

SRAM 芯片和 DRAM 芯片

工作原理

通常把存放一个二进制位的物理器件称为存储元，它是存储器的最基本的构件。地址码相同的多个存储元构成一个存储单元。若干存储单元的集合构成存储体。

SRAM	非破坏性读出 存取速度快 集成度低 成本高	静态随机存储器 (SRAM) 的存储元是双稳态触发器 (六晶体管 MOS) 来记忆信息的，静态是指即使信息被读出后，它仍保持其原状态而不需要再生 (非破坏性读出)。
DRAM SDRAM 同步 DRAM	破坏性读出 存取速度慢 集成度高	动态随机存储器 (DRAM) 是利用存储元电路上栅极电容上的电荷来存储信息的，DRAM 的基本存储元通常只使用一个晶体管，所以它比 SRAM 的密度要高很多。

DRAM 刷新

DRAM 电容上的电荷一般只维持 1~2ms，因此即使电源不断电，信息也会自动消失。此外，读操作会使其状态发生改变 (破坏性读出)，需读后再生，这也是称其为动态存储器的原因。

刷新可采用读出的方法进行，根据读出内容对相应单元进行重写，即读后再生。对同一行进行相邻两次刷新的时间间隔称为刷新周期，通常取 2ms。

集中刷新	读/写操作时不受刷新工作的影响 在集中刷新期间 (死区) 不能访问存储器 有死区	在一个刷新周期内，利用一段固定的时间，依次对存储器的所有行进行逐一再生，在此期间停止对存储器的读/写操作，称为死时间，也称访存死区。
分散刷新	没有死区 加长了系统的存取周期	将一个存储器系统的工作周期分为两部分：前半部分用于正常的读/写操作；后半部分用于刷新。这种刷新方式增加了系统的存取周期，如存储芯片的存取周期为 0.5μs，则系统的存取周期为 1μs。
异步刷新	有死区	结合了前两种方法，使得在一个刷新周期内每一行仅刷新一次。具体做法是将刷新周期除以行数，得到相邻两行之间刷新的时间间隔 t，每隔时间 t 产生一次刷新请求。这样就使“死时间”的分布更加分散，避免让 CPU 连续等待过长的时间。

SRAM 和 DRAM 的比较

	SRAM	DRAM
存储信息	触发器	电容
破坏性读出	非	是
需要刷新	不要	需要
送行列地址	同时送	分两次送 (复用)
运行速度	快	慢
集成度	低	高
存储成本	高	低
主要用途	高速缓存	主机内存

