

EE2000 Logic Circuit Design

Minimization of Logic Functions Part 2

Outline

- 3.4 Minimization using Karnaugh map
(continue)
- 3.5 Boolean Functions with Don't-care Cases
- 3.6 Minimization using Quine-McCluskey
Method

3.4 Minimization using Karnaugh Map (continue)

Re-visit This Example Again

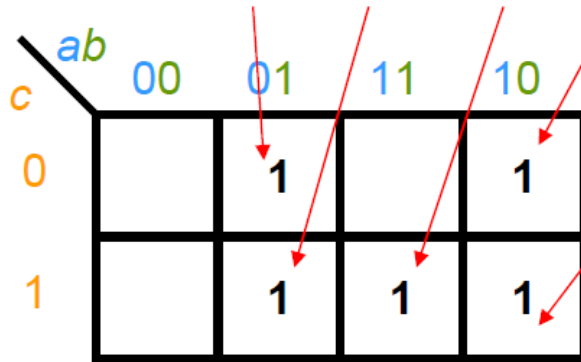
- e.g. Given 4 equivalent Boolean functions f_1 to f_4 expressed in SOP form already (to be proved in next chapter).
 - $f_1(a,b,c) = a'bc' + a'bc + ab'c' + ab'c + abc$ (5 product terms, 15 literals)
 - $f_2(a,b,c) = a'b + ab' + abc$ (3 product terms, 7 literals)
 - $f_3(a,b,c) = a'b + ab' + ac$ (3 product terms, 6 literals)
 - $f_4(a,b,c) = a'b + ab' + bc$ (3 product terms, 6 literals)
 - Both f_3 & f_4 are the minima

- Now we use K-map to simplify the function and verify our answer

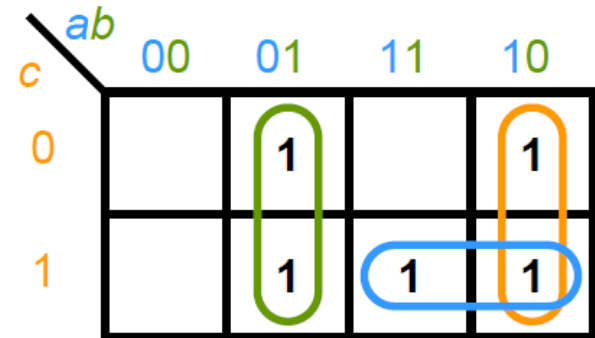
Verify Our Answer

First plot the 1's into K-map

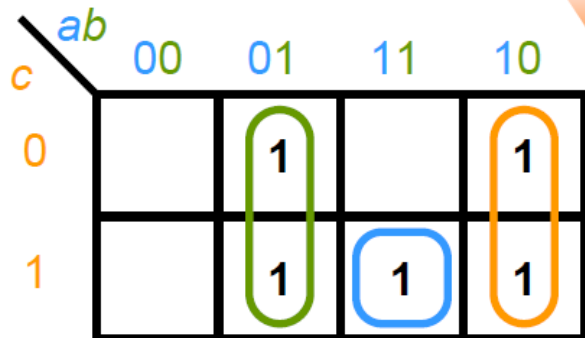
$$f_1(a, b, c) = a'bc' + a'bc + abc + ab'c' + ab'c$$



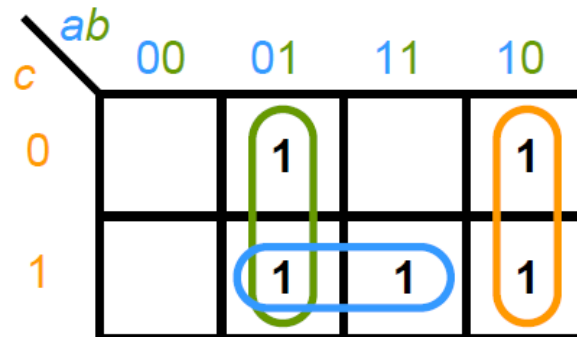
We can see that f_1 to f_4 are equivalent Boolean functions!
And we can obtain the solutions f_3 & f_4 easily



$$f_3(a, b, c) = a'b + ab' + ac$$



$$f_2(a, b, c) = a'b + ab' + abc$$



$$f_4(a, b, c) = a'b + ab' + bc$$

3.4 Minimization using Karnaugh Map (continue)

- Previous examples only show the simplified Boolean functions in **SOP** form
- How to obtain functions in **POS** form
 - [Step 1] Group the 0s to obtain the complement of the f in **SOP** form
 - [Step2] Apply DeMorgan's Theorem to find f' in **POS** form

