第八讲 组合逻辑优化
K-Map and QM Algorithms

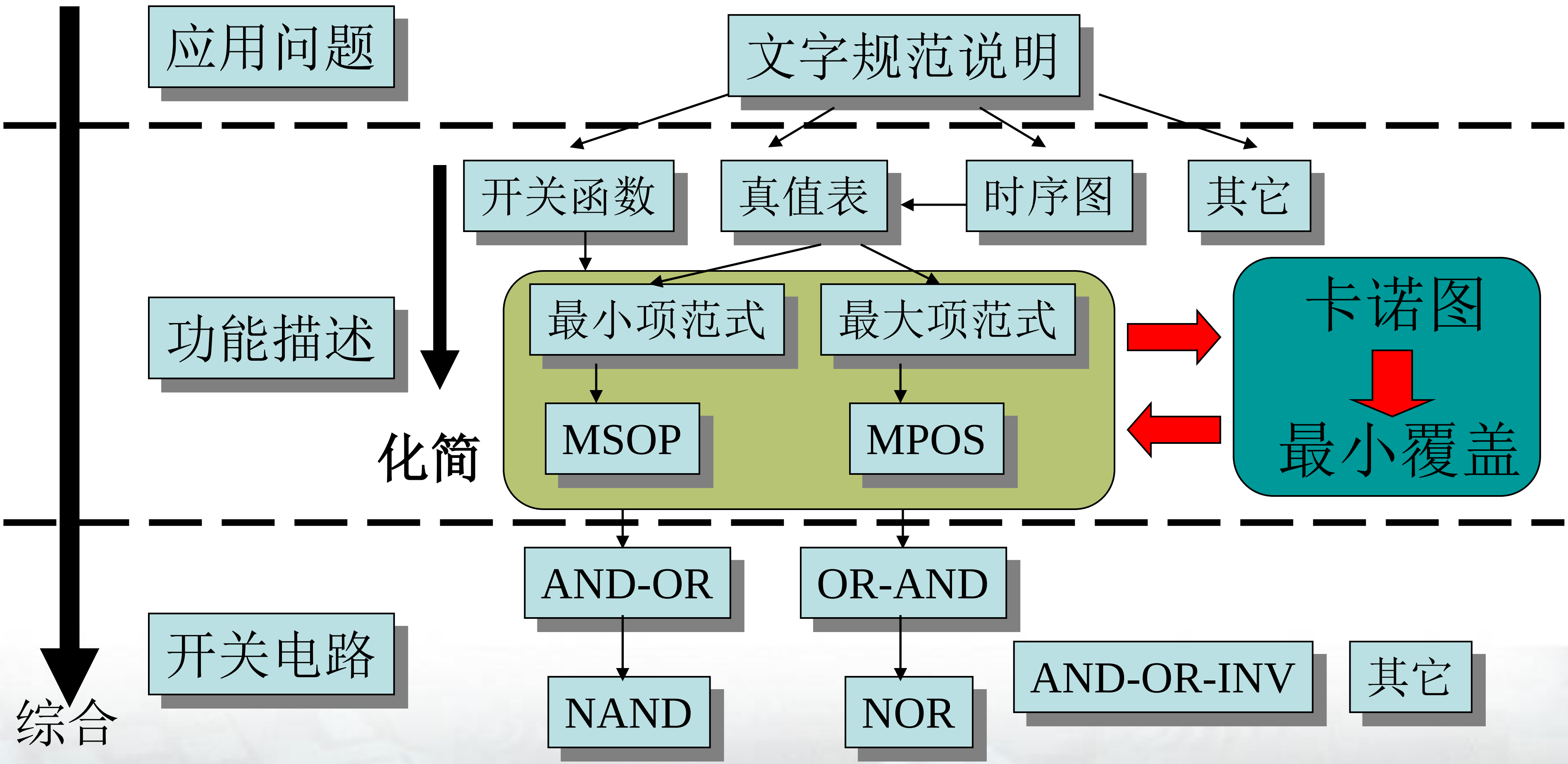
佟冬
tongdong@pku.edu.cn

微处理器研究开发中心（MPRC）
计算机科学技术系

北京大学

电路化简和函数最小化

- 组合电路的分析与综合



布尔表达式的化简

- 合并定理是化简的核心公式: A (B’ + B) = A
- 两级逻辑化简的本质

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

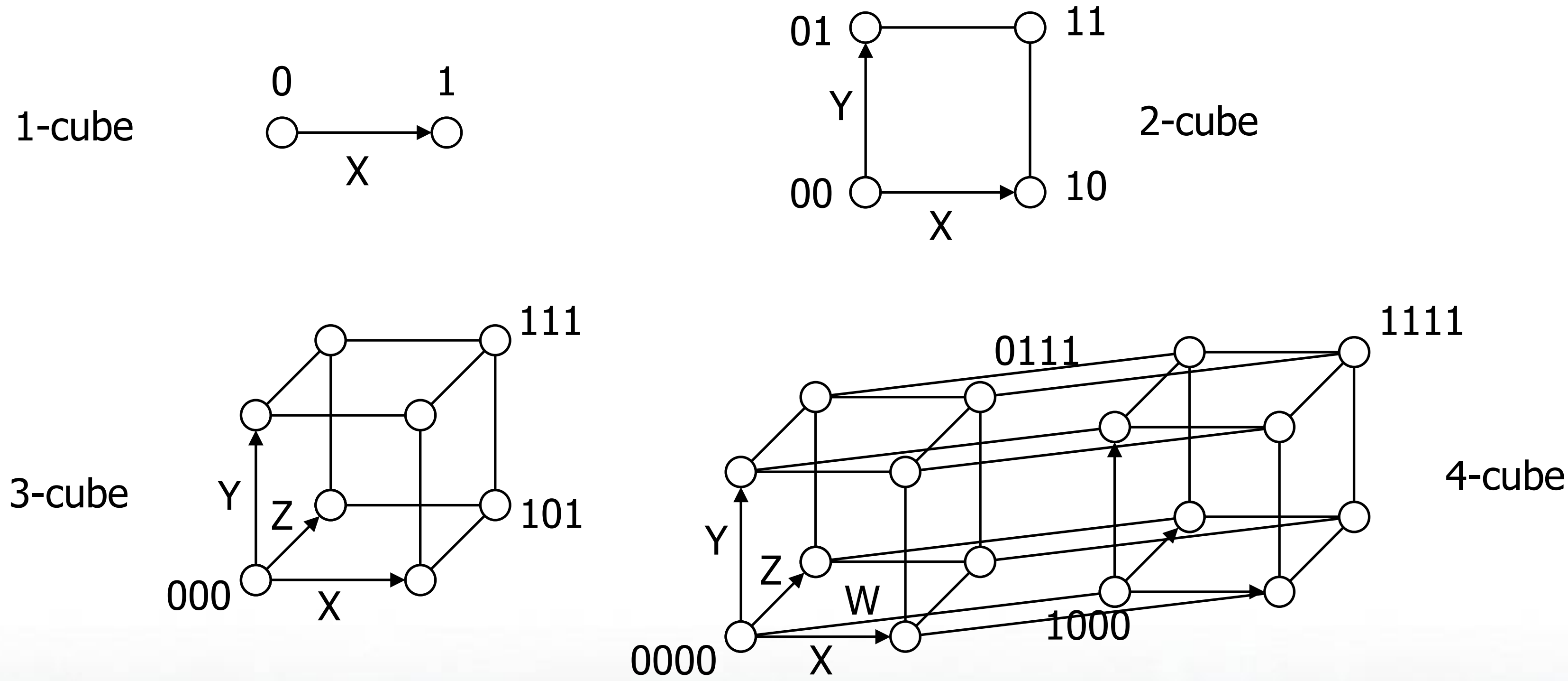
A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

