EECS 270 *Final Exam*

Spring 2023

Name: Key unique name: Key 2

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

Key

NOTES:

1. ***The exam is open book and open class notes plus anything you wrote yourself. You may not use any device capable of communication (cell phones, calculators with wireless, etc.)***
2. You have about 120 minutes for the exam total.
3. Some questions may be harder than others. Manage your time wisely.
4. If you need additional room for an answer, the last 2 pages are blank and you may use them as part of your answer. Just be sure to indicate you've done so on the page of the question.

1. Fill in each blank or circle the best answer. **[13 points, -2 per wrong or blank answer, min 0]**

   a) The 5-bit 2's complement representation of -12 is 10100.

   b) If a given function has A, B, and C as inputs, there are 2 maxterms in the equation A+!B.

   c) You are building a memory where each address references 8 bits of data. You are using a 1024 by 1024 memory array to do so. You need 17 address bits, and the column MUX select input requires 7 of those address bits.

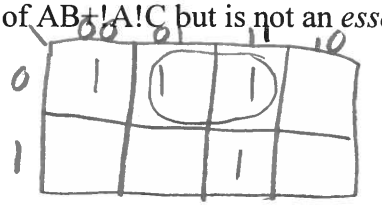
   d) SRAM cells tend to be ***larger / smaller*** than DRAM cells.

   e) "4A.C" base 16 is equal to 74.75 base 10.

   f) If a 4-bit ripple-carry adder (RCA) has a worst-case delay of 1ns, you would expect a 256-bit RCA to have a delay of *about* ***4 / 20 / 60 / 120 / 1024*** ns.

   g) If a 4-bit carry look-ahead adder (CLA) has a worst-case delay of 1ns, you would expect a 256-bit cascaded CLA (with a base size of 4-bits, as done in class) to have a delay of *about* ***4 / 20 / 30 / 60 / 1024*** ns.

   h) $B\bar{C}$ is a prime implicant of AB+!A!C but is not an *essential* prime implicant.



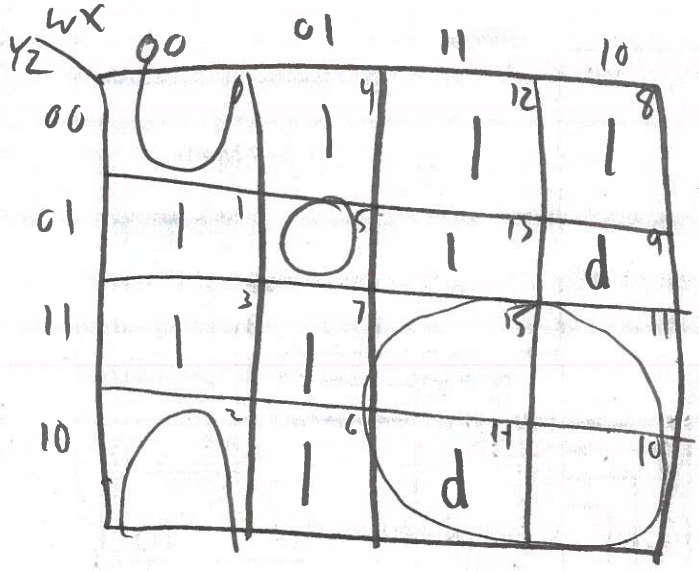
2. Say you have an external signal attached to the D input of a flip-flop. That signal is not synchronized to the local clock. And say that the flip-flop has a set-up time of 2ns, a hold time of 3ns, and a clock-to-Q time of 4ns. **[4 points]**

   a) If the D flip-flop's clock is 20 MHz, what is the probability of a timing violation? 10%

   b) If the D flip-flop's clock is 100MHz, what is the probability of a timing violation? 50%

Basic theme: you have a timing violation if the data changes during setup or hold time. If the clock is 50ns and the hold and setup time cover 5ns, that's 10%.

