1	/18
2	/17
3	/18
4	/19
5	/18
6	/10

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.191 Computation Structures
Fall 2025

Quiz #1

Name	Athena login name	Score

Recitation section

□ WF 10, 34-301 (Jan)	□ WF 2, 34-302 (Varun)	□ WF 12, 34-303 (Nathan)
□ WF 11, 34-301 (Jan)	□ WF 3, 34-302 (Varun)	□ WF 1, 34-303 (Nathan)
□ WF 12, 34-302 (Abdullah)	□ WF 10, 34-302 (Christina)	□ WF 2, 34-303 (Grace)
□ WF 1, 34-302 (Abdullah)	□ WF 11, 34-302 (Christina)	□ WF 3, 34-303 (Grace)
		□ opt-out

Please enter your name, Athena login name, and recitation section above. Enter your answers in the spaces provided below. Show your work for potential partial credit. You can use the extra white space and the back of each page for scratch work.

Problem 1. Digital Abstraction (18 points)

Suppose we define all signaling thresholds in our digital system to be relative to the supply voltage V_{DD}:

$V_{OL} = 0.1V_{DD}$
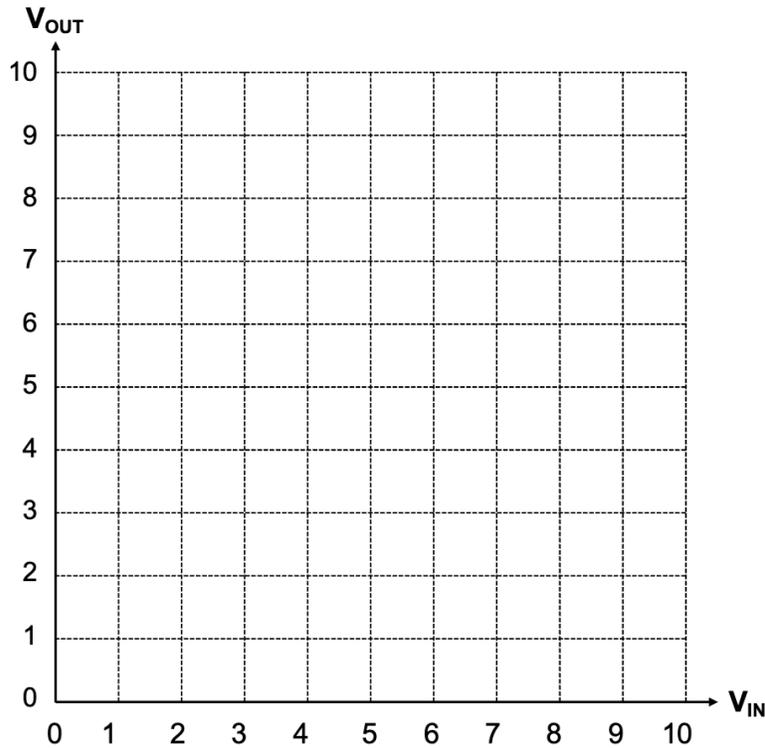$V_{IL} = 0.5V_{DD}$
$V_{IH} = 0.6V_{DD}$
$V_{OH} = 0.8V_{DD}$

Suppose we have some valid digital **inverter** F with supply voltage $V_{DD,F} = 10V$ and propagation delay of 50ns.

(A) (2 points) Given the above specification for inverter F, determine the noise immunity of the device.

Noise Immunity (V): _____

(B) (3 points) Draw one potential voltage transfer curve for F. **Please clearly label all signaling thresholds.**

(C) (3 points) We apply constant input voltage V_{in} to F for 30ns starting at time $t = 0$ns and measure that $V_{out} = 9$V at $t = 25$ns. What can we conclude about V_{in}? Circle the correct answer and provide an explanation.

V_{in} **is (select one):** $V_{in} < 1$V $\quad$ $V_{in} > 5$V $\quad$ $V_{in} < 6$V $\quad$ $V_{in} > 8$V $\quad$ None of the above

Explanation:

(D) (3 points) We apply constant input voltage V_{in} to F for 60ns starting at time $t = 0$ns and measure that $V_{out} = 0.5$V at $t = 50$ns. What can we conclude about V_{in}? Circle the correct answer and provide an explanation.

