

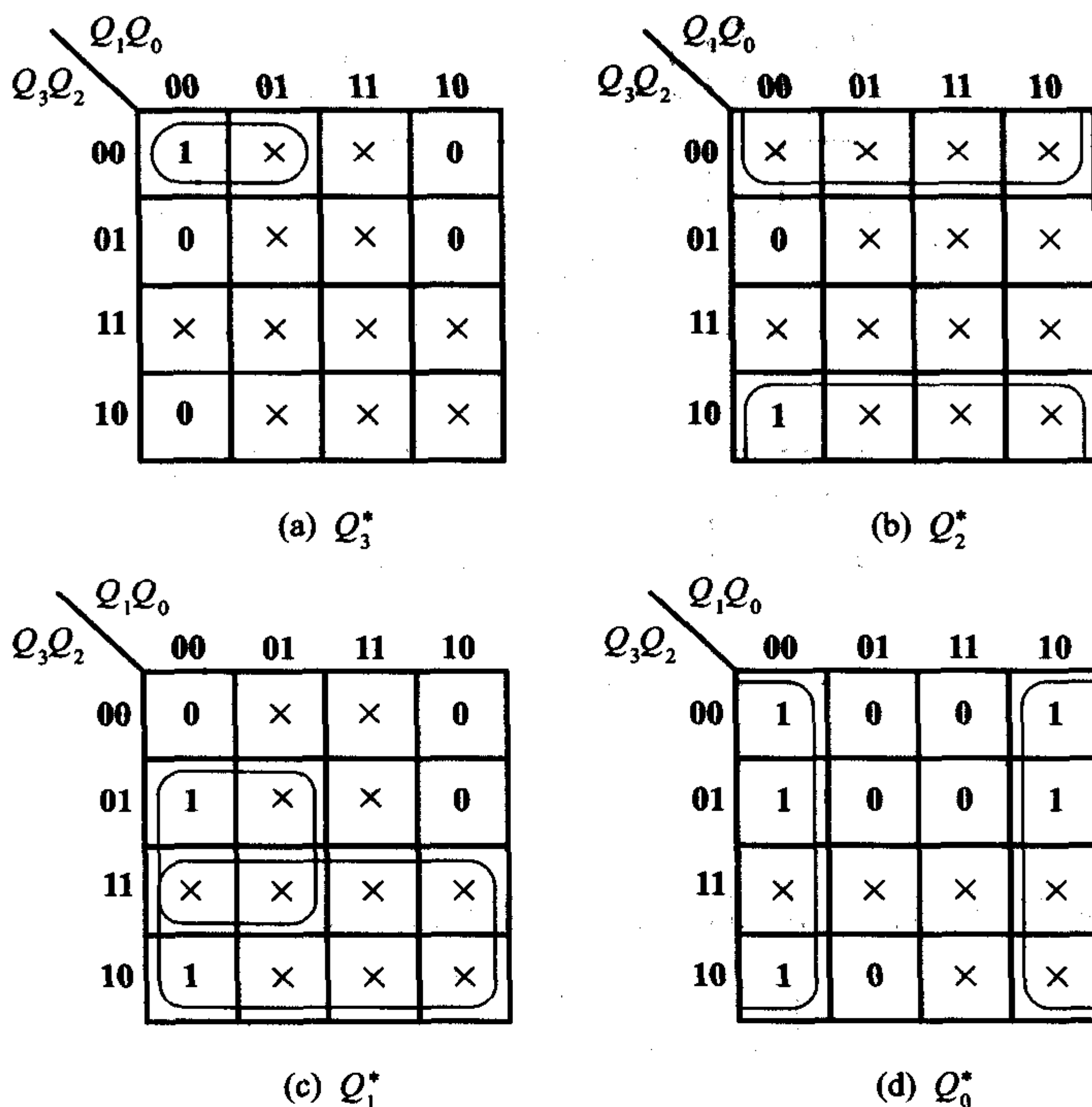


图 6.4.33 图 6.4.32 卡诺图的分解

$$\begin{cases}
 Q_3^* = [(Q_2'Q_1')Q_3' + 1' \cdot Q_3] \cdot clk_3 \\
 Q_2^* = [1 \cdot Q_2' + 1' \cdot Q_2] \cdot clk_2 \\
 Q_1^* = [Q_3(Q_1 + Q_1') + Q_2Q_1'] \cdot clk_1 \\
 \quad = [(Q_3 + Q_2)Q_1' + Q_3Q_1] \cdot clk_1 = [(Q_3'Q_2')'Q_1' + 1' \cdot Q_1] \cdot clk_1 \\
 Q_0^* = [1 \cdot Q_0' + 1' \cdot Q_0] \cdot clk_0
 \end{cases}
 \quad (6.4.22)$$

因为电路正常工作时不会出现 $Q_3Q_1 = 1$ 的情况, 所以在 Q_1^* 的方程式中删去了这一项。

从式(6.4.22)得到每个触发器应有的驱动方程为

$$\begin{cases}
 J_3 = Q_2'Q_1', & K_3 = 1 \\
 J_2 = K_2 = 1 \\
 J_1 = (Q_3'Q_2')', & K_1 = 1 \\
 J_0 = K_0 = 1
 \end{cases}
 \quad (6.4.23)$$

根据状态转换表画出的输出 B 的卡诺图如图 6.4.34 所示。由图得到

$Q_3Q_2 \backslash Q_1Q_0$		00	01	11	10
		00	01	11	10
00		1	0	0	0
01		0	0	0	0
11		×	×	×	×
10		0	0	×	×