3. Using a Kmap, find the minimal *product-of-sums* for $A=\Sigma_{(W,X,Y,Z)}=(1, 3, 4, 6, 7, 8, 12, 13) + d(9, 14)$. Clearly show your work. **[6 points]**



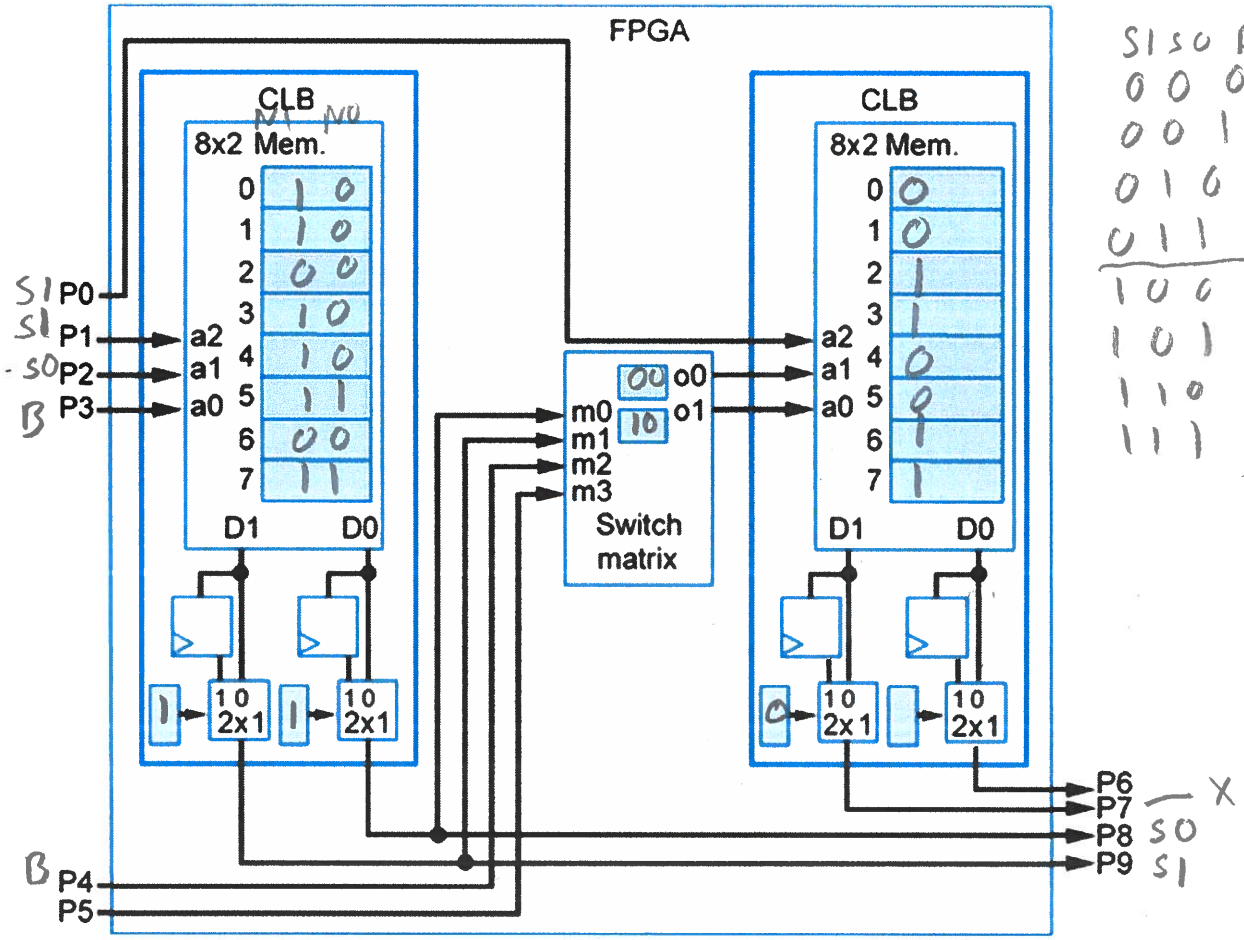
$$\bar{A} = WY + \bar{W}\bar{X}\bar{Z} + \bar{W}X\bar{Y}Z$$