B 在真值表不同的两行有相同的值
– 保留B

A 在这两行有不同的值
– 消除A

布尔立方体

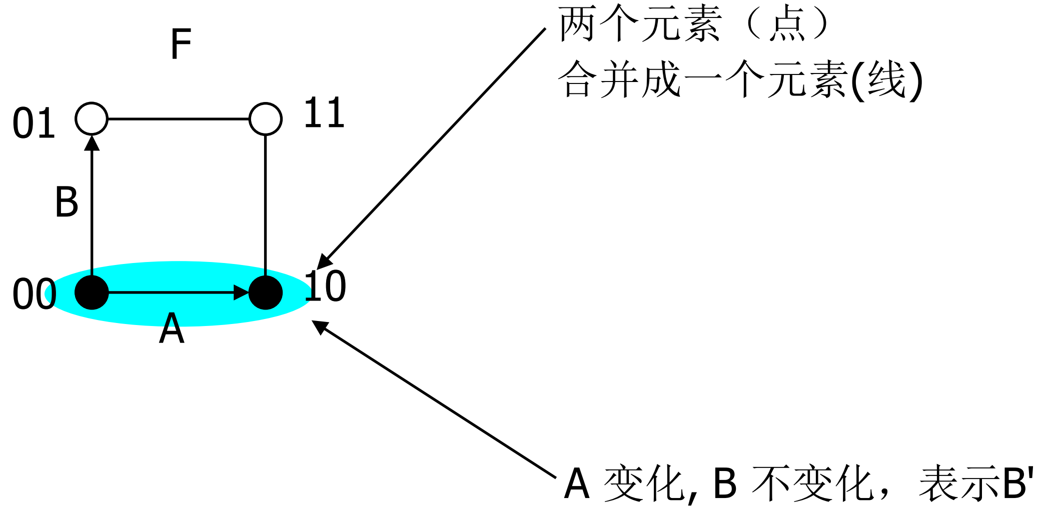
- 使用合并定理的图形化技术
- n个输入变量 = n维 立方体



真值表到布尔立方体的转换

- 合并定理将立方体的两个元素合并成一个元素
- Example:

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0



黑点：开状态集合

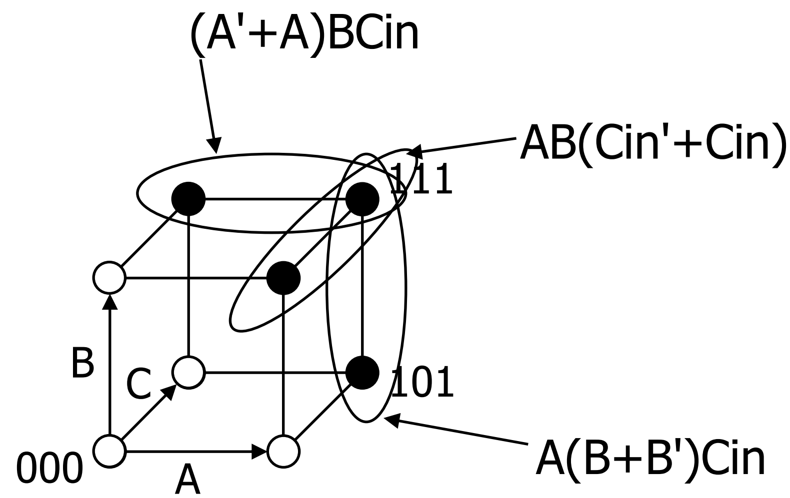
白点：关状态集合

X点：无关项集合

三变量例子

- 二进制全加器进位产生逻辑

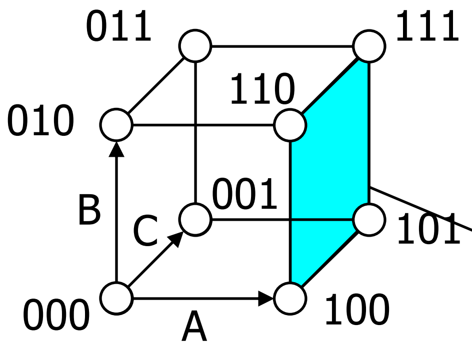
A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Cout = BCin+AB+ACin

更高维度的例子

- 维度大于2的立方体的子立方体



$F(A,B,C) = \Sigma m(4,5,6,7)$

开状态集形成一个正方形
例如，维度为2的立方体

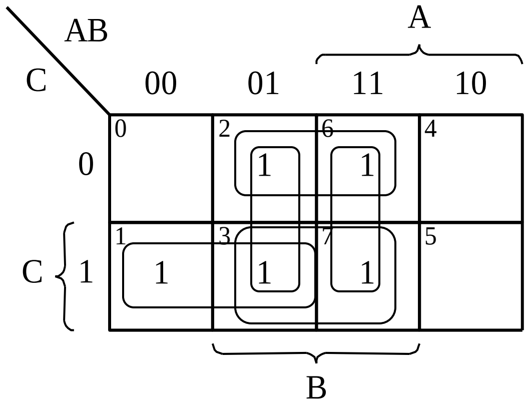
A 是 1（真）不变化
B 和 C 变化

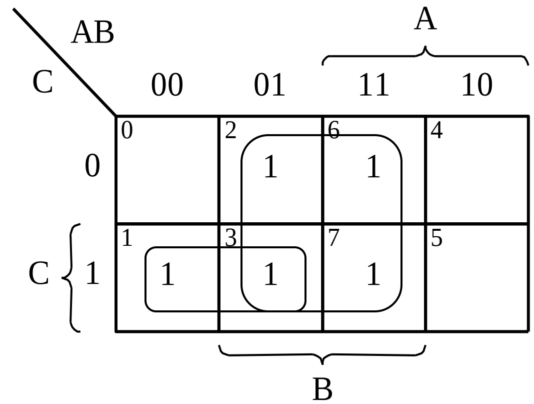
子立方体代表A

卡诺图化简：质蕴涵项和覆盖(SOP)

- 蕴涵项(*implicant*)是一个积项，它能覆盖一个函数的一些最小项。
- 质蕴涵项 (*prime implicant*) 是一个不被该函数的任何其它蕴涵项覆盖的积项。
- 实质蕴涵项(*essential prime implicant*)是一个至少包含一个最小项不能被任何其它质蕴涵项集合覆盖的质蕴涵项。
- 一组蕴涵项被称为函数的一个覆盖(Cover)，如果函数的每个最小项至少被该组内的蕴涵项覆盖。
- 最小覆盖是一个包含最少的质蕴涵项和最少的符号的覆盖。

卡诺图中的蕴涵项





最小项: $\{A'B'C, A'BC', A'BC, ABC', ABC\}$

含有两个最小项的组：$\{A'B, AB, A'C, BC', BC\}$

含有四个最小项的组：$\{B\}$

质蕴涵项: $\{A'C, B\}$

实质蕴含项 = $\{A'C, B\}$

覆盖 = $\{A'C, B\}$

MSOP = $A'C + B$

算法 1 – 产生和选择质蕴涵项

1. 计算出卡诺图中每个最小项的相邻单元个数。

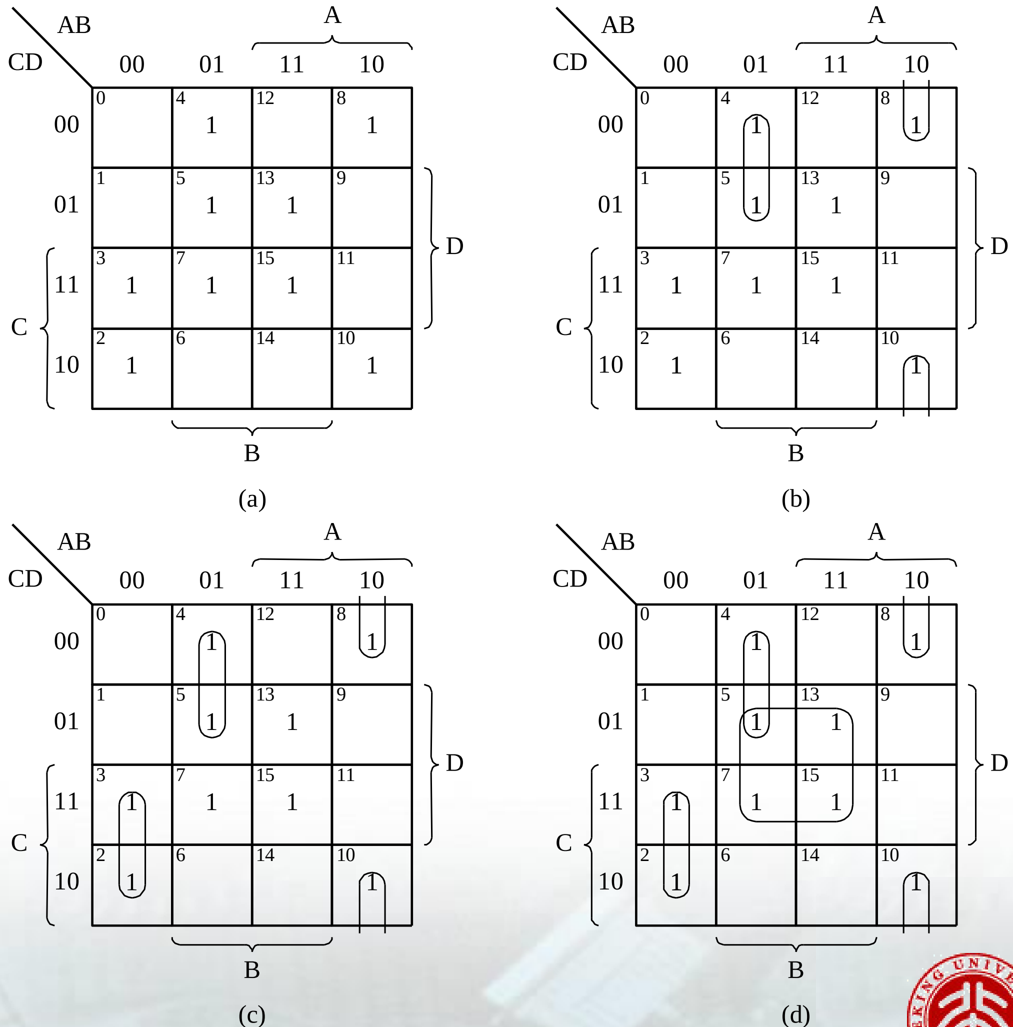
2. 选择没有被包含的具有最小相邻数的最小项。如果存在多个选择可能，任意选择一种。