Example: Find POS

Simplify $f(a, b, c, d) = \Sigma m(0, 1, 2, 5, 8, 9, 10)$ in POS form

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

Fill the 1s and 0s into the map

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

Group the 0s using the same procedure as grouping the 1s

$$f'(a, b, c, d) = ab + cd + bd'$$

$$f(a, b, c, d) = (a' + b')(c' + d')(b' + d)$$

3.5 Boolean functions with Don't-care cases

The output of Boolean functions are **incompletely specified functions**,

- For some input conditions, the outputs are unspecified
- Input condition has no effects to the function
- Output values are defined as **don't-care**
- Don't-care term can be minterm / maxterms
- Don't-care term indicates by an $\times$, d , ϕ or φ

Truth Table with Don't Care

a	b	f
0	0	0
0	1	1
1	0	1
1	1	X

What the table says is:

f is 0 if $(a = 0 \text{ AND } b = 0)$
 f is 1 if $(a = 0 \text{ AND } b = 1)$, or
 $(a = 1 \text{ AND } b = 0)$
 f can be 0 or 1 if $(a = 1 \text{ AND } b = 1)$

$$f(a, b) = \sum m(1, 2) + \sum d(3)$$

a	b	f_1	f_2
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

Both f_1 or f_2 of table on the left are acceptable

Don't-care in K-map

Which solution is better?

$b \backslash a$	0	1
0	0	1
1	1	X

$b \backslash a$	0	1
0	0	1
1	1	0

f_1 implementation

2 groups

$$f = a'b + ab'$$

$b \backslash a$	0	1
0	0	1
1	1	1

f_2 implementation

2 groups

$$f = a + b$$

Procedure for K-map in don't-care cases

1. Must include all 1s in the map (but $\times$ is optional)
2. Select the EPIs first, then remaining PIs
3. Choose the largest PI terms that may contains don't-care terms $\times$

Simplify $f(a, b, c, d) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$

$cd \backslash ab$	00	01	11	10
00	$\times$	0	0	0
01	1	$\times$	0	0
11	1	1	1	1
10	$\times$	0	0	0

$cd \backslash ab$	00	01	11	10
00	$\times$	0	0	0
01	1	$\times$	0	0
11	1	1	1	1
10	$\times$	0	0	0

$$f(a, b, c, d) = a'b'd + cd$$

Is it a good solution?

Other solutions

1. Must include all 1s in the map (but $\times$ is optional)
2. Select the EPIs first, then remaining PIs
3. Choose the largest PI terms that may contains don't-care terms $\times$

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

$$f(a, b, c, d) = a'b' + cd$$

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

$$f(a, b, c, d) = a'd + cd$$

Choose to include those Xs that give largest PIs

3.6 Minimization using Quine-McCluskey (QM) Method

- Developed by W. V. Quine and E. J. McCluskey in 1956
- Functionally identical to Karnaugh map
- More efficient in computer algorithms
- Ease to handle large number of variables

For number of variables is less than 4, we use K-map; otherwise, QM method will be more efficient.

Procedure of QM-method

- Partitioning
- Combining
- Identifying Prime Implicants (PI)
- Generating PI chart
- Reducing chart
- Reporting result

QM-method

- Partitioning

Simplify $f(a, b, c, d) = \sum m(1, 4, 5, 6, 8, 9, 10, 12, 14)$

List all minterms of the function first

Minterms	<i>abcd</i>
m_1	0001
m_4	0100
m_5	0101
m_6	0110
m_8	1000
m_9	1001
m_{10}	1010
m_{12}	1100
m_{14}	1110



Partition them into groups

Minterms	<i>abcd</i>	
m_1	0001	} 1 1s
m_4	0100	
m_8	1000	
m_5	0101	} 2 1s
m_6	0110	
m_9	1001	
m_{10}	1010	
m_{12}	1100	} 3 1s
m_{14}	1110	

QM-method

- Combing

Compare the partitioned terms that follows Gary code property

Combine adjacent group implicants into $(n-1)$ -variable implicants

Mark the changed bit as “-” and give a ✓ to the combined implicants

Minterms	<i>abcd</i>
m_1	0001 ✓
m_4	0100 ✓
m_8	1000 ✓
m_5	0101 ✓
m_6	0110 ✓
m_9	1001 ✓
m_{10}	1010 ✓
m_{12}	1100 ✓
m_{14}	1110 ✓

implicants

Minterms	<i>abcd</i>
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0
m_4, m_{12}	-100
m_8, m_9	100-
m_8, m_{10}	10-0
m_8, m_{12}	1-00
m_6, m_{14}	-110
m_{10}, m_{14}	1-10
m_{12}, m_{14}	11-0

implicants

0001
0101

QM-method

- Combing

Further compare the new partitioned terms that follows Gary

Further combine adjacent group
implicants into $(n-2)$ -variable implicants

Minterms	<i>abcd</i>
m_1, m_5	0-01
m_1, m_9	-001
m_4, m_5	010-
m_4, m_6	01-0 ✓
m_4, m_{12}	-100 ✓
m_8, m_9	100-
m_8, m_{10}	10-0 ✓
m_8, m_{12}	1-00 ✓
m_6, m_{14}	-110 ✓
m_{10}, m_{14}	1-10 ✓
m_{12}, m_{14}	11-0 ✓