V_{in} **is (select one):** $V_{in} < 1$V $\quad$ $V_{in} > 5$V $\quad$ $V_{in} < 6$V $\quad$ $V_{in} > 8$V $\quad$ None of the above
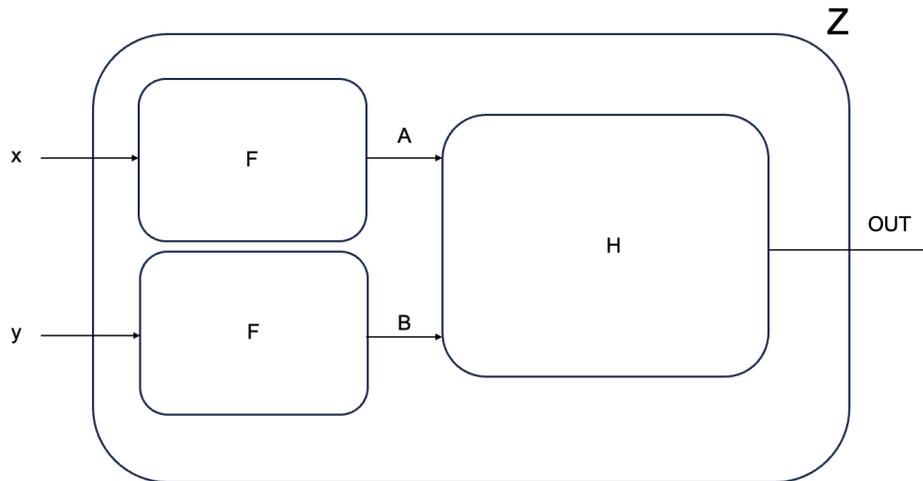
Explanation:

(E) (4 points) Now consider device G with the same signaling thresholds relative to its supply voltage $V_{DD,G}$. In the circuit below, under what range of supply voltages $V_{DD,G}$ will the system work correctly?

```
[G] → [F] → [G] → [F] → [G] → [F]
```
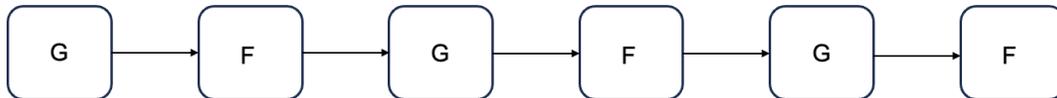
Acceptable Range for $V_{DD,G}$: _____

(F) (3 points) Now consider the following circuit below.

Circuit H follows the voltage transfer characteristic given below.

$$V_{out} = \begin{cases} 8V & (\max(V_A, V_B) \geq 8V) \\ 1V & (\max(V_A, V_B) \leq 1V) \end{cases}$$

Circuit F behaves as it did before.

What Boolean expression does $H(A, B)$ implement? Express your answer in terms of A and B.

Boolean Expression for $H(A, B)$: _____

What gate does Z implement? Express your answer in terms of X and Y.

Gate implemented by Z: _____

Problem 2. Boolean Algebra (17 points)

(A) (5 points) Consider the truth table given below:

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

1. What is the normal form for $F(a, b, c)$?

Normal form for F: _____

2. Find two different minimal sum-of-products expressions for $F(a, b, c)$.

First minimal sum-of-products for F: _____

Second minimal sum-of-products for F: _____

(B) (3 points) Can you construct a circuit for any possible Boolean function using just XOR and AND gates? **If yes, prove it. If no, provide a counterexample**.

(C) (3 points) Implement the function $G = A \oplus B = XOR(A, B)$ using one 2-input OR, one 2-input AND, and one 2-input NAND gate. **Using Boolean algebraic properties, demonstrate that your circuit is equivalent in functionality to an XOR gate**.

(D) (6 points) Determine whether each of the Boolean expressions below are satisfiable. If it is satisfiable, give an input assignment that makes the expression satisfiable. You must provide a valid value for each variable. If it is not satisfiable, show why it is not.

1. Expression $(\overline{a} + b)(\overline{b} + c)(\overline{c} + a)(\overline{c} + d + e)(\overline{d} + f)(\overline{f} + \overline{a} + \overline{b})$

2. Expression $(\overline{a} + b)(\overline{b} + c)(\overline{c} + d)(\overline{d} + e)(a + \overline{e})(a \oplus c + f)(\overline{f})$

Problem 3. CMOS Logic (18 points)

(A) (12 points) For each of the following four functions, determine if it can be implemented as a single CMOS gate. **If it can, draw the complete CMOS implementation using a minimal number of transistors. If not, describe why it cannot be implemented as a single CMOS gate.**

    1. $F(A, B, C, D) = \overline{ABD} + \overline{C}$

    2. $F(A, B, C, D) = \overline{AB + C} \cdot \overline{D}$

    3. $F(A, B, C, D) = \overline{\overline{A} + D} \cdot \overline{B + C}$

4. $F(A, B, C, D) = \overline{D \cdot (AB + AC)}$

(B) (6 points) Given the following truth tables, find whether it is possible for F to be implemented using a single CMOS gate. **If it is possible, draw the CMOS gate using the least number of transistors possible. If it is not possible, explain why**.

