

# 数字电路

## Digital Circuits and System

李文明

liwenming@ict.ac.cn



# 可编程逻辑器件



# 可编程逻辑器件

- 可编程器件概述
- 简单可编程逻辑器件 (SPLD)
  - 可编程阵列逻辑 (PAL)
  - 通用阵列逻辑 (GAL)
- 大容量可编程逻辑器件
  - 复杂的可编程逻辑器件 (CPLD)
  - 现场可编程门阵列 (FPGA)
- PLD的编程



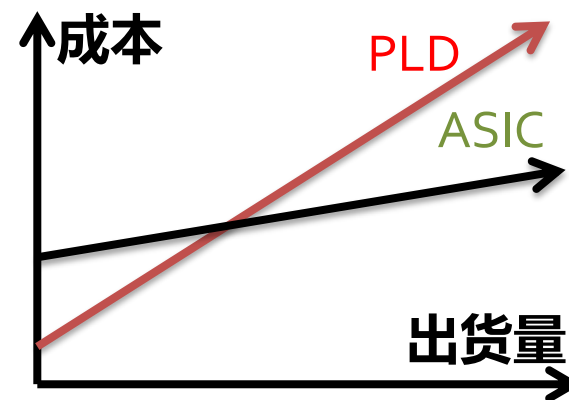
# 可编程逻辑器件

- 可编程器件概述
  - 简单可编程逻辑器件 (SPLD)
    - 可编程阵列逻辑 (PAL)
    - 通用阵列逻辑 (GAL)
  - 大容量可编程逻辑器件
    - 复杂的可编程逻辑器件 (CPLD)
    - 现场可编程门阵列 (FPGA)
  - PLD的编程

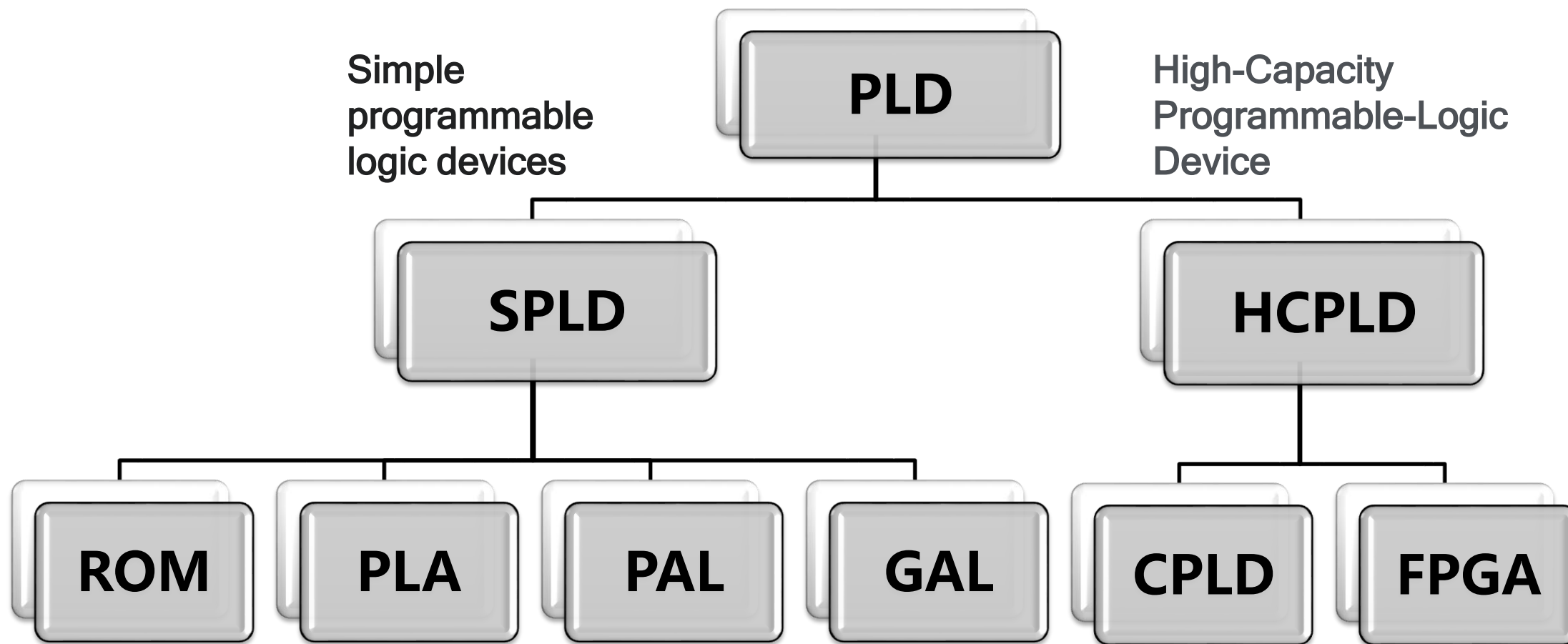


# 可编程器件概述

- 通用器件： **54/74、74HC、74HCT**等系列
  - 随系统规模扩大：焊点多，可靠性下降，功耗增加、成本升高，占用空间扩大
- 专用型：ASIC (Application Specific Integrated Circuit)
  - 要承担设计风险、周期长、成本高
- 系统设计师们希望自己设计ASIC芯片，缩短设计周期，能在实验室设计好后，立即投入实际应用
- 可编程器件 (PLD : Programmable Logic Device )
  - Xilinx was founded in 1984
  - Altera was founded in 1983
  - Lattice 1983, Cypress 1982, Actel 1985



# 可编程逻辑器件分类



# 可编程逻辑器件

- 可编程器件概述
- 简单可编程逻辑器件 (SPLD)
  - 可编程阵列逻辑 (PAL)
  - 通用阵列逻辑 (GAL)
- 大容量可编程逻辑器件
  - 复杂的可编程逻辑器件 (CPLD)
  - 现场可编程门阵列 (FPGA)
- PLD的编程



# 简单可编程逻辑阵列 (SPLD)

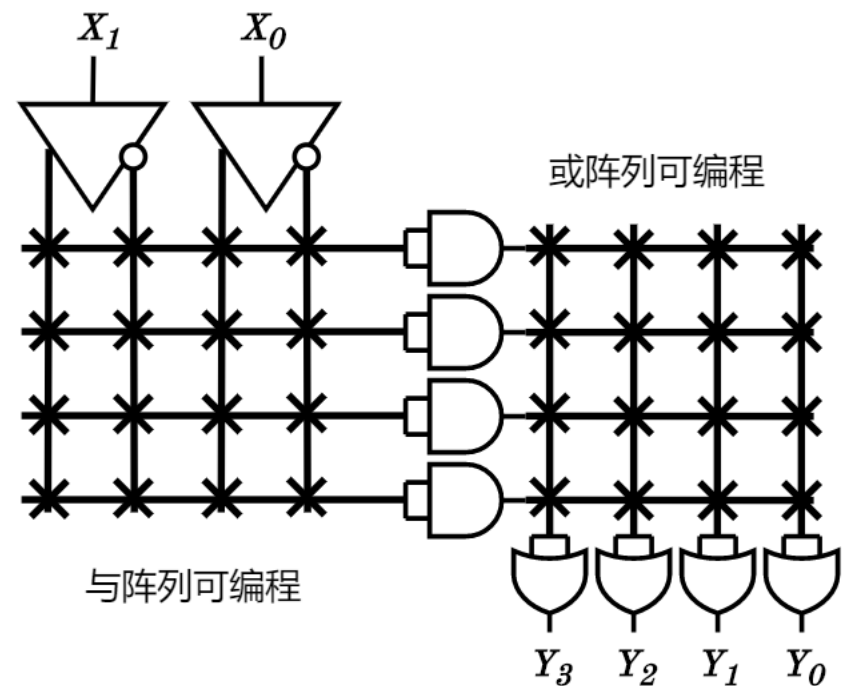
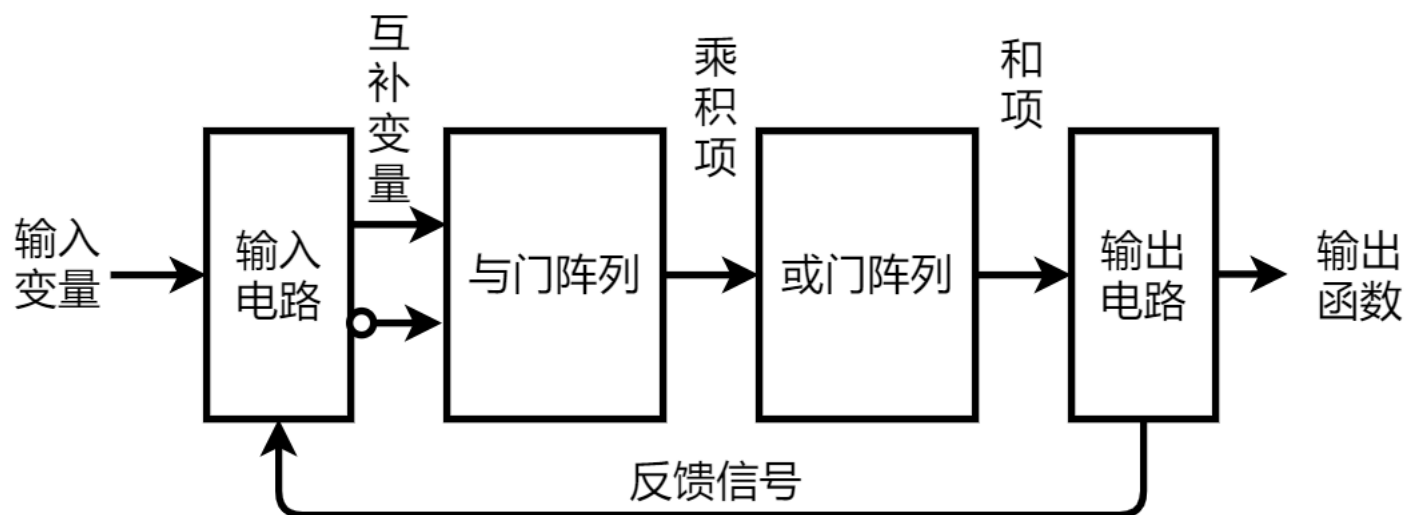
- SPLD包括3类器件：
  - PLA: Programmable Logic Array
  - PAL: Programmable Array Logic
  - GAL: Generic Array Logic
- PLA的结构指其中的“与”逻辑、“或”逻辑都可以编程，对于单输出逻辑，无需对“或”逻辑编程
- PAL：一般只能编程一次 (One Time Programmable, OTP)
- GAL：是一类特定的PAL，可重复编程使用，GAL最早由Lattice半导体公司推出，后来授权给其他公司
- PAL/GAL的基本结构是“或”逻辑固定，“与”逻辑可编程





# PLD的基本结构

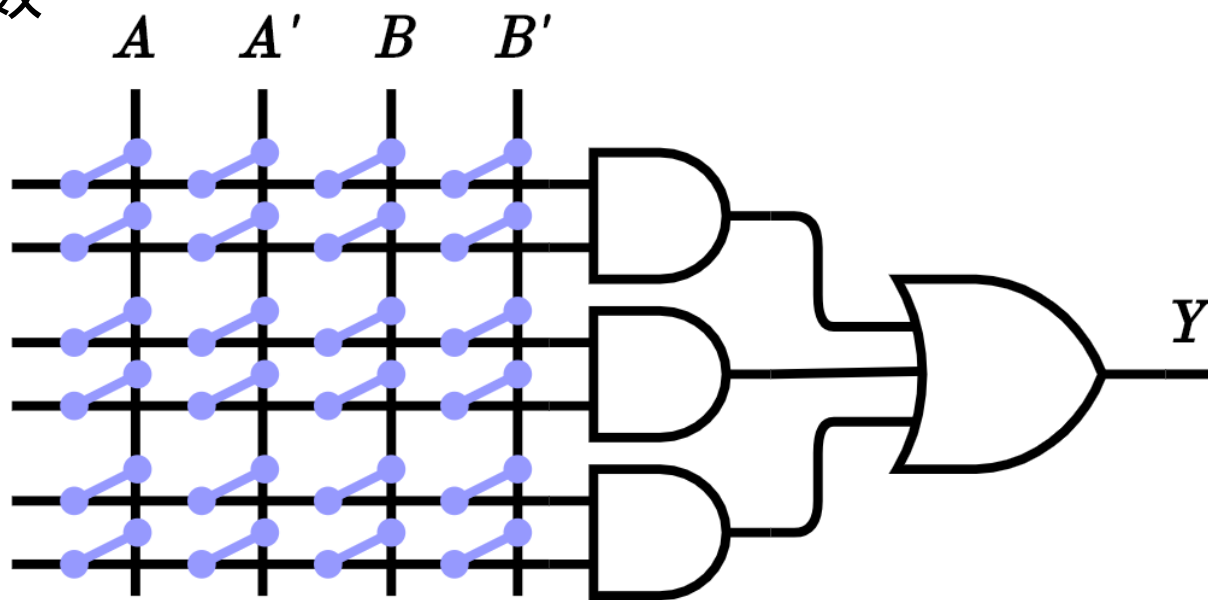
- 如图为PLD的一般结构
- 通过或阵列直接输出，可构成组合逻辑函数
- 通过寄存器输出，附加反馈逻辑，可构成时序逻辑



