Homework Assignment 7

CS/ECE 3810: Computer Organization
Oct 19, 2020

Boolean Expressions and Logic Design

Due Date: Oct 26, 2020
(100 points)

Important Notes:

- Solutions turned in must be your own. Please, mention references (if any) at the end of each question. *Please refrain from cheating.*
- All solutions must be accompanied by the equations used/logic/intermediate steps. Writing only the final answer will receive **zero** credits.
- Partial score of every question is dedicated to each correct final answer provided by you. Please ensure both your equation/logic and final answer are correct. Moreover, you are expected to provide explanation for your solutions.
- All units must be mentioned wherever required.
- Late submissions **(after 11:59PM on 10/26/2020)** will not be accepted.
- For Question 2, hand drawn solutions are accepted. However, the drawings have to be legible. Illegible drawings may result in a penalty.
- We encourage all solutions to be typed in for which you could use software programs like LaTeX, Microsoft Word etc. If you submit handwritten solutions, they must be readable by the TAs to receive credits.

Boolean expression reduction. Lecture 14 introduces boolean algebra and logic , along with some laws that govern boolean algebra.

Refer to Slide #8 of the Hardware for Arithmetic lecture for the basic boolean algebra laws.

Question 1. Simplify the given boolean expressions, only using the specific laws mentioned in each sub-question. If multiple laws are used, show stepwise how each law helps in reducing the expression. **(30 points)**

1. Use the *distributive* and *inverse* laws to simplify the following term: $(a + b)(\overline{a} + \overline{b})$
2. Use the *distributive* and *zero & one* laws to simplify the following term: $b + \overline{a}bc\overline{d} + ab\overline{c}\overline{d} + \overline{a}bc\overline{d}$

