

第5章 可编程逻辑

5.1 PLD的基本概念

5.2 现场可编程门阵列FPGA

5.3 在系统可编程ISP

5.4 可编程逻辑的原理图方式设计

5.5 可编程逻辑的VHDL文本方式设计



[返回目录](#)

5.1 PLD的基本概念

5.1.1 可编程阵列

5.1.2 PLD的类型

导入

数字电路经历了分离元件 --> 中小规模标准化集成电路 --> 可编程逻辑器件 (PLD) 这样的发展历程。PLD是用户根据需要自行设计芯片中特定逻辑电路的器件。

5.1.1 可编程阵列

所有的PLD是用可编程阵列组成的。

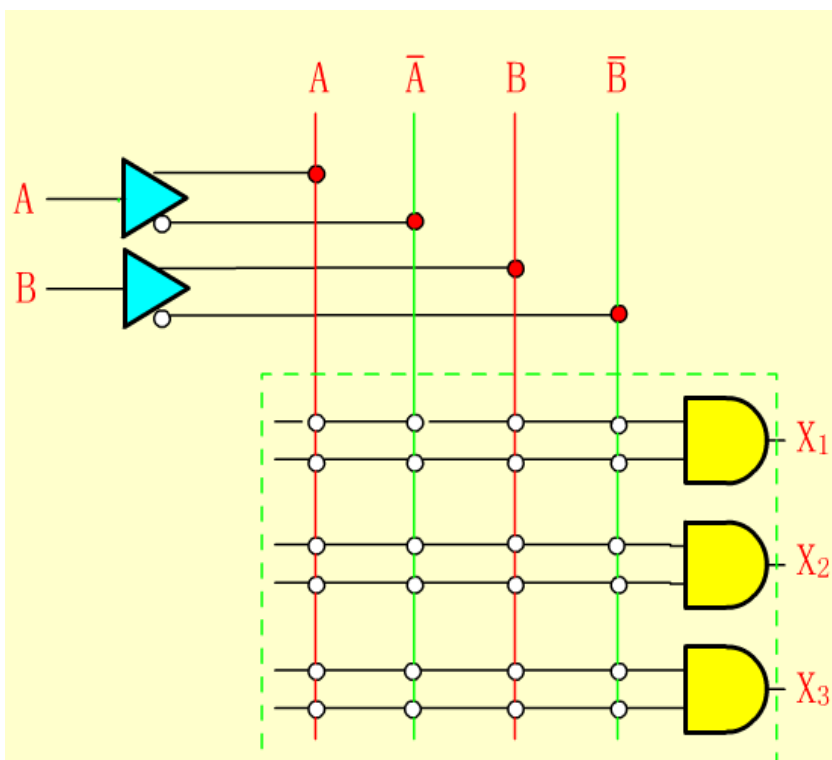
可编程阵列本质上是行、列导线组成的导电网格。

在网格的交叉点上，通过熔断金属丝或E²COMS管等连接技术来实现逻辑 **1** 或逻辑 **0** 。

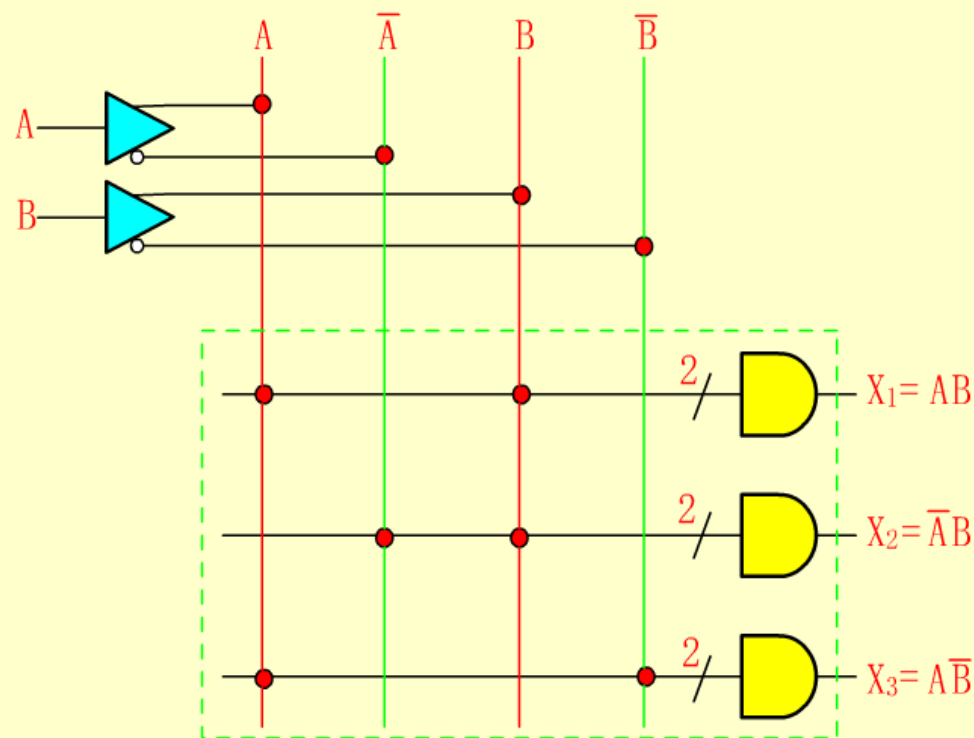
可编程阵列分类：与阵列和或阵列

5.1.1 可编程阵列

1. 与阵列



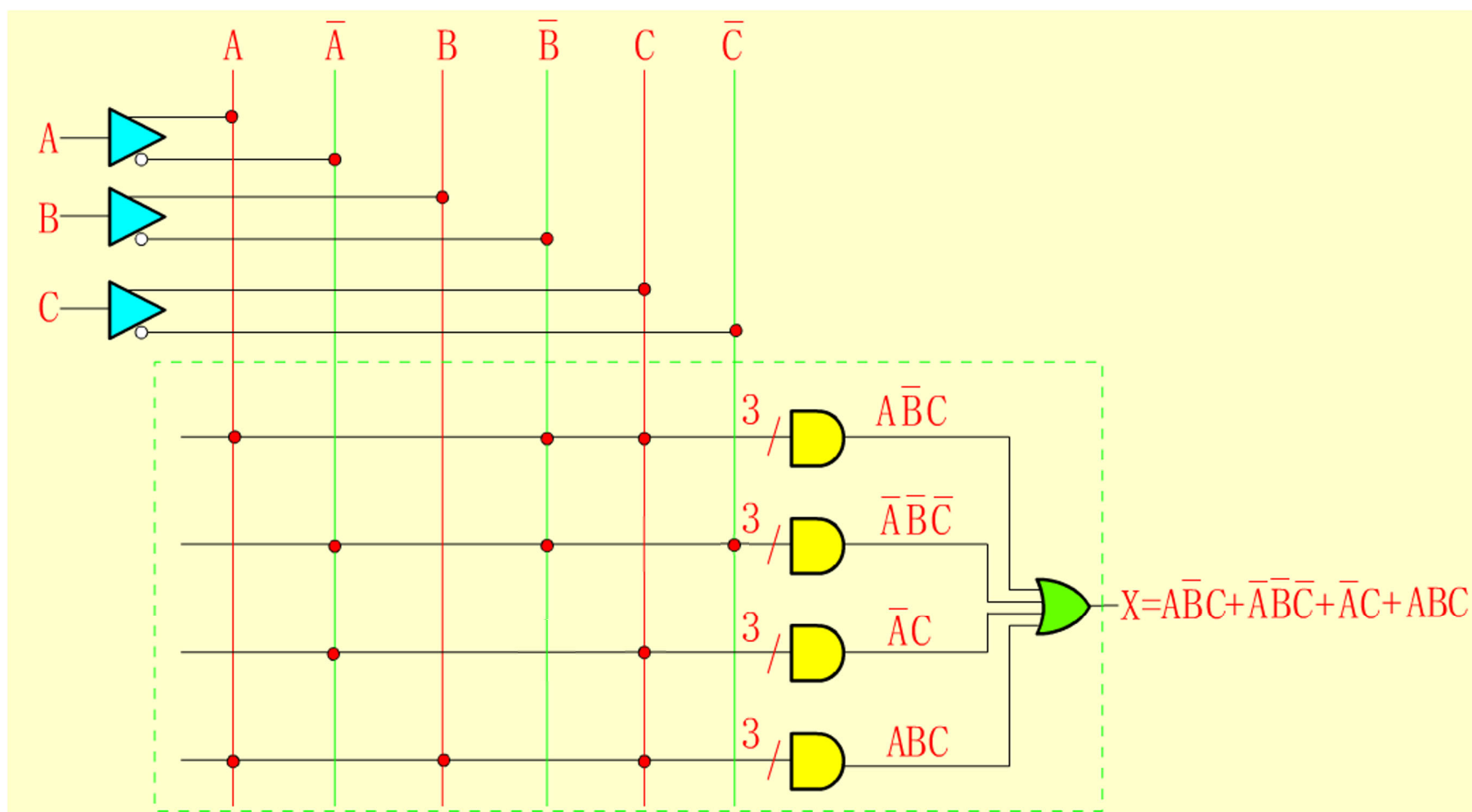
(a) 未编程



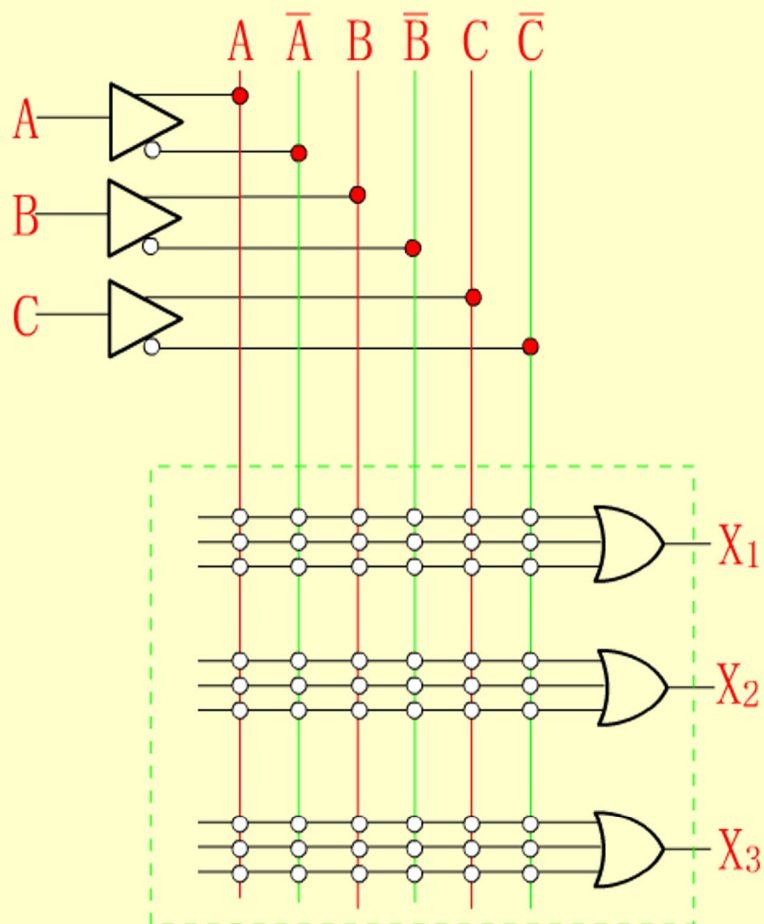
(b) 已编程

【例1】 用与阵列实现三变量A、B、C的与或表达式

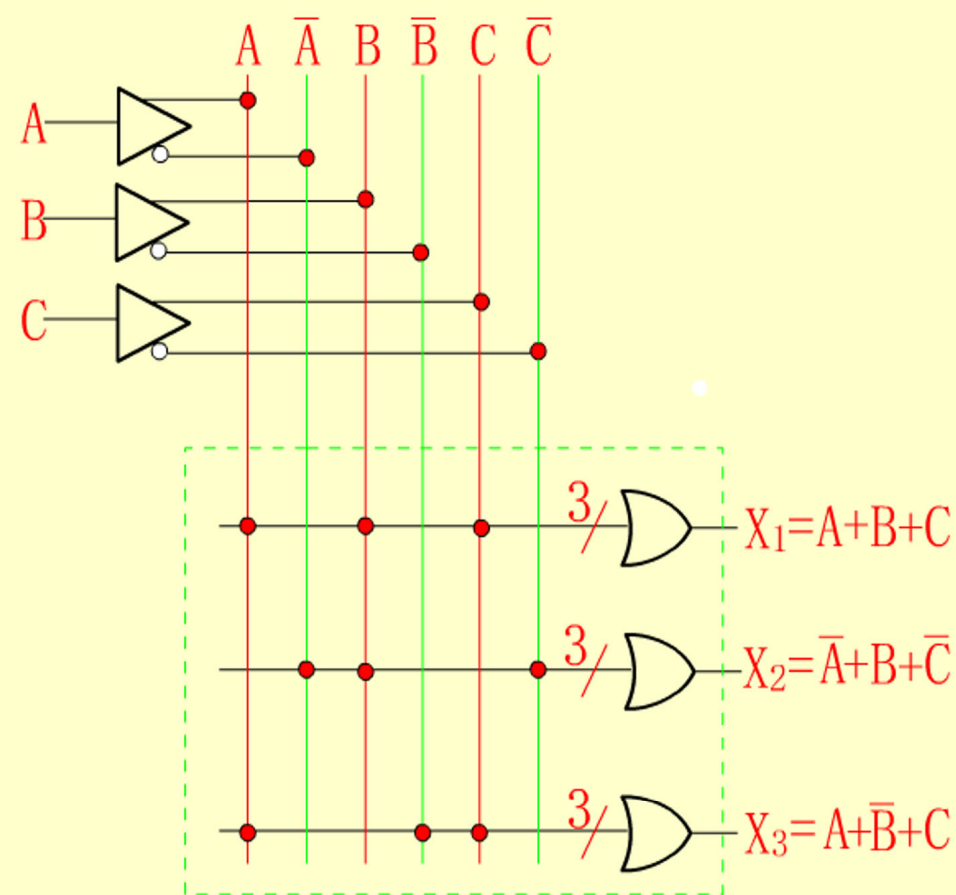
$$X = A\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}C + ABC$$