3. 生成这个最小项的一个质蕴涵项并将它放入覆盖中。如果这个最小项被其它多个质蕴涵项覆盖，选择一个覆盖最多没有被覆盖最小项的质蕴涵项。

4. 重复2-3步，直到覆盖所有的最小项。

算法1的例子

$$f(A,B,C,D) = \sum m(2,3,4,5,7,8,10,13,15)$$

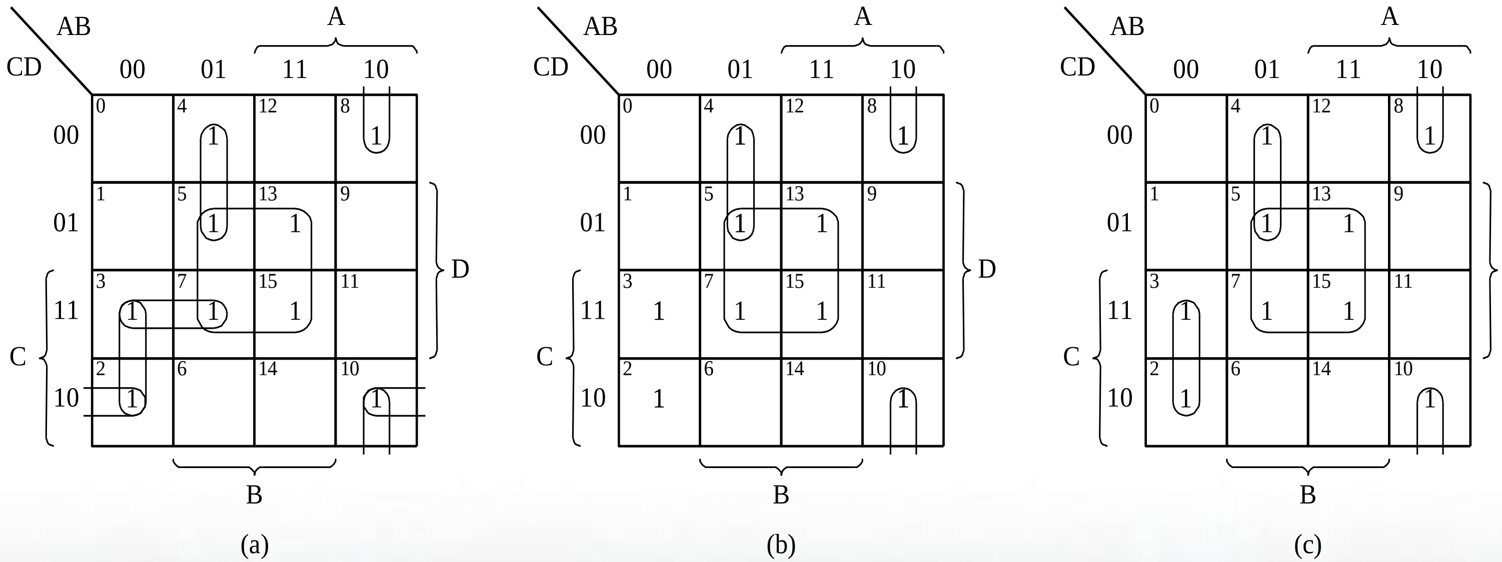


算法 2 --产生和选择质蕴涵项

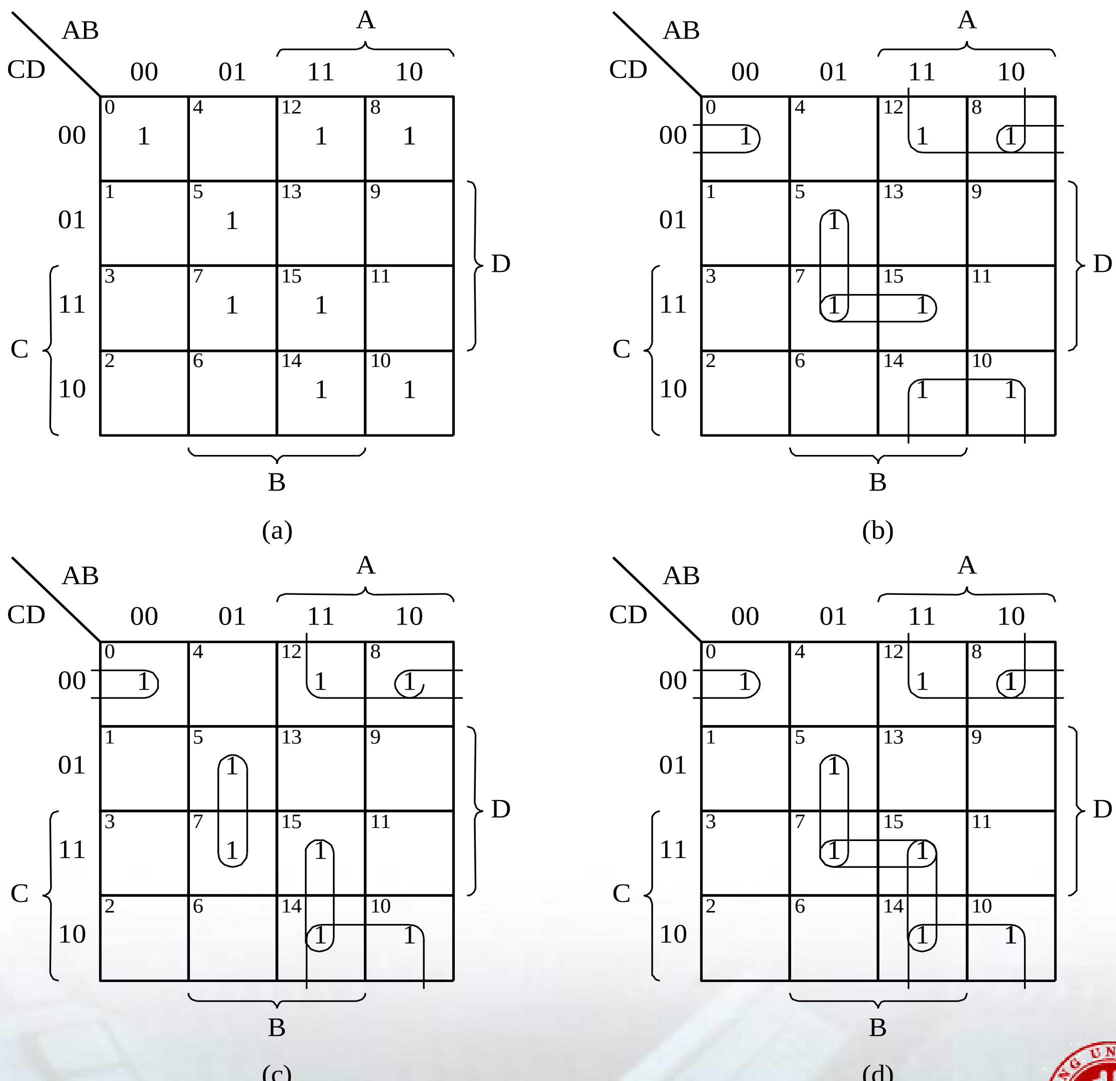
- 算法：用卡诺图求得最小的积之和
  - 第一步：任意选择一个单元
  - 第二步：找到包含这个单元的由1和X组成的最大的组
    - 注意最上/最下行，最左/最右列，以及四角单元是相邻项；
    - 形成一个质蕴涵项（单元数量是2的幂次方）
  - 重复步骤1和2，直到找到所有的质蕴涵项
  - 第三步：重新考察卡诺图中的所有的1
    - 如果只被一个质蕴涵项覆盖，则该质蕴涵项是实质蕴涵项，应该出现在最终的覆盖中；
    - 被实质蕴涵项覆盖的所有的1，不用再被重新考察了
  - 第四步：如果还存在一些1没有被任一实质蕴涵项覆盖
    - 选择最小的质蕴涵项集合覆盖剩余的所有1

