

# Unit 13

## —Programmable Logic Devices

张彦航

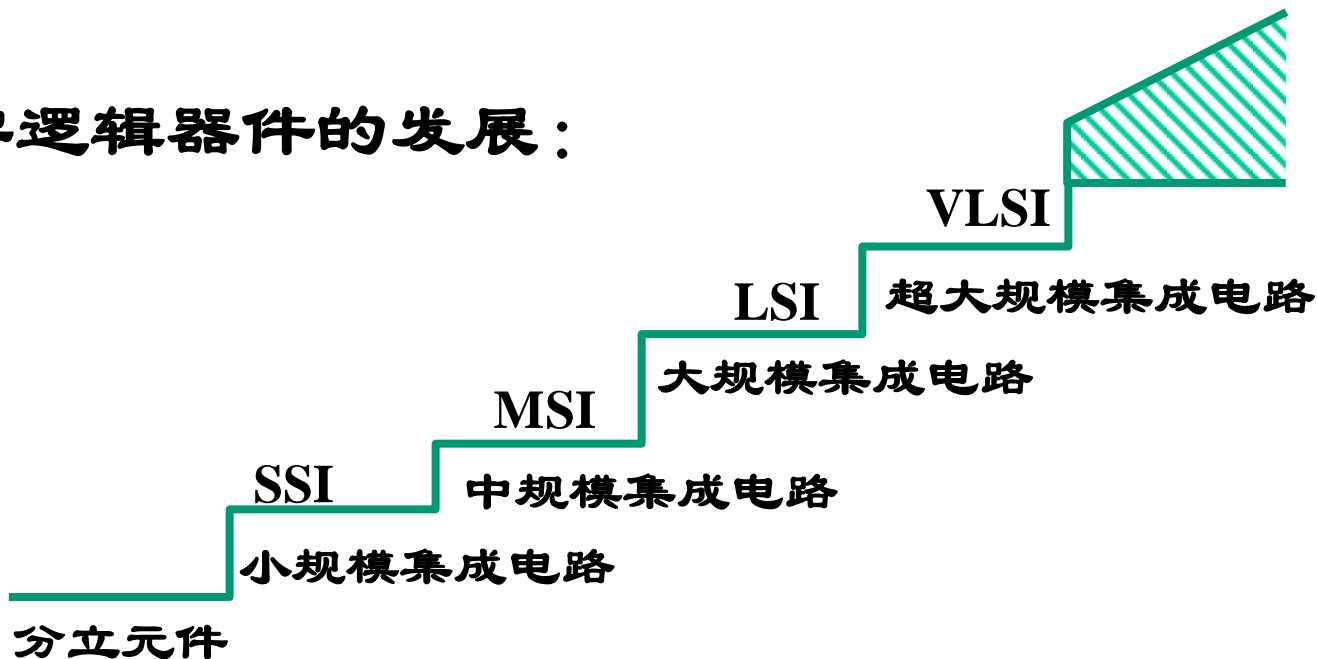
School of Computer Science  
Zhangyanhang@hit.edu.cn

# ROM及其应用

---

- PLD的背景知识
- ROM及其应用

## 数字逻辑器件的发展：



### 按芯片设计方法

- **标准产品** 主要是一些中小规模集成电路，如：逻辑门、触发器、译码器、计数器、寄存器、ALU...
- **由软件组态的大规模集成电路** 如：单片机
- **专用或特定用途的大规模集成电路** 如：可编程逻辑器件 **PLD**

## 现代数字系统设计方法 (基于可编程芯片)

1. 画真值表或状态表
2. 编写HDL代码
3. EDA软件自动逻辑综合、布局布线
4. 利用EDA软件功能仿真、时序仿真
5. 下载到PLD芯片，在实验台验证
6. 若存在错误，在计算机上直接修改设计，重复上述过程。

对比

## 传统数字系统设计方法 (基于电路板，全部手动完成)

1. 画真值表或状态表
2. 表达式化简
3. 画出电路原理图
4. 基于面包板实验验证
5. 绘制PCB板图，送交制板
6. 对PCB板安装、焊接、调试。  
若存在大错误，重复上述过程

