Digital System Design

Lecture 2
Logic Minimization

Reading Assignment:

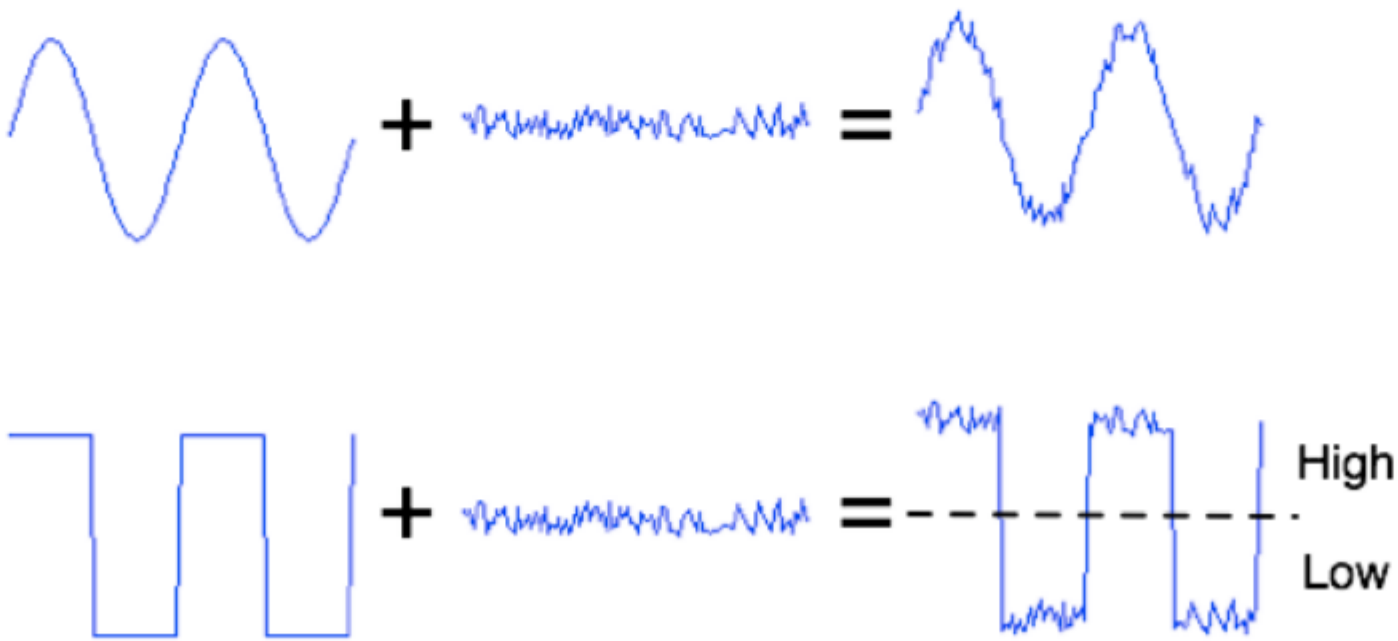
- **Brown, “Fundamentals of Digital Logic with VHDL, pp. 22 - 56, pp. 168 - 207, pp.211 - 219**

Learning Objective:

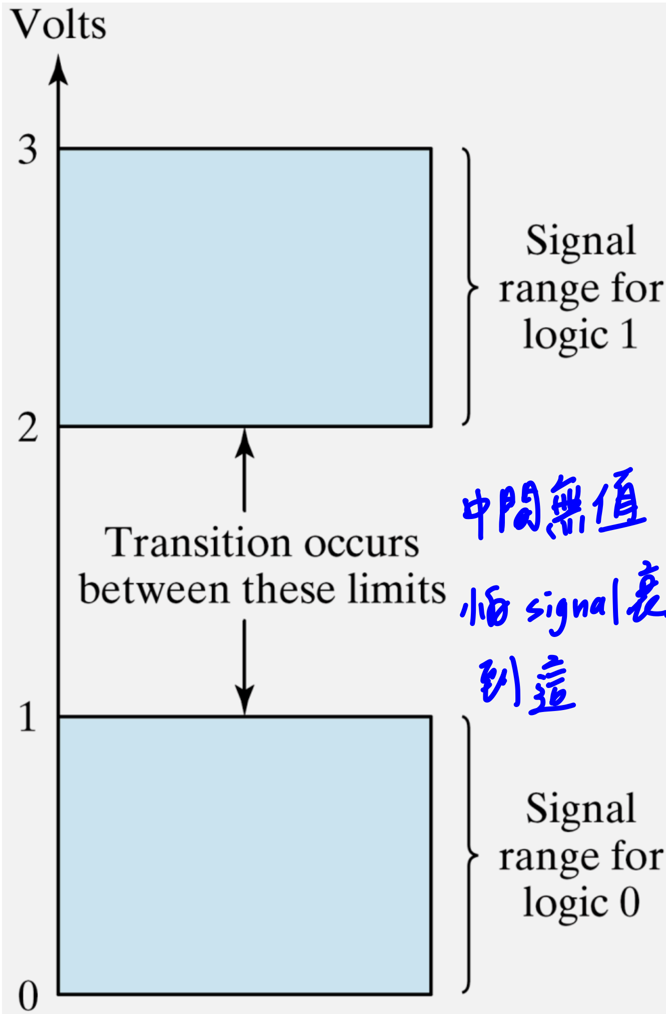
- **Review the Boolean Algebra**
- **Review all aspects of the synthesis process, starting with an initial design and performing the optimization steps needed to generate a desired final circuit**

