

第 5 章

可编程逻辑器件

前面几章介绍的中小规模数字集成电路(如 74 系列)的逻辑功能是固定的,不能为了适合某一特定设计而变更,而且这些器件功能简单,包含的门数少,因此,在构造大型逻辑电路时效果较差。20 世纪 70 年代出现了可编程逻辑器件(Programmable Logic Device,PLD),一片 PLD 能容纳的逻辑门可达到数百、数千甚至更多,片上的逻辑功能可以由用户编程指定。

自从 PLD 问世以来,经历了从 PLA、PAL 等低密度 PLD 到 CPLD、FPGA 等高密度 PLD 的发展过程。PLD 的出现和发展,打破了数字系统被中小规模通用型集成电路和大规模专用集成电路垄断的局面。与中小规模通用型集成电路相比,用 PLD 实现数字系统存在集成度高、速度快、功耗小、可靠性高等优点。与大规模专用集成电路相比,用 PLD 实现数字系统具有研制周期短、前期投资少、无风险、修改逻辑设计方便、小批量生产成本低等优点。

5.1 可编程逻辑阵列

20 世纪 70 年代中期出现的可编程逻辑阵列(Programmable Logic Array,PLA)是最早问世的 PLD 器件。其结构如图 5-1 所示,它由一个与阵列和一个或阵列组成。PLA 的输入是 x_1, x_2, \dots, x_n , 与阵列的每一个输出 p_1, p_2, \dots, p_k 是关于 x_1, x_2, \dots, x_n 原反变量的乘积项,乘积项作为或阵列的输入,或阵列的输出信号是 f_1, f_2, \dots, f_m , 每个输出信号实现乘积项的任意的或运算。由此可见,PLA 可以实现输入 x_1, x_2, \dots, x_n 的乘积项之和。



图 5-1 PLA 的总体结构

图 5-2 是一个 PLA 的内部结构示意图,它有 3 个输入信号和 2 个输出信号。与阵列中的每个与门有 6 个输入,分别对应于 x_1, x_2, x_3 及其反变量。每个输入和与门的连接关系由编程决定,图 5-2 中的波浪线表示输入和与门相连,断开表示不相连。或阵列同样也是可编程的。图 5-3 是对图 5-2 的一种简化,称之为方阵图。每一个与门连接到水平横线,与门的输入则画成与水平横线相交的垂直线,在水平线和垂直线的交叉处加 \cdot 表示该输入被编程为和与门相连。或阵列与此相似,或门连接到一根垂直线,和与门的输出线相交,对交叉点连接进行编程,实现所需的逻辑。在图 5-3 中,输出为 P_1 的与门的输入端连接到变量 x_1

和 x_2 ,因此 $P_1 = x_1x_2$ 。同样的道理, $P_2 = \overline{x_1}x_2\overline{x_3}$, $P_3 = x_1\overline{x_2}$, $P_4 = x_1x_3$ 。输出为 f_1 的或门的输入端连接到乘积项 P_1 、 P_2 、 P_4 ,因此 $f_1 = x_1x_2 + \overline{x_1}x_2\overline{x_3} + x_1x_3$,同样的道理, $f_2 = \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}$ 。这个例子说明,通过对与阵列及或阵列分别编程,每一个输出 f_1 和 f_2 都可以实现输入的各种函数。

