

Lecture 5: K-Map minimization in larger input dimensions and K-map minimization using max terms

CSE 140: Components and Design Techniques for Digital Systems

CK Cheng
Dept. of Computer Science and Engineering
University of California, San Diego

Definitions: Sum of Products and Product of Sums Expressions

Sum of Products Expression

Implicant: A **product term** that has non-empty intersection with on-set $\mathcal{F}$ and does not intersect with off-set $\mathcal{R}$.

Prime Implicant: An **implicant** that is not a proper subset of any other **implicant**.

Essential Prime Implicant: A prime **implicant** that has an element in on-set $\mathcal{F}$ but this element is not covered by any other prime **implicants**.

Product of Sums Expression

Implicate: A **sum term** that has non-empty intersection with off-set $\mathcal{R}$ and does not intersect with on-set $\mathcal{F}$.

Prime Implicate: An **implicate** that is not a proper subset of any other **implicate**.

Essential Prime Implicate: A prime **implicate** that has an element in off-set $\mathcal{R}$ but this element is not covered by any other prime **implicates**.

4WA

Part I. Combinational Logic

1. Specification
2. Implementation

K-map:

Sum of products $\sum P_i$
Product of sums $\prod S_i$

Ex: SOP $f(a,b,c) = ab + a'b' + a'c$
 $f(a,b,c) = 1$ if $ab=1$, $a'b'=1$, or $a'c=1$
 $f(a,b,c) = 0$? NPComplete (SAT: Satisfiability)
POS $g(a,b,c) = (a+b)(a'+b')(a'+c)$
 $g(a,b,c) = 0$ if $(a+b)$, $(a'+b')$ or $(a'+c) = 0$
 $g(a,b,c) = 1$? (SAT)

K-Map to Minimized Product of Sums

• Sometimes it is easier to reduce the K-map by considering the offset

$$F_1 = a'b'c'd + abc'd + a'b'cd' + abcd$$

$$F_2 = (a+b')(a'+b)(c'+d')(c+d)$$

	ab	00	01	11	10
cd	00	0	4	12	8
01	1	5	0	13	9
11	3	0	7	0	15
10	2	1	6	0	14

Q: Which function is simpler for a two-level logic implementation?

- A. F_1
- B. F_2
- C. Two are the same

$\Pi M(0,4,8,12) \quad c+d$
 $\Pi M(3,7,11,15) \quad c+d'$
 $\Pi M(4,5,6,7) \quad a+b'$
 $\Pi M(8,9,10,11) \quad a'+b$

$F_1: 4 \text{ prod terms, } 16 \text{ literals}$
 $F_2: 4 \text{ sum terms, } 8 \text{ literals}$

K-Map to Minimized Product of Sums

• Sometimes it is easier to reduce the K-map by considering the offset

$$F_1 = a'b'c'd + abc'd + a'b'cd' + abcd'$$

$$F_2 = (a + b')(a' + b)(c' + d')(c + d)$$

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

5

Another min product of sums example

Given $\mathcal{R}(a,b,c,d) = \Sigma m(3, 11, 12, 13, 14)$

$\mathcal{D}(a,b,c,d) = \Sigma m(4, 8, 10)$

K-map

ab \ cd	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

6

Another min product of sums example

Given $\mathcal{R}(a,b,c,d) = \Sigma m(3, 11, 12, 13, 14)$ Is $\pi M(12,14)$ a prime implicate?
 $\mathcal{D}(a,b,c,d) = \Sigma m(4, 8, 10)$ A: Yes
 B: No

ab \ cd	00	01	11	10
00	0	4	12	8
01	1	X	0	X
11	3	7	15	11
10	2	6	14	10

a

d

7

Prime Implicates:

$\pi M(0,8), \pi M(11,15), \pi M(12,13,14,15), \pi M(6,14)$

Q: Which of the following is a not an essential prime implicate?

