中国科学
1989 年 5 月　　A 辑　　第 5 期

基于传输函数理论的四值 CMOS 电路*

吴训威　陈偕雄
(杭州大学物理系)
F. P. Prosser
(美国印第安纳大学计算机系)

摘　要

本文提出了传输函数理论. 在这一理论中,四值变量与逻辑变量被区分, 同时, 三种新运算——阈比较运算、传输运算及并运算被引入. 由于这些运算能描写 MOS 晶体管的工作原理, 因此用它们及逻辑运算表示的传输函数能直接指导四值 CMOS 电路的实现. 文中提出了若干基本传输函数的 CMOS 电路设计,它们经计算机模拟被证明具有正确的逻辑功能与理想的 DC 传输特性. 此外, 本文还介绍了四值 CMOS 传输网络的综合技术.

关键词: 传输函数理论, CMOS 电路,多值逻辑,四值电路

一、引　言

四值 MOS 电路的研究首先在存贮器上取得成功,其中一个方法是在 ROM 的制造过程中用四种不同阈值电压的 MOS 晶体管(存贮元件)来存贮四种不同的信息[1]. 1 个 MOS 管阈值电压的变化可以靠注入杂质原子的离子于晶体管沟道衬底来达到.

然而, 对于一般的四值 CMOS 组合电路的研究却遇到起源于代数的困难. 在传统的 Post 代数系统中[2],三种基本运算为取最小、取最大及文字运算:

$$x \wedge y \triangleq \min(x, y),$$
$$x \vee y \triangleq \max(x, y),$$
$${}^{i}x^{j} \triangleq \begin{cases} 3 & (i \leqslant x \leqslant j), \\ 0 & (\text{其他}). \end{cases}$$

上述定义中 $x, y, i, j \in \{0, 1, 2, 3\}$, 而在文字运算中的左标 $i = 0$ 及右标 $j = 3$ 常可省略,故有 ${}^{i}x^{j} = {}^{i}x^{3} \cdot {}^{0}x^{j} = {}^{i}x \cdot x^{j}$. 利用这些基本运算可将表 1 所示的二变量四值函数写成如下规范展开形式:

$$f(x, y) = (c_0 \wedge x^0 \wedge y^0) \vee (c_1 \wedge x^3 \wedge {}^{1}y \wedge y^1) \vee (c_2 \wedge x^0 \wedge {}^{2}y \wedge y^2)$$

1988 年 7 月 16 日收到修改稿.
* 国家自然科学基金资助项目.

$$\vee(c_3\wedge x^0\wedge {}^3y)\vee(c_4\wedge {}^1x\wedge x^1\wedge y^0)\vee(c_5\wedge {}^1x\wedge x^1\wedge {}^1y\wedge y^1)$$
$$\vee(c_6\wedge {}^1x\wedge x^1\wedge {}^2y\wedge y^2)\vee(c_7\wedge {}^1x\wedge x^1\wedge {}^3y)\vee(c_8\wedge {}^2x\wedge x^2\wedge y^0)$$
$$\vee(c_9\wedge {}^2x\wedge x^2\wedge {}^1y\wedge y^1)\vee(c_{10}\wedge {}^2x\wedge x^2\wedge {}^2y\wedge y^2)\vee(c_{11}\wedge {}^2x\wedge x^2\wedge {}^3y)$$
$$\vee(c_{12}\wedge {}^3x\wedge y^0)\vee(c_{13}\wedge {}^3x\wedge {}^1y\wedge y^1)\vee(c_{14}\wedge {}^3x\wedge {}^2y\wedge y^2)$$
$$\vee(c_{15}\wedge {}^3x\wedge {}^3y), \tag{1}$$

式中 $c_k\in\{3,2,1,0\}$ 为真值表中相应输入下的四值函数值．注意到如上函数的规范形式中包含大量的蜕化四值信号为二值信号的文字运算，因此，如果用多值逻辑门实现所有的运算∧和∨，无疑是能力上的浪费．为此，可对(1)式中的四值系数括因子而得到

$$f(x,\ y)=(3\wedge {}^3f^3)\vee(2\wedge {}^2f^2)\vee(1\wedge {}^1f^1)\vee(0\wedge {}^0f^0), \tag{2}$$

