Chapter 2 Combinationalal Logic Circuit Practice

2-1 Practice purpose

1. To understand the character of Boolean Algebra
2. To know how to use Karnaugh maps
3. To understand the simplified techniques of logic gate and the design techniques of combinational logic.
4. To understand conversion among all the logic gates.

2-2 Practice theory

The basic combinational logic circuit square figure is as figure 2-1 which is combined with three parts, including input variables, logic gate circuit, and output variables. A logic gate usually accepts input signals and necessary outputs signals as well. This procedure is to convert a pre set input signal into a necessary output signal. Therefore, basically combinational logic circuit is just a code converter.

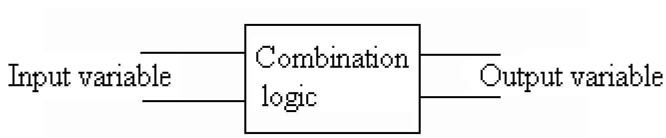


Figure 2-1

2-2-1 The logic analysis of NAND and NOR

1. The logic analysis of NAND

We use NAND to design all kinds of logic circuits and there are rules for design. Let us have some examples.

Example 2-1, find Figure 2-2 function F.

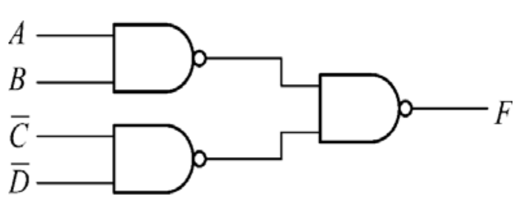


Figure 2-2

Solution:

According to NAND's definition and Boolean algebra, we can solve F as:

$$
\begin{aligned}
F &= \overline{\overline{AB}\,\overline{\overline{C}\,\overline{D}}} \\
&= \overline{\overline{AB}} + \overline{\overline{\overline{C}\,\overline{D}}} \\
&= AB + \overline{C}\,\overline{D}
\end{aligned}
$$

Therefore, we can draw an circuit as:

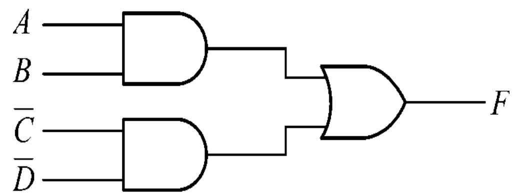


Figure 2-3

Comparing the two figures, we realize that the original input NAND gate's small circle is missing and it turns out to be AND gate. The output gate becomes OR. Here is one more example.

Example 2-2, solve figure 2-4 and F.

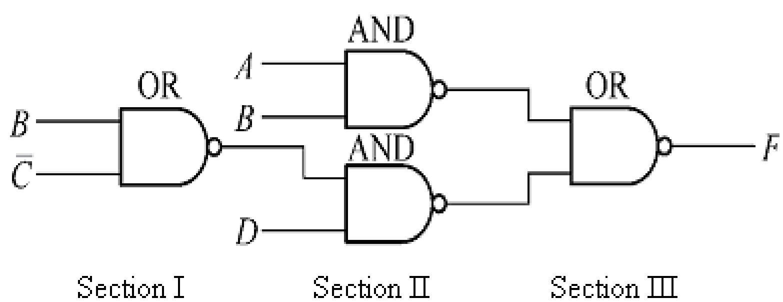


Figure 2-4

Solution:
From output to input, the first class NAND has become OR
Second class NAND has become AND
Third class NAND has become OR
The third class is an odd number input; that is why input variable is the supplement

To conver $B\overline{C}$ to $\overline{B}+C$

Therefore, output $F=AB+(\overline{B}+C)D$, as figure2-5

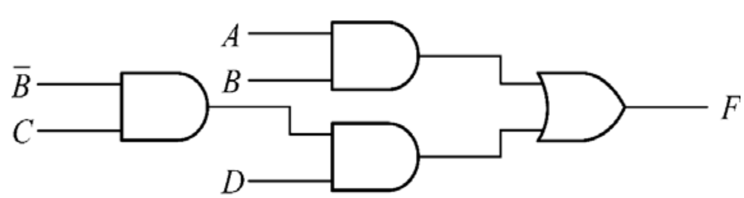


Figure 2-5

Example 2-3, solve figure 2-6 F.

