HW 1

Due: Wednesday, May 15, 2019 at 23:00

CSID 1: d6a2b

CSID 2: b9i2b

Instructions:

1. Do not change the problem statements we are giving you. Simply add your solutions by editing this latex document.

2. If you need more space, add a page between the existing pages using the `\newpage` command.

3. Where possible, include formatting to clearly distinguish your solutions from the given problem text (e.g. use a different font colour for your solutions).

4. Export the completed assignment as a PDF file for upload to gradescope.

5. On Gradescope, upload only **one** copy per partnership. (Instructions for uploading to Gradescope are posted on the assignments page of the course website.)

6. Late submissions will be accepted up to 24 hours past the deadline with a penalty of 20% of the assignment's maximum value.

Academic Conduct: I certify that my assignment follows the academic conduct rules for of CPSC 121 as outlined on the course website. As part of those rules, when collaborating with anyone outside my group, (1) I and my collaborators took no record but names away, and (2) after a suitable break, my group created the assignment I am submitting without help from anyone other than the course staff.

Version history:

- 2019-05-08 16:15 – parenthesis added in Q1a; Q3 gate limit relaxed slightly
- 2019-05-05 20:31 – Initial version for release

1. **[8 marks]** Apply the rules of logical equivalence to reduce the following expressions to the indicated number of literals. Recall that a literal is one Boolean variable or its negation. Show your intermediate steps.

   (a) $\sim p \wedge q \wedge (\sim s \vee (\sim r \wedge s)) \vee (q \wedge (p \vee (\sim p \wedge r \wedge s)))$, to 1 literal

   (b) $((w \wedge x) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim(w \wedge y)$, to 4 literals

1(a).
Proof:
$\equiv \sim p \wedge q \wedge (\sim s \vee (\sim r \wedge s)) \vee (q \wedge (p \vee (\sim p \wedge r \wedge s)))$ [Start]
$\equiv \sim p \wedge q \wedge (\sim s \vee (\sim r \wedge s)) \vee (q \wedge ((p \vee \sim p) \wedge (p \vee r) \wedge (p \vee s)))$ [DIST]
$\equiv (\sim p \wedge q \wedge ((\sim s \vee \sim r) \wedge (\sim s \vee s))) \vee (q \wedge ((p \vee \sim p) \wedge (p \vee r) \wedge (p \vee s)))$ [DIST]
$\equiv (\sim p \wedge q \wedge ((\sim s \vee \sim r) \wedge T)) \vee (q \wedge ((p \vee \sim p) \wedge (p \vee r) \wedge (p \vee s)))$ [NEG]
$\equiv (\sim p \wedge q \wedge ((\sim s \vee \sim r) \wedge T)) \vee (q \wedge (T \wedge (p \vee r) \wedge (p \vee s)))$ [NEG]
$\equiv (\sim p \wedge q \wedge (\sim s \vee \sim r)) \vee (q \wedge (T \wedge (p \vee r) \wedge (p \vee s)))$ [I]
$\equiv (\sim p \wedge q \wedge (\sim s \vee \sim r)) \vee (q \wedge (p \vee r) \wedge (p \vee s))$ [I]
$\equiv (q \wedge \sim p \wedge (\sim s \vee \sim r)) \vee (q \wedge (p \vee r) \wedge (p \vee s))$ [COM]
$\equiv q \wedge ((\sim p \wedge (\sim s \vee \sim r)) \vee ((p \vee r) \wedge (p \vee s)))$ [DIST]
$\equiv q \wedge ((\sim p \wedge (\sim s \vee \sim r)) \vee (p \vee (r \wedge s)))$ [DIST]
$\equiv q \wedge ((\sim p \wedge \sim(s \wedge r)) \vee (p \vee (r \wedge s)))$ [DM]
$\equiv q \wedge (\sim(p \vee (s \wedge r)) \vee (p \vee (r \wedge s)))$ [DM]
$\equiv q \wedge (\sim(p \vee (r \wedge s)) \vee (p \vee (r \wedge s)))$ [COM]
$\equiv q \wedge T$ [NEG]
$\equiv q$ [I]
□