其中 ${}^if^i$ 为由文字之间的运算而构成的二值函数．这样，由(1)，(2)式指导的函数实现过程为：先将四值输入信号译码成二值信号以实现文字运算；其次由纯二值电路产生二值输出 ${}^if^i$；最后按(2)式用编码电路得到所需的四值函数．上述分析导致了对于多值 CMOS 电路的许多讨论都集中于译码器与编码器的各种设计[3-7]．基于(2)式的编码器可以方便地用 ${}^if^i$ 控制选通 4 个源 (3, 2, 1, 0) 的 CMOS 传输门实现最后的四值输出[3-4]，使得人们在近年来重视对 Pass 晶体管多值传输网络的研究[8-13]．这些讨论中的基本点是：先由多阈晶体管组成文字电路对输入信号译码，然后用它产生的二值信号控制 Pass 晶体管或传输门以决定与某函数值相应的源是否传输到输出．在这些讨论中已取消对专门编码器的要求，处理二值文字信号的二值夹心电路也转化为传输途径上开关的串联与并联结构．

表 1

x	y	$f(x,y)$
0	0	c_0
0	1	c_1
0	2	c_2
0	3	c_3
1	0	c_4
1	1	c_5
1	2	c_6
1	3	c_7
2	0	c_8
2	1	c_9
2	2	c_{10}
2	3	c_{11}
3	0	c_{12}
3	1	c_{13}
3	2	c_{14}
3	3	c_{15}

应该指出，上述工作中已经把(1)式中的函数值 c_i 与文字的地位作了区分：首先，在电路实现中函数值作为传输源处理，而文字信号却用于控制传输的开关（Pass 晶体管或 CMOS 传输门）；其次，文字之间的运算及 c_i 与文字之间运算的实现方法也不相同．这一切均暗示了(1)式中的文字与 c_k 不应属于同一类型的量，然而，以前的工作均未揭示两者在本质上的差别．据此分析，我们确定本文的研究目标为：

1. 建立一种新的代数系统，在该代数系统中存在两类变量——四值变量与逻辑变量，后者不表示信号，而是表示两种开关状态(开或关)，在这两类变量之间将用新的运算建立联系，这些运算应能描写信号决定开关状态及信号传输受开关控制的物理过程．

2. 讨论这些运算的性质，并用 MOS 管实现．

3. 使用新运算表示具体的函数形式，发展相应的函数化简技术，并根据函数的简化形式指导相应的电路实现．

二、传输函数理论

在逻辑代数中设 α,β 为逻辑变量，它们取两个逻辑值——T (真)及 F (假)．该两个逻辑值仅有相反的关系而不存在可比较的大小关系．它们可用于表示两个相反的状态，如一个开关的开与关．在逻辑变量之间的三种基本运算为非(～)，与(·)及或(+)，其中后两种运算

常用开关的串联及并联来解释。

在四值 Post 代数中,设 $x, y\cdots\cdots$ 为四值变量,它们取值为 3, 2, 1, 0,而这些取值是可以相互比较的,并构成全序关系: $3>2>1>0$。如果引入检测阈值 t, $t\in\{2.5, 1.5, 0.5\}$,则全序关系为 $3>2{,}5>2>1{,}5>1>0{,}5>0$。在物理上,该全序关系可描写四种信号电平及 3 个电路检测电平之间的大小关系。在四值变量之间的基本运算为补(—)、取最小(∧)、取最大(∨)及文字运算 $({}^{t}x^{t})$,所有这些运算都蕴涵大小比较的意义。例如,补运算是以 3 为被减数的减法运算,取最小、取最大运算是输入变量之间相互比较运算,而文字运算则是变量与定值之间比较运算,用以判定变量的取值。为了描写物理上无电压信号的状态,还应引入一个空值 ϕ,由于它无法参加与有值信号的比较,因此不参与上述基本运算。

首先,可引入如下两种阈比较运算以描写 MOS 管的栅极信号决定 MOS 管开关状态的物理过程:

高阈比较运算

$$ {}^{t}x \triangleq \begin{cases} T, & \text{如 } x>t, \\ F, & \text{如 } x<t; \end{cases} \tag{3} $$

