Combinatorial Logic Project 2
Lab 3 - RBN - 2306173113 - Athazahra Nabila Ruby - KKI

Design Procedure

1. Specification

- BCD to Excess-5 converter
- Transforms BCD code for the decimal digits to Excess-5 for the decimal digits
- BCD code words for digits 0 through 9: 4-bit patterns 0000 to 1001, respectively
- Excess-5 code words for digits 0 through 9: 4-bit patterns consisting of 5 (binary 0101) added to each BCD code word
- Implementation:
  - NOR gates

2. Formulation

Variables:

- BCD (input): A, B, C, D
- Excess-5 (output): W, X, Y, Z

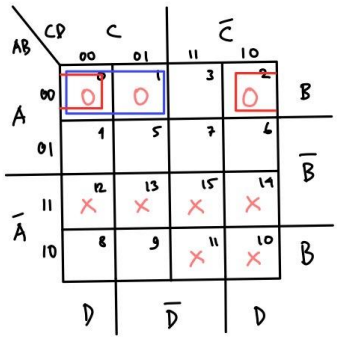
Don't Cares: BCD 1010 to 1111

Truth Table

BCD				Excess-5			
A	**B**	**C**	**D**	**W**	**X**	**Y**	**Z**
0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	1	1	0	1
1	0	0	1	1	1	1	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

3. Optimization

POS of W



Optimized POS

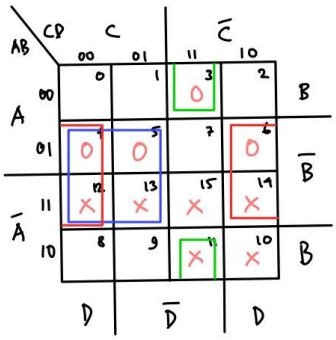
$W = (A+B+C) \cdot (A+B+D)$

double complement

$\overline{\overline{W}} = \overline{(A+B+C) \cdot (A+B+D)}$

$W = \overline{(A+B+C)} + \overline{(A+B+D)}$

POS of X



Optimized POS

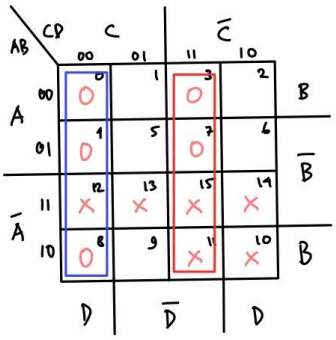
$X = (\bar{B}+C) \cdot (\bar{B}+D) \cdot (B+\bar{C}+\bar{D})$

double complement

$\overline{\overline{X}} = \overline{(\bar{B}+C) \cdot (\bar{B}+D) \cdot (B+\bar{C}+\bar{D})}$

$X = \overline{(\bar{B}+C)} + \overline{(\bar{B}+D)} + \overline{(B+\bar{C}+\bar{D})}$

POS of Y



Optimized POS

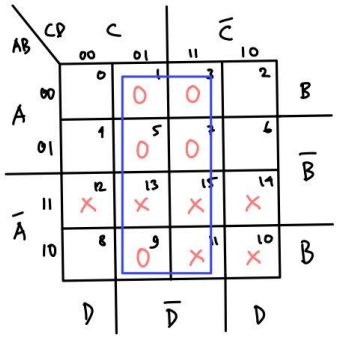
$Y = (C+D) \cdot (\bar{C}+\bar{D})$

double complement

$\overline{\overline{Y}} = \overline{(C+D) \cdot (\bar{C}+\bar{D})}$

$Y = \overline{(C+D)} \cdot \overline{(\bar{C}+\bar{D})}$

POS of Z



Optimized POS

$Z = \bar{D}$

Note: double complement is used to make converting into NOR gates easier.

4. Technology Mapping