



## 8.4 复杂的可编程逻辑器件(CPLD)

---

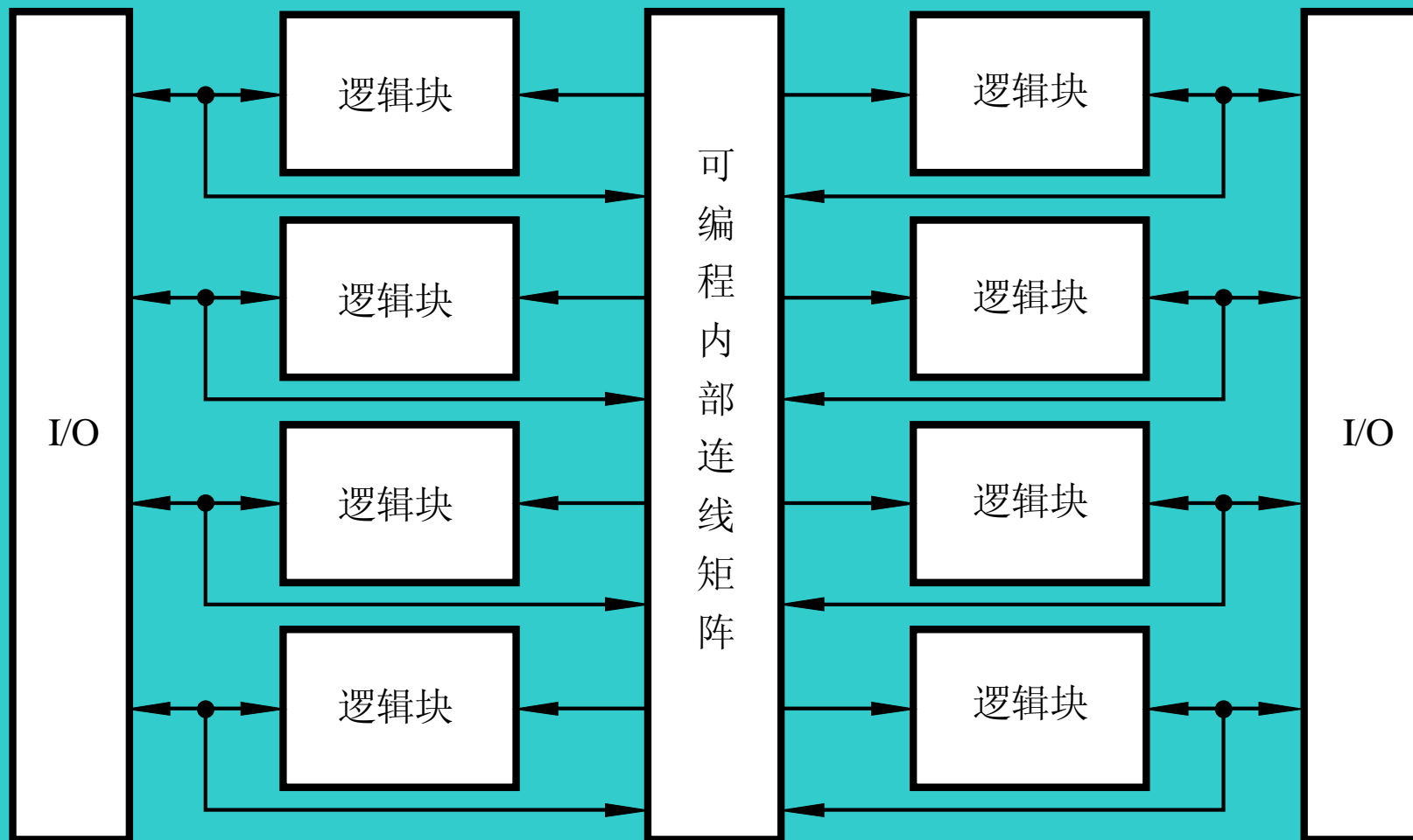
---

### \*8.4.1 MAX7000系列CPLD

### 8.4.2 ispLSI 1032E

- CPLD器件内部含有多个逻辑单元块，每个逻辑块就相当于一个GAL器件；
- 这些逻辑块可以使用可编程内部连线（可编程的开关矩阵）实现相互之间的连接；
- CPLD允许有更多的输入信号、更多的乘积项和更多的宏单元定义；