$$A=(\bar{W}+\bar{Y})(W+X+Z)(W+\bar{X}+Y+\bar{Z})$$

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
S1	S1	S0	B	B				S0	S1

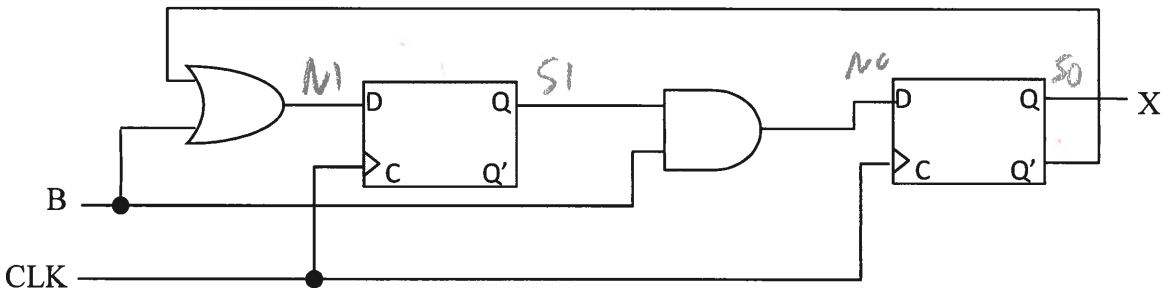
$N1 = B + \overline{S0}$

$N0 = S1 \cdot B$



S1	S0	B	N1	N0	X
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

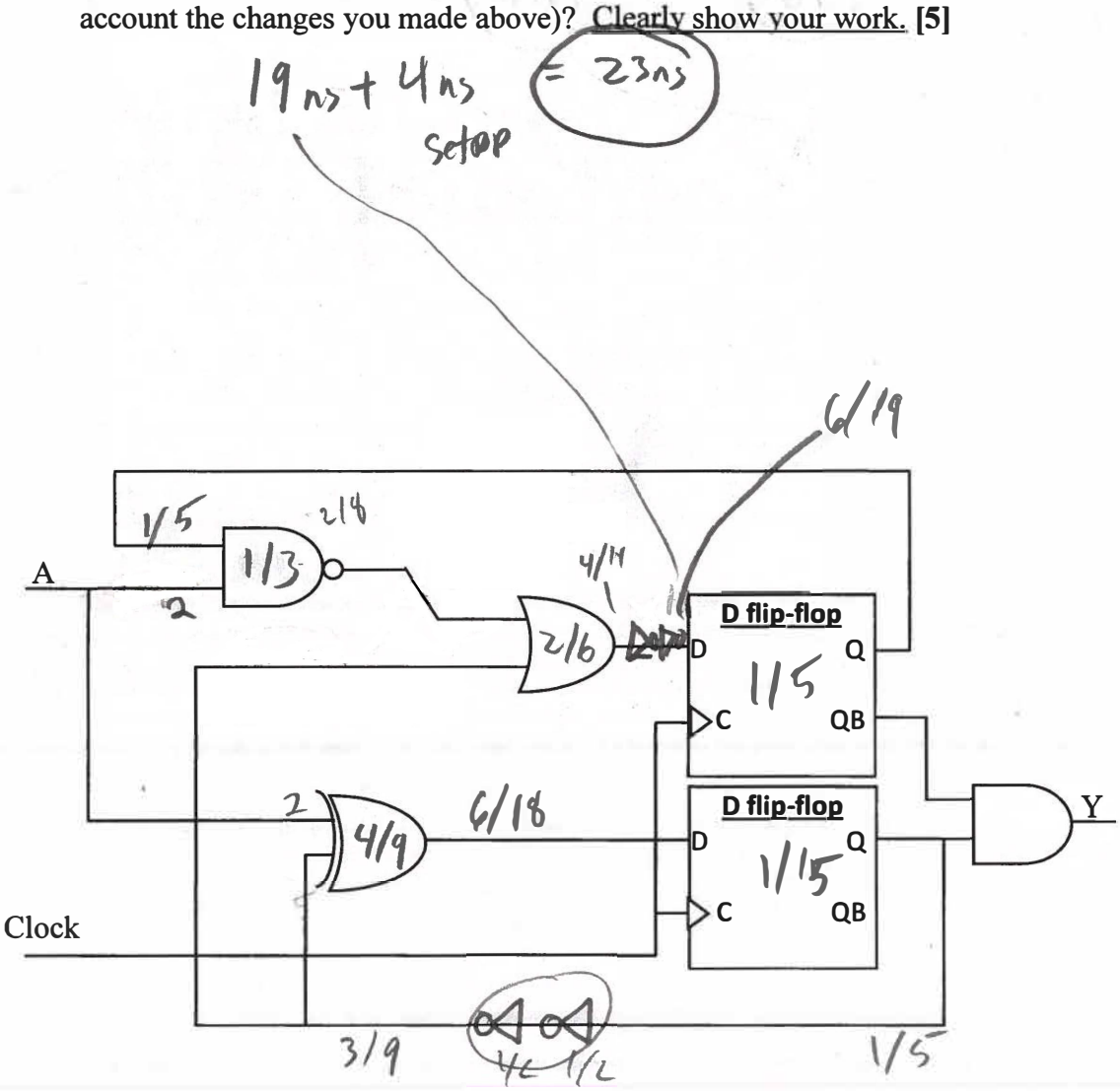
4. Fill in blank boxes as needed to implement the following circuit including the ones on the top. Leave boxes blank if their values don't matter. You may use temporary values to connect outputs back into inputs. Be sure to use the names given (B, X) for inputs and outputs. **[11 points]**



