1	/14
2	/13
3	/14
4	/15
5	/14
6	/12

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.004 Computation Structures
Updated Spring 2022

Quiz #1

Name	*Athena login name*	*Score*

Recitation section

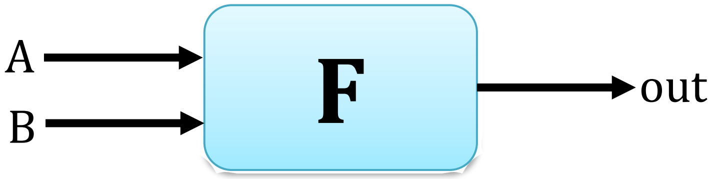
□ WF 10, 34-302 (Francis)
□ WF 11, 34-302 (Francis)
□ WF 12, 34-302 (Grace)
□ WF 1, 34-302 (Grace)
□ WF 2, 34-302 (Robert)
□ WF 3, 34-302 (Robert)
□ WF 10, 35-310 (Sara)
□ WF 11, 35-310 (Sara)
□ WF 12, 35-310 (Kendall)
□ WF 1, 35-310 (Kendall)
□ opt-out

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 backs of the pages for scratch work.

Problem 1. Static Discipline (14 points)

The F module below outputs 0.5V when $0.5 * V_A + V_B > 2V$ for 25ns and outputs 5V when $0.5 * V_A + V_B < 1.5V$ for 25ns. Furthermore, V_A and V_B are both between 0 and 5V. This is summarized in the equation below:

$$V_{out} = \begin{cases} 0.5V, & 0.5 * V_A + V_B > 2V \\ 5V, & 0.5 * V_A + V_B < 1.5V \\ 0 \leq ??? \leq 5V, & otherwise \end{cases}$$



(A) (2 points) If we apply constant V_A, V_B for 25ns and then measure a $\boldsymbol{V_{out} = 5}$, what can we conclude about V_B?

C1: $V_B < 1.5V$
C2: $V_B \leq 2V$
C3: $V_B > 2V$
C4: $V_B \geq 1.5V$
C5: None of the above

If V_{out} = 5 then
$0.5 * V_A + V_B \leq 2V$
$V_B \leq 2V$

Best conclusion about V_B (Circle one): C1 … [✓ C2] … C3 … C4 … C5

(B) (2 points) If we apply constant V_A, V_B for 25ns and then measure a $\boldsymbol{V_{out} = 4}$, what can we conclude about V_A?

C1: $V_A < 3V$
C2: $V_A \leq 4V$
C3: $V_A > 4V$
C4: $V_A \geq 3V$
C5: None of the above

If V_{out} = 4 then
$1.5V \leq 0.5 * V_A + V_B \leq 2V$
$V_A \leq 4V$

Best conclusion about V_A (Circle one): C1 … [✓ C2] … C3 … C4 … C5

(C) (2 points) What Boolean expression does the F module implement? Specify an equation using A and B.

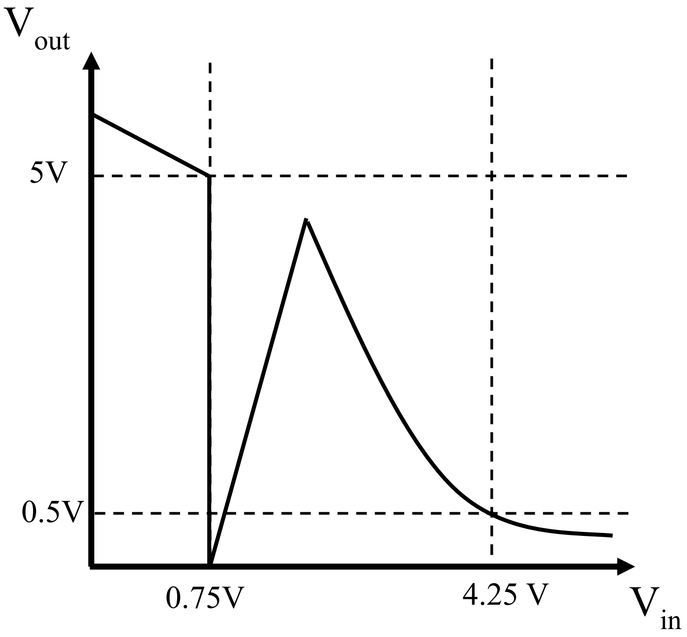
Boolean Expression: out = ____$\overline{\boldsymbol{A + B}}$______________

(D) (4 points) What are the parameters that produce a maximum noise immunity for the F module shown above?

