

电后数据便随之丢失。因此,每次开始工作时都要重新装载编程数据,并需要配备保存编程数据的 EPROM。这些都给使用带来一些不便。

此外,FPGA 的编程数据一般是存放在 EPROM 中的,而且要读出并送到 FPGA 的 SRAM 中,因而不便于保密。而 CPLD 中设有加密编程单元,加密后可以防止编程数据被读出。

可见,FPGA 和 CPLD 各有不能取代的优点,这也正是两种器件目前都得到广泛应用的原因所在。

*8.7.2 FPGA 的 IOB 和 CLB

现以 Xilinx 公司生产的 XC2064 为例,介绍 FPGA 的 IOB 和 CLB 的电路结构和工作原理。

一、IOB

XC2064 是 Xilinx 公司 FPGA 器件中结构比较简单的一种,它一共有 56 个可编程的 I/O 端。

由图 8.7.3 所示的电路结构图可见,每个 IOB 由输出三态缓冲器 G_1 、触发器、输入缓冲器 G_2 和两个数据选择器 MUX1、MUX2 组成。在图中所用的数据选择器符号上只标出了数据输入端和数据输出端,省略了地址输入端。实际上每个 2 选 1 数据选择器都应当有一位输入地址代码,每个 4 选 1 数据选择器应当有两位输入地址代码。这些代码都存放在 FPGA 内部的编程数据存储单元中。

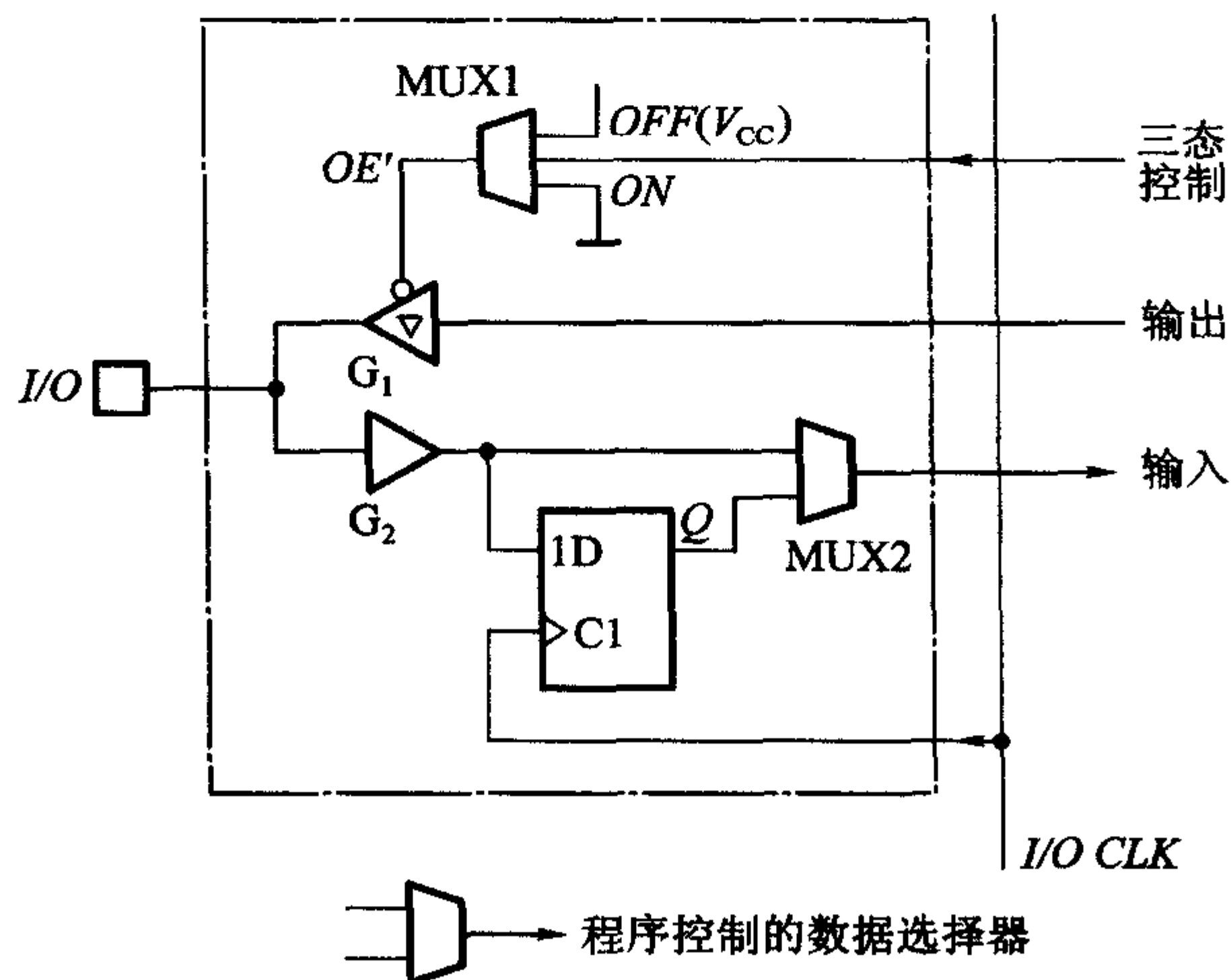


图 8.7.3 XC2064 的 IOB 电路

输出三态缓冲器的控制信号 OE' 由 MUX1 给出。MUX1 输出低电平时 IOB 工作在输出状态, FPGA 内部产生的信号通过 G_1 送至 I/O 端。MUX1 输出高电平时 G_1 为高阻态, IOB 工作在输入状态。

MUX2 用于输入方式的选择。当 MUX2 的输出选中输入缓冲器 G_2 的输出时, 为异步输入方式, 加到 I/O 端的输入信号立刻就通过 G_2 、MUX2 送往 FPGA 内部。当 MUX2 的输出选中触发器的输出时, 为同步输入方式。在同步输入方式下, 必须等到时钟信号 I/O CLK 到达后, 加到 I/O 端的输入信号才能经过 MUX2 送往内部电路。

输入缓冲器 G_2 的阈值电平是可编程的, 既可设置为 TTL 电路的阈值电平 (1.4 V), 也可以设置为高速 CMOS 电路的阈值电平 (2.2 V)。在 XC2064 中, 所有 IOB 的时钟信号是公用的。

二、CLB

在 XC2064 中有 64 个 CLB, 排列成 8×8 的矩阵。每个 CLB 的电路中包含组合逻辑电路、存储电路和由一些数据选择器组成的内部控制电路, 如图 8.7.4 所示。