低阈比较运算

$$ x^{t} \triangleq \begin{cases} T, & \text{如 } x<t, \\ F, & \text{如 } x>t. \end{cases} \tag{4} $$

对四值变量的阈比较运算的结果为逻辑变量。可以证明,对于同一四值变量的各种阈比较运算存在如下逻辑关系:

1. 非关系

$$ \left.\begin{aligned} \widetilde{x^{t}} &= {}^{t}x, \\ \widetilde{{}^{t}x} &= x^{t}, \end{aligned}\right\} \tag{5} $$

2. 与关系

$$ \left.\begin{aligned} x^{t_1}\cdot x^{t_2} &= x^{\min(t_1,t_2)}, \\ {}^{t_1}x\cdot{}^{t_2}x &= {}^{\max(t_1,t_2)}x, \end{aligned}\right\} \tag{6} $$

3. 或关系

$$ \left.\begin{aligned} x^{t_1}+x^{t_2} &= x^{\max(t_1,t_2)}, \\ {}^{t_1}x+{}^{t_2}x &= {}^{\min(t_1,t_2)}x. \end{aligned}\right\} \tag{7} $$

由于阈比较运算建立了四值变量与逻辑变量之间的关系,因此,通过阈运算能建立四值 Post 代数的基本运算——补、取最小、取最大及逻辑代数中的基本运算——非、与、或之间的联系。容易证明存在如下关系:

$$ \left.\begin{aligned} 1.\ {}^{t}(\bar{x}) &= \widetilde{{}^{3-t}x}, \\ (\bar{x})^{t} &= \widetilde{x^{3-t}}, \end{aligned}\right\} \tag{8} $$

$$ \left.\begin{aligned} 2.\ {}^{t}(x\wedge y) &= {}^{t}x\cdot{}^{t}y, \\ (x\wedge y)^{t} &= x^{t}+y^{t}, \end{aligned}\right\} \tag{9} $$

$$ \left.\begin{aligned} 3.\ {}^{t}(x\vee y) &= {}^{t}x+{}^{t}y, \\ (x\vee y)^{t} &= x^{t}\cdot y^{t}. \end{aligned}\right\} \tag{10} $$

其次,可引入如下两种新运算来描写 MOS 管网络的开关状态控制源信号传输的物理过程:

传输运算

$$y*\alpha \triangleq \begin{cases} y, & 如\ \alpha = T, \\ \phi, & 如\ \alpha = F; \end{cases} \tag{11}$$

并运算

$$y_1*\alpha_1 \# y_2*\alpha_2 \triangleq \begin{cases} y_1*\alpha_1, & 如\ \alpha_2 = F, \\ y_2*\alpha_2, & 如\ \alpha_1 = F. \end{cases} \tag{12}$$

上述定义中 y 被称为传输源，ϕ 表示无信号状态，即悬空状态，并运算表示多个传输支路的线结，因此在与四值信号的并运算中 ϕ 是无效的。此外，在并运算的定义中如果 $y_1 \neq y_2$，则 $\alpha_1 = \alpha_2 = T$ 将不被允许，因为它在物理上导致不同电源之间的冲突。根据上述运算的定义与电路意义容易得到如下性质：

1. 重迭律

$$\begin{aligned} (y*\alpha)*\alpha &= y*\alpha, \\ y*\alpha \# y*\alpha &= y*\alpha; \end{aligned} \tag{13}$$

2. 串联传输律

$$(y*\alpha_1)*\alpha_2 = y*(\alpha_1 \cdot \alpha_2); \tag{14}$$

3. 并联传输律

$$y*\alpha_1 \# y*\alpha_2 = y*(\alpha_1 + \alpha_2); \tag{15}$$

4. 交换律

$$y_1*\alpha_1 \# y_2*\alpha_2 = y_2*\alpha_2 \# y_1*\alpha_1; \tag{16}$$

5. 结合律

$$(y_1*\alpha_1 \# y_2*\alpha_2) \# y_3*\alpha_3 = y_1*\alpha_1 \# (y_2*\alpha_2 \# y_3*\alpha_3) = y_1*\alpha_1 \# y_2*\alpha_2 \# y_3*\alpha_3; \tag{17}$$