图 6.4.34 例 6.4.6 电路输出的卡诺图

$$B = Q'_3 Q'_2 Q'_1 Q'_0 \quad (6.4.24)$$

按照式(6.4.23)和式(6.4.24)画出的逻辑图如图 6.4.35 所示。

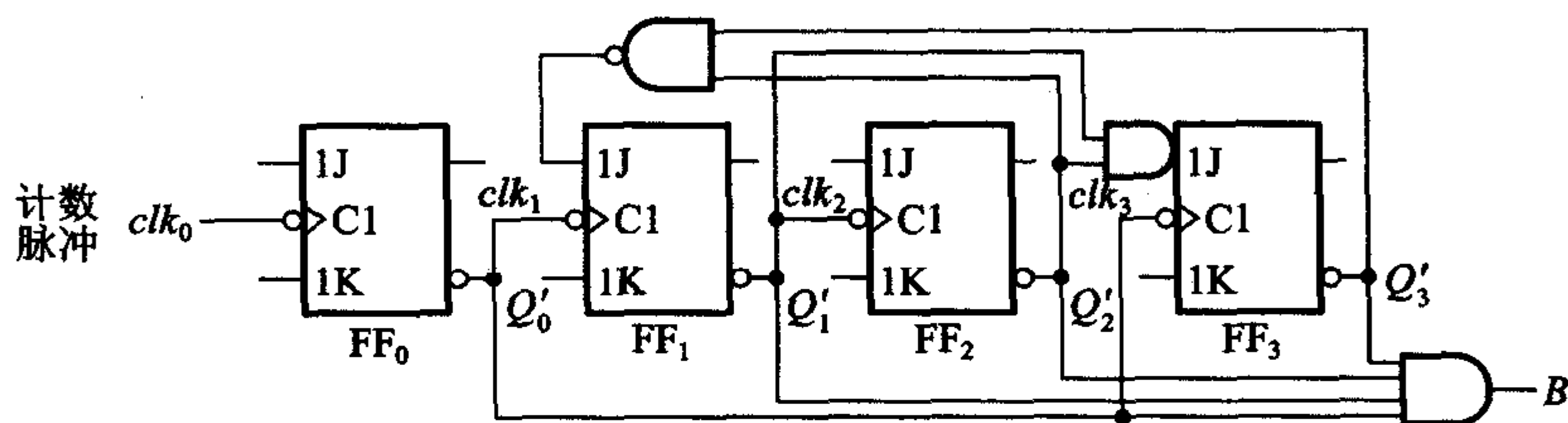


图 6.4.35 异步十进制减法计数器的逻辑图

最后需要检查一下设计的电路能否自启动。将 1010 ~ 1111 这 6 个无效状态分别代入状态方程求其次态,结果表明电路是可以自启动的。完整的电路状态转换图如图 6.4.36 所示。

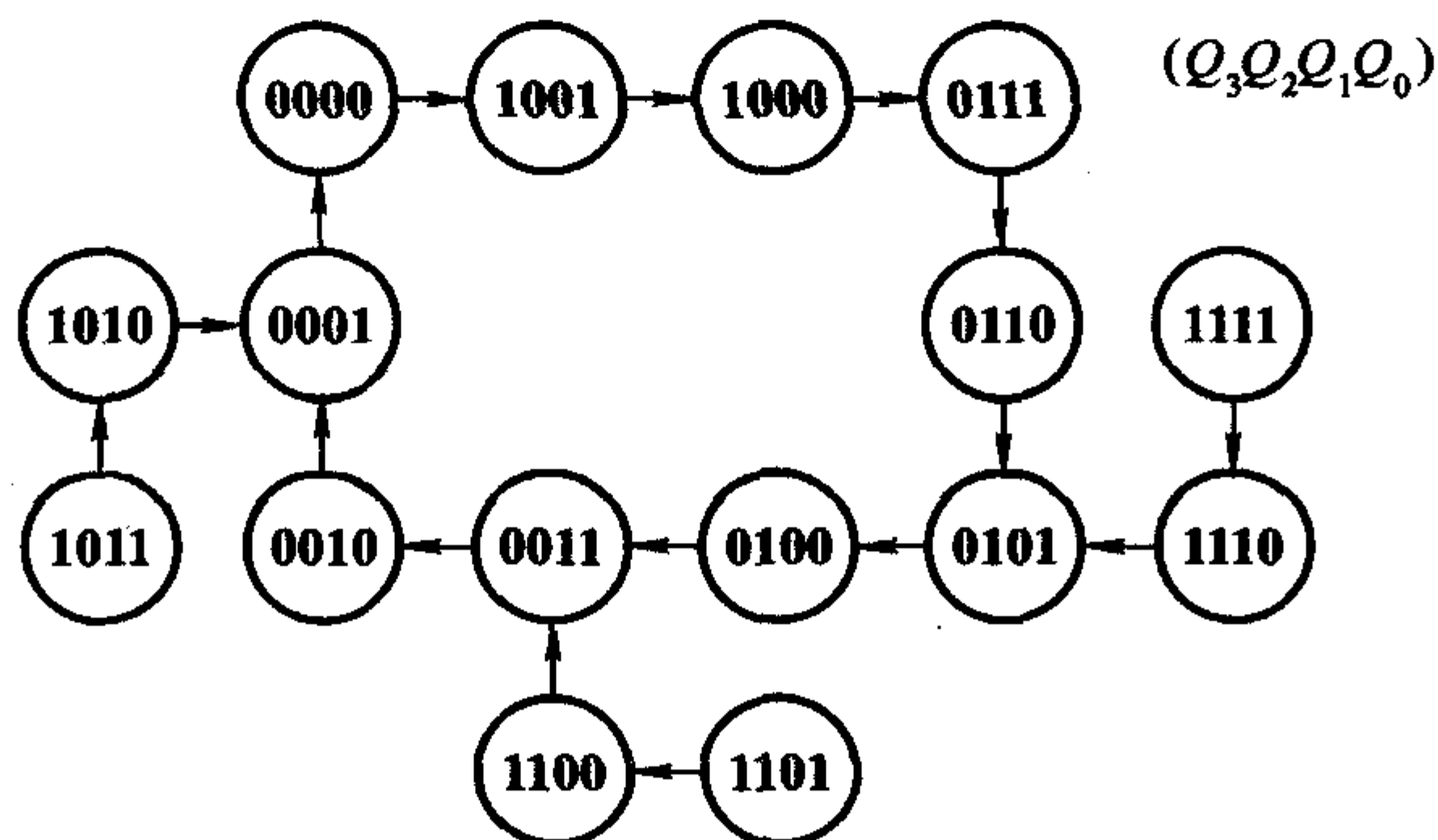


图 6.4.36 图 6.4.35 电路的状态转换图

* 6.4.4 复杂时序逻辑电路的设计

在一些复杂的时序电路中,往往会包含为数众多的输入变量、输出变量、电路状态,而且存在多种状态循环和需要完成各种逻辑运算。这时已难于用一组状态方程、驱动方程和输出方程描述整个电路的逻辑功能了,因此简单地套用前面讲过的设计方法显然已经行不通了。

在这种情况下,通常采用层次化结构设计方法,或者称为模块化设计方法。层次化结构设计方法有“自顶向下”(Top-down)和“自底向上”(Bottom-up)两种做法。采用自顶向下的做法时,首先需要将所设计电路的功能逐级划分为更简单的功能模块,直到这些模块都能用简单的逻辑电路实现为止。这些简单的逻辑电路都可以用我们前面讲过的设计方法来设计。由于电路规模较大、功能复杂,所以经常需要有一个控制模块去协调各模块之间的操作。有人又将这类能明显地划分出控制模块的数字电路称为数字系统。其实究竟电路复杂到什么程度才叫做系统并无十分明确的界定。

由于自顶向下划分模块的过程中完全是从获得最佳电路性能出发的,并未考虑这些模块电路是否已经有成熟的设计存在了,所以必须从头设计每个模块电路,然后进行仿真和测试。在发现问题时,还需反复修改。即便如此,在做成硬件电路以后,也不能保证绝对不出现问题。

在采取自底向上的做法时,首先要考虑有哪些已有的、成熟的模块电路可以利用。这些模块电路可能是标准化的集成电路器件,也可能是经过验证的计算机软件。将电路划分为功能模块时,最后要划分到能利用这些已有的模块电路来实现为止。直接采用这些模块电路能大大减少设计的工作量。然而有时由于需要迁就已有的模块电路,这就会使电路的某些性能受到一些影响。另外,也不可能任何一种功能模块都有现成的成熟设计,因此多数情况下都采用自顶向下和自底向上相结合的方法,以求达到既能满足设计要求,又能提高设计速度、降低设计成本的目标。

