

PLD/FPGA 结构与原理初步（一）

一. 基于乘积项 (Product-Term) 的 PLD 结构

采用这种结构的 PLD 芯片有: Altera 的 MAX7000, MAX3000 系列 (EEPROM 工艺), Xilinx 的 XC9500 系列 (Flash 工艺) 和 Lattice, Cypress 的大部分产品 (EEPROM 工艺)

我们先看一下这种 PLD 的总体结构 (以 MAX7000 为例, 其他型号的结构与此都非常相似):

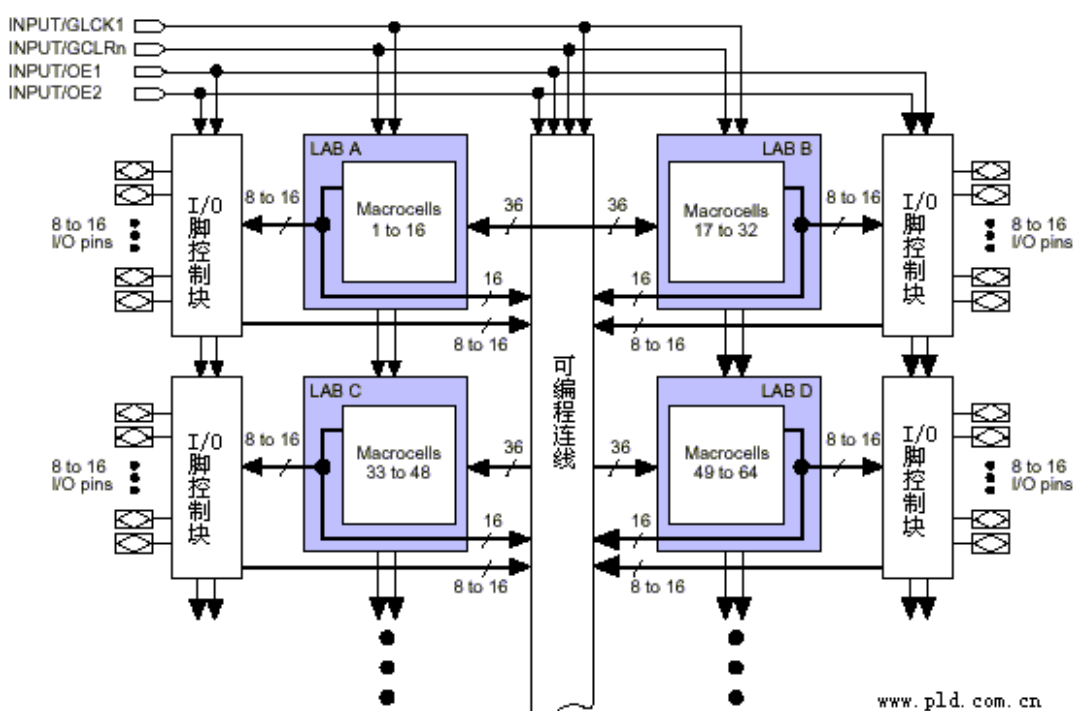


图 1 基于乘积项的 PLD 内部结构

这种 PLD 可分为三块结构: 宏单元 (Macrocell), 可编程连线 (PIA) 和 I/O 控制块。宏单元是 PLD 的基本结构, 由它来实现基本的逻辑功能。图 1 中蓝色部分是多个宏单元的集合 (因为宏单元较多, 没有一一画出)。可编程连线负责信号传递, 连接所有的宏单元。I/O 控制块负责输入输出的电气特性控制, 比如可以设定集电极开路输出, 摆率控制, 三态输出等。图 1 左上的 INPUT/GCLK1, INPUT/GCLRn, INPUT/OE1, INPUT/OE2 是全局时钟, 清零和输出使能信号, 这几个信号有专用连线与 PLD 中每个宏单元相连, 信号到每个宏单元的延时相同并且延时最短。

宏单元的具体结构见下图:

