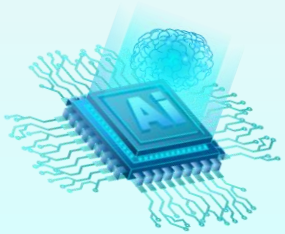
第9章 存储器



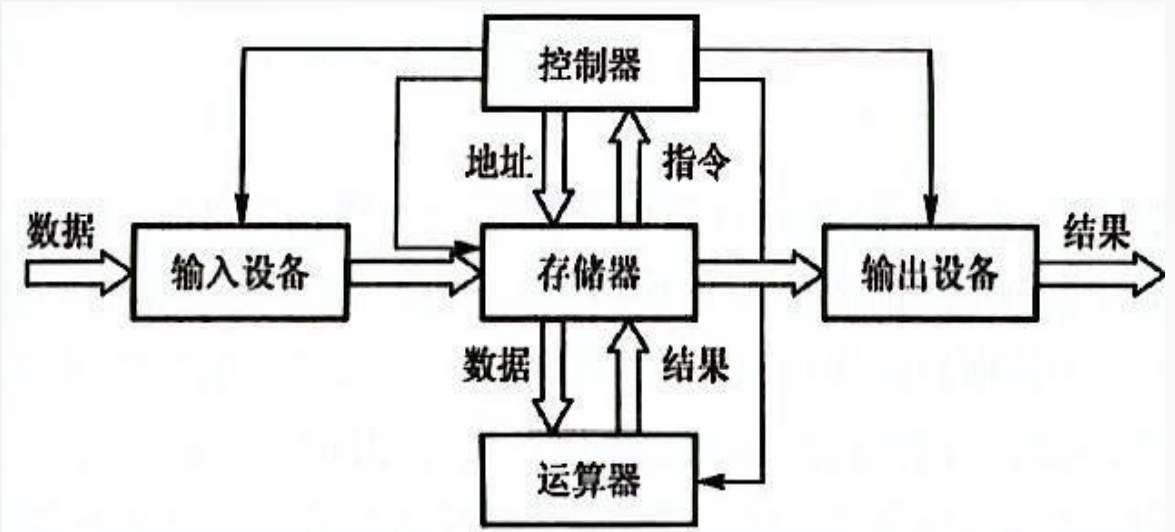
本章导读：

存储器是计算机系统的核心部件之一，在冯·诺依曼结构的计算机中，指令和数据均以二进制的形式存储在存储器中，当计算机运行程序时，主存储器中存放正在运行的指令和正在被处理的数据，并能由中央处理器直接随机存取。计算机系统中的存储器种类很多，本章首先介绍存储器的分类，然后再介绍随机读写存储器、高速缓冲存储器和Flash存储器等几种重要存储器。

9.1 存储器的分类与层次结构

9.1.1 存储器的基本概念

存储器是具有“记忆功能”的设备，它采用二进制的“0”和“1”来存储信息。现代冯·诺伊曼计算机是以存储器为中心，任何数据在传输、输入和输出的过程中都必须经过存储器的中转。



明确几点：字节地址、字、字地址

9.1.2 存储器的分类

1．按存储介质分类

目前存储元件所使用的存储介质主要有半导体介质、磁性介质和光介质。半导体类存储器应用最广：如静态随机读写存储器（主要用于高速缓存Cache和内存条）、U盘和固态硬盘；

2．按在计算机系统中的层次分类

根据在计算机系统中的位置和作用，存储器可以分为寄存器、高速缓冲存储器（Cache，简称高速缓存）、主存储器和辅助存储器等四个类型，所有这些存储器根据各自的不同特性和性价比，组成计算机系统中各个层次的存储设备。

3．按存取方式分类

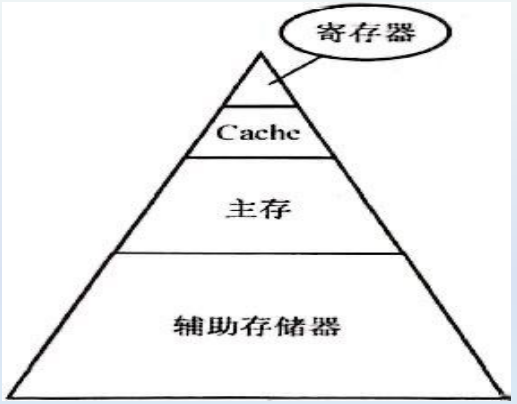
按对存储器的存取方式可分为随机读写存储器（Random Access Memory，RAM）和只读存储器（Read-Only Memory，ROM）两大类。

