



EDA技术实用教程

第2章

FPGA/CPLD结构原理

2.1 概述

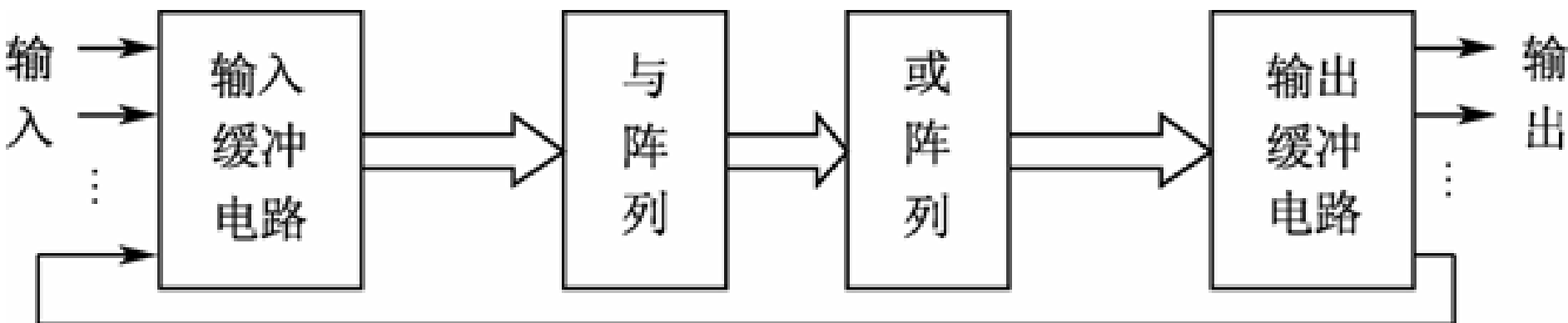


图 2-1 基本 PLD 器件的原理结构图

2.1 概述

2.1.1 PLD的发展历程

2.1.2 PLD分类

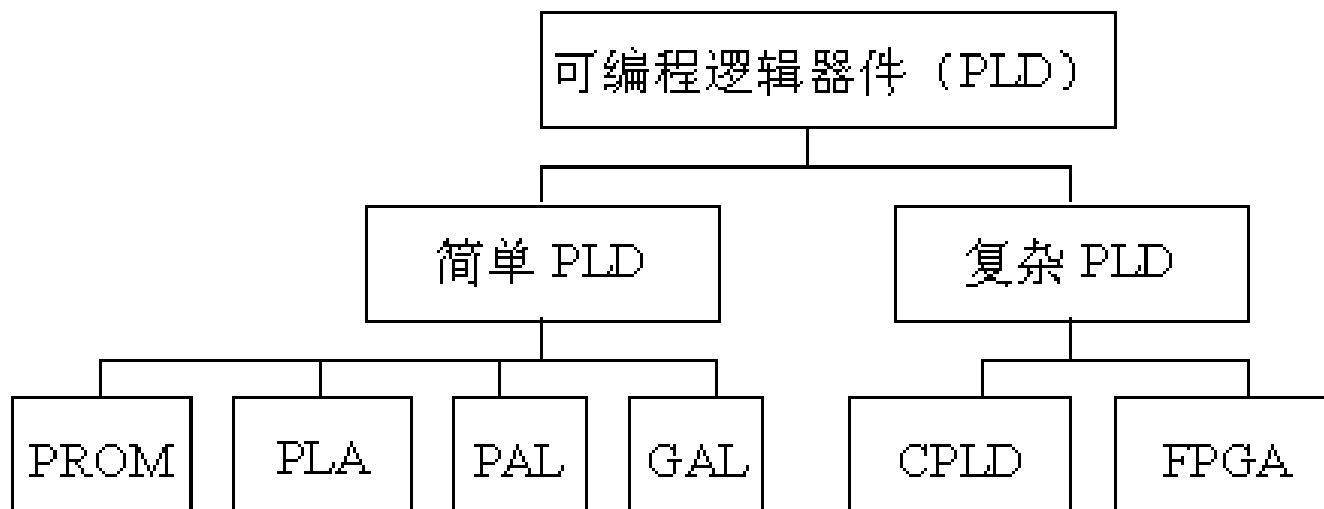


图 2-2 PLD 按集成度分类

2.2 简单PLD结构原理

2.2.1 逻辑元件符号表示

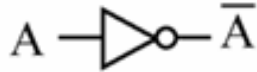



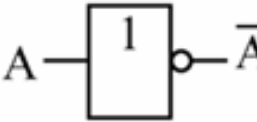
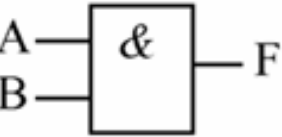
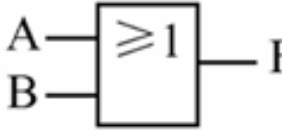
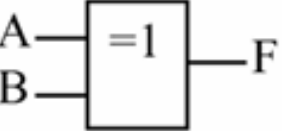
	非门	与门	或门	异或门
IEEE 1991 版 标准逻辑符号				
IEEE 1984 版 标准逻辑符号				
逻辑表达式	$\bar{A} = \text{NOT } A$	$F = A \cdot B$	$F = A + B$	$F = A \oplus B$

图 2-3 两种不同版本的国际标准逻辑门符号对照表

2.2 简单PLD结构原理

2.2.1 逻辑元件符号表示



图 2-4 PLD 的互补缓冲器

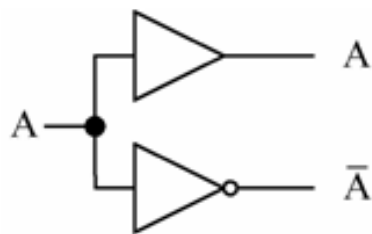


图 2-5 PLD 的互补输入



图 2-6 PLD 中与阵列的表示



图 2-7 PLD 中或阵列的表示



图 2-8 阵列线连接表示

2.2 简单PLD结构原理

2.2.2 PROM结构原理

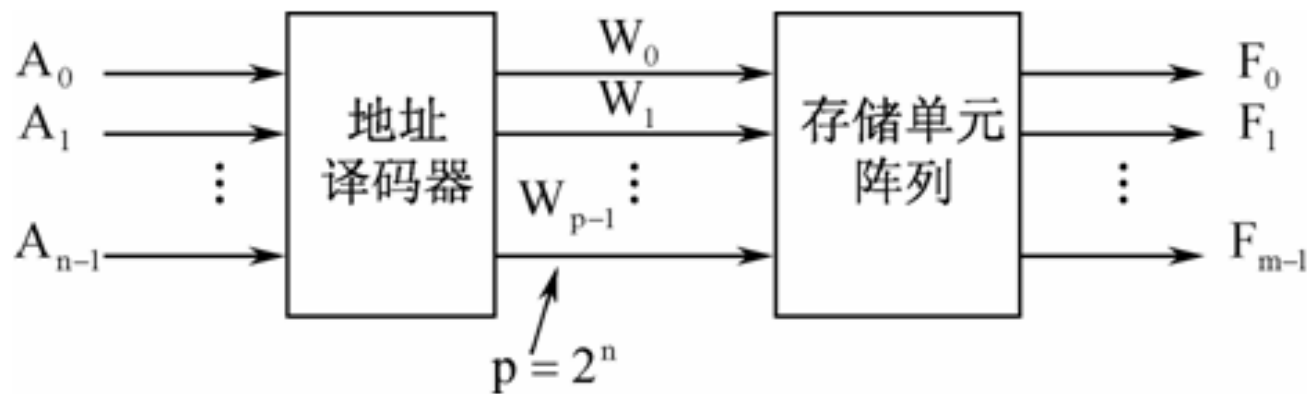


图 2-9 PROM 基本结构

$$W_0 = \bar{A}_{n-1} \cdots \bar{A}_1 \bar{A}_0$$

$$W_1 = \bar{A}_{n-1} \cdots \bar{A}_1 A_0$$

\vdots

$$W_{p-1} = A_{n-1} \cdots A_1 A_0$$

(2-1)

$$p = 2^n$$

2.2 简单PLD结构原理

2.2.2 PROM结构原理

$$\begin{aligned} F_0 &= M_{p-1,0} W_{p-1} + \cdots + M_{1,0} W_1 + M_{0,0} W_0 \\ F_1 &= M_{p-1,1} W_{p-1} + \cdots + M_{1,1} W_1 + M_{0,1} W_0 \\ &\vdots \\ F_{m-1} &= M_{p-1,m-1} W_{p-1} + \cdots + M_{1,m-1} W_1 + M_{0,m-1} W_0 \end{aligned} \quad (2-2)$$