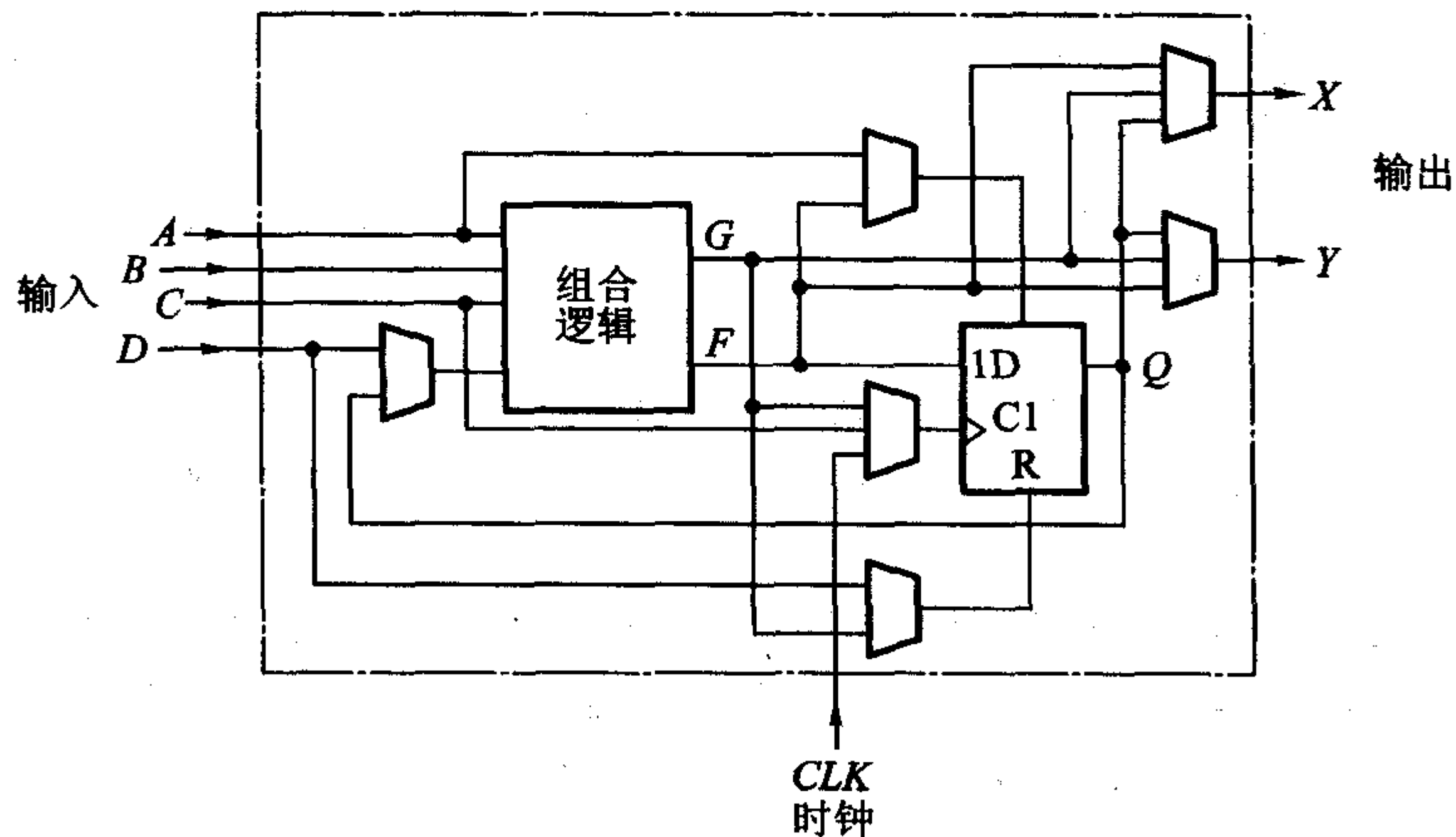


图 8.7.4 XC2064 的 CLB 电路

组合逻辑电路部分是有 4 个输入端、两个输出端的通用逻辑模块。根据设计的需要可以将组合逻辑部分设置成 3 种不同的组态。第一种组态如图 8.7.5(a) 所示, 通过编程可以产生任何形式的四变量组合逻辑函数。第二种组态如图 8.7.5(b) 所示, 可以产生两个三变量的任何形式的逻辑函数。第三种组态如图 8.7.5(c) 所示, 可以产生含有 A 、 B 、 C 、 D 、 Q 的五变量逻辑函数。

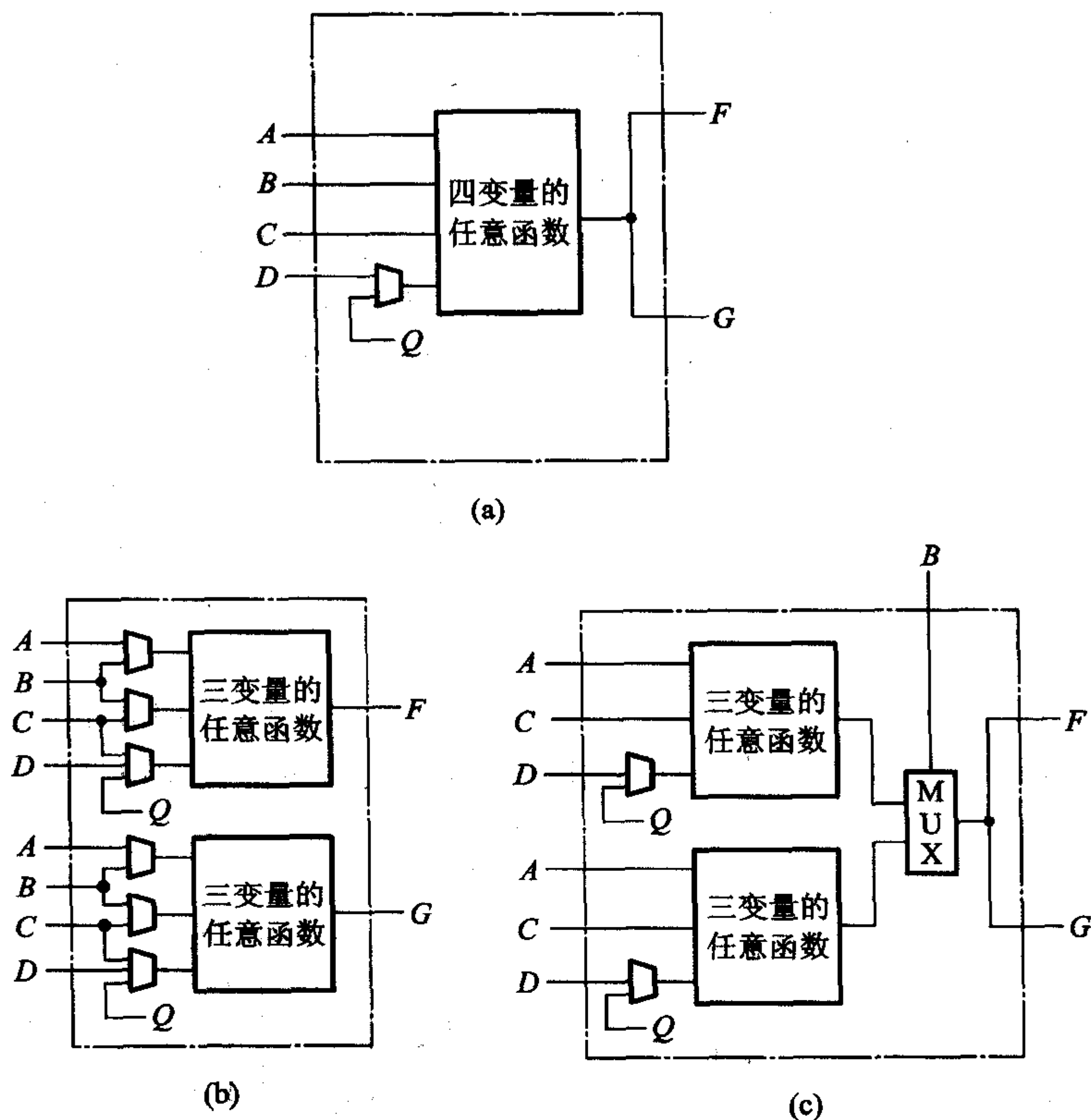


图 8.7.5 XC2064 中 CLB 的 3 种组态

(a) 四变量任意函数 (b) 两个三变量任意函数 (c) 五变量逻辑函数

这种通用逻辑模块由 N 沟道 MOS 管和 CMOS 反相器组成,输出与输入间的逻辑函数关系由一组编程控制信号决定。将编程控制信号与函数对应关系列成函数表,在编程过程中通过查表即可找出所需的编程数据。

