

# 基于传输函数理论的多值多变量 CMOS 电路综合\*

陈偕雄 赵小杰 吴训威

(杭州大学电子工程系, 杭州)

**摘要** 本文讨论了基于传输函数理论的多值 CMOS 电路综合技术。通过对基于传输函数理论的 CMOS 电路与 T 门的分析比较, 指出了它们的工作原理的一致性。在此基础上提出了用函数分解综合多值多变量 CMOS 电路的方法。

**关键词** 传输函数理论; 多值逻辑; 多值 CMOS 电路

## 一、引言

在传统的 Post 代数系统中, 三种基本运算为取大、取小和阈运算。由于利用 Post 代数综合多值电路时, 是以实现这些基本运算的电路作为基本单元, 因此它不能有效地指导以电路的基本元件(如晶体管)为基本单元的逻辑设计。此外, 由于 Post 代数引入了阈运算, 从而导致了多值电路的三明治式结构<sup>[1]</sup>。这种电路结构的主要部分仍是二值电路, 因此不能充分发挥多值电路的优势。传输函数理论正是为了克服上述缺点而提出的基于 MOS 管的工作原理, 并能有效地指导 CMOS 电路设计的代数系统。

在传输函数理论<sup>[1-6]</sup>中, 引入了如下三种运算:

### 阈比较运算

$$x' \triangleq \begin{cases} T, & \text{如 } x < t \\ F, & \text{如 } x > t \end{cases}$$
$${}^t x \triangleq \begin{cases} T, & \text{如 } x > t \\ F, & \text{如 } x < t \end{cases}$$

### 传输运算

$$c * \alpha \triangleq \begin{cases} c, & \text{如 } \alpha = T \\ \phi, & \text{如 } \alpha = F \end{cases}$$

上式中  $\phi$  表示物理上无信号的状态。

### 并运算

$$c_1 * \alpha_1 \# c_2 * \alpha_2 \triangleq \begin{cases} c_1 * \alpha_1, & \text{如 } \alpha_2 = F \\ c_2 * \alpha_2, & \text{如 } \alpha_1 = F \end{cases}$$

1990年9月3日收到, 1991年1月31日修改定稿。

\* 国家自然科学基金资助课题。

利用上述运算,可以获得任意多值函数的传输函数形式。

## 二、传输函数形式的化简

由于函数的传输函数形式直接与电路实现相对应,为获得尽量简单的电路实现,必须对函数的形式进行化简。化简可采用代数法或  $k$  图法。以三值函数为例,主要化简公式如下<sup>[3]</sup>:

$$x^{t_1} + x^{t_2} = x^{\max(t_1, t_2)} \quad (1)$$

$${}^{t_1}x + {}^{t_2}x = \min(t_1, t_2)x \quad (2)$$

$$x^{0.5} + {}^{0.5}x^{1.5} + {}^{1.5}x = T \quad (3)$$

$$c_1 * (\alpha \cdot \gamma) \# c_2 (\beta \cdot \gamma) = (c_1 * \alpha \# c_2 * \beta) * \gamma \quad (4)$$

$$0 * x^{0.5} \# 1 * {}^{0.5}x^{1.5} \# 2 * {}^{1.5}x = x \quad (5)$$

在变量较少时用  $k$  图化简较为方便。如二变量模 3 乘函数的  $k$  图如图 1(a) 所示。由  $k$  图可得

$$(x_1 \cdot x_2)_{\text{mod}3} = 0 * (x_1^{0.5} + x_2^{0.5}) \# x_2 * {}^{0.5}x_1^{1.5} \# x_1 * {}^{0.5}x_2^{1.5} \# 1 * ({}^{1.5}x_1 \cdot {}^{1.5}x_2)$$