工作量大，  
开发周期长

数字系统  
设计趋势

EDA软件



计算机

编程



空白PLD

数字系统

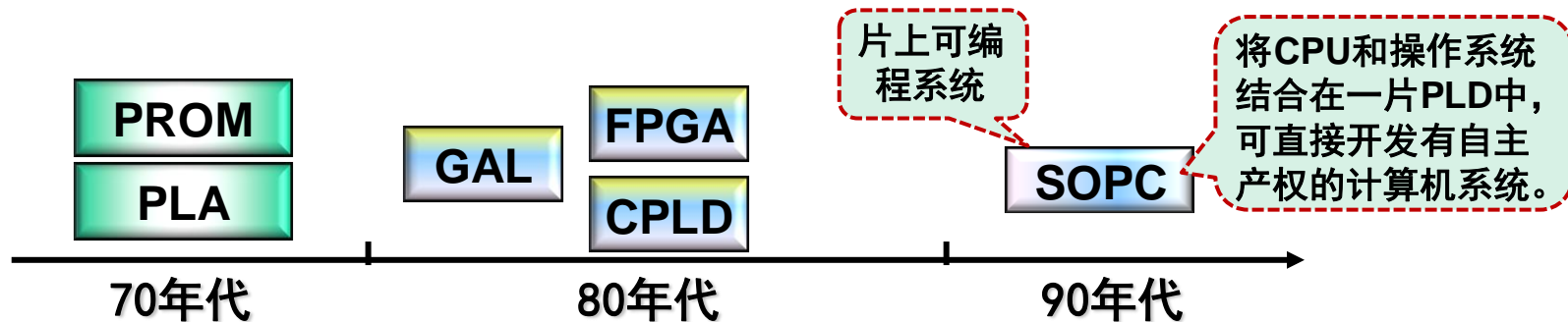
特点	传统方法	现代方法
采用器件	通用型器件	PLD
设计对象	电路板	可编程芯片
仿真验证时期	硬件设计后期	硬件设计早期
设计文件	电路原理图	HDL编写的代码

# ROM系列

## PLD结构原理

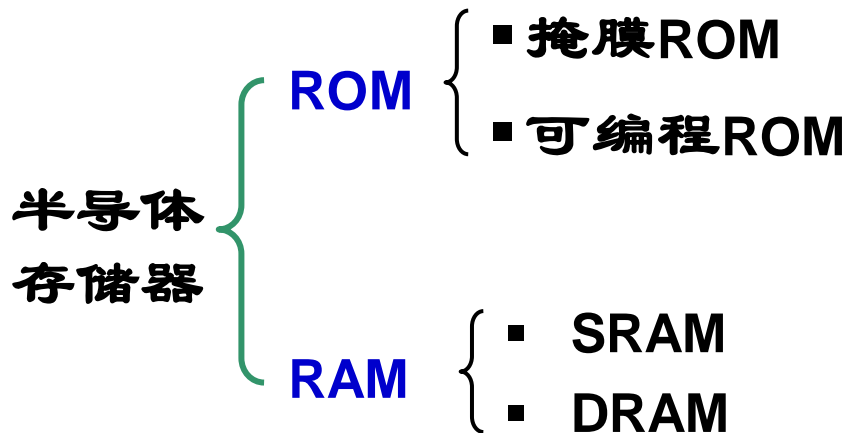
- 任何**组合逻辑**函数均可化为“与或”表达式，用“与门-或门”二级电路实现。
- 任何**时序电路**都是由组合电路加上存储元件（触发器）构成的。
- 从原理上说，**与或阵列加上寄存器**的结构可以实现任何数字逻辑电路。
- PLD：采用“与或阵列 + 寄存器 + 可灵活配置的互连线的结构”，即可实现任意的逻辑功能。

# PLD的分类及发展



# ROM系列

半导体存储器：能存储大量二值数字信息，是现代数字系统特别是计算机的重要组成部分。



(Random-Access Memory)