为了说明这种查询表方式的工作原理,下面来分析一个二输入变量的通用逻辑模块的逻辑功能。图 8.7.6 所示电路是用 NMOS 管构成的通用逻辑模块, A 、 B 是两个输入变量, F 为输出逻辑函数, C_0 、 C_1 、 C_2 、 C_3 是编程控制信号。在 $C_0C_1C_2C_3$ 的 16 种不同取值下,得到 A 和 B 的 16 种函数关系,如表 8.7.1 所示。

例如,当 $C_0C_1C_2C_3 = 1100$ 时,若 $A = 1$ 、 $B = 0$,则 T_1 和 T_2 导通, $F = 1$;若 $A = 0$ 、 $B = 1$,则 T_3 、 T_4 导通, $F = 1$;若 $A = B = 1$ 或 $A = B = 0$,则 4 条支路皆不导通, $F = 0$ 。因此,得到 $F = AB' + A'B = A \oplus B$ 。

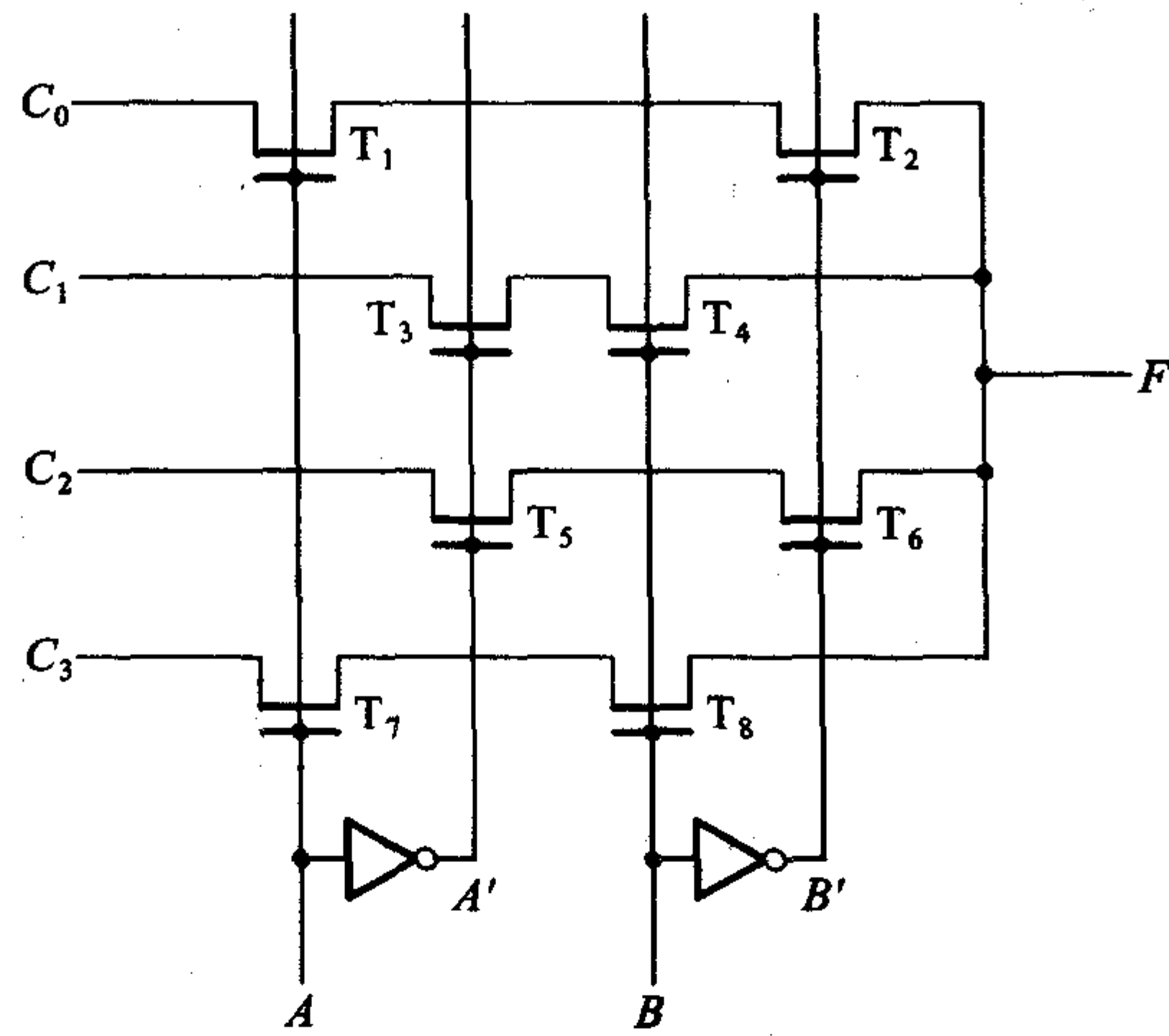


图 8.7.6 二变量通用逻辑模块的原理图

表 8.7.1 二变量通用逻辑模块的函数表

C_0	C_1	C_2	C_3	F
0	0	0	0	0
0	0	0	1	AB
0	0	1	0	$A'B' = (A+B)'$
0	0	1	1	$AB + A'B' = (A \oplus B)'$
0	1	0	0	$A'B$
0	1	0	1	B
0	1	1	0	A'
0	1	1	1	$A' + B$
1	0	0	0	AB'
1	0	0	1	A
1	0	1	0	B'
1	0	1	1	$A + B'$
1	1	0	0	$A'B + AB' = A \oplus B$
1	1	0	1	$A + B$
1	1	1	0	$A' + B' = (AB)'$
1	1	1	1	1

同理,可以找出三变量、四变量通用逻辑模块的这种函数表,并通过查询表的方式进行设计。三变量通用逻辑模块的函数关系由一组 8 位代码指定,四变量通用逻辑模块的函数关系由一组 16 位代码指定。

CLB 中存储电路的结构如图 8.7.7 所示,它只包含一个触发器(在 XC3000 和 XC4000 系列的 FPGA 器件中,每个 CLB 中有两个触发器)。根据设计要求,这个触发器可以编程为边沿触发的 D 触发器,也可以编程为电平触发的 D 型锁存器。

触发器的 D 端接收组合电路部分的输出 F。时钟信号由数据选择器 MUX1

给出,既可以选择片内公共时钟 CLK 作为时钟信号,工作在同步方式;又可以选择组合电路的输出 G 或输入变量 C 作为时钟信号,工作在异步方式。而且,用数据选择器 MUX2 还可以选择用时钟的上升沿或下降沿(或高电平和低电平)触发。