6. 分配律

$$(y_1*\alpha_1 \# y_2*\alpha_2)*\beta = y_1*(\alpha_1 \cdot \beta) \# y_2*(\alpha_2 \cdot \beta); \tag{18}$$

7. 透阈律

$$\left.\begin{aligned} {}^t(y_1*\alpha_1 \# y_2*\alpha_2) &= {}^t y_1 \cdot \alpha_1 + {}^t y_2 \cdot \alpha_2, \\ (y_1*\alpha_1 \# y_2*\alpha_2)^t &= y_1^t \cdot \alpha_1 + y_2^t \cdot \alpha_2, \end{aligned}\right\} \tag{19}$$

上述诸式中已设定传输运算 $*$ 优先于并运算 $\#$。

利用新引入的三种运算及原有的逻辑运算，可以获得任意四值函数的传输函数表示形式。以表 1 定义的函数为例，相应的传输函数为

$$\begin{aligned} f(x, y) = {} & c_0*(x^{0.5} \cdot y^{0.5}) \# c_1*(x^{0.5} \cdot {}^{0.5}y \cdot y^{1.5}) \\ & \# c_2*(x^{0.5} \cdot {}^{1.5}y \cdot y^{2.5}) \# c_3*(x^{0.5} \cdot {}^{2.5}y) \# c_4*({}^{0.5}x \cdot x^{1.5} \cdot y^{0.5}) \\ & \# c_5*({}^{0.5}x \cdot x^{1.5} \cdot {}^{0.5}y \cdot y^{1.5}) \# c_6*({}^{0.5}x \cdot x^{1.5} \cdot {}^{1.5}y \cdot y^{2.5}) \\ & \# c_7*({}^{0.5}x \cdot x^{1.5} \cdot {}^{2.5}y) \# c_8*({}^{1.5}x \cdot x^{2.5} \cdot y^{0.5}) \# c_9*({}^{1.5}x \cdot x^{2.5} \cdot {}^{0.5}y \cdot y^{1.5}) \\ & \# c_{10}*({}^{1.5}x \cdot x^{2.5} \cdot {}^{1.5}y \cdot y^{2.5}) \# c_{11}*({}^{1.5}x \cdot x^{2.5} \cdot {}^{2.5}y) \# c_{12}*({}^{2.5}x \cdot y^{0.5}) \\ & \# c_{13}*({}^{2.5}x \cdot {}^{0.5}y \cdot y^{1.5}) \# c_{14}*({}^{2.5}x \cdot {}^{1.5}y \cdot y^{2.5}) \# c_{15}*({}^{2.5}x \cdot {}^{2.5}y). \end{aligned} \tag{20}$$

上述函数形式与(1)式在形式上相似，但无论是变量的性质或者运算的性质，都有了本质上的差别。应用(15)式所示的并联传输律可对上式抽因子而得到

$$f(x, y) = 3*{}^{2.5}f \# 2*({}^{1.5}f \cdot f^{2.5}) \# 1*({}^{0.5}f \cdot f^{1.5}) \# 0*f^{0.5}$$

$$= 3 * \alpha_3 \# 2 * \alpha_2 \# 1 * \alpha_1 \# 0 * \alpha_0, \tag{21}$$

式中开关函数 α_i 满足互补与互斥的逻辑关系:

$$\left.\begin{aligned} &\alpha_3 + \alpha_2 + \alpha_1 + \alpha_0 = T, \\ \text{及}\quad &\alpha_i \cdot \alpha_j = F, \quad (i \neq j). \end{aligned}\right\} \tag{22}$$

表 2 给出了若干单变量四值函数的真值表,它们的运算意义及相应的传输函数为

表 2

x	$\bar{x}$	M	$\tilde{x}$	$\overline{x(0.5)}$	$\overline{x(1.5)}$	$\overline{x(2.5)}$
0	1	c_0	3	3	3	3
1	2	c_1	2	0	3	3
2	3	c_2	1	0	0	3
3	0	c_3	0	0	0	0