4．按信息的可保存性分类

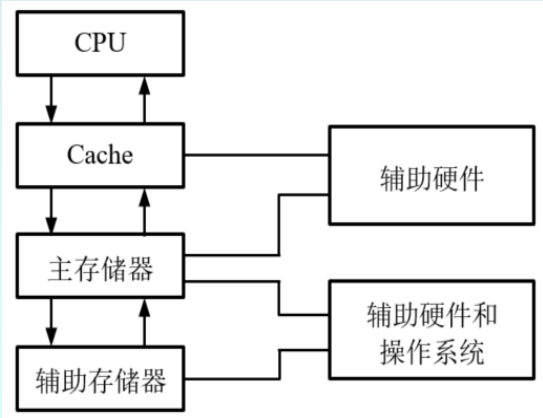
在断电后还能保存信息的存储器称为非易失性存储器；反之，称为易失性存储器。

9.1.3 存储器的层次结构

存储器的速度：处于最上层的是寄存器，访问时间是几个纳秒，第二层是高速缓存访问时间是寄存器访问时间的几倍。第三层是主存，计算机当前正在执行的程序和处理的数据都是存放在主存中的，任何程序如果要在计算机中执行，首先必须将其调入主存才能由CPU执行。



存储系统的工作原理是：CPU首先访问Cache，如果Cache中没有所需要的内容，则存储系统通过辅助硬件到主存中去查找；如果主存没有CPU要访问的内容，则存储系统通过辅助硬件或软件到辅存中去查找；然后，把找到的数据逐级调入相应的存储器中。