$\boldsymbol{V_{OL}}$ = __**0.5**__ , $\boldsymbol{V_{IL}}$ = __**1**__ , $\boldsymbol{V_{IH}}$ = __**4**__ , $\boldsymbol{V_{OH}}$ = __**5**__

Noise Immunity = _____0.5___________

(E) (4 points) Now suppose we have a new device G, whose voltage transfer characteristic (VTC) is depicted below. We want to use both F and G in the same circuit, but we want to use the same signaling specification for both devices. What are the parameters that will produce a maximum noise immunity while satisfying both devices?

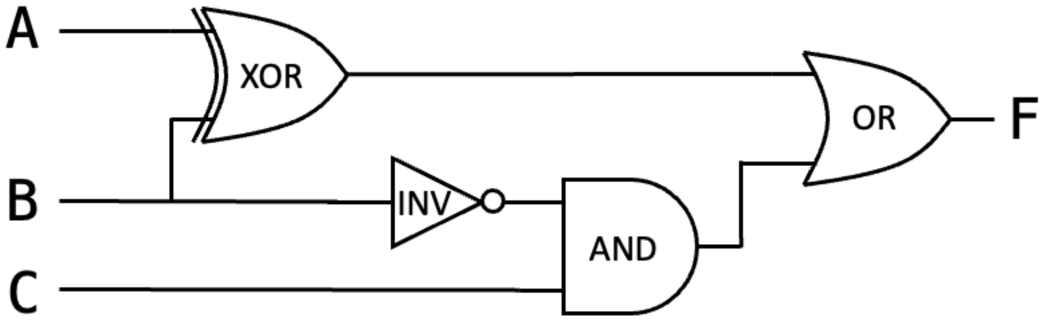


V_{OL} = __0.5__ , V_{IL} = __0.75__ , V_{IH} = __4.25__ , V_{OH} = __5___

Noise Immunity = ___0.25____________

Problem 2. Boolean Algebra and Combinational Logic (13 points)

(A) (2 points) Consider the schematic of the circuit F(A, B, C) below. Derive its truth table.



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	0
1	1	1	0

(B) (3 points) Find the normal form and a minimal sum-of-products expression for G(A, B, C).

A	B	C	G
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

1. **Normal form for G =** ___________ $\bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}c + abc$ __________

2. **Minimal sum of products for G =** _____ $c + \bar{a}\bar{b}$____________________

$$\bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}c + abc = \bar{a}\bar{b}\bar{c} + \bar{a}c + ac = \bar{a}\bar{b}\bar{c} + c = c + \bar{a}\bar{b}$$

(C) (2 points) Is the circuit G(A, B, C) functionally complete? Explain.

F is functionally complete as we can construct a NOR gate. F(A, B, 0) = NOR(A, B).

(D) (4 points) Simplify the following Boolean expressions by finding a minimal sum-of-products expression for each one. (Note: These expressions can be reduced into a minimal SOP by repeatedly applying the Boolean algebra properties we saw in lecture.)

1. $(x+y)(x+\bar{y})(\bar{x}+z)(\bar{x}+\bar{z}) = (x + xy + x\overline{y} + y\overline{y})(\overline{x} + \overline{x}z + \overline{xz} + z\overline{z})$
   $= x\overline{x} = 0$

2. $(x+y+z)(x+\bar{y}+\bar{z}) = x + x\overline{y} + x\overline{z} + xy + y\overline{y} + y\overline{z} + xz + \overline{y}z + z\overline{z}$
   $= x + y\overline{z} + \overline{y}z$

(E) (2 points) For each of the expressions in part D, find a set of variable assignments that makes the expression true or explain why the expression cannot be satisfied with any variable assignment.

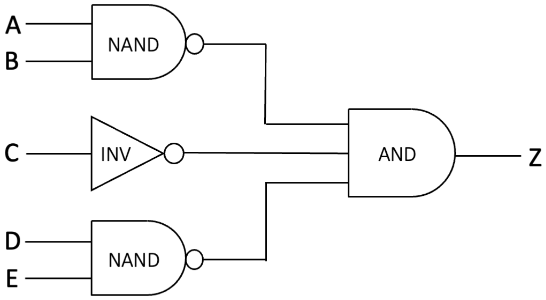
1. $x =$ ____ , $y =$ ____, $z =$ ____ or [✓ UNSATISFIABLE.]