# CPLD的结构



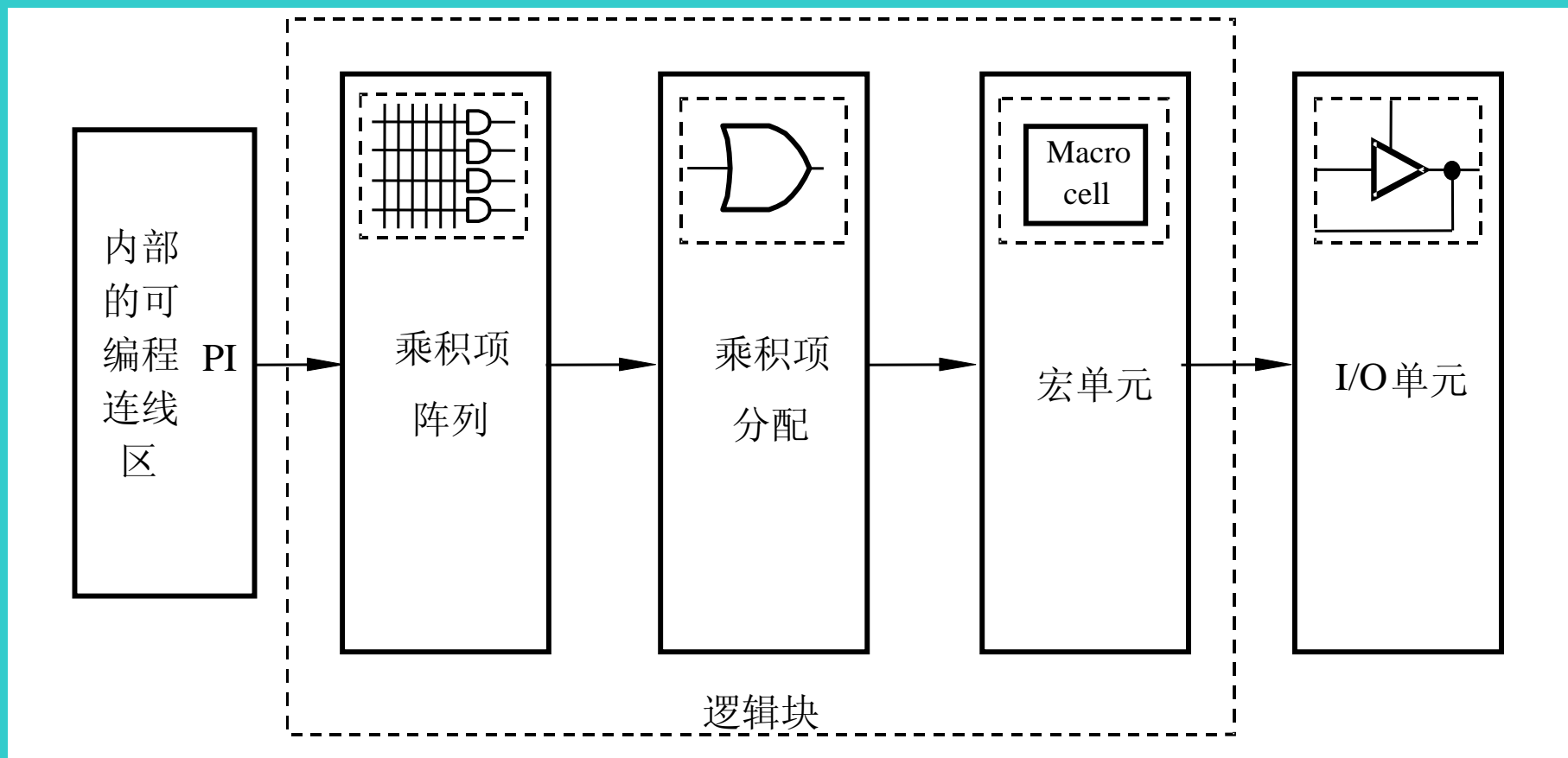
HOME

BACK

NEXT



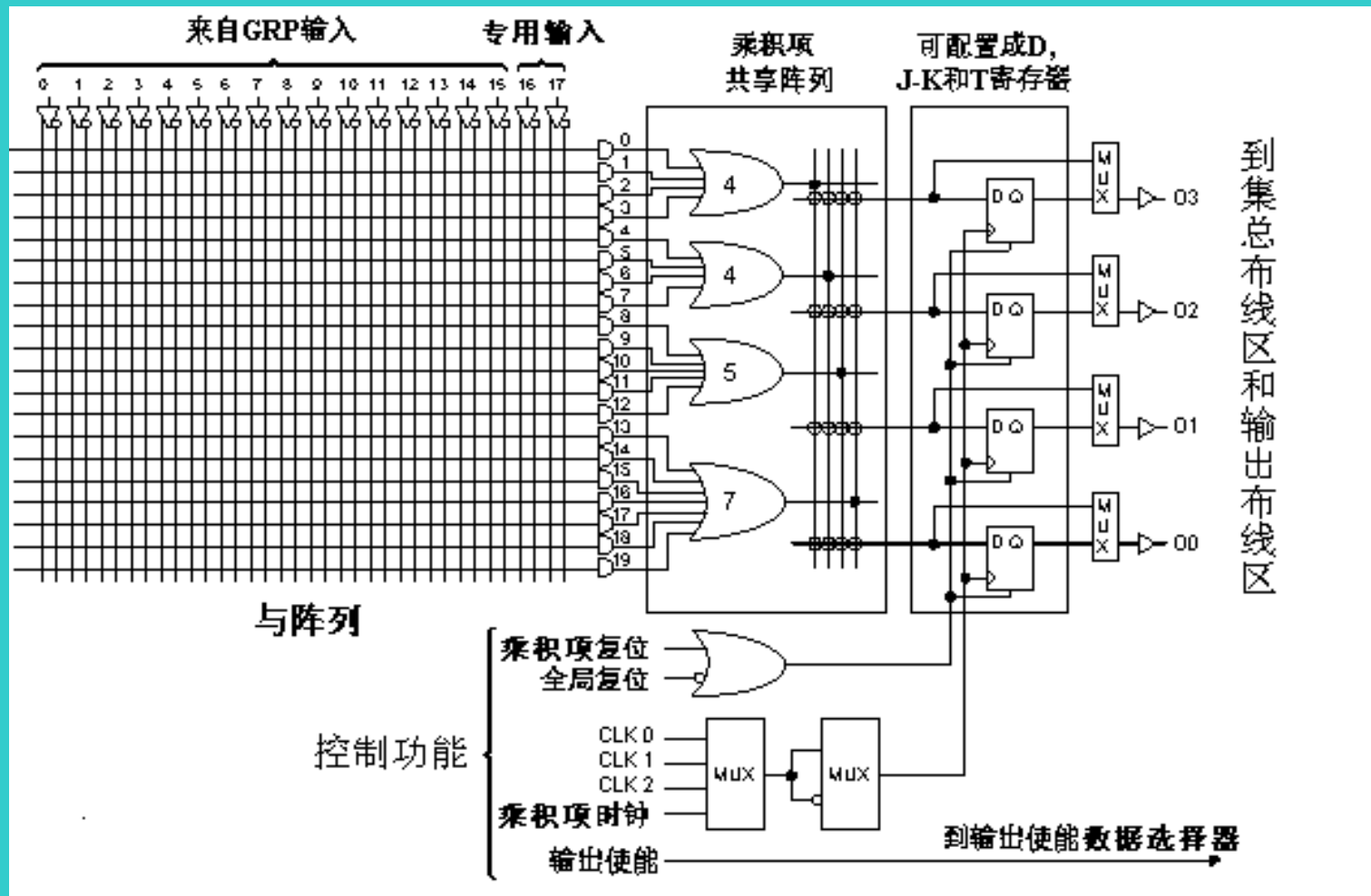
## CPLD的 逻辑块结构







## ispLSI1032器件GLB结构



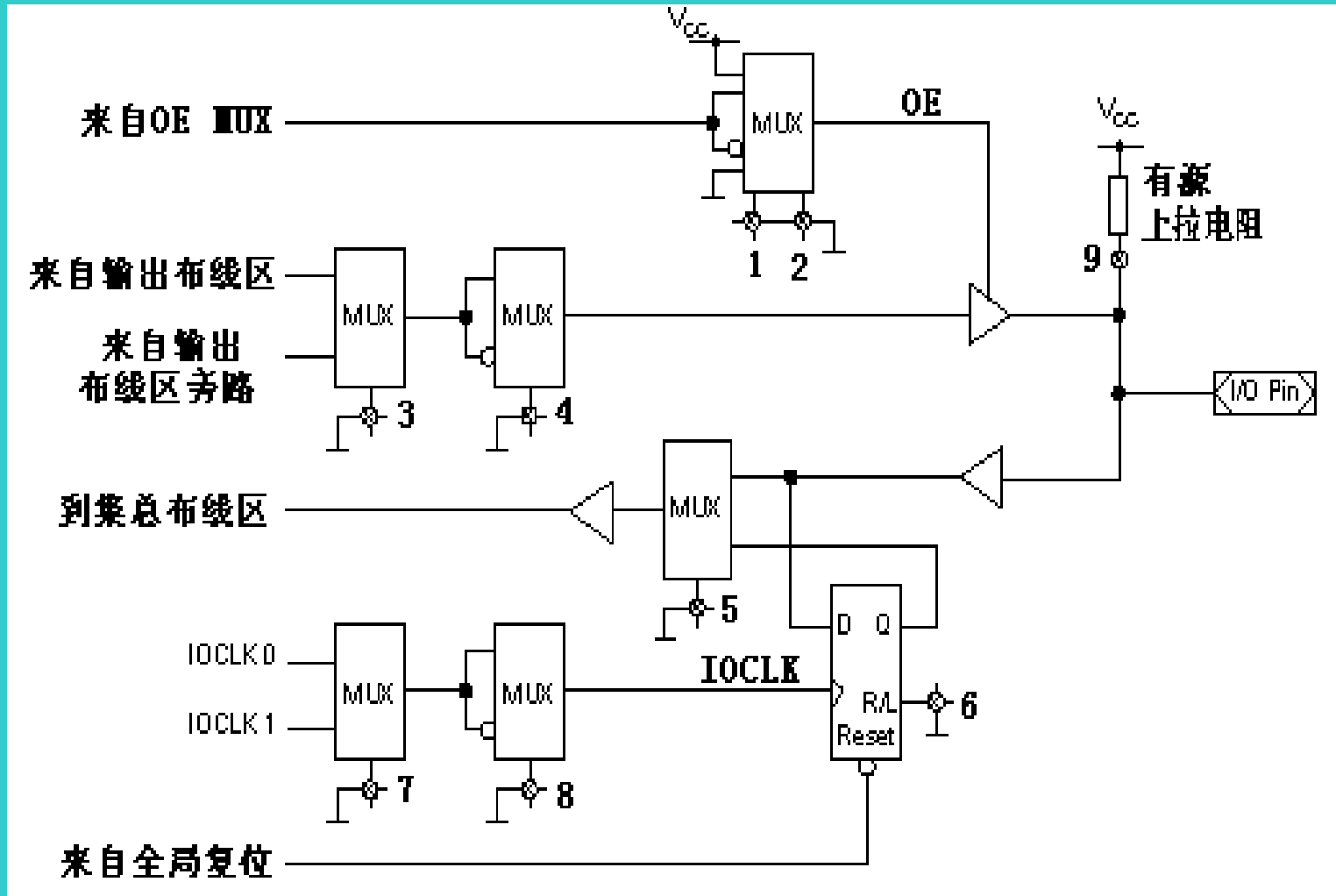


## ispLSI1032 GLB 的配置

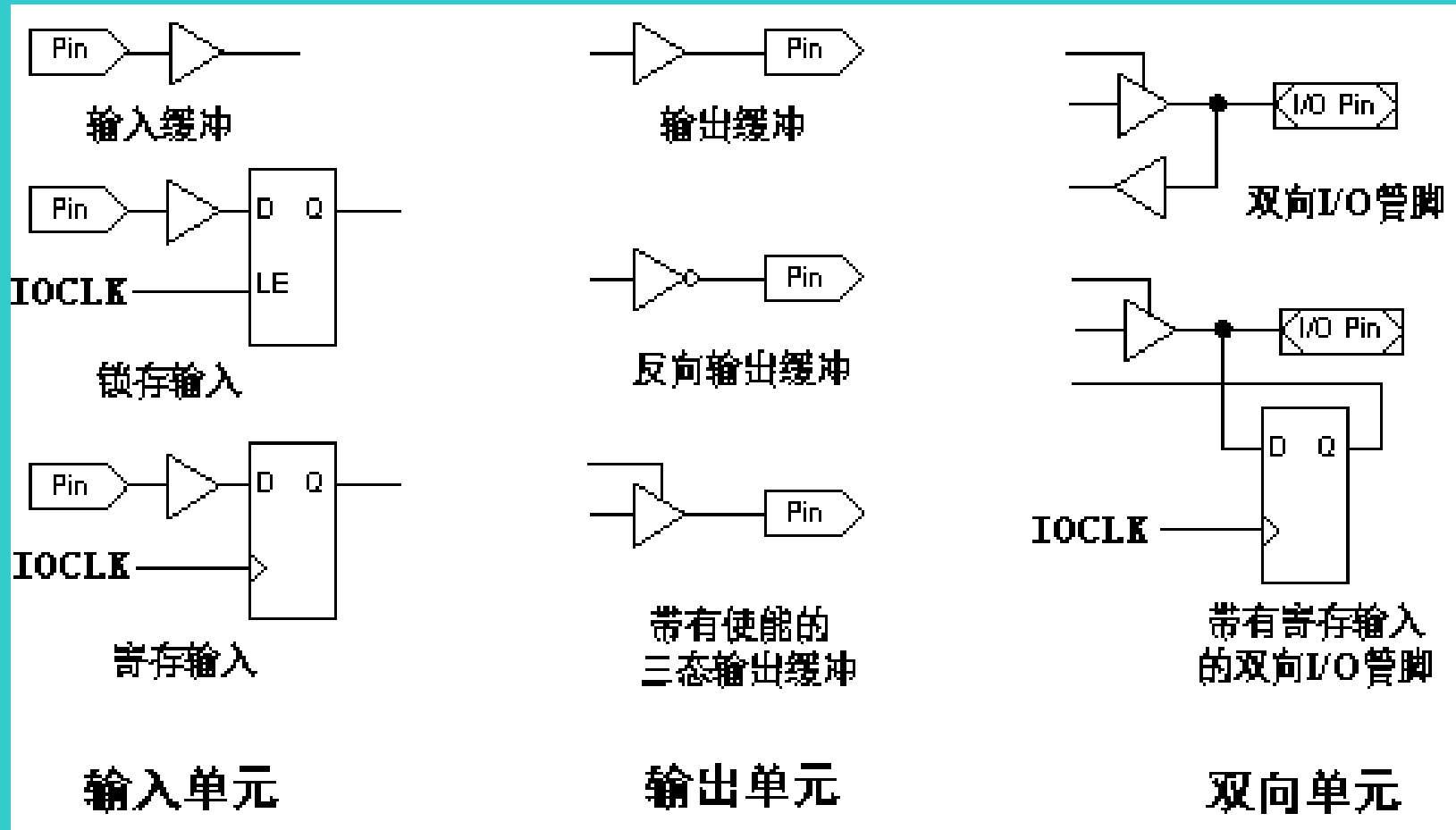


