

PAL14H4、PAL10L8、PAL14L4、PAL16C1 等都是专用输出结构的器件。其中 PAL10H8 和 PAL14H4 的输出端是与或门结构，以高电平作输出信号（高电平有效）；PAL10L8 和 PAL14L4 的输出端是与或非门结构，以低电平作输出信号（低电平有效）；PAL16C1 的输出端是互补输出的或门结构，同时输出一对互补的信号。

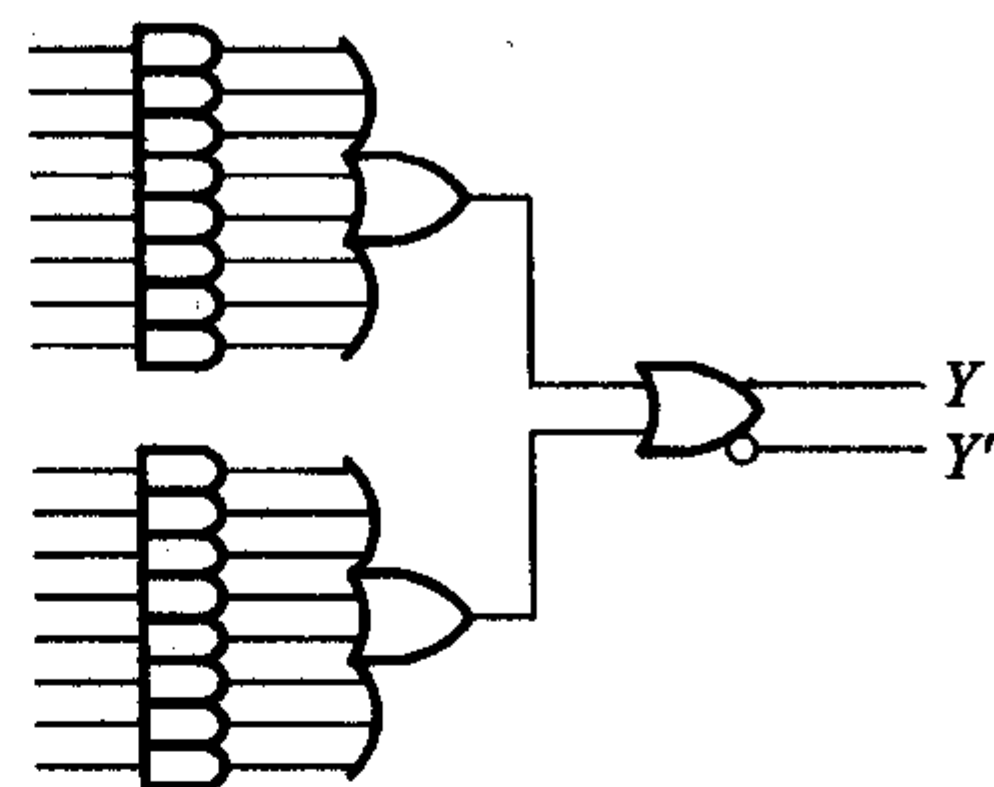


图 8.3.3 具有互补输出的专用输出结构

### 二、可编程输入/输出结构

可编程输入/输出结构（简称可编程 I/O 结构）的电路结构图如图 8.3.4 所示。它的输出端是一个具有可编程控制端的三态缓冲器，控制端由与逻辑阵列的一个乘积项给出。同时，输出端又经过一个互补输出的缓冲器反馈到与逻辑阵列上。

在图 8.3.4 所示的编程情况下，当  $I_1 = I_2 = 1$  时，上面一个缓冲器  $G_1$  的控制端  $C_1 = 1$ ， $I/O_1$  处于输出工作状态。对下面一个缓冲器  $G_2$  而言，它的控制端  $C_2$  恒等于零， $G_2$  处于高阻态，因此可以将  $I/O_2$  作为变量输入端使用。这时加到  $I/O_2$  上的输入信号经  $G_3$  接到与逻辑阵列的输入端（图中的第 6、7 列）。