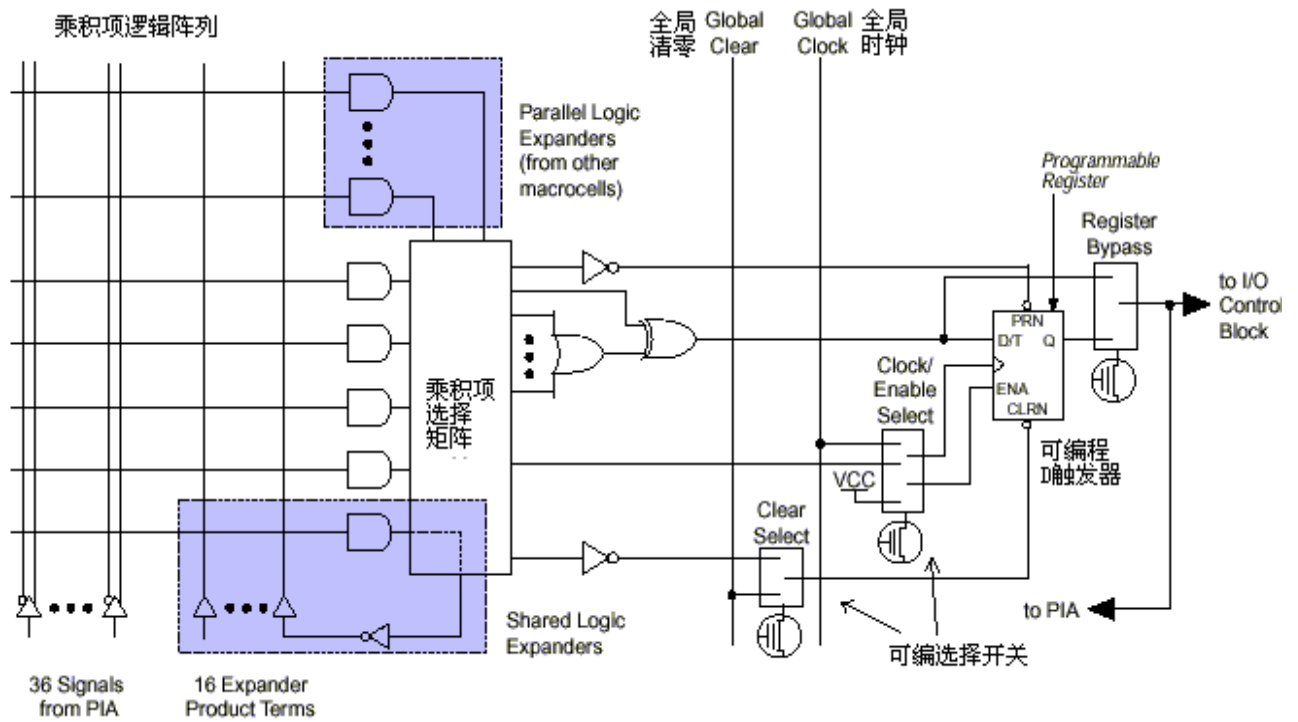


图2 宏单元结构

左侧是乘积项阵列，实际就是一个与或阵列，每一个交叉点都是一个可编程熔丝，如果导通就是实现“与”逻辑。后面的乘积项选择矩阵是一个“或”阵列。两者一起完成组合逻辑。图右侧是一个可编程D触发器，它的时钟，清零输入都可以编程选择，可以使用专用的全局清零和全局时钟，也可以使用内部逻辑（乘积项阵列）产生的时钟和清零。如果不需要触发器，也可以将此触发器旁路，信号直接输给PIA或输出到I/O脚。

二. 乘积项结构PLD的逻辑实现原理

下面我们以一个简单的电路为例，具体说明PLD是如何利用以上结构实现逻辑的，电路如下图：

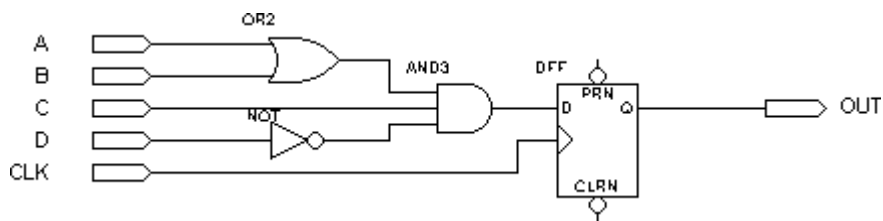


图3

假设组合逻辑的输出 (AND3 的输出) 为 f , 则 $f = (A+B) * C * (!D) = A * C * !D + B * C * !D$ (我们以 $!D$ 表示 D 的“非”)