1. GLB的标准组态
2. GLB的单乘积项组态
3. GLB的多模式组态
4. GLB的高速直通组态
5. GLB的异或逻辑组态

## ispLSI1032 器件I/O单元结构



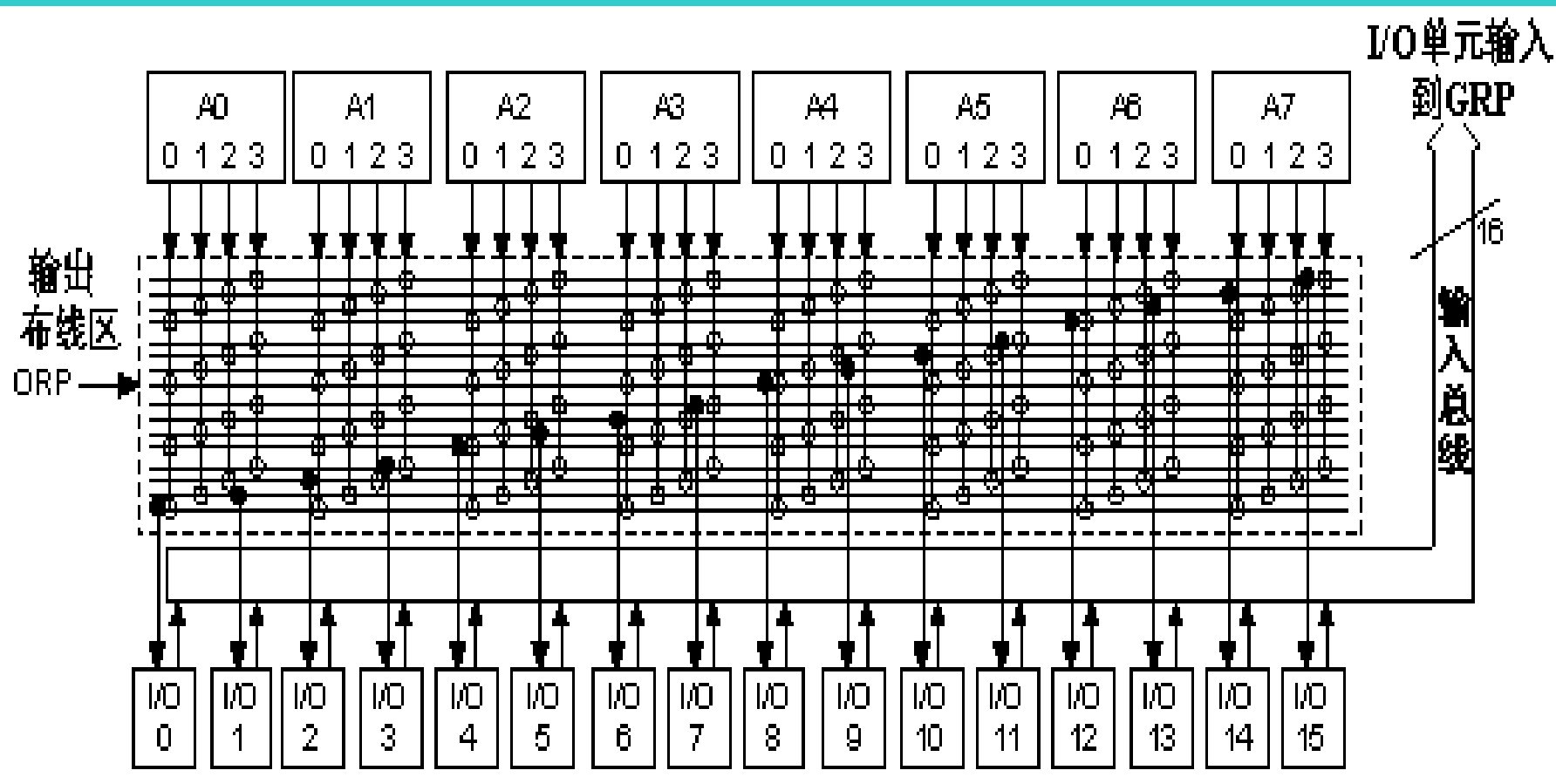
## ispLSI1032 I/O单元的配置





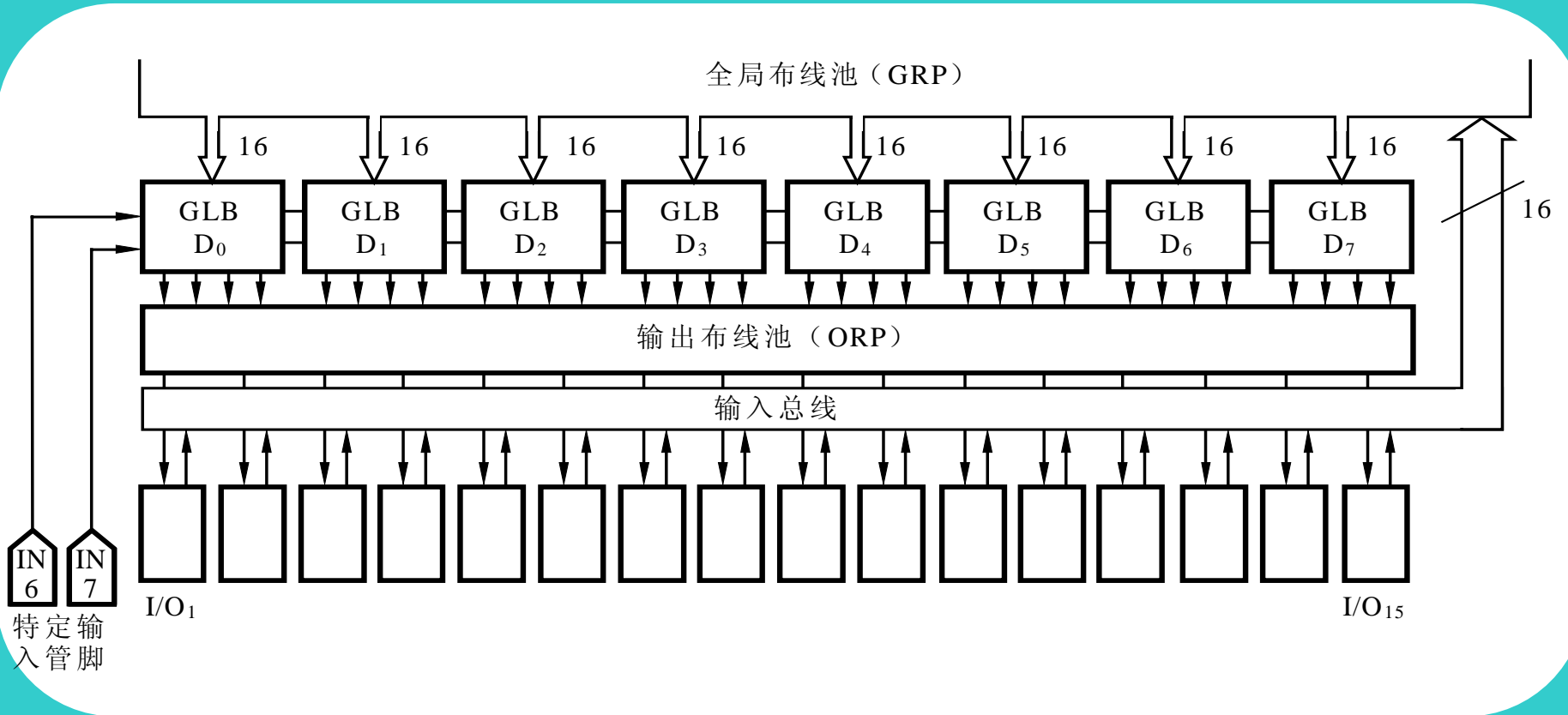


## ispLSI1032器件输出布线区（ORP）的结构





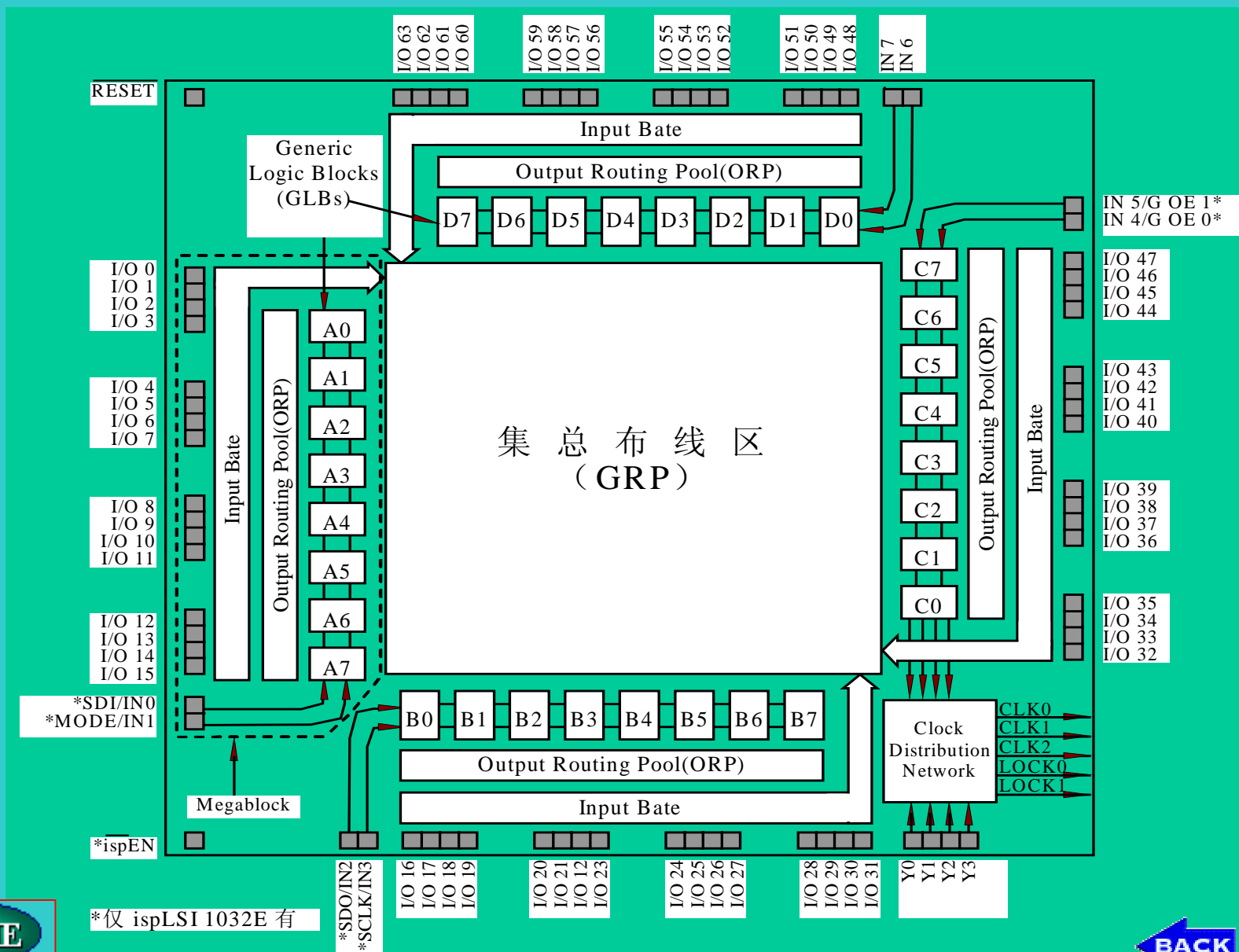
