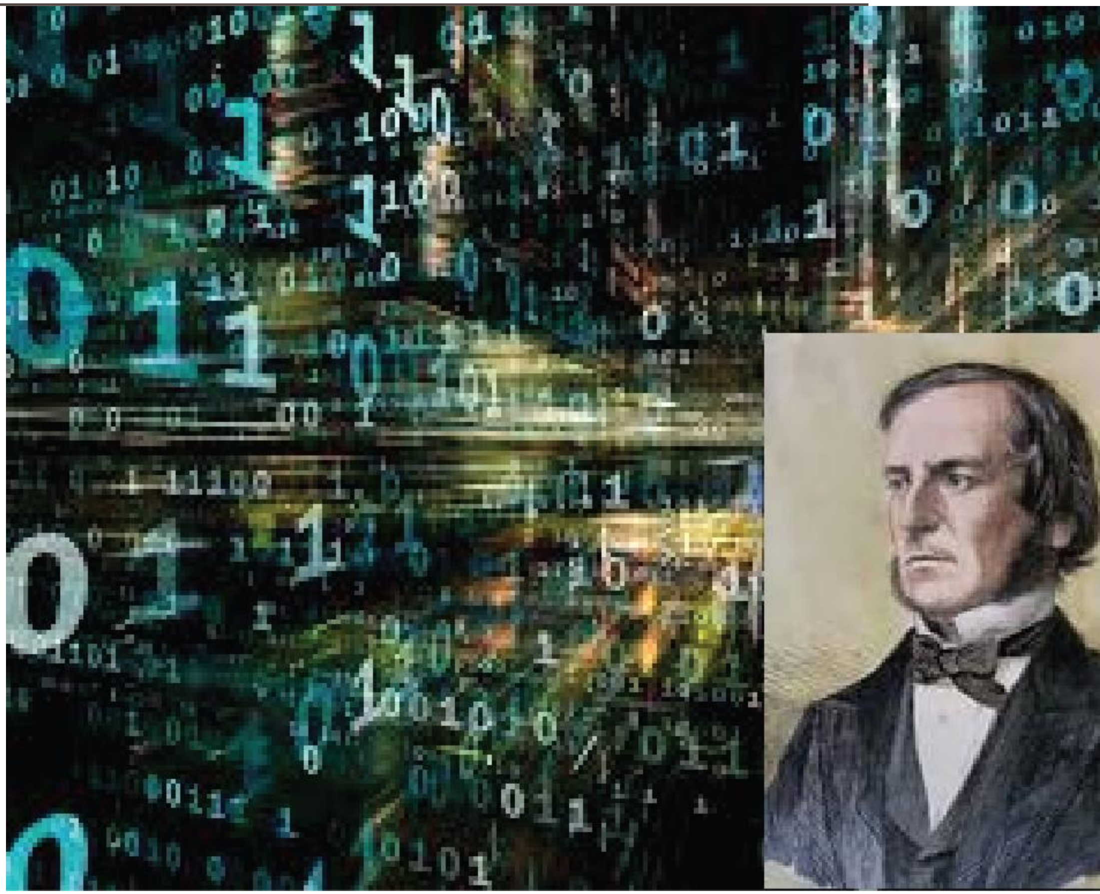
DIGITAL CIRCUITS

Week-7, Lecture-2
K-map



Sneh Saurabh
12th September, 2018

Digital Circuits: Announcements/Revision

Principles of Logic Optimization: Procedure

Steps:

1. Generate all the ***prime implicants*** of a function.

2. Find the list of ***essential prime implicants***

3. a) If the set of essential prime implicants covers all minterms, then this set is the desired minimum cover

   b) Otherwise, determine the non-essential prime implicants that should be added to form a complete minimum cover.

Principles of Logic Optimization: Example

Problem:

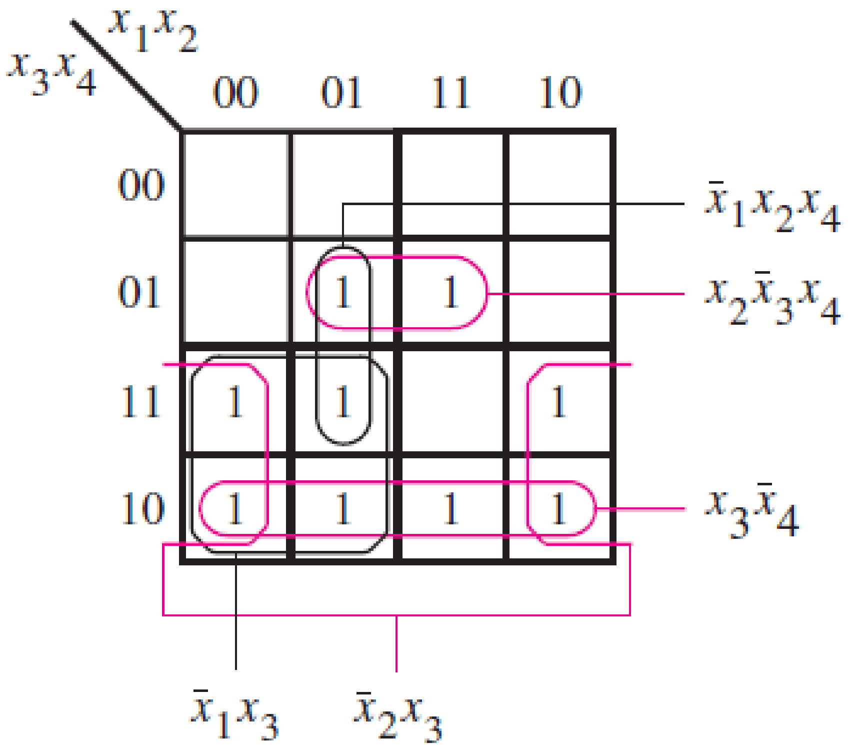
A function $f(x_1, x_2, x_3, x_4) = \Sigma m(2,3,5,6,7,10,11,13,14)$ is given.