算法 2的例子

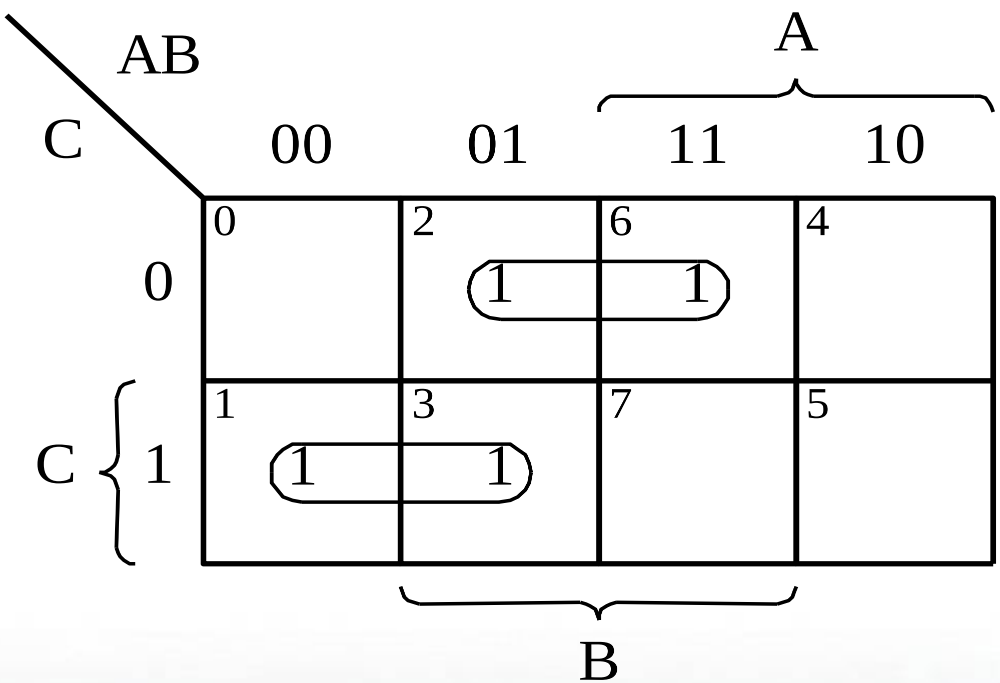
$$f(A,B,C,D) = \sum m(2,3,4,5,7,8,10,13,15)$$



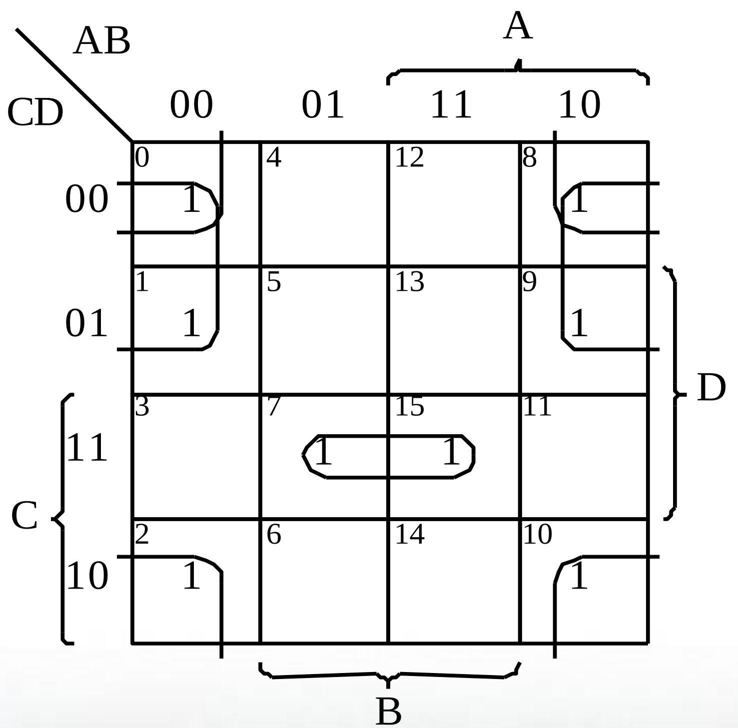
$f(A,B,C,D) = \sum m(0,5,7,8,10,12,14,15)$



$f(A,B,C,D) = \sum m(1,2,3,6) = A'C + BC'$

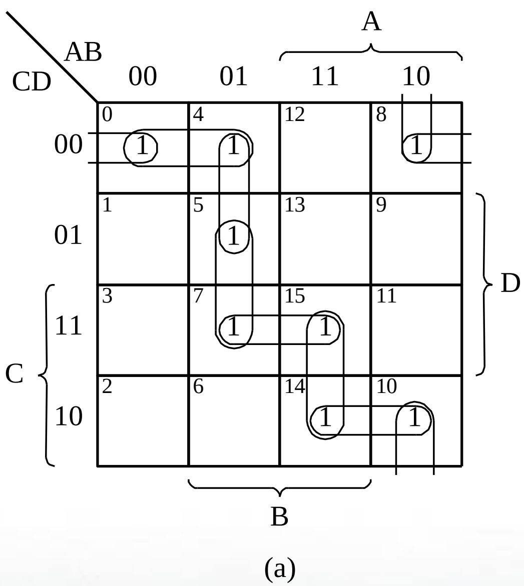


$f(A,B,C,D) = B'D' + B'C' + BCD$



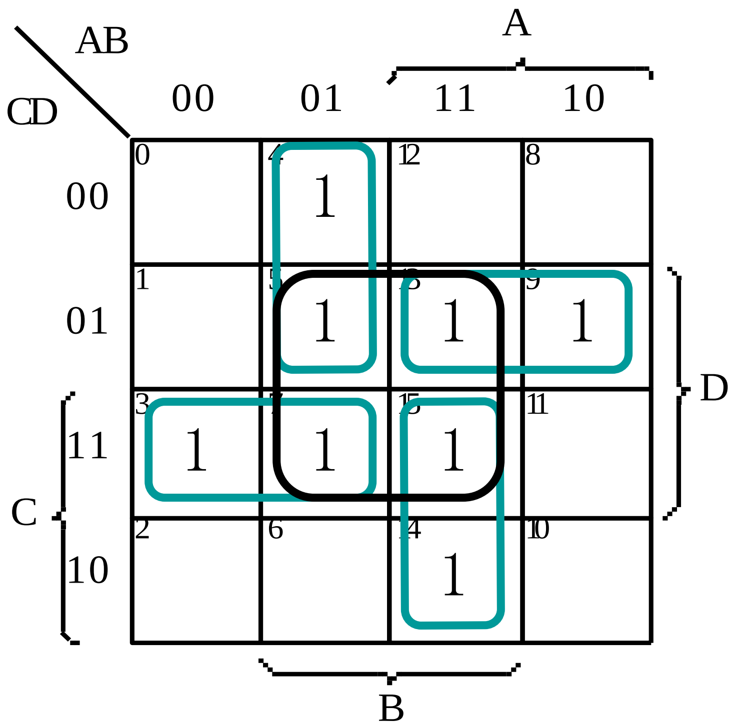
不包含实质蕴涵项的函数例子

$$f(A,B,C,D) = \sum m(0,4,5,7,8,10,14,15)$$



北京大学

一个典型的例子



图中蓝色的组不在最后的覆盖中

质蕴涵项和覆盖(POS)

- 蕴涵项(*implicant*)是一个和项，它能覆盖一个函数的一些最大项。
- 质蕴涵项 (*prime implicant*) 是一个不被该函数的任何其它蕴涵项覆盖的和项。
- 实质蕴涵项(*essential prime implicant*)是一个至少包含一个不能被其它任何质蕴涵项集合所覆盖。
- 一组蕴涵项被称为函数的一个覆盖(Cover)，如果函数的每个最大项至少被该组内的蕴涵项覆盖。
- 最小覆盖是一个包含最少的质蕴涵项和最少的符号的覆盖。

