Name:	Entry Number:

Department of Computer Science and Engineering, IIT Delhi
Digital Logic and System Design (COL 215)
I Semester 2024-25, Mid-semester Exam, Maximum Marks: 60, 17 Sep 2024, 8:00 AM to 10:00 AM
Please write your answers ONLY IN THE SPACE BELOW THE QUESTIONS.
Use reverse side of paper for rough work.

1. **[4+4=8 Marks]** Consider the following 4 messages:
   A. Do not (drink and drive)
   B. Do not (drink or drive)
   C. (Do not drink) and (do not drive)
   D. (Do not drink) or (do not drive)
   Encode the messages as Boolean expressions, with appropriate interpretations for each Boolean variable.
   p: drink
   q: drive
   A: (pq)’ B: (p + q)’ C: p’q’ D: p’ + q’

   Do the 4 messages convey different meanings? Justify.

   A and D are identical.
   B and C are identical

2. **[5 Marks]** Consider a K-map with 0/1/x in its cells, from which minimal Boolean expressions are derived in two different ways: grouping 0's vs grouping 1's. Will the resulting Boolean functions be identical? Why?

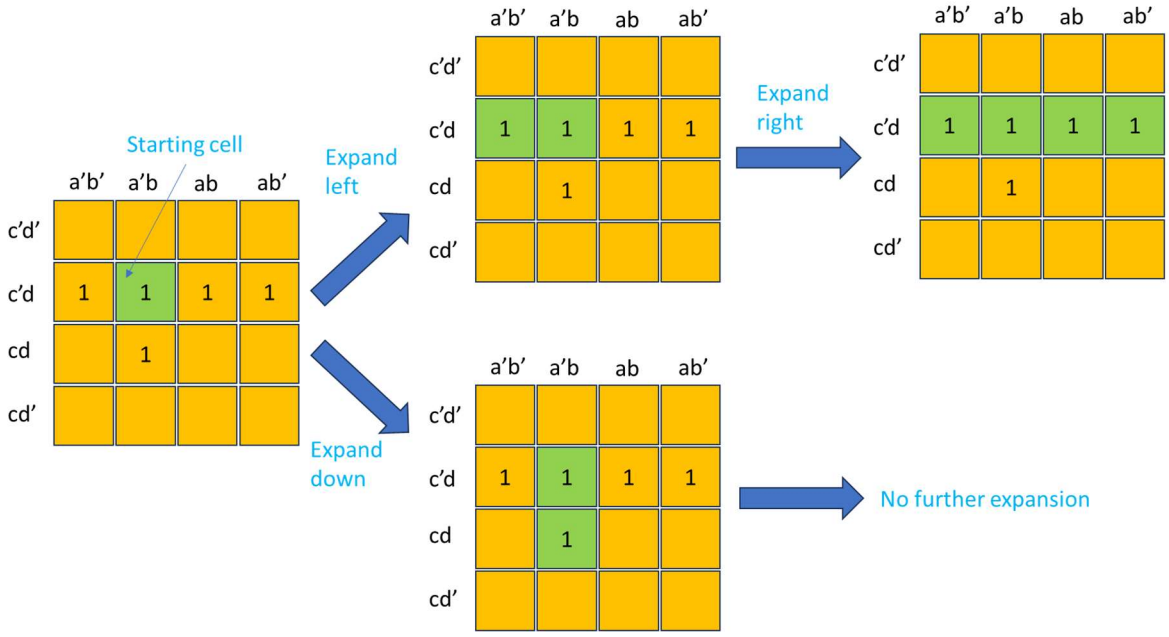
   They may not be identical.
   Don’t cares may be interpreted differently.

3. **[5 Marks]** The number 72 is not considered a valid representation in Base 5. Why? Can't we just interpret it as: $7 \times 5^1 + 2 \times 5^0 = 37_{10}$? Explain.

