

数字逻辑设计

高翠芸

School of Computer Science

gaocuiyun@hit.edu.cn

目 录

- 布尔代数的应用
- 最大项、最小项表达式
- 不完全给定函数

组合逻辑电路的设计方法

已知 —— 设计要求
待求 —— 逻辑图

• 步骤:

1. 根据设计要求确定 —— 真值表
2. 根据真值表 —— 卡诺图(表达式)
3. 化简
4. 按设计要求, 变换逻辑表达式
5. 画出逻辑图



组合逻辑电路的设计方法——续

- 逻辑设计目标

- 实现逻辑功能
- 满足性能指标
- 综合考虑各项因素：
规模、功耗、价格、可靠性、
速度、易实现、易维修、美观等

设计不唯一，最佳设计方案随新技术的不断推出而变化

怎样设计组合逻辑电路？

■方法1：直接转换（简单情况下）

- 将文字描述的功能直接转换为真值表或表达式

■方法2：真值表转换

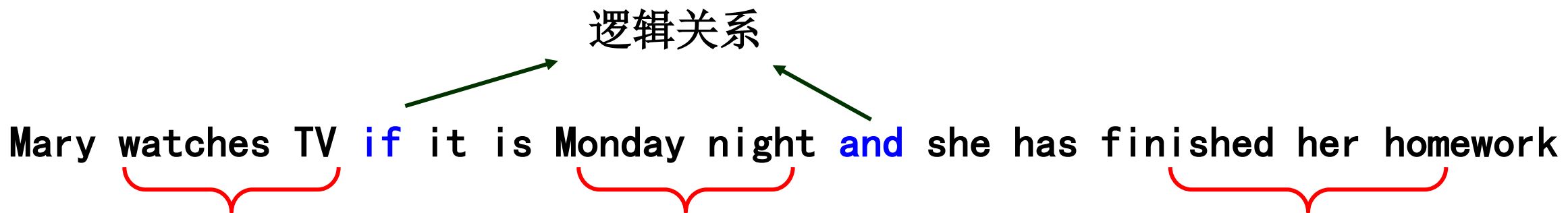
- 由真值表可直接写出标准形式的逻辑表达式

- 标准与或式（最小项表达式：and-or）

- 标准或与式（最大项表达式：or-and）

布尔代数的应用

方法1. 将文字描述的功能直接转换为表达式



Define:

$F = 1$: 看电视

$F = 0$: 没看电视

$A = 1$: 周一晚上

$A = 0$: 不是周一晚上

$B = 1$: 完成作业

$B = 0$: 没完成作业

$$F = A \cdot B$$

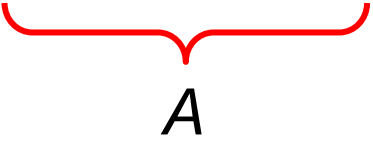
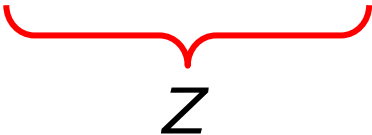
布尔代数的应用

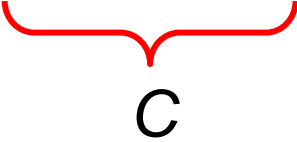
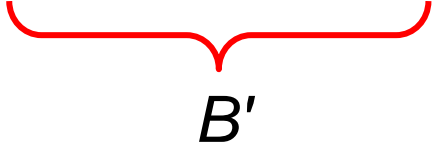
方法1. 将文字描述的功能直接转换为表达式

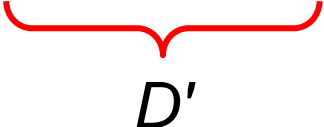
The alarm will ring if the alarm switch is turned on and the door is not closed, or it is after 6 P.M. and the window is not closed.