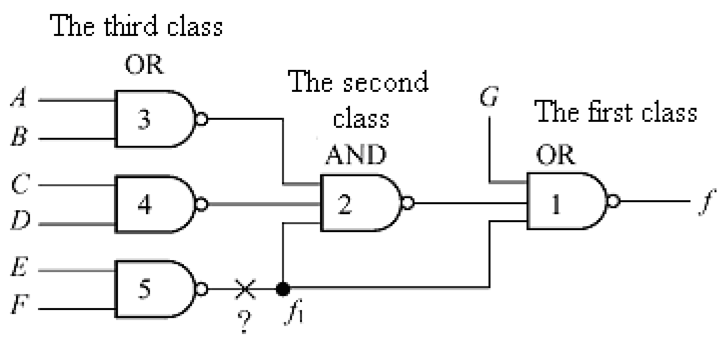
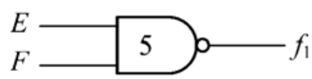


Figure 2-6

Solution:



Solution:

$$f_1 = \overline{E} + \overline{F}$$

$$= (\overline{EF})$$

$$\therefore f_1 = \overline{G} + \overline{f_1} + (\overline{A} + \overline{B})(\overline{C} + \overline{D})(\overline{E} + \overline{F})$$

$$= \overline{G} + EF + (\overline{A} + \overline{B})(\overline{C} + \overline{D})(\overline{E} + \overline{F})$$

According to the above example, we realize the rules of NAND combinational logic gate are as follows:

(1) Output to input, convert odd number NAND to OR.
(2) Output to input, convert even number NAND to AND
(3) Odd and even numbers become supplements
(4) The rule shows that a series of NAND converts to OR-AND-OR-AND.

2. Logic analysis of NOR

NOR and NAND gates are the same, there are some rules we notice when we process the logic analysis as follows:

(1) Output to input, turn odd NOR to AND.
(2) Output to input, turn even NOR to OR.
(3) Invert odd input variable to supplement.
(4) A series of NOR turn to AND-OR-AND-OR from output to input conversion.

Follow the examples.

Example 2-4, solve figure 2-7 output function F.

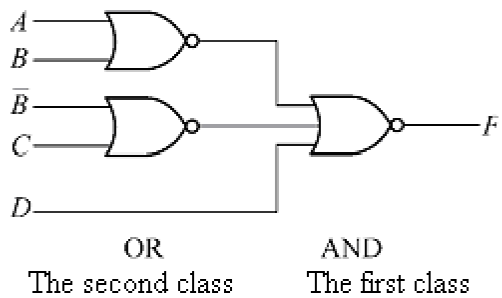


Figure 2-7

Solution:

From output to input, the first class NOR turn to AND

From output to input, the second class NOR turn to OR

The first class input is odd AND, therefore input variable D converts to

$\overline{D}$，therefore the function formula is $F=(A+B)(\overline{B}+C)\overline{D}$，as figure 2-8

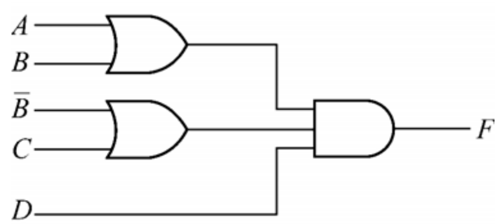


Figure 2-8

Example 2-5, as figure 2-9, to solve F

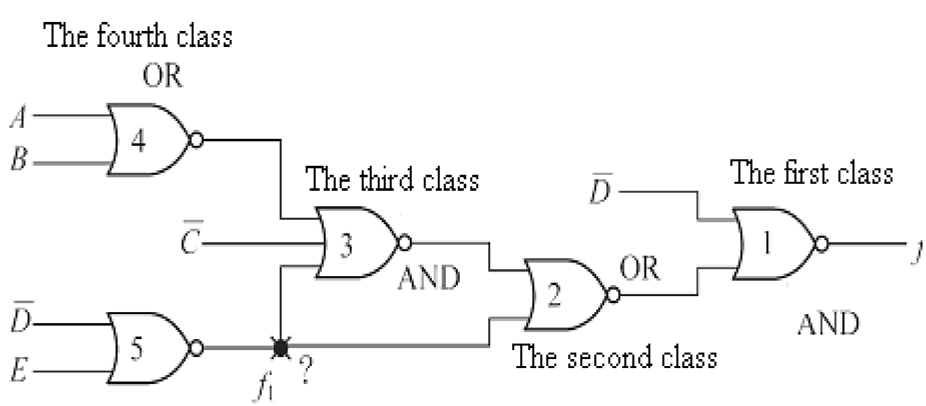
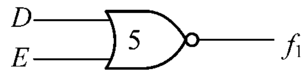


Figure 2-9

Solution：



$$f=(\overline{\overline{D}})\,(f_1+(A+B)(C\cdot\overline{f_1})$$

$$=D[\overline{D}\,\overline{E}+(A+B)C(\overline{\overline{D+E}})]$$

$$=D[\overline{D}\,\overline{E}+(A+B)\cdot C(C+E)]$$

2-2-2 Karnaugh map

Any kind of logic function can always be expressed with two standard forms which are the sum of product and product of sum. Then, we can simplify them with Boolean algebra to get the result of a synthetic circuit. But the result we get is not necessarily the most economical and simply one. Therefore we are going to discuss how to use a Karnaugh map.