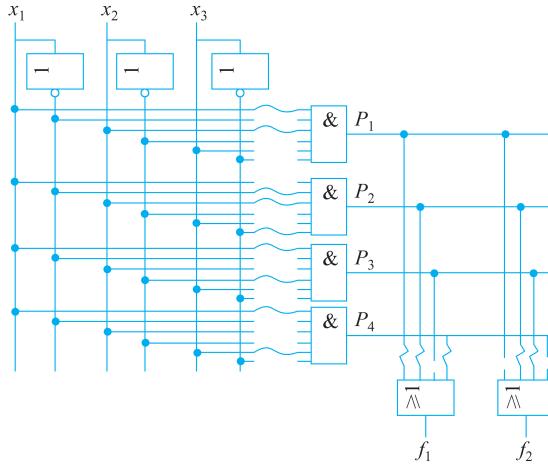


图 5-2 PLA 的内部结构示意图

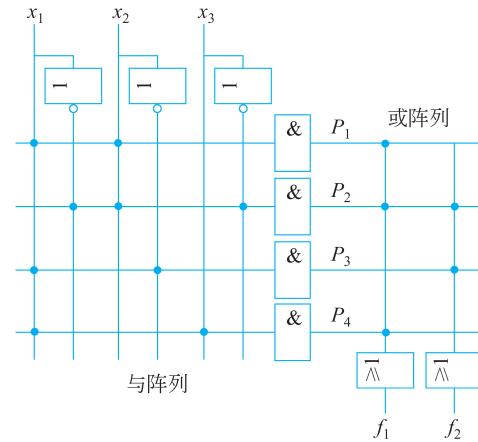


图 5-3 PLA 的阵列图

5.2 可编程阵列逻辑

PLA 的与阵列和或阵列都是可编程的,可编程开关的制造工艺复杂,降低了 PLA 的速度。因此,出现了一种叫作可编程阵列逻辑(Programmable Array Logic, PAL)的器件,它同样采用阵列逻辑结构,其中与阵列可编程,或阵列固定。这种结构比 PLA 工艺简单,易于编程,而且能提供更高的速度。同时,也可以实现灵活多变的逻辑功能。

图 5-4 是一个 PAL 的例子,它的与阵列是可编程的,而两个或门的连接是固定的,乘积项 P_1 和 P_2 固定连接到一个或门, P_3 和 P_4 固定连接到另一个或门。该 PAL 被编程实现下述逻辑函数:

$$f_1 = \overline{x_1} + x_2x_3$$

$$f_2 = x_1\overline{x_2} + x_3$$

与图 5-3 所示的 PLA 相比较,由于或门的输入是固定的,PAL 的灵活性较小,为了增加 PAL 的灵活性,许多 PAL 在或门之后还附加了一些电路,这种电路习惯上被称为宏单元。图 5-5 所示的是宏单元的一个实例,它的或门输出的后面是一个动态触发的 D 触发器,在时钟信号的作用下,保存来自或门的信号值。二选一多路开关用于为 PAL 输出端选择信号:或者把或门的输出作为 PAL 的输出,或者把触发器的输出作为 PAL 的输出。图 5-4 中还有一个三态缓冲器连接在多路开关和 PAL 输出端之间,同时多路开关的输出还反馈到 PAL 的与阵列。

综上所述,PAL 能提供多种灵活的内部结构,可实现组合逻辑和时序逻辑功能。用 PAL 进行逻辑设计时,一般先按常规设计方法对要求实现的功能进行正确的逻辑描述(如

作出真值表、状态转移图等),写出相应的函数表达式,然后根据具体要求(如输入数、输出数、寄存器数以及与项数等)选择合适的器件,最后按函数表达式进行编程。

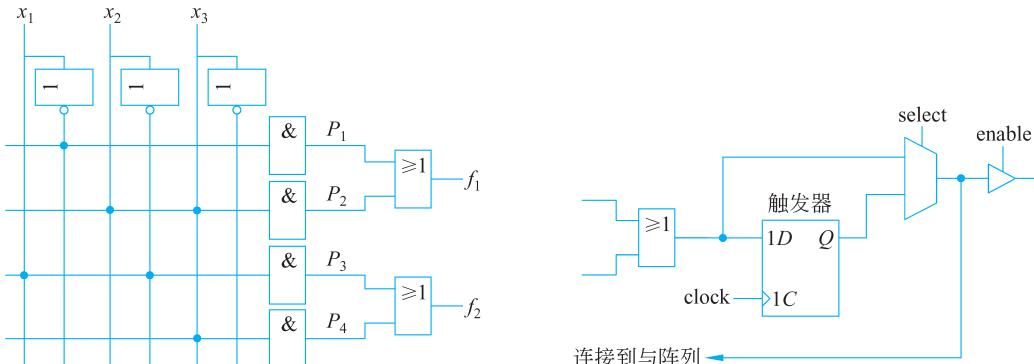


图 5-4 一个 PAL 的例子

图 5-5 PAL 中的宏单元

例 5-1 用 PAL 设计一个 8421BCD 码同步计数器。

解：有寄存器输出的 PAL 能用来实现时序逻辑电路。8421BCD 码同步计数器状态转移表如表 5-1 所示。

表 5-1 8421BCD 码同步计数器状态转移表

Q_4	Q_3	Q_2	Q_1	$Q_4^{\frac{n}{4}+1}$	$Q_3^{\frac{n}{3}+1}$	$Q_2^{\frac{n}{2}+1}$	$Q_1^{\frac{n}{1}+1}$	Z
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	0	1