布尔代数的应用

$$Z = AB' + CD'$$

The alarm will ring **if** the alarm switch is on **and**


the door is not closed, **or** it is after 6 P.M. **and**


the window is not closed


怎样设计组合逻辑电路？

■方法1：直接转换(简单情况下)

- 将文字描述的功能直接转换为真值表或表达式

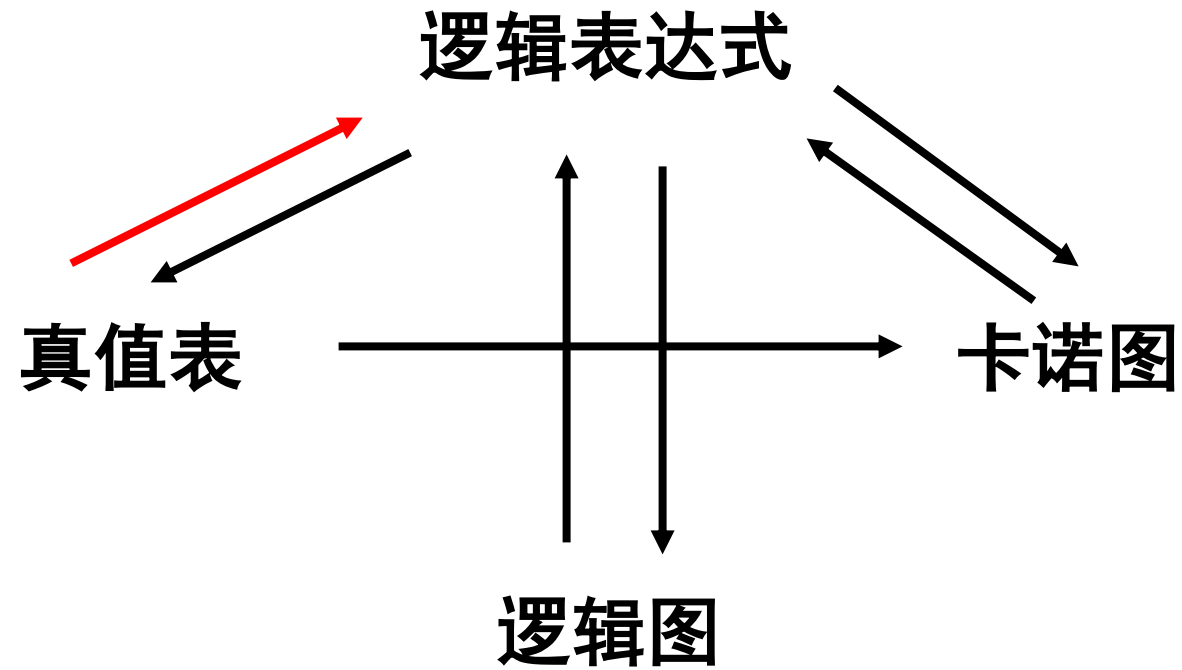
■方法2：真值表转换

- 由真值表可以直接写出两种标准形式的逻辑表达式

- 标准与或式 (最小项表达式: and-or)