	Min	Max
OR/AND	2ns	6ns
NOR/NAND	1ns	3ns
NOT	1ns	2ns
XOR/XNOR	4ns	9ns

DFF:		**Min**	**Max**
	Clock to Q	1ns	5ns
	Set-up time	4 ns	
	Hold time	6 ns	

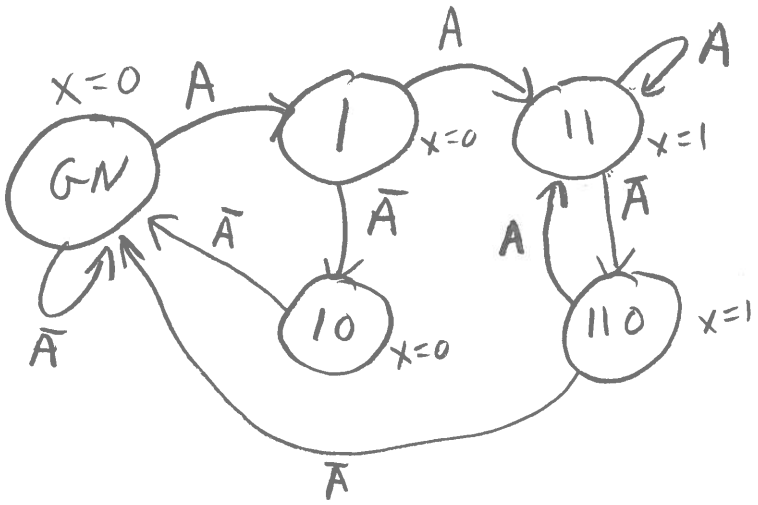
5. Assuming the input A ***always*** arrives 2ns after the rising edge of the clock, answer the following questions. **[10 points]**
   a) Where do inverter pairs need to be added needed to ensure that there are no hold time violations? Add them in a way that maximized the frequency this circuit can be clocked at. *If there is more than one way to add inverter pairs which maximizes the frequency of the circuit, add as few as possible.* You may not make any changes other than adding inverter pair(s). **[5]**
   b) What is the lowest clock period that could be safely used to clock this circuit (after taking into account the changes you made above)? Clearly show your work. **[5]**