## ispLSI 1032 宏逻辑块



- ispLSI 1032具有四个宏逻辑块；
- ispLSI 1032中的每个宏逻辑块包含8个GLB、16个 I/O单元、1个ORP；



## ispLSI 1032 的全局布线区



\*仅 ispLSI 1032E 有

HOME

BACK NEXT  
BACK



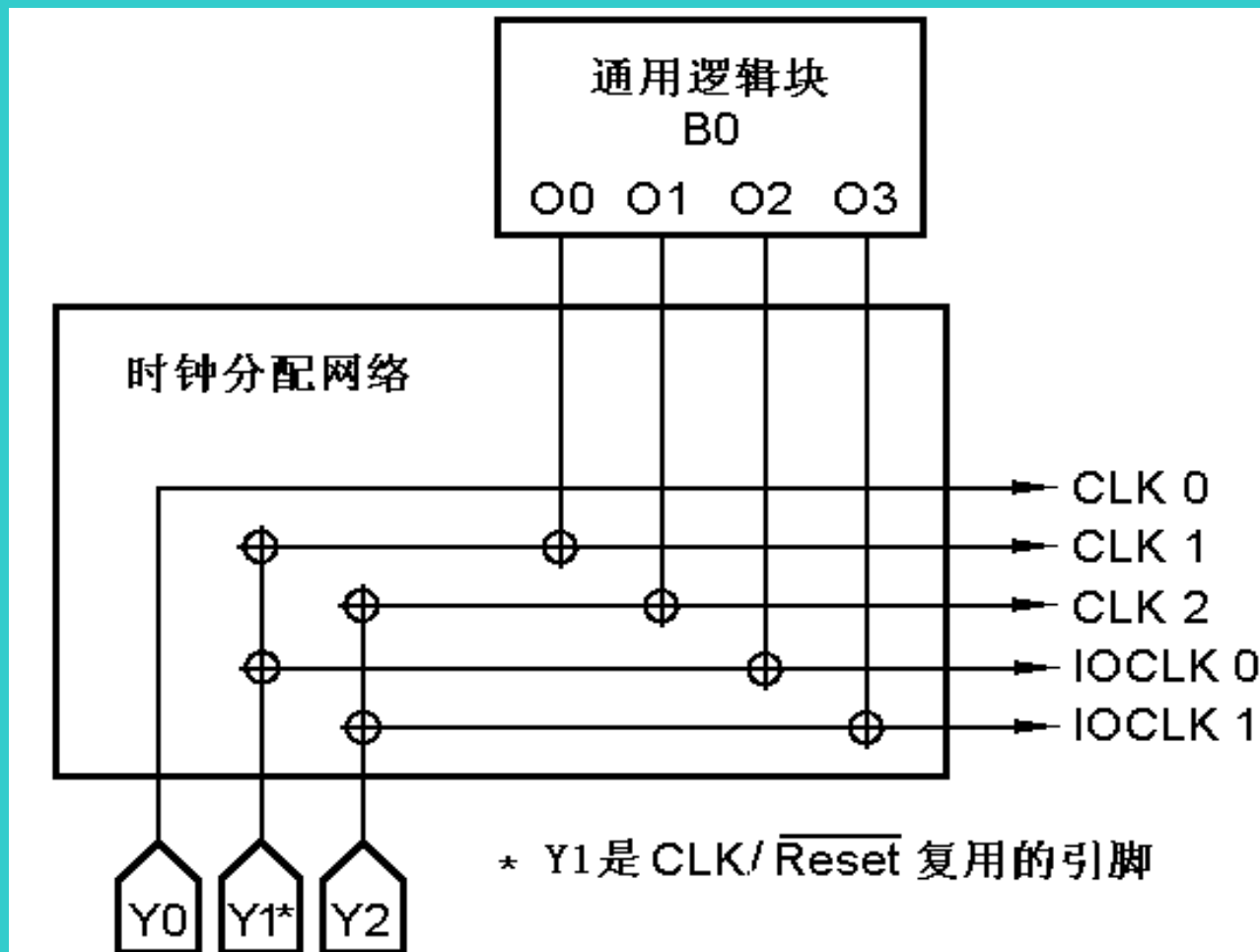
## CPLD资源比较

器件	宏逻辑块	GLBs	I/O 单元
