Digital Electronics ECM1106

Tutorial & Practice

Karnaugh Maps

What is a Karnaugh Map?

Recap: (Lecture # 4)

A **Karnaugh map** comprises a box for every line in the truth table; the binary value for each box is the binary value of the input terms in the corresponding table row.

Unlike a truth table, in which the input values typically follow a standard binary sequence (00, 01, 10, 11), the Karnaugh map's input values must be ordered such that the values for adjacent columns vary by only a single bit, for example, 00, 01, 11, and 10. This ordering is known as a *Gray code*.

We use a **Karnaugh map** to obtain the simplest possible Boolean expression that describes a truth table.

Relationship between a Karnaugh Map and a Truth Table

Recap: (Lecture # 4)

Each row in the table (or minterm) is equivalent to a a cell on the Karnaugh Map.

Example #1:

Here is a two-input truth table for a digital circuit:

Row	**Inputs**		**Output**
	A	**B**	**F**
Row # 0	0	0	0
Row # 1	0	1	1
Row # 2	1	0	1
Row # 3	1	1	1

The corresponding K-map is:

A \ B	0	1
0	Row # 0: 0	Row # 1: 1
1	Row # 2: 1	Row # 3: 1

Example #2:

Here is a three-input truth table for a digital circuit:

Row	**Inputs**			**Output**
	A	**B**	**C**	**F**
Row # 0	0	0	0	0
Row # 1	0	0	1	1
Row # 2	0	1	0	1
Row # 3	0	1	1	1
Row # 4	1	0	0	1
Row # 5	1	0	1	1
Row # 6	1	1	0	0
Row # 7	1	1	1	1

The corresponding K-map is:

C \ AB	00	01	11	10
0	Row # 0 0	Row # 2 1	Row # 6 0	Row # 4 1
1	Row # 1 1	Row # 3 1	Row # 7 1	Row # 5 1

Example #3:

Here is a four-input truth table for a digital circuit:

Row	**Inputs**				**Output**
	A	**B**	**C**	**D**	**F**
Row # 0	0	0	0	0	0
Row # 1	0	0	0	1	1
Row # 2	0	0	1	0	1
Row # 3	0	0	1	1	1
Row # 4	0	1	0	0	1
Row # 5	0	1	0	1	1
Row # 6	0	1	1	0	0
Row # 7	0	1	1	1	1
Row # 8	1	0	0	0	1
Row # 9	1	0	0	1	0
Row # 10	1	0	1	0	1
Row # 11	1	0	1	1	1
Row # 12	1	1	0	0	1
Row # 13	1	1	0	1	1
Row # 14	1	1	1	0	1
Row # 15	1	1	1	1	0

The corresponding K-map is:

CD \ AB	00	01	11	10
00	Row # 0 0	Row # 4 1	Row # 12 1	Row # 8 1
01	Row # 1 1	Row # 5 1	Row # 13 1	Row # 9 0
11	Row # 3 1	Row # 7 1	Row # 15 0	Row # 11 1
10	Row # 2 1	Row # 6 0	Row # 14 1	Row # 10 1

Simplifying Boolean Expressions using Karnaugh map

To simplify the resulting Boolean expression using a Karnaugh map adjacent cells containing one are looped together. This step eliminated any terms of the form $A\overline{A}$.

Adjacent cells means:

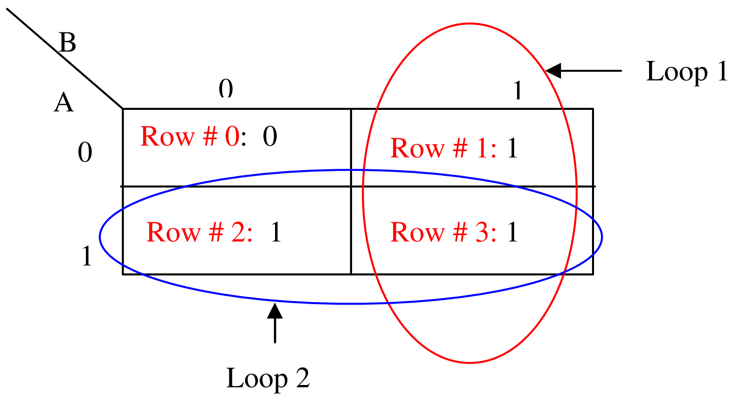
1. Cells that are side by side in the horizontal and vertical directions (but not diagonal).
2. For a map row: the leftmost cell and the rightmost cell.
3. For a map column: the topmost cell and the bottom most cell.
4. For a 4 variable map: cells occupying the four corners of the map.

Cells may only be looped together in twos, fours, or eights. As few groups as possible must be formed. Groups may overlap one another and may contain only one cell.

The larger the number of 1s looped together in a group the simpler is the product term that the group represents.