- A. $\pi M(0,8)$
- B. $\pi M(11,15)$
- C. $\pi M(12,13,14,15)$
- D. $\pi M(6,14)$

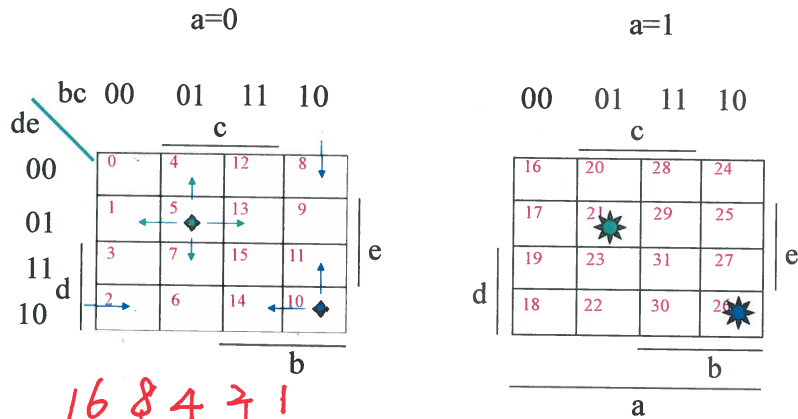
ab \ cd	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

a

d

8

Five variable K-map

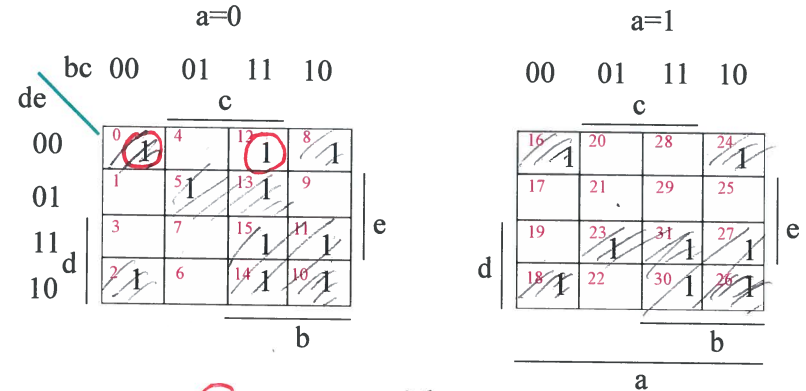


16 8 4 2 1
a b c d e

Neighbors of m_5 are: minterms 1, 4, 7, 13, and 21
Neighbors of m_{10} are: minterms 2, 8, 11, 14, and 26

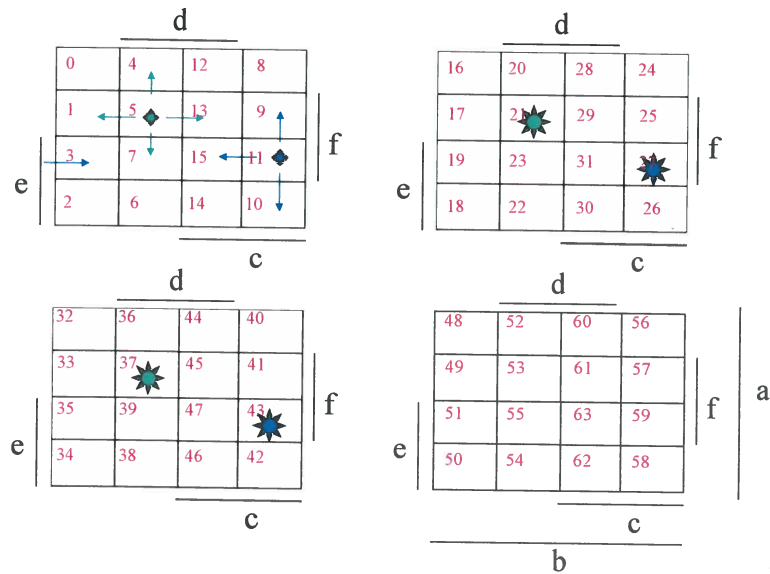
m_5 00101 5
00100 4
00111 7
00001 1
01101 13
10101 21

Reading a Five variable K-map: An example



I $\Sigma m(0, 2, 16, 18, 8, 10, 24, 26)$ E
II $\Sigma m(12, 13, 14, 15)$ NE
III $\Sigma m(10, 11, 14, 15, 26, 27, 30, 31)$ E
IV $\Sigma m(5, 13)$ E
V $\Sigma m(23, 31)$ E
VI $\Sigma m(12, 13, 14, 15)$ NE

