Spring 2024
Exam I
2024-02-23

Instructions

This exam contains 10 pages (including this cover page) and 28 questions. It is out of 100 points.

You have **50 minutes** to complete the examination. As a courtesy to your classmates, we ask that you not leave during the last fifteen minutes.

For this exam, you have been given a separate scantron answer sheet to fill in your responses. We scan this into Gradescope so color in the bubble without going outside of the lines.

It is **OK** to write on your test pages.

It is **NOT OK** to write anything on your scantron answer sheet except for your name, computing ID, signature, and coloring in your bubbles (small circles) to answer the questions.

You may assume all questions on this test are single-select **unless otherwise indicated.**

You may **not** use a calculator or notes.

The next page contains reference material which you are welcome to refer to during the test if you would like.

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:

7	6	5	4	3	2	1	0
`0`	`icode`			`a`		`b`	

byte at `pc`

7	6	5	4	3	2	1	0
immediate							

byte at `pc + 1`

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

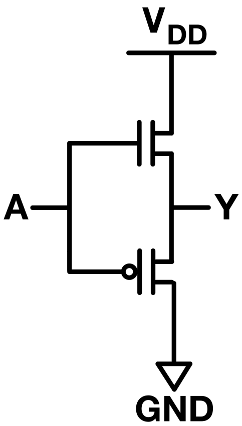
In the table below `rA` means "the value stored in register number `a`" and `rB` means "the value stored in register number `b`."

`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 Logic Gates & Boolean Algebra

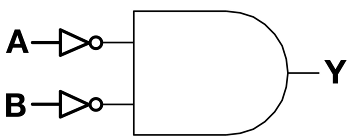
1. (4 points) Given input A, what is the output Y of the following circuit?



(A) A (B) ∼A (C) GND (D) V_{DD}

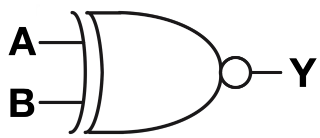
2. (6 points) Which of the following representations are equivalent to A NOR B? **Select all that apply.**

(A)



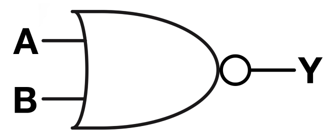
(B) `Y = ~(A & B)`

(C)



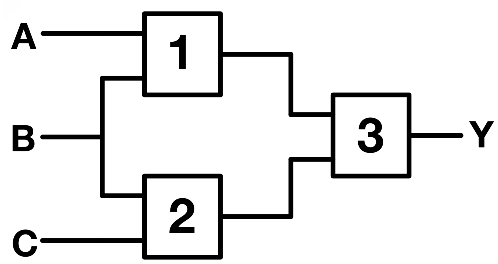
(D) `Y = ~A | ~B`

(E)

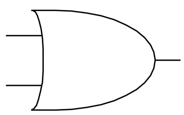


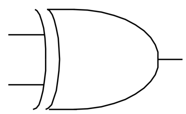
(F) `Y = ~(A | B)`

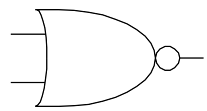
Without any simplification, the logic expression `Y = (A ^ B) & ~(B & C)` is to be implemented in hardware using the skeleton shown below.

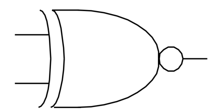


3. (2 points) Which gate should be placed in Box 1?

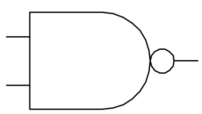
(A) 

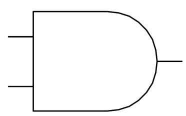
(B) 

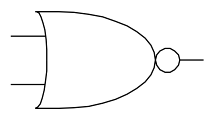
(C) 

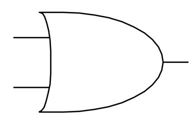
(D) 

4. (2 points) Which gate should be placed in Box 2?

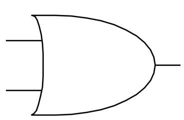
(A) 

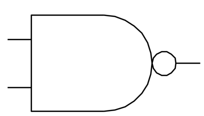
(B) 

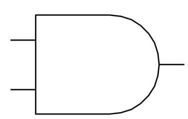
(C) 

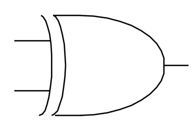
(D) 

5. (2 points) Which gate should be placed in Box 3?

(A) 

(B) 

(C) 

(D) 

2 Number Representations

6. (4 points) The two's complement number representation has a "weird" number that has no positive counterpart because it is its own two's complement. Which of the following is the "weird" number in 4-bit two's complement?