1(b).
Proof:
$\equiv ((w \wedge x) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim(w \wedge y)$ [Start]
$\equiv (((w \wedge x) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z))) \vee \sim w \vee \sim y$ [DM]
$\equiv (((w \wedge x) \vee (\sim w \wedge \sim x)) \vee \sim w \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [DIST]
$\equiv (((w \wedge x) \vee (\sim w \wedge \sim x) \vee \sim w) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [ASS]
$\equiv (((w \wedge x) \vee \sim w \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [COM]
$\equiv ((((w \vee \sim w) \wedge (x \vee \sim w)) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [DIST]
$\equiv (((T \wedge (x \vee \sim w)) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [NEG]
$\equiv ((((x \vee \sim w)) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [I]
$\equiv ((((\sim w \vee x)) \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [COM]
$\equiv ((\sim w \vee x \vee (\sim w \wedge \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [ASS]
$\equiv ((\sim w \vee (x \vee \sim w \wedge x \vee \sim x)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [DIST]
$\equiv ((\sim w \vee (x \vee \sim w \wedge T)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [NEG]
$\equiv ((\sim w \vee (x \vee \sim w)) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [I]
$\equiv ((\sim w \vee x \vee \sim w) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [ASS]
$\equiv ((\sim w \vee \sim w \vee x) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [COM]
$\equiv ((\sim w \vee x) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim w) \vee \sim y$ [ID]
$\equiv ((\sim w \vee x) \wedge \sim w \vee ((\sim y \wedge \sim z) \vee (y \wedge z))) \vee \sim y$ [COM]
$\equiv \sim w \vee (x \wedge ((\sim y \wedge \sim z) \vee (y \wedge z))) \vee \sim y$ [DIST]
$\equiv \sim w \vee (x \vee \sim y \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim y)$ [DIST]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z)) \vee \sim y)$ [ASS]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee (y \wedge z) \vee \sim y))$ [ASS]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee ((y \vee \sim y) \wedge (z \vee \sim y))))$ [DIST]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee (T \wedge (z \vee \sim y))))$ [NEG]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee ((z \vee \sim y))))$ [I]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \wedge \sim z) \vee z \vee \sim y))$ [ASS]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \vee z) \wedge (\sim z \vee z) \vee \sim y))$ [ASS]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \vee z) \wedge (\sim z \vee z) \vee \sim y))$ [DIST]
$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \vee z) \wedge T \vee \sim y))$ [NEG]

$\equiv \sim w \vee ((x \vee \sim y) \wedge ((\sim y \vee z) \vee \sim y))$ [I]
$\equiv \sim w \vee ((x \vee \sim y) \wedge (\sim y \vee z \vee \sim y))$ [ASS]
$\equiv \sim w \vee ((x \vee \sim y) \wedge (z \vee \sim y \vee \sim y))$ [COM]
$\equiv \sim w \vee ((x \vee \sim y) \wedge (z \vee \sim y))$ [COM]
$\equiv \sim w \vee (x \wedge z) \vee \sim y$ [DIST]

□

2. **[12 marks]** Here is a list of four propositional forms and two circuits. This list contains three pairs of logically equivalent entries (we will say a circuit is equivalent to a propositional form if the propositional form describes the value of the circuit's output for all possible combinations of input values). For instance, maybe $A \equiv B$, $C \equiv D$, and $E \equiv F$). First determine the three pairs, and then prove for each pair that one element of the pair is logically equivalent to the other one.