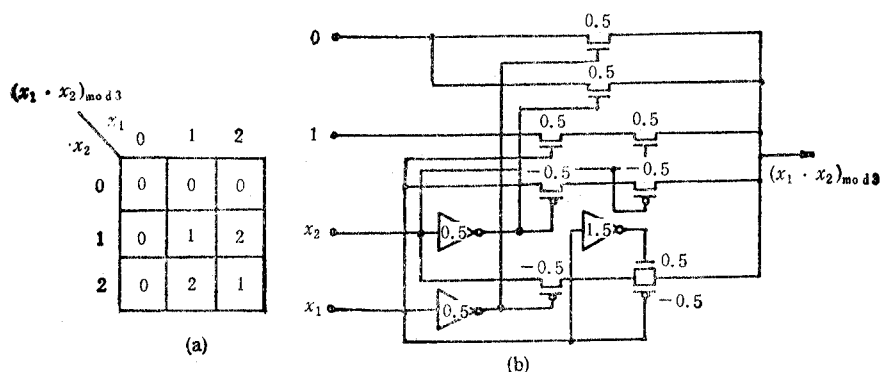


图 1 模 3 乘函数的  $k$  图和电路

(a)  $k$  图 (b) 电路

相应的电路实现如图 1(b) 所示。图中出现传输变源的支路。注意到在传输变源时,电路对输入变量无整形作用。当用这种电路作为基本模块构成系统或实现多值多变量电路时,由于电路中会出现许多 MOS 管串联的传输支路,从而导致输出阻抗和延迟增大。此外,当变量数较多时,不但函数化简困难,而且直接根据函数表达式实现电路时会出现多个 MOS 管串联的问题,因此必须寻找适合 CMOS 电路特点的、多变量多值电路的综合方法。

## 三、多变量多值 CMOS 电路的综合

在多值逻辑中, T 门可作为多值通用逻辑门用于实现任意多值函数。一个三值 T 门的定义为  $T(y_0, y_1, y_2; x) = y_i$ , 如  $x = i$ 。它相当于一个以输入变量  $x$  控制数据传输的

单刀三掷开关。当  $x = i$  时, 将数据  $y_i$  送至输出端。用传输函数理论设计的 CMOS 电路是用输入变量控制各个传输支路上的 MOS 开关, 将所需数据传送到输出端。对比基于传输函数理论的 CMOS 电路和 T 门电路, 可发现它们的工作原理是完全类似的, 因此, 对 T 门网络的化简方法可推广用于基于传输函数理论的 CMOS 电路设计。文献[4]指出, 用三值 T 门树型网络实现  $n$  变量的三值函数时, 传输支路会出现  $2n$  个 MOS 管串联的情况。当  $n$  较大时, 显然是不合理的。解决该问题的方法之一是在 T 门输出端加接整形电路<sup>[4]</sup>。解决该问题的另一途径是在设计时尽量引入中间变量以减少串联的 MOS 管数。

降维分解是处理多变量函数的一种常用手段, 可用于使用多值 T 门的逻辑设计<sup>[7-10]</sup>。以五变量函数  $f(x_1, x_2, x_3, x_4, x_5)$  为例, 可以对其中任意两个变量进行收缩处理。如收缩  $x_1, x_2$ , 则将它对其余三个变量  $x_3, x_4, x_5$  进行规范展开。此时展开系数均为变量  $x_1 x_2$  的函数(称子函数), 我们可根据它们的传输函数形式直接设计 CMOS 电路, 而这些子电路经过整形后的输出可以直接参与已降为三变量函数的电路实现, 从而解决了在传输通路上串联 MOS 管过多的问题。在上述讨论的基础上, 可以提出如下综合算法:

(1) 对函数逐级进行分解。每级对两个变量分解, 所选变量应使分解产生的子函数最少。

(2) 考察引入中间变量的可能性。若对于两个变量的  $R^2$  种取值 ( $R$  为多值逻辑基数), 分解产生的子函数对应的取值组合种类少于或等于  $R$ , 则可引入一中间变量来实现这些子函数。

(3) 分别实现中间变量和各级子函数, 在各级之间接入整形电路(引入中间变量后某些级间的整形电路可以省去), 最后构成总电路。

(4) 检查各传输支路串联 MOS 管数, 超过要求时, 可根据下式插入中间级电路<sup>[3]</sup>:

$$\alpha = '[2 * \alpha \# 0 * \bar{\alpha}] = [2 * \bar{\alpha} \# 0 * \alpha]'$$