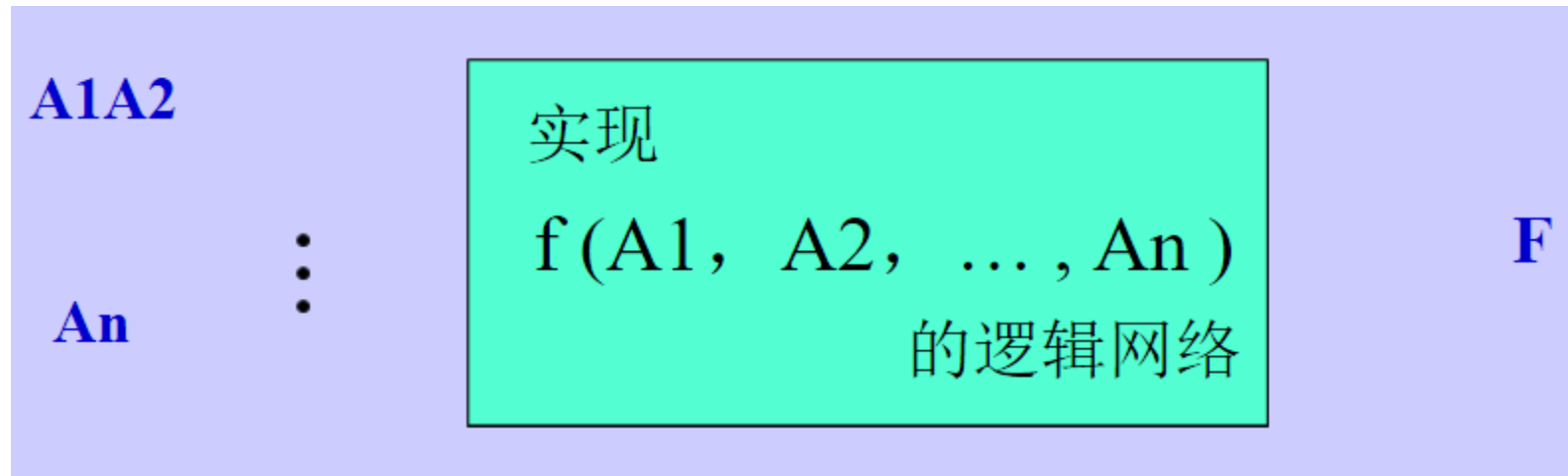
- 标准或与式 (最大项表达式: or-and)

逻辑函数的表示方法



逻辑函数

输入逻辑变量 $A_1, A_2, A_3, \dots, A_n$ ；输出逻辑变量 F ；记为
 $F = f(A_1, A_2, A_3, \dots, A_n)$ ，关系如下图所示：