循环运算

$$\tilde{x} = 1 * x^{0.5} \# 2 * ({}^{0.5}x \cdot x^{1.5}) \# 3 * ({}^{1.5}x \cdot x^{2.5}) \# 0 * {}^{2.5}x, \tag{23}$$

数据选择运算

$$M = c_0 * x^{0.5} \# c_1 * ({}^{0.5}x \cdot x^{1.5}) \# c_2 * ({}^{1.5}x \cdot x^{2.5}) \# c_3 * {}^{2.5}x, \tag{24}$$

反相运算

$$\bar{x} = 3 * x^{0.5} \# 2 * ({}^{0.5}x \cdot x^{1.5}) \# 1 * ({}^{1.5}x \cdot x^{2.5}) \# 0 * {}^{2.5}x, \tag{25}$$

阈 t 反相运算

$$\overline{x(t)} = 3 * x^t \# 0 * {}^t x, \; t \in \{0.5, 1.5, 2.5\}. \tag{26}$$

注意阈 t 反相运算具有二值(3,0)的函数值,对于这种二值(3, 0)的函数使用(19)式所示的透阈律能达到变换阈比较运算的效果:

$$\left.\begin{aligned} {}^{t'}[\overline{x(t)}] &= x^t, \\ [\overline{x(t)}]^{t'} &= {}^t x, \end{aligned}\right\} \tag{27}$$

式中阈 t' 可任意取值,这一关系在下节的电路实现中将有重要用途.

三、传输运算的 MOS 电路实现

首先假定四值信号的电平分别为 $3E$, $2E$, E, 0 (地),它们分别用相应的逻辑级 3,2, 1, 0 予以表示. 电路中只用增强型的 MOS 晶体管,其中 NMOS 晶体管具有三种阈值电压: $V_{TN} = 2.5E$, $1.5E$, $0.5E$, 它们分别用 2.5, 1.5, 0.5 表示,并称为 NMOS 晶体管的阈. PMOS 晶体管的三种阈值电压 V_{TP} 为 $-2.5E$, $-1.5E$, $-0.5E$, 对应的阈值为 -2.5, -1.5, -0.5. NMOS 管的导通条件是栅源间电压降大于该管阈值电压: $V_G - V_S > V_{TN}$, 而 PMOS 管的导通条件为: $V_G - V_S < V_{TP}$.

1. 高阈比较运算控制的传输运算 $y * {}^t x$ 的 MOS 电路实现

高阈比较运算表示信号电压高才能够开启 MOS 开关,因此,NMOS 管可选作 Pass 晶体管,如图 1(a) 所示. 图中晶体管阈为 $t - y$. 当 $x > t$ 时, $x - y > t - y$, 于是满足 NMOS 管的导通条件. 然而, $t - y$ 即 NMOS 管的阈必须大于 0, 因此如果发生 $t - y < 0$ (此时必然不为 0), 则可利用上节的变阈关系(27)式得到

$$y * {}^t x = y * [\overline{x(t)}]^{t'}.$$

这表示可改用 PMOS 管作 Pass 管来实现 $y*^t x$，如图 1(b) 所示。图中使用了阈 t 反相器，其电路实现将在下节中讨论。图 1(b) 中 PMOS 管的阈值取为 -0.5，它可用于传输 $y=1$，2，3 的情况。

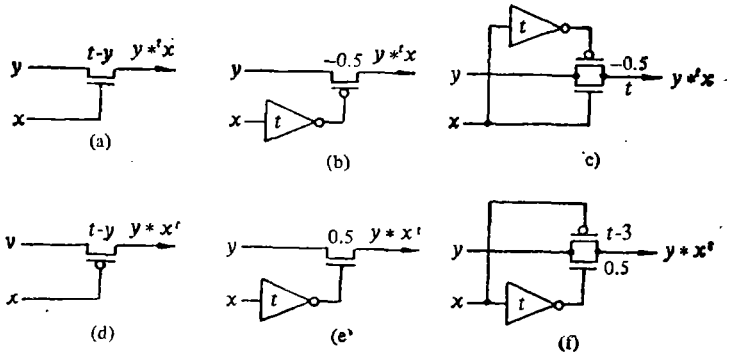

图 1　传输运算的 MOS 电路实现

(a) $t-y>0$，(b) $y\in\{1,2,3\}$，(c) $y\in\{0,1,2,3\}$，(d) $t-y<0$，(e) $y\in\{0,1,2\}$，(f) $y\in\{0,1,2,3\}$