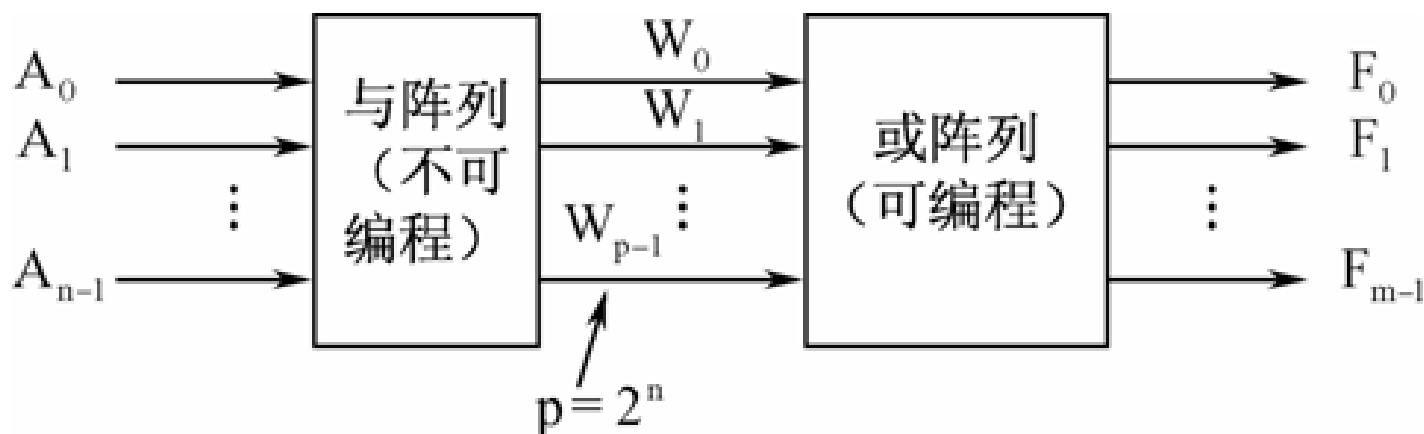


图 2-10 PROM 逻辑阵列结构

2.2 简单PLD结构原理

2.2.2 PROM结构原理

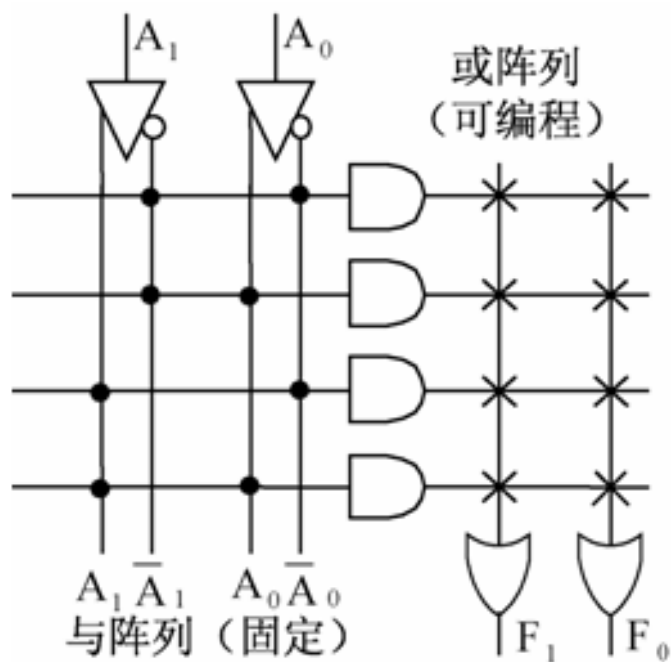


图 2-11 PROM 表达的 PLD 阵列图

2.2 简单PLD结构原理

2.2.2 PROM结构原理

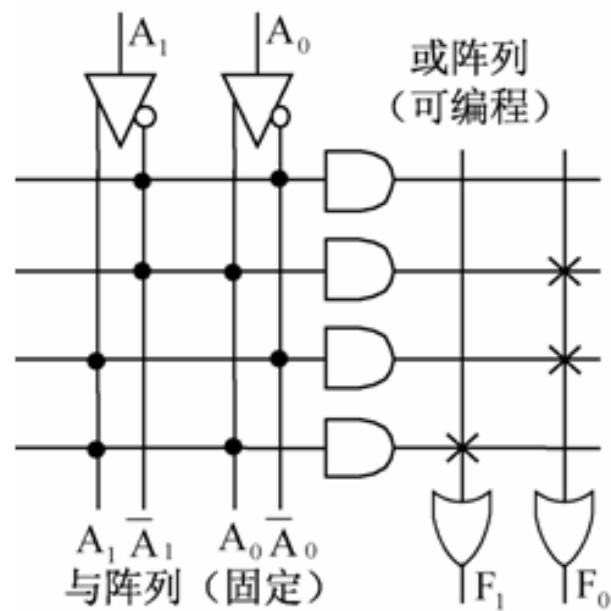


图 2-12 用 PROM 完成半加器逻辑阵列

2.2 简单PLD结构原理

2.2.3 PLA结构原理

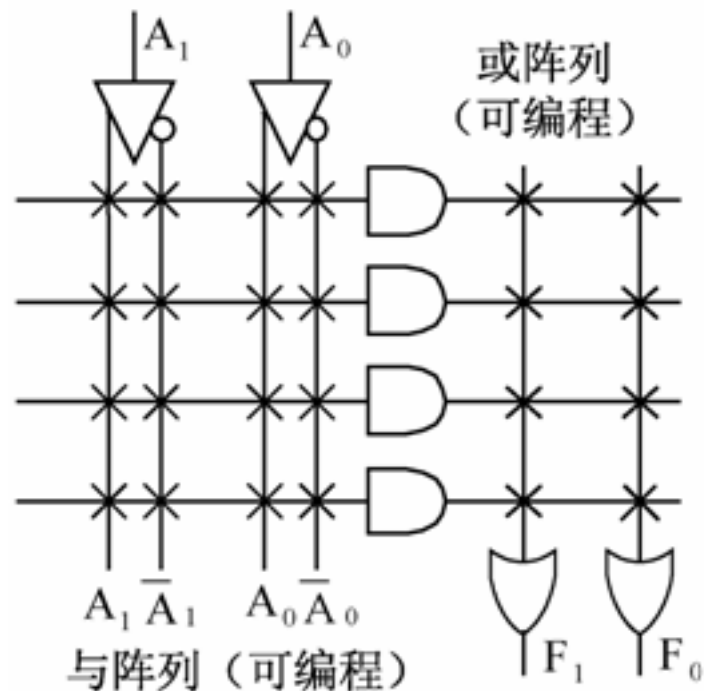


图 2-13 PLA 逻辑阵列示意图

2.2 简单PLD结构原理

2.2.3 PLA结构原理

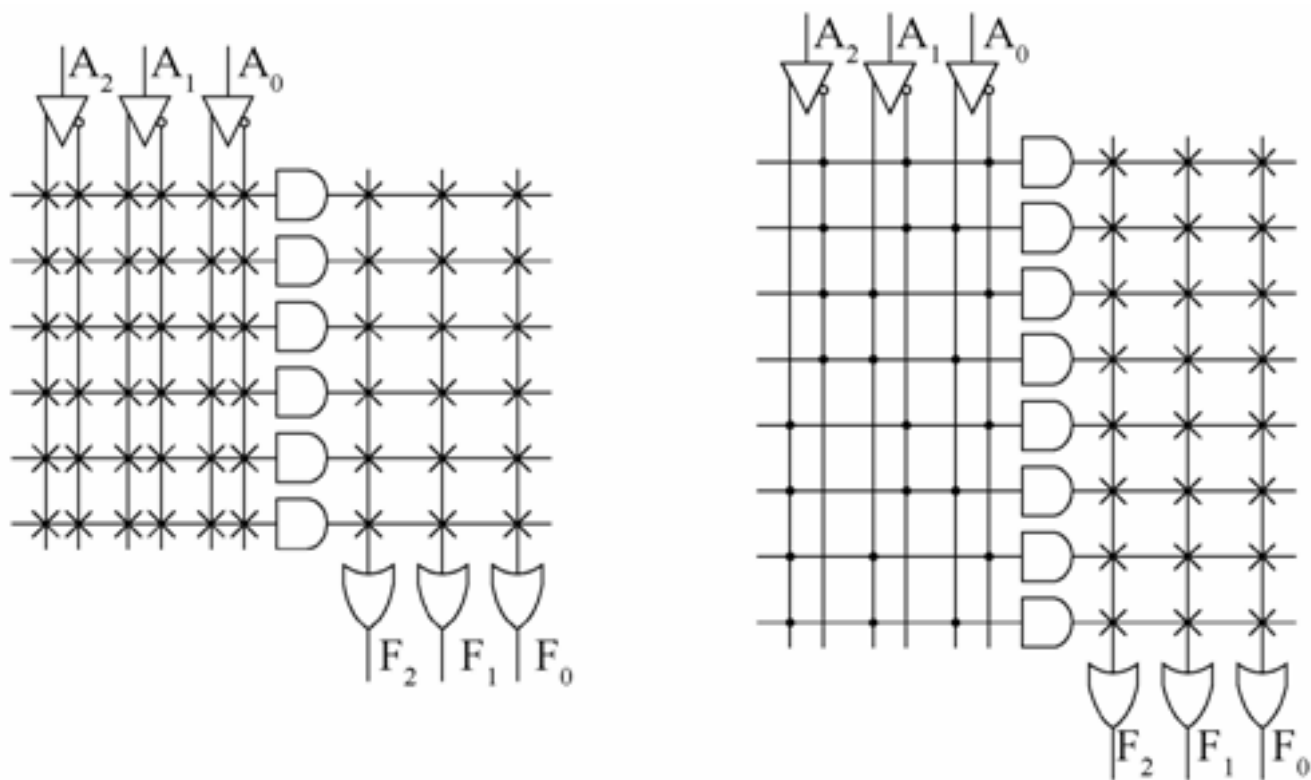


图 2-14 PLA 与 PROM 的比较

2.2 简单PLD结构原理

2.2.4 PAL结构原理

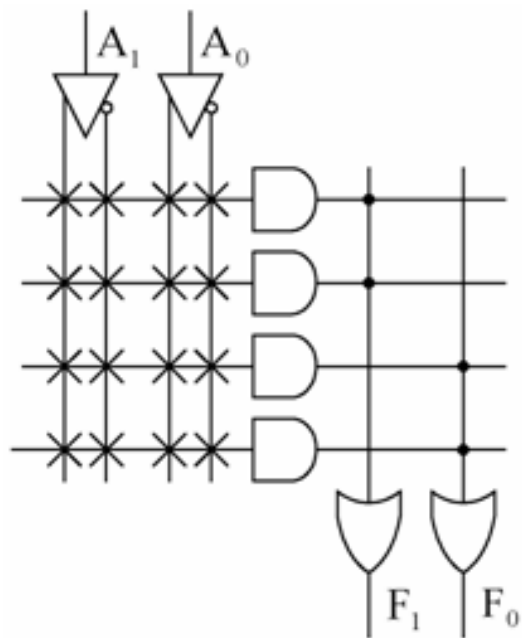


图 2-15 PAL 结构

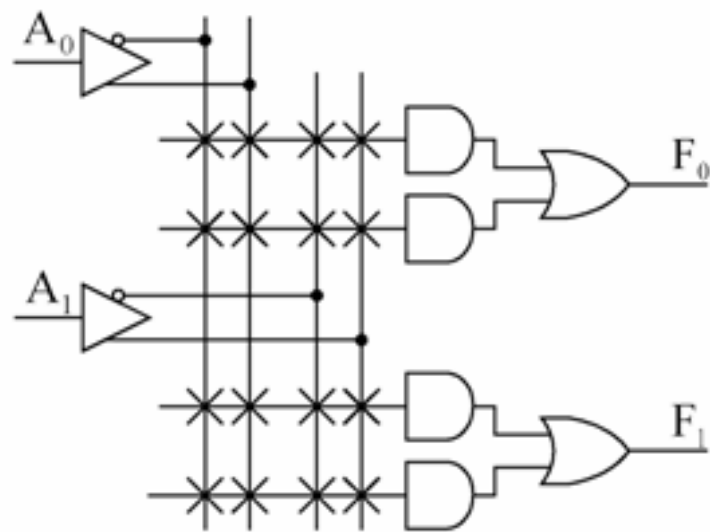


图 2-16 PAL 的常用表示

2.2 简单PLD结构原理

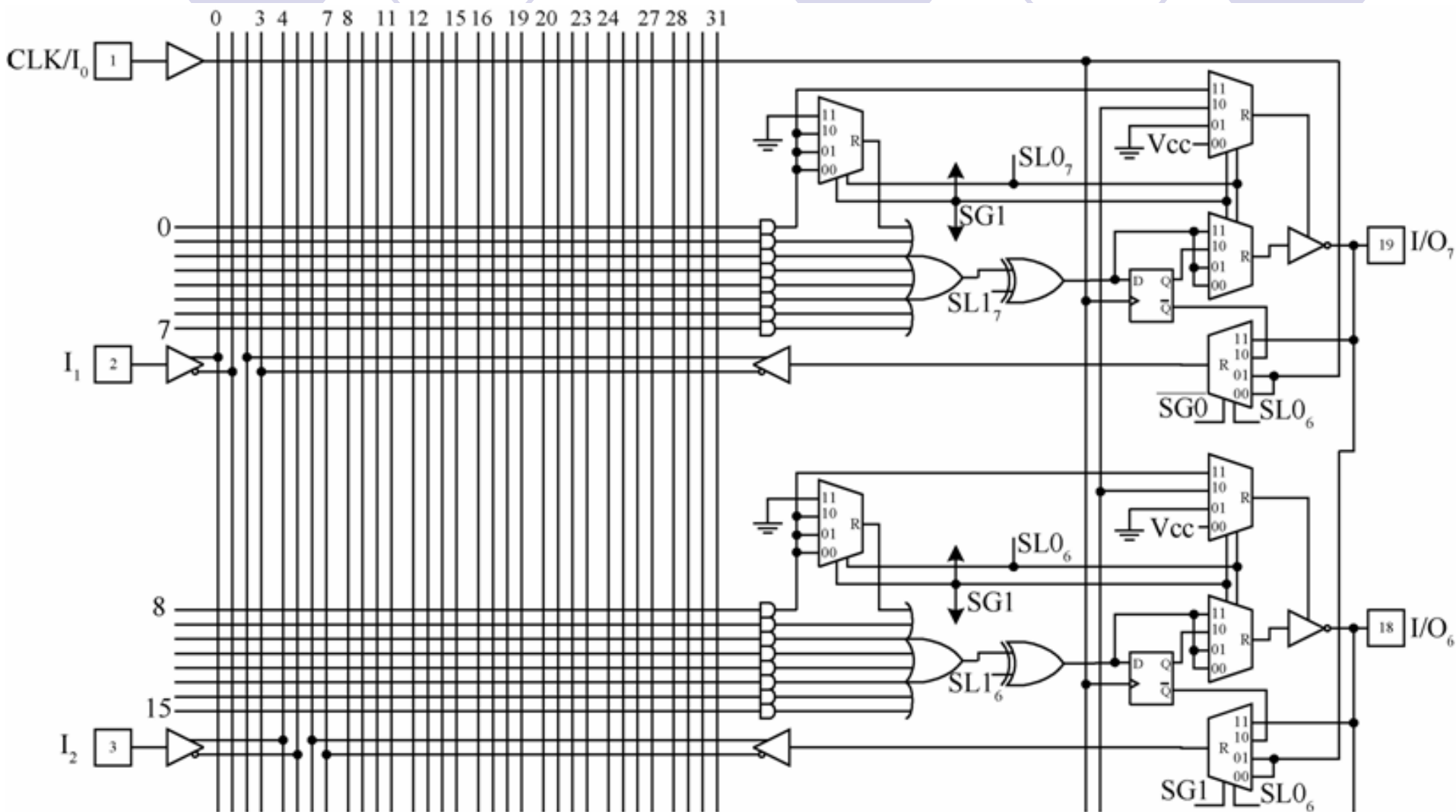
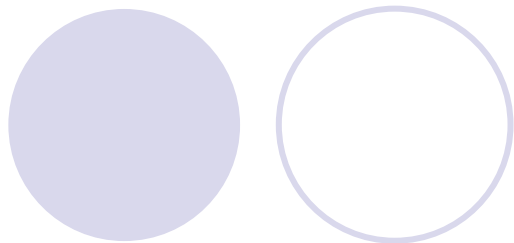