A Karnaugh map can be called the explanation of real value figures. Using a figure to explain how something works started from Veitch and then Karnaugh reorganized it. Karnaugh was an American electronic engineer who developed the Karnaugh map in 1953. The advantage of a Karnaugh map is to use maps to understand, which makes everything simpler but the disadvantage is that it is not easy to simplify more than four variables. This will be discussed in chapter 2-2-3, Quine-McCluskey.

Karnaugh maps:

1. One variable Karnaugh map

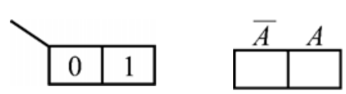


Figure 2-10 one variable Karnaugh map

2. Two variable Karnaugh map

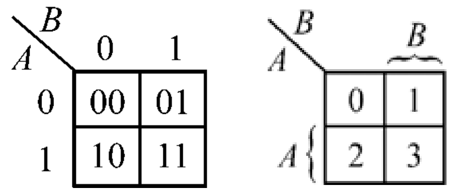


Figure 2-11 two variable Karnaugh map

3. Three variable Karnaugh map

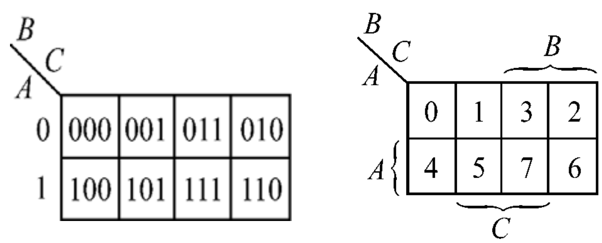


Figure 2-12 three variable Karnaugh map

4. Four variable Karnaugh map

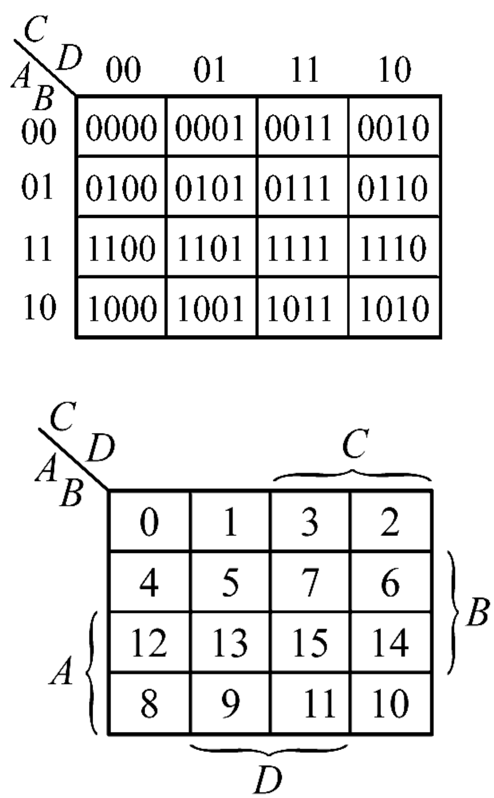


Figure 2-13 four variable Karnaugh map

5. Five variable Karnaugh map

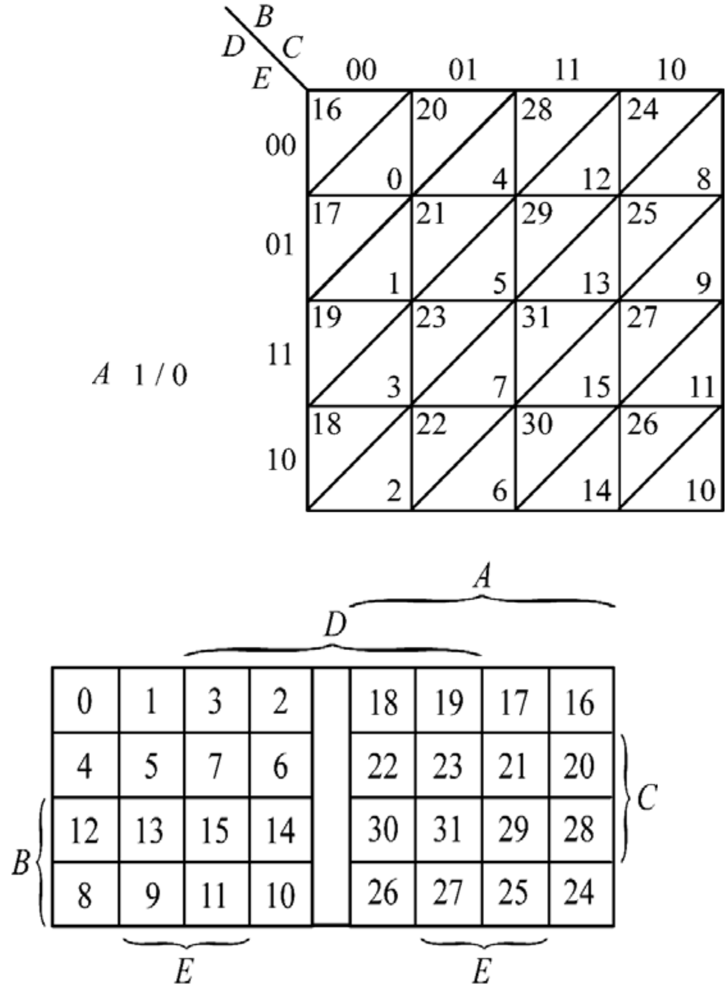
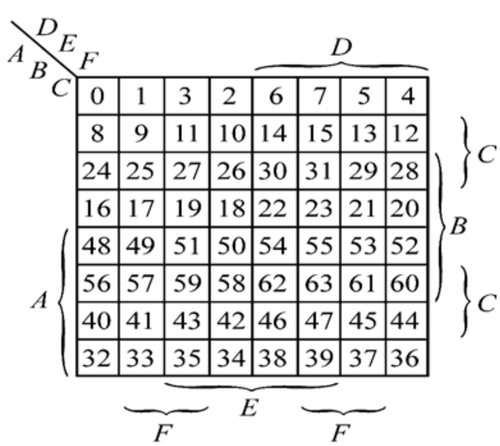