PLD 将以下面的方式来实现组合逻辑 f :

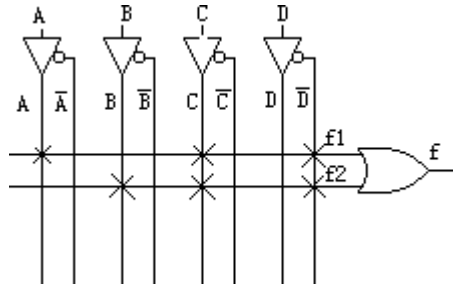


图 4

A, B, C, D 由 PLD 芯片的管脚输入后进入可编程连线阵列 (PIA), 在内部会产生 A, A 反, B, B 反, C, C 反, D, D 反 8 个输出。图中每一个叉表示相连(可编程熔丝导通), 所以得到: $f = f1 + f2 = (A * C * !D) + (B * C * !D)$ 。这样组合逻辑就实现了。图 3 电路中 D 触发器的实现比较简单, 直接利用宏单元中的可编程 D 触发器来实现。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的全局时钟专用通道, 直接连接到可编程触发器的时钟端。可编程触发器的输出与 I/O 脚相连, 把结果输出到芯片管脚。这样 PLD 就完成了图 3 所示电路的功能。(以上这些步骤都是由软件自动完成的, 不需要人为干预)

图 3 的电路是一个很简单的例子, 只需要一个宏单元就可以完成。但对于一个复杂的电路, 一个宏单元是不能实现的, 这时就需要通过并联扩展项和共享扩展项将多个宏单元相连, 宏单元的输出也可以连接到可编程连线阵列, 再做为另一个宏单元的输入。这样 PLD 就可以实现更复杂逻辑。

这种基于乘积项的 PLD 基本都是由 EEPROM 和 Flash 工艺制造的, 一上电就可以工作, 无需其他芯片配合。

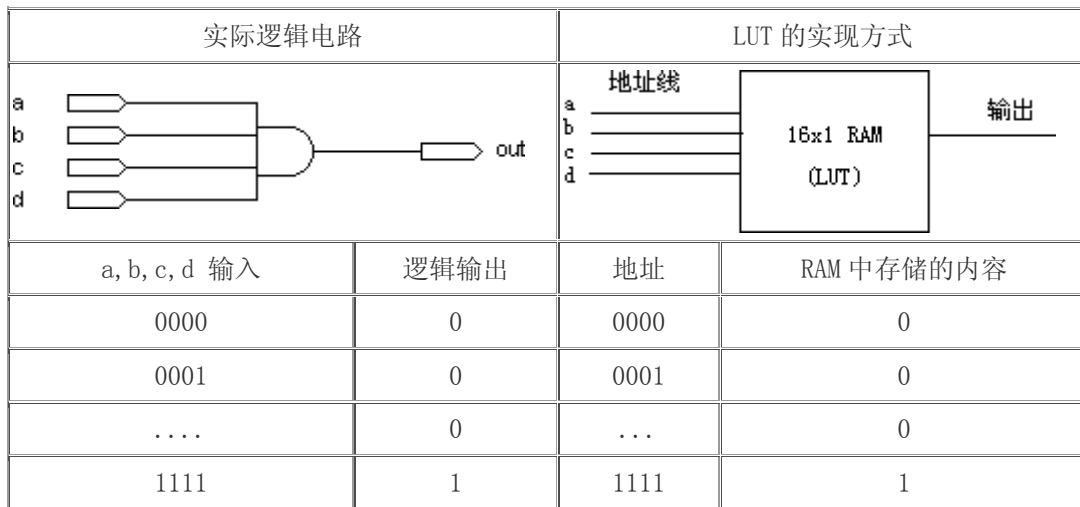
PLD/FPGA 结构与原理初步 (二)

一. 查找表 (Look-Up-Table) 的原理与结构

采用这种结构的 PLD 芯片我们也可以称之为 FPGA: 如 altera 的 ACEX, APEX 系列, xilinx 的 Spartan, Virtex 系列等。

