ENGG2020 DIGITAL LOGIC AND SYSTEMS

CHAPTER 3: GATE LEVEL MINIMIZATION

By Dr. Anthony Sum
Department of Computer Science and Engineering
The Chinese University of Hong Kong

1

CONTENTS

- Canonical and Standard Forms
- Gate level minimization
- K-map

2

MINTERM

- Normal form and complement form
  - A binary variable may appear either in its normal form, x, or in its complement form, x'
- Minterm
  - If there are 2 variables, x and y, there are 4 possible combinations: x'y', x'y, xy', xy
  - Each of these 4 AND terms is called a minterm, or standard product
  - If there are N variables, there are 2^N minterms
  - A symbol for each minterm is denoted as m_j, where the subscript j denotes the decimal equivalent of the binary number of the minterm designated

3

MAXTERM

- Maxterm
  - If there are 2 variables, x and y, there are another 4 possible combinations: x+y, x+y', x'+y, x'+y'
  - Each of these 4 OR terms is called a maxterm, or standard sum
  - If there are N variables, there are 2^N maxterms
  - A symbol for each maxterm is denoted as M_j, where the subscript j denotes the decimal equivalent of the binary number of the maxterm designated
- For minterm, "1" ➔ Normal (True), and "0" ➔ Complement (False)
- For maxterm, "0" ➔ Normal (True), and "1" ➔ Complement (False)
- Therefore, m_j is the complement of M_j

4

MINTERMS AND MAXTERMS

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$ =111=1	m_0	$x + y + z$ =0+0+0=0	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

5

EXAMPLE (MINTERMS & MAXTERMS)

- For example, we have the truth table of the following functions f_1

Functions of Three Variables

x	**y**	**z**	**Function f_1**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

6

EXAMPLE (MINTERMS & MAXTERMS)

- A Boolean function can be expressed algebraically in terms of either minterms or maxterms

- In term of minterms, we have
- $f_1(x,y,z) = x'y'z + xy'z' + xyz$
- $f_1(x,y,z) = m_1 + m_4 + m_7$
- $f_1(x,y,z) = \sum m(1,4,7)$
- $f_1'(x,y,z) = \sum m(0,2,3,5,6)$

Functions of Three Variables

x	*y*	*z*		Function f_1
0	0	0	term 0	0
0	0	1	term 1	1
0	1	0	term 2	0
0	1	1	term 3	0
1	0	0	term 4	1
1	0	1	term 5	0
1	1	0	term 6	0
1	1	1	term 7	1

7

EXAMPLE (MINTERMS & MAXTERMS)

- In term of maxterms, we have
- $f_1(x,y,z) = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$
- $f_1(x,y,z) = M_0M_2M_3M_5M_6$
- $f_1(x,y,z) = \prod M(0,2,3,5,6)$
- $f_1'(x,y,z) = \prod M(1,4,7)$

Functions of Three Variables

x	*y*	*z*	Function f_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

8

EXAMPLE OF MINTERM & MAXTERM

- Given a truth table:

A	B	X
0	0	0
0	1	1
1	0	0
1	1	1

- By using Minterm, 1 = Ture, 0 = False
  - X = AB'+AB sum of product (SOP)
- By using Maxterm, 0 = True, 1 = False
  - X = (A+B)(A'+B) product of sum(POS)
    - If A=0, B=0, X = (0+0)(1+0) = 0
    - If A=0, B=1, X = (0+1)(1+1) = 1
    - If A=1, B=0, X = (1+0)(0+0) = 0
    - If A=1, B=1, X = (1+1)(0+1) = 1

9

CANONICAL AND STANDARD FORMS

- Standard Sum-Of-Products (SOP) forms
  - Equations are written as AND terms summed with OR operators
  - $F = xyz + xy' + z'$
- Standard Product-Of-Sums (POS) forms
  - Equations are written as OR terms multiplied with AND operators
  - $F = (x + y + z)(x + y')(z')$
- Canonical forms, e.g.
  - Sum of minterms: $F = xyz + xy'z + x'yz'$
  - Product of maxterms: $F = (x + y + z)(x + y' + z)(x' + y + z')$
- Mixed forms is neither SOP nor POS, e.g.
  - $F = (xy + z)(x + y'z) + (x'y + z')$