图 2-17 PAL16V8 的部分结构图



2.2.5 GAL结构原理

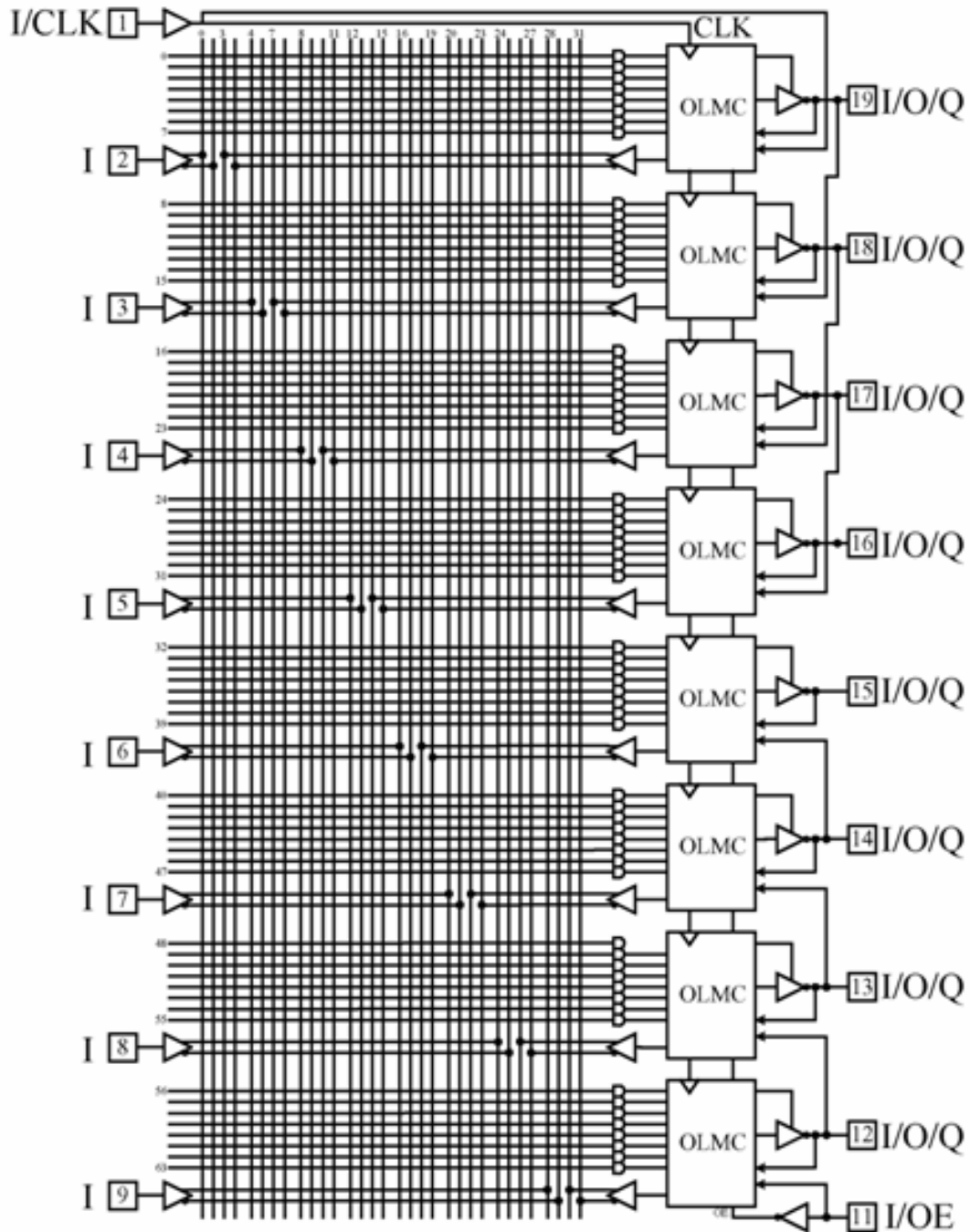


图 2-18 GAL16V8 的结构图

2.2 简单PLD结构原理

2.2.5 GAL结构原理

(1) 寄存器模式

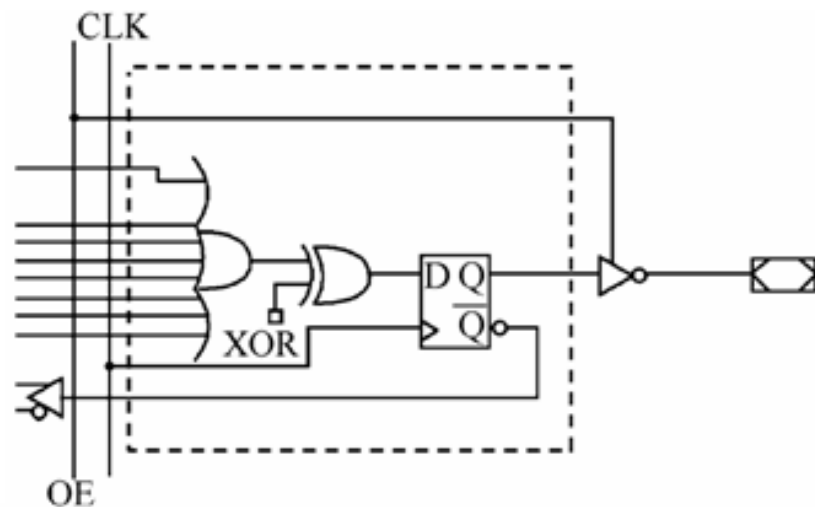


图 2-19 寄存器输出结构

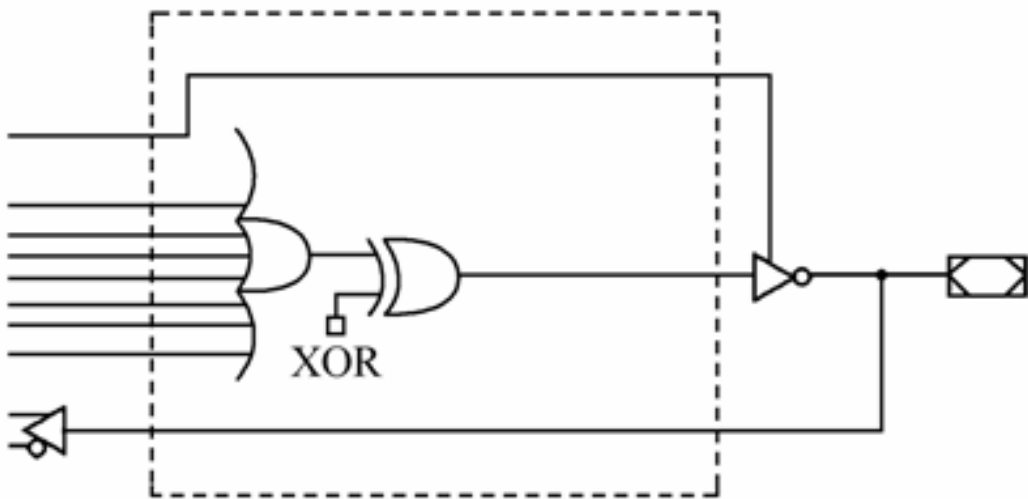


图 2-20 寄存器模式组合双向输出结构

2.2 简单PLD结构原理

(2) 复合模式

1、组合输出双向口结构

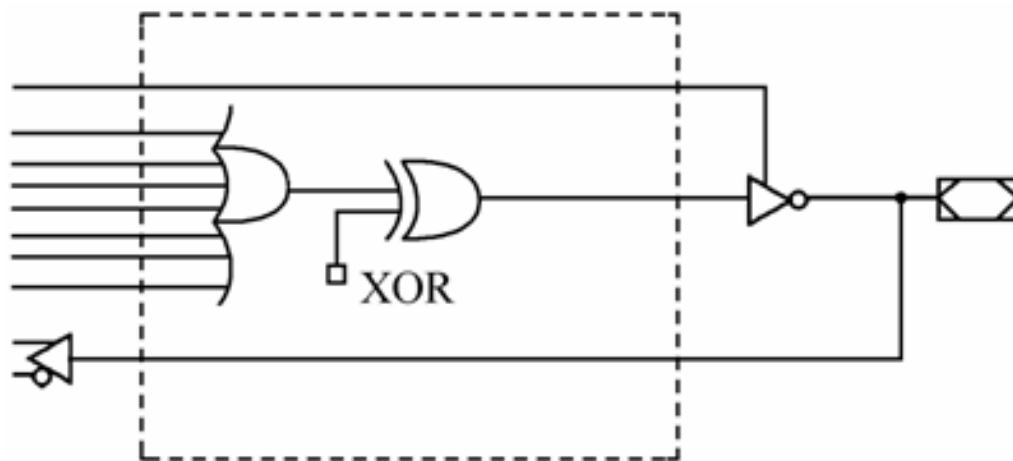


图 2-21 组合输出双向结构

2.2 简单PLD结构原理

(2) 复合模式

2、组合输出结构

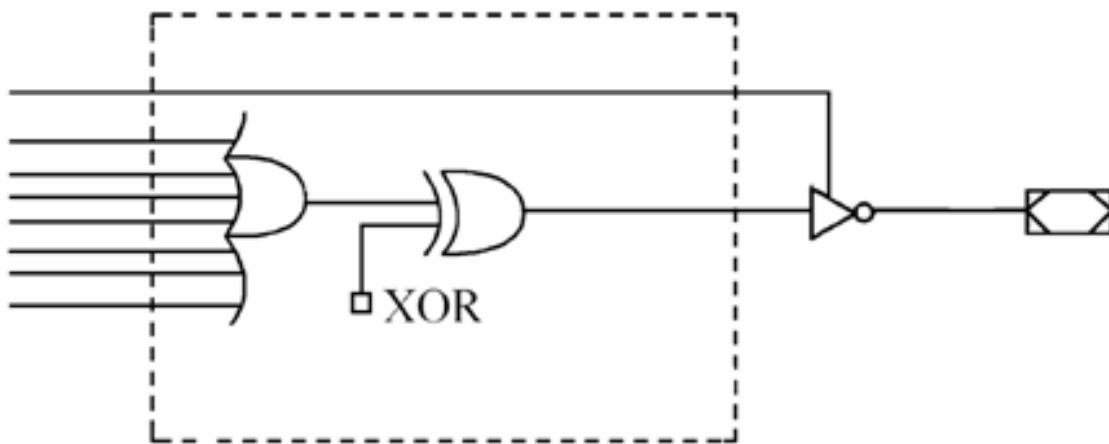


图 2-22 复合型组合输出结构

2.2 简单PLD结构原理

(3) 简单模式

1、反馈输入结构

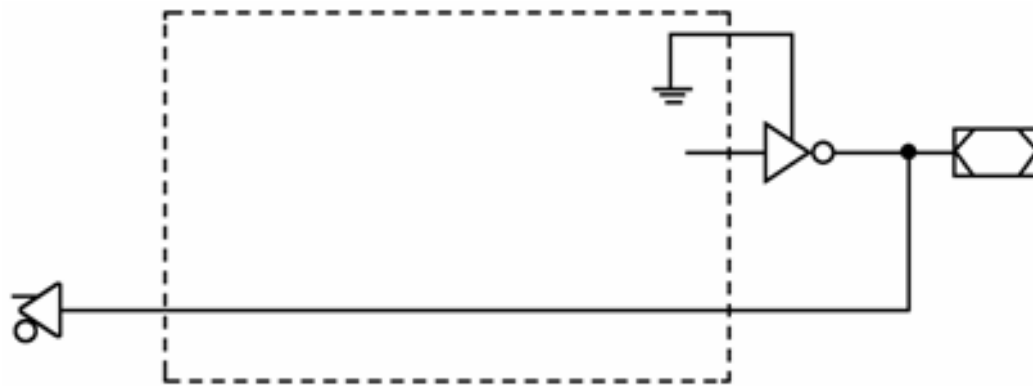


图 2-23 反馈输入结构

2.2 简单PLD结构原理

(3) 简单模式

2、输出反馈结构

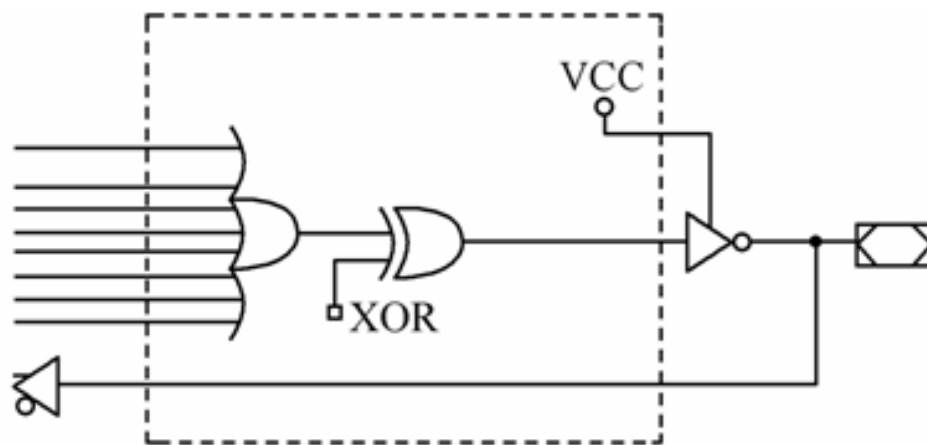


图 2-24 输出反馈结构

2.2 简单PLD结构原理

(3) 简单模式

3、输出结构

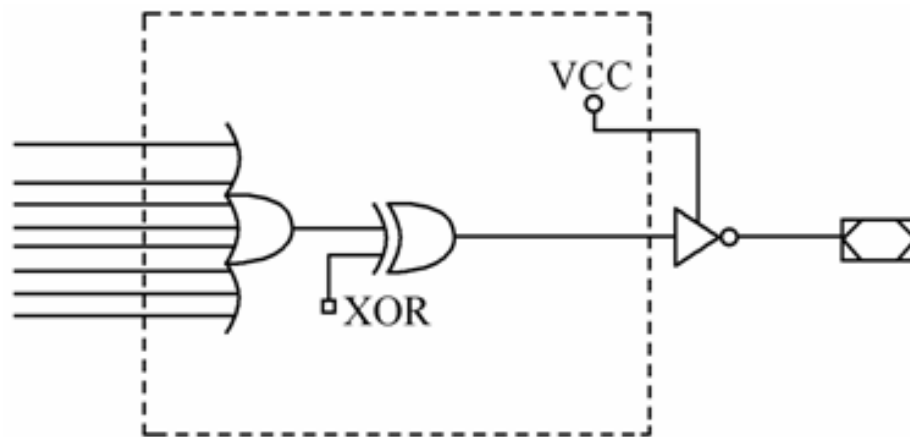


图 2-25 简单模式输出结构

2.3 CPLD的结构及其工作原理

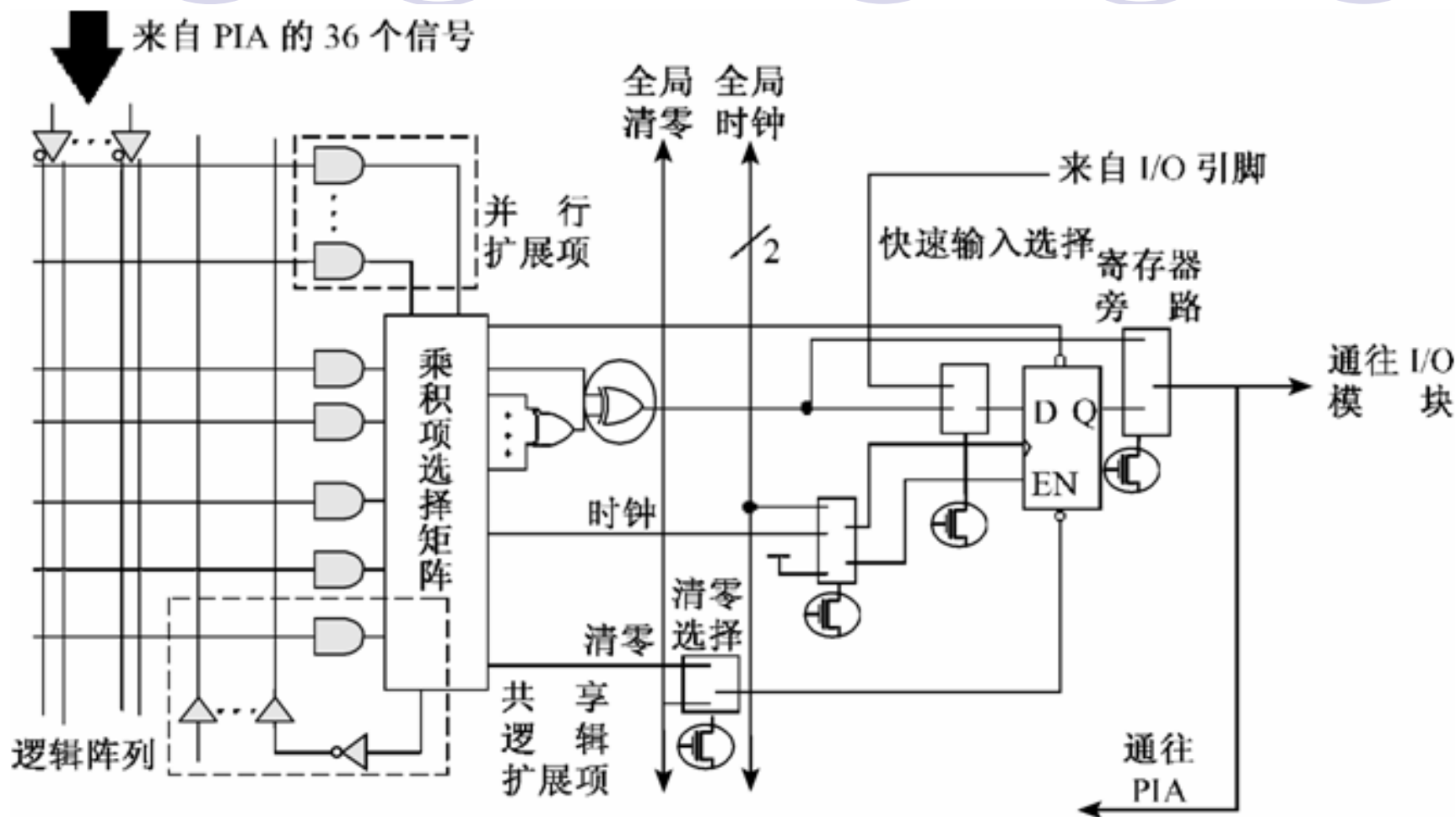


