

Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
THIRD SEMESTER B.TECH DEGREE EXAMINATION(S), MAY 2019

Course Code: CS203

Course Name: SWITCHING THEORY AND LOGIC DESIGN

Max. Marks: 100

Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks.

Marks

- 1 List out any *three* advantages of digital systems over analogue systems. (3)
- 2 Do the following number conversions: (3)
 (i) *base-7* number 3456 to decimal (ii) *base-4* number 1213 to binary.
- 3 Show the *K-map* contents for the following Boolean functions : (i) $F(x,y,z) = (x+y)(y+z)$ (ii) $F(x,y,z) = \Pi(0,3,5,7)$. (3)
- 4 Use *De-Morgan's principle* to find the complement of $A+BC'(D+EF)'$ (3)

PART B

Answer any two full questions, each carries 9 marks.

- 5 a) Do the following operations: (7)
 - (i) Compute *1's complement* of the binary number 1101.01.
 - (ii) Compute *8's complement* of the octal number 672.23.
 - (iii) Add *base-16* numbers 1FE and EF1.
- b) Assume that floating point numbers are represented in the following format. (2)
 The *mantissa* is represented in *sign-magnitude* form. Magnitude of mantissa is adjusted such that the Most significant bit (MSB) is 1 and the (assumed) *binary point* is to the left of MSB.

Sign bit of Mantissa (1bit)	Exponent (6 bits signed-2's complement)	Mantissa (9 bits)
--------------------------------	--------------------------------------------	----------------------

Represent the decimal number 6.25 in binary.

- 6 Use *tabulation method* to identify the *simplified Boolean expression* for the function, (9)
 $F(w,x,y,z) = \Pi(1,3,4,6,9,11,12,14)$.
- 7 a) Use *algebraic manipulation* to convert: (5)
 - (i) $F(x,y,z) = xy+y+z$ into *canonical PoS*.
 - (ii) $F(x,y,z) = (x+y+z)(x'+y+z)(x+y'+z)(x+z)$ into *standard PoS*.
- b) Subtract the BCD number 1671 from BCD number 837 using *10's complement addition*. (4)

PART C

Answer all questions, each carries 3 marks.

- 8 Show how a *master-slave J-K flip-flop* can be realized using NOR and AND gates. (3)
- 9 Write the truth table of a 4x1 de-multiplexer and show the corresponding logic diagram. (3)
- 10 Show how a *full-subtractor* can be implemented using a decoder. (3)
- 11 Realize a *half-adder* using NAND gates. (3)

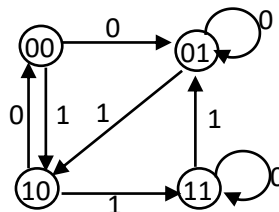
PART D

Answer any two full questions, each carries 9 marks.

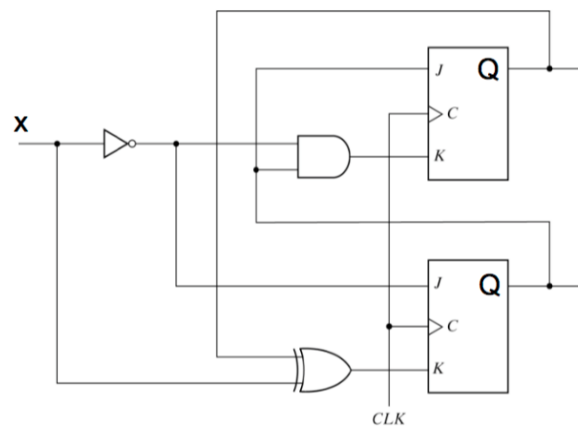
- 12 a) Implement the following Boolean functions using a 2X1 multiplexer and additional gates as needed: $F(x,y,z) = \sum(1,2,4,5)$. (3)
- b) Design a code converter with the following mapping specifications: (6)

Input code	000	001	010	011	100	101	110	111
Output code	001	010	011	100	101	110	111	000

- 13 a) Given a 2-bit subtractor (block diagram), design a circuit with additional gates to use it as a comparator. (3)
- b) Design a sequential circuit for the following *state diagram* using *T flip-flops*. (6)



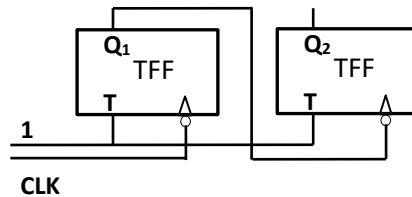
- 14 Deduce the *state table* and *state diagram* that represents the behaviour of the following sequential circuit: (9)



PART E

Answer any four full questions, each carries 10 marks.

- 15 With the help of a neat diagram discuss how a *serial adder* can be designed using full-adder, shift registers and flip-flop. (10)
- 16 Design a *synchronous counter*, using edge-triggered *J-K flip-flops*, that generates the binary sequence: 001, 011, 010, 110, 111, 101, 001, 000, 001, ... (10)
- 17 Draw a *mod-16 ripple up-counter* using J-K flip-flops. Show how this counter can be converted to a *mod-12 ripple counter*. (10)
- 18 a) How is *static RAM* different from *dynamic RAM*? (3)
- b) Write explanatory notes on *read-only memory* and *read-write memory*. (4)
- c) Assuming that both the *T flip-flops* in the diagram below are initially at state 1, show the *timing diagram* for Q_1 and Q_2 with respect to the *falling edge* of the first four clock pulses. (3)



- 19 a) Write a short note on *PLA*. (4)
- b) Implement the following Boolean functions using a *3-by-4-by-2 PLA*. (6)
- (i) $F_1 = \Sigma(1,4,5,6)$
- (ii) $F_2 = \Sigma(0,2,3,4,6,7)$
- 20 Briefly discuss the algorithms for *floating point addition* and *floating point subtraction*. (10)
