Digital Design

Ms. Umarah Qaseem

1

4

Chapter 3

Gate Level Minimization

2

Karnaugh Map Method

4

Gate Level Minimization

- Although truth tables representation of a function is unique, it can be expressed algebraically in different forms
- The procedure of simplifying Boolean expressions is difficult since it lacks specific rules to predict the successive steps in the simplification process.
- Alternative Karnaugh Map (K-map) Method

5

Gate Level Minimization

- The Karnaugh map, like Boolean algebra, is a simplification tool applicable to digital logic.
- Boolean simplification is actually faster than the Karnaugh map for a task involving two or fewer Boolean variables. It is still quite usable at three variables, but a bit slower.

6

Gate Level Minimization

- At four input variables, Boolean algebra becomes tedious. Karnaugh maps are both faster and easier.
- Karnaugh maps work well for up to six input variables, are usable for up to eight variables.
- For more than six to eight variables, simplification should be by **CAD (Computer-Aided Design)**.

7

Gate Level Minimization

- Recall: a given function has
  - A unique representation using a truth table
  - A unique representation as a sum of minterms
  - Several equivalent representations as algebraic expressions
- Boolean minimization is a bit awkward
- Solution: Karnaugh map method minimization

8

Gate Level Minimization

- Basic Procedure:
- Minimization is performed by visual identification of *logic* blocks
  - The larger the blocks, the fewer literals in a term

9

K-Map

- A K-Map is a graphical representation of a logic function's truth table

m_0	m_1
m_2	m_3

$x \backslash y$	0	1 (y)
0	$x'y'$	$x'y$
1 (x)	xy'	xy

10

Re-arranging the Truth Table

- A two-variable function has four possible minterms. We can re-arrange these minterms into a Karnaugh map

x	y	minterm
0	0	x'y'
0	1	x'y
1	0	xy'
1	1	xy

x \ y	0	1
0	x'y'	x'y
1	xy'	xy

11

K-Map Simplification

- Imagine a two-variable sum of minterms:

$$x'y' + x'y$$

- Both of these minterms appear in the top row of a Karnaugh map, which means that they both contain the literal x'

		y
	$x'y'$	$x'y$
X	xy'	xy

1	1
0	0

- What happens if you simplify this expression using Boolean algebra?

$x'y' + x'y$	$= x'(y' + y)$	[Distributive]
	$= x' \bullet 1$	[$y + y' = 1$]
	$= x'$	[$x \bullet 1 = x$]

13

K-Map: Two-Variable Examples

- Another example expression is $x'y + xy$
  - Both minterms appear in the right side, where y is uncomplemented
  - Thus, we can reduce $x'y + xy$ to just y

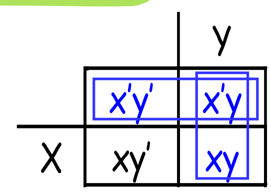
		y
	$x'y'$	$x'y$
X	xy'	xy

0	1
0	1

14

K-Map: Two-Variable Examples: Activity

- Solve for x’y’ + x’y + xy
- We have x’y’ + x’y in the top row, corresponding to x’
- There’s also x’y + xy in the right side, corresponding to y
- This whole expression can be reduced to x’ + y



1	**1**
0	**1**

15

K-Map: Two-Variable Examples

- The truth table for the function *F(x,y) = xy* is shown at the right along with its corresponding Kmap.

F(X,Y) = XY

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X \ Y	0	1
0	0	0
1	0	1

16

K-Map: Two-Variable Examples

- As another example, we give the truth table and KMap for the function, *F(x,y) = x + y*.
- This function is equivalent to the OR of all of the minterms that have a value of 1. Thus:

F(X,Y) = X+Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

$$F(x,y) = X+Y = \bar{X}Y + X\bar{Y} + XY$$

X \ Y	0	1
0	0	1
1	1	1

17

K-Map Simplification

- We can reduce complicated expression to its simplest terms by finding adjacent 1s in the Kmap that can be collected into groups that are powers of two.
- In our example, we have two such groups: so solved form is X + Y

X \ Y	0	1
0	0	1
1	1	1

X \ Y	0	1
0	0	1
1	1	1

18

K-Map Simplification: Rules

- Groupings can contain only 1s; no 0s.
- Groups can be formed only at right angles; diagonal groups are not allowed.
- The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.