Analog vs Digital

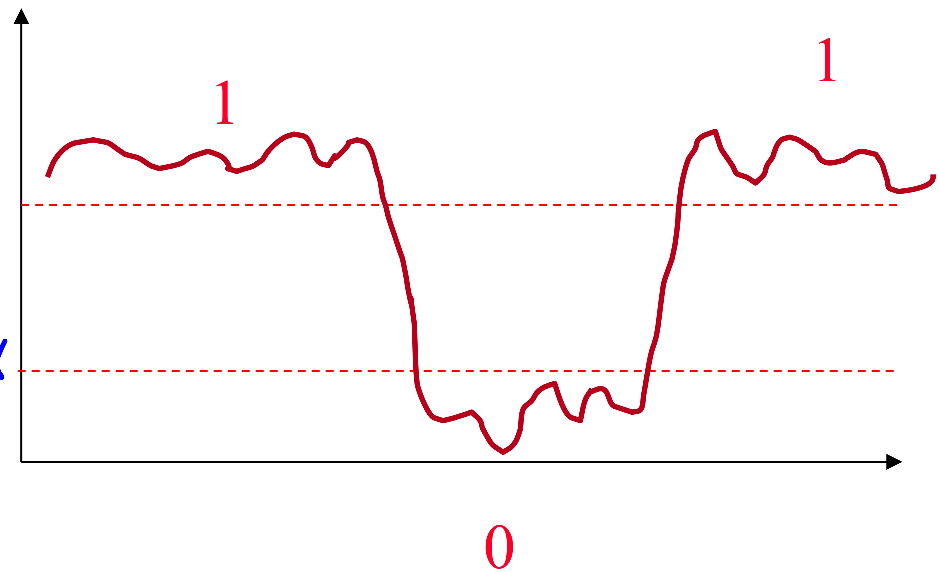
- Digital Information is more robust to noise than analog information.
- In digital information, exact voltage values are not important, only their class (1 or 0).



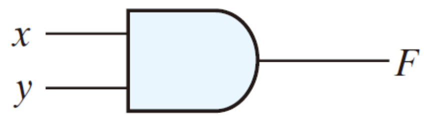
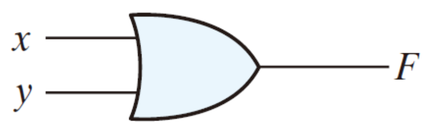
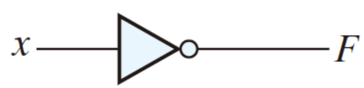
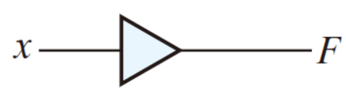
Signal Levels for Binary Logic Values



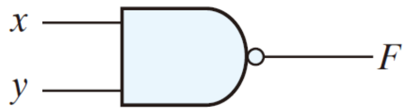
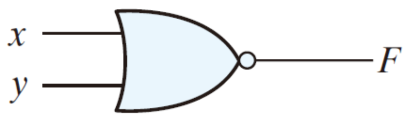
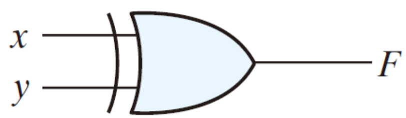
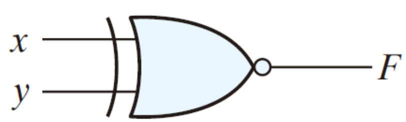
中間無值
怕 signal 衰減
到這



Digital logic gates

<table>
<tr><th>Name</th><th>Graphic symbol</th><th>Algebraic function</th><th>Truth table</th></tr>
<tr><td>AND</td><td>
</td><td>$F = x \cdot y$</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OR</td><td>
</td><td>$F = x + y$</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>Inverter</td><td>
</td><td>$F = x'$</td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>Buffer</td><td>
</td><td>$F = x$</td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td></tr>
</table>

Digital logic gates

NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Basic Theorems and Properties of Boolean Algebra

- Duality
  - the binary operators are interchanged; AND $\Leftrightarrow$ OR
  - the identity elements are interchanged; 1 $\Leftrightarrow$ 0

Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

$\downarrow$

$\underline{x(1+x)} = x$

Minterms and Maxterms

- each maxterm is the complement of its corresponding minterm, and vice versa $m_j' = M_j$

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	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

Conversion Between Canonical Forms

- **Easy to convert between minterm and maxterm representations**
- **For maxterm representation, select rows with 0's**

x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$G = xyz + xyz' + x'yz$$

$$\downarrow$$

$$G = m_7 + m_6 + m_3 = \Sigma(3, 6, 7)$$

$$\downarrow$$

$$G = M_0M_1M_2M_4M_5 = \Pi(0,1,2,4,5)$$

$$\downarrow$$

$$G = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)(x'+y+z')$$

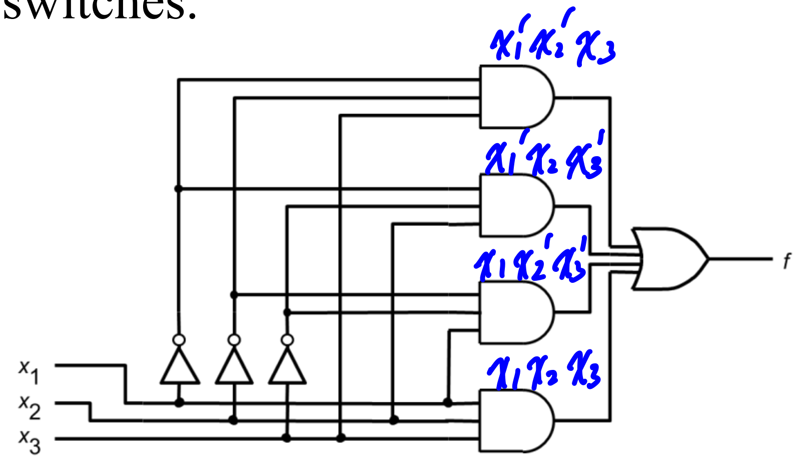
- sum of minterms = product of maxterms

Three-Way Light Control (XOR)

