Karnaugh maps

Jia Chen
jiac@ucr.edu

Karnaugh map

- A systematic and graphical way to reduce the product or sum terms in an expression.
- Key: apply the uniting property/theorem/law as judiciously as possible

Uniting Theorem	$a\,b + a\,b' = a$	$(a + b)\cdot(a + b')=a$

Application examples:

$$x_1x_2x_3 + x_1x_2x_3' = x_1x_2\,,\ \text{where } a = x_1x_2,\ b=x_3$$

$$(x_1 + x_2 + x_3)(x_1 + x_2 + x_3') = x_1 + x_2\,,\ \text{where } a = x_1 + x_2,\ b=x_3$$

Karnaugh maps for SoP

Truth table

Row number	x_1	x_2	x_3	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Use Boolean algebra:

$f = m_0 + m_2 + m_4 + m_5 + m_6$

$= (m_0 + m_2 + m_4 + m_6)$

$\quad + (m_4 + m_5)$

$= (x_1'x_2'x_3' + x_1'x_2x_3' + x_1x_2'x_3' + x_1x_2x_3')$

$\quad + (x_1x_2'x_3' + x_1x_2'x_3)$

$= x_1'x_3' + x_1x_3' + x_1x_2'$

$= x_3' + x_1x_2'$

$f=1$

	x_1	x_2	x_3
m_0	0	0	0
m_2	0	1	0
m_4	1	0	0
m_6	1	1	0

If $x_3=0, f=1$ regardless of the values of x_1 and x_2

$m_0 + m_2 + m_4 + m_6 = x_3'$

$f=1$

	x_1	x_2	x_3
m_4	1	0	0
m_5	1	0	1

If $x_1=1$ and $x_2=0, f=1$ regardless of the value of x_3

$m_4 + m_5 = x_1x_2'$

How to easily discover groups of minterms for $f=1$ that can be combined into single terms?

Karnaugh map

- An alternative to the truth-table form for representing a function

- A map consists of *cells* corresponding to the *rows* of the truth table

Location of two-variable minterms

x_1	x_2	
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

(a) Truth table

x_2 \ x_1	0	1
0	m_0	m_2
1	m_1	m_3

(b) Karnaugh map

- Advantage: minterms in any two cells that are adjacent, either in the same row or the same column, can be combined.

- Test m_2+m_3 ? $m_2+m_3 = x_1x_2'+x_1x_2 = x_1$

2-variable K-map example

Input		Output	
A	B		
0	0	1	m_0
0	1	1	m_1
1	0	1	m_2
1	1	0	m_3

B \ A	0	1
0	m_0	m_2
1	m_1	m_3

		A'	A
	B \ A	0	1
B'	0	1	1
B	1	1	0

Combining m_0+m_2 = B'

Combining $m_0 + m_1$ = A'

Output = $m_0 + m_1 + m_2$ = A' + B'

Observation

Output = A’ + B’

		A’	A
	B \ A	0	1
B’	0	1	1
B	1	1	0

- In the two cells (A=0,**B=0**), & (A=1,**B=0**)
- Result doesn’t include A as A has two different values 0 and 1
- Result only includes B as B is always 0
- As B=0 in both cells, result is **B’**

- In the two cells (**A=0**,B=0), & (**A=0**,B=1)
- Result doesn’t include B as B has two different values 0 and 1
- Result only includes A as A is always 0
- As A=0 in both cells, result is **A’**

Conclusion:

(1) A resulting term from a combination only contains the variables having constant values

(2) If the variable (denoted as X)=0 in all cells combined, X’ shows up in the resulting term; otherwise X shows up in the resulting term.

Steps to find the simplest SoP

Step 1) Create a 2 dimensional truth table with input variables on each dimension, and adjacent column(j)/row(i) only change one bit in the variable.

Step 2) Fill each (i , j) with the corresponding output result in the truth table

Step 3) Combine neighboring 2, 4, 8, 16 , …, 2^n Minterms to obtain a SINGLE product term

- Therefore, in a K map, we can only circle 2, 4, 8, 16 , …, 2^n adjacent cells to obtain a single term!
- How to get a SINGLE product term (see next slide)

Step 4) Find the “minimum cover” that covers all 1s in the map

Step 5) OR the product terms in the "minimum cover"

How to get a SINGLE product term (Step 3)?

