|  |  |
| --- | --- |
| **Ex.No. : 4** | **SINGLE PASS ASSEMBLER FOR SIC** |
| **Date :** 03/02/2022 |

**Aim :**

To simulate the concept of single pass assembler for simplified instructional computer.

**Algorithm :**

1. Open and Read the input file

2. If the input line has the opcode START do the following

2.1 Find if there is any operand field after START initialize the LC to the operand

Value.

2.2 Otherwise if there is no value in the operand field then LC is set to 0.

3. Write the input line to the intermediate file.

4. Do the following steps until the opcode is END.

4.1 Check the Symbol table, if the symbol isnot available then enter that symbol

into the SYMTAB, along with the memory address in which it is

stored.Otherwise, the error message will be displayed.

4.2 If there is a opcode

4.2.1 1 If opcode is present in the OPTAB, then increment the LC by 3 and

Start writing the location counter, opcode and operand fields of the

corresponding statement to the output file, along with the object code.

4.2.2 If opcode isWORD then increment LC by 3

4.2.3 If opcode is BYTE then increment LC by 1

4.2.4 If opcode is RESW then increment LC by the integer equivalentof the

operand value \* 3

4.2.5 If opcode is RESB then increment LC by the integer equivalent of the

operand value

4.2.6 If there is no symbol/label in the operand field, then the

Operand address is assigned as zero and it is assembled with the object

code of the instruction

4.2.7 Write the processed lines in the intermediate file along with their along

with the location counters.

5. To find the length of the program, Subtract the starting address of the program from

the LC.

6. Close all the files.

**Program :**

i=open('D:\Hello Rajat\Comp\input\_4.txt','r').readlines()

op=open('D:\Hello Rajat\Comp\optab\_4.txt','r').readlines()

optab={};

l=[];

m=[];

n=[];

a=[];

opcode=[];

object\_code=[]

for line in op:

op=line.split()

optab.update({op[0]: op[1]})

for line in i:

if 'START' in line:

start=line.split()[2]

x=int(start, 16)

a.append('')

else:

a.append(str(hex(x)[2:]))

x+=3

for line in i:

line=line.split()

if len(line)==3:

l.append(line[0])

m.append(line[1])

n.append(line[2])

elif len(line)==2:

l.append('')

m.append(line[0])

n.append(line[1])

else:

m.append(line[0])

for i in m:

if i == 'START':

continue

if i in optab:

x=str(optab[i])

y=str(a[l.index(n[m.index(i)])])

object\_code.append('^'+str(x+y))

ob=''.join(object\_code)

leng=len(object\_code)

x=str(hex(leng\*3)[1:])

if 'x' in x:

x=x.replace('x', '0')

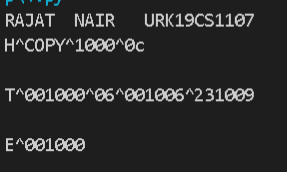
print("ASHLEY JOHN ARAKAL URK19CS1064")

print('H^'+l[0]+'^'+a[1]+'^'+a[len(a)-1][2:]+'\n')

print('T^00'+a[1]+'^'+x+ob+'\n')

print('E^'+'00'+a[1]+'\n')

**Output :**

****

**Video Link :**

**Result :**

The program to simulate the concept of single pass assembler for simplified instructional computer has been successfully executed and the output is verified.