

这个章节包含了定义逻辑单元 (LE) 和逻辑阵列模块 (LAB) 的特性。具体信息体现在 LE 如何运作, LAB 如何容纳 LE 组, 以及 LAB 接口如何与 Cyclone® IV 器件中的其他模块连接。

逻辑单元

逻辑单元 (LE) 在 Cyclone IV 器件结构中最小的逻辑单位。LE 紧密且有效的提供了高级功能的逻辑使用。每个 LE 有以下特性:

- 一个四口输入的查找表 (LUT), 以实现四种变量的任何功能
- 一个可编程的寄存器
- 一个进位链连接
- 一个寄存器链连接
- 可驱动以下的相互连接:
 - 本地
 - 行
 - 列
 - 寄存器链
 - 直联
- 寄存器套包支持
- 寄存器反馈支持

除了三个通用布线输出之外，在一个 LAB 上的 LE 有寄存器链输出，使得同一个 LAB 中的寄存器能够串联在一起。寄存器链输出使 LUT 能够被用于组合功能，实现寄存器被用于一个不相关的移位寄存器。这些资源加速了 LAB 之间的连接，同时节省本地互联资源。

LE 操作模式

Cyclone IV LE 在以下模式下操作：

- 正常模式
- 算术模式

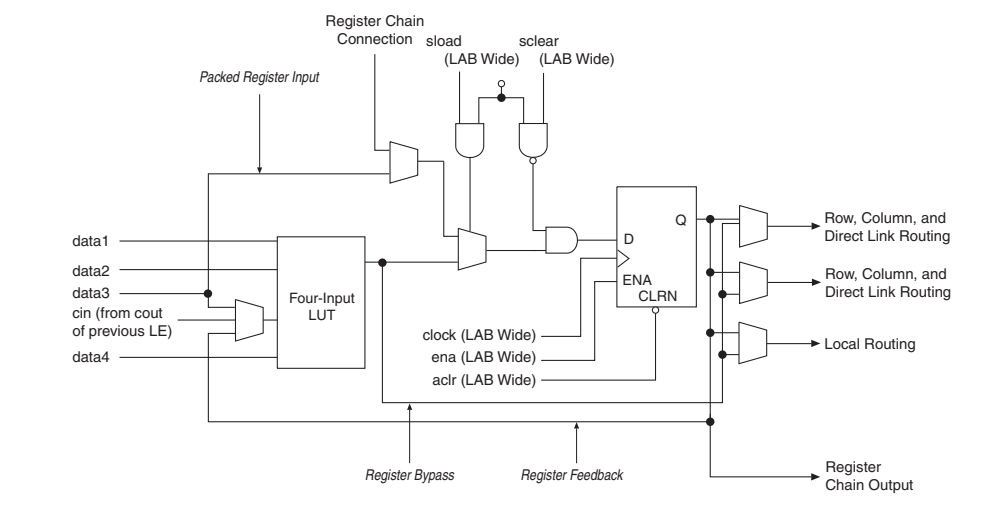
The Quartus® II 软件自动为普通功能选择适用的模式，例如计数器，加法器，减法器 和算术功能，与参数化功能如参数化模块库 (LPM) 功能一起。如果需要，您也可以创建指定的特用功能以用于 LE 操作模式的性能优化。

正常模式

正常模式适用于一般的逻辑运用和组合功能。在正常模式中，来自 LAB 本地互联的四个数据输入输入到一个四口输入的 LUT 中（图 2-2）。Quartus II 编译器自动选择 carry-in (cin) 或 data3 信号作为其中一个输入到 LUT。LE 在正常模式中支持套包寄存器和寄存器的反馈。