存储器芯片的内部结构

存储器芯片由存储体、I/O 读/写电路、地址译码器和控制电路等部分组成。

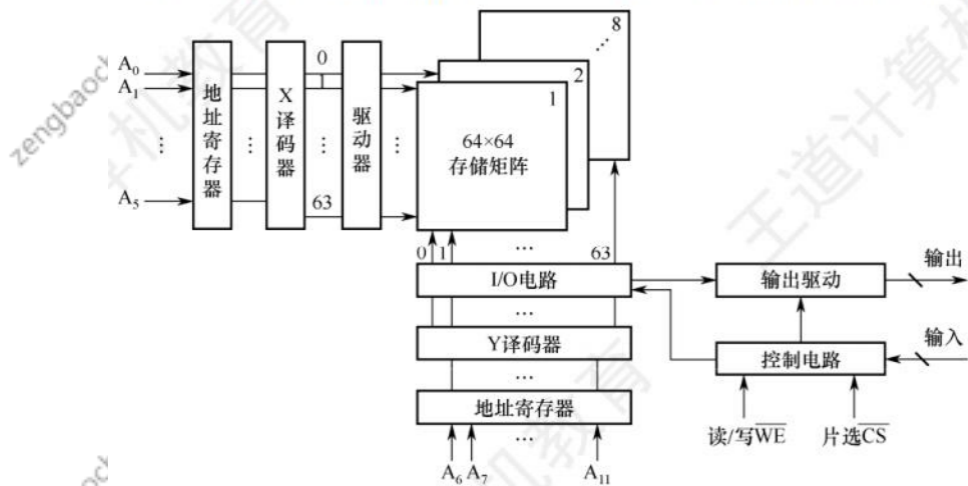


图 3.5 存储器芯片结构图

- 存储体 (存储矩阵)
存储体是存储单元的集合，它由行选择线 (X) 和列选择线 (Y) 来选择所访问单元，存储体的相同行、列上的多位 (位平面数) 同时被读出或写入。
- 地址译码器
用来将地址转换为译码输出线上的高电平，以便驱动相应的读/写电路。地址译码有单译码法 (一维译码) 和双译码法 (二维译码) 两种方式。
- I/O 控制电路
用以控制被选中的单元的读出或写入，具有放大信息的作用。
- 片选控制信号
单个芯片容量太小，往往满足不了计算机对存储器容量的要求，因此需要用一定数量的芯片进行存储器的扩展。在访问某个字时，必须“选中”该存储字所在的芯片，而其他芯片不被“选中”，因此需要有片选控制信号。
- 读/写控制信号
根据 CPU 给出的读命令或写命令，控制被选中单元进行读或写。

只读存储器

ROM 的类型	
MROM 掩模式只读存储器	\
PROM 一次可编程只读存储器	\
EPROM 可擦除可编程只读存储器	不仅可以由用户利用编程器写入信息，而且可以对其内容进行多次改写。EPROM 虽然既可读又可写，但它不能取代 RAM，因为 EPROM 的编程次数有限，且写入时间过长。
Flash 存储器	Flash 存储器是在 EPROM 的基础上发展起来的，它兼有 ROM 和 RAM 的优点，可在不加电的情况下长期保存信息，又能在线进行快速擦除与重写。Flash 存储器既有 EPROM 价格便宜、集成度高的优点，又有 E2PROM 电可擦除重写的特点，且擦除重写的速度快。
固态硬盘 Solid State Drive SSD	基于闪存的固态硬盘是用固态电子存储芯片阵列制成的硬盘，由控制单元和存储单元 (Flash 芯片) 组成。保留了 Flash 存储器长期保存信息、快速擦除与重写的特性。对比传统硬盘也具有读/写速度快、低功耗的特性，缺点是价格较高。

多模块存储器

CPU 的速度比存储器快得多，若同时从存储器中取出 n 条指令，就可以充分利用 CPU 资源，提高运行速度。

单体多字存储器	<p>在单体多字系统中，每个存储单元存储 m 个字，总线宽度也为 m 个字，一次并行读出 m 个字。在一个存取周期内，从同一地址取出 m 条指令，然后将指令逐条送至 CPU 执行，即每隔 1/m 存取周期，CPU 向主存取一条指令。这显然提高了单体存储器的工作速度。</p> <p>缺点：只有指令和数据在主存中连续存放时，这种方法才能有效提升存取速度。一旦遇到转移指令，或操作数不能连续存放时，这种方法的提升效果不明显。</p>	
多体并行存储器	多体并行存储器由多体模块组成。每个模块都有相同的容量和存取速度，各模块都有独立的读/写控制电路、地址寄存器和数据寄存器。它们既能并行工作，又能交叉工作。	