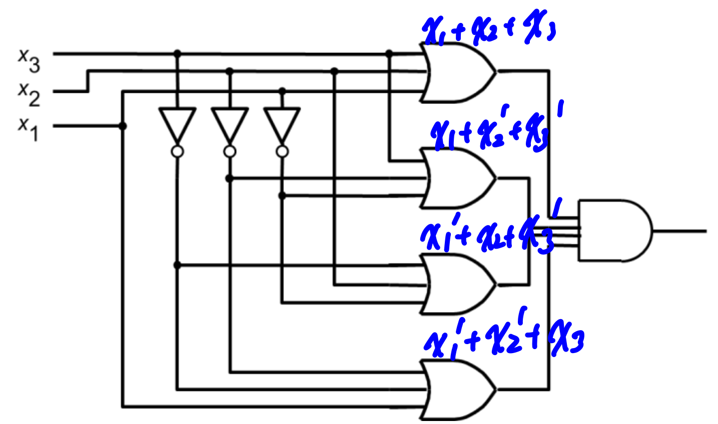
Assume that a large room has three doors and that a switch near each door controls a light in the room. It has to be possible to turn the light on or off by changing the state of any one of the switches.

意思就是每多一個1, status change

x_1	x_2	x_3	f	
0	0	0	0	
0	0	1	1	x_3
0	1	0	1	x_2
0	1	1	0	$x_2 x_3$
1	0	0	1	x_1
1	0	1	0	$x_1 x_3$
1	1	0	0	$x_1 x_2$
1	1	1	1	$x_1 x_2 x_3$