a. Draw the K-map of this function.

b. List out all the prime implicants of the function.

c. Identify essential prime implicants.

d. Find the minimum cover.



Prime implicants are: $x_2x_3'x_4$, $x_1'x_2'x_4$, $x_1'x_3$, $x_2'x_3$, x_3x_4'

Essential Prime implicants are: $x_2x_3'x_4$, $x_2'x_3$, x_3x_4'

Minimum cover: $\{x_2x_3'x_4,\ x_2'x_3,\ x_3x_4',\ x_1'x_3\}$

$f(x_1, x_2, x_3, x_4) = x_2x_3'x_4 + x_2'x_3 + x_3x_4' + x_1'x_3$

Combinational Circuit Design

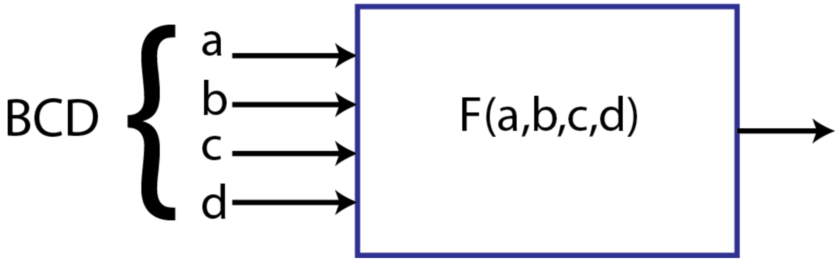
Don’t Care Conditions

Don’t Care Conditions: Meaning

- Don’t care conditions are the input combinations for which a function is not specified (0 or 1)

Example 1:

- In BCD, first 10 binary numbers are used, rest 6 combinations are not used



- If a function is considered which takes BCD as input, then it need not consider the input combinations: {1010, 1011, 1100, 1101, 1110, 1111}
- This is an example of ***input controllability don’t care***

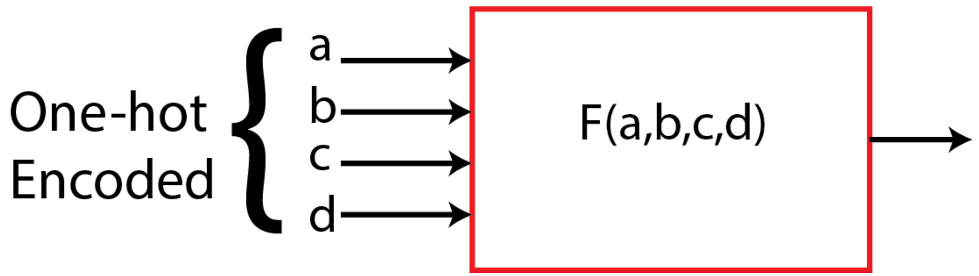
Table 5.3 Binary-coded decimal digits.

Decimal digit	BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Don’t Care Conditions: Example-2

Example 2:

- In one-hot encoding, only one bit is 1, rest all are 0’s.

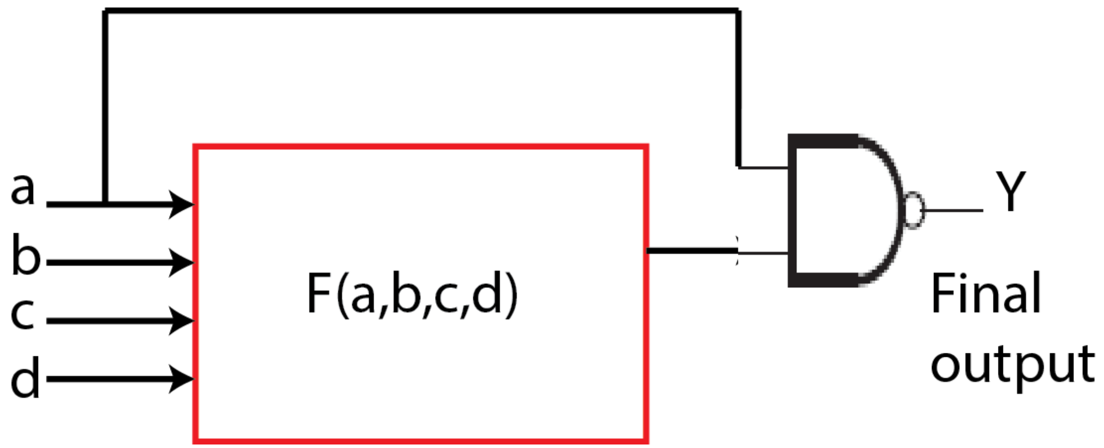


- If a function $F(a,b,c,d)$ is considered which takes one-hot encoded input, then it need to consider only following combinations: {0001, 0010, 0100, 1000}
- Rest all combinations are don’t care for $F(a,b,c,d)$
- This is an example of ***input controllability don’t care***