Figure 2-14 Five variable Karnaugh map

6. Six variable Karnaugh map



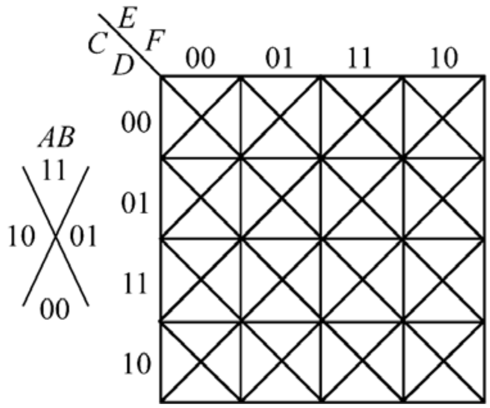


Figure 2-15 six variable Karnaugh map

The usage of Karnaugh map

As figure 2-16 shows that 011, 001, 010, and 111 are next to each other. 100 and 110 are next to each other.

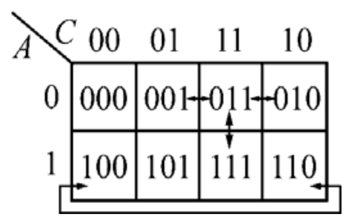
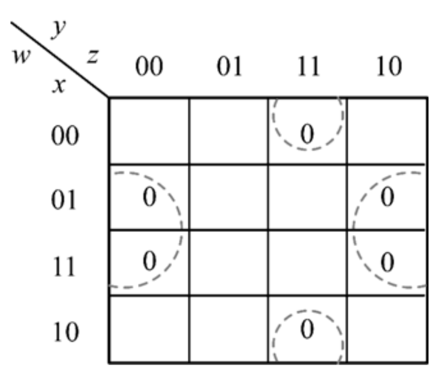


figure2-16

Example: figure 2-16 Simplify $f(w,x,y,z)=\pi(3,4,6,11,12,14)$。

Solution:



$f=(w,x,y,z)$

$=(\bar{x}+z)(x+\bar{y}+\bar{z})$

Example2-17 Karnaugh map function F

Solution: It seems there is no solution, but actually XOR and XNOR can solve the problem.

F: AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
01	1	0	1	0

$\rightarrow \overline{A}\overline{B}\overline{C}D,\ \overline{A}\overline{B}C\overline{D} \rightarrow \overline{A}\overline{B}(C\oplus D) \longrightarrow ①$

$\rightarrow \overline{A}B\overline{C}\overline{D},\ \overline{A}BCD \rightarrow \overline{A}B(\overline{C\oplus D}) \longrightarrow ②$

$\rightarrow AB\overline{C}D,\ ABC\overline{D} \rightarrow AB(C\oplus D) \longrightarrow ③$