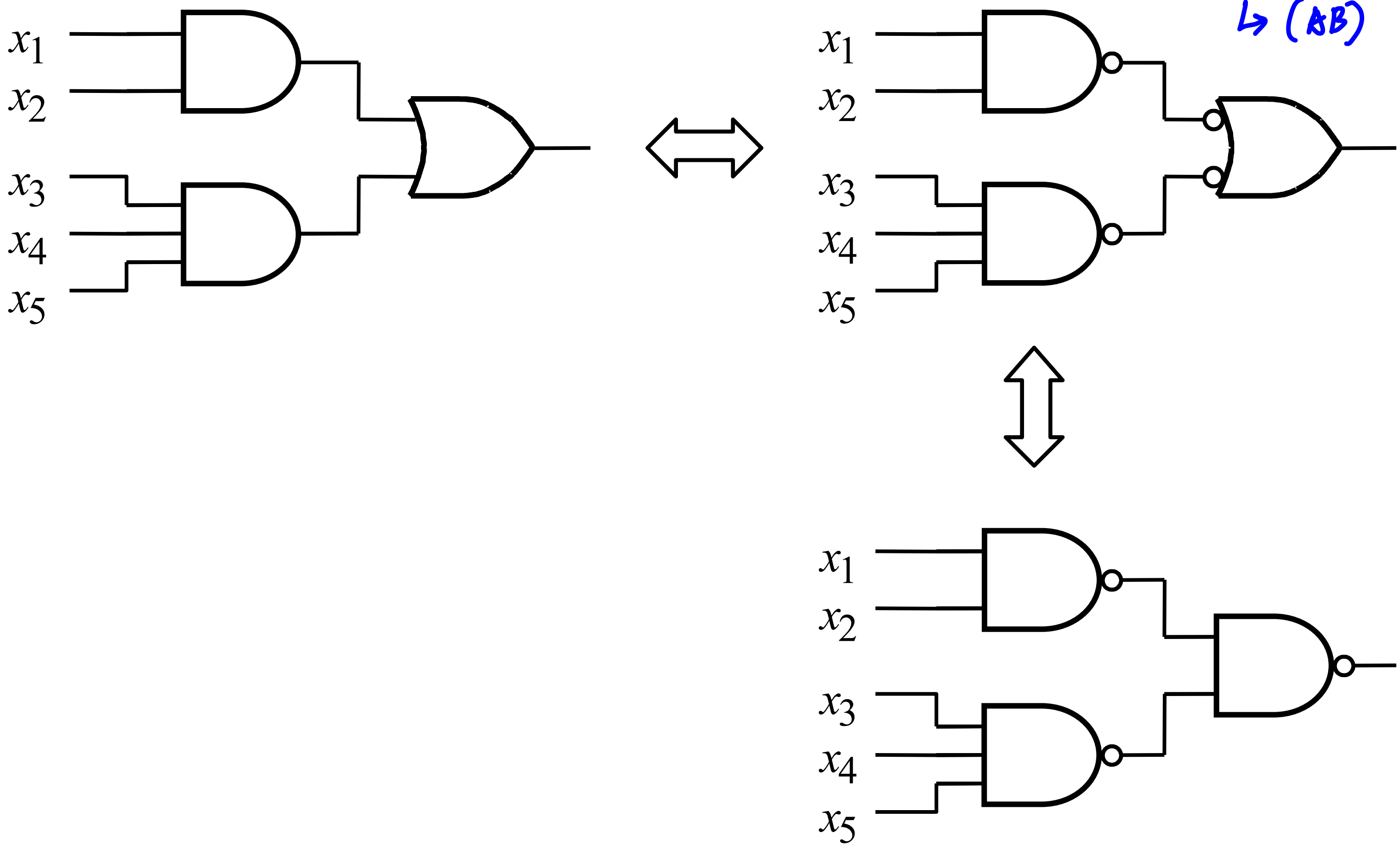
Sum-of-products realization



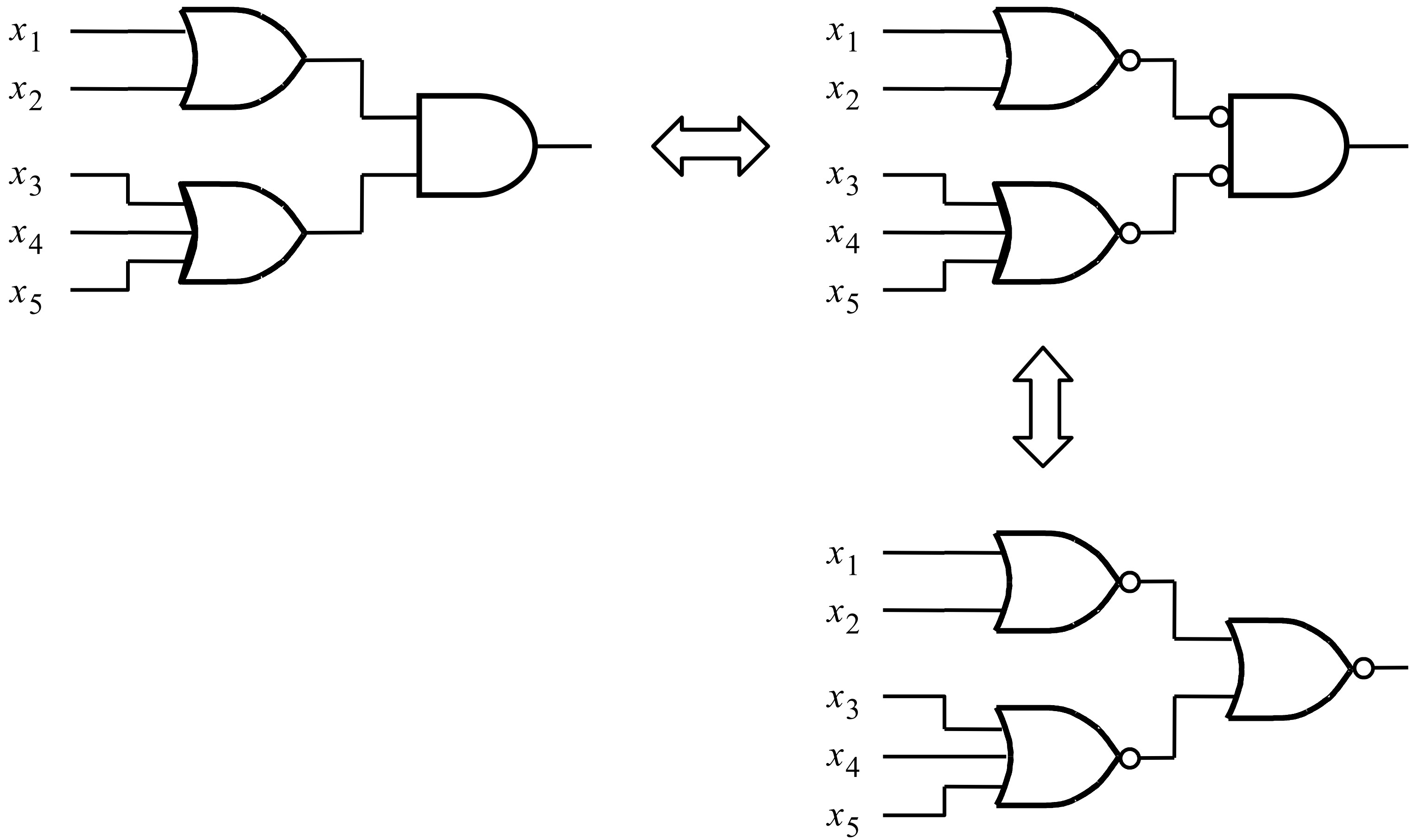
Product-of-sums realization

NAND Network

$(A' + B')$
$\hookrightarrow (AB)'$

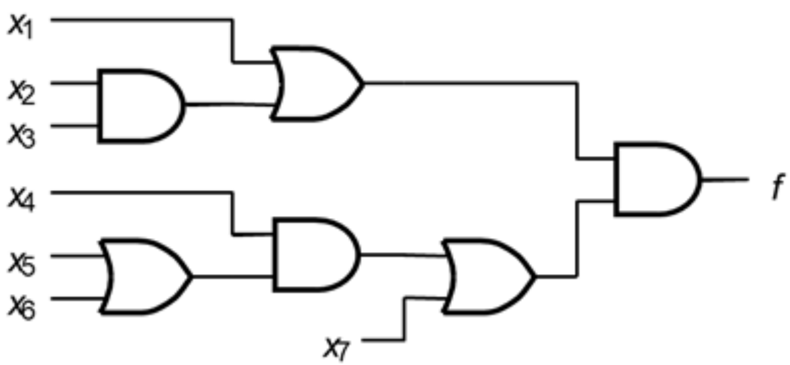


NOR Network

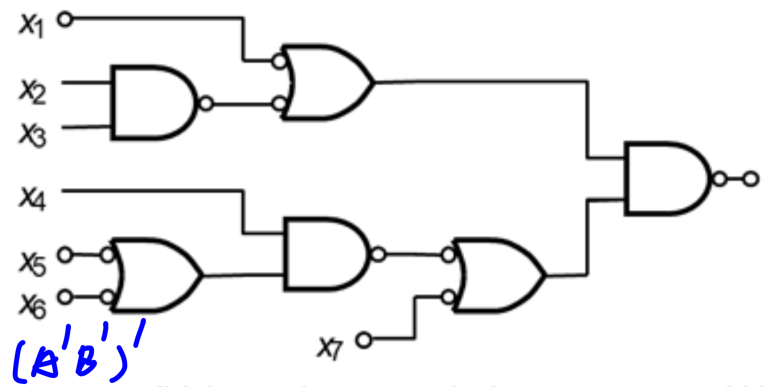