## 两个重要技术指标

- 存储容量: 能存放多少比特二值信息
- 存储时间: 读出/写入数据的时间, 一般用读/写周期表示

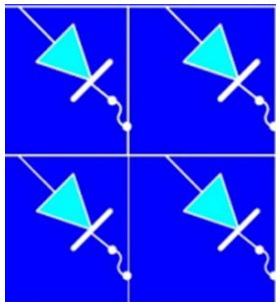
# ROM系列

Read-Only  
Memories

ROM

存放字库，  
数学用表

采用熔丝工艺，  
只能编程一次。

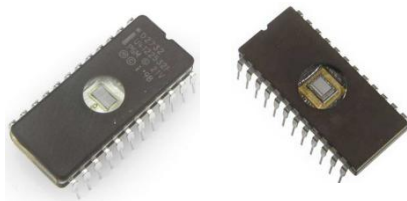


PROM

可编程ROM

EPROM

采用浮栅MOS管，需  
要专用的写入器，编  
错了可清除整体信息  
(紫外线照射)。



电可擦除可编  
程，不需编程  
器和擦除器



E<sup>2</sup>PROM

Flash Memory

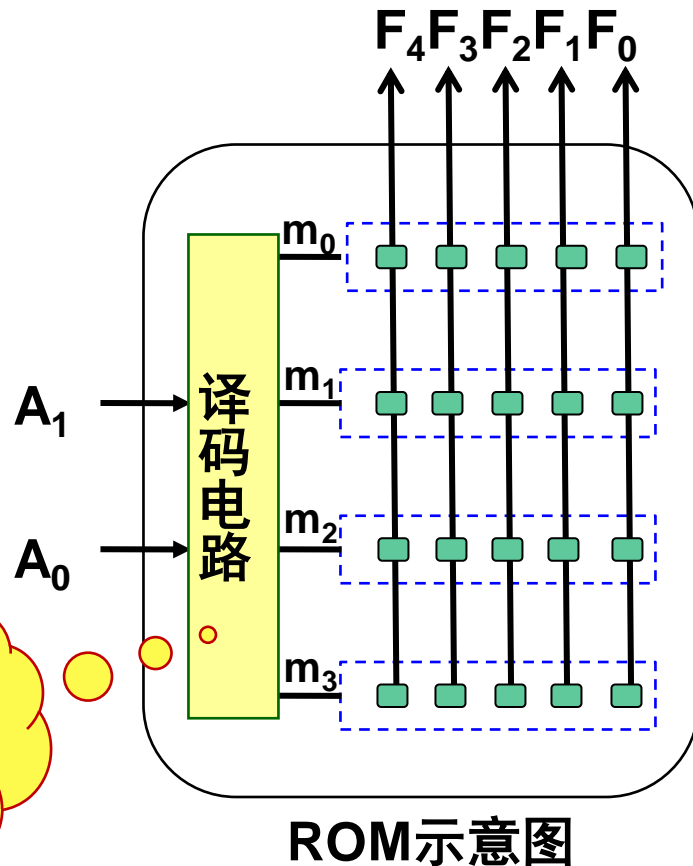
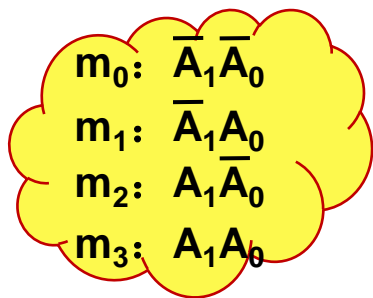
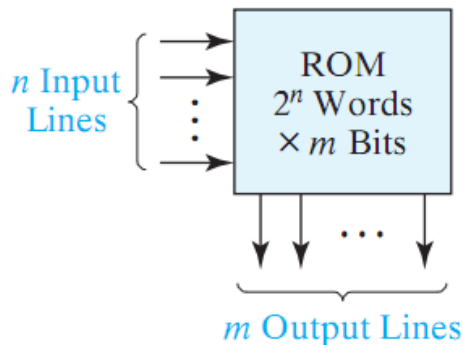
新一代的  
E<sup>2</sup>PROM