$\rightarrow A\overline{B}\overline{C}\overline{D},\ A\overline{B}C\overline{D} \rightarrow A\overline{B}(\overline{C\oplus D}) \longrightarrow ④$

figure 2-17

① + ③ = $(C\oplus D)(\overline{A}\ \overline{B}+AB)$

$= (C\oplus D)(A\odot B)$

$= (7C\oplus D)(\overline{A\oplus B})$ ……… ⑤

② + ③ = $(\overline{C\oplus D})(\overline{A}B+A\overline{B})$

$= (\overline{C\oplus D})(A\oplus B)$ ……… ⑥

if $E = A\oplus B, G=C\oplus D$

then F = ⑤ + ⑥

$= (C\oplus D)(\overline{A\oplus B})+(\overline{C\oplus D})(A\oplus B)$

$= G\cdot\overline{E}+\overline{G}\cdot E$

$= G\oplus E$

$= A\oplus B\oplus C\oplus D$

Example 2-18 Solve Karnaugh map function F

Solution: similar to example 2-17

F: AB \ CD	00	01	11	10
00	1	0	1	0
01	0	1	0	1
11	1	0	1	0
01	0	1	0	1

$\rightarrow \overline{A}\overline{B}\overline{C}\overline{D},\ \overline{A}\overline{B}CD \rightarrow \overline{A}\overline{B}(\overline{C\oplus D}) \longrightarrow ①$

$\rightarrow \overline{A}B\overline{C}D,\ \overline{A}BC\overline{D} \rightarrow \overline{A}B(C\oplus D) \longrightarrow ②$

$\rightarrow AB\overline{C}\overline{D},\ ABCD \rightarrow AB(\overline{C\oplus D}) \longrightarrow ③$

$\rightarrow A\overline{B}\overline{C}D,\ A\overline{B}C\overline{D} \rightarrow A\overline{B}(C\oplus D) \longrightarrow ④$

figure 2-18

$$
\begin{aligned}
F &= ① + ② + ③ + ④ \\
&= A\overline{B}(\overline{C \oplus D}) + \overline{A}B(C \oplus D) + AB(\overline{C \oplus D}) + A\overline{B}(C \oplus D) \\
&= (\overline{C \oplus D})(\overline{A}\,\overline{B} + AB)(C \oplus D)(\overline{A}B + A\overline{B}) \\
&= (\overline{C \oplus D})(\overline{A \oplus B}) + (C \oplus D)(A \oplus B)
\end{aligned}
$$

$$
\text{if } X = A \odot B, Y = C \odot D
$$

$$
\begin{aligned}
\text{Then } F &= \overline{X}\,\overline{Y} + XY = X \odot Y \\
&= A \odot B \odot C \odot D
\end{aligned}
$$

2-2-3 Quine-McCluskey method

As for normal simplification, using the Karnaugh map is a very effective way, of simplifying switching functions with a small amount of variables.

List method is also called Quine-McCluskey method which has two simplified steps. The first one is the ensuring of necessity and the second one is the choice of necessity.

1. The ensuring of necessity.

The way to find the ensuring of necessity is using matching procedure. This procedure is to make a comparison between " Prime" and other "Prime". If there is only one variable difference between two Primes it means the variable can be deleted. For example, 010$\underline{0}$(4), 010$\underline{1}$(5), has formed a new 010$\underline{X}$. At the same time, make a "√" to note the unnecessary one.

There is one thing we have to remember when using the List method is to combine 0 of 1, 1 of 1 and 2 of 1 together as figure 2-19.

W	X	Y	Z	
0	0	0	0	0 Prime has zero 1
0	0	0	1	1
0	0	1	0	2
0	1	0	0	4 Prime has one 1
1	0	0	0	8
0	0	1	1	3
0	1	0	1	5
0	1	1	0	6 Prime has two 1
1	0	0	1	9
1	0	1	0	10
1	1	0	0	12
0	1	1	1	7
1	0	1	1	11
1	1	0	1	13 Prime has three 1
1	1	1	0	14
1	1	1	1	15 Prime has four 1

Figure 2-19

Example: using list method to simplify Boolean function

Solution:

WXYZ	
0000	0 ✓ (Prime has zero)
0001	1 ✓
0010	2 ✓ (Prime has one 1)
0100	8 ✓
1010	10 ✓
1011	11 ✓ (Prime has three 1)
1110	14 ✓
1111	15 ✓ (Prime has four 1)

(a)

WXYZ	
000X	(0, 1)
00X0	(0, 2) ✓
X000	(0, 8) ✓
X010	(2, 10) ✓
10X0	(8, 10) ✓
101X	(10, 10) ✓
111X	(10, 14) ✓
1X11	(11, 15) ✓
111X	(14, 15) ✓

(b)

W X Y Z	
× 0 × 0	0 , 2 , 8 ,10
× 0 × 0	0 , 8 , 2 ,10
1 × 1 ×	10,11,14,15
1 × 1 ×	10,14,11,15

(c)

∵Necessity with no
“ √ ”)

$\overline{W}\,\overline{X}\,\overline{Y}, \overline{X}\,\overline{Z}\ , WY$

$\therefore F = \overline{W}\,\overline{X}\,\overline{Y} + \overline{X}\,\overline{Z} + WY$

Figure 2-20

2. The choice of necessity

The solution above is not necessary the min. one

The combinational function simplification has to be chosen from the necessities. Each necessity in the necessity figure has one row. The row with X means the prime of necessity. Therefore we have to choose the min. necessity to combine the function ‘s prime. The example below shows that not only do we need to ensure the necessity, but also have to choose necessity. Otherwise it is not a simplified function.

