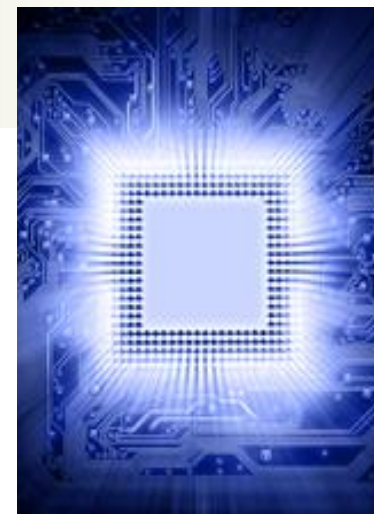




南京大学计算机科学与技术系
Department of Computer Science and Technology



第2讲 可编程逻辑器件



南京大学计算机学院



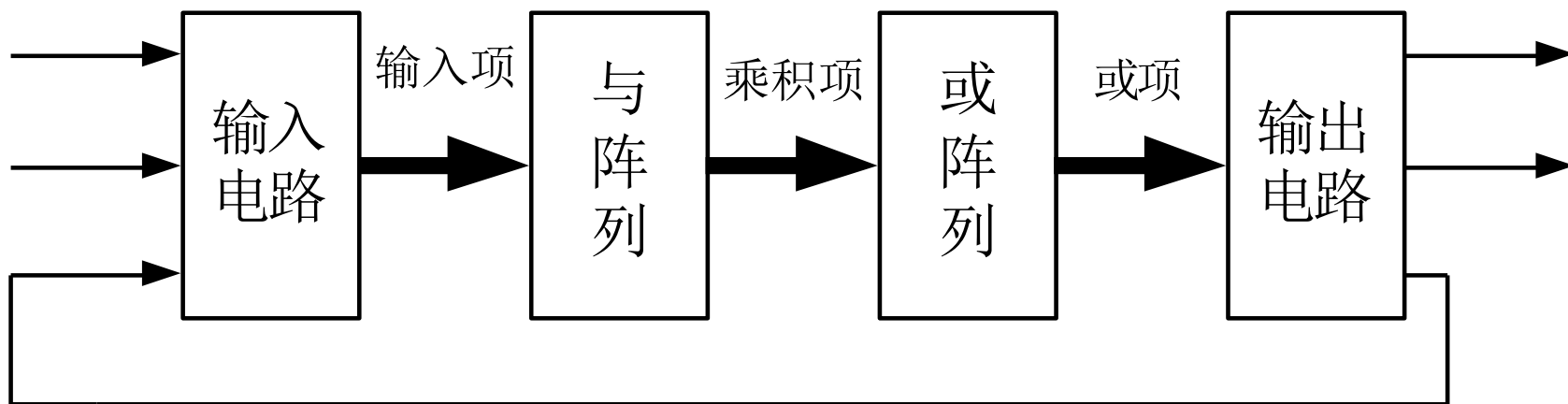
主要内容

- 可编程器件PLD
- 复杂可编程器件CPLD
 - CPLD
 - FPGA
- 数字系统设计



数字电路的基本架构

- 任何组合逻辑函数都可表示为输入信号的**最小项之和**或者**最大项之积**的形式。
- 组合电路有两种基本架构：
 - **与或电路**：积之和POS，先“**与**”后“**或**”
 - **或与电路**：和之积SOP，先“**或**”后“**与**”
- 时序逻辑电路包含记忆单元：**存储器件/反馈结构**，其记忆信号通过组合逻辑输出到激励信号。



可编程逻辑器件是指一切通过软件手段更改、配置器件内部连接结构和逻辑单元，完成既定设计功能的数字集成电路。



可编程逻辑器件

- Programmable Logic Devices, PLD的发展历程
 - 第一阶段：
 - ◆ 可编程只读存储器PROM：与阵列固定，或阵列可编程；
 - ◆ 可编程逻辑阵列PLA：与阵列可编程，或阵列可编程；
 - 第二阶段：
 - ◆ 可编程阵列逻辑PAL：与阵列可编程，或阵列固定。具有多种输出和反馈结构，一次编程使用。
 - 第三阶段：
 - ◆ 通用阵列逻辑GAL：与阵列可编程，或阵列固定。采用高速电擦除CMOS工艺、多次反复使用。
 - 第四阶段：
 - ◆ 复杂可编程逻辑器件CPLD：包括CPLD、FPGA、ISP、SOC。

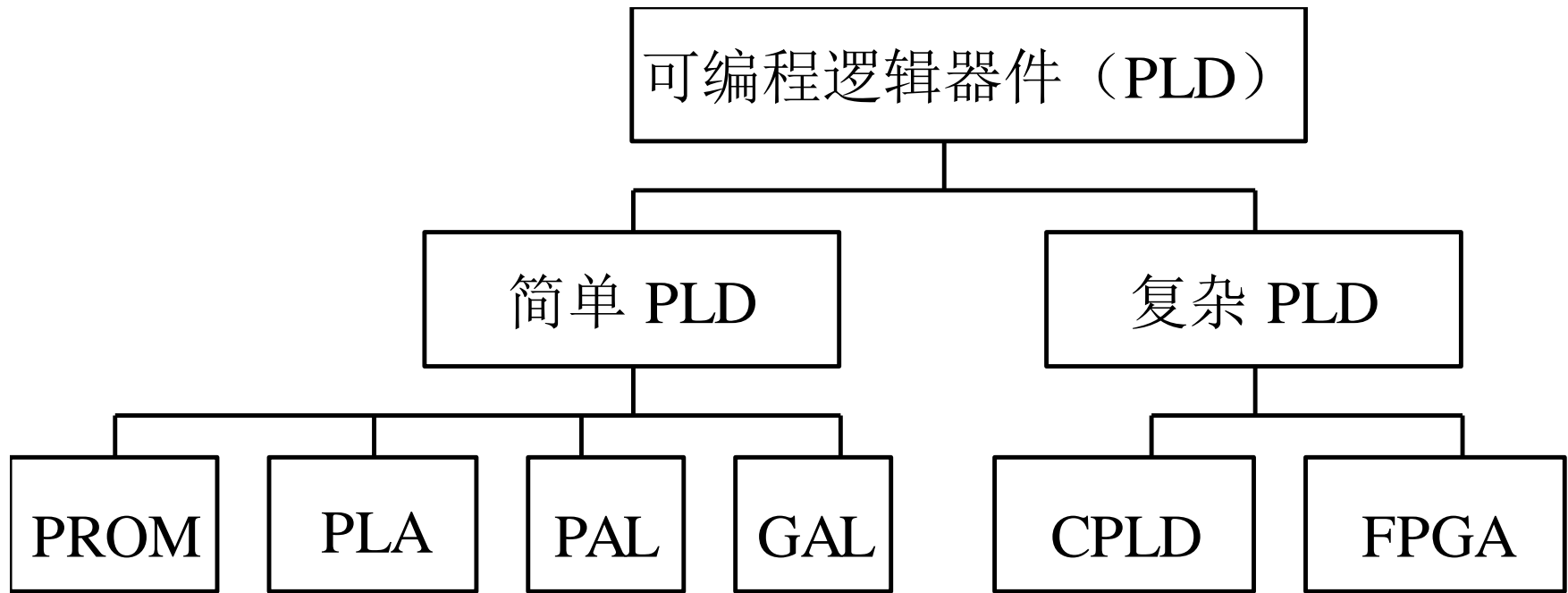


可编程逻辑器件制造工艺

- 熔丝或反熔丝编程器件 - - Actel的FPGA器件
 - 体积小，集成度高，速度高，易加密，抗干扰，耐高温
 - 只能编程一次，使用不灵活。
- SRAM - - 大多数的FPGA器件
 - 可反复编程，实现系统功能的动态重构
 - 每次上电需重新下载，实际应用时可外挂E²PROM用于保存程序
- E²PROM、Flash - - 大多数的CPLD器件
 - 可反复编程
 - 不用每次上电重新下载，速度慢，功耗较大
 - 上电后编程信息自动加载到芯片内部SRAM，才能工作

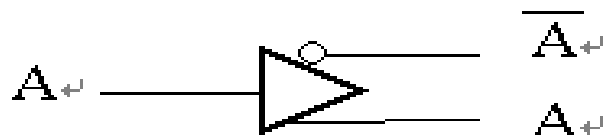


可编程逻辑器件按集成度分类

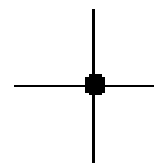




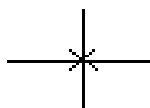
可编程逻辑器件连线表示



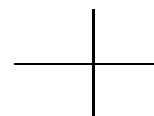
缓冲器



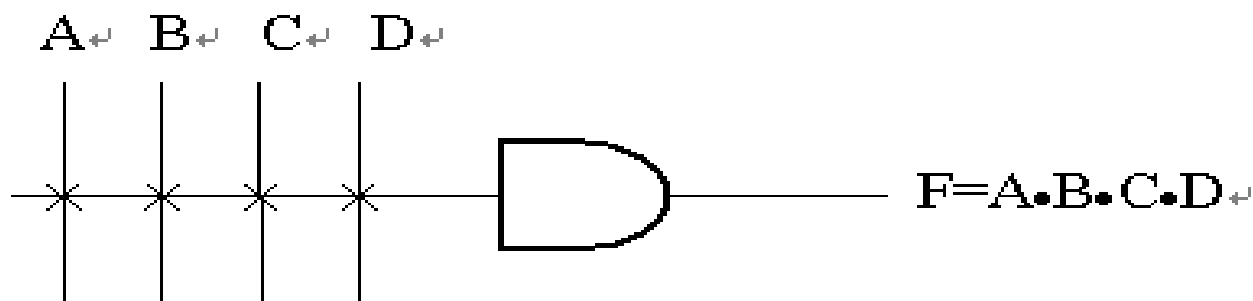
硬线连接(不可编程)



被编程单元



被擦除(断开)单元



“与”阵列
(可编程)

PLD逻辑符号绘图约定

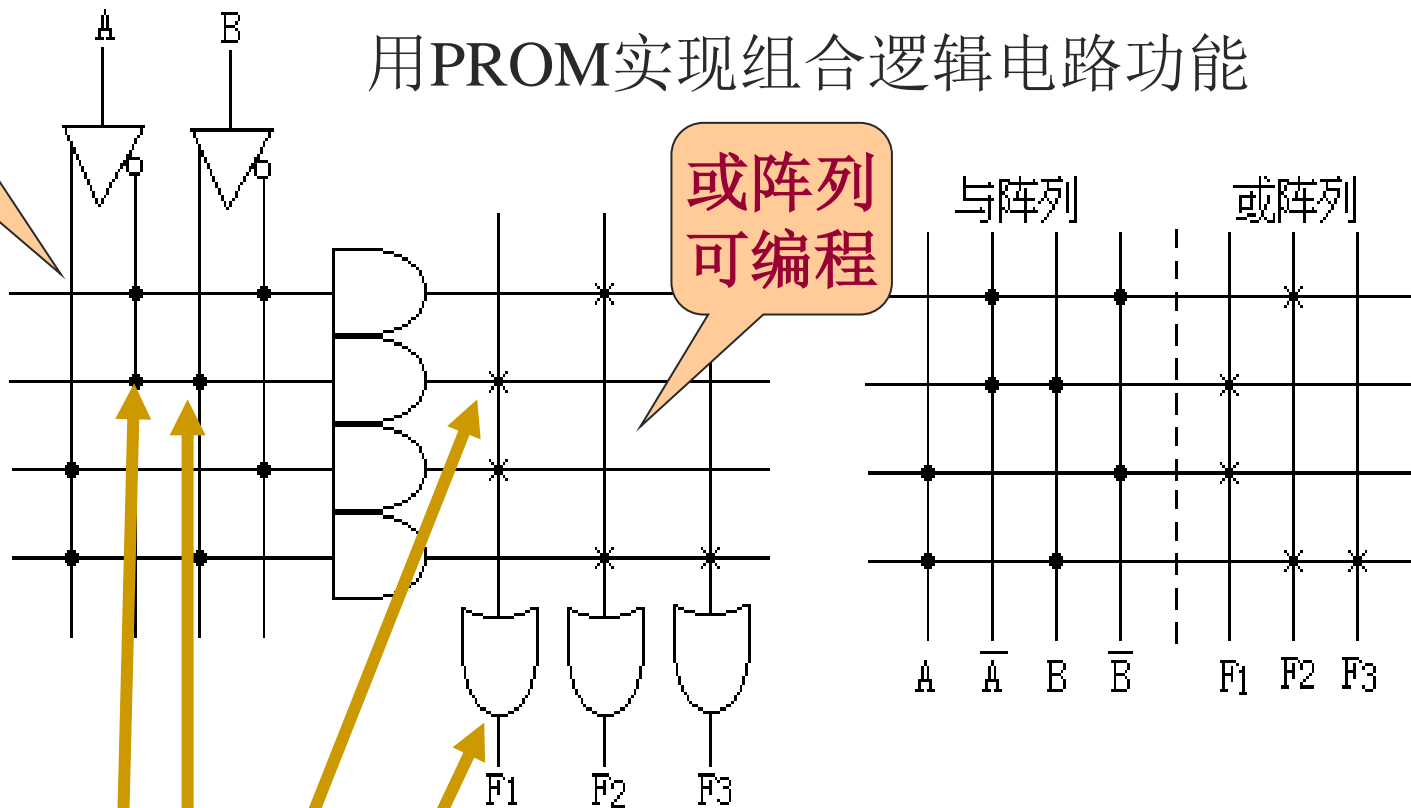


可编程逻辑器件EPROM的应用

用PROM实现组合逻辑电路功能

与阵列
固定

或阵列
可编程



实现的函数为:

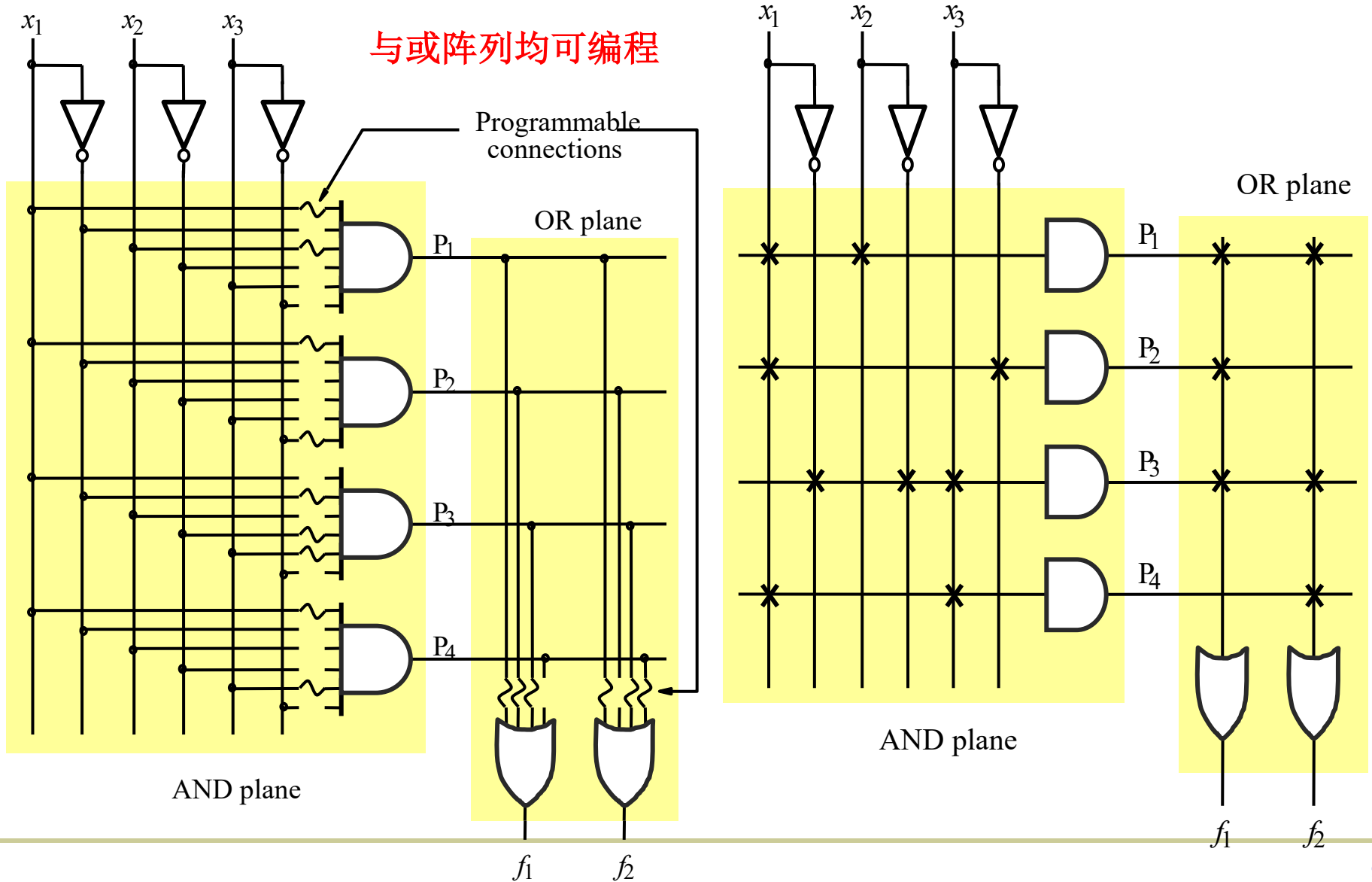
$$F_1 = \bar{A} \bullet B + A \bullet \bar{B}$$

$$F_2 = \bar{A} \bullet \bar{B} + A \bullet B$$

$$F_3 = A \bullet B$$



可编程逻辑器件PLA的应用



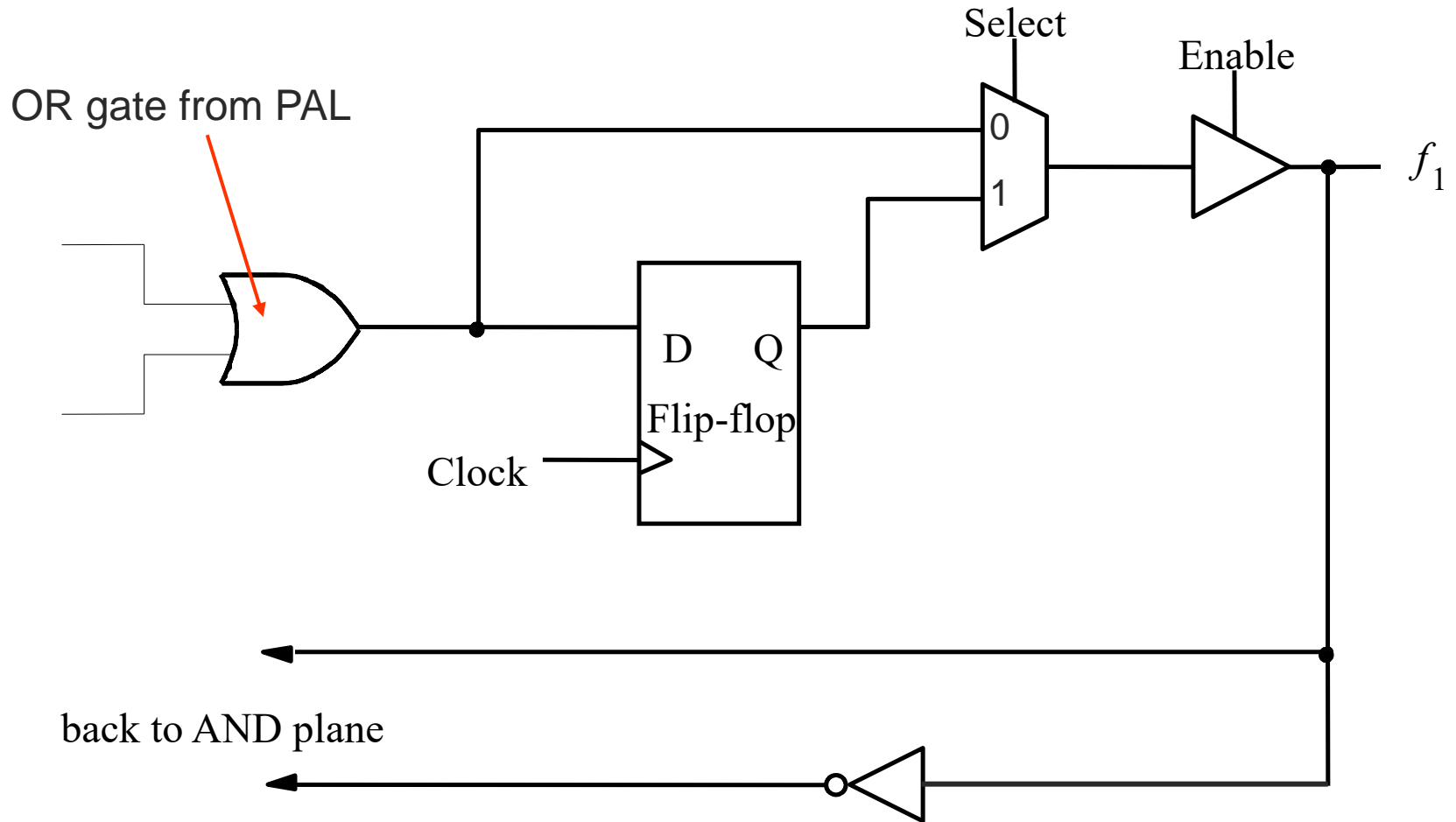


可编程逻辑器件PAL

- PAL的逻辑结构：由一个**可编程**的“**与**”阵列和一个**固定连接**的“**或**”阵列组成。
- PAL每个输出包含的“**与**”项数目是由固定连接的“**或**”阵列提供的。
- 按其输出和**反馈**结构，大致可将其分为**5种**基本类型（专用输出的基本门阵列结构、带反馈的可编程I/O结构、带反馈的寄存器输出结构、加“异或”、带反馈的寄存器输出结构和算术选通反馈结构）。

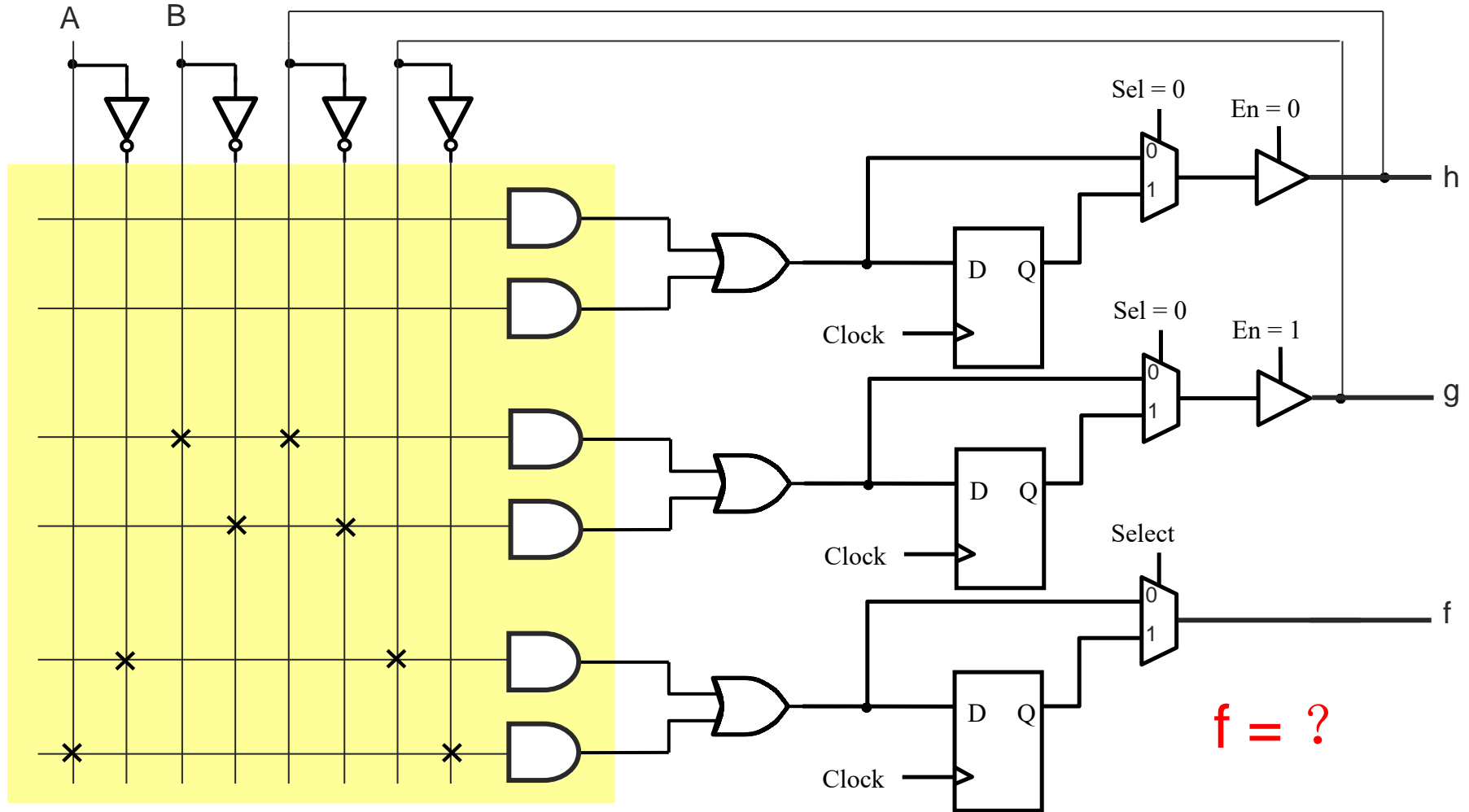


PAL输出宏单元





PAL的应用



改变sel,en的值呢?