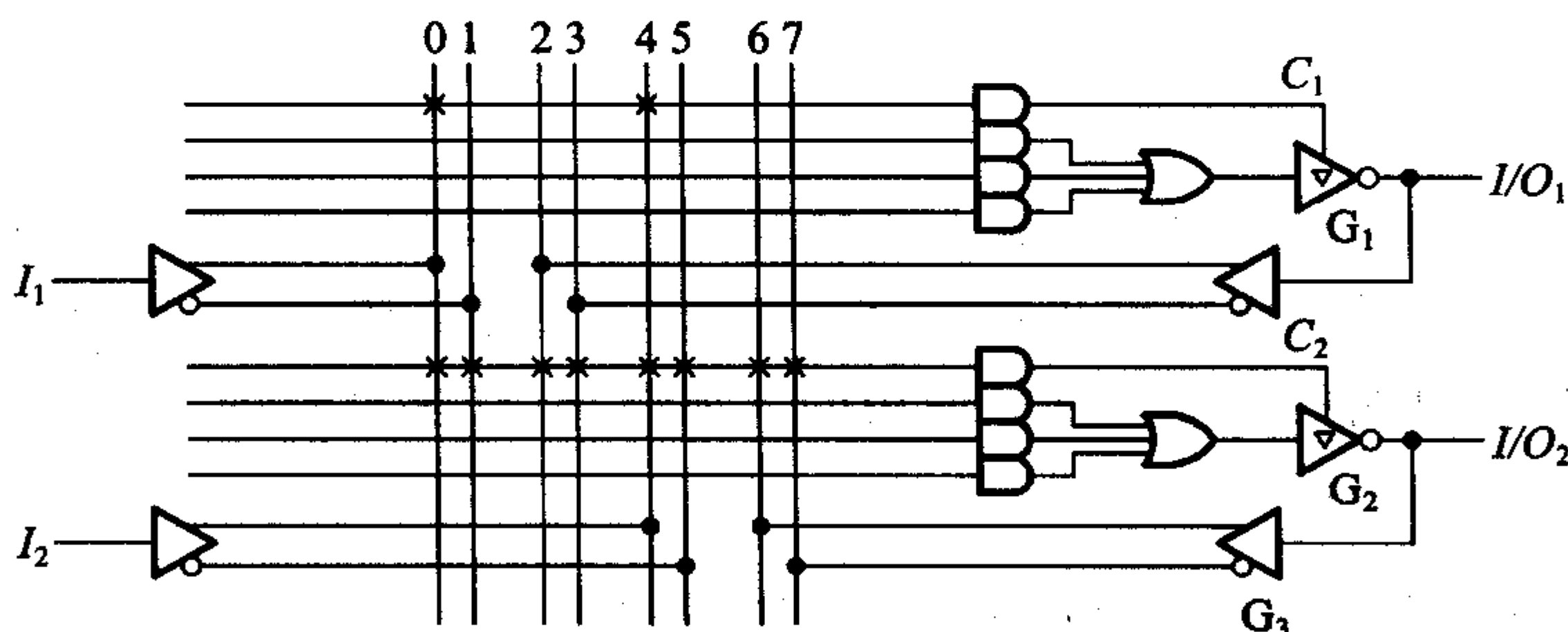


图 8.3.4 PAL 的可编程输入/输出结构

属于这种输出结构的器件有 PAL16L8、PAL20L10 等。

在有些可编程 I/O 结构的 PAL 器件中，在与或逻辑阵列的输出和三态缓冲器之间还设置有可编程的异或门，如图 8.3.5 所示。通过对异或门一个可编程输入端的编程可以控制输出的极性。当  $XOR = 0$  时， $Y$  与  $S$  同相；而当  $XOR = 1$  时， $Y$  与  $S$  反相。在用 PAL 设计组合逻辑电路时经常会遇到求反函数的情况。例如，所设计的与或逻辑函数的乘积项数多于

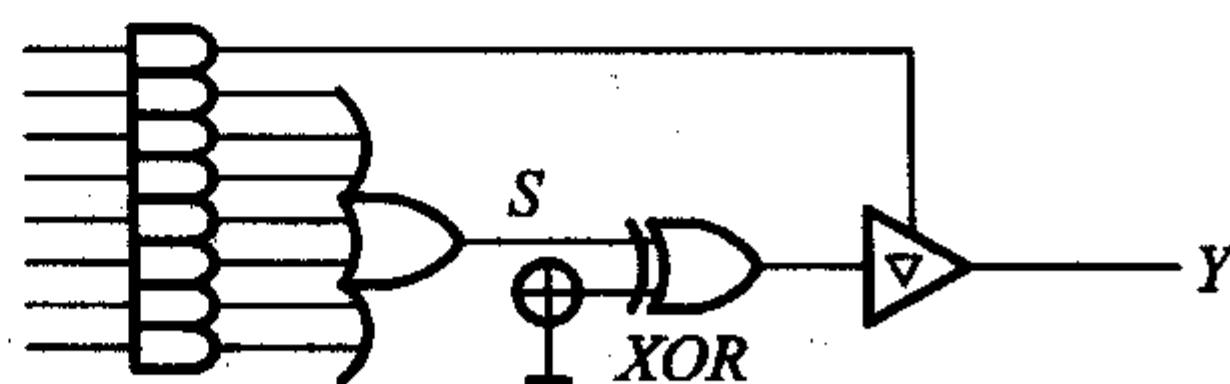


图 8.3.5 带有异或门的可编程输入/输出结构

或门的输入端个数,而它的反函数包含的乘积项数少于或门的输入端数目时,可以先通过对与逻辑阵列编程产生反函数,然后再利用对异或门编程求反,最后得到所求的函数。

### 三、寄存器输出结构

PAL 的寄存器输出结构如图 8.3.6 所示,它在输出三态缓冲器和与-或逻辑阵列的输出之间串进了由  $D$  触发器组成的寄存器。同时,触发器的状态又经过互补输出的缓冲器反馈到与逻辑阵列的输入端。