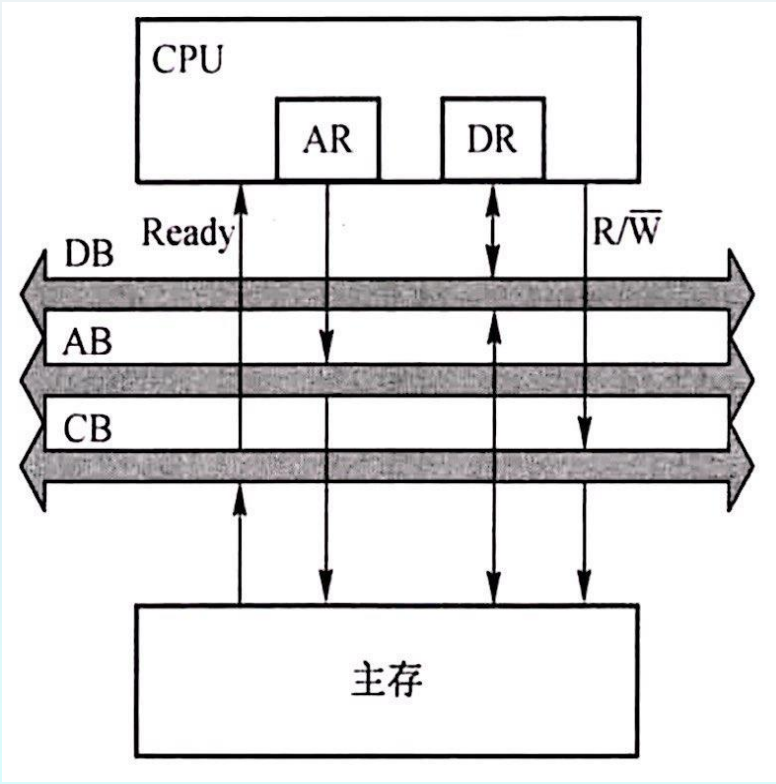
9.1.4 主存储器的基本原理（重点）

主存储器的概念：CPU通过地址、数据、控制三总线可以直接触及的地方。CPU内的程序计数器PC中可以存储的地址。

CPU与存储器通过总线连接，

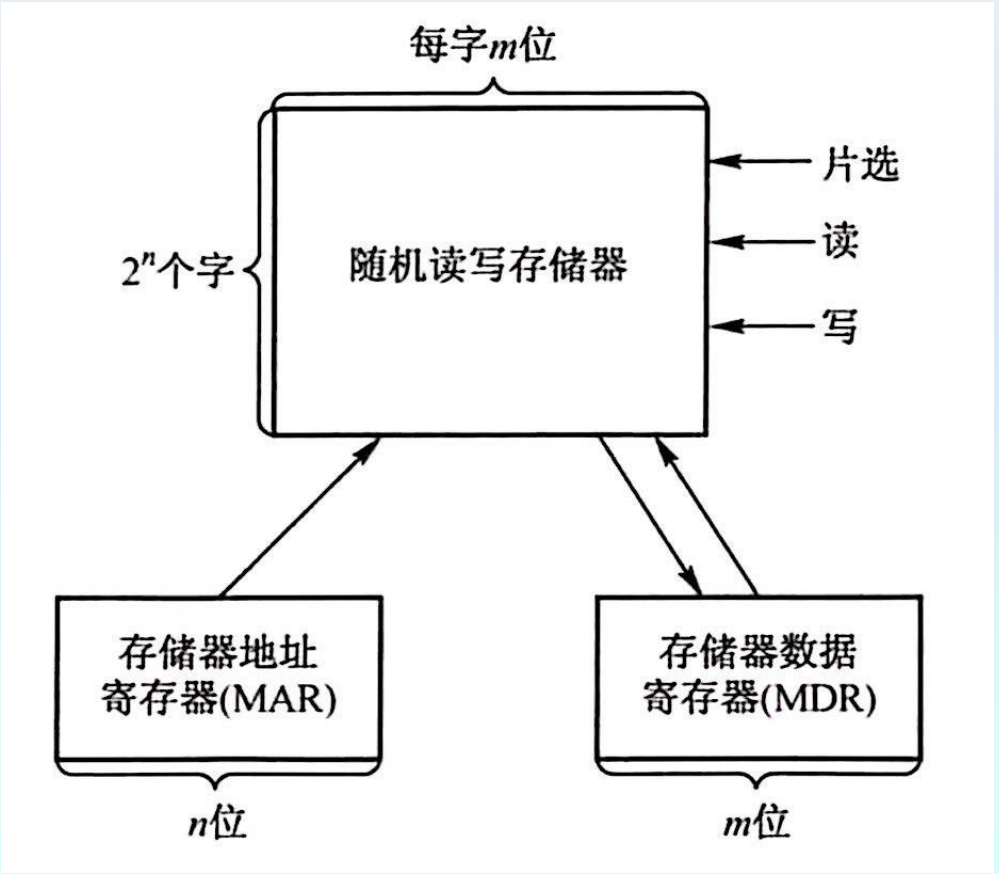
在“读”存储器时，CPU通过地址总线（AB）和控制总线（CB）分别给出所需字的存储器地址和访存“读”控制信号，然后等待存储器将读出的字送到数据总线（DB）上，并通知CPU“读”操作完成，CPU 从数据总线上读取这个字，结束“读”存储器的工作；

在“写”存储器时，CPU通过地址总线和数据总线分别给出欲写字的存储器地址和字的内容，CPU还通过控制总线给出访存“写”控制信号，然后等待，存储器将数据总线上送来的字写入指定地址的存储单元，并通知CPU“写”操作完成，CPU结束“写”存储器的工作。



关于MAR、MBR、MDR

计算机中的任何操作都是按照时间节拍有序进行的，由于各器件的操作都存在延时，为保证读/写的可靠性，应当在存储器的地址输入端和数据端分别安排一个寄存器，用来存放地址信号和数据信号，这两个寄存器分别称为**存储器地址寄存器**（MAR）和**存储器缓冲寄存器**（MBR）或**存储器数据寄存器**（MDR）。



9.2 RAM的存储器单元

随机读写存储器RAM可以随时从任何一个指定地址的存储单元中读取数据，也可以随时将数据写到任何一个指定地址的存储单元中。RAM掉电后将丢失数据，即一旦断开电源，RAM中所存储的信息就会随之丢失。