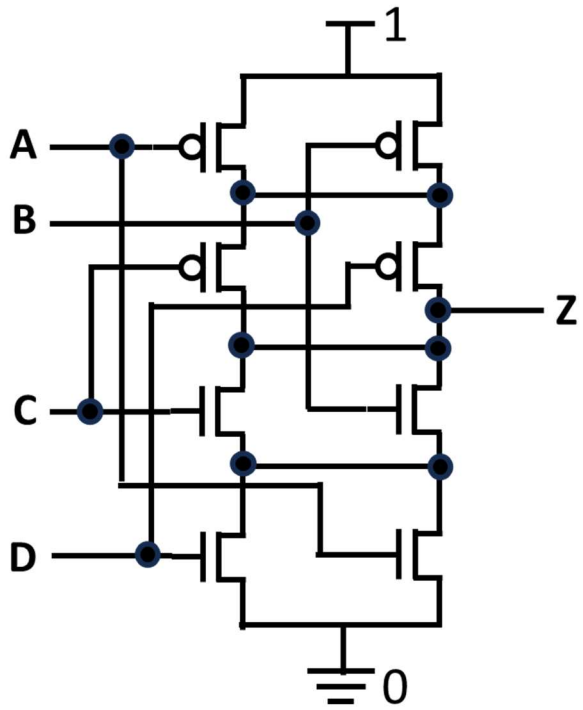
Such interpretations would lead to the same integer having multiple representations. For example, 37_{10} could also be represented as: $122_5 = 1 \times 5^2 + 2 \times 5^1 + 2 \times 5^0 = 37_{10}$

4. **[8 Marks]** In the K-map minimisation process, we start with a region consisting of an individual cell (corresponding to a minterm) and keep expanding the region until it cannot be expanded further. Does the final region depend on the order in which the expansion happens (e.g., left/right/up/down in the 4-variable K-map)? Explain.

Yes. Different orders may lead to different final regions. Example below.



5. **[8 Marks]** What function of A, B, C, and D does the following CMOS transistor circuit implement? Justify. [Note that crossing wires are connected only when there is a black circle at the intersection]



This is an invalid CMOS circuit, as the n- and p-sections are not complementary. For example, for A=1, B=0, C=1, and D=0, there is a path to the output Z from both V_{dd} and Ground, resulting in a short circuit between V_{dd} and Ground.

6. **[8 Marks]** Can the truth table for the NOT function be inferred from the postulates of Boolean algebra alone? Justify.

Yes, it can be inferred from postulates alone. We need to show:
$(1)\ x = 0 \Rightarrow x' = 1$ and $(2)\ x = 1 \Rightarrow x' = 0$

$(1)\ x = 0$
$\Rightarrow x' + x = x'$ (2a: Identity)
$\Rightarrow 1 = x'$ (5a: Complement)

$(2)\ x = 1$
$\Rightarrow x'x = x'$ (2b: Identity)
$\Rightarrow 0 = x'$ (5b: Complement)

Other arguments are possible.

7. **[8 Marks]** We understood that the delay through an *n*-bit Carry Look Ahead adder varies as *O(log n)*. Its area varies as what function of *n*? Justify. Consider only gate area, and ignore wires.

Number of P, G blocks: one per bit position. O(n)
Number of C/L blocks: $\frac{n}{4} + \frac{n}{16} + \frac{n}{64} + \ldots < \frac{n}{4} \left(\frac{1}{1-1/4}\right)$ = O(n)

8. **[5 + 5 = 10 Marks]** Consider the simulation of a set of VHDL processes, each with an associated sensitivity list of signals (no wait statements). The heart of the simulation algorithm has processes being executed in response to events on signals. Two possible algorithms are sketched below. The algorithms are not intended to be complete; they are an abstract view of what happens when events occur on signals.

Algorithm 1

S = all signals on which events occurred just now (same time/delta)
For each signal *s* in *S*
  P_s = all processes sensitive to signal *s*
  For each process *p* in P_s
    EXECUTE *p*

Algorithm 2

For each process *p* in the design
  S_p = all signals in sensitivity list of *p*
  For each signal *s* in S_p
    If event has occurred on signal *s* just now (same time/delta)
      EXECUTE *p*

Is Algorithm 1 a fair representation of the simulator's working? Why?

No. If a process is sensitive to two signals, it might be executed twice in the same delta if both signals change value. This is not allowed because processes have state (local variables), which might get updated twice.

Is Algorithm 2 a fair representation of the simulator's working? Why?

No. Same reason as above.