Example: find the function’s necessities
$F(W,X,Y,Z)=\Sigma(1,4,6,7,8,9,10,11,15)$

Solution: ensuring of necessity

W X Y Z		
0 0 0 1	1 ✓	
0 1 0 0	4 ✓	Prime has one 1
1 0 0 0	8 ✓	
0 1 0 0	6 ✓	
1 0 0 1	9 ✓	Prime has two 1
1 0 1 0	10✓	

W X Y Z		
0 0 0 1	7 ✓	
0 1 0 0	11✓	Prime has three 1
1 1 1 1	15✓	Prime has four 1

(a)

W X Y Z	
× 0 0 1	(0,2)
0 1 × 0	(4,6)
1 0 0 ×	(8,9) ✓
1 0 × 0	(8,10) ✓
0 1 1 ×	(6,7)
1 0 × 1	(9,11) ✓
1 0 1 ×	(10,11) ✓
× 1 1 1	(7,15)
1 × 1 1	(11,15)

(b)

W X Y Z	
1 0 × ×	(8,9,10,11)
1 0 × ×	(8,9,10,11)

(c)

Figure2-21

we know, from simplification, that the ensuring of necessity is

$$F=W\overline{X} + \overline{X}\,\overline{Y}Z + \overline{W}X\overline{Z} + \overline{W}XY + XYZ + WYZ$$

Not the most simplified function, as K map proves

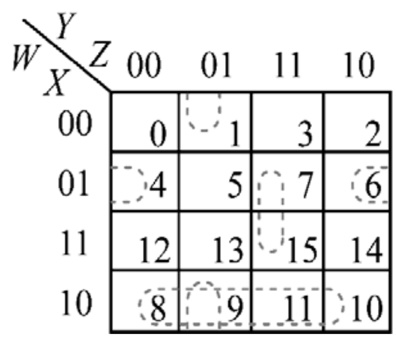


Figure 2-22

$$F\,(W,X,Y,Z)=\overline{X}\,\overline{Y}Z + \overline{W}X\overline{Z} + XYZ + W\overline{X}$$

(2) The choice of necessity

	1	4	6	7	8	9	10	11	15
$\overline{X}\,\overline{Y}Z$ (1,9)	×					×			
$\overline{W}X\overline{Z}$ (4,6)		×	×						
$\overline{W}XY$ (6,7)			×	×					
XYZ (7,15)				×					×
WYZ (11,15)								×	×
$W\overline{X}$ (8,9,10,11)					×	×	×	×	
	✓	✓	✓		✓	✓	✓	✓	✓

Figure2-23

the figure above including"1 " is $\overline{X}\,\overline{Y}Z$

including "4" is $\overline{W}X\overline{Z}$

including "8" is $W\overline{X}$

That is why in figure 2-23 the necessity has $\overline{X}\,\overline{Y}Z$、$\overline{W}X\overline{Z}$、$W\overline{X}$ ， the function is $F(W,X,Y,Z)=\overline{X}\,\overline{Y}Z+\overline{W}X\overline{Z}+W\overline{X}+XYZ$(plus " XYZ " , the reason is " XYZ " " 7 " and " 15 " do not have " √ ")

Example: 2-11 simplify the function with the List method

$f(W,X,Y,Z) = \pi(3,4,5,6,7,9,12,13,14)$

solution: $f(W,X,Y,Z) = \pi(3,4,5,6,7,9,12,13,14)$

$= \Sigma(0,1,2,8,10,11,15)$

(1) The ensuring of necessity

W X Y Z			
0 0 0 0	0	Prime has zero 1	√√√
0 0 0 1	1	Prime has one 1	√
0 0 1 0	2		√√
1 0 0 0	8		√√
1 0 1 0	10	Prime has two 1	√√√
1 0 1 1	11	Prime has three 1	√
1 1 1 1	15	Prime has four 1	

(a)

W X Y Z	
0 0 0 ×	(0,1)
0 0 × 0	(0,2) √
× 0 0 0	(0,8) √
× 0 1 0	(2,10) √
1 0 × 0	(8,10) √
1 0 1 ×	(10,11)
1 × 1 1	(11,15)

