

# 数字电路

## Digital Circuits

### 16\_存储器与PLD

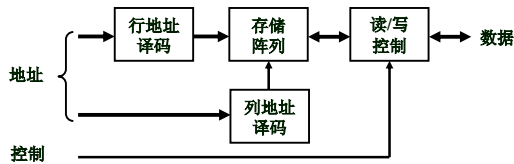
张俊霞  
zjx@ustc.edu.cn

## 内容提纲

- 只读存储器
- 随机存取存储器
- 存储器容量扩展
- GAL与高密度PLD

## 半导体存储器

- 能够保存大量二值信息的半导体器件
- 存储容量 = 字数 \* (位数/字)
  - 每个字具有唯一的编号, 称为地址
- 一般结构



## 半导体存储器分类

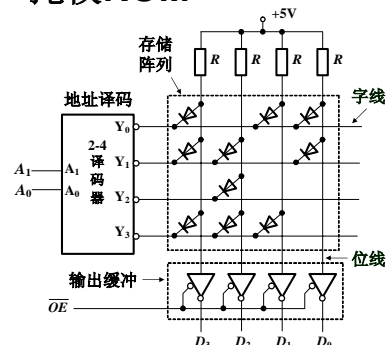
- 只读存储器
  - Read Only Memory, 简称ROM
  - 正常工作状态只能读出信息, 断电后信息不会丢失, 常用于存放固定信息(如程序、常数等)
- 随机存储器
  - Random Access Memory, 简称RAM
  - 在运行状态可以随时进行读出或写入操作, 存储信息必须有电源供应才能保存, 一旦掉电, 信息全部丢失

## ROM

- MROM (Masked ROM)
    - 掩模ROM, 不可改写
  - PROM (Programmable ROM)
    - 可编程ROM (只可编程一次)
  - EPROM (Erasable PROM)
    - 紫外线擦除的可编程ROM
  - EEPROM (Electrically EPROM)
    - 电擦除的可编程ROM
  - Flash memory
    - 闪存, 一种特殊的EEPROM
- } 离线编程  
} 在线编程

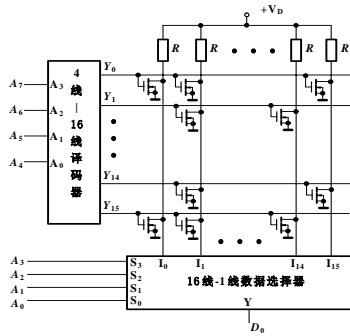
## 掩模ROM

- 存储单元
  - 有二极管 相当存1
  - 无二极管 相当存0
- 存储容量
  - = 4 x 4 (位)
  - 4个字 → 2位地址
  - 每字4位 → 4位数据



## 地址二维译码

- 高位地址译码
  - 选择一行
- 低位地址译码
  - 从选中的行选择一列
- 存储单元
  - 有MOS管相当于存0, 无MOS管相当于存1



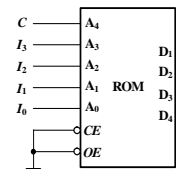
## 示例—ROM应用

- 利用ROM实现二进制码与格雷码相互转换

编码顺序	二进制码	格雷码	编码顺序	二进制码	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

## 示例—ROM应用(续)

- C = 0
  - 二进制码→格雷码
- C = 1
  - 格雷码→二进制码



C	I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> (二进制码)	O <sub>3</sub> O <sub>2</sub> O <sub>1</sub> O <sub>0</sub> (格雷码)	C	I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> (格雷码)	O <sub>3</sub> O <sub>2</sub> O <sub>1</sub> O <sub>0</sub> (二进制码)
0	0000	0000	1	0000	0000
0	0001	0001	1	0001	0001
0	0010	0011	1	0010	0011
0	0011	0010	1	0011	0010
0	0100	0110	1	0100	0111
0	0101	0111	1	0101	0110
0	0110	0101	1	0110	0100
0	0111	0100	1	0111	0101
0	1000	1100	1	1000	1111
0	1001	1101	1	1001	1110
0	1010	1111	1	1010	1100
0	1011	1110	1	1011	1101
0	1100	1010	1	1100	1000
0	1101	1011	1	1101	1001
0	1110	1001	1	1110	1011
0	1111	1000	1	1111	1010