Multilevel NAND Network

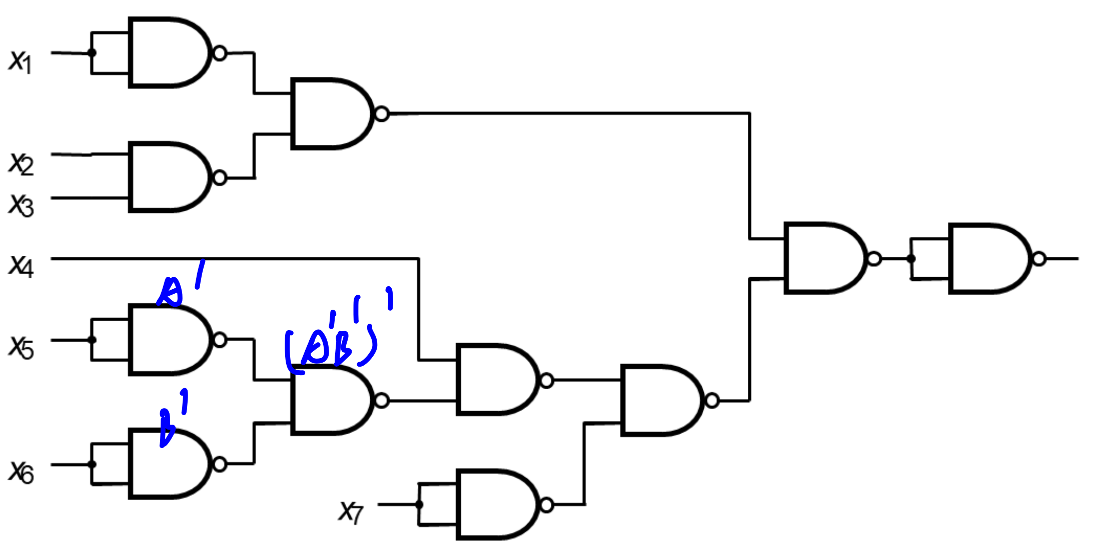
$A' + B' = (AB)'$



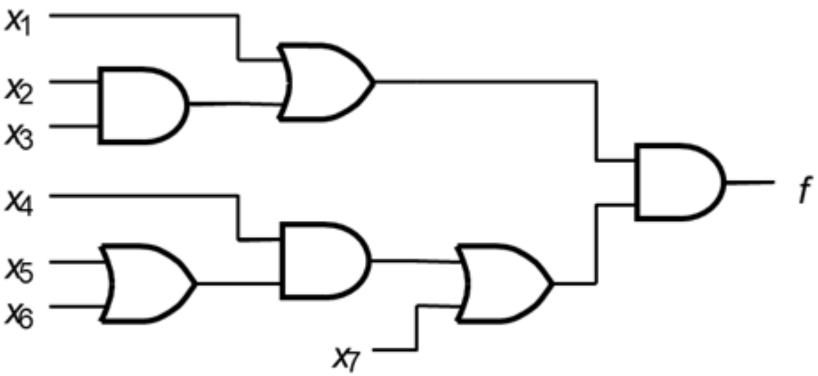
(a) Circuit with AND and OR gates



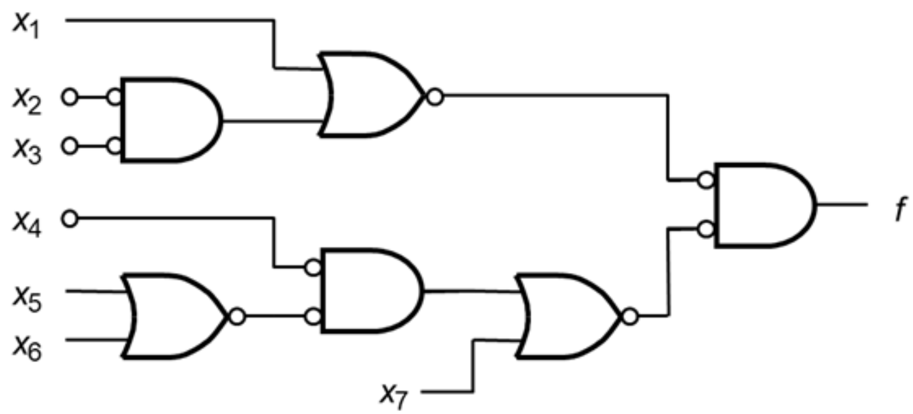
(b) Inversions needed to convert to NANDs