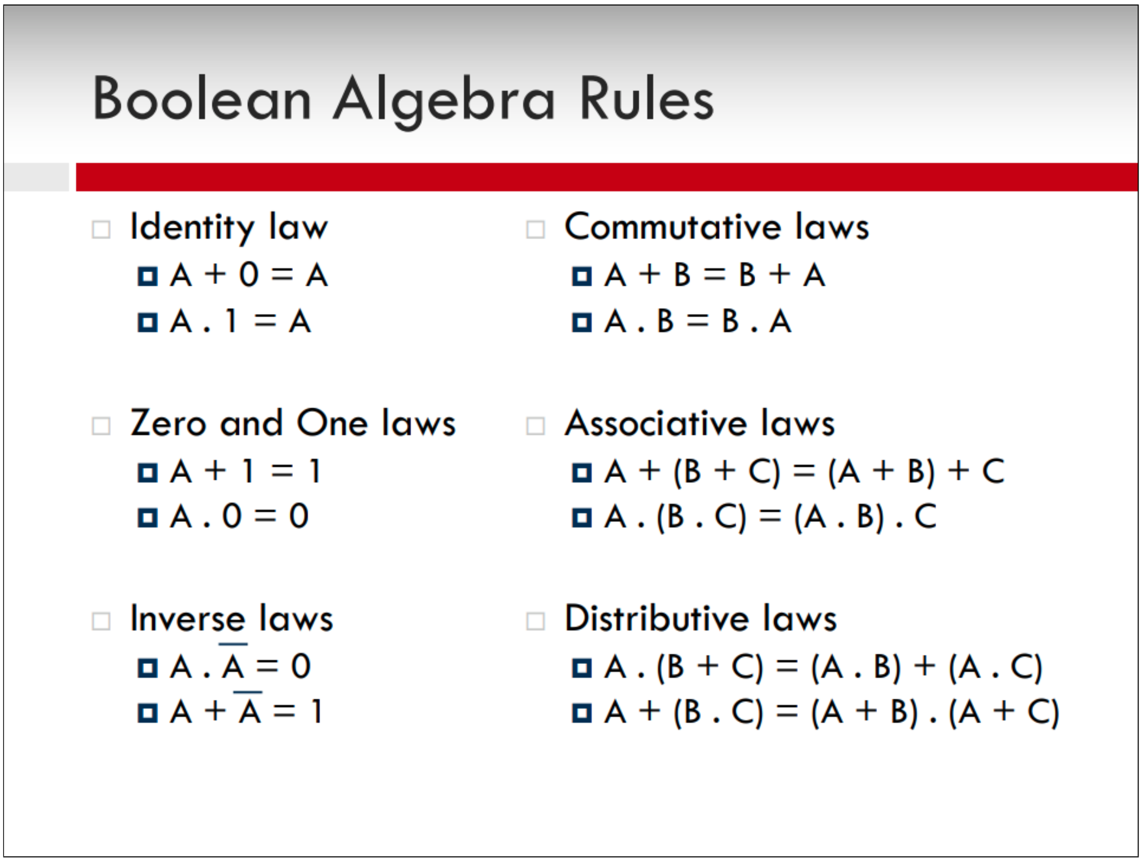


Figure 1: Boolean Laws

3. Use the *inverse* law to simplify the following term: $(a + \overline{a})(b + \overline{b})(c + \overline{c})$
4. Use the *distributive* and *zero & one* laws to simplify the following term: $ab + ab\overline{c}\overline{d} + \overline{c}\overline{d}$

1. $(a + b)(\overline{a} + \overline{b}) \Rightarrow a\overline{a} + a\overline{b} + b\overline{a} + b\overline{b}$ by the Distributive law.
   $a\overline{a} + a\overline{b} + b\overline{a} + b\overline{b} \Rightarrow 0 + a\overline{b} + b\overline{a} + 0$ by the Inverse law
   $0 + a\overline{b} + b\overline{a} + 0 \Rightarrow a\overline{b} + b\overline{a}$ by the Identity law.

2. $b + \overline{a}bc\overline{d} + ab\overline{c}\overline{d} + \overline{a}bc\overline{d} \Rightarrow b + (1 + \overline{a}c\overline{d}) + (1 + a\overline{c}\overline{d}) + (1 + \overline{a}c\overline{d})$ by the Distributive law
   $b + (1 + \overline{a}c\overline{d}) + (1 + a\overline{c}\overline{d}) + (1 + \overline{a}c\overline{d}) \Rightarrow b$ by the zero and one law.

3. $(a + \overline{a})(b + \overline{b})(c + \overline{c}) \Rightarrow (1)(1)(1)$ by the Inverse law. This simplifies to 1.

4. $ab + ab\overline{c}\overline{d} + \overline{c}\overline{d} \Rightarrow ab + (1 + \overline{c}\overline{d}) + \overline{c}\overline{d}$ by the Distributive law.
   $ab + (1 + \overline{c}\overline{d}) + \overline{c}\overline{d} \Rightarrow ab + \overline{c}\overline{d}$ by the zero and one law.

Truth tables and universal gate representations. Lecture 14 introduces truth tables, drawing truth tables, and finding sum-of-product expressions using the drawn truth tables. The following question tests the student's understanding of truth tables, and finding sum-of-product representations of given expressions.

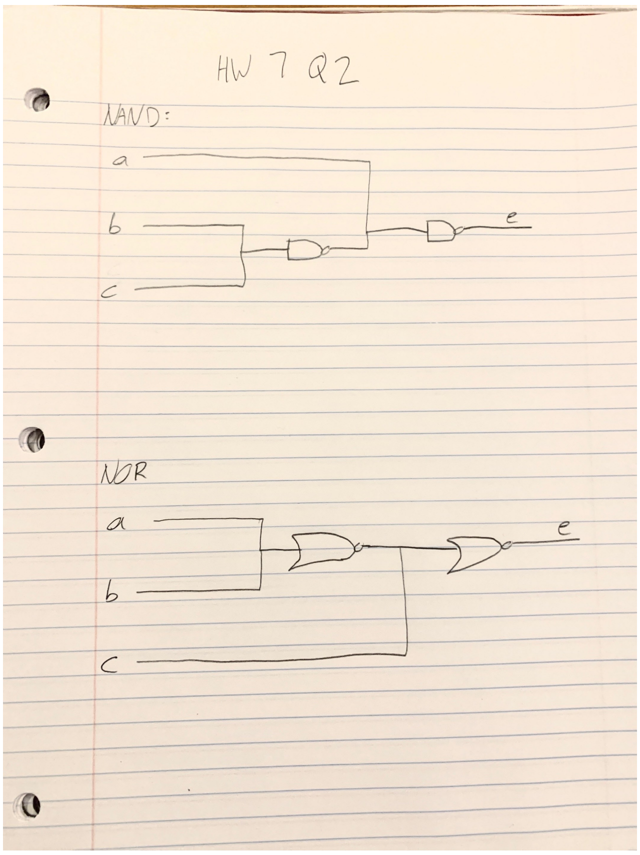
Question 2. The aim of this question is to find the universal gate(NAND-only/NOR-only) representations of boolean expressions. **(30 points)**