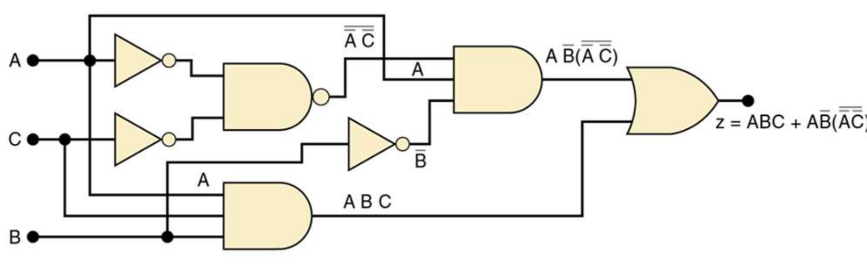
10

11

GATE LEVEL MINIMIZATION

SIMPLIFYING A LOGIC CIRCUIT

- Simplify the logic circuit below:

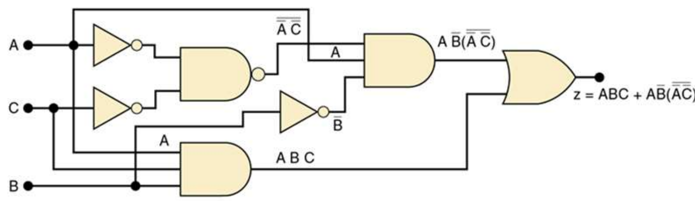


$$
\begin{aligned}
z &= ABC + A\overline{B}(\overline{\overline{A}} + \overline{\overline{C}}) && \text{[theorem (17)]} \\
&= ABC + A\overline{B}(A + C) && \text{[cancel double inversions]} \\
&= ABC + A\overline{B}A + A\overline{B}C && \text{[multiply out]} \\
&= ABC + A\overline{B} + A\overline{B}C && [A \cdot A = A]
\end{aligned}
$$

12

SIMPLIFYING A LOGIC CIRCUIT

- Simplify the logic circuit below:

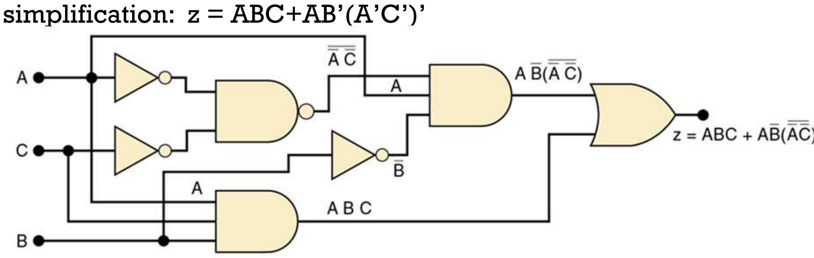


$$
\begin{aligned}
z &= ABC + A\overline{B}(\overline{\overline{A}} + \overline{\overline{C}}) && \text{[theorem (17)]}\\
&= ABC + A\overline{B}(A + C) && \text{[cancel double inversions]}\\
&= ABC + A\overline{B}A + A\overline{B}C && \text{[multiply out]}\\
&= ABC + A\overline{B} + A\overline{B}C && [A \cdot A = A]\\
z &= AC(B + \overline{B}) + A\overline{B} &&\\
z &= AC(1) + A\overline{B} &&\\
&= AC + A\overline{B} &&
\end{aligned}
$$

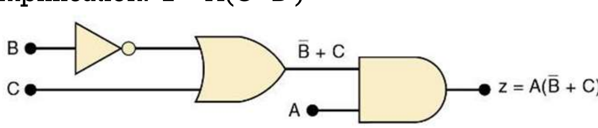
13

SIMPLIFYING A LOGIC CIRCUIT

- Before simplification: z = ABC+AB'(A'C')'



- After simplification: z = A(C+B')



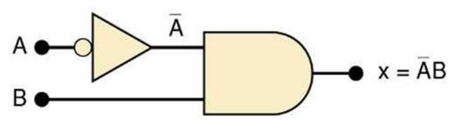
14

SIMPLIFYING A LOGIC CIRCUIT

- Step 1: setup its truth table, figure (a)
- Step 2: write an AND term for each row where output is 1
- Step 3: combine the AND terms in SOP form
- Step 4: simplify the expression by Boolean algebra, if possible
- Step 5: implement the simplified circuit, figure (b)

A	B	x
0	0	0
0	1	1
1	0	0
1	1	0

(a)



(b)

15

EXAMPLE (SIMPLIFICATION)

- Simplify a logic circuit with the following truth table

Truth table.