输入变量（自变量）取值：0、1；

输出变量（逻辑函数值）取值：0、1。

使用真值表设计组合逻辑电路

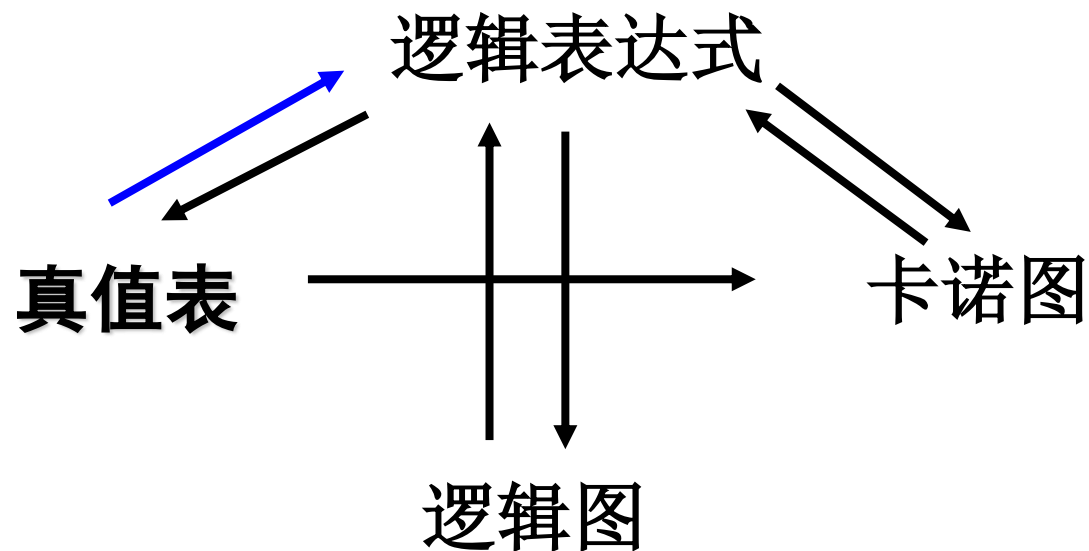
Truth table

AB C	F
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

真值表 \longrightarrow 表达式

① 写出标准与或式（乘积之和）

关注表中输出值为1的所有输入取值组合



使用真值表设计组合逻辑电路

Truth table

AB C	F
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 ✓

真值表 \longrightarrow 表达式

① 写出标准与或式（乘积之和）

关注表中输出值为1的所有输入取值组合

$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

输入取值组合中

1——原变量

0——反变量

使用真值表设计组合逻辑电路

真值表 \longrightarrow 表达式

②写出标准**或与式**（和之积）

关注表中**输出值为0**的所有输入取值组合

输入取值组合中

0——原变量

1——反变量

Truth table

AB C	F
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

$$F = (A+B+C) \cdot (A+B+\bar{C}) \cdot (A+\bar{B}+C) \cdot (\bar{A}+B+C)$$

布尔代数的应用

Example. 某电路有三个输入端 A , B , C , 当 $ABC \geq 011$ 时, 输出 $f = 1$, 否则 $f = 0$.

- 步骤:

1. 根据设计要求确定 —— 真值表
2. 根据真值表 —— 卡诺图(表达式)
3. 化简
4. 按设计要求, 变换逻辑表达式
5. 画出逻辑图

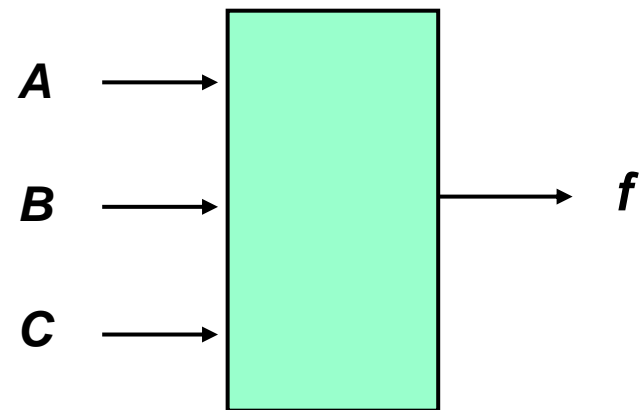
布尔代数的应用

Example. 某电路有三个输入端 A , B , C , 当 $ABC \geq 011$ 时, 输出 $f = 1$, 否则 $f = 0$.

穷举法

① True Table

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



布尔代数的应用

② Algebraic Expression

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

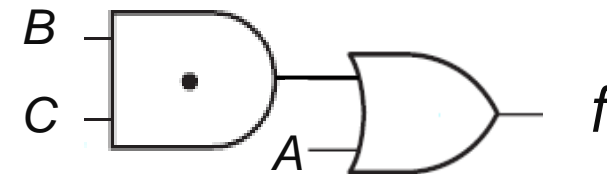
③ Simplification

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

$$= A'BC + AB' + AB$$

$$= A'BC + A = BC + A$$

④ Logic Circuit



A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

设计一个比赛需要用到的三人投票器，如果有两人或者三人投赞成票，则显示通过；否则不通过。请设计逻辑电路。

正常使用主观题需2.0以上版本雨课堂

作答

目 录

- 布尔代数的应用
- 最大项、最小项表达式
- 不完全给定函数

最小项 $minterm$ 、最大项 $Maxterm$ 的定义

序号		最小项	最大项
0	000	$A'B'C' = m_0$	$A + B + C = M_0$
1	001	$A'B'C = m_1$	$A + B + C' = M_1$
2	010	$A'B C' = m_2$	$A + B' + C = M_2$
3	011	$A'B C = m_3$	$A + B' + C' = M_3$
4	100	$A B'C' = m_4$	$A' + B + C = M_4$
5	101	$A B'C = m_5$	$A' + B + C' = M_5$
6	110	$A B C' = m_6$	$A' + B' + C = M_6$
7	111	$A B C = m_7$	$A' + B' + C' = M_7$

