Paper 7, TDC Part-3
Chapter– 4, Combinational Logic Design
Lecture - 6

By:

Mayank Mausam

Assistant Professor (Guest Faculty)

Department of Electronics

L.S. College, BRA Bihar University,

Muzaffarpur, Bihar

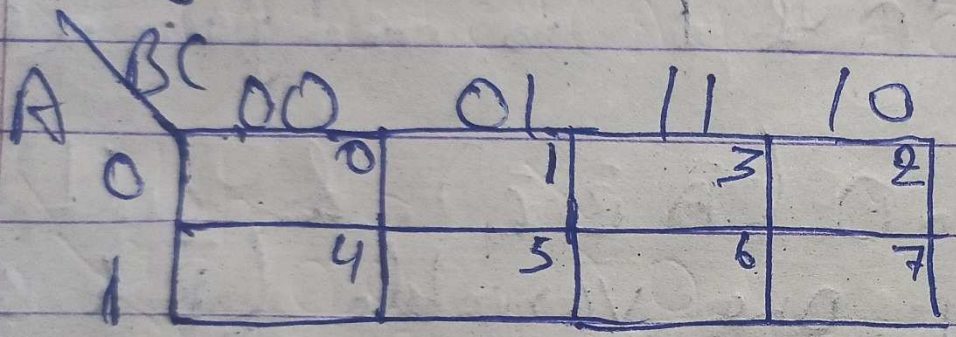
Combinational Logic Design

(5.4) Simplification of Logic Functions / Boolean expression using K-Map.

Using K-map we can simplify the logic function or Boolean expression more conveniently than ~~the~~ using the Boolean theorems. Simplication with K-map is based on the principle of combining terms in adjacent cells. Two ~~are~~ cells are said to be adjacent if they are adjacent either horizontally or vertically but not diagonally (if the two cells differ in the state of only one variable).

Combinational Logic Design

For example, in a 3-variable K-Map as shown below, let us see the adjacent cell.



In ~~the~~ 3-Variable K-Map as shown above we see that cell with decimal value '0' is adjacent to cells '1', '2' & '4'

Combinational Logic Design

Cell with Decimal Value	Variable State
0	$\bar{A}, \bar{B}, \bar{C}$
1	$\bar{A}, \bar{B}, C$
2	$\bar{A}, B, \bar{C}$
3	$\bar{A}, B, C$
4	$A, \bar{B}, \bar{C}$
5	$A, \bar{B}, C$
6	$A, B, \bar{C}$
7	A, B, C

Table 1.1 :-

Table 1.1 represents decimal value of the cells of a 3-variable K-Map and the state of the variables in the respective cells.

Combinational Logic Design

From table we see that cell '0' have variable as $\bar{A}\bar{B}\bar{C}$ and cell '1' have variables as $\bar{A}\bar{B}C$. So we can see that cell '0' and '1' has only variable which state is change so "0 & 1" is adjacent.

Hardy in cell 0 ($\bar{A}\bar{B}\bar{C}$) & cell '2' ($\bar{A}B\bar{C}$) has only one variable ($\bar{B}$ & B) which state is change so "0 & 2" is adjacent.

Again we can find that cell 0 ($\bar{A}, \bar{B}, \bar{C}$) & cell '4' ($A, \bar{B}, \bar{C}$) has only one variable ($\bar{A}$ & A)

Combinational Logic Design

which state is change so "0 & 4" is also adjacent.

While in cell '0' ($\bar{A}, \bar{B}, \bar{C}$) & cell 3 ($\bar{A}BC$) have two variables with different state ($\bar{B}$ & B and $\bar{C}$ & C). So the cell '0' and '3' is not adjacent.

Again in cell '0' ($\bar{A}, \bar{B}, \bar{C}$) & cell 5 ($A, \bar{B}, C$) have two variables with different state ($\bar{A}$ & A and $\bar{C}$ & C). So the cell '0' and '5' is not adjacent.