A	*B*	*C*	*x*	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \bar{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\bar{B}C$
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	1	$\rightarrow ABC$

AND terms for each case where output is 1.

SOP expression for the output:

$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

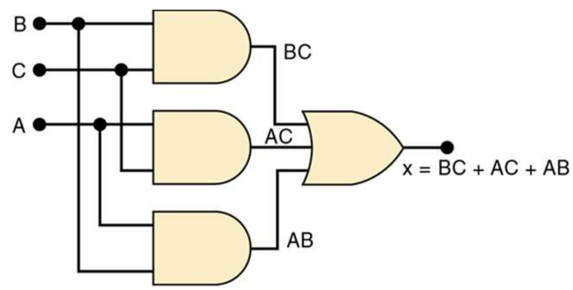
16

EXAMPLE (SIMPLIFICATION)

- Design a logic circuit with three inputs, A, B, and C.

A	B	C	X
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

$$x = BC + AC + AB$$



17

SIMPLIFYING A BOOLEAN EXPRESSION

- Simplify the Boolean function, $\mathrm{F}(A, B, C) = \sum m(0,2,3,4,5,7)$
- Calculation 1:

$$\begin{aligned} F &= \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}BC + \overline{A}B\overline{C} + ABC \\ &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC + A\overline{B}\,\overline{C} + A\overline{B}C \\ &= \overline{A}\,\overline{C}(B + \overline{B}) + BC(A + \overline{A}) + A\overline{B}(\overline{C} + C) \\ &= \overline{A}\,\overline{C} + BC + A\overline{B} \end{aligned}$$

- Calculation 2:

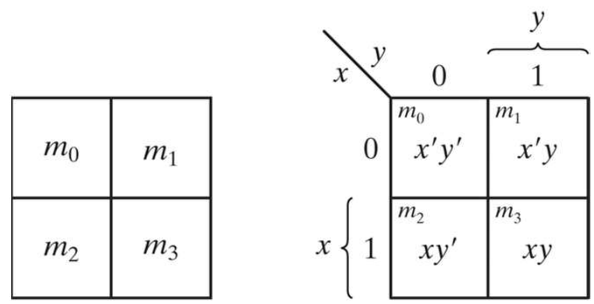
$$\begin{aligned} F &= \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}BC + \overline{A}B\overline{C} + ABC \\ &= \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} + \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + ABC \\ &= \overline{B}\,\overline{C}(\overline{A} + A) + \overline{A}B(C + \overline{C}) + AC(B + \overline{B}) \\ &= \overline{B}\,\overline{C} + \overline{A}B + AC \end{aligned}$$

- Both calculations are correct and having the same numbers of terms, is there any systematic way to simplify it?

18

KARNAUGH MAP METHOD

- K-map method is a graphical method of simplifying logic equations or truth tables
- Theoretically, it can be used for any number of input variables, but practically limited to 5 or 6 variables
- For 2 input variables, x and y

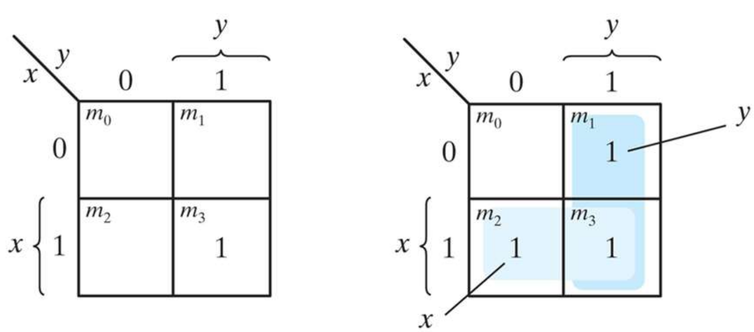


(a) (b)

19

2-VARIABLE K-MAP

- For the following expressions, we have the K-maps below:
  - (a) xy
  - (b) x+y



(a) xy (b) $x + y$

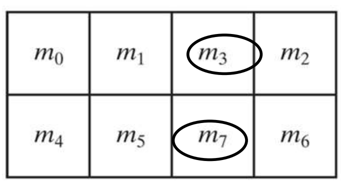
m1m3->y
m2m3->x
so x +y
or
m1m3->y
m2->xy'
so xy' +y=y+xy'
note:x+x' y=x+y
so y+xy' =y+x=x+y