2. 或阵列

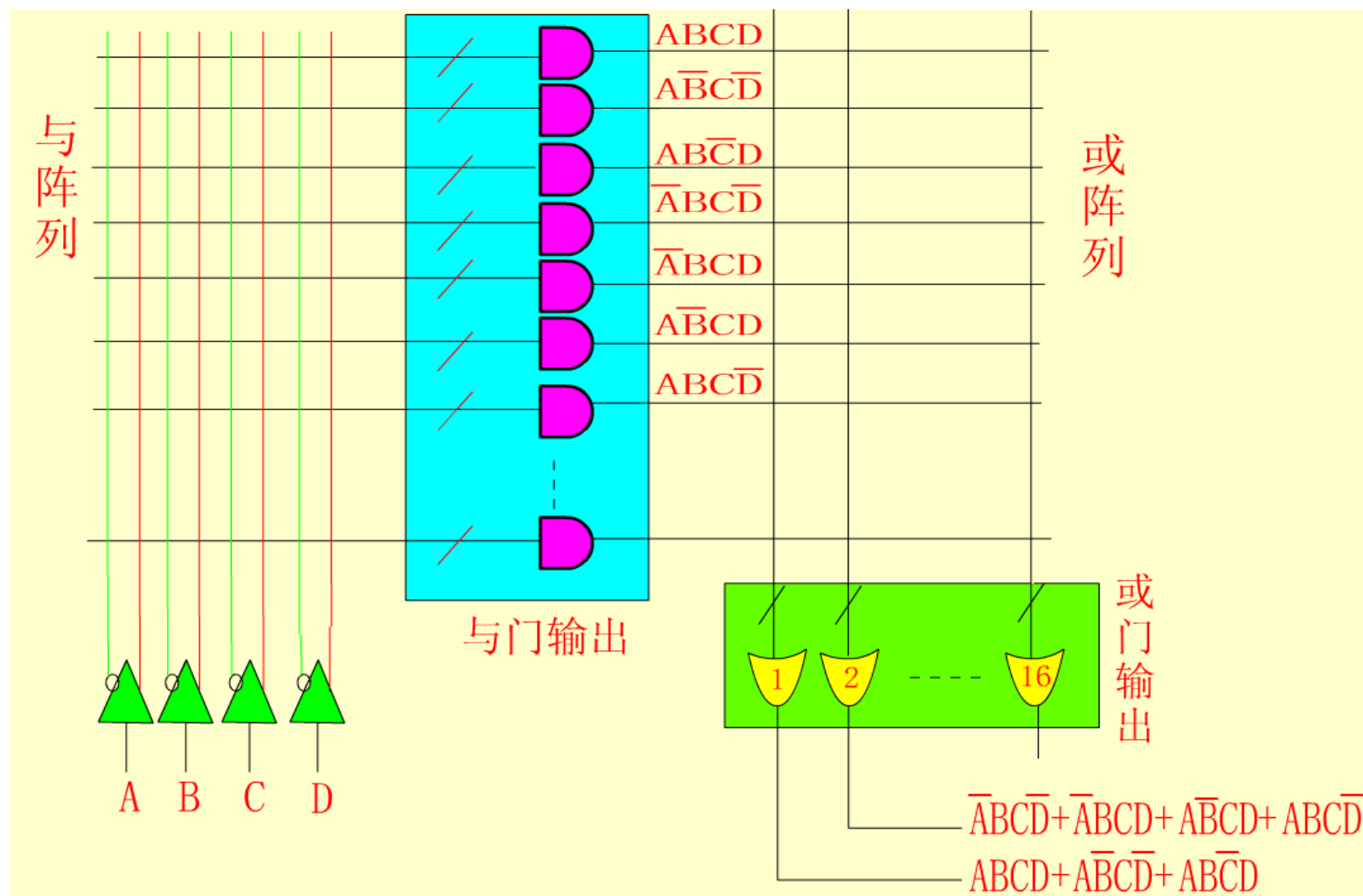


(a) 未编程



(b) 已编程

【例3】图5.4是一个与阵列和一个或阵列组成的SOP形式复合阵列，或阵列制定的逻辑表达式由或门1、或门2输出。请在图中画出编程点，并写出与门输出的表达式。



3.可编程连接技术

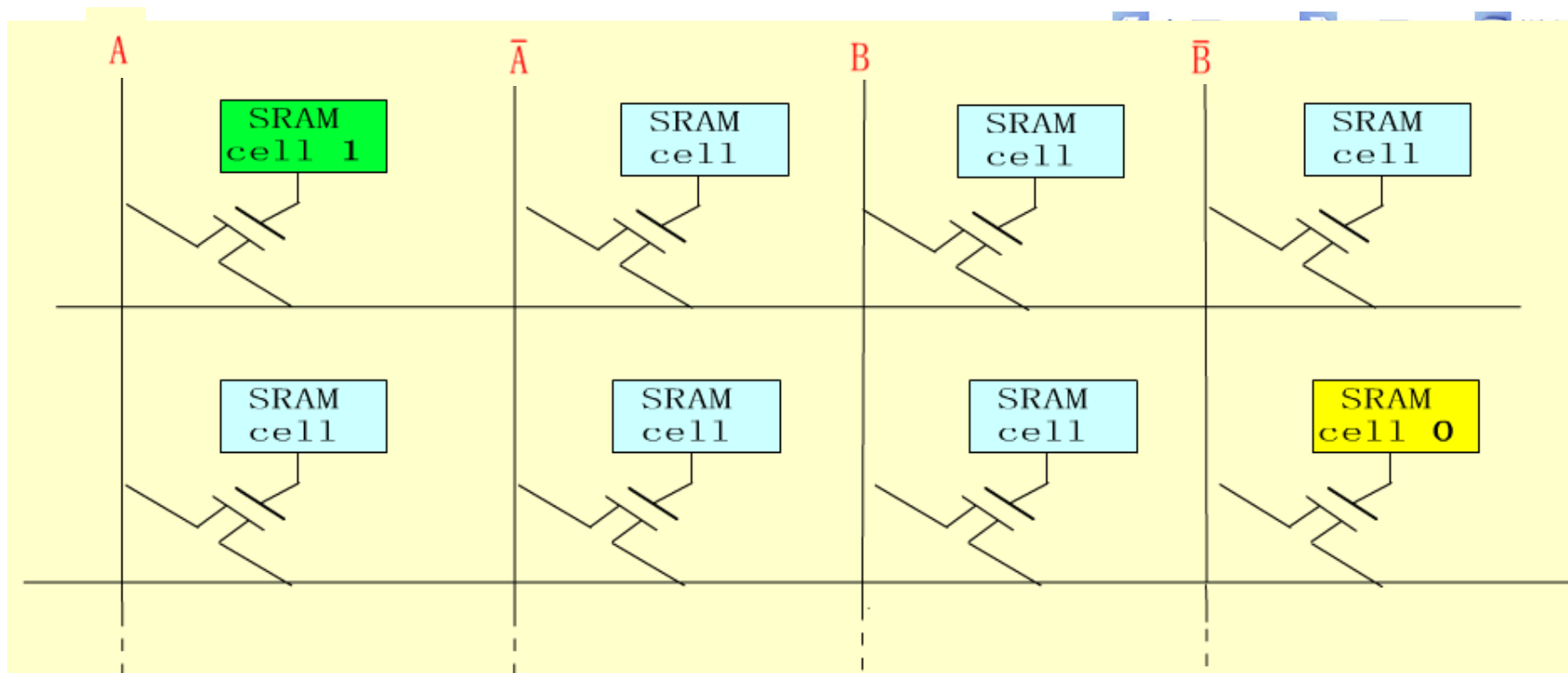
连接技术有以下4种：

熔丝技术, 编程之前, 熔丝连接（原始），状态称为逻辑 1；对选定熔丝**熔断**后的状态称为逻辑 0。**非易失**。

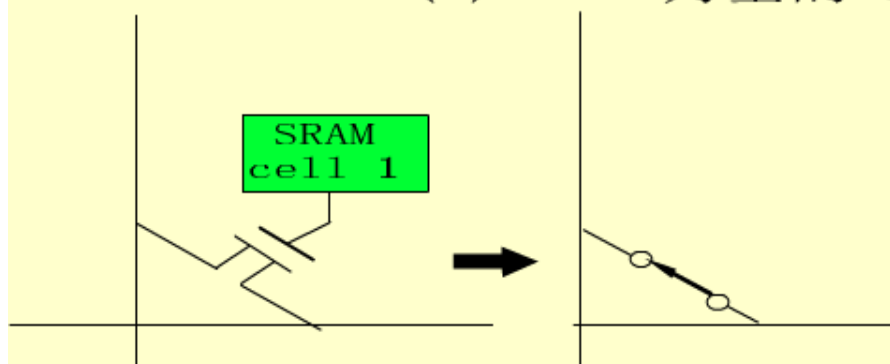
反熔丝技术, 这种连接与熔丝连接正好相反，不是破坏连接，而是建立连接。**非易失**。

E²PROM技术, 加电的方式可以擦除或重写，ISP在系统编程

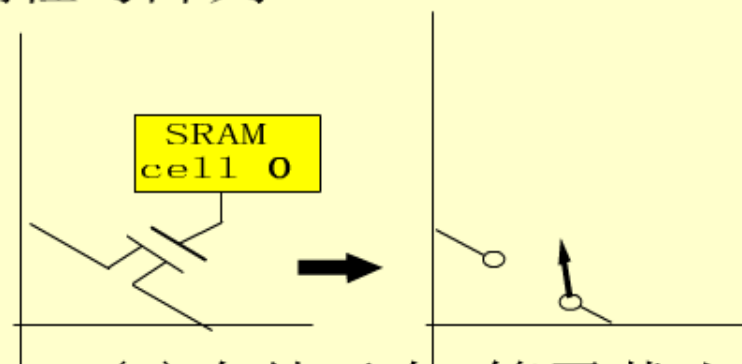
SRAM技术, SRAM存储元通过触发器实现连接、断开。**易失**，每次加电，PLD编程数据必须**重新写入**。