【例 6.4.7】 设计一个自动售火柴机的逻辑电路。每次可投入一枚 1 分、2 分或 5 分的硬币,累计投入超过 8 分以后,输出一小盒火柴,同时找回多于 8 分的钱。

解: 首先仍然需要进行逻辑抽象,把要求实现的逻辑功能抽象为一个逻辑函数问题。取投币信号为输入变量,以 I_1 、 I_2 、 I_3 分别表示投入 1 分、2 分、5 分硬币的信号,同时以 Y 表示输出火柴的信号,以 Z_1 、 Z_2 、 Z_3 分别表示找回 1 分、2 分、4 分钱的信号。

考虑到各种可能的投币情况,电路可能出现多种状态和许多种可能的状态

循环,因此宜于采用层次化结构设计的方法进行设计。根据电路应实现的逻辑功能,可以将它划分为图 6.4.37 所示的模块电路。首先将表示整个电路功能的顶级模块划分为下一级的运算电路、输出电路、输入电路和控制电路四个模块。运算电路又可以划分为加法器和寄存器两个模块电路。

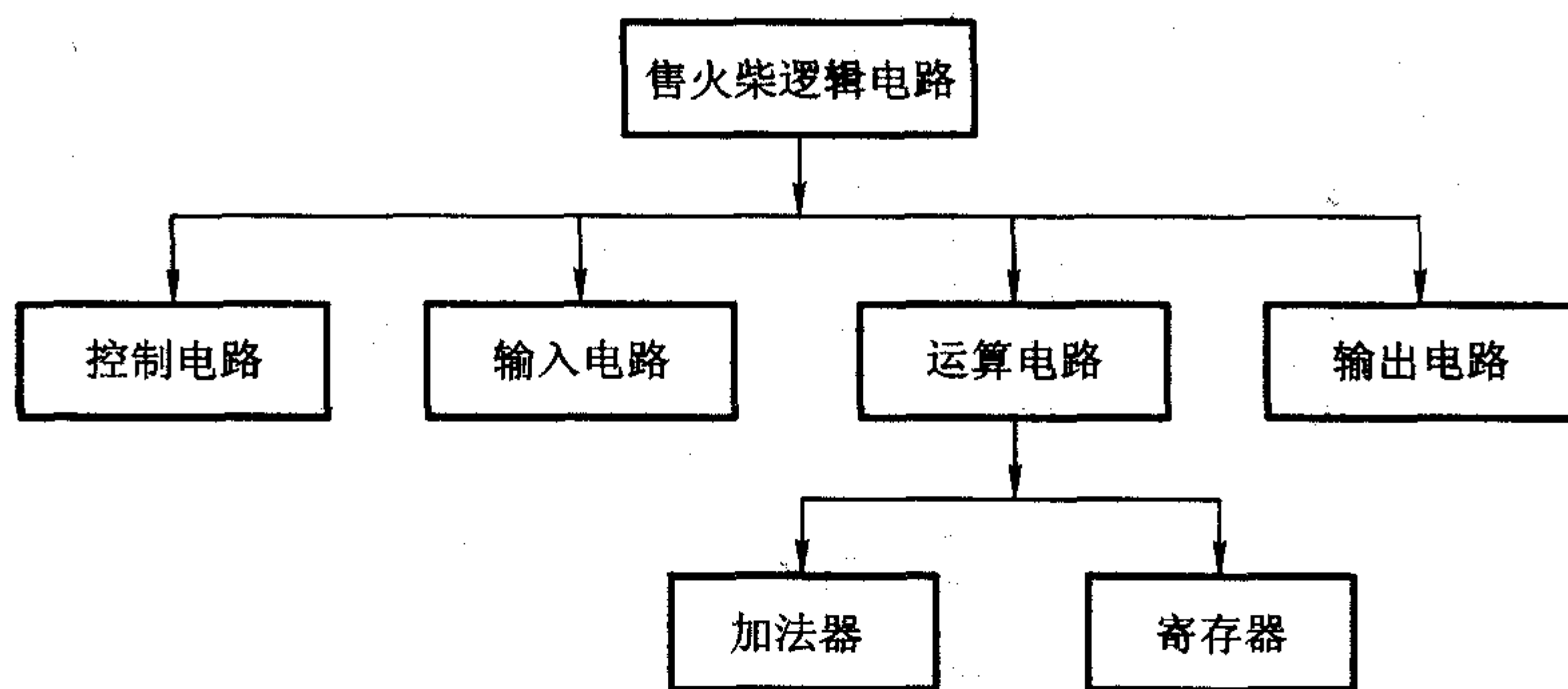


图 6.4.37 例 6.4.7 电路的模块划分

运算电路的功能是对每一次的输入做累加运算,所以它就是一个累加器。每当有投币信号到达时,将输入的钱数与寄存器中原有的钱数相加,并且将结果送回寄存器。当寄存器中的数大于、等于 8 时,输出电路给出输出火柴和找钱信号。输入电路中的整形电路用投币动作产生一定宽度的输入脉冲信号 I_1 、 I_2 和 I_3 ,并将它们转换为加法器输入的二进制数。控制电路产生累加器的操作信号 CLK 和寄存器的异步置 0 信号。

如果采用标准化的集成电路设计各个模块,就可以得到图 6.4.38 所示的逻辑图了。图中的 4 位超前进位加法器 74LS283 和 4 位寄存器 74LS175 组成了运算电路,门电路 $G_1 \sim G_4$ 和阻容电路 C_1 、 R_1 组成控制电路,门电路 $G_5 \sim G_7$ 组成输出电路,整形电路 $L_1 \sim L_3$ 和门电路 G_8 组成输入电路。

接通电源电压以后, $R_1 C_1$ 电路输出的瞬时高电平经过 G_1 反相后将寄存器置 0,电路处于准备状态。每当出现投币输入信号, I_1 、 I_2 或 I_3 等于 1 时,便有 001、010 或 101 加到加法器的输入端 B_2 、 B_1 、 B_0 上。与此同时, G_2 输出的低电平信号将经 G_3 反相后产生的 CLK 上升沿将加法器的输出存入寄存器中,完成一次累加操作。

当寄存器中的数大于、等于 8 时,寄存器的 Q_3 变为 1,使输出 $Y=1$,给出输出火柴的信号,同时在 Z_3 、 Z_2 、 Z_1 给出找钱信号。 I_1 、 I_2 或 I_3 回到 0 以后,反相器 G_4 输出高电平,经过 G_1 反相后将寄存器置 0。电路回到起始的准备状态。

为了确保 CLK 上升沿到达寄存器时寄存器数据输入端 $D_0 \sim D_3$ 的状态已经稳定地建立起来了,还可以在门 G_2 的输出端加入一个由电容 C_2 构成的延迟环节。这个电容的数值通常只需数十至数百皮法。

