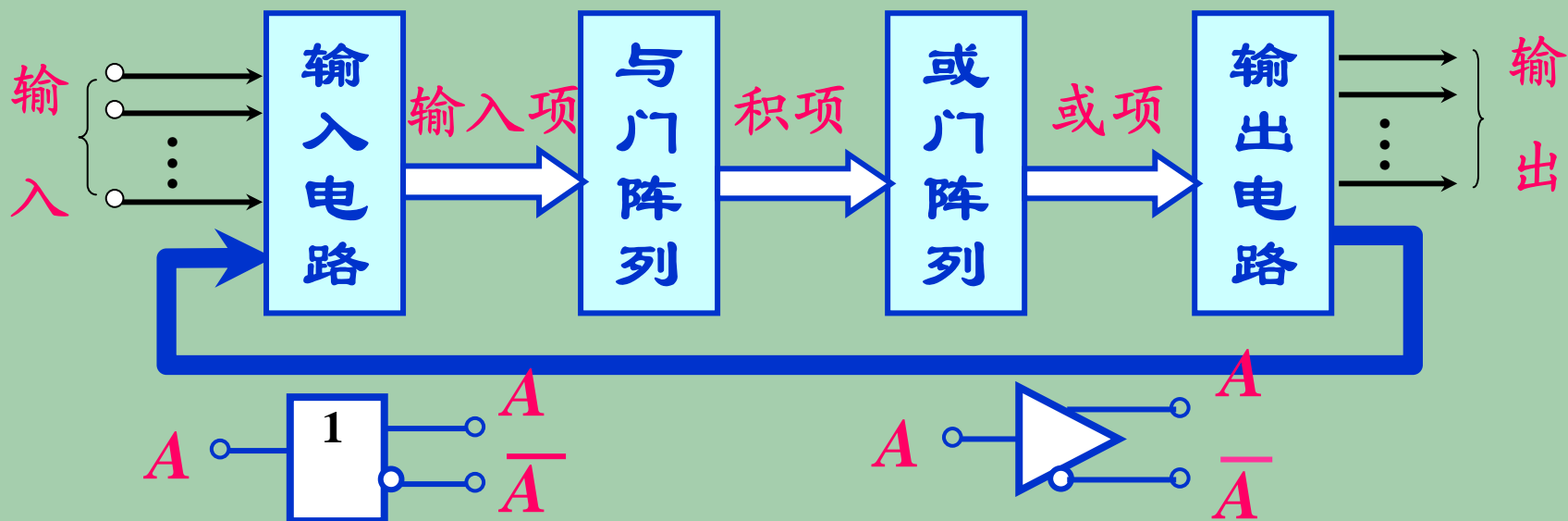


5.5 可编程逻辑器件和时序逻辑电路的VHDL及其仿真

5.5.1 可编程逻辑器件 (PLD) (Programmable Logic Device)

一、PLD的基本结构和分类

1. 基本结构



PLD的输入缓冲电路



2. 分类

(1) 按可编程情况分

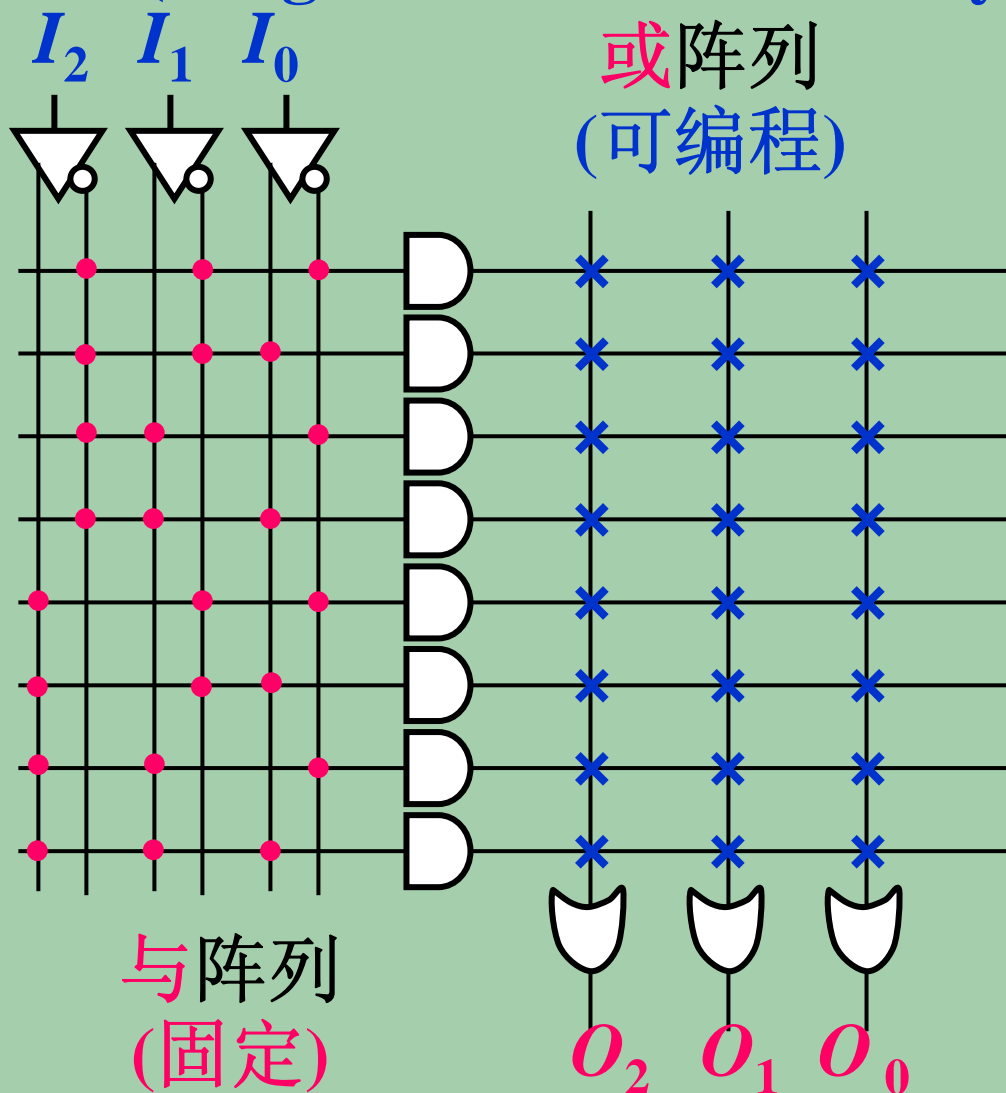
分 类	与阵列	或阵列	输出电路	出现年代
PROM	固定	可编程	固定	70年代初
PLA	可编程	可编程	固定	70年代中
PAL	可编程	固定	固定	70年代末
GAL	可编程	固定	可组态	80年代初