   (A) `0b0111`
   (B) `0b1000`
   (C) `0b1111`
   (D) `0b0000`

7. (2 points) **True or False:** When using 4 bits to represent numbers, the bit representation of 5 is the same in two's complement as in the unsigned representation.

   (A) True (B) False

8. (5 points) While reviewing code on a 32-bit system, you come across the following lines of code:

```
X = 0x1234
mask = ~0x0F
extraction = X & mask
print(extraction)
```

   What is the output in hex?

   (A) `0x1234`
   (B) `0x0034`
   (C) `0x0030`
   (D) `0x1230`
   (E) `0x123F`
   (F) `0xFFF4`
   (G) `0x0000`

9. (6 points) If a little-Endian x86 computer were to read the 32-bit number `0x12345678` from a big-Endian network without accounting for the difference in Endian-ness, how would the computer interpret the number?

3 IEEE Floating Point

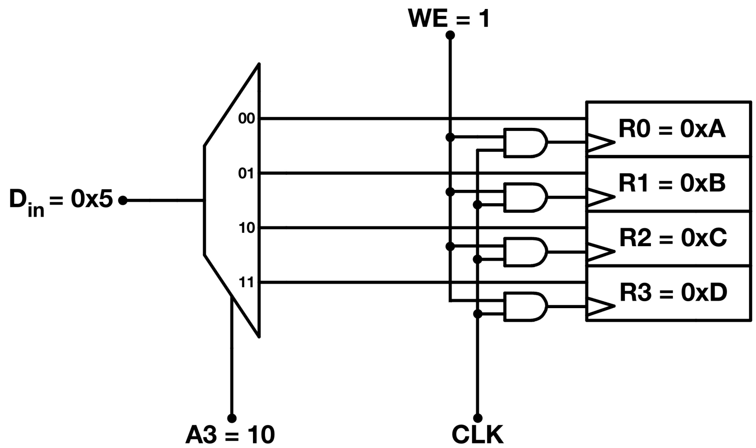
Write the following binary number as a 16-bit IEEE-style floating-point number assuming a 4-bit exponent value:

`-100100.11010001`

10. (3 points) What should the sign bit be set to?
11. (5 points) How would the 4-bit exponent be represented in binary?
12. (5 points) How would the mantissa be represented in binary?

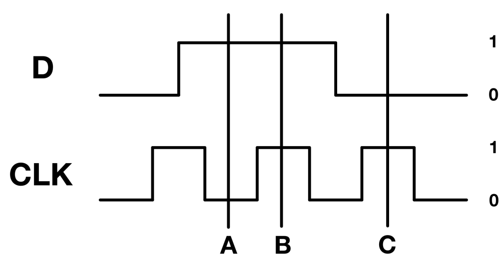
4 Circuits & Registers

13. (5 points) Consider the following diagram (note this is not our standard set-up for writing to a register file).



What is the value in each register after the next rising CLK edge?

(A) R0=0x0, R1=0x0, R2=0x5, R3=0x0
(B) R0=0xA, R1=0xB, R2=0xC, R3=0xD
(C) R0=0x0, R1=0x0, R2=0x0, R3=0x0
(D) R0=0xA, R1=0xB, R2=0x5, R3=0xD
(E) R0=0x1, R1=0x1, R2=0x1, R3=0x1
(F) R0=0x1, R1=0x1, R2=0x5, R3=0x1

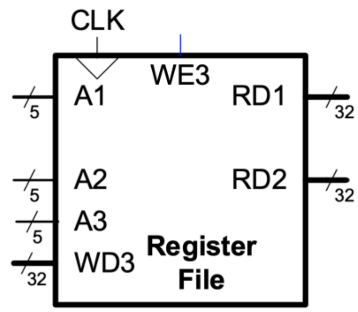


14. (5 points) Given the timing diagram for a positive edge-triggered D flip-flop shown above, what is the value of the output Q at times A, B, and C?

(A) A: Q = 0, B: Q = 0, C: Q = 0
(B) A: Q = 0, B: Q = 0, C: Q = 1
(C) A: Q = 0, B: Q = 1, C: Q = 0
(D) A: Q = 0, B: Q = 1, C: Q = 1
(E) A: Q = 1, B: Q = 0, C: Q = 0
(F) A: Q = 1, B: Q = 0, C: Q = 1
(G) A: Q = 1, B: Q = 1, C: Q = 0
(H) A: Q = 1, B: Q = 1, C: Q = 1

The questions on this page consider the register file pictured below, with the inputs and outputs as defined in-class:

- A1 and A2 are the two addresses to read from
- A3 is the address to write to
- WE3 is the write enable
- WD3 is the write data input
- RD1 and RD2 are the two read outputs



You may assume that this register file is implemented using **positive edge-triggered D flip-flops**.