1016/1016E	2	16	32
1024	3	24	48
1032/1032E	4	32	64
1048/1048C/1048E	6	48	96





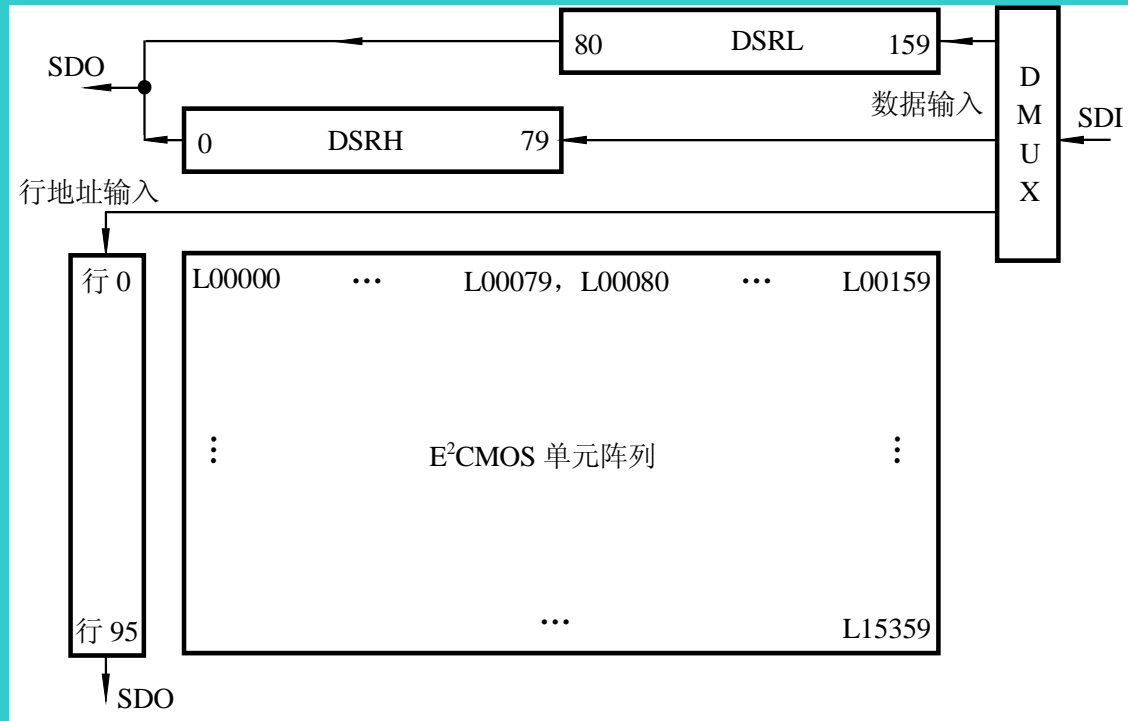
## ispLSI1032器件的时钟分配网络



## 8.4.3 CPLD的编程



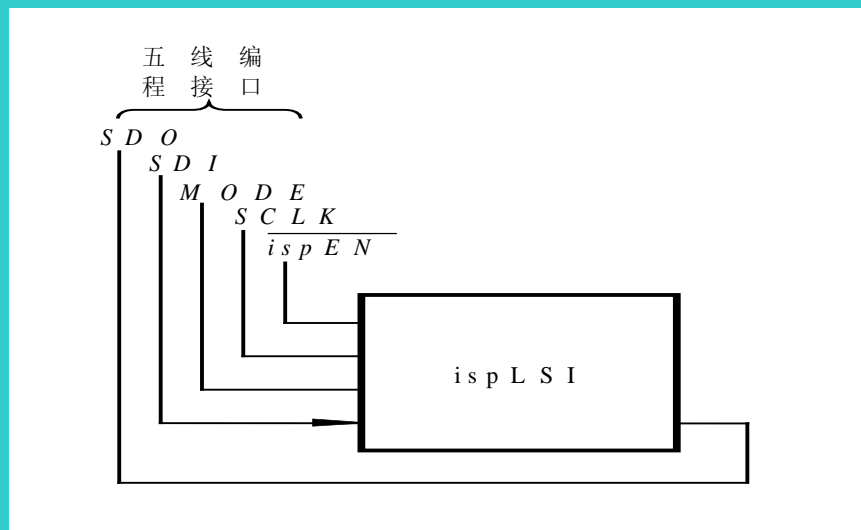
### ispLSI1016 的 E<sup>2</sup>CMOS 单元编程结构