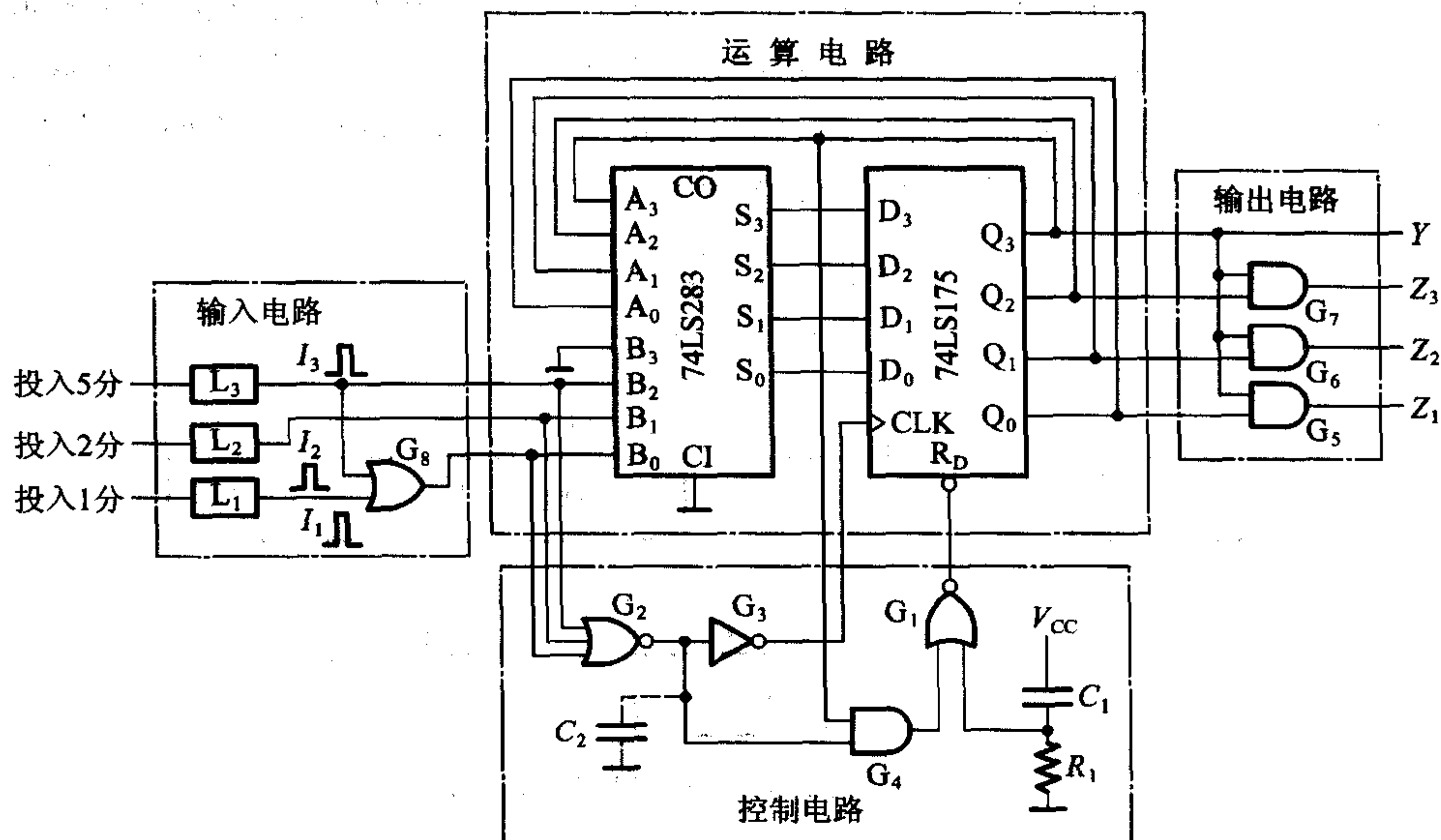


图 6.4.38 例 6.4.7 的电路

复习思考题

R6.4.4 “自顶向下”和“自底向上”的设计方法有什么区别？

6.5 时序逻辑电路中的竞争 - 冒险现象

因为时序逻辑电路通常都包含组合逻辑电路和存储电路两个组成部分,所以它的竞争 - 冒险现象也包含两个方面。

一方面是其中的组合逻辑电路部分可能发生的竞争 - 冒险现象。产生这种现象的原因已在 4.4.1 节中讲过。这种由于竞争而产生的尖峰脉冲并不影响组合逻辑电路的稳态输出,但如果它被存储电路中的触发器接收,就可能引起触发器的误翻转,造成整个时序电路的误动作,这种现象必须绝对避免。消除组合逻辑电路中竞争 - 冒险现象的方法已在 4.4.3 节中做了介绍,这里不再重复。

另一方面是存储电路(或者说是触发器)工作过程中发生的竞争 - 冒险现象,这也是时序电路所特有的一个问题。

在讨论触发器的动态特性时曾经指出,为了保证触发器可靠地翻转,输入信

号和时钟信号在时间配合上应满足一定的要求。然而当输入信号和时钟信号同时改变,而且途经不同路径到达同一触发器时,便产生了竞争。竞争的结果有可能导致触发器误动作,这种现象称为存储电路(或触发器)的竞争 - 冒险现象。

