CS 207 Digital Logic - Spring 2019
Assignment 1

Deadline: Friday, Mar. 22, 2019

Digital Logic Theory

Write down your answer to the questions on a new sheet with detailed procedures.

1. (0.3 points) List the octal and hexadecimal numbers from 10 to 32. With A and B as the last two digits, list the numbers from 10 to 32 in base-12.

Solution:

$12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 36, 37, 40,$

$A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1F, 20,$

$A, B, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 20, 21, 22, 23, 24, 25, 26, 27, 28$

2. (0.3 points) What is the largest binary number that can be expressed with 12 bits? What are the equivalent decimal and hexadecimal numbers?

Solution: 111111111111, $2^{12} - 1 = 4095$, FFF

3. (0.5 points) Perform subtraction on the given unsigned numbers using the 10's complement of the subtrahend. Where the result should be negative, find its 10's complement and affix a minus sign. Verify your answers.

   (a) $4637 - 2579$

   (b) $125 - 1800$

   (c) $2043 - 4361$

   (d) $1631 - 745$

Solution:

(a) 2579 $\rightarrow$ 9's comp: 7420 $\rightarrow$ 10's comp: 7421 $\rightarrow$ $4637 + 7421 = 12058$ $\rightarrow$ Drop carry: 2058.
(b) 1800 $\rightarrow$ 9's comp: 8199 $\rightarrow$ 10's comp: 8200 $\rightarrow$ $125 + 8200 = 8325$ $\rightarrow$ Negative 10's comp: -1675.
(c) 4361 $\rightarrow$ 9's comp: 5638 $\rightarrow$ 10's comp: 5639 $\rightarrow$ $2043 + 5639 = 7682$ $\rightarrow$ Negative 10's comp: -2318.
(d) 745 $\rightarrow$ 9's comp: 9254 $\rightarrow$ 10's comp: 9255 $\rightarrow$ $1631 + 9255 = 10886$ $\rightarrow$ Drop carry: 886.

4. (0.3 points) Write the expression "G. Boole" without quotes in ASCII, using an eight-bit code. Include the period and the space. Treat the leftmost bit of each character as a parity bit. Each eight-bit code should have odd parity.

> **Solution:** 11000111 10101110 00100000 11000010 11101111 11101111 11101100 11100101

5. (0.5 points) The following is a string of ASCII characters whose bit patterns have been converted into hexadecimal for compactness: 73 F4 E5 76 E5 4A EF 62 73. Of the eight bits in each pair of digits, the leftmost is a parity bit. The remaining bits are the ASCII code.
   (a) Convert the string to bit form and decode the ASCII.
   (b) Determine the parity used: odd or even?

> **Solution:**
> (a) 01110011 11110100 11100101 01110110 11100101 01001010 11101111 01100010 0111011
> steveJobs
> (b) Odd parity

6. (0.3 points) Find the complement of $F = wx + yz$; then show that $FF' = 0$ and $F + F' = 1$.

> **Solution:**
> $F' = (w' + x')(y' + z')$
> $FF' = (wx + yz)(wx)'(yz)' = (wx)(wx)' + (yz)(yz)' = ((wx)' + (wx))' + ((yz)' + (yz))' = 1' + 1' = 0$
> $F + F' = (wx + yz) + (w' + x')(y' + z') = ((wx)'(yz)')' + (wx)'(yz)' = 1$

7. (0.5 points) Draw logic diagrams to implement the following Boolean expressions:
   (a) $y = [(u + x')(y' + z)]$
   (b) $y = (u \oplus y)' + x$
   (c) $y = u(x \oplus z) + y'$
   (d) $y = u + x + x'(u + y')$

8. (0.5 points) The logical product of all maxterms of a Boolean function of n variables is 0.
   (a) Prove the previous statement for $n = 3$.
   (b) Suggest a procedure for a general proof of n.

> **Solution:**
> (a)
>
> $$\begin{aligned} F &= M_7 M_6 M_5 M_4 M_3 M_2 M_1 \\ &= (a + b + cc')(a + b' + cc')(a' + b + cc')(a' + b' + cc') \\ &= (a + bb')(a' + bb') = aa' = 0 \end{aligned}$$
>
> (b) There always exists one maxterm that yields 0 whatever the variable values are.

9. (0.5 points) Obtain the truth table of the following functions, and express each function in sum-of-minterms and product-of-maxterms form:
   (a) $(c' + d)(b + c')$
   (b) $bd' + acd' + ab'c + a'c'$

Solution:
(a)$\sum(0, 1, 4, 5, 7)$, $\prod(2, 3, 6)$
(b)$\sum(0, 1, 4, 5, 6, 10, 11, 12, 14)$, $\prod(2, 3, 7, 8, 9, 13, 15)$

10. (0.3 points) Show that a positive logic NAND gate is a negative logic NOR gate and vice versa.