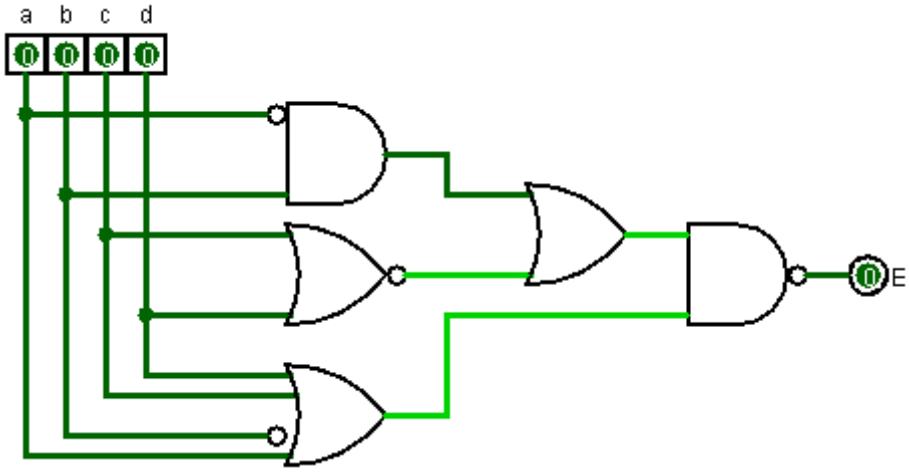
(A) $(\sim a \wedge b) \leftrightarrow \sim(c \vee d)$.

(B) $(\sim a \vee b) \wedge (\sim a \vee c) \wedge (\sim a \vee \sim d)$

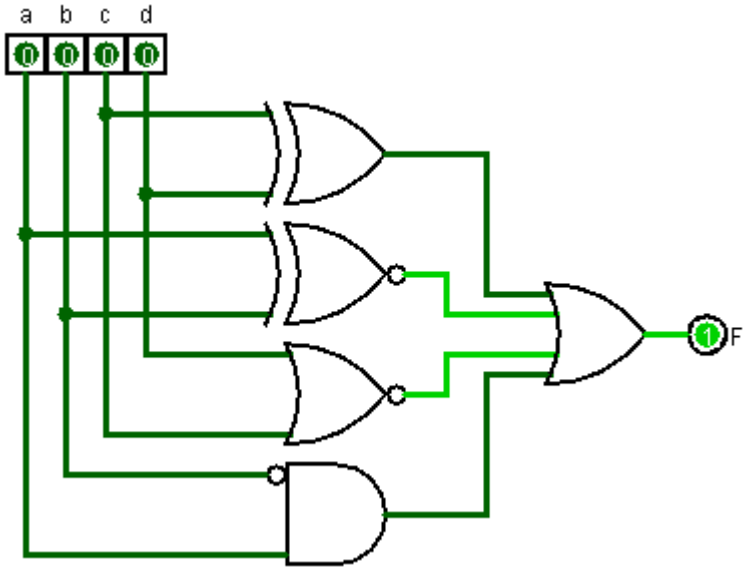
(C) $(a \wedge b \wedge c) \vee \sim(b \wedge c \wedge d)$.

(D) $a \rightarrow (b \wedge c \wedge \sim d)$.

(E)



(F)



You are allowed to use truth tables to figure out the equivalences; however at least two of your proofs must use a sequence of known logical equivalences (see Epp table 2.1.1, or Dave's formula sheet; you can also assume that $x \rightarrow y \equiv \sim x \vee y$ and that $x \oplus y \equiv (x \vee y) \wedge \sim(x \wedge y) \equiv (\sim x \wedge y) \vee (x \wedge \sim y)$. The third proof can use either a sequence of known logical equivalences, or a truth table.

Hint: you might want to translate the circuits into formulas that mirror exactly as the circuit is implemented, before using any logical equivalence rules.

(a) _____≡_____

expression 1		expression 2	reason
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		

(b) _____≡_____

expression 1		expression 2	reason
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		
	≡		

(c) _____≡_____

2(a). B $\equiv D$
$expression1, expression2, reason$
$(\sim a \vee b) \wedge (\sim a \vee c) \wedge (\sim a \vee \sim d) \equiv a \rightarrow (b \wedge c \wedge \sim d)$ [Start]
$(\sim a \vee b) \wedge (\sim a \vee c) \wedge (\sim a \vee \sim d) \equiv \sim a \vee (b \wedge c \wedge \sim d)$ [IMP]
$\sim a \vee (b \wedge c \wedge \sim d) \equiv \sim a \vee (b \wedge c \wedge \sim d)$ [DIST]