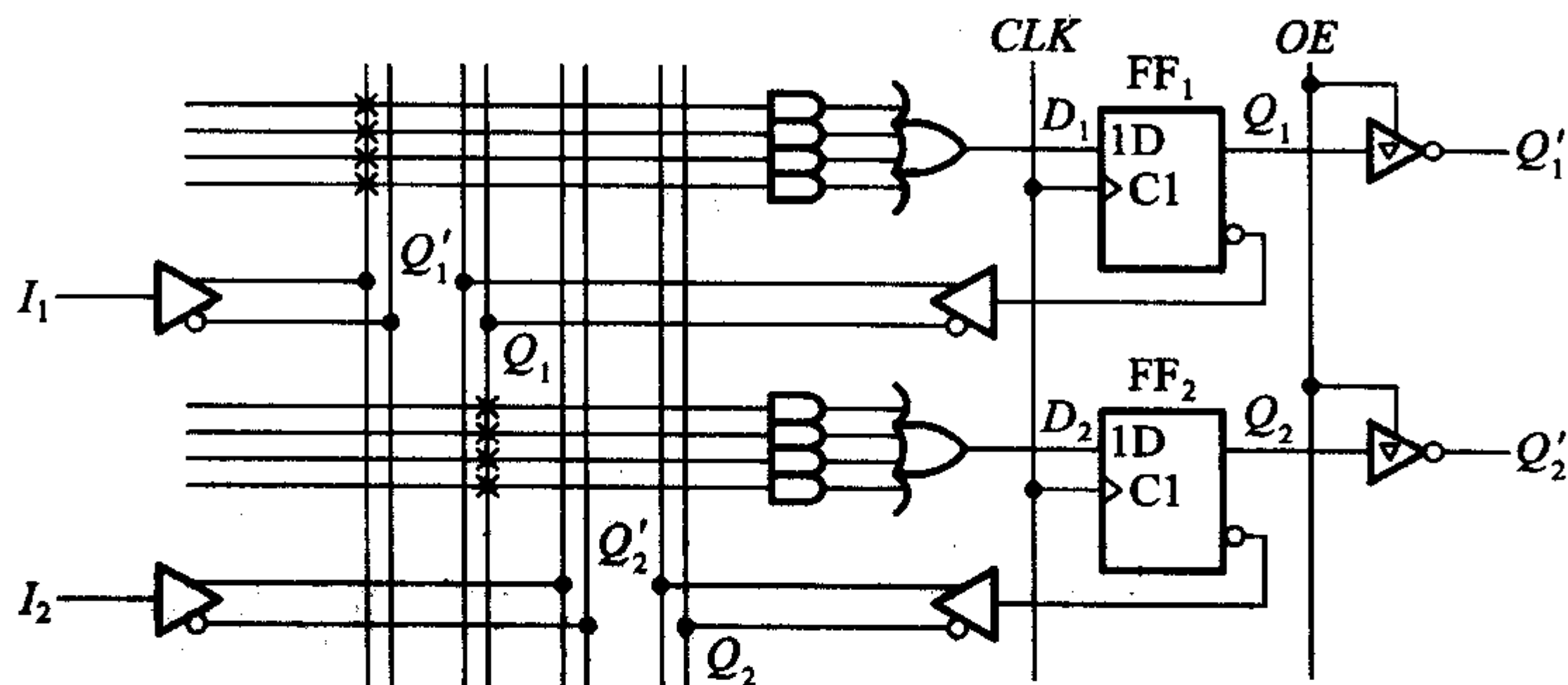


图 8.3.6 PAL 的寄存器输出结构

利用这种输出结构不仅可以存储与-或逻辑阵列输出的状态,而且能很方便地组成各种时序逻辑电路。例如,将与逻辑阵列按图 8.3.6 所示的情况编程,则得到  $D_1 = I_1, D_2 = Q_1$ 。因此,两个触发器和与-或逻辑阵列一起组成了移位寄存器。

属于寄存器输出结构的 PAL 器件有 PAL16R4、PAL16R6、PAL16R8 等。

### 四、异或输出结构

异或输出结构型 PAL 的电路结构如图 8.3.7 所示。它的电路结构与寄存器输出结构类似,只是在与-或逻辑阵列的输出端又增设了异或门。

利用这种电路结构不仅便于对与-或逻辑阵列输出的函数求反,还可以实现对寄存器状态进行保持的操作。例如,在图 8.3.7 所示的编程情况下,当  $I_1 = 0$  时  $D_1 = Q_1$ ,所以  $Q_1^* = Q_1$ ,在时钟信号到来时触发器的状态保持不变。当  $I_1 = 1$  时  $D_1 = Q_1'$ ,所以  $Q_1^* = Q_1'$ 。

对于下面一个触发器,当  $I_1 = 0$  时  $D_2 = Y_2 = Q_1 I_2 + Q_1' I_2'$ ;而当  $I_1 = 1$  时  $D_2 = Y_2' = (Q_1 I_2 + Q_1' I_2')'$ ;即得到  $Y_2$  的反函数。