■ **编号次序**：自左向右从高到低

最小项输入取值组合中
原变量——1
反变量——0

最大项输入取值组合中
原变量——0
反变量——1

- n 个变量组成的最小项 m_i ：是一个与项（包含 n 个变量）
- n 个变量组成的最大项 M_i ：是一个或项（包含 n 个变量）
- 每个变量以原变量或反变量的形式出现，并且只出现一次。因子：
原变量或反变量
- n 个变量能组成的最小/大项的个数是 2^n

最小项 m_i 的特点

m_i

	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}\bar{C}$	$A\bar{B}C$	$AB\bar{C}$	ABC
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

① $\sum_{i=0}^{2^n-1} m_i = 1$

② $m_i \cdot m_j = 0 \quad (i \neq j)$

③ 对所有输入组合，
只有一个最小项为
1 (输入组合等于最
小项编号时)

最大项 M_i 的特点

A,B,C	$A + B + C$	$A + B + \bar{C}$	$A + \bar{B} + C$	$A + \bar{B} + \bar{C}$	$\bar{A} + B + C$	$\bar{A} + B + \bar{C}$	$\bar{A} + \bar{B} + C$	$\bar{A} + \bar{B} + \bar{C}$
000	0	1	1	1	1	1	1	1
001	1	0	1	1	1	1	1	1
010	1	1	0	1	1	1	1	1
011	1	1	1	0	1	1	1	1
100	1	1	1	1	0	1	1	1
101	1	1	1	1	1	0	1	1
110	1	1	1	1	1	1	0	1
111	1	1	1	1	1	1	1	0

- ① $\prod_{i=0}^{2^n-1} M_i = 0$
- ② $M_i + M_j = 1 \quad (i \neq j)$
- ③ 对所有输入组合，只有一个最大项为0(输入组合等于最小项编号)

最小项和最大项的性质

1. 最小项的反是最大项，最大项的反是最小项（编号相同）；

$$\overline{\overline{A}\overline{B}\overline{C}} = \overline{m_0} = A + B + C = M_0$$

$$\overline{A + \overline{B} + \overline{C}} = \overline{M_3} = \overline{A}BC = m_3$$

2. 全部最小项之和恒等于“1”；

$$m_0 + m_1 + m_2 + m_3 = 1$$

3. 全部最大项之积恒等于“0”；

$$M_0 M_1 M_2 M_3 = 0$$

4. 一部分最小项之和的反等于其余所有最小项之和

$$\overline{m_1 + m_2} = m_0 + m_3$$

$$\overline{m_0} = m_1 + m_2 + m_3$$

最小项和最大项的性质——续

5. 两个不同的最小项之积恒等于“0”；

例如：
$$ABC \cdot ABC\overline{C} = 0$$

6. 两个不同的最大项之和恒等于“1”；

例如：
$$(A + B + C) + (A + B + \overline{C}) = 1$$

与或标准型

$$Y = \sum m_i = \sum m(0,1,4,6,7) = m_0 + m_1 + m_4 + m_6 + m_7$$

或与标准型

$$Y = \prod M_i = \prod M(0,1,4,6,7) = M_0 M_1 M_4 M_6 M_7$$

最小项表达式

- 标准与或式
- list of “1”

011 101 110 111

$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= m_3 + m_5 + m_6 + m_7$$

$$= \Sigma m(3, 5, 6, 7)$$

最大项表达式

- 标准或与式
- list of 0

000

001

010

100

$$F = (A+B+C) \cdot (A+B+\bar{C}) \cdot (A+\bar{B}+C) \cdot (\bar{A}+B+C)$$

$$= M_0 \cdot M_1 \cdot M_2 \cdot M_4$$

$$= \prod M(0, 1, 2, 4)$$

最大项、最小项表达式

练习：

	Minterm Expansion of f	Maxterm Expansion of f	Minterm Expansion of f'	Maxterm Expansion of f'
$f =$ $\Sigma m(3, 4, 5, 6, 7)$	<div></div>	<div></div>	<div></div>	<div></div>
$f =$ $\Pi M(0, 1, 2)$	<div></div>	<div></div>	<div></div>	<div></div>