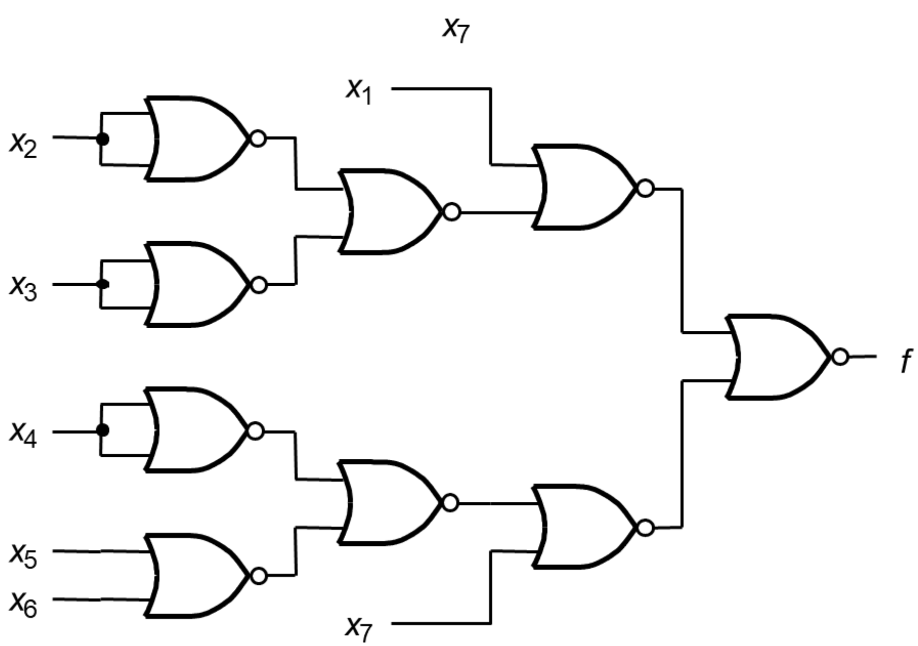
Multilevel NOR Network



(a) Circuit with AND and OR gates



(b) Inversions needed to convert to NORs



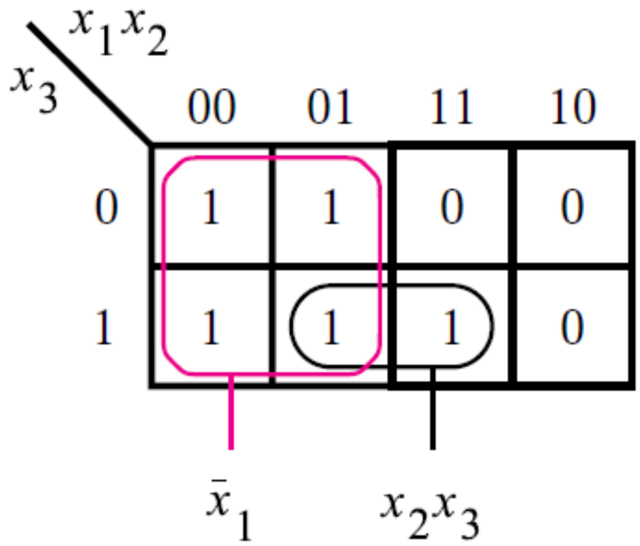
2018.02.18

Strategy for Minimization

- A variable either uncomplemented or complemented is called a literal.
- A product term that indicates when a function is equal to 1 is called an *implicant.*
- An implicant that cannot have any literal deleted and still be a valid implicant is called a *prime implicant*.

- A collection of implicants that accounts for all input combinations in which a function evaluates to 1 is called a *cover*.
- An *essential prime implicant* includes a minterm covered by no other prime.
- *Cost* is number of gates plus number of gate inputs. Assume primary inputs available in both true and complemented form.

An Example



There are 11 implicants : $\bar{x}_1\bar{x}_2\bar{x}_3$ $\bar{x}_1\bar{x}_2x_3$ $\bar{x}_1x_2\bar{x}_3$ $\bar{x}_1x_2x_3$ $x_1x_2x_3$

$\bar{x}_1\bar{x}_2$ $\bar{x}_1x_2$ $\bar{x}_1\bar{x}_3$ $\bar{x}_1x_3$ x_2x_3 $\bar{x}_1$

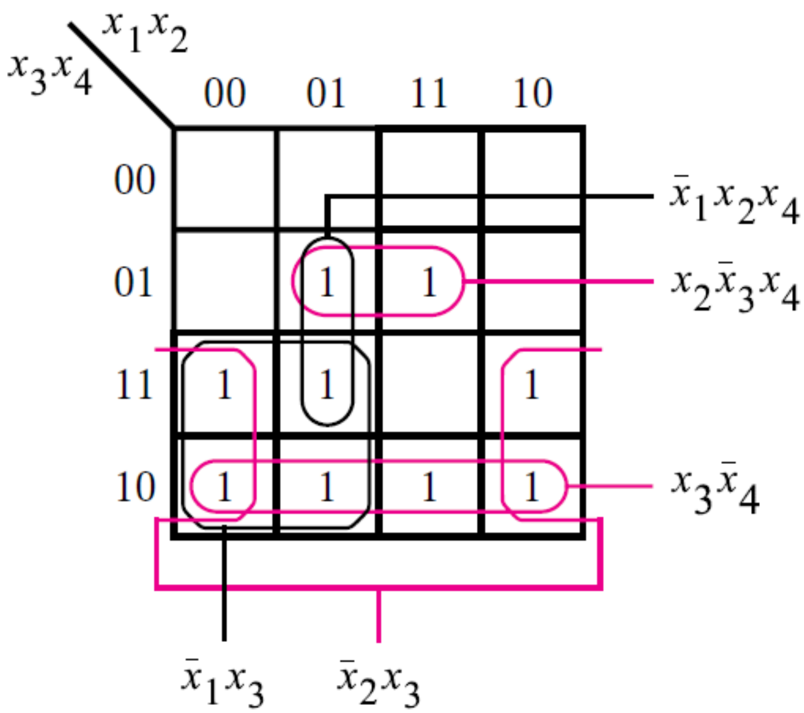
There are two prime implicants : $\bar{x}_1$ x_2x_3

