San Diego State University

CS 370 Computer Architecture

Homework Assignment #1

- Please type the solutions using a word processor such as WORD, Latex, etc., or write by hand very neatly and legibly, comparable to typing*.
- Please pay special attention to the due date – no late turn ins or special case consideration.
- Please do not email your homework. Instead, please turn in a paper version of your homework in the classroom. Neither the instructor nor the grader will print your homework for you.
- Please do the following problems from the textbook, and submit solutions:

1. Problem 1-9 {please ignore the "Octal" part}
2. Problem 1-12
3. Problem 2-3 (a)
4. Problem 2-6
5. Problem 2-11 {please ignore "maxterms of each function" in (a)}
6. Problem 2-17
7. Problem 2-20
8. Problem 3-9
9. Problem 3-10
10. Problem 3-11

* The preferred format is typing with a word processor for the following reasons:
1. You have a copy in your computer that you can study for exams or future use.
2. You will learn to use a word processor (if not already learned) to do math, diagrams, etc. This will be one of the most useful things in your career.
3. You have a backup copy in case of lost or misplaced assignment.
4. A typed assignment helps the grader to spend less time and to be more accurate in grading. Our class is multi-national, and it is hard and time consuming to decode many different hand writing styles.

1-9. *Convert the following numbers from the given base to the other three bases listed in the table:

Decimal	Binary	Hexadecimal
369.3125	?	?
?	10111101.101	?
?	?	F3C7A

1-12. Perform the following binary multiplications:

(a) 1010×1100 **(b)** 0110×1001 **(c)** 1111001×011101

2-3. +Prove the identity of each of the following Boolean equations, using algebraic manipulation:

(a) $AB\overline{C} + B\overline{C}\,\overline{D} + BC + \overline{C}D = B + \overline{C}D$

2-6. Simplify the following Boolean expressions to expressions containing a minimum number of literals:

(a) $\overline{A}\,\overline{C} + \overline{A}BC + \overline{B}C$

(b) $(\overline{A + B + C}) \cdot \overline{ABC}$

(c) $AB\overline{C} + AC$

(d) $\overline{A}\,\overline{B}D + \overline{A}\,\overline{C}D + BD$

(e) $(A + B)(A + C)(A\overline{B}C)$

2-11. For the Boolean functions E and F, as given in the following truth table:

X	Y	Z	E	F
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

(a) List the minterms of each function.

(b) List the minterms of $\overline{E}$ and $\overline{F}$

(c) List the minterms of $E + F$ and $E \cdot F$.

(d) Express E and F in sum-of-minterms algebraic form.

(e) Simplify E and F to expressions with a minimum of literals.

2-17. Optimize the following Boolean functions, using a map:

(a) $F(W, X, Y, Z) = \Sigma m(0, 1, 2, 4, 7, 8, 10, 12)$

(b) $F(A, B, C, D) = \Sigma m(1, 4, 5, 6, 10, 11, 12, 13, 15)$

2-20. Optimize the following Boolean functions by finding all prime implicants and essential prime implicants and applying the selection rule:

(a) $F(A, B, C, D) = \Sigma m(1, 5, 6, 7, 11, 12, 13, 15)$

(b) $F(W, X, Y, Z) = \Sigma m(0, 1, 2, 3, 4, 5, 10, 11, 13, 15)$

(c) $F(W, X, Y, Z) = \Sigma m(0, 1, 2, 5, 7, 8, 10, 12, 14, 15)$

3-9. +Design a combinational circuit that accepts a 4-bit number and generates a 3-bit binary number output that approximates the square root of the number. For example, if the square root is 3.5 or larger, give a result of 4. If the square root is < 3.5 and ≥ 2.5, give a result of 3.

3-10. Design a circuit with a 4-bit BCD input A, B, C, D that produces an output W, X, Y, Z that is equal to the input $+3$ in binary. For example, 9 (1001) + 3 (0011) = 12 (1100). The outputs for invalid BCD codes are don't-cares.

3-11. A traffic metering system for controlling the release of traffic from an entrance ramp onto a superhighway has the following specifications for a part of its controller. There are three parallel metering lanes, each with its own stop (red)–go (green) light. One of these lanes, the car pool lane, is given priority for a green light over the other two lanes. Otherwise, a "round robin" scheme in which the green lights alternate is used for the other two (left and right) lanes. The part of the controller that determines which light is to be green (rather than red) is to be designed. The specifications for the controller follow:

Inputs

PS	Car pool lane sensor (car present—1; car absent—0)
LS	Left lane sensor (car present—1; car absent—0)
RS	Right lane sensor (car present—1; car absent—0)
RR	Round robin signal (select left—1; select right—0)

Outputs

PL	Car pool lane light (green—1; red—0)
LL	Left lane light (green—1; red—0)
RL	Right lane light (green—1; red—0)

Operation

1. If there is a car in the car pool lane, PL is 1.
2. If there are no cars in the car pool lane and the right lane, and there is a car in the left lane, LL is 1.
3. If there are no cars in the car pool lane and in the left lane, and there is a car in the right lane, RL is 1.
4. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 1, then LL = 1.
5. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 0, then RL = 1.
6. If any PL, LL, or RL is not specified to be 1 above, then it has value 0.

(a) Find the truth table for the controller part.

(b) Find a minimum multiple-level gate implementation with minimum gate-input cost using AND gates, OR gates, and inverters.