触发器的异步置位信号由数据选择器 MUX3 给出,异步置位信号可以从输入变量 A 和组合电路输出 F 当中选择。异步置零信号由数据选择器 MUX4 给出,既可以选组合电路输出 G 作为异步置零信号,也可以选输入变量 D 作为异步置零信号。

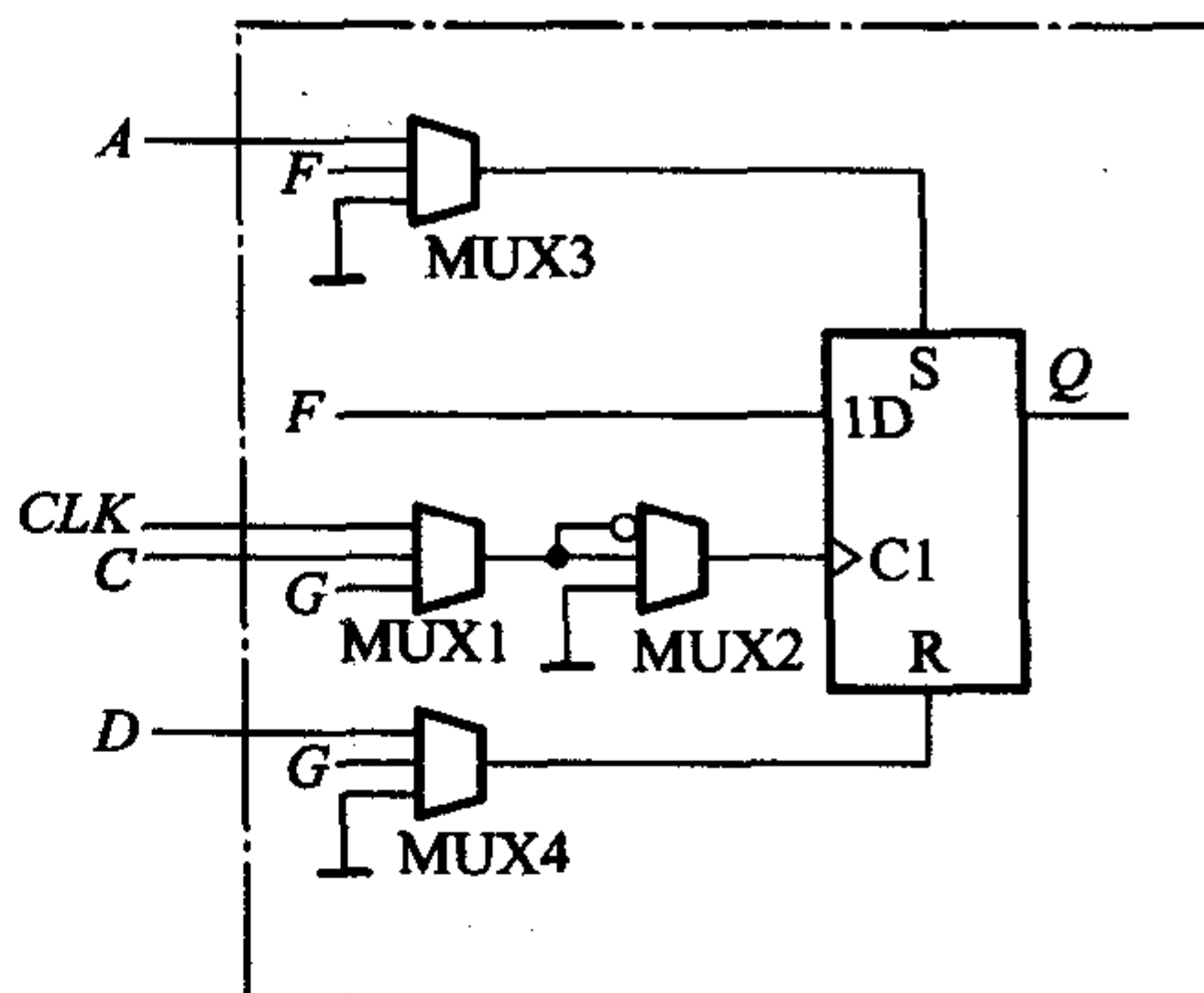


图 8.7.7 XC2064 中 CLB 的存储电路

*8.7.3 FPGA 的互连资源

为了能将 FPGA 中数目很大的 CLB 和 IOB 连结成各种复杂的系统,在布线区内布置了丰富的连线资源。这些互连资源可以分为三类,即金属线、开关矩阵 SM(Switching Matrices)和可编程连接点 PIP(Programmable Interconnect Points)。在图 8.7.8 中示出了这些互连资源的布局状况。