Again, mark the changed bit as “-” and
give a ✓ to the combined implicants

Minterms	<i>abcd</i>
m_4, m_6, m_{12}, m_{14}	-1-0
$m_8, m_{10}, m_{12}, m_{14}$	1--0

} No more combination

m_4, m_6 : 01-0
 m_{12}, m_{14} : 11-0

m_4, m_{12} : -100
 m_6, m_{14} : -110

QM-method

- Identifying Prime Implicants (PI)

Minterms	<i>abcd</i>	Minterms	<i>abcd</i>	Minterms	<i>abcd</i>
m_1	0001 ✓	m_1, m_5	0-01 PI_3	m_4, m_6, m_{12}, m_{14}	-1-0 PI_1
m_4	0100 ✓	m_1, m_9	-001 PI_4	$m_8, m_{10}, m_{12}, m_{14}$	1--0 PI_2
m_8	1000 ✓	m_4, m_5	010- PI_5	The remaining unmarked implicants are prime implicants. Label them as PI_i (label the rightmost column first)	
m_5	0101 ✓	m_4, m_6	01-0 ✓		
m_6	0110 ✓	m_4, m_{12}	-100 ✓		
m_9	1001 ✓	m_8, m_9	100- PI_6		
m_{10}	1010 ✓	m_8, m_{10}	10-0 ✓		
m_{12}	1100 ✓	m_8, m_{12}	1-00 ✓		
m_{14}	1110 ✓	m_6, m_{14}	-110 ✓		
		m_{10}, m_{14}	1-10 ✓		
		m_{12}, m_{14}	11-0 ✓		

QM-method

- Generating PI chart

Mark an X in the corresponding columns if this PI covers that minterms

e.g. PI_1 involves m_4, m_6, m_{12}, m_{14}

Essential
prime
implicants ✓
✓

PI	Numeric	1	4	5	6	8	9	10	12	14
PI_1	-1-0		x		x				x	x
PI_2	1--0					x		x	x	x
PI_3	0-01	x		x						
PI_4	-001	x					x			
PI_5	010-		x	x						
PI_6	100-					x	x			

These two columns have one X only!

Minterms 6 & 10 are covered by only one prime implicant
i.e. PI_1 and PI_2 are essential prime implicants

QM-method

- Reducing chart

Reduce the chart by removing those rows of essential prime implicants and columns that covered by them

PI	Numeric	1	4	5	6	8	9	10	12	14
✓ ~~PI₁~~	~~-1-0~~				⊗					
✓ ~~PI₂~~	~~1--0~~					×		⊗	×	×
PI ₃	0-01	x		x						
PI ₄	-001	x					x			
PI ₅	010-		x	x						
PI ₆	100-					x	x			

QM-method

- Reducing chart

The reduced PI chart may not have any essential prime implicants

PI	Numeric	1	5	9
PI_3	0-01	x	x	
PI_4	-001	x		x
PI_5	010-		x	
PI_6	100-			x

How to further reduce the chart?

QM-method

- Reducing chart (covering rule)

Now go back to the reduced PI chart

PI	Numeric	1	5	9
PI ₃	0-01	x	x	
PI ₄	-001	x		x
PI ₅	010-		x	
PI ₆	100-			x



PI	Numeric	1	5	9
PI ₃	0-01	x	x	
PI ₄	-001	x		x

PI₅ is covered by PI₃ (i.e. PI₅ can be removed)

PI₆ is covered by PI₄ (i.e. PI₆ can be removed)