- ispLSI 器件内部具有控制编程逻辑电路，其工作受MODE和SDI信号控制；
- 编程数据和命令以串行方式从SDI送入ispLSI；
- 在写入同时以串行方式将写入的数据从SDO读出进行校验；



### 在系统可编程 (ISP)



### 五线编程接口定义:

ispEN: 编程使能信号; ispEN=0时编程使能, 所有IOC被置成高阻态;

MODE: 模式控制信号; SCLK: 串行时钟输入;

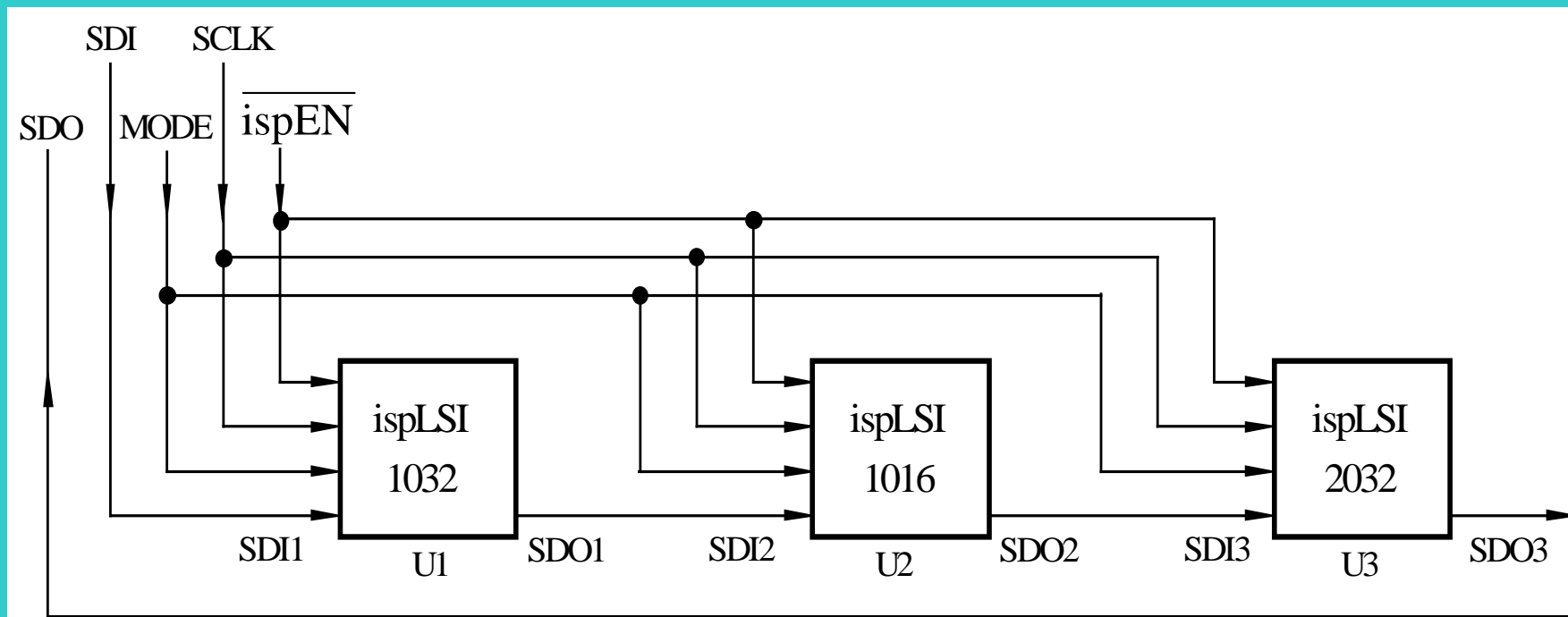
SDI: 串行数据和命令输入; SDO: 串行数据输出;