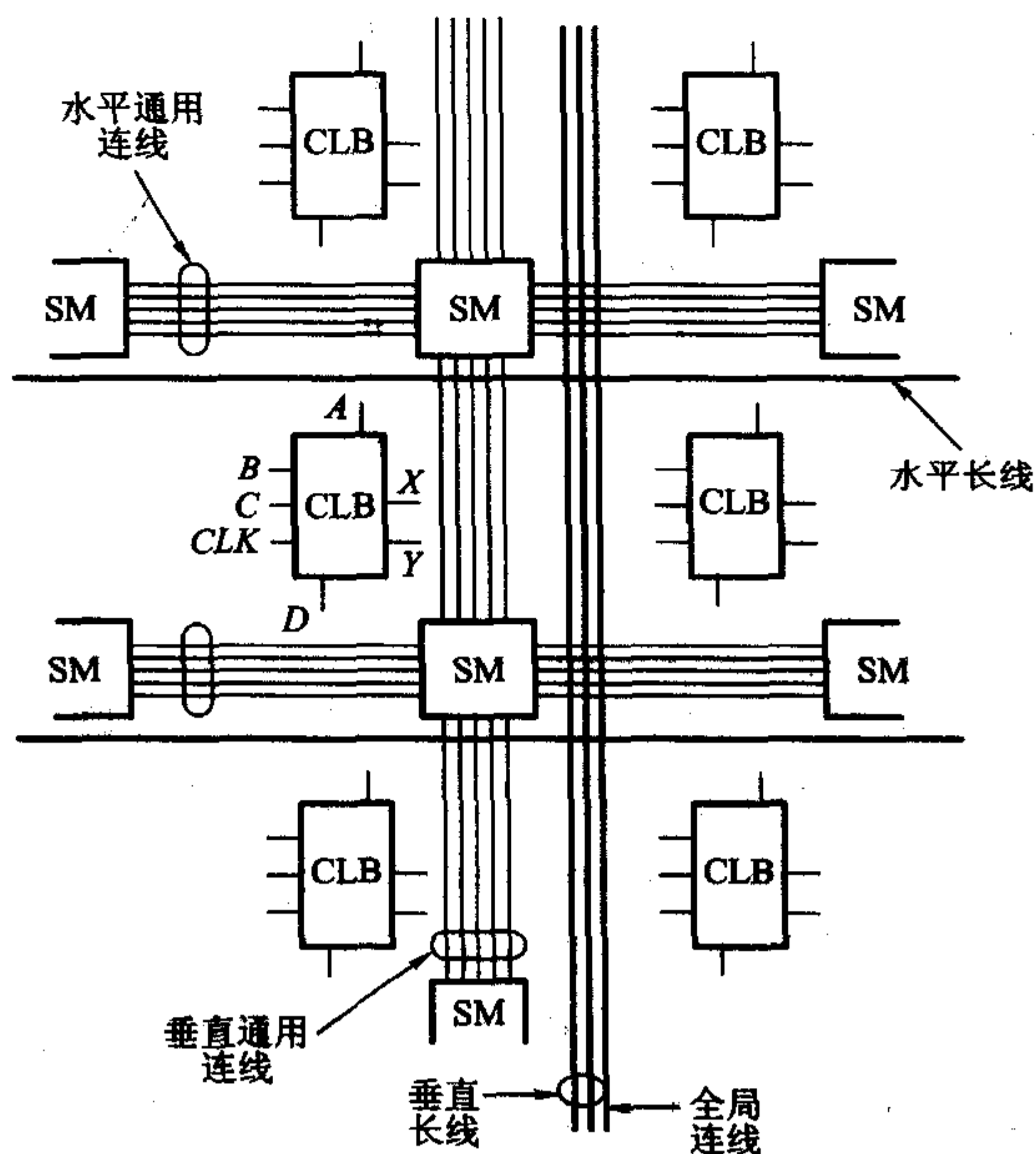


图 8.7.8 FPGA 内部的互连资源

布线区里的金属线分为水平通用连线、垂直通用连线、水平长线、垂直长线、全局连线和直接连线等几种。这些金属线经可编程的连接点与 CLB、IOB 和开关矩阵相连。其中的通用连线主要用于 CLB 之间的连接,长线主要用于长距离或多分支信号的传送,全局连线则用于输送一些公共信号(如公用的 RESET'信号)等。

图 8.7.9 中给出了开关矩阵和可编程连接点的布置图。开关矩阵的作用如同一个可以实现多根导线转接的接线盒,通过对开关矩阵编程,可以将来自任何方向上的一根导线转接至其他方向的某一根导线上。图 8.7.9 中还列出了开关矩阵在不同编程情况下的连接状态。

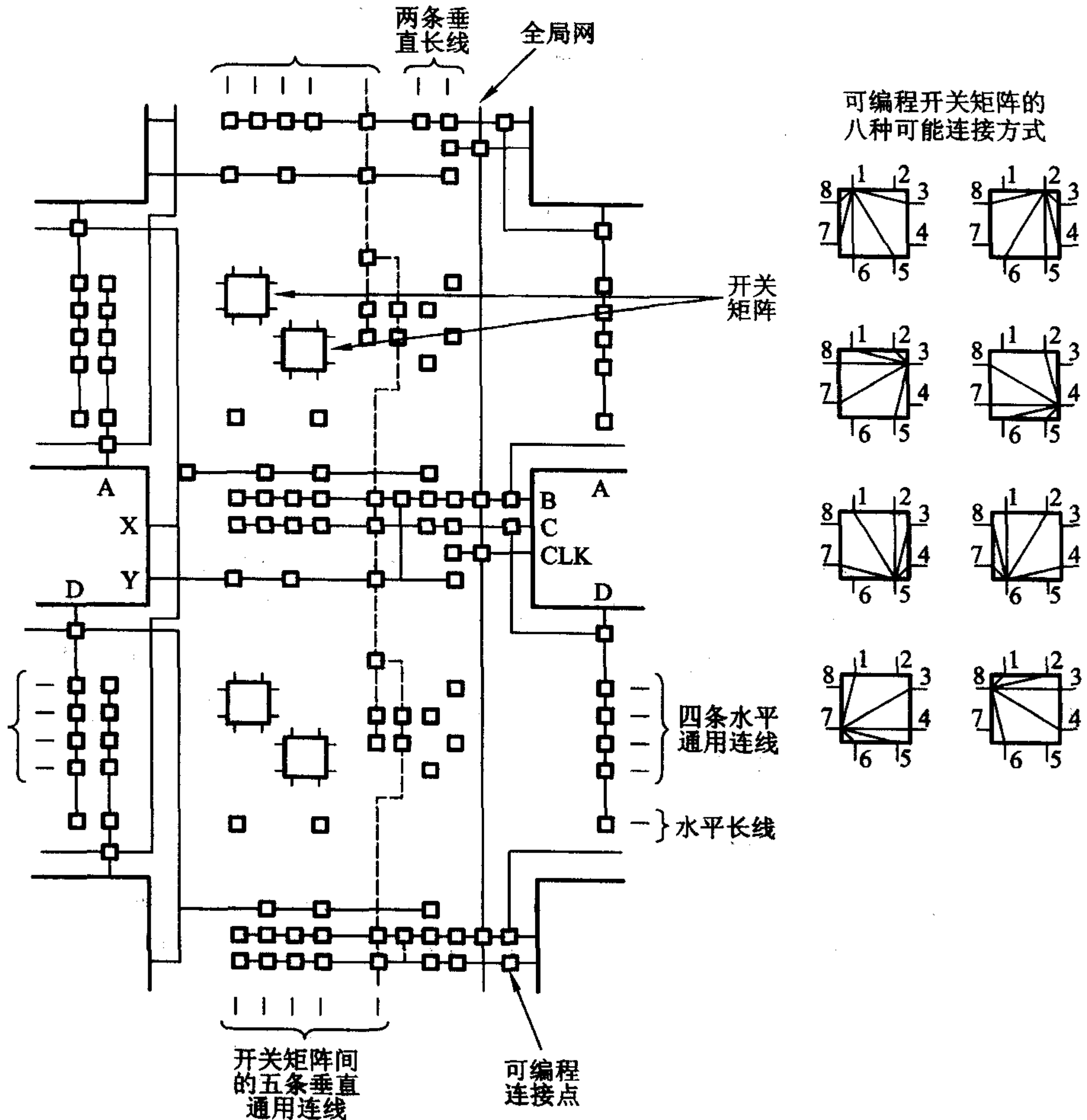


图 8.7.9 开关矩阵和可编程连接点

图 8.7.10 中以粗线示出了经过编程后的一种连线情况,CLB(1) 输出 X 经过开关矩阵分别送到了 CLB(2) 的 D 端、CLB(3) 的 A 端和 CLB(4) 的 C 端。

