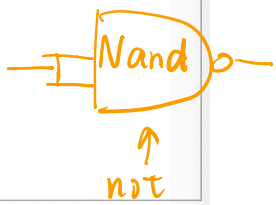
Question 1

1 / 1 pts

Category: Minimum Nands

What is the smallest number of **Nand** gates required to implement a **Not** chip? Your answer must only contain decimal digits.

1



Incorrect

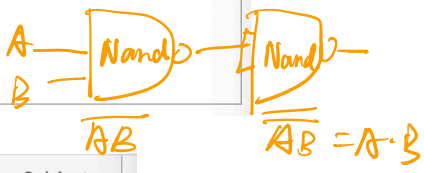
Question 2

0 / 1 pts

Category: Minimum Nands

What is the smallest number of **Nand** gates required to implement an **And** chip?

1



Incorrect

Question 3

0 / 1 pts

Category: Combinational Logic

The Hack ALU has only 18 instructions listed in figure 2.6 of the textbook but there are 2^6 = 64 possible input wire configurations. Why aren't all of these instructions available?

- Actually there are 64 instructions possible - in the machine code - but most are redundant so the assembler doesn't give the programmer a way to express these. ✓✓
- To implement all 64 instructions would greatly increase the complexity of the processor and for no benefit since we can do almost everything we want with the 18 instructions we have.
- Because most of the other instructions would conflict with the 18 instructions that we have - making it so that they don't work properly.
- The other instructions are mostly slower than the 18 very fast instructions. We don't implement these other instructions because we don't want to slow down the machine.

Question 4

1 / 1 pts

Category: Combinational Logic

Consider the following statements on this course's HDL programs.

Which one is wrong?

- HDL is case sensitive and its keywords are written in uppercase letters
- Names of chips and pins may be any sequence of letters and digits not starting with a digit.
- Whenever an HDL program XXX.hdl is loaded and parsed into the hardware simulator, the corresponding simulator's *builtIn* chip will be loaded.

  This statement is wrong.The simulator will not load the XXX.hdl in the builtIn chips directory if the XXX.hdl exists in the current directory.
- A chip definition consists of a header and a body. The header specifies the chip interface and the body its implementation.

Question 5

1 / 1 pts

Category: Combinational Canonical

Select all boolean expressions below that are part of the canonical representation of the **And** boolean function?

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

Note: In the boolean expressions

- the **not** operator is represented by **!**
- the **and** operator is represented by **&**

- [x] a & b
- [ ] !a & !b
- [ ] !a & b
- [ ] a & !b

Question 6

0.5 / 1 pts

Category: Combinational Canonical

Select all boolean expressions below that are part of the canonical representation of the **Mux** boolean function?

a	b	sel	out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

考点：SOP

$(\bar{a} \cdot b \cdot sel) + (a \bar{b} \overline{sel}) + (a b \overline{sel}) + (a b sel)$

Note: In the boolean expressions

- the **not** operator is represented by **!**
- the **and** operator is represented by **&**

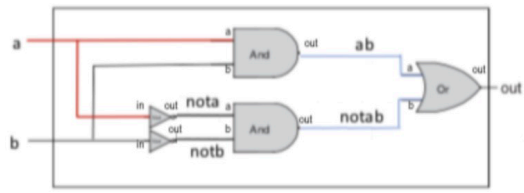
- [x] a & b & sel ①
- [ ] a & b & !sel ③
- [ ] a & !b & sel
- [x] a & !b & !sel ②
- [ ] !a & b & sel ④
- [ ] !a & b & !sel
- [ ] !a & !b & sel
- [ ] !a & !b & !sel

Question 7

0.6 / 1 pts

Category: Circuit HDL

Select the lines of HDL code shown below that will implement an Xnor chip that is wired as shown in this diagram:




```
Chip Xnor
{
        IN a, b ;
        OUT out ;

        PARTS:
        // select the lines of HDL code that should go here ...
}
```

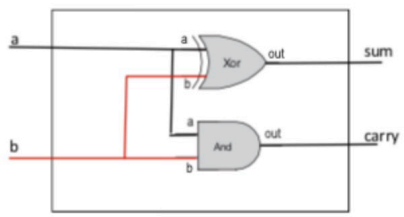
- [x] Not(in=a,out=nota);
- [x] Not(in=b,out=notb);
- [ ] And(a=a,out=ab,b=b);
- [x] And(a=nota,b=notb,out=notab);
- [x] Or(a=ab,b=notab,out=out);
- [ ] Nand(a=a,b=a,out=nota);
- [ ] Nand(a=b,b=b,out=notb);
- [x] And(a=b,b=a,out=ab);
- [ ] Or(out=out,a=notab,b=ab);
- [ ] Nand(a=nota,b=notb,out=notab);