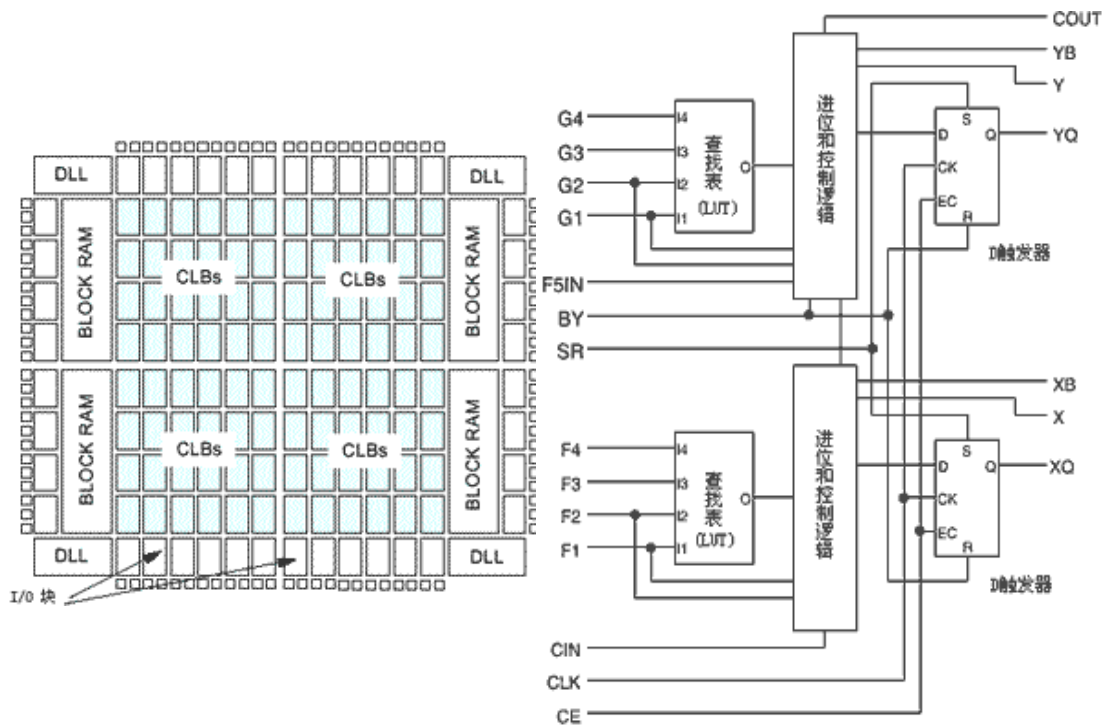
查找表 (Look-Up-Table) 简称为 LUT, LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT, 所以每一个 LUT 可以看成是一个有 4 位地址线的 16×1 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后, PLD/FPGA 开发软件会自动计算逻辑电路的所有可能的结果, 并把结果事先写入 RAM, 这样, 每输入一个信号进行逻辑运算就等于输入一个地址进行查表, 找出地址对应的内容, 然后输出即可。