1.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

2.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Problem 4. Combinational Minispec (19 points)

(A) (4 points) The following parametric function g#(n) performs a specific operation using a, b and c. We want the function g_copy#(n) to implement g#(n) in a single line of code. Fill in the blanks in g_copy#(n) to make the two functions equivalent. Write a single-line expression that uses the ternary operator (? :).

```
function Bit#(n) g#(Integer n)(Bit#(n) a, Bit#(1) b, Bit#(1) c);
  Bit#(n) ret;
  for(Integer i = 0; i < n; i = i + 1) begin
    ret[i] = a[i] | (b ^ c);
  end
  return ret;
endfunction

function Bit#(n) g_copy#(Integer n)(Bit#(n) a, Bit#(1) b, Bit#(1) c);


  return (_____) ? _____ : _____;


endfunction
```

(B) (4 points) The following parametric function f#(n) performs a specific operation using a. We want the function f_copy#(n) to implement f#(n) in a single line of code. Fill in the blank with a single expression to make f_copy#(n) equivalent to f#(n).

```
function Bit#(n) f#(Integer n)(Bit#(n) a);
  Bit#(n) ret;
  for(Integer i = 0; i < n-1; i = i + 1) begin
    ret[i] = a[i] | a[i+1];
  end
  ret[n-1] = a[n-1] | a[0];
  return ret;
endfunction

function Bit#(n) f_copy#(Integer n)(Bit#(n) a);


  return _____;


endfunction
```

(C) (5 points) Finish the following circuit diagram to implement function h, given below. You may use **at most four** 2-bit 2-to-1 multiplexers (shown below), and any number of constants (0, 1, 2, 3, ...). **When drawing multiplexers clearly specify each of the two inputs, the selector, and the output.**

```
function Bit#(2) h(Bit#(3) a);
  Bit#(2) ret = 2;
  for(Integer i = 0; i < 3; i = i + 1) begin
    if(a[i] == 1) begin
      ret = i ^ 1;
    end
  end
  return ret;
endfunction
```

a[0] →

a[1] →   → ret

a[2] →
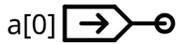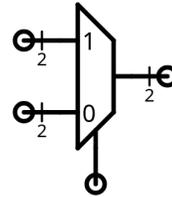
(D) (6 points) Alice wants to send Bob a secret message. For that purpose she implemented the following encode#(n) function in minispec which obfuscates an *n*-bit message. She showed Bob her implementation and now he has to come up with a decode#(n) function which will retrieve the original message (i.e., decode#(n)(encode#(n)(x))==x for all *n*-bit x). **Help Bob implement the decode#(n) function and answer the question about delay below. Make sure to provide a justification for the delay.**

```
function Bit#(n) encode#(Integer n)(Bit#(n) message);
  Bit#(n/2) upper;
  Bit#(n - n/2) lower;
  for(Integer i = 0; i < n; i = i + 1) begin
    if(i % 2 == 0) begin
      lower[i/2] = message[i];
    end else begin
      upper[i/2] = message[i];
    end
  end
  return {upper, lower};
endfunction

function Bit#(n) decode#(Integer n)(Bit#(n) encr_message);
  // Implement Bob's function here



endfunction
```

The delay of Alice's implementation of encode#(n) is $\Theta(\underline{\hspace{4cm}})$

Justification:

Problem 5. Combinational and Sequential Logic Timing (18 points)

The TAs made a new circuit to be part of their Important-Computation-Machine™ but were busy writing this exam so they're asking the 6.191 students for help analyzing and improving their circuit. The gates they used have the following characteristics:

| Gate	t_{PD}	t_{CD}	t_{SETUP}	t_{HOLD}
AND2	1.4ns	0.5ns	—	—
OR2	1.6ns	0.2ns	—	—
XOR2	1.2ns	0.4ns	—	—
INV1	0.7ns	0.2ns	—	—
BUF1	1.0ns	1.0ns	—	—
IREG	0ns	0ns	0ns	0ns
REG	2.5ns	0.6ns	1.0ns	0.4ns