# ROM的基本结构

## ROM从整体上看

- 具有 $n$ 个输入 $m$ 个输出的组合电路
- 输入：称为地址输入
- 以类似矩阵的形式存储，每次存取其中的一行



# ROM的基本结构

## 字 (Word) ——

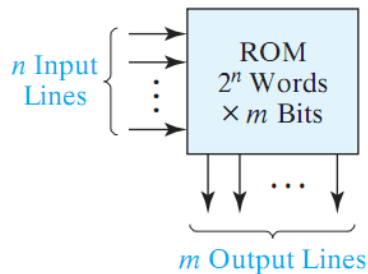
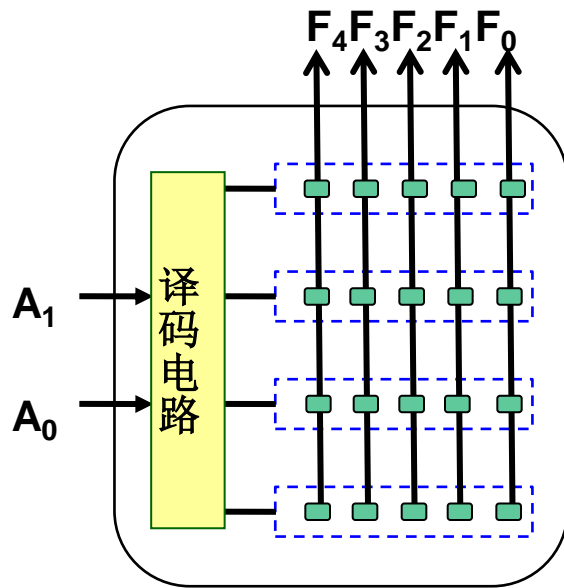
- 即存储单元
- 每个单元 (字) 由若干个二进制位组成

对应ROM  
的输出数量

## 地址 (Address) ——

- $n$  位地址能寻址  $2^n$  个存储单元 (字)
- 每个存储单元存放  $m$  位 (每个字  $m$  位长)
- 存储容量:  $2^n \times m$

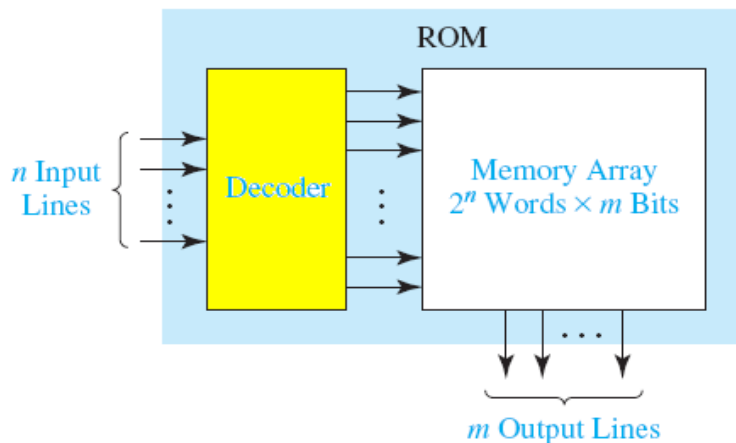
ROM的  $m$   
个输出



# ROM的基本结构

从本质上讲——

- ROM中存放的是一个  $n$  个输入、 $m$  个输出的组合逻辑的真值表
- 每个输入组合选中一个存储单元，该存储单元中存放的内容出现在输出端



每一组输入  
对应ROM  
的一个存储  
单元的地址

$n$ Input Variables	$m$ Output Variables
00 ... 00	100 ... 110
00 ... 01	010 ... 111
00 ... 10	101 ... 101
00 ... 11	110 ... 010
...	...
11 ... 00	001 ... 011
11 ... 01	110 ... 110
11 ... 10	011 ... 000
11 ... 11	111 ... 101

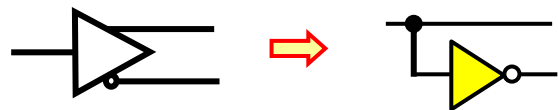
每一组输出对  
应ROM的一  
个存储单元中  
的存放内容

Typical Data  
Array Stored  
in ROM  
( $2^n$  words of  
 $m$  bits each)