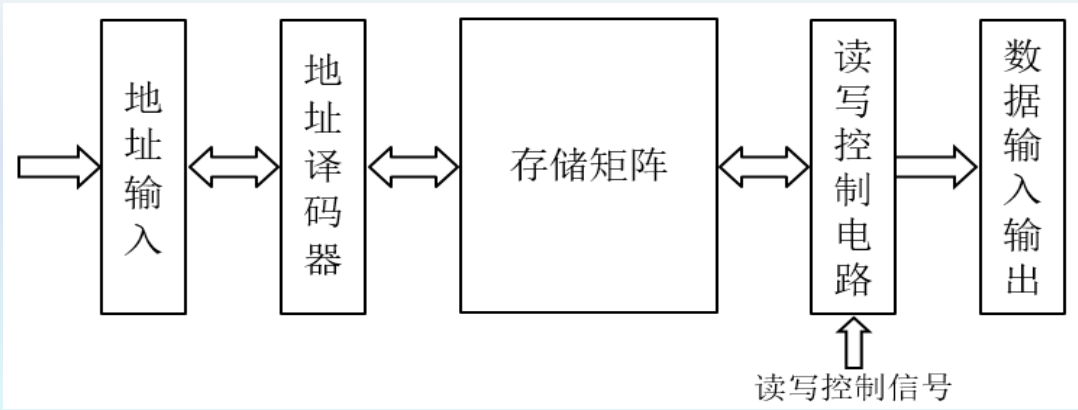
9.2.1 RAM的整体结构

随机读写存储器（RAM，Random Access Memory）两层含义：

（1）相对于顺序访问的存储器而言，CPU可以随机地访问RAM的任意单元；

（2）RAM掉电后将丢失数据，重新上电后，RAM的数据是“随机的”。

RAM 电路：



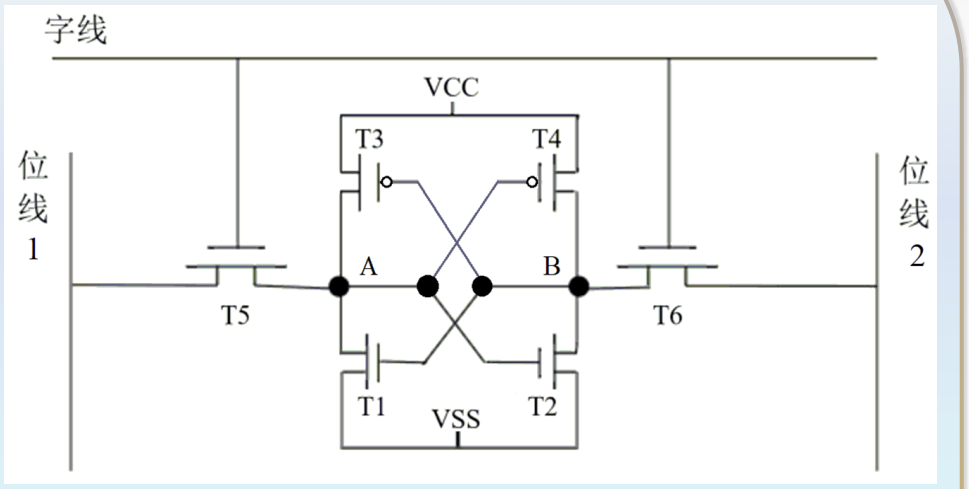
RAM 的分类：按照RAM存储单元的工作原理，RAM又分为静态随机存储器（Static RAM，SRAM）和动态随机存储器（Dynamic RAM，DRAM）。

9.2.2 SRAM存储单元

1. SRAM的工作原理（重点）

（1）**约定**。设存储数字0：A=0，B=1；反之为存储数据1：A=1，B=0。读出数字：以位线1的状态为准；写入0时，位线1=0，位线2=1；写入1时，位线1=1，位线2=0。VCC为电源，VSS为地。不论是写入还是读出操作，字线均为1，T5、T6导通，保持时，字线为0，T5、T6截止。

（2）**信息保持**。假设目前存储的数据为0，下面看看“0”是如何被保持的。按照约定，存储数据0：A=0，B=1，分析一下，可否保持？当处于数据保持时（即没有读写操作），字线为0，T5、T6截止，相当于没有这两个MOS管。由于A=0→T4导通，T2截止→B=1，又由于B=1→T3截止，T1导通→A=0，处于稳定状态。



（3）**信息读出**。0如何被读出？RAM单元处于0状态，即A=0，B=1。按照约定，读出数字以位线1的状态为准，只要分析出位线1=0即可。读出时，字线为1，T5、T6导通，位线1=A=0，位线2=B=1，系统对位线1进行采样，获得0。

（4）**信息写入**。如何写入1？按照约定，写入1时，位线1=1，位线2=0。设当前处于0状态，则即A=0，B=1。写入时，字线为1，T5、T6导通，则A=位线1=1；由于A=1→T4截止，T2导通→B=0（位线2=0在T6导通情况下，也保证了这一点）；由于B=0→T3导通，T1截止→A=1（位线1=1在T5导通情况下，也保证了这一点）。因此，该单元变成了保持，撤出写信号后，数据得以保持。