由表 5-1 得到 4 个 D 触发器的激励函数, 化简后为

$$D_1 = \overline{Q}_1$$

$$D_2 = \overline{Q_4} \overline{Q_2} Q_1 + Q_2 \overline{Q_1}$$

$$D_3 = \overline{Q_3} Q_2 Q_1 + Q_3 \overline{Q_1} + Q_3 \overline{Q_2}$$

$$D_4 = \overline{Q_4} Q_3 Q_2 Q_1 + Q_4 \overline{Q_1}$$

实现中需要 4 个触发器 FF_4 、 FF_3 、 FF_2 和 FF_1 ，根据触发器的激励函数及输出方程，可作出 PAL 阵列图，如图 5-6 所示。

在图 5-3 和图 5-4 中,用符号 \cdot 表示 PLA 或 PAL 中的信号被编程为和逻辑门相连,PLD 芯片中含有数千个可编程开关,其编程是依靠 EDA 工具完成的。支持 PLD 器件的 EDA 工具能够自动产生对 PLD 器件中每一个开关编程的信息,当用户完成了电路设计,EDA 工具就会产生一个编程文件(或称为熔丝映射表),它规定 PLD 器件中每一个开关的

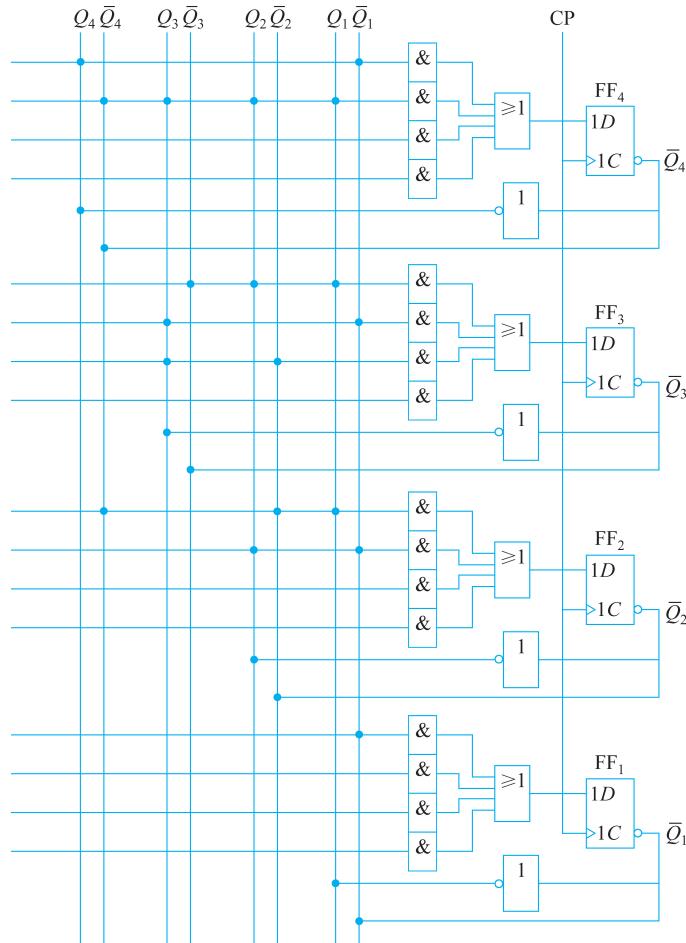


图 5-6 实现 8421BCD 码同步计数器的 PAL 阵列图

状态,开关处于上述指定的状态就能正确实现用户设计的电路。

运行 EDA 工具的计算机通过电缆和专用的编程器(programming unit)相连,PLD 器件放置在编程器的上面。编程器处于特定的编程模式。对 PLD 器件编程的过程大概需要几分钟,编程完毕之后,编程器通常会自动从 PLD 器件中读回每一个开关的状态,以检验对该芯片编程的正确性。