下面通过设计实例<sup>[9-10]</sup>演示用函数分解实现多变量函数的 CMOS 电路的设计方法。

**例** 试实现如表 1 所示的三值五变量函数。

表 1

| $x_5 \backslash x_4 \backslash x_3 \backslash x_2 \backslash x_1$ |       |       |       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------------------------------------------------------------------|-------|-------|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|                                                                   | $x_5$ | $x_4$ | $x_3$ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|                                                                   | 0     | 0     | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|                                                                   | 0     | 0     | 0     | 1 | 1 | 1 | 2 | 2 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
|                                                                   | 0     | 1     | 2     | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| 00                                                                | 1     | 0     | 0     | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |   |
| 01                                                                | 1     | 2     | 2     | 2 | 2 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 2 |
| 02                                                                | 2     | 0     | 0     | 0 | 0 | 2 | 1 | 2 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 2 | 0 |
| 10                                                                | 2     | 0     | 0     | 0 | 0 | 2 | 1 | 2 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 2 | 0 |
| 11                                                                | 1     | 0     | 0     | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 12                                                                | 1     | 2     | 2     | 2 | 2 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 2 |
| 20                                                                | 1     | 2     | 2     | 2 | 2 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 2 |
| 21                                                                | 1     | 0     | 0     | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 22                                                                | 2     | 0     | 0     | 0 | 0 | 2 | 1 | 2 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 1 | 2 | 0 |   |

对于一个五变量函数有 10 种可能的二变量分解矩阵, 经比较确定以  $x_1 x_2$  作为子函

表 2

| $x_2 \backslash x_1$ | 00  | 01  | 02  | 10  | 11  | 12  | 20  | 21  | 22  |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0                    | $a$ | $b$ | $b$ | $b$ | $b$ | $a$ | $c$ | $a$ | $b$ |
| 1                    | 0   | $d$ | $d$ | $d$ | $d$ | 0   | $e$ | 0   | $d$ |
| 2                    | $a$ | $b$ | $b$ | $b$ | $b$ | $a$ | $c$ | $a$ | $b$ |

表 3

| $x_2$ | $x_1$ | $a$ | $b$ | $c$ | $d$ | $e$ |
|-------|-------|-----|-----|-----|-----|-----|
| 0     | 0     | 1   | 0   | 1   | 1   | 2   |
| 0     | 1     | 1   | 2   | 0   | 0   | 2   |
| 0     | 2     | 2   | 0   | 1   | 2   | 0   |
| 1     | 0     | 2   | 0   | 1   | 2   | 0   |
| 1     | 1     | 1   | 0   | 1   | 1   | 2   |
| 1     | 2     | 1   | 2   | 0   | 0   | 2   |
| 2     | 0     | 1   | 2   | 0   | 0   | 2   |
| 2     | 1     | 1   | 0   | 1   | 1   | 2   |
| 2     | 2     | 2   | 0   | 1   | 2   | 0   |

表 4

| $y_1$ | $a$ | $b$ | $c$ | $d$ | $e$ |
|-------|-----|-----|-----|-----|-----|
| 0     | 1   | 0   | 1   | 1   | 2   |
| 1     | 1   | 2   | 0   | 0   | 2   |
| 2     | 2   | 0   | 1   | 2   | 0   |

数变量进行分解时产生的子函数最少。该函数对  $x_1x_2$  进行分解,即对  $x_3x_4x_5$  进行展开,可得表 2, 其中五个子函数  $a, b, c, d, e$  分别如图 2(a)—2(e) 所示。我们将子函数  $a—e$  列于同一表中,如表 3 所示。由表可见,对于  $x_1x_2$  的全部取值,  $a—e$  只有三种取

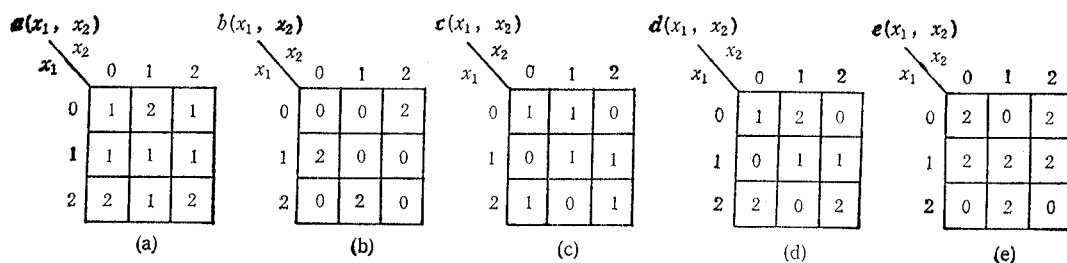


图 2 子函数  $a—e$  的  $k$  图 (a)  $a$  的  $k$  图 (b)  $b$  的  $k$  图 (c)  $c$  的  $k$  图 (d)  $d$  的  $k$  图 (e)  $e$  的  $k$  图

