**NAME:** ______________________________

**PERM ID:** ______________________________

# Assignment 6: Introduction to Digital Logic

| | |
|---|---|
| Assigned: | Thu. 2/21/19, 8:00 AM |
| Due: | Mon. 2/25/19, **5:00 PM** in class |
| Points: | 100 |

***Goals for This Week:***
By the time you have completed this work, you should be able to:

- Write out the truth table corresponding to a Boolean function
- Write the truth table in an equivalent sum-of-products form
- Optimize Boolean functions using Boolean algebra and K-Maps

***Instructions:***
For this week, you will answer a series of questions regarding Boolean functions in order to simplify them. This week, you will not be using **`turnin`**. Instead, you will use Gradescope to turn in either a scan of your homework (if so, PLEASE write in pen and clearly/cleanly) OR write your answers electronically on this PDF using your favorite program (e.g. Adobe Acrobat). Either way, you will be uploading a PDF file with your answers onto Gradescope. I'll announce more about the use of the Gradescope tool on Piazza.

Turn in this assignment before the due date (please note the date and the time above!). IF YOU CANNOT SCAN OR EDIT A PDF, then you may turn in your homework as a hardcopy in the mailroom next door to the CS Department Office in **HFH, Room 2104**. ***Please place your finished assignment in the box marked "CS 64".*** Note that the deadline applies to this as well.

Remember to solve these individually, working with your partner only when you get stuck, and seeking help from the TA only when you are both stuck. Show your work wherever possible.

1. In class, we went through the example of splitting a binary addition into single bit additions. This is simplest in terms of design time, but it does not lead to the fastest implementation. Instead, let's create a 2-bit adder as the smallest unit. In this case, there are four bits of input from the two numbers, in addition to the single input carry. There are two result bits and a single output carry. A0 is the least significant bit (the one on the right). inputs: A1, A0, B1, B0, Cin outputs: R0, R1, Cout. For example, if we are performing an addition, it looks like this:

```
      A1 A0
      B1 B0
         Cin
-------------
Cout R1 R0
```

(Question on next page…)

**NAME:** ____________________________________________

**PERM ID:** ____________________________________________

a. Fill in this truth table for the function described above (16 pts)

| $C_{in}$ | A1 | B1 | A0 | B0 | $C_{out}$ | R1 | R0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

b. Write the unoptimized sum of products equations for **each** of the 3 outputs (12 pts)

**NAME:** ______________________

**PERM ID:** ______________________

2. Given this truth table:

| **A** | **B** | **C** | **O** |
|---|---|---|---|
| 0 | 0 | 0 | **0** |
| 0 | 0 | 1 | **1** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **1** |
| 1 | 0 | 1 | **0** |
| 1 | 1 | 0 | **1** |
| 1 | 1 | 1 | **0** |

   a. Write the unoptimized sum of products equation for the output **O** (4 pts)

3. Given this truth table:

| **A** | **B** | **C** | **D** | **O** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **1** |
| 0 | 0 | 0 | 1 | **1** |
| 0 | 0 | 1 | 0 | **1** |
| 0 | 0 | 1 | 1 | **1** |
| 0 | 1 | 0 | 0 | **1** |
| 0 | 1 | 0 | 1 | **0** |
| 0 | 1 | 1 | 0 | **0** |
| 0 | 1 | 1 | 1 | **0** |
| 1 | 0 | 0 | 0 | **1** |
| 1 | 0 | 0 | 1 | **0** |
| 1 | 0 | 1 | 0 | **0** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | **1** |
| 1 | 1 | 0 | 1 | **0** |
| 1 | 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | 1 | **0** |

   a. Write the unoptimized sum of products equation for the output **O** (4 pts)

**NAME:** ______________________________

**PERM ID:** ______________________________

4. Given this truth table:

| **A** | **B** | **C** | **D** | **O** |
|---|---|---|---|---|
| 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 | **1** |
| 0 | 1 | 1 | 0 | **0** |
| 0 | 1 | 1 | 1 | **0** |
| 1 | 0 | 0 | 0 | **0** |
| 1 | 0 | 0 | 1 | **0** |
| 1 | 0 | 1 | 0 | **0** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | **1** |
| 1 | 1 | 0 | 1 | **1** |
| 1 | 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | 1 | **1** |

a. Write the unoptimized sum of products equation for the output **O** (4 pts)

5. For each problem use **Boolean algebra** to simplify the equation. SHOW YOUR WORK, one step per line. You should use only variable names, negation (use `!`), AND (by putting terms next to each other), OR (with `+`), and parentheses. In addition, each line should start with an equal sign (`=`). The first step has been done for you to show how.