# PAL基本结构

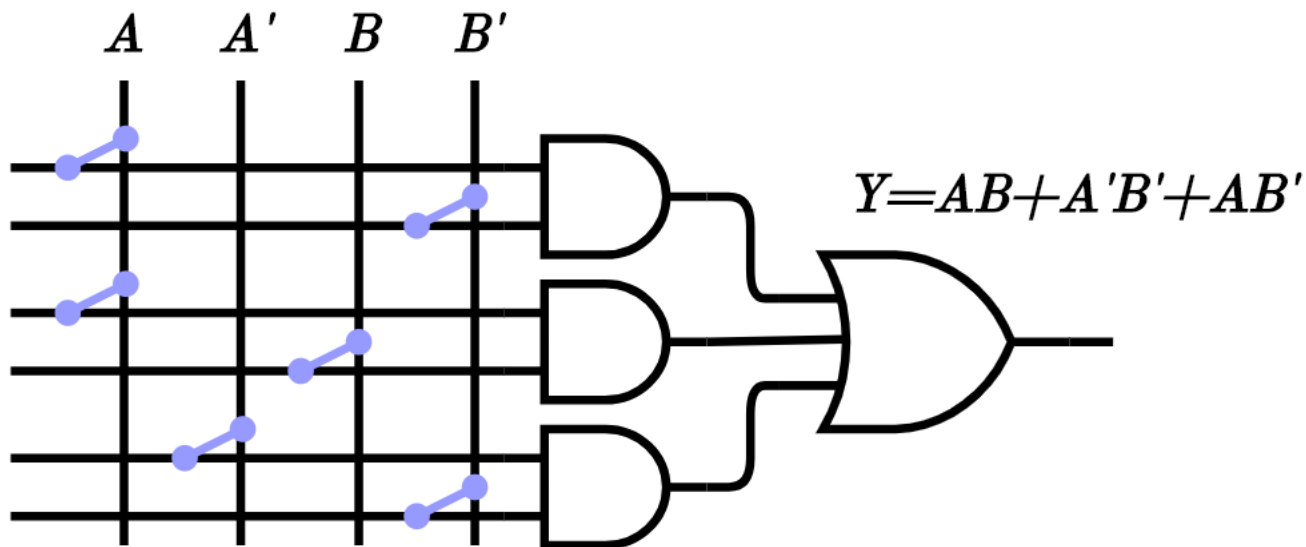
- PAL包括可编程的与阵列，连接到一个固定的或门上
- 一般PAL采用熔丝技术实现，只能编程一次
- PAL可以实现任意最小项和逻辑函数

- 熔丝编程前联通，变更后熔断
- 熔丝可以采用二极管
- 也可以采用MOS管实现



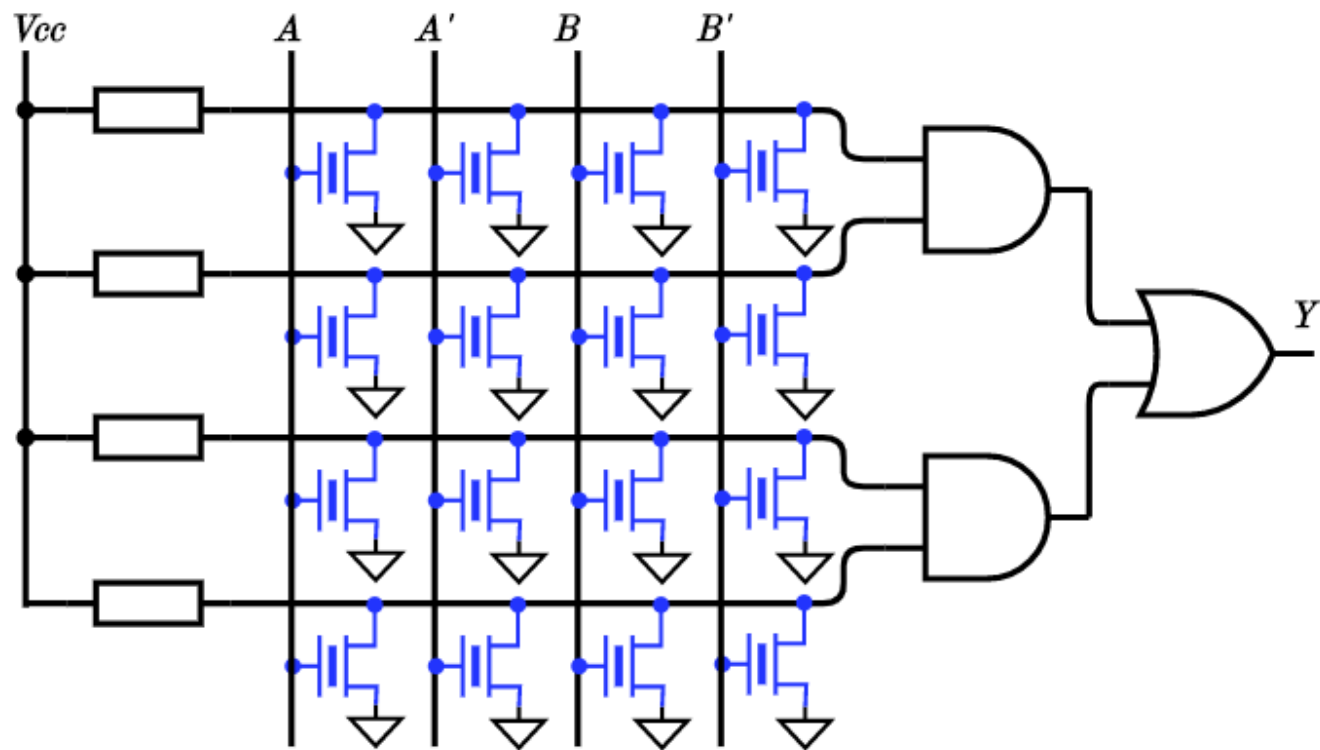
# PAL编程举例

- 如图所示
- 编程后，不需要连接的熔丝开路
- 需要连接的熔丝保留
- 可以产生图中所示逻辑函数
- PAL的主要问题：
  - 只能编程一次
  - 灵活性差
  - 考虑更改熔丝的实现技术，使其具备重复编程的能力



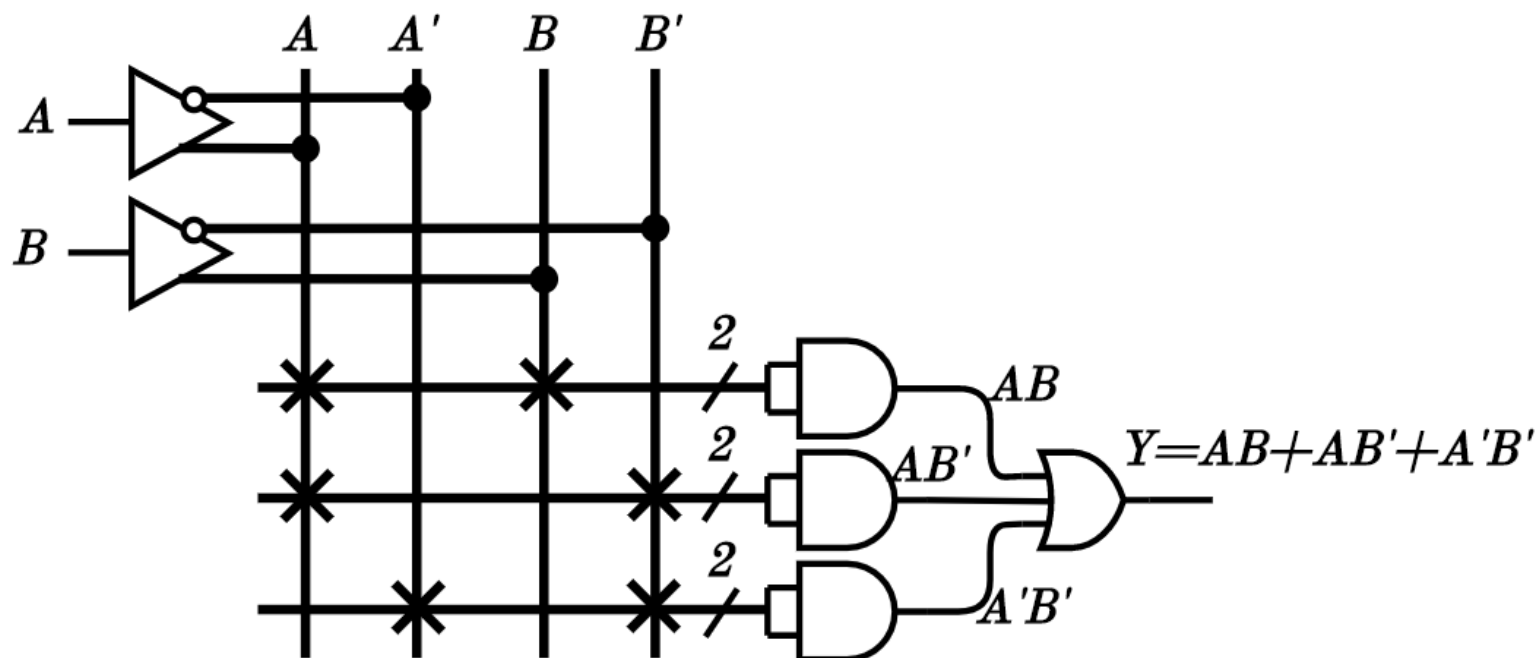
# GAL基本结构

- 如图所示
- GAL的基本结构与PAL相同
- 与逻辑可编程，或逻辑固定
- 可反复编程，实用电可擦除的浮栅晶体管代替熔丝，实现与逻辑的编程和擦除功能



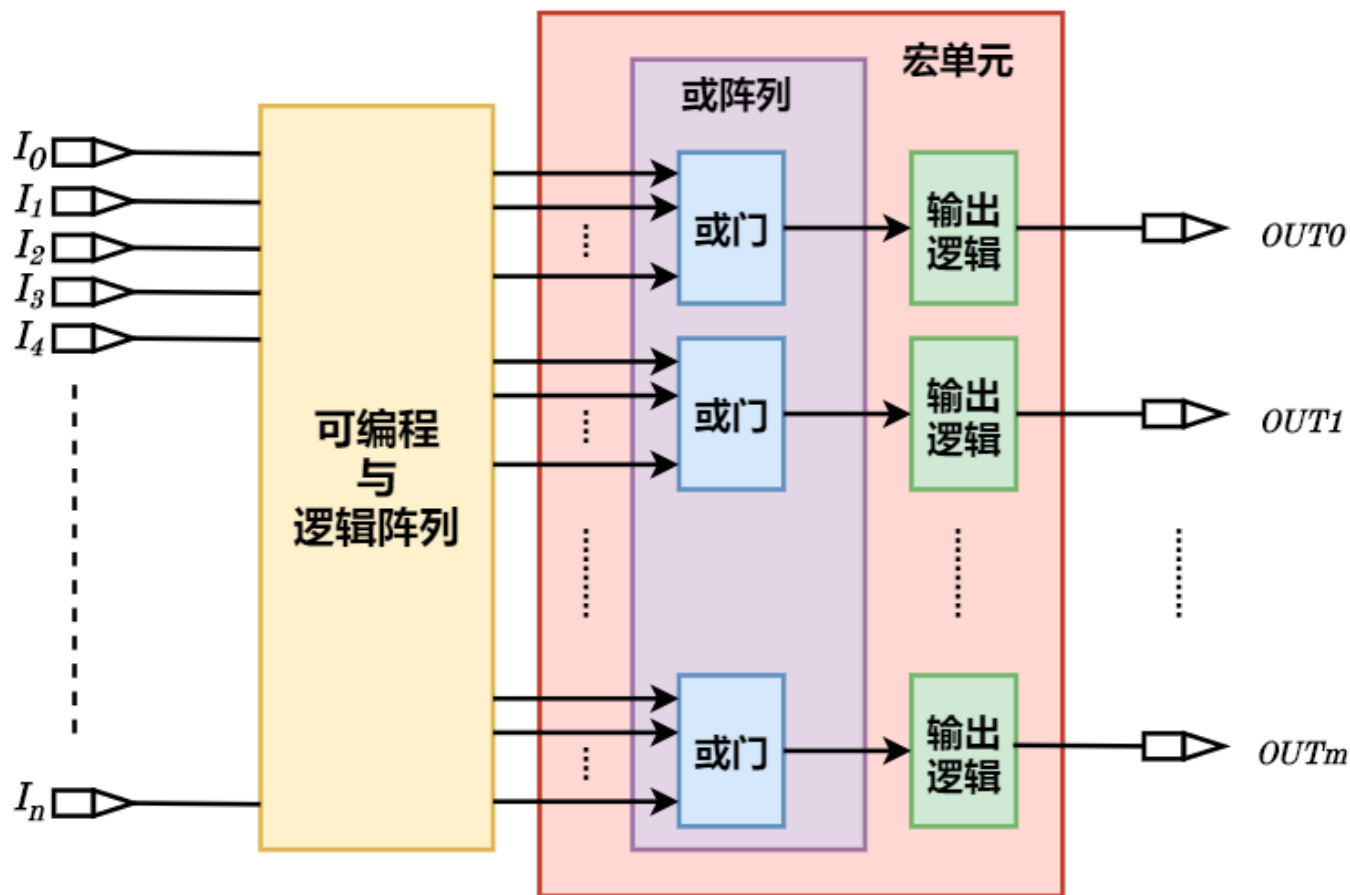
# PAL/GAL的简单符号表示

- 实际的PAL/GAL包括多个可编程的与门阵列，以及多个或门
- 还包括输入输出驱动、输入反相器、输出三态门等功能模块
- 一个实现组合逻辑函数的举例



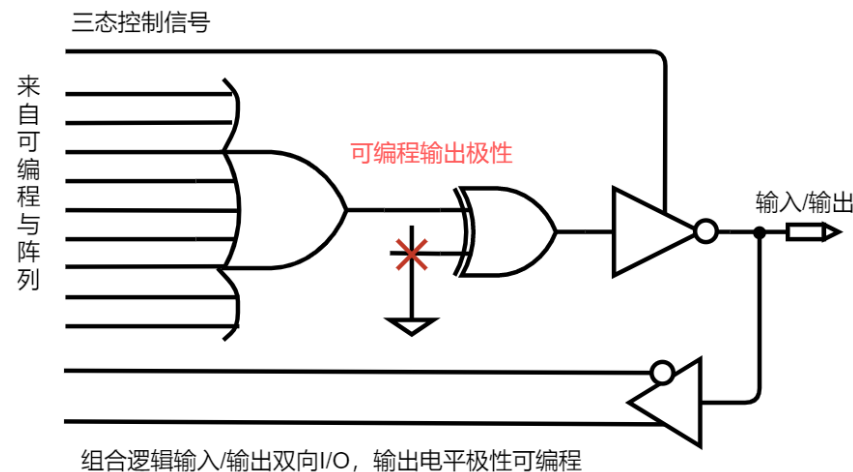
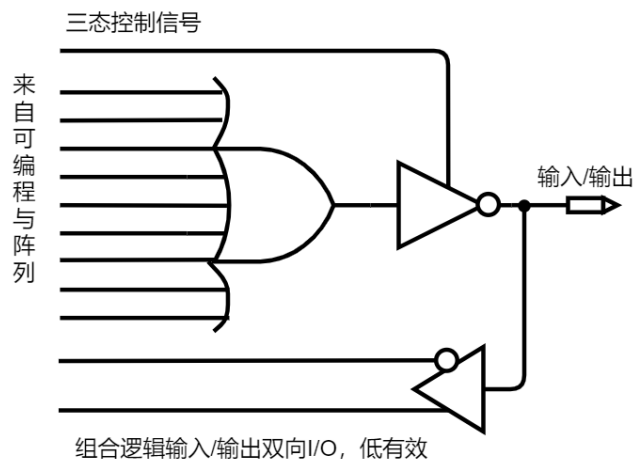
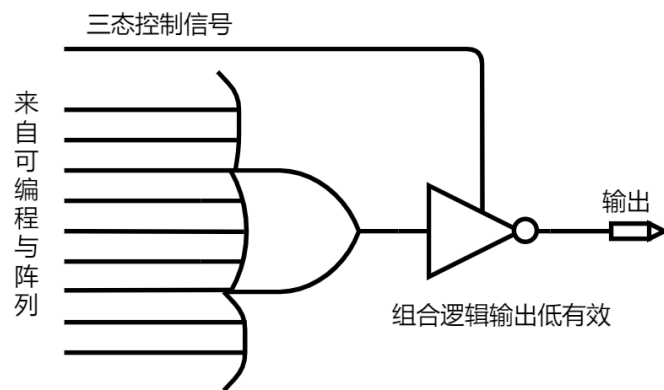
# PAL/GAL框图