(a) SRAM为基的可编程与阵列



(b) 存储元存1管子导通



(c) 存储元存0管子截止

SRAM 为基的与阵列

5.1.2 PLD的类型

1.简单可编程逻辑器件SPLD

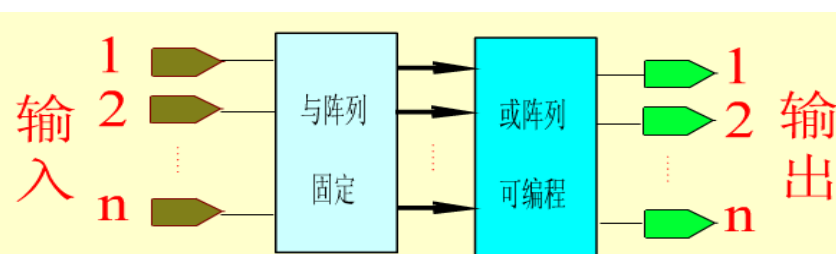
SPLD是指简单可编程逻辑器件。根据与阵列、或阵列是否可以编程，分类：

PROM（一次可编程只读存储器）；

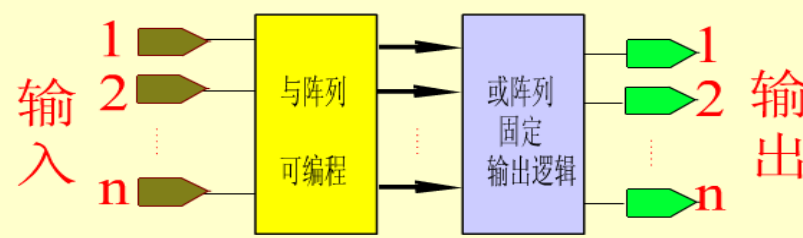
PLA（可编程逻辑阵列）；

PAL（可编程阵列逻辑）；

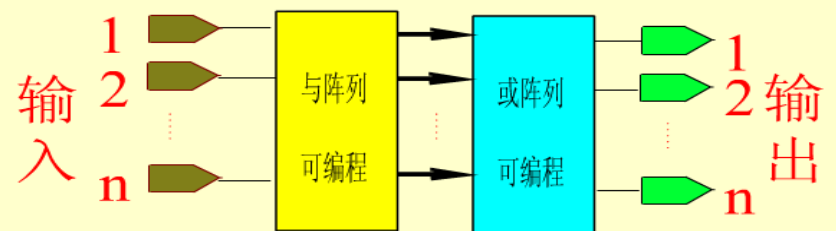
GAL（通用阵列逻辑）。



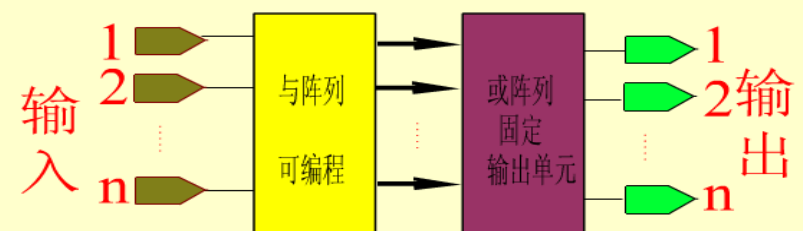
(a)PROM