源 y 取各种值时的传输情况综合如下：$y=0$ 必须由图 1(a) 实现；$y=3$ 必须由图 1(b)实现。$y=1,2$ 可由图 1(b) 实现，但如果满足 $t-y>0$，则也可由图 1(a) 实现，例如 (1) $t=2.5$，$y=1,2$；(2) $t=1.5$，$y=1$。然而，这种传输电路具有输出倒灌的危险。以 $t=1.5$，$y=1$ 为例，此时 NMOS 管的阈为 $t-y=0.5$，如果 $x=1$，该管不应导通。但如果漏极输出端的信号为 0，该管也将导通，从而发生电源 1，0 冲突的情况。显然，如果输入 x 只取二值信号(3,0)，则这种倒灌现象便不发生，因为 $x=0$ 对 NMOS 管的关闭将是绝对的。因此，如果(1)源为中间电平；(2)栅极输入为中间电平；(3)在不应导通的条件下输出可能出现比源低的电平，则使用单个 NMOS 管会发生倒灌，必须改用图 1(b) 所示的传输电路。图 1(c) 为变源情况下 $y*^t x$ 的电路实现，图中 NMOS 管只传输 $y=0$，而 PMOS 部分用于传输 $y=1,2,3$，其功能相当于文献[8]中的 Make 门。

2. 低阈比较运算控制的传输运算 $y*x^t$ 的 MOS 电路实现

类似于上面的讨论，几种情况下 $y*x^t$ 的 MOS 电路实现如图 1(d)—(f) 所示。$y=3$ 必须由图 1(d) 实现，$y=0$ 必须由图 1(e) 实现。$y=1,2$ 必可由图 1(e) 实现，但如果满足 $t-y<0$，则也可由图 1(d) 实现，例如 (1) $t=0.5$，$y=1,2$；(2) $t=1.5$，$y=2$。然而，此时必须验明 x 为应使此管截止的中间电平输入时漏极是否存在较源高的输出电平而导致倒灌的发生。图 1(f) 为变源情况下 $y*x^t$ 的电路实现，图中 PMOS 管只传输 $y=3$，而 NMOS 管传输 $y=0,1,2$。它的功能相当于文献[8]中 Break 门。

四、CMOS 传输网络的实现与简化

基于上节讨论，可从(23)—(26)式所示的传输函数得到电路实现，如图 2 所示。图 2(a) 中源 1 与源 2 分支均使用二值信号 (3, 0) 控制的 MOS 管作为 Pass 管，因此不会发生倒灌。图 2(b) 中 $c_i\in\{3,2,1,0\}$，变源传输结构被采用，与文献[12]比较，其结构较简单。这是由于绕过文字电路，直接从传输函数形式出发进行设计的结果，在图 2(c)中。1 与源

2 分支中第一个 Pass 管均使用 x 直接控制的方式，经验明不会发生倒灌。图 2(d) 具有与一般的二值反相器相同的结构，十分简单。它可用于产生必要的二值中间信号而毋需顾虑它们会过分增加电路的复杂性。