下面是一个 4 输入与门的例子，



二. 基于查找表 (LUT) 的 FPGA 的结构

我们看一看 xilinx Spartan-II 的内部结构，如下图：

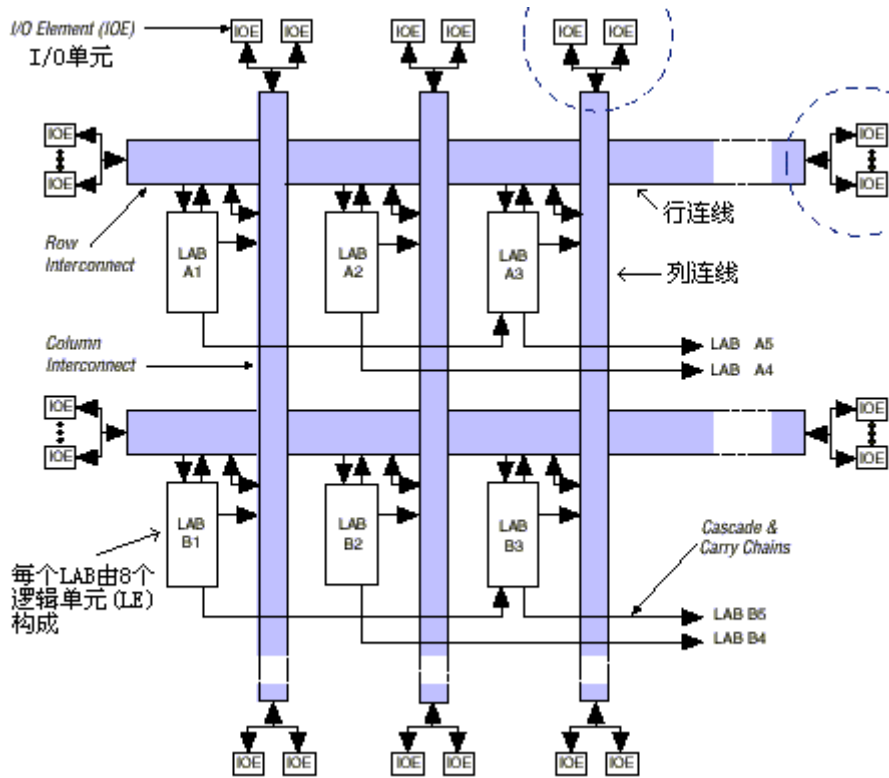


xilinx Spartan-II 芯片内部结构

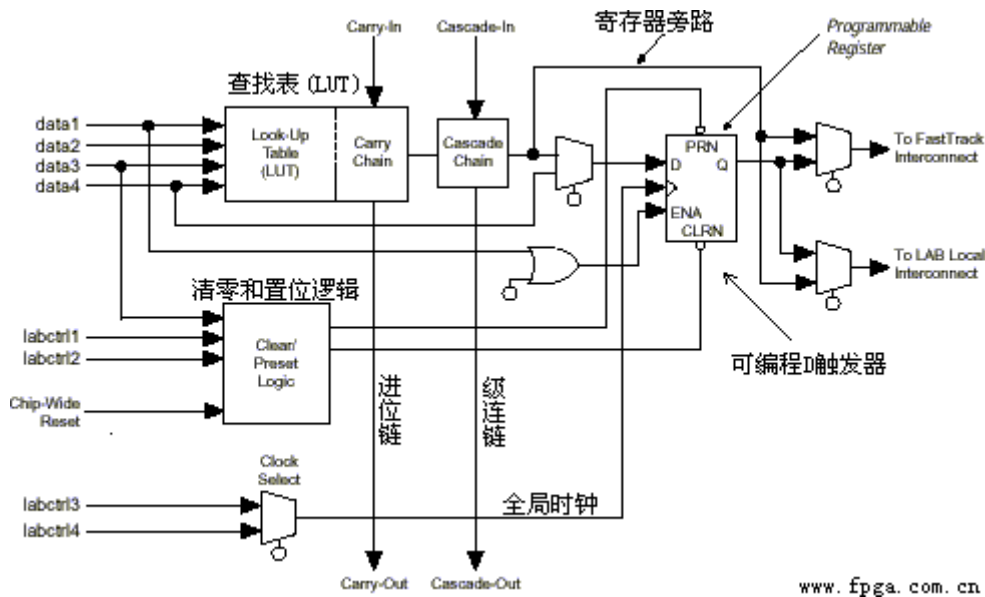
Slices 结构

Spartan-II 主要包括 CLBs, I/O 块, RAM 块和可编程连线 (未表示出)。在 spartan-II 中, 一个 CLB 包括 2 个 Slices, 每个 slices 包括两个 LUT, 两个触发器和相关逻辑。Slices 可以看成是 SpartanII 实现逻辑的最基本结构 (xilinx 其他系列, 如 SpartanXL, Virtex 的结构与此稍有不同, 具体请参阅数据手册)

altera 的 FLEX/ACEX 等芯片的结构如下图:



altera FLEX/ACEX 芯片的内部结构

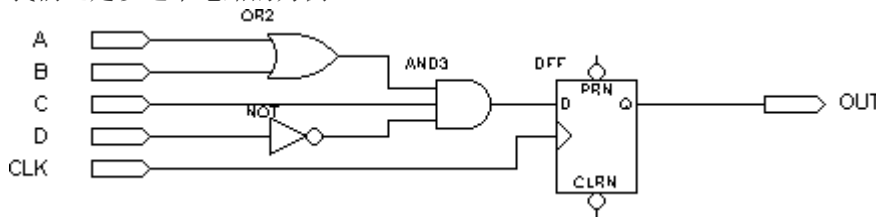


逻辑单元 (LE) 内部结构

FLEX/ACEX 的结构主要包括 LAB, I/O 块, RAM 块 (未表示出) 和可编程行/列连线。在 FLEX/ACEX 中, 一个 LAB 包括 8 个逻辑单元 (LE), 每个 LE 包括一个 LUT, 一个触发器和相关的相关逻辑。LE 是 FLEX/ACEX 芯片实现逻辑的最基本结构 (altera 其他系列, 如 APEX 的结构与此基本相同, 具体请参阅数据手册)

二. 查找表结构的 FPGA 逻辑实现原理

我们还是以这个电路的为例:



A, B, C, D 由 FPGA 芯片的管脚输入后进入可编程连线, 然后作为地址线连到到 LUT, LUT 中已经事先写入了所有可能的逻辑结果, 通过地址查找到相应的数据然后输出, 这样组合逻辑就实现了。该电路中 D 触发器是直接利用 LUT 后面 D 触发器来实现。时钟信号 CLK 由 I/O 脚输入后进入芯片内部时钟专用通道, 直接连接到触发器的时钟端。触发器的输出与 I/O 脚相连, 把结果输出到芯片管脚。这样 PLD 就完成了图 3 所示电路的功能。(以上这些步骤都是由软件自动完成的, 不需要人为干预)

这个电路是一个很简单的例子, 只需要一个 LUT 加上一个触发器就可以完成。对于一个 LUT 无法完成的电路, 就需要通过进位逻辑将多个单元相连, 这样 FPGA 就可以实现复杂的逻辑。

由于 LUT 主要适合 SRAM 工艺生产，所以目前大部分 FPGA 都是基于 SRAM 工艺的，而 SRAM 工艺的芯片在掉电后信息就会丢失，一定需要外加一片专用配置芯片，在上电的时候，由这个专用配置芯片把数据加载到 FPGA 中，然后 FPGA 就可以正常工作，由于配置时间很短，不会影响系统正常工作。也有少数 FPGA 采用反熔丝或 Flash 工艺，对这种 FPGA，就不需要外加专用的配置芯片。

三. 其他类型的 FPGA 和 PLD

随着技术的发展，在 2004 年以后，一些厂家推出了一些新的 PLD 和 FPGA，这些产品模糊了 PLD 和 FPGA 的区别。例如 Altera 最新的 MAXII 系列 PLD，这是一种基于 FPGA (LUT) 结构，集成配置芯片的 PLD，在本质上它就是一种在内部集成了配置芯片的 FPGA，但由于配置时间极短，上电就可以工作，所以对用户来说，感觉不到配置过程，可以传统的 PLD 一样使用，加上容量和传统 PLD 类似，所以 altera 把它归作 PLD。还有像 Lattice 的 XP 系列 FPGA，也是使用了同样的原理，将外部配置芯片集成到内部，在使用方法上和 PLD 类似，但是因为容量大，性能和传统 FPGA 相同，也是 LUT 架构，所以 Lattice 仍把它归为 FPGA。

四. 选择 PLD 还是 FPGA?

根据上一篇 PLD 的结构和原理可以知道，PLD 分解组合逻辑的功能很强，一个宏单元就可以分解十几个甚至 20—30 多个组合逻辑输入。而 FPGA 的一个 LUT 只能处理 4 输入的组合逻辑，因此，PLD 适合于设计译码等复杂组合逻辑。但 FPGA 的制造工艺确定了 FPGA 芯片中包含的 LUT 和触发器的数量非常多，往往都是几千上万，PLD 一般只能做到 512 个逻辑单元，而且如果用芯片价格除以逻辑单元数量，FPGA 的平均逻辑单元成本大大低于 PLD。所以如果设计中使用到大量触发器，例如设计一个复杂的时序逻辑，那么使用 FPGA 就是一个很好选择。同时 PLD 拥有上电即可工作的特性，而大部分 FPGA 需要一个加载过程，所以，如果系统要可编程逻辑器件上电就要工作，那么就应该选择 PLD。