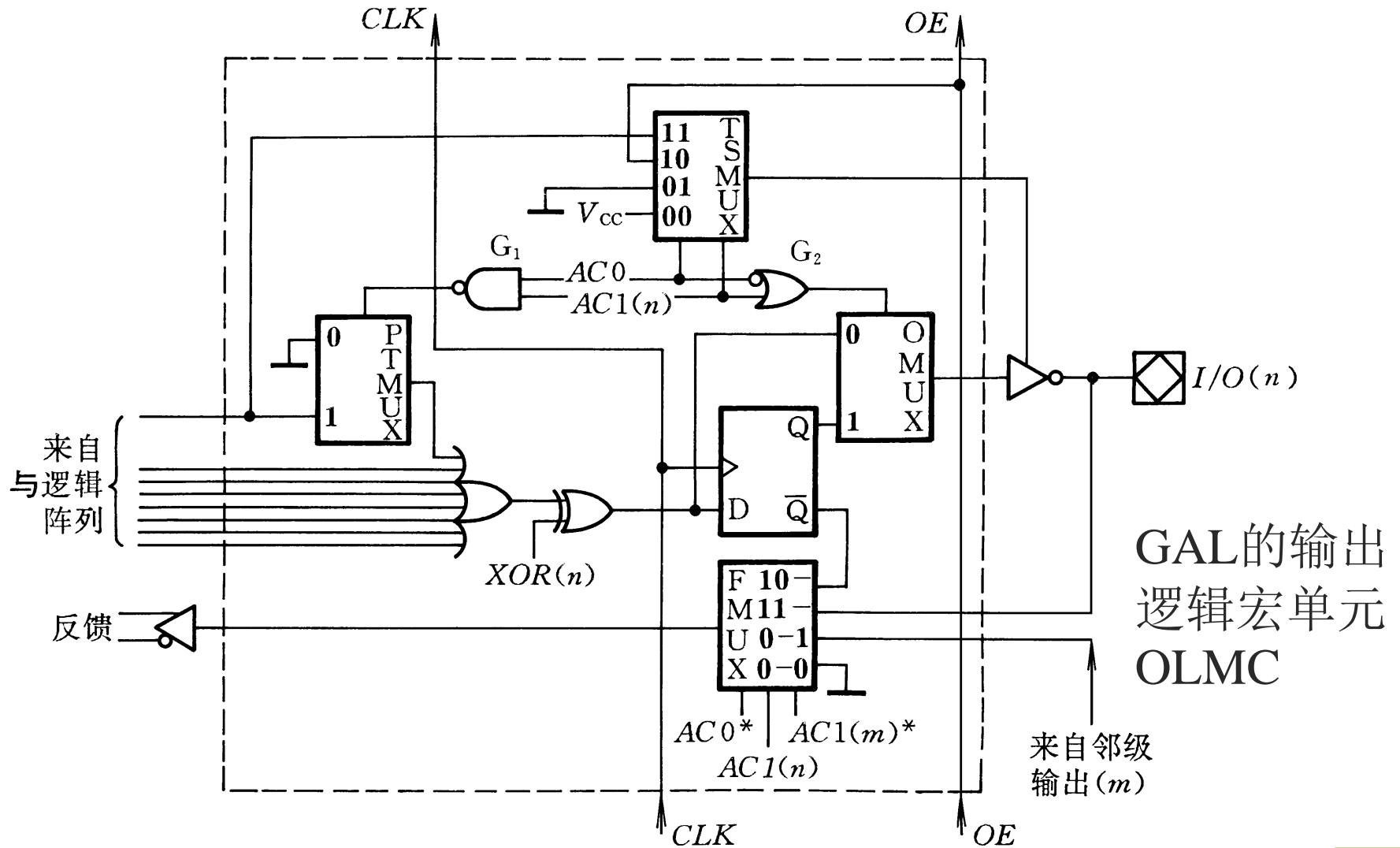


通用阵列逻辑GAL

- GAL (Generic Array Logic) 器件是1985年由LATTICE公司开发并商品化的一种新的PLD器件。它是在PAL器件的基础上综合了E²PROM和CMOS技术发展起来的一种新型技术。
- PAL器件采用的是**双极型熔丝工艺**，一旦编程不能修改，且输出结构的类型繁多，给设计带来不便。
- GAL通用逻辑阵列，采用**电可擦除的CMOS**制作，输出端设置了可编程的**输出逻辑宏单元OLMC**，通过编程可设置不同的输出状态，增强器件的通用性。



通用阵列逻辑GAL



GAL的输出
逻辑宏单元
OLMC



PLD器件的优点

- 集成度高，可以替代通用IC芯片
 - 极大减小电路的面积，降低功耗，提高可靠性
- 具有完善先进的**开发工具**
 - 提供语言、图形等设计方法，十分灵活
 - 通过仿真工具来验证设计的正确性
- 可以**反复擦除、编程**，方便设计的修改和升级
- 灵活地定义管脚功能，减轻设计工作量，缩短系统开发时间
- 保密性好

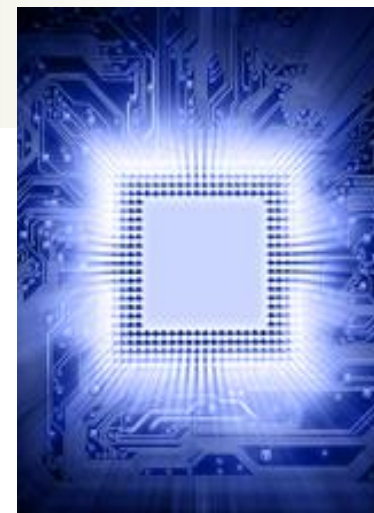


南京大学计算机科学与技术系
Department of Computer Science and Technology, Nanjing University



CPLD

CPLD
FPGA



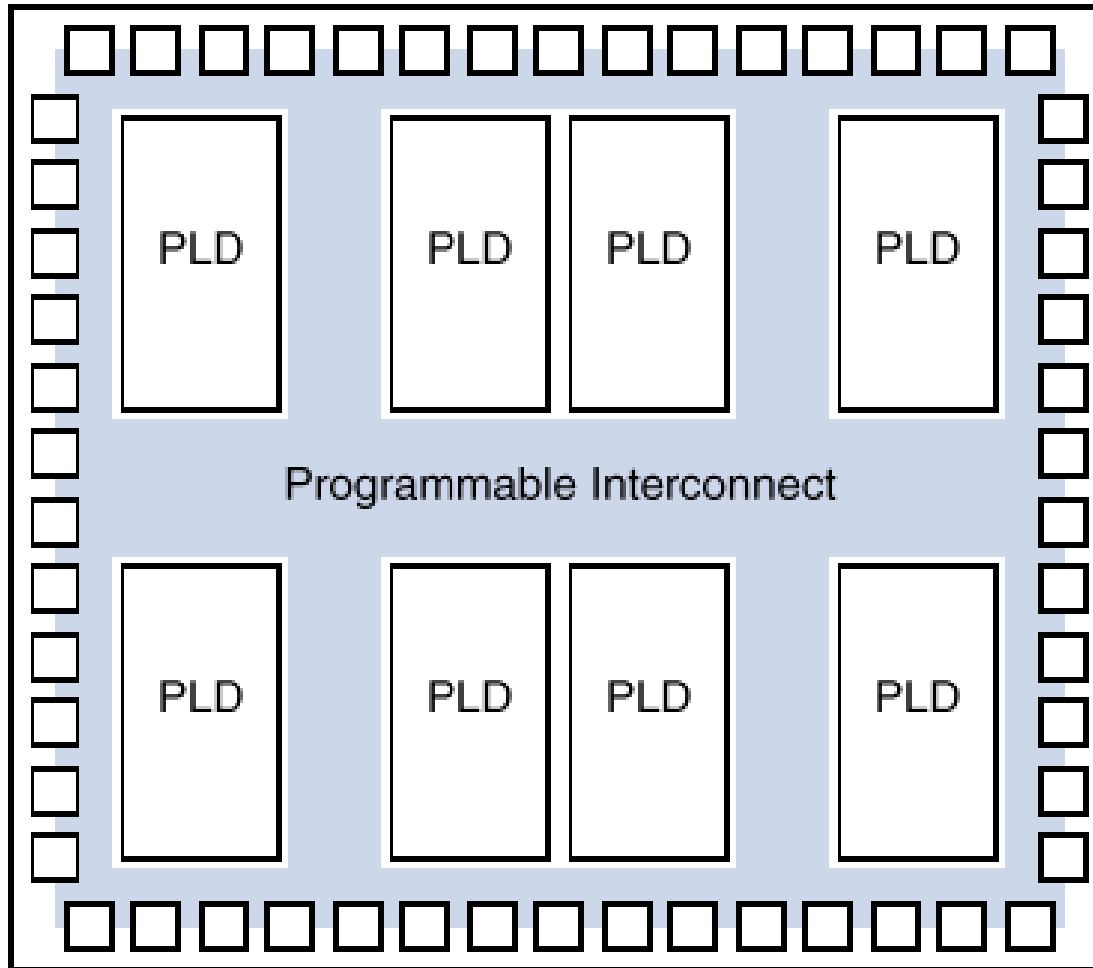


复杂可编程逻辑器件

- 为什么没有对PAL、GAL的体系结构加以扩充？
 - 增加 n 倍输入端，由于电容效应和漏电流等因素，与阵列的操作速度会下降 n 倍。
 - 芯片面积需增加 n^2 倍。
 - 不如使用 n 个单独的PLD。
- 在PAL、GAL基础上发展起来的阵列型PLD。
- 用EPROM、E²PROM、Flash Memory和SRAM等编程技术，构成了高密度、高速度和低功耗的CPLD。
- 基本结构：由内部PLD、输入输出模块和可编程内部连线三个部分组成。
- CPLD集成度远远高于PAL和GAL。



通用的CPLD体系结构



□ = input/output block

片内PLD间的互连
方式可编程。



Xilinx XC9500 CPLD 系列

- 内部PLD称为（可重配）**功能块FB**(Configurable Functional Blocks)
- 每个内部PLD具有36个输入和18个宏单元（36V18）

Table 10-8 Function blocks and external I/O pins in Xilinx 9500-series CPLDs.

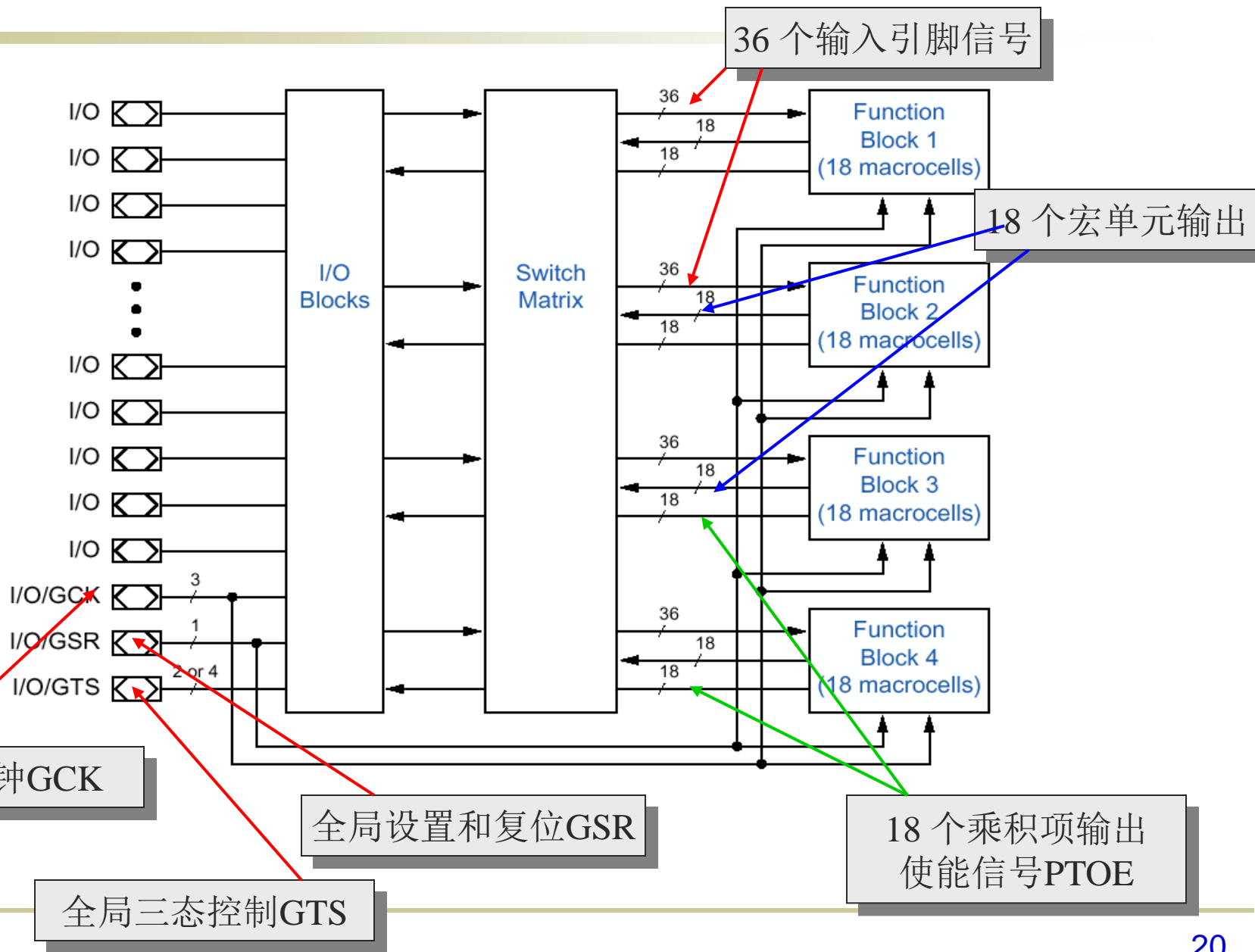
	<i>Part Number</i>					
	<i>XC9536</i>	<i>XC9572</i>	<i>XC95108</i>	<i>XC95144</i>	<i>XC95216</i>	<i>XC95288</i>
FBs / macrocells	2 / 36	4 / 72	6 / 108	8 / 144	12 / 216	16 / 288
<i>Package</i>	<i>Device I/O Pins</i>					
44-pin VQFP	34					
44-pin PLCC	34	34				
48-pin CSP	34					
84-pin PLCC		69	69			
100-pin TQFP		72	81	81		
100-pin PQFP		72	81	81		
160-pin PQFP			108	133	133	
208-pin HQFP					166	168
352-pin BGA					166	192