● PROM

— 可编程只读存储器

(Programmable Read Only Memory)



缺点:

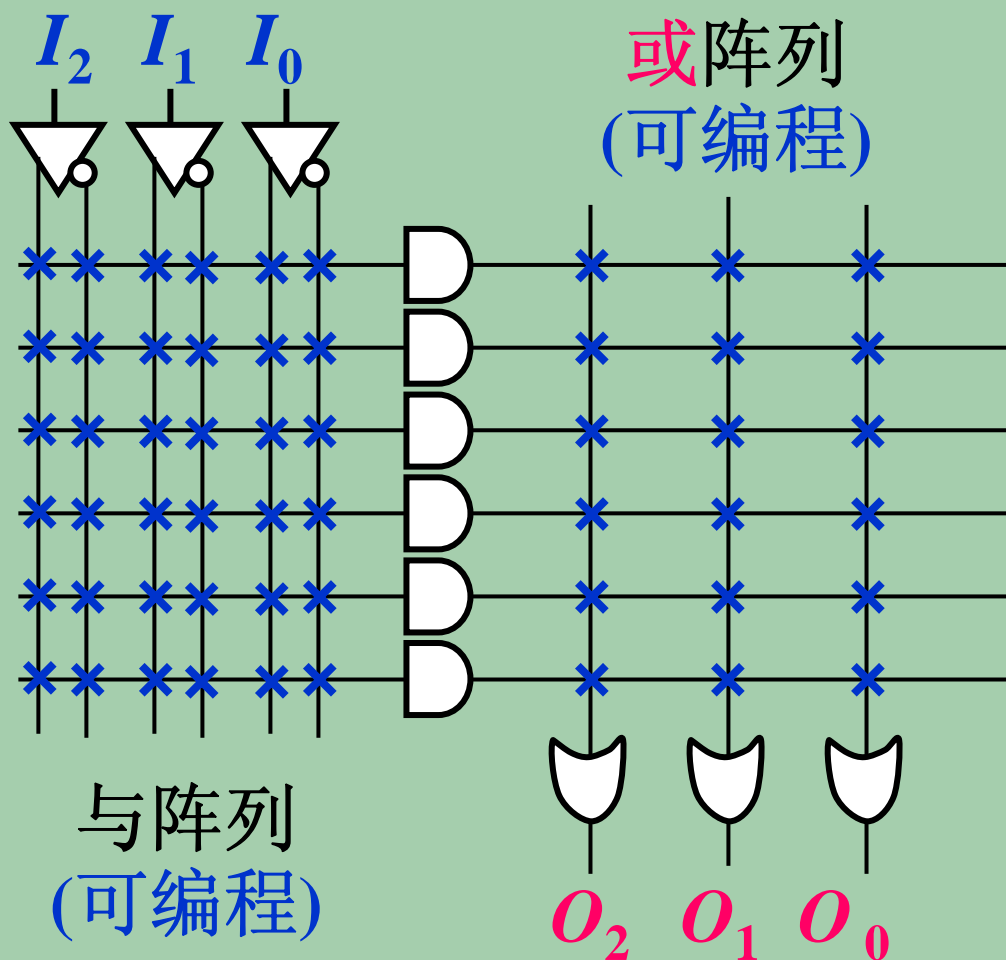
- 只能实现标准与或式
- 芯片面积大
- 利用率低,不经济

用途:

- 存储器
- 函数表
- 显示译码电路



● PLA — 可编程逻辑阵列 (Programmable Logic Array)



优点:

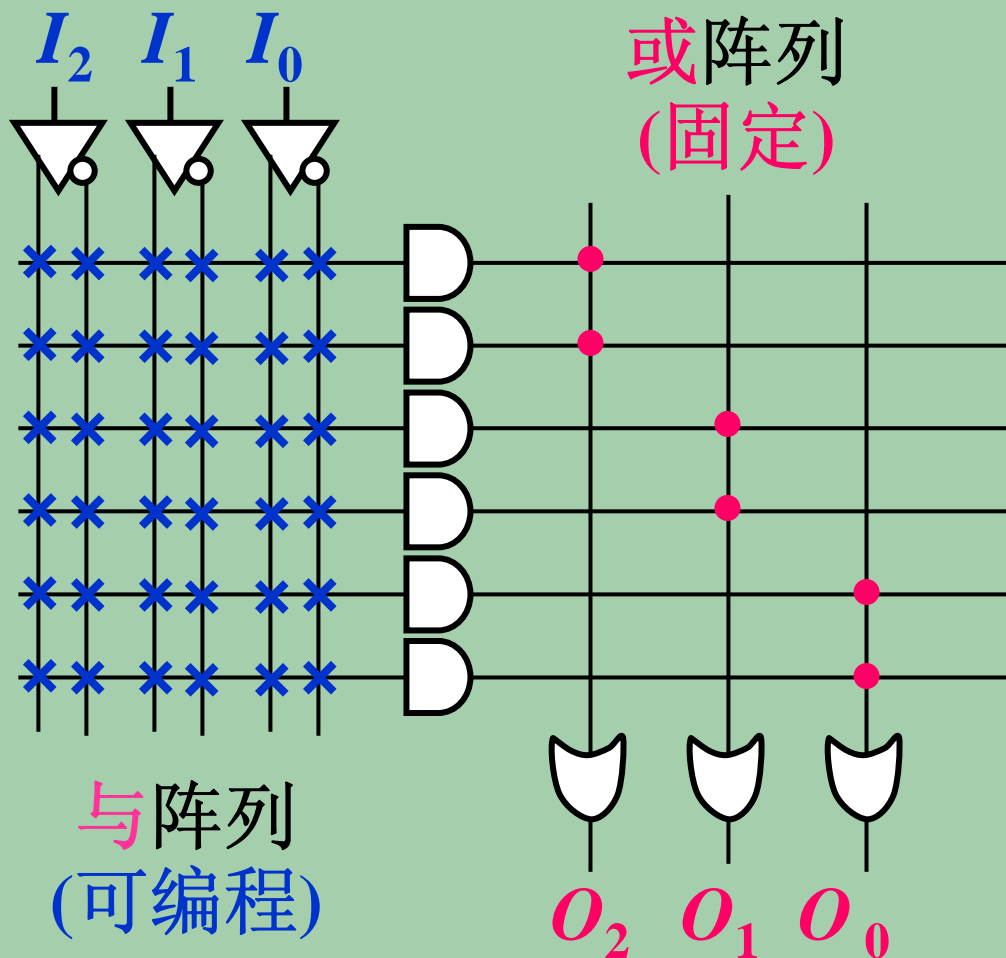
- 与阵列、或阵列都可编程
- 能实现最简与或式

缺点:

- 价格较高
- 门的利用率不高



● PAL — 可编程阵列逻辑 (Programmable Array Logic)



优点:

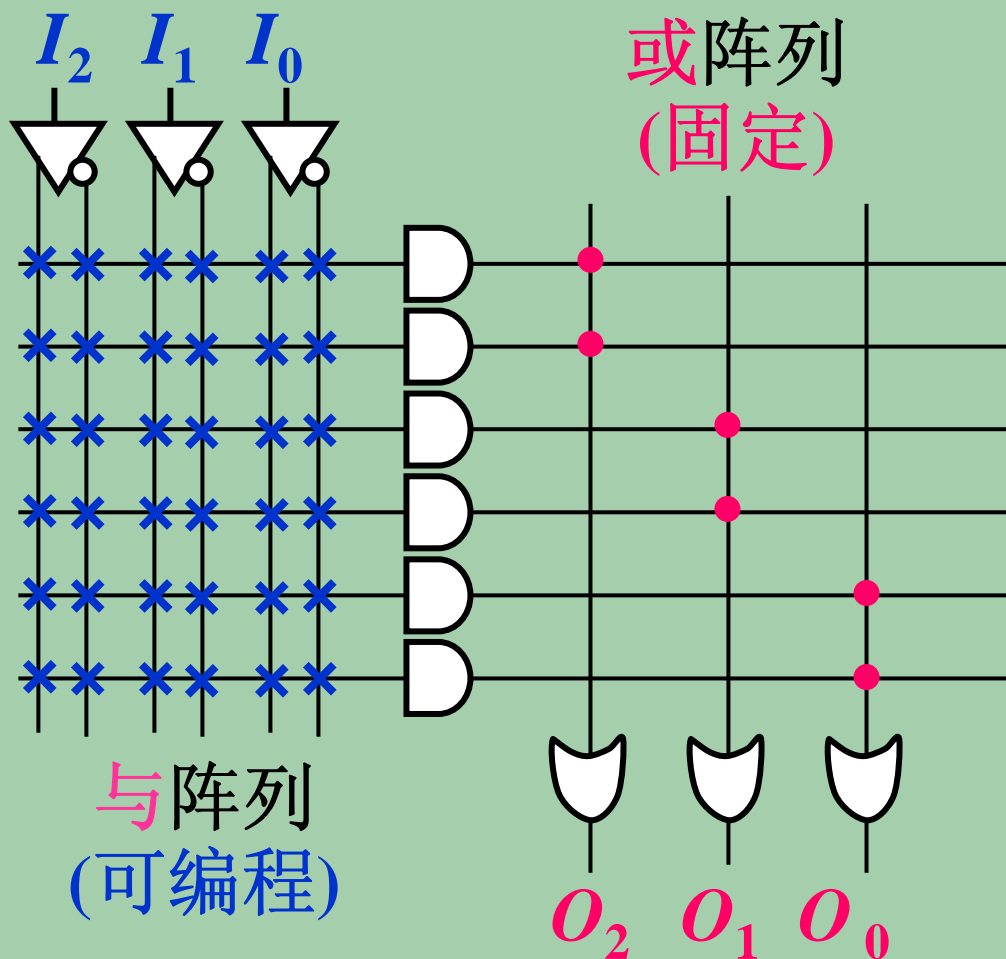
- 速度快
- 价格低
- 采用编程器现场编程

缺点:

- 输出方式固定
一次编程



● GAL — 通用阵列逻辑 (Generic Array Logic)



优点:

- 具有 PAL 的功能
- 采用逻辑宏单元使输出自行组态
- 功能更强, 使用灵活, 应用广泛



(2) 按可编程和改写方法分

PLD	编程方式	改写方法	特点、用途
第一代	一次性掩模 (厂家)	不能改写	固定程序、数据、函数表、字符发生器
第二代	编程器(用户)	紫外光擦除	先擦除, 后编程
第三代	编程器(用户)	电擦除	擦除、编程同时进行
第四代	在系统可编程	软件	直接在目标系统或线路板上编程

(3) 按组合、时序分

组合 电路	{	PROM、PLA	{	时序型 PAL
		组合型 PAL		GAL (也可实现组合电路)