Solution:

x	y	NAND	x(-ve)	y(-ve)	NOR(-ve)
0/L	0/L	1/H	1/L	1/L	0/H
0/L	1/H	1/H	1/L	0/H	0/H
1/H	0/L	1/H	0/H	1/L	0/H
1/H	1/H	0/L	0/H	0/H	1/L

11. (0.3 points) Write the Boolean equations and draw the logic diagram of the circuit whose outputs are defined by the following truth table:

f_1	f_2	a	b	c	f_1	f_2	a	b	c
1	1	0	0	0	1	0	1	0	0
0	1	0	0	1	0	1	1	0	1
1	0	0	1	0	1	0	1	1	1
1	1	0	1	1					

Solution: $f_1 = \sum(0, 2, 3, 4, 7) = bc + b'c' + a'b$, $f_2 = \sum(0, 1, 3, 5) = a'b' + a'c + b'c$, or $f_1 = b + c'$, considering the missing row as don't care.

12. (0.3 points) Write the following Boolean expressions in sum of products form: $(b + d)(a' + b' + c)$.

Solution: $(b+d)(a'+b'+c) = a'b+a'd+bb'+b'd+bc+cd = a'b+a'd+b'd+bc+cd = a'b+b'd+bc$

13. (0.5 points) Simplify the following Boolean functions, using three-variable maps:
   (a) $F(x, y, z) = \sum(1, 2, 3, 6, 7)$
   (b) $F(x, y, z) = \sum(3, 4, 5, 6, 7)$

Solution:
(a) $F = y + x'z$
(b) $F = x + yz$

14. (0.5 points) Simplify the following Boolean expressions, using four-variable maps:
    (a) $A'B'C'D + AB'D + A'BC' + ABCD + AB'C$
    (b) $A'B'C'D' + BC'D + A'C'D + A'BCD + ACD'$

> **Solution:**
> (a) $F = A'BC' + B'C'D + AB'C + ACD$
> (b) $F = A'B'C' + BC'D + A'BD + ACD'$

15. (0.5 points) Simplify the following Boolean functions by first finding the essential prime implicants:
    (a) $F(w, x, y, z) = \sum(0, 2, 5, 7, 8, 10, 12, 13, 14, 15)$
    (b) $F(A, B, C, D) = \sum(0, 2, 3, 5, 7, 8, 10, 11, 14, 15)$

> **Solution:**
> (a) $F = xz + w'x + x'z'$
> (b) $F = AC + B'D' + CD + A'BD$

16. (0.3 points) Express the following Boolean function with eight or fewer literals: $F = A'BC'D + AB'CD + A'B'C' + ACD'$.

> **Solution:**
> (1) By K-map, $F = A'B'C' + A'C'D + AB'C + ACD' = A'C'(B' + D) + AC(B' + D')$
> (2) $F = A'BC'D + AB'CD + A'B'C' + ACD' = A'C'(B' + BD) + AC(B'D + D') = A'C'(B' + D) + AC(B' + D')$

17. (0.6 points) Implement the following Boolean function F, together with the don't-care conditions d, using no more than two NOR gates and no other gates:

$$F(A, B, C, D) = \sum(2, 4, 10, 12, 14)$$
$$d(A, B, C, D) = \sum(0, 1, 5, 8)$$

Assume that both the normal and the complemented inputs are available.

> **Solution:**
> $F' = D + A'BC$, $F = [D + A'BC]' = [D + (A + B' + C')']'$

Digital Logic Experiment

> Pack (`tarball`, `zip`, `7z`, etc.) the source and output files as indicated in the respective sections in lab sheets.

1. (0.3 points) Section 5.4 of Lab 1 (a component that has AND, OR, and NOT gates).

2. (0.4 points) Section 2.1 of Lab 2 (sum-of-products and productof-sums transformation).
3. (0.3 points) Section 3.1 of Lab 2 (two-input XOR with only AND, OR, and NOT gates).
4. (0.4 points) Section 1 of Lab 3 (textual output of all operators)
5. (0.4 points) Section 2.2 of Lab 3 (UDP implementation of Boolean expression).
6. (0.3 points) Section 1 of Lab 4 (textual output of `begin-end` and `fork-join`).
7. (0.3 points) Section 2 of Lab 4 (textual output of blocking and nonblocking assignment).
8. (0.3 points) Section 3.1 of Lab 4 (textual output of nested `if-else` example).
9. (0.3 points) Section 3.2 of Lab 4 (textual output of nested `casez` example).