给定的芯片有不同的封装。

相同的封装有不同的器件。

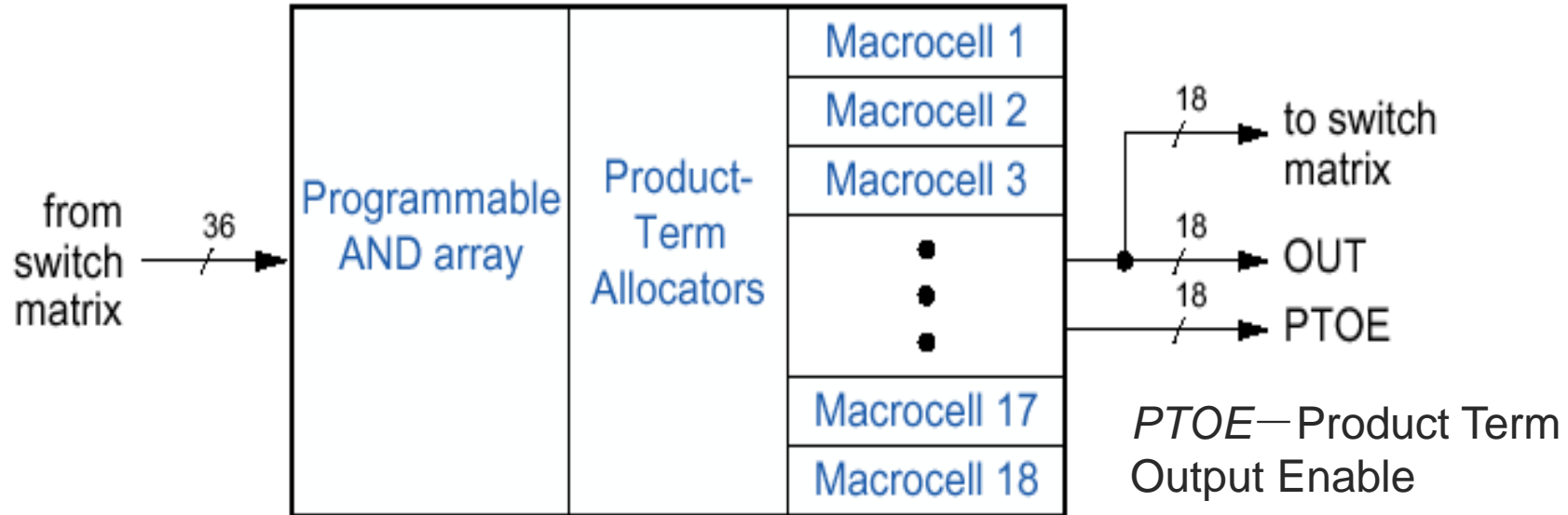


XC9500系列内部结构图

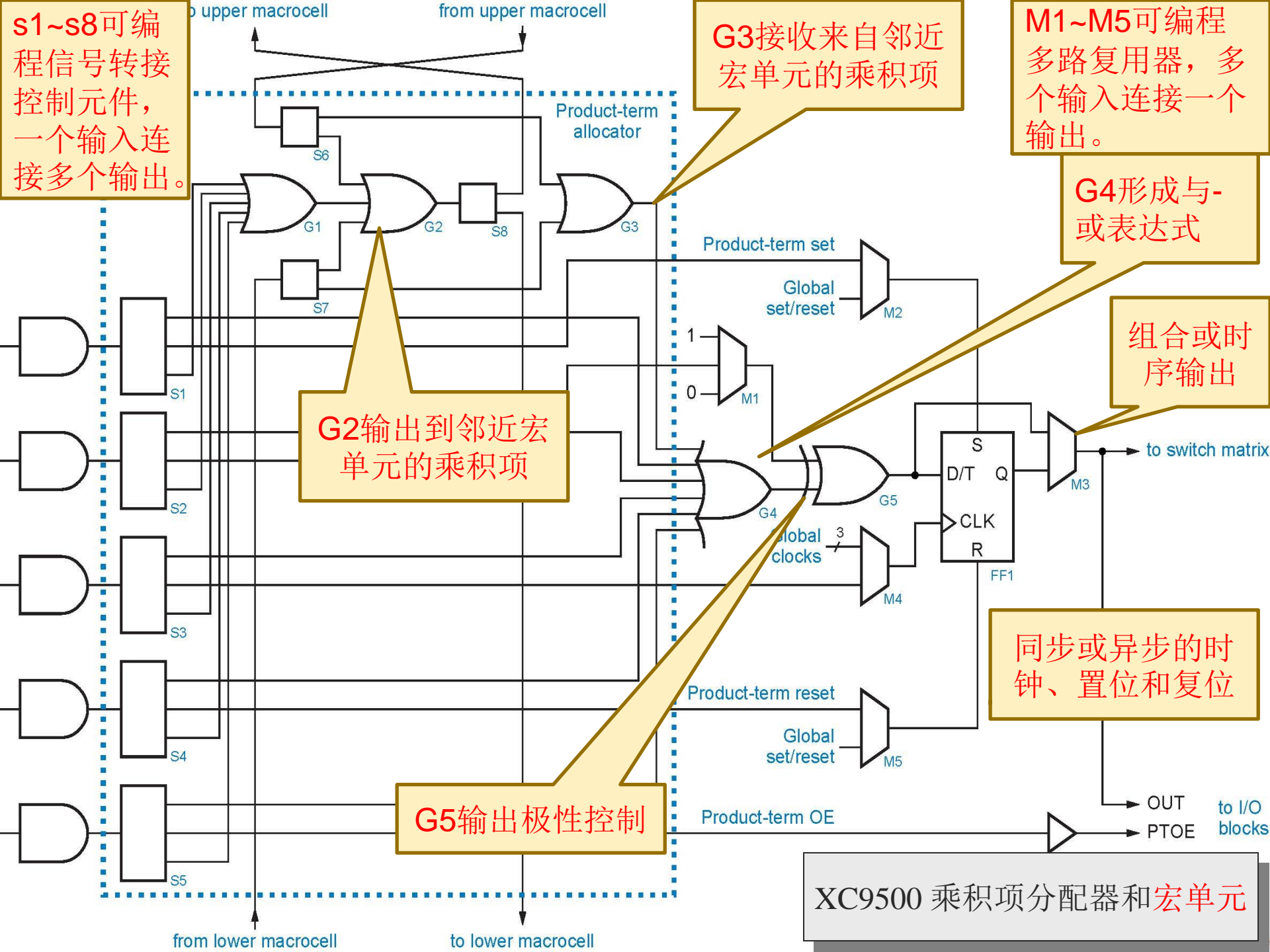




功能块体系结构



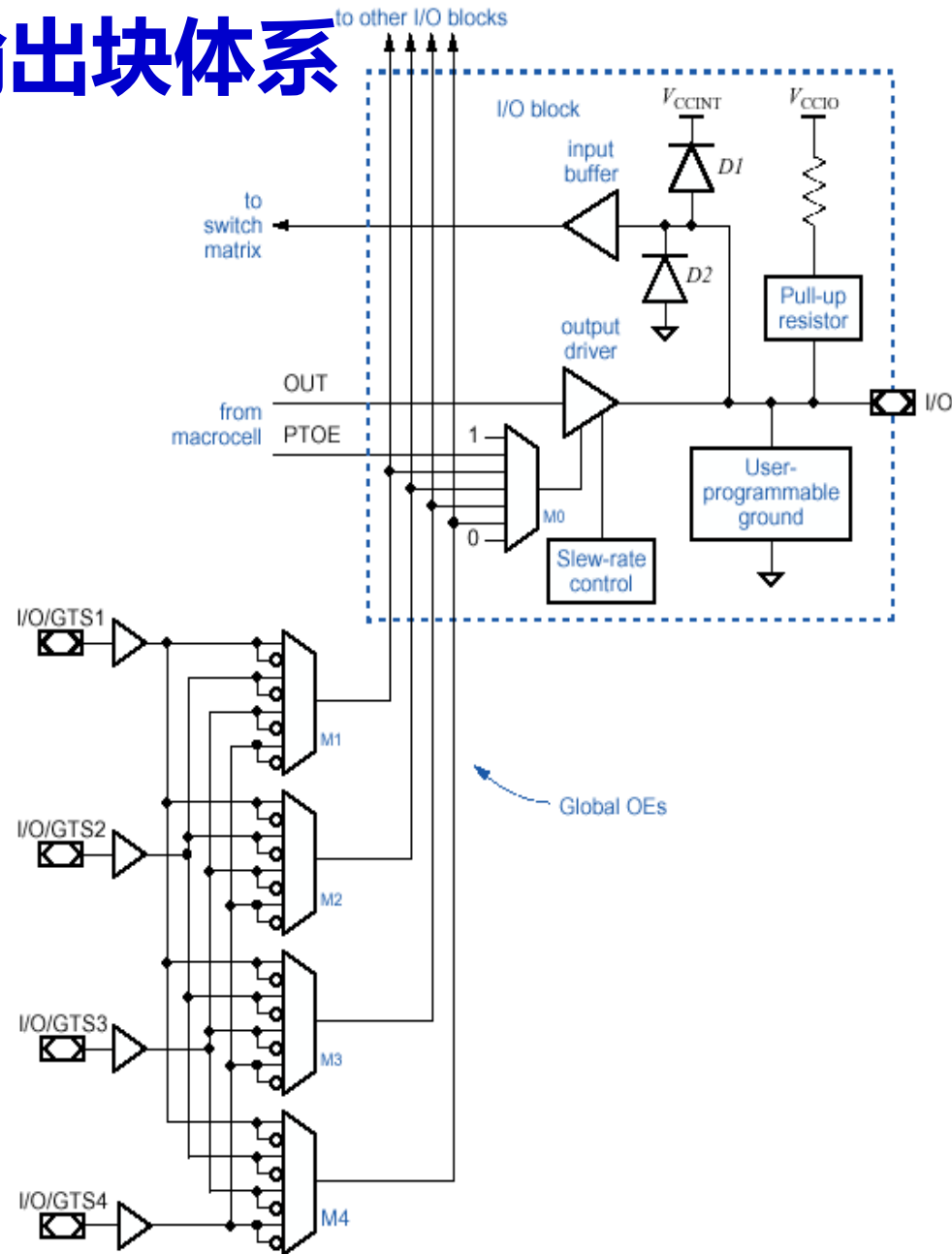
- XC9500具有5个与项。
- 拥有乘积项分配器，可以将本宏单元中未用的乘积项分配给邻近的宏单元使用。
- 最多90个乘积项。





- to other I/O blocks

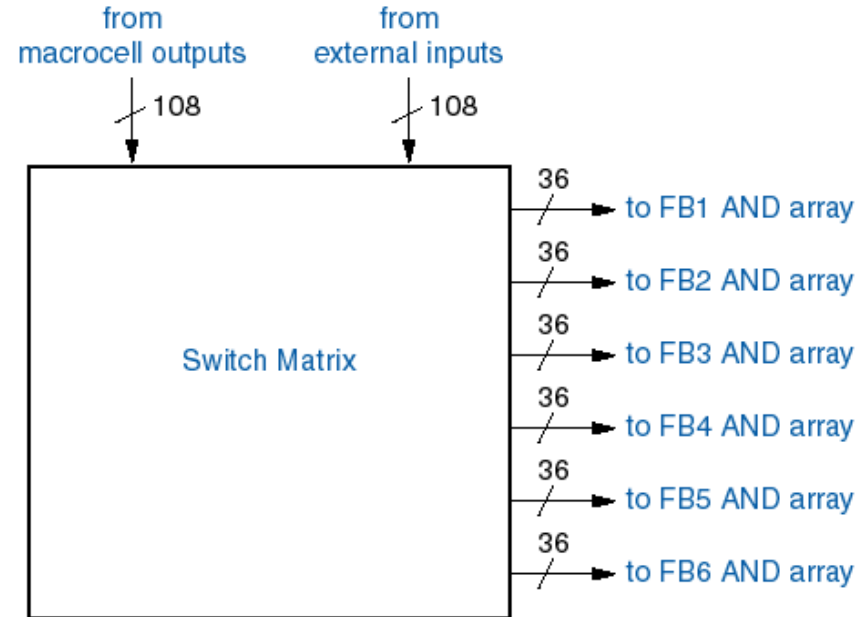
- 输入输出块IOB由输入缓冲器、输出缓冲器、输出选择器 (OE MUX) 和接地控制单元等构成。
- 7种输出使能信号，用于三态驱动缓冲器。
- 提供三种不同的“模拟”控制
 - 转换速率控制
 - 上拉电阻
 - 用户可编程接地
- 不同电压外部器件的兼容。





开关矩阵

- XC95108为例
 - 216个输入
 - 216个输出
 - 每个输出带216输入端的多路复用器
- 最小多路复用器方案
- 完全的、无阻塞交叉阵列
- fitter软件查找



Non-deterministic (非确定性) Polynomial

寻找通过稀疏开关矩阵的一个完备集，**NP问题**。

开关矩阵是芯片性能与fitter软件能力的折中。

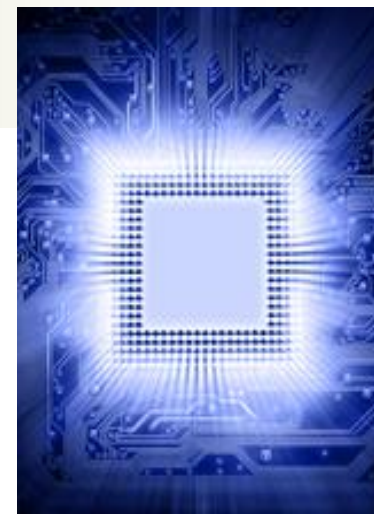


南京大学计算机科学与技术系
Department of Computer Science and Technology, Nanjing University



FPGA

现场可编程逻辑阵列
Field Programmable Gate Array





FPGA

- FPGA是在 CPLD 的基础上发展起来的新型高性能可编程逻辑器件，一般采用SRAM工艺。
- FPGA的集成度很高，可以完成极其复杂的时序与组合逻辑电路功能，适用于高速、高密度的数字逻辑电路设计领域。
- FPGA提高数字电路的重复利用性，通过可编程的方式实现不同的电路逻辑功能。
- 随着FPGA结构的不断演变和工艺的不断提升，在FPGA内部集成了越来越多的资源，比如块RAM、PLL、DSP、Transceiver、DDR IP以及处理器ARM等。
- 功能越来越强大
- 应用越来越广泛



