USN - 1KS18CS097

SIGN - Sourabh

Name: Sourabh Santosh Kamble

USN: 1KS18CS097

Semester: VI

Section: ISE 'B'

Subject: System Software and Compilers

Subject Code: 18CS61

Signature: Sourabh

INTERNAL ASSESMENT - 1

PART - A

1.a. **ALGORITHM FOR PASS-1**

```
Begin
  Read the first line
  if OPCODE = 'START' then
  Save #[operand] to starting address
  Initialize starting address to LOCCTR
  write the line to intermediate file
  Read the next line
  End
  else
  Initialize LOCCTR to 0
  While OPCODE != 'END' do
  Begin
    if this is not a comment line then
    Begin
    if a symbol in the LABEL field then
    Begin
    Search SYMTAB for LABEL
    if found then
       Set error flag
  else
```

Search OPTAB for OPCODE
if found then
    Add 3 to LOCCTR
else if OPCODE = 'BYTE' then
~~Add~~
Begin
find length of constant in bytes.
Add length to LOCCTR
End
else if OPCODE = 'WORD' then
~~Add 3 * #~~
Add 3 to LOCCTR
else if OPCODE = 'RESB' then
Add #[operand] to LOCCTR
else if OPCODE = 'RESW' then
Add 3 * #[operand] to LOCCTR
else
set error flag
End.
Write line to intermediate file
Read next input line
End
~~write last input line~~
Write line to intermediate file
~~Read next input~~
Save {LOCCTR - Starting address} to program length
End.

USN: 1KS18CS097

SIGN: Aswath

1.b. SIC/XE MACHINE ARCHITECTURE:

SIC/XE stands for Simple Instructional Computer (SIC) / Extra equipment or extra expensive (XE).

i) MEMORY: The total memory used or opted by SIC/XE is about 2^{20} bytes. Number of address lines are 20.

ii) REGISTERS:

REGISTER	NUMBER/ CODE	FUNCTION
A	0	Accumulator
X	1	Index Register
L	2	Linkage Register
B	4	Base register } General purpose
S	5	Register
T	6	
F	7	Floating point integer
PC	8	Program Counter
SW	9	status word.

iii) DATA FORMAT

a) Integers - 3 bytes

b) character - 1 byte

c) Floating point - 6 byte

USN. 1KS18CS097
SIGN: Souradip

iv> INSTRUCTION FORMAT

a> FORMAT 1 :

8 → 8 bits

opcode

b> FORMAT-2 :

8	4	4
opcode	R1	R2

→ 16 bits

c> FORMAT-3 :

6	1	1	1	1	1	1	12
opcode	n	i	x	b	p	e	Disp/Data

→ 24 bits

d> FORMAT-4 :

6	1	1	1	1	1	1	20
opcode	n	i	x	b	p	e	Disp/Data

→ 32 bits

Here
- n → Indirect bit
- i → Immediate bit
- x → Index bit
- b → base bit
- p → PC relative bit
- e → extended bit

when,
- n=0, i=0 it represents simple SIC instruction
- n=0, i=1 it represents immediate addressing
- n=1, i=0 it represents indirect addressing
- n=1, i=1 it represents direct addressing.
- x=0 it represent direct addressing.
- x=1 it represent indexed addressing
- b=0, P=0 neither base nor pc relative addressing

b = 1, p = 0 it represents base relative addressing
b = 0, p = 0 it represent PC relative addressing
e = 0 it represents format 3 instruction
e = 1 it represents format 4 instruction

v) ADDRESING MODES:
a) Base relative
b) Program counter relative.
c) Direct
d) Index
e) Immediate
f) Indirect
g) Extended addressing modes.

vi) INSTRUCTION SET:
a) Load and store instructions: LDB, LDA, STB, STA.
b) Arithematic instructions: ADD, SUB, MUL.
c) Register arithematic instructions: ADDR, SUBR, MULR.

vii) INPUT AND OUTPUT:
a) Test Device (TD): It is to Test whether the device is ready or not. The condition code (CC) is used to Test, if CC = less then device is ready or else device is not ready.
b) Read Data (RD): Reads the byte from device and store it in register A.

c) write data: Writes byte from register A to device.

d) SIO - Starts IIO operation.

e) HIO - Halts IIO operation.

f) TIO - Test IIO operations.

