CBCS SCHEME

USN □□□□□□□□□□

17EE35

Third Semester B.E. Degree Examination, June/July 2019
Digital System Design

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. What is a K-map? Which is the code used to number K-map and why? Write a neat numbered 4-variable K-map. **(06 Marks)**

b. Using K-map, evaluate the minimal POS expression of,
$f(a,b,c,d) = \pi M(0,2,4,6,8) \cdot \pi D(1,9,12,15)$ **(08 Marks)**

c. Evaluate the minimal SOP expression using VEM with 'Z' as MEV.
$f(w,x,y,z) = \sum m(0,3,5,7,8,9,10,12,13) + \sum d(1,6,11,14)$ **(06 Marks)**

OR

2 a. What is don't care condition? What are its advantages? **(04 Marks)**

b. Evaluate the canonical SOP expression and hence determine the minimal expression using K-map $F = A\overline{B}\overline{D} + \overline{A}B + \overline{A}C + CD$. **(08 Marks)**

c. Minimize the following SOP expression using Quine McCluskey method:
$f(w, x, y, z) = \sum m(3, 7, 8, 10, 11, 12, 14, 15) + \sum d(2, 6)$ **(08 Marks)**

Module-2

3 a. With a neat diagram, explain the operation of a carry look-ahead adder circuit. **(10 Marks)**

b. Implement the following Boolean function using a 4:1 MUX with a and b as select inputs.
$f(a, b, c, d) = \sum m(4, 5, 7, 8, 10, 12, 15)$ **(06 Marks)**

c. Implement a 2 to 4 decoder using 1 to 2 decoders. **(04 Marks)**

OR

4 a. Design a full adder using 4:1 MUX. **(06 Marks)**

b. Implement a 1-bit comparator using a suitable decoder. **(06 Marks)**

c. With a logic diagram, explain the operation of a decimal to BCD priority encoder. **(08 Marks)**

Module-3

5 a. Explain the operation of an SR latch with a circuit diagram and characteristic table. Justify its application as switch debouncer with a relevant circuit and waveforms. **(08 Marks)**

b. Design a 4-bit shift register using DFFs. Design a twisted ring counter using a 4-bit shift register. **(06 Marks)**

c. Design a mod-11 up ripple counter using TFF. **(06 Marks)**

OR

6 a. What is the problem in SRFF? How is it eliminated in JKFF? Explain with a neat diagram. **(06 Marks)**

b. With a neat logic diagram, explain the different modes of operation of universal shift register. **(07 Marks)**

c. Design a synchronous counter using JKFF with counting sequence 0, 2, 6, 1, 3, 7, 0.... **(07 Marks)**

Module-4

7 a. List out the merits and demerits of synchronous sequential circuit asynchronous sequential circuits. **(04 Marks)**

b. Design a clocked synchronous sequential circuit that operates according to the state diagram shown in Fig.Q7(b). Use DFF in the circuit.

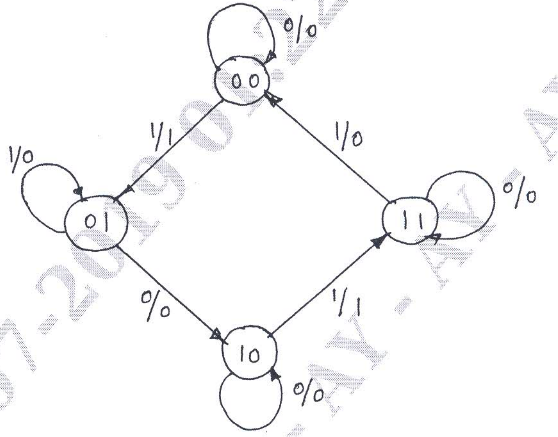

Fig.Q7(b) **(10 Marks)**

c. Analyze the following sequential circuit and obtain: [Fig.Q7(c)]

i) FF input and output equations ii) Transition table

iii) State table iv) State diagram

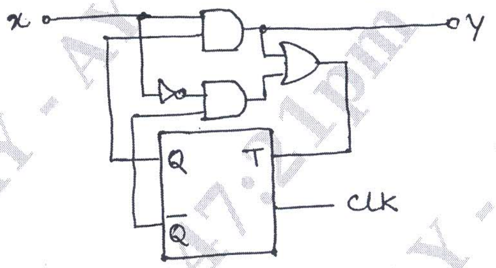

Fig.Q7(c) **(06 Marks)**

OR

8 a. Compare and contrast Mealy and Moore synchronous sequential networks with neat block diagrams. **(04 Marks)**

b. A sequential network has one input and one output. The state diagram is as shown in Fig.Q8(b). Design the sequential network using TFF.

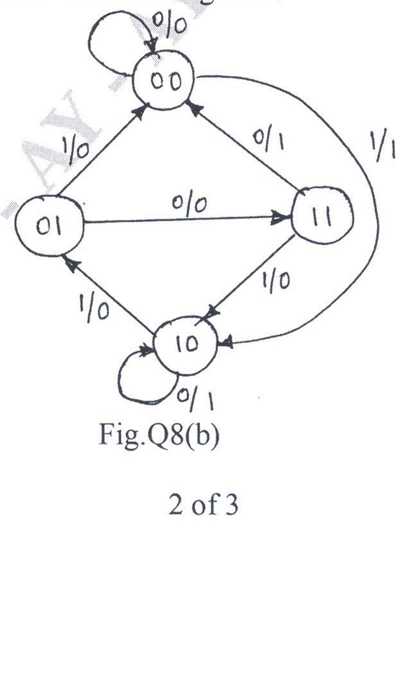

Fig.Q8(b) **(10 Marks)**

c. An edge triggered DFF is connected as shown in Fig.Q8(c). Assuming Q = 0 initially, sketch the output waveform and determine the frequency of the output signal.

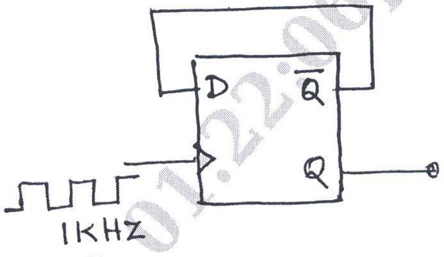

Fig.Q8(c) **(06 Marks)**

Module-5

9 a. Explain entity and architecture with reference to VHDL code of full adder circuit. **(08 Marks)**

b. Write VHDL code to implement 2:1 MUX. **(06 Marks)**

c. List out all operators in VHDL with examples. **(06 Marks)**

OR

10 a. Explain various data types supported in VHDL with examples. **(08 Marks)**

b. Implement a 1-bit comparator either using VHDL code or verilog. **(04 Marks)**

c. Implement a JKFF with active low asynchronous inputs $\overline{pr}$ and $\overline{clr}$ along with clock input using verilog/VHDL. **(08 Marks)**

* * * * *