Name:

Entry Number:

Department of Computer Science and Engineering, IIT Delhi
Digital Logic and System Design (COL 215)
I Semester 2023-24, Major Exam, Maximum Marks: 65, 24 Nov 2023, 4:00 PM to 6:00 PM
Please write your answers ONLY IN THE SPACE BELOW THE QUESTIONS
Use reverse side of paper for rough work.

1. **[5 Marks]** Consider the following two methods of initialising a VHDL variable in a design that is targeted for synthesis into hardware:
   a. Using the variable initialisation syntax
   b. Using a RESET signal as an input port

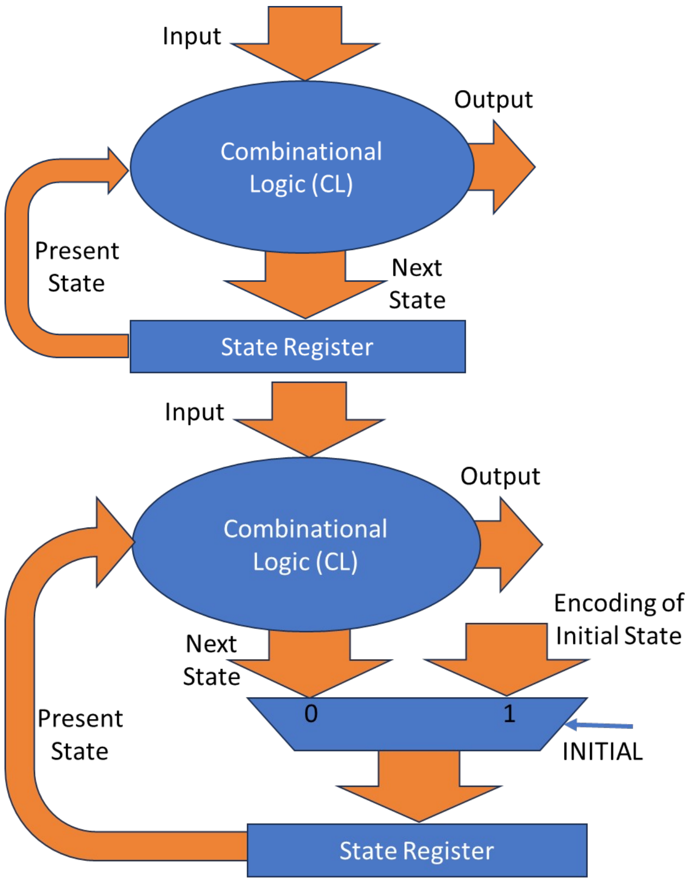
   What advantage does the second method above have over the first?

   Reset can be performed at any time during execution, not just once during Power-Up.
   Reset can be controlled by the user.

2. **[5+5+5 = 15 Marks]** Answer the following questions regarding Finite State Machines (FSMs).
   a. Why should we encode the states of a Finite State Machine?
   So that we can implement it. If hardware, we would first generate a truth table or other forms, and then combinational logic (if software, we would generate appropriate comparisons with the codes, etc.)

   b. Suppose we observe that a subset S of FSM states are UNREACHABLE from other states. No path exists from other states to any state in S. Would we be justified in deleting from the FSM all the states in S and their connecting transitions? Explain your answer.
   No. One of them might be an INITIAL/RESET state, in which case it is reached upon initialisation, even though it is never reached later.

   c. Consider an FSM implemented as combinational logic and flip-flops shown below. We would like to implement an INITIAL STATE feature in the FSM: when the INITIAL input signal is ON, we set the FSM to a known state whose encoding is given. Draw a modified circuit that achieves this. You are allowed to add new elements to the circuit, but cannot change the state register or the combinational logic block (CL) given in the figure.



3. **[5 Marks]** In the error correction strategy using Hamming Codes, the 8 bits were expanded into 12 bits by adding 4 parity bits. In our numbering of the bits, why did we start from position 1 instead of position zero?
Position zero has special meaning: this parity position of the C bits indicates “no error”.

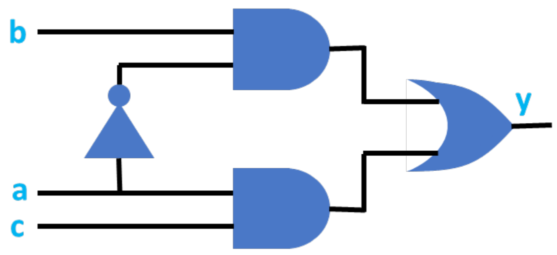
4. **[5 Marks]** Why is the post-silicon validation step needed? Can’t we just do simulation, followed by testing of the manufactured chip?
When a chip is complex, the post-silicon validation step is used to validate the chip on a few lab samples; if major errors are discovered at this stage, a re-design may be required. If testing is done directly after a chip is manufactured in volume, then re-designing after a major error could be very expensive.

5. **[10 Marks]** Do you think the D Algorithm for automatic test pattern generation would succeed in generating a suitable test vector for testing Stuck-At-0 or Stuck-At-1 faults for ANY combinational circuit? Justify your answer. Assume that there is only one stuck-at fault in the circuit.
No. If multiple paths starting from the stuck-at point re-converge at a gate, we may not be able to propagate the D value. Recall that, in order to propagate the D value at a gate, we need to be able to fix the values of ALL

other inputs of the gate.

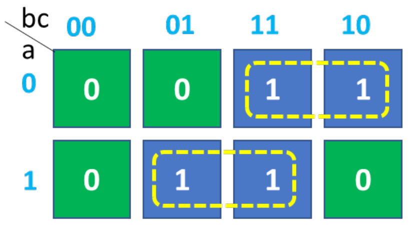
6. **[5 Marks]** When writing HDL for a design that is synthesised to combinational logic, it is recommended that variables in a process should be initialised before any branches occur in the control flow. Why?
If not initialised, it is possible that a variable might be assigned in one branch but not in the other, which would require the system to remember the previous value of the variable, making the synthesised circuit sequential.

7. **[5+5+5 = 15 Marks]** Consider the combinational circuit shown below.



   a. The circuit could exhibit a hazard/glitch at its output **y** when one of its input values changes. Under what condition would this occur?
   b = c = 1, and a goes from 1 to 0. Because the inverter takes non-zero time, output y (originally 1) would fall to 0 first, then rise to 1.

   b. Hazard conditions such as the above, could be noticed by observing the Karnaugh Map of the Boolean function. The K-Map corresponding to the above circuit is shown below, along with the minterm grouping that leads to the circuit. How would we infer the presence of a hazard from the K-Map?



a \ bc	00	01	11	10
0	0	0	1	1
1	0	1	1	0

y = a’b + ac

   Adjacent cells corresponding to **a’bc** and **abc** belong to different product terms, which might cause a hazard when **a** changes.

   c. The hazard detected in the above example could be fixed by adding some logic (which might make the circuit non-minimal). How would we fix the above circuit to make it hazard free?
   Add a product term **bc**, so that the transition between adjacent cells occurs within the same product term (which is independent of **a**. To make the circuit hazard-free, we implement: **y = a’b + ac + bc**

8. **[5 Marks]** Would splitting a sequential design into 100 pipeline stages result in the circuit now being able to run correctly on a clock with 100 times the original frequency? Why or why not?
No. Flip-flop delays such as setup time and D-to-Q propagation delay will place limits on the minimum clock period for the circuit to work correctly.