

CSCI-261

Prineekumar
Diyorze

Midterm 1

656570553

Q

- ① False \rightarrow bcz I think it accept 16 bit address space
- ② True
- ③ False

b

11111110 ob

↓

flipping the digits

0000 0001

+1 (adding 1 to it)

0000 0010 ob

$\begin{matrix} \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\ 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 \end{matrix}$

(\because but we can see that
in the beginning there is
① and it points to negative
sign)

$= 2^1 \times 1$

$= -2$ \rightarrow it represents -2 in decimal number

c

opcode is the 0's and 1's which correspond to particular instructions, and also in machine language might opcode is followed by the operands or it can be the references to Operands. whereas, mnemonic is the symbolic name for the opcodes. Here, it can be one opcode or group of opcodes.

- (d) In the immediate mode of addressing opcode is immediately followed by the operand on the stack.
- In the extended mode of addressing opcode is followed by the addresses. Here, the addresses follows the big endian notation. So, opcode, bigger part of address, lower part of address, and then operand. This four will be on stack in given order.
- In the inherent mode of addressing there will be only operator on the stack, and there is no operands on stack.

(Assumption: I am showing the value to)
the Port A

Q.2 (a)

→ nor operation does not set the Z flag (0 to 1 transition)

Assembly Code:-

Pushimm 128 -- Pushing 128 on top of the stack

Popext PSW -- clearing out Z flag

Pushimm 1 -- Pushing Value 1 on stack

Pushimm 2 -- Pushing Value 2 on stack

nor -- nor operation between two top values of stack

Pushext PSW -- go to PSW and put the value on stack

Popext A -- it will take the value from the stack and leave it at Port A

halt

(b)

SSBC assembly language code

Pushimm 255

Pushimm 0

Pushimm 0

add

Pushext PSW

Popext A

halt

Question 3:-

(Assumption: Just passing the parameter n on top of the stack by Pushimm n .)

-- mein -- -- -- --

Pushimm 0 -- clearing out ZF n
Popext PSW flag

Pushim RL(14) -- Pushing lower return address

Pushimm RH(0) and higher return address

Pushimm n on stack. 14 & 0 which is

JNZ subRoutine(15) Value in decimal

YY: halt -- jump to the subroutine
and 15 is location in decimal

-- Sub routine -- -- -- --

subRoutine: Pushimm 1 -- Pushing 1 on the stack
Sub -- subtracting 1 from
 n ($\because n-1$)

popinh -- taking out $n-1$

Popext RH(26) -- take the RH & RL and put

Popext RL(27) it to the given location

JNZ returnAddress

Pushimm 2

-- Pushing 2 on stack

Popext A

-- save it at Post A

Pushimm n

-- now our stack is empty

Popext A

-- so pushing value n on stack

JNZ returnAddress(26) -- again

-- and then save it to
the Post A

-- go back to the given location

(Our subroutine is starting from line number 32 in decimal)

Q.4 Machine language code

XX : 00000010 Push imm 1

00000001

00000101

Popext A

11111111

11111100

00000010

Push imm 0

00000000

00000101

Popext PSW

11111111

11111011

00000101

popext xx+17d (49) ↴
in decimal

00000000

00110001

00000101

Popext xx+18d (50) ↴
in decimal

00000000

00110010

00000010

jnz

00000000

00000000