5.3 复杂可编程逻辑器件

PLA、PAL 以及类似的电路属于简单 PLD(Simple PLD, SPLD),一般用于规模较小的数字电路。复杂可编程逻辑器件(Complex PLD, CPLD)是从 PLA、PAL 发展而来的高密度可编程逻辑器件,它的规模可以达到几十万门甚至上百万门,而工作速度可以达到 100MHz 以上。目前 CPLD 已经成为主流的可编程逻辑器件之一。

5.3.1 CPLD 的基本结构

CPLD 包含多个类似于 PLA 或 PAL 的电路块,并且通过内部连线资源把这些电路块

连接起来。

CPLD一般由3部分组成：逻辑阵列块（Logic Array Block, LAB）、可编程互连阵列（Programmable Interconnect Array, PIA）和I/O控制块（I/O control block）。图5-7是CPLD器件的内部结构。

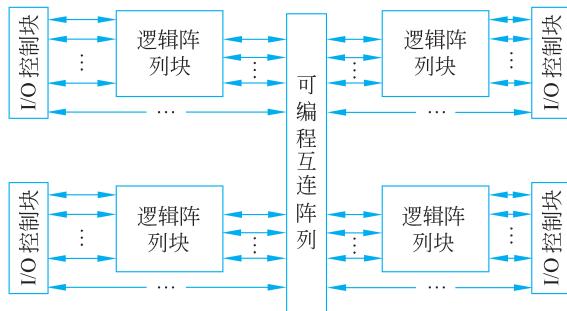


图5-7 CPLD器件的内部结构

逻辑阵列块是CPLD中最重要的单元，每个逻辑阵列块在结构上和PLA或PAL相似，可以独立地配置成组合逻辑电路或者时序逻辑电路来使用。每个逻辑阵列块和可编程互连阵列相连，而且还连接到一个I/O控制块。

可编程互连阵列包含许多可编程开关，提供内部逻辑阵列块之间的互连。这种互连可以在不改变引脚配置的情况下改变内部设计，因此具有很大的灵活性。CPLD的可编程互连阵列还有一个重要特征，就是其连线延时是累加的，因此CPLD的延时是可预测的，在设计时可以得到较好的时序性能。

I/O控制块和外部输入输出引脚相连，每个引脚都配备一个三态缓冲器。三态缓冲器起到开关的作用，使得每一个引脚既可用于CPLD的输出，又可用于CPLD的输入。

在前面介绍过对于PAL和PLA使用编程器编程，但是这种方法对于大型CPLD却很不方便。原因之一是引脚多而细，易折易弯；原因之二是需要一个价格昂贵的插座，这种插座的价格甚至高于芯片本身。为了满足CPLD编程的需要，出现了在系统可编程（In-System Programmability, ISP）技术，即CPLD芯片所在的印制电路板上还焊有一个很小的连接器，再用一根电缆把连接器和计算机系统连接起来。EDA系统产生的编程信息通过电缆和连接器到达CPLD芯片，从而实现对CPLD的编程。CPLD器件被编程之后，其编程状态保持不变，即使电源掉电也能保持原有信息，这种性质称作非易失性编程。

5.3.2 典型的CPLD芯片

图5-8展示了Altera公司的MAX7000芯片的总体结构，其中有4个专用输入引脚，这4个专用输入可以用作通用输入，也可以用作每个宏单元和I/O引脚的高速全局控制信号（时钟、清零及两个输出使能信号）。图5-8中每个灰色方框为一个逻辑阵列块，其中包含16个宏单元。每个逻辑阵列块都连接到一个I/O控制块，其中有连接到芯片封装引脚上的三态缓冲门；每个这样的引脚都可被用作一个输出引脚或输入引脚。每个逻辑阵列块还连接到可编程互连阵列上。可编程互连阵列由一组扩展到整个器件中的连线组成，所有宏单元之间的连接都通过可编程互连阵列完成。

