1. Simplify the following Boolean function and find its SOP:
2. F(a,b,c,d) = П (0,1,3,7,8,9,10,13,15) + d(5,11)
3. F(w,x,y,z) = Σ(5,6,7,12,14,15) + d(13,9,11)
4. A combinational switching circuit has 4 inputs (A, B, C, D) and one output (Y). The value of Y=1, if three or four of the inputs are zero.

i. Obtain the truth table.

ii. Apply the k-map to find the minimal SOP.

iii. Realize using basic gates.

3. A digital system is to be designed in which the month of the year is given as input in four-bit form. The month January is represented as ‘0000’,February as ‘0001’ and so on. The output of the system should be ‘1’ corresponding to the input of the month containing 31 days or otherwise it is ‘0’. Consider the excess numbers in the input beyond ‘1011’ as don’t care conditions. For this system of four variables (A,B,C,D),find the following:

i)Boolean expression in Ʃm and M form.

ii)Write the truth table.

iii)Using K-map,simplify the Boolean expression of canonical min term form.

iv)Implement the simplified equation using NAND gates.

4. Switching circuit has two control inputs (C1 and C2), two data inputs (X1 and X2) and one output Z. The circuit performs one of the logic functions such as OR, XOR, AND, EQU for control inputs combination C1, C2 as 00,01,10,11 respectively:

(i) Derive the truth table for Z

(ii) Use a K-map to find minimum AND-OR gate circuit to realize Z.

1. Minimize the following function for **POS** using K-map and implement it using basic gates:

F(a,b,c,d)= П M (6,7,9,10,13)+d(1,4,5,11)

1. Apply K-map and draw the circuit using gates for the following Boolean expressions.

f(A, B, C,D)=A’B’C’D’+AB’CD+ A’BCD+ABCD+ ABCD’

1. Obtain all prime implicants using Quine-McCluskey method for the function

f(a,b,c,d) = Ʃ m (0,1,10,11,13,15) + d(2,3,12,14). Identify all essential prime implicants.

1. Apply Quine-McCluskey method to find the essential prime implicants for the Boolean expression f(a,b,c,d) = Ʃ m (1,3,6,7,9,10,12,13,14,15)
2. Design the EX-OR gate using (ii) NAND gates (ii) NOR gates and write the HDL Verilog Code for EX-OR Gate.
3. Design an HDL code for the given circuit: Using Structured level and Data Flow Modelling. Implement the function using universal gates.

A diagram of a diagram

Description automatically generated

1. Implement the full-subtractor using 3-to-8 decoder
2. Realize the Boolean expression f(w,x,y,z)=Ʃm (0,1,3,5,7,11,12,13,14) using a 8 to 1 line multiplexer.
3. Design a 1-to-16 De-multiplexer using two 1-to-8 De-multiplexer and one 1-to-2 De-multiplexer.
4. Design a 1-bit comparator using basic gates.
5. Analyse the truth table and design the priority encoder.

The order of priority of inputs is X1>X2>X3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S | X1 | X2 | X3 | A | B |
| 0 | X | X | X | 0 | 0 |
| 1 | 1 | X | X | 0 | 1 |
| 1 | 0 | 1 | X | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |

1. Construct Decimal Adder.