Name:_____

Honor Pledge: I am adhering to the Honor Code while taking this test.

Signature:_____ Date:_____

25 points total. (Don't miss the questions on the back of the last page.)

1. Represent each of the following base-10 numbers in 8-bit two's complement.

A. -23

+23 is 00010111, so -23 is 11101001 (=-128+64+32+8+1)

B. 17

00010001

2. Convert 10011011 to base 10, interpreting it as

A. unsigned binary

128+16+8+2+1 = 155

B. 8-bit two's complement

-128+16+8+2+1 =-101

3. The trustees of Karnaugh University (Alice, Bob, Carol, and David) are voting on whether the MW section of Digital Electronics is better than the TT section. The result of the vote is determined by extremely peculiar rules:

If an even number of trustees vote in favor of MW, MW is determined to be the better section.

If Alice, Bob, and Carol all vote the same way, and David votes the opposite way, then TT is determined to be the better section.

In all other cases, the trustees don't even care which section is determined to be better.

A. Complete this truth table, where 1 as an input is a vote for MW as the better section, and 1 as an output means the MW was determined to be the better section. X as an output represents "don't care."

A	B	C	D	MW
0	0	0	0	1
0	0	0	1	0
0	0	1	0	X
0	0	1	1	1
0	1	0	0	X
0	1	0	1	1
0	1	1	0	1
0	1	1	1	X
1	0	0	0	X
1	0	0	1	1
1	0	1	0	1
1	0	1	1	X
1	1	0	0	1
1	1	0	1	X
1	1	1	0	0
1	1	1	1	1

B. Write out a K map equivalent to the truth table.

		CD			
		00	01	11	10
AB	00	1	0	1	X
	01	X	1	X	1
	11	1	X	1	0
	10	X	1	X	1

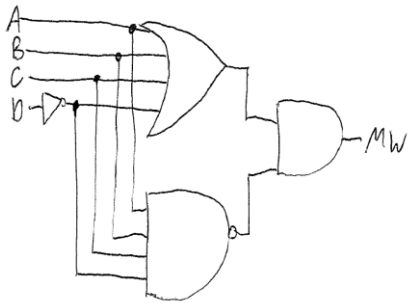
C. Write out any logic equation that implements the truth table. You may use as many gates as you like.

~MW = ~A&~B&~C&D|A&B&C&~D

so MW = ~(~A&~B&~C&D|A&B&C&~D) = ~(~A&~B&~C&D)&~(A&B&C&~D) =
(A|B|C|~D)&~(A&B&C&~D)

Alternatively, MW = ~(A^B^C^D): a single XNOR gate!

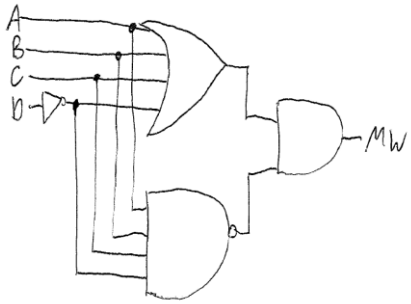
D. Write out the circuit diagram equivalent to the logic equation.



E. Write out an equivalent logic equation that requires at most four **standard** gates with any number of inputs. (Standard gates: NOT, AND, OR, NAND, NOR, XOR, XNOR). Your responses to C and E may be identical.

MW = (A|B|C|~D)&~(A&B&C&~D)

F. Write out the circuit diagram equivalent to your response to E. Your responses to D and F may be identical.



4.

If Edmund or Peter votes 1, it's a vote to go through the wardrobe. Susan and Lucy decide whether Edmund's vote or Peter's vote will determine the group's action.

If Susan votes 1, she is voting to support Peter's vote; if she votes 0, she is voting to support Edmund's vote.

Lucy makes the final decision. If she votes 1, she is voting to support Susan's selection (of either Edmund or Peter's vote). If she votes 0, she is voting to reverse Susan's selection (and choose the other boy's vote).