(b)

W X Y Z	
× 0 × 0	(0,2,8,10)
× 0 × 0	(0,2,8,10)

(c)

Figure2-24

(2) The choice of necessity

		0	1	2	8	10	11	15
✓	$\overline{W}\,\overline{X}\,\overline{Y}$	×	×				×	
	$\overline{W}\,\overline{X}\,Y$					×	×	
✓	$W\,Y\,Z$						×	×
✓	$\overline{X}\,\overline{Z}$	×		×	×	×		
		✓	✓	✓	✓	✓	✓	✓

Figure2-25

as figure2-25shows, only $\overline{W}\,\overline{X}\,\overline{Y}$ has "0"

pm;u wyz has "15"

pm;u $\overline{X}\,\overline{Z}$ has "2, 8"

so $(W,X,Y,Z) = \overline{W}\,\overline{X}\,\overline{Y} + WYZ + \overline{X}\,\overline{Z}$

Example:2-12 simplify Boolean function with the List method:

$$F(A,B,C,D,E,F,G)=\Sigma(20,28,38,39,52,60,102,103,127)$$

Solution: (1) the ensuring of necessity

A B C D E F G			
0 0 1 0 1 0 0	20	Prime has two 1	✓ ✓
0 0 1 1 1 0 0	28	Prime has three 1	✓ ✓
0 1 0 0 1 1 0	38		✓ ✓
0 1 1 0 1 0 0	52		✓ ✓
0 1 0 0 1 1 1	39		✓ ✓
0 1 1 1 1 0 0	60	Prime has four 1	✓ ✓
1 1 0 0 1 1 0	102		✓ ✓
1 1 0 0 1 1 1	103	Prime has five 1	✓ ✓
1 1 1 1 1 1 1	127	Prime has seven 1	

(a)

A B C D E F G	
0 0 1 × 1 0 0	(20,28) ✓
0 × 1 0 1 0 0	(20,52) ✓
0 × 1 1 1 0 0	(28,60) ✓
0 1 0 0 1 1 ×	(38,39) ✓
× 1 0 0 1 1 0	(38,102) ✓
0 1 1 × 1 0 0	(52,60) ✓
× 1 0 0 1 1 1	(39,103) ✓
1 1 0 0 1 1 ×	(102,103) ✓

(b)

A B C D E F G	
0 × 1 × 1 0 0	(20,28,52,60)
0 1 × 1 0 0	(20,28,52,60)
× 1 0 0 1 1 ×	(38,39,102,103)
× 1 0 0 1 1 ×	(38,39,102,103)

(c)

Figure2-26

(2) The choice of necessity

	20	28	38	39	52	60	102	103	127
√ $ABCDEFG$									×
√ $\overline{A}CE\overline{F}\overline{G}$	×	×			×	×			
√ $B\overline{C}\overline{D}EF$			×	×			×	×	
	√	√	√	√	√	√	√	√	√

Figure2-27

$$\therefore F(A,B,C,D,E,F,G)=ABCDFG+\overline{A}CE\overline{F}\,\overline{G}+B\overline{C}\,\overline{D}EF$$

2-3 Practice items

2-3-1 Experiment steps

1. AND-OR circuit as figure 2-28

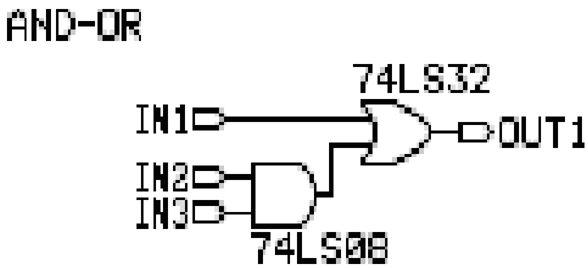


Figure 2-28

2. Input: Connect CON6 PULSE A to CON1 IN1
   Connect CON6 PULSE $\overline{A}$ to CON1 IN2
   Connect CON6 PULSE B to CON1 IN3
   Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1

3. Press A ·Ā·B after finished connecting, as table 2-1， 0 means low logic, LED off, 1 means high logic, LED on.
4. Record LED changes in table 2-1.