算法 3 – 产生和选择质蕴涵项

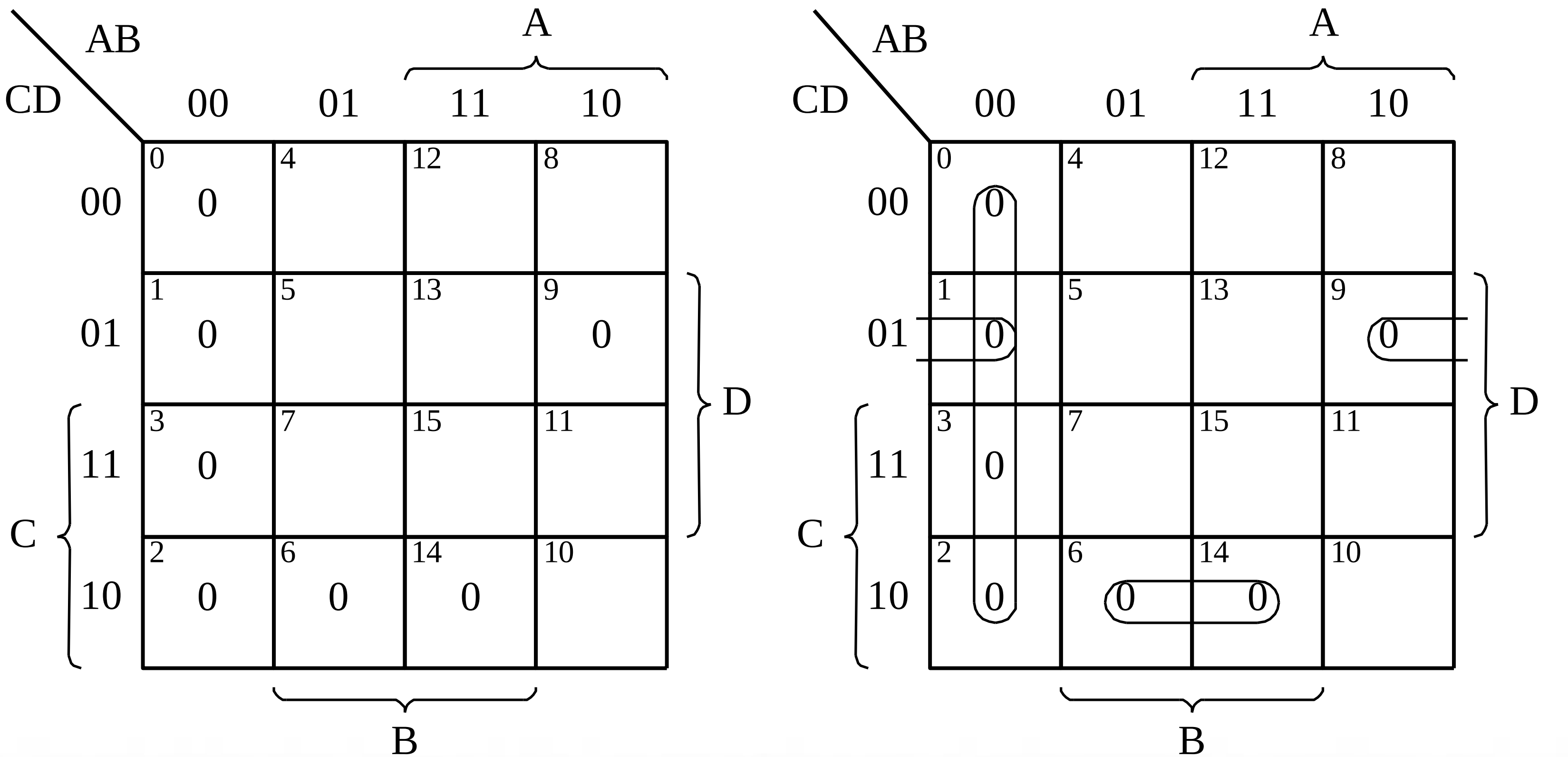
1. 计算出卡诺图中每个最大项的相邻单元个数。

2. 选择没有被包含的具有最小相邻数的最大项。如果存在多个选择可能，任意选择一种。

3. 生成这个最大项的一个质蕴涵项并将它放入覆盖中。如果这个最大项被其它多个质蕴涵项覆盖，选择一个覆盖最多没有被覆盖最大项的质蕴涵项。

4. 重复2-3步，直到覆盖所有的最大项。

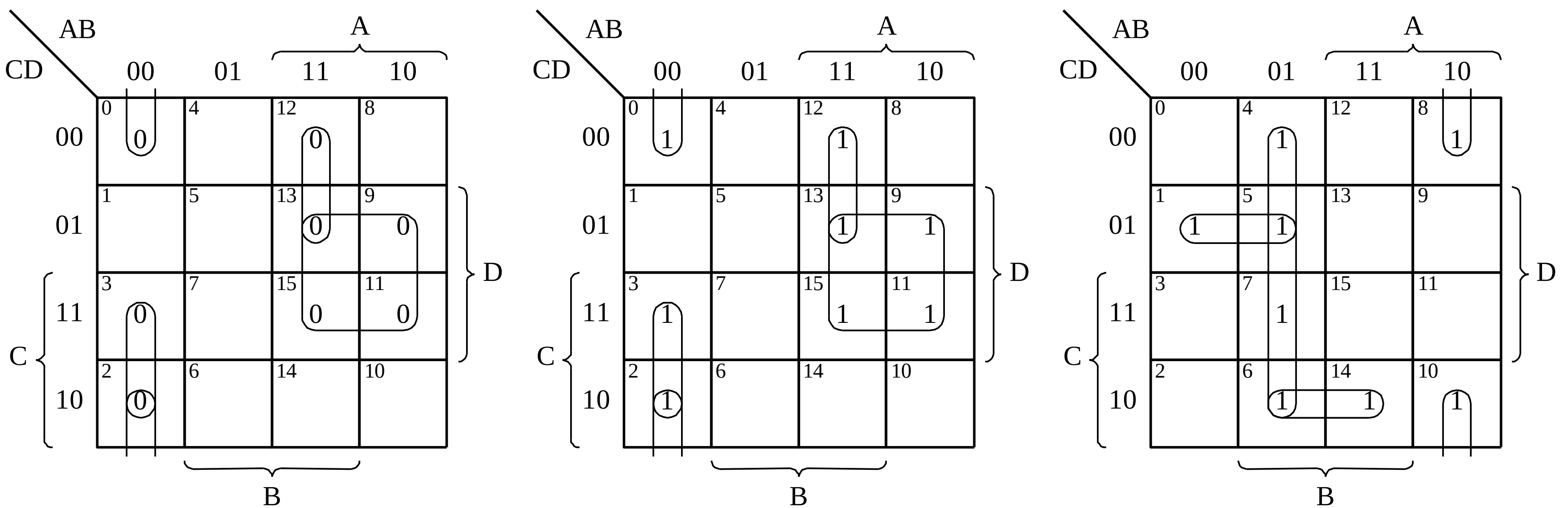
$f(A,B,C,D) = \prod M(0,1,2,3,6,9,14)$



(a) (b)

北京大學

函数的MSOP和MPOS形式化简



(a) (b) (c)

f 的最大项　　f `的最小项　　f 的最小项

回忆：非确定函数

- 开关函数非完全确定的原因
  - 一些确定的输入组合不会产生
  - 一些输出为0或1只对特定的输入组合成立
- 无关(don’t care)最小项：忽略一些最小项
- 无关最大项：忽略一些最大项
- 表示
  - 无关最小项：d_i
  - 无关最大项：D_i
- 例子：
  - 无关最小项：$f(A,B,C)=\Sigma m(0,3,7)+d(4,5)$
  - 无关最大项：$f(A,B,C)=\Pi M(1,2,6)\cdot D(4,5)$

北京大學

非确定函数的化简

- 在化简非确定函数时，可以选择无关项为1或者0的确定值。
- 原则是使卡诺图中划分的单元组大于不包括无关项的卡诺图的单元组。
- 在选择最小覆盖时，忽略那些没有被选择为确定值的非确定项。
- 可以根据无关项是否对函数的化简有帮助来决定无关项的确定值。
- 包含无关项的函数的SOP化简函数和POS化简函数并不一定在形式上互补。
- 产生非确定结果的输入组合，在真实的电路中有确定的结果。
- 非确定性函数卡诺图化简中，质蕴涵项和实质蕴涵项应该至少包含一个确定性最小项。

Quine-McCluskey最小化方法

- Q-M方法相对于卡诺图的优点
  - 直接的、系统的、可计算的方法
  - 可以处理多于六个变量的函数（无法用卡诺图）
  - 可以处理多输出函数
- 算法概况
  - 列出函数的所有最小项
  - 找出所有的质蕴涵项（第一步和第二步）
    - 第一步：将最小项按重量（1的个数）分组。
    - 第二步：穷尽地找出所有的质蕴涵项。
  - 找出最小的质蕴涵项覆盖（第三步和第四步）
    - 构造质蕴涵项图
    - 选择最小数目的质蕴涵项覆盖