(b)PAL



(c)PLA

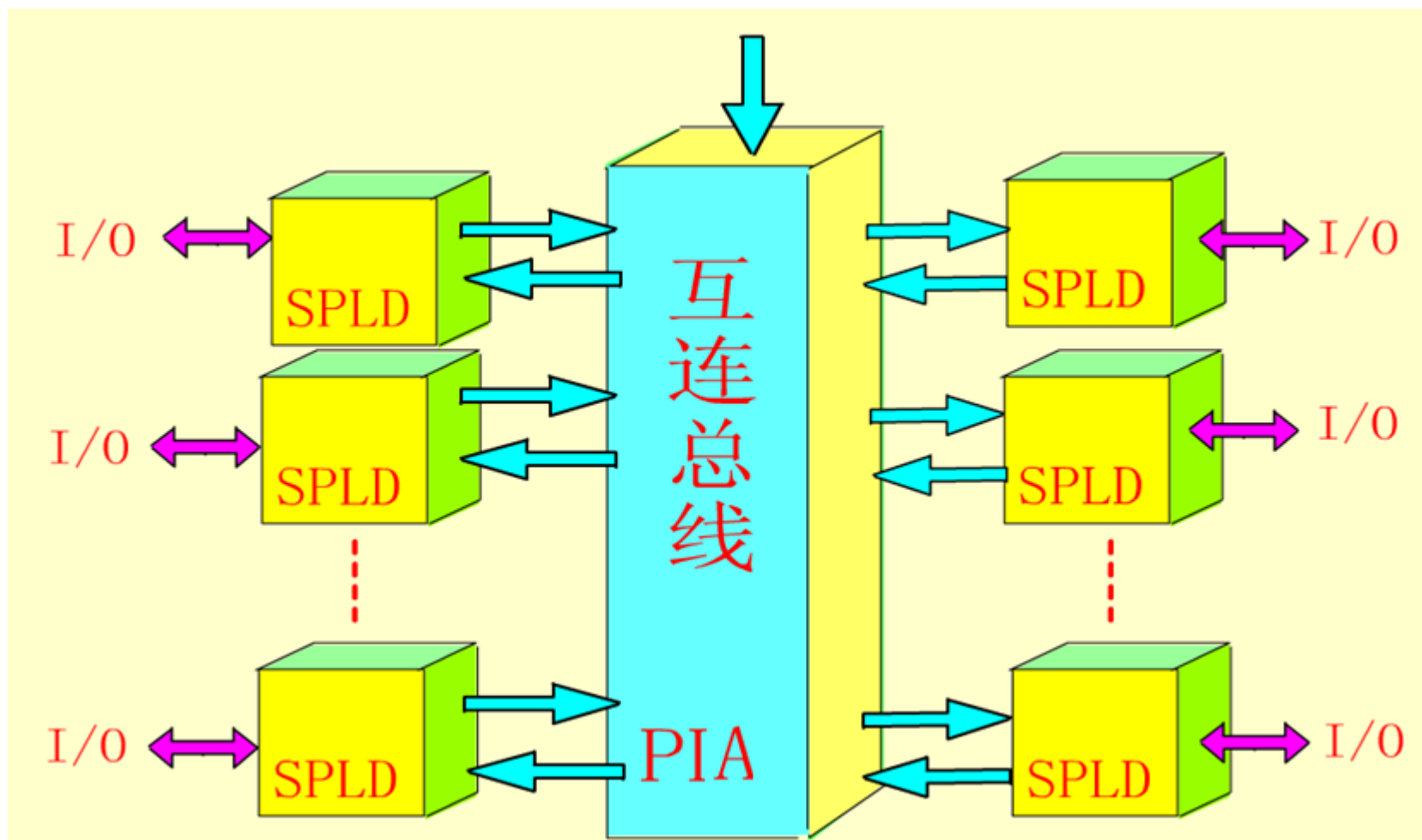


(d)GAL

SPLD内部结构框图

2.复杂可编程逻辑器件CPLD

CPLD本质上是：利用可编程的互连总线连接起来的多路SPLD。



CPLD结构框图

5.2 现场可编程门阵列FPGA

5.2.1 FPGA的基本结构

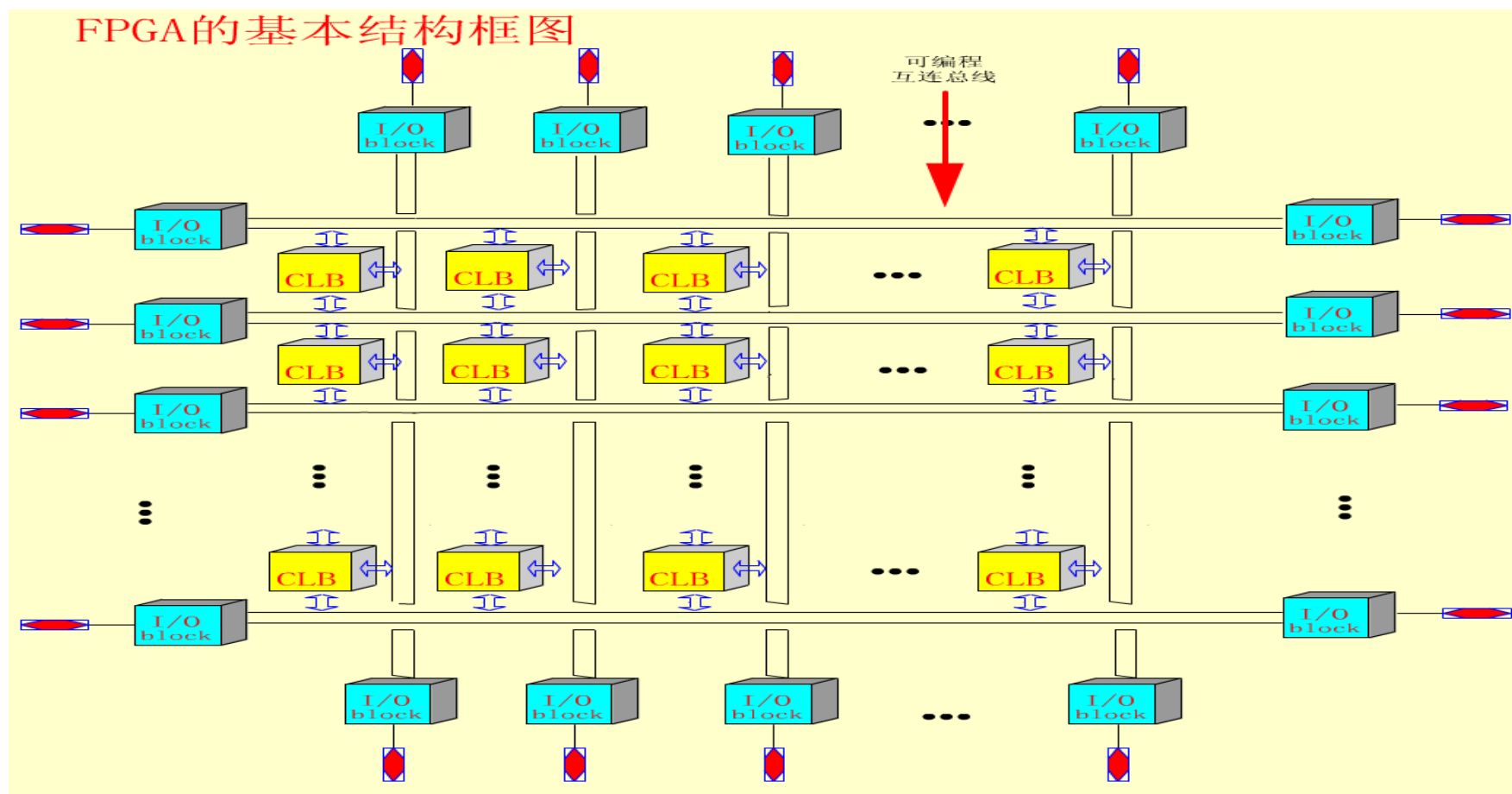
5.2.2 可组态逻辑块CLB

5.2.3 SRAM为基础的FPGA

5.2.1 FPGA的基本结构

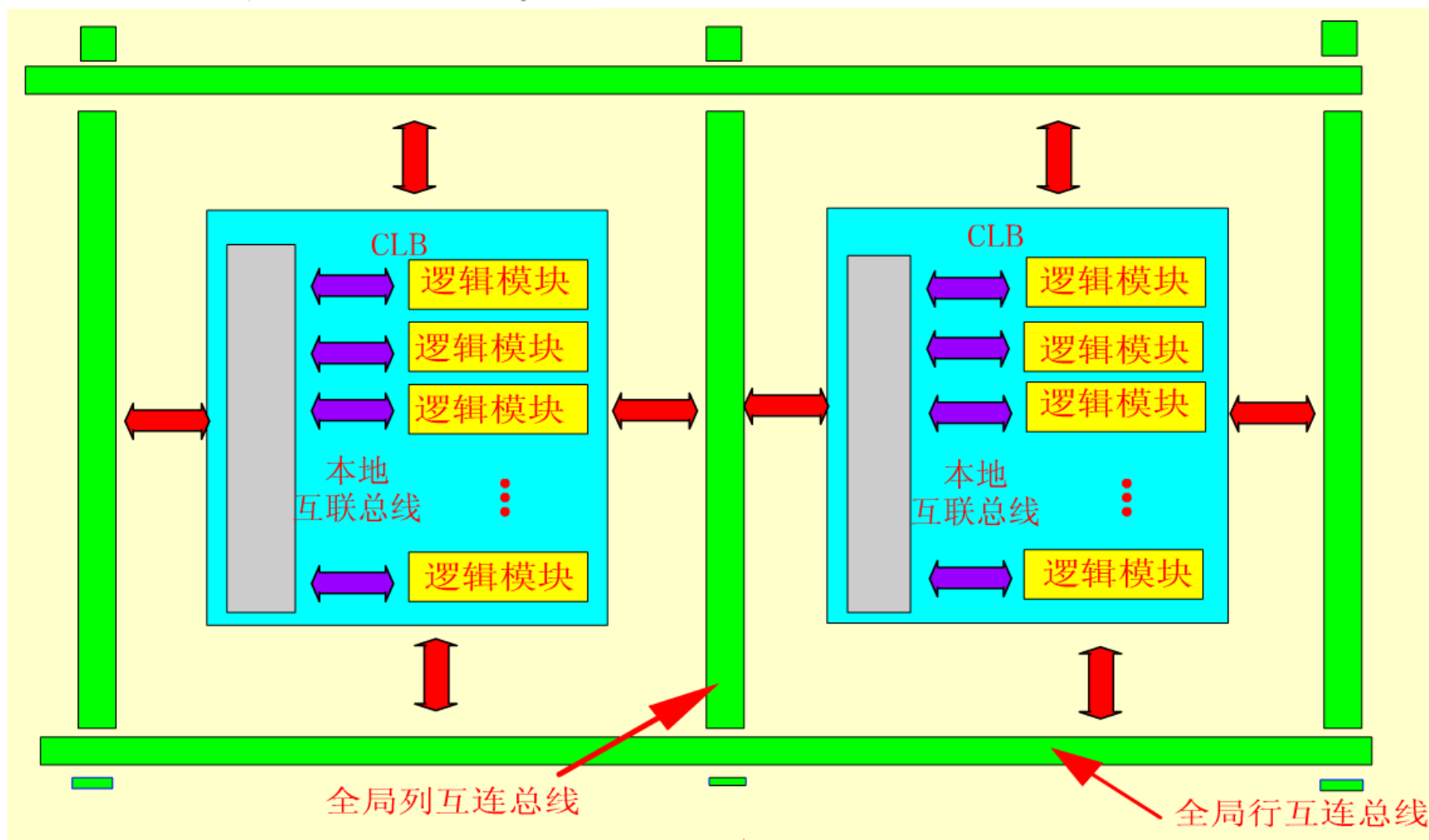
FPGA是现场可编程门阵列的英文缩写，比CPLD还要复杂，逻辑块数量大，体系结构也不同，有细粒度、粗粒度之分