5线编程接口加1条地线和1条对ispLSI系统电源电压的监测线共7线与计算机的并行口相连





## 多个ispLSI器件的菊花链连接

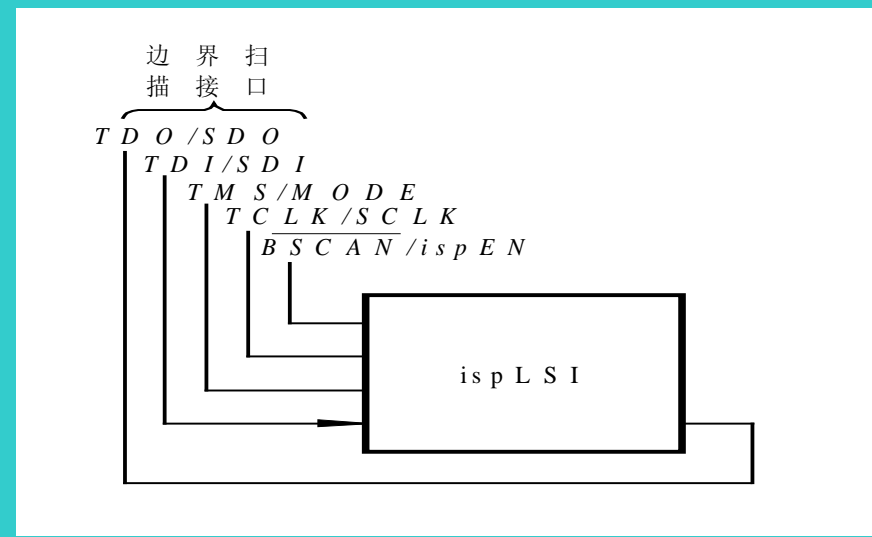




## 边界扫描测试技术



边界扫描：  
高密度器件测试  
国际标准



边界扫描接口定义：与五线编程接口定义复用，另加复位信号

BSCAN：边界扫描使能信号；BSCAN=1时边界扫描使能；

TMS：测试模式选择； TCLK：边界扫描时钟输入；

TDI：测试数据输入； TD0：测试数据输出；

TRST：测试复位；低电平有效；





## 8.5 现场可编程门阵列 (FPGA)

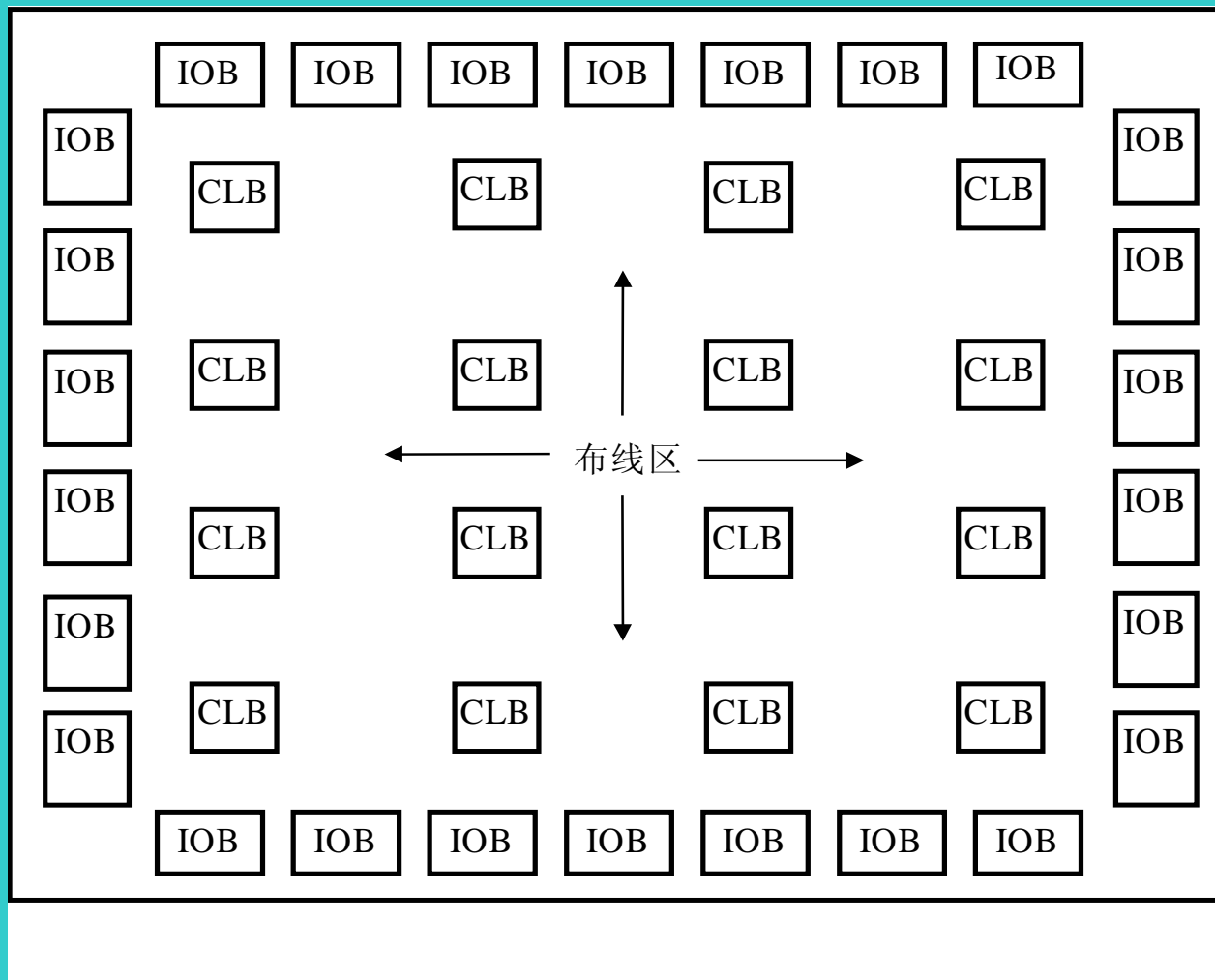
PLD器件基于“与-或”阵列结构；

FPGA器件基于门阵列结构；

特点：

- 不受“与-或”阵列结构限制和宏单元中触发器和I/O端数量限制；
- 依靠内部的门阵列逻辑单元以及它们的连接构成任何复杂的逻辑电路；
- 具有更高的密度和更大的灵活性；