(also essential prime implicants)

Minimization Procedure

- Generate all prime implicants.
- Find all essential prime implicants.
- If essential primes do not form a cover, then select minimal set of non-essential primes.

An Example



There are 5 prime implicants : $\overline{x}_1x_3$ $\overline{x}_2x_3$ $x_3\overline{x}_4$ $\overline{x}_1x_2x_4$ $x_2\overline{x}_3x_4$

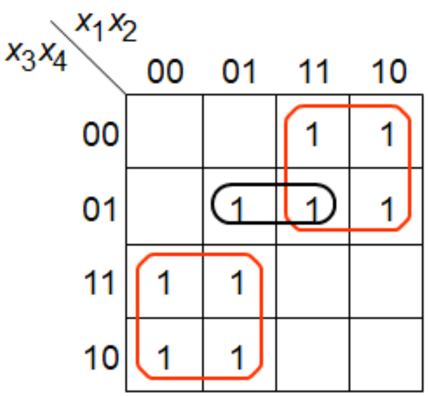
There are 3 essential prime implicants : $\overline{x}_2x_3$ $x_3\overline{x}_4$ $x_2\overline{x}_3x_4$

To form a cover: $\overline{x}_2x_3 + x_3\overline{x}_4 + x_2\overline{x}_3x_4 + ?$

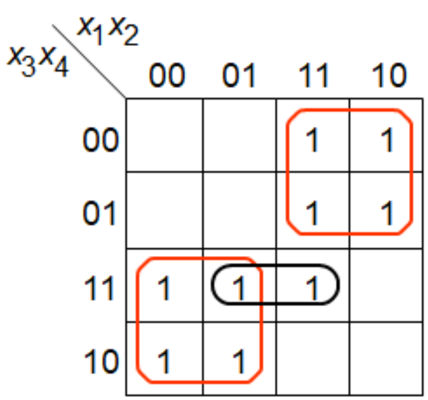
The minimum cost cover: $\overline{x}_2x_3 + x_3\overline{x}_4 + x_2\overline{x}_3x_4 + \overline{x}_1x_3$

Multiple-Output Circuits

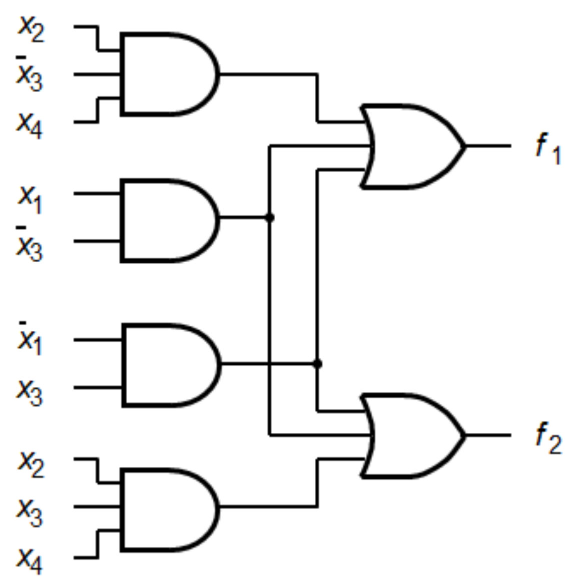
- Necessary to implement multiple functions.
- Circuits can be combined to obtain lower cost solution by sharing some gates.



(a) Function f_1



(b) Function f_2



(c) Combined circuit for f_1 and f_2

A Tabular Method for Minimization

The function is defined as

0000
0100
1000
1010
1011
1100
1101
1111

$$f(x_1, \ldots, x_4) = \sum m(0, 4, 8, 10, 11, 12, 13, 15)$$

■ Generation of Prime Implicants

$$P = \{10x0, 101x, 110x, 1x11, 11x1, xx00\}$$
$$= \{p_1, p_2, p_3, p_4, p_5, p_6\}$$

List 1

0	0 0 0 0	✓
4	0 1 0 0	✓
8	1 0 0 0	✓
10	1 0 1 0	✓
12	1 1 0 0	✓
11	1 0 1 1	✓
13	1 1 0 1	✓
15	1 1 1 1	✓

List 2

找有一個不同的

0,4	0 x 0 0	✓
0,8	x 0 0 0	✓
8,10	1 0 x 0	
4,12	x 1 0 0	✓
8,12	1 x 0 0	✓
10,11	1 0 1 x	
12,13	1 1 0 x	
11,15	1 x 1 1	
13,15	1 1 x 1	

List 3

0,4,8,12	x x 0 0

Figure 4.36

A Tabular Method for Minimization

Determination of a Minimum Cover

To find a minimum-cost cover, we construct a *prime implicant cover table* in which there is a row for each prime implicant and a column for each minterm that must be covered. Then we place check marks to indicate the minterms covered by each prime implicant. Figure 4.37*a* shows the table for the prime implicants derived in Figure 4.36. If there is a single check mark in some column of the cover table, then the prime implicant that covers the minterm of this column is *essential* and it must be included in the final cover. Such is the case with p_6, which is the only prime implicant that covers minterms 0 and 4. The next step is to remove the row(s) corresponding to the essential prime implicants and the column(s) covered by them. Hence we remove p_6 and columns 0, 4, 8, and 12, which leads to the table in Figure 4.37*b*.