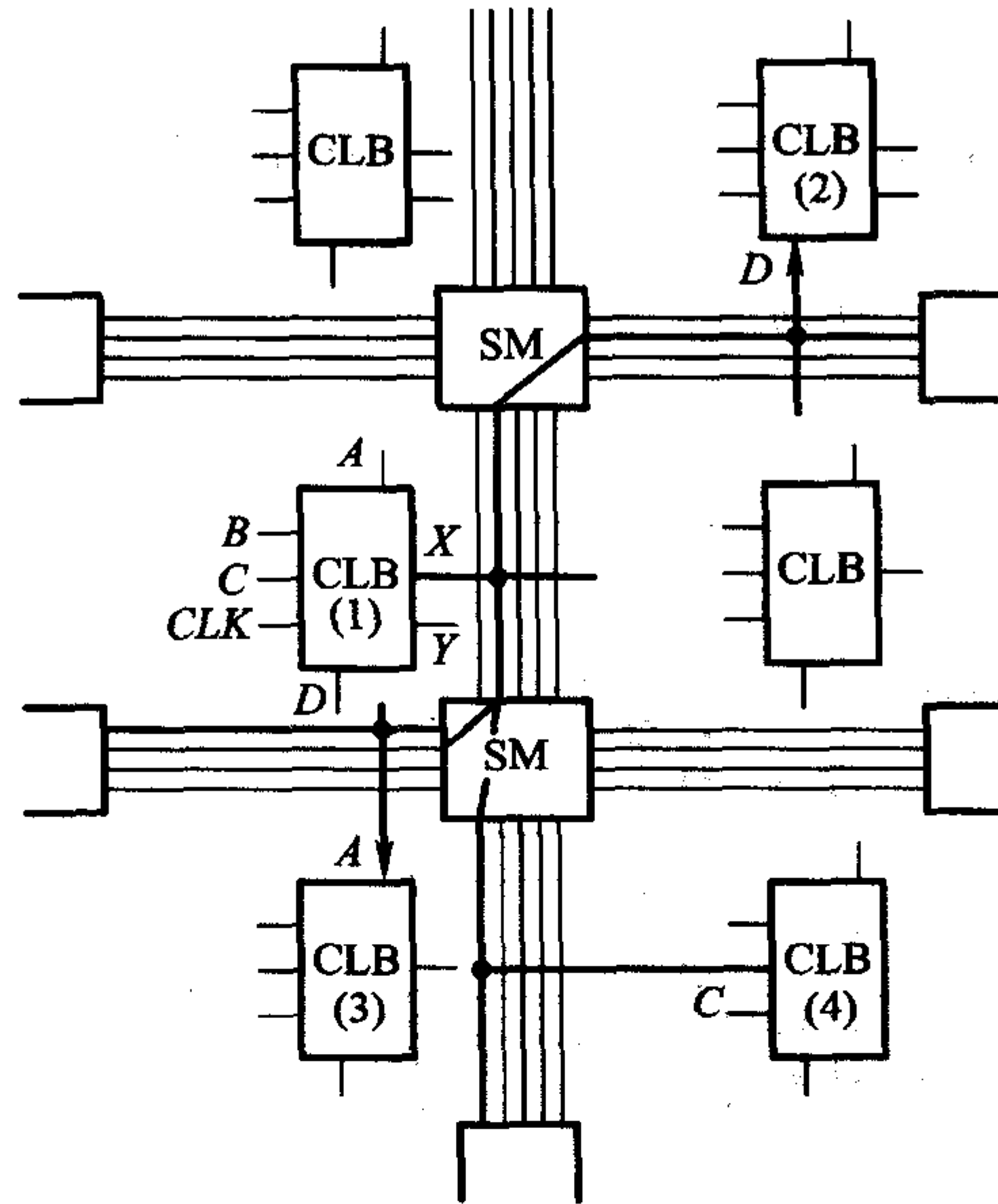


图 8.7.10 利用水平和垂直通用连线和开关矩阵实现连接

此外,为了减少传输延迟时间和简化编程,在相邻的 CLB 中间还设置了直接连线,如图 8.7.11 所示。每个 CLB 的输出端 X 可以与下边一个 CLB 的输入端 A 、 B 和上边一个 CLB 的输入端 C 、 D 直接相连;输出端 Y 可以和右边一个 CLB 的输入端 B 直接相连。使用这些直接连线连接时不经过开关矩阵和通用连线,只要对信号接收端的开关点编程就行了。

*8.7.4 编程数据的装载

将编程数据写入 FPGA 内部编程数据存储器的称为装载。整个装载过程是在 FPGA 内的控制电路操作下自动进行的。下面仍以 XC2000 系列 FPGA 为例,说明装载的过程。

装载过程在接通电源后自动开始,也可以由外加控制信号启动。编程数据通常存放在一个 EPROM 中,也可以存放在计算机的存储器中。装载的操作有不同模式,由模式选择信号 M_0 、 M_1 及 M_2 指定,有主、从模式之分和数据并行输入、串行输入模式之分。