- 一个  $2^n \times m$  的ROM能实现一个  $n$  输入、 $m$  个输出的组合逻辑函数

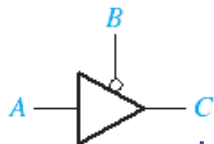
# PLD阵列图中的常见符号

输入缓冲器



- 增强输入信号的驱动能力
- 产生输入信号的原变量和反变量。

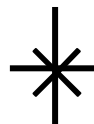
不同于三态门



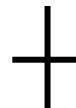
阵列连接方式



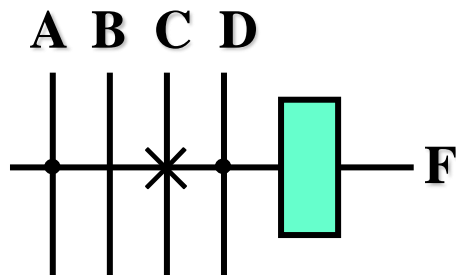
固定连接



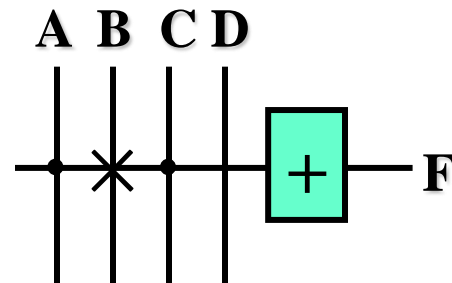