例:用Q-M方法找出函数的MSOP

$$f(A,B,C,D) = \prod m(2,4,6,8,9,10,12,13,15)$$

CD \ AB	00	01	11	10
00	0	4: 1	12: 1	8: 1
01	1	5	13: 1	9: 1
11	3	7	15: 1	11
10	2: 1	6: 1	14	10: 1

第一步：列出所有的最小项

最小项	ABCD	
2	0010	
4	0100	组1（一个1）
8	1000	
6	0110	
9	1001	组2（两个1）
10	1010	
12	1100	
13	1101	组3（三个1）
15	1111	组4（四个1）

第二步：生成质蕴涵项

列表1			列表2			列表3		
最小项	ABCD		最小项	ABCD		最小项	ABCD	
2	0 0 1 0	√	2, 6	0 – 1 0	PI_2	8, 9, 12, 13	1 – 0 –	PI_1
4	0 1 0 0	√	2, 10	– 0 1 0	PI_3			
8	1 0 0 0	√	4, 6	0 1 – 0	PI_4			
6	0 1 1 0	√	4, 12	– 1 0 0	PI_5			
9	1 0 0 1	√	8, 9	1 0 0 –	√			
10	1 0 1 0	√	8, 10	1 0 – 0	PI_6			
12	1 1 0 0	√	8, 12	1 – 0 0	√			
13	1 1 0 1	√	9, 13	1 – 0 1	√			
15	1 1 1 1	√	12, 13	1 1 0 –	√			
			13, 15	1 1 – 1	PI_7			

第三步：质蕴涵项图表 实质蕴涵项

				?	?		?	?	?
	2	4	6	8	9	10	12	13	15
* * PI_1				?	?		?	?	
PI_2	?		?						
PI_3	?					?			
PI_4		?	?						
PI_5		?					?		
PI_6				?		?			
* * PI_7								?	?

第四步 简化的质蕴涵项图表 最小覆盖

	? 2	? 4	? 6	? 10
PI_2	?		?	
*PI_3	?			?
*PI_4		?	?	
PI_5		?		
PI_6				?

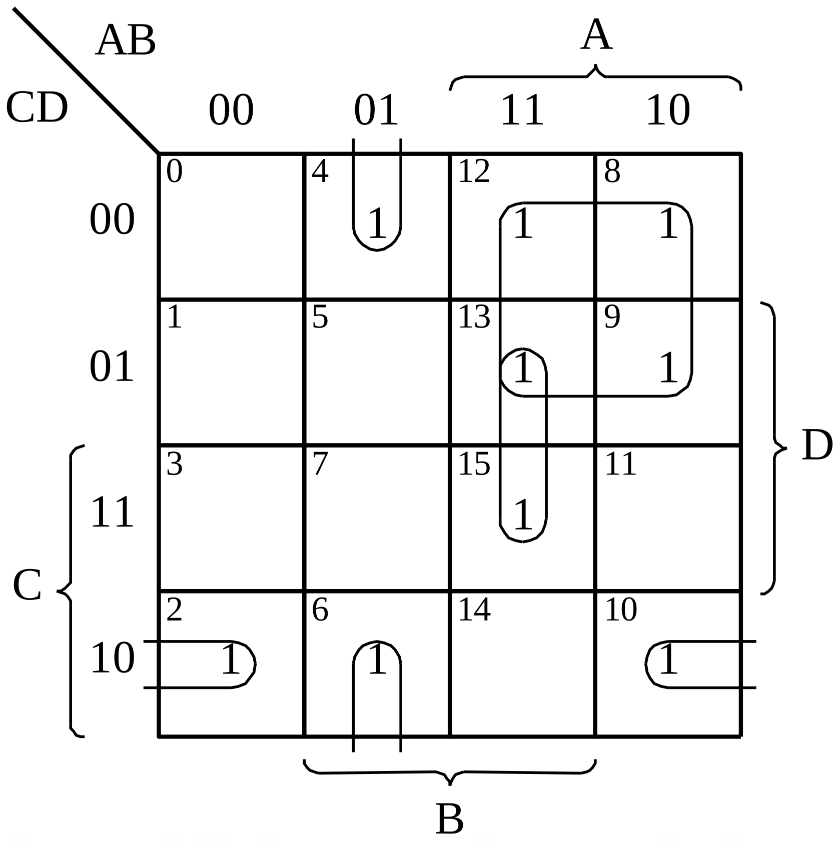
函数的最小化结果

$$f(A,B,C,D) = \mathrm{PI}_1 + \mathrm{PI}_3 + \mathrm{PI}_4 + \mathrm{PI}_7$$

$$= 1\text{-}0\text{-} + \text{-}010 + 01\text{-}0 + 11\text{-}1$$

$$= AC' + B'CD' + A'BD' + ABD$$

结果卡诺图



覆盖过程(Covering Procedure)

- 第一步：标示出所有只被某个PI覆盖的最小项，将这些PI加入覆盖中（实质蕴含项）。
- 第二步：在质蕴涵项图中去掉所有在第一步中标示出的PI覆盖的行，同时去掉被这些PI覆盖的最小项的列。如果结果为空表，过程结束，否则按如下规则进行简化：
  - 规则1：如果一行被其它行覆盖，去掉该行
  - 规则2：如果一列覆盖其它的列，去掉该列
- 第三步：如果一个循环图表，进入第五步。否则再次执行第一部和第二步。
- 第四步：重复执行第五步，任意选择一个质蕴涵项（优先选择包含最小项最多的PI）。直到产生一个空图表或者非循环表。如果表中还有符号，进入第一步。

覆盖的例子

$$f(A,B,C,D) = \sum m(0,1,5,6,7,8,9,10,11,13,14,15)$$

	0	1	5	6	7	8	9	10	11	13	14	15
	?	?		?	?	?	?				?	?
* * PI_1	?	?				?	?					
PI_2		?	?				?			?		
PI_3			?		?					?		?
PI_4						?	?	?	?			
PI_5							?		?	?		?
PI_6								?	?		?	?
* * PI_7				?	?						?	?

简化的图表

	5	10	11	13
PI_2	?			?
PI_3	?			?
PI_4		?	?	
PI_5			?	?
PI_6		?	?	

	?	?
	5	10
* PI_2	?	
*PI_4		?

循环PI图表

- 1 没有实质蕴涵项PI
- 2 没有行或列的覆盖

	?		?			
	1	2	3	4	5	6
*PI_1	?		?			
PI_2		?	?			
PI_3		?				?
PI_4				?		?
PI_5				?	?	
PI_6	?				?	

	2	4	5	6
PI_2	?			
PI_3	?			?
PI_4		?		?
PI_5		?	?	
PI_6			?	

	?	?	?	?
	2	4	5	6
*PI_3	?			?
PI_4		?		?
*PI_5		?	?	

非确定函数的Q-M方法

- 1 在生成质蕴涵项时，同时使用最小项和无关项。
- 2 在生成最小项覆盖时，只需覆盖最小项。
  - 在构造PI图时，不能使用无关项。

- 利用M-Q方法化简下列函数

$$f\,(A, B, C, D, E) = \Sigma m(2, 3, 7, 10, 12, 15, 27) + d(5, 18, 19, 21, 23)$$

质蕴涵项

列表1

最小项	ABCDE	
2	0 0 0 1 0	√
3	0 0 0 1 1	√
5	0 0 1 0 1	√
10	0 1 0 1 0	√
12	0 1 1 0 0	PI_7
18	1 0 0 1 0	√
7	0 0 1 1 1	√
19	1 0 0 1 1	√
21	1 0 1 0 1	√
15	0 1 1 1 1	√
23	1 0 1 1 1	√
27	1 1 0 1 1	√

列表2

最小项	ABCDE	
2, 3	0 0 0 1 –	√
2, 10	0 – 0 1 0	PI_4
2, 18	– 0 0 1 0	√
3, 7	0 0 – 1 1	√
3, 19	– 0 0 1 1	√
5, 7	0 0 1 – 1	√
5, 21	– 0 1 0 1	√
18, 19	1 0 0 1 –	√
7, 15	0 – 1 1 1	PI_5
7, 23	– 0 1 1 1	√
19, 23	1 0 – 1 1	√
19, 27	1 – 0 1 1	PI_6
21, 23	1 0 1 – 1	√

列表3

最小项	ABCDE	
2, 3, 18, 19	– 0 0 1 –	PI_1
3, 7, 19, 23	– 0 – 1 1	PI_2
5, 7, 21, 23	– 0 1 – 1	PI_3

质蕴涵项图表

	?		?	?	?	?	?
	2	3	7	10	12	15	27
PI_1	?	?					
PI_2		?	?				
PI_3			?				
* * PI_4	?			?			
* * PI_5			?			?	
* * PI_6							?
* * PI_7					?		