2. $x =$ _1_ , $y =$ _0_, $z =$ _0_ or UNSATISFIABLE.

   1xx, x10, and x01 are all valid solutions.

Problem 3: CMOS (14 points)

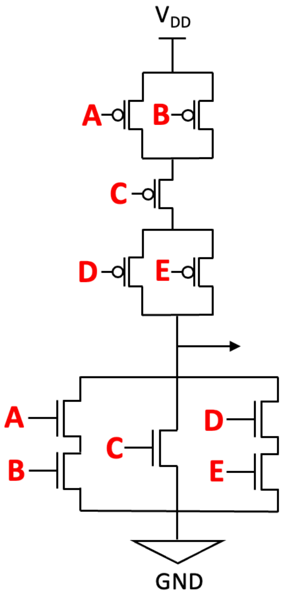
(A) (3 points) The following 5-input circuit consists of two 2-input NAND gates, an inverter, and a 3-input AND gate. Write a Boolean expression for the function Z. If the function Z can be implemented using a **single CMOS gate**, please draw the corresponding single CMOS gate. **For full credit, use a minimum number of FETs.** If it cannot be implemented using a single CMOS gate, **explain why not.**



$Z(A,B,C,D,E) = (\overline{AB})(\bar{C})(\overline{DE}) = (\bar{A}+\bar{B})(\bar{C})(\bar{D}+\bar{E})$

The function Z is inverting (note we can express Z as a function of only the negations of the inputs) so we can implement Z using a single CMOS gate.

Pullup (pFETs): (A in parallel with B) in series with C in series with (D in parallel with E)
Pulldown (nFETs): (A in series with B) in parallel with C in parallel with (D in series with E).



(B) (3 Points) A single CMOS gate, consisting of an output node connected to a single pFET-based pullup circuit and a single nFET-based pulldown circuit (as described in lecture), computes F(A, B, C, D). It is observed that F(0, 1, 1, 1) = 1. What can you say about the following values? **Circle one of the options (0, 1, can't tell).**

a. F(1, 1, 0, 1) = 0 … 1 … [✓ can't tell]
This has both rising and falling inputs compared to the reference we are given, so we can't predict what will happen to the output.

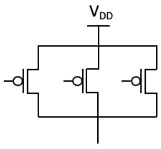
b. F(0, 1, 0, 0) = 0 … [✓ 1] … can't tell
This has falling inputs compared to the reference, so the output must either stay the same or rise. Since the reference output is 1 the output cannot rise, so it must stay the same.

c. F(1, 1, 1, 1) = [✓ 0] … 1 … can't tell
An inverting function (a single CMOS gate can only implement inverting functions) must output 0 when the input is all 1s.

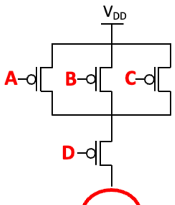
(C) (8 Points) For each of the following Boolean functions, choose the appropriate pullup design, i.e., the one which, properly connected, implements that gate's pullup using the **minimum number of active** transistors (a FET that is never on is not active). You can connect one of the inputs (A, B, C, or D) or a constant (GND, or VDD) to each pFET.

For each expression, circle the name of your chosen pullup design (e.g. PU1) and label each pFET in your chosen design with the appropriate input or constant. You can use a design more than once. **If none of the above pullups implements the function's pullup, circle "NONE".**

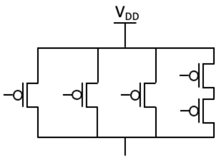
a. $F(A,B,C,D) = (\overline{ABC})\overline{D}$



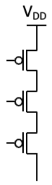
PU1



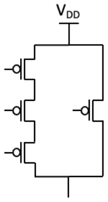
[✓ PU2]