可编程连接



不连接

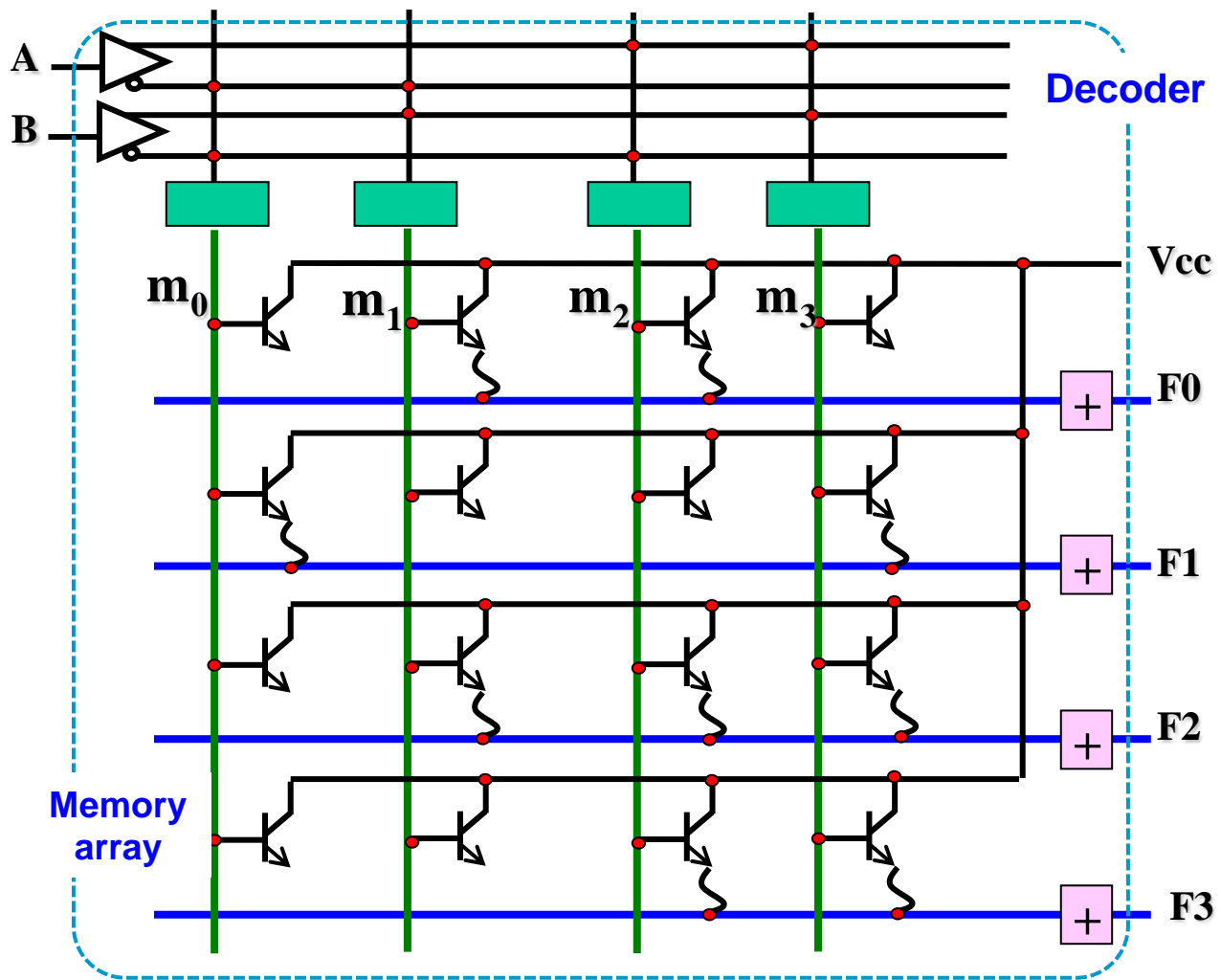


$$F = ACD$$



$$F = A + B + C$$

# ROM 举例



# ROM举例

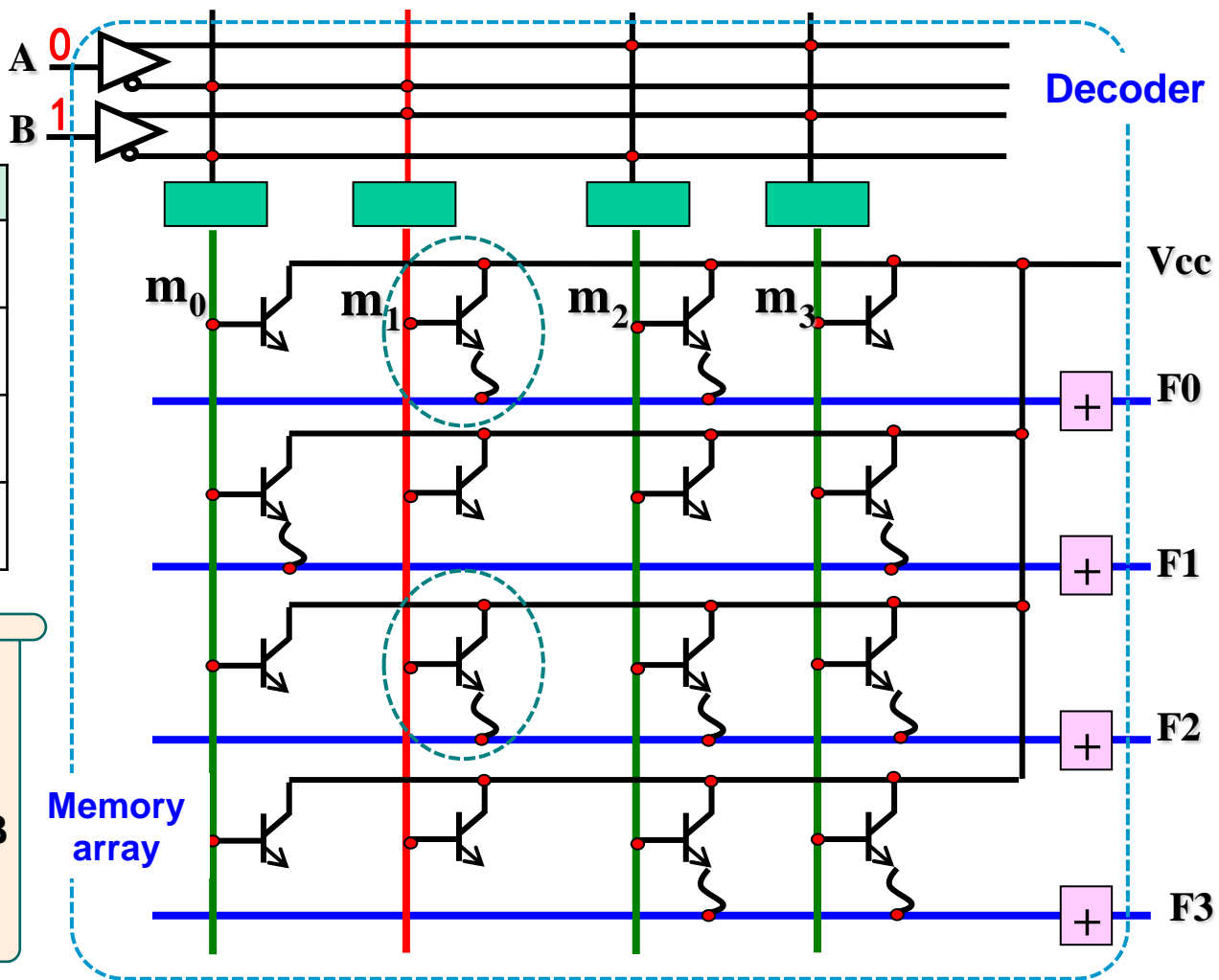
AB		$F_3 F_2 F_1 F_0$
00	$m_0=1$ ( $m_1=0, m_2=0, m_3=0$ )	0 0 1 0
01	$m_1=1$ ( $m_0=0, m_2=0, m_3=0$ )	0 1 0 1
10	$m_2=1$ ( $m_0=0, m_1=0, m_3=0$ )	1 1 0 1
11	$m_3=1$ ( $m_0=0, m_1=0, m_2=0$ )	1 1 1 0

