



Roll No

## National Institute of Technology Goa

Programme Name: **B.Tech**

End Semester Examinations, May-2019

B. Tech  
CSE.  
4<sup>th</sup> Sem

Course Name: Systems Programming

Date: **02/05/2019**

Duration: **3 Hours**

Course Code: **CS251**

Time: **2:00 PM-5:00 PM**

Max. Marks: **100**

### ANSWER ALL QUESTIONS

**NOTE:** Opcodes for a selected set of instructions are given in Page 4 of this paper. Same codes must be used.

1. Explain the various instruction formats supported in SIC/XE giving 2 examples for each. [10 marks]
2.
  - a. Consider the following instruction  
LDA #3  
Suppose with a minor extension to literal notation, we could write the same instruction as  
LDA =W' 3'  
specifying as the literal operand a word with the value 3. Would this be a good idea? Discuss
  - b. What is the advantage of writing  
MAXLEN EQU BUFEND-BUFFER  
Instead of  
MAXLEN EQU 4096  
Explain
  - c. Write short notes on one-pass assembler and a multi-pass assembler. [3+3+4=10 marks]
3. Generate the object program for the SIC/XE program given in Pages 5 & 6 of this question paper. Write the location counter as well as object code for the instructions on the Pages 5 & 6 and attach the same to the answer booklet. Write the final object program(s) in the answer booklet. [20 marks]
4.
  - a. What are the data structures used in a linking loader? Explain
  - b. Compare and contrast a linking loader and a linkage editor.
  - c. Write a short note on dynamic linking [3+4+3=10 marks]

5. Consider the three (separately assembled) programs (PROGA, PROGB and PROGC) given in Pages 7 & 8 of this question paper each of which consists of a single control section. Each program contains a list of items (LISTA, LISTB and LISTC) and the ends of these lists are marked by the labels ENDA, ENDB and ENDC. Each program consists of same set of references to these symbols. Note that all portions of the programs which are not involved in relocation and linking are omitted. Write the object programs that are generated at the end of the assembly process. Assume PROGA to be the main program. Write the object code for the instructions and other lines wherever applicable on the Pages 7 & 8 and attach the same to the final answer booklet. Write the final object programs in the answer script.

6. The object program given below need to be loaded into memory. Show the program in memory after loading considering the starting address as 2000.

```

HPROG 00000000107A
T00000001E0001400334810390000362800303000154810613C000300002A0C003900002D
T00001E15E000C00364810610800334C0000454F46000003000000
T0010391E000C040030000030E0105D30103FD8105D2800303010575480392C105E38103F
T0010570A8001000364C0000F1001000
T00106119E00040030E01079301064508039DC10792C00363810644C000005
E0000000

```

7. Consider a macro defined as follows:

```

RDBUFF      MACRO          &INDEV, &BUFADR, &RECLTH, &EOR, &MAXLTH
&EORCT      SET            %NITEMS(&EOR)
             CLEAR         X
             CLEAR         A
             IF             (&MAXLTH EQ `')
             +LDT           #4096
             ELSE
             +LDT           #&MAXLTH
             ENDIF
$LOOP        TD            =X`&INDEV`
             JEQ            $LOOP
             RD             =X`&INDEV`
&CTR        SET            1
             WHILE         (&CTR LE &EORCT)
             COMP          =X`0000&EOR[&CTR]`
             JEQ            $EXIT
&CTR        SET            &CTR+1
             ENDW
             STCH          &BUFADR,X
             TIXR          T
             JLT            $LOOP
$EXIT       STX            &RECLTH
             MEND

```

Expand the following macro invocation statements for the macro definition given above

- ```
a. RDBUFF      F1, BUF, LENGTH, ,
b. RDBUFF      0E, BUFF, LENG, (00), 100
c. RDBUFF      F2, BUFFER, LEN, (00, 05, 03, 04)
```

[4+4+4=12 marks]

8.