FPGA优点

- 计算资源利用率：可以**进行任意精度**的计算
- **低功耗**：相比于GPU和CPU的功耗而言
- **硬实时**：在一个周期内要完成的任务是固定的，周期内必须做出响应。
- FPGA的并行处理体现在**逻辑单元的并发性**和**流水线的处理方式**上。
 - 并发性指的是各个逻辑模块相互独立处理，依赖各自的电路资源同时工作；
 - 流水线的处理方式就是将任务切片，同一时刻可以多任务执行（FPGA的并行优势更多的体现于此）。
 - FPGA带有丰富的**片上存储**资源，可以大大减少访问片外存储的延迟，提高计算性能。



FPGA的基本结构

I/O单元是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配需求。

可编程
I/O单元

基本可编程
逻辑单元

基本可编程逻辑单元是**可编程逻辑的主体**，由查找表（LUT）和寄存器组成的。

底层嵌入
功能单元

内嵌专用硬核
Hard core

内嵌专用硬核通用性相对较弱，不是所有FPGA器件都包含硬核。

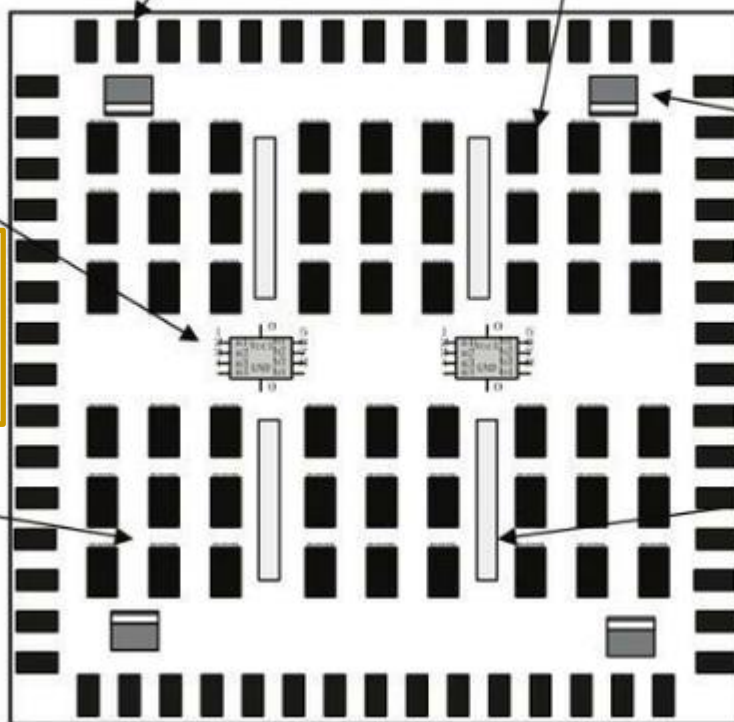
指的是那些通用程度较高的嵌入式功能模块，比如PLL、DLL、DSP、CPU等。

丰富的
布线资源

嵌入式块RAM

布线资源连通FPGA内部所有单元，连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。

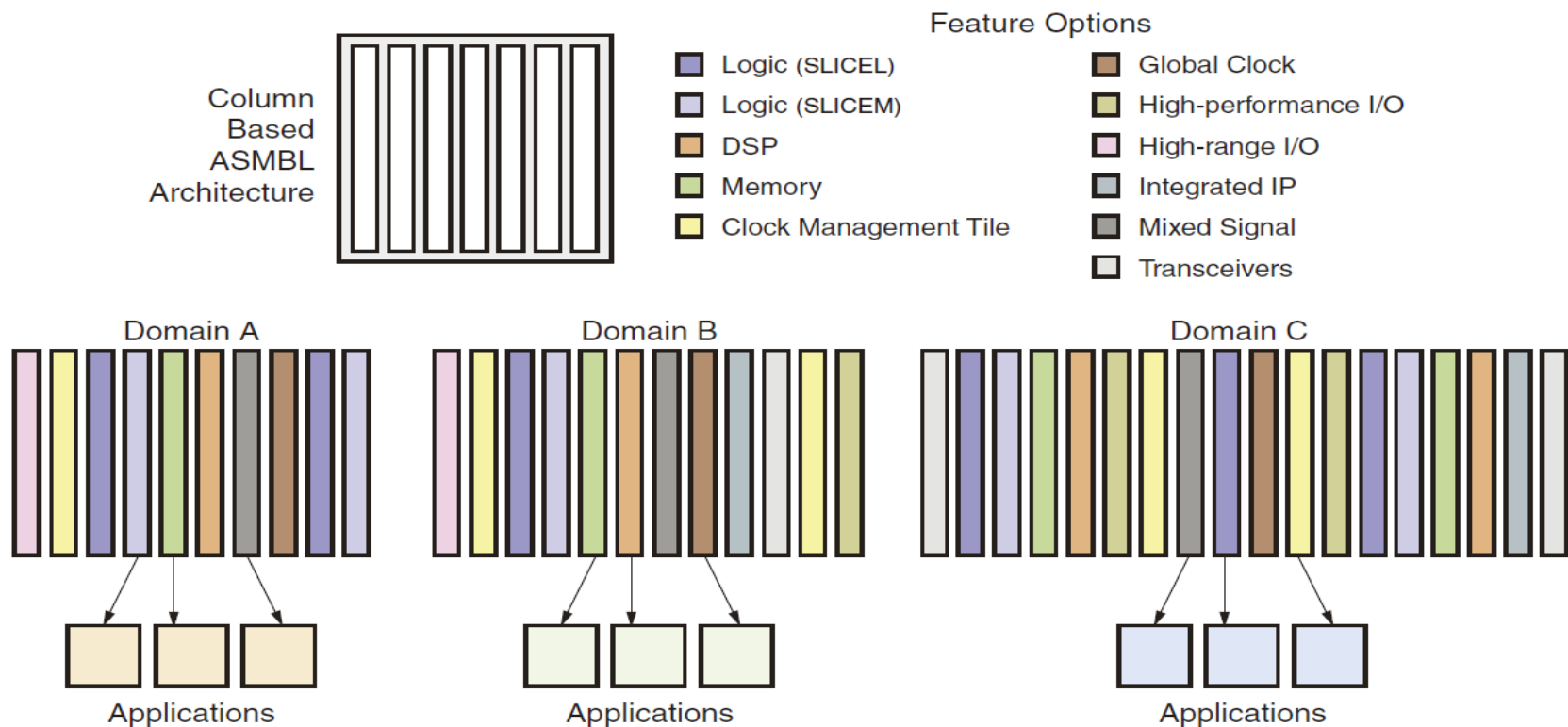
块RAM可以灵活配置为单端口RAM、双端口RAM、CAM、FIFO等常用存储结构。





Xilinx 7系列FPGA架构

- Xilinx创建了高级硅模块（ASMBL）架构提供的独特柱状方法。



UG474_c2_24_071014



Xilinx 7系列FPGA架构

块状RAM
Block SelectRAM
resource(BRAM)

底层功能模块，
如DSP等。

可编程输入输出块
I/O Blocks(IOBs)

可编程互连
Programmable Interconnect

可配置逻辑块
Configurable Logic
Blocks(CLBs)

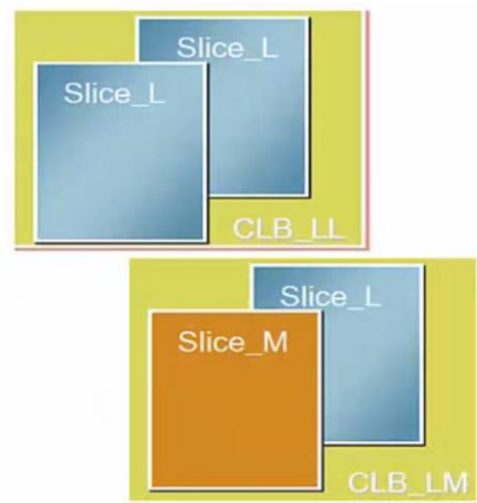
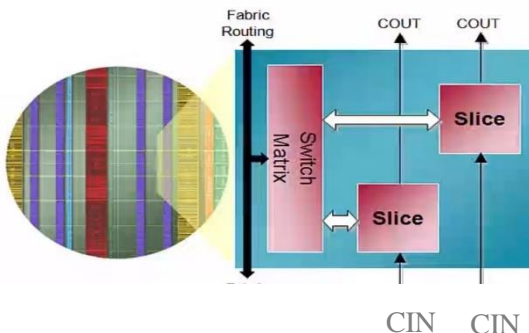
Artix-7系列 XC7A100T包括:
7925个可配置逻辑块CLB
4860Kb的快速块RAM
6个时钟管理片
240 DSP片

数字时钟管理
Clock Management
(DCMs,BUFGMUXes)



CLB模块

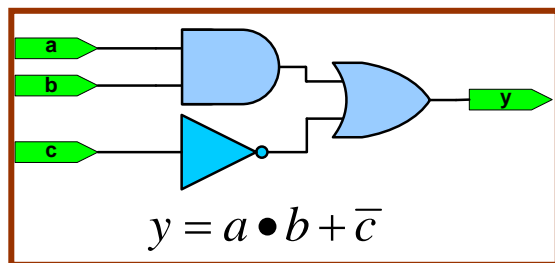
- 可配置逻辑块**CLB**是FPGA内的**基本逻辑单元**
- 一个CLB由**两个Slices**和**开关矩阵**构成，两个Slice相互独立，每一个Slice都被组织为一列，都有一个独立的进位链。
- **Slice**分为**两种**:
 - **SliceL**: 仅仅用来实现组合逻辑和时序逻辑。
 - **SliceM**: **不仅**可以实现组合逻辑和时序逻辑，**还可以**被用来实现**分布式RAM**或32位的**移位寄存器SRL**。
- 每个CLB包含
 - 两个**SliceL**
 - 一个**SliceL**与一个**SliceM**
- 7925个可配置逻辑块CLB
 - 15,850个逻辑片Slice
 - ◆ 11100SliceL
 - ◆ 4750SliceM





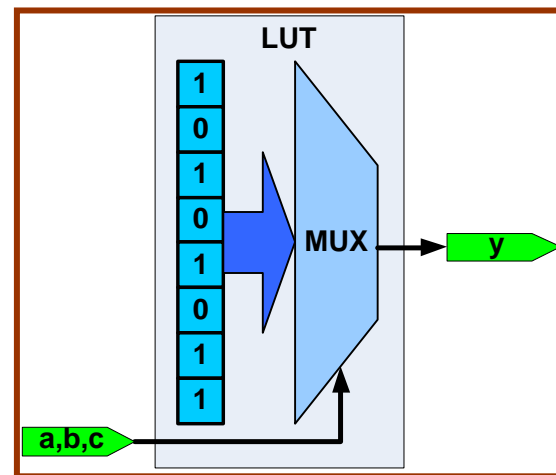
查找表

- 查找表look-up-table(LUT)，也称为函数发生器，本质上是一个存储器RAM。
- 先把逻辑函数的输出数据写入RAM，每当输入一组信号就等于输入一个地址进行寻址，找出该地址对应的存储内容输出。
- LUT对应于逻辑函数的真值表。
- 通过查找表的时延是固定的。



Truth Table

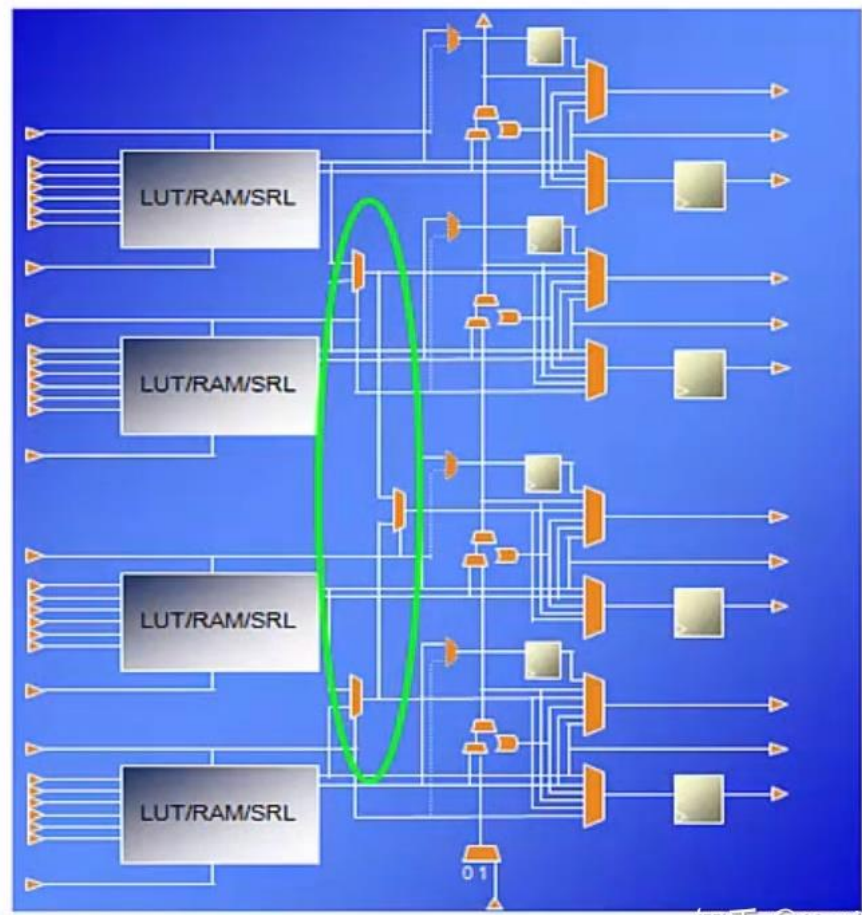
a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