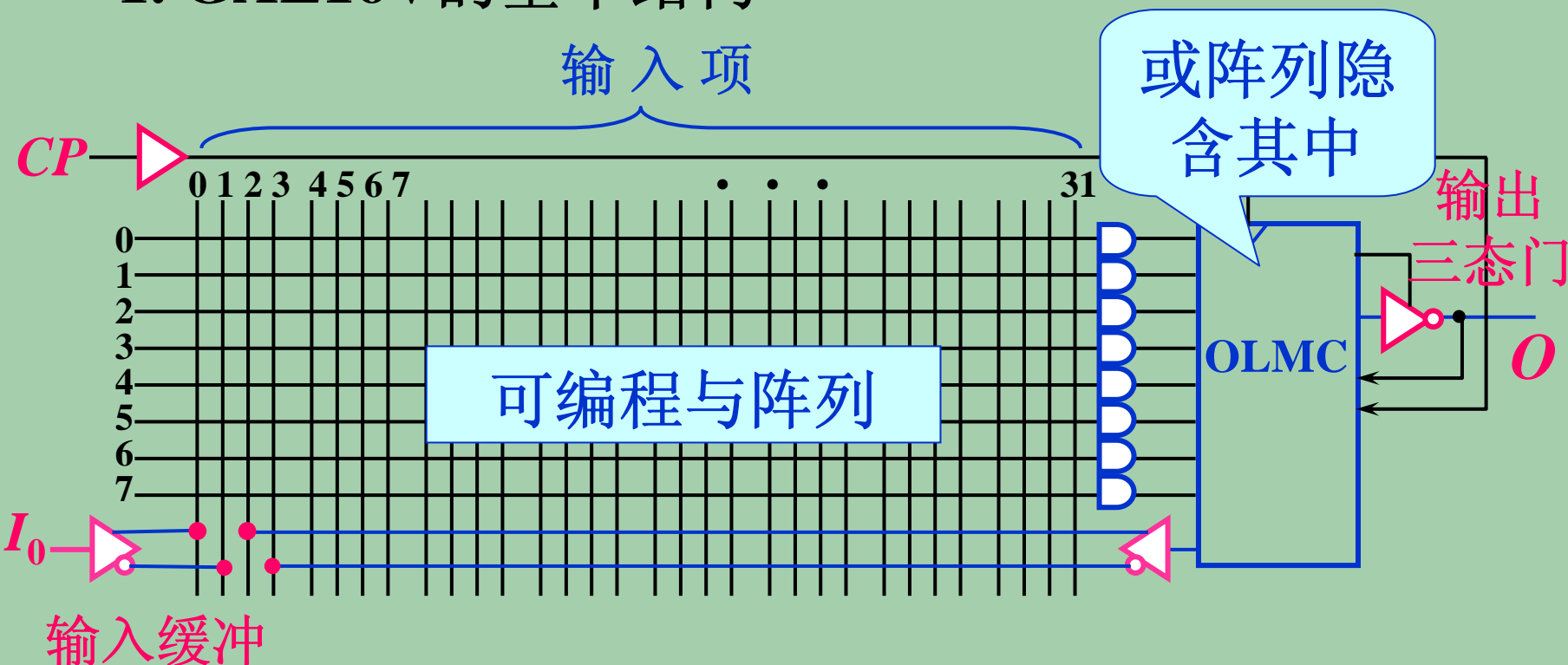


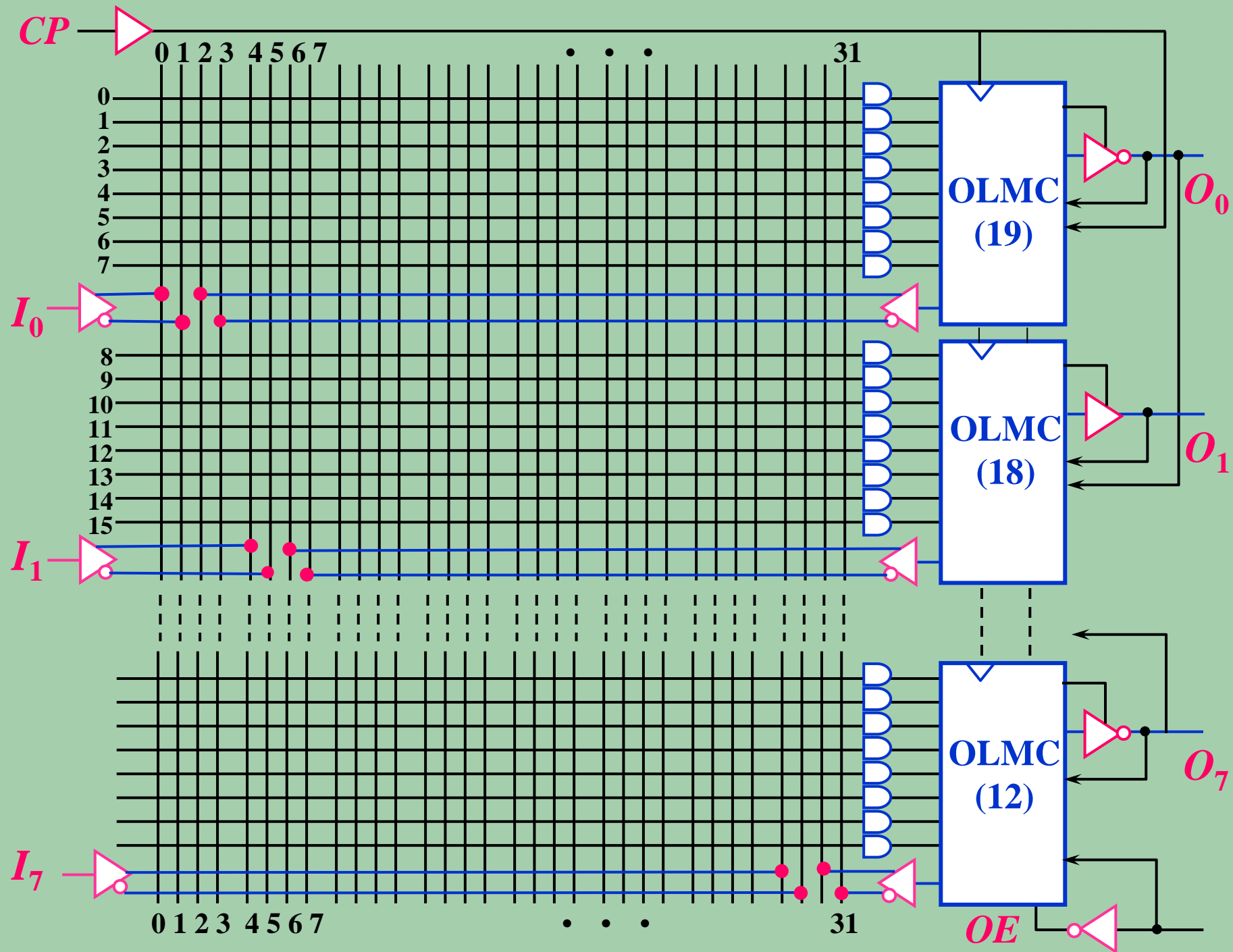
二、PLD的基本原理

PROM的原理已在第三章介绍，不赘述。

