PASS- Computer Systems

Week 8

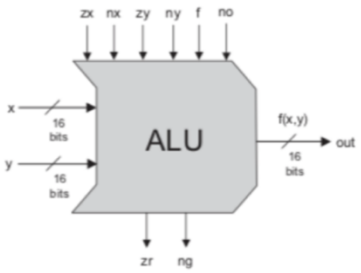
1. There is a limit on the number of ways to express a Nand chip in terms of other chips. True or False?

2. In eight-bit two's complement notation write down the binary for -126
    a. -125 is too large to represent in 8 bits 2's complement
    b. 1000 0010
    c. 1000 0011
    d. 0111 1101

3. Imagine that you had a data flip-flop DDF and, during successive clock cycles the signal on its input is: 0,0,1,0,1,0,0,1. What is the output of the DFF on these same clock cycles?

    a. ?,0,0,1,0,1,0,0
    b. 0,0,1,0,1,0,0,0
    c. 0,1,0,1,0,0,1,0
    d. ?,0,1,0,1,0,0,1

Use the following for questions 4-11



zx	nx	zy	ny	f	no

if zx==0 then
x1=x
else
x1=0
	if nx==0 then
	x2=x1
	else
	x2=!x1
		if zy==0 then
		y1=y
		else
		y1=0
			if ny==0 then
			y2=y1
			else
			y2=!y1
				if f==0 then
				fout=x2&y2
				else
				fout=x2+y2
					if no==0 then
					out=fout
					else
					out=!fout

Notes:

- The values of **x1**, **x2**, **y1**, **y2**, **fout** and **out** must be expressed as simplified arithmetic expressions and may include a single **x**, a single **y**, a single digit (**0**, **1** or **2**) and the operators, **+** and **-**.
- If an expression starts with **-**, all operators must be **-**.
- If an expression is **0**, it may be expressed as **0** or **zero**.
- The values of **zr** and **ng** must be expressed as **true** or **false**.
- Your answers must not include any spaces.

What are the values x1, x2, y1, y2, fout and **out** when the ALU control inputs have the following values?

4. if zx == 0 then x1 =

5. if nx == 0 then x2 =

6. if zy == 0 then y1 =

7. if ny == 0 then y2 =

8. If f == 0 then fout =

9. If no == 1 then out =

What values would be output on the zr and ng wires if the values of x and y are as follows?

10. If x == true and y == false then zr =

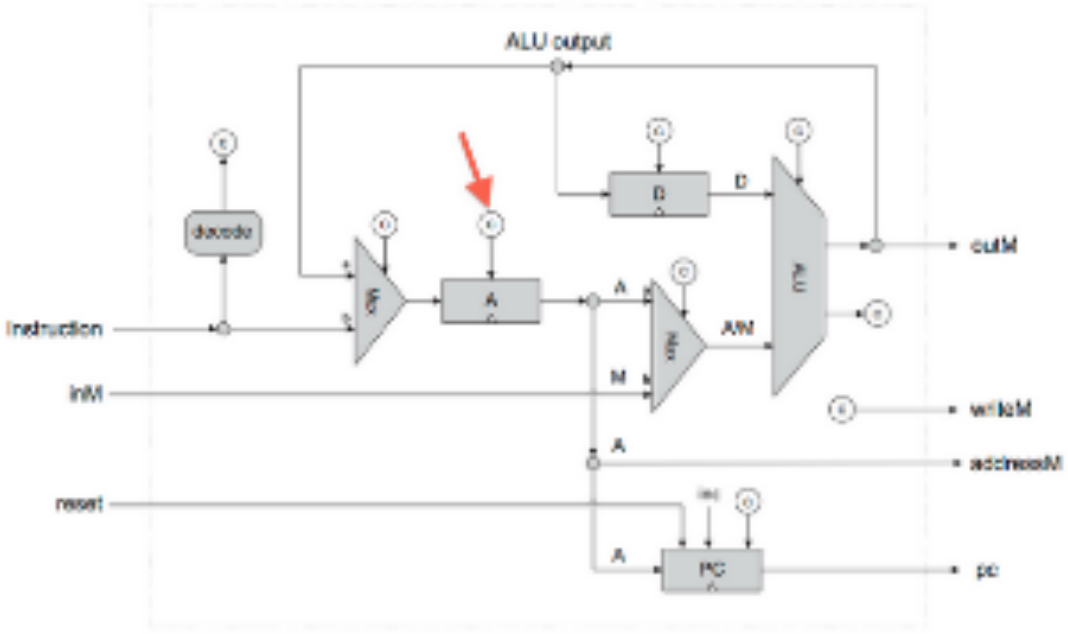
11. If x == true and y == true then ng =

12. What does the following Hack assembler code do?

```
(LOOP)
@KBD
D=M
@48
D=D-A
@num
M=D
@LOOP
D;JLT
@10
D=D-A
@LOOP
D;JGE
(END)
@END
0;JMP
```

a. It loops, putting the scan code into "num", until the key pressed is between scan code 48 and 57.
b. It loops, putting the scan code into "num", until the keyboard scan code is less than 48 (which translates to the char '0').
c. It doesn't work because it goes into an infinite loop at the end of the code.
d. It loops, putting the scan code minus 48 into "num", until the key pressed is between scan code 48 and 57.

13. Look at the following (incomplete) diagram of the Hack CPU. Look at the wire pointed to by the large red arrow. Where does the signal on this wire come from?



a. d1 (the left most ***destination*** bit in an instruction)
b. i15(left most bit of the instruction)
c. d1 OR i15
d. d1 OR (Not i15)

14. What does the following code do to current value in register D?

D = !D

D = D +1

a. Set D to be D-1
b. Set D to be 1-D
c. Set D to be –D
d. D does not change

15. Translate below HACK assembly instruction to 16-bit binary code:
    a. !D;JNE

16. How does a programmer finish a program in Hack Assembler?
    a. By specifying a label followed by an instruction that unconditionally jumps to itself.
    b. By setting the address of the A register to zero.
    c. By using halt instruction
    d. By including an instruction that is all zero bits at the end of the code and then jumping conditionally to that instruction.

17. Assume that we have a stack where the value of the top element is 7 and the second top element is 12. If the VM command ***sub*** is executed, and the value 12 is stored at memory address 568, what is the new value of the stack pointer?
    a. 567
    b. 568
    c. 569
    d. 570

18. How many memory segments does the Hack Virtual Machine store in the data memory?
    a. 8
    b. 7
    c. 6
    d. 5

19. Which of the following fragments of Hack assembly language would implement the following virtual machine command? If there is more than one correct answer, select all of them.

    push constant 0

    a. @SP

       M=M+1

       A=M

       M=0

       @SP

    b. @0

       D=A

       @SP

```
M=D
@SP
M=M+1
```

c.
```
A=0
M=M+1
AD=M-1
M=0
```

d.
```
@SP
A=M
M=0
@SP
M=M+1
```