- 如图所示为PAL/GAL的基本结构
- PAL/GAL的差别在于GAL可重复编程，而PAL只能编程一次
- 或门阵列及其相应的输出逻辑称为宏单元
- 一般封装为20~28脚，决定了输入/输出个数，如16V8，16个I/O，最多8个输出



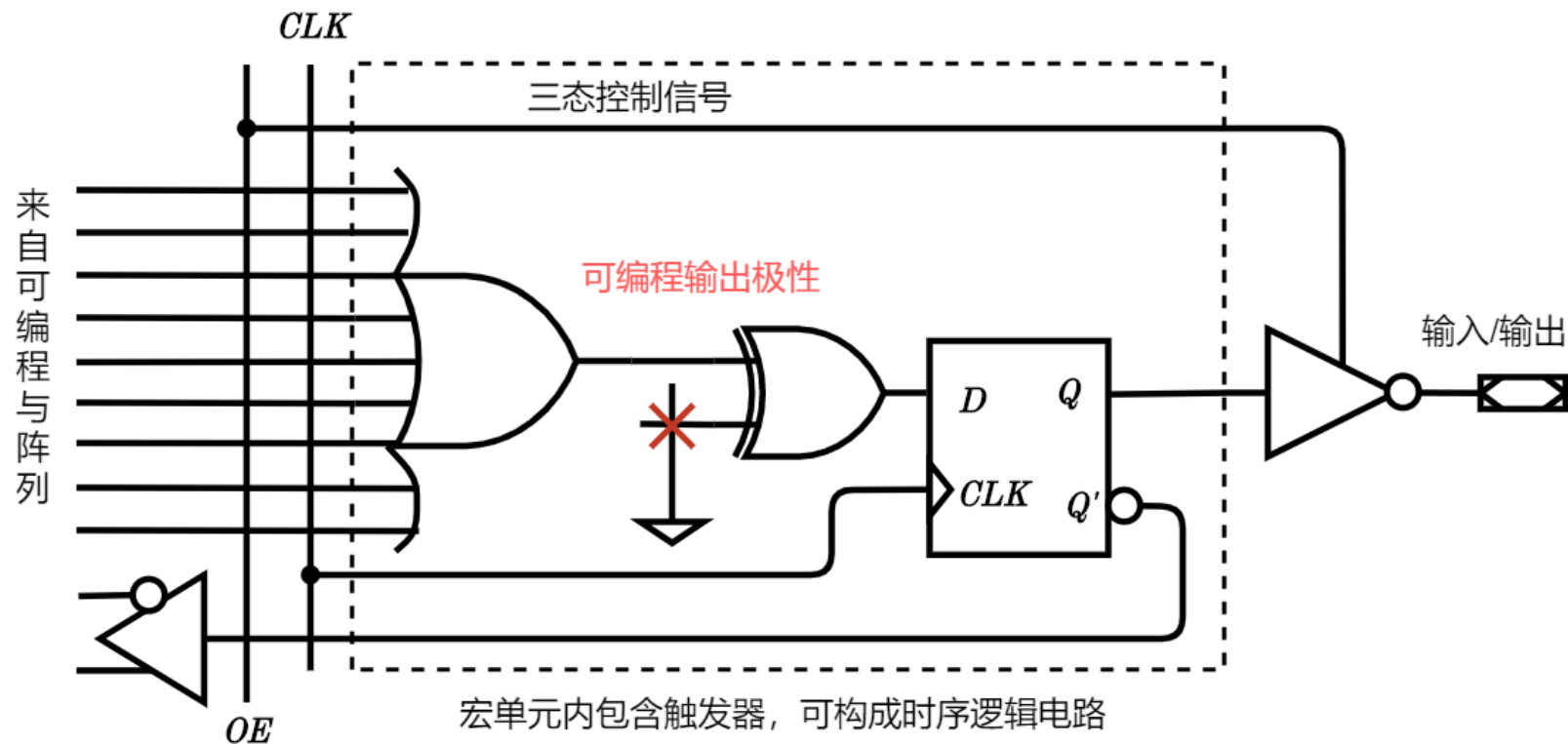
# PAL/GAL宏单元结构

- 不同器件的输出宏单元结构不同
- 宏单元可以配置成组合逻辑电路
- 如果宏单元中包含触发器，可以配置成时序逻辑电路
- 输出引脚可以只配置成输出，也可以配置成输入/输出复用，输出电平的有效极性可以配置修改
  - $A \oplus 0 = A$ ,  $A \oplus 1 = A'$



# PAL/GAL带触发器的宏单元结构

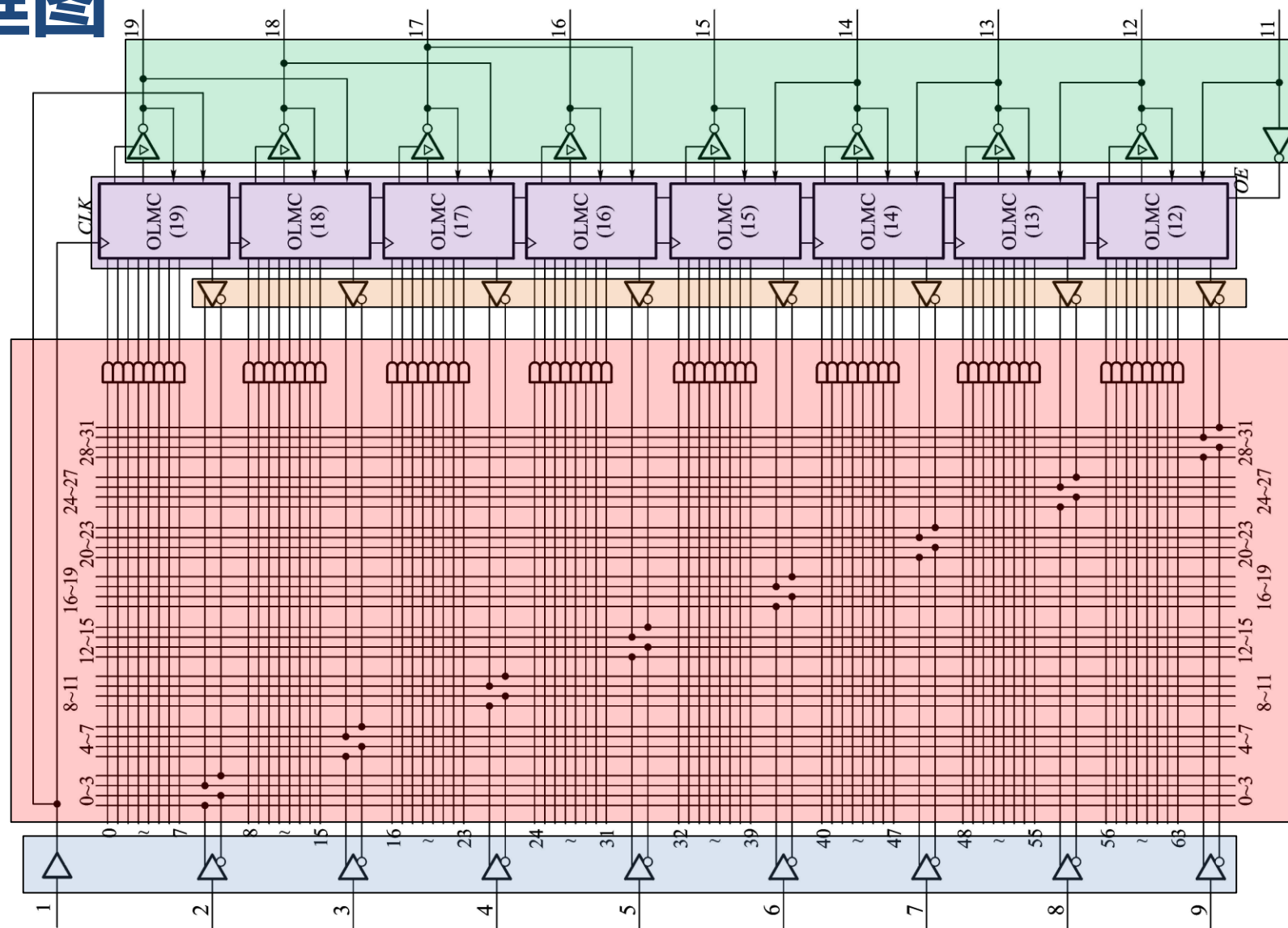
- 有些GAL器件的宏单元内包含触发器
- 可构成时序逻辑电路，GAL16V8的宏单元如下





# GAL16V8原理框图

- 输入缓冲器
- 可编程“与”阵列 64X32
- 反馈输入缓冲器
- 输出逻辑宏单元 (OLMC)
- 三态输出缓冲器



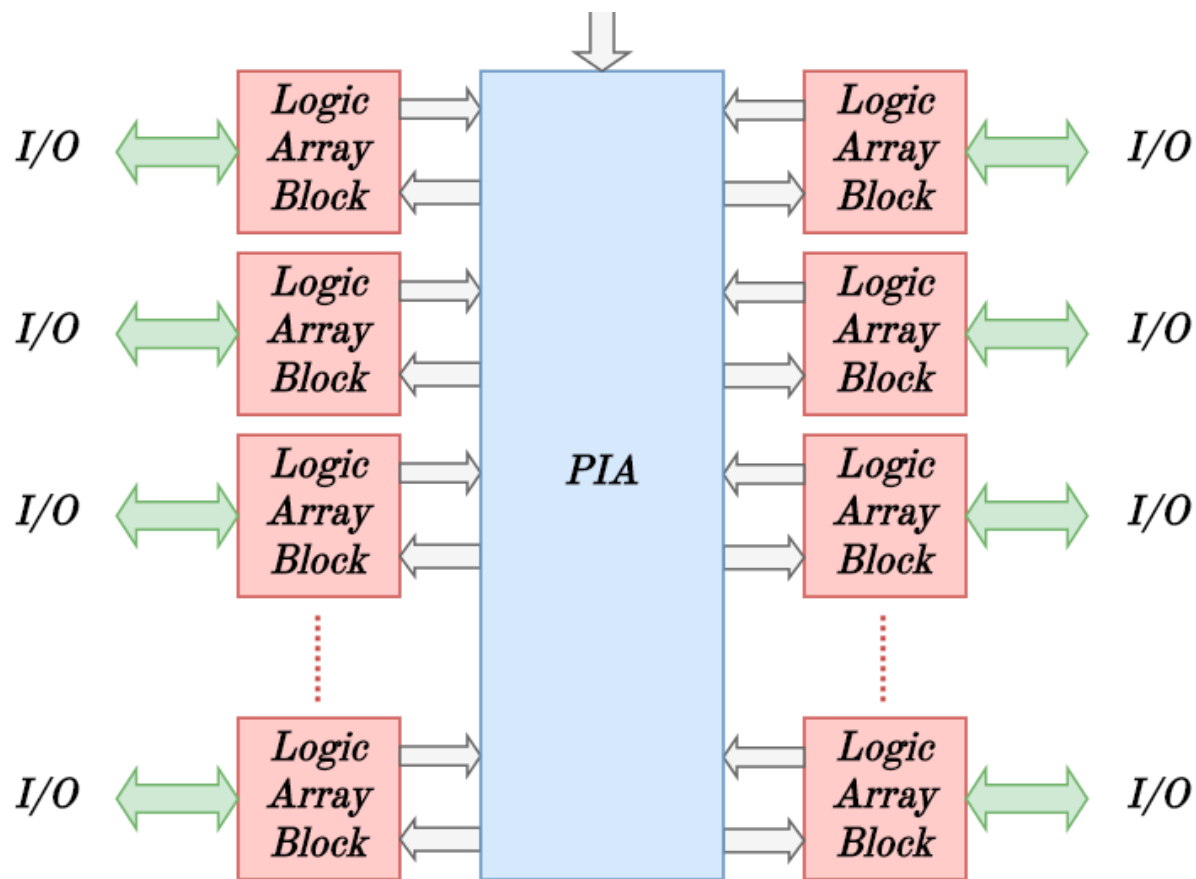
# 可编程逻辑器件

- 可编程器件概述
- 简单可编程逻辑器件 (SPLD)
  - 可编程阵列逻辑 (PAL)
  - 通用阵列逻辑 (GAL)
- 大容量可编程逻辑器件
  - 复杂的可编程逻辑器件 (CPLD)
  - 现场可编程门阵列 (FPGA)
- PLD的编程



