

ECE2030A Introduction to Computer Engineering
Fall 2008
Homework Assignment #2
Assigned 09/29/08 **Due in the first 5 min in class 10/10/08**
No late turn-in accepted

1. (20%) Use Quine-McClusey Method to simplify the following canonical SOP forms. Find all the Essential Prime Implicants based the Q-M Method, and identify the non-Essential Prime Implicants. List all possible solutions of your results.

1.1. $F(A, B, C, D, E) = \sum m(0, 1, 4, 5, 8, 9, 11, 13, 17, 20, 21, 28, 30, 31) + d(10, 12, 16)$

1.2. $F(A, B, C, D, E) = \sum m(0, 4, 12, 15, 27, 29, 30) + d(1, 5, 9, 10, 14, 16, 20, 28, 31)$

2. (10%) Implement the following Boolean Expressions using Mixed Logic. For each equation, show two implementations of the logic circuits either completely in NAND gates or completely in NOR gates. Please discuss how many NAND (or NOR) gates are needed for each logic circuit. For each gate, you are allowed to use any arbitrary number of inputs. Do not simplify the equations, leave them intact.

2.1. $F = A + B \cdot C \cdot \bar{D}$

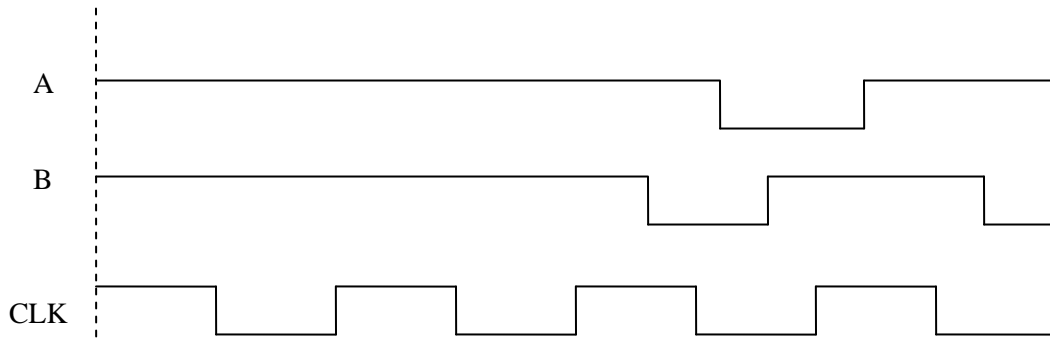
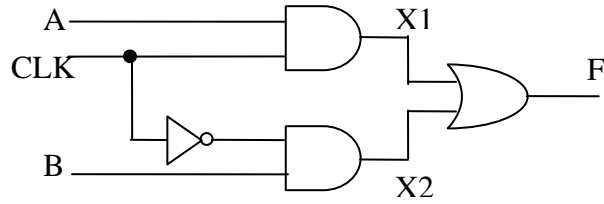
2.2. $F = \overline{\overline{A + B} \cdot C + \overline{\overline{D} + C} \cdot E} + B$

3. (10%) For the given canonical POS representation below, please finish the sub-problems. You can draw your Multiplexor and Decoder using block diagram with appropriate input and output labels. In other words, you do not need to provide the design details within a MUX block or Decoder block.

$$F(A, B, C, D) = \prod M(2, 3, 6, 7, 9, 12, 14, 15)$$

- 3.1. Design it using an **8-to-1 Multiplexor**
3.2. Design it using a **4-to-16 Decoder and OR gates**

4. (10%) Given the following combinational logic and input waveforms A, B, and CLK, draw the corresponding timing diagram for signal X1, X2 and F.



5. (20%) Binary Coded Decimal (or BCD) is a coding method which uses 4-bit binary to represent a decimal from 0 up to 9 as shown below. Note that, 4-bit binaries beyond 1001 are not used.