Now, we can use the concept of *row dominance* to reduce the cover table. Observe that p_1 covers only minterm 10 while p_2 covers both 10 and 11. We say that p_2 *dominates* p_1. Since the cost of p_2 is the same as the cost of p_1, it is prudent to choose p_2 rather than p_1, so we will remove p_1 from the table. Similarly, p_5 dominates p_3, hence we will remove p_3 from the table. Thus, we obtain the table in Figure 4.37*c*. This table indicates that we must choose p_2 to cover minterm 10 and p_5 to cover minterm 13, which also takes care of covering minterms 11 and 15. Therefore, the final cover is

A Tabular Method for Minimization

Figure 4.37

前面已經化簡 0,4,8,12了

剩下的

Prime implicant	0	4	8	10	11	12	13	15
$p_1 = 1\ 0\ x\ 0$			✓	✓				
$p_2 = 1\ 0\ 1\ x$				✓	✓			
$p_3 = 1\ 1\ 0\ x$						✓	✓	
$p_4 = 1\ x\ 1\ 1$					✓			✓
$p_5 = 1\ 1\ x\ 1$							✓	✓
$p_6 = x\ x\ 0\ 0$	✓	✓	✓			✓		

(a) Initial prime implicant cover table

P_2可 cover P_1

P_5可 cover P_3

Prime implicant	10	11	13	15
p_1	✓			
p_2	✓	✓		
p_3			✓	
p_4		✓		✓
p_5			✓	✓

(b) After the removal of essential prime implicants

刪掉可被cover的

Prime implicant	10	11	13	15
p_2	✓	✓		
p_4		✓		✓
p_5			✓	✓

(c) After the removal of dominated rows

1010
1011

1101
1111

0,4,8,12的

$$C = \{p_2, p_5, p_6\} = \{101x,\ 11x1,\ \boxed{xx00}\}$$

the minimum-cost implementation of the function is

$$f = x_1\bar{x}_2x_3 + x_1x_2x_4 + \bar{x}_3\bar{x}_4$$

101x 11x1 xx00

A Tabular Method Example

Problem: Use the tabular method to derive a minimum-cost SOP expression for the function

$$f(x_1, \ldots, x_4) = \bar{x}_1\bar{x}_3\bar{x}_4 + x_3x_4 + \bar{x}_1\bar{x}_2x_4 + x_1x_2\bar{x}_3x_4$$

assuming that there are also don't-cares defined as $D = \sum(9, 12, 14)$.

Solution: $f(x_1, \ldots, x_4) = \sum m(0, 1, 3, 4, 7, 11, 13, 15) + D(9, 12, 14)$

List 1

0	0 0 0 0	✓
1	0 0 0 1	✓
4	0 1 0 0	✓
3	0 0 1 1	✓
9	1 0 0 1	✓
12	1 1 0 0	✓
7	0 1 1 1	✓
11	1 0 1 1	✓
13	1 1 0 1	✓
14	1 1 1 0	✓
15	1 1 1 1	✓

List 2

0,1	0 0 0 x	
0,4	0 x 0 0	
1,3	0 0 x 1	✓
1,9	x 0 0 1	✓
4,12	x 1 0 0	
3,7	0 x 1 1	✓
3,11	x 0 1 1	✓
9,11	1 0 x 1	✓
9,13	1 x 0 1	✓
12,13	1 1 0 x	✓
12,14	1 1 x 0	✓
7,15	x 1 1 1	✓
11,15	1 x 1 1	✓
13,15	1 1 x 1	✓
14,15	1 1 1 x	✓

List 3

1,3,9,11	x 0 x 1
3,7,11,15	x x 1 1
9,11,13,15	1 x x 1
12,13,14,15	1 1 x x

$$P = \{000x, 0x00, x100, x0x1, xx11, 1xx1, 11xx\}$$
$$= \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$$

A Tabular Method Example

Prime implicant	Minterm 0	1	3	4	7	11	13	15
p_1 = 0 0 0 x	✓	✓						
p_2 = 0 x 0 0	✓			✓				
p_3 = x 1 0 0				✓				
p_4 = x 0 x 1		✓	✓			✓		
p_5 = x x 1 1			✓		✓	✓		✓
p_6 = 1 x x 1						✓	✓	✓
p_7 = 1 1 x x							✓	✓

(a) Initial prime implicant cover table

Prime implicant	Minterm 0	1	4	13
p_1 = 0 0 0 x	✓	✓		
p_2 = 0 x 0 0	✓		✓	
p_4 = x 0 x 1		✓		
p_6 = 1 x x 1				✓

$C = \{p_2, p_4, p_5, p_6\}$

the function is implemented as

$$f = \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_2 x_4 + x_3 x_4 + x_1 x_4$$

(b) After the removal of rows p_3, p_5 and p_7, and columns 3, 7, 11 and 15