图 2-26 MAX3000A 系列的单个宏单元结构

2.3 CPLD的结构及其工作原理

1. 逻辑阵列块LAB

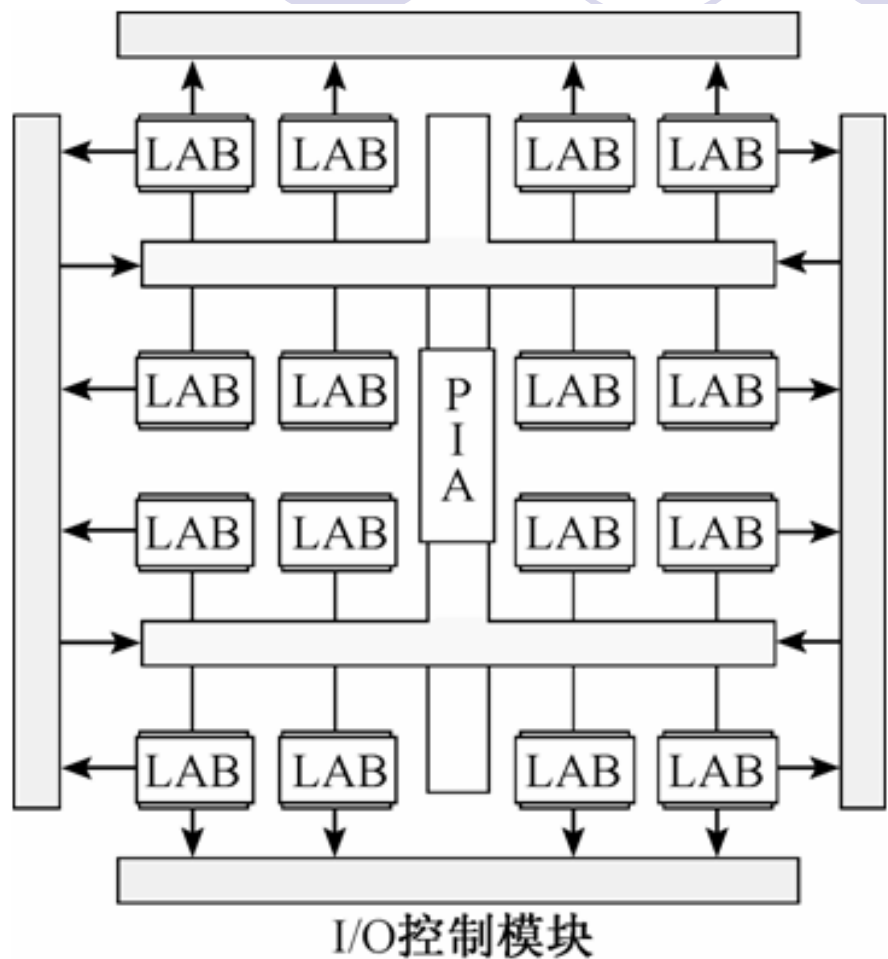


图 2-27 MAX3000 的结构

2.3 CPLD的结构及其工作原理

2. 宏单元

三种时钟输入模式

- (1) 全局时钟信号。
- (2) 全局时钟信号由高电平有效的时钟信号使能。
- (3) 用乘积项实现一个阵列时钟。

2.3 CPLD的结构及其工作原理

3. 扩展乘积项

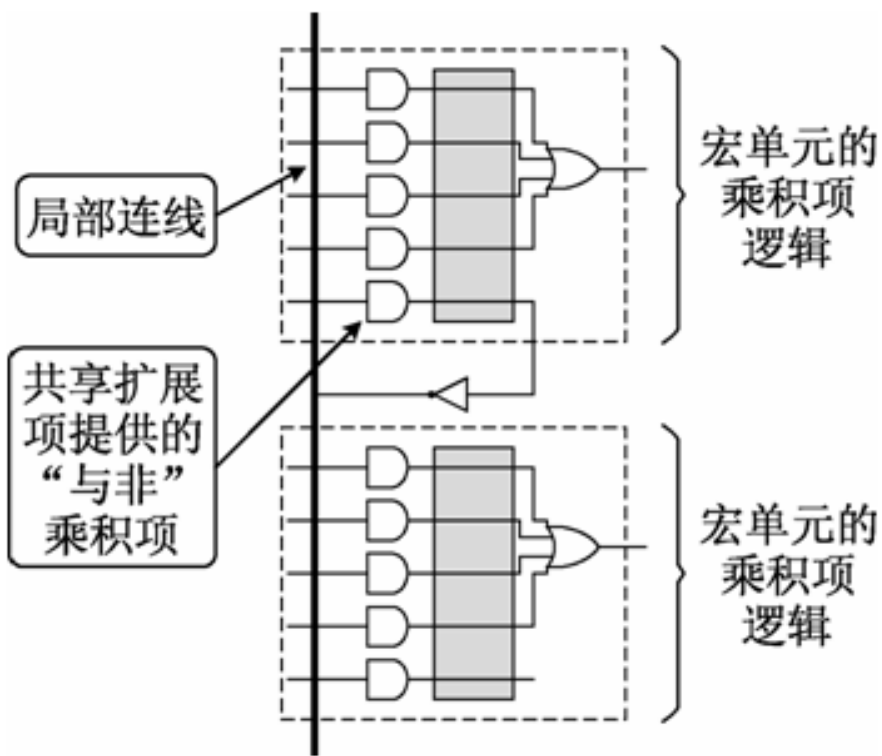


图 2-28 共享扩展乘积项结构

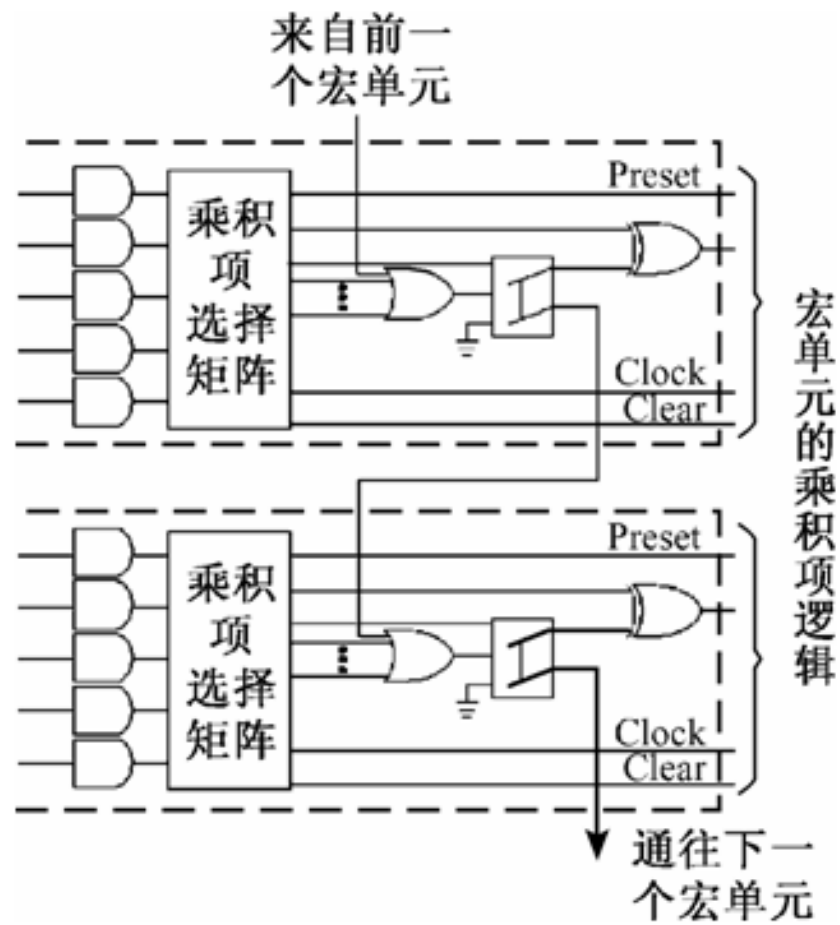


图 2-29 并联扩展项馈送方式

2.3 CPLD的结构及其工作原理

4. 可编程连线阵列PIA

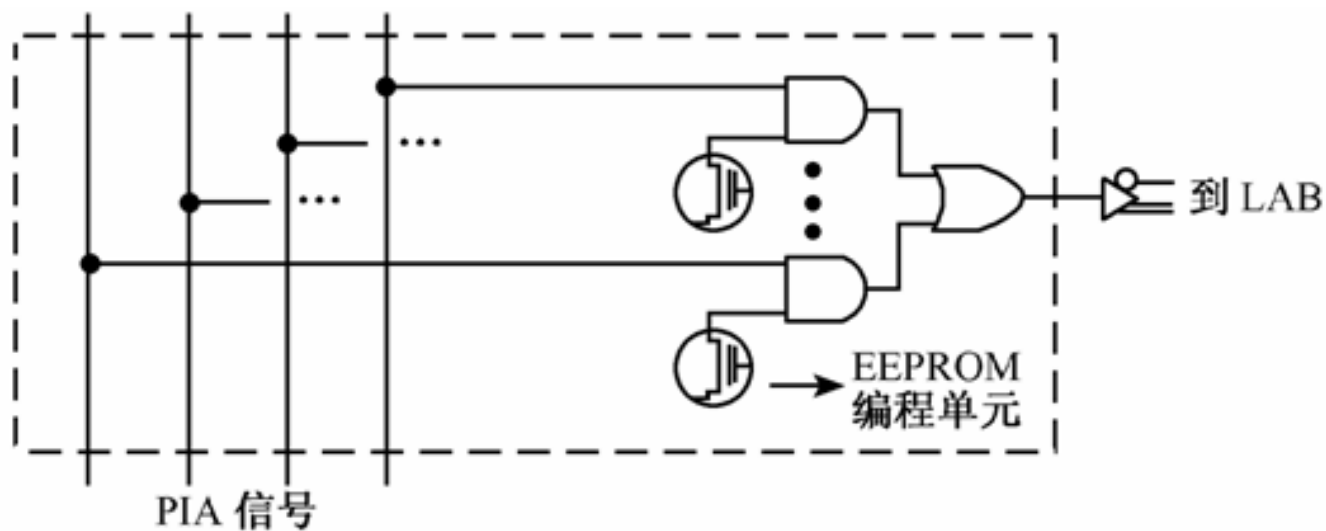


图 2-30 PIA 信号布线到 LAB 的方式

2.3 CPLD的结构及其工作原理

5. I/O控制块

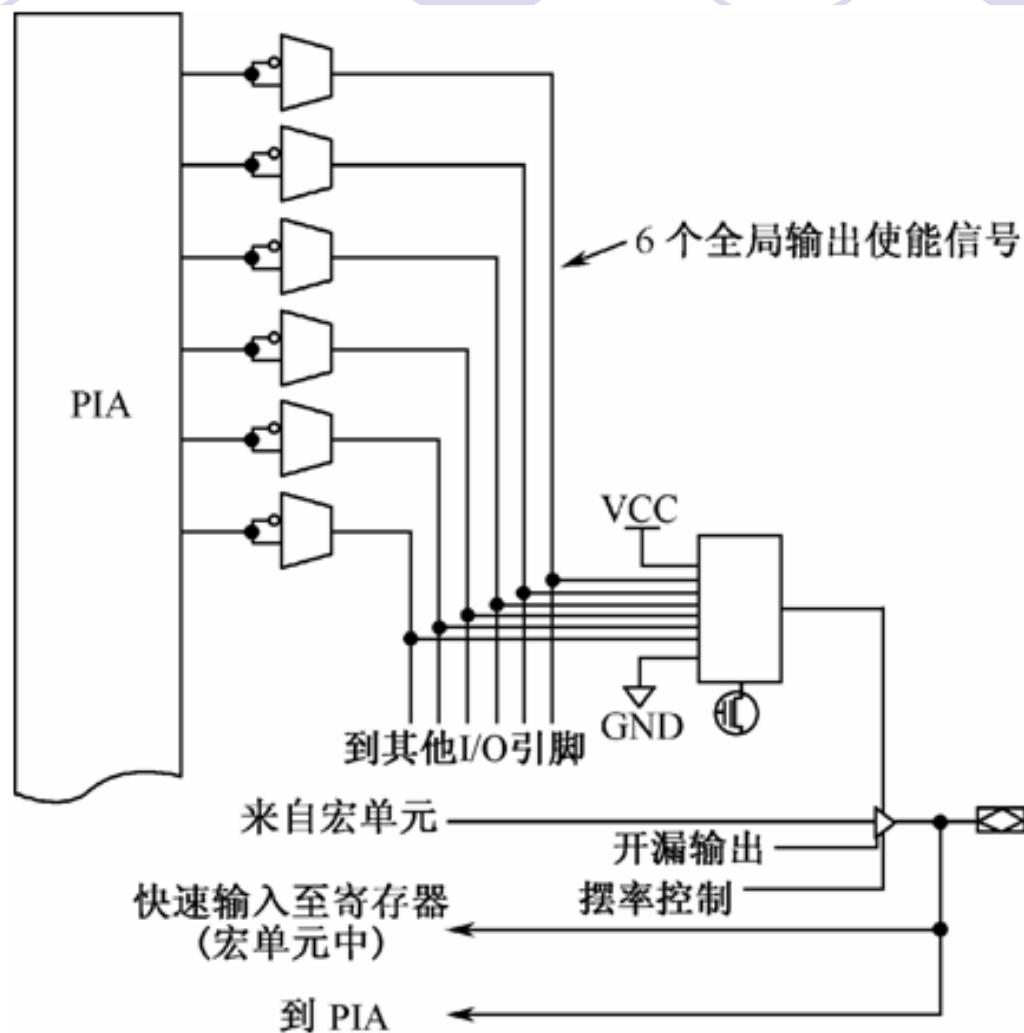


图 2-31 MAX3000A 系列器件的 I/O 控制块

2.4 FPGA的结构及其工作原理

2.4.1 查找表逻辑结构

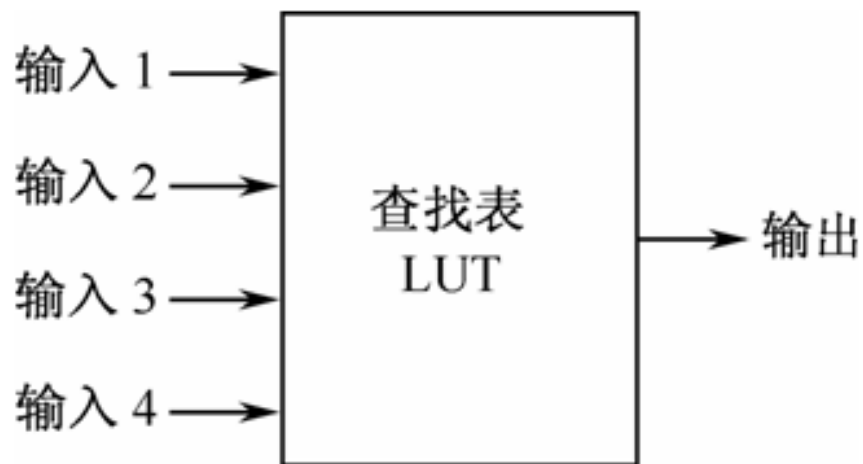


图 2-32 FPGA 查找表单元

2.4 FPGA的结构及其工作原理

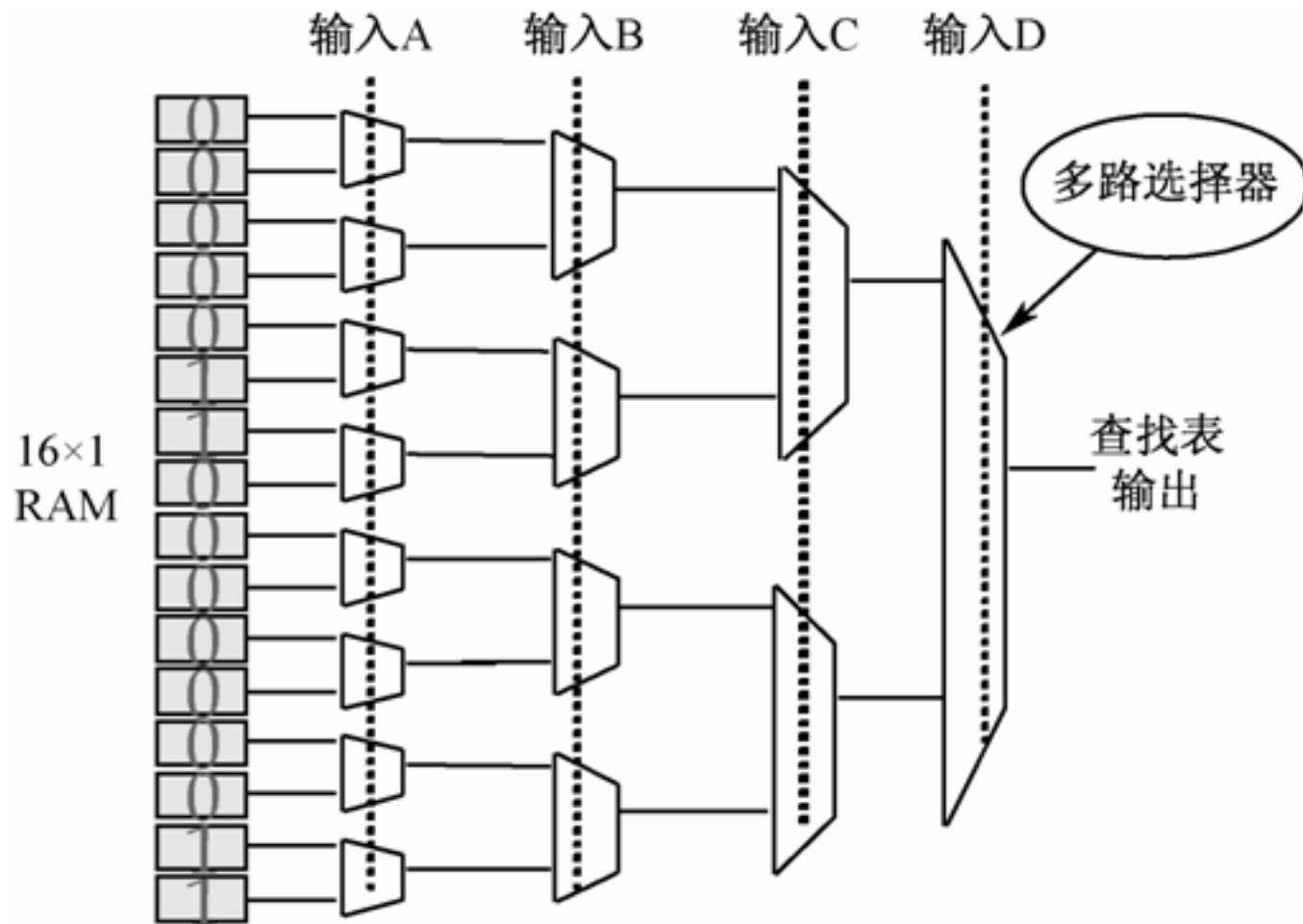
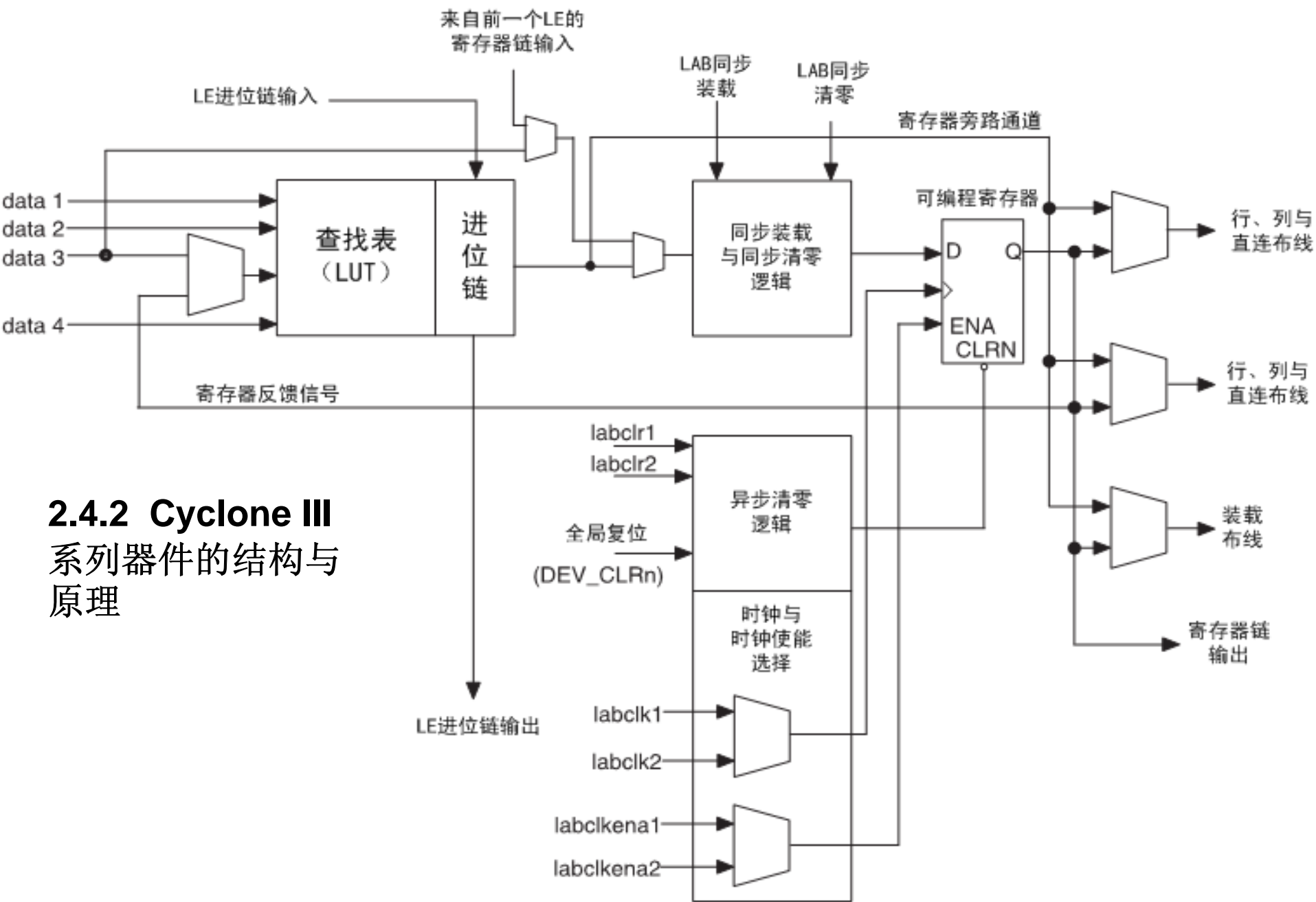