属于这种输出结构的器件有 PAL20X4、PAL20X8 及 PAL20X10 等。

### 五、运算选通反馈结构

在异或输出结构的基础上再增加一组反馈逻辑电路,就构成了如图 8.3.8

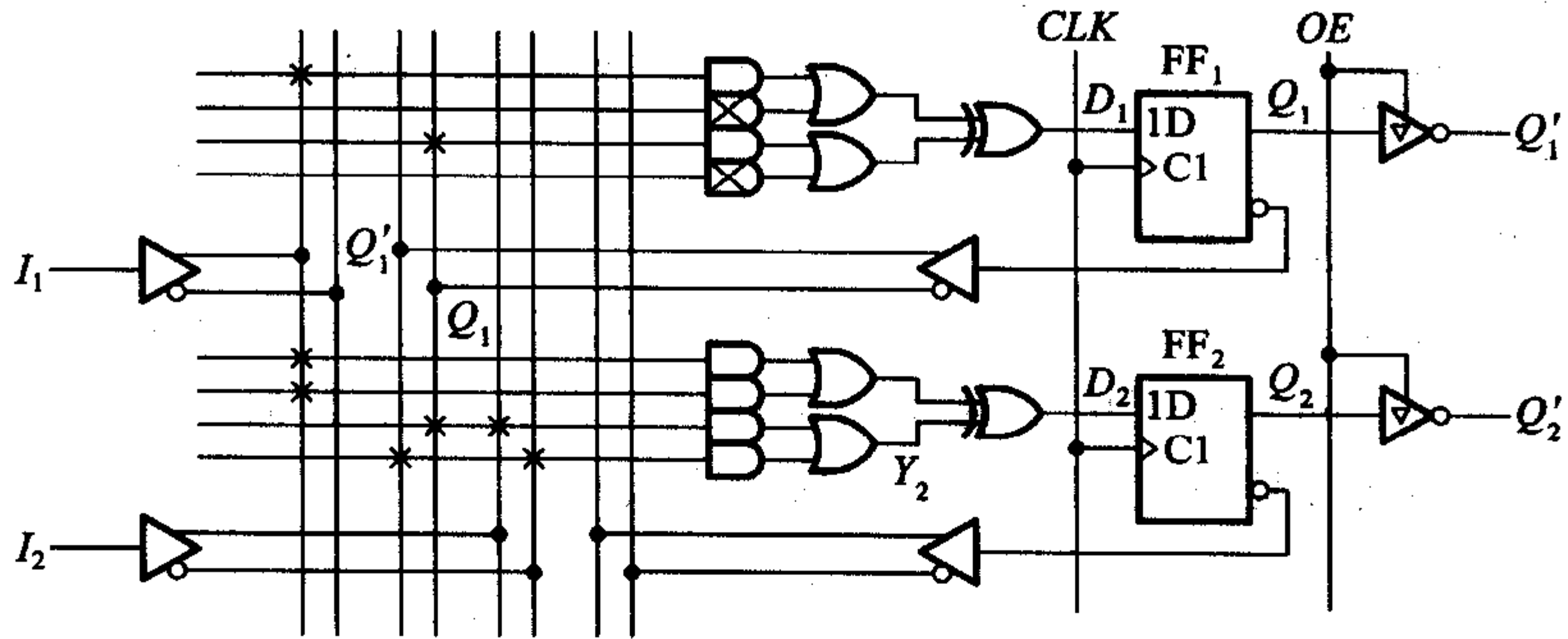


图 8.3.7 PAL 的异或输出结构

所示的运算选通反馈结构。

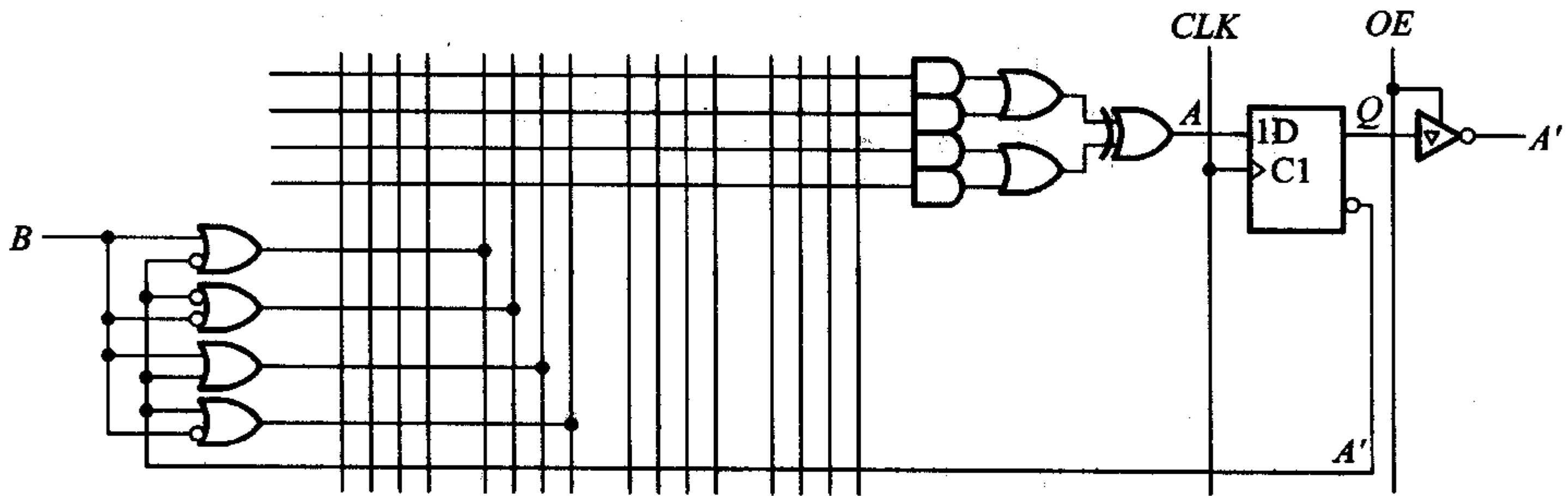


图 8.3.8 PAL 的运算选通反馈结构

反馈选通电路分别给出了输入变量  $B$  和反馈变量  $A$  产生的  $(A+B)$ 、 $(A+B')$ 、 $(A'+B)$ 、 $(A'+B')$  4 个反馈量,并接至与逻辑阵列的输入端。通过对与逻辑阵列的编程,能产生  $A$  和  $B$  的 16 种算术运算和逻辑运算的结果。图 8.3.9 给出了产生这 16 种运算的编程情况。

属于运算选通反馈结构的器件有 PAL16X4、PAL16A4 等。

### 8.3.3 PAL 的应用举例

在这一小节里我们将通过两个例子进一步深入说明 PAL 的工作原理以及用它设计逻辑电路的过程。在实际的设计工作中,只要把设计任务抽象成逻辑函数形式以后,余下的工作都是使用 EDA 软件在计算机上完成的,而不需要像下面的例子中那样用手工方法完成。