图 2-2 显示在正常模式中的 LE。

图 2-2. 在 Cyclone IV 器件上正常模式中的 LE

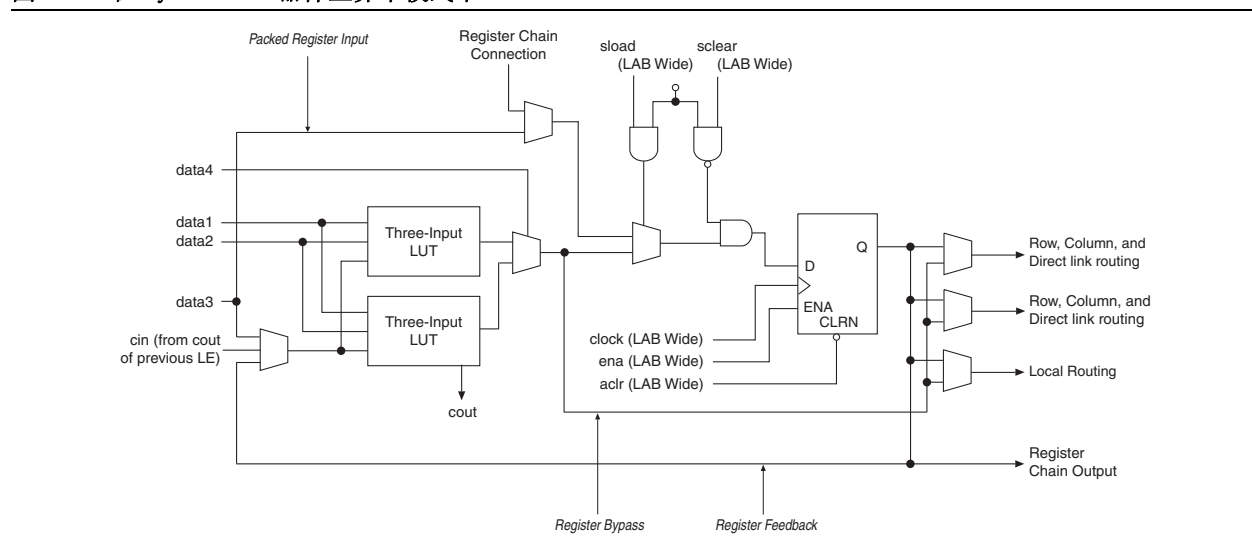


算术模式

算术模式对于加法器，计数器，蓄能器和比较器的实现是理想的。一个 LE 在算术模式中实现一个 2 位全加器和基本的进位链（图 2-3）。LE 在算术模式中驱动 LUT 输出存储与未存储的版本。寄存器反馈和寄存器套包都支持 LE 用于算术模式。

图 2-3 显示在算术模式中的 LE。

图 2-3. 在 Cyclone IV 器件上算术模式中 LE



Quartus II 编译器在设计处理期间自动创建进位链逻辑。您也可以在设计输入期间手动创建进位链逻辑。参数化功能如 LPM 功能，自动优化进位链于合适的功能中。

Quartus II 编译器通过同一列中自动链接 LAB 创建长过 16 个 LE 的进位链。为了增强布局，一条长进位链纵向运行，通过直接链接互联实现迅速横向连接到 M9K 存储器模块或嵌入式乘法器。例如，如果一个设计有一条长进位链在一 LAB 列中与一系列 M9K 存储器模块紧挨着，那么任何 LE 输出通过直接链接互联都可以连到相邻的 M9K 存储器模块上。如果进位链横向运行，任何不与 M9K 存储器模块列紧挨的 LAB 使用其他的行和列互联以驱动一个 M9K 存储器模块。一条进位链继续运行直到全列。