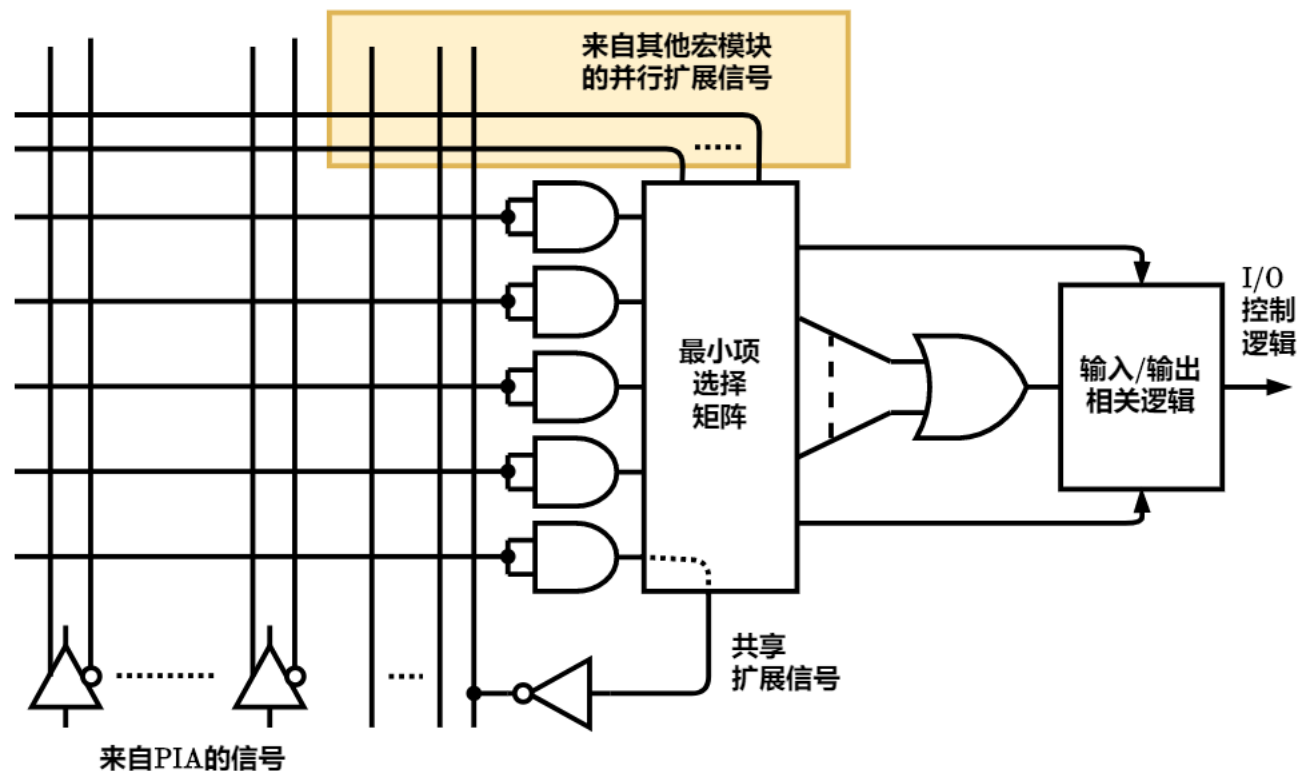
# 复杂可编程逻辑器件 (CPLD)

- Complex programmable Devices 包含多个简单可编程阵列，以及可编程的互连结构
- 其中的简单可编程阵列可以称为LAB (Logic Array Block)，每个LAB可以包含多个Logic Element (LE)
- 可编程互连结构称为PIA (Programmable Interconnection Array)
- 输入信号可以通过编程配置连接到所有LAB
- 使用EDA工具软件实现对CPLD器件内的各部分进行编程配置
- CPLD的编程配置采用EEPROM或SRAM技术实现



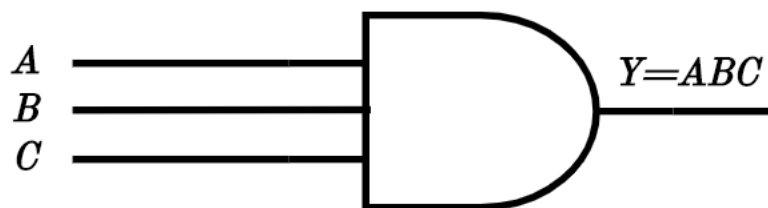
# CPLD中的宏单元 (MacroCell)

- 宏单元包括：
  - 可编程 “与” 逻辑
  - 乘积项选择矩阵
  - 或门
  - 输出/输入粘合逻辑
  - 并行扩展器连线
  - 共享扩展连线
- 并行扩展、共享扩展都是为了实现多个宏单元之间直接连接，扩展实现更复杂的逻辑函数