PAL的输出方式固定而不能重新组态，且编程是一次性的，使用有较大的局限。

1. GAL16V的基本结构







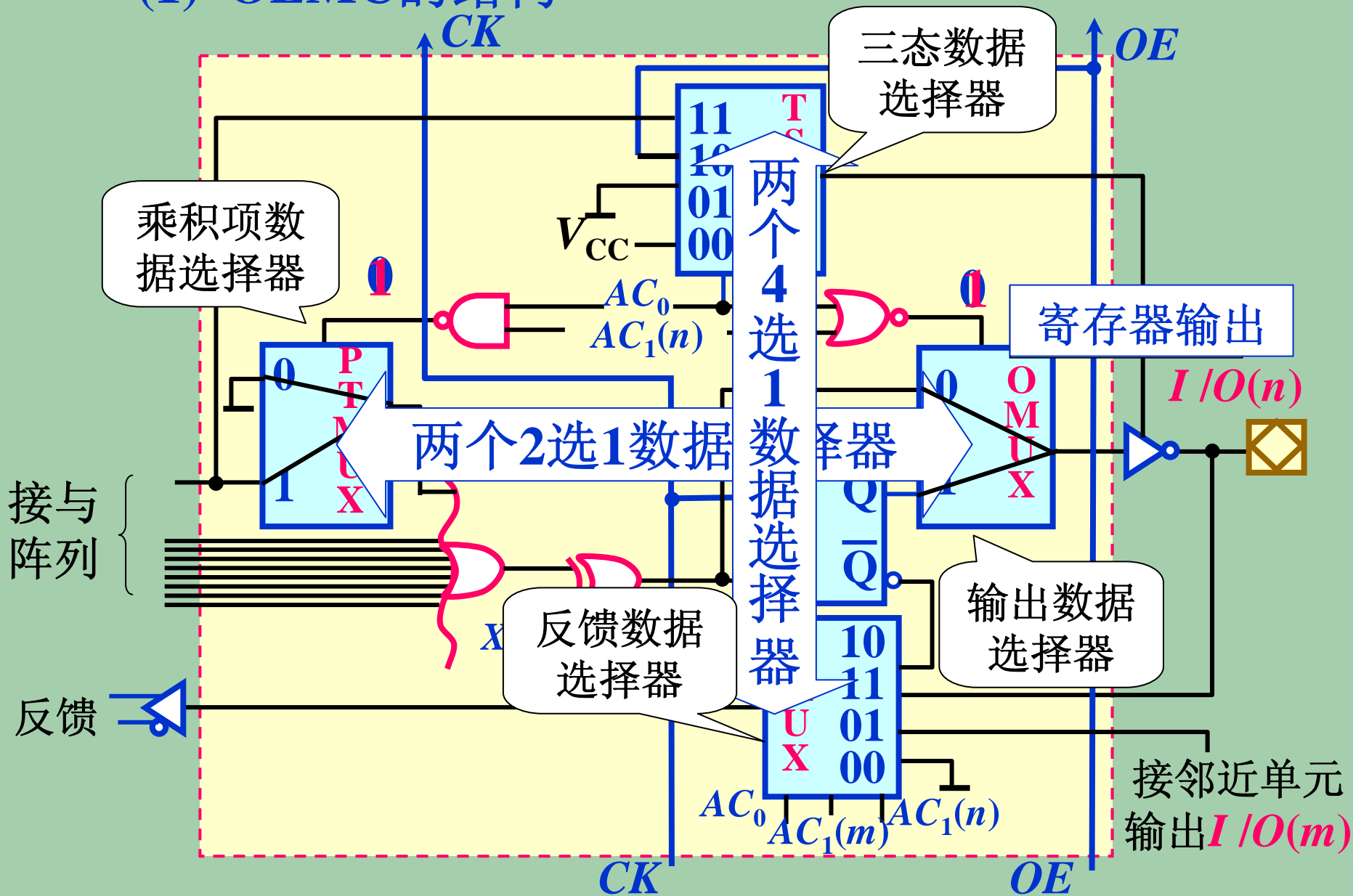
2. 输出逻辑宏单元

输出逻辑宏单元 (OLMC — Out Logic Cell)