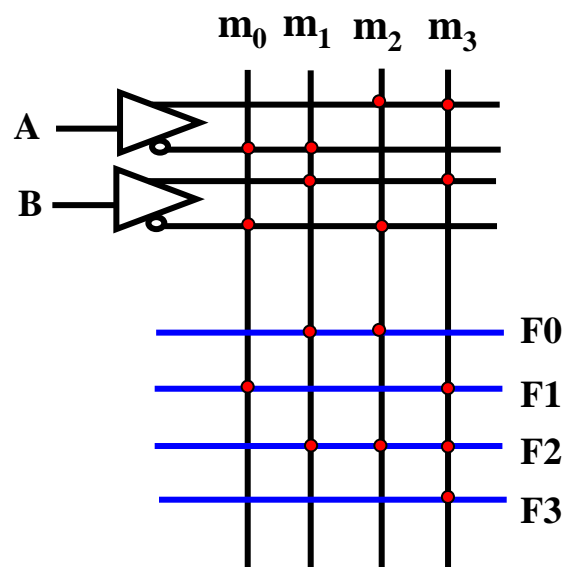
$$F_0 = m_1 + m_2 = \bar{A}B + A\bar{B}$$

$$F_1 = m_0 + m_3 = \bar{A}\bar{B} + AB$$

$$F_2 = m_1 + m_2 + m_3 = \bar{A}B + A\bar{B} + AB$$

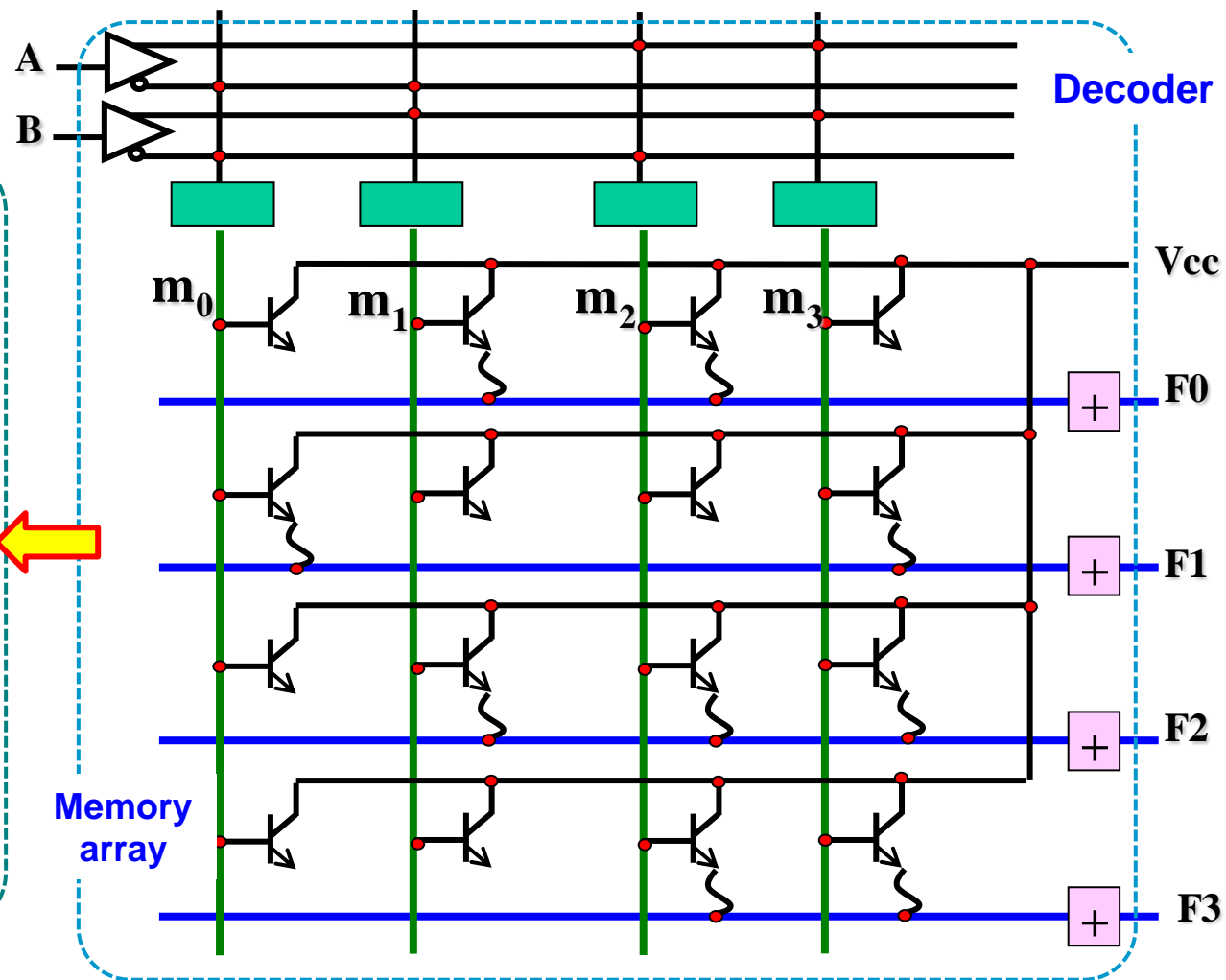
$$F_3 = m_2 + m_3 = A\bar{B} + AB$$





简化的ROM阵列图：

用X ( or • ) 表示开关元件



# 利用ROM实现组合逻辑函数

AB		F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>
00	$m_0=1(m_1=0, m_2=0, m_3=0)$	0 0 1 0
