CPSC3300 – Computer Systems Organization  
Homework #2 – Boolean Algebra and Adders

Due: Before class on Monday, February 11

Submit to Canvas

Total 100pts



1. [20pts] Logical completeness



* 1. Show that you can use only two-input NAND gates to implement each of the following two-input logic functions, and draw the used NAND gates and wiring.



* + 1. NOR function



* + 1. XOR function



* 1. Show that you can use only two-input NOR gates to implement each of the following two-input logic functions, and draw the used NOR gates and wiring.
     1. NAND function



* + 1. XOR function



1. [10pts] A 2-1 Mux takes two inputs and uses a 1-bit selector value to select one of the inputs for the output. A 1-2 Demux does the opposite taking a single input and using a 1-bit selector to select which one of the two output lines will have the input value while the other output line has a 0 value. Draw the logic gates and wiring for a 1-2 Demux.



1. [10pts] Demonstrate by means of truth tables the validity of the following identities:

false

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C |  |  |
| T | T | T | F | F |
| T | T | F | T | F |
| T | F | T | T | F |
| T | F | F | T | F |
| F | T | T | T | F |
| F | T | F | T | F |
| F | F | T | T | F |
| F | F | F | T | T |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C |  |  |
| T | T | T | T | T |
| T | T | F | T | T |
| T | F | T | T | T |
| T | F | F | T | T |
| F | T | T | T | T |
| F | T | F | F | F |
| F | F | T | F | F |
| F | F | F | F | F |



1. [20pts] Prove the identity of each of the following Boolean equations, using algebraic manipulation:
   * 2. = 1
     3. = 1
2. [20pts] For the Boolean function O1 and O2, as given in the following truth table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | | |  | **Output** | |
| **x** | **y** | **z** | **minterms** | **O1** | **O2** |
| 0 | 0 | 0 | x’y’z | 1 | 0 |
| 0 | 0 | 1 | x’y’ | 1 | 0 |
| 0 | 1 | 0 | x’yz’ | 0 | 1 |
| 0 | 1 | 1 | x’yz | 1 | 0 |
| 1 | 0 | 0 | xy’z | 0 | 1 |
| 1 | 0 | 1 | xy’z’ | 1 | 0 |
| 1 | 1 | 0 | xyz’ | 1 | 0 |
| 1 | 1 | 1 | xyz | 0 | 1 |

* 1. List the minterms for a three-variable function with variables x, y, and z.
     1. See above
  2. Express O1 and O2 in sum-of-product algebraic form.



* + 1. x’y’z’ + x’y’z + x’yz + xy’z + xyz’
    2. x’yz + xy’z’+ xyz

1. [20pts] In class, we learned the implementation for a 4-bit carry lookahead adder. We can use the same idea and extend to build a 16-bit carry lookahead adder. Denote this implementation as a one-level carry lookahead adder.

In the textbook, Figure 8.6.3 shows a two-level implementation of a 16-bit carry lookahead adder. This adder uses 4-bit carry lookahead adders at the lower level, and uses a carry lookahead unit at the higher level.

Compare these two implementations and provide your explanation why the two-level implementation could be preferred.