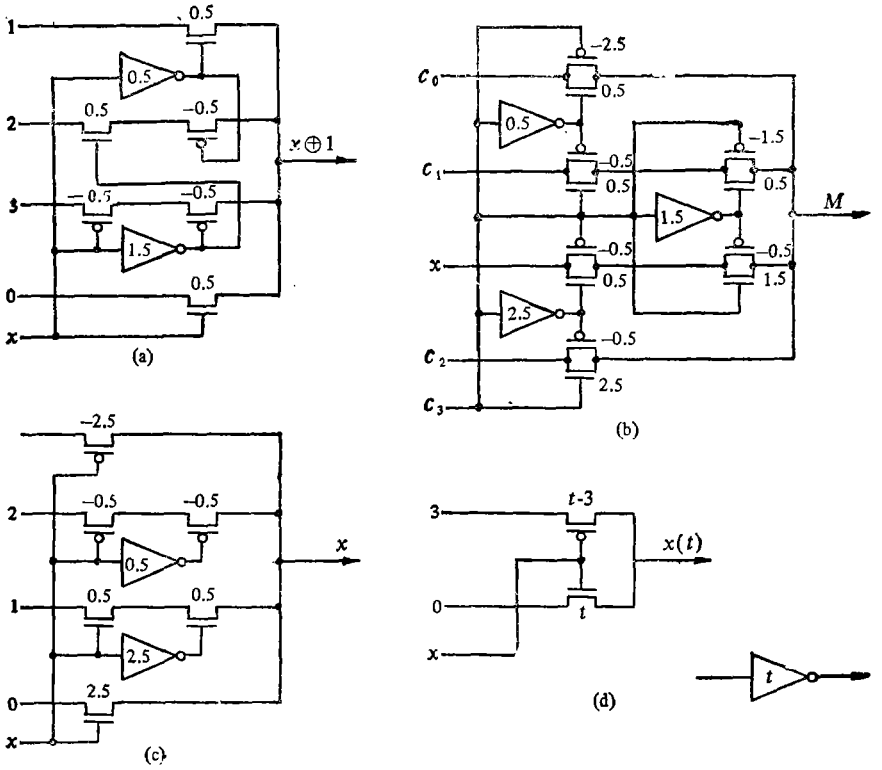

图 2 (a) 循环门 (b) 四选-数据选择器 (c) 反相器 (d) 阈 t 反相器

图 2 中的电路已用电路分析程序 SPICE II 予以计算机模拟，结果表明它们具有正确的逻辑功能及理想的 DC 传输特性。分析还表明，如采用单个传输门传输中间源，则可获得较低的输出阻抗。MOS 管的模拟参数取自文献[8]。

根据 CMOS 电路的特点，减少 Pass 管的数量，即减少传输函数形式中阈比较运算的总数，乃是简化 CMOS 传输网络的关键。由于开关函数 α_i 即为一般的二值逻辑函数，因此只要注意到同一变量的各种阈运算彼此并不独立，而满足(5)—(7)式的关系，便可按传统的逻辑函数化简技术进行化简。然而，简化后的函数并不拘泥于积之和及和之积两种二级形式，而允许继续抽因子化简之。因为逻辑变量的与、或由开关的串联、并联实现，而并非用门的级联实现，因此抽因子过程并不增加电路的"级数"。

例 1：由图 3(a) 所示的 K 图[8]可得到四值二变量函数 $f_1(x, y)$ 的传输函数形式中 4 个开关函数的简化形式：

$$\alpha_0 = {}^{1.5}x \cdot {}^{0.5}y \cdot y^{2.5} + x^{0.5} \cdot {}^{2.5}y,$$

$$\alpha_1 = x^{0.5} \cdot y^{2.5} + x^{1.5} \cdot y^{0.5} + {}^{0.5}x \cdot x^{1.5} \cdot {}^{2.5}y,$$

$$\alpha_2 = {}^{0.5}x \cdot x^{1.5} \cdot {}^{0.5}y \cdot y^{2.5} + {}^{1.5}x \cdot x^{2.5} \cdot {}^{2.5}y,$$