01	$m_1=1(m_0=0, m_2=0, m_3=0)$	0 1 0 1
10	$m_2=1(m_0=0, m_1=0, m_3=0)$	1 1 0 1
11	$m_3=1(m_0=0, m_1=0, m_2=0)$	1 1 1 0

$$F_0 = m_1 + m_2 = \bar{A}B + A\bar{B}$$

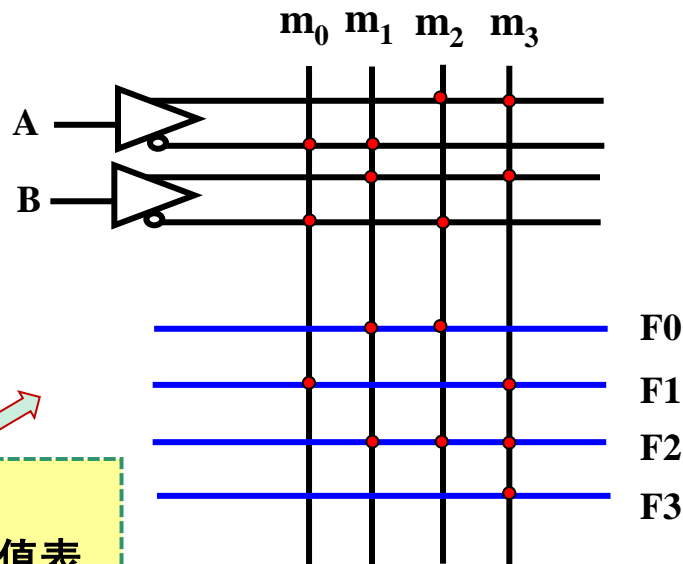
$$F_1 = m_0 + m_3 = \bar{A}\bar{B} + AB$$

$$F_2 = m_1 + m_2 + m_3 = \bar{A}B + A\bar{B} + AB$$

$$F_3 = m_2 + m_3 = A\bar{B} + AB$$

ROM——

存放真值表



简化的ROM阵列图：

用X ( or • ) 表示开关元件