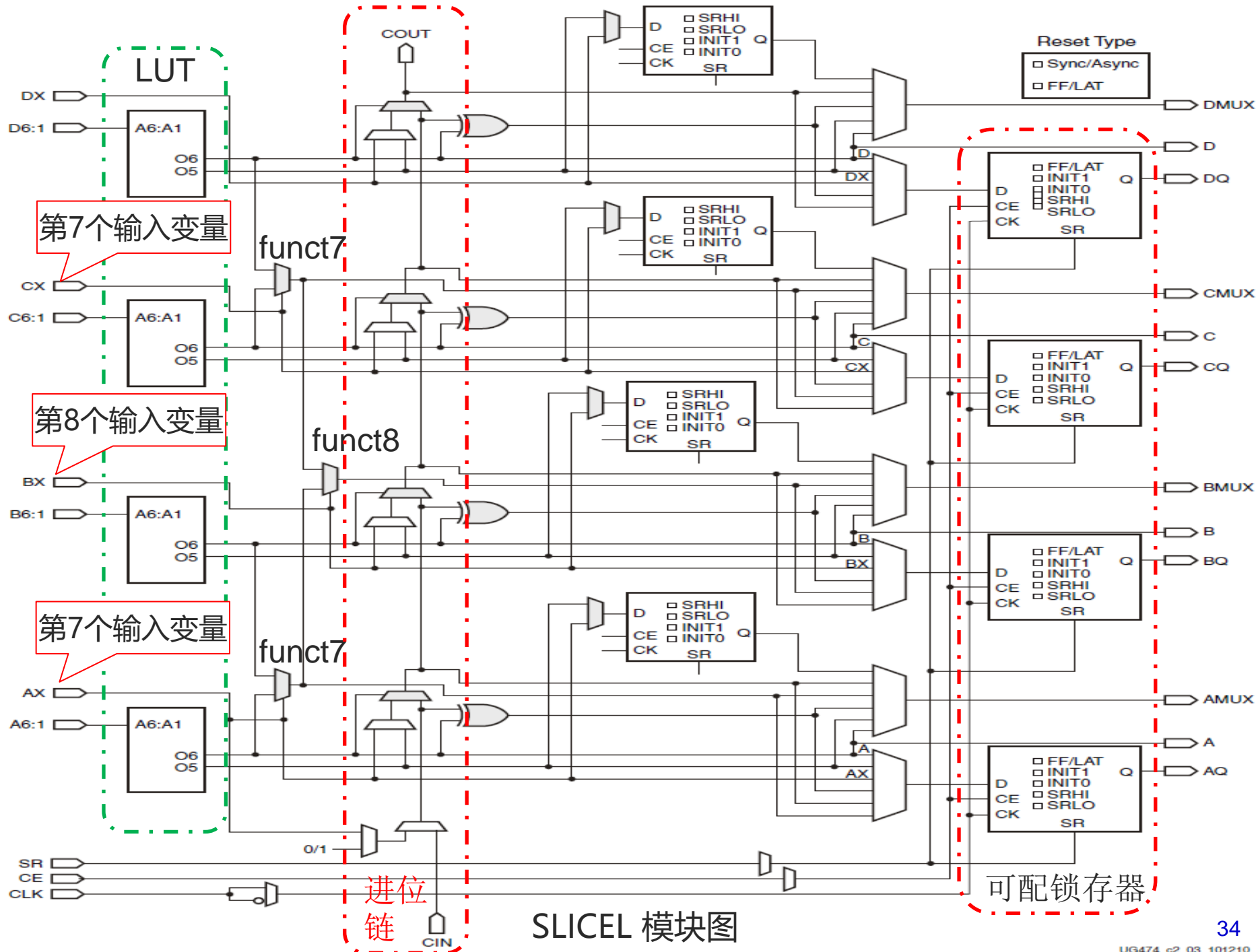


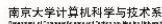
Slice结构

- 每个slice包含：
 - 4个6输入的LUT
 - 多个多路复用器MUX
 - 8个触发器FF/LATCH
 - 进位链的逻辑，实现算术功能
 - 2个独立输出（O5和O6）。
- 6输入LUT可以实现：
 - 任意的6输入逻辑函数
 - 连接A,B两个LUT 用于产生7输入的逻辑函数
 - 连接C,D两个LUT用于产生7输入的逻辑函数
 - 连接4个LUT 的多路选择器产生8输入的逻辑函数



香农定理: $F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + \overline{X_1} \cdot F(0, X_2, \dots, X_n)$





- 一个6输入LUT可配置成4:1MUX，4路输入2个选择位。
- 两个LUT可配置成最多8:1 MUX
- 四个LUT可配置成16个MUX

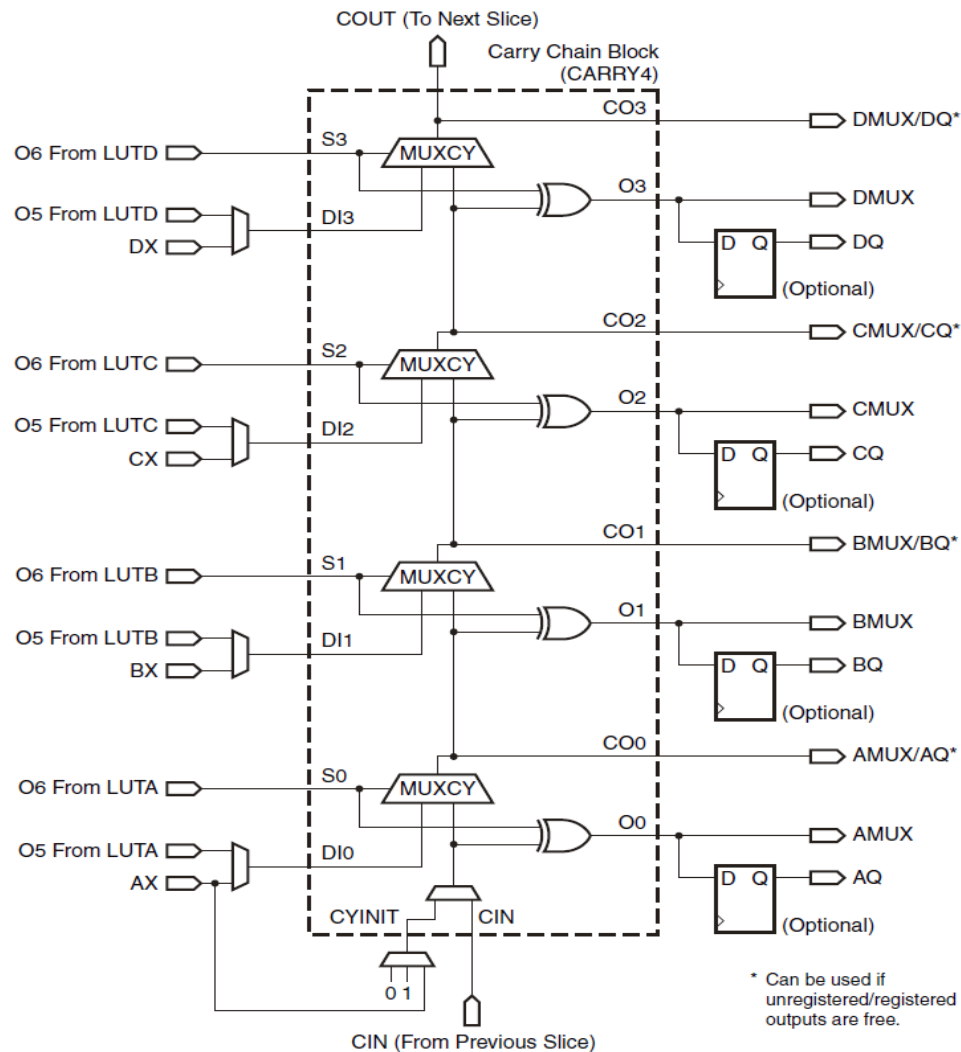


35



进位链

- 每个SLICE有4bit的进位链。每bit都由一个进位MUX(MUXCY)和一个异或门组成，可在实现加法/减法器时生成进位逻辑。
- 该MUXCY与XOR也可用于产生一般逻辑。



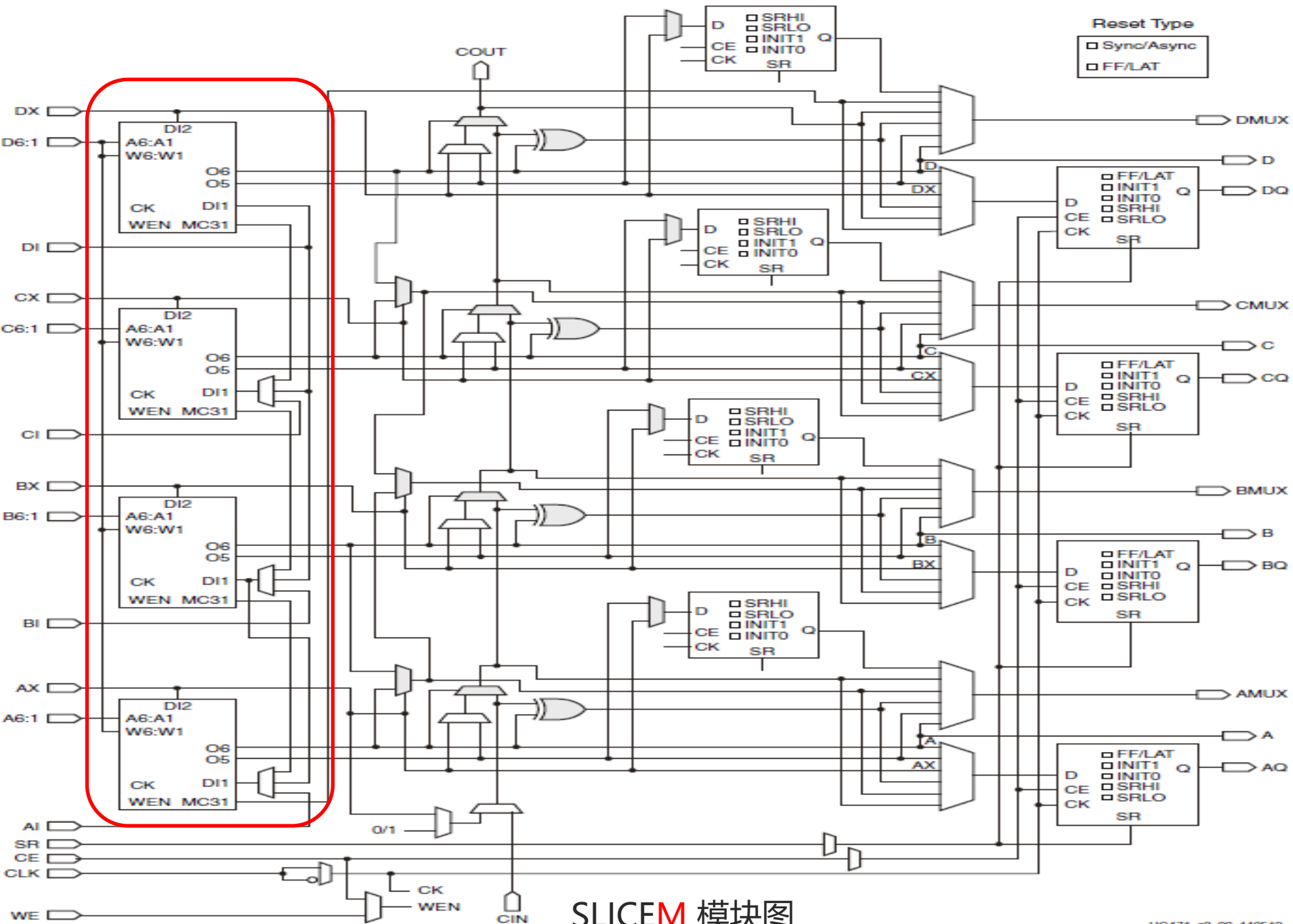
UG474_c2_23_071813

Figure 2-24: Fast Carry Logic Path and Associated Elements



SLICE的存储元件

- 每个Slice有8个存储元件。
 - 其中4个可以被配置成边沿触发的D触发器或电平触发的锁存器。
 - ◆ D输入可由LUT输出通过AFFMUX, BFFMUX, CFFMUX或DFFMUX直接驱动, 或通过BYPASS slice输入绕过函数发生器(LUT)通过AX, BX, CX或DX输入。
 - 另外4个存储元件只能被配置成边沿触发的D触发器。
 - ◆ D输入可以由LUT的输出O5驱动, 或者BYPASS slice输入通过AX, BX, CX或DX输入。
 - 当原来的4个存储元件被配置成锁存器时, 另外4个额外的存储元件就不能被使用。