- OLMC 有 5 种不同的输出组态
- 5种输出组态由结构控制字来决定
- 通过编程对GAL芯片内部的结构控制字寄存器进行设置

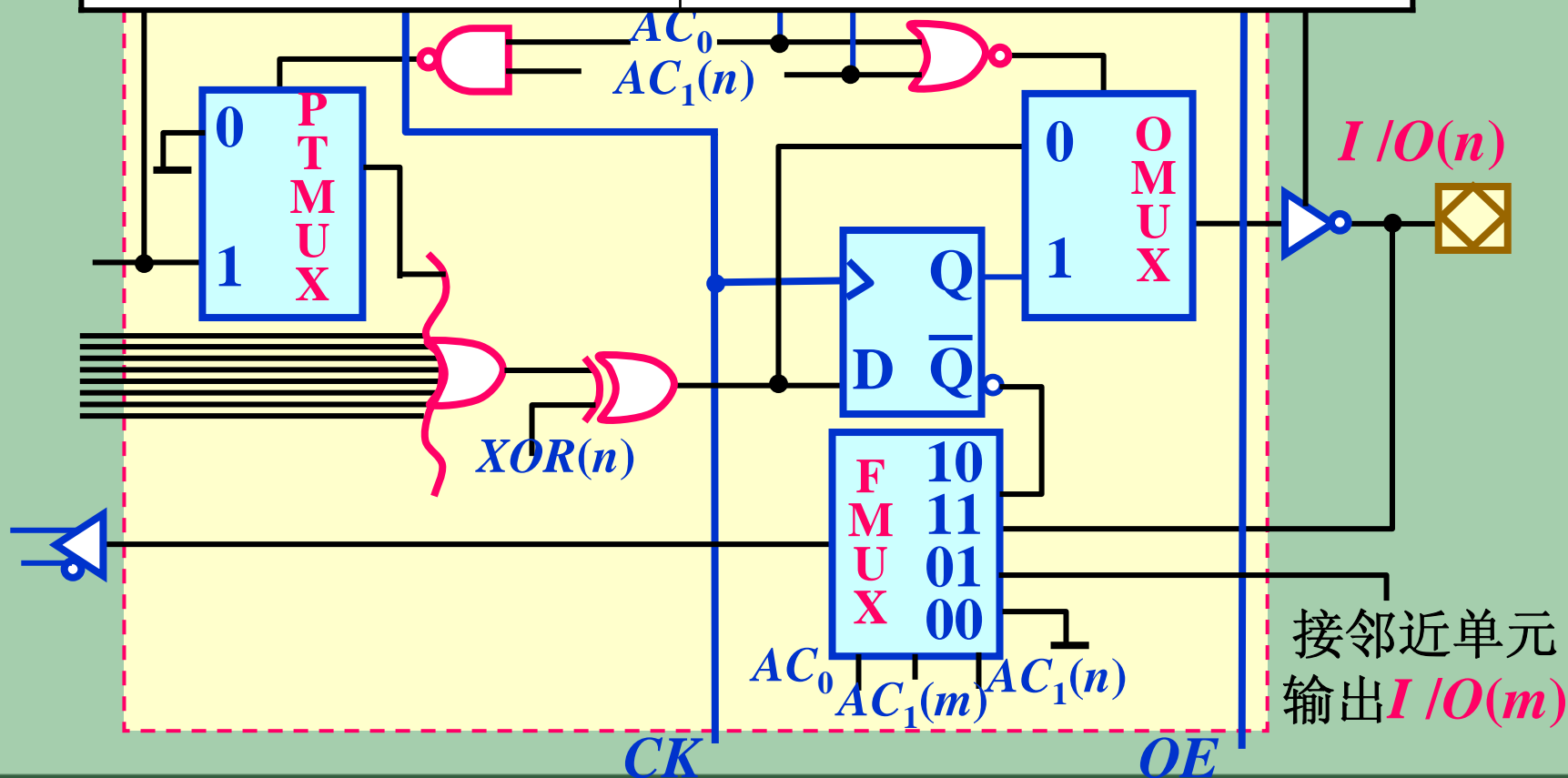


(1) OLMC的结构



(2) FMUX的输出与三个结构控制字的关系

AC_0	$AC_1(n)$	$AC_1(m)$	FMUX的选择
1	0	×	D 触发器的 Q
1	1	×	本单元输出 $I/O(n)$
0	×	1	邻近单元输出 $I/O(m)$
0	×	0	地





(3) OLMC 的输出组态

SYN	AC_0	$AC_1(n)$	功 能	注
0	0	0		不用
0	0	1		不用
0	1	0	寄存器输出	纯时序输出
0	1	1	组合与寄存器输出	本宏单元为组合输出，一个以上宏单元寄存器输出
1	0	0	纯组合输出	无内部反馈和使能控制
1	0	1	纯输入方式	输入为 $I/O(m)$ 三态门禁止
1	1	0		不用
1	1	1	组合输出	组合 I/O 输出, 乘积项 P_1 控制输出使能



3. GAL的主要特点

(1) 通用性强

- 每一个OLMC均可组态成组合或时序电路
- 输入引脚不够时可将OLMC组合成输入端
- 可构成较复杂的时序电路

(2) 100%可编程

- 可重复擦写上百次甚至万次, PAL为一次编程

(3) 100%可测试

(4) 隐含成本低

- 与原始成本大致相同



4. 几种常见的GAL器件

型 号	与阵列规模 (乘积项×输入项)	OLMC 最大输出数	特 点
GAL16V8	64×32	8	普通型
GAL20V8	64×40	8	普通型
isp GAL16Z8	64×32	8	可擦写万次
GAL39V18	64×78	10	与、或阵列 均可编程