逻辑函数的标准形式

1、积之和的标准形式, 即最小项之和的形式

• 是哪些最小项之和呢? 函数输出为1的行对应的最小项

$$F = \Sigma_{X,Y,Z}(0,3,4,6,7)$$

$$F = \Sigma_{X,Y,Z}(0,3,4,6,7)$$

$$= X' \cdot Y' \cdot Z' + X' \cdot Y \cdot Z + X \cdot Y' \cdot Z' + X \cdot Y \cdot Z' + X \cdot Y \cdot Z$$

• 利用互补律 $X + X' = 1$ 可以把任何一个逻辑函数化为最小项之和的标准形式

例: 给定逻辑函数的积之和形式为

$$F = A + B' C$$

化为积之和的标准形式

逻辑函数的标准形式

2、和之积的标准形式 即最大项之积的形式

- 函数输出为0的行对应的最大项之积
符号 $\Pi A, B, C(1, 2, 4, 5)$ 是最大项列表.
- 利用互补律 $X \cdot X' = 0$ ，在缺少某一变量的和项中加上该变量，然后利用分配律 $A = A + X \cdot X' = (A + X)(A + X')$ 展开，就可以把任何一个逻辑函数化为最大项之积的标准形式

写出 $F=A+B'C$ 的最大项表达式

最小项和最大项之间的转换关系

A	B	C	F
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	0

$$G = \Pi_{A,B,C}(3,5,6) = F'$$