Question 8

1 / 1 pts

Category: Circuit HDL

Select the lines of HDL code shown below that will implement a HalfAdder chip that is wired as shown in this diagram:




```
Chip HalfAdder
{
        IN a, b ;
        OUT sum, carry ;

        PARTS:
        // select the lines of HDL code that should go here ...
}
```

- [x] Xor(a=a,b=b,out=sum) ;
- [ ] Xor(b=a,a=b,out=sum) ;
- [ ] Xor(a=a,b=b,sum=out) ;
- [x] And(a=a,b=b,out=carry) ;
- [ ] And(a=b,b=a,out=carry) ;
- [ ] And(a=a,b=b,carry=out) ;

unsigned
可表示 2^n 个数, 从 0 - 2^n-1

one's complement:
可表示 2^n-1 个数,
从 $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

two's complement:
可表示 2^n 个数:
从 -2^{n-1} to $(2^{n-1}-1)$

Question 9

1 / 1 pts

Category: Maximum 8-bit Twos Complement

What is the largest decimal number that can be represented by a two's complement 8-bit binary number?

127

$-(2^{n-1})$ to $(2^{n-1}-1)$

$2^{8-1}-1$
$=2^7-1=127$

Question 10

1 / 1 pts

Category: Maximum 3-bit Unsigned

What is the largest decimal number that can be represented by an unsigned 3-bit binary number?

7

Max:
$2^n - 1$
$2^3 - 1 = 7$

Question 11

1 / 1 pts

Category: Unsigned Minimums

What is the decimal value of the smallest number that can be represented by a 13-bit unsigned binary number?

0

Question 12

1 / 1 pts

Category: Unsigned Minimums

What is the decimal value of the smallest number that can be represented by a 16-bit unsigned binary number?

0

Question 13

1 / 1 pts

Category: Unsigned 10000000

What is the decimal value of the unsigned binary number 10000000?

128

10,000,000

76 543 210

$2^7 = 128$

Question 14

1 / 1 pts

Category: Twos Complement 10000000

What is the decimal value of the 8-bit two's complement binary number 10000000?

-128

10,000,000

$-2^7 = -128$

Question 15

0 / 1 pts

Category: Twos Complement 24 + 8

Assuming that all numbers are represented in 6-bit two's complement binary, what is the decimal value of the result of adding 24 and 8?

32

24 → 2|24 0, 2|12 0, 2|6 0, 2|3 1, 1

011000

8 → 2|8 0, 2|4 0, 2|2 0, 1

001000

011000
+ 001000
100000
↓
-32

Question 16

0 / 1 pts

Category: 4-bit Twos Complement 4 + 5

Assuming that all numbers are represented in 4-bit two's complement binary, what is the decimal value of the result of adding 4 and 5?

7

4 → 0100

5 → 0101

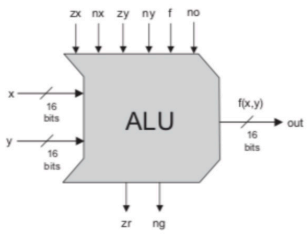
0100
+ 0101
1001
↓
-7

Question 17

1 / 1 pts

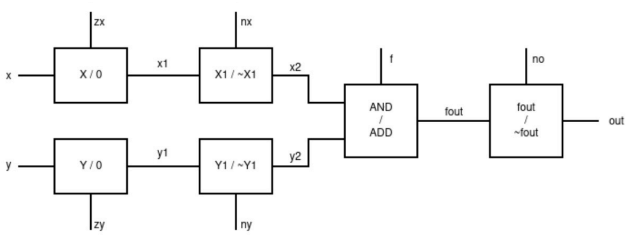
Category: ALU

The following diagrams show the interface to the Hack ALU and the effect of the six control inputs zx, nx, zy, ny, f and no.



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



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

if zx == 0 then x1 = x

if nx == 0 then x2 = x

if zy == 1 then y1 = false

if ny == 1 then y2 = true

if f == 0 then fout = x

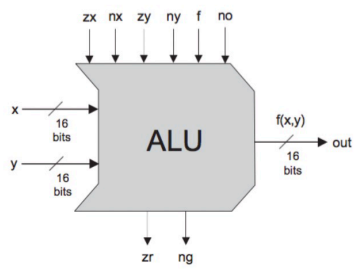
if no == 1 then out = !x

Question 19

1 / 1 pts

Category: ALU

The following diagrams show the interface to the Hack ALU.



What are the values on the **zr** and **ng** pins when the **out** value is `true`?

zr	false (0)
ng	true (1)

Question 20

0 / 1 pts

Category: Sequential DFF

The clock signal goes to every DFF in the Hack machine. What does the DFF do in response the clock signal?

