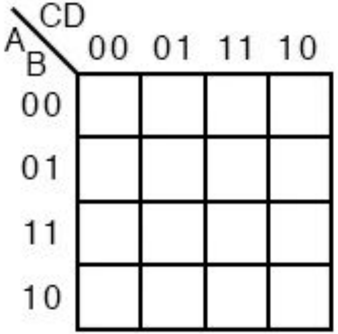
IS1102 Computer Systems

Boolean Expression Simplification
Part 03

Karnaugh Map (Recap)



- A gate level minimization technique.
- For a boolean function with n variables, corresponding K Map have 2^n cells.
- Only one variable changed when moving to an adjacent column or row.
- Steps → (1) Mapping (2) Grouping (3) Deriving

SoP and PoS Expressions

- If the simplified expression is needed in the form
  - Sum of Products → 1s (ones) are grouped. (Already Discussed)
  - Product of Sums → 0s (zeros) are grouped.

Mapping

- Draw the grid
- Put 1 to the described terms ; others 0.

Grouping

- Group 0s (zeros) for Product of Sums and Group 1s (ones) for Sum of Products.
- Grouping can be done vertically or horizontally (Not diagonal).
- Number of cells in a group should be a power of 2 (1,2,4,8 etc.)
- Select the largest group possible.
- Groups can be overlapped to form the largest group.
- Groups can be formed by wrapping around the grid.
- There should be as few groups as possible while covering all the 1s or 0s

Deriving

- Write the unchanged terms in each group

K Maps (Standard SOP)

- Draw the K-Map for the following Boolean function and derive the simplified expression in Standard Sum of Products form by applying grouping rules.

- $F = \bar{a}.\bar{b}.\bar{c}.\bar{d} + \bar{a}.\bar{b}.c.\bar{d} + \bar{a}.b.\bar{c}.\bar{d} + \bar{a}.b.c.\bar{d} + a.\bar{b}.\bar{c}.\bar{d} + a.\bar{b}.c.\bar{d}$

K Maps (Standard SOP) - Answer

- $F = \bar{a}.\bar{b}.\bar{c}.\bar{d} + \bar{a}.\bar{b}.c.\bar{d} + \bar{a}.b.\bar{c}.\bar{d} + \bar{a}.b.c.\bar{d} + a.\bar{b}.\bar{c}.\bar{d} + a.\bar{b}.c.\bar{d}$

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	1

Simplified expression:

$$\bar{a}.\bar{d} + \bar{b}.\bar{d}$$

K Maps (Standard POS)

What if we want to derive the simplified expression in Standard Product of Sums form?

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	1

K Maps (Standard POS)

Grouping

Group 0s (zeros) by following grouping rules

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	1

K Maps (Standard POS)

Deriving the simplified expression

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	1

$\bar{d}$

$\bar{a} + \bar{b}$

Simplified expression would be F’.

$$F' = (\bar{a} + \bar{b}).(\bar{d})$$

Don’t Care Condition - x

- There are functions that output is not defined for its input patterns.
- These are also called
  - Incompletely specified functions
  - Incompletely Defined Functions
- These undefined/ unspecified input patterns are called Don’t Care Conditions.
- We can exploit these don’t care conditions in KMap simplifications.

Don't Care Condition - x

- Allow us to replace the empty cell of a K-Map to form a grouping of the variables which is larger than that of forming groups without don't cares.
- Can use the "don't care" cells as either 0 or 1, and if required, we can also ignore that cell.
- The cross(×) symbol is used to represent the "don't care" cell in K-map. This cross symbol represents an invalid combination.
- No group can be encircled whose all the elements are X.

Don't Care Condition - x

Example :

Let's Assume function F is defined as,

$$F(a,b,c,d) = \sum(1,3,5,7,9)$$

F's don't care conditions are defined as,

$$d(a,b,c,d) = \sum(11,13)$$

Don't Care – Example (Mapping)

Example : $$F(a,b,c,d) = \sum(1,3,5,7,9)$$

$$d(a,b,c,d) = \sum(11,13)$$

cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	X	1
11	1	1	0	X
10	0	0	0	0

Don't Care – Suboptimal Grouping

Example :
$$F(a,b,c,d) = \sum(1,3,5,7,9)$$
$$d(a,b,c,d) = \sum(11,13)$$

cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	X	1
11	1	1	0	X
10	0	0	0	0

$\overline{a}\,d + \overline{b}\,\overline{c}\,d$

Don’t Care – Optimal Grouping #1

Example : $$F(a,b,c,d) = \sum(1,3,5,7,9)$$

$$d(a,b,c,d) = \sum(11,13)$$

cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	X	1
11	1	1	0	X
10	0	0	0	0

$$\bar{c}d + \bar{a}d$$

Don’t Care – Optimal Grouping #2

Example :
$$F(a,b,c,d) = \sum(1,3,5,7,9)$$
$$d(a,b,c,d) = \sum(11,13)$$

cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	X	1
11	1	1	0	X
10	0	0	0	0

$$\bar{b}d + \bar{a}d$$

Exercise 1

A fire alarm system consist of 3 smoke detectors ; S1 , S2 and S3. Fire alarm will be activated if at least 2 of 3 smoke detectors are activated. Assume 1 for activation of smoke detectors and Output F should be 1 to activate fire alarm. (15 Marks - 2014)