$$F = \Sigma_{A,B,C}(3,5,6)$$

$$F = \Pi_{A,B,C}(0,1,2,4,7)$$

标号互补

$$(A' \cdot B \cdot C)' = A + B' + C'$$

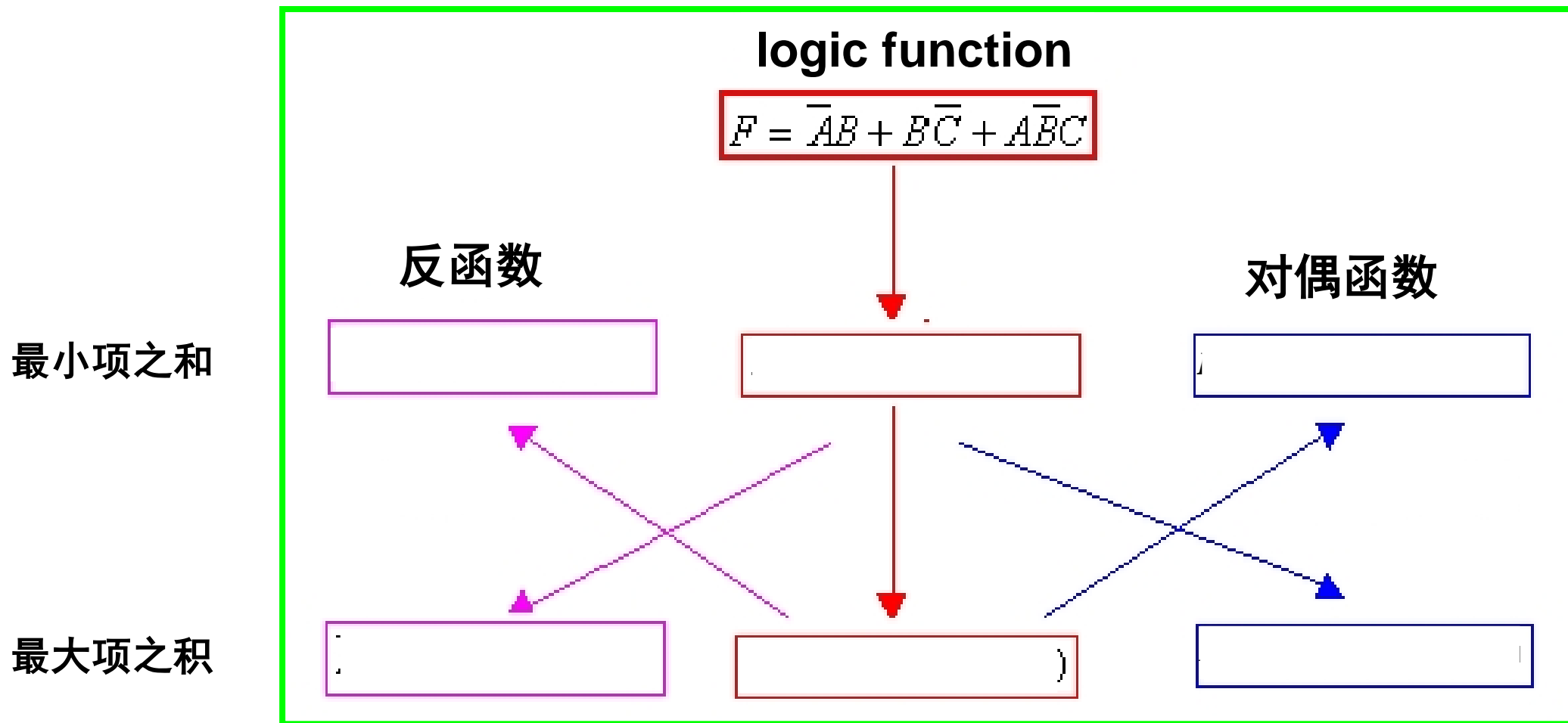
$$(A \cdot B' \cdot C)' = A' + B + C'$$

$$(A \cdot B \cdot C')' = A' + B' + C$$

$$M_i = m_i'$$

$$m_i = M_i'$$

最大项、最小项表达式



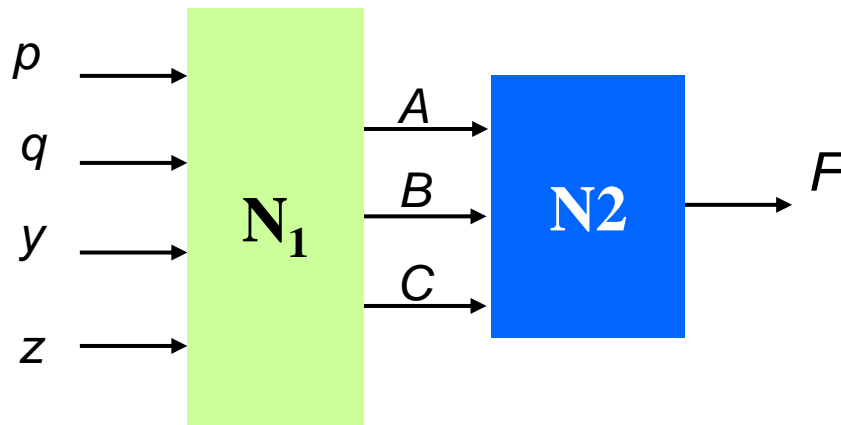
目 录

- 布尔代数的应用
- 最大项、最小项表达式
- 不完全给定函数

无关项 (Don't care terms)

- **约束项**: 不可能存在的输入取值组合。
- **任意项**: 某种输入取值组合存在, 但是并不关心它导致的输出结果是0还是1, 因为不影响电路功能。
- **约束项**和**任意项**统称为无关项, 即把这些最小项写入逻辑函数式无关紧要, 可以写入也可以删除。

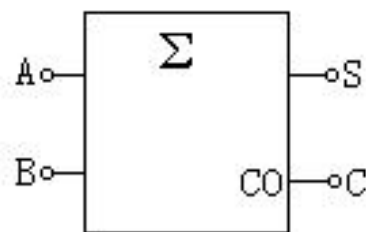
假设: 无论pqyz取何值, 电路N1输出都不为001或110。



A B C	F
0 0 0	1
0 0 1	X
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	X
1 1 1	1

无关项 (Don't care terms)