2(b). A $\equiv E$
$expression1, expression2, reason$
$(\sim a \wedge b) \leftrightarrow \sim(c \vee d) \equiv \sim((a \vee \sim b \vee c \vee d) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [Start]
$\sim((\sim a \wedge b)(+) \sim(c \vee d)) \equiv \sim((a \vee \sim b \vee c \vee d) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [Equivalence definition]
$\sim(\sim((\sim a \wedge b) \wedge \sim(c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d))) \equiv \sim((a \vee \sim b \vee c \vee d) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [Xor definition]
$\sim(\sim((\sim a \wedge b) \wedge \sim(c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d))) \equiv \sim((\sim(\sim a \wedge b) \vee c \vee d) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [DM]
$\sim(\sim((\sim a \wedge b) \wedge \sim(c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d))) \equiv \sim((\sim(\sim a \wedge b) \vee (c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [ASS]
$\sim(\sim((\sim a \wedge b) \wedge \sim(c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d))) \equiv \sim(\sim((\sim a \wedge b) \wedge \sim(c \vee d)) \wedge ((\sim a \wedge b) \vee \sim(c \vee d)))$ [DM]

3(c). C $\equiv$ F
Truth table of C:

a	b	c	d	$(a \wedge b \wedge c)$	$\sim(b \wedge c \wedge d)$	$f \equiv (a \wedge b \wedge c) \vee \sim(b \wedge c \wedge d)$
1	1	1	X	1	X	1
X	0	X	X	X	1	1
X	X	0	X	X	1	1
X	X	X	0	X	1	1

Since the statement is separated by an Or gate, we can find what makes either terms true and ignore the other term. That lets us deduce the truth table quickly. Unmentioned input combination(s) is false.

Truth table of F:

a	b	c	d	$w \equiv (c(+)d)$	$x \equiv \sim(a(+)b)$	$y \equiv \sim(c \vee d)$	$z \equiv (a \wedge \sim b)$	$f = w \vee x \vee y \vee z$
0	0	0	0	0	1	1	0	1
0	0	0	1	1	1	0	0	1
0	0	1	0	1	1	0	0	1
0	0	1	1	0	1	0	0	1
0	1	0	0	0	0	1	0	1
0	1	0	1	1	0	0	0	1
0	1	1	0	1	0	0	0	1
0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	0	1	1
1	0	1	1	0	0	0	1	1
1	1	0	0	0	1	1	0	1
1	1	0	1	1	1	0	0	1
1	1	1	0	1	1	0	0	1
1	1	1	1	0	1	0	0	1

These two truth tables' output values are the same for given input value (ie only 0111 input makes output 0), therefore the two logical statements are equivalent.

3. **[14 marks]** In this problem we will explore circuit design with limited resources by taking advantage of *functionally complete* sets of logic gates. A set of logic gates is functionally complete if it is able to simulate all of the operations $\{AND, OR, NOT\}$. Here, we will show that $\{NOR\}$ alone is functionally complete.

   For reference, a 2-input NOR operation on inputs a and b is written as $a \downarrow b$, and is equivalent to $\sim(a \vee b)$.

   (a) Using rules of logical equivalence, show that the 2-input NOR operation can be used to simulate a 1-input NOT operation.

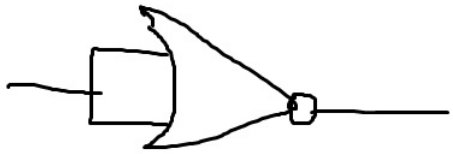
   (b) Using rules of logical equivalence, show that two 2-input NOR operations can be used to simulate a 2-input OR operation.

   (c) Using a circuit diagram with accompanying truth table, show that multiple 2-input NOR gates (with non-inverted inputs) can be used to simulate a 2-input AND gate. Be sure to label your input and output signals.

(d) Below is a truth table for a system with 4 inputs. Design a circuit to implement the function, using only 2-input NOR gates with non-inverted inputs. A modest penalty will be applied to solutions using 16 or more gates. Hint: begin with a circuit using other gates, that can be easily converted to NOR gates using the double-negative law and DeMorgan's laws.