$$\alpha_3 = {}^{1.5}x \cdot y^{0.5} + {}^{2.5}x \cdot {}^{2.5}y.$$

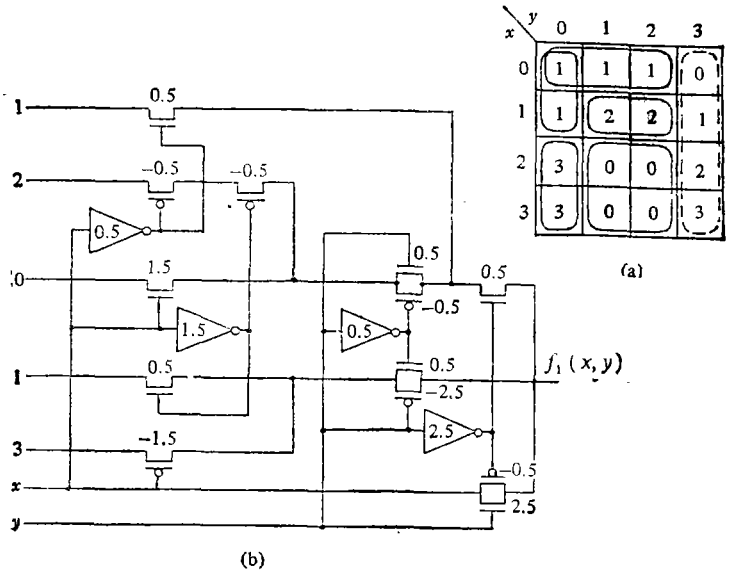

图 3　(a) K 图综合方案和(b)CMOS电路实现

与上式相应的电路实现共需 23 个 Pass 晶体管及 5 个阈 t 反相器，总计 33 个 MOS 管。但是，α_0，α_1，α_2 中的第一项可先后抽因子 $y^{2.5}$ 及 $^{0.5}y$；α_1 的第二项与 α_3 的第一项可抽因子 $y^{0.5}$；α_0，α_1，α_2，α_3 的最后一项可抽因子 $^{2.5}y$。于是，该传输函数简化为

$$f_1(x, y) = \{1 * {}^{0.5}x \# [0 * {}^{1.5}x \# 2 * ({}^{0.5}x \cdot x^{1.5})] * {}^{0.5}y\} * y^{2.5} \# (1 * x^{1.5} \# 3 * {}^{1.5}x)$$
$$* y^{0.5} \# [0 * x^{0.5} \# 1 * ({}^{0.5}x \cdot x^{1.5}) \# 2 * ({}^{1.5}x \cdot x^{2.5}) \# 3 * {}^{2.5}x] * {}^{2.5}y.$$

根据(21)式，上式中最后一个方括弧内可简化为 x，如图 3(a) 中虚线所示，它可用变源传输实现。由于变源传输能同时输入信号于 MOS 管源极与栅极，充分利用了 MOS 管的功能，因此可获较好的化简效果。与上式相应的 CMOS 电路实现如图 3(b) 所示。

五、结　论

本文通过引入阈比较运算建立了 Post 代数与逻辑代数之间的联系。由于在四值函数的表示中引入的逻辑变量描写开关状态，并且引入的阈比较运算、传输运算及并运算能反映 MOS 管的作用原理，因此，在此基础上建立的传输函数理论能很好地指导四值 CMOS 电路实现，也能推广应用于其他基数的多值 CMOS 电路设计[14]。

应该指出，本文提出的四值 CMOS 电路能与目前的二值 CMOS 电路兼容及可相比较，它们具有相同的开关作用原理，无被动元件(电阻)及低功耗等特点，适宜于 VLSI 技术，因此四值 CMOS 芯片期望被设计与制作，并象四值 ROM 一样付诸实用。

参　考　文　献

[1] Rich, D. A., *IEEE Trans. Comput.*, C-35(1986), 99—106.
[2] Post, E. L., *Amer. J. Math.*, 43(1921), 163—186.
[3] Freitas, D. A. and Current, K. W., *Proc. of ISMVL*, 1984, 164—168.
[4] Mangin, J. L. and Current, K. W., *IEEE Trans. Comput*, C-35(1986), 157—161.

[5] Abd-Elbarr, M. H. and Vranesic, Z. G., *Proc. of ISMVL*, 1985, 323—330.
[6] Singh, A. D., *ibid.*, 1986, 224—231.
[7] Singh, A. D., *ibid.*, 1987, 128—133.
[8] Zukeran, C. et al., *ibid.*, 1985, 84—90.
[9] Watanabe, T. et al., *ibid.*, 1986, 26—32.
[10] Ishizuka, C., *ibid.*, 1986, 51—57.
[11] Horie, K. et al., *ibid.*, 1987, 142—149.
[12] Ueno, F. et al., *ibid.*, 1987, 268—274.
[13] Ishizuka, C. and Xu, J., *ibid.*, 1987, 292—297
[14] Wu, X. W. (吴训威) and Prosser, F. P., *ibid.*, 1988, 381—387.