Mid-semester Examination marking scheme: F215 Digital Design (Sem-I, 2018-19)

Q1 For the Boolean expression $F(A,B,C,D) = \sum m(1,3,6,7,10,11,13,15) + \sum d(12)$ **[18]**

where $\boldsymbol{A}$ is MSB and $\boldsymbol{D}$ is LSB

a) Plot the K-map.

- 2 marks

b) Identify all Prime Implicants (PI). Express each PI in product form.

- 6 PIs (6 marks); any extra PI (negative 0.5 per PI)

c) Identify all Essential Prime Implicants (EPI). Express each EPI in product form.

- 3 EPIs (3 marks); any extra EPI (negative 0.5 per EPI)

d) Write down expression for the possible minimal SOP of output function **F**.

- 1 mark

e) Realize the minimal SOP using minimum number of 2-input NAND gates only. Only true (unprimed) inputs are available.

- 8 NAND gate (6 marks); any extra NAND gate (negative 0.5 per gate)

Q2 Two four-bit signed numbers $P = p_3p_2p_1p_0$ and $Q = q_3q_2q_1q_0$ are compared by using the substractor circuit which performs the operation $P - Q$ and produces the following three outputs: **[18]**

$W = 1$, if the result of $P - Q$ is zero; otherwise $W = 0$.

$X = 1$, if the result of $P - Q$ is negative; otherwise $X = 0$

$Y = 1$, if the result of $P - Q$ produces an overflow; otherwise $Y = 0$.

Components available for implementation:

1-bit **Full Adder**- 4 Nos., **NOT** gate- 6 Nos., 4-input **NOR** gate- 1 Nos.,

2-input **OR** gate -1 No., 2-input **XOR** gate- 2 Nos.

a. Design the substractor circuit to produce W, X and Y.

- Realization of Substractor: 4 marks
- W (2 marks); X (1 mark); Y (2 marks)

b. Further show how these outputs can be used to produce A, B, C, D and E which indicates the cases $P = Q$, $P < Q$, $P \leq Q$, $P > Q$ and $P \geq Q$ respectively.

- A (1 mark); B, C, D, E (2 marks each)

Q3 A combinational circuit has 4-bit input $\boldsymbol{A} = a_3a_2a_1a_0$ (a_3: MSB and a_0: LSB) and 1-bit output $\boldsymbol{P}$. The output $\boldsymbol{P}$ represents the state of the Parity bit. If Odd-Parity is being used, the 4-bit data and the parity bit should add-up to give odd number of 1's. If Even-Parity is being used, the 4-bit data and the parity bit should add-up to give even number of 1's. Design an Even-Parity bit generator circuit assuming that the Parity bit ($\boldsymbol{P}$) is set to 0 when the 4-bit data input combinations have all zeros. **[18]**

Components available for above realization:

2 to 4-line **decoder** with active low enable- 1No., 2:1 **Mux** with active high enable- 1No., 4:1 **Mux** with active high enable- 1No., NOT gate- 1No.

- Truth table (2 marks); K-map (2 marks); Equation (2 marks)
- Correct Realization (12 marks); Correct realization with any extra component (5 marks)

Q4 A positive edge triggered D flip-flop with input D, output Q ($Q\ and\ \bar{Q}$) and clock is available. **[18]**

a) Convert the given D flip-flop into positive edge T flip-flop with **synchronous** reset (clear) pin $\boldsymbol{R}$ (active high, i.e. $R = 1, Q = clear$). Realize the design using basic gates around D flip-flop. Draw the symbolic block of designed $\boldsymbol{T}$ flip-flop with input $\boldsymbol{T}$, output Q ($Q\ and\ \bar{Q}$), clock and reset (clear) pin $\boldsymbol{R}$.

- Design and realization **with synchronous reset** (6 marks, else 0 mark)
- Symbolic block **with synchronous reset** (2 marks, else 0 mark)

b) Construct the truth table and derive the characteristic equation for designed $\boldsymbol{T}$ flip-flop.

- Truth table (2 marks); Characteristics equation (2 marks)

c) Using the symbolic block of designed $\boldsymbol{T}$ flip-flop, construct a 3-bit binary Mod 5 UP-counter which counts $0 -\ 1 -\ 2 -\ 3 -\ 4$ and back to 0 and repeat. 3-bit counter outputs are Q_A, Q_B and Q_C (Q_A: MSB and Q_C: LSB). Only two, 2-input **AND** gates are available for design. Clearly mention the counter output pins and show the counter reset logic.

- Counter designed and realized using **synchronous reset** (6 marks, else 0 mark)

Q5 Design synchronous non-sequential counter using positive edge triggered $J-K$ flip-flop which counts $0-4-7-2-3$ and repeat assuming that there is no constraint on the unused states (i.e. NEXT state is don't care). Is the design self-correcting? Justify. **[18]**

Counter outputs are Q_A, Q_B and Q_C (Q_A: MSB and Q_C: LSB)

Counter Inputs are $J_A - K_A$, $J_B - K_B$ and $J_C - K_C$ ($J_A - K_A$: MSB and $J_C - K_C$: LSB)

- Design table: 3 marks
- Input equation for $J_A - K_A, J_B - K_B$ and $J_C - K_C$ (1 mark each) (total 6 marks)
- Correct realization (clock, input, MSB/LSB, output) 7 marks
- Self-correcting with justification: 2 marks