函数最小化结果

$$f(A,B,C,D,E) = PI_1 + PI_4 + PI_5 + PI_6 + PI_7 \ \textbf{OR}$$

$$= PI_2 + PI_4 + PI_5 + PI_6 + PI_7$$

选择最小覆盖的Petrick算法

- 1 找出所有的质蕴涵项
- 2 画质蕴涵项图表，利用M-Q方法去除所有的实质蕴涵项。
- 3 对于剩余的质蕴涵项表，写出所有可能覆盖的POS表示，包含每一个剩余的最小项列的质蕴涵项积项。
- 4 转化为SOP
- 5 找出包含最少质蕴涵项的积项，得到最小覆盖。

例子

	? 2	? 4	? 6	? 10
PI_2	?		?	
$*PI_3$	?			?
$*PI_4$		?	?	
PI_5		?		
PI_6				?

例子（续）

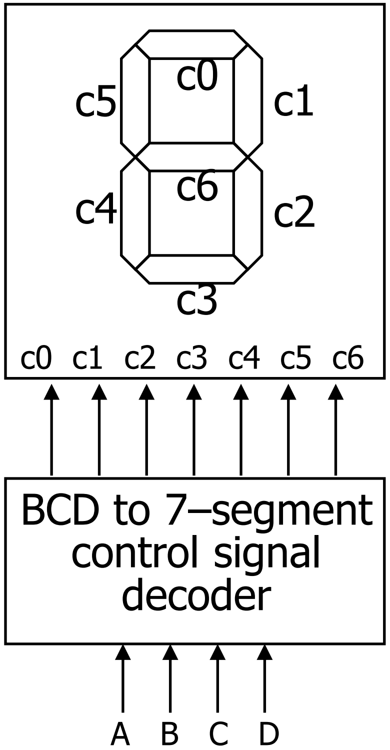
$$C = (PI_2 + PI_3)(PI_4 + PI_5)(PI_2 + PI_4)(PI_3 + PI_6)$$

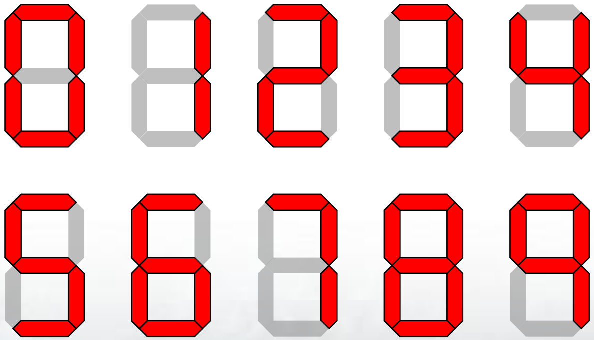
$$= PI_2PI_3\,PI_5 + PI_3PI_4 + PI_2\,PI_4PI_6 + PI_2\,PI_5PI_6$$

Minimal cover = $\{PI_1^*, PI_7^*, PI_3, PI_4\}$

BCD到7段显示控制器

- 理解问题
  - 输入是4位BCD数字 (A, B, C, D)
  - 输出是7段显示的控制信号
    (7个输出 C0 – C6)
- 结构图





问题形式化描述

- 卡诺图
  - 无关项
- 选择实现目标
- 进行实现流程
  - 化简卡诺图

A	B	C	D	C0	C1	C2	C3	C4	C5	C6
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	–	–	–	–	–	–	–	–
1	1	–	–	–	–	–	–	–	–	–

化简卡诺图

- 分别化简形成15个与项

			A	A	
	1	0	X	1	
	0	1	X	1	D
C	1	1	X	X	D
C	1	1	X	X	
		B	B		

			A	A	
	1	1	X	1	
	1	0	X	1	D
C	1	1	X	X	D
C	1	0	X	X	
		B	B		

			A	A	
	1	1	X	1	
	1	1	X	1	D
C	1	1	X	X	D
C	0	1	X	X	
		B	B		

			A	A	
	1	0	X	1	
	0	1	X	0	D
C	1	0	X	X	D
C	1	1	X	X	
		B	B		

			A	A	
	1	0	X	1	
	0	0	X	0	D
C	0	0	X	X	D
C	1	1	X	X	
		B	B		

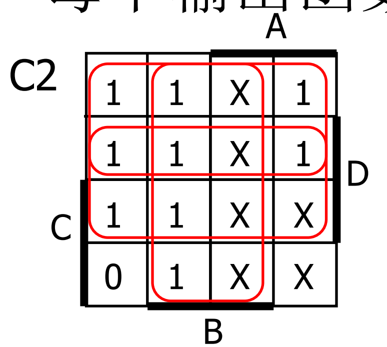
			A	A	
	1	1	X	1	
	0	1	X	1	D
C	0	0	X	X	D
C	0	1	X	X	
		B	B		

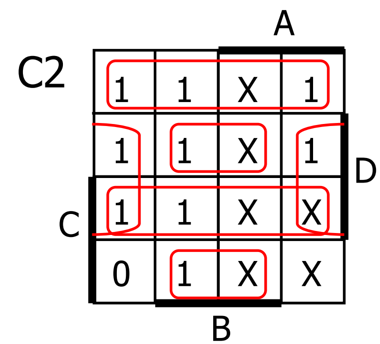
			A	A	
	0	1	X	1	
	0	1	X	1	D
C	1	0	X	X	D
C	1	1	X	X	
		B	B		

C0 = A + B D + C + B' D'
C1 = C' D' + C D + B'
C2 = B + C' + D
C3 = B' D' + C D' + B C' D + B' C
C4 = B' D' + C D'
C5 = A + C' D' + B D' + B C'
C6 = A + C D' + B C' + B' C

多输出函数化简

- 可以更好（多输出函数的QM方法）
  - 9个积项就够了
  - 输出公用积项
  - 每个输出函数不一定最简化





C0 = A + B D + C + B' D'
C1 = C' D' + C D + B'
C2 = B + C' + D
C3 = B' D' + C D' + B C' D + B' C
C4 = B' D' + C D'
C5 = A + C' D' + B D' + B C'
C6 = A + C D' + B C' + B' C

C0 = B C' D + C D + B' D' + B C D' + A
C1 = B' D + C' D' + C D + B' D'
C2 = B' D + B C' D + C' D' + C D + B C D'
C3 = B C' D + B' D + B' D' + B C D'
C4 = B' D' + B C D'
C5 = B C' D + C' D' + A + B C D'
C6 = B' C + B C' + B C D' + A

多输出函数的化简

- 1. 将每一个最小项，标示出函数符号
- 2. 只有两个最小项标示一个或多个共同的函数符号时，才能合并两个最小项。同时，用共同的函数符号标示合并项。
- 3. 在最小项表中的项，所有的符号都应该被处理，一个被处理当且仅当它所有的函数符号出现在一个合并项中。

质蕴涵项

$$f_\alpha(A,B,C,D)=\sum m(0,2,7,10)+d(12,15)$$
$$f_\beta(A,B,C,D)=\sum m(2,4,5)+d(6,7,8,10)$$
$$f_\gamma(A,B,C,D)=\sum m(2,7,8)+d(0,5,13)$$

列表1				列表2				列表3			
最小项	ABCD	标示		最小项	ABCD	标示		最小项	ABCD	标示	
0	0 0 0 0	αγ	√	0, 2	0 0 – 0	αγ	PI_2	4, 5, 6, 7	0 1 – –	β	PI_1
2	0 0 1 0	αβγ	PI_{10}	0, 8	– 0 0 0	γ	PI_3				
4	0 1 0 0	β	√	2, 6	0 – 1 0	β	PI_4				
8	1 0 0 0	βγ	PI_{11}	2, 10	– 0 1 0	αβ	PI_5				
5	0 1 0 1	βγ	√	4, 5	0 1 0 –	β	√				
6	0 1 1 0	β	√	4, 6	0 1 – 0	β	√				
10	1 0 1 0	αβ	√	8, 10	1 0 – 0	β	PI_6				
12	1 1 0 0	α	PI_{12}	5, 7	0 1 – 1	βγ	PI_7				
7	0 1 1 1	αβγ	PI_{13}	5, 13	– 1 0 1	γ	PI_8				
13	1 1 0 1	γ	√	6, 7	0 1 1 –	β	√				
15	1 1 1 1	α	√	7, 15	– 1 1 1	α	PI_9				