逻辑阵列模块

逻辑阵列模块 (LAB) 包含 LE 组。

拓扑

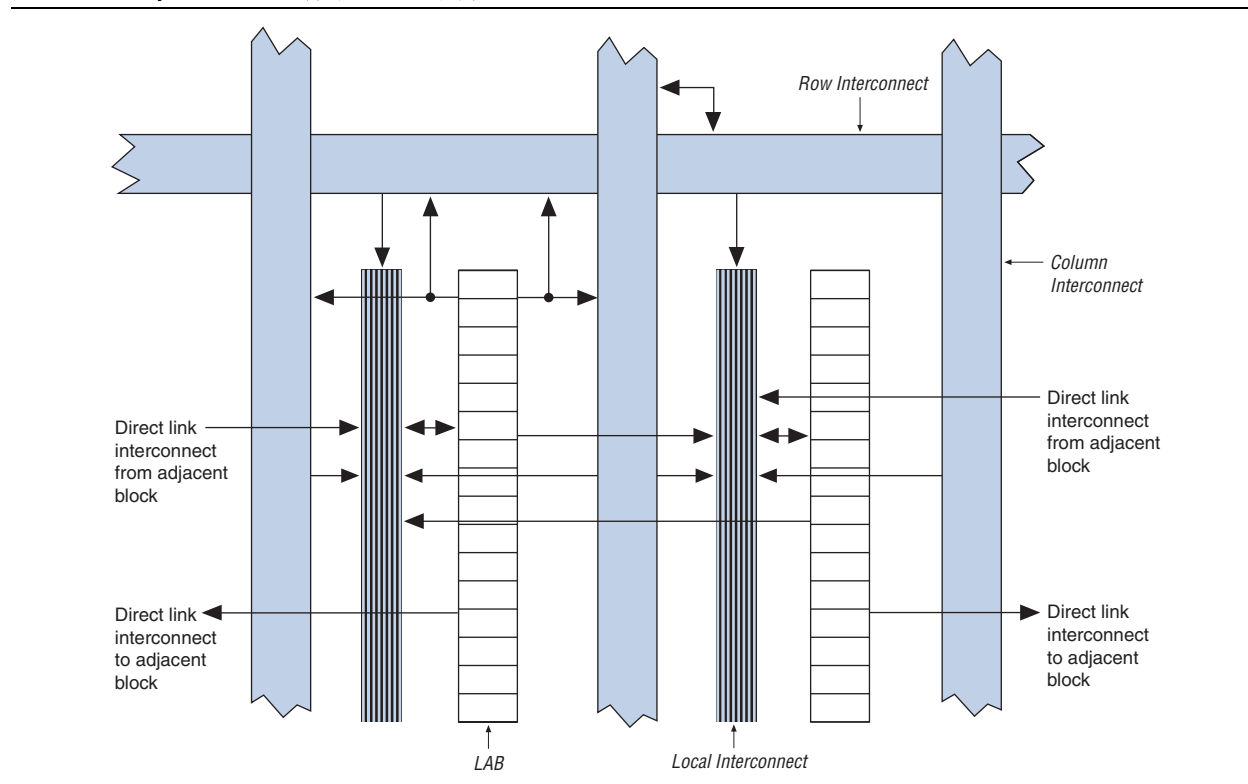
每个 LAB 包括以下特性:

- 16 LE
- LAB 控制信号
- LE 进位链
- 寄存器链
- 本地互联

本地互联在同一个 LAB 的 LE 之间传输信号。寄存器链连接把一个 LE 寄存器的输出传输到 LAB 中相邻的 LE 寄存器上。Quartus II 编译器放置相关的逻辑在 LAB 或相邻的 LAB 中, 允许使用本地和寄存器链连接以实现性能和面积效率。

图 2-4 显示在 Cyclone IV 器件中的 LAB 结构。

图 2-4. 在 Cyclone IV 器件中的 LAB 结构

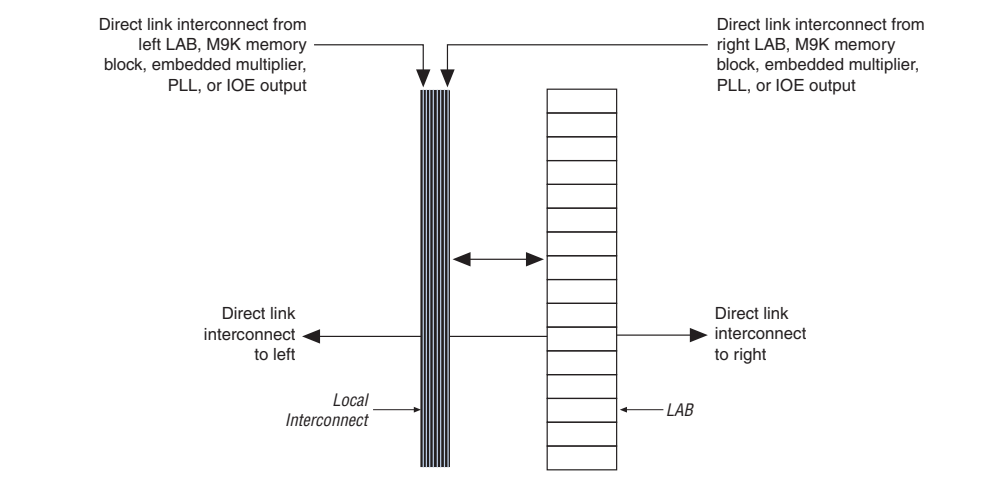


LAB 互联

LAB 本地互联是由列和行互联以及 LE 在相同的 LAB 中的输出端驱动的。邻近的 LAB, 锁相环 (PLL), M9K RAM 模块和嵌入式乘法器由左到右通过直接链接连接也可以驱动 LAB 的本地互联。直接链接连接功能最小化行和列互联的使用, 以提供更高的性能和灵活性。每个 LE 通过加速本地和直接链接互联可以驱动多达 48 个的 LE。