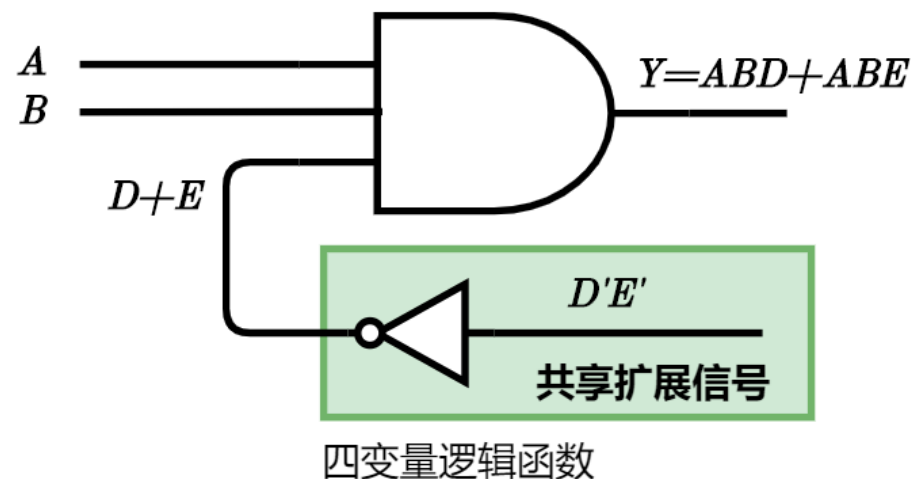


# 共享扩展和并行扩展方法

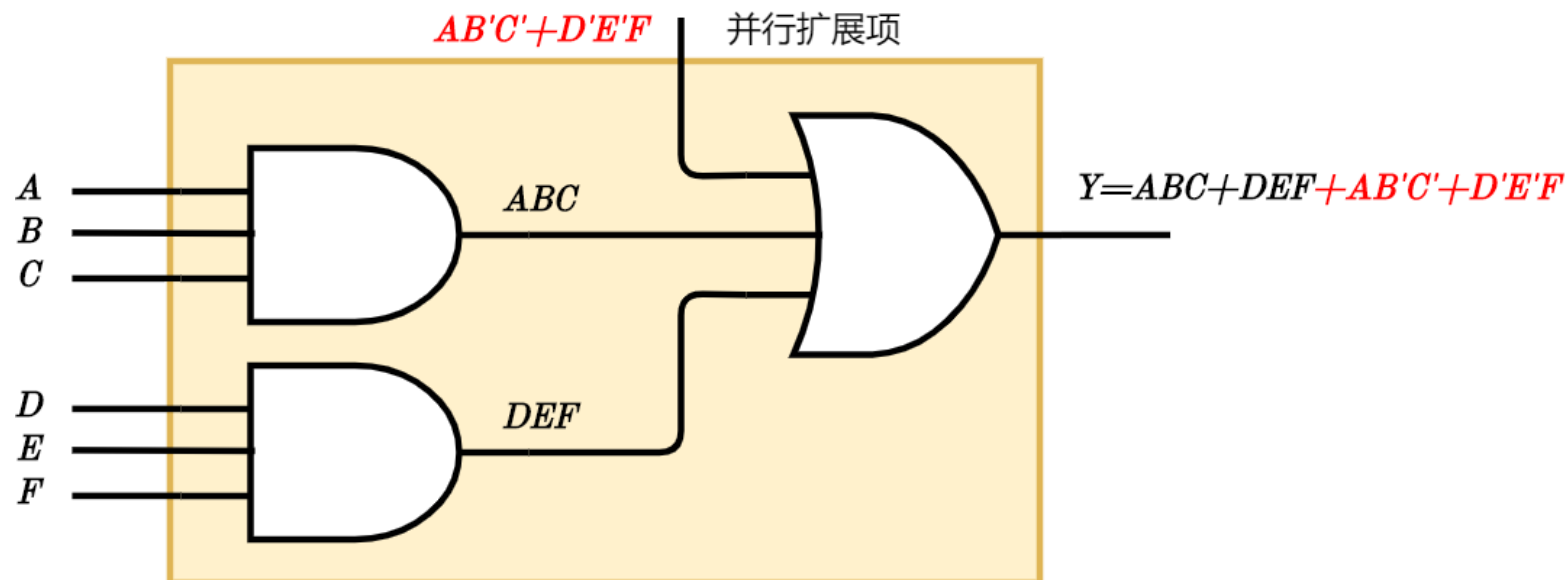
- 共享扩展可以增加组合逻辑函数的变量数，以及逻辑函数的复杂度



三变量逻辑最小项

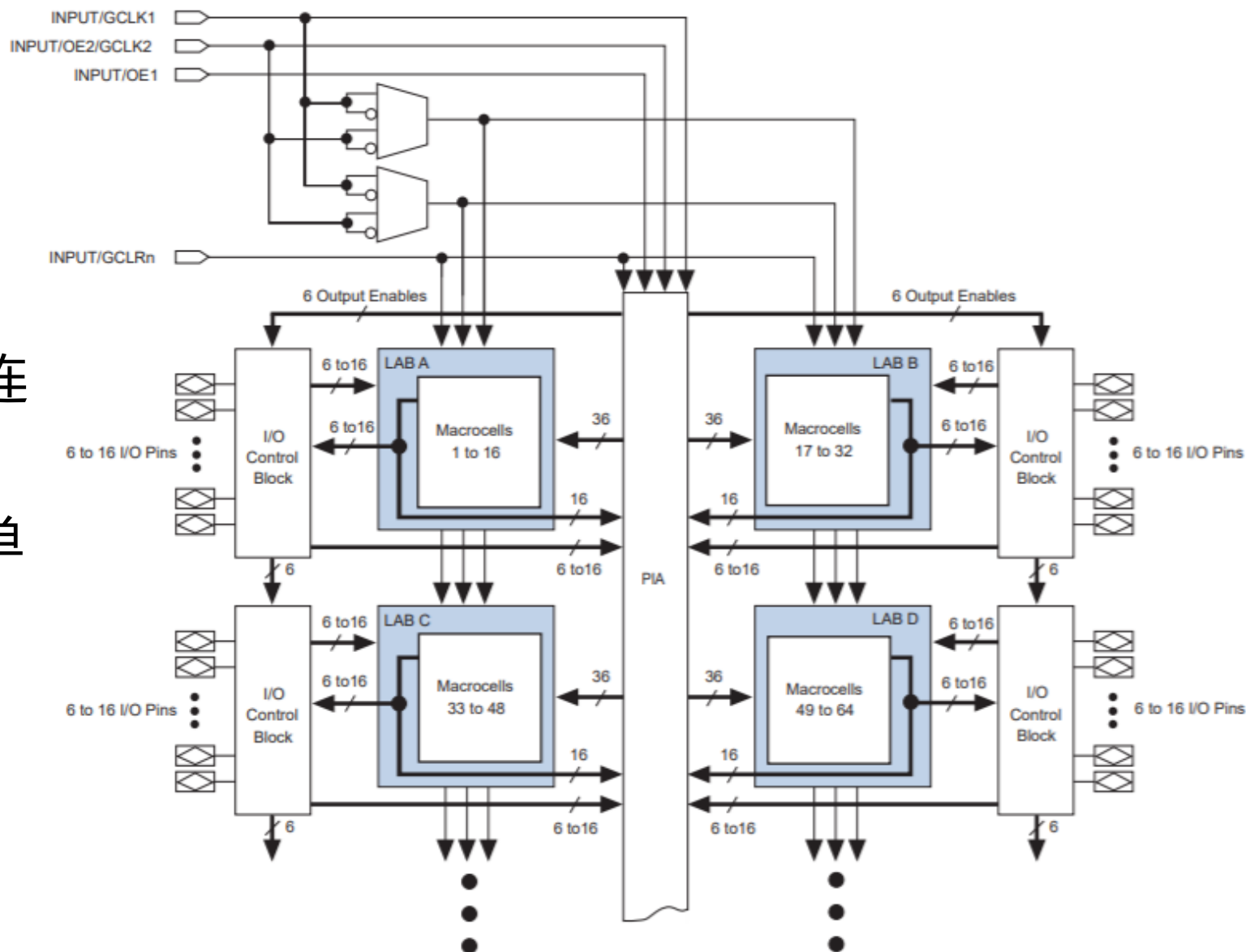


- 并行扩展也可以增加组合逻辑函数的变量数，逻辑函数的复杂度



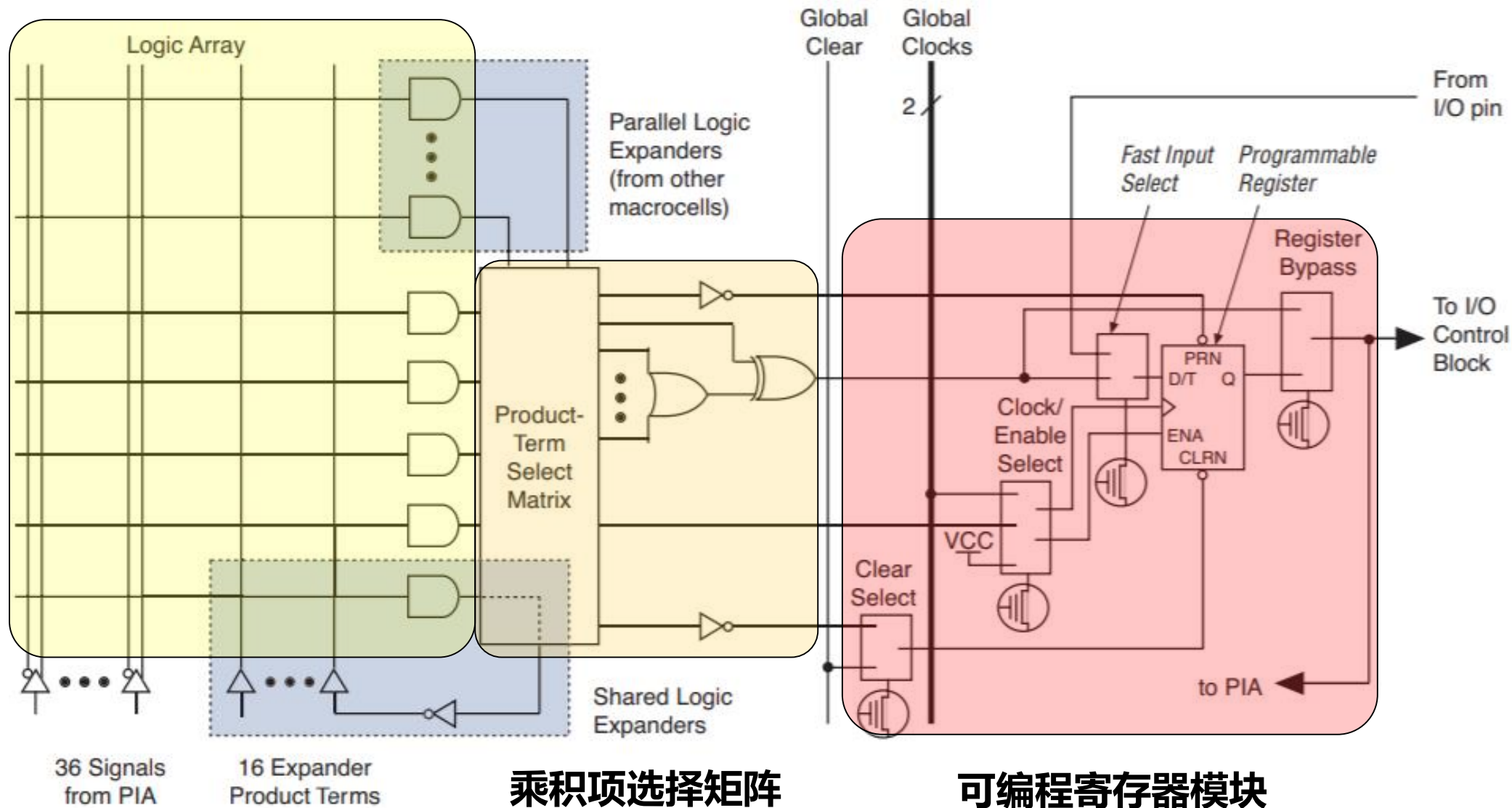
# Altera Max7000 CPLD结构

- 每个LAB包含16个宏单元
- 多个宏单元间通过PIA相连
- 专用引脚、I/O引脚与宏单元间通过全局总线连接



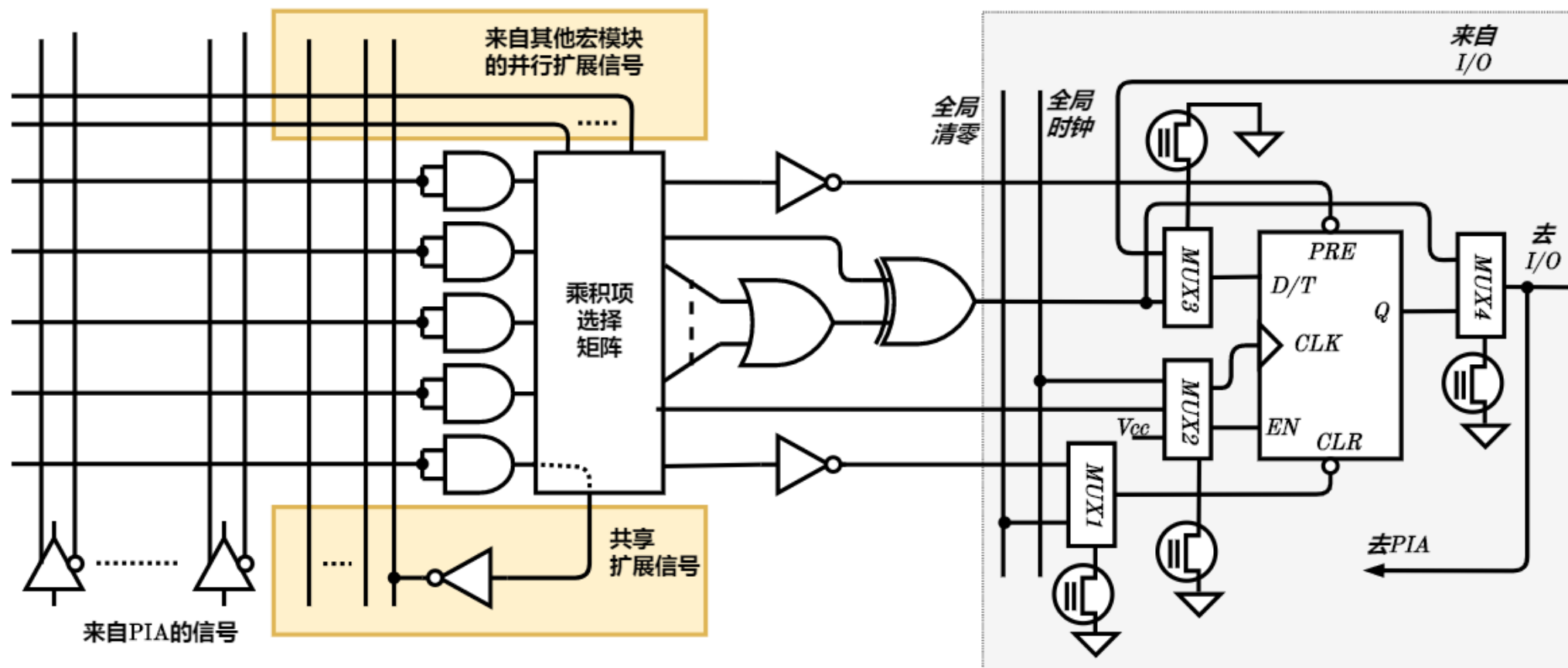
# Altera Max7000 CPLD宏单元

可编程逻辑“与”阵列



# 宏单元的工作模式

- 通过对数据选择器MUX的编程，实现不同的工作模式

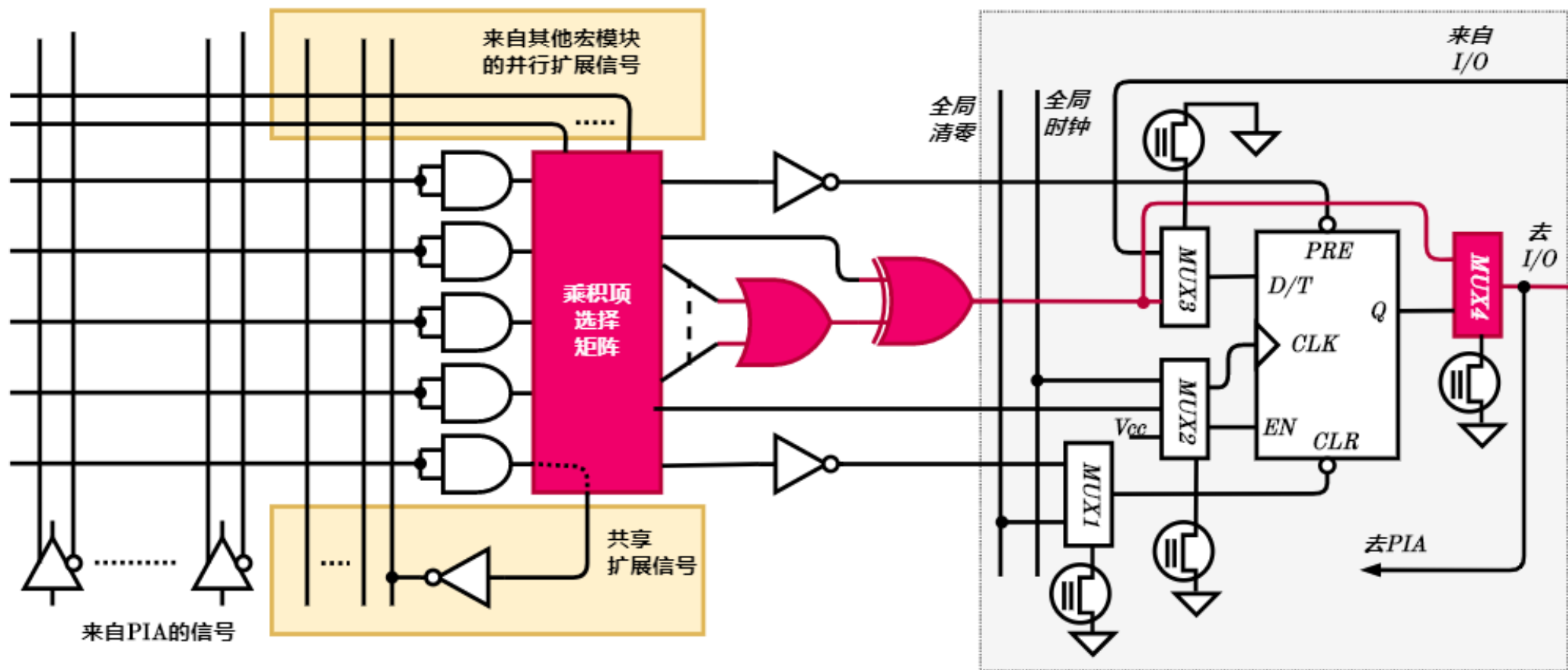


- MUX1清0选择；MUX2时钟和触发器使能选择
- MUX3触发器输入来源选择；MUX4输出旁路选择



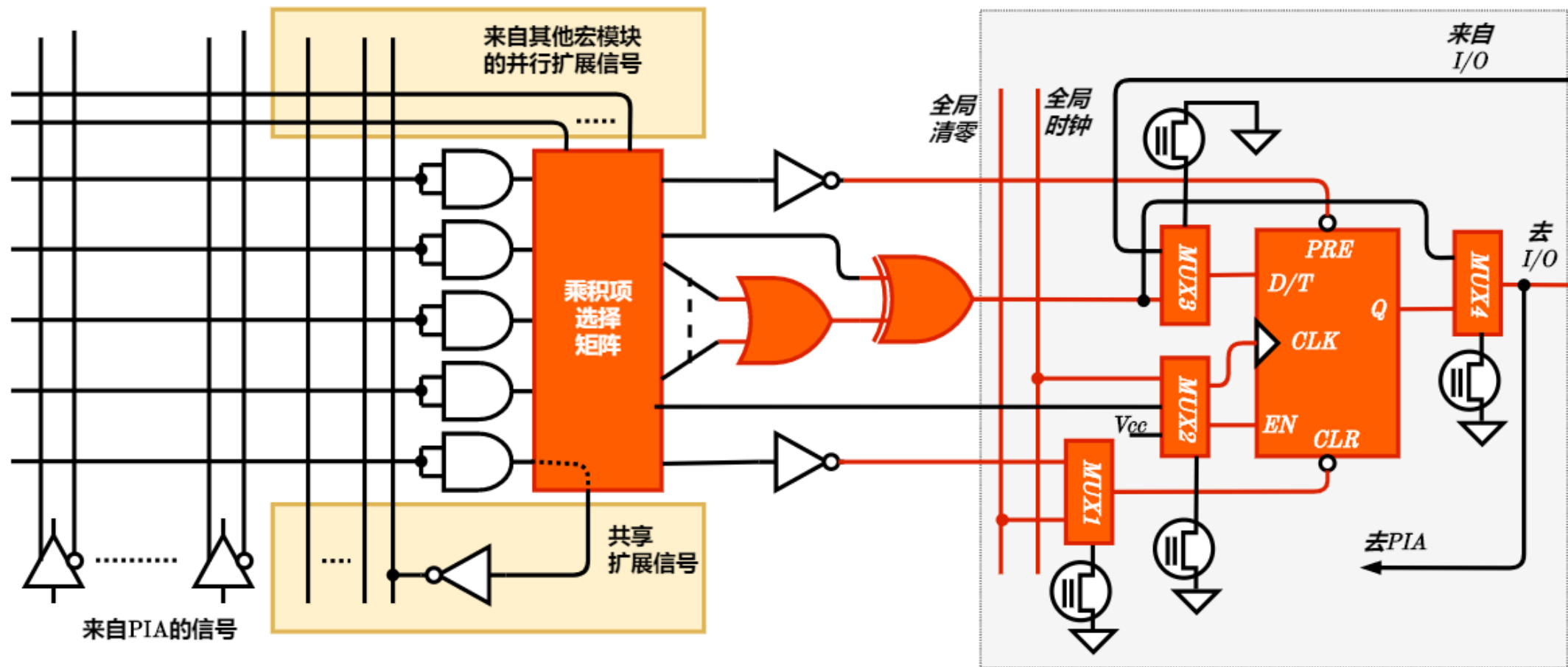
# 宏单元组合逻辑工作模式

- 组合逻辑模式跳过触发器，逻辑函数直接连到I/O引脚上



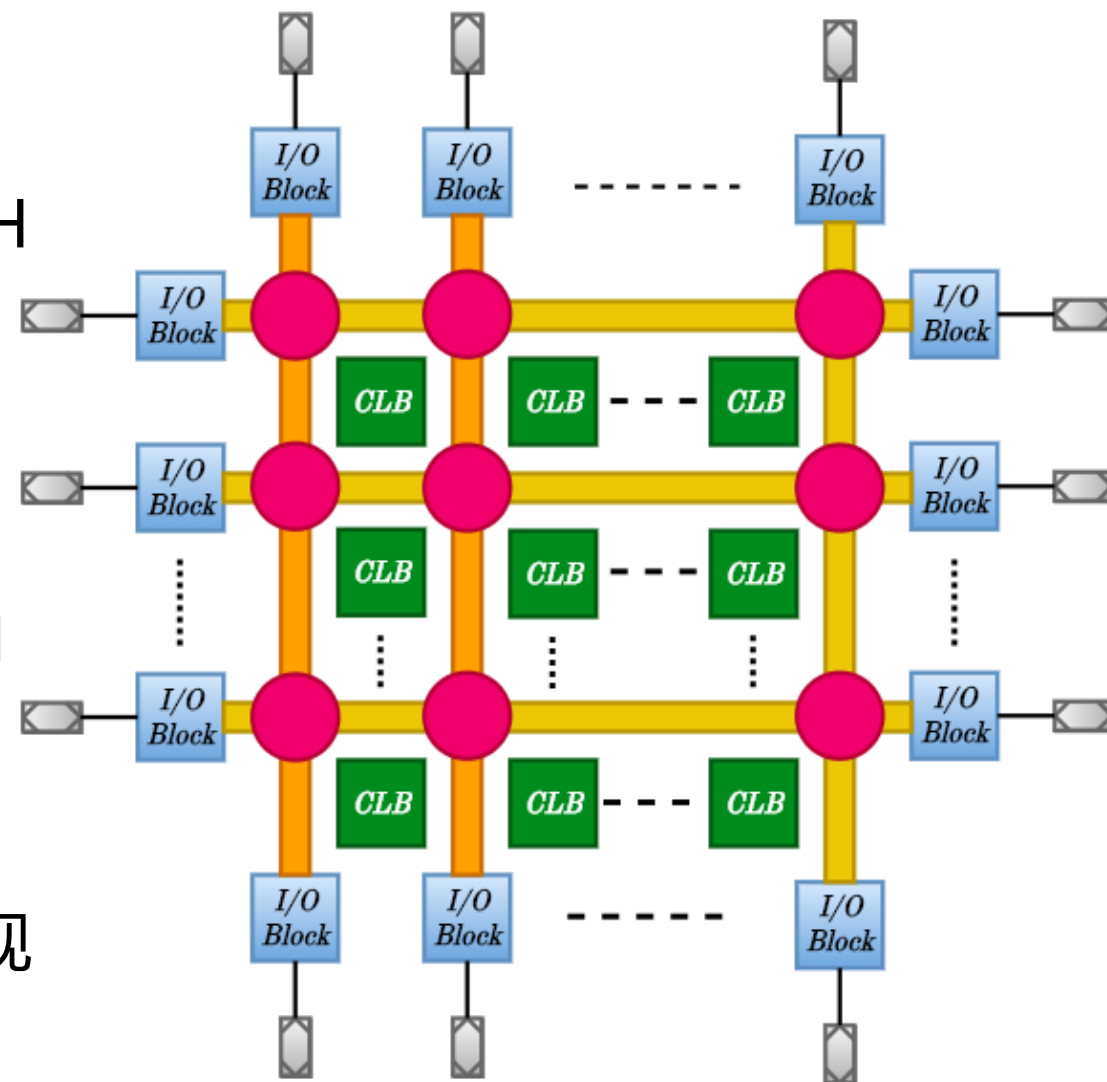
# 宏单元寄存器工作模式

- 寄存器模式下，联通触发器的数据和时钟信号，乘积项输出被锁存



# Field Programmable Gate Array (FPGA)

- FPGA采用与PAL/GAL不同的可编程逻辑结构，比CPLD有更高密度
- FPGA的可编程互连结构采用行列MESH
- FPGA包括三种基本模块
  - Configurable Logic Block (CLB)
    - 细粒度CLB，CLB功能简单
    - 粗粒度CLB，CLB具有更复杂的功能
  - 分布式MESH结构的互连网络，连接CLB和I/O块
  - I/O Block, 连接I/O和互连网络，内部也有触发器和可编程逻辑
- FPGA一般采用SRAM或反熔丝技术实现编程配置