Example #1:

Simplifying the corresponding K-map of a two-input truth table for a digital circuit:



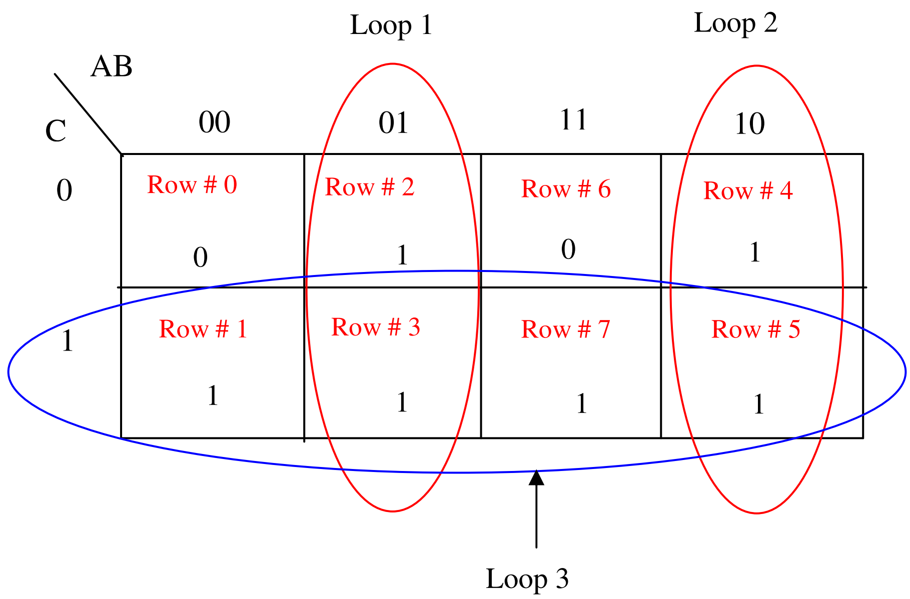
In Loop 1 the variable A has both logic 0 and logic 1 values in the same loop. B has a value of 1. Hence minterm equation is: $F = B$.

In Loop 2 Variable B has both logic 0 and 1 values in the same loop. A = 1, hence minterm equation is: $F = A$.

The overall Boolean expression for F is therefore: $F = A + B$

Example #2:

Simplifying the corresponding K-map of a three-input truth table for a digital circuit:



In Loop 1 the variable C has both logic 0 and logic 1 values in the same loop. A has a value of 0 and B has a logic value of 1. Hence minterm equation is: $F = \overline{A}B$

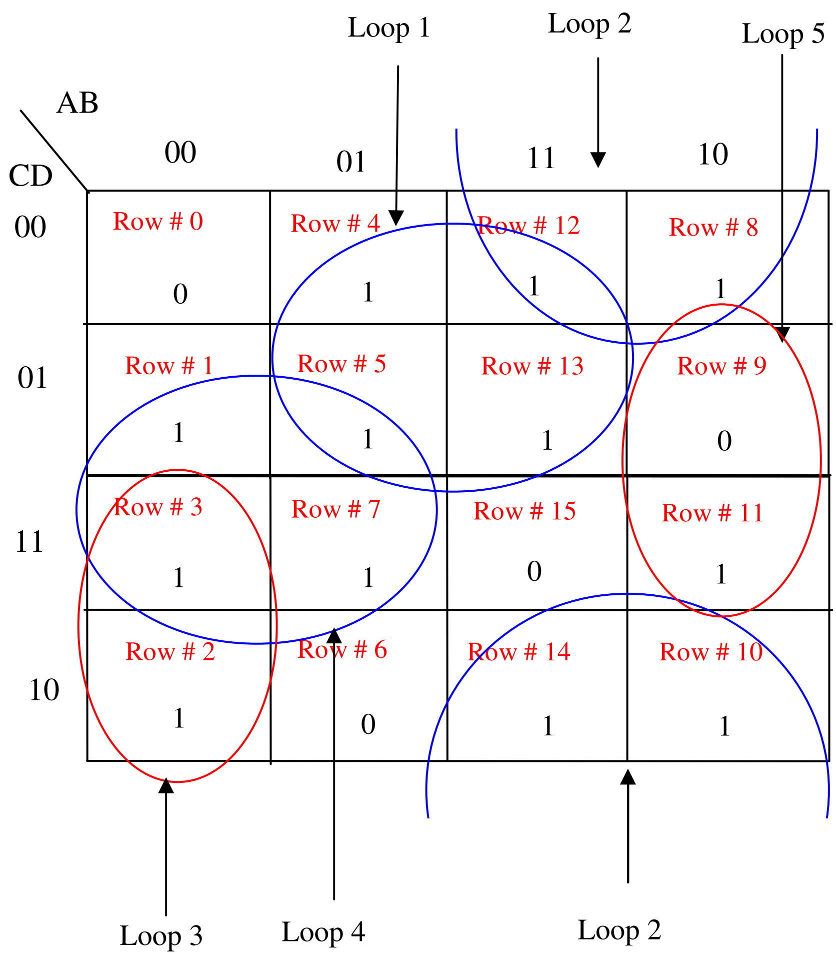
In Loop 2 the variable C has both logic 0 and 1 values in the same loop. A = 1 and B = 0, hence minterm equation is: $F = A\overline{B}$.