三个基本元素（逻辑块CLB、可编程互连总线、I/O输入输出块）



5.2.2 可组态逻辑块CLB

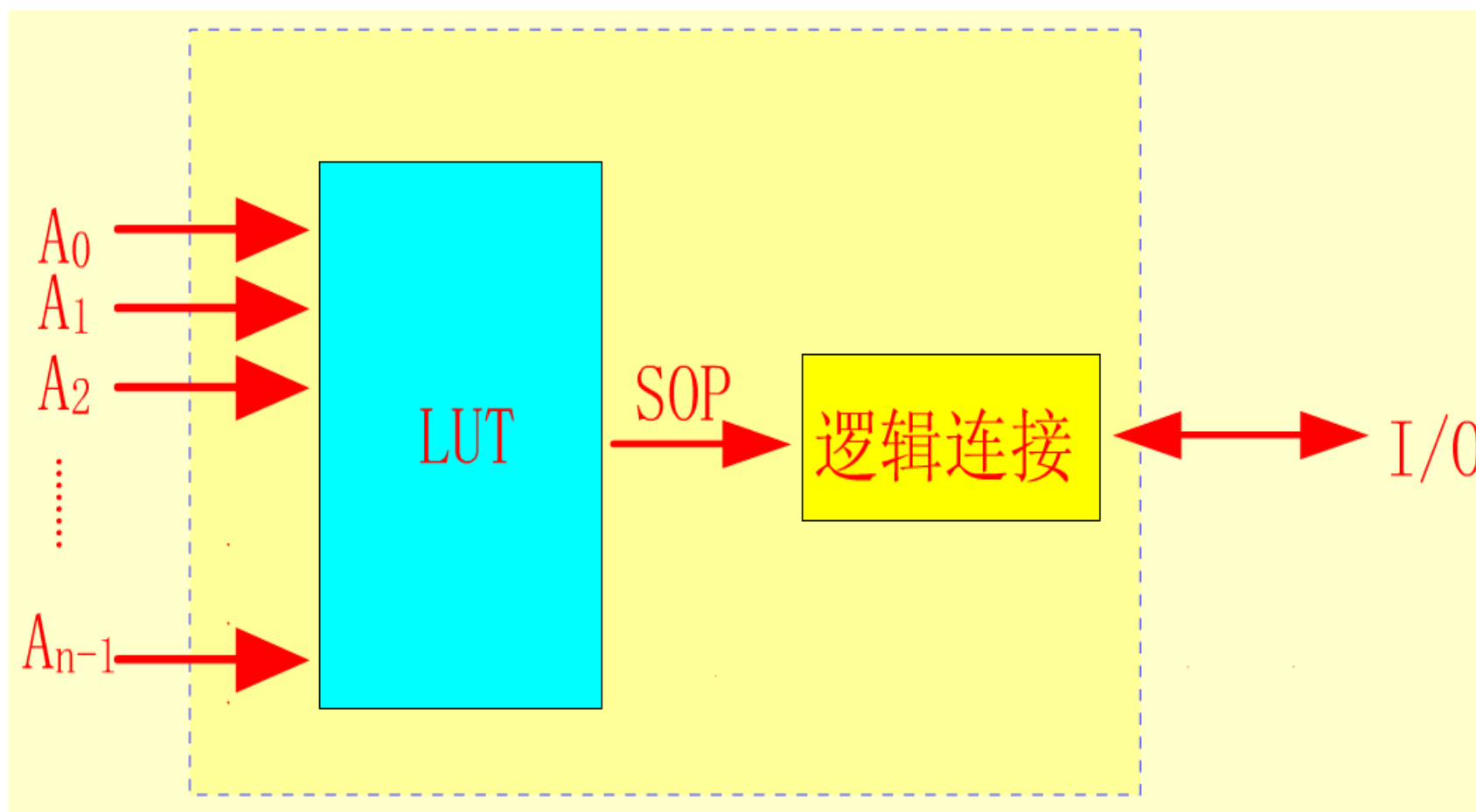
CLB 包含若干个较小的逻辑模块



基本可组态CLB

5.2.2 可组态逻辑块CLB

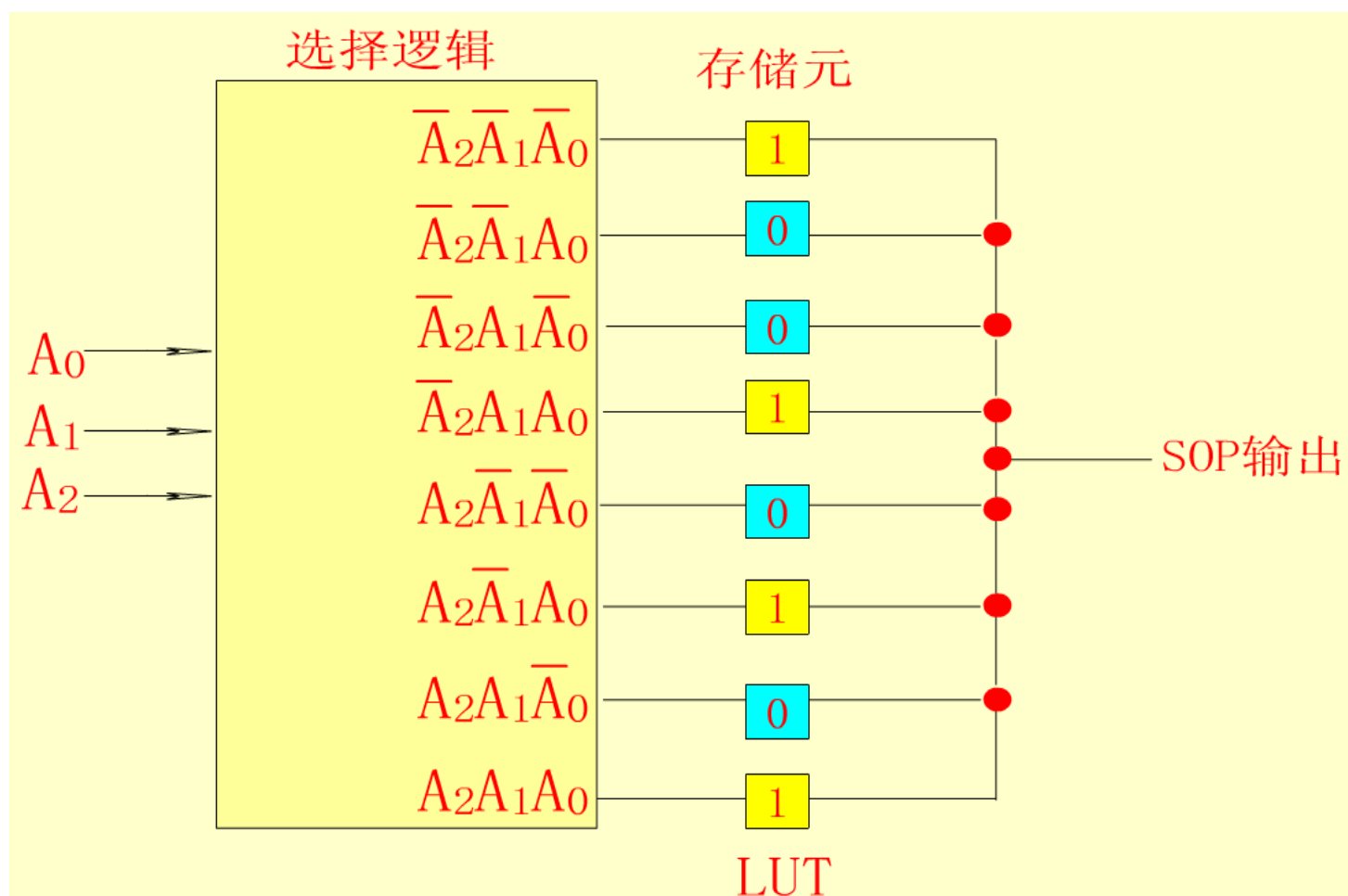
一个用逻辑模块实现的典型LUT（查找表类型的存储器）的框图



FPGA中一个逻辑模块的框图

5.2.2 可组态逻辑块CLB

LUT由数目等于 2^n 的存储元组成



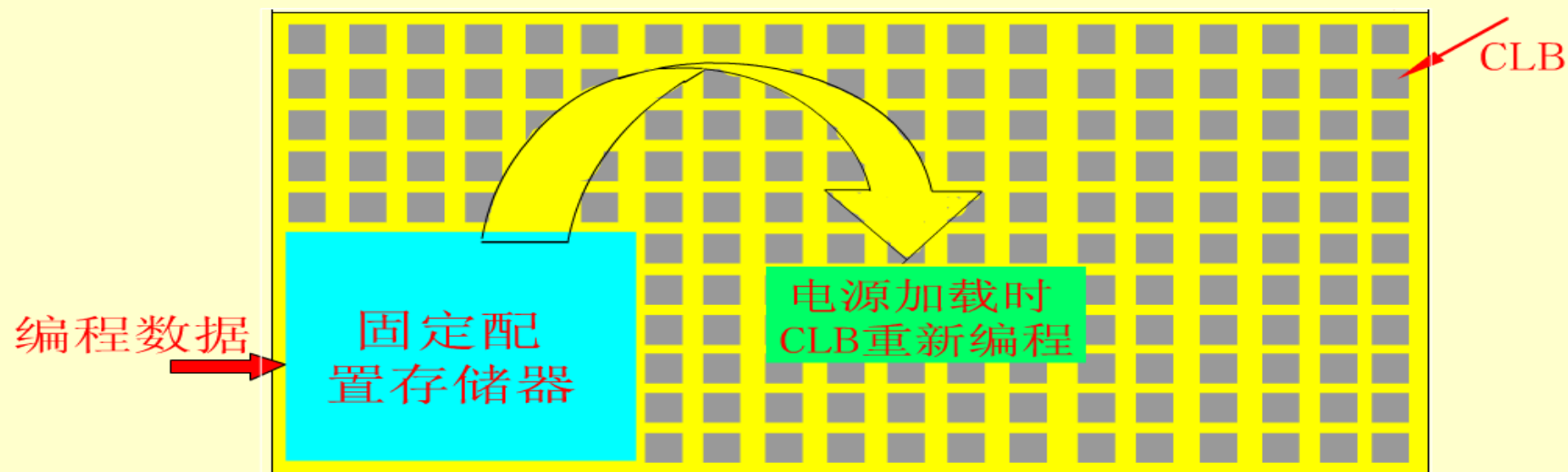
LUT编程后用作SOP表达式输出

5.2.3 SRAM为基础的FPGA