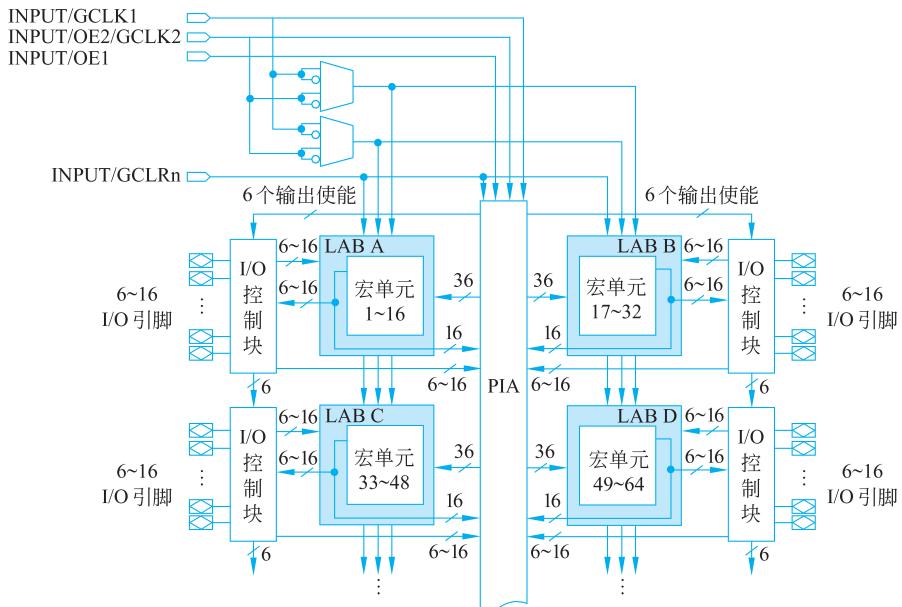


图 5-8 MAX7000 的总体结构

图 5-9 给出了 MAX7000 宏单元的结构。其中有 5 个乘积项，它们可通过乘积项选择阵列(product term select matrix)连接到一个或门。对该或门可进行配置，使它只使用宏单元中实现的电路需要的那些乘积项。或门的输出通过一个异或门连接到一个触发器上。如果需要超过 5 个以上的乘积项，多出来的乘积项可以通过从其他宏单元中“共享”得到，宏单元中的或门有一个额外的输入，它可以被连接到上方宏单元中或门的输出，称为并行扩展器(parallel expander)，被用于实现包含多于 20 个乘积项的逻辑函数。如果还需要更多的乘积项，可以使用另一个被称为共享扩展器(shared expander)的特性。如图 5-9 下部灰色方

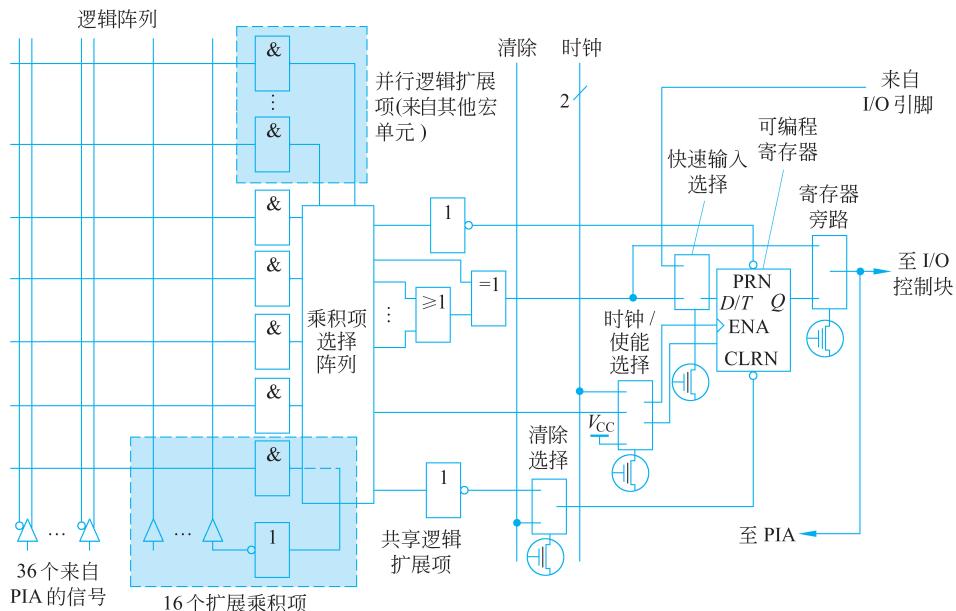


图 5-9 MAX7000 宏单元的结构

5.4 现场可编程门阵列

前面已经介绍了多种器件,包括74系列、SPLD和CPLD,它们可以实现许多逻辑电路。上述器件中除了CPLD之外,规模都偏小,一般只适于简单应用。即使是CPLD,一个芯片中也只能装入中等规模的逻辑电路。衡量电路规模的常用指标是实现某电路所使用的二输入与非门(NAND)的个数,或者称该器件中包含的等价门个数。技术发展到今天,由2万个门所组成的电路已经不能算作大型逻辑电路。对于大型逻辑电路,可以使用包含门数更多的现场可编程门阵列(Field-Programmable Gate Array, FPGA)实现,用FPGA实现的电路规模可以超过几十万个等价门。

