Digital Electronics Assignment

Answer all questions

Question 1

A lock has five buttons and a knob used to open the lock when the five buttons have been pressed in the correct sequence. Whenever the knob is turned, successfully or not, the state of the lock is reset. The correct sequence is hardwired into the control logic of the lock.

a. Describe the inputs and outputs for a logic circuit which can be used to control the lock. [5]

b. Draw a state diagram for the logic circuit. [10]

c. Derive equations for the logic system using J-K flip- flops to hold state variables. Do not minimize the equations. [5]

Question 2

a. In the context of digital logic, what is a multiplexer? [5]
b. Sketch the design of an 8-bit to 2-bit multiplexer. [5]
c. Suppose you had to implement an arbitrary function of seven variables and were provided with nine 8-bit to 1-bit multiplexers and an inverter. How would you implement the function? [10]

Question 3

a. What is the maximum number of terms there can be in a minimal sum of products form of a function of n Boolean variables? [2]

b. Consider a two-bit multiplier with inputs x_1, x_0, y_1, y_0 and outputs z_3, z_2, z_1, z_0 such that $Z = Y \times X$ where Z, Y, X are the positive integers represented by $z_3 z_2 z_1 z_0$, $y_1 y_0$ and $x_1 x_0$ using the obvious representation. Find a minimal sum of products expression for each of z_3, z_2, z_1 and z_0. [10]

c. Comment on the complexity of building an eight-bit multiplier using a minimal sum of products form. [3]

d. Describe another way of building an eight-bit multiplier. [5]

Question 4

a. Use Boolean algebra to simplify the following functions:

$$X = \overline{A}.B.C + \overline{B.\overline{C}} + B.C$$

$$Y = \overline{(A + B + C)}.D + A.D + B$$

[6]

$$X = \overline{A}.\overline{B}.\overline{C} + B.\overline{C} + B.C$$

b. Implement the Boolean function using
   i. An 8:1 Multiplexor; [2]
   ii. A 4:1 Multiplexor, plus a NOT gate; [2]
   iii. A 2:1 Multiplexor, plus a NOT gate, plus an OR gate. [2]

c. A priority encoder has 2^N inputs. It produces an N-bit binary output indicating the most significant bit of the input that is TRUE, or 0 if none of the inputs is TRUE. It also produces output NONE that is TRUE if none of the inputs is TRUE.

i. Write down the Truth Table showing all inputs and all outputs for an eight-input priority encoder. [2]

d. Give simplified Boolean expressions for all outputs of the eight-input priority encoder. [6]

Question 5

a. Consider a 4-input Boolean function that outputs a binary 1 whenever an odd number of its inputs are binary 1.

i. Using Boolean logic or otherwise, show how the above function can be implemented using only 2-input XOR gates. [4]

ii. Show how the above function may alternatively be implemented using one 4-input decoder, and a minimum number of 4-input NOR and 4-input NAND gates. [3]

$$F = \overline{B}.\overline{C} + \overline{A}.B.C + A.C.\overline{D}$$

b. Consider the following Boolean expression

i. Show that F can be represented by the following Product of Sums (POS) form

$$F = (\overline{B} + C).(\overline{A} + \overline{C} + \overline{D}).(A + B + \overline{C})$$

[3]

ii. Show how F can be implemented in a 2-level form using OR gates followed by an AND gate. Remember to indicate any NOT gates required, since only uncomplemented input variables are available. [2]

c. Consider your implementation in part b. ii

i. Assume that the gates have finite propagation delay. Describe in detail what happens at the output F when the inputs {A, B, C, D} change from {1, 1, 0, 1} to {1, 1, 1, 1}. [4]

ii. Using a Karnaugh map or otherwise, determine the other single input variable change that will give rise to a similar problem to that observed in part (c)(i). [2]

iii. Using a Karnaugh map or otherwise, determine a modied POS expression for F that will eliminate the problems observed in parts (c)(i) and (c)(ii) [2]

Question 6

a. State De Morgan's theorems. [4]

b. Simplify the function. $f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c} + a\bar{b}\bar{d}$ with don't care states $\bar{a}\bar{b}\bar{c}d$ and $\bar{a}\bar{b}c\bar{d}$ to give expressions in the following forms:

i. Sum of products; [3]

ii. Product of sums. [3]

c. Simplify the function. $f = (\bar{a} + \bar{b} + \bar{c}).(b + d)$ to give an expression in the sum of products form. [6]

d. Implement with 2-level logic the function in part (c) using only
   i. NOR gates; [2]
   ii. NAND gates. [2]

Question 7

a. Briefly explain the differences between combinational and sequential logic. [2]

b. With the aid of appropriate diagrams, briey explain the operation of Moore and Mealy finite state machines and highlight their diferences. [6]

c. The state sequence for a binary counter is as follows:

A	B	C	D
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

The counter is to be implemented using four synchronously clocked D-type flip flops

i. Draw a state table for the counter, showing the required D inputs. [4]

ii. Find expressions for the D inputs, making use of unused states if appropriate. [6]

iii. What problem could occur when the counter circuit is powered-up? Give two possible general methods for overcoming the problem. [2]

Question 8

a. In a particular computer system, numbers are represented using words having a length of 4 bits.
   i. What is the range of positive numbers that can be represented using unsigned binary numbers? [2]
   ii. Explain how the 2's complement representation can be used to describe signed binary numbers, using 4-bit words as an example. [3]
   iii. Using decimal (base 10) representation for the answers, perform the following 2's complement 4-bit additions, noting any problems:

- 0110 + 1101 [2]
- 1010 + 1011 [2]

b. Complete the following truth table that describes a single-bit full adder:

C_{IN}	A	B	C_{OUT}	sum
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

[2]

c. Show how 4 single-bit full-adders can be combined to implement a 4-bit ripple carry-adder.[2]

d. Briefly describe how the speed of operation of the approach in part (c) could be improved.[4]

e. Show how COUT in part (b) can be implemented using only NAND gates .[3]