20

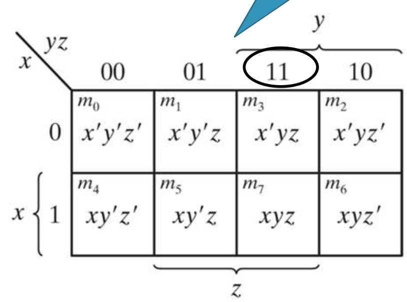
3-VARIABLE K-MAP

- 3-variable K-map

Only 1 variable changed at the neighborhoods



(a)

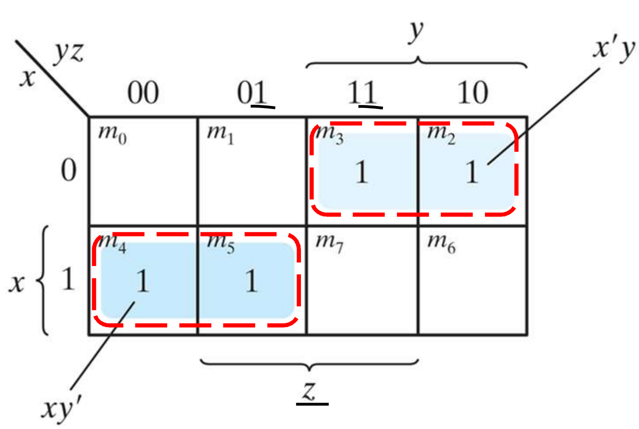


(b)

21

EXAMPLE (3-VARIABLE K-MAP)

- Simplify $F(x,y,z) = \sum(2,3,4,5)$



- Ans: $F(x,y,z) = x'y + xy'$

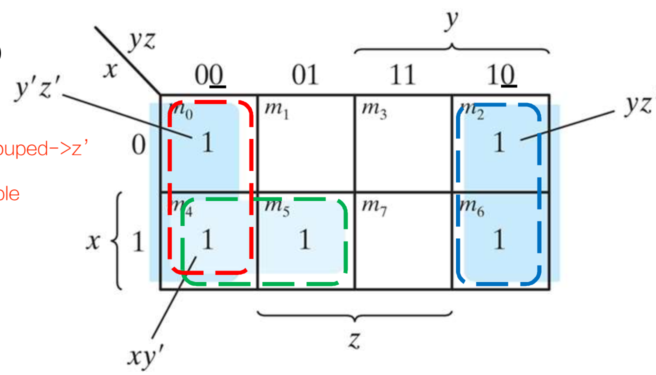
22

EXAMPLE (3-VARIABLE K-MAP)

- Simplify $F(x,y,z) = \sum(0,2,4,5,6)$

red and blue region can be grouped->z'

larger group, term is more simple
so m4m5 group-> xy'



Note: $y'z' + yz' = z'$

- Ans: $F(x,y,z) = y'z' + yz' + xy' = z' + xy'$

4-VARIABLE K-MAP

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(a)

wx \ yz	00	01	11 (y)	10 (y)
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01 (x)	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11 (w, x)	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10 (w)	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

z

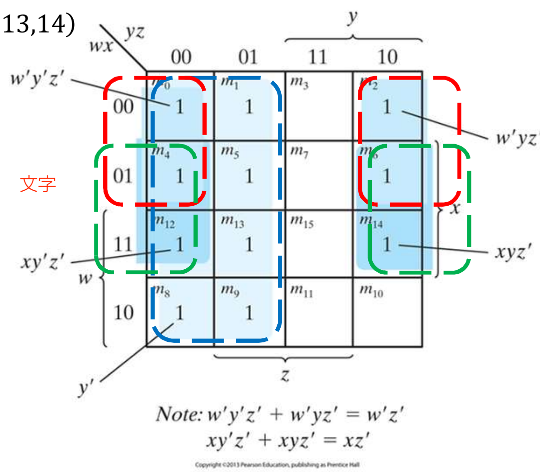
(b)

EXAMPLE (4-VARIABLE K-MAP)