Decimal Symbol	BCD Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Figure 3(a)

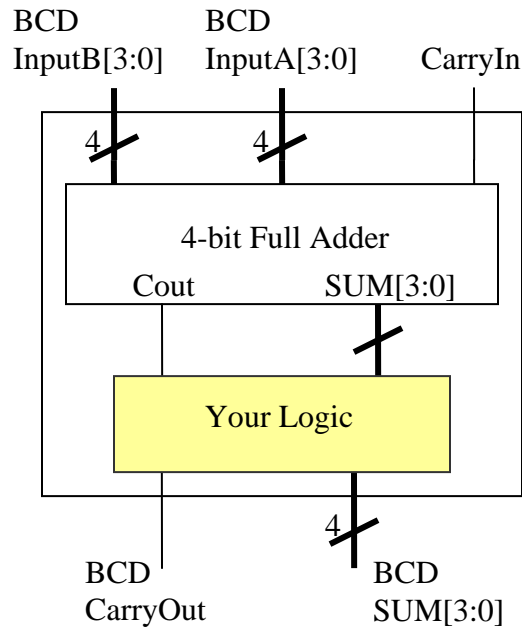


Figure 3(b)

The first intuitive step to design a functional BCD adder is to have a “carry generation logic” which generate a ‘1’ and the corresponding correct BCD SUM output when the output of the sum is equal to or greater than 10. For example, if the BCD inputs A and B are 8 (1000) and 9 (1001), then the BCD output should have a BCD CarryOut=“1” and an encoded BCD SUM[3:0] = 0111 (“7”). In contrast, the same inputs to a 4-bit Full Adder will generate a Cout=1 and SUM[3:0]=0001. Design a combinational logic, the shaded area in Figure 3(b), which can convert a Cout and SUM[3:0] from a Full Adder into the BCD style Carryout and sum. (You can use any method you like to design this, but try to minimize the logic using either K-map or Q-M method.) **HINT:** This problem has little to do with an Adder design, it is more a conventional combinational logic design problem. You have to consider the truth table of those cases when the output of a Full Adder is greater than 9.

6. (30%) Seven-Segment Display. A DVD player display contains 5 different modes: On (ON), Off (OF), Stop (SP), Fast-forward (FF), and Play (PL). Using 2 seven-segment displays, each mode is encoded and illustrated in Figure 6(b). There are 5 push buttons that control the entering of the 5 states. Assume that when a button is pushed, the state will stay until another button is pushed. At any given time, only one button can stay pushed (or enabled). If more than one button is pushed simultaneously, the priority is determined as specified in Figure 6(b) and only the one with the highest priority will be enabled. To display correctly, you have to design a Decoder to convert a given push to the 14 control signals (A to G, 2 bits each) that control the 2 seven-segment displays. Please show and minimize your logic design of such a decoder. You can use any basic gates.

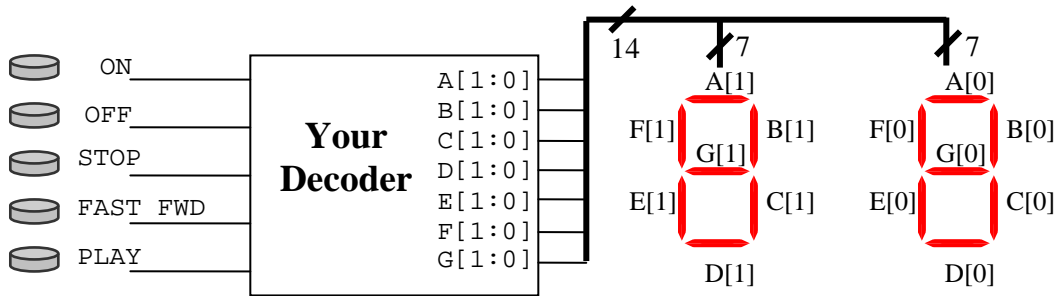


Figure 6(a)



Figure 6(b)