5.4.1 FPGA的基本结构

FPGA的基本结构框图如图5-10所示。FPGA不采用与阵列和或阵列结构,与SPLD和CPLD的内部结构有明显的区别。FPGA的结构基本由3部分组成:可配置逻辑块(Configurable Logic Block, CLB)、输入输出功能块(I/O Block, IOB)以及可编程连线(Programmable Interconnect, PI)。可配置逻辑块用于实现所需的逻辑功能,I/O块用于与封装的引脚相连。可配置逻辑块安排成二维阵列,可编程连线则安排成垂直和水平布线通道,位于可配置逻辑块的行和列之间。布线通道中包含连线和可编程开关,允许可配置逻辑块以多种方式相连。I/O块和可编程连线之间也有可编程开关,用于实现它们之间的连接。

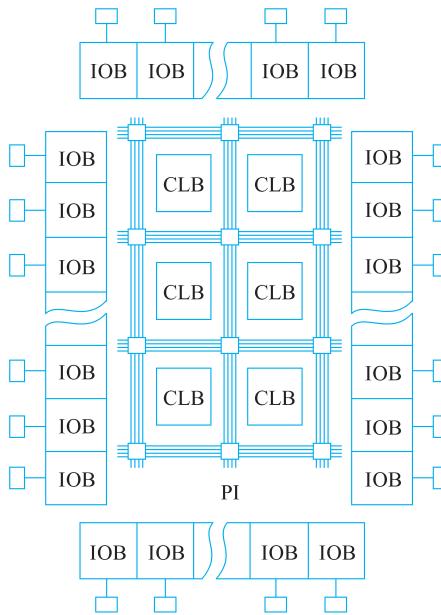


图 5-10 FPGA 的基本结构框图

FPGA器件结构的差别主要表现在可配置逻辑块上。最常用的可配置逻辑块结构是查找表(LookUp Table, LUT),典型的LUT有一个输出和很少的输入,用于实现一个规模不大的逻辑函数。LUT内部有若干存储单元和多路选择器。存储单元的值可以是1或0。任何形式的组合逻辑函数均能通过向存储单元写入相应的数据实现。LUT的规模由其输入个数来定义,可以创建各种规模的LUT。图5-11是一个二输入的LUT,它有两个输入 x_1 、 x_2 和一个输出 f ,可以实现两个输入变量的任何逻辑函数。

例5-2 用一个二输入LUT实现逻辑函数 $f = x_1 \overline{x_2} + \overline{x_1}x_2$ 。

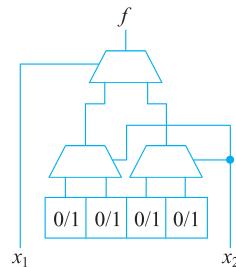


图 5-11 二输入 LUT 电路

解:列出函数 $f = x_1 \overline{x_2} + \overline{x_1}x_2$ 的真值表,如图5-12(a)所示。因为两个变量的真值表有4行,所以这个LUT具有4个存储单元,一个存储单元和真值表中一行的输出值相对应。输入变量 x_1 和 x_2 连接到3个多路选择器的选择端。根据LUT中3个多路选择器的安排,在输入变量 x_1 和 x_2 的取值确定时,只有一个存储单元能够通过多路选择器到达LUT的输出端。例如,当 $x_1 = x_2 = 0$ 时,第一个存储单元的值输出,由真值表知此时 $f = 0$,因此第一个存储单元取值0。

同理,当输入变量取其他值时,LUT的输出值也都和真值表一致,从而确定每一个相应存储单元的值。实现函数 $f = x_1 \overline{x_2} + \overline{x_1}x_2$ 的二输入 LUT 如图 5-12(b) 所示。

x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

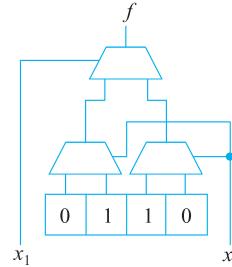
(a) 函数 $f = x_1 \overline{x_2} + \overline{x_1}x_2$ 的真值表(b) 实现函数 $f = x_1 \overline{x_2} + \overline{x_1}x_2$ 的二输入 LUT