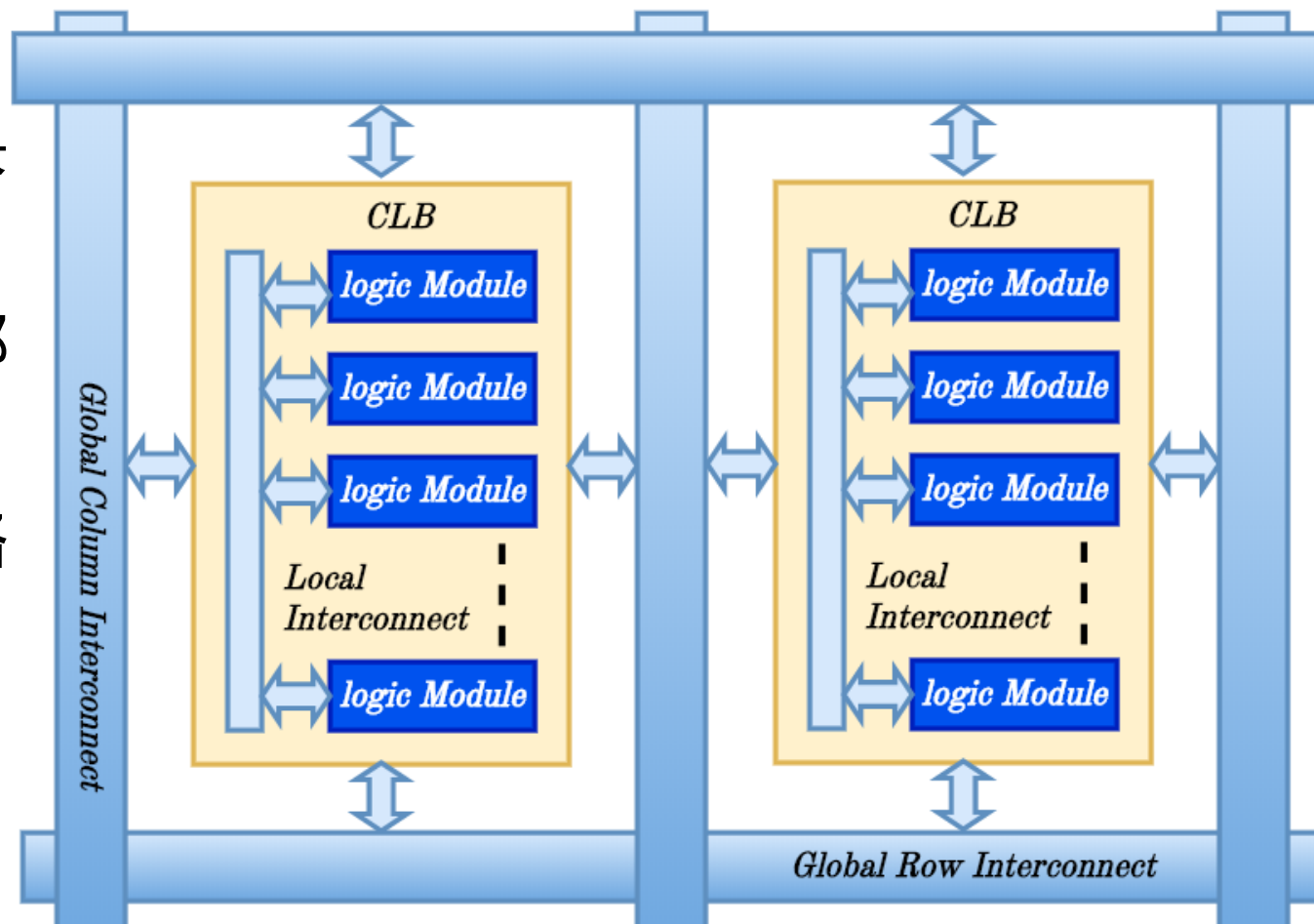
# FPGA资源容量举例 (Xilinx)

## Device Resources

	Kintex UltraScale FPGA	Kintex UltraScale+ FPGA	Virtex UltraScale FPGA	Virtex UltraScale+ FPGA	Zynq UltraScale+ MPSoC	Zynq UltraScale+ RFSoc
MPSoC Processing System					✓	✓
RF-ADC/DAC						✓
SD-FEC						✓
System Logic Cells (K)	318–1,451	356–1,143	783–5,541	862–3,780	103–1,143	678–930
Block Memory (Mb)	12.7–75.9	12.7–34.6	44.3–132.9	23.6–94.5	4.5–34.6	27.8–38.0
UltraRAM (Mb)		0–36		90–360	0–36	13.5–22.5
HBM DRAM (GB)				0–8		
DSP (Slices)	768–5,520	1,368–3,528	600–2,880	2,280–12,288	240–3,528	3,145–4,272
DSP Performance (GMAC/s)	8,180	6,287	4,268	21,897	6,287	7,613
Transceivers	12–64	16–76	36–120	32–128	0–72	8–16
Max. Transceiver Speed (Gb/s)	16.3	32.75	30.5	58.0	32.75	32.75
Max. Serial Bandwidth (full duplex) (Gb/s)	2,086	3,268	5,616	8,384	3,268	1,048
Memory Interface Performance (Mb/s)	2,400	2,666	2,400	2,666	2,666	2,666
I/O Pins	312–832	280–668	338–1,456	208–832	82–668	280–408

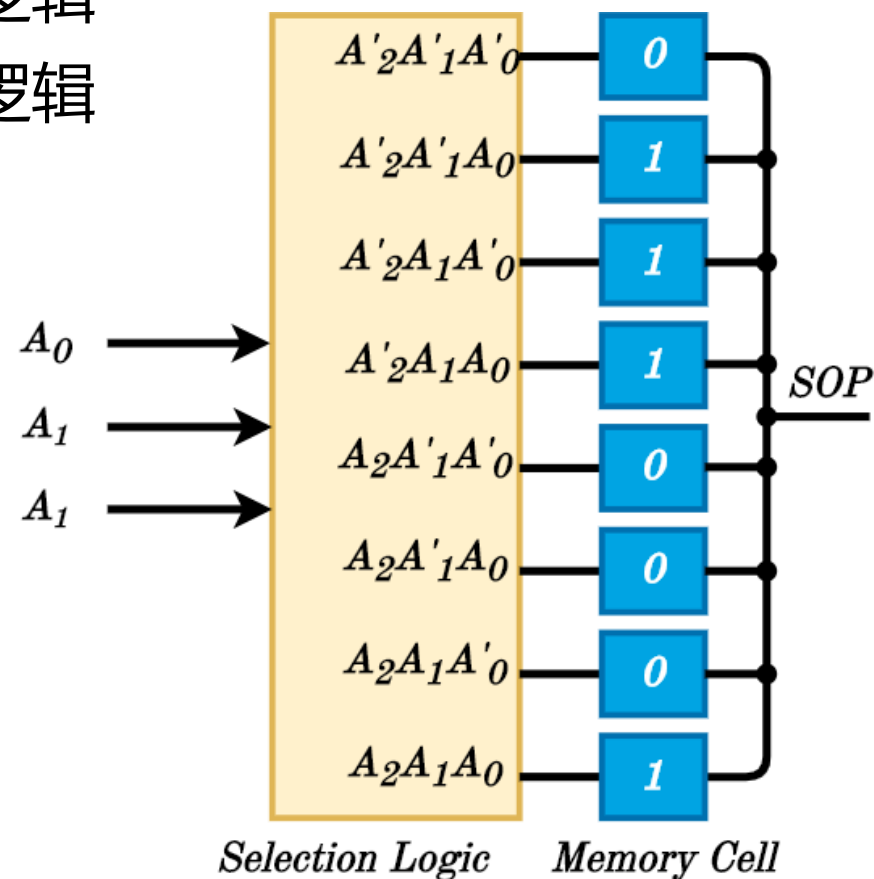
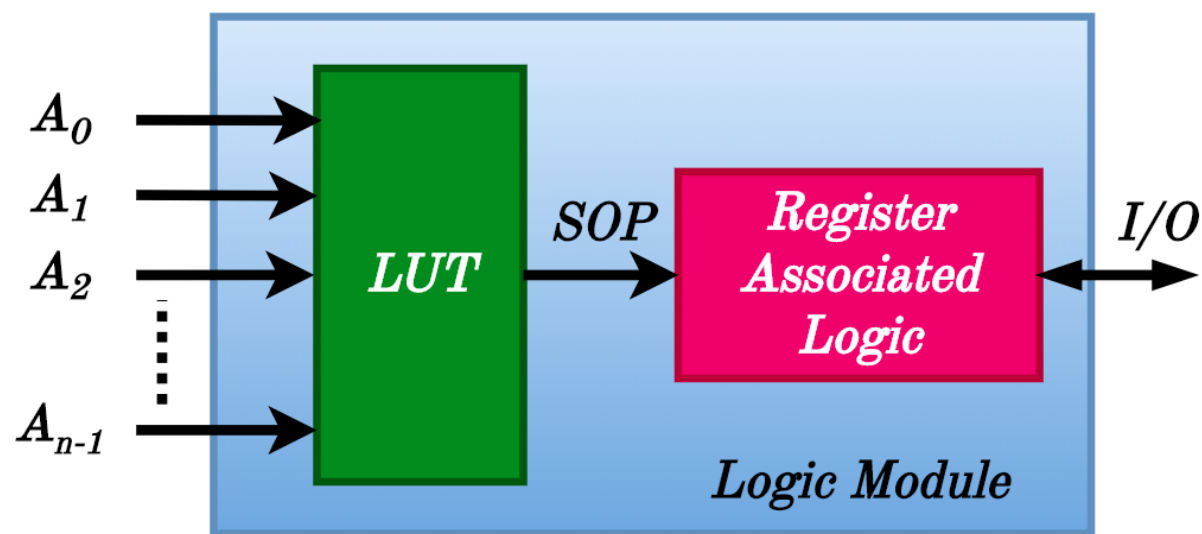
# FPGA的CLB结构