(A) (3 points) What are the propagation and contamination delays for this combinational circuit they create as their first design?
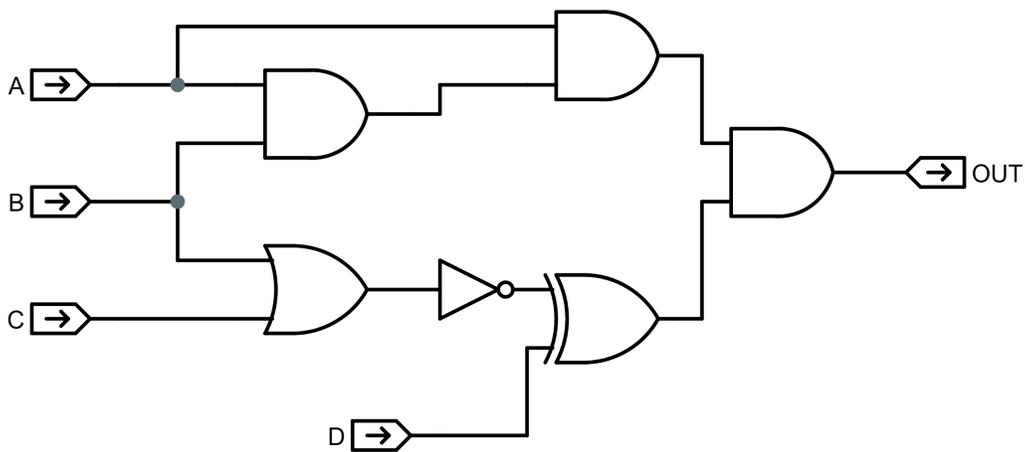
t_{PD} **(ns):** _____

t_{CD} **(ns):** _____

Using the timing constraints from above, they designed a new sequential circuit based on the original combinational circuit. **Note that they used ideal registers, labeled IREG, for the input signals while the other registers are regular registers, labeled REG**.
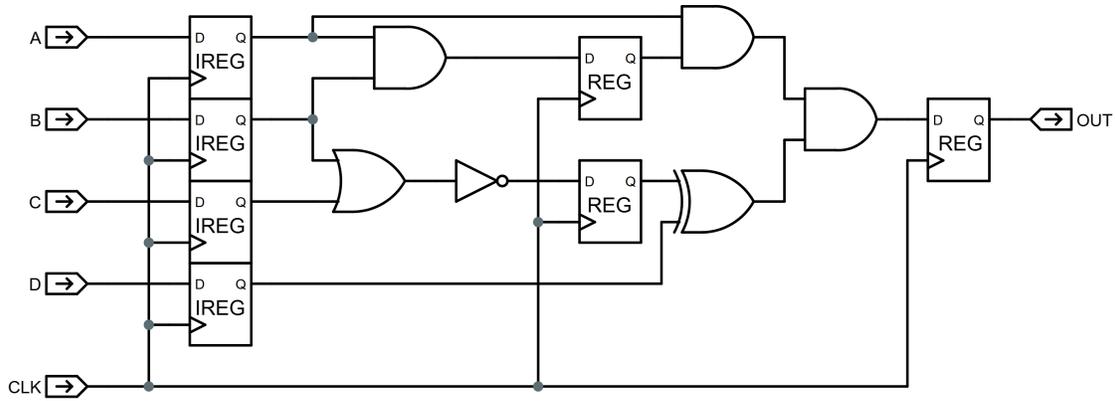
(B) (2 points) Demonstrate that the sequential circuit satisfies all hold time constraints.

(C) (3 points) What is the shortest clock period that can be used for this circuit?

Minimum t_{CLK} (ns): _____

(D) (2 points) What are the t_{PD} and t_{CD} values of the sequential circuit?

t_{PD} **(ns):** _____

t_{CD} **(ns):** _____

(E) (2 points) The TAs would like to switch out their AND gates for a different model, what's the smallest t_{CD} the new AND gates could have such that the circuit is still valid?

t_{CD} **(ns):** _____

(F) (3 points) These TAs are so fickle! After switching their AND gates they now also want to switch the REG registers. Which of the following registers allows them to have the fastest clock frequency while still having a valid circuit? Provide an explanation for your answer. **Note that you should use the new AND gates and that only the REG registers are being changed**. The IREG ideal registers remain the same.