图 2-5 显示直接链接连接。

图 2-5. 在 Cyclone IV 器件中的直连接方式



LAB 控制信号

每个 LAB 都包含专用的逻辑以驱动控制信号各自 LE。控制信号包括:

- 两个时钟
- 两个时钟使能
- 两个异步清零
- 一个同步清零
- 一个同步加载

您可以一次使用多达 8 个的控制信号。寄存器套包和同步加载不能被同时使用。

每个 LAB 可以有多达四个的非全局控制信号。您可以使用其他的 LAB 控制信号只要这些信号是全局信号。

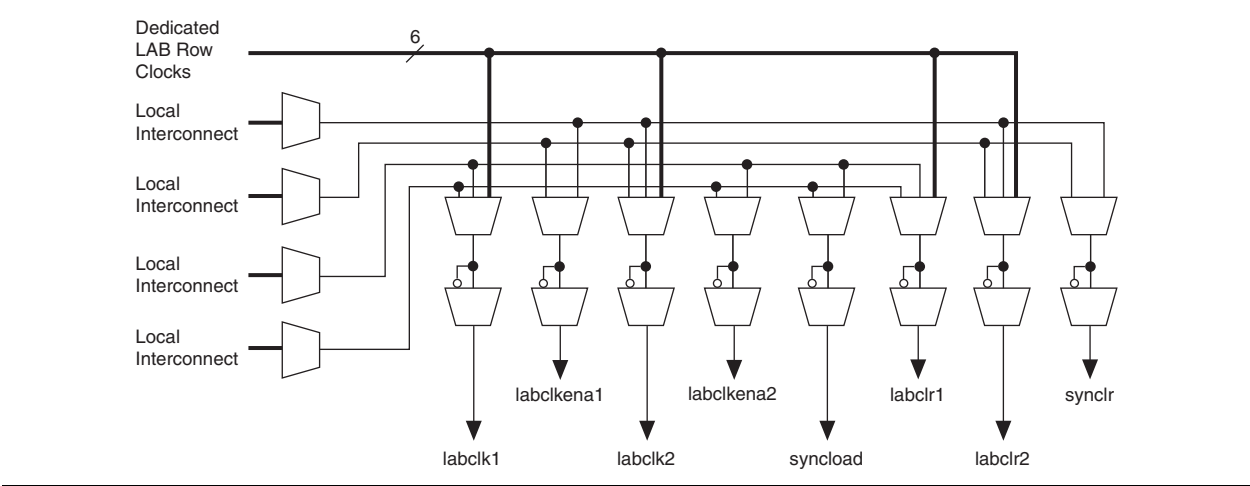
同步清零和加载对于实现计数和其他功能是很有用的。同步清零和同步加载是 LAB-wide 信号, 其影响所有寄存器上的 LAB。

每个 LAB 可以使用两个时钟和两个时钟使能信号。时钟和时钟使能信号在各个 LAB 上是相连接的。例如，在一特定的 LAB 上的任何一个 LE 中使用 labclk1 信号时也在使用 labclkena1。如果 LAB 同时使用上升沿和下降沿时钟，也使用 LAB-wide 时钟信号。释放时钟使能信号关闭 LAB-wide 时钟。

LAB 行时钟 [5..0] 和 LAB 本地的互联生成 LAB-wide 控制信号。MultiTrack 互联固有的低偏移除数据分配之外还可以实现时钟和控制信号分配。

图 2-6 显示 LAB 控制信号生成电路。

图 2-6. Cyclone IV 器件 LAB-Wide 控制信号



LAB-wide 信号控制寄存器上清零信号的逻辑。LE 直接支持一个异步清零功能。每个 LAB 支持多达两个异步清零信号 (labclr1 和 labclr2)。

一个 LAB-wide 异步加载信号不可被用为寄存器的预置信号控制逻辑。寄存器预置是用 NOT 门推回 (push-back) 的技术完成的。Cyclone IV 器件仅支持预置或异步清零信号。

除了清零端口外，Cyclone IV 器件提供一个芯片级重置管脚 (DEV_CLRn) 对器件中所有的寄存器重置。在编辑 Quartus II 软件之前设置一个选项以控制这个管脚。这个芯片级重置覆盖了所有其他的控制信号。

文档修订历史

表 2-1 显示了本章节的修订历史。

表 2-1. 文档修订历史

| 日期 | 版本 | 修订内容 |
|-------------|-----|-------|
| 2009 年 11 月 | 1.0 | 首次发布。 |