值组合,即 10112, 12002 和 20120, 因此可引入中间变量  $y_1(x_1, x_2)$ 。如按图 3 定义  $y_1$ , 则可得到如表 4 所示的子函数真值表。由图 3 和表 4 可得子函数和  $y_1$  的表示式:

$$y_1 = 0 * (x_1^{0.5} \cdot x_2^{0.5} + {}^{0.5}x_1^{1.5} \cdot {}^{0.5}x_2) \# 1 * ({}^{6.5}x_1^{1.5} \cdot x_2^{0.5} + {}^{1.5}x_1 \cdot {}^{0.5}x_2^{1.5} + x_1^{0.5} \cdot {}^{1.5}x_2) \# 2 * ({}^{1.5}x_1 \cdot x_2^{0.5} + x_1^{0.5} \cdot {}^{0.5}x_2^{1.5} + {}^{1.5}x_1 \cdot {}^{1.5}x_2) \quad (6)$$

$$a = 1 * y_1^{1.5} \# 2 * {}^{1.5}y_1 \quad (7)$$

$$b = 0 * (y_1^{0.5} + {}^{1.5}y_1) \# 2 * {}^{0.5}y_1^{1.5} \quad (8)$$

$$c = 0 * {}^{0.5}y_1^{1.5} \# 1 * (y_1^{0.5} + {}^{1.5}y_1) \quad (9)$$

$$d = 0 * {}^{0.5}y_1^{1.5} \# 1 * y_1^{0.5} \# 2 * {}^{1.5}y_1 \quad (10)$$

$$e = 0 * {}^{1.5}y_1 \# 2 * y_1^{1.5} \quad (11)$$

对表 2 所示三变量函数进一步分解, 不难发现以  $x_3x_4$  作为子函数变量为最佳。该函数对  $x_3x_4$  进行分解,

图 3  $y_1$  的  $k$  图表示

| $y_1$ | $x_1$ | $x_2$ |   |   |
|-------|-------|-------|---|---|
|       |       | 0     | 1 | 2 |
| 0     | 0     | 0     | 1 | 2 |
| 1     | 2     | 2     | 0 | 1 |
| 2     | 1     | 0     | 2 |   |

| $g(x_3, x_4)$ | $x_3$ | $x_4$ |   |   |
|---------------|-------|-------|---|---|
|               |       | 0     | 1 | 2 |
| 0             | 0     | a     | b | b |
| 1             | 1     | b     | b | a |
| 2             | 2     | c     | a | b |

| $h(x_3, x_4)$ | $x_3$ | $x_4$ |   |   |
|---------------|-------|-------|---|---|
|               |       | 0     | 1 | 2 |
| 0             | 0     | 0     | d | d |
| 1             | 1     | d     | d | 0 |
| 2             | 2     | e     | 0 | d |

图 4 子函数  $g, h$  的  $k$  图(a)  $g$  的  $k$  图, (b)  $h$  的  $k$  图

表 5

| $x_5$ |     |     |
|-------|-----|-----|
| 0     | 1   | 2   |
| $g$   | $h$ | $g$ |

| $y_2$ | $x_3$ | $x_4$ |   |   |
|-------|-------|-------|---|---|
|       |       | 0     | 1 | 2 |
| 0     | 0     | 0     | 2 | 2 |
| 1     | 2     | 2     | 2 | 0 |
| 2     | 1     | 0     | 2 |   |