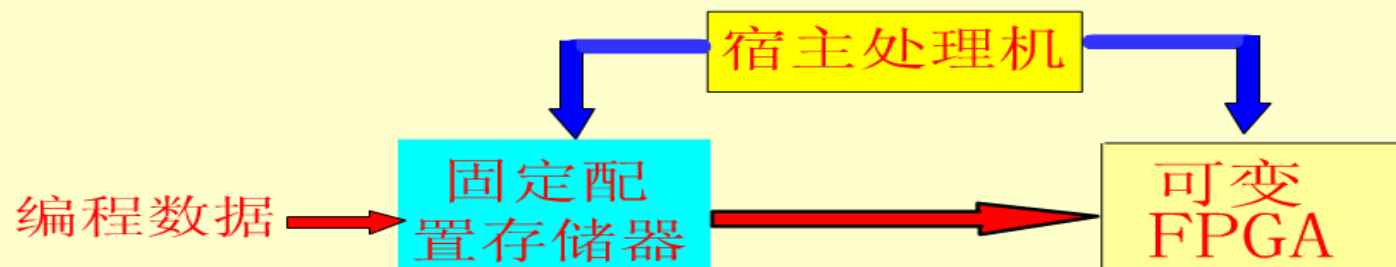
根据制造技术不同，FPGA在体系结构上分可变或不变两种。

‘可变’意味着：当电源关闭时，被编程到组态逻辑块的所有数据将丢失。

可变FPGA配置的基本概念



(a) 可变FPGA利用芯片上的固定配置存储器



(b) 可变FPGA利用板上存储器和宿主处理机

5.4 可编程逻辑的原理图方式设计

5.4.1 编程环境 and 设计流程图

5.4.2 设计输入

5.4.3 功能模拟

5.4.4 综合和实现（软件）

5.4.5 时序模拟

5.4.6 器件下载

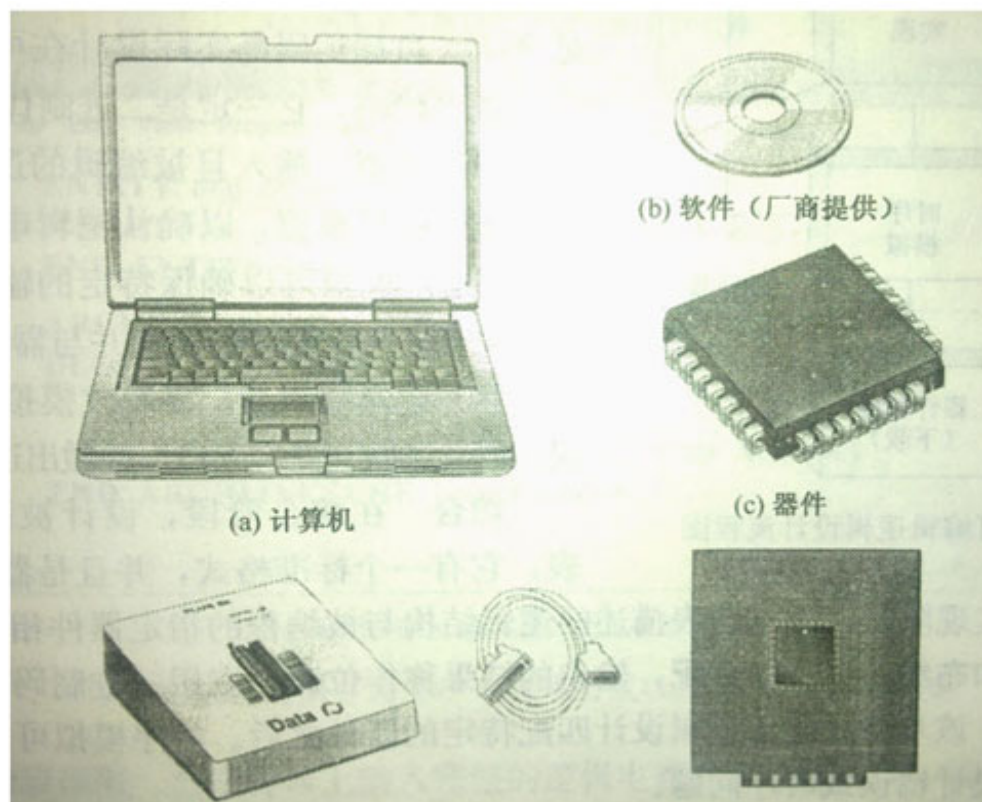
5.4.1 编程环境 and 设计流程图

1. 编程环境

EDA叫电子设计自动化

目标器件的放置采用两种方法:

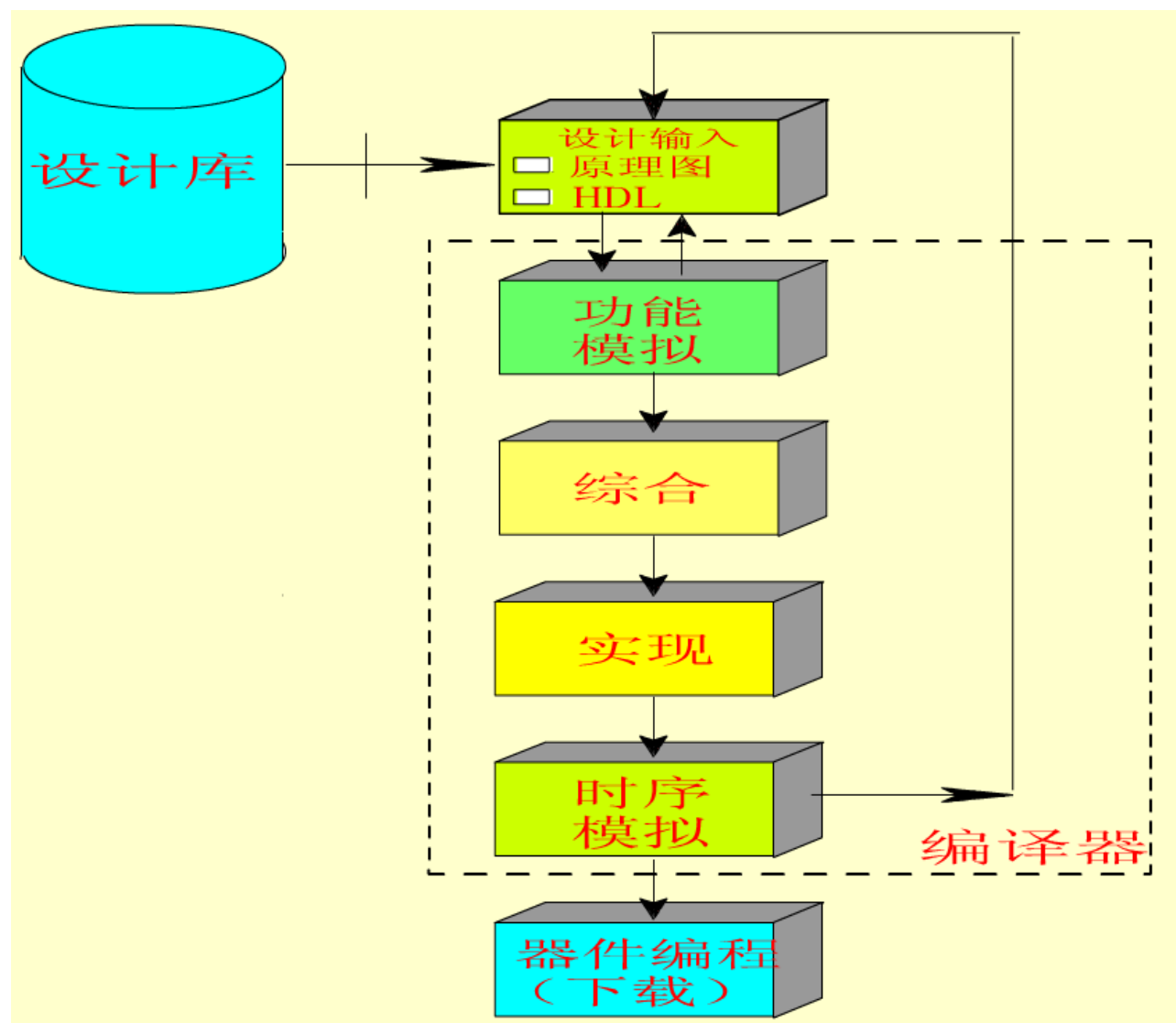
1. 用编程器
2. 用开发板




编程环境

5.4.1 编程环境 and 设计流程图

2. 设计流程图





设计输入：将所设计的电路输入到计算机。

1. 文本方式, 如: 用硬件描述语言完成  逻辑设计
2. 原理图方式

编译状态：源代码 目标代码（二进制代码）

功能模拟：确认逻辑设计实现预期的功能，利用波形编辑器

综合：设计被翻译成 网表Netlist

实现：网表描述的逻辑结构 被编程的指定器件

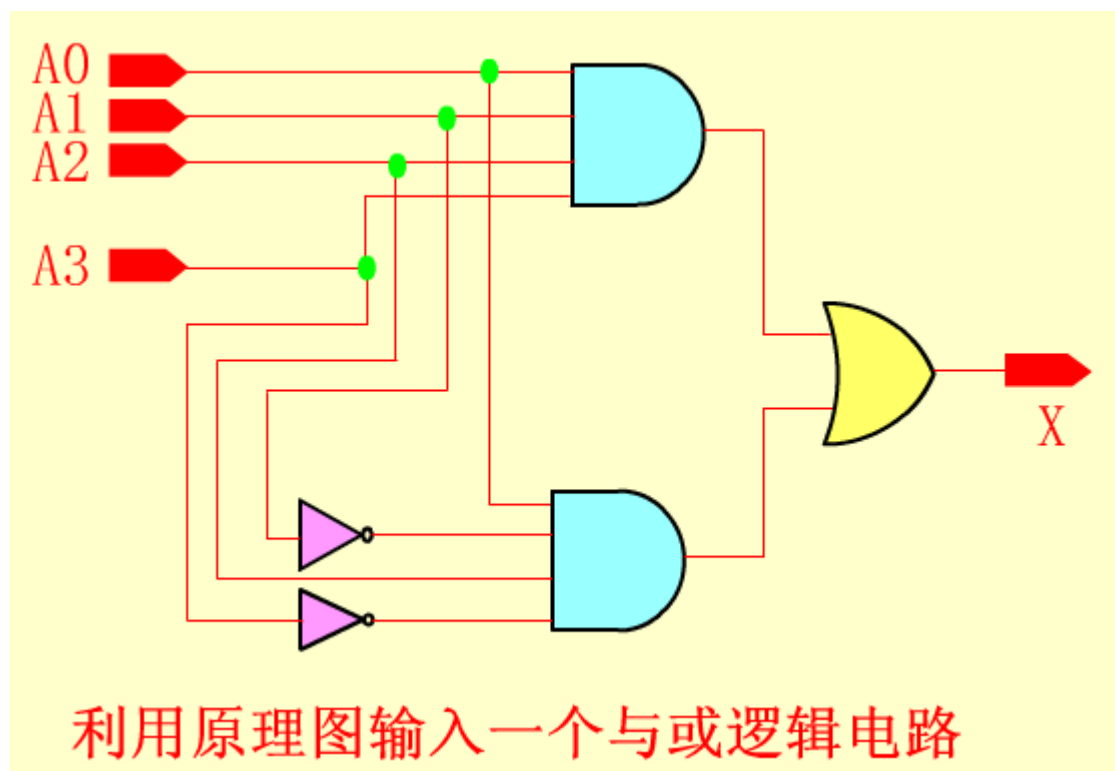
时序模拟：确保没有导致传播延迟的设计错误或时序问题

下载：位流下载到器件上

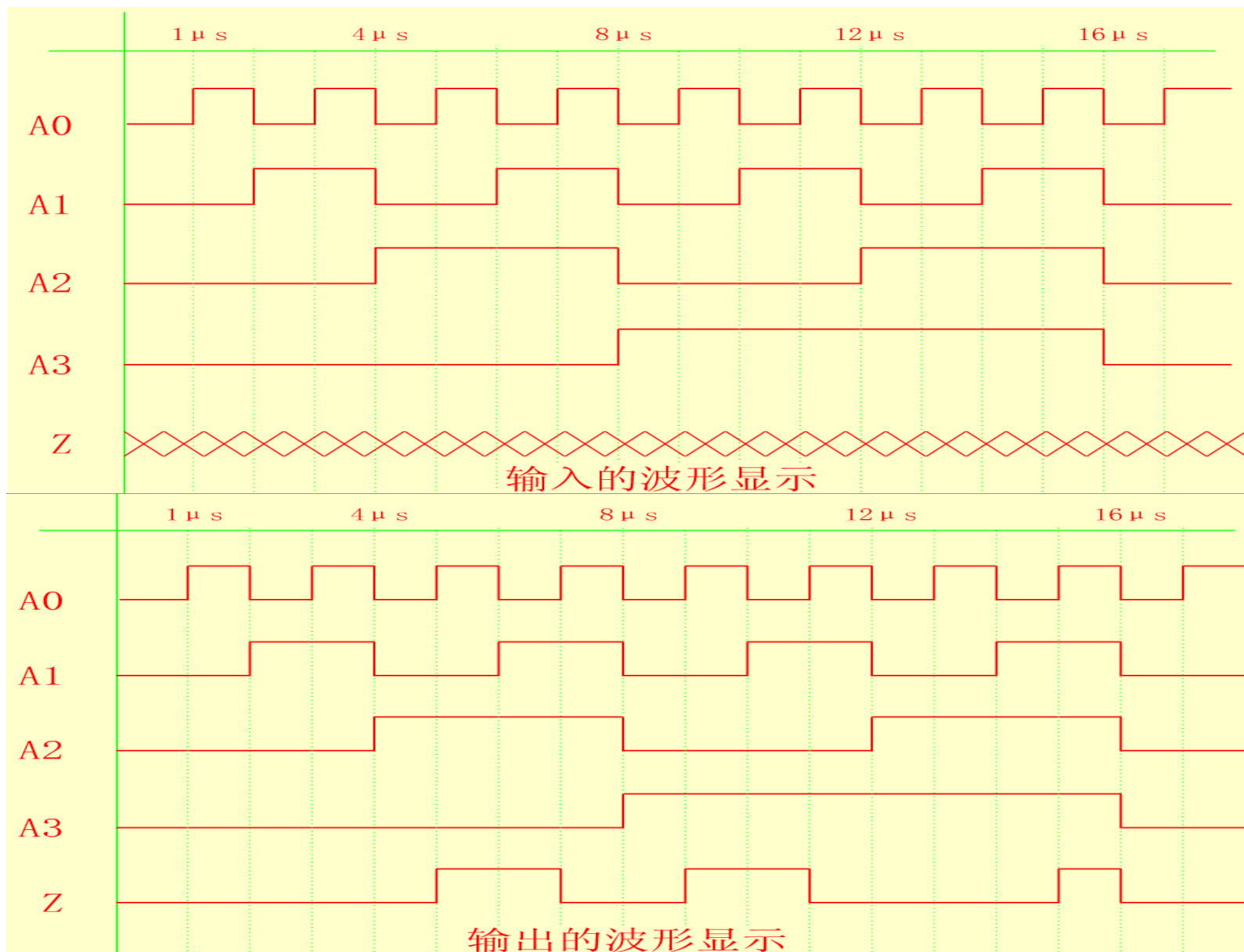
5.4.2 设计输入

可编程逻辑器件中实现的逻辑电路设计，可采用两种基本方式之一进行输入：

1. **文本输入**：用**硬件描述语言**完成，通用性好
2. **原理图输入**：可视化逻辑元件拖放、连接，式直观简单，难做大。



5.4.3 功能模拟