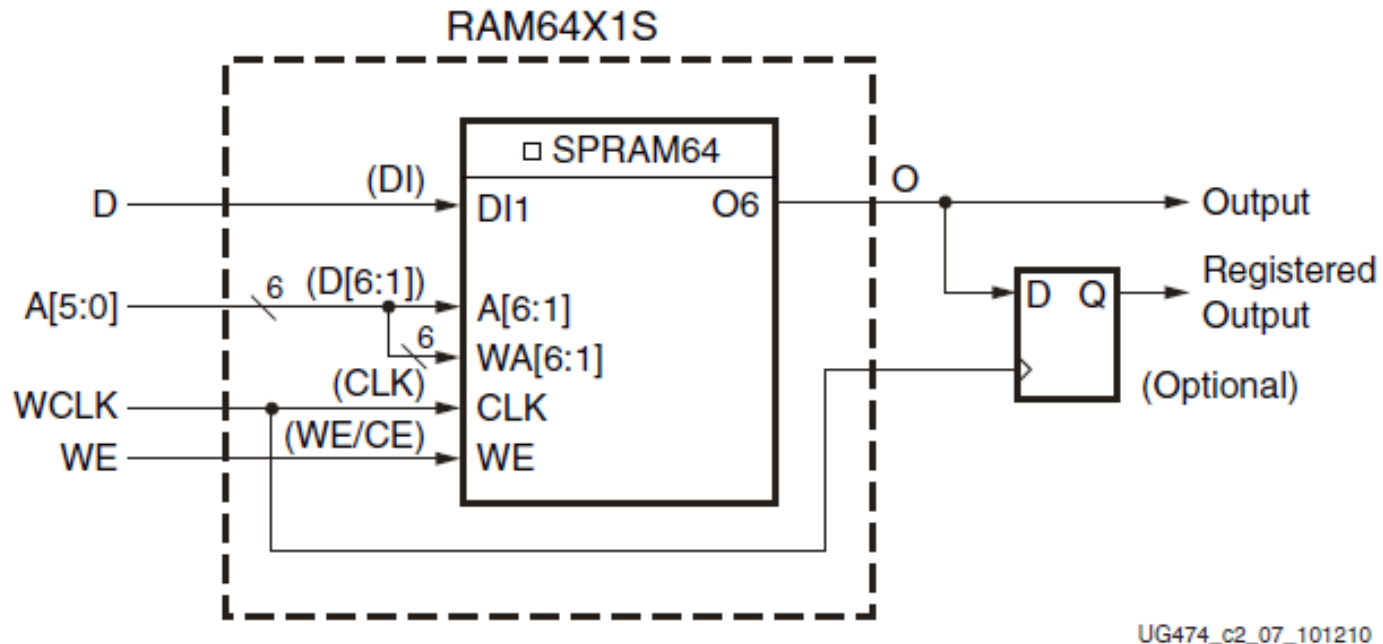
SLICEM 模块图

Figure 2-3: Diagram of SLICEM



64 X 1 Single Port Distributed RAM (RAM64X1S)

- 1个6输入LUT可用作64 x 1存储器，作为分布式RAM，满足小型存储要求。
- 4个LUT可实现64 X 4。
- 分布式RAM容量1188kb。



UG474_c2_07_101210

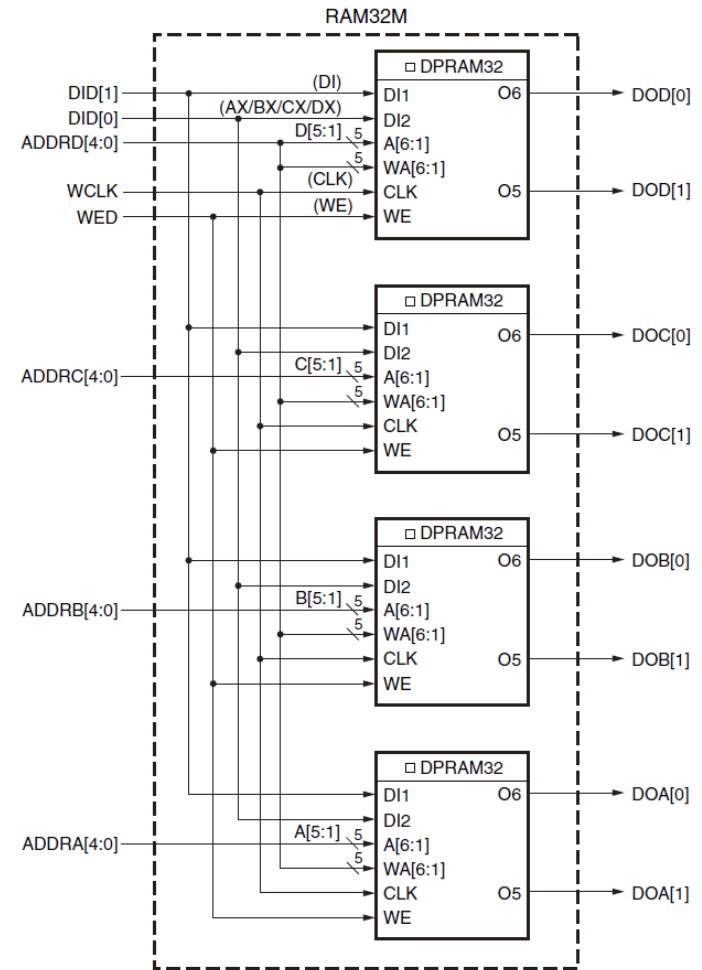
Figure 2-8: 64 X 1 Single Port Distributed RAM (RAM64X1S)



32 X 2 Quad Port Distributed RAM (RAM32M)

■ 分布式RAM和 BLOCK RAM的选择遵循以下方法：

1. 小于或等于64bit容量的都用分布式实现
2. 深度在64~128之间的，若无额外的block可用分布式RAM。要求异步读取就使用分布式RAM。数据宽度大于16时用block ram.
3. 分布式RAM有比block ram更好的时序性能。分布式RAM在**逻辑资源CLB**中。而BLOCK RAM则在**专门的存储器列**中，会产生较大的布线延迟，布局也受制约。



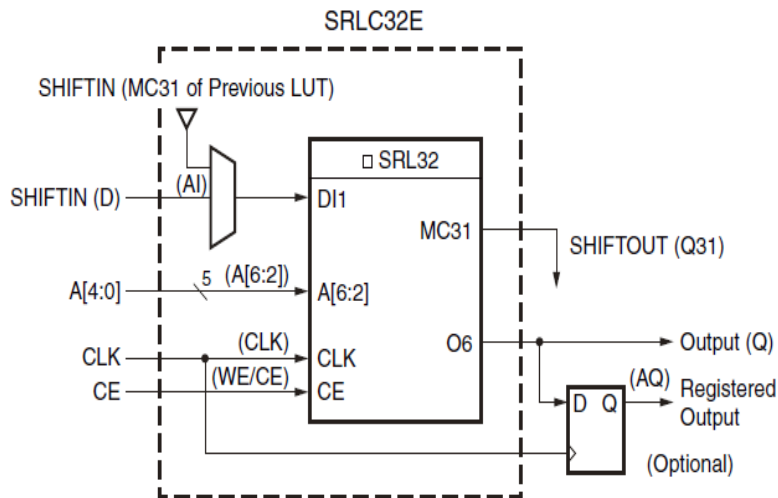
UG474_c2_06_070914

Figure 2-6: 32 X 2 Quad Port Distributed RAM (RAM32M)



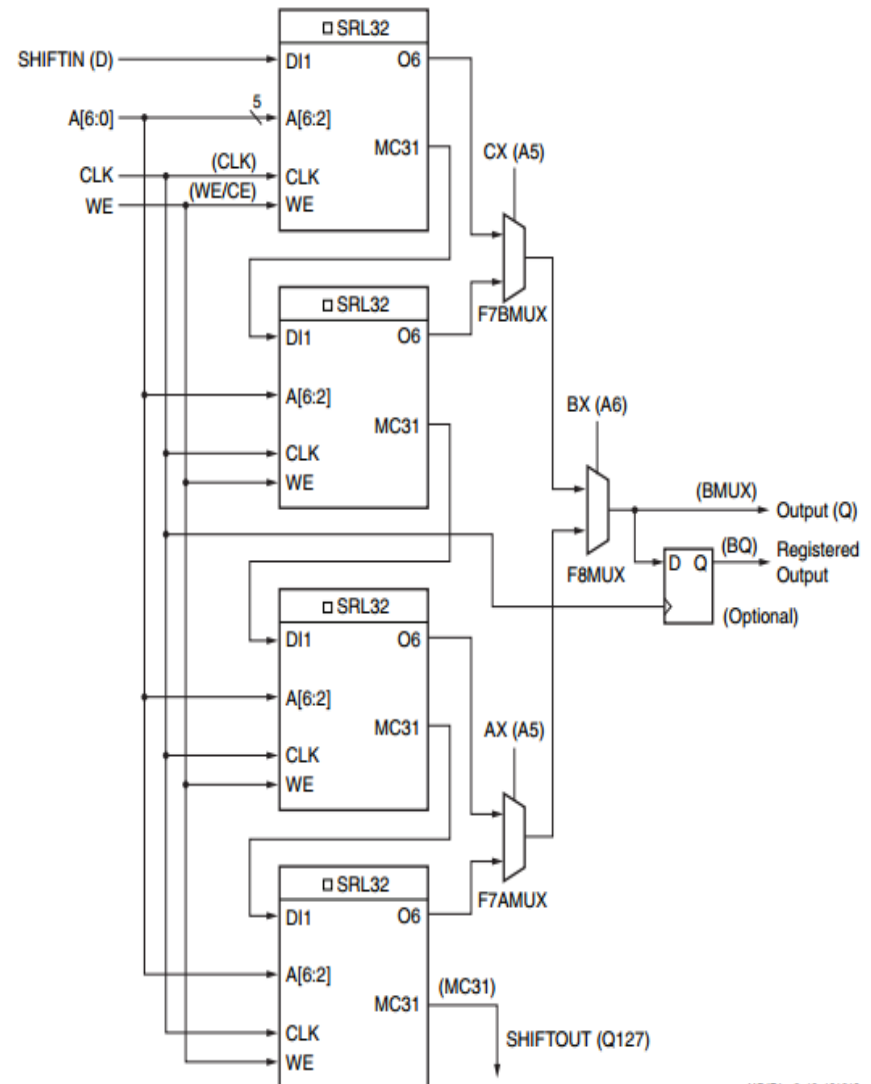
32-Bit Shift Register Configuration

- SLICEM中的LUT能在设置成32bit的移位寄存器
- 4个LUT可级联成128bit的移位寄存器。



UG474_c2_14_110510

Figure 2-15: 32-Bit Shift Register Configuration

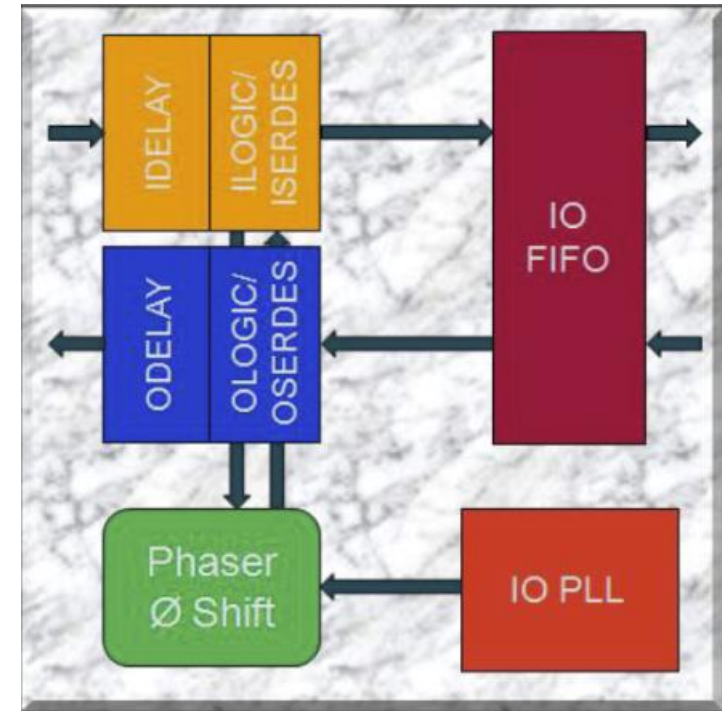


UG474_c2_19_101210



IOB

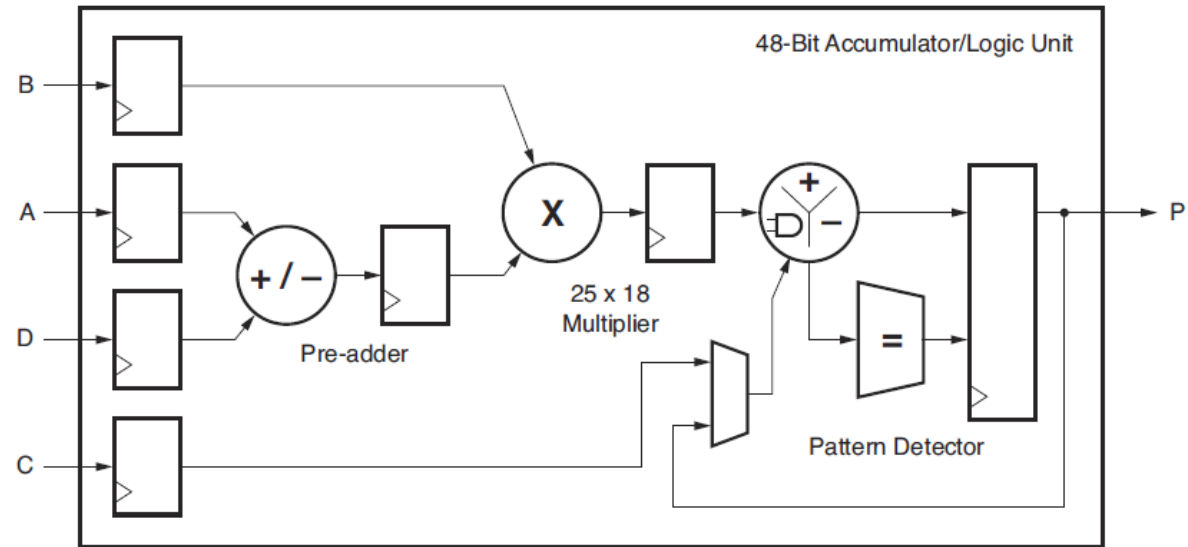
- 7系列FPGA I/O bank分为高性能 (HP) banks和宽范围 (HR) banks。
 - HP I/O banks可以实现高性能存储器接口或者芯片间高速接口，支持电压最大1.8V
 - HR I/O banks支持更宽范围的I/O电平标准，**最大支持3.3V**。
- SelectIO管脚可以配置为单端和差分信号：
 - 单端I/O标准 (例如**LVC**MOS、LVTTTL、HSTL、PCI和SSTL)
 - 差分I/O标准 (例如，Mini LVDS、RSDS、PPDS、BLVDS和差分HSTL和SSTL)