Select the inputs below required to write the number 0x45 to register number 0x02. If an input's value does not matter, match it to "any value."

15. (2 points) `A1`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

16. (2 points) `A2`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

17. (2 points) `A3`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

18. (2 points) `WD3`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

19. (2 points) `WE3`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

20. (2 points) `CLK`:

    (A) 1 (B) 0 (C) 0x45 (D) 0x02 (E) Rising Edge (F) Falling Edge (G) Any value

5 Toy ISA

Questions 21-23 consider the following block of memory.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	60	18	64	20	68	22	6C	24	**36**	00	01	02	03	04	05	06
10	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
20	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26

The first 8 bytes of memory load values into the registers such that the state of the machine is

- `pc = 0x08`
- `r0 = 0x18`
- `r1 = 0x20`
- `r2 = 0x22`
- `r3 = 0x24`

The following questions concern the next instruction (boxed above), which will impact the `pc` and one of the registers.

21. (4 points) What is the value of the `pc` after the command executes?

22. (2 points) Which register will change value?

    (A) `r0`
    (B) `r1`
    (C) `r2`
    (D) `r3`

23. (6 points) What will the new value stored be in that register be?

---

24. (5 points) Select the instruction below that stores the value of the program counter in register 2.

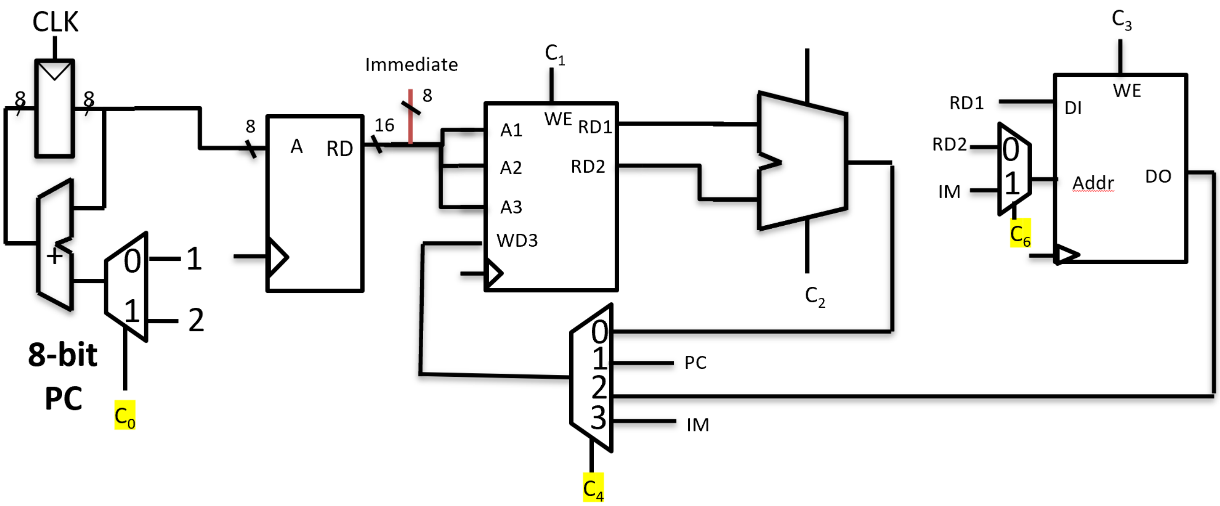
    (A) `0 101 10 11`
    (B) `0 011 10 11`
    (C) `0 110 10 11`
    (D) `0 101 11 11`
    (E) `0 101 11 10`

25. (5 points) Using the example ISA, which of the following lines of code loads the value `0x12` into register 3?

    (A) `0x0A 0x12`
    (B) `0x03 0x12`
    (C) `0x6C 0x12`
    (D) `0x63 0x12`
    (E) `0x6D 0x12`

6 Control signals

Consider the following diagram, which is a subset of our toy ISA architecture.



The following three questions ask you to determine the control signals C_0, C_4, and C_6 when `icode=6` and `b=0`.

`icode`	`b`	Behavior	add to `pc`
6	0	`rA` = read from memory at `pc + 1`	2

26. (3 points) What is C_0?
    - (A) `0`
    - (B) `1`
    - (C) Does not matter

27. (4 points) What is C_4?
    - (A) `0`
    - (B) `1`
    - (C) `2`
    - (D) `3`
    - (E) Does not matter

28. (3 points) What is C_6?
    - (A) `0`
    - (B) `1`
    - (C) Does not matter