图 5  $y_2$  的  $k$  图表示

所有取值, 子函数  $g, h$  只有三种不同取值组合,  $a0, bd$  和  $ce$ , 因此可引入中间变量  $y_2(x_3, x_4)$ ,  $y_2$  的  $k$  图表示如图 5 所示,  $g$  和  $h$  的真值表如表 7 所示。由此可得

$$g = a * y_2^{0.5} \# b * {}^{1.5}y_2 \# c * {}^{0.5}y_2^{1.5} \quad (12)$$

$$h = 0 * y_2^{0.5} \# d * {}^{1.5}y_2 \# e * {}^{0.5}y_2^{1.5} \quad (13)$$

$$y_2 = 0 * (x_3^{0.5} \cdot x_4^{0.5} + {}^{1.5}x_3 \cdot {}^{0.5}x_4^{1.5} + {}^{0.5}x_3^{1.5} \cdot {}^{1.5}x_4) \# 1 * (x_3^{0.5} \cdot {}^{1.5}x_4) \# 2 * ({}^{0.5}x_3 \cdot x_4^{1.5} + x_3^{1.5} \cdot {}^{0.5}x_4^{1.5} + {}^{1.5}x_3 \cdot {}^{1.5}x_4) \quad (14)$$

由表 5 得

$$f(x_1, x_2, x_3, x_4, x_5) = g * (x_5^{0.5} + {}^{1.5}x_5) \# h * {}^{0.5}x_5^{1.5} \quad (15)$$

图 6(a) 给出了实现该五变量函数的逻辑框图, 图中各子函数  $a-h$  和中间变量  $y_1, y_2$  可

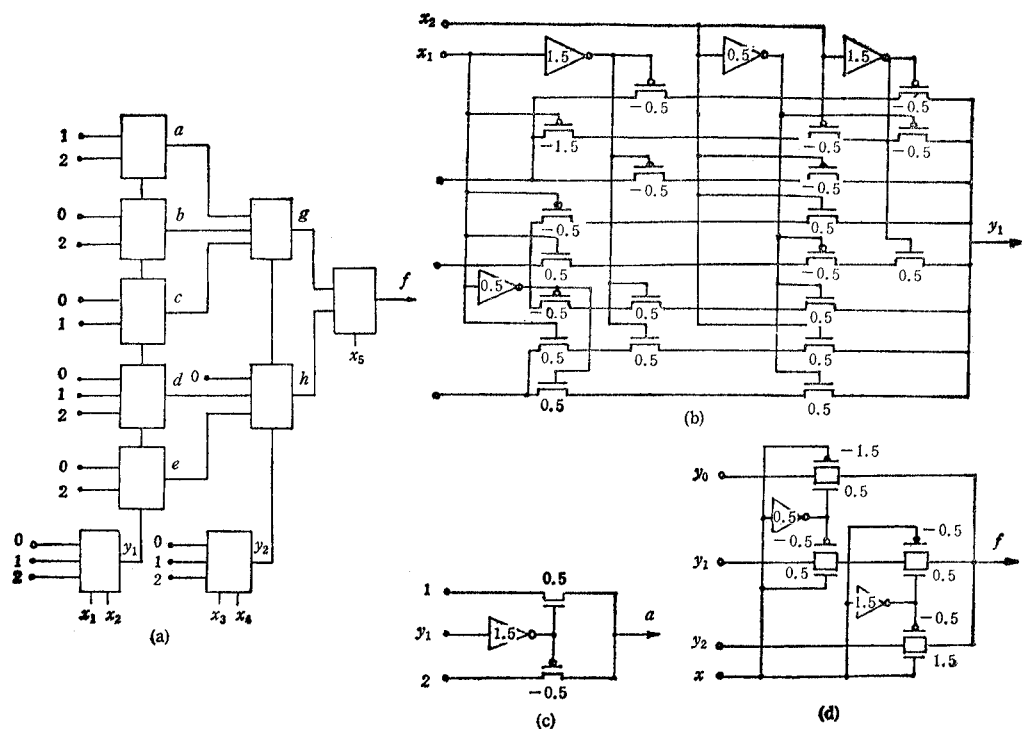
表 6

| $x_4$ | $x_3$ | $g$ | $h$ |
|-------|-------|-----|-----|
| 0     | 0     | $a$ | 0   |
| 0     | 1     | $b$ | $d$ |
| 0     | 2     | $b$ | $d$ |
| 1     | 0     | $b$ | $d$ |
| 1     | 1     | $b$ | $d$ |
| 1     | 2     | $a$ | 0   |
| 2     | 0     | $c$ | $e$ |
| 2     | 1     | $a$ | 0   |
| 2     | 2     | $b$ | $d$ |