- A product terms include only those variables having the same value for all cells in the group represented by this term
- If the variable is 1 in the group, it appears uncomplemented (e.g., X)
- If the variable is 0 in the group, it appears complemented (e.g., X’)

Strategy for SoP simplification

- Intuitive strategy: find as few as possible for the number of groups & as large as possible for the number of cells with 1s for each group

- Each group of 1s has to comprise cells that can be represented by a single product term

- The larger the group of 1s, the fewer the number of variables in the corresponding product term

Practicing 2-variable K-map

- What's the simplified function of the given K-map?
  A. A'
  B. A'B
  C. AB'
  D. B
  E. A

B \ A	0	1
0	0	0
1	1	1

Practicing 2-variable K-map

- What's the simplified function of the given K-map?
  - A. A'
  - B. A'B
  - C. AB'
  - D. B
  - E. A

B \ A	0	1
0	0	0
1	1	1

B (as B has the same value 1 in both cells while A has different values 0/1 in two cells)

3-variable K-map example

- One dimension (row or column) will represent two variables and the other dimension represents one variable
- Adjacent cells should differ by only 1 bit; so we only change one variable in the neighboring columns and rows

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

Solution 1:

		A'B'	A'B	AB	AB'
	C \ AB	0,0	0,1	1,1	1,0
C'	0	1	1	1	1
C	1	1	1	0	0

C'

A'

Output = A' + C'

Observation

Solution 1:

		A'B'	A'B	AB	AB'
	C \ AB	0,0	0,1	1,1	1,0
C'	0	1	1	1	1
C	1	1	1	0	0

C' (only C is always 0 for all the four cells, A=0/1, B=0/1)

A' (only A is always 0 in all the four cells, B=0/1, C=0/1)

$\boxed{\text{Output} = A' + C'}$

Schedule for rows/columns is not unique

- One dimension (row or column) will represent two variables and the other dimension represents one variable
- Adjacent cells should differ by only 1 bit; so we only change one variable in the neighboring columns and rows

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

Solution 2:

		B'C'	B'C	BC	BC'	
	A \ BC	0,0	0,1	1,1	1,0	
A'	0	1	1	1	1	A'
A	1	1	0	0	1	

C'

The most left and the most right columns are adjacent; The most top and the most bottom rows are adjacent.

Schedule for rows/columns is not unique

- One dimension (row or column) will represent two variables and the other dimension represents one variable
- Adjacent cells should differ by only 1 bit; so we only change one variable in the neighboring columns and rows

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

Solution 3:

		B'	**B**
	AC \ B	0	1
AC	1,1	0	0
A'C	0,1	1	1
A'C'	0,0	1	1
AC'	1,0	1	1

A'

C'

Output = A' + C'

And more solutions …

Combine/circle adjacent cells

Circle $2=2^1$ adjacent cells

		B'	B
	AC \ B	0	1
AC	1,1	1	0
A'C	0,1	1	0
A'C'	0,0	0	0
AC'	1,0	1	1

		B'	B
	AC \ B	0	1
AC	1,1	1	0
A'C	0,1	0	0
A'C'	0,0	0	0
AC'	1,0	1	1

Combine/circle adjacent cells

Circle $4=2^2$ adjacent cells

	AC \ B	B' 0	B 1
AC	1,1	1	1
A'C	0,1	1	1
A'C'	0,0	0	0
AC'	1,0	0	0

	AC \ B	B' 0	B 1
AC	1,1	1	1
A'C	0,1	0	0
A'C'	0,0	0	0
AC'	1,0	1	1

Combine/circle adjacent cells

Circle $8=2^3$ adjacent cells

		B'	B
	AC \ B	0	1
AC	1,1	1	1
A'C	0,1	1	1
A'C'	0,0	1	1
AC'	1,0	1	1

Output = 1

Special case!

4-variable K-map

- Usually, row represents 2 variables and column represents 2 variables
- Adjacent columns/rows should differ by only 1 bit; so we only change one variable in the neighboring column/row

A’B’C’

		A’B’	A’B	AB	AB’
		00	01	11	10
C’D’	00	1	0	0	0
C’D	01	1	0	0	0
CD	11	0	0	0	0
CD’	10	1	0	0	1

F(A, B, C) = A’B’C’+B’CD’

B’CD’

Observation

		A'B'	A'B	AB	AB'
		00	01	11	10
C'D'	00	1	0	0	0
C'D	01	1	0	0	0
CD	11	0	0	0	0
CD'	10	1	0	0	1

The resulting product is A'B'C' because:

- In the two cells, A=0,B=0,C=0,D=0/1
- Only variables with constant values are kept which are A, B, C
- As A=0, complemented form, aka A'
- Similarly for B, C

The resulting product is **B'CD'** because:

- In the two cells, A=0/1,B=0,C=1,D=0
- Only variables with constant values are kept which are B, C, D
- As B=0, complemented form, aka B'
- As C=1, uncomplemented form, aka C
- As D=0, complemented form, aka D'

F(A, B, C) = A'B'C'+B'CD'

Alternatives of 4-variable K-map

		D'B'	D'B	DB	DB'
		00	01	11	10
C'A'	00				
C'A	01		?		
CA	11				
CA'	10				

Fill the cell with 0 if output = 0
when A = 1, B = 1, C = 0, D = 0
Fill the cell with 1 if output = 1
when A = 1, B = 1, C = 0, D = 0

Alternatives of 4-variable K-map

	A'B'C'	AB'C'	ABC'	A'BC'	A'BC	ABC	AB'C	A'B'C
	000	100	110	010	011	111	101	001
D' 0								
D 1								

BUT, not very convenient!

