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 numbers in 8-bit two's complement.

A. -79

+79 = 01001111, so -79 = 10110001 (=-128+32+16+1)

B. 103

01100111

2. Convert 00011011 to base 10, interpreting it as

A. unsigned binary

27

B. 8-bit two's complement

27

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

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

If Alice and Bob vote the same way as each other, while Carol and David vote the same way as each other but opposite to Alice and Bob, MW 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 TT as the better section, and 1 as an output means the TT was determined to be the better section. X as an output represents "don't care."

A	B	C	D	TT
0	0	0	0	X
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	X
0	1	1	0	X
0	1	1	1	1
1	0	0	0	1
1	0	0	1	X
1	0	1	0	X
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	X

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

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

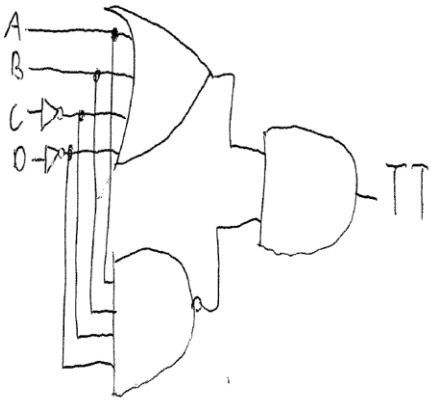
C. Write out any logic equation that implements the truth table. You may use as many gates as you like.

~TT = ~A&~B&C&D|A&B&~C&~D,

so TT = ~(~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: A^B^C^D: one XOR gate!

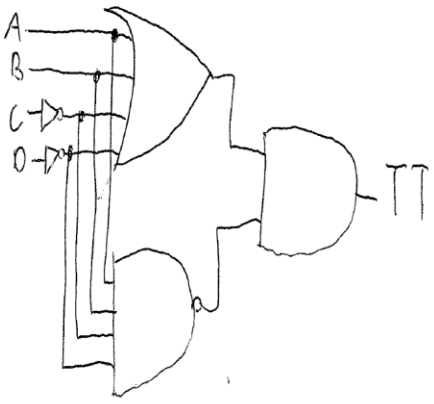
D. Write out the circuit diagram equivalent to the logic equation.



E. Write out an equivalent logic equation that requires at most five **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.

TT = (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 Han Solo or Chewbacca votes 1, it's a vote to go to Endor. Luke Skywalker and Princess Leia decide whether Han Solo's vote or Chewbacca's vote will determine the group's action.

If Luke Skywalker votes 1, he is voting to support Han Solo's vote; if he votes 0, he is voting to support Chewbacca's vote.

Princess Leia makes the final decision. If she votes 0, she is voting to support Luke's selection (of either Han Solo's vote or Chewbacca's vote). If she votes 1, she is voting to reverse Luke's selection (and choose the other vote; for example, if Luke supports Han Solo, Leia instead chooses to make Chewbacca's vote the ultimate choice).

A. Complete this truth table.

Leia	Luke	Han Solo	Chewbacca	Endor
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

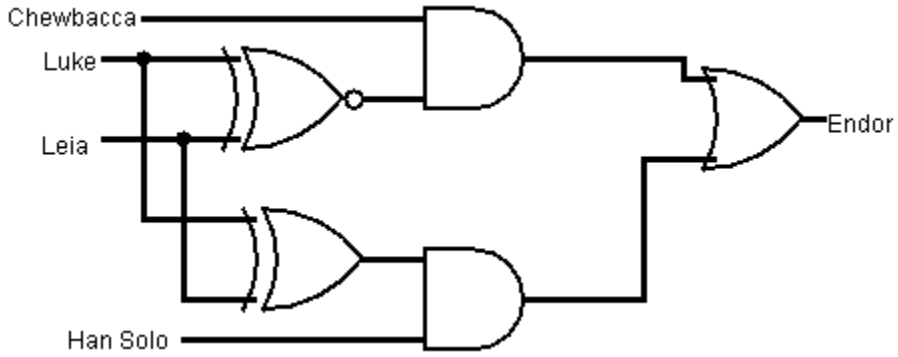
B. Write out a K map equivalent to the truth table.

		HC			
		00	01	11	10
LeLu	00	0	1	1	0
	01	0	0	1	1
	11	0	1	1	0
	10	0	0	1	1

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

Endor = ~Le&~Lu&C|~Le&Lu&H|Le&Lu&C|Le&~Lu&H = C&~(Le^Lu)|H&(Le^Lu)

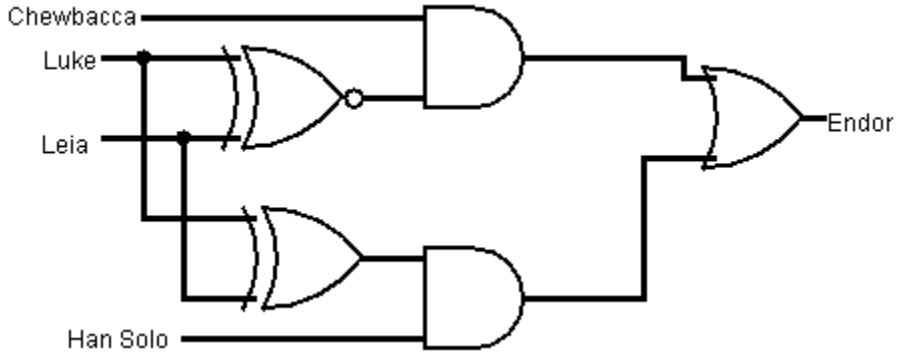
D. Draw the circuit diagram equivalent to your response to part C.



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.

Endor = C&~(Le^Lu)|H&(Le^Lu)

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 Le, input Lu, input H, input C, output Endor);

assign Endor = C&~(Le^Lu)|H&(Le^Lu);

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 partG (input Le, input Lu, input H, input C, output reg Endor);

always @ (*)

if ((Le==1)&(Lu==1)|(Le==0)&(Lu==0)) Endor = C;

else Endor = H;

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.