Indian Institute of Technology (IIT-Bombay)

AUTUMN Semester, 2025
COMPUTER SCIENCE AND ENGINEERING

CS230: Digital Logic Design and Computer Architecture

Tutorial - I

Full Marks: 0

Time allowed: ∞ hours

1. Determine the canonical sum-of-products representation of the following functions:

(a) $f(a,b,c) = c' + (a+b')(a'+b)$

(b) $f(a,b,c) = a' + (ab + a'c')'$

(c) $f(a,b,c) = (a+b)(a'+c) + bc'$

2. Consider the Karnaugh map given below in fig 1, where X represents "don't care" and blank represents 0.

dc \ ba	00	01	11	10
00		X	1	
01	1			X
11	1			1
10		X	X	

Figure 1

Assume for all inputs (a,b,c,d), the respective complements $(\bar{a},\bar{b},\bar{c},\bar{d})$ are also available. The above logic is implemented using 2-input NOR gates only. The minimum number of gates required is ____________

3. Show that $f(X,Y,Z) = XY'Z' + X'Y + YZ'$ is a universal operation.

4. Design an 8×1 multiplexer using a combination of 2×1 multiplexers.

5. Find the simplest function $h(A, B, C, D)$ that will make the function

$$f = A'BC + (AC + B)D + h(A, B, C, D)$$

self-dual.
Hint: The dual of a Boolean function can be obtained by replacing all the ANDs with ORs and vice versa. The constant terms are replaced with their respective complements. Boolean function can be self-dual if the truth table contains the same number of 0's and 1's, and none of the minterm pairs are mutually exclusive (that is, no two minterms are complements to each other).

6. You are given three-input logic gates, each realizing the function $g(x, y, z) = x \oplus yz$. Use this function as a block to implement the following function

$$f = (a + b)c + ab'.$$

Prove/disprove that by using these gates only, you can realize any Boolean function. Show all your steps.

7. Design a circuit that takes in a 3-bit signed (2's complement) number X and produces an output $Z = X^2 + 2X + 1$

8. Design a 1-bit comparator that takes in a bit X and a bit Y and outputs $X < Y, X > Y, X = Y$. Use a single 2-to-4 decoder and 1 single OR gate

9. A new flip-flop, called MN flip-flop, is constructed from a JK flip-flop as follows:

$$J = M, \quad K = N \oplus M$$

(a) Derive the state-transition table and excitation table for this new flip-flop (b) Derive the characteristic equation (c) Construct a D flip-flop from this new flip-flop (d) Construct a T flip-flip from this new flip-flop

10. Construct a counter for the following sequence using T flip-flops: 000 → 010 → 111 → 100 → 011. (a) Draw the state-table (b) Construct the next-state map (c) Derive the inputs of the T flip-flops (d) Draw the final circuit

11. A sequential circuit has one flip flop Q, two inputs X and Y and one output S. The circuit consists of a full subtractor circuit connected to a D flip flop as shown in Figure 2 below. Derive the state table and state diagram for the sequential circuit.

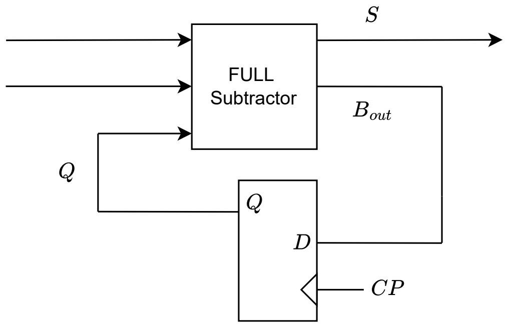
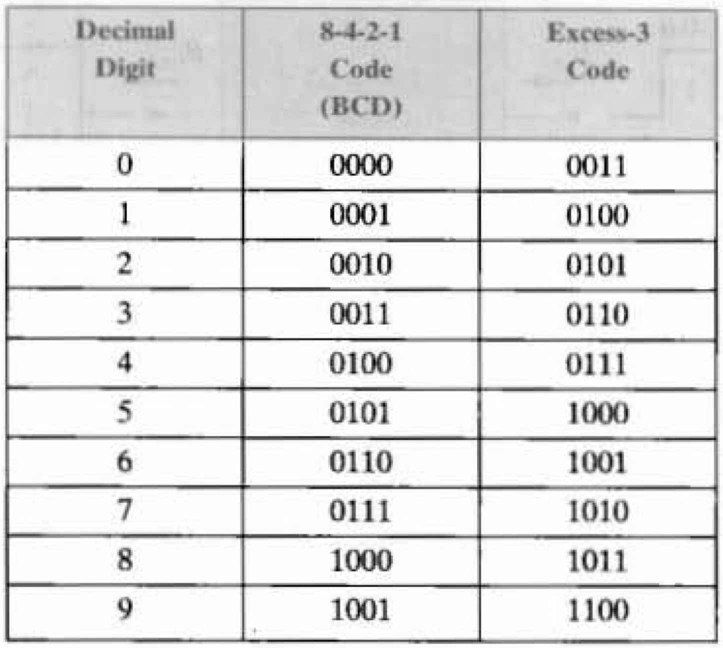