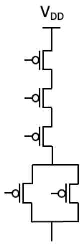
PU3



PU4



PU5



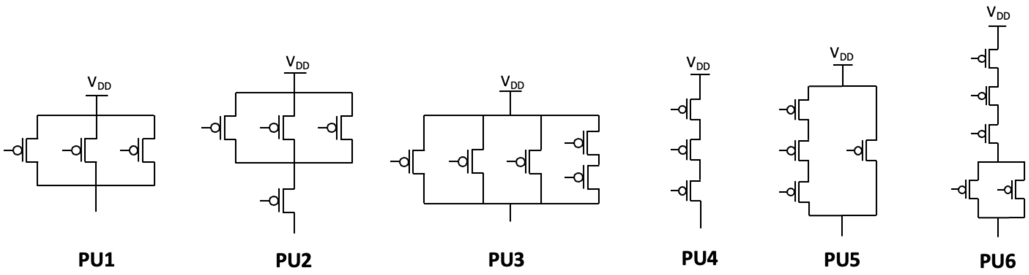
PU6

NONE

$F(A,B,C,D) = (\overline{ABC})\overline{D} = (\bar{A} + \bar{B} + \bar{C})\overline{D}$

The pullup circuit has (A in parallel with B in parallel with C) in series with D.

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

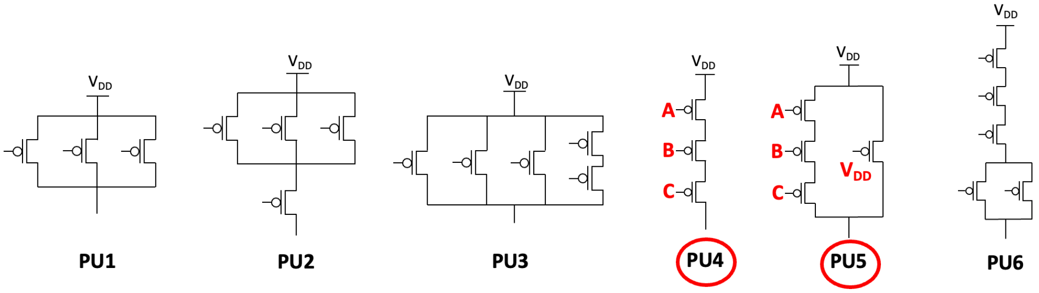


[✓ NONE]

$F(A,B,C,D) = \overline{(A+B)(\bar{C}+\bar{D})} = \overline{A+B} + \overline{\bar{C}+\bar{D}} = \bar{A}\bar{B} + CD$

This function is non-inverting (note the presence of C and D in the expression for F) so the answer is NONE.

c. $F(A,B,C,D) = (\overline{B+C})(\bar{A})(\overline{BD})$

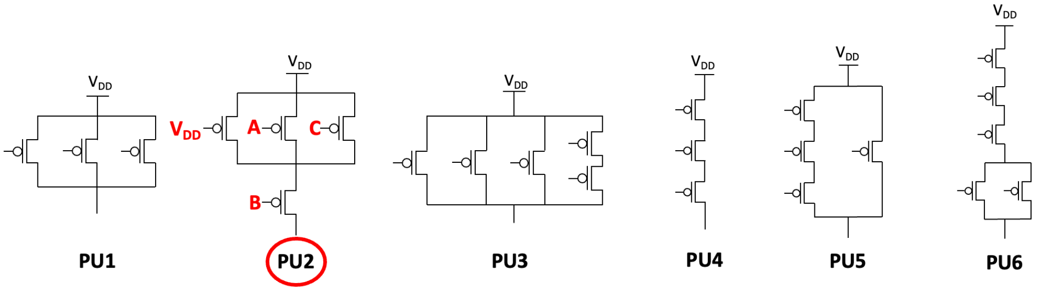


NONE

$(A,B,C,D) = (\overline{B+C})(\bar{A})(\overline{BD}) = \bar{B}\bar{C}\bar{A}(\bar{B}+\bar{D}) = \bar{B}\bar{C}\bar{A}\bar{B} + \bar{B}\bar{C}\bar{A}\bar{D} = \bar{B}\bar{C}\bar{A} + \bar{B}\bar{C}\bar{A}\bar{D} = \bar{B}\bar{C}\bar{A}(1+\bar{D}) = \bar{B}\bar{C}\bar{A}$

The pullup circuit has A, B, and C all in series. This can be implemented using PU4, or using PU5 with the additional parallel pFET connected to V_{DD} (both solutions have the same number of active transistors since the pFET connected to V_{DD} is never on).

d. $F(A,B,C,D) = (\bar{A}+\bar{C})(\bar{B})(D+\bar{D})$



NONE

$(A,B,C,D) = (\bar{A}+\bar{C})(\bar{B})(D+\bar{D}) = (\bar{A}+\bar{C})(\bar{B})$

The pullup circuit has (A in parallel with C) in series with B. This can be implemented using PU2 by connecting the extra parallel pFET to V_{DD}.