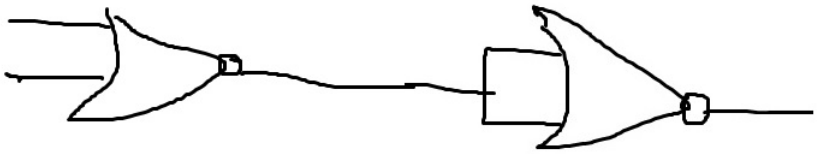
x_3	x_2	x_1	x_0	f
F	F	F	F	F
F	F	F	T	F
F	F	T	F	T
F	F	T	T	F
F	T	F	F	T
F	T	F	T	F
F	T	T	F	F
F	T	T	T	F
T	F	F	F	F
T	F	F	T	F
T	F	T	F	T
T	F	T	T	F
T	T	F	F	T
T	T	F	T	T
T	T	T	F	F
T	T	T	T	F

3(a).



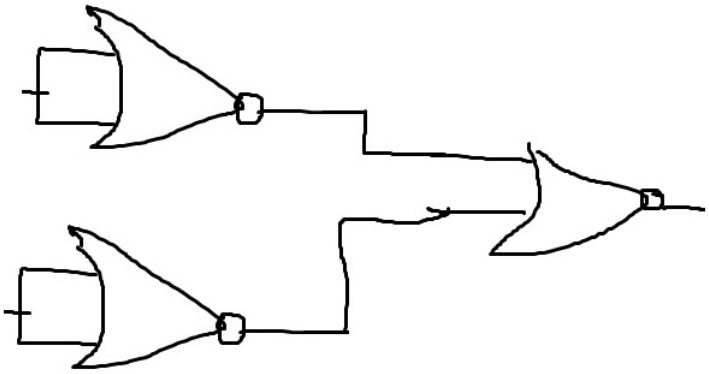
Proof:
$\equiv \sim(a \vee a)$ [Start]
$\equiv \sim(a)$ [ID]
$\equiv \sim a$ [ASS]
□

3(b).



Proof:
$\equiv \sim(\sim(a \vee b) \vee \sim(a \vee b))$ [Start]
$\equiv \sim(\sim(a \vee b))$ [ID]
$\equiv (a \vee b)$ [DNEG]
□

3(c).



a	b	$\sim(a \vee a) \equiv \sim a$	$\sim(b \vee b) \equiv \sim b$	$\sim(\sim(a \vee a) \vee \sim(b \vee b)) \equiv \sim(\sim a \vee \sim b)$	$a \wedge b$
0	0	1	1	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	1	1

3(d).
Since the question didn't specify how we should simplify the truth table, I will use

K-map for simplicity:

–	$x1 \wedge x0$	$\sim x1 \wedge x0$	$\sim x1 \wedge \sim x0$	$x1 \wedge \sim x0$
$x3 \wedge x2$	0	1	1	0
$\sim x3 \wedge x2$	0	0	1	0
$\sim x3 \wedge \sim x2$	0	0	0	1
$x3 \wedge \sim x2$	0	0	0	1

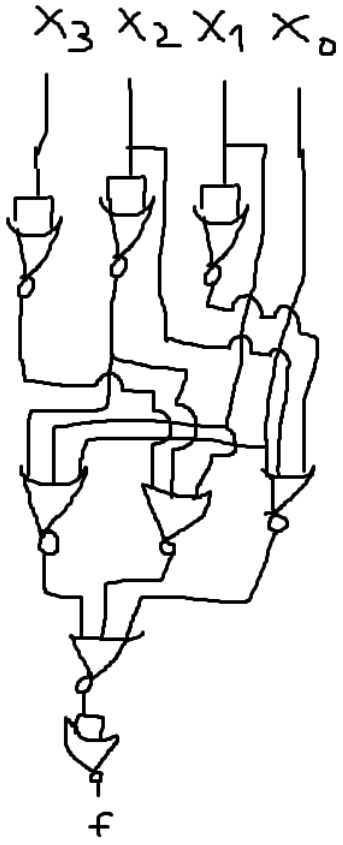
Finding sum of minterms, I can get:
$f \equiv (x2 \wedge \sim x1 \wedge \sim x0) \vee (x3 \wedge x2 \wedge \sim x1) \vee (\sim x2 \wedge x1 \wedge \sim x0)$