6. Design a Moore-type state-transition diagram which has one input A and one output X. X should go high iff A has been high for at least two cycles (total, not in a row) since either reset (when the FSM starts) or the last time A has been low for two cycles in a row. For example:

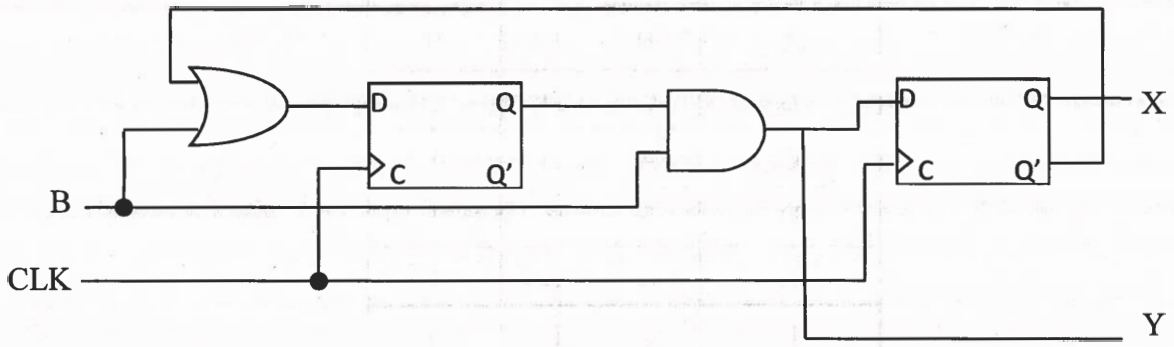
   A=01010101001001001010101
   X=00011111100000000011111

   For full credit you must use as few states as possible. **[10 points]**



X=1 in
11 & 110

7. Write a Verilog module, Bob, which implements the following circuit. The inputs are B and Clock. The outputs are X and Y. The code must be correct, clear, and reasonably concise to get full points. **[10 points]**