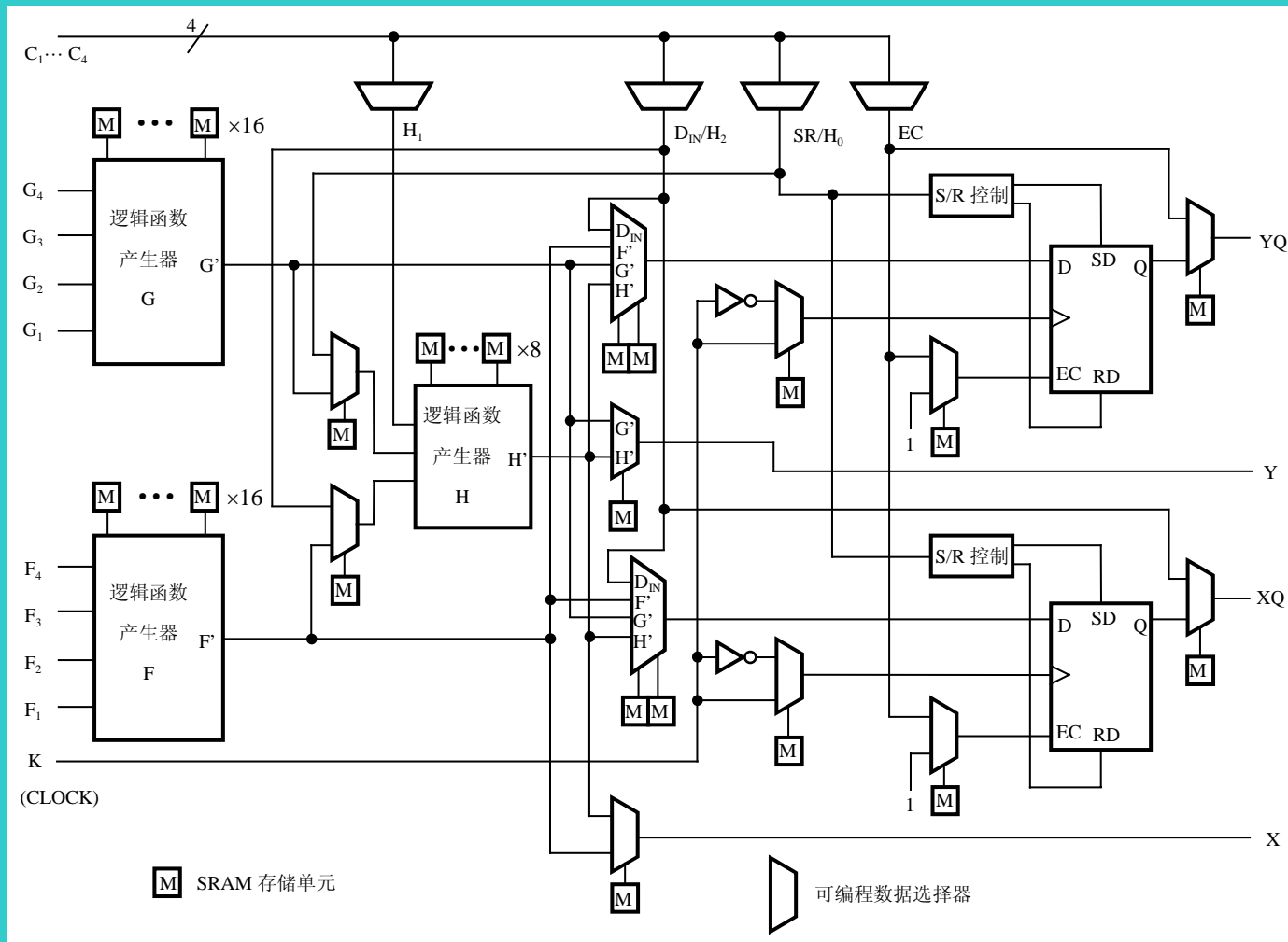
## FPGA 的结构示意图



# 8.5 现场可编程门阵列(FPGA)



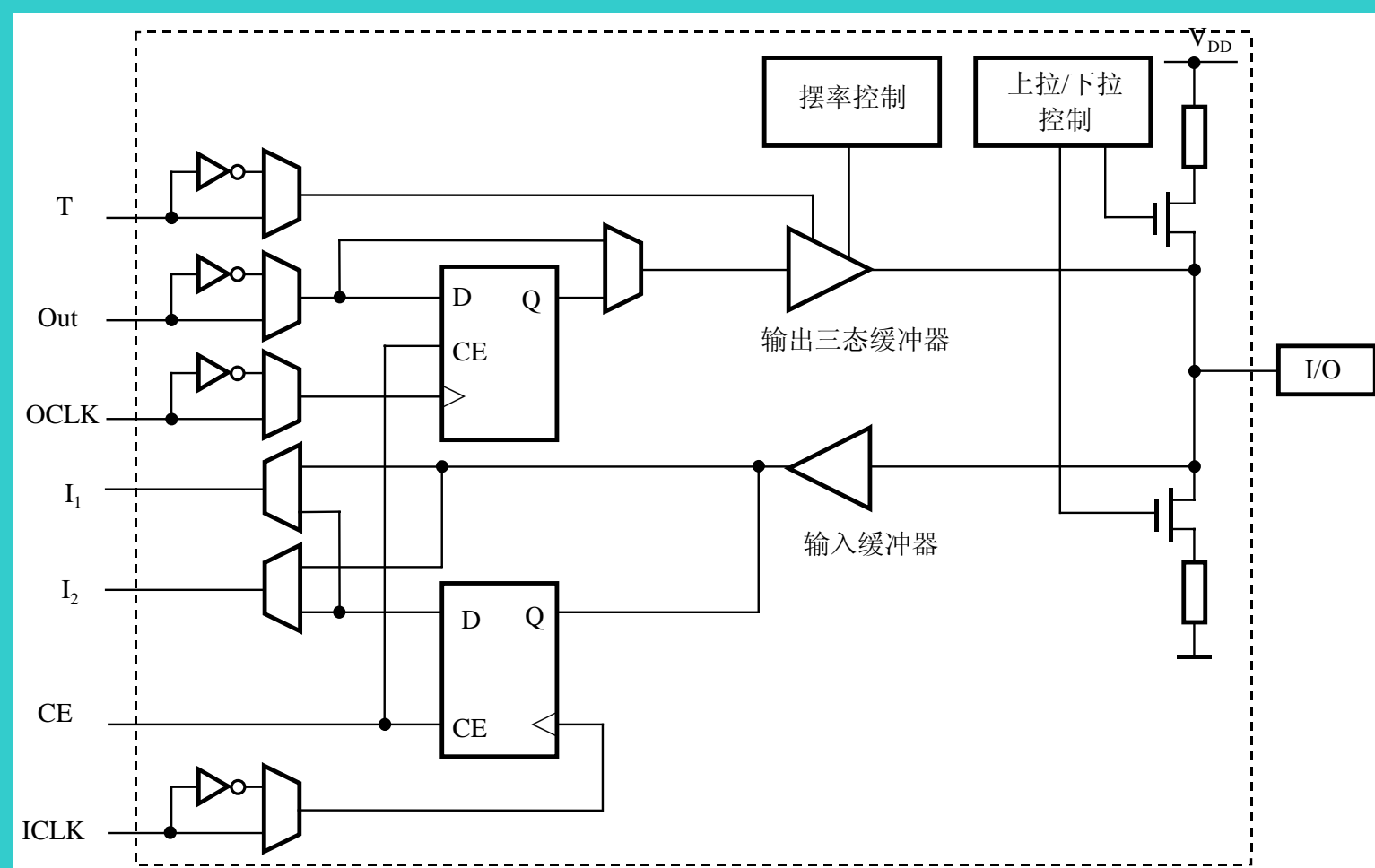
## 简化的CLB原理框图



# 8.5 现场可编程门阵列(FPGA)

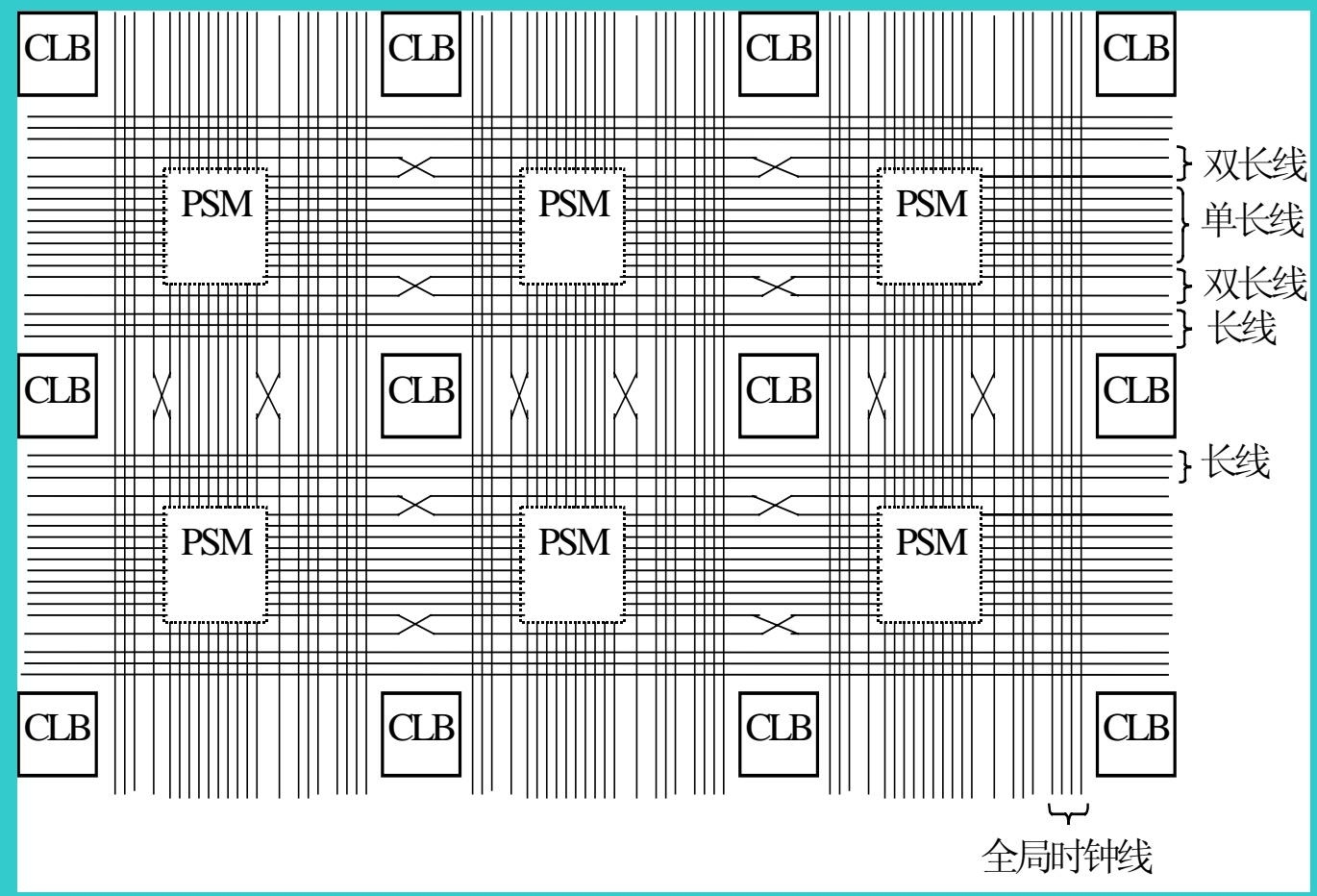


## 简化的IOB原理框图





## 可编程连线资源示意图



# 技术的推动与电路设计的革命

## —用CPLD和FPGA设计数字系统



器件：高集成度、超小型、低功耗、表面贴装

ISP技术与强大的开发系统功能

硬件描述语言（HDL）

计算机功能极大增强与EDA技术普及

硬件设计软件化与软件实现硬件化—ASIC

自顶向下的设计流程

SOC——System On Chip

