**TWO PASS ASSEMBLER**

**Computer Organization**

**CSE112**

Steps followed while converting assembly code to machine code:

**1) TAKING INPUT**

used to open the assembly language code present in file: input.txt . Next, each line is read line by line and stored in form of a list. Further, each line is checked to see if it is a comment or not. If it is not a comment, it is stored in a list. Then, each line which is appended is split and the contents are stored as a nested list. Instruction length counter is incremented at each step to find final address.

**2) THE FIRST PASS**

A pointer is traversed in the assembly code to look through all possible cases.

**If the number of words in a line is 1,**

then it can only be CLA or STP opcodes. In any other case, error will be reported.

**If the number of words in a line is 2, the possible cases are as follows:**

**Opcode and Operand:**

we checked if the opcode was valid or not. If it is valid if it is appended to the opcode table. It is invalid if it is CLA or STP, hence error is reported. If the opcodes is one of SAC, INP, STP, BRZ, BRN, BRP and the operand is a literal, then error is reported. Else, the instruction is correct.

**Label and Opcode:**

If the opcode is a valid opcode, it is appended to the opcode table. Next, the label is checked if it is correct or not. If the label with the same name is declared earlier in the program, error is reported. If the name of the label is same as an opcode or a symbol, error is reported. If there is no error, then it is a valid label and hence added to the symbol table. The opcode must be either CLA or STP, Else, error is reported.

If it is a case different from those mentioned, invalid instruction error is reported.

**If the number of words in a line is 3, the following cases are possible:**

**Variable assignment instruction:**

It is checked if the variable is declared more than once. If the name of variable is same as an opcode or a label already present in the opcode. If it is any one of the cases, error is reported. Else, the variable is added to the symbol table along with its value.

**Label+Opcode+Operand:**

It is checked if the opcode is valid or not. If valid, it is appended to the opcode table, else error is reported. Checks are made to see if the label if correct or not. If correct, it is added to the symbol table else error is reported. If the opcode is either CLA or STP, error is reported. If opcodes like BRN, BRP, BRZ, INP, STP, SAC are present, checks are made if particular addresses are provided or not. If no error is reported, instruction is correct.

**Opcode+Operand+Operand:**

Error is reported as opcode is provided with more than required opcodes.

If the error is is none from the ones mentioned or the number of words in a line is **more than 3**, **invalid instruction error** is reported or else the instruction is correct.

**Next, repeated elements are removed from all the tables and the new tables are printed.**

**3) THE SECOND PASS**

This function is called only when there are no errors in the assembly code.

If there are errors in the assembly code, just the tables will be created and the assembler would terminate.

Another pointer is made to traverse the code for the second time, now to convert the assembly code to corresponding machine code. The converted code, the opcode table, the symbol table and literal table is written to a text file : Output.txt and the errors are reported on the console.

**4) ERROR HANDLING:**

The following errors have been dealt with in our assembler:

**1)**Invalid Opcode

**2)**Variable undeclared but used

**3)**Variable declared more than once

**4)**Opcode provided with more or less than required opcodes

**5)**Name of symbol is same as an opcode or a label which is already used in the program

**6)**Name of variable is same as that of can opcode or a label which is already used in the program

7) Opcode using more operands than needed

**8)**Any other error is named as invalid instruction error

**AMAN PRIYADARSHI** 2019294

**MEDHAVI SABHERWAL** 2019371