End Semester Examination (5th sem), 2021

→ Subject Name : → Microprocessor and Microcontroller.

→ Subject Code : → IT 3101

→ Date of Examination : → 02.12.2021

→ Name : → Aniket Majhi.

→ Examination Roll Number : → 510819019.

→ G Suite ID : → 510819019.aniket@students.iiests.ac.in

→ Number of sheets uploaded : → 13.

ANSWER

(3)

(a) After Execution of each instruction the values are :→

Initial	A	B	C	D	E	H	L	1000	1001	1002	1003	1004
Initial	35	28	41	10	02	25	00	27	25	37	41	56
LDAX D	37	28	41	10	02	25	00	27	25	37	41	56
XCHG	37	28	41	25	00	10	02	27	25	37	41	56
MVI M,56H	37	28	41	25	00	10	02	27	25	56	41	56
MVI A,25H	25	28	41	25	00	10	02	27	25	56	41	56
MOV C,D	25	28	25	25	00	10	02	27	25	56	41	56

(b.) The assembly code to transfer entire block of data to 8070H from 8050H is :→

```
      LXI  8050H
      LXI  8070H
      MVI  B, 10H
Loop: MOV  A, M
      STAX D
      INX  H
      INX  D
      DCR  B
      JNZ  LOOP
      HLT.
```

(2)

(a) Difference between Compare and Subtract instruction in 8085: →

In Subtract instruction the final result is placed in accumulator.

Example: → SUB B

it updates the A with A - B.

A ← A - B.

In compare instruction, accumulator does not store the final result it just compares the result with the accumulator so accumulator contents will not get updated.

Example: → CMP B.

Checks if A > B or not

(b.) CALL Instruction: →

Mnemonics: →

LXI SP, 8100H. → Initialize the stack pointer at 8100H.

SP ← 8100H.

CALL 8050H → Call the Subroutine located at 8050 H.

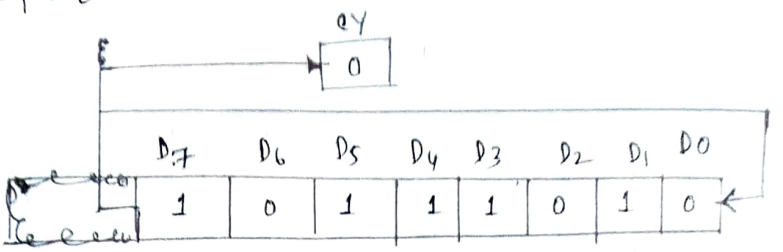
RET [returns the execution in main program].

Explanation :-

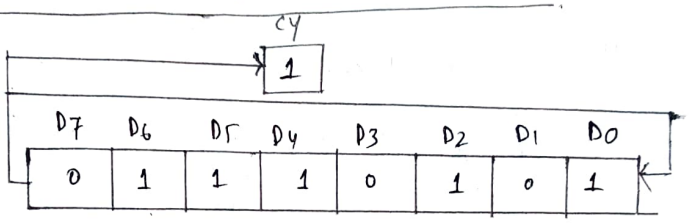
(a.)

A = B AH

CY = 0

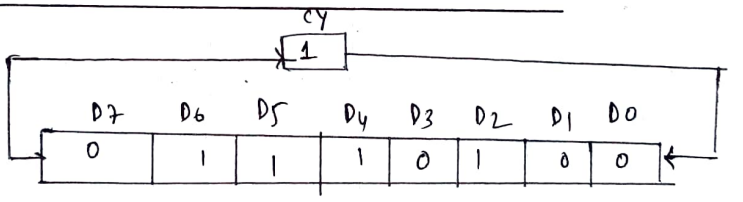


☒ After Execution of a (RLC instruction) :→



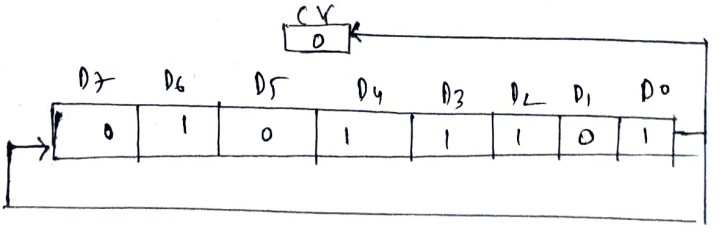
A = 75H, CY = 1.

☒ After Execution of b (RAL instruction) :→



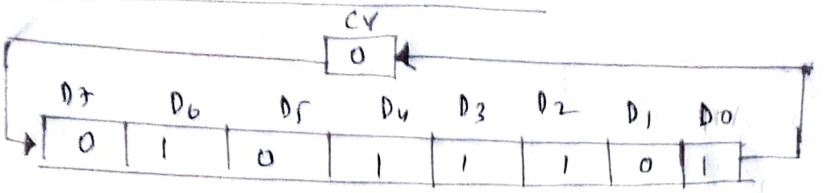
A = 74 H , CY = 1

☒ After Execution of c (RRC instruction) :→



A = 5DH , CY = 0

(a.) After Execution of ~~ind~~ d(RAR) : ⟶



A = 50H , CY = 0

(b.)

In the operation of XCHG instruction Register HL pair and DE pair involves.

Basically, it means,

Exchange H & L with D & E.

the Contents of register H are exchanged with the contents of register D & the contents of register L are exchanged with Contents of register E.

(5)

(a.) Difference between memory mapped I/O and I/O mapped I/O :→

	Memory mapped I/O	I/O mapped I/O
	(1) Here I/O devices are treated as memory	(1) Here I/O devices are treated as I/O devices.
	(2) The alloted address size is 16 bit ($A_0 - A_{15}$)	(2) The alloted address size is 8 bit ($A_0 - A_7$)
	(3.) The data transfer instruction is same for memory and I/O devices	(3) The data transfer instruction is different for memory and I/O devices
	(4) Here memory read and memory write cycles are involved	(4) Here I/O read and I/O write cycles are involved.