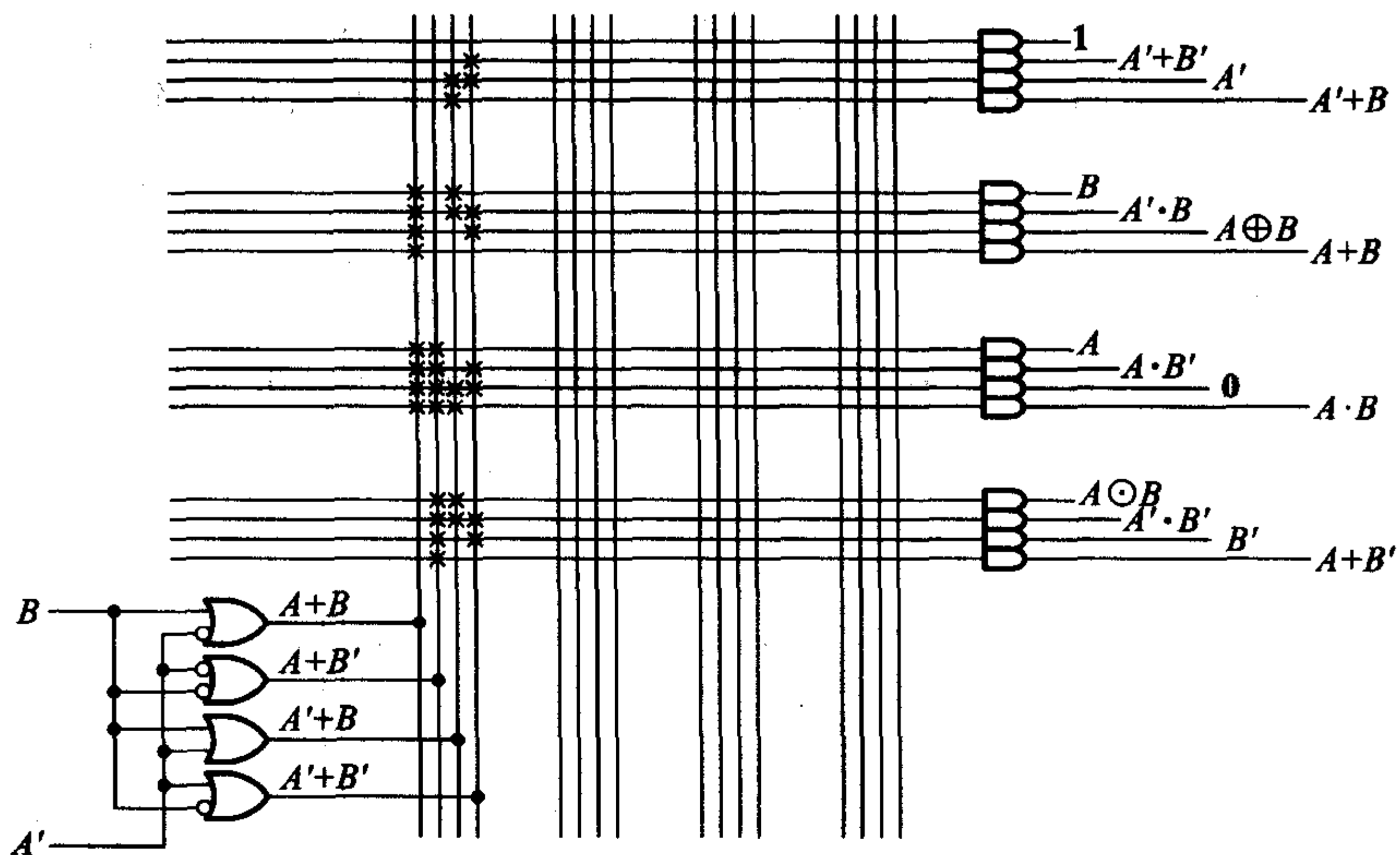


图 8.3.9 产生 16 种算术、逻辑运算的编程情况

【例 8.3.1】用 PAL 器件设计一个数值判别电路。要求判断 4 位二进制数  $DCBA$  的大小属于  $0 \sim 5$ 、 $6 \sim 10$ 、 $11 \sim 15$  三个区间的哪一个之内。

解：若以  $Y_0 = 1$  表示  $DCBA$  的数值在  $0 \sim 5$  之间；以  $Y_1 = 1$  表示  $DCBA$  的数值在  $6 \sim 10$  之间；以  $Y_2 = 1$  表示  $DCBA$  的数值在  $11 \sim 15$  之间，则得到表 8.3.1 中的函数真值表。

表 8.3.1 例 8.3.1 的函数真值表

十进制数	二进制数				$Y_0$	$Y_1$	$Y_2$	十进制数	二进制数				$Y_0$	$Y_1$	$Y_2$
	$D$	$C$	$B$	$A$					$D$	$C$	$B$	$A$			
0	0	0	0	0	1	0	0	8	1	0	0	0	1	0	
1	0	0	0	1	1	0	0	9	1	0	0	1	0	1	0
2	0	0	1	0	1	0	0	10	1	0	1	0	0	1	0
3	0	0	1	1	1	0	0	11	1	0	1	1	0	0	1
4	0	1	0	0	1	0	0	12	1	1	0	0	0	0	1
5	0	1	0	1	1	0	0	13	1	1	0	1	0	0	1
6	0	1	1	0	0	1	0	14	1	1	1	0	0	0	1
7	0	1	1	1	0	1	0	15	1	1	1	1	0	0	1

从真值表写出  $Y_0$ 、 $Y_1$ 、 $Y_2$  的逻辑函数式，经化简后得到

$$\begin{cases} Y_0 = D'C' + D'B' \\ Y_1 = D'CB + DC'B' + DC'A' \\ Y_2 = DC + DBA \end{cases} \quad (8.3.2)$$

这是一组有 4 个输入变量、3 个输出的组合逻辑函数。如果用一片 PAL 器件产生这一组逻辑函数,就必须选用有 4 个以上输入端和 3 个以上输出端的器件。而且由式(8.3.2)可以看到,至少还应当有一个输出包含 3 个以上乘积项。

根据上述理由,选用 PAL14H4 比较合适。PAL14H4 有 14 个输入端、4 个输出端。每个输出包含 4 个乘积项。图 8.3.10 是按照式(8.3.2)编程后的逻辑图。