- **Simplify** $F(w,x,y,z) = \sum(0,1,2,4,5,6,8,9,12,13,14)$

recall: group term in 2^n times
n=0,1,2

文字



- **Ans:** $F(x,y,z) = y' + w'z' + xz'$

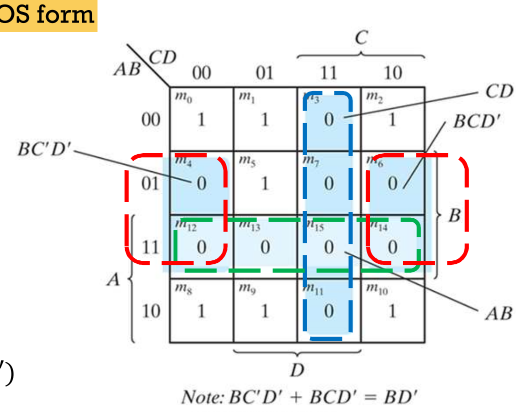
SOP

EXAMPLE (4-VARIABLE K-MAP TO POS)

- **Simplify** $F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$ **in POS form**
- $F'(A,B,C,D) = \sum(3,4,6,7,11,12,13,14,15)$
- $F'(A,B,C,D) = BD' + CD + AB$

De Morgen thm

- $F(A,B,C,D) = (BD' + CD + AB)'$
- $F(A,B,C,D) = (BD')'(CD)'(AB)'$



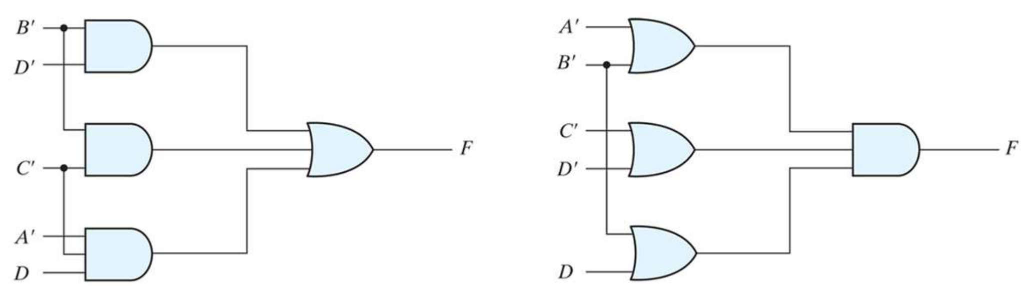
- **Ans:** $F(A,B,C,D) = (B' + D)(C' + D')(A' + B')$

TWO-LEVEL IMPLEMENTATIONS

- From the previous example, $F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$ can be simplified
  - In SOP form by collecting the 1s as $F(A,B,C,D) = B'D' + B'C' + A'C'D$
  - In POS form by collecting the 0s as $F(A,B,C,D) = (B'+D)(C'+D')(A'+B')$
- The implementation of a function in a standard form is said to be a two-level implementation
  - In SOP form, AND gates are connected to a single OR gate
  - In POS form, OR gates are connected to a single AND gate

TWO-LEVEL IMPLEMENTATIONS

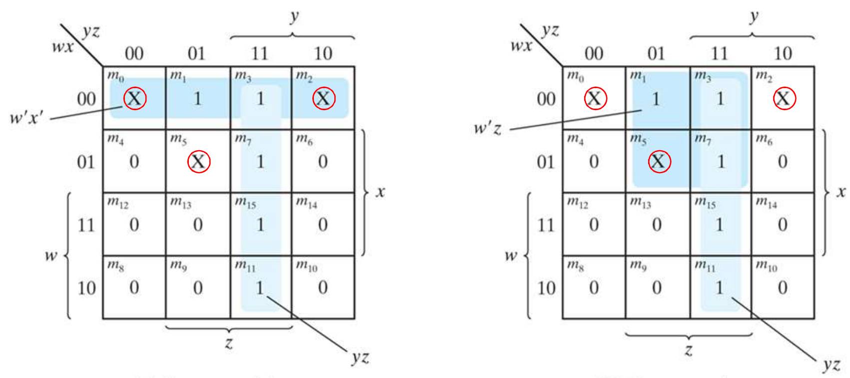
- (a) In SOP form, AND gates are connected to a single OR gate
- (b) In POS form, OR gates are connected to a single AND gate



(a) $F = B'D' + B'C' + A'C'D$ (b) $F = (A'+B')(C'+D')(B'+D)$

EXAMPLE (K-MAP WITH "DON'T CARE")

- Simplify $F(w,x,y,z) = \sum(1,3,7,11,15)$ with m_0, m_2, and m_5 are "Don't Care".