- 每个CLB包括多个小的逻辑模块，类似于CPLD的宏单元
- CLB内部逻辑模块之间采用局部互连总线连接
- CLB再通过行、列全局互连网络连接其他CLB、IOB、BRAM、DSP、CPU Core等全局资源



# Logic Module结构

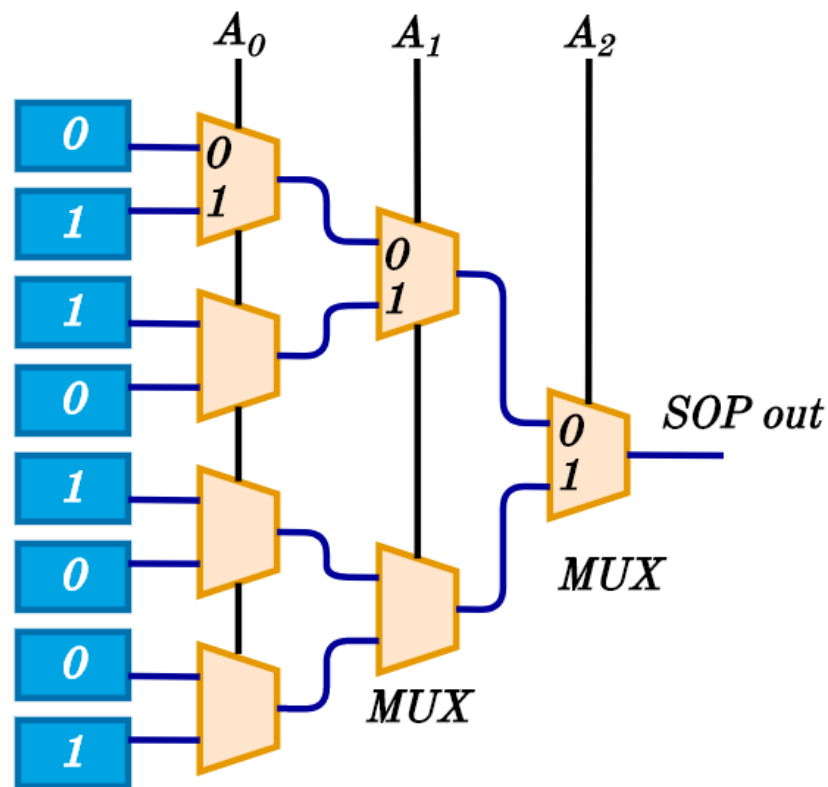
- 逻辑模块包括一个LUT (Look Up Table) ,和一个关联的逻辑电路
- 关联电路内包含有触发器, 及其相关控制逻辑
- 逻辑模块可以配置成组合逻辑, 或者时序逻辑
- LUT可以实现类似PAL/GAL的逻辑功能



# 查找表 (LUT)

- LUT由RAM单元和数据选择器实现
- 对于n输入LUT，可以实现n变量的任意组合逻辑函数
- LUT可以工作在以下模式
  - 普通LUT模式（逻辑函数），单LUT实现组合逻辑
  - 扩展LUT模式，多个LUT，增加变量数
- 写出SOP out的逻辑函数：

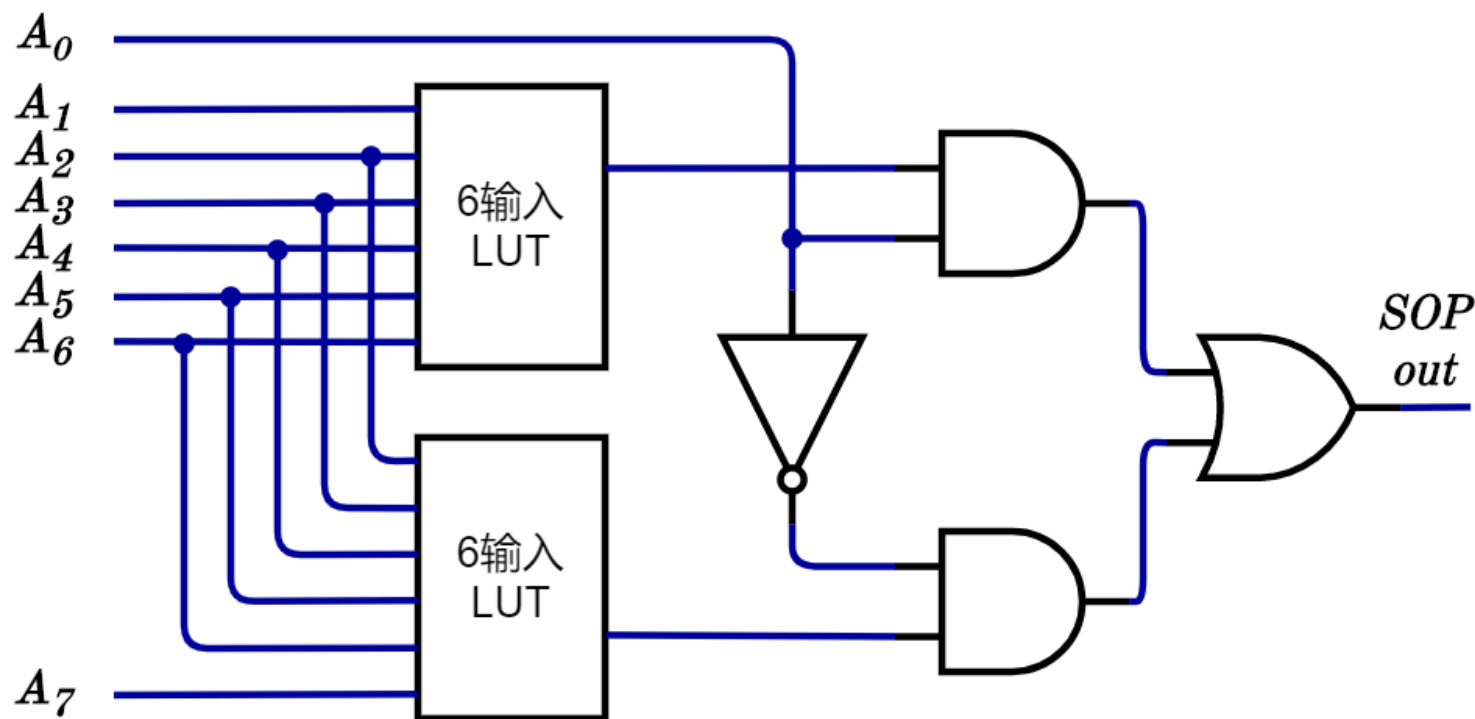
$$SOPout = A'_2 A'_1 A_0 + A'_2 A_1 A'_0 + A_2 A'_1 A'_0 + A_2 A_1 A_0$$





# LUT扩展工作模式

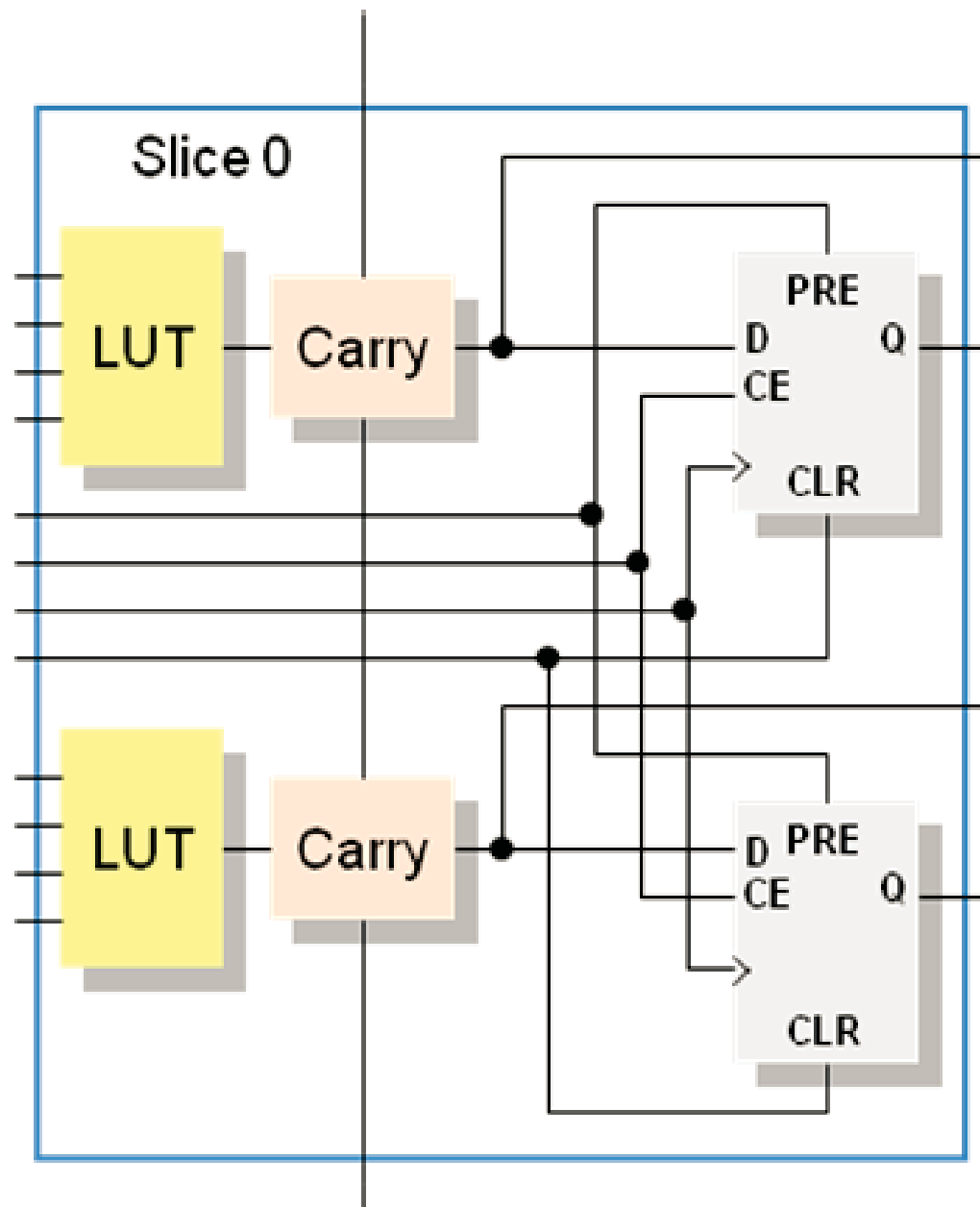
- 多个LUT扩展，可以增加逻辑函数的变量数，需要LM中的附加可编程逻辑实现所需的MUX
- 如图实用2个6输入LUT实现8变量逻辑函数
- 该电路是否可以实现8变量的所有函数？



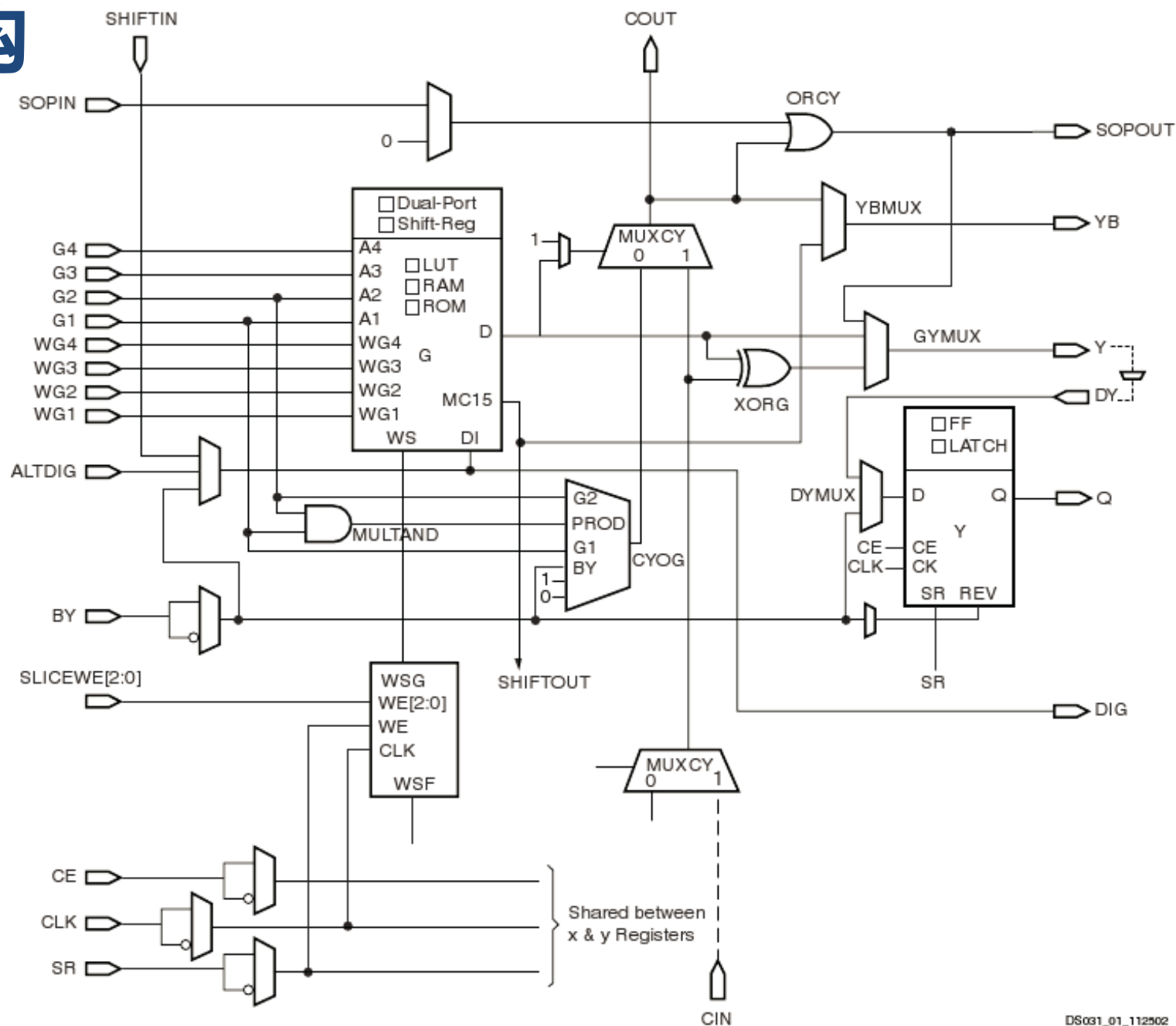


# FPGA CLB举例

- Xilinx Spartan 3 CLB包括：
  - 2个LUT
  - 2个触发器
  - 4个输出
    - 2个组合逻辑输出
    - 2个寄存器输出
  - 对触发器的控制输入
  - I/O进位链