Don’t Care Conditions: Example-3

Example 3:

- Consider the circuit alongside
- The final output of the circuit is Y
- When the input $a = 0$, then $Y = 1$ is forced, irrespective of $F(a, b, c, d)$
- Therefore, when $a = 0$, $F(a, b, c, d)$ can take any value, without affecting Y



- All input combinations in which $a = 0$, is a don’t care condition for $F(a, b, c, d)$
- This is an example of ***output observability don’t care***

Don’t care conditions arise due to the environment (external conditions) under which a given Boolean function is used.

Don’t Care Conditions: Exploiting in simplification

- Functions that have unspecified outputs for some input combinations are called ***incompletely specified functions***.
- We simply ***don’t care*** what value is assumed by the function for the unspecified minterms
  - Therefore, unspecified minterms of a function are called don’t-care conditions

- Don’t-care conditions provide additional degree of freedom and further simplification of the Boolean expression can be carried out by exploiting the don’t care conditions

- Don’t care minterms are marked as X in K-map
- These terms can take any value 0 or 1 during minimization (combining adjacent squares)
- A value of 0 or 1 is chosen for don’t care minterms, depending on which of the two values will give a more simplified expression

Principles of Logic Optimization: Example

Problem:

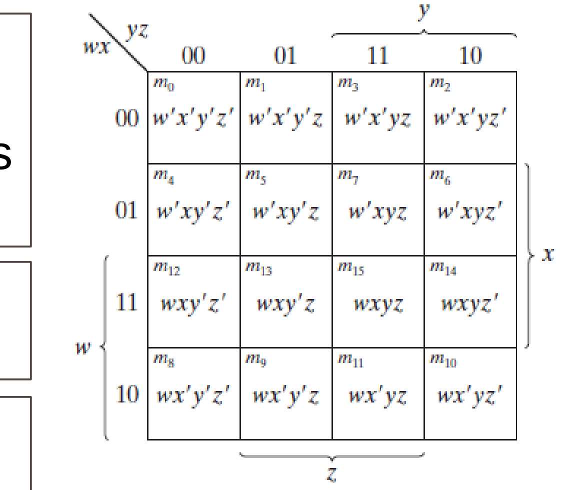
Simplify the function: $F(w,x,y,z) = \Sigma m(1,3,7,11,15)$ which has don't care conditions as: $d(w,x,y,z) = \Sigma m(0,2,5)$

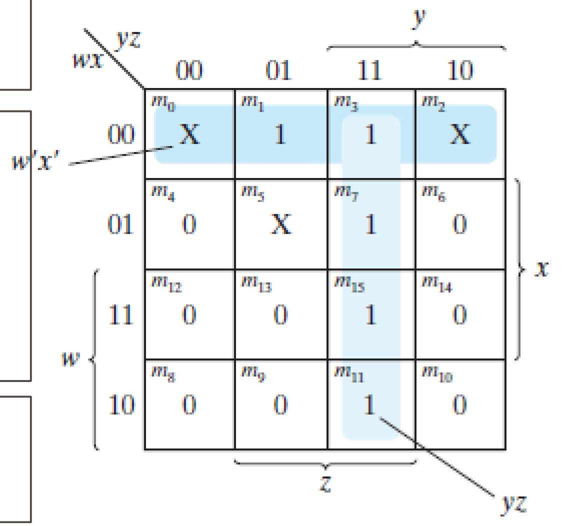
- First represent the given function on the K-map

- All five 1's must be included
- We may include X, if simplification possible

- Without use of don't care: $F(w,x,y,z) = yz + w'x'z$
- With including X in first row: $F_1(w,x,y,z) = yz + w'x'$
- With including X in m_5: $F_2(w,x,y,z) = yz + w'z$

$F(w,x,y,z) \neq F_1(w,x,y,z) \neq F_2(w,x,y,z)$





Digital Circuits: Practice Problems

Problem:

For the function specified in the previous slide, find the minimized expression for $F'(w, x, y, z)$. Thereafter, find the minimized $F'(w, x, y, z)$ in POS form.

Problems 3.15, 3.23

from "Digital Design"– M. Morris Mano & Michael D. Ciletti, Ed-5, Pearson (Prentice-Hall).

Problems 4.1-4.7, 4.9, 4.32

from Fundamentals of Digital Logic with Verilog Design - S. Brown, Z. Vranesic