1) Draw the truth table
2) Get most simplified SoP and PoS expressions using K Maps.
3) Design logic gates using single logic gates (NAND & NOR)

Answer - Exercise 1 (Q1)

S1	S2	S3	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$SOP = \overline{S1}.S2.S3 + S1.\overline{S2}.S3 + S1.S2.\overline{S3} + S1.S2.S3$$

Answer - Exercise 1 (Q2)

$$SOP = \overline{S1}.S2.S3 + S1.\overline{S2}.S3 + S1.S2.\overline{S3} + S1.S2.S3$$

S3 \ S1 S2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

F = S1.S2 + S2.S3 + S1.S3 ← simplified expression in SOP form

Answer - Exercise 1 (Q2)

$$SOP = \overline{S1}+S2+S3 \;.\; S1+\overline{S2}+S3 \;.\; S1+S2+\overline{S3} \;.\; S1+S2+S3$$

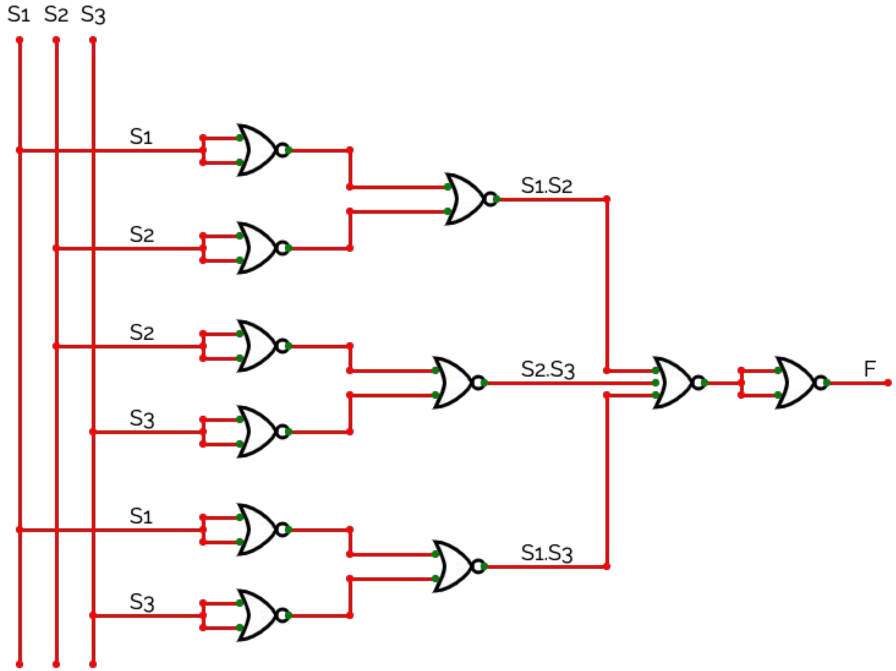
S3 \ S1 S2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

F’ = (S1+S3).(S1+S2).(S2+S3) ← simplified expression in POS form

Answer - Exercise 1 (Q3)

Using NOR Gate

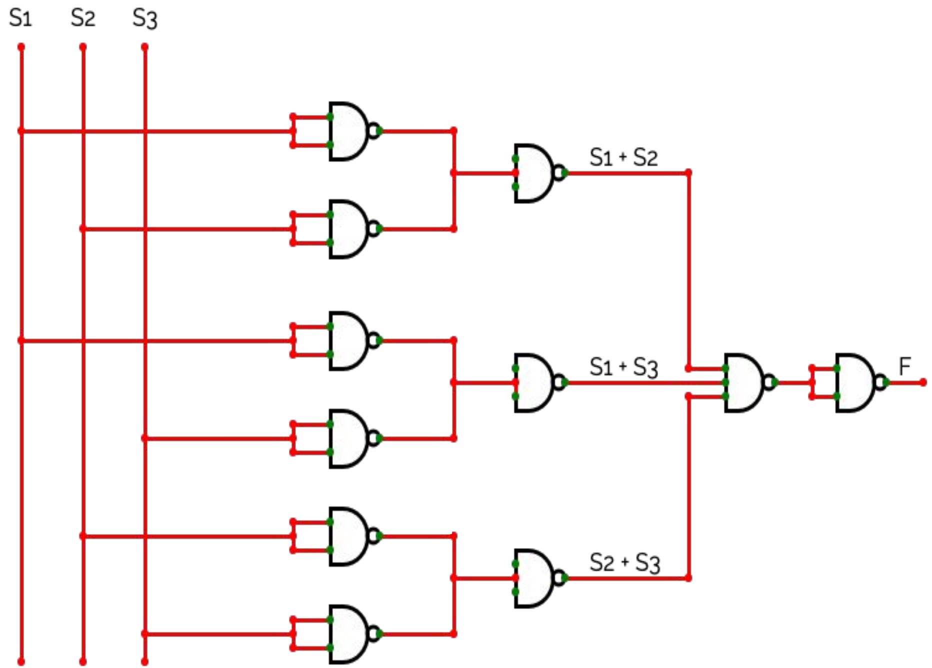
$$F = S1.S2 + S2.S3 + S1.S3$$



Answer - Exercise 1 (Q3)

Using NAND Gate

$F' = (S1+S3).(S1+S2).(S2+S3)$



Thank You!!