Problem 4. Combinational Circuits (15 points)

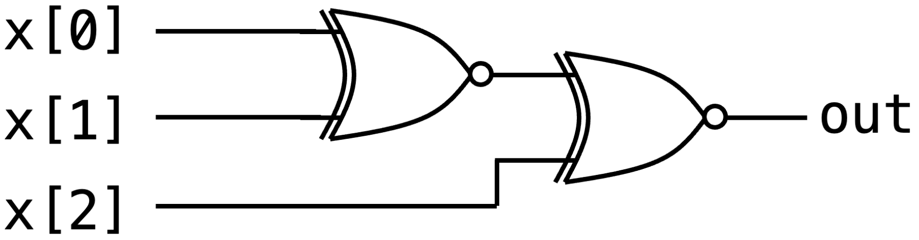
(A) (2 points) Fill out the truth table for the 3-input parity function parity that returns true if an odd number of bits are true.

x[0]	x[1]	x[2]	out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

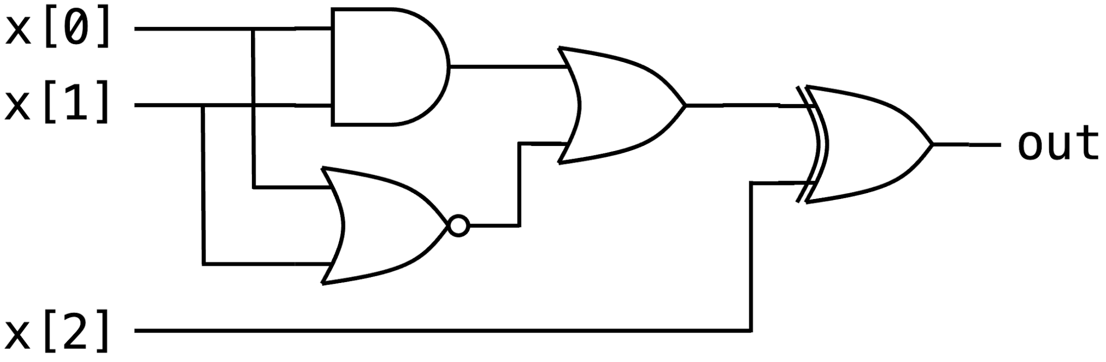
(B) (2 points) Implement the function above in Minispec.

```
function Bit#(1) parity3(Bit#(3) x);

    return x[0] ^ x[1] ^ x[2];

endfunction
```

(C) (6 points) Determine whether each of the following circuits implements the `parity3` function. **If it does not, change one gate in the circuit to have it implement `parity3`.**



Implements the function `parity`: [✓ YES] NO



Implements the function `parity`: YES [✓ NO]

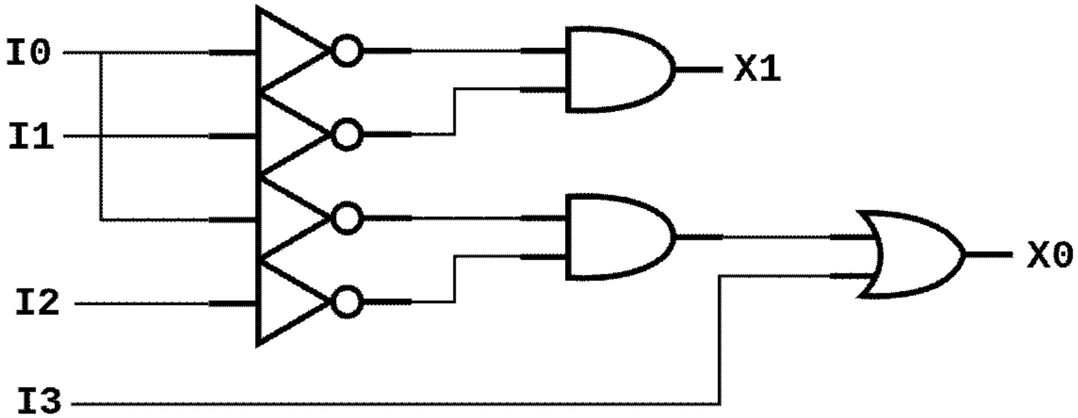
XOR should be replaced with XNOR, or OR should be replaced with NOR.