Figure 2

12. In this question we construct a serial BCD to Excess-3 code converter from serial input. In other words, the input will be provided, and the output will be generated one bit at a time. Excess-3 code can be generated by adding $(0011)_2$ with the BCD code. Therefore, you need to consider 4 bits of input at a time as a valid BCD code. The expected input and outputs provided in Figure 3. Observe that the inputs will be provided from LSB to MSB and the outputs will be generated in the same manner (more precisely, the LSB of the input will be processed first and the LSB of the output will be generated first). So you have to consider each entry in the table from right to left while processing.

- In this question you will be generating a circuit for this state machine. First generate the state-transition graph and the state-transition table. *Hint:* The state-transition graph will have the states as the nodes and transitions of the form input/output as the links. The number of states can be quite large in this case. To keep the number of states low, we use some tricks.
  - **Observation 1:** Since the input is of 4 bits, you need to transit through 4 states for each input.
  - **Observation 2:** Let S_t be any state processing the i-th bit of input for some string. It will have two possible transitions out of it for $x_i = 0$ and $x_i = 1$. Observe (from the table) that, for any bit location i, the output $y_i = \overline{x_i}$ or $y_i = x_i$ for $x_i \in \{0, 1\}$. In other words, the two transitions from any given state will be either (0/1), (1/0) or (0/0), (1/1). There will be no other transitions, such as, (0/1), (1/1).
  - **Observation 3:** There can be total 16 possible states. But many of these states will be equivalent and, therefore, can be merged together.
  - **Trick 1:** There will some state transitions which are impossible. Find them out and add dummy transitions for them keeping observation 2 in mind.
  - **Trick 2:** The initial state S_0 will also be the final state.
  - **Trick 3:** Eliminate the equivalent states. Two states S_i and S_j are equivalent if and only if for every possible input sequence, the same output sequence is produced, regardless of whether S_i or S_j is the starting state. This much should be sufficient for you to solve the problem.

(b) Perform the state assignment using binary encoding. If you have any unused state, use it as don't care.

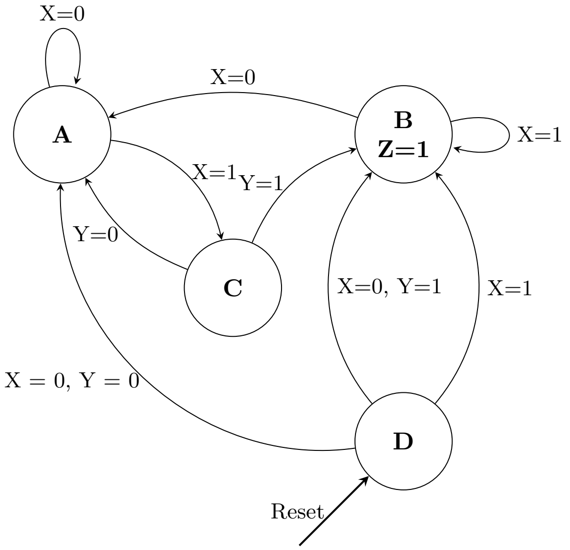


Decimal Digit	8-4-2-1 Code (BCD)	Excess-3 Code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Figure 3: BCD to Excess-3 Code Converter

13. Design a circuit that detects the signal 1010 or 101 in a sequence of bits. For example, when given a sequence of 101011011110001 as an input, the output has to be 001100010000001. Use any flip-flop possible.

14. Design the circuit for the following finite state machine with inputs X and Y. Define a method to represent the states in the circuit


X=0
X=0
A
B
Z=1
X=1
X=1
Y=1
Y=0
C
X=0, Y=1
X=1
X = 0, Y = 0
D
Reset