In Loop 3 the two variables A and B both have logic 0 and logic 1 values in the same loop. C has a value of 1. Hence minterm equation is: $F = C$.

The overall Boolean expression for F is therefore: $F = \overline{A}B + A\overline{B} + C$

Example #3:

Simplifying the corresponding K-map of a four-input truth table for a digital circuit:



In Loop 1 the two variables A and D both have logic 0 and logic 1 values in the same loop. C has a value of 0 and B has a value of 1. Hence minterm equation is: $F = B\overline{C}$.

In Loop 2 the two variables B and C both have logic 0 and logic 1 values in the same loop. A has a value of 1 and D has a value of 0. Hence minterm equation is: $F = A\overline{D}$.

In Loop 3 the variable D has logic 0 and logic 1 values in the same loop. A and B both have a value of 0 and C has a value of 1. Hence minterm equation is: $F = \overline{A}\overline{B}C$.

In Loop 4 the two variables B and C both have logic 0 and logic 1 values in the same loop. A has a value of 0 and D has a value of 1. Hence minterm equation is: $F = \overline{A}D$.

In Loop 5 the variable C has logic 0 and logic 1 values in the same loop. A and D both have s a value of 1 and B has a value of 0. Hence minterm equation is: $F = A\overline{B}D$.

The overall Boolean expression for F is therefore: $F = B\overline{C} + A\overline{D} + \overline{A}\overline{B}C + \overline{A}D + A\overline{B}D$

Practice Examples:

Q.1

A house has two lights to illuminate the stairs leading from the hall to the upstairs landing. The lights can be switched OFF and ON by either one of two switches, one in the hall and one in the landing. The lights are to be OFF when both switches are either ON or OFF together, and the lights are to be ON when one switch is ON and the other is OFF.

i) Obtain the truth table of the system
ii) Obtain the sum of products (SOP) Boolean expression for the system.
iii) Implement the system using AND, NOT and OR logic gates.
iv) Implement the system using exactly ONE logic gate.

Q.2

A bank vault has 3 locks with a key for each lock. Key A is owned by the bank manager. Key B is owned by the senior bank teller. Key C is owned by the trainee bank teller. In order to open the vault door at least two people must insert their keys into the assigned locks at the same time. The trainee bank teller) can only open the vault when the bank manager is present in the opening.

i) Determine the truth table for such a digital locking system
ii) Design, using Karnaugh Map techniques, a minimum AND-OR gate network to realise this locking system.

Q.3

A car seat belt interlock requires that the car should only start if the driver's seat belt is fastened and either the front passenger seat is unoccupied or the front passenger seat is occupied and the passenger seat belt is fastened.

i) Obtain the truth table of the system
ii) Obtain the SOP Boolean expression for the system.
iii) Use a Karnaugh map to simplify the SOP Boolean expression
iv) Implement the system using AND, NOT and OR logic gates.

Q.4

Design a circuit that will indicate whether a 4-bit number is either odd and greater than 8 or even and less than 5. Assume decimal 0 to be an even number.

Don't Care Conditions

The truth tables of some digital circuits contain certain combinations of input variables for which the output F is unimportant and so they can be either 0 or 1. Such combinations are said to be "don't care" conditions or states.

When a Boolean function is mapped, any don't care terms are represented by a $\times$ and can be looped in with either the 1 cells or the 0 cells in the simplification of that function.
A $\times$ cell should be looped with a group of 1 cells if the looping gives a greater reduction in the resulting equation.

Given the Karnaugh map for a function:

CD \ AB	00	01	11	10
00	Row # 0 1	Row # 4 1	Row # 12 X	Row # 8 1
01	Row # 1 0	Row # 5 1	Row # 13 1	Row # 9 1
11	Row # 3 0	Row # 7 X	Row # 15 1	Row # 11 1
10	Row # 2 0	Row # 6 0	Row # 14 1	Row # 10 X

Q.1

i) Obtain a simplified Boolean expression by drawing loops <u>excluding</u> the don't care conditions.

ii) Now obtain a simplified Boolean expression by drawing loops <u>including</u> the don't care conditions.

Q.2

Use a Karnaugh map to simplify the equation $F = \sum(0,2,4,5,6,7,8)$ with don't cares $\times = \sum(12,13,14,15)$.

Q.3

Use a Karnaugh map to simplify the equation $F = \sum(1,3,5,7,9)$ with don't cares $\times = \sum(6,12,13)$.