I. Given a three-input Boolean function \( f(a, b, c) = \sum m(0, 2, 4, 6, 7) + \sum d(1) \).
   a. Implement the function using a minimal network of 2:4 decoders and OR gates.
   b. Implement the function using a minimal network of 4:1 multiplexers.
   c. Implement the function using a minimal network of 2:1 multiplexers.

II. Adders: Draw the logic diagram to show the following designs.
   II.(1). Design a full adder with a minimal number of 2:1 multiplexers (no other gates). Draw the schematic diagram.
   II.(2). A sequential three-at-a-time adder inputs \( a_i, b_i, e_i \), the \( i \)'th bit of three binary numbers in each clock cycle for \( i = 0 \) to \( n - 1 \) and outputs the sum. Implement the adder with a minimal numbers of Half Adders, OR gates and two D flip-flops.
   III. A sequential adder inputs \( a_i, b_i \), the \( i \)'th bit of two binary numbers in each clock cycle for \( i = 0 \) to \( n - 1 \) and outputs the sum. Implement the adder with a JK flip-flop, and a minimal AND-OR-NOT network (if the network is needed). Draw the schematic diagram.

IV. Given modulo-16 counters, draw the logic diagram to show the following designs.
   IV(1). Design a module-200 counter with a repeated output.
   IV(2). Design a counter with a repeated output sequence 15, 0, 1, 2, 8, 9, 10, 6, 7, with a modulo-16 counter and a minimal combinational network. Write the Boolean expression and draw the schematic diagram.

V. Design a counter with a repeated output sequence 0, 1, 2, 4, 5, 6, 3, with a modulo-8 counter and a minimal AND-OR-NOT network. Write the Boolean expression and draw the schematic diagram.

VI. System Designs:
   Implement the following algorithm:
   Alg(X, Y, Z, start, U, done);
   Input X[7:0], Y[7:0], Z[7:0], start;
   Output U[7:0], done;
   Local-object A[7:0], B[7:0], C[7:0];
   S1: If start' goto S1;
   S2: done<= 0 || A<= X || B<= Y || C<= Z;
   S3: A<= Add(A,B);
   S4: If B'[7] goto S3 || B<= Inc(B);
   S5: If C'[7] goto S3 || C<= Inc(C);
   S6: U<= A || done<= 1 || goto S1;
   End Alg
   VI(1). Design a data subsystem that is adequate to execute the algorithm. Draw the schematic diagram to show the design.
   VI(2). Design the control subsystem (i) draw the state diagram; (ii) draw the logic diagram that implements the control subsystem with a one hot encoding design.