Register	t_{PD}	t_{CD}	t_{SETUP}	t_{HOLD}
REG (current)	2.5ns	0.6ns	1.0ns	0.4ns
REGA	1.4ns	0.7ns	0.5ns	0.5ns
REGB	1.6ns	0.1ns	0.8ns	0.3ns

Select one: REGA   REGB   Neither

Explanation:

(G) (3 points) After experimenting with their sequential circuit, they realized they need to strengthen the clock signal by using a buffer to drive the clock of the rightmost register. Unfortunately, this adds skew to their clock signal. **Using the original AND and REG timing specifications, what is the minimum t_{CLK} the circuit can now use? If the circuit is not valid, then write "Invalid" and provide an explanation for why the circuit is not valid**.
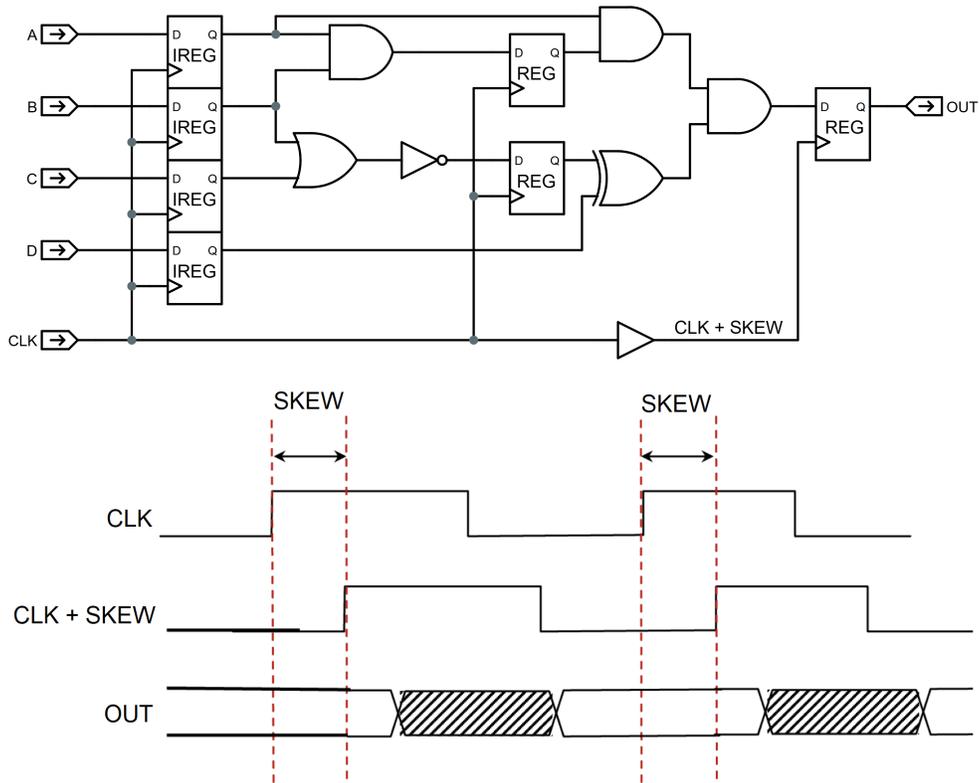
Minimum t_{CLK} (ns) or Invalid + Explanation: _____

Problem 6. Bit Invader Finite State Machine (10 points)

You're designing the logic for a simple enemy AI, the "Bit Invader", for a new retro arcade game. The invader's behavior is determined by a finite state machine that responds to the player's position.

FSM Specifications:
- Inputs (2-bit SR): S (Player is in Sight (S=1) or not in Sight (S=0)), R (Player is in Range (R=1) or not in Range (R=0)).
- Output (1-bit U): U=1 when the invader is attacking, U=0 otherwise.
- Initial State: The FSM's default state is PATROL.

Behavioral Description:
- The invader starts in and remains in the PATROL state as long as the player is unseen. It begins a chase as soon as the player is spotted.
- If visual contact is lost while in a chase, it reverts to PATROL.
- During a chase, the invader will attack if the player moves into close range.
- An attack is a single-cycle action that is always immediately followed by a single-cycle evasive maneuver.
- After the evasive maneuver, the invader resumes its chase if the player is still visible; otherwise, it returns to PATROL.

(A) (5 points) Draw the state transition diagram for the Bit Invader FSM.
- You must determine the minimum number of states required. Give each state a label (e.g., PATROL (P), etc.).
- Label each state with its name and its corresponding output value for U.
- Label each transition arc with the input condition(s) (SR) that cause it.

(B) (4 points) For the following input sequence, determine the state and output for each cycle.

Cycle	1	2	3	4	5	6	7	8
Current State	P							
Input (SR)	10	10	11	00	10	11	10	00
Next State								
Output (W)								

(C) (1 points) How many flip-flops are required to implement this FSM?

Number of flip-flops: _____

END OF QUIZ 1!