Roll no									

National Institute of Technology Goa

Programme Name: B.Tech
Mid Semester Examinations

Course Name: System Programming **Course Code: CS251**
Date:08/03/2022 **Time:09:30 AM-11:00 AM**
Duration: 1 hour 30 minutes **Max. Marks:** 50

ANSWER ALL QUESTIONS

NOTE:

- **The opcode table is attached for your reference.**

1. Write an assembly language program for simplified instructional computer (SIC) to multiply a number with another number using successive addition. Assume that the multiplier and the multiplicand are available in memory. Let the product be stored in memory and also sent to an output device numbered '05'. Assume the starting address to be 3000.
   [*NOTE: This question is based on the individual thought process and creativity.*]

   (15 marks)

2. Assign the address and generate an object code for each of the instruction and constant declaration in the assembly language program written in Question 1. Write the object program also. Clearly state the outputs after the execution of pass 1 and pass 2 of the assembler.

   (15 marks)

3. Explain the process of object code generation for the following instructions

   a. 1059 RSUB
   b. 1051 TIXR T
   c. 104E STCH BUFFER,X
   d. 103C +LDT #4096
   e. 002D EOF BYTE C`EOF`

   in the following intermediate code generated after the execution of pass 1 of the assembler for SIC/XE

```
0000    COPY        START   0
0000    FIRST       STL     RETADR
0003                LDB     #LENGTH
                    BASE    LENGTH
0006    CLOOP       +JSUB   RDREC
000A                LDA     LENGTH
000D                COMP    #0
0010                JEQ     ENGFIL
0013                +JSUB   WRREC
0017                J       CLOOP
001A    ENDFIL      LDA     EOF
001D                STA     BUFFER
0020                LDA     #3
0023                STA     LENGTH
```

```
0026                +JSUB       WRREC
002A                J           @RETADR
002D    EOF         BYTE        C`EOF`
0030    RETADR      RESW        1
0033    LENGTH      RESW        1
0036    BUFFER      RESB        4096
        .
        .
        .
1036    RDREC       CLEAR       X
1038                CLEAR       A
103A                CLEAR       S
103C                +LDT        #4096
1040    RLOOP       TD          INPUT
1043                JEQ         RLOOP
1046                RD          INPUT
1049                COMPR       A, S
104B                JEQ         EXIT
104E                STCH        BUFFER,X
1051                TIXR        T
1053                JLT         RLOOP
1056    EXIT        STX         LENGTH
1059                RSUB
105C    INPUT       BYTE        X`F1`
        .
        .
        .
105D    WRREC       CLEAR       X
            .
```

. **--------10 marks**

4.

a. Compare and contrast SIC and SIC/XE
b. Which among absolute addressing and relative addressing is better? Justify your answer.
c. Suppose that we make the program written for the Question 1 a relocatable program. To support relocation, nearly every instruction in the object program would need to have its operand address modified at load time. This would mean a large number of Modification records. How could we include the required relocation information without the large increase in the object program size?

(3+3+4=10 marks)

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

SIC/XE Instruction Set

Op codes in blue are SIC/XE only instructions
Op codes in red are not implemented by the simulator

Notes: P=privileged, C=CC set (<,=,>), F=floating point
See Appendix A of *System Software* by Beck for information on instruction formats and addressing modes.

Mnemonic	Format	Opcode	Effect	Notes
ADD m	3/4	18	A ← (A) + (m..m+2)	
ADDF m	3/4	58	F ← (F) + (m..m+5)	F
ADDR r1,r2	2	90	r2 ← (r2) + (r1)	
AND m	3/4	40	A ← (A) & (m..m+2)	
CLEAR r1	2	4	r1 ← 0	
COMP m	3/4	28	A : (m..m+2)	C
COMPF m	3/4	88	F : (m..m+5)	CF
COMPR r1,r2	2	A0	(r1) : (r2)	C
DIV m	3/4	24	A : (A) / (m..m+2)	
DIVF m	3/4	64	F : (F) / (m..m+5)	F
DIVR r1,r2	2	9C	(r2) ← (r2) / (r1)	
FIX	1	C4	A ← (F) [convert to integer]	
FLOAT	1	C0	F ← (A) [convert to floating]	F
HIO	1	F4	Halt I/O channel number (A)	P
J m	3/4	3C	PC ← m	
JEQ m	3/4	30	PC ← m if CC set to =	
JGT m	3/4	34	PC ← m if CC set to >	
JLT m	3/4	38	PC ← m if CC set to <	
JSUB m	3/4	48	L ← (PC); PC ← m<	
LDA m	3/4	00	A ← (m..m+2)	
LDB m	3/4	68	B ← (m..m+2)	
LDCH m	3/4	50	A [rightmost byte] ← (m)	
LDF m	3/4	70	F ← (m..m+5)	F
LDL m	3/4	08	L ← (m..m+2)	
LDS m	3/4	6C	S ← (m..m+2)	
LDT m	3/4	74	T ← (m..m+2)	
LDX m	3/4	04	X ← (m..m+2)	
LPS m	3/4	D0	Load processor status from information beginning at address m (see Section 6.2.1) 6.2.1)	P
MUL m	3/4	20	A ← (A) * (m..m+2)	
MULF m	3/4	60	F ← (F) * (m..m+5)	
MULR r1,r2	2	98	r2 ← (r2) * (r1)	
NORM	1	C8	F ← (F) [normalized]	F
OR m	3/4	44	A ← (A) | (m..m+2)	
RD m	3/4	D8	A [rightmost byte] ← data from device specified by (m)	P
RMO r1,r2	2	AC	r2 ← (r1)	
RSUB	3/4	4C	PC ← (L)	
SHIFTL r1,n	2	A4	r1 ← (r1); left circular shift n bits. [*for assembled instruction, r2 is n-1*]	
SHIFTR r1,n	2	A8	r1 ← (r1); right shift n bits with vacated bit positions set equal to leftmost bit of (r1) [*for assembled instruction, r2 is n-1*]	
SIO	1	F0	Start I/O channel number (A); address of channel program is given by (S)	P
SSK m	3/4	EC	Protection key for address m ← (A) (see Section 6.2.4)	P
STA m	3/4	0C	m..m+2 ← (A)	
STB m	3/4	78	m..m+2 ← (B)	
STCH m	3/4	54	m ← (A) [rightmost byte]	
STF m	3/4	80	m..m+5 ← (F)	F
STI m	3/4	D4	Interval timer value ← (m..m+2) (see Section 6.2.1)	P

STL m	3/4	14	m..m+2 ← (L)	
STS m	3/4	7C	m..m+2 ← (S)	
STSW m	3/4	E8	m..m+2 ← (SW)	P
STT m	3/4	84	m..m+2 ← (T)	
STX m	3/4	10	m..m+2 ← (X)	
SUB m	3/4	1C	A ← (A) - (m..m+2)	
SUBF m	3/4	5C	F ← (F) - (m..m+5)	F
SUBR r1,r2	2	94	r2 ← (r2) - (r1)	
SVC n	2	B0	Generate SVC interrupt. {*for assembled instruction, r1 is n*]	
TD m	3/4	E0	Test device specified by (m)	PC
TIO	1	F8	Test I/O channel number (A)	PC
TIX m	3/4	2C	X ← (X) + 1; (X) : (m..m+2)	C
TIXR r1	2	B8	X ← (X) + 1; (X) : (r1)	C
WD m	3/4	DC	Device specified by (m) ← (A) [rightmost byte to device specified by m]	P