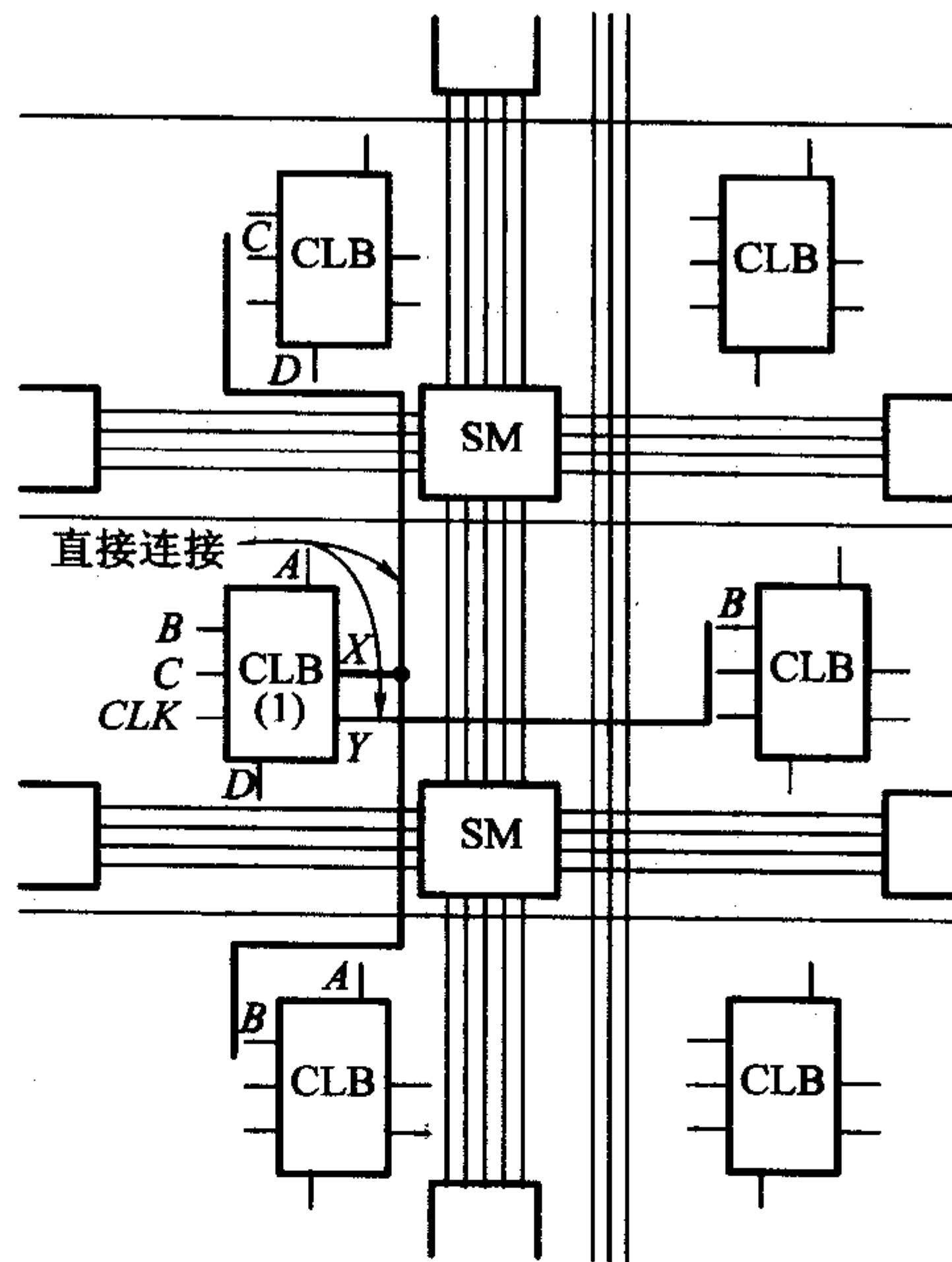


图 8.7.11 用直接连线实现连接

图 8.7.12(a) 是“主并模式”装载的电路图,编程数据事先已存入 EPROM 中。

首先简要介绍一下图中 XC2064 引脚功能的设置情况。 $A_0 \sim A_{15}$ 是地址代码输出端,装载开始后从 $A_0 \sim A_{15}$ 便自动地顺序输出地址代码(可以设置成从 0000 号地址递增,也可以设置成从 FFFF 号地址递减)给 EPROM。 $D_0 \sim D_7$ 是数据输入端,接收来自 EPROM 的编程数据。装载完成后 $A_0 \sim A_{15}$ 和 $D_0 \sim D_7$ 都可作为用户可编程的 I/O 引脚用。 M_0 、 M_1 和 M_2 给出装载模式选择信号, M_0 为专用输入引脚, M_2 在装载结束后作为用户可编程的 I/O 引脚使用。如果在装载完成后需要回读(即读出已经装入 FPGA 中的编程数据),则回读的数据将从 M_1 脚输出。 $DOUT$ 是数据输出端,在装载过程中,写入数据的同时就把数据又变成串行数据从 $DOUT$ 端输出。在将多片 FPGA 同时编程时, $DOUT$ 的输出将作为另一片 FPGA 的串行数据输入。装载结束后它也是一个用户可编程的 I/O 端。 HDC 和 LDC' 是两个表示装载数据正在进行的信号,装载过程中 HDC 始终为高电平, LDC' 始终为低。装载完成后它们也是用户可编程的 I/O 引脚。

D/P' 为装载完成信号端。当装载过程结束,电路进入用户编程设定的工作状态后, D/P' 变成高电平。如果装载完成后再令 D/P' 由高电平变为低电平,则

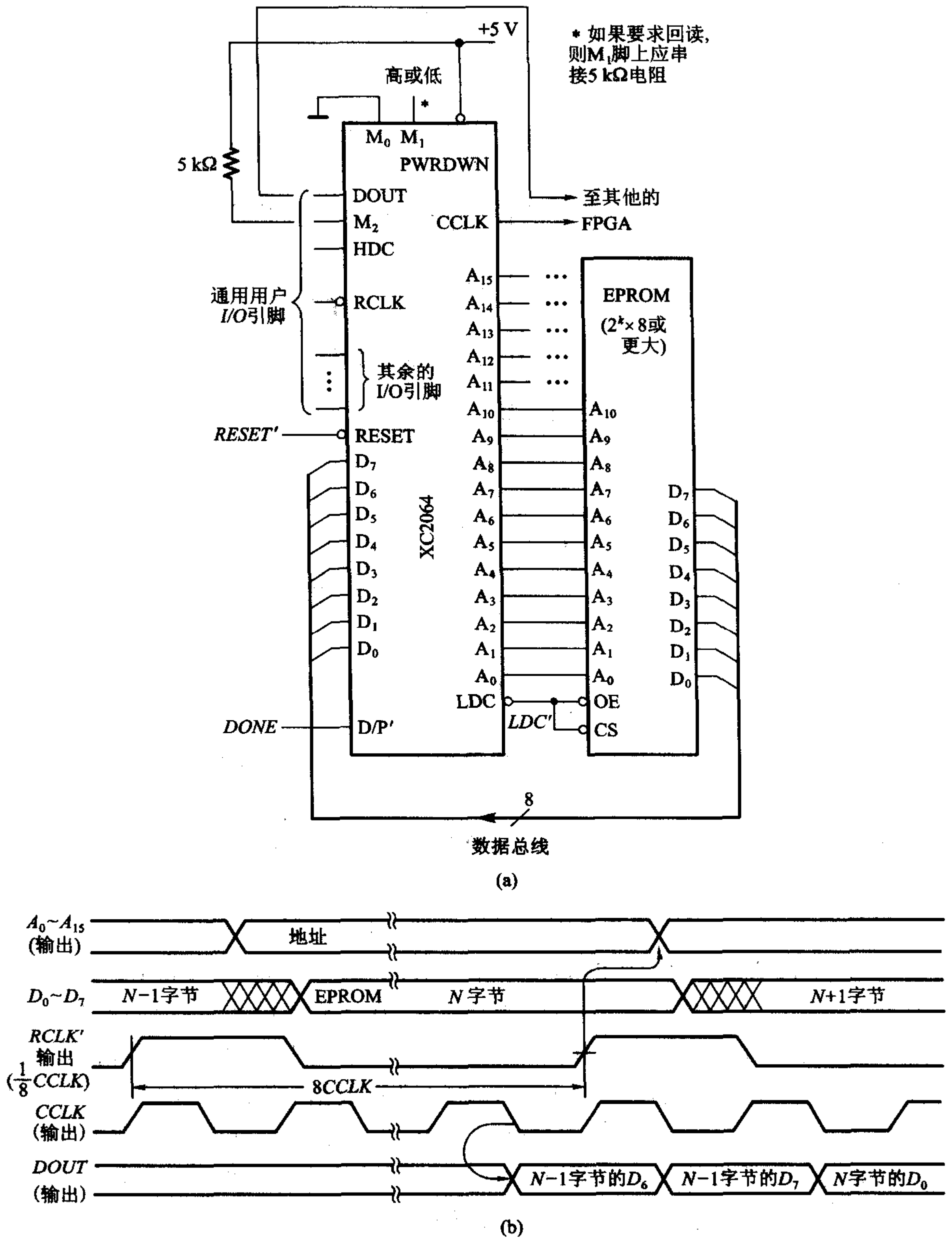


图 8.7.12 XC2064 的主并装载模式

(a) 电路接法 (b) 写入数据的时序图

电路将重新初始化,并重新进入装载过程。*PWRDWN'*是掉电输入信号。当

$PWRDWN'$ 变成低电平时,电路停止工作,并使所有输出变为高阻态。这时只要器件的供电电源不低于2.3 V,编程数据存储器的数据仍能保留,在 $PWRDWN'$ 返回高电平以后电路还能恢复原来的工作状态。

$RESET'$ 为复位信号。如果在装入数据的过程中 $RESET'$ 端出现低电平输入信号,则编程数据存储器将被清除,重新开始装载过程。

CCLK为时钟引脚。在主并模式下它输出时钟信号。这时需要利用外接石英晶体和器件内部的反相器构成(Pierce)振荡器,如图8.7.13所示。在回读时CCLK是一个输入时钟信号端, $RCLK'$ 是另一个输出时钟信号,它的频率为CCLK的1/8。当外接动态存储器时,装载期间作读信号端使用,一般情况下不用。

