(14)

Name: _Parul Shandilya_   Roll No. _16UCS126_

COMPUTER ORGANIZATION &ARCHITECTURE (2017-18, I)

MIDTERM (LAB)

Max. Marks: 20   **Time: 50 mins.**

1. A set of data bytes is stored in memory locations starting at 100H. The values are 32,45,65,73,28,00. The following ALP is run for this set of values: (2+2+1=5)

        MOVE #100, R1   Rough

        MOVE #0, R5   [72 | 45]
                                                          ↑
L0:   LDB @R1, R2   100

Ending { CMP #0, R2   33
       { JE $L3   R2 = 1 odd   R1 = 100
                         R2 = 0 even.   R5 = 1
        AND #1, R2   //Statement 1   R1 = 2 102
                                                       R5 = 2
if even. [CMP #0, R2   R1 = 2 1

           JNE $L2

           INC R5   ②

L2:   INC R1

           JMP $L0

L3:   OUT R5, 0

           HLT

a) What is the output of this program? ___③ ✗ (R5↑ if odd)___

b) Why is Statement 1 executed?
   To check whether R2 is even or odd
   R2 = 1 → odd   R2 ⇒ 0 → even.

c) What will be the final value in register R1?___② ✗ ___

2. You are given a set of ten resistors assumed to be having 10 ohm value. The following ALP tests the resistors and rejects all resistors that are outside the tolerance. The values of resistors are stored as data bytes from memory location starting at 200. The values are: 7, 8, 9, 9, 10, 11, 13, 9, 12, 10.

(2+1+1+1=5)

200

```
            MOVE #10, R0
            MOVE #0, R5
            MOVE #200, R10
L0:         LDB @R10, RJ
            MSF
            CAL $L1
            CMP #0, R0
            JNE $L0
            OUT R5, 0,
            HLT

L1:         CMP #9, RI
            JLT $L2
            CMP #11, RI          // Statement 2
            _____          // Statement 3
            INC R5
L2:         INC R10
            DEC R0
            RET
```

(handwritten notes, right margin)

RS increases
if $9 \leq R1$
or $10 \geq R1$
$11 \leq R1$

$R1 \geq 9$
$R1 \leq 11$ RS ↑
$9 > R1$
$9 \leq R1$
$9 > R1$

$11 \geq R1$

JLT

(circled) 4

a) What is the output of the program? **6** ✓ $(9 \leq R1 \leq 11)$

b) Complete Statement 3.

_____ ~~MOVE R10~~ JGT $L2 ✓

c) What will be the output if Statements 2 and 3 are removed?

~~6 8~~ 7 will be output ✗ $(9 \leq R1)$

d) What is the accepted range of resistor values? **from 9 to 11**

$9 \leq R1 \leq 11$ ✓

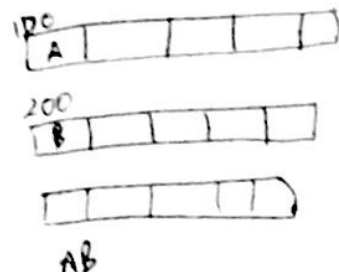(handwritten, right margin)
1. p + 11
R0 + 9.

3. There are two vectors which start at memory locations #100 and #200 respectively. Consider the program given below. Each instruction follows the convention " Opcode *Destination, Source*"

	Move	R1, #100
	Move	R2, #200
	Load	R3, #N
	Clear	R0
LOOP:	Load	R4, (R1)+
	Load	R5, (R2)+
	Multiply	R5, R4
	Add	R0, R5
	Decrement	R3
	Branch>0	$LOOP
	HLT	

a. What is the program doing?   (3)

It is multiplying the respective i^{th} value of a vector and adding these values and storing it in R0. N element in a vector.

b. What will be the total number of memory reference of the program?   (2)

Address at R1 and R2 reffered N times. So total 2N+1 times.

c. What is the meaning of "(R1)+" ?   (1)

It is indirect accessing. Address stored in R1 is accessed and then the value of R1 is increased.

4. The following program counts the length of a string where "#string" points to the start of the string and "#0x0D" points to the end of the string. First operand is destination operand in all the instructions. Length of the string should be stored at memory location 1001.

```
                Move R2, #string
                Clear R3
                Move R4, #0x0D
LOOP:           LoadByte R5, (R2)
                Branch_if_ [R5=R4]  $DONE
                Add R2, #1                      //Statement 4
                Add R3, #1                      //Statement 5
                Branch $LOOP
DONE:           Store R3
```

STB   1001   R3.

a. What are the two Add instructions doing?   (2)

Statement 4: ___R2 is increasing the address that should be accessed.___

Statement 5: ___R3 is counting the string length.___

b. Is there anything missing in the program? If yes, complete instruction.   (2)

___STB   1001   R3___

② ② ②