## RAM

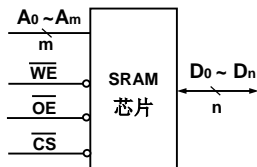
- 根据是否需要刷新, RAM分为SRAM和DRAM
  - 刷新: 保证存储信息不丢失的措施
- SRAM: 静态RAM (Static)
  - 存储单元是触发器
- DRAM: 动态RAM (Dynamic)
  - 存储单元是电容

### SRAM与DRAM比较

	SRAM	DRAM
刷新	不需要	需要
速度	快	慢
控制	简单	复杂
集成度	低	高
价格	昂贵	便宜
功耗	高	低

## SRAM芯片

- 地址引脚数
  - 决定存储字数
- 数据引脚数
  - 决定存储字长
- 控制引脚
  - WE: 写允许(Write Enable)
  - OE: 输出允许(Output Enable)
  - CS: 片选(Chip Select)
- 存储容量 =  $2^m \times n$  (位) =  $2^m \times n / 8$  (字节)

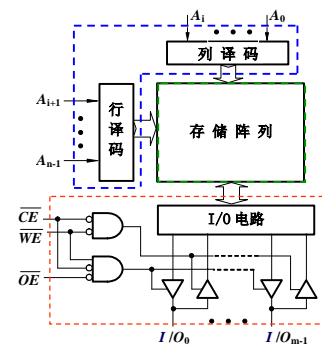


功能表

CS	WE	OE	D <sub>0</sub> ~ D <sub>n</sub>	操作
1	x	x	高阻	无操作
0	0	x	输入	写
0	1	0	输出	读
0	1	1	高阻	无输出

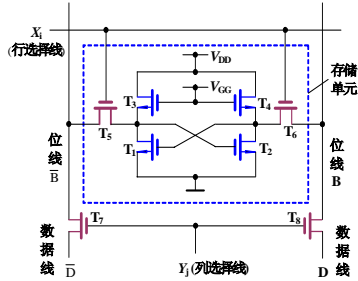
## SRAM芯片内部结构

- 存储阵列
- 地址译码
- 输入/输出控制



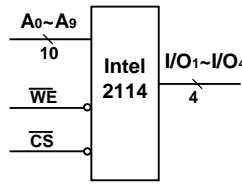
## SRAM存储单元

- $X_i=1$ 
  - T5、T6导通
  - 触发器与位线接通
- $Y_j=1$ 
  - T7、T8导通
  - 触发器的输出与数据线接通
  - 通过数据线读取该单元数据

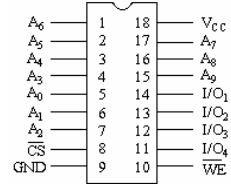


## 示例—SRAM芯片2114

- 容量 =  $1K \times 4 = 4Kb$
- 地址:  $A_0 \sim A_9$
- 数据:  $I/O_1 \sim I/O_4$
- 控制: 片选  $\overline{CS}$  和写允许  $\overline{WE}$
- 电源和地:  $V_{CC}$  和  $GND$