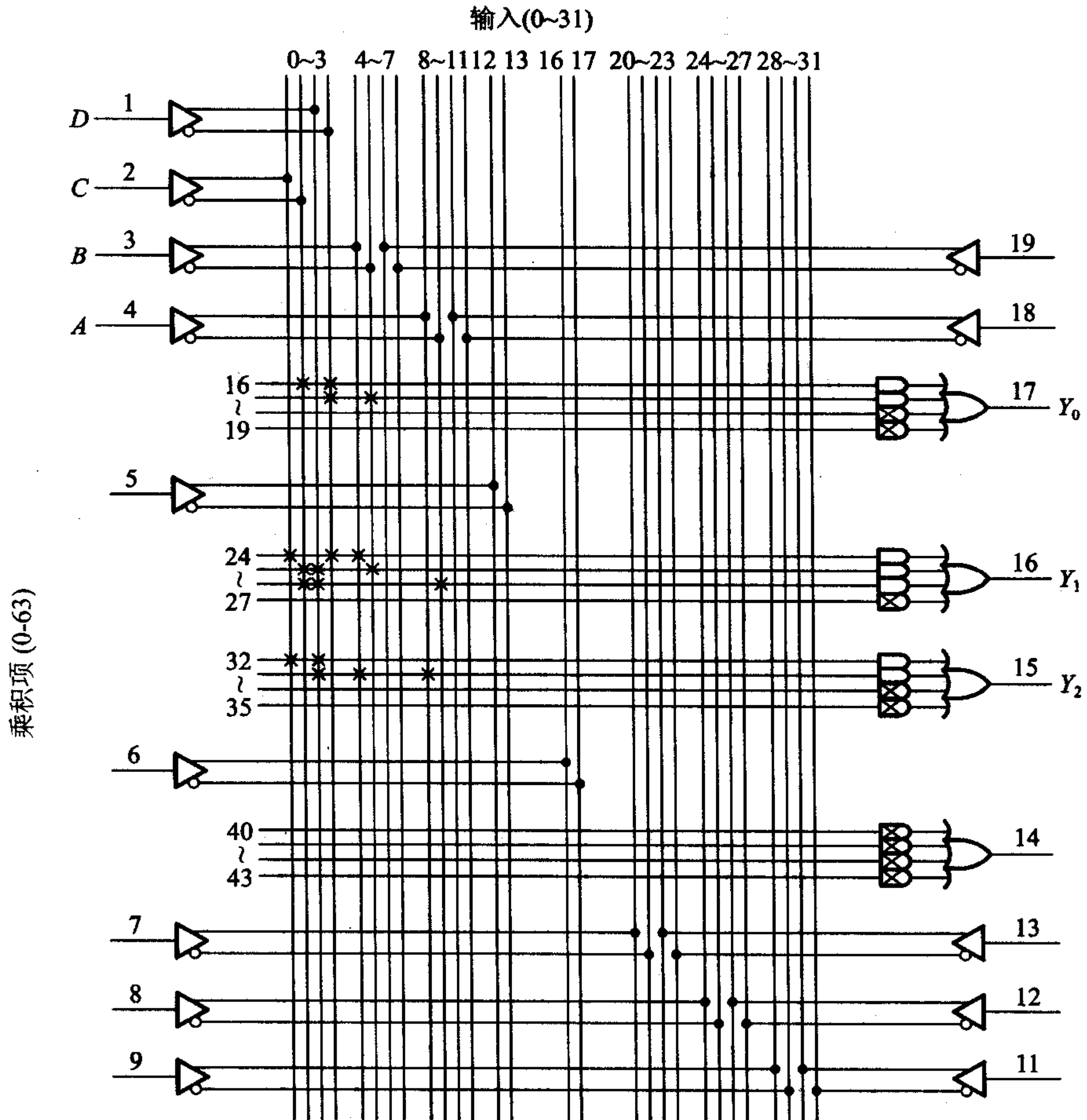


图 8.3.10 PAL14H4 按式(8.3.2)编程后的逻辑图

图中画“×”的与门表示编程时没有利用。由于未编程时这些与门的所有输入端均有熔丝与列线相连,所以它们的输出恒为 0。为简化作图起见,所有输入端交叉点上的“×”就不画了,而用与门符号里面的“×”来代替。

**【例 8.3.2】** 用 PAL 设计一个 4 位格雷码计数器,并要求所设计的计数器具有置零和对输出进行三态控制的功能。

**解:** 根据格雷码的计数顺序可以列出在一系列时钟信号作用下 4 位格雷码的变化顺序表,如表 8.3.2 所示。

如果用 PAL 器件设计这个计数器,则所用的器件中至少应包含 4 个触发器和相应的与-或逻辑阵列。从手册上可以查到, PAL16R4 可以满足上述要求。由图 8.3.12 可见, PAL16R4 的电路中有 4 个触发器,而且触发器的输出端设置有三态缓冲器。它有 8 个变量输入端,除了 4 个寄存器输出端以外还有 4 个可编程 I/O 端。

因为输出缓冲器是反相器,所以 4 个触发器  $Q$  端的状态与表 8.3.2 中  $Y$  的状态相反。因此,  $Q_3Q_2Q_1Q_0$  的状态转换顺序应如表 8.3.3 所示。这也就是  $Q_3Q_2Q_1Q_0$  的状态转换表。

表 8.3.2 4 位格雷码的计数顺序表

CLK	$Y_3$	$Y_2$	$Y_1$	$Y_0$	$C$ (进位)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	1	0
3	0	0	1	0	0
4	0	1	1	0	0
5	0	1	1	1	0
6	0	1	0	1	0
7	0	1	0	0	0
8	1	1	0	0	0
9	1	1	0	1	0
10	1	1	1	1	0
11	1	1	1	0	0
12	1	0	1	0	0
13	1	0	1	1	0
14	1	0	0	1	0
15	1	0	0	0	1
16	0	0	0	0	0

表 8.3.3 PAL 器件中触发器的状态转换表

CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$C'$ (进位)
0	1	1	1	1	1
1	1	1	1	0	1
2	1	1	0	0	1
3	1	1	0	1	1
4	1	0	0	1	1
5	1	0	0	0	1
6	1	0	1	0	1
7	1	0	1	1	1
8	0	0	1	1	1
9	0	0	1	0	1
10	0	0	0	0	1
11	0	0	0	1	1
12	0	1	0	1	1
13	0	1	0	0	1
14	0	1	1	0	1
15	0	1	1	1	0
16	1	1	1	1	1