1.c. SIC/XE

LOCATION	LENGTH	LABEL	MNEMONIC	OPERAND	OBJECT CODE
		~~START~~ SUM	START	0	
0000	3	FIRST	LDX	#0	050000
0003	3		LDA	#0	010000
0006	4		+LDB	#TABLE2	69101790
			BASE	TABLE2	
000A	3	LOOP	ADD	TABLE,X	1BA013
000D	3		ADD	TABLE2,X	1BC000
0010	3		TIX	COUNT	2F200A
0013	3		JLT	LOOP	3B2FF4
0016	4		+STA	TOTAL	0F102F00
001A	3		RSUB		4F0000
001D	1x3 = 3	COUNT	RESW	1	
0020	1770	TABLE	RESW	2000	
1790	1770	TABLE2	RESW	2000	
2F00	1x3 = 3	TOTAL	RESW	1	
2F03			END	FIRST	

USN: 1RS18CS041

SIGN: Nourabh

OBJECT PROGRAM:

H^ SUM ^000000 ^ 002F03

T^ 000000^ 1D^ 050000^ 010000^ 69101790^

1BC000^ 2F200A^ 3B2FF4^ 0F102F00^4F0000

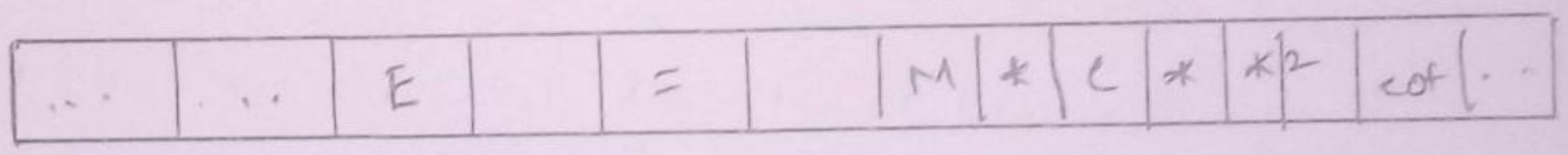
M^ 000007^ 05

M^ 000017^05

E^ 000000

PART-B

4.a. BUFFER PAIRS:



i) Buffering technique is used to reduce the amount of overhead required to process single buffering when the program has to be loaded and processed.

WORKING:

a) The figure above is a buffer (array) of ~~length~~ one, n is number of characters on a disk. It is divided into two n-character halves.

b) Two pointers are used here lexemeBegin and forward.

c) Initially both pointers point to the first character of lexeme to be found.

d) Now the foward pointer increments and scan until a match for pattern is obtained.

e) Once next lexeme is determined, processed and then both the pointers are set to the character immediately.

f) The string within lexemeBegin and forward is the lexeme to be processed.

LOOK-AHEAD CODE:

```
switch (*forward ++) {
  case eof:
     if (forward is at end of first buffer) {
        reload second buffer;
        forward = beginning of second.buffer;
     }
     else if ( forward is at end of second buffer) {
        reload first buffer;
        forward = beginning of first buffer;
     }
     else
        terminate lexical analysis;
     break;
}
```

USN 1KS18CS097
SIGN: Soumabh

4.a. DRAWBACKS OF BUFFER PAIRS:

i) Look ahead is limited. and it is impossible to recognize tokens.

ii) If distance between two points is more the buffer length has to be increased.

iii) To move forward to next lexeme we have to confirm whether we have reached the end of one of buffers.

USN: 1KS18CS097
SIGN: [signature]

4.c. TOKEN:

a> A token is a terminal symbol in grammar for the source language.

b> A token is a pair of token name and an optional attribute value.

c> The token name is abstract symbol representing a lexical unit like keyword or identifier.

PATTERN:

a> A pattern is a rule describing set of lexemes that can represent a token in the source programs.

b> It is a description of form that lexemes of a token may take.

LEXEME:

a> It is a sequence of characters in source program that matches the pattern with a token.

b> It is identified by lexical analyzer as an instance of token.

NO. OF PAGES - 10
QUESTIONS ATTEMPTED:
1.a, 1.b, 1.c
4.a, 4.b.