图 2-33 FPGA 查找表单元内部结构



2.4.2 Cyclone III 系列器件的结构与原理

图 2-34 Cyclone III LE 结构图

2.4.2 Cyclone III系列器件的结构与原理

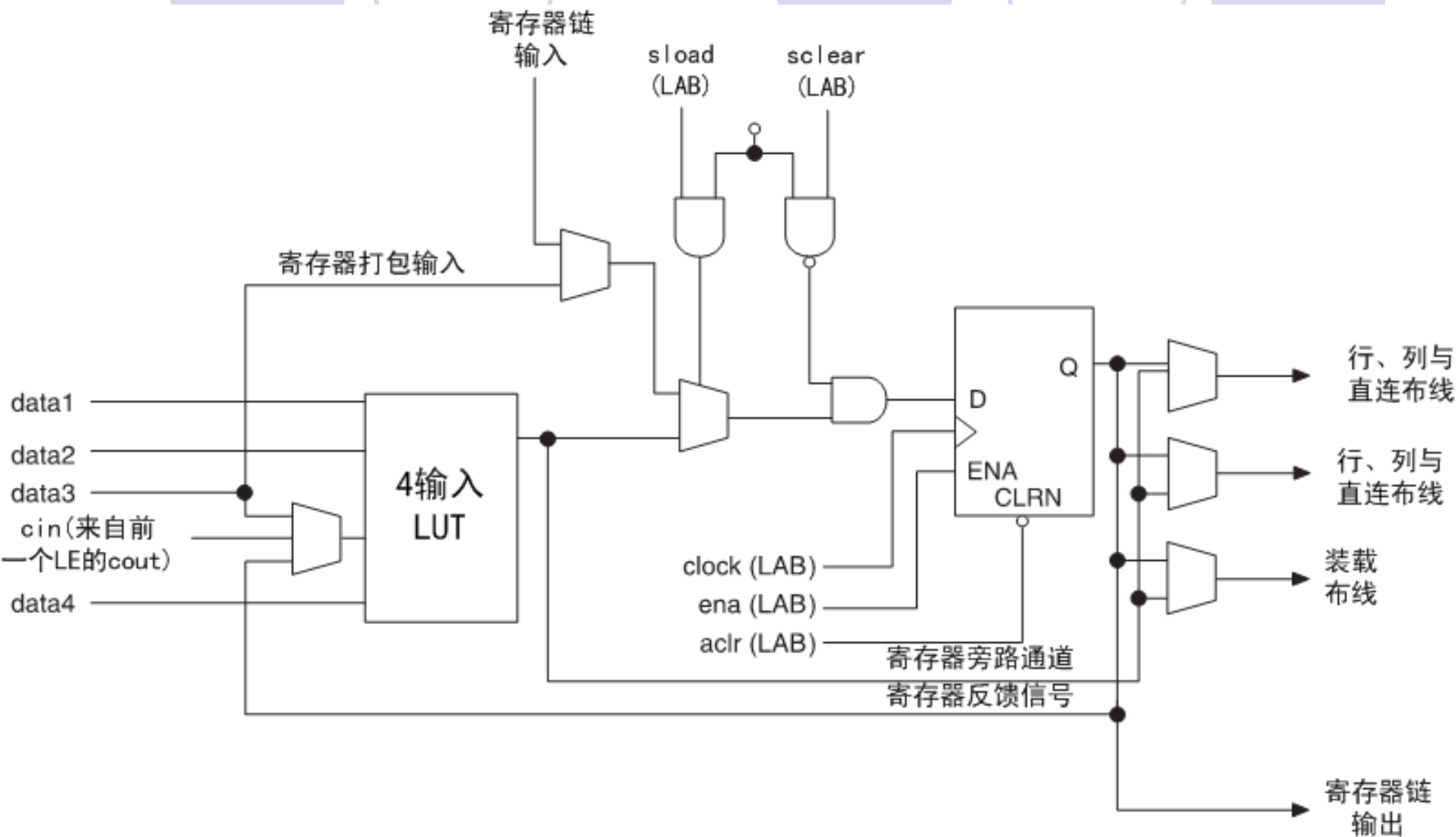


图 2-35 Cyclone III LE 普通模式

2.4.2 Cyclone III系列器件的结构与原理

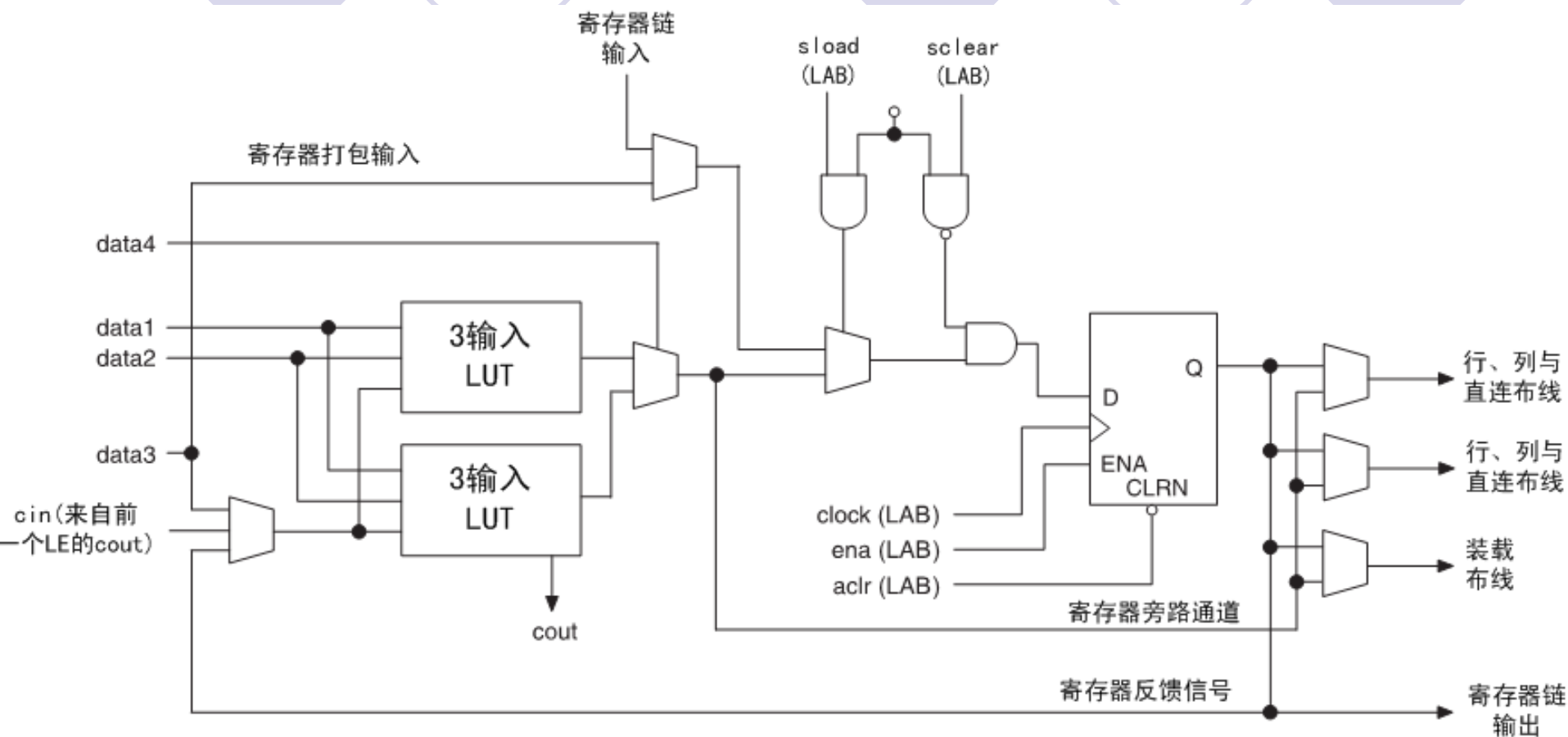


图 2-36 Cyclone III LE 动态算术模式

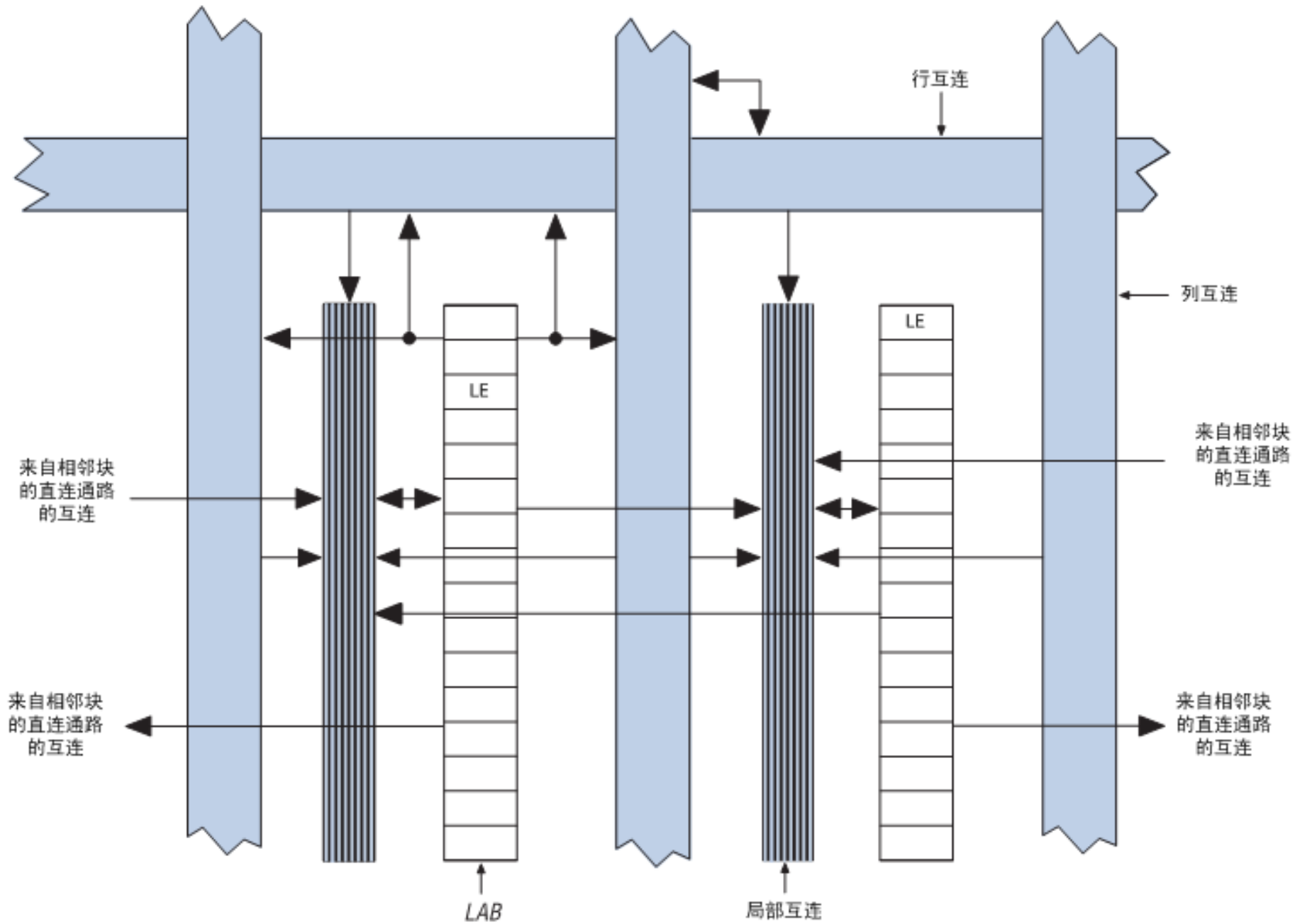


图 2-37 Cyclone III LAB 结构

2.4.2 Cyclone III系列器件的结构与原理

来自于左侧LAB、M9K存储器块、嵌入式乘法器、PLL、IOE输出的直连通路互连

来自于右侧LAB、M9K存储器块、嵌入式乘法器、PLL、IOE输出的直连通路互连

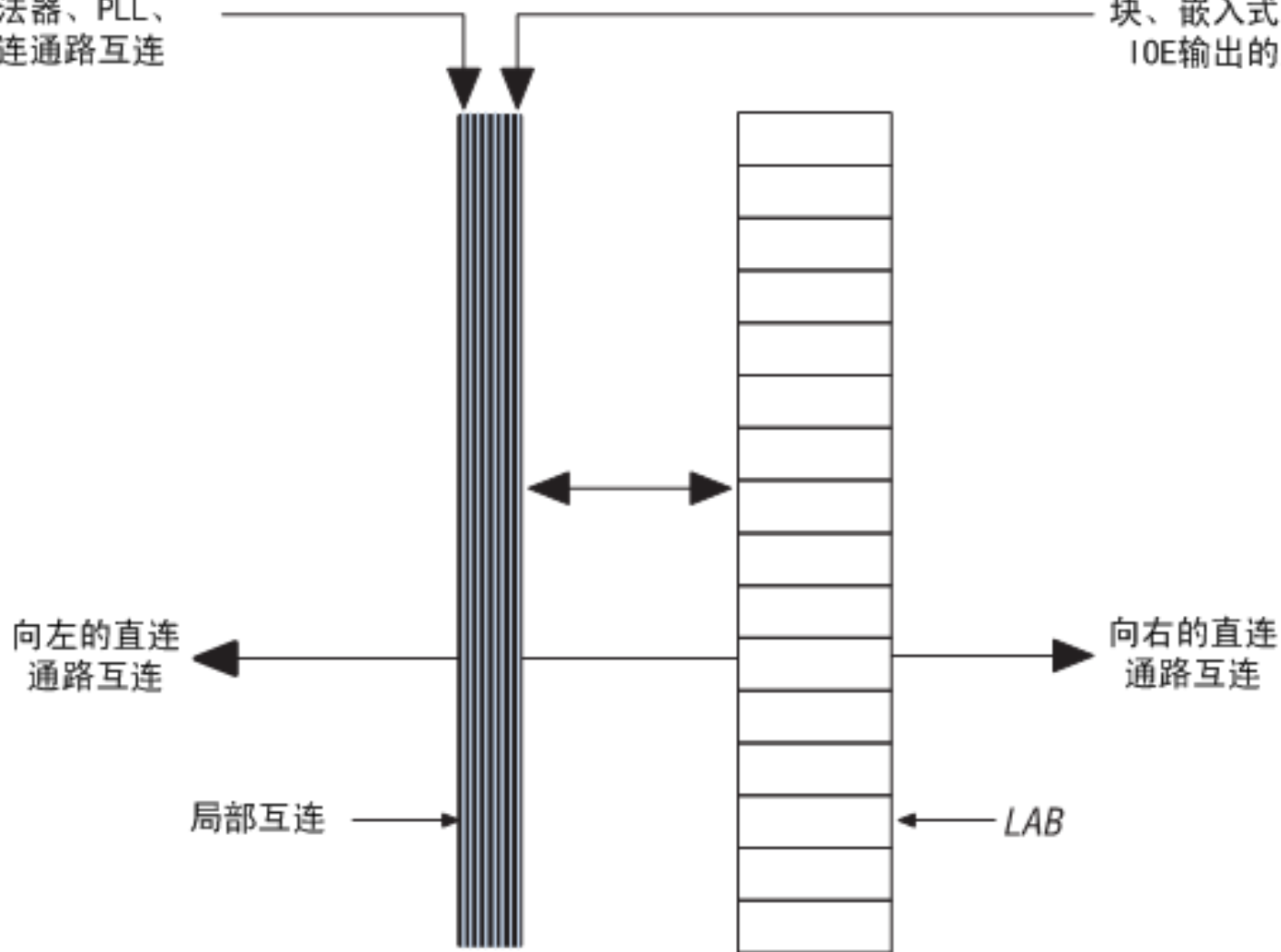


图 2-38 LAB 阵列间互连

2.4 FPGA的结构及其工作原理

2.4.2 Cyclone III系列器件的结构与原理

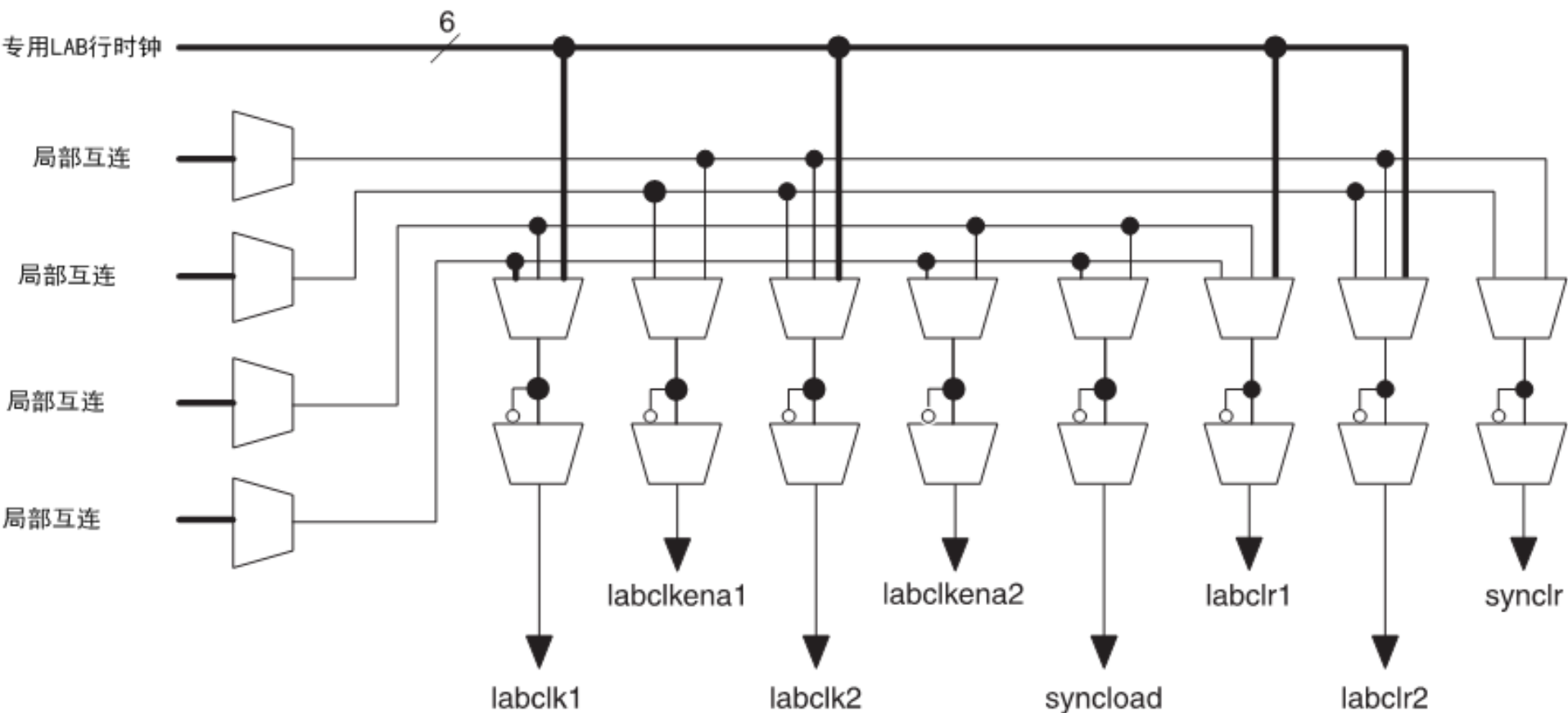


图 2-39 LAB 控制信号生成

2.4 FPGA的结构及其工作原理

2.4.2 Cyclone III系列器件的结构与原理