Six variable K-map



7 Variables and Beyond

- Procedure
 - Specification
 - Prime enumeration
 - Essential prime selection
 - Logic synthesis
- Theory
- Methodology
 - Data structure
 - Algorithms
 - Software packages
 - Benchmarks



CSE 140: Components and Design Techniques for Digital Systems

Lecture 6: Universal Gates

CK Cheng
Dept. of Computer Science and Engineering
University of California, San Diego

1

Universal Set of Gates: Motivation

- AND, OR, NOT: Logic gates related to reasoning from Aristotle (384-322BCE).
- NAND, NOR: Inverted AND, Inverted OR gates. For VLSI technologies, all gates are inverted (AND, OR operation with a bubble at output).
- XOR: Exclusive OR gates. Parity check.
- Multiplexer + input table: Table based logic for programmability. FPGA technology. *Intel, Xilinx*
- Neuron and Synapse: Neural network.
- Reversible Gates: Quantum computing.

In the future, we may have new sets of gates due to new technologies.
Given a set of gates, are we able to construct all possible switching functions using the gates in the set?

3

Combinational Logic: Various Types of Gates

- Universal Set of Gates
 - Motivation
 - Definition
 - Examples
- Other Types of Gates
 - XOR
 - NAND / NOR
 - Block Diagram Transfers

2

Universal Set

Universal set is a powerful concept to identify the coverage of a set of gates afforded by a given technology.

Criterion: If the set of gates can implement AND, OR, and NOT gates, then the set is universal.

4

Universal Set Definition

Universal Set: A set of gates such that every switching function can be implemented with the gates in this set.

Examples

{AND, OR, NOT} Univ.

{AND, NOT} Univ. column'

$\{\text{OR, NOT}\}$ *univ. $a+b = (a+b)$*

$$a \cdot b = (a' + b')'$$

Example

Is the set $\{\text{AND}, \text{OR}\}$ (but no NOT gate) universal?

A. Yes

B. No

Note that the set was used in a once popular design style as **domino logic** for high performance computing.

Universal Set: Examples

Universal Set: A set of gates such that every switching function can be implemented with the gates in this set.

Examples

{AND, OR, NOT}

{AND, NOT} OR can be implemented with AND &

NOT gates: $a + b = (a'b')'$

{OR, NOT} AND can be implemented with OR &

NOT gates: $ab = (a' + b')'$

$\{\text{XOR}\}$ is not universal

$\{\text{XOR}, \text{AND}\}$ is universal

$a \Rightarrow \neg a \vee b$
 $b \Rightarrow \neg a \vee b$
 $a \Rightarrow \neg a \vee a' = a'$
 $1 \Rightarrow a \vee a' = 1$



AND
 OR

Example

Is the set $\{f(x, y) = xy'\}$ universal?

A. Yes.

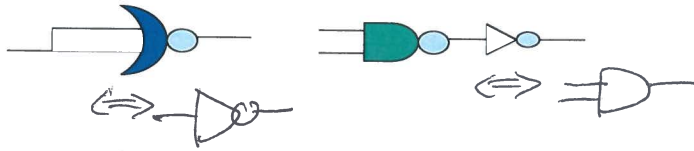
B. No

$$y \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow y' \Leftrightarrow \Delta$$

x — y — $\neg y$ — f — $x(y')' = xy$
 $\Leftrightarrow x \cdot y = xy$

Universal Set: Examples

{NAND, NOR} Univ.



{XOR} Not Univ.

{XOR, AND} 1 Univ.

$x \oplus 1 = x1' + x'1 = x'$ if constant "1" is available.

9

Other Types of Gates: Properties and Usage

1. XOR $x \oplus y = xy' + x'y$
2. NAND, NOR
3. Block Diagram Transfers

{NAND} Univ.



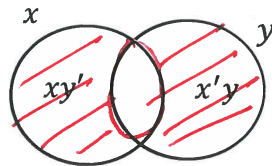
{NOR} Univ.