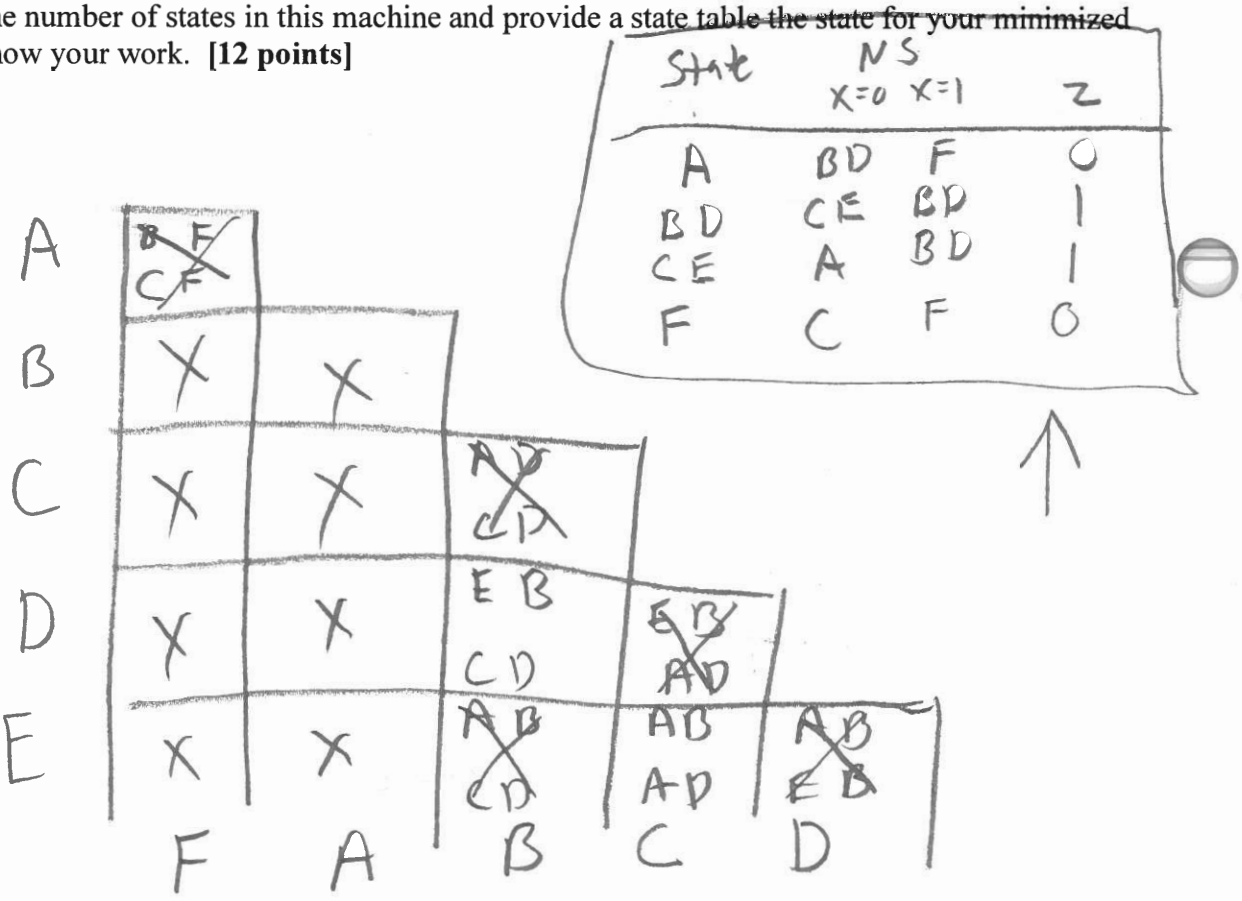



```
module Bob(
                input   B,
                input   CLK,
                output Y,
                output X
);

reg   Q0, Q1;
assign  Y = Q0 & B;
assign  X = Q1;

always@(posedge CLK) begin
        Q0 <= B | ~X;
        Q1 <= Y;
end

endmodule
```

8. Consider the following state table.

Present state	Next state x=0	Next state x=1	Output Z
A	B	F	0
B	C	D	1
C	A	D	1
D	E	B	1
E	A	B	1
F	C	F	0

Minimize the number of states in this machine and provide a state table ~~the state for your minimized~~ machine. Show your work. **[12 points]**

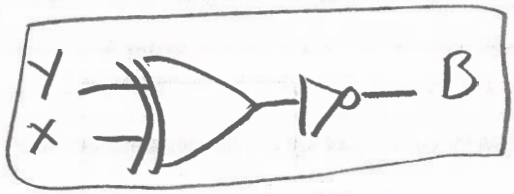


9. Implement the following circuits. Use the given names for inputs and outputs. **[14 points]**

a) Provide a circuit which implements the following logic using only 2-input AND/OR/XOR gates and inverters. For full credit you must use as few 2-input gates as possible. All inputs and outputs are one bit. **[4]**

```
assign B=y?x:!x;
```

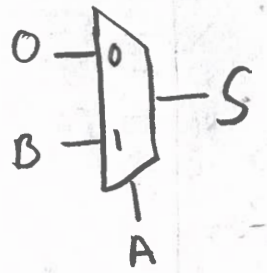
Y	X	B
0	0	1
0	1	0
1	0	0
1	1	1

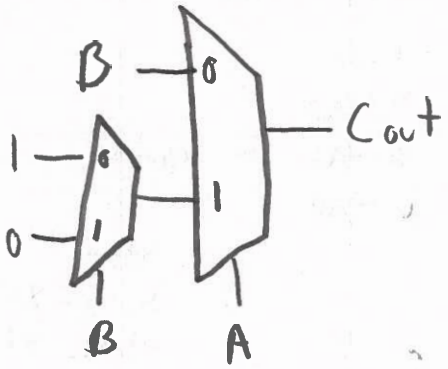


b) Provide a circuit which implements a half-adder (inputs A, B; outputs S, Cout) using only 2-to-1 MUXes. For full credit you must uses as few MUXes as possible. **[5]**

