

Exam #1 Solutions

Oct 26, 2009

1. (10 points) Express -36_{10} as an **8-bit twos-complement** binary number.

$$\text{Answer: } \underline{\mathbf{1\ 1\ 0\ 1\ 1\ 1\ 1\ 0}}$$

msb lsb

2. (10 points) Find the complement of F. Do **not** simplify the expression.

$$F = (A + B')(A'D' + E) + (AC' + BD)[(C + E')(B' + D) + 0]$$

Using the set of rules for complement, this turns out to be a one-step process. We get

$$F' = [A'B + (A + D)E'] [(A' + C)(B' + D') + (C'E + BD')1]$$

3. (10 points) Find the dual of F. Do **not** simplify the expression.

$$F = AB + (A' + C)(B + E')[D'E(A' + B) + 1]$$

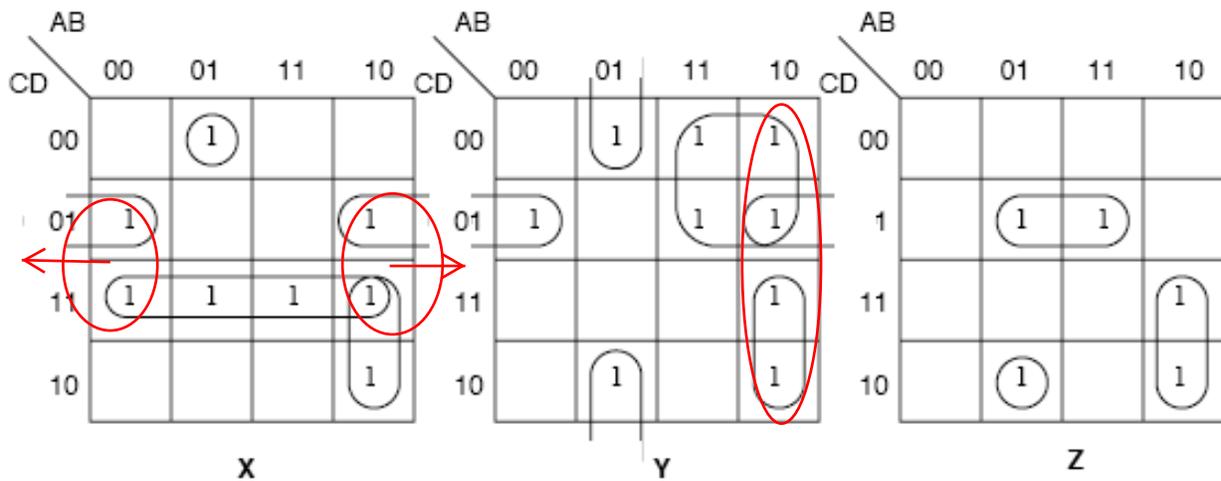
Using the set of rules for duals, this turns out to be a one-step process. We get

$$F_D = (A + B)[A'C + BE' + (D' + E + A'B)0]$$

4. A leading medical electronics company is designing a logic network that will perform preliminary diagnosis on patients based on the presence or absence of four symptoms ABCD. The diseases to be diagnosed are XYZ, and their relationship with the symptoms is as follows —

ABCD	XYZ	ABCD	XYZ
0000	000	1000	010
0001	110	1001	110
0010	000	1010	111
0011	100	1011	111
0100	110	1100	010
0101	001	1101	011
0110	011	1110	000
0111	100	1111	100

a) (15 points) Draw the K-maps for X, Y and Z to implement this network as a 4-input 3-output system:

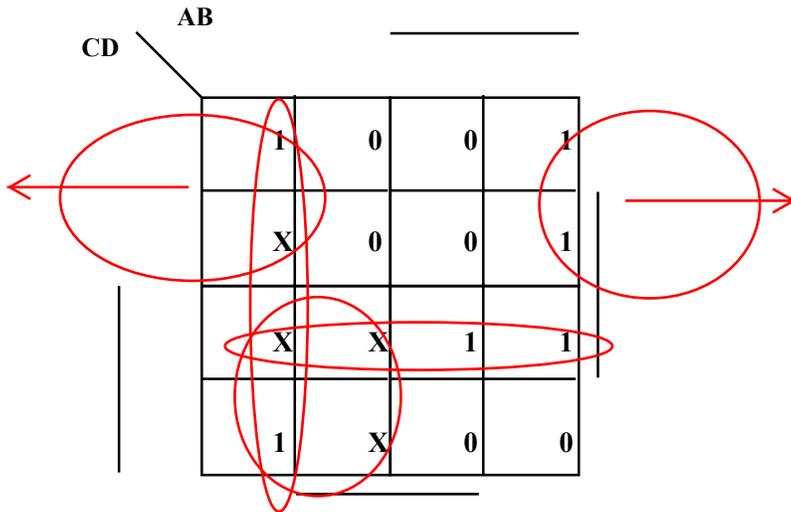


b) (15 points) Find the minimum 2-level AND-OR expressions to implement this system:

$$\begin{aligned}
 X &= CD + AB'C + A'BC'D' + B'D \\
 Y &= A'C + AB'D + A'BD' + B'C'D \\
 Z &= AB'C + A'BCD' + BCD
 \end{aligned}$$

5. Given the function: $F(a, b, c, d) = \sum m(0,2,8,9,11,15) + D(1,3,6,7)$

- a) (10 points) Draw and label a K-Map of the function and circle the elements of the on-set that correspond to a minimized expression for F



- b) (15 points) Find the minimum SOP form of the function.

$$\mathbf{A'B' + B'C' + CD \quad \text{or} \quad A'C + B'C' + CD}$$

- c) (15 points) Write a verilog module that implements your SOP form:

```

module somename (X,A,B,C,D);
  input A,B,C,D;
  output X;
  assign X = ~A&~B|~B&~C|C&D;
endmodule

```

or

```

module somename (X,A,B,C,D);
  input A,B,C,D;
  output X;
  assign X = ~A&C|~B&~C|C&D;
endmodule

```