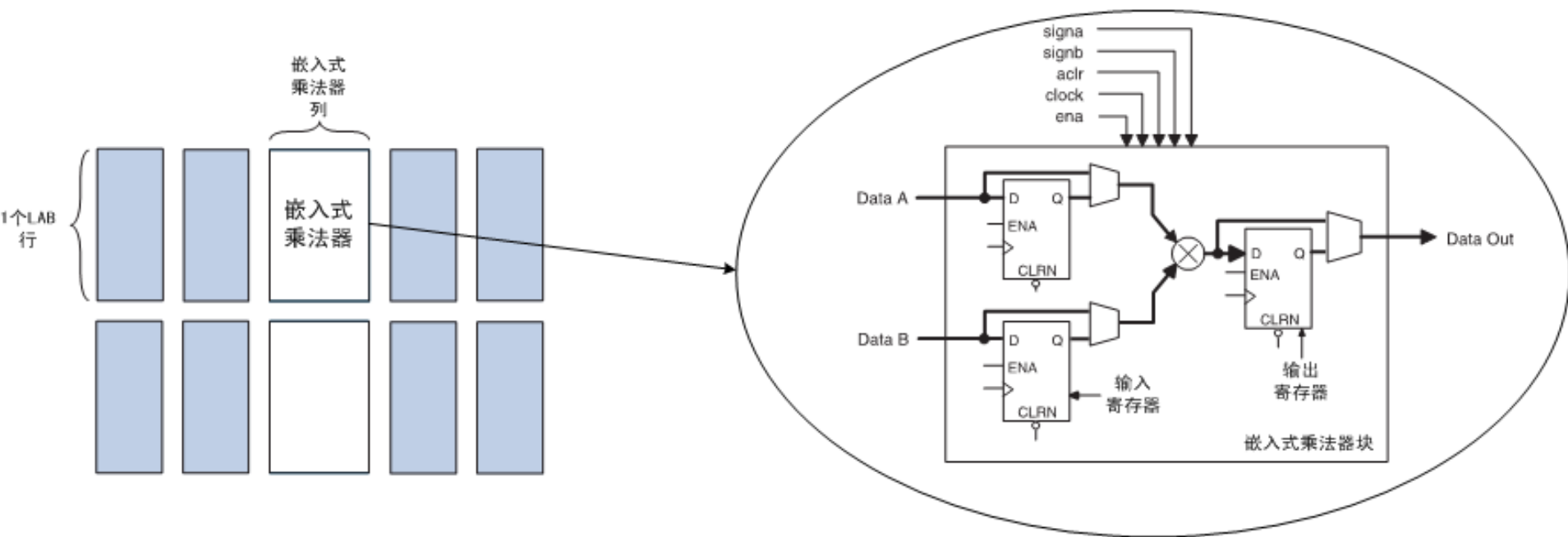


图 2-40 嵌入式乘法器

2.4 FPGA的结构及其工作原理

2.4.2 Cyclone III系列器件的结构与原理

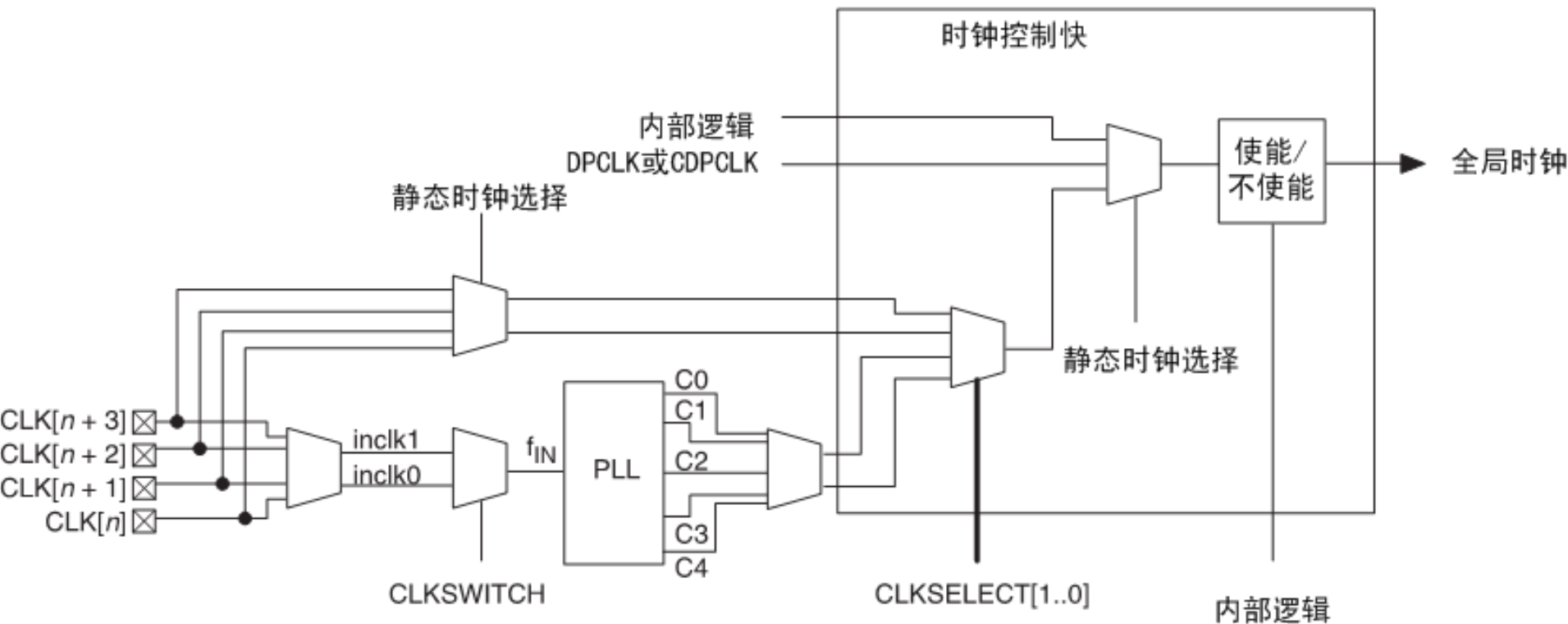


图 2-41 时钟网络的时钟控制

2.4 FPGA的结构及其工作原理

2.4.2 Cyclone III系列器件的结构与原理

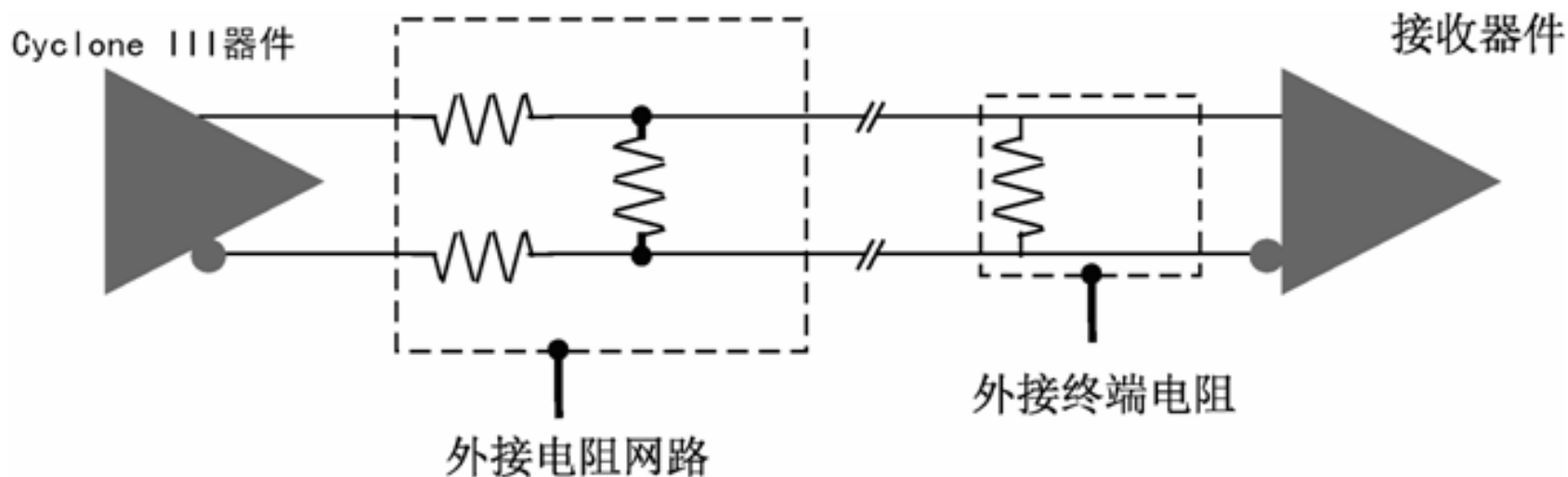


图 2-42 LVDS 连接

2.5 硬件测试

2.5.1 内部逻辑测试

2.5.2 JTAG边界扫描测试

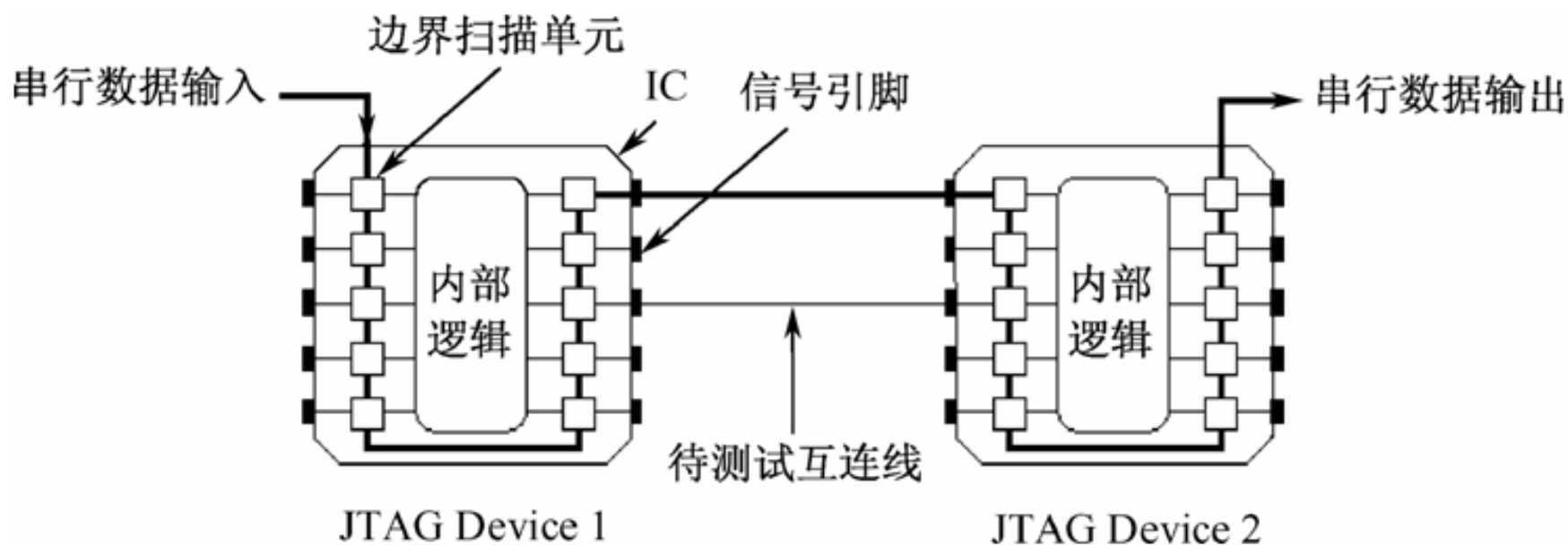


图 2-43 边界扫描电路结构

2.5 硬件测试

2.5.2 JTAG边界扫描测试

表 2-1 边界扫描 I/O 引脚功能

引 脚	描 述	功 能
TDI	测试数据输入	测试指令和编程数据的串行输入引脚。数据在 TCK 的上升沿移入
TDO	测试数据输出	测试指令和编程数据的串行输出引脚，数据在 TCK 的下降沿移出。如果数据没有被移出时，该引脚处于高阻态
TMS	测试模式选择	控制信号输入引脚，负责 TAP 控制器的转换。TMS 必须在 TCK 的上升沿到来之前稳定
TCK	测试时钟输入	时钟输入到 BST 电路，一些操作发生在上升沿，而另一些发生在下降沿
TRST	测试复位输入	低电平有效，异步复位边界扫描电路（在 IEEE 规范中，该引脚可选）