- a. Briefly explain the Define and Expand procedure of a 2-pass macro processor
- b. What is the limitation of a standard 2-pass macro processor when a macro is defined within another macro? How it is addressed?
- c. Can a macro invoke another macro? How is it handled in a macro processor? Explain

[3+3+2=8 marks]

\*\*\*\*\*ALL THE BEST\*\*\*\*\*



### Opcode for a few selected instructions

|              |    |
|--------------|----|
| ADD m        | 18 |
| ADDF m       | 58 |
| ADDR R1, R2  | 90 |
| CLEAR R1     | B4 |
| COMP m       | 28 |
| COMPF m      | 88 |
| COMPR R1, R2 | A0 |
| DIV m        | 24 |
| DIVR R1, R2  | 9C |
| J m          | 3C |
| JEQ m        | 30 |
| JGT m        | 34 |
| JLT m        | 38 |
| JSUB m       | 48 |
| LDA m        | 00 |
| LDB m        | 68 |
| LDF m        | 70 |
| LDL m        | 08 |

|             |    |
|-------------|----|
| LDS m       | 6C |
| LDT m       | 74 |
| LDX m       | 04 |
| MUL m       | 20 |
| MULR R1, R2 | 98 |
| RD m        | D8 |
| RSUB        | 4C |
| STA m       | 0C |
| STB m       | 78 |
| STF m       | 80 |
| STL m       | 14 |
| STS m       | 7C |
| STX m       | 10 |
| SUB m       | 1C |
| TD m        | E0 |
| TIX m       | 2C |
| TIXR R1     | B8 |
| WD m        |    |

### Register numbers

|    |   |
|----|---|
| A  | 0 |
| X  | 1 |
| L  | 2 |
| B  | 3 |
| S  | 4 |
| T  | 5 |
| F  | 6 |
| PC | 8 |
| SW | 9 |

# Assembly language (SIC/XE) program for Question (3)

|        |        |                              |
|--------|--------|------------------------------|
| ARRMP  | START  | 0                            |
|        | EXTDEF | ALPHA, MAXLEN                |
|        | EXTREF | SUMARR, FMAXA, FMINA, MKZERA |
|        | +JSUB  | SUMARR                       |
|        | +JSUB  | FMAXA                        |
|        | +JSUB  | FMINA                        |
|        | +JSUB  | MKZERA                       |
|        | RSUB   |                              |
| ALPHA  | RESW   | 100                          |
| AEND   | EQU    | *                            |
| MAXLEN | EQU    | AEND-ALPHA                   |
|        |        |                              |
| SUMARR | CSECT  |                              |
|        | EXTREF | ALPHA, MAXLEN                |
|        | CLEAR  | X                            |
|        | CLEAR  | A                            |
|        | LDS    | THREE                        |
|        | +LDT   | MAXLEN                       |
|        | LDA    | #0                           |
|        | STA    | SUM                          |
| ADDLP  | +ADD   | ALPHA, X                     |
|        | ADDR   | S, X                         |
|        | COMPR  | X, T                         |
|        | JLT    | ADDLP                        |
|        | STA    | SUM                          |
|        | RSUB   |                              |
| THREE  | WORD   | 3                            |
| SUM    | RESW   | 1                            |
|        |        |                              |
| FMAXA  | CSECT  |                              |
|        | EXTREF | ALPHA, MAXLEN                |
|        | CLEAR  | X                            |
|        | CLEAR  | A                            |
|        | LDS    | THREE                        |
|        | STX    | MAXVAL                       |
|        | +LDT   | MAXLEN                       |
| MAXLP  | +LDA   | ALPHA, X                     |
|        | COMP   | MAXVAL                       |
|        | JLT    | INCR                         |
|        | STA    | MAXVAL                       |
| INCR   | ADDR   | S, X                         |
|        | COMPR  | X, T                         |
|        | JLT    | MAXLP                        |
|        | RSUB   |                              |
| THREE  | WORD   | 3                            |
| MAXVAL | RESW   | 1                            |