三、高密度可编程逻辑器件**HDPLD**

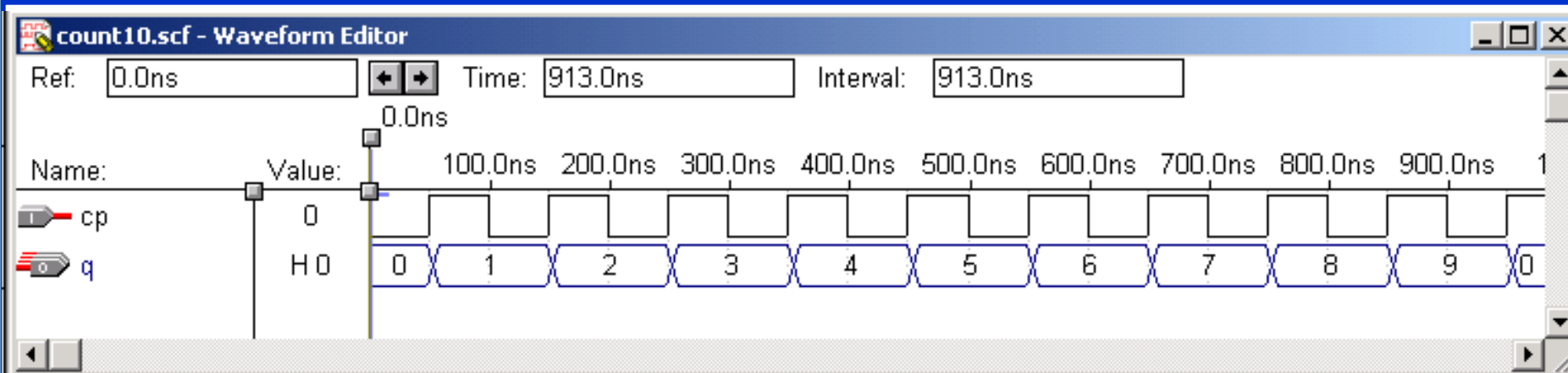
是一种高密度、高性能的超大规模集成电路

分类	{	阵列型 HDPLD	在 GAL 基础上发展起来 主体为与、或阵列
		单元型 HDPLD	由许多逻辑宏单元组成阵列

四、**PLD**编程

5.5.2 时序逻辑电路的**VDHL**描述及仿真

[例5.5.1] 十进制计数器的VHDL描述及仿真



ARCHITECTURE one OF count10 IS

SIGNAL count :STD_LOGIC_VECTOR(3 DOWNT0 0) ;

BEGIN

PROCESS (cp)

BEGIN

IF cp'EVENT AND cp='1' THEN

IF count <="1001" THEN

count ="0000";

ELSE count <= count +1;

END IF;

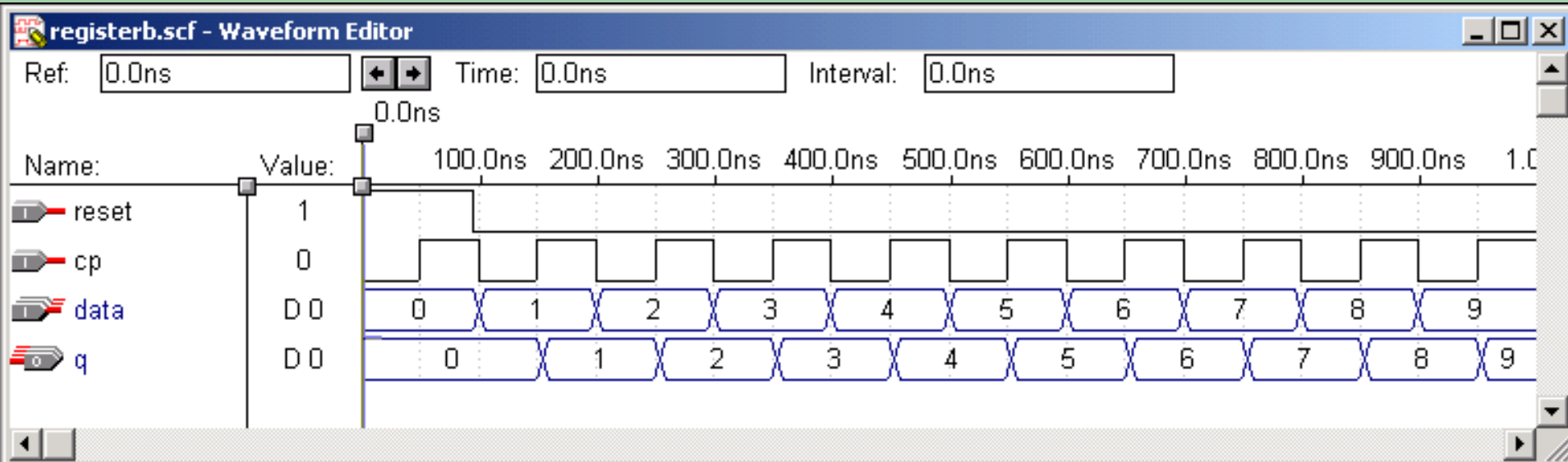
END IF;

END PROCESS;

q<= count;

END one;

[例5.5.2] 4位基本寄存器的VHDL描述及仿真



```
BEGIN
PROCESS (cp)
BEGIN
    IF cp'EVENT AND cp='1' THEN
        IF reset='1' THEN
            q<="0000";
        ELSE
            q<= data;
        END IF;
    END IF;
END PROCESS;
END one;
```