10

Other Types of Gates: XOR

$$x \oplus y = xy' + x'y$$

It is a parity function (examples)
useful for testing because the
flipping of a single input changes the
output.



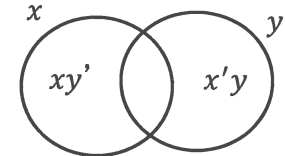
id	x	y	$x \oplus y$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

	x=0	x=1
y=0	0	1
y=1	1	0

11

Other Types of Gates: XOR

- 1) XOR $x \oplus y = xy' + x'y$



(a) Commutative $x \oplus y = y \oplus x$

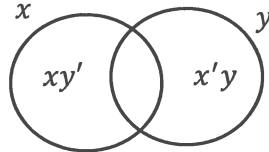
(b) Associative $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ = 1 iff # true at inputs is an odd #.

$$\begin{aligned}
 \text{LHS} &= (xy' + x'y) \oplus z \\
 &= (xy' + x'y)z' + (xy' + x'y)z \\
 &= (x'y + x'y')z' + (xy + x'y)z \\
 &= x'yz' + x'y'z' + xyz + x'y'z \\
 &= x'y + xy
 \end{aligned}$$

12

Other Types of Gates: XOR

1) XOR $x \oplus y = xy' + x'y$



a) Commutative $x \oplus y = y \oplus x$

b) Associative $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

c) $1 \oplus x = x', 0 \oplus x = x$

d) $x \oplus x = 0, x \oplus x' = 1$

$$1 \oplus x = 1x' + 1'x = x'$$

$$0 \oplus x = 0x' + 0'x = x$$

$$x \oplus x = xx' + x'x = 0$$

$$x \oplus x' = xx'' + x'x' = x + x' = 1$$

13

Other Types of Gates: XOR

e) If $ab = 0$, then $a \oplus b = a + b$

Proof: If $ab = 0$, then

$$a = a(b + b') = ab + ab' = ab'$$

$$b = b(a + a') = ba + ba' = a'b$$

$$\text{Therefore, } a + b = ab' + a'b = a \oplus b$$

Note that in full adder, we have

$$c_{out} = ab + bc + ac = ab + c(a + b)$$

From property e), we can also write $c_{out} = ab + c(a \oplus b)$

$$\text{Sum} = \underbrace{a \oplus b}_{\text{wavy line}} \oplus c$$

14

Other Types of Gates: XOR

f) $f(x, y) = x \oplus xy' \oplus x'y \oplus (x + y) \oplus x = ?$

(Priority of operations: AND, $\oplus$, OR)

Hint: We apply Shannon's Expansion.

15

Other Types of Gates: XOR

Shannon's Expansion: Recap for switching functions

Formula: $f(x, y) = xf(1, y) + x'f(0, y)$

Proof by enumeration:

If $x = 1, f(x, y) = f(1, y):$

$$xf(1, y) + x'f(0, y) = 1f(1, y) + 1'f(0, y) = f(1, y)$$

If $x = 0, f(x, y) = f(0, y):$

$$xf(1, y) + x'f(0, y) = 0f(1, y) + 0'f(0, y) = f(0, y)$$

16

Other types of gates: XOR

Simplify the function (Priority of operations: AND, $\oplus$, OR)

$$f(X, Y) = X \oplus XY' \oplus X'Y \oplus (X + Y) \oplus X$$

$$\text{Case } X = 1: f(1, Y) = 1 \oplus Y' \oplus 0 \oplus 1 \oplus 1 = Y$$

$$\text{Case } X = 0: f(0, Y) = 0 \oplus 0 \oplus Y \oplus Y \oplus 0 = 0$$

Thus, using Shannon's expansion, we have

$$f(X, Y) = Xf(1, Y) + X'f(0, Y) = XY$$

$$xy + x'0 \rightarrow xy$$

17

XOR gates

Example: Is the equation

$$a + (b \oplus c) \neq (a + b) \oplus (a + c)$$

true?