|        |        |               |
|--------|--------|---------------|
| FMINA  | CSECT  |               |
|        | EXTREF | ALPHA, MAXLEN |
|        | CLEAR  | X             |
|        | CLEAR  | A             |
|        | LDS    | THREE         |
|        | STX    | MINVAL        |
|        | +LDT   | MAXLEN        |
| MINLP  | +LDA   | ALPHA, X      |
|        | COMP   | MINVAL        |
|        | JGT    | INCR          |
|        | STA    | MINVAL        |
| INCR   | ADDR   | S, X          |
|        | COMPR  | X, T          |
|        | JLT    | MINLP         |
|        | RSUB   |               |
| THREE  | WORD   | 3             |
| MINVAL | RESW   | 1             |

|        |        |               |
|--------|--------|---------------|
| MKZERA | CSECT  |               |
|        | EXTREF | ALPHA, MAXLEN |
|        | CLEAR  | X             |
|        | CLEAR  | A             |
|        | LDS    | THREE         |
|        | +LDT   | MAXLEN        |
| ZERLP  | STA    | ALPHA, X      |
|        | ADDR   | S, X          |
|        | COMPR  | X, T          |
|        | JLT    | ZERLP         |
|        | RSUB   |               |
| THREE  | WORD   | 3             |

# Assembly language (SIC/XE) program for Question (5)

```

0000  PROGA  START 0
          EXTDEF LISTA, ENDA
          EXTREF LISTB, ENDB, LISTC, ENDC
          .
          .
          .
0010  REF1  +LDA  LISTB-LISTA
0014  REF2  +LDT  LISTB+7
0018  REF3  LDX   #ENDA-LISTA
001B  REF4  LDS   LISTA
          .
          .
          .
0030  LISTA EQU   *
          .
          .
          .
0047  ENDA  EQU   *
0047  REF5  WORD  ENDA-LISTA+LISTC
004A  REF6  WORD  ENDC-LISTC+LISTA-1
004D  REF7  WORD  LISTB-LISTA
0050  REF8  WORD  LISTB+LISTC
0053  REF9  WORD  LISTA-LISTB-ENDA+ENDB
          END    REF1

```

```

0000  PROGB  START 0
          EXTDEF LISTB, ENDB
          EXTREF LISTA, ENDA, LISTC, ENDC
          .
          .
          .
0023  REF1  +LDA  LISTB-LISTA
0027  REF2  LDT   LISTB+7
002A  REF3  +LDX  #ENDA-LISTA
002E  REF4  +LDS  LISTA
          .
          .
          .
0038  LISTB EQU   *
          .
          .
          .
0044  ENDB  EQU   *
0044  REF5  WORD  ENDA-LISTA+LISTC
0047  REF6  WORD  ENDC-LISTC+LISTA-1
004A  REF7  WORD  LISTB-LISTA
004D  REF8  WORD  LISTB+LISTC
0050  REF9  WORD  LISTA-LISTB-ENDA+ENDB
          END

```

```

0000  PROGC  START 0
        EXTDEF LISTC, ENDC
        EXTREF LISTA, ENDA, LISTB, ENDB
        .
        .
0028  REF1  +LDA  LISTB-LISTA
002C  REF2  +LDT  LISTB+7
0030  REF3  +LDX  #ENDA-LISTA
0034  REF4  +LDS  LISTA
        .
        .
004F  LISTC EQU  *
        .
005D  ENDC  EQU  *
005D  REF5  WORD  ENDA-LISTA+LISTC
0060  REF6  WORD  ENDC-LISTC+LISTA-1
0063  REF7  WORD  LISTB-LISTA
0066  REF8  WORD  LISTB+LISTC
0069  REF9  WORD  LISTA-LISTB-ENDA+ENDB
        END

```