表 7

| $y_2$ | $g$ | $h$ |
|-------|-----|-----|
| 0     | $a$ | 0   |
| 1     | $c$ | $e$ |
| 2     | $b$ | $d$ |

根据上述诸式直接得到相应的 CMOS 电路实现, 从而构成完整的电路。以中间变量  $y_1$

图 6 函数  $f$  的电路实现

(a) 框图, (b)  $y_1$  电路实现, (c)  $a$  电路实现, (d) 三值 T 门电路实现

和子函数  $a$  为例, 与 (6)、(7) 式相应的电路如图 6(b)、6(c) 所示. 类似地可以根据 (8) — (14) 式得到子函数  $b-h$  和  $y_2$  的电路实现, 在此不一一举例. 作为对比, 文献 [9] 给出了用三值 T 门实现该函数的电路, 其中  $y_1$  与子函数  $a$  分别使用 3 个和 1 个三值 T 门, 而 1 个三值 T 门的 CMOS 电路如图 6(d) 所示. 由图可见,  $y_1$  和  $a$  分别需用 24 个 MOS 管、6 个阈值反相器和 8 个 MOS 管、2 个阈值反相器, 而图 6(b) 和 6(c) 分别仅需 9 个 MOS 管、3 个阈值反相器和 2 个 MOS 管、1 个阈值反相器. 显然, 本文设计优于文献 [9] 的设计.

## 四、结 论

本文提出了基于传输函数理论的多值多变量 CMOS 电路的综合算法. 该算法使用函数分解方法以解决 CMOS 电路中传输支路串联 MOS 管过多的问题. 由于该算法是以 MOS 管作为基本单元的, 因此它能有效地指导 CMOS 电路设计, 可能获得较传统设计更为简单的电路结构, 从而可望用于 LSI 和 VLSI 内部电路设计. 最后应该指出, 该算法可以推广应用于四值或更高基数的多值多变量 CMOS 电路设计.

## 参 考 文 献

- [1] 吴训威, 陈偕雄, F. P. Prosser, 电子学报, **17** (1989) 5, 8—13.
- [2] 吴训威, 陈偕雄, F. P. Prosser, 电路与系统学报, **1** (1989) 1, 1—7.
- [3] Wu Xunwei, F. P. Prosser, *Int. J. Electronics*, **65** (1988) 5, 891—905.
- [4] Wu Xunwei, F. P. Prosser, Ternary CMOS sequential circuits, *IEEE Proc. of 18th Int. Symp. on MVL, Spain*, (1988), 307—313.
- [5] Wn Xunwei, Chen Xiexiong, F. P. Prosser, An investigation into quaternary CMOS full-adder based on transmission function theory, *IEEE Proc. of 19th Int. Symp. on MVL, Guangzhou*, (1989), 58—62.
- [6] 吴训威, 陈偕雄, F. P. Prosser, 中国科学(A 辑), 1989 年, 第 5 期, 第 528—536 页.
- [7] M. Kameyama, T. Higuchi, *IEEE Trans. on C*, **C-26** (1977) 12, 1297—1302.
- [8] K. Y. Fang, A. S. Wojcik, An approach to the modular design of multiple-valued logic function, *IEEE Proc. of 12th Int. Symp. on MVL, Paris*, (1982), 260—266.
- [9] K. Y. Fang, A. S. Wojcik, Synthesis of multiple-valued logic functions based on a modular design approach, *IEEE Proc. of 13th Int. Symp. on MVL, Kyoto*, (1983), 397—407.
- [10] K. Y. Fang, A. S. Wojcik, *IEEE Trans. on C*, **C-37** (1988) 10, 1293—1301.

# SYNTHESIS OF MULTIVALUED MULTIVARIABLE CMOS CIRCUITS BASED ON TRANSMISSION FUNCTION THEORY

Chen Xiexiong Zhao Xiaojie Wu Xunwei

(Hangzhou University, Hangzhou)

**Abstract** The synthesis technique for multivalued CMOS circuits based on transmission function theory is discussed. By comparing the CMOS circuits based on transmission function theory with the T gate, it shows that their action principles are identical. Based on it, the synthesis method for multivalued multivariable CMOS circuits using function decomposition is proposed.

**Key words** Transmission function theory; Multivalued logic; Multivalued CMOS circuits