I can use logical properties to change the logical statement into one that's made up of only Nor gates:

Proof:
$\equiv (x2 \wedge \sim x1 \wedge \sim x0) \vee (x3 \wedge x2 \wedge \sim x1) \vee (\sim x2 \wedge x1 \wedge \sim x0)$ [Start]
$\equiv \sim(\sim x2 \vee x1 \vee x0) \vee \sim(\sim x3 \vee \sim x2 \vee x1) \vee \sim(x2 \vee \sim x1 \vee x0)$ [DM to all And terms]
$\equiv \sim(\sim(x2 \vee x2) \vee x1 \vee x0) \vee \sim(\sim(x3 \vee x3) \vee \sim(x2 \vee x2) \vee x1) \vee \sim(x2 \vee \sim(x1 \vee x1) \vee x0)$ [ID to all Not terms]
$\equiv \sim(\sim(\sim(\sim(x2 \vee x2) \vee x1 \vee x0) \vee \sim(\sim(x3 \vee x3) \vee \sim(x2 \vee x2) \vee x1) \vee \sim(x2 \vee \sim(x1 \vee x1) \vee x0)))$ [DNEG]

□

This logical statement can be simplified further, but it already uses under 16 gates, as shown here:



4. **[8 marks]** Design a circuit that takes five inputs x_4, x_3, x_2, x_1, x_0 representing an unsigned binary number in the range of $[0, 31]$ and a single output z which is `true` if (and only if) the binary number specified by the inputs is a prime number. Assume that 0 and 1 are not prime, and 2 is prime. For instance, your circuit should output `true` in the following cases:

   - $x_4 =$ `false`, $x_3 =$ `false`, $x_2 =$ `false`, $x_1 =$ `true`, $x_0 =$ `false` (decimal 2).
   - $x_4 =$ `true`, $x_3 =$ `false`, $x_2 =$ `false`, $x_1 =$ `true`, $x_0 =$ `true` (decimal 19).
   - $x_4 =$ `true`, $x_3 =$ `true`, $x_2 =$ `true`, $x_1 =$ `false`, $x_0 =$ `true` (decimal 29).

   but it should output false in these cases:

   - $x_4 =$ `false`, $x_3 =$ `false`, $x_2 =$ `false`, $x_1 =$ `false`, $x_0 =$ `true` (decimal 1).
   - $x_4 =$ `false`, $x_3 =$ `true`, $x_2 =$ `true`, $x_1 =$ `false`, $x_0 =$ `false` (decimal 12).
   - $x_4 =$ `true`, $x_3 =$ `true`, $x_2 =$ `false`, $x_1 =$ `false`, $x_0 =$ `true` (decimal 25).

   Justify your answer! You may use any gates available in Logisim, with any number of inputs and/or inverted inputs. A modest penalty will be applied to solutions using 11 or more gates.

4.
I know that in the range of [0, 31] the prime numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31. I can write a truth table from these outputs.

