CSSE2010 / CSSE7201 – Introduction to Computer Systems
Exercises – Week Two
Logic Gates and Functions

Exercises

Many of the problems below are taken from or based on questions in Tanenbaum, Structured Computer Organisation, 5th edition, Appendix A and Chapter 3. A number of questions are taken from "Digital Design", 3rd edition by M. Morris Mano, Prentice-Hall, 2002.

1. Perform the following calculations on 8-bit two's complement numbers.

00101101	11111111	00000000	11110111
+ 01101111	+ 11111111	– 11111111	– 11110111

   Verify the results by converting the numbers to decimal.

2. Consider the following addition problems for 3-bit binary numbers in two's complement. For each sum, state
   (a) Whether the sign bit of the result is 1.
   (b) Whether the low-order 3-bits are 0.
   (c) Whether an overflow occurred.

000	000	111	100	100
+ 001	+ 111	+ 110	+ 111	+ 100

3. Show how a 4-input NOR gate and a 3-input XOR gate (i.e. the odd function) can be implemented with only
   (a) 2 input NAND gates
   (b) inverters and 2 input NOR gates

4. There are four possible Boolean functions of one variable (A). These are X = 0, X = 1, X = A and X = $\bar{A}$.
   (a) Write down the equations for the 16 possible functions of two variables (A, B).
   (b) How many possible functions of three input variables exist? of n variables?

5. A *literal* in a Boolean function is a single variable (which may or may not be complemented). For example, in the function $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$ there are 8 literals. In the equivalent function, $F = \overline{X}Y + XZ$, there are 4 literals. The literal count can be thought of as the number of gate inputs in a logic diagram representation of the function. Using Boolean algebra, modify each of the following functions to a logically equivalent function using as few literals as possible:
   (a) $XY + X\overline{Y}$
   (b) $(B\overline{C} + \overline{A}D)(A\overline{B} + C\overline{D})$
   (c) $\overline{A}B(\overline{D} + \overline{C}D) + B(A + \overline{A}CD)$

6. Given the Boolean function $F = X\overline{Y}Z + \overline{X}\overline{Y}Z + \overline{W}XY + W\overline{X}Y + WXY$
   (a) Obtain the truth table of the function
   (b) Draw the logic diagram using the original Boolean expression
   (c) Simplify the function to a minimum number of literals using Boolean algebra
   (d) Obtain the truth table of the function from the simplified expression and show that it is the same as the one in part (a)
   (e) Draw the logic diagram from the simplified expression and compare the total number of gates with the diagram of part (b)

7. Write a sum-of-products Boolean expression for the function defined by the following truth table. Draw the logic diagram equivalent also. Can you simplify the expression? If so, draw the equivalent logic diagram.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

8. By using truth tables, prove
   (a) DeMorgan's laws for three variables: $\overline{ABC} = \overline{A} + \overline{B} + \overline{C}$ and $\overline{A + B + C} = \overline{A}\,\overline{B}\,\overline{C}$
   (b) the distributive law: $A + BC = (A+B)(A+C)$

9. Find the complement of the following expressions:
   (a) $\overline{X}Y + X\overline{Y}$
   (b) $(A\overline{B} + C)\overline{D} + E$

10. Use a truth table to show that $P = PQ + P\overline{Q}$.

11. Using Boolean algebra, modify each of the following functions to a logically equivalent function using as few literals as possible:
    (a) $(X + Y)(X + \overline{Y})$
    (b) $(\overline{A} + C)(\overline{A} + \overline{C})(A + B + \overline{C}D)$

12. Write down the function table and draw the logic symbol for an 8-to-1 multiplexer. Draw a logic circuit which implements this function.

13. Show how a 4-to-1 multiplexer can be used to implement a 2-input XOR gate. (Hint: compare the function table of a 4-to-1 multiplexer with the truth table of a 2-input XOR gate, and consider how you can make them the same.) Is there any logic function of 2 variables which can NOT be implemented using a 4-to-1 multiplexer?

14. Write down a sum-of-products function representation of a full-adder's outputs in terms of its inputs. Draw the logic circuit equivalent. (This is an alternative way to build a full-adder instead of using two half-adders and an OR gate.)

15. A comparator is a circuit which compares two input words. Draw a logic diagram of a comparator which takes two 8-bit input words ($A_7A_6A_5A_4A_3A_2A_1A_0$ and $B_7B_6B_5B_4B_3B_2B_1B_0$) and produces a 1 if they are equal and 0 if they are not equal. (Hint: consider the two-input equivalence circuit shown in the lecture.)

16. Draw the logic diagram of a 2-bit encoder, a circuit with four input lines, exactly one of which is high at any instant, and two output lines whose 2-bit binary value tells which input is high.

17. Implement a full adder with two 4-to-1 multiplexers.