(b.) Operating modes of 8255 programming peripheral interface :→

There are two different modes of 8255. These modes are —

(i) Bit set reset (BSR) mode.

(ii) Input output mode.

(i) Bit set Reset Mode :⟶

This mode is used to set or reset the bits of the port-c only. For BSR mode always D7 will be 0. The control register is looking like this ⟶

Bits	D7	D6	D5	D4	D3	D2	D1	D0
Val	0	X			PC bit Number			0 or 1

The (D3, D2, D1) will be 000 to 111.

In this mode it affects only one bit of port C at a time. when user set the bits it remains set until user unset it. The user needs to load the bit pattern in control register to change the bit.

(ii) I/O mode :⟶

This mode is selected when the D7 bit of the control register is 1.

The mode has also three different modes.

(i) Mode 0 :⟶ Simple or basic I/O mode

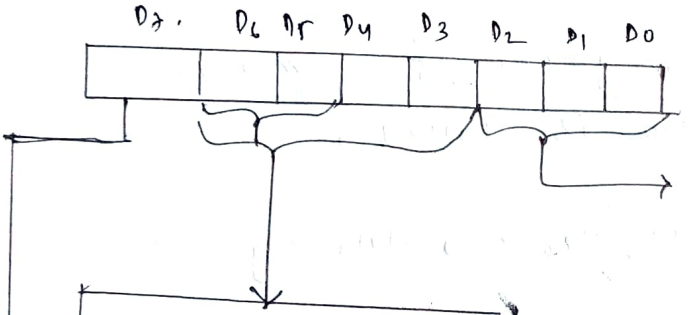
(ii) Mode 1 :⟶ Handshake or Strobed I/O

(iii) Mode 3 :⟶ Bidirectional I/O.

(e:) Format for control word of 8255 PPI chip.

As we know, a port can be programmed to act as an input port or an output port. To tell about that we generate a bit pattern or we may say a word which is called control word. A control word is, therefore, to be formed for Programming the ports of 8255A.

The format for the control word format for 8255A is below,



Group A

PORT C (upper) PC_7-PC_0

1 → input
0 → output.

PORT A

1 → input
0 → output.

Mode selection,

00 for 0
01 for 1
1x for 2.

Group B.

PORT C (lower) PC_3-PC_0

1 → input
0 → output

PORT B,

1 → input
0 → output.

Mode selection for Port B

0 → Mode 0
1 → Mode 1

1 for I/O Mode
0 for BSR Mode

6

a) Segmentation is the process in which the main memory is logically divided into segments with each segment having its base address.

In 8086, there are 4 special purpose 16 bit register ~~Bus~~ Bus Integer Unit (BIU).

1) Code Segment register :— Addressing memory locations ~~Add~~ Code segment of memory, where program is stored.

2) Data segment register :— data segment register stores data.

3) Extra segment register :—

4) Stack segment register :—

(b) Advantages of the segmented memory : →

(i) To generate 20 bit Physical address, the Processor just maintains two 16-bit registers, which are within the word length capacity of the machine.

(ii) All these segments are logical segments. They may or may not be Physically separated.

(c) Physical addressing of 8086 : →

(i) Complete Physical address of 20 bits long is generated by using 16-bit segment and 16 bit offset registers.

(ii) segment register contains 16 bit segment base address corresponding to segment.

(iii) Any of the pointers and index registers or BX may contains offset of location to be addressed.

(iv) For generating a Physical address from the contents of these two registers, the contents of the segment register is shifted left with 4 bit-wise 4 times in 20 bit address format and to this result, content of an offset register also called as offset address is added, to produce a 20 bit Physical address.

(a.)

segment address —

1004H → 0001 0000 0000 0100

offset address —

5434H → 0101 0100 0011 0100

Shifted by 4 bit positions,

0001 0000 0000 0100 0000

(10040H)

offset address.

0101 0100 0011 0100

(5434H)

0001 0101 0100 0111 0100

(15475H).

(7)

(8) (a) Loop :- In loop instructions the code is gonna execute several times. The block of codes are executed continuously till the expression is true.

Jump :- In JUMP instruction, ~~the result~~ is used to jump from one block of statement to another. It is used to skip some of the parts.

(b) Program :-

```
MOV  AX, 3000H   ; Load 3000H in AX
MOV  DS, AX      ; Initialize data segment with 3000H.
MOV  BX, [0300H] ; Move contents of addr with 0300H offset in BX
ADD  [0300H], 0FH ; Add byte 0FH to contents in BX.
MOV  DX, [0300H] ; Store result in DX.
MOV  [0400H], DX ; Store result in 0400H
HLT              ; Stop execution.
```

(6)

Program:→

```
        MOV   CX, 0FH    ; Initialize counter (CX)
                         for number of iterations.
        MOV   AX, 3000H  ; load AX with 3000H
        MOV   DS, AX     ; Initialize data segment
                         (DS) with 3000H.
        MOV   SI, 0300H  ; initialize source pointer
                         with 0300H.
        MOV   AX, [SI]   ; load first number in
                         AX
XX:     INC   SI         ; increment source
                         Pointer from 0300H.
        CMP   AX, [SI]   ; Compare with next
                         number.
        JNC   YY         ; If the next
                         number is not
                         larger, go to YY.
        MOV   AX, [SI]   ; replace the
                         previous one
                         with the next in
                         AX
YY:     LOOP  XX         ; Repeat the proces for 15 times.
        HLT              ; Halt (stop execution)
```