A. Complete this truth table.

Edmund	Peter	Susan	Lucy	Narnia
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

B. Write out a K map equivalent to the truth table.

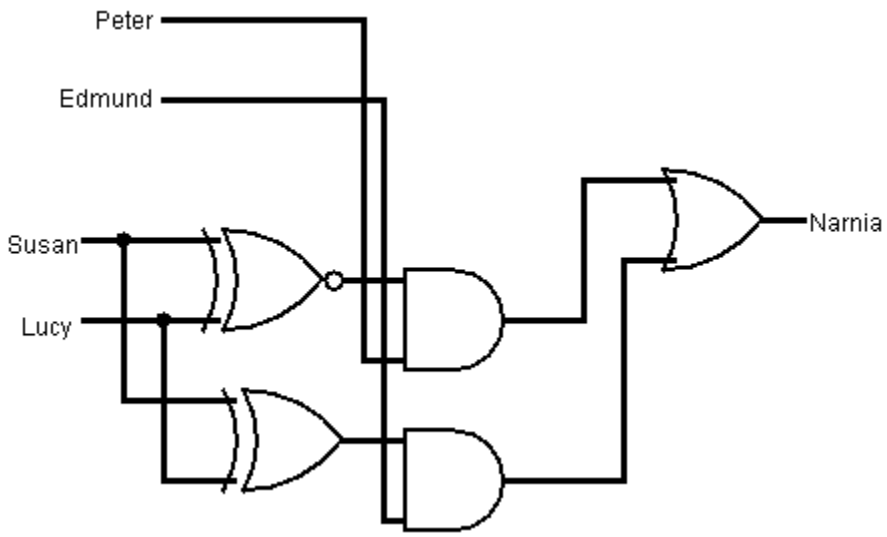
		SL			
		00	01	11	10
EP	00	0	0	0	0
	01	1	0	1	0
	11	1	1	1	1
	10	0	1	0	1

C. Write out any logic equation that implements the truth table. You may use as many gates as you like.

Narnia = ~S&~L&P|~S&L&E|S&L&P|S&~L&E = E&P|P&~(S^L)|E&(S^L)

(Both results are correct, though one does not follow from the other.)

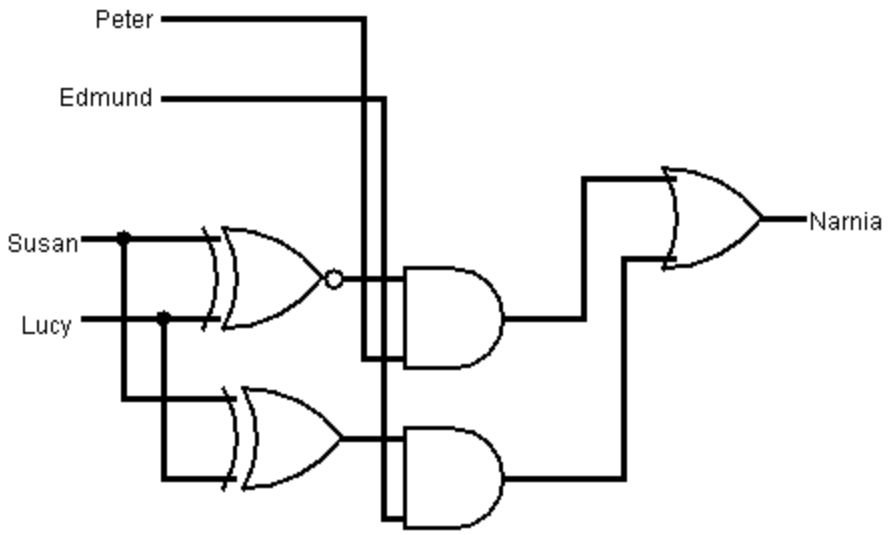
D. Draw the circuit diagram equivalent to your response to part C.



E. Write out an equivalent logic equation that requires at most six **standard** gates with any number of inputs. (Standard gates: NOT, AND, OR, NAND, NOR, XOR, XNOR). Your responses to C and E may be identical.

Narnia = P&~(S^L)|E&(S^L)

F. Write out the circuit diagram equivalent to your response to E. Your responses to D and F may be identical.



G. (Three points.) Without using an always block, write a Verilog module to implement the circuit.

module partG (input E, input P, input S, input L, output Narnia);

assign Narnia = P&~(S^L)|E&(S^L);

endmodule

H. (Five points.) Without using any assign statements, write a Verilog module to implement the circuit. Hint: Use one or more if statements (inside an always block, of course).

module partH (input E, input P, input S, input L, output reg Narnia);

always @(*)

if (S==1&L==1|S==0&L==0) Narnia=P;

else Narnia=E;

endmodule

I. Now show how to implement the circuit using a single multiplexer and a single XOR gate. Please specify which multiplexer input is selected by a 0 on the select line, and which is selected by a 1.