例如,在图 6.5.1 给出的八进制异步计数器电路中,就存在着这种存储电路的竞争 - 冒险现象。

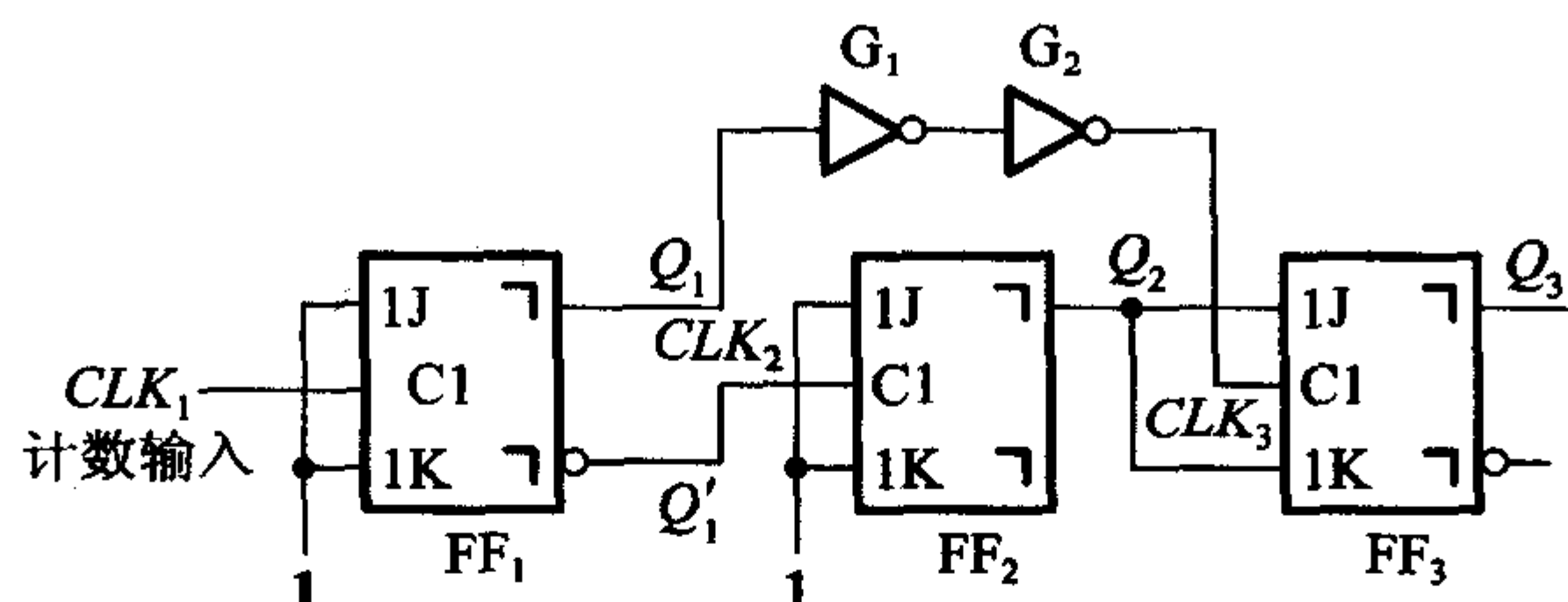


图 6.5.1 说明时序电路竞争 - 冒险现象的例子

计数器由 3 个主从 JK 触发器 FF_1 、 FF_2 、 FF_3 和两个反相器 G_1 、 G_2 组成。其中 FF_1 工作在 $J_1 = K_1 = 1$ 的状态,每次 CLK_1 的下降沿到达时它都要翻转。 FF_2 同样也工作在 $J_2 = K_2 = 1$ 的状态,所以每次 Q_1' 由高电平跳变为低电平时都要翻转。 FF_3 的情况要复杂一些。由于 CLK_3 取自 Q_1 (经过两级反相器延迟),而 $J_3 = K_3 = Q_2$, FF_2 的时钟信号又取自 Q_1' ,因而当 FF_1 由 0 变成 1 时 FF_3 的输入信号和时钟电平同时改变,导致了竞争 - 冒险现象的发生。

如果 Q_1 从 0 变成 1 时 Q_2 的变化首先完成, CLK_3 的上升沿随后才到,那么在 $CLK_3 = 1$ 的全部时间里 J_3 和 K_3 的状态将始终不变,就可以根据 CLK_3 下降沿到达时 Q_2 的状态决定 FF_3 是否该翻转。由此即可得到表 6.5.1(a) 的状态转换表和图 6.5.2 中以实线表示的状态转换图。显然这是一个八进制计数器。

表 6.5.1 图 6.5.1 电路的状态转换表

(a)			
计数 顺序	电路状态		
	Q_1	Q_2	Q_3
0	0	0	0
1	1	1	0
2	0	1	1
3	1	0	1
4	0	0	1
5	1	1	1
6	0	1	0
7	1	0	0
8	0	0	0

(b)			
计数 顺序	电路状态		
	Q_1	Q_2	Q_3
0	0	0	0
1	1	1	0
2	0	1	1
3	1	0	1
4	0	0	0

反之,如果 Q_1 从 0 变成 1 时 CLK_3 的上升沿首先到达 FF_3 , 而 Q_2 的变化在后, 则 $CLK_3 = 1$ 的期间里 J_3 和 K_3 的状态可能发生变化, 这就不能简单地凭 CLK_3 下降沿到达时 Q_2 的状态来决定 Q_3 的次态了。例如, 在 $Q_1 Q_2 Q_3$ 从 011 变成 101 时, FF_1 从 0 变为 1。由于 CLK_3 首先从低电平变成了高电平而 Q_2 原来的 1 状态尚未改变, 所以在很短的时间里出现了 J_3 、 K_3 、 CLK_3 同时为高电平的状态, 使 FF_3 的主触发器翻转成 0 状态。在下一个计数脉冲到达后, 产生 CLK_3 的下降沿, 虽然这时 Q_2 已