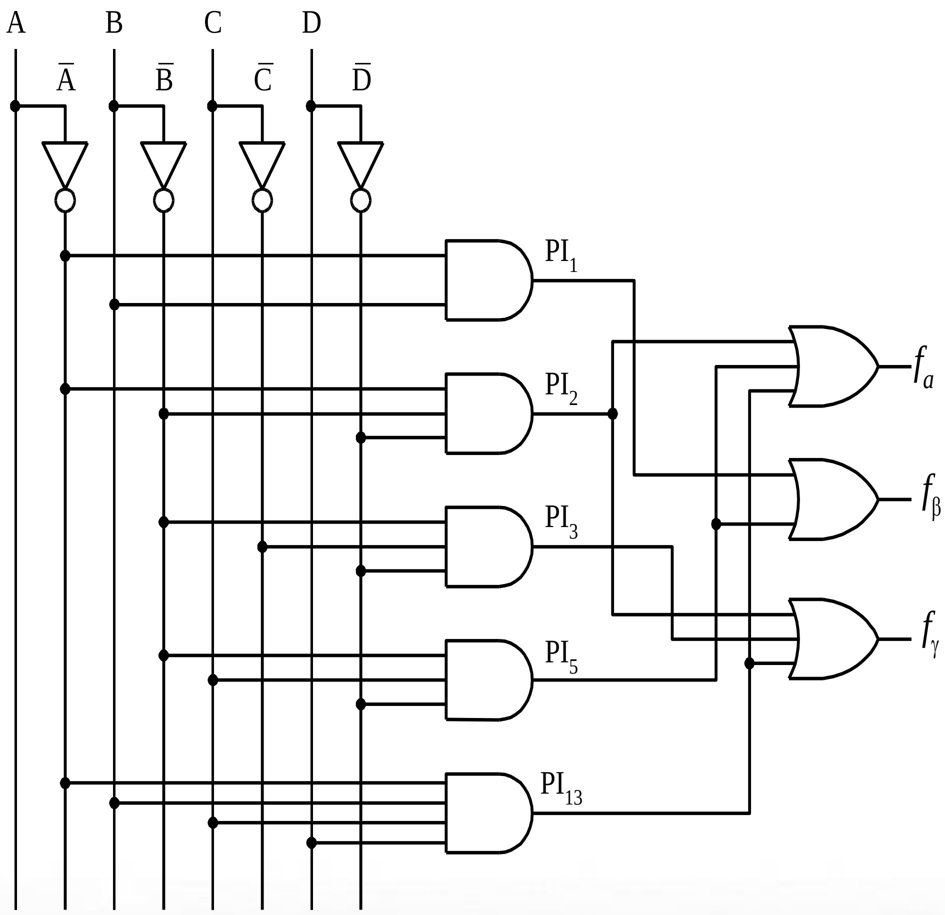
质蕴涵项图表

	$\phi\alpha$				$\phi\beta$			$\phi\gamma$		
	?	?		?	?	?	?	?		
	0	2	7	10	2	4	5	2	7	8
* * PI_1 β						?	?			
* * PI_2 $\alpha\gamma$	?	?						?		
PI_3 γ										?
PI_4 β					?					
* * PI_5 $\alpha\beta$		?		?	?					
PI_6 β										
PI_7 $\beta\gamma$							?		?	
PI_8 γ										
PI_9 α			?							
PI_{10} $\alpha\beta\gamma$		?			?			?		
PI_{11} $\beta\gamma$										?
PI_{12} α										
PI_{13} $\alpha\beta\gamma$			?						?	

化简后的质蕴涵项图表

	$\phi\alpha$	$\phi\gamma$	
	?	?	?
	7	7	8
* PI_3 γ			?
PI_7 $\beta\gamma$		?	
PI_9 α	?		
PI_{11} $\beta\gamma$			?
* PI_{13} $\alpha\beta\gamma$	?	?	

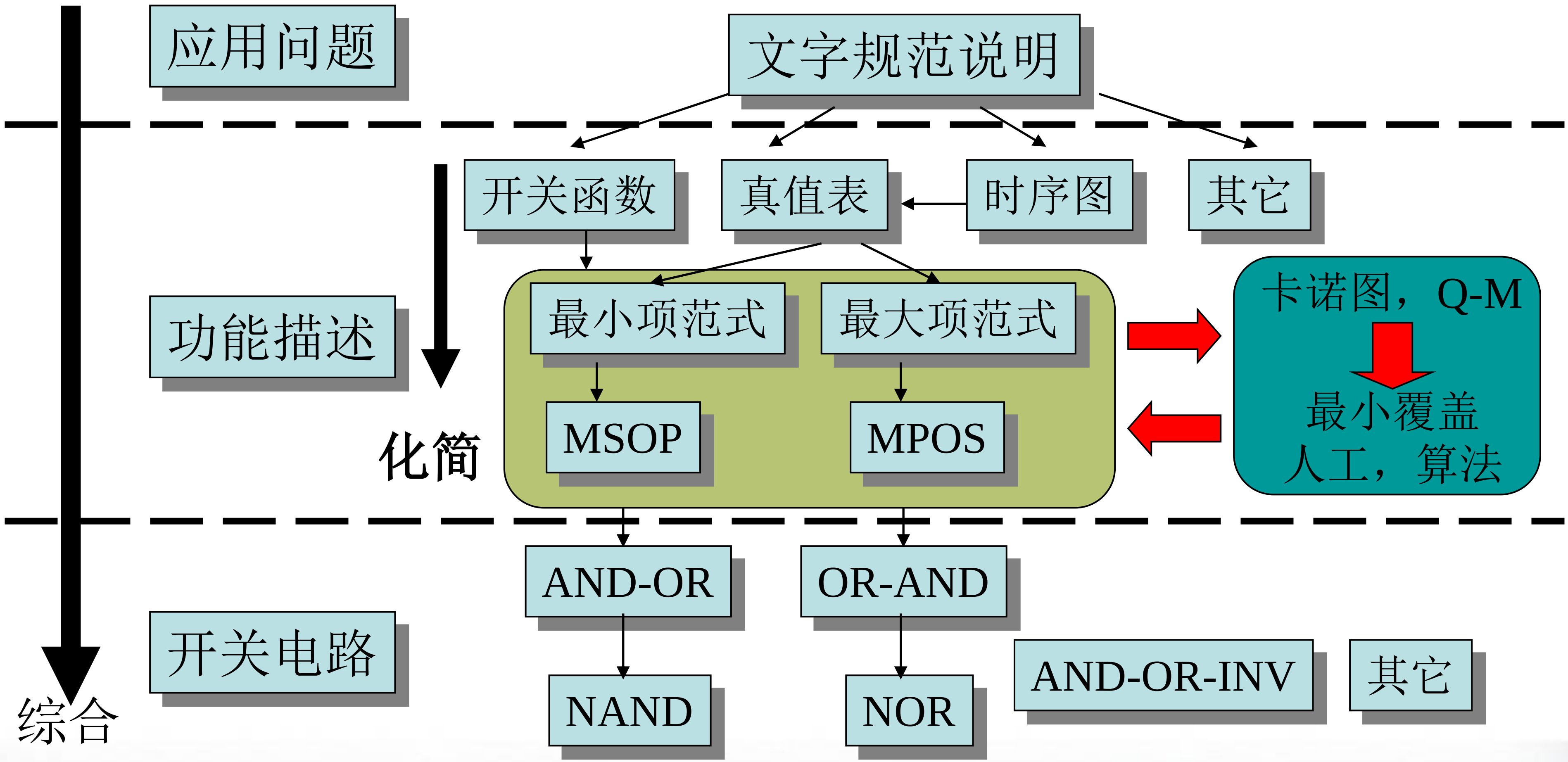
最小的实现



Espresso算法

- 质蕴含项数$3^n/n$；最小覆盖问题是NP完全问题
- Espresso算法
  - 1. 扩展（EXPAND）最大程度扩展蕴含项，被覆盖的蕴含项不再被考察，不再需要找出所有蕴含项。
  - 2. 必要覆盖（IRREDUNDANT COVER），类似QM方法中寻找质蕴含项。
  - 3. 缩减（REDUCE），缩减质蕴含项，减少项数或者每项的变量数。
  - 4. 重复扩展和缩减的过程，直到无法产生比最近结果更好的覆盖。
  - 5. 提速：
    - 实质蕴涵项、反函数和特殊步骤保证覆盖增加

总结



应用问题
文字规范说明
开关函数
真值表
时序图
其它
功能描述
最小项范式
最大项范式
MSOP
MPOS
化简
卡诺图，Q-M
最小覆盖
人工，算法
开关电路
AND-OR
OR-AND
NAND
NOR
AND-OR-INV
其它
综合