(a) $F = yz + w'x'$

(b) $F = yz + w'z$

GUIDELINES OF K-MAP

- Group number of cells in powers of 2
- Group as many cells as possible
- The larger the group, the less number of variables in that term
- Make as few groups as possible to reduce the number of terms

EXAMPLE (ALARM DESIGN)

- A manufacturing plant is going to setup an alarm system to monitor a chemical production process
- Sensor inputs include Temperature (T), Pressure (P), Weight (W), and Level (L)
- The process may fail if any one of the following conditions occur:
  - Temperature, Pressure, and Level are high
  - Low Level with high Temperature and Weight
  - Low Level and Temperature with high Pressure
  - Low Level and Weight with high Temperature

31

EXAMPLE (ALARM DESIGN)

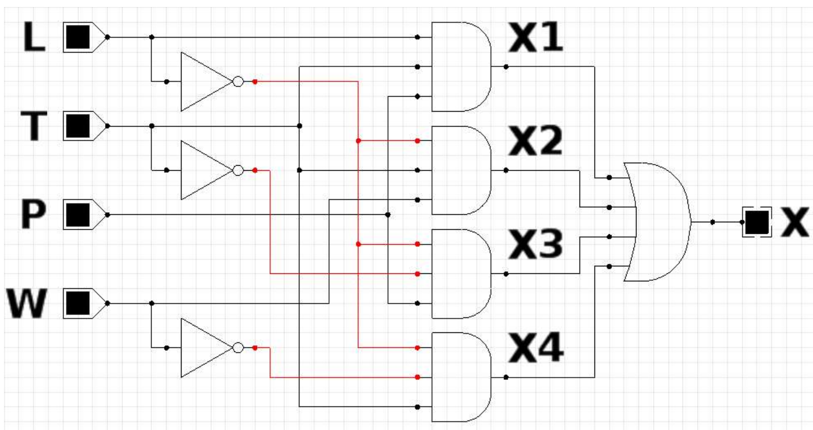
- Covert the conditions to Boolean expressions
  - X1 = LTP
  - X2 = L'TW
  - X3 = L'T'P
  - X4 = L'W'T

1. Temperature, Pressure, and Level are high
2. Low Level with high Temperature and Weight
3. Low Level and Temperature with high Pressure
4. Low Level and Weight with high Temperature

- The alarm output becomes:
  - X = X1 + X2 + X3 + X4
  - X = LTP + L'TW + L'T'P + L'W'T

32

EXAMPLE (ALARM DESIGN)



EXAMPLE (ALARM DESIGN)

- Let's simplify the circuit/expression, X = LTP + L'TW + L'T'P + L'W'T
- Truth Table

L	P	T	W	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1

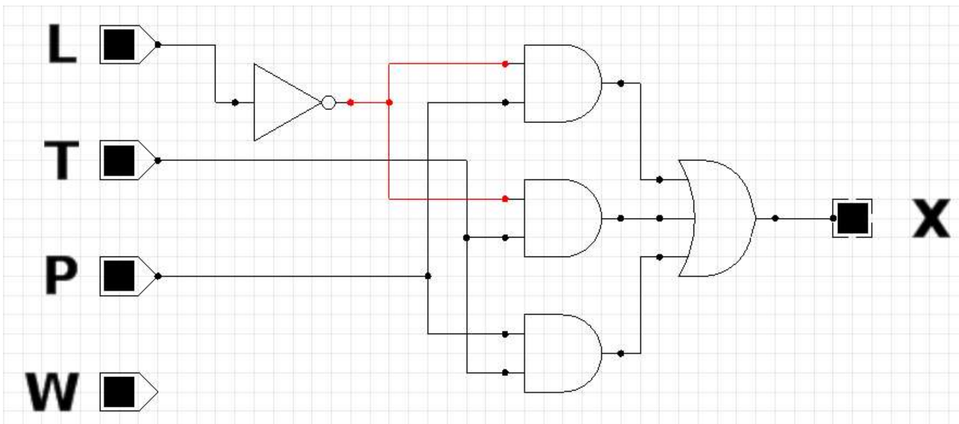
L	P	T	W	X
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

K-map

	T'W'	T'W	TW	TW'
L'P'			1	1
L'P	1	1	1	1
LP			1	1
LP'				

- Then we have: X = L'P + L'T + PT

EXAMPLE (ALARM DESIGN)



35

ANY QUESTIONS ?

36