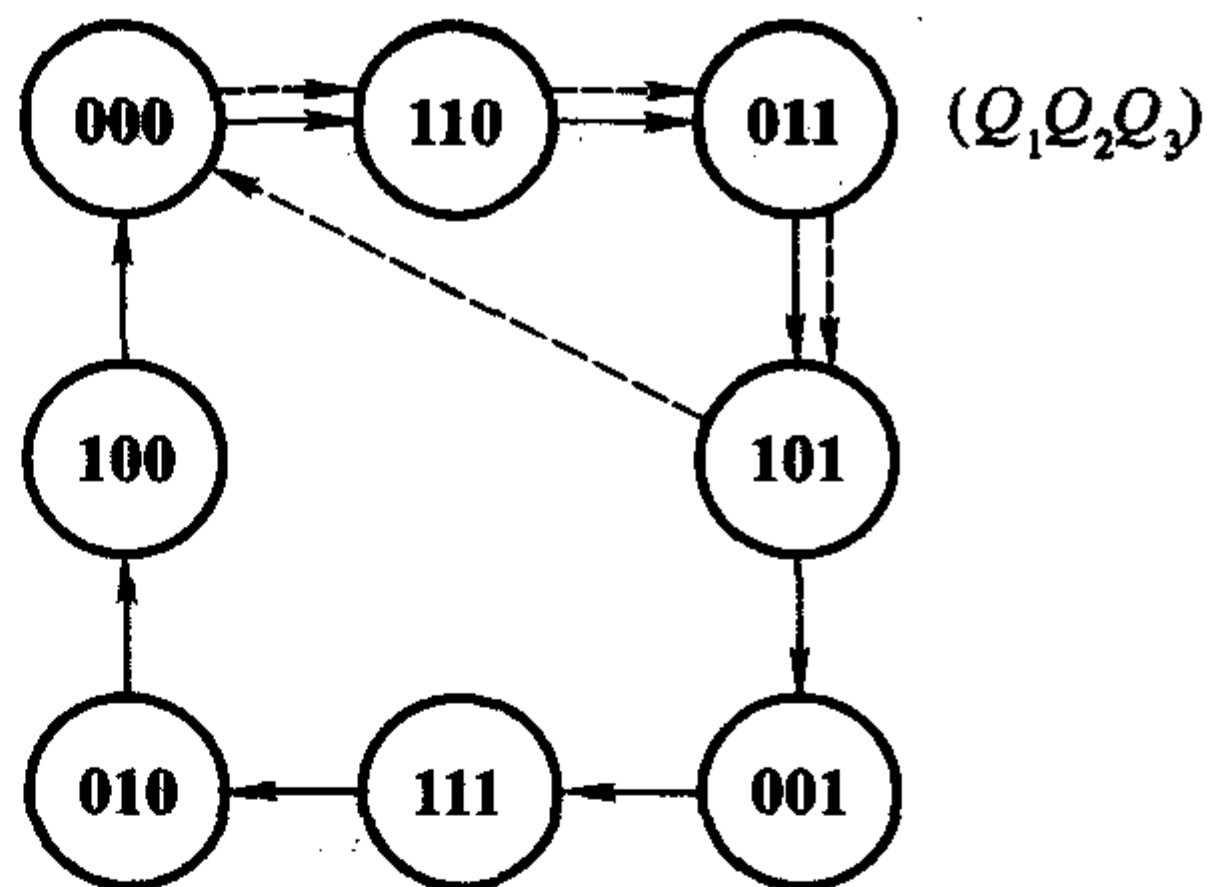


图 6.5.2 图 6.5.1 电路的状态转换图

变为 0 状态, 使 $J_3 = K_3 = 0$, 但由于 FF_3 的主触发器已经是 0 状态了, 从触发器仍要翻转为 0 状态, 使 $Q_1 Q_2 Q_3 = 000$ 。于是又得到另外一个状态转换表, 如表 6.5.1(b) 所示。对应的状态转换图将如图 6.5.2 中的虚线所示。倘若在设计时无法确切知道 CLK_3 和 Q_2 哪一个先改变状态, 那么也就不能确定电路状态转换的规律。

为了确保 CLK_3 的上升沿在 Q_2 的新状态稳定建立以后才到达 FF_3 , 可以在 Q_1 到 CLK_3 的传输通道上增加延迟环节。图 6.5.1 中的两个反相器 G_1 和 G_2 就是作延迟环节用的。只要 G_1 和 G_2 的传输延迟时间足够长, 一定能使 Q_2 的变化先于 CLK_3 的变化, 保证电路按八进制计数循环正常工作。

在同步时序电路中, 由于所有的触发器都在同一时钟操作下动作, 而在此之前每个触发器的输入信号均已处于稳定状态, 因而可以认为不存在竞争现象。因此, 一般认为存储电路的竞争 - 冒险现象仅发生在异步时序电路中。

在有些规模较大的同步时序电路中, 由于每个门的带负载能力有限, 所以经常是先用一个时钟信号同时驱动几个门电路, 然后再由这几个门电路分别去驱动若干个触发器。由于每个门的传输延迟时间不同, 严格地讲系统已不是真正的同步时序电路了, 故仍有可能发生存储电路的竞争 - 冒险现象。

图 6.5.3(a) 中的移位寄存器就是这样的一个例子。由于触发器的数目较多, 所以采用分段供给时钟信号的方式。触发器 $FF_1 \sim FF_{12}$ 的时钟信号 CLK_1 由门 G_1 供给, $FF_{13} \sim FF_{24}$ 的时钟信号 CLK_2 由门 G_2 供给。如果 G_1 和 G_2 的传输延迟时间不同, 则 CLK_1 和 CLK_2 之间将产生时间差, 发生时钟偏移现象。

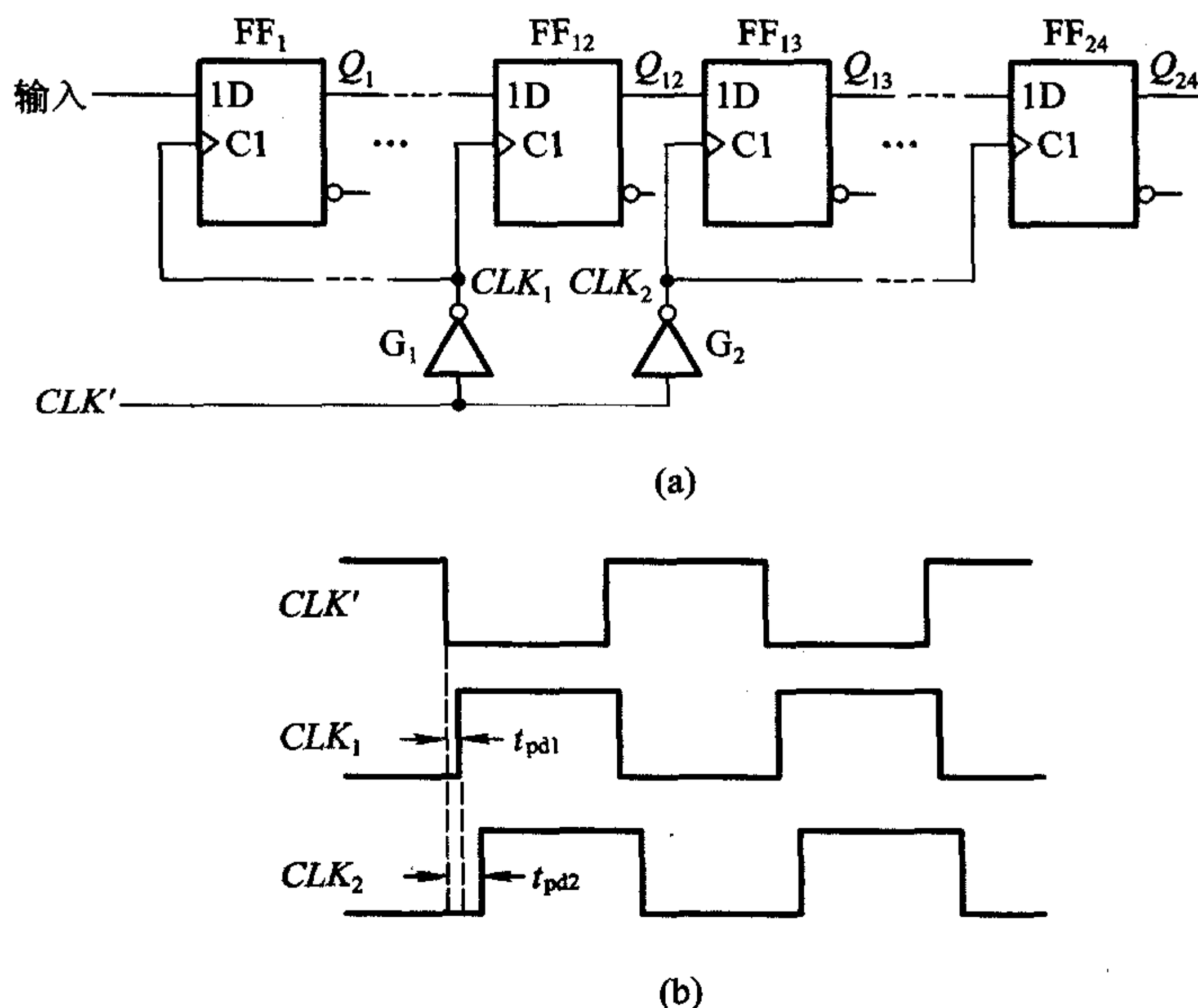


图 6.5.3 移位寄存器中的时钟偏移现象
(a) 电路图 (b) 时钟信号波形

时钟信号偏移有可能造成移位寄存器的误动作。譬如说, G_1 的传输延迟时间 t_{pd1} 比 G_2 的传输延迟时间 t_{pd2} 小得多, 如图 6.5.3(b) 所示, 则当 CLK' 输入一个负跳变时 CLK_1 的上升沿将先于 CLK_2 的上升沿到达, 使 FF_{12} 先于 FF_{13} 动作。如果两个门的传输延迟时间之差大于 FF_{12} 的传输延迟时间, 那么 CLK_2 的上升沿到 FF_{13} 时 FF_{12} 已经翻转为新状态了。这时 FF_{13} 接收的是 FF_{12} 的新状态, 而把 FF_{12} 原来的状态丢失了, 移位的结果是错误的。