2.5 硬件测试

2.5.2 JTAG边界扫描测试

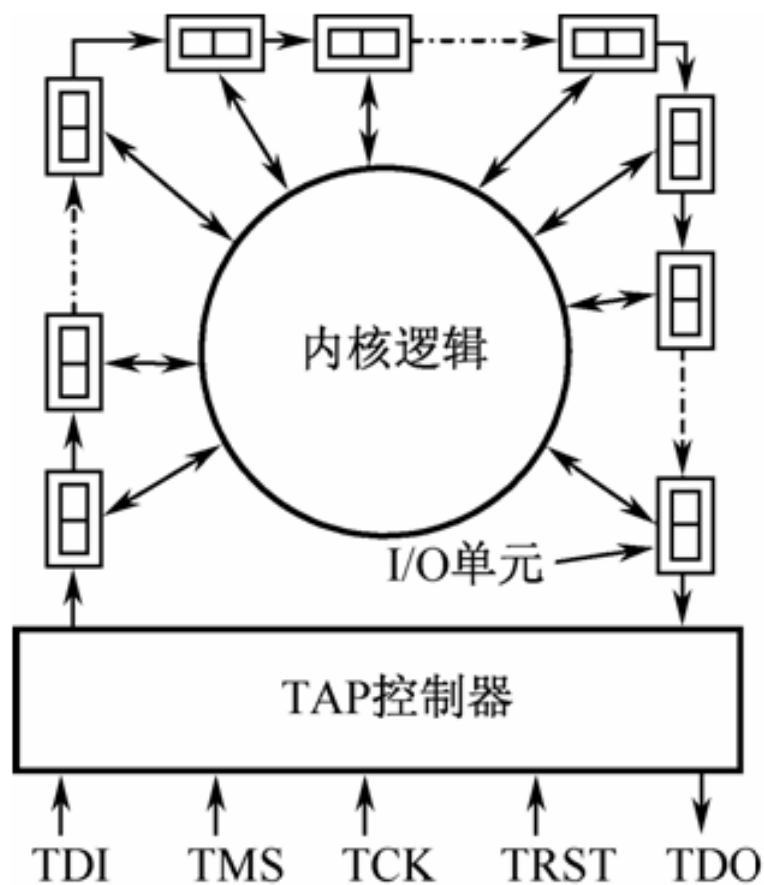


图 2-44 边界扫描数据移位方式

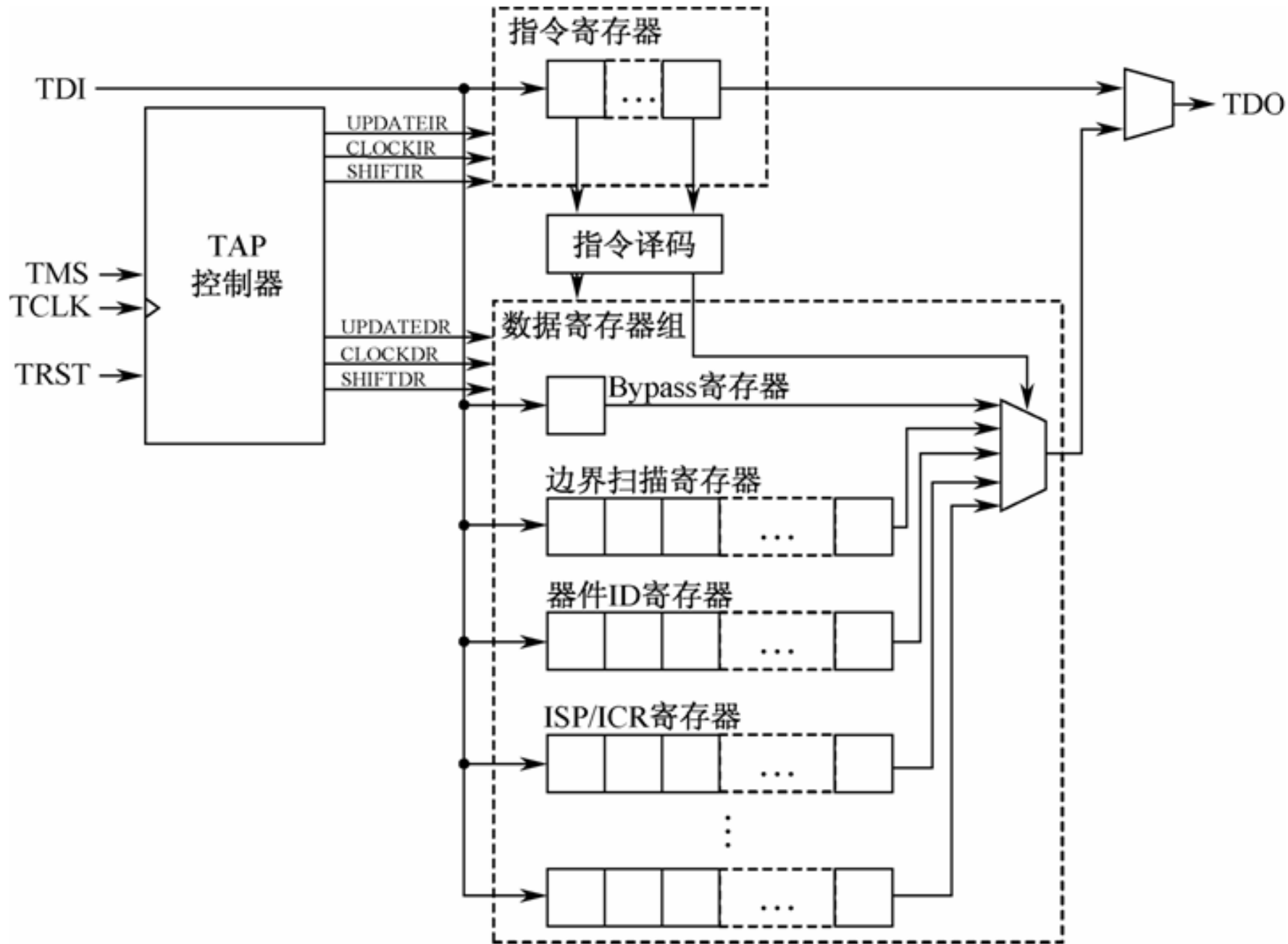


图 2-45 JTAG BST 系统内部结构

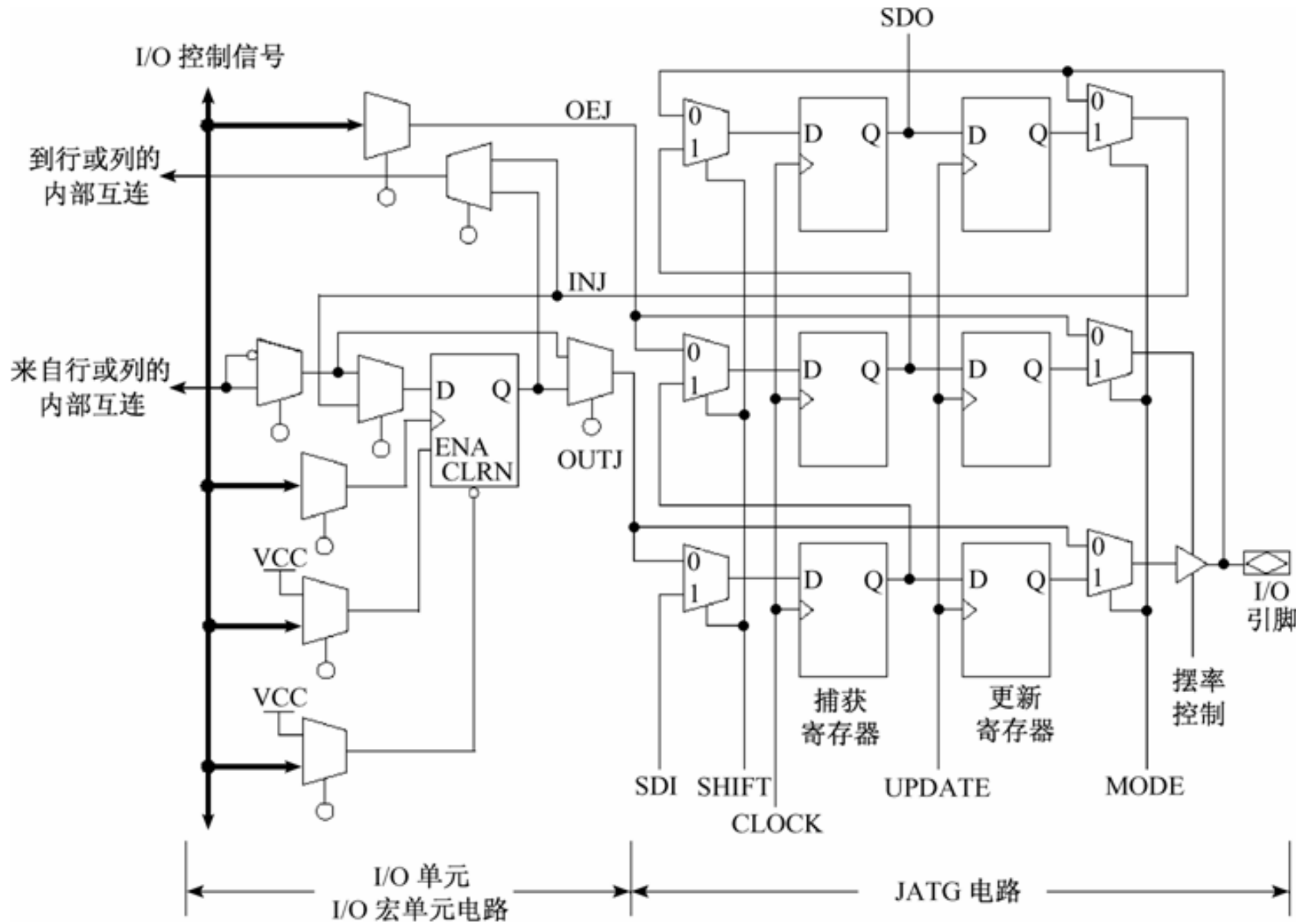


图 2-46 JTAG BST 系统与 FPGA 器件关联结构图

2.5 硬件测试

2.5.2 JTAG边界扫描测试

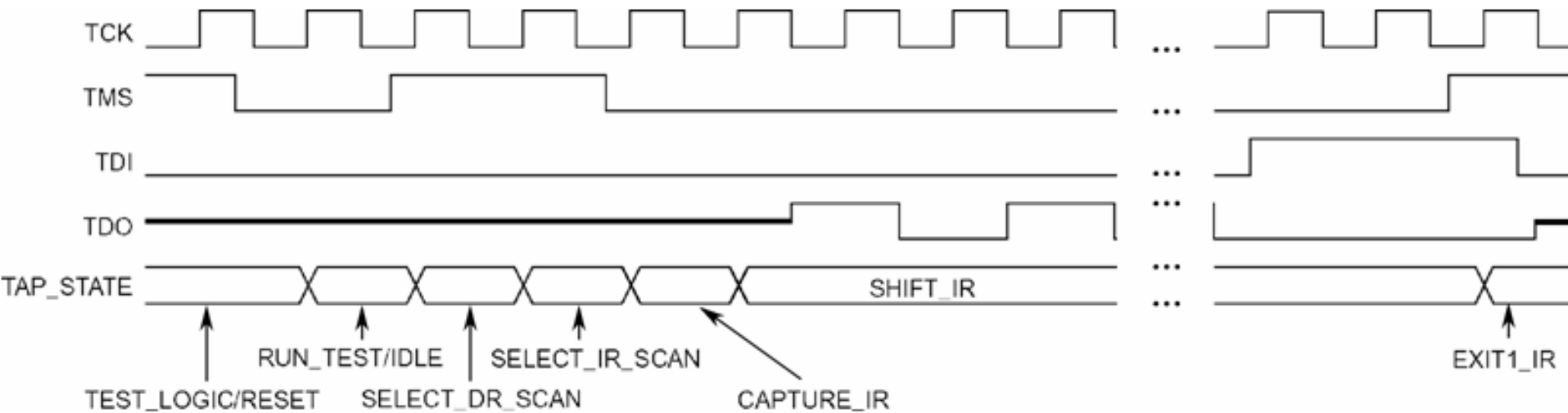
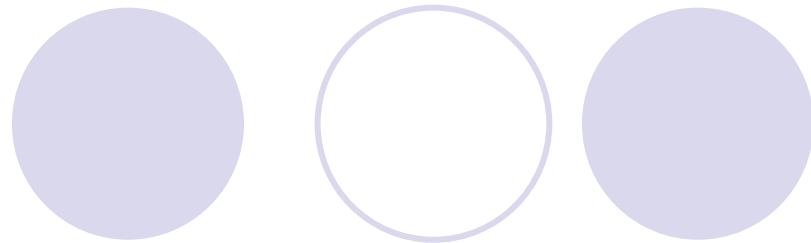


图 2-47 JTAG BST 选择命令模式时序

2.5.3 嵌入式逻辑分析仪

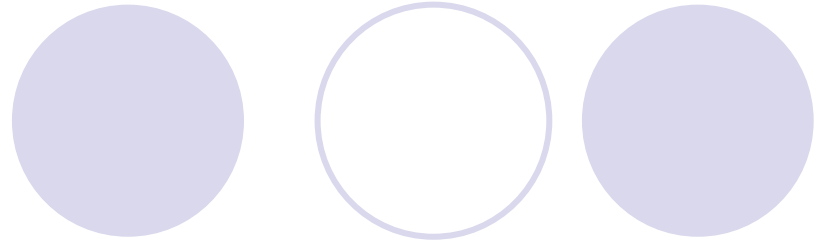
2.6 PLD产品概述



2.6.1 Lattice公司的PLD器件

1. ispLSI系列器件
2. MACHXO系列
3. MACH4000系列
4. LatticeSC FPGA系列
5. LatticeECP3 FPGA系列

2.6 PLD产品概述



2.6.2 Xilinx公司的PLD器件

1. Virtex-6系列FPGA
2. Spartan-6器件系列
3. XC9500/XC9500XL系列CPLD
4. Xilinx Spartan-3A系列器件
5. Xilinx的IP核

2.6 PLD产品概述



2.6.3 Altera公司的PLD器件

1. **Stratix 4/6 系列FPGA**
2. **Cyclone 4系列FPGA**
3. **Cyclone系列FPGA（低成本FPGA）**
4. **Cyclone II系列FPGA**
5. **Cyclone III系列FPGA**
6. **MAX系列CPLD**
7. **MAX II系列器件**
8. **Altera宏功能块及IP核**

2.6 PLD产品概述

2.6.4 Actel公司的PLD器件

2.6.5 Altera的FPGA配置方式

表 2-2 Altera FPGA 常用配置器件

器 件	功能描述	封装形式
EPCS1	1M×1 位, 3.3/5V 供电	8 脚
EPCS4	4M×1 位, 3.3/5V 供电	8 脚
EPC2	1695680×1 位, 3.3/5V 供电	20 脚 PLCC、32 脚 TQFP

2.7 CPLD/FPGA的编程与配置

- (1) 基于电可擦除存储单元的**EEPROM**或**Flash**技术。
- (2) 基于**SRAM**查找表的编程单元。
- (3) 基于一次性可编程反熔丝编程单元。

表 2-3 各引脚信号名称

引 脚	1	2	3	4	5	6	7	8	9	10
JTAG 模式	TCK	GND	TDO	VCC	TMS	-	-	-	TDI	GND
PS 模式	DCK	GND	CONF_DONE	VCC	nCONFIG	-	nSTATUS	-	DATA0	GND