Combinational Logic Design

Hardy cell "0 & 6" and "0 & 7" are not adjacent.

For other cells like "1" the adjacent cells are '0', '3' & '5'. For '3' the adjacent cells are '1', '2' & '7'. In this way we can find out the adjacent cells in any variable K-map.

Table in next slide gives the adjacent cells of each cell in 2-, 3- and 4-variable K-Maps. From this it is clear that the left-most cells are adjacent to their corresponding right-

Combinational Logic Design

most cells. Similarly the top most cells are adjacent to their corresponding bottom cells.

The simplification of logical function is achieved by grouping adjacent 1's ~~and 0's~~ in case of SOP (minterms) form or 0's in case of POS (maxterm) form.

Grouping of adjacent cells is performed in groups of 2^i, where $i = 1, 2, 3, \ldots n$ where n is number of variable in the logical function.

Combinational Logic Design

The process of simplification involves grouping of minterms and identifying prime-implicants (PI) and essential prime implicants (EPI).

A 'PI' is a group of minterms that can't be combined with any other minterms or groups. An EPI is a prime-implicant in which one or more minterms are unique, i.e. it contains at least one minterm which is not contained in any other prime-implicant.

Combinational Logic Design

Cell with decimal number	Decimal numbers of adjacent cells		
	2-variable	3-variable	4-variable
0	1, 2	1, 2, 4	1, 2, 4, 8
1	0, 3	0, 3, 5	0, 3, 5, 9
2	0, 3	0, 3, 6	0, 3, 6, 10
3	1, 2	1, 2, 7	1, 2, 7, 11
4		0, 5,6	0, 5, 6, 12
5		1, 4, 7	1, 4, 7, 13
6		2, 4, 7	2, 4, 7, 14
7		3, 5, 6	3, 5, 6, 15
8			0, 9, 10,12
9			1, 8, 11,13
10			2, 8, 11,14
11			3, 9, 10,15
12			4, 8, 13,14
13			5, 9, 12,15
14			6,10, 12,15
15			7,11, 13,14

Combinational Logic Design

Grouping of cells simplified the logical function using K-map by reducing the number of literals in the resulting term. Group of 2 cells will result in one less literal than the original 2 terms.

Grouping of 4 cells will results in two less literal than the original 4 terms.

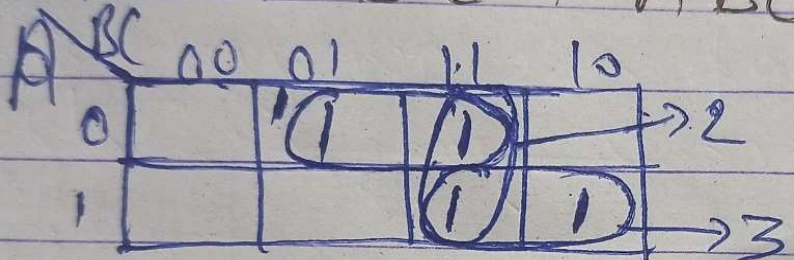
Hardy grouping 8 cells will result in 3 less literals than the original 8 terms.

Combinational Logic Design

Example 5-10) Simplify the logical function Y in canonical SOP form using K-Map.

$$Y = \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C} + ABC$$

Soln:



$$Y = \Sigma m(1,3,6,7)$$

Here 3 ~~groups~~ PI's can be formed as numbered 1, 2 & 3. PI, 1 & 3 are essential prime implicant and PI 2 is included in EPI 1 & 3. So the simplified

Combinational Logic Design

equation have only 2 groups. 1 & 3 and the function Y is given by.

$Y = \bar{A}C + AB$ ——— (i)

This can also be verified using theorem.

$Y = \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C} + ABC$

$= \bar{A}C(\bar{B} + B) + AB(\bar{C} + C)$

$= \bar{A}C + AB$ ——— (ii)

(i) & (ii) are same so using K-map we can simplify the logic function.

Combinational Logic Design

Refer book- Modern Digital Electronics by RP Jain.

Thank You