

Roll No. 

--	--	--	--	--	--	--	--	--	--

Total No. of Pages : 02

Total No. of Questions : 18

B.Tech (CSE) (Sem.-3)

**DIGITAL CIRCUITS & LOGIC DESIGN**

Subject Code : CS-205

M.Code : 56503

Time : 3 Hrs.

Max. Marks : 60

**INSTRUCTIONS TO CANDIDATES :**

1. SECTION-A is COMPULSORY consisting of TEN questions carrying TWO marks each.
2. SECTION-B contains FIVE questions carrying FIVE marks each and students have to attempt any FOUR questions.
3. SECTION-C contains THREE questions carrying TEN marks each and students have to attempt any TWO questions.

**SECTION-A****Write briefly :**

- 1) Explain the working of OR, AND and NOT gates.
- 2) Distinguish between SOP and POS forms.
- 3) What are minimization techniques?
- 4) What do you mean by toggle?
- 5) What are Min and Max terms?
- 6) What is the use of A/D and D/A conversions?
- 7) What are shift registers?
- 8) Draw the circuit diagram of a 8-input multiplexer.
- 9) What is the race condition in S-R flip-flop? How is it resolved in D-flip-flop?
- 10) Define a Flip-Flop with a suitable sketch?



**SECTION-B**

- 11) What is PLA? Design BCD to Excess-3 code convertor using PLA.
- 12) What is a ROM? What are its types? Give characteristics of each.
- 13) Convert D flip-flop to JK flip flop.
- 14) Draw a circuit for the 4-input successive operation A/D convertor. Discuss its limitations.
- 15) Reduce the following function using Karnaugh map technique :

$$F(A,B,C,D) = \sum m(5,6,7,12,13) + \sum d(4,9,14,15)$$

**SECTION-C**

- 16) Explain the step-wise process for the sequential circuit design using state tables.
- 17) What is a Programmable logic devices? What are their advantages? Explain in detail the architecture of a programmable logic device.
- 18) a) Implement the function

$$F = A'BC + ABC' + A'BC' + AC' \text{ using PAL.}$$

- b) With the help of a diagram explain the working of R-2R ladder type DAC.

**NOTE : Disclosure of identity by writing mobile number or making passing request on any page of Answer sheet will lead to UMC case against the Student.**