根据表 8.3.3 画出 4 个触发器次态的卡诺图,如图 8.3.11 所示。经化简后得到各个触发器的状态方程为

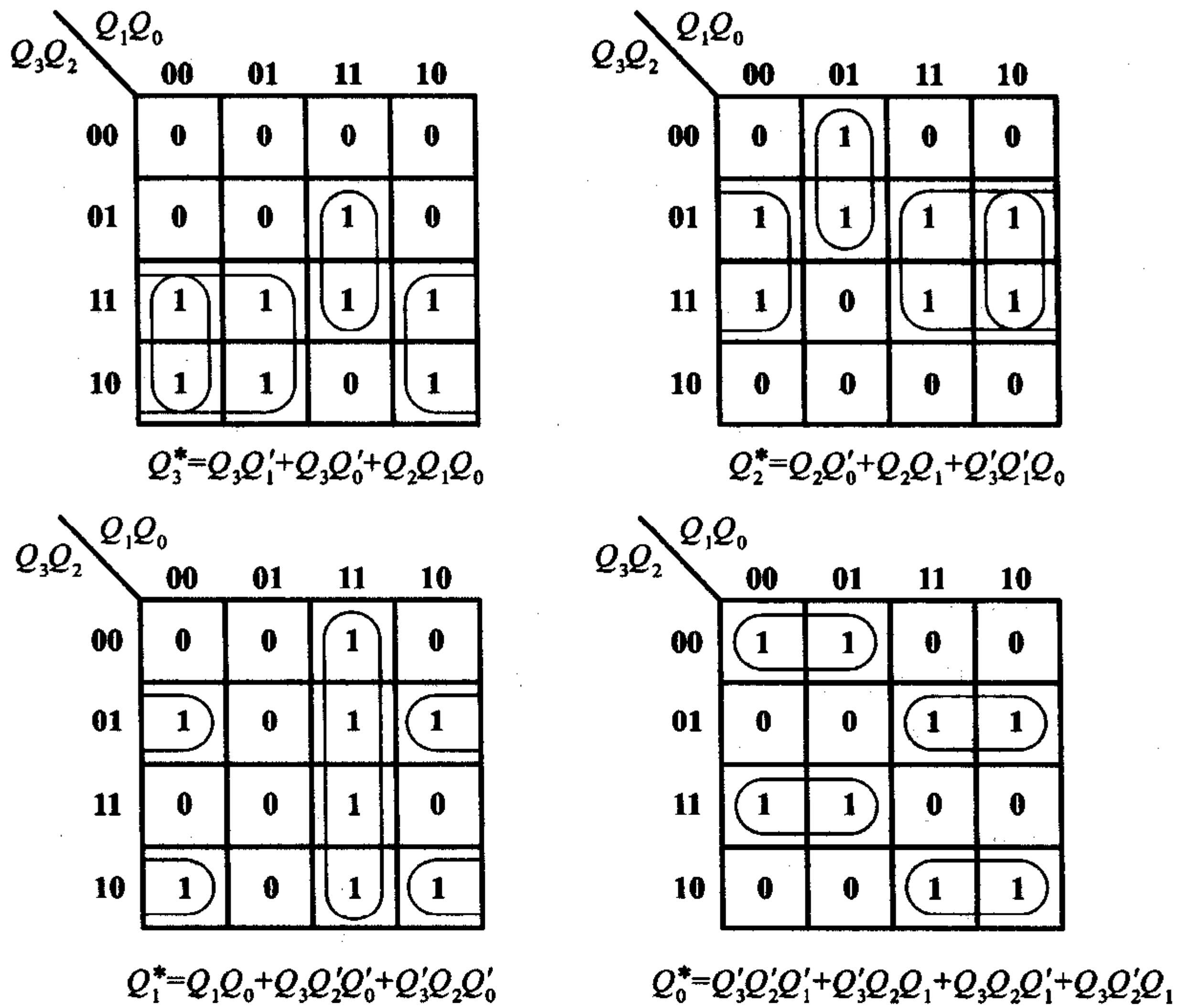


图 8.3.11 例 8.3.2 输出状态的卡诺图

$$\begin{cases}
 Q_3^* = Q_3 Q_1' + Q_3 Q_0' + Q_2 Q_1 Q_0 \\
 Q_2^* = Q_2 Q_0' + Q_2 Q_1 + Q_3' Q_1' Q_0 \\
 Q_1^* = Q_1 Q_0 + Q_3 Q_2' Q_0' + Q_3' Q_2 Q_0' \\
 Q_0^* = Q_3' Q_2' Q_1' + Q_3' Q_2 Q_1 + Q_3 Q_2 Q_1' + Q_3 Q_2' Q_1
 \end{cases} \quad (8.3.3)$$

从上式即可写出每个触发器的驱动方程,即  $D$  端的逻辑函数式。同时,考虑到要求具有置零功能,故应在驱动方程中加入一项  $R$ 。当置零输入信号  $R=1$  时,在时钟信号到达后将所有的触发器置 1,反相后的输出得到  $Y_3 Y_2 Y_1 Y_0 = 0000$ 。于是得到驱动方程为

$$\begin{cases}
 D_3 = Q_3 Q_1' + Q_3 Q_0' + Q_2 Q_1 Q_0 + R \\
 D_2 = Q_2 Q_0' + Q_2 Q_1 + Q_3' Q_1' Q_0 + R \\
 D_1 = Q_1 Q_0 + Q_3 Q_2' Q_0' + Q_3' Q_2 Q_0' + R \\
 D_0 = Q_3' Q_2' Q_1' + Q_3' Q_2 Q_1 + Q_3 Q_2 Q_1' + Q_3 Q_2' Q_1 + R
 \end{cases} \quad (8.3.4)$$

进位输出信号的逻辑函数式为

$$C' = (Q_3' Q_2 Q_1 Q_0)' \quad (8.3.5)$$

按照式(8.3.4)和式(8.3.5)编程后 PAL16R4 的逻辑图如图 8.3.12 所示。

图中1脚接时钟输入,亦即计数输入;11脚接输出缓冲器的三态控制信号 $OE'$ ;2脚接置零信号 $R$ ,正常计数时 $R$ 应处于低电平;17、16、15、14脚分别为输出 $Y_3$ 、 $Y_2$ 、 $Y_1$ 、 $Y_0$ ;18脚为 $C'$ 输出端。若从 $Y_3Y_2Y_1Y_0 = 0000$ 开始计数,则输入16个时钟信号时 $C'$ 从低电平跳回到高电平,给出一个正跳变的进位输出信号。