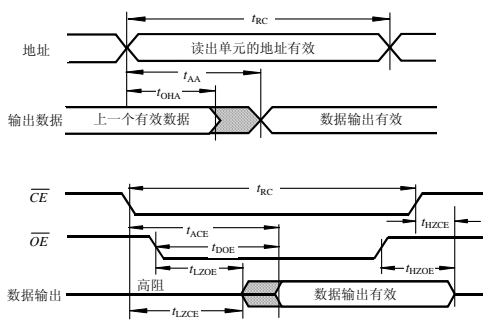


Intel 2114 逻辑图

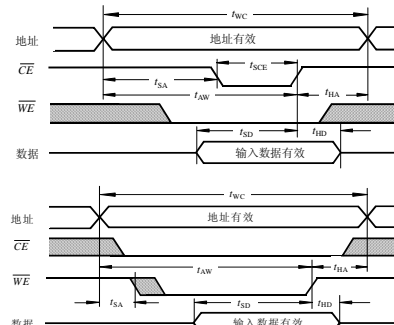


Intel 2114 引脚图

## SRAM读操作时序图

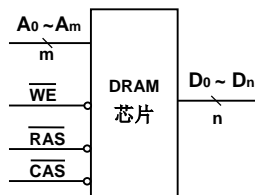


## SRAM写操作时序图



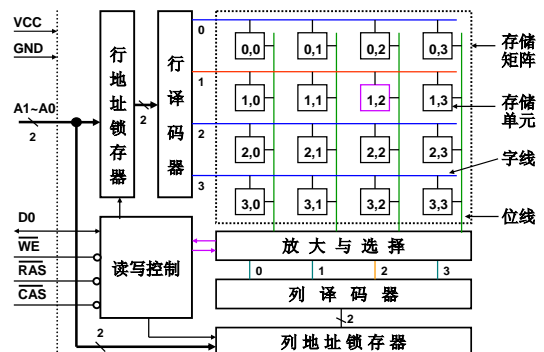
## DRAM芯片

- 地址: 决定存储字数
  - 为减少引脚数, 地址分两次输入
- 数据: 存储字大小
- 控制
  - WE: 写允许
  - RAS: 行地址选通 (Row Address Strobe)
  - CAS: 列地址选通 (Column Address Strobe)



- 存储容量
  - =  $2^{2m} \times n$  (位)
  - =  $2^{2m} \times n / 8$  (字节)

## DRAM芯片内部结构



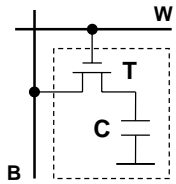
## DRAM存储单元

- 由1个三极管和1个电容组成

- 根据电容是否充有电荷来存储一位信息
- T用于控制读写

- 读写过程

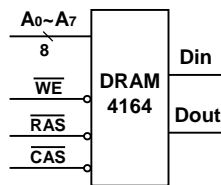
- 写
  - (1) B上设置待写数据
  - (2) W设置成有效
- 读
  - (1) B预充电至高电平
  - (2) W设置成有效



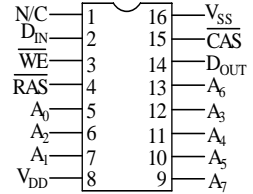
## 示例—DRAM芯片4164

容量 = 64K x 1 = 64Kb

- 地址: A<sub>0</sub> ~ A<sub>7</sub>
- 数据: Din, Dout
- 控制: WE, RAS和CAS
- 电源和地: V<sub>SS</sub>和V<sub>DD</sub>



4164 逻辑符号



4164 引脚图

## 存储器容量扩展

- 由多个存储芯片, 构成较大容量的存储器
- 容量扩展方式

- 位扩展

- 由容量为P x L位的芯片, 构成容量为P x N位的存储器 (N>L), 所需芯片数=[N/L]

- 字扩展

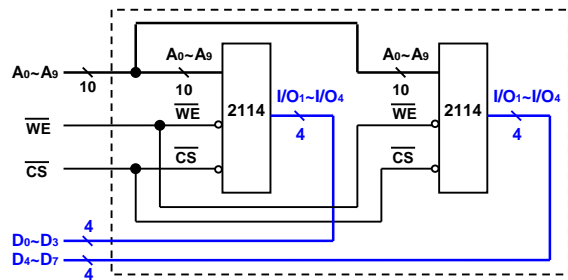
- 由容量为P x L位的芯片, 构成容量为M x L位的存储器 (M>P), 所需芯片数=[M/P]

- 字位同时扩展

- 由容量为P x L位的芯片, 构成容量为M x N位的存储器 (M>P, N>L), 所需芯片数=[M/P] \* [N/L]