By choosing PI₃ & PI₄, all minterms (m_1, m_5, m_9) have been selected

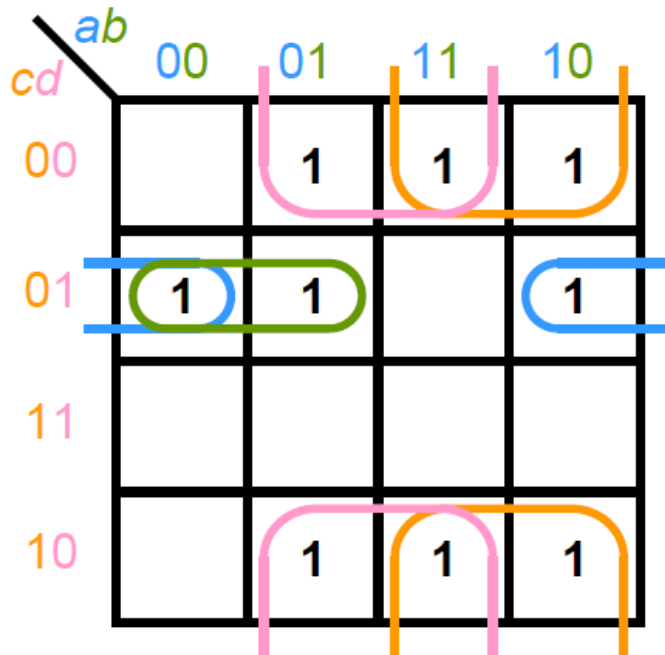
So combining with previous result (all rows with ✓), the minimized function is

$$\begin{aligned}
 f(a, b, c, d) &= \text{PI}_1 + \text{PI}_2 + \text{PI}_3 + \text{PI}_4 \\
 &= -1-0 + 1--0 + 0-01 + -001 \\
 &= bd' + ad' + a'c'd + b'c'd
 \end{aligned}$$

Result!!!

Verify the Result by K-map

- Simplify $f(a, b, c, d) = \sum m(1, 4, 5, 6, 8, 9, 10, 12, 14)$



$$f(a, b, c, d) = bd' + ad' + a'c'd + b'c'd$$

Don't Care Conditions

- How to minimize incompletely specified functions using Q-M method?
 - The first three steps are the same (list, partition and combine)
 - But do **NOT** list the don't care minterms in the PI chart in step 4
- The reason is
 - We set don't care terms to minterms so as to find PIs
 - But omit them as the don't care terms are not essential to be covered

Example: Don't Care

Simplify $f(a, b, c, d) = \Sigma m(4, 8, 9, 10, 12, 15) + \Sigma d(2, 6, 13)$

The don't care terms are listed together Partition the terms as usual

Minterms	<i>abcd</i>
m_2	0010
m_4	0100
m_6	0110
m_8	1000
m_9	1001
m_{10}	1010
m_{12}	1100
m_{13}	1101
m_{15}	1111



Minterms	<i>abcd</i>
m_2	0010
m_4	0100
m_8	1000
m_6	0110
m_9	1001
m_{10}	1010
m_{12}	1100
m_{13}	1101
m_{15}	1111

Example: Don't Care

Combine them to form prime implicants

Minterms	<i>abcd</i>	Minterms	<i>abcd</i>	Minterms	<i>abcd</i>
m_2	0010 ✓	m_2, m_6	0-10 PI_2	m_8, m_9, m_{12}, m_{13}	1-0- PI_1
m_4	0100 ✓	m_2, m_{10}	-010 PI_3		
m_8	1000 ✓	m_4, m_6	01-0 PI_4		
m_6	0110 ✓	m_4, m_{12}	-100 PI_5		
m_9	1001 ✓	m_8, m_9	100- ✓		
m_{10}	1010 ✓	m_8, m_{10}	10-0 PI_6		
m_{12}	1100 ✓	m_8, m_{12}	1-00 ✓		
m_{13}	1101 ✓	m_9, m_{13}	1-01 ✓		
m_{15}	1111 ✓	m_{12}, m_{13}	110- ✓		
		m_{13}, m_{15}	11-1 PI_7		

Example: Don't Care

Note that the don't care terms m_2, m_6, m_{13} are not listed in this chart!

Find the essential prime implicants and reduce the chart as usual

PI	Numeric	4	8	9	10	12	15
PI_1	1-0-		x	(x)		x	
PI_2	0-10						
PI_3	-010				x		
PI_4	01-0	x					
PI_5	-100	x				x	
PI_6	10-0		x		x		
PI_7	11-1						(x)



PI	Numeric	4	10
PI_3	-010		x
PI_4	01-0	x	
PI_5	-100	x	
PI_6	10-0		x



PI	Numeric	4	10
PI_3	-010		x
PI_4	01-0	x	



$$\begin{aligned}
 f(a, b, c, d) &= PI_1 + PI_3 + PI_4 + PI_7 \\
 &= 1-0- + -010 + 01-0 + 11-1 \\
 &= ac' + b'cd' + a'bd' + abd
 \end{aligned}$$

Summary