DSP

- 25X18补码乘法器
- 48位运算器
- 48位逻辑运算
- 低功耗先行加法器
- SIMD运算单元
- 模式检测
- 有240个DSP slice



UG479_c1_21_032111

Figure 1-1: Basic DSP48E1 Slice Functionality



BLOCK RAM

- 每个块RAM存储**36kb**的数据
- 可以配置为**两个**独立的18 Kb RAM 或一个36Kb RAM。
- 每个36kb块RAM可以配置为64K x 1
- **写**和**读**是同步操作;
- 两个端口**对称**且完全**独立**，只**共享**存储的数据。
- 每一个端口可以配置在一个**可用的宽度**，独立于其他端口。
- Artix-7 100T有135个块，BRAM容量：4860kb。

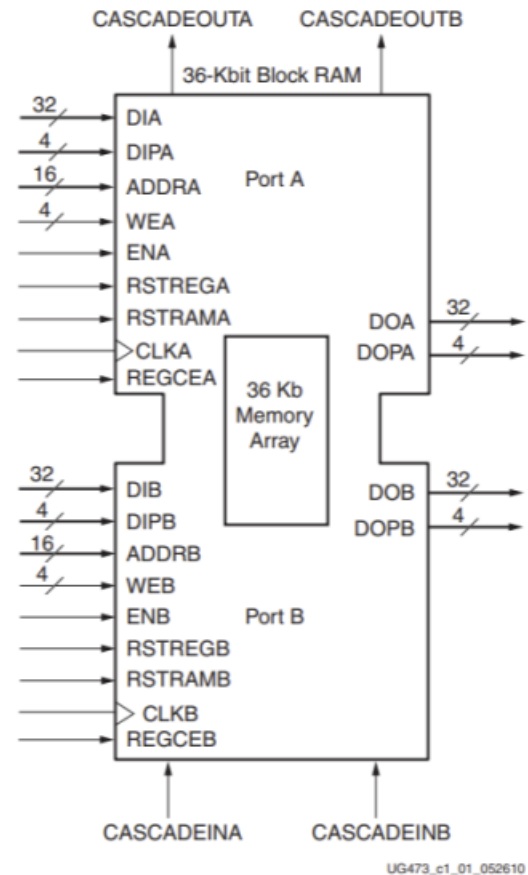


Figure 1-1: True Dual-Port Data Flows for a RAMB36

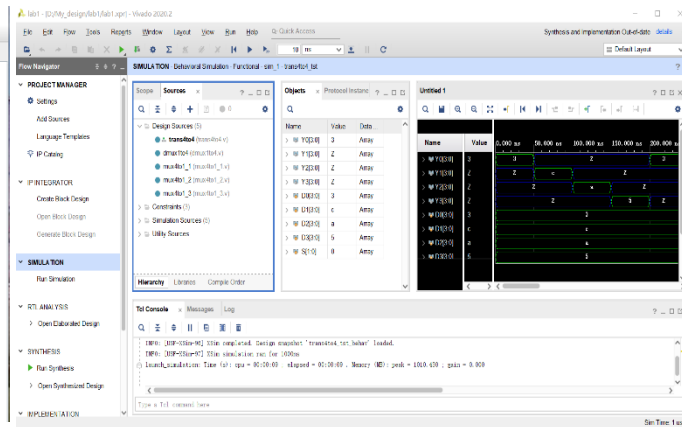
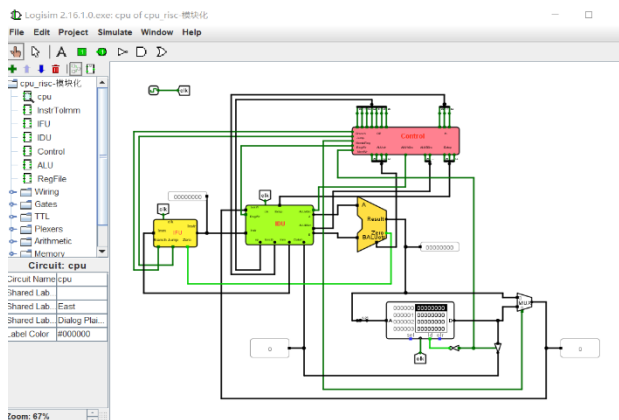
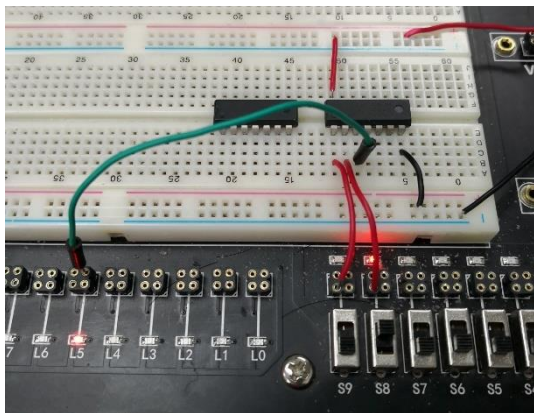


数字系统设计

■ 分为三个发展阶段:

- 1、全人工设计阶段：手工设计、手工测试
- 2、Computer Aided Design：计算机设计、手工测试
- 3、Electronic Design Automatic：计算机设计和测试

- ◆ 使用硬件描述语言Hardware Descriptions Languages
- ◆ 利用自动化综合软件，自动生成电路及版图，并用仿真软件完成功能测试仿真





硬件描述语言

- 硬件描述语言 (Hardware Description Language, HDL)
 - 是用来描述数字电路功能、行为的语言，可以在逻辑门级、寄存器传输级、行为级等对数字电路系统进行描述。
 - 被逻辑综合工具识别，并自动转换到逻辑门级网表，使得硬件描述语言可以被用来进行电路系统设计，并能通过逻辑仿真的形式验证电路功能。
 - 常见的硬件描述语言包括Verilog、VHDL、Chisel等。
- 与原理图设计方法相比：
 - 更适于描述大规模的数字系统；
 - 在抽象的层次上描述系统的结构与功能；
- 与传统的软件编程语言相比：
 - 能够对于硬件电路的时序特性进行描述。
 - 小到简单的触发器，大到复杂的超大规模集成电路（如处理器），都可以利用硬件描述语言来描述。

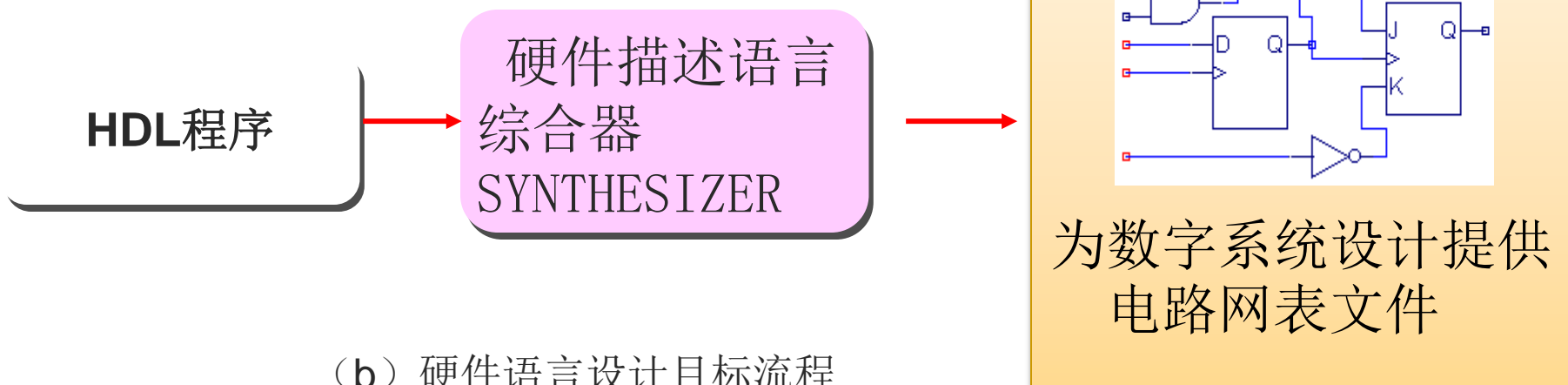


逻辑综合器的功能

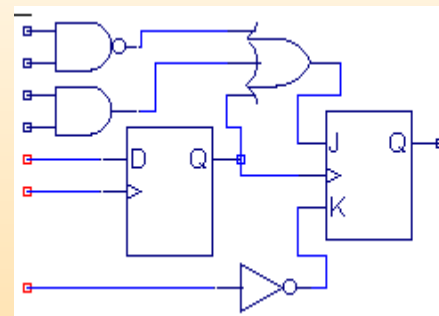
■ 与编程语言编译器功能类似



(a) 软件语言设计目标流程



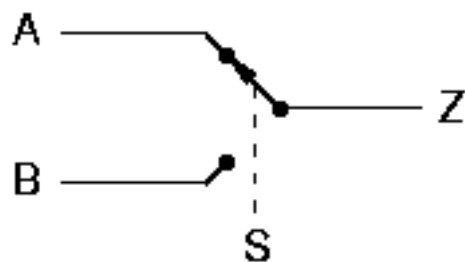
(b) 硬件语言设计目标流程



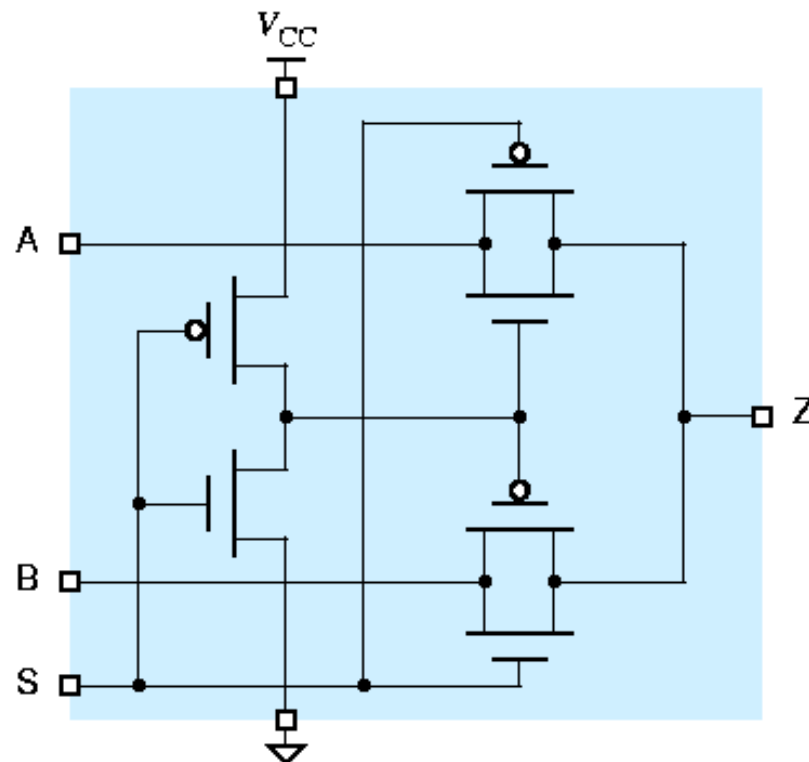


数字系统设计举例

- 多路复用器：2个输入（A， B）、1个控制输入S和一个输出Z)



开关模型



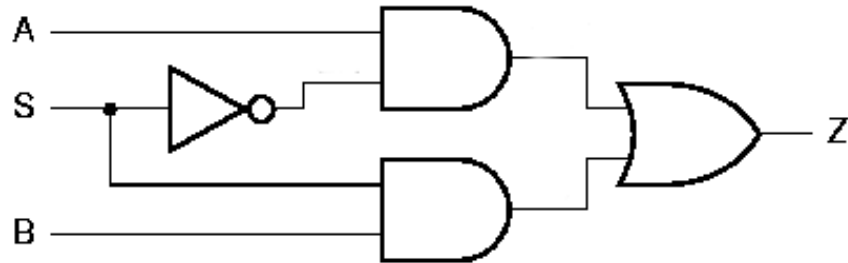
使用CMOS传输门设计的多路复用器



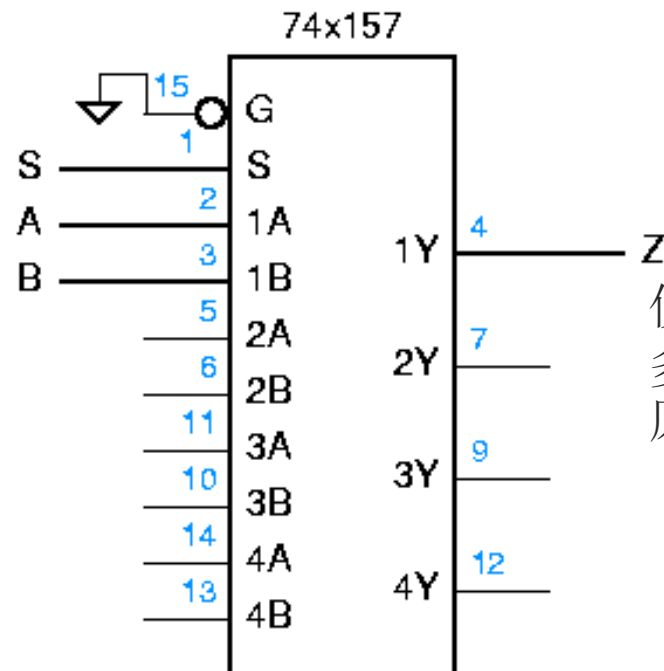
数字系统设计举例

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

多路复用器真值表



多路复用器的门级逻辑原理图



使用MSI构件的
多路复用器逻辑
原理图



数字系统设计举例

```
library IEEE;
use IEEE.std_logic_1164.all;

entity Vchap1mux is
    port ( A, B, S: in  STD_LOGIC;
          Z:          out STD_LOGIC );
end Vchap1mux;

architecture Vchap1mux_arch of Vchap1mux is
begin
    Z <= A when S = '0' else B;
end Vchap1mux_arch;
```

多路复用器VHDL程序

```
module Vrchap1mux(A,B,S,Z);
    input A,B,S;
    output Z;

    assign Z= (S==0) A :B;
endmodule
```

多路复用器Verilog HDL程序