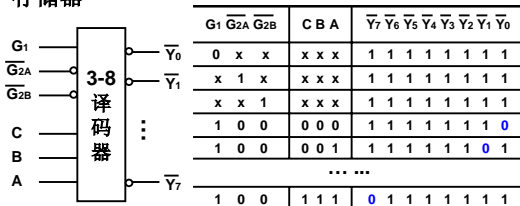
## 示例一位扩展

- 方法: 除数据引脚外, 芯片的其他引脚(地址和控制)并联
- 例如: 用容量为1Kx4位的2114构成1K x 8位的存储器

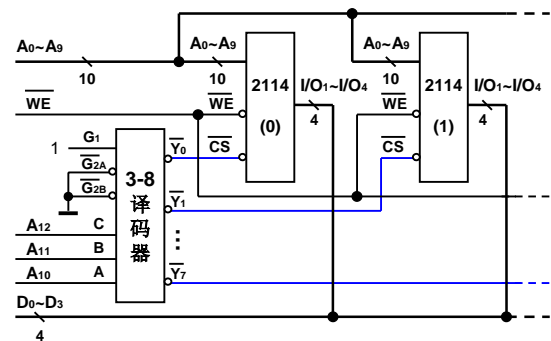


## 示例一字扩展

- 方法: 除片选引脚外, 芯片的其他引脚(地址、数据、写允许)并联, 高位地址经译码选择芯片
- 例如: 用容量为1Kx4位的2114构成8K x 4位的存储器



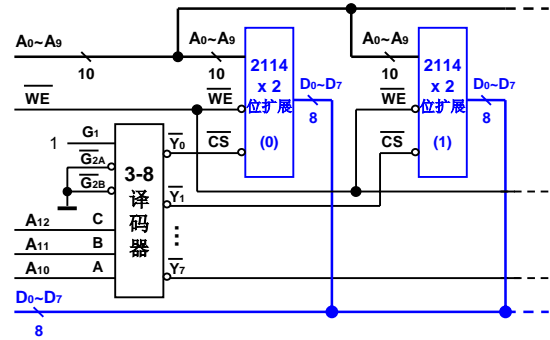
## 示例一字扩展 (续)



## 示例一字位同时扩展

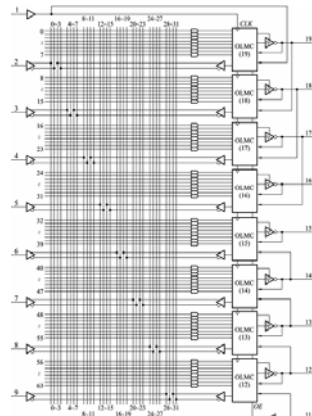
- 方法：将芯片分组，组内按位扩展，组间按字扩展
- 例如：用容量为1Kx4位的2114构成8K x 8位的存储器  
 所需芯片数 =  $(8K \times 8) / (1K \times 4)$   
 =  $(8K / 1K) \times (8 / 4)$   
 =  $8 \times 2$   
 = 16 (片)  
 这些芯片分成8组，每组2片

## 示例一字位同时扩展 (续)



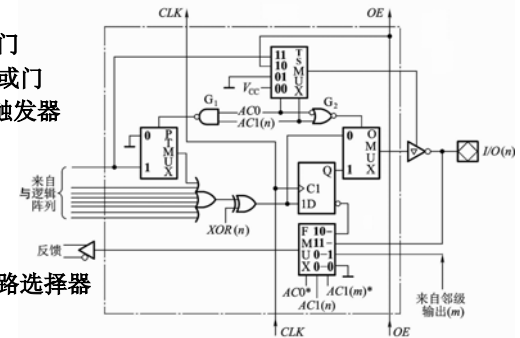
## GAL结构

- Generic Array Logic, 通用阵列逻辑
- 相较PAL器件, 用可编程的输出逻辑宏单元 (OLMC) 代替固定的或阵列
- 可实现多种形式的输出