- The DFF will only load its input value if the clock signal is on an edge. VV
- The DFF will only output a value if the clock signal is on an edge.
- The DFF oscillates its output value between 1 and 0 with the clock signal.

No the output of the DFF depends on its stored value.

- The rest of the machine is only able to "see" output value of the DFF when the clock signal is on an edge.

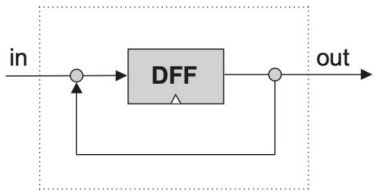
This question concerns how the clock allows the architecture to have a consistent view of the memory state.

Question 21

1 / 1 pts

Category: Sequential DFF

Consider this diagram taken from figure 3.1 in the textbook showing a DFF feeding back into itself.



Why does this not work as a solution to storing a bit of information?

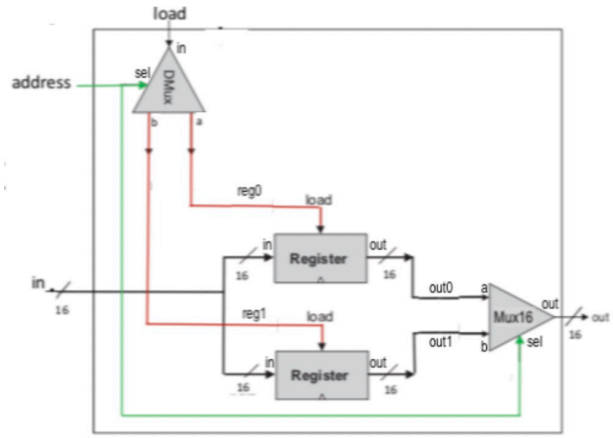
- Because it is impossible to make wires go backwards
- Because there is no reliable way to chose from the value in the input and the value in the output if they are different.
- Because we need to boost the voltage on the return wire.
- Because the circuit is too complex.

Question 22

1 / 1 pts

Category: Circuit HDL

Select the lines of HDL code shown below that will implement a RAM chip with two 16-bit registers that is wired as shown in this diagram:




```
Chip RAM2
{
        IN in[16], load, address ;
        OUT out[16] ;

        PARTS:
```

```
        // select the lines of HDL code that should go here ...
}
```

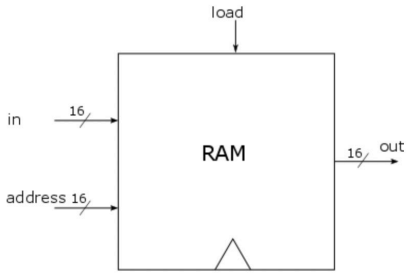
- [x] DMux(in=load,sel=address,a=reg0,b=reg1) ;
- [x] Register(in=in,load=reg0,out=out0) ;
- [x] Register(in=in,load=reg1,out=out1) ;
- [x] Mux16(a=out0,b=out1,sel=address,out=out);
- [ ] DMux(in=load,sel=address,b=reg0,a=reg1) ;
- [ ] DMux16(in=in,sel=address,a=reg0,b=reg1) ;
- [ ] Register(in=reg0,load=in,out=out0) ;
- [ ] Register(in=reg1,load=in,out=out1) ;
- [ ] Register(in=in,load=address,out=out0) ;
- [ ] Register(in=in,load=reg0,out=out) ;
- [ ] Mux16(in=load,sel=address,a=reg0,b=reg1) ;
- [ ] Mux(a=out1,b=out0,sel=address,out=out);
- [ ] DMux16(a=out0,b=out1,sel=address,out=out);

Question 23

1 / 1 pts

Category: Sequential RAM

What is the capacity of the following memory in terms of bytes (B) or bits (b)?



- ( ) 1 MB
- ( ) 256 Mb
- (•) 128 KB

Excellent.

Question 24

0 / 1 pts

Category: Bitwise Arithmetic

C++, Java & Python have the following bitwise operators, **&** (and), **|** (or) and **~**(not).

What is the decimal value of the expression, `~3`, if all numbers are represented in 8-bit two's complement.

Write your answer as a number.

2 ✗ -4

3 → 00000011

not 3 → 11111100 ⇐ two's complement

7 6 5 4 3 2 1 0

$-2^7+2^6+2^5+2^4+2^3+2^2$

$= -128 + 64+32+16+8+4$

$= -4$

Question 25

0 / 1 pts

Category: Bitwise Arithmetic

C++, Java & Python have the following bitwise operators, **&** (and), **|** (or) and **~**(not).

What is the decimal value of the expression, `19 & ~7`, if all numbers are represented in 8-bit two's complement.

Write your answer as a number.

4 ✗ 16

7 → 00000111

~7 → 11111000

19 → 00010011

```
  11111000
x 00010011
----------
  00010000
```

$2^4 = 16$