$S = A \cdot B$

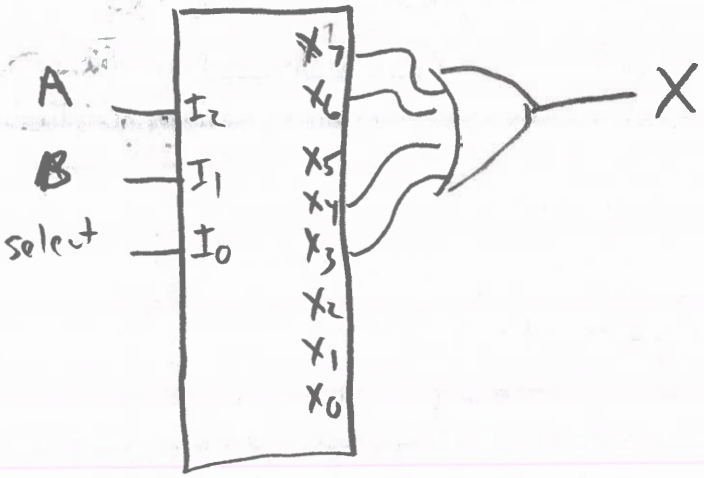
$Cout = A \oplus B$





c) Provide a circuit which implements a 2-to-1 MUX (inputs A, B, and select; outputs X) using only a decoder and one other standard gate (any number of inputs). **[5]**