## 输出逻辑宏单元 (OLMC)

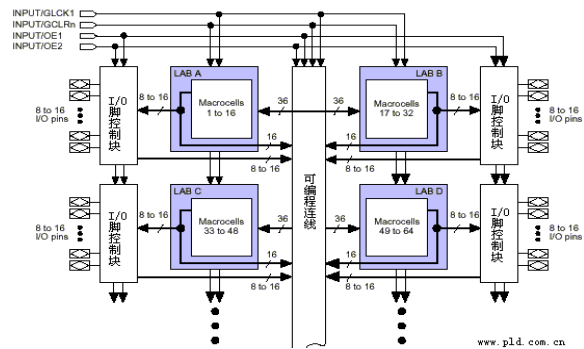
- 或门
- 异或门
- D触发器
- 多路选择器



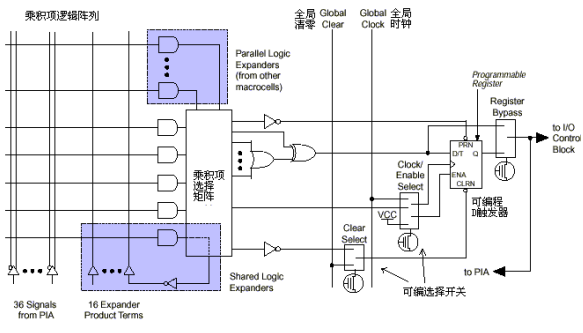
## 高密度PLD

- Complex Programmable Logic Device, CPLD, 复杂可编程逻辑器件
  - 基于与或阵列和宏单元结构, 采用E<sup>2</sup>PROM存储技术, 具有非易失性、互连通路延时可预测等优点, 适于实现大规模组合逻辑
- Field Programmable Gate Array, FPGA, 现场可编程门阵列器件
  - 多采用查找表 (Look Up Table, LUT) 技术, SRAM 存储技术, 具有集成度高、触发器资源丰富等优点, 易于实现时序逻辑

## CPLD结构



## 逻辑阵列块LAB



## 可编程的I/O单元

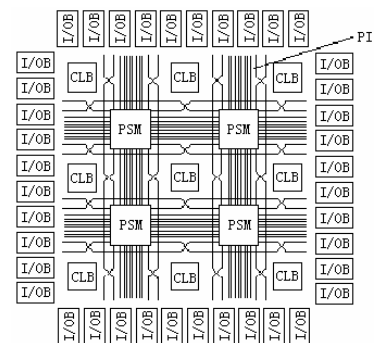
- 能兼容TTL和CMOS多种接口和电压标准
- 可配置为输入、输出、双向、集电极开路和三态等形式
- 能提供适当的驱动电流
- 降低功耗，防止过冲和减少电源噪声
- 支持多种接口电压（降低功耗）
  - 1.2um~0.5um, 5V
  - 0.35um, 3.3V
  - 0.25um, internal 2.5V, I/O 3.3V
  - 0.18um, internal 1.8V, I/O 2.5V and 3.3V

## 可编程连线阵列PIA

- 在各个逻辑宏单元之间以及逻辑宏单元与I/O单元之间提供信号连接的网络
- CPLD中一般采用固定长度的线段来进行连接，因此信号传输的延时是固定的，使得时间性能容易预测

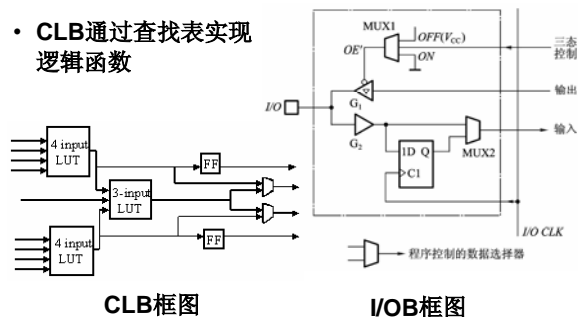
## FPGA结构

- 可配置逻辑块CLB
- 可编程开关矩阵PSM
- 可编程互连PI
- 可编程输入/输出块I/OB

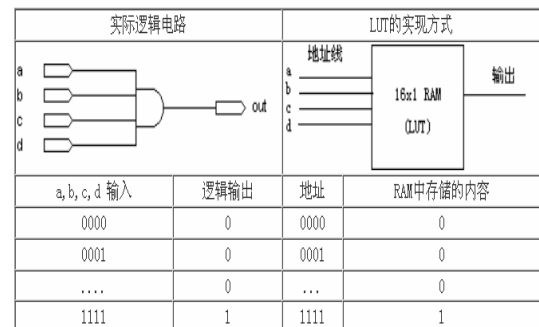


## CLB和I/OB

- CLB通过查找表实现逻辑函数



## 示例—LUT实现组合逻辑



## CPLD与FPGA比较

	CPLD	FPGA
内部结构	乘积项	查找表
配置存储	内部E <sup>2</sup> PROM	SRAM, 外挂E <sup>2</sup> PROM
资源类型	组合电路资源丰富	触发器资源丰富
集成度	低	高
使用场合	完成控制逻辑	能完成比较复杂的算法
速度	慢	快
其他资源	—	EAB, 锁相环, DSP, MCU等
保密性	可加密	一般不能保密

The End