ASSIGNMENT NUMBER: A - 02

TITLE : Pass II of a two pass assembler.

PROBLEM STATEMENT : Implement Pass-II of two pass assembler for pseudo-machine in Java using object oriented features. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.

OBJECTIVES:

- Synthesis of the object code.
- Understand the use of data structures required in the design of assembler.

OUTCOMES:

The students will be able to

- Parse and tokenize the intermediate code file
- Perform the LC processing
- Generate the target code file
- Demonstrate the use of symbol table, literal table, pooltab

THEORY:

Assembler is a program which converts assembly language instructions into machine language form. A two pass assembler takes two scans of source code to produce the machine code from assembly language program.

Assembly process consists of following activities:

- Convert mnemonics to their machine language opcode equivalents
- Convert symbolic (i.e. variables, jump labels) operands to their machine addresses
- Translate data constants into internal machine representations
- Output the object program and provide other information required for linker and loader

Pass II Tasks:

- Assemble instructios(generate opcode and look up addresses)
- Generate data values defined by BYTE, WORD
- Perform processing of assembler directives(not done in pass I)
- Write the object program and the assembly listing

Description using set THEORY:

Let ‘S’ be set which represents a system

S={I,O,T,D,Succ,Fail}

where,

I=Input
O=Output
T=Type

D=Data Structure

I={Ic,St,Lt}

where,

Ic=Intermediate Code File
St=Symbol table
Lt=Literal table

St={N,A}

where,

N=Name Of Symbol
A=Address Of Symbol

Lt={N,A}

where,

N=Name Of Literal
A=Address Of Literal

O={o}

Where,

o=Output File(M/C Code File)
T=Varient II

D={Ar,Fl,Sr}

Where,

Ar=Array
Fl=File
Sr=Structure

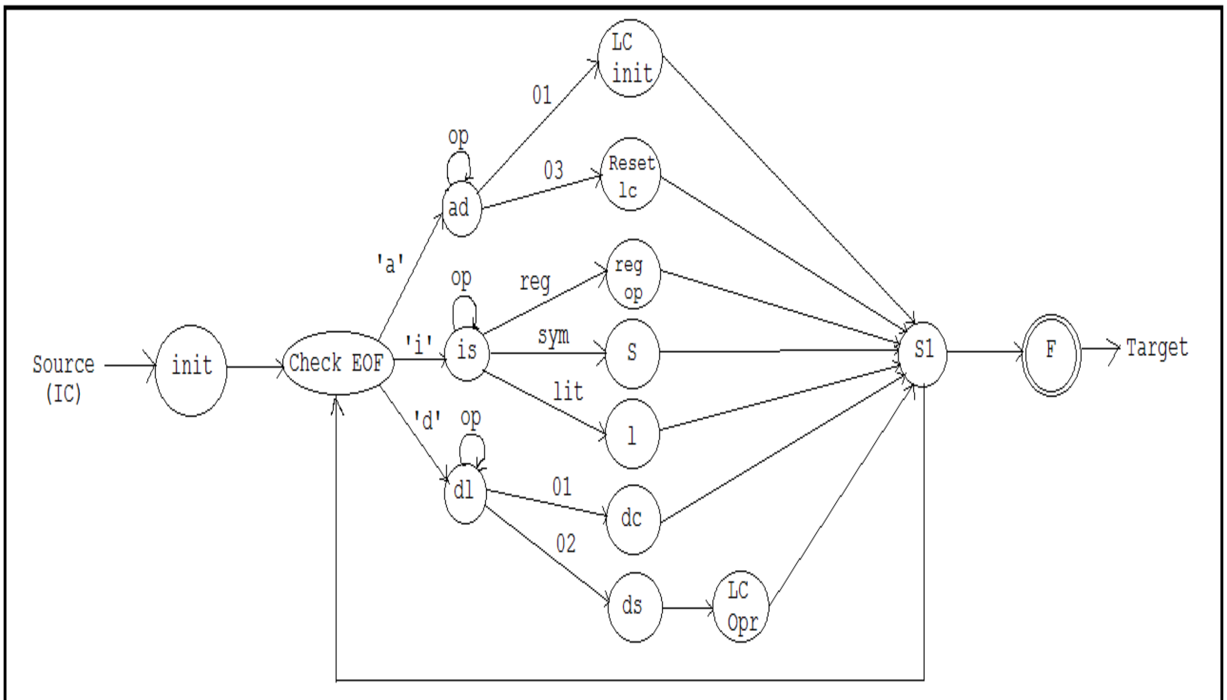
Success Succ={x |x is set of all cases that are handled in program}

Succ=
{
Undefined Symbol
Undefined mnemonic,
}

Failures Fail={x |x is set of all cases that are not handled in program}

Fail=
{Multiple statements in a line}

Turing machine/state diagram:



State Diagram
(Pass 2 of 2 Pass Assembler)

Steps to do /algorithm:

- Read the intermediate code file generated in pass I.
- Search symbol and literal tables to use in machine code generation.
- Generate the machine code.

CONCLUSION: We have successfully performed pass-II of two pass assembler.