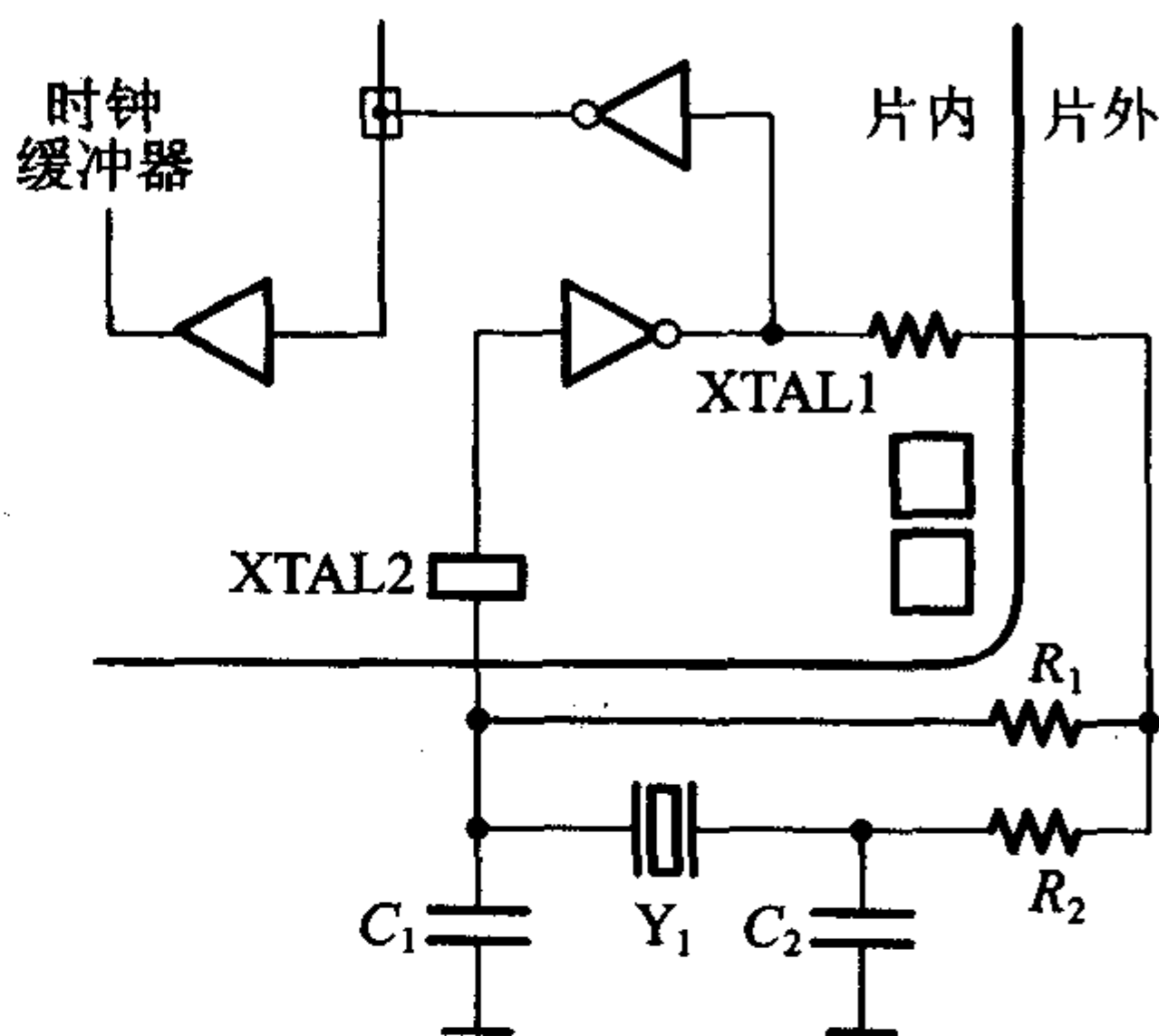


图 8.7.13 石英晶体振荡器电路

在图8.7.14中画出了装载过程的流程图。接通电源后,内部的复位电路被触发,开始清除编程数据存储器。在 $RESET'$ 信号无效的条件下,电路自动检测 M_0 、 M_1 、 M_2 的状态,以确定装载模式。

然后启动数据读入操作,从 $A_0 \sim A_{15}$ 顺序输出地址代码给EPROM,同时从 $D_0 \sim D_7$ 逐个字节读入数据。并行读入的数据在器件内部被转换成串行数据,并且在DOUT可以同时读出。装载完成后DONE变为高电平,电路开始进入用户逻辑状态。如果数据读入过程中出现 $RESET' = 0$ 信号,则装载过程停止,并清除编程存储器,重新开始装载过程。

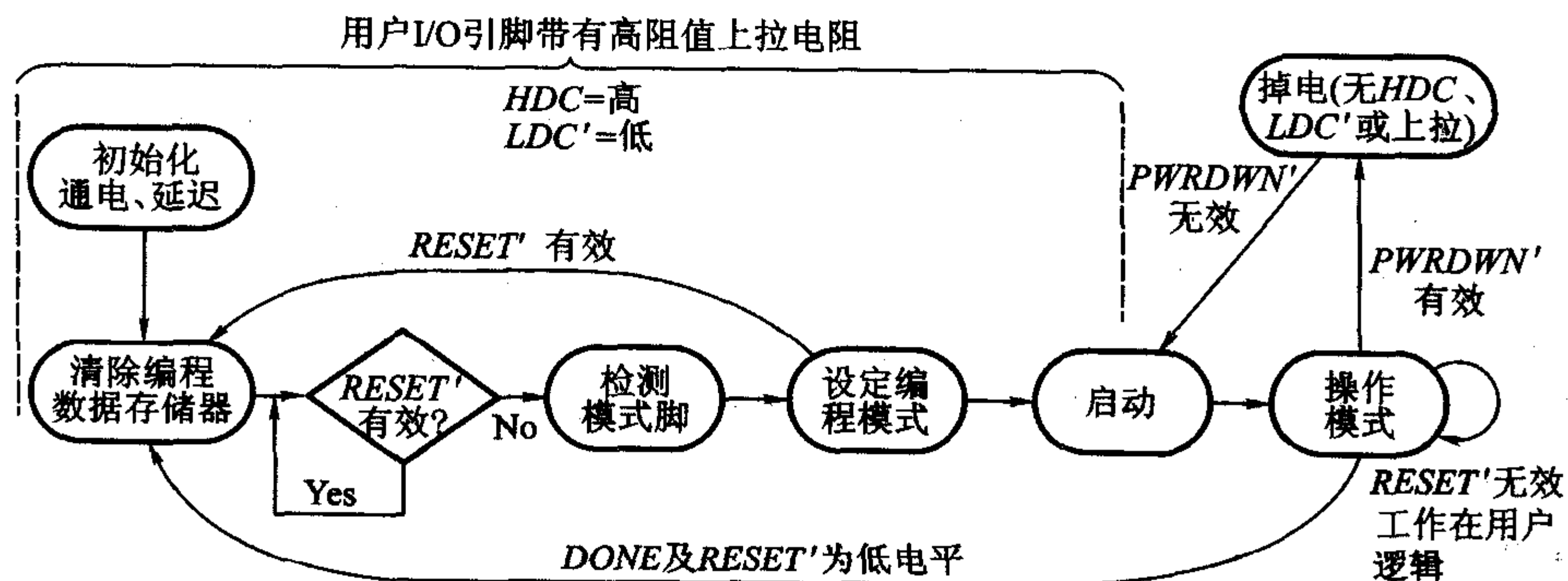


图 8.7.14 装载过程的流程图

除了主并装载模式外,还有主串模式、从串模式、外设模式等,这里就不一一