Karnaugh maps for PoS

Steps to find the simplest PoS

Step 1) Create a 2 dimensional truth table with input variables on each dimension, and adjacent column(j)/row(i) only change one bit in the variable.

Step 2) Fill each (i , j) with the corresponding result in the truth table

Step 3) Combine neighboring 2, 4, 8, 16 , …, 2^n Maxterms to obtain a SINGLE sum term

- Therefore, in a K map, we can only circle 2, 4, 8, 16 , …, 2^n adjacent cells to obtain a single term!
- How to get a SINGLE sum term (see next slide)

Step 4) Find the “minimum cover” that covers all 0 s in the map

Step 5) AND all the sum terms from the "minimum cover"

How to get a SINGLE sum term ?

- A sum terms include only those variables having the same value for all cells in the group represented by this term

- If the variable is 1 in the group, it appears complemented (e.g., X’)

- If the variable is 0 in the group, it appears uncomplemented (e.g., X)

Strategy for POS simplification

- Intuitive strategy: find as few as possible for the number of groups & as large as possible for the number of cells with 0s for each group
- Each group of 0 s has to comprise cells that can be represented by a single sum term
- The larger the group of 0 s, the fewer the number of variables in the corresponding sum term

Simplest PoS: 2-variable K-map

- What's the simplified function of the given K-map?

B \ A	0	1
0	0	0
1	1	1

The resulting sum term is B because:

(1) B has the same value 0 in both cells while A has different values 0/1 in two cells

(2) B = 0, uncomplemented form, aka B

Simplest PoS: 3-variable K-map

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**0**
1	1	1	**0**

	A \ BC	B'C' 0,0	B'C 0,1	BC 1,1	BC' 1,0
A'	0	1	1	0	0
A	1	1	1	0	0

The resulting sum term is B' because:

(1) B has the same value 1 in all the four cells while A, C have different values 0/1

(2) B = 1, complemented form, aka B'

Simplest PoS: 4-variable K-map

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	0	1
10	1	1	1	1

$(x_3 + x_4)$

$(x_2 + x_3)$

$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4)$

The resulting sum is $x_3 + x_4$ because:

- In the four cells, $x_3 = 0$, $x_4 = 0$, $x_1 = 0/1$, $x_2 = 0/1$
- Only variables with constant values are kept which are x_3 , x_4
- As $x_3 = 0$, uncomplemented form, aka x_3
- As $x_4 = 0$, uncomplemented form, aka x_4

The resulting sum is $x_2 + x_3$ because:

- In the four cells, $x_2 = 0$, $x_3 = 0$, $x_1 = 0/1$, $x_4 = 0/1$
- Only variables with constant values are kept which are x_2 , x_3
- As $x_3 = 0$, uncomplemented form, aka x_3
- As $x_2 = 0$, uncomplemented form, aka x_2

The resulting sum is $\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4$ because:

- No neighbors; a single *Maxterm*
- As all the variables are 1, complemented form

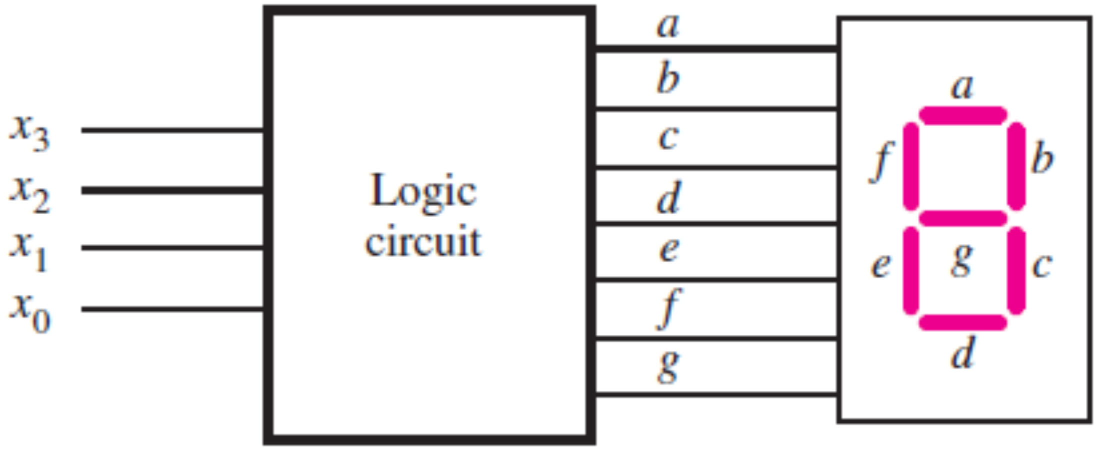
$$F = (x_3 + x_4)(x_2 + x_3)(x_1' + x_2' + x_3' + x_4')$$