相反, 如果 CLK_2 领先于 CLK_1 到达, 就不会发生错移位的现象。

假如使用的是维持阻塞结构触发器, 则对输入信号还要求有一段保持时间 t_H (见 5.7.4 节), 因而能计算出对时钟信号偏移的限制为

$$t_{pd2} - t_{pd1} < t_{PLH} - t_H \quad (6.5.1)$$

为了提高电路的工作可靠性, 防止错移位现象发生, 应挑选延迟时间长的反相器作 G_1 , 延迟时间短的作 G_2 。但这种做法显然是不方便的。实际上可以利用增加 FF_{12} 的 Q 端到 FF_{13} 的 D 端之间的传输延迟时间来解决。具体的做法可以在 FF_{12} 的 Q' 端与 FF_{13} 的 D 端之间串进一级反相器 (如图 6.5.4(a) 所示), 也可以在 FF_{12} 的 Q 端与地之间接入一个很小的电容 (如图 6.5.4(b) 所示)。

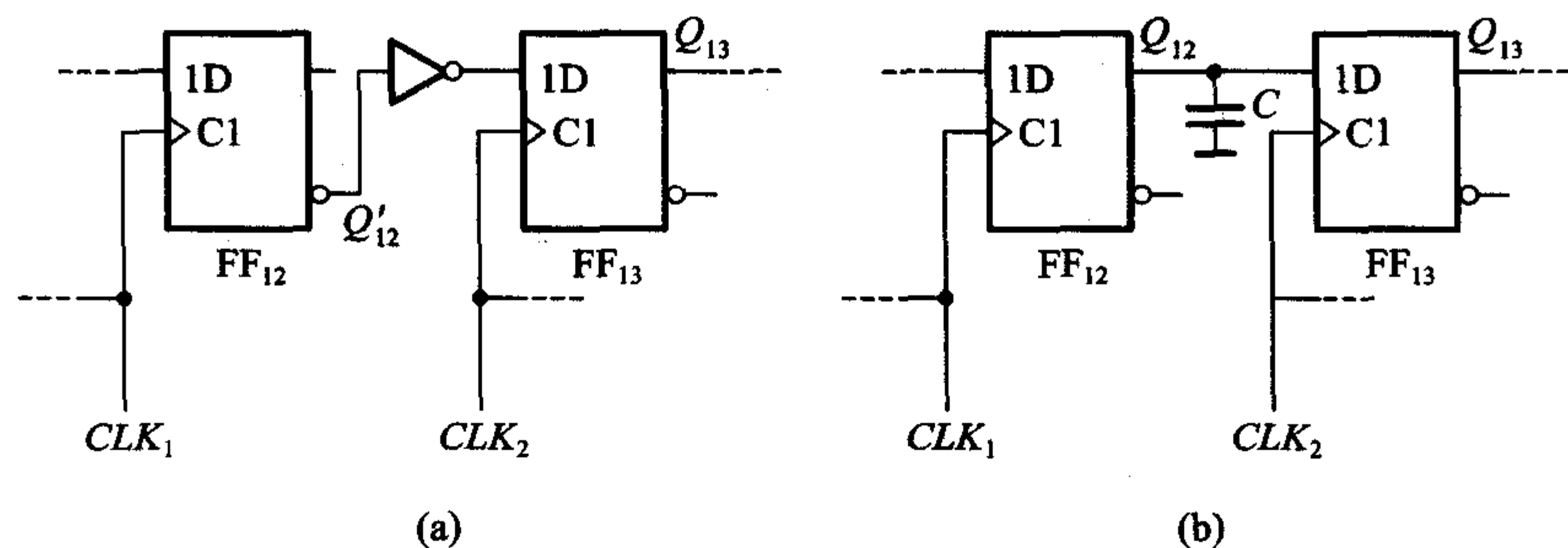


图 6.5.4 防止移位寄存器错移的方法
(a) 接入反相器作延迟环节 (b) 接入延迟电容

* 6.6 用 Multisim 7 分析时序逻辑电路

我们在 4.5 节中已经讲过,用 Multisim 7 可以分析组合逻辑电路,得到给定逻辑电路的逻辑函数式,并且可以方便地将它化为最简单的与或形式。Multisim 7 同样可以用于分析时序逻辑电路。下面我们就通过一个实例说明如何具体使用 Multisim 7 分析时序逻辑电路。