A	B	Select	X
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



Assume X=A if select=0

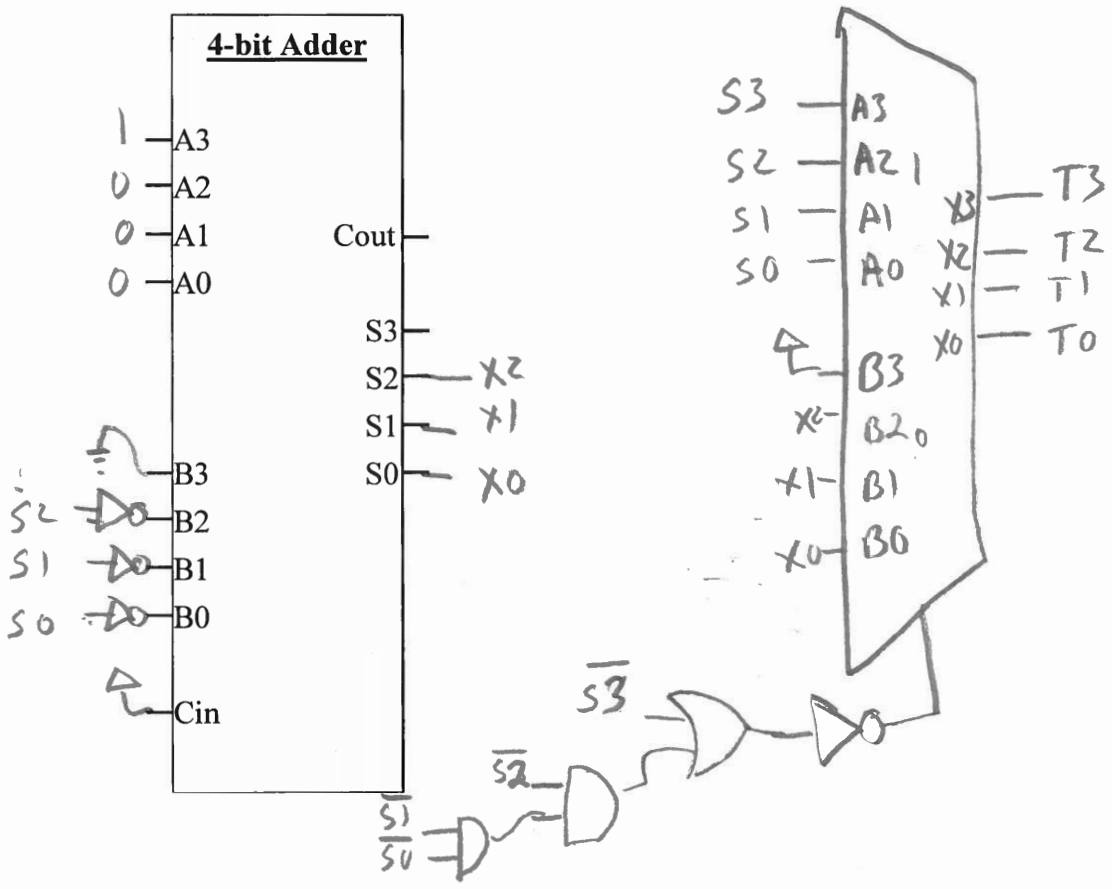
3, 4, 6, 7

10. Create a circuit which takes a 4-bit signed-magnitude number and converts it into a 4-bit 2's complement number. You may use a 4-bit adder, a 4-bit 2 to 1 MUX, and up to 3 standard 2-input gates and any number of inverters. The 4-bit adder has been drawn for you. You may freely make use of ground and power. **[10 points]**

This converter takes its input, S[3:0] and treats it as a signed-magnitude number.

The output, T[3:0] is a 4-bit 2's complement number.

1 = power
0 = Ground



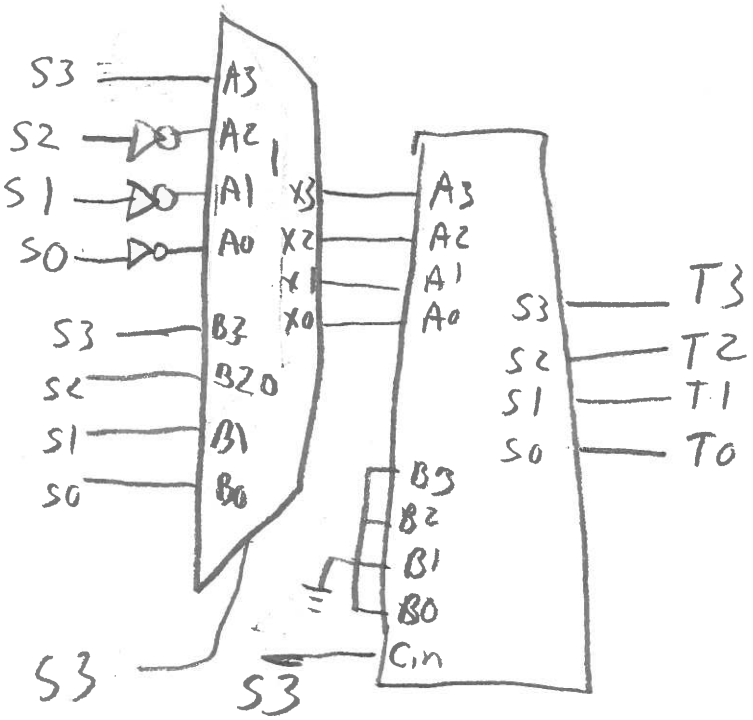
If S3=0 T=S
If S3=1 T3=1
T[2:0] = 8 - S[2:0]

S 1111 = -7 = 1001
S 1001 = -1 = 1111
S 1000 = 0 = 0000 ← hard

Other answer on page 12.

This page is intentionally left blank. See the cover page for an explanation.

This page is intentionally left blank. See the cover page for an explanation.



examples — T

S 0000 → 0000 + 0 = 0000

S 1100 → 1011 + 1 = 1100

S 1110 → 1001 + 1 = 1010

S 1000 → 1111 + 1 = 0000 ✓