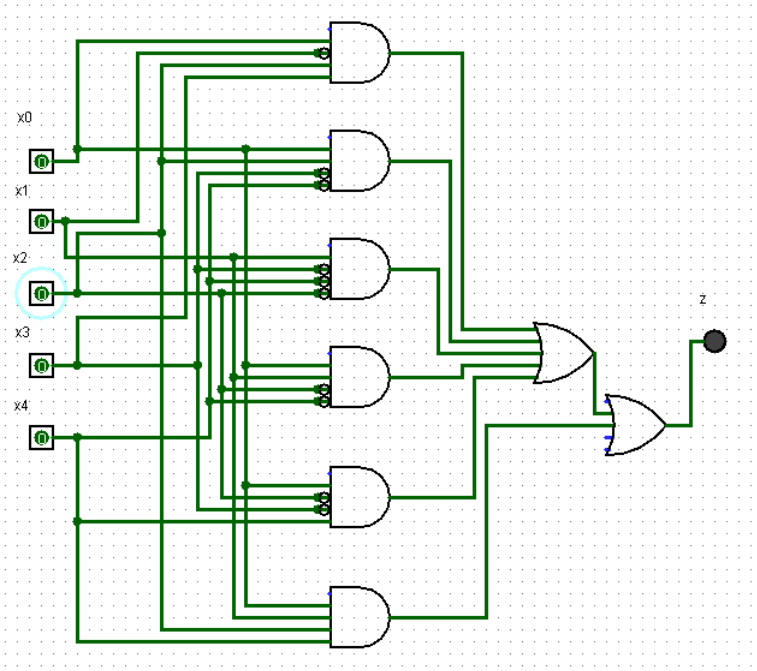
$x4$	$x3$	$x2$	$x1$	$x0$	*Decimal*	z
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	0	0	1	0	2	1
0	0	0	1	1	3	1
0	0	1	0	0	4	0
0	0	1	0	1	5	1
0	0	1	1	0	6	0
0	0	1	1	1	7	1
0	1	0	0	0	8	0
0	1	0	0	1	9	0
0	1	0	1	0	10	0
0	1	0	1	1	11	1
0	1	1	0	0	12	0
0	1	1	0	1	13	1
0	1	1	1	0	14	0
0	1	1	1	1	15	0
1	0	0	0	0	16	0
1	0	0	0	1	17	1
1	0	0	1	0	18	0
1	0	0	1	1	19	1
1	0	1	0	0	20	0
1	0	1	0	1	21	0
1	0	1	1	0	22	0
1	0	1	1	1	23	1
1	1	0	0	0	24	0
1	1	0	0	1	25	0
1	1	0	1	0	26	0
1	1	0	1	1	27	0
1	1	1	0	0	28	0
1	1	1	0	1	29	1
1	1	1	1	0	30	0
1	1	1	1	1	31	1

Since the question did not specify the method, I can use K-map made from this truth table to get logical statement of z:

–	$x1 \wedge x0$	$\sim x1 \wedge x0$	$\sim x1 \wedge \sim x0$	$x1 \wedge \sim x0$
$x4 \wedge x3 \wedge x2$	1	1	0	0
$x4 \wedge \sim x3 \wedge x2$	1	0	0	0
$x4 \wedge \sim x3 \wedge \sim x2$	1	1	0	0
$x4 \wedge x3 \wedge \sim x2$	0	0	0	0
$\sim x4 \wedge x3 \wedge \sim x2$	1	0	0	0
$\sim x4 \wedge \sim x3 \wedge \sim x2$	1	0	0	1
$\sim x4 \wedge \sim x3 \wedge x2$	1	1	0	0
$\sim x4 \wedge x3 \wedge x2$	0	1	0	0

We can rewrite these two K-maps into logical statements as follow:
$z = (x3 \wedge x2 \wedge \sim x1 \wedge x0) \vee (\sim x4 \wedge \sim x3 \wedge x2 \wedge x0) \vee (\sim x4 \wedge \sim x3 \wedge \sim x2 \wedge x1) \vee (\sim x4 \wedge \sim x2 \wedge x1 \wedge x0) \vee (x4 \wedge \sim x3 \wedge \sim x2 \wedge x0) \vee (x4 \wedge x2 \wedge x1 \wedge x0)$

Using K-map, we can be sure that this is the most simplified form of the logical statement. While there are one Xor and one Xnor simplifications we can make with $(4(+)2)$ and $\sim(4(+)2)$, it doesn't decrease the number of logic gates since we still need to wrap the Xor and Xnor with an And gate (respectively) to pair it with other terms ($(\sim x3 \wedge x0)$ and $(x1 \wedge x0)$, respectively). We can make this into a logic circuit:



Note that I have to use two separate Or gates at the output since Logisim has 5 input maximum for their Or gate.