**Example**: f(Z,Y,X) = !ZY!X + ZY!X + !YX
= (!Z + Z)Y!X + !YX
= Y!X + !YX

a. f(A,B,C) = !A!BC + A!B!C + !ABC + !AB!C + A!BC (6 pts)

**NAME:** ______________________________________________

**PERM ID:** ______________________________________________

b. f(A,B,C,D) = (A!D + !AC)(!B(C + BD)) (6 pts)

c. f(A,B,C,D) = (!A!C + AD)(B(D + !BC)) (6 pts)

6. Given the following truth table:

| **A** | **B** | **C** | **D** | **O** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | 1 | **1** |
| 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 1 | 1 | **0** |
| 0 | 1 | 0 | 0 | **1** |
| 0 | 1 | 0 | 1 | **0** |
| 0 | 1 | 1 | 0 | **1** |
| 0 | 1 | 1 | 1 | **0** |
| 1 | 0 | 0 | 0 | **0** |
| 1 | 0 | 0 | 1 | **1** |
| 1 | 0 | 1 | 0 | **0** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | **0** |
| 1 | 1 | 0 | 1 | **0** |
| 1 | 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | 1 | **0** |

a. Write the **<u>unoptimized</u>** sum of products equation for the output **O** (4 pts)

**NAME:** ______________________________________________

**PERM ID:** ______________________________________________

b. Draw the K-map, mark it clearly (cleanly), and simplify the output function (4 pts)

c. Write the ***optimized*** sum of products equation for the output **O** (2 pts)

d. Draw the circuit for the optimized output function (3 pts)

**NAME:** ____________________

**PERM ID:** ____________________

7. Given this truth table (and note the use of don't-cares, represented by **X**):

| **A** | **B** | **C** | **D** | **O** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **1** |
| 0 | 0 | 0 | 1 | **X** |
| 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 1 | 1 | **1** |
| 0 | 1 | 0 | 0 | **X** |
| 0 | 1 | 0 | 1 | **0** |
| 0 | 1 | 1 | 0 | **X** |
| 0 | 1 | 1 | 1 | **0** |
| 1 | 0 | 0 | 0 | **X** |
| 1 | 0 | 0 | 1 | **0** |
| 1 | 0 | 1 | 0 | **1** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | **1** |
| 1 | 1 | 0 | 1 | **0** |
| 1 | 1 | 1 | 0 | **X** |
| 1 | 1 | 1 | 1 | **0** |

a. Write the unoptimized sum of products equation for the output **O** (4 pts)

b. Draw the K-map, mark it clearly (cleanly), and simplify the output function (4 pts)

**NAME:** ____________________

**PERM ID:** ____________________

c. Write the ***optimized*** sum of products equation for the output **O** (2 pts)

d. Draw the circuit for the optimized output function (3 pts)

**NAME:** ______________________________

**PERM ID:** ______________________________

8. The 8-input MUX can be constructed entirely with 2-input MUXes. I will start you out: put inputs A and B into one MUX and inputs C and D into another MUX. Continue on with E F and G H like this. This is the first stage of the solution. Now, decide how to reconcile the two results from those MUXes. Look hard at the logic for the 2-input MUX vs the 8-input MUX. Look at the discussion we've had in class (Lecture 13) on this.

   a. How many 2-input MUXes are needed to create an 8-input MUX? (2 pts)

   b. How many select lines are needed? (2 pts)

   c. Draw the circuit, using 2-input muxes as your building blocks. (4 pts)

9. Now think of a MUX that has $2^n$ inputs. All of your answers will have the variable n in them.

   a. How many select bits are there? (2 pts)

   b. If it is implemented with 2-input MUXes, how many MUXes are there in the first stage? (2 pts)

   c. How many 2-input MUXes are there in the second stage? (2 pts)

   d. How many stages of 2-input MUXes are there? (2 pts)