Draw the truth table for the expression $e = abc + (a + b)(b + c)(a + c)$, and express the resulting sum-of-product equation in

1. NAND-only form
2. NOR-only form

Note. For this question, hand drawn NAND-only/NOR-only submissions are allowed. However, please make sure that the drawings are legible. Lack of clarity in the drawings may result in a penalty.

a	b	c	e
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Logic Block. A logic block comprises binary inputs and binary outputs. The output is only a function of the inputs. A truth table defines the outputs of a logic block for each set of inputs. (Refer '14. Hardware for Arithmetic' lecture Slide #10-#13)

In this question, let's create truth tables and Boolean equations based on the circuit behaviors defined below:

Question 3.A. An addition circuit takes two 2-bit unsigned numbers and a carry-in as input, and produces a sum and carry-out as outputs.
Consider an example of adding 2 2-bit numbers - 01 and 11 and carry in of 1:
In this case, A1 = 0, A0 = 1; B1 = 1, B0 = 1 and Ci = 1
Adding them would give the output as 01 and carry out of 1.
Thus, Co = 1, S1 = 0 and S0 = 1
(20 points)

i) Fill in the truth table below to achieve the above behavior:

A1	A0	B1	B0	Ci	Co	S1	S0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	1	0
0	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	0	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	0	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	0	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	0	1	0	0
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

ii) Based on the truth table filled above, write boolean equation(s) that describe how each output is computed from the inputs. Write each equation in sum-of-products form, and make sure your equation matches the truth table.

Co $= B1B0C_i + A0B0C_i + A0B1C_i + A0B1B0 + A0B1B0C_i + A1B0C_i + A1B1 + A1B1C_i +$

$A1B1B0 + A1B1B0C_i + A1A0C_i + A1B1B0 + A1B1B0C_i + A1A0B1 + A1A0B1C_i + A1A0B1B0 + A1A0B1B0C_i$

S1 $= B0C_i + B1 + B1C_i + B1B0 + A0C_i + A0B0 + A0B1 + A1 + A1C_i + A1B0 + A1A0 + A1A0B1C_i + A1A0B1B0 + A1A0B1B0C_i$

S0 $= C_i + B0 + B1C_i + B1B0 + A0 + A0B0C_i + A0B1 + A0B1B0C_i + A1C_i + A1B0 + A1B1C_i + A1B1B0 + A1B1B0C_i + A1A0 + A1A0B0C_i + A1A0B1B0C_i$

Question 3.B. A logic circuit takes in one 4 bit binary number. It outputs 1 only if the binary number represents a prime number. **(20 points)**

i) Fill in the truth table below to achieve the above behavior:

S3	S2	S1	S0	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

ii) Based on the truth table filled above, write boolean equation(s) that describe how each output is computed from the inputs. Write each equation in sum-of-products form, and make sure your equation matches the truth table.

Output $= \overline{S3S2}S1S0 + \overline{S3}S2\overline{S1}S0 + \overline{S3}S2S1S0 + S3\overline{S2}S1S0 + S3S2\overline{S1}S0$