A. Yes

B. No

If $a=1$.

$$\text{LHS} = 1$$

$$\begin{aligned} \text{RHS} &= (1+b) \oplus (1+c) \\ &= 1 \oplus 1 = 0 \end{aligned}$$

18

Other Types of Gates: NAND, NOR

2) NAND, NOR gates

NAND (NOR) gates are not associative

$$\text{Let } a | b = (ab)'$$

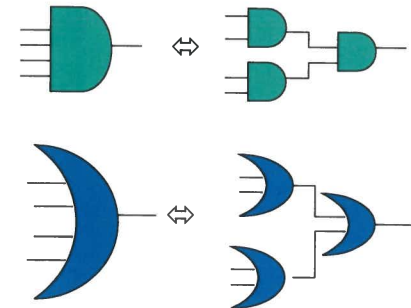
$$(a | b) | c \neq a | (b | c)$$

19

Other Types of Gates: NAND, NOR Block Diagram Transform

3) Block Diagram Transformation

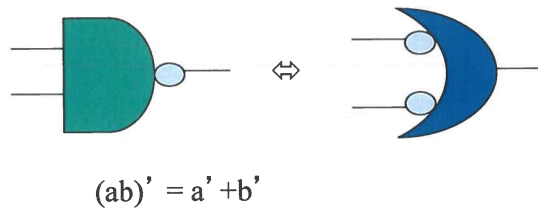
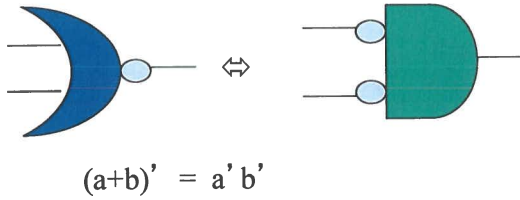
a) Reduce # of inputs.



20

Other Types of Gates: NAND, NOR Block Diagram Transform

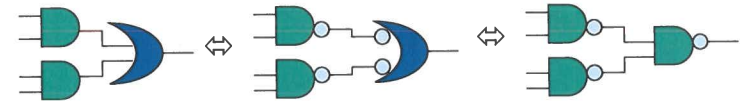
b. DeMorgan's Law



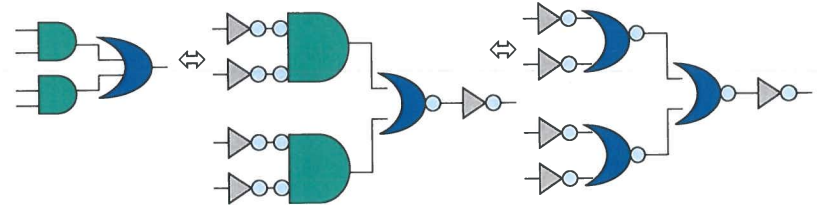
21

Other Types of Gates: NAND, NOR Block Diagram Transform

c. Sum of Products (Using only NAND gates)



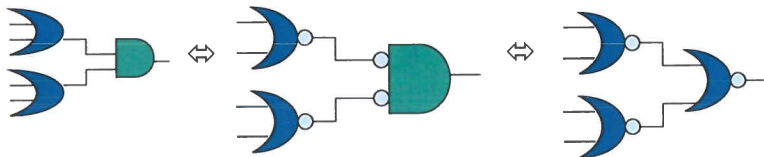
Sum of Products (We create many bubbles with NOR gates)



22

Other Types of Gates: NAND, NOR Block Diagram Transform

d. Product of Sums (NOR gates only)



We will create many bubbles with NAND gates.

23

Other Types of Gates: NAND, NOR Block Diagram Transform

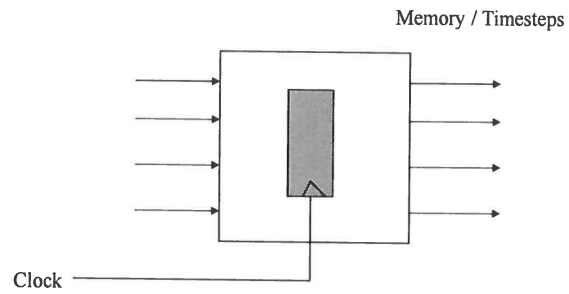
Remark:

Two level NAND gates: Sum of Products

Two level NOR gates: Product of Sums

24

Part II. Sequential Networks



Flip flops
Specification
Implementation