Karnaugh maps for Incompletely Specified Functions

Incompletely Specified Functions

- Situations where the output of a function can be either 0 or 1 for a particular combination of inputs
- This is specified by a don’t care in the truth table
- This happens when
  - The input does not occur. e.g. Decimal numbers 0… 9 use 4 bits, so (1,1,1,1) does not occur.
  - The input may happen but we don’t care about the output. E.g. The output driving a seven segment display – we don’t care about illegal inputs (greater than 9)

Recall: Truth table with Don't Cares



(a) Logic circuit and 7-segment display

	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
Don't-care conditions	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0	x	x	x	x	x	x	x
	1	1	0	1	x	x	x	x	x	x	x
	1	1	1	0	x	x	x	x	x	x	x
	1	1	1	1	x	x	x	x	x	x	x

- Each "x" for these valuations is either 1 or 0, whichever is more useful

K-map with Don't Cares

	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
Don't-care conditions	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0	x	x	x	x	x	x	x
	1	1	0	1	x	x	x	x	x	x	x
	1	1	1	0	x	x	x	x	x	x	x
	1	1	1	1	x	x	x	x	x	x	x

Consider the output a

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

- Each "X" for these valuations is either 1 or 0, whichever is more useful

The simplest SoP with Don't Cares

You can treat "X" as either 0 or 1 — depending on which is more advantageous

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

What is the simplest product term including the "1" ?

The simplest SoP with Don't Cares

What is the simplest product term including the "1" ?

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	$\boxed{1}$	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

If we circle one cell

$x_3' x_2' x_1 x_0$

The simplest SoP with Don't Cares

What is the simplest product term including the "1" ?

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

If we circle two cells

$x_3' x_2' x_1$

The simplest SoP with Don't Cares

What is the simplest product term including the "1" ?

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

If we circle four cells

$x_3' x_1$

The simplest SoP with Don't Cares

What is the simplest product term including the "1" ?

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

If we circle eight cells

x_1

The simplest!

The simplest SoP with Don't Cares

What is the simplest product term including the "1" ?

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X(1)	X(1)
10	1	1	X(1)	X(1)

If we circle eight cells

x_1

Treat the four "X"s as "1"

The simplest SoP with Don't Cares

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Green "1"s have been circled

Red "1"s have not been circled

The simplest SoP with Don't Cares

What is the simplest product term including the "1" on the left top corner?

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

The simplest SoP with Don’t Cares

What is the simplest product term including the “1” on the left top corner? (circle as many cells as possible)

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X(1)

We can circle as many as four cells

$x_0' x_2'$

The simplest SoP with Don't Cares

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Green "1"s have been circled

Red "1"s have not been circled

The simplest SoP with Don’t Cares

What is the simplest product term including the “1”

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Green “1”s have been circled

Red “1”s have not been circled

The simplest SoP with Don't Cares

What is the simplest product term including the "1"

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X(1)	X(1)	X
10	1	1	X	X

We can circle as many as four cells

$$x_0 x_2$$

The simplest SoP with Don’t Cares

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Green “1”s have been circled

Red “1” has not been circled

The simplest SoP with Don’t Cares

What is the simplest product term including the “1”

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

The simplest SoP with Don’t Cares

Which of the following circling will lead to the simplest product term including the “1”

(1).

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(2).

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(3).

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(4).

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

The simplest SoP with Don’t Cares

Which of the following circling will lead to the simplest product term including the “1”

(1).

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(2).

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(3).

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(4).

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

x_3

Summary

	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
Don't-care conditions	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0	x	x	x	x	x	x	x
	1	1	0	1	x	x	x	x	x	x	x
	1	1	1	0	x	x	x	x	x	x	x
	1	1	1	1	x	x	x	x	x	x	x

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = x_1 + x_0'x_2' + x_0x_2 + x_3$$