(D) (5 points) Implement the parametric function `parity#(n)` in Minispec, which computes the parity of an
`n`-bit input, returning 1 if and only if an odd number of bits are 1. **For full credit, `parity#(n)`must have logarithmic delay** with respect to `n`. **`n` is greater than or equal to 1, and not necessarily a power of two.**

```
  function Bit#(1) parity#(Integer n) (Bit#(n) x);

   if (n == 1) begin
       return x
   end else begin
       return parity#(n/2)(x[n/2-1:0]) ^ parity#(n-n/2)(x[n-
1:n/2]);
   end
  endfunction
```

Problem 5. Combinational and Sequential Logic Timing (14 points)

(A) (3 points) Given the timing parameters in the table below, what is the propagation delay of this circuit?

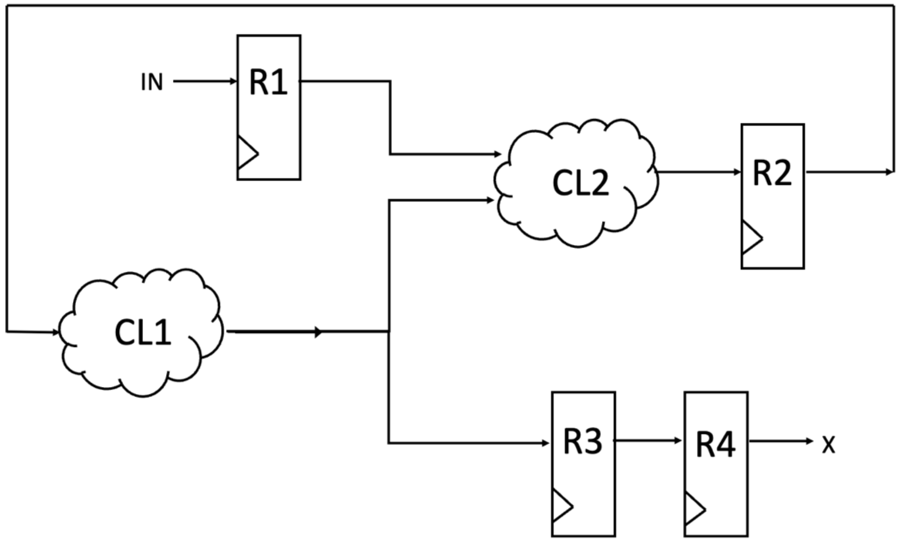


Gate	t_{pd} (ns)	t_{cd} (ns)
OR2	4	1.5
INV	2	1
AND2	3	2

Propagation delay (ns): _____9_____

Longest Path goes through INV + AND2 + OR2 = 2 + 3 + 4 = 9

(B) (6 points) Given the timing parameters in the table below, please indicate whether or not each constraint is satisfied by the circuit below, and explain your answer. All registers are driven by a common clock with period $t_{clk} = 10\mathbf{ns}$.



Component	t_{pd} **(ns)**	t_{cd} **(ns)**	t_{SETUP} **(ns)**	t_{HOLD} **(ns)**
Register (R1, R2, R3, R4)	5.5	1	2	1.5
Combinational Logic (CL1, CL2)	2.5	1	N/A	N/A

Setup time constraint (select one):

SATISFIED [✓ **NOT SATISFIED**]

Explanation:

R2 -> R2: $t_{PD,R2} + t_{PD,CL1} + t_{PD,CL2} + t_{Setup,R2}$ = 5.5 + 2.5 + 2.5 + 2 = 12.5 > t_{CLK} = 10ns so setup constraint is not satisfied.

Hold time constraint (circle one):

SATISFIED [✓ **NOT SATISFIED**]

Explanation:

R3 -> R4: $t_{CD,R3}$ = 1 > $t_{Hold,R4}$ = 1.5 so hold time not satisfied.