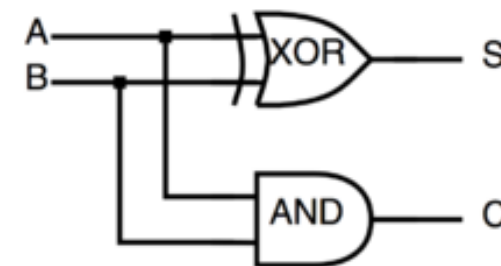
■ 半加器 (Half-adder)



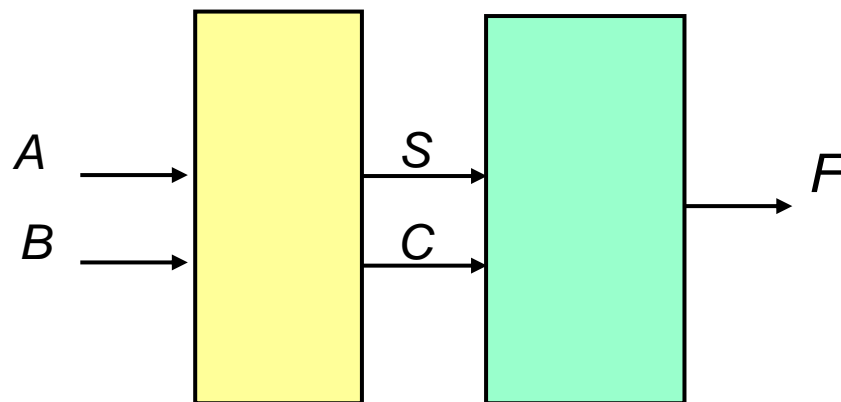
半加器逻辑符号

输入		输出	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

逻辑表达式: $S = A \oplus B$; $C = A \cdot B$ 。

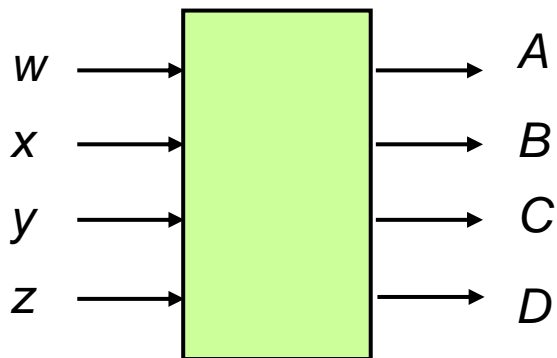


半加器的逻辑实现



例：8421BCD转余三码

将输入的
8421BCD码转
换为余3码



Decimal	8421 BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
....
9	1001	1100
	1010	XXXX

	1110	XXXX
	1111	XXXX

不完全给定函数

$$F = \sum m(0, 3, 7) + \sum d(1, 6)$$

$$F = \prod M(2, 4, 5) \cdot \prod D(1, 6)$$

<i>A B C</i>	<i>F</i>
0 0 0	1
0 0 1	X
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	X
1 1 1	1

例：8421BCD转余三码

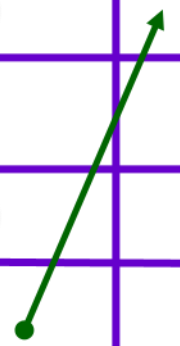
将输入的
8421BCD码转
换为余3码

$$A = \sum m(5, 6, 7, 8, 9) + \sum d(10, 11, 12, 13, 14, 15)$$

$$D = \sum m(0, 2, 4, 6, 8) + \sum d(10, 11, 12, 13, 14, 15)$$

Decimal	8421 BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
....
9	1001	1100
	1010	XXXX

	1110	XXXX
	1111	XXXX



目 录

- 布尔代数的应用
- 最大项、最小项表达式
- 不完全给定函数

以下哪些内容有不明白的，需要再讲解一下：

- ☐ A 最大项
- ☐ B 最小项
- ☐ C 不完全给定函数
- ☐ D 其他

提交