19

Three Variable Kmap

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

yz yz

x x

$x \backslash yz$	00	01	11 (y)	10 (y)
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x 1	$xy'z'$	$xy'z$	xyz	xyz'

z

20

Three Variable K-Map

- For a three-variable expression with inputs x, y, z, the arrangement of minterms is more tricky:

x \ yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

- Another way to label the K-map (use whichever you like):

			y	y
	x'y'z'	x'y'z	x'yz	x'yz'
x	xy'z'	xy'z	xyz	xyz'
		z	z	

			y	y
	m_0	m_1	m_3	m_2
x	m_4	m_5	m_7	m_6
		z	z	

Adjacency

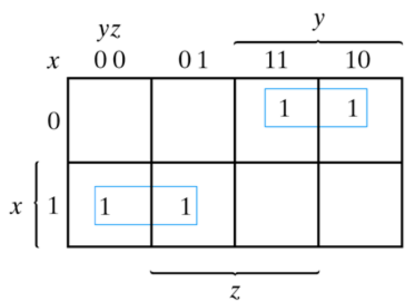
- "Adjacency" includes wrapping around the left and right sides:

- We'll use this property of adjacent squares to do our simplifications.

			y	y
	x'y'z'	x'y'z	x'yz	x'yz'
x	xy'z'	xy'z	xyz	xyz'
		z	z	

Examples: Three Variable Kmap

$$F = \Sigma(m_2, m_3, m_4, m_5) = x'yz' + x'yz + xy'z' + xy'z$$



$$x'y + xy'$$

24

Examples: Three Variable Kmap

Given: $F(A, B, C) = A'C + A'B + AB'C + BC$

(a) Express F in sum of minterms.
(b) Find the minimal sum of products using K-Map

$$A'C(B + B') = A'BC + A'B'C$$
$$A'B(C + C') = A'BC + A'BC'$$
$$AB'C$$
$$BC(A + A') = ABC + A'BC$$

$$\boxed{\begin{aligned} &F(A, B, C) = A'B'C + A'BC' + A'BC + AB'C + ABC \\ &= \Sigma(1,2,3,5,7) \end{aligned}}$$

25

Examples: Three Variable Kmap

$$F(A,B,C) = \Sigma(1,2,3,5,7)$$

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

A \ BC	00	01	11	10
0		1	1	1
1		1	1	

$C + A'B$

26

Examples: Three Variable Kmap

$$F = \Sigma(m_1, m_5, m_6, m_7)$$

	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x	$xy'z'$	$xy'z$	xyz	xyz'

	m_0	m_1	m_3	m_2
x	m_4	m_5	m_7	m_6

	0	1	0	0
x	0	1	1	1

$y'z$ $\quad$ xy

F(x,y,z)= y'z + xy

27

Examples: Three Variable Kmap

$$F = \Sigma(m_1, m_3, m_5, m_6)$$

y

	m_0	m_1	m_3	m_2
X	m_4	m_5	m_7	m_6

Z

y

	0	1	1	0
X	0	1	0	1

Z

x'z + y'z + xyz'

28

Kmaps can be tricky

y

	0	1	0	1
X	0	1	1	1

Z

y

	0	1	0	1
X	0	1	1	1

Z

y'z + yz' + xy

y

	0	1	0	1
X	0	1	1	1

Z

y'z + yz' + xz

29

Three Variable Kmaps

- One square represents one minterm → a term of 3 literals
- Two adjacent squares → a term of 2 literals
- Four adjacent squares → a term of 1 literal
- Eight adjacent squares → the function equals to 1

30

References

- Chapter 3 – Digital Design Morris Mano
- Template is taken from slides carnival.

SlidesCarnival

31