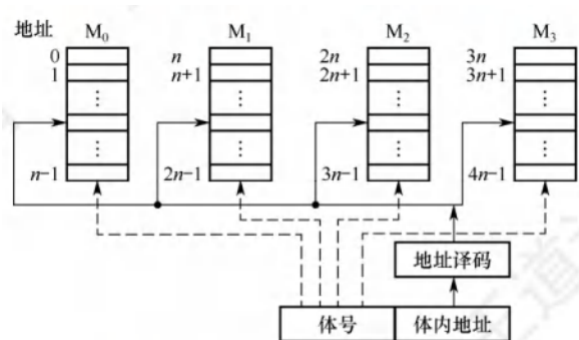


图 3.7 高位交叉编址的多体存储器

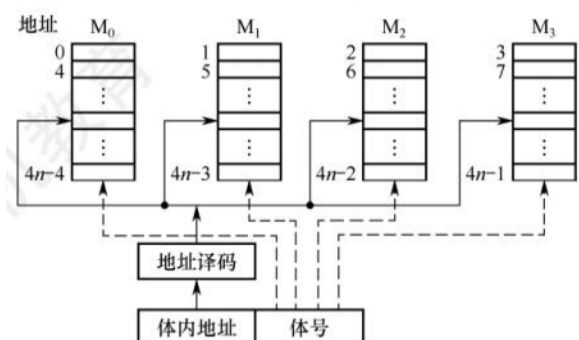


图 3.8 低位交叉编址的多体存储器

多体并行存储器			
高位交叉编址 (顺序方式)	高位地址表示模块号 (或体号), 低位地址为模块内地址 (或体内地址)。	\	在高位交叉方式下, 总把低位的体内地址送到由高位体号确定的模块内进行译码。访问一个连续主存块时, 总是先在一个模块内访问, 等到该模块访问完才转到下一个模块访问, CPU 总是按顺序访问存储模块, 各模块不能被并行访问, 因而不能提高存储器的吞吐率。
低位交叉编址 (交叉方式)	低位地址为模块号, 高位地址为模块内地址。	轮流启动同时启动	低位交叉方式下, 总是把高位的体内地址送到由低位体号所确定的模块内进行译码。程序连续存放在相邻模块中, 因此称采用此编址方式的存储器为交叉存储器。

低位交叉编址		
轮流启动	若每个模块一次读/写的位数正好等于数据总线位数, 模块的存取周期为 T , 总线周期为 r , 为实现轮流启动方式, 存储器交叉模块数应大于或等于 $m=T/r$ 。	
同时启动	若所有模块一次并行读/写的总位数正好等于数据总线位数, 则可以同时启动所有模块进行读/写。设每个模块一次读/写的位数为 16 位, 模块数 $m=4$, 数据总线位数为 64 位, 4 个模块一共提供 64 位, 正好构成一个存储字, 因此应该同时启动 4 个模块进行并行读/写。	

交叉存储器存取时间和带宽的计算

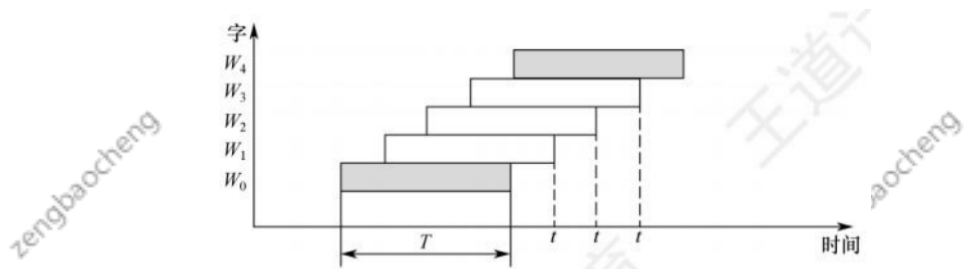


图 3.9 低位交叉轮流启动的存取时间示意图

按每隔 $1/m$ 个存取周期轮流启动各模块，则每隔 $1/m$ 个存取周期就可读出或写入一个数据，存取速度提高 m 倍，上图展示了 4 体交叉轮流启动的时间关系。交叉存储器要求其模块数大于或等于 m ，以保证启动某模块后经过 $m \times r$ 的时间后再次启动该模块时，其上次的存取操作已经完成 (以保证流水线不间断)。这样连续存取 m 个字所需的时间为

$$t_1 = T + (m - 1)r$$

而顺序方式连续读取 m 个字所需的时间为

$$t_2 = mT$$

交叉存储器中访存冲突的分析

在理想情况下， m 体交叉存储器每隔 $1/m$ 存取周期可读/写一个数据，若相邻的 m 次访问的访存地址出现在同一个模块内，则会发生访存冲突，此时需延迟发生冲突的访问请求。