

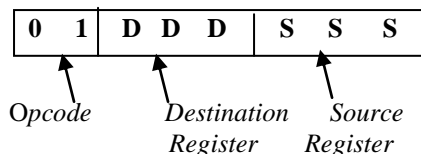
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

SYSTEMS PROGRAMMING CSC 2209

TEST 1

1. **Use examples to explain the difference between Application Software and Systems Software.**
Application programs are written by the user for the purpose of solving particular problems using a computer as a tool e.g. application packages while System Software of a computer consists of a collection of programs whose purpose is to make more effective use of the computer. They control the operation of the machine and carry out the most basic functions the computer performs. They control the way in which the computer receives input, produces output, manages and stores data, carries out or executes instructions of other programs etc.
2. Explain the following:
 - (i) **Operation Code**
The portion of the machine language instruction that specifies what the instruction does.
 - (ii) **Operand**
An address or piece of data that is required by the instruction to complete its execution
 - (iii) **Source Operand**
A location where data is taken from
3. (a) **What are addressing modes?**
They are the methods used to locate and fetch an operand from an internal CPU register or from a memory location
- (b) **Use examples to explain any two addressing modes.**
4. Assume the following Intel 8085 Instruction Formats:

Register to register transfer



Load accumulator from memory

0	0	1	1	1	0	1	0
Low order Address							
High Order Address							

Add contents of a register to the accumulator

1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

Store accumulator contents to memory

0	0	1	1	0	0	1	0
Low order Address							
High Order Address							

Write an INTEL 8085 Machine Language program that picks two numbers from memory locations 1234H and 5678H respectively, adds them together and stores the answer in memory location ABCDH and another copy in register E. Give your answer in Hexadecimal.

3A	3A	32
34	78	CD
12	56	AB
47	80	5F

4. Explain what each of the following directives means and in each case state the number of bytes that have been reserved when the directive is given.
 - (i) **Num1: DS 20;**
Reserve 20 bytes of memory and label the first byte NUM1
 - (ii) **Num2: RESW 20**
Reserve 60 bytes of memory and label the first byte NUM2
 - (iii) **Num3: RESB 20**
Reserve 20 bytes of memory to store characters and label the first byte NUM3
5. (a) **List and give the uses of the registers used by the Simple SIC machine.**

Mnemonic	Use
A	<i>Accumulator; used for arithmetic Operations</i>
X	<i>Index register; used for addressing</i>
L	<i>Linkage register; the Jump to subroutine (JSUB)instruction stores the return address in this register.</i>
PC	<i>Program Counter; Contains the address of the next instruction to be fetched for execution.</i>
SW	<i>Status word; contains the condition codes</i>

(b) **How would the following numbers be stored by the SIC/XE machine?**

- (i) **+20**
 $20_{10} = 10100_2 = 0000000000000000000010100 = \mathbf{000014}_{16}$
- (ii) **-20** = \mathbf{FFFFFC}_{16}
- (iii) **-20.25** = $10100.01 = 0.1010001 \times 2^5$; Offset = 1024
Exponent = $1024 + 5 = 1029 = 10000000101$
1100 0000 0101 1010 0010 00000000 = **C05A20000000**

6. Assume that the **SIC/XE** registers B, PC and X contain numbers 500_{16} , 600_{16} and 700_{16} respectively and that some memory locations are loaded with contents as shown here:

500	300	200	1000
1000	200	1200	4000

Each of the following **SIC/XE** instructions loads some data into the accumulator. Calculate the target address in each case and give what is loaded into the accumulator.

(i) **03C600**

	Op	n i x b p e	disp/address
03C600	000000	1 1 1 1 0 0	0110 0000 0000
TA = (B) + (X) + Disp = 500 + 700 + 600 =			1200
Accumulator is loaded with			= 200

(ii) **02C600**

	Op	n i x b p e	disp/address
02C600	000000	1 0 1 1 0 0	0110 0000 0000
TA = (B) + (X) + Disp = 500 + 700 + 600 =			1200
Accumulator is loaded with			= 300

(iii) **02104000**

	Op	n i x b p e	address
02104000	000000	1 0 0 0 0 1	0000 0100 0000 0000 0000
TA =			4000 ; Accumulator is loaded with = 1000

7. Write a **SIC/XE** Assembler Language program that picks two numbers from memory locations **NUM1** and **NUM2** and then stores the big number in a memory location **BIG** and the small number in a memory location **SMALL**. Append the appropriate directives to this program.

```

LDA    NUM1
COMP   NUM2
JLT    SMALLER
LDA    NUM2
STA    BIG
LDA    NUM2
STA    SMALL
J      OUT
SMALLER: STA SMALL
        LDA NUM2
        STA BIG
OUT
```