2.7 CPLD/FPGA的编程与配置

2.7.1 CPLD在系统编程

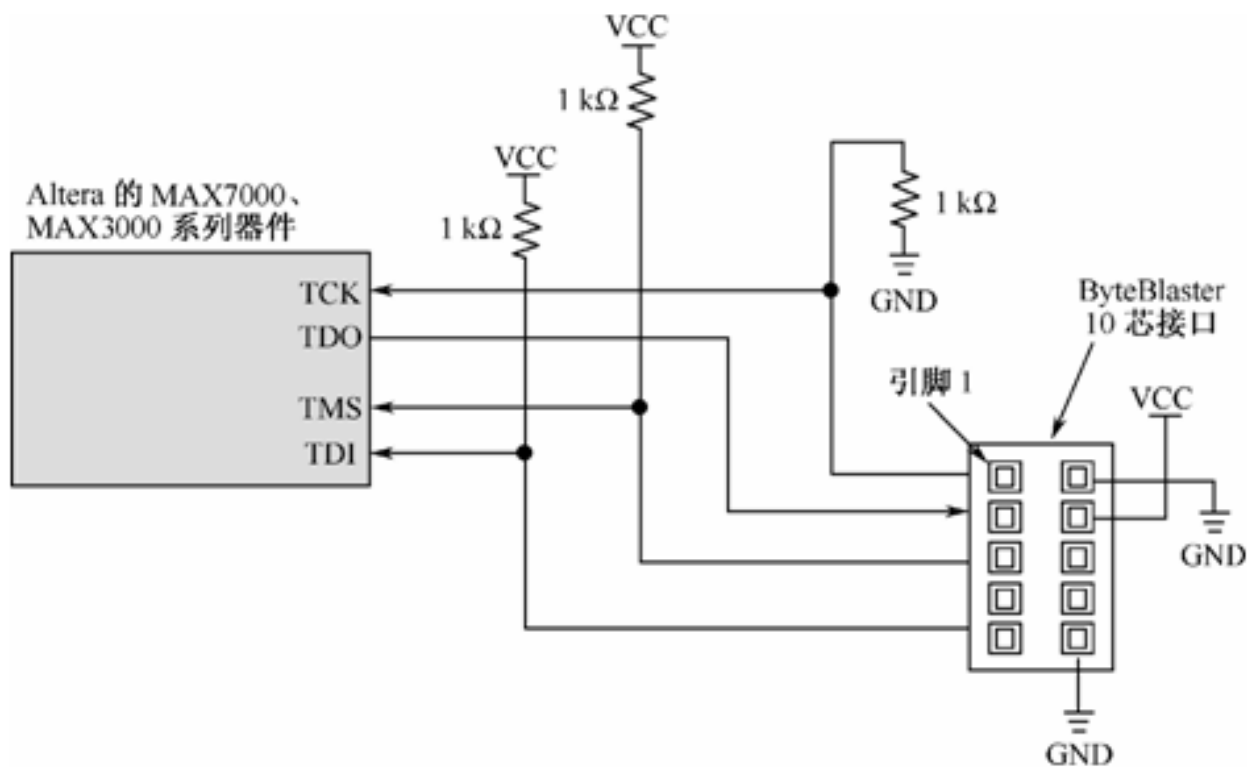


图 2-48 CPLD 编程下载连接图

2.7 CPLD/FPGA的编程与配置

2.7.1 CPLD在系统编程

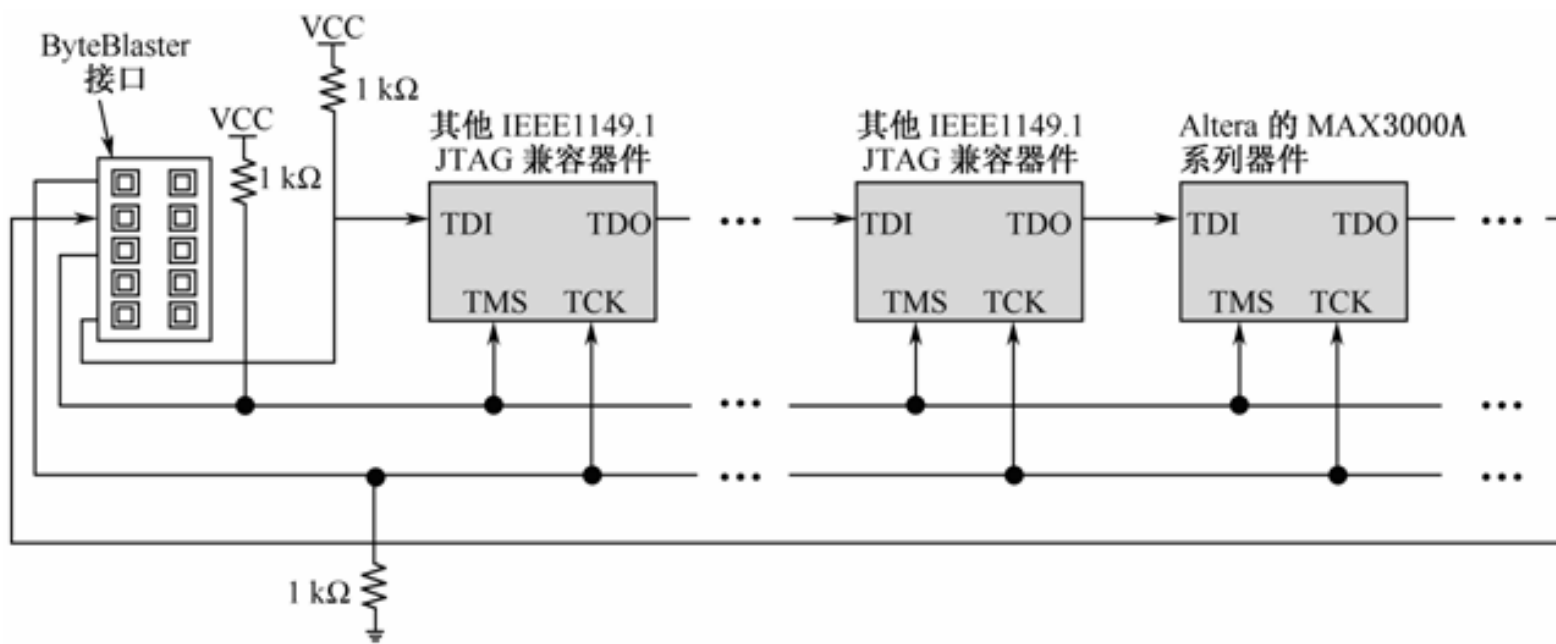


图 2-49 多 CPLD 芯片 ISP 编程连接方式

2.7 CPLD/FPGA的编程与配置

2.7.2 FPGA配置方式

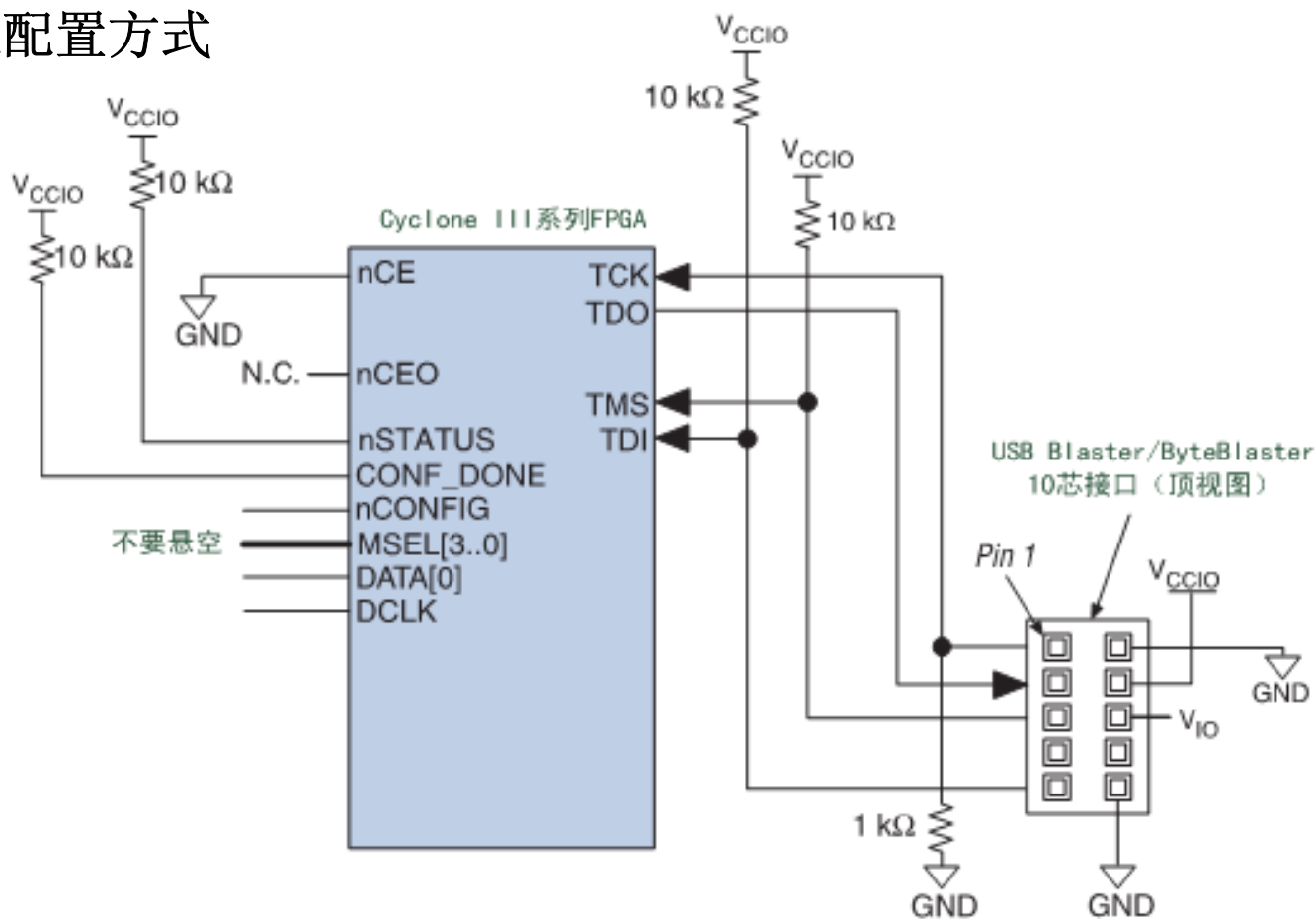


图 2-50 JTAG 在线配置 FPGA 的电路原理图

2.7 CPLD/FPGA的编程与配置

2.7.3 FPGA专用配置器件

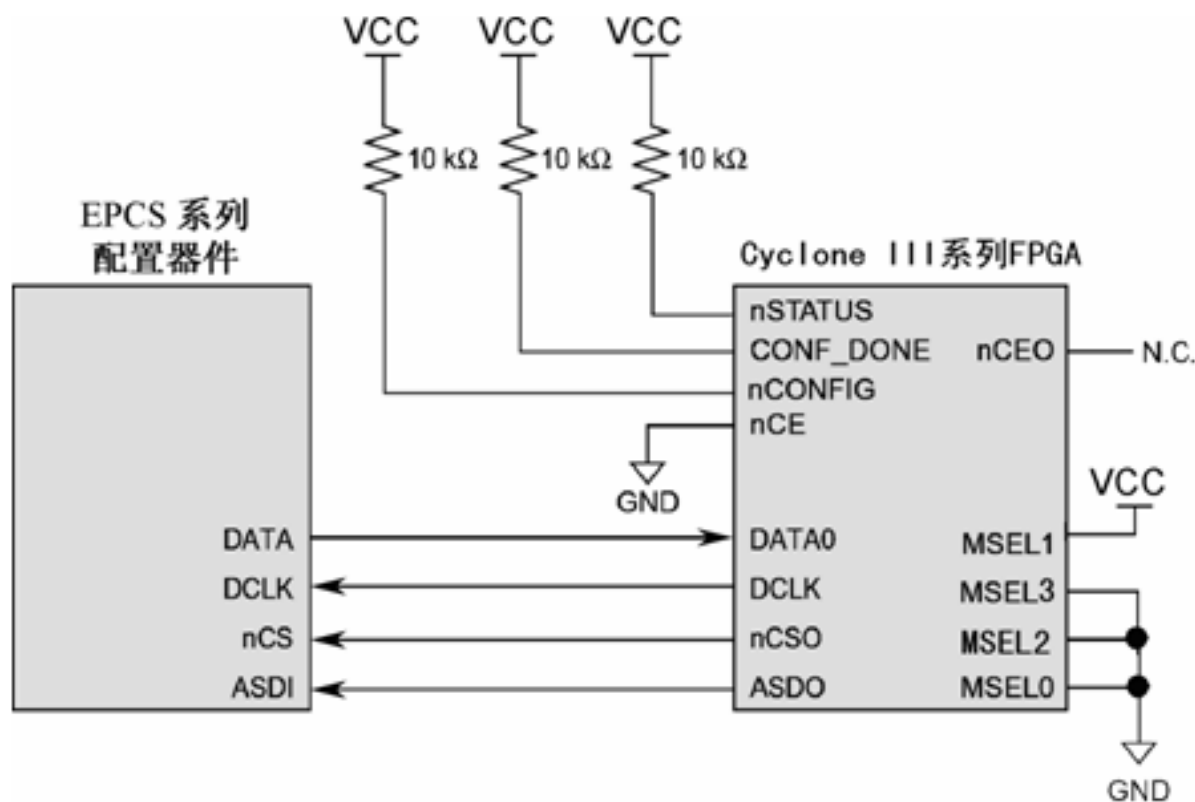


图 2-51 EPCS 器件配置 FPGA 的电路原理图

2.7 CPLD/FPGA的编程与配置

2.7.4 使用单片机配置FPGA

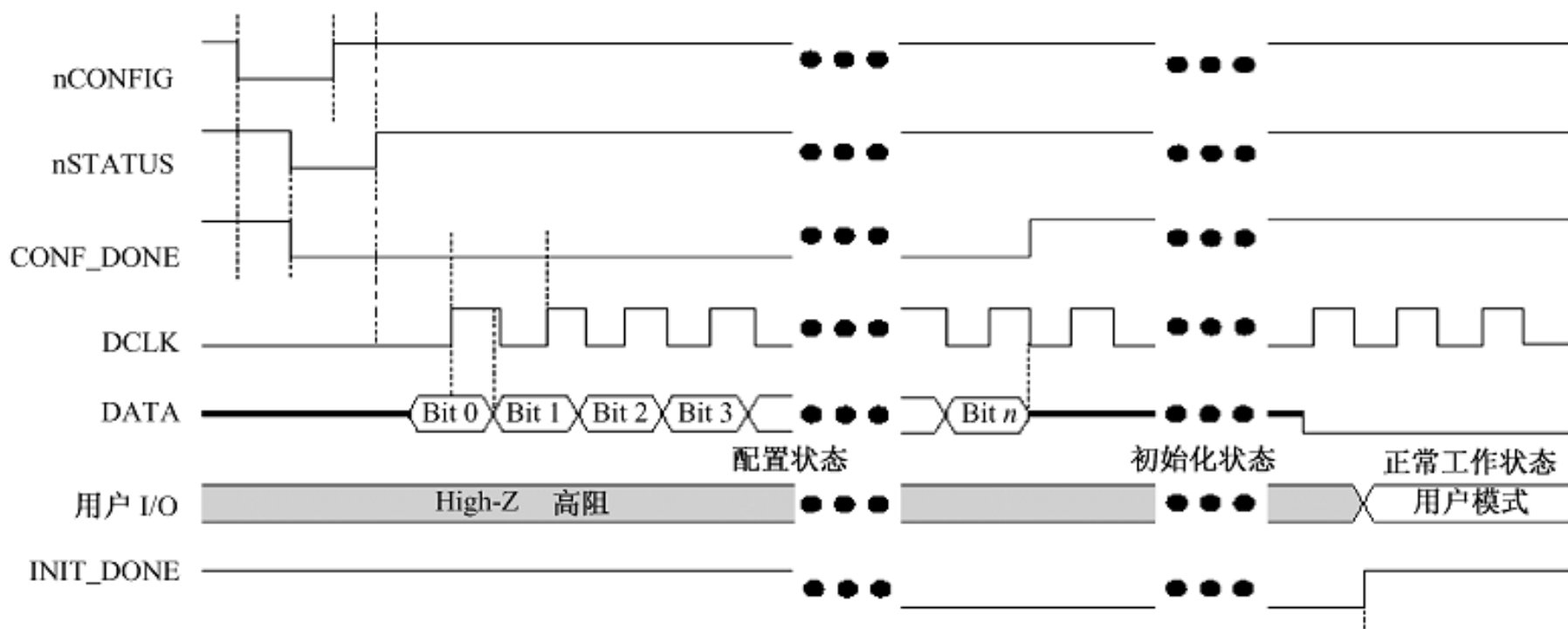


图 2-52 PS 模式的 FPGA 配置时序

2.7 CPLD/FPGA的编程与配置

2.7.4 使用单片机配置FPGA

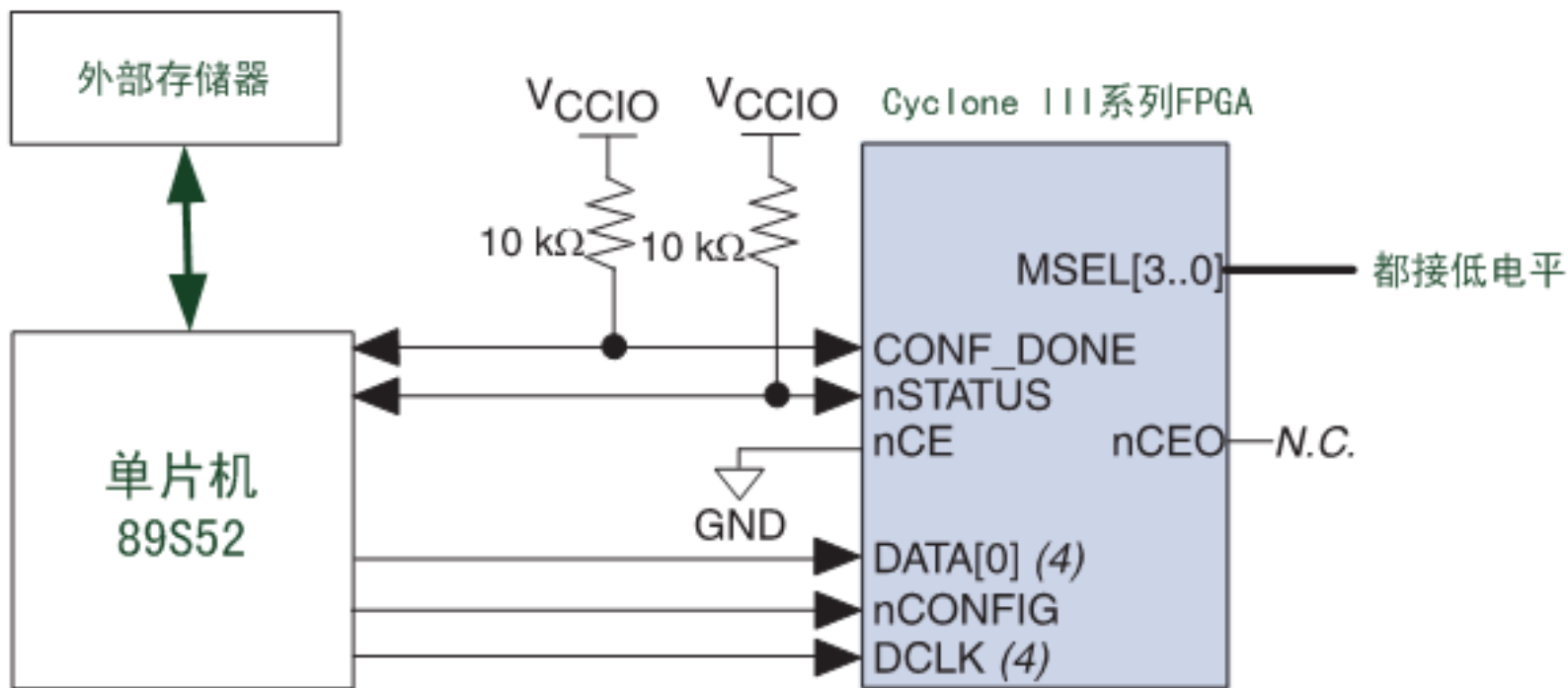


图 2-53 用 89C52 进行配置

2.7 CPLD/FPGA的编程与配置

2.7.5 使用CPLD配置FPGA

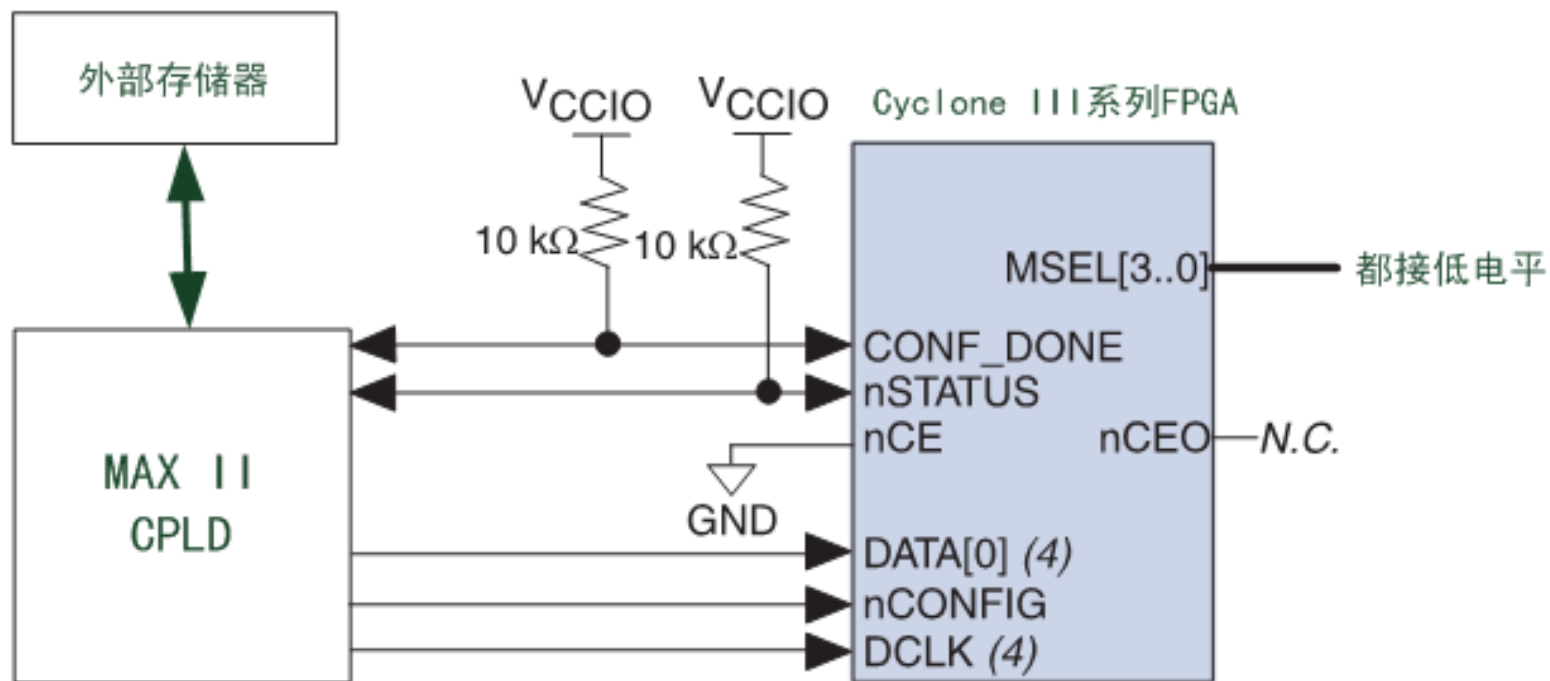


图 2-54 用 MAX II CPLD 进行配置



习 题

- 2-1** **OLMC**有何功能？说明**GAL**是怎样实现可编程组合电路与时序电路的。
- 2-2** 什么是基于乘积项的可编程逻辑结构？什么是基于查找表的可编程逻辑结构？
- 2-3** **FPGA**系列器件中的**EAB/M9K**有何作用？
- 2-4** 与传统的测试技术相比，边界扫描技术有何优点？
- 2-5** 解释编程与配置这两个概念。
- 2-6** 请参阅相关资料，并回答问题：按本章给出的归类方式，将基于乘积项的可编程逻辑结构的**PLD**器件归类为**CPLD**；将基于查找表的可编程逻辑结构的**PLD**器件归类为**FPGA**，那么，**APEX**系列属于什么类型**PLD**器件？**MAX II**系列又属于什么类型的**PLD**器件？为什么