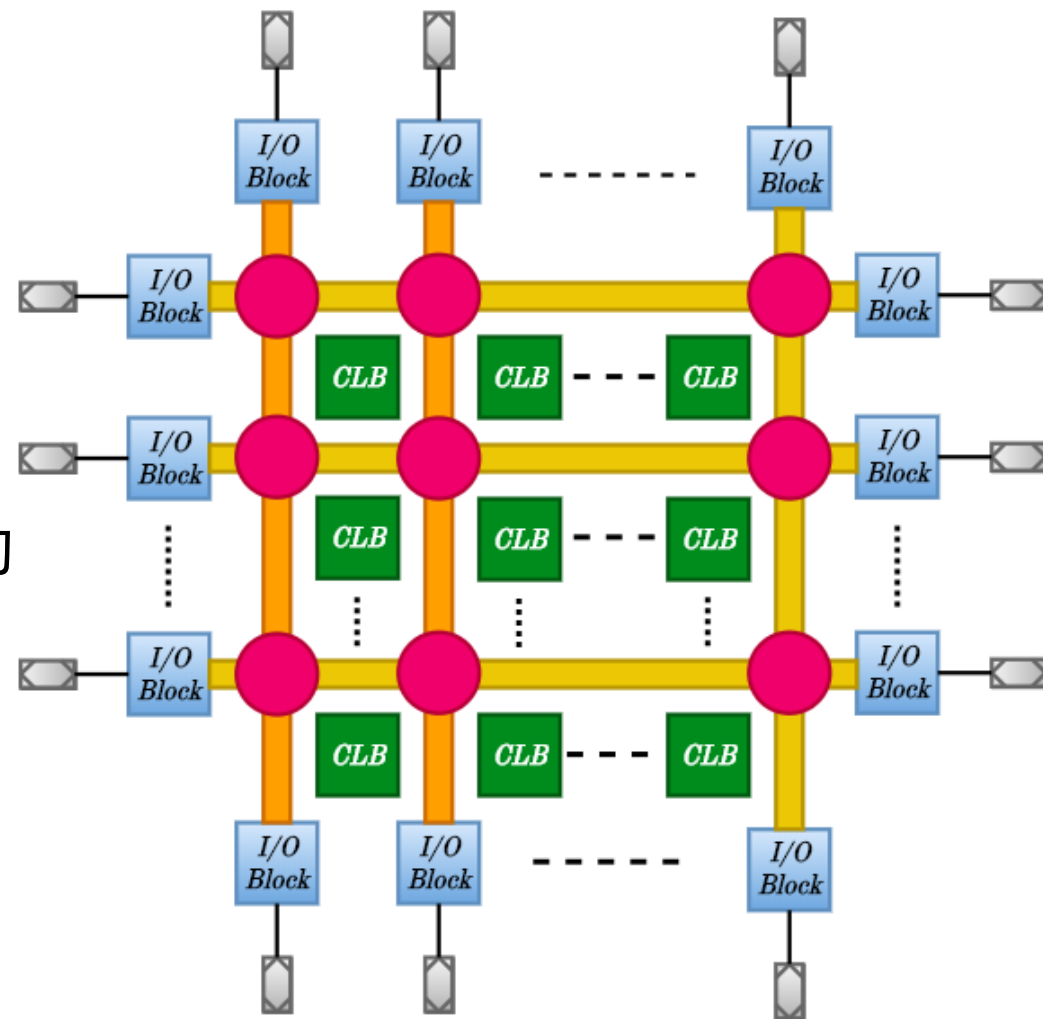
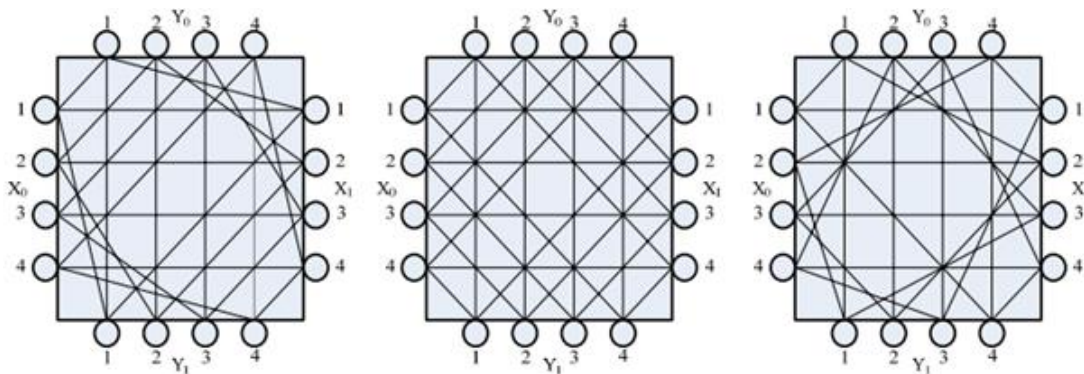


# 1/2 CLB详细结构

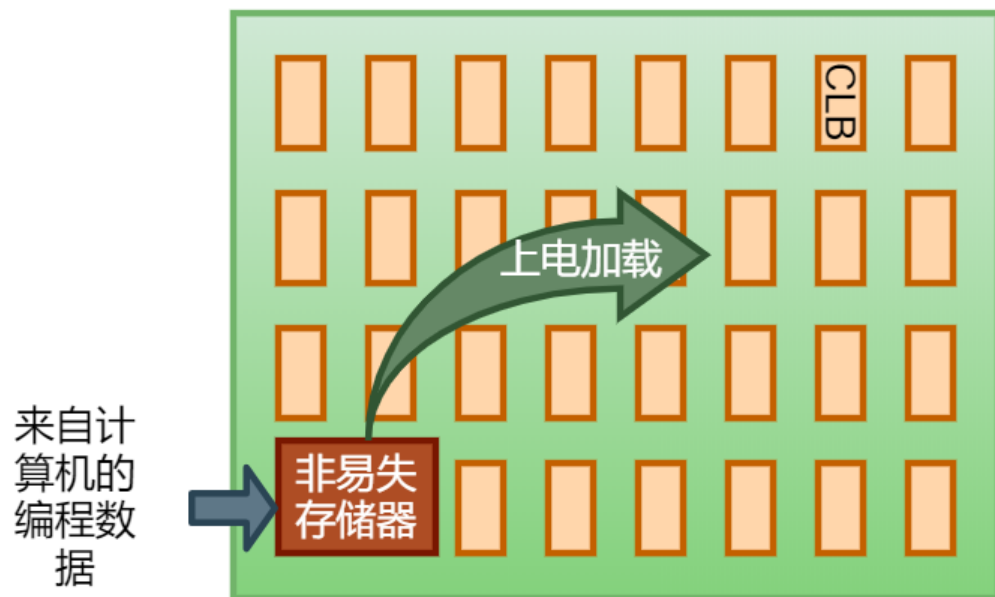


# 可编程互连结构 (PIA)

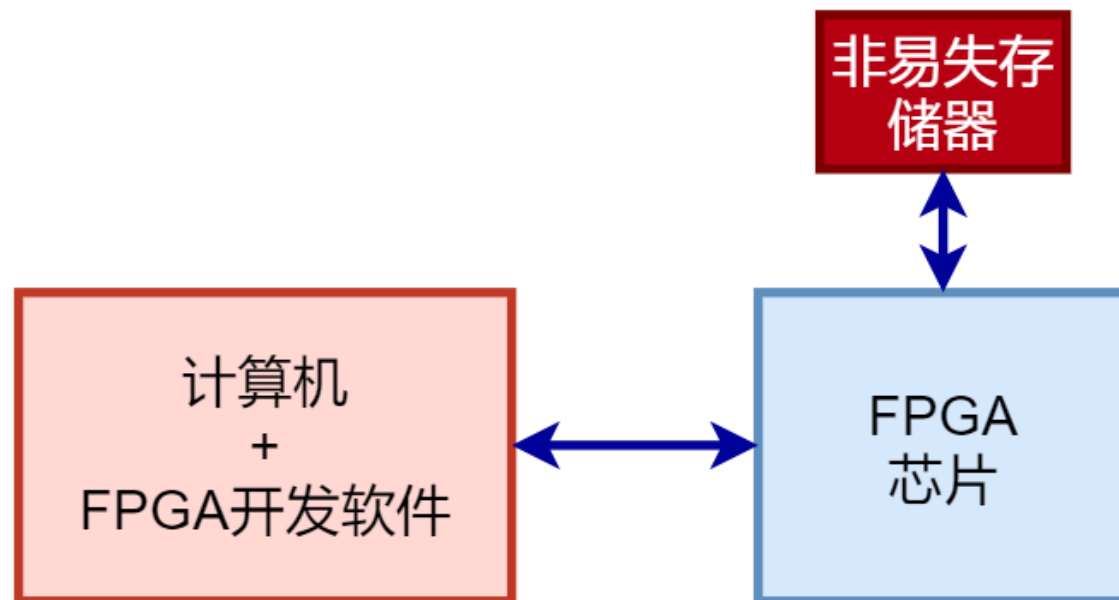
- 金属连线 (经可编程的连接点与CLB、IOB的开关矩阵相连)
  - 通用连线用于CLB之间的连接
  - 长线用于长距离或多分支信号的传送
  - 全局连线用于输送一些公共信号
- 开关矩阵SM
  - 通过对开关矩阵编程, 可以将来自任何方向上的一根导线转接至其他方向的某一根导线上



# FPGA编程数据存储

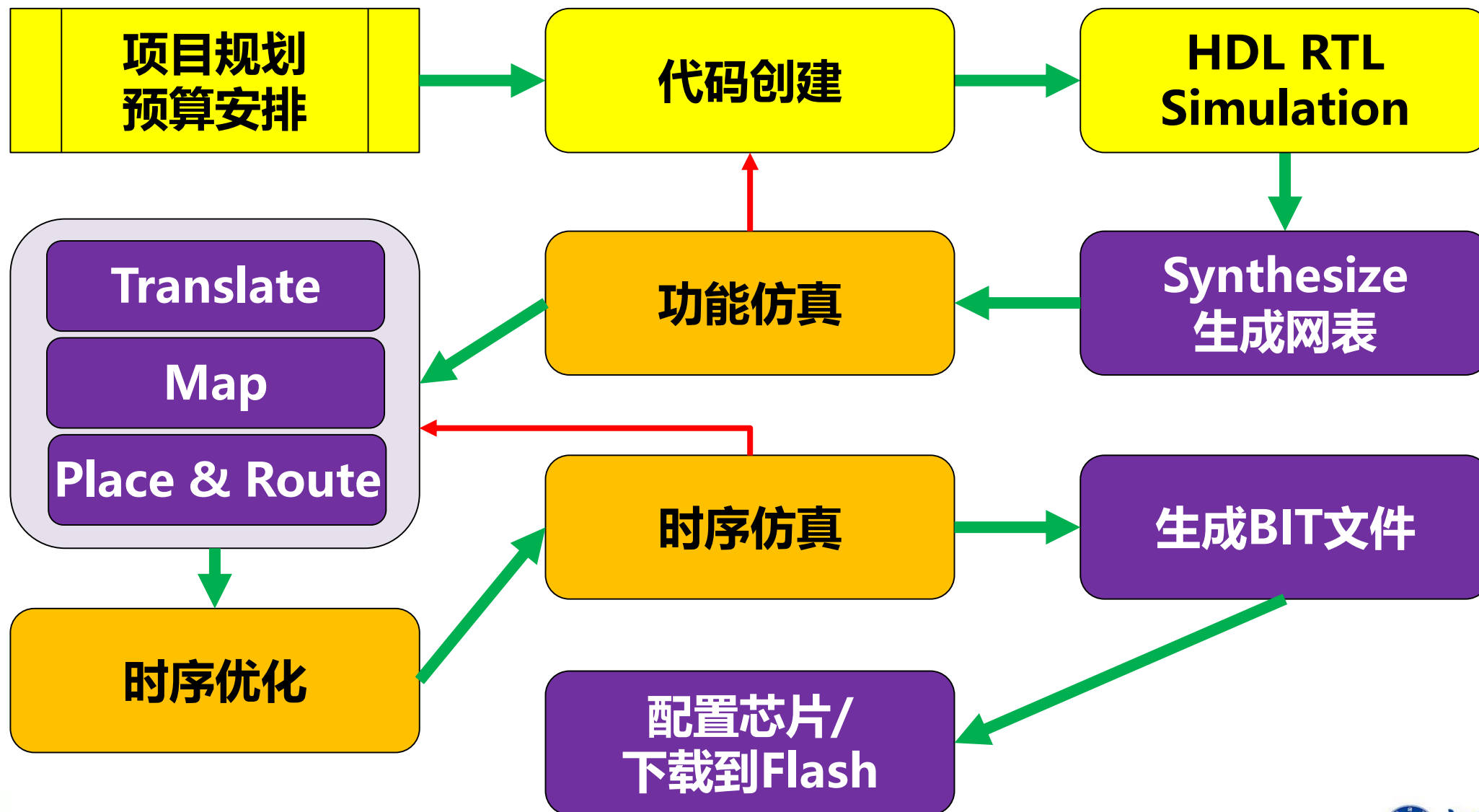


- FPGA内建非易失存储器，存储编程数据



- FPGA外接Flash芯片，用于存储编程数据

# 基于可编程器件系统开发流程



# 问题和建议?