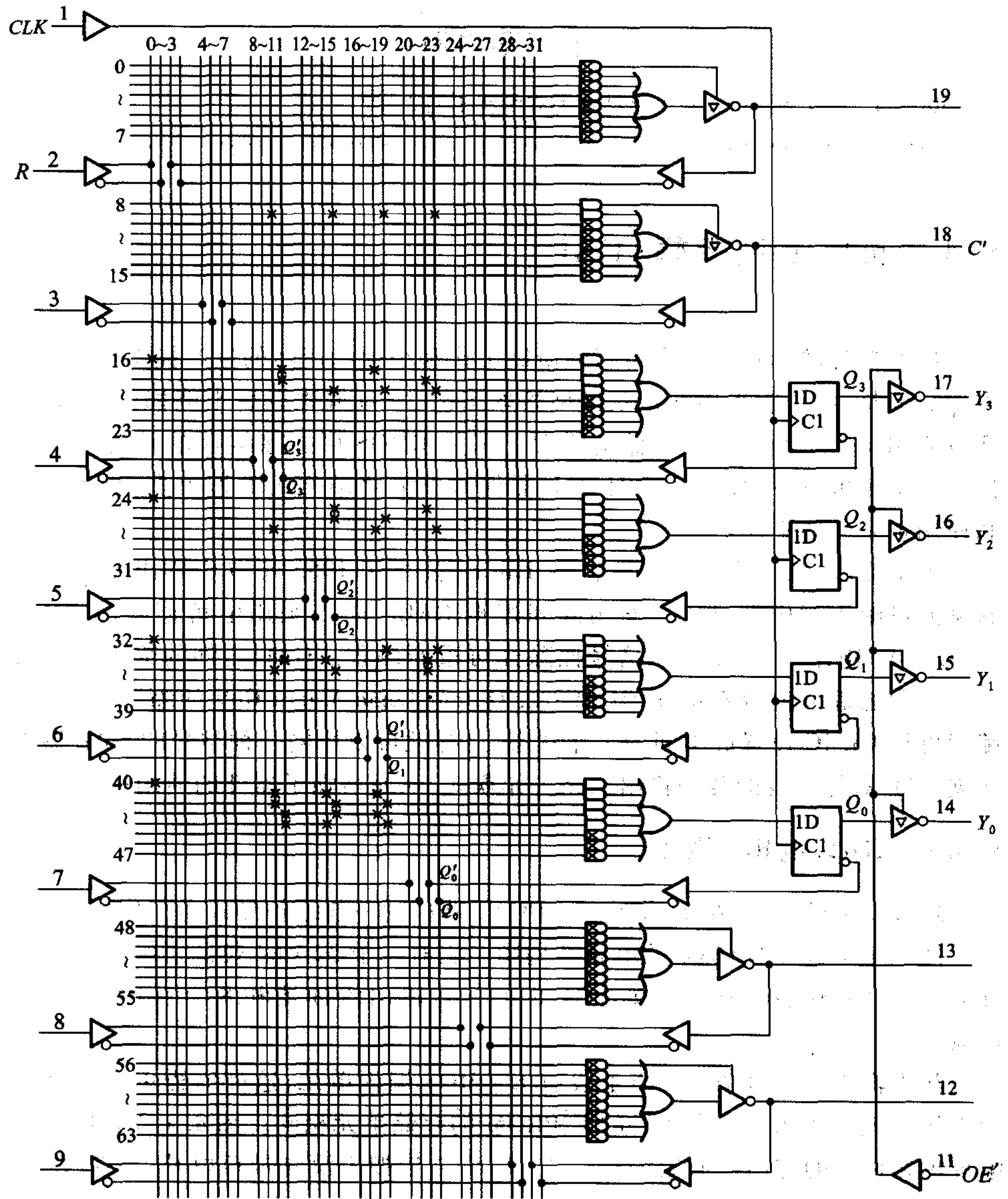


图 8.3.12 例 8.3.2 中编程后的 PAL16R4 的逻辑图



以上所讲的设计工作都可以在开发系统上自动进行。只要按照编程软件规定的格式输入逻辑真值表,后面的工作都由计算机去完成。

### 复习思考题

R8.3.1 PAL 和 FPLA 在电路结构上有哪些不同之处?

## 8.4 通用阵列逻辑(GAL)

PAL 器件的出现为数字电路的研制工作和小批量产品的生产提供了很大的方便。但是,由于它采用的是双极型熔丝工艺,一旦编程以后不能修改,因而不适应研制工作中经常修改电路的需要。采用 CMOS 可擦除编程单元的 PAL 器件克服了不可改写的缺点,然而 PAL 器件输出电路结构的类型繁多,仍给设计和使用带来一些不便。

为了克服 PAL 器件存在的缺点,Lattice 公司于 1985 年首先推出了另一种新型的可编程逻辑器件——通用阵列逻辑 GAL。GAL 采用电可擦除的 CMOS ( $E^2$ CMOS) 制作,可以用电压信号擦除并可重新编程。GAL 器件的输出端设置了可编程的输出逻辑宏单元 OLMC(系 Output Logic Macro Cell 的缩写)。通过编程可将 OLMC 设置成不同的工作状态,这样就可以用同一种型号的 GAL 器件实现 PAL 器件所有的各种输出电路工作模式,从而增强了器件的通用性。

### 8.4.1 GAL 的电路结构

现以常见的 GAL16V8 为例,介绍 GAL 器件的一般结构形式和工作原理。

图 8.4.1 是 GAL16V8 的电路结构图。它有一个  $32 \times 64$  位的可编程与逻辑阵列,8 个 OLMC,10 个输入缓冲器、8 个三态输出缓冲器和 8 个反馈/输入缓冲器。

与逻辑阵列的每个交叉点上设有  $E^2$ CMOS 编程单元,这种编程单元的结构和工作原理与 7.2.3 节中所讲的  $E^2$ PROM 的存储单元相同。图 8.4.2 是用三个编程单元构成的与门。假定编程后  $T_2$ 、 $T_4$  的浮置栅上没有带负电荷,而  $T_6$  的浮置栅上存储了足够的负电荷,则  $T_2$ 、 $T_4$  导通而  $T_6$  截止。因此,A、B 和 P 之间是编程连接,而 C 和 P 之间没有连接,于是得到  $P = A \cdot B$ 。

组成或逻辑阵列的 8 个或门分别包含于 8 个 OLMC 中,它们和与逻辑阵列