【例 6.6.1】 分析图 6.6.1 所示的计数器电路,求电路的时序图,说明这是几进制计数器。

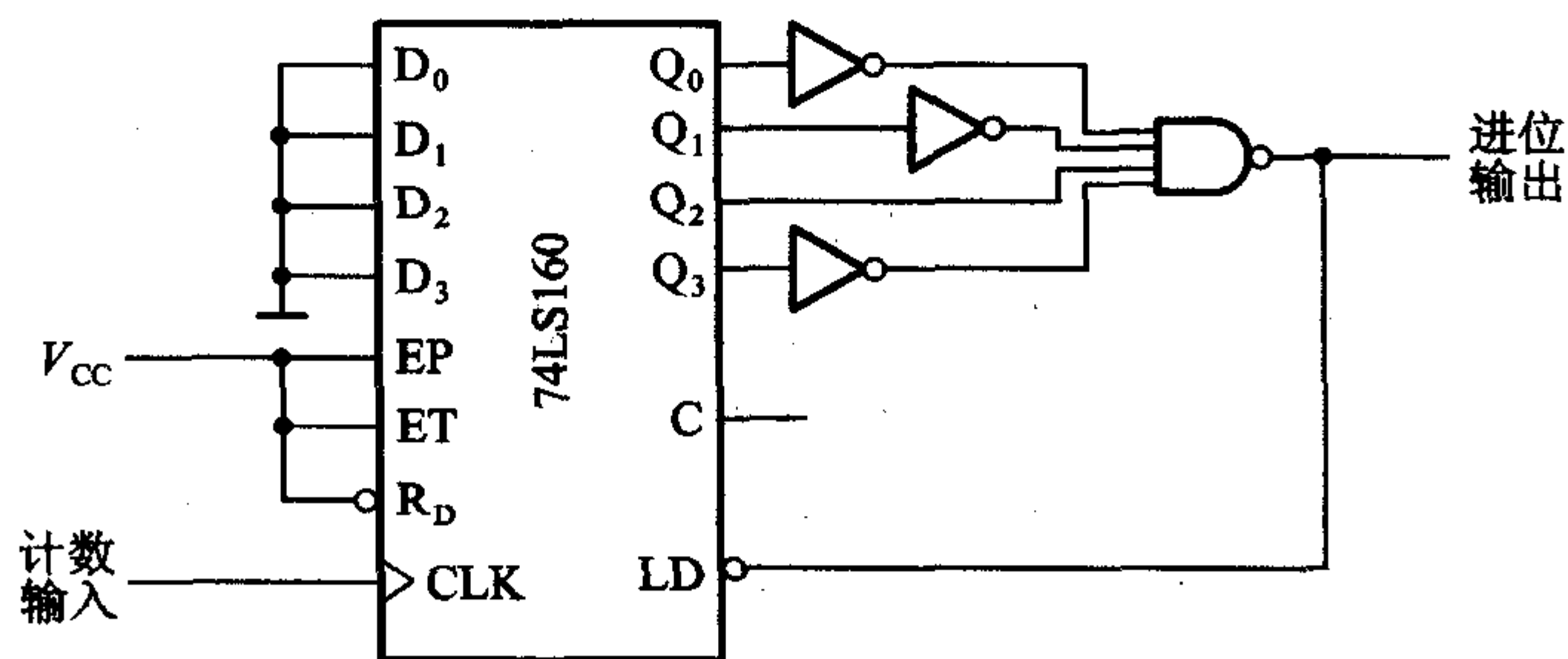


图 6.6.1 例 6.6.1 的时序逻辑电路

解: 在 Multisim 7 中选用 TTL 器件库中的 74LS160、反相器 7404 以及与非

门 7420 构成图 6.6.1 中的电路,并接入信号发生器 XFG1 和逻辑分析仪 XLA1,如图 6.6.2 所示^①。图 6.6.2 中的 Q_A 、 Q_B 、 Q_C 、 Q_D 与图 6.6.1 中的 Q_0 、 Q_1 、 Q_2 、 Q_3 对应。

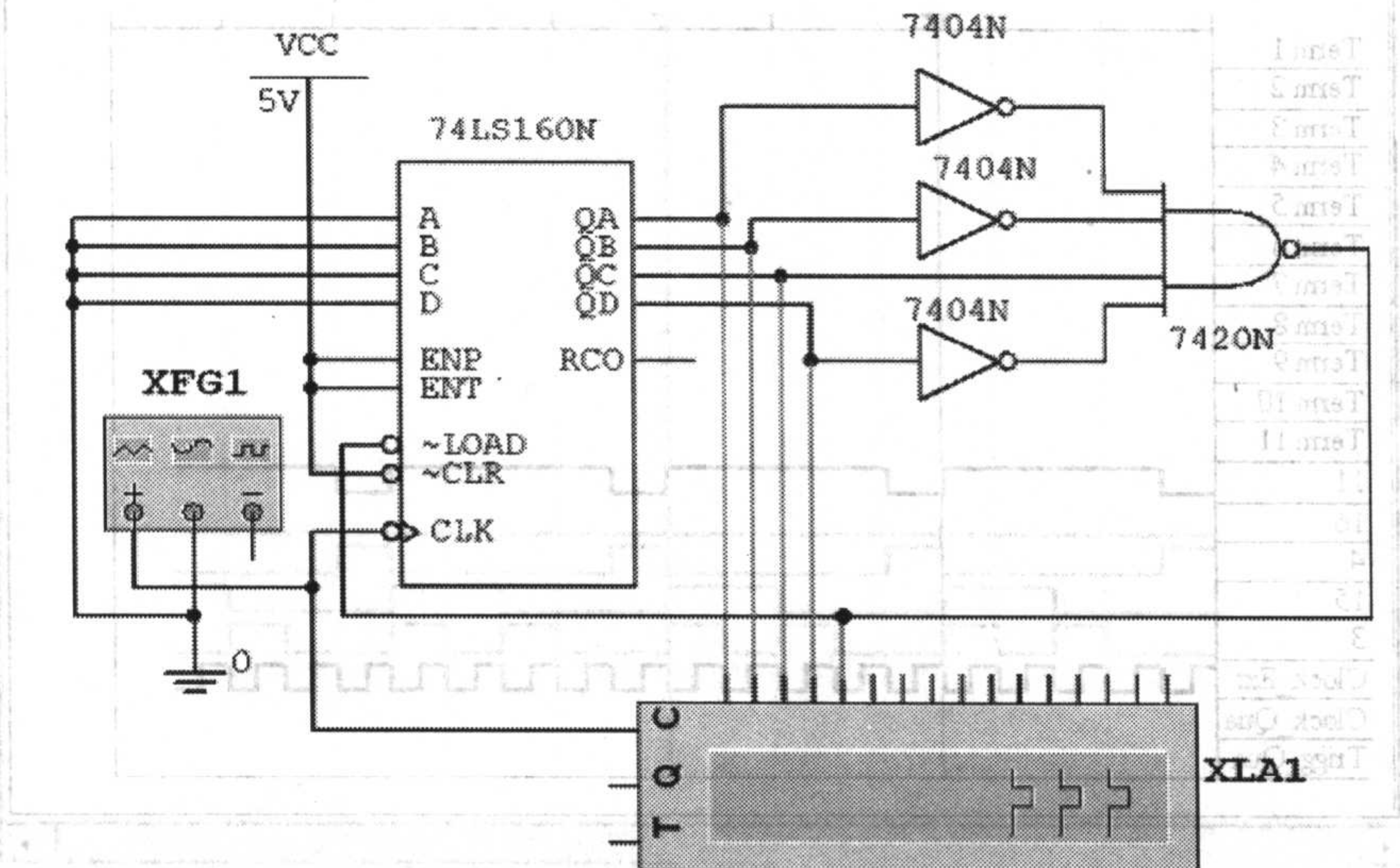


图 6.6.2 用 Multisim 7 构建图 6.6.1 的电路

利用 Multisim 7 中的逻辑分析仪对计数器的时钟波形和输出波形进行观测,得图 6.6.3 所示的波形图。分析波形图可见,每 5 个时钟周期输出波形就重复一遍,在 7420 的输出端产生一个输出进位脉冲。因此,这是一个五进制计数器。

从逻辑分析仪给出的 Q_D 、 Q_C 、 Q_B 、 Q_A 的波形图,还可以画出电路的状态转换图,如图 6.6.4 所示。

图 6.6.1 中的计数器采用了同步预置数的工作方式,当计数器处于 $Q_D Q_C Q_B Q_A = 0100$ 时,用 7404 和 7420 译出 $LD' = 0$ 的信号,将 $D_3 D_2 D_1 D_0 = 0000$ 的信号预置入计数器,作为计数循环的初始状态。由此分析可得,该计数器是五进制计数器。因此,用 Multisim 7 得到的仿真结果与理论分析结果完全吻合。

^① 按本书中的规定画法,图 6.6.2 中 74LS160 的 CLK 输入端不应有小圆圈。