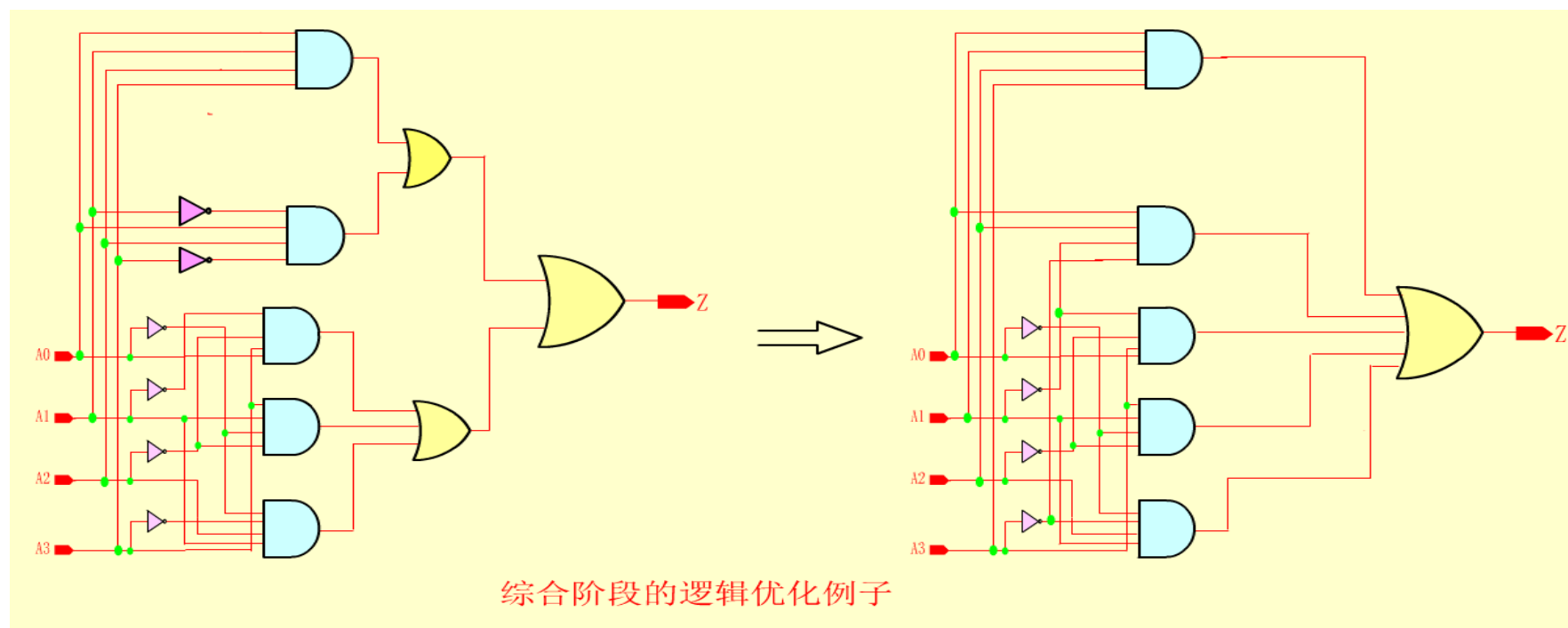
5.4.4 综合和实现（软件）

1. 综合阶段

优化：门的数量最小化

综合阶段输出：网表。

网表： 由综合软件生成，它基本上是一个描述元件和它们怎样相互连接在一起的连接表。



5.4.4 综合和实现（软件）

2.实现阶段

完成网表描述的逻辑结构被编程的指定器件的映射：

使设计和器件自身体系结构、引脚配置的特定目标器件相适应。（设置和选径，简称适配）。

所有可能用到的目标器件的完整数据，通常保存在软件库中。

5.4.5 时序模拟

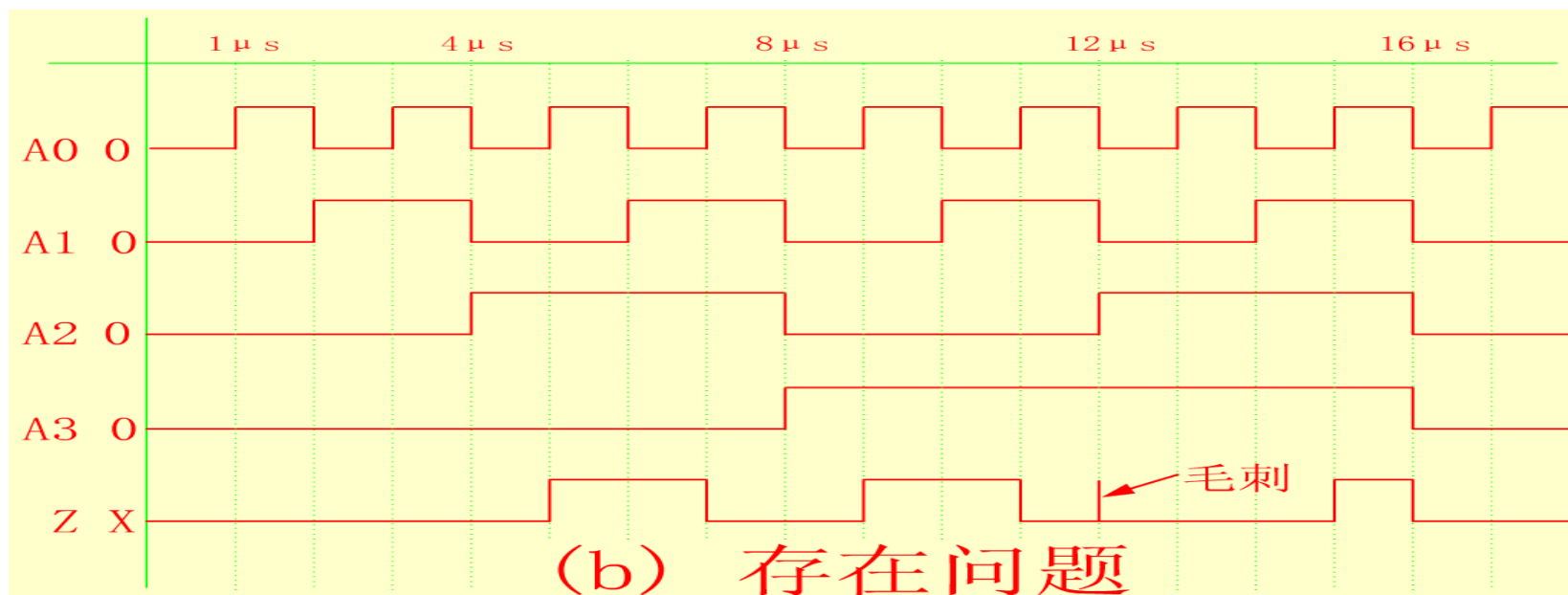
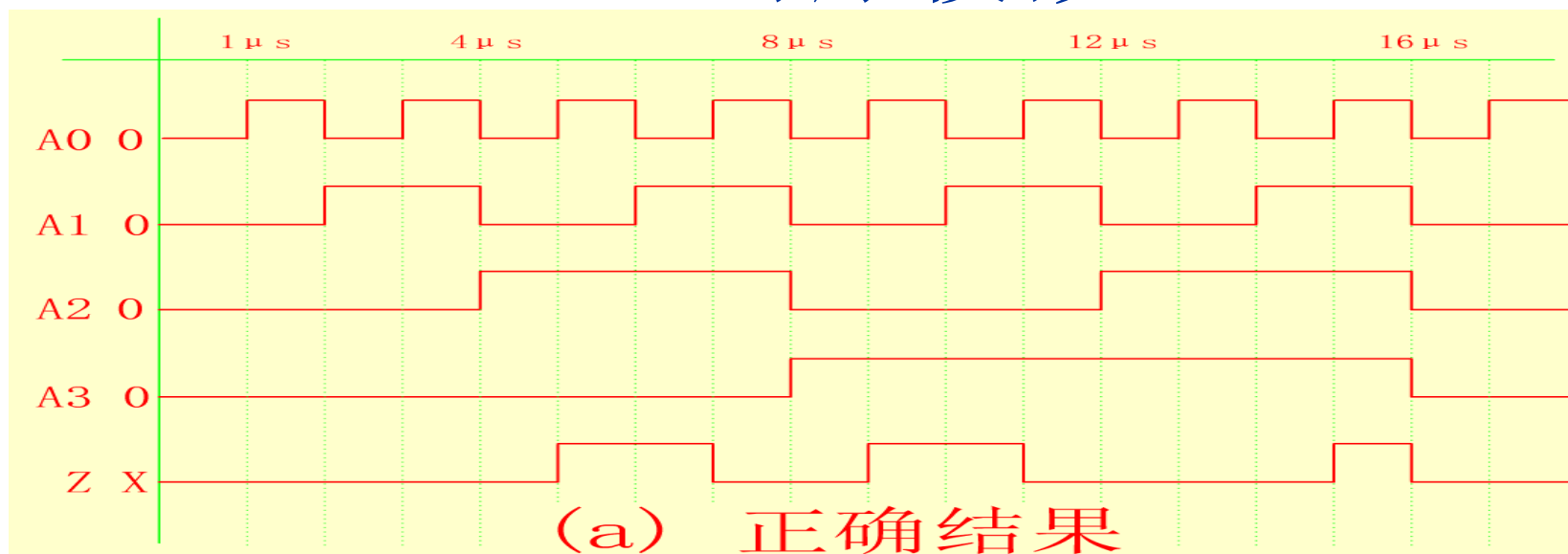
功能模拟已经通过了，但考虑特定目标器件的信息（如门的传输延迟），可能存在时序问题。

注：功能模拟中不需要选定目标器件。

时序模拟的目的：

确保电路以设计频率工作，且没有传输延迟或其他影响全局操作的时序问题。

5.4.5 时序模拟



5.4.6 器件下载

产生二进制码表示的位流后，发送到目标器件进行自动配置
设计已放置在目标器件上，并且可以在电路上进行测试。