(C) (5 points) We would like for the circuit from (B) to be able to function with a clock period of 9ns. Using the replacement parts below, modify the circuit so that it can operate using a clock period of 9ns *while satisfying the setup and hold time constraints.* For full credit, minimize the number of replacements made.

Component	t_{pd} **(ns)**	t_{cd} **(ns)**	t_{SETUP} **(ns)**	t_{HOLD} **(ns)**
Register-New	3	2	2	0.5
CombinationalLogic-New	2	0.5	N/A	N/A

CL1: [✓ **REPLACED (with CombinationalLogic-New)**] **NOT REPLACED**

CL2: [✓ **REPLACED (with CombinationalLogic-New)**] **NOT REPLACED**

R1: [✓ **REPLACED (with Register-New)**] **NOT REPLACED**

R2: [✓ **REPLACED (with Register-New)**] **NOT REPLACED**

R3: **REPLACED (with Register-New)** [✓ **NOT REPLACED**]

R4: [✓ **REPLACED (with Register-New)**] **NOT REPLACED**

Replace R3 instead of R4 is also valid.

To satisfy the setup time on the R2 -> R2 path, we need $t_{PD,R2} + t_{PD,CL1} + t_{PD,CL2} + t_{Setup,R2} <= 9$ns.
Since $t_{Setup,R2}$ is not changing this means that the first 3 terms must sum to less than or equal to 7. The only way to achive this is to replace R2, CL1, and CL2.

Next, we check the R1 -> R2 path which is $t_{PD,R1} + t_{PD,CL2} + t_{Setup,R2} <= 9$ns. So $t_{PD,R1} + t_{PD,CL2}$ must be less than or equal to 7. If we keep the old R1, then this adds up to 9.5 which does not satisfy the necessary constraints, so we also need to replace R1.

With these changes in place the R2 -> R3 and the R3 -> R4 setup time constraints are also satisfied.

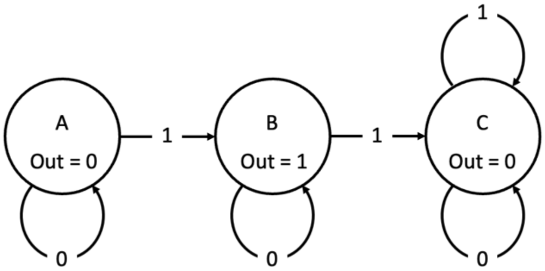
Finally, looking at the R3 -> R4 path, we see that without any changes the hold time constraint is not satisfied because $t_{CD,R3} = 1 < t_{Hold,R4} = 1.5$ns. So, we either need to

replace R3 or R4. Either one will result in satisfying the hold time constraint as well as the setup time constraint.

If we replace R3: $t_{CD,R3} = 2 > t_{Hold,R4} = 1.5$ns.
If we replace R4: $t_{CD,R3} = 1 > t_{Hold,R4} = 0.5$ns.

Problem 6. Finite State Machines (12 points)

Consider the following finite state machine. The initial state is A. The output of B is 1 and the output of all other states is 0.



(A) (3 points) Given the following inputs, determine what the next state will be after the last input is processed. Assume the FSM starts at initial State A for each series of inputs.

(i) 00010

Final state: _______B___________

(ii) 010000001

Final state: _______C___________

(iii) 0101010100111111011101110101

Final state: _______C___________

(B) (3 points) For the second sequence of inputs in the question above (copied below), determine what the outputs are for each cycle. Note that the FSM starts in State A in Cycle 1.

Cycle	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Input	0	1	0	0	0	0	0	0	1
Output	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Current State	**A**	**A**	**B**	**B**	**B**	**B**	**B**	**B**	**B**

(C) (4 points) To encode the states, two bits S_1 and S_0 are used. States A, B, and C are encoded with bits 00, 01, and 10, respectively. Determine truth table for the FSM.

S_1^t	S_0^t	**input**	S_1^{t+1}	S_0^{t+1}	**output**
0	0	0	**0**	**0**	**0**
0	0	1	**0**	**1**	**0**
0	1	0	**0**	**1**	**1**
0	1	1	**1**	**0**	**1**
1	0	0	**1**	**0**	**0**
1	0	1	**1**	**0**	**0**

(D) (2 points) Find the minimal sum-of-products expression for the output.

output = ________$\overline{S_1^t}S_0^t$ **or** S_0^t____

END OF QUIZ 1!