图 5-12 例 5-2 图

例5-3 用一个三输入 LUT 实现函数 $f = x_1x_2x_3 + \overline{x_1}\overline{x_2}\overline{x_3}$ 。

解:因为三变量真值表有8行,所以它有8个存储单元。函数 $f = x_1x_2x_3 + \overline{x_1}\overline{x_2}\overline{x_3}$ 的真值表如图5-13(a)所示。实现函数 $f = x_1x_2x_3 + \overline{x_1}\overline{x_2}\overline{x_3}$ 的三输入 LUT 如图 5-13(b) 所示。

前面介绍过,PAL除了含有与阵列和或阵列之外,还有一些附加电路。FPGA的可配置逻辑块中除了有LUT之外,也同样包含一些附加电路。例如,可以在其中添加一个触发器,如图5-14所示,这样可以实现时序逻辑电路。

5.4.2 FPGA 的编程

通常一个LUT的输入个数是4或5,因而存储单元的个数也就是16或32。由于一个可配置逻辑块的规模较小,它只能实现规模足够小的逻辑函数。对于规模较大的逻辑电路

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

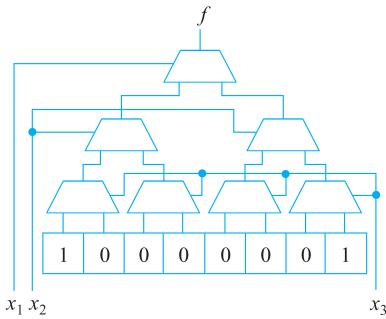
(a) 函数 $f=x_1x_2x_3+\bar{x}_1\bar{x}_2\bar{x}_3$ 的真值表(b) 实现函数 $f=x_1x_2x_3+\bar{x}_1\bar{x}_2\bar{x}_3$ 的三输入 LUT

图 5-13 例 5-3 图

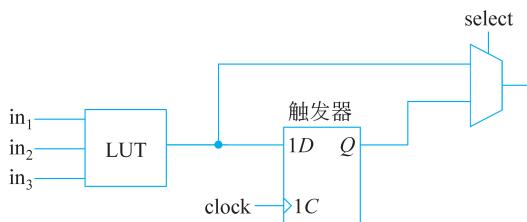


图 5-14 在 FPGA 的逻辑块中添加一个触发器

来说,为了能装入 FPGA 芯片,必须被分解为若干规模较小的子电路,以适应可配置逻辑块的规模。实际上,把用户设计的电路转化为满足 FPGA 所需的形式,是由 EDA 工具自动实现的。如果一个电路用 FPGA 实现,则 FPGA 的可配置逻辑块被编程以实现必需的逻辑功能,布线通道被编程以实现可配置逻辑块之间的互连。

FPGA 采用 ISP 方法。FPGA 中 LUT 的存储单元是易失性的,一旦电源关闭,则芯片中的内容全部丧失。因此,每次加电时都必须对 FPGA 进行编程。通常采用的方法是:在同一个印制电路板上焊有一片可编程只读存储器(Programmable Read-Only Memory, PROM)芯片,其中保存着 FPGA 的编程数据,当给该 FPGA 所在的印制电路板加电时,数据将自动从 PROM 中加载到 FPGA 的存储单元中。

5.4.3 典型的 FPGA 芯片

XC4000 系列 FPGA 是 Xilinx 公司于 20 世纪 90 年代末推出的最成功的产品之一,是 Xilinx 公司最成熟的 FPGA 产品,是查找表结构 FPGA 的代表。

XC4000 系列芯片的结构与图 5-10 所示的 FPGA 的结构很类似。它有一个二维的可配置逻辑块阵列,这些可配置逻辑块可以使用垂直和水平的布线通道互相连接起来。XC4000 系列芯片的规模从 XC4002 到 XC40250 为 2000~250 000 个等效的逻辑门。

