CSO1 **Name:** _____
Spring 2022
Midterm 1
2022-02-24
Time Limit: 75 minutes Computing ID _____

Instructions:

1. This exam contains 7 pages (including this cover page) and 16 questions.
2. You have **75 minutes** to complete the examination. As a courtesy to your classmates, we ask that you not leave during the last fifteen minutes.
3. Write your answers in this booklet. We scan this into GradeScope, so **please try to avoid writing on the backs of pages**.
4. If a question presents several options in a list, mark the bubble next to the one correct answer. All such questions on this test are single-select.
5. You may not use a calculator or notes.
6. Because this assessment is being given in several places, we cannot fairly answer questions during it. If you find a question ambiguous or unclear, please explain that on the page by the question itself and we will consider your explanation during grading.
7. Please sign the below Honor Code statement.

I have neither given nor received aid on this exam.

Signature: _____

Our Example ISA

This is the same ISA used in HW03 and HW04, but presented to fit onto one printed page.

Each instruction is one or two bytes, with the meaning of those bytes being:

Not all instructions have the second byte; those that do describe it below as the byte "at `pc + 1`".

In the table below `rA` means "the value stored in register number `a`" and `rB` means "the value stored in register number `b`."

`icode`	`b`	Behavior	add to `pc`
0		`rA = rB`	1
1		`rA += rB`	1
2		`rA &= rB`	1
3		`rA` = read from memory at address `rB`	1
4		write `rA` to memory at address `rB`	1
5	0	`rA = ~rA`	1
5	1	`rA = -rA`	1
5	2	`rA = !rA`	1
5	3	`rA = pc`	1
6	0	`rA` = read from memory at `pc + 1`	2
6	1	`rA +=` read from memory at `pc + 1`	2
6	2	`rA &=` read from memory at `pc + 1`	2
6	3	`rA` = read from memory at the address stored at `pc + 1`	2
7		if `rA <= 0`, set `pc = rB`	N/A
7		if `rA > 0`, do nothing	1

If the first bit of the byte at `pc` is `1` instead of `0`, the above text does not define what the instruction means, but some other source (such as a question on this exam) might. If it has no defined meaning either here or elsewhere, leave the `pc` and all other registers and memory values unchanged.

1. Assuming that you are operating on 1 bit binary values `a` and `b`. What is "`a ^ b`" equivalent to?

   ◯ `a == b`
   ◯ `a != b`
   ◯ `a >> b`
   ◯ `b >> a`

2. Assuming 8-bit machine with signed integers, what is the value of "`-4 >> 2`"?

   Answer as a signed base-10 number, like `+3` or `-14`.

3. If two's complement is used, how many non-negative values (i.e., zero or positive) can be represented by an 8 bit number?

   Answer as base-10 number, like `0` or `300`.

4. Which of the following circuits correctly implements an adder of two one-bit values, `x` and `y`?

   ◯ `output = x | y; carry = x ^ y;`
   ◯ `output = x ^ y; carry = x & y;`
   ◯ `output = x & y; carry = x ^ y;`
   ◯ `output = x >> y; carry = x | y;`

5. Which of the following is equivalent to "`p ^ q`" for all 4-bit values `p` and `q`?

   ◯ `(p | q) & (~(p & q))`
   ◯ `(p >> q) & p`
   ◯ `(p & ~q) | p`
   ◯ `(~p & q) & (q | p)`

6. Which of the following constructs the following one byte mask `11001100`?

   ◯ `~0x33`
   ◯ `1100 << 4`
   ◯ `1100 >> 4`
   ◯ `0xBB`

7. Consider the following stack operations. What is the value of `z`?

```
push(9)
push(4)
pop()
push(2)
push(1)
pop()
z = pop()
```

- ◯ 4
- ◯ 1
- ◯ 2
- ◯ 9

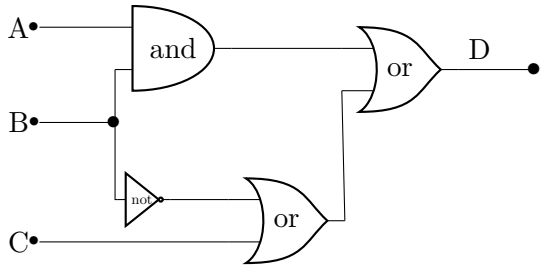
8. Consider the following program in our reference ISA (see page 2)

```
0x00 0x60 0x01 0x64 0xFF 0x08 0x01 0x06
```

Break this code into instructions and fill in the following table to provide the PC of the instruction, the bytes of the instruction, and the values of each program register after the instruction completes. (There are more lines in the table than you will need)

PC	Byte(s) of instruction	r0	r1	r2	r3
0	00	0	0	0	0
1	60 01				

9. Consider the following circuit.



Fill in the following truth table for this circuit:

A	B	C	D
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

10. Order the following bytes

```
0x02 0x60 0xF7 0x61
```

so that they implement the following program in our example ISA (see page 2)

```
x = 2
x -= 9
```

11. If the 32-bit integer `0x0A0B0C0D` is to be stored in a **little** endian machine, it will be stored the same way as which array of 8-bit values?

- ◯ `[0x0D, 0x0C, 0x0B, 0x0A]`
- ◯ `[0xD0, 0xC0, 0xB0, 0xA0]`
- ◯ `[0xA0, 0xB0, 0xC0, 0xD0]`
- ◯ `[0x0A, 0x0B, 0x0C, 0x0D]`
- ◯ none of the above

12. If the 32-bit integer `0x0A0B0C0D` is to be stored in a **big** endian machine, it will be stored the same way as which array of 8-bit values?

- ◯ `[0x0D, 0x0C, 0x0B, 0x0A]`
- ◯ `[0xD0, 0xC0, 0xB0, 0xA0]`
- ◯ `[0xA0, 0xB0, 0xC0, 0xD0]`
- ◯ `[0x0A, 0x0B, 0x0C, 0x0D]`
- ◯ none of the above

13. Which of the following will result in integer overflow on 4 bit machine?

- ◯ `-1 + -1`
- ◯ `7 + -5`
- ◯ `7 + 2`
- ◯ none of the above.

14. Consider the following code:

```
r3 = 0xA0
r0 = 50
r2 = pc
r0 += 1
r1 = r0
r1 += -60
if r1 <= 0, set pc = r2
M[r3] = r0
```

What is the value put into memory at address `0xA0` on the last line of the code?

15. Break the first instruction in the byte sequence `0x7A 0x52 0x60 0x00` into its parts by filling in the following:

If some box is not part of the instruction, write "N/A" in that box.

16. How difficult was this midterm?

- ◯ too easy
- ◯ easy but fair
- ◯ fair
- ◯ difficult but fair
- ◯ too difficult

.....

You may use the space below as scratch paper