

U.G. 1st Semester Examination - 2021**COMPUTER SCIENCE****[HONOURS]****Course Code : COM.SC-H-CC-P-102****(Digital System Design Lab)****[PRACTICAL]**

Full Marks : 75

Time : 4 Hours

Marks Distribution :

Experiment : 60 marks

Lab Notebook : 5 marks

Viva : 10 marks

Answer any **one** from each group to be allotted on lottery basis.

30×2=60

GROUP-A

1. Design Half-Adder using NAND gates only.
2. Design Half-subtractor using NAND gates only.
3. Design Full Adder using XOR and basic gates.
4. Implement the following function using basic gates.
 $F = \sum m(3,4,5,7,9,13,15).$
5. Design a 2×4 decoder using NAND gates only.

6. Design 2-bit magnitude comparator using basic gates.
7. Design half adder using basic gates only.
8. Assume that a 3-bit message is to be transmitted with even parity. Design a circuit for even parity generator.
9. Design and implement subtractor multiplexer IC's of your choice.
10. Implement the basic gates using universal gates.
11. Design 2-bit magnitude comparator using universal gates.
12. Design 4×1 MUX using basic gates.
13. Design and implement full-subtractor using universal gates only.
14. Implement 2-input XOR operation using Universal gates only.
15. Implement the following function using NOR gates only.
 $F(A,B,C) = \prod m(0,1,4,6).$
16. Design a 4-to-2 priority encoder using basic gates.
17. Implement the following function using 4×1 Mux and other necessary gates:
 $F(A,B,C,D) = AB'C + A'C'D + AD'$

[Turn over]

18. Design and implement full adder using decoder IC's.
19. Design and implement BCD-to- excess-3code converter.
20. Implement and design 1×4 Demultiplexer using Logic Gates.
21. Realize the following expression using NOR gates only.

$$((A+B) + C)'D$$

GROUP-B

1. Design and implement S-R flip flop.
2. Design a shift register (shift left) using J-K flip-flop.
3. Design and implement J-K flip flop.
4. Design a mod-10 Ripple counter using suitable flip-flop.
5. Implement and verify the truth table for Master-slave J-K flip-flop.
6. Design and implement T flip flop.
7. Design a Mod-8 binary up counter.
8. Design a 3-bit binary up counter.

9. Design and implement D flip flop.
10. Design and realize Johnson's counter.
11. Design and Verify the 4-Bit Synchronous Counter using J-K flip-flop.
12. Design a 4-bit Serial-in to Parallel-out Shift Register.
13. Design and implement D flip-flop using T flip-flop.
14. Design a counter using J-K flip-flop for the following sequence
00 → 110 → 101 → 100 → 010 → 001.
15. Design a counter that goes through the following sequence of binary states: 0, 1, 2, 3, 6, 7 and back to 0 to repeat.

—————