A	$\overline{A}$	B	$\overline{B}$	Y(Q1)
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	

Table 2-1

2-3-2

Experiment steps

1. NAND-NAND Circuit, as figure 2-29

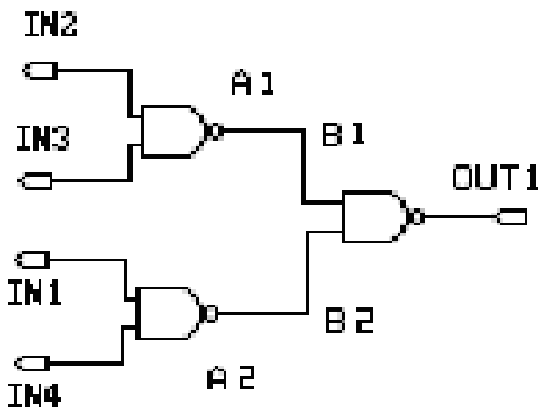


Figure 2-29

2. Input: Connect CON6 A to CON1 IN1
   Connect CON6 $\bar{A}$ to CON1 IN2
   Connect CON6 B to CON1 IN3
   Connect CON6 $\overline{B}$ to CON1 IN4
   Output: CON4 OUT1 to LED DISPLAY CON15 Q1
3. Press A、$\overline{A}$、B、$\overline{B}$, as table 2-2, after finished connecting. 0 means low logic, LED off, 1 means high logic, LED on.
4. Record LED changes in the table 2-2.

A	$\overline{A}$	B	$\overline{B}$	Y(Q1)
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	

Table 2-2

2-3-3

Experiment steps

1. NOR-NOR circuit as figure 2-30

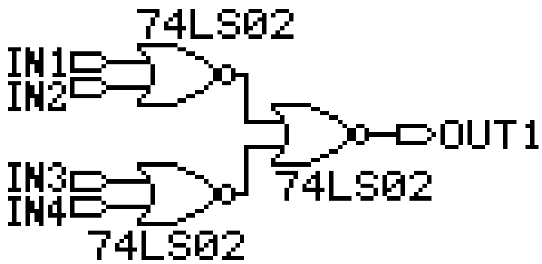


Figure 2-30

2. Input: Connect CON6 A to CON1 IN1
   Connect CON6 Ā to CON1 IN2
   Connect CON6 B to CON1 IN3
   Connect CON6 $\overline{B}$ to CON1 IN4
   Output: CON4 OUT1 to LED DISPLAY CON15 Q1
3. Press A • $\overline{A}$ • B • $\overline{B}$, as table 2-3, after finished connecting. 0 means low logic, LED off, 1 means high logic, LED on.
4. Record LED changes in the table 2-3

A	$\overline{A}$	B	$\overline{B}$	Y(Q1)
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	

Table 2-3

2-3-4

Experiment steps

1. NOR-NOR circuit, as figure 2-31

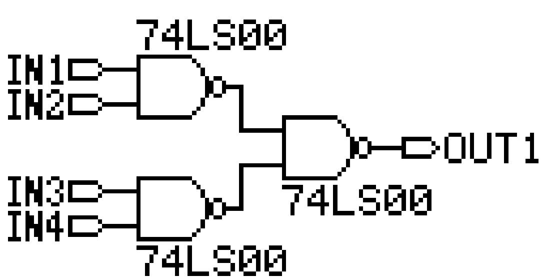


Figure 2-31

2. Input: Connect CON7 S1 to CON1 IN1
   Connect CON7 S2 to CON1 IN2
   Connect CON7 S3 to CON1 IN3
   Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1
3. Press S1, S2, and S3 as figure 2-4, after finished connecting. 0 means low logic, LED off, 1 means high logic, LED on.
4. Record LED changes in table 2-4.

A(S1)	B(S2)	C(S3)	Y(Q1)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 2-4

2-3-5

Experiment steps.

1. Matching circuit, as figure 2-32.

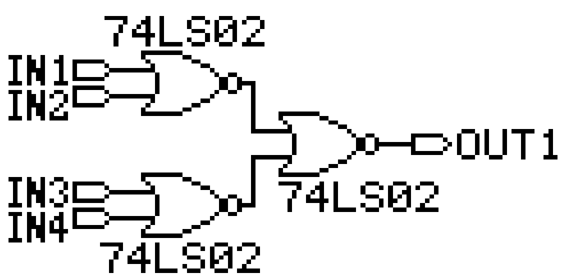


Figure 2-32

2. Input: Connect CON7 S1 to CON1 IN1
   Connect CON7 S2 to CON1 IN2

Connect CON7 S3 to CON1 IN3
Connect CON2 A1 to CON3 B1
Connect CON2 A2 to CON3 B2
Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1

3. Press S1, S2 and S3, as figure 2-5. 0 means low logic, LED off. 1 means high logic, LED on.
4. Record LED changes in figure 2-5

A(S1)	B(S2)	C(S3)	Y(Q1)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 2-5

2-4 Questions & Discussion

1. What is the total of any variables and their supplement?
2. What is the difference between a Truth table and a Karnaugh map?
3. What is the difference between SOP and POS?
4. The first 8 inputs are 1; the last inputs are 0 in the 4 variable Truth Table. Try to draw a Karnaugh map with it and the simple formula.
5. What is x output? Show it with Boolean Algebra.