图 5-15 是简化的可配置逻辑块结构,一个可配置逻辑块包含两个四输入 LUT,这样可以实现四变量的任意逻辑函数。这些 LUT 的输出可以可选地保存在触发器中。可配置逻辑块中还包含了一个三输入 LUT,它的两个输入可以来自 F 和 G ,其余的输入来自可配置逻辑块外,所以一个可配置逻辑块可以实现多达 9 个变量的逻辑函数。

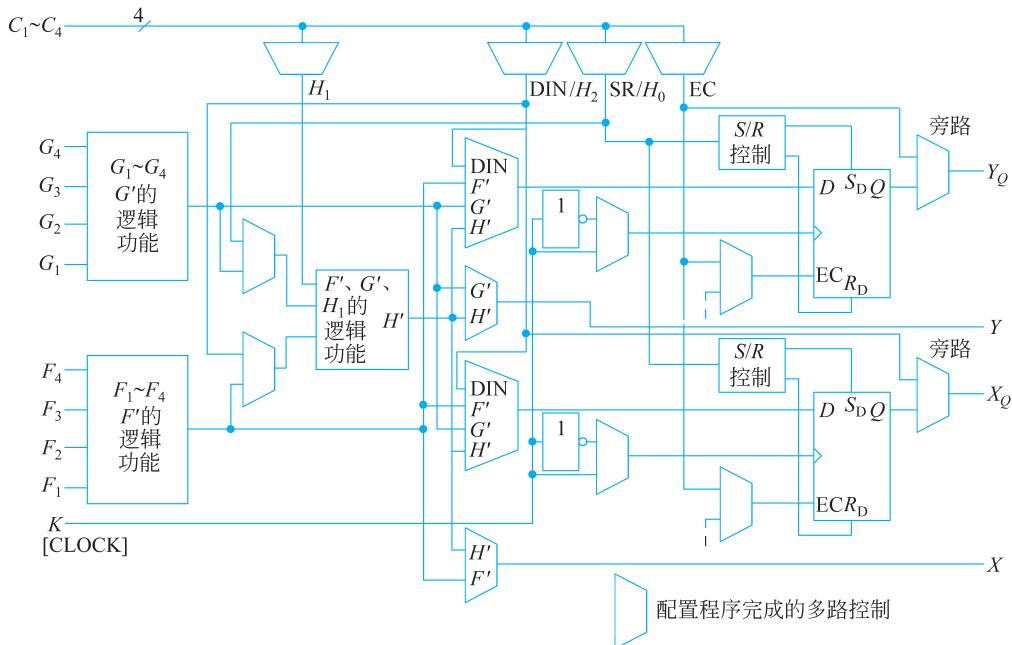


图 5-15 简化的 XC4000 芯片的可配置逻辑块结构

5.5 标准单元和定制芯片

在 PLD 中通过对可编程开关的编程实现用户所需的电路。可编程开关的存在占用了芯片的一部分面积,影响了可装入 PLD 电路的规模,并降低了电路的运行速度。

在电路规模很大、运行速度要求极高的情况下,通常使用定制芯片(custom chip),定制芯片中不含可编程开关,其中包含大量的逻辑门,运行速度也远远高于 PLD。定制芯片的设计者有很大的自由度,自己决定电路形式、芯片的规模、晶体管在芯片中的位置以及晶体管的互连方式。通常把确定晶体管的位置及相互连接的过程叫作布线(layout)。定制芯片的设计者完全可以自由地创建自己希望的任何布线。定制芯片所包含的晶体管数可以超过数百万,因而布线所需的工作量十分大,成本十分昂贵,而且只有在销售量很大的情况下才能补偿设计成本。定制芯片最常见的例子是微处理器和存储器芯片。

使用标准单元(standard cell)可以减少定制芯片设计的工作量,使用这种技术制成的芯片通常叫作应用专用集成电路(application specific integrated circuit, ASIC)。所谓标准单元是由 ASIC 销售商提供的一些已经完成了连接和紧密布线的晶体管单元。一般来说,ASIC 中使用的标准单元可以有许多种,预先存储在库中,设计者可以随时选用。ASIC 销售商还提供各自的 EDA 工具,整个芯片的布线图由 EDA 工具自动实现。

5.6 可编程器件的设计流程

可编程器件设计是指利用 EDA 工具和编程器对器件进行开发的过程。一般来说,可编程器件设计包括设计准备、设计输入、设计处理和器件编程 4 个设计步骤以及相应功能