（5）**举一反三**。同理，可以分析其他情况。

2. SRAM的优缺点（了解）

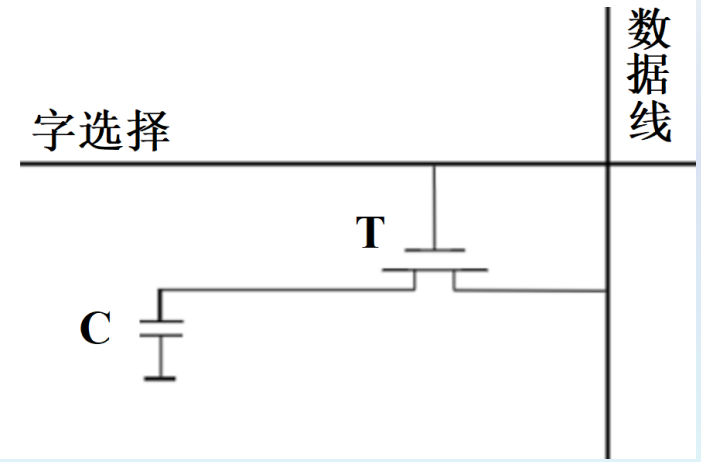
SRAM的**优点**是工作速度较高，性能稳定，不需要外加额外刷新电路，一般用于规模较小的快速存储器。**缺点**是每个存储单元需要由6个MOS管组成，集成度较低，功耗也较大，价格高（相对于DRAM）。在电路工作时，即使不进行读写操作，只要保持在加电状态下，电路中就一定有MOS管导通，带来功率消耗，因此，SRAM功耗较大，集成度不能做得很高。

对于SRAM，数据一旦写入，其信息就稳定的保存在电路中等待读出。无论读出多少次，只要不断电，此信息会一直保持下去。SRAM初始加电时，其状态是随机的。写入新的状态，原来的旧状态就消失了，新状态会一直维持到写入新的状态为止。

9.2.3 DRAM存储单元

1. DRAM的工作原理

常见的DRAM存储单元结构有四管MOS型、三管MOS型和单管MOS型，这里介绍单管MOS型DRAM存储单元的工作原理。单管MOS型动态存储单元结构如图9-8所示，由一个电容和一个访问MOS管组成。当存储单元被选中后，字选择线加载高电平，使得控制管T被打开，电流在数据线和存储电容C之间流动。写入1时，数据线呈高电平状态，电流通过T流入C中，给出C充电；写入0时，数据线呈低电平状态，将C中的电流导出，电容C放电，使其内部不存有正电荷。读出时，此时数据线处于输入状态，如果C中有正电荷，将有电流流过T管，拉升数据线上的电平状态；否则数据线仍保持低电平状态。可以说，动态随机存储器DRAM的存储单元是靠电容的充放电存储数据的。



DRAM的读取是破坏性的读取，一旦进行读取操作，将可能导致电容中失去正电荷，呈现无电荷状态。因此，必须在读出后进行重写工作，即还原读取前电容的存储状态。电容在实际工作中会有漏电的现象，从而导致内部存储正电荷不足而丢失存储数据。虽然读出后的重写工作可以为电容补充电荷，但是并不是每一个电容在电荷泄漏前都可以被访问到。因此，为了使其能正常工作，即使没有读取操作，也要进行周期性的重写工作，否则无法长期保持存储状态。正是由于需要周期性的定时**刷新**，这种利用电容存储电荷性质制成的存储器被称作DRAM（动态之含义）。

在电路上加上电源不进行读写及刷新操作时，只是保持在加电状态下，电路中没有MOS管导通，也就没有电流流过（会有极其微小的漏电流存在），也就没有功率消耗（或功耗忽略不计）。因此DRAM的功耗很小，其集成度可以做的很高，当前的一片DRAM芯片的集成度可以达到GB级别。常说的内存条，大多是由DRAM构成，随着时间发展，DRAM经历若干代变更，早期的PM DRAM、EDO DRAM均已淘汰，目前仍在使用的主要是SDRAM和DDR SDRAM。

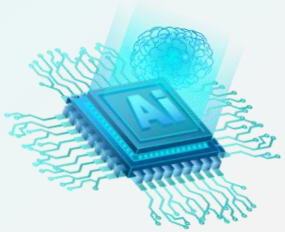
2．DRAM的优缺点

与SRAM相比，DRAM的主要优点是芯片集成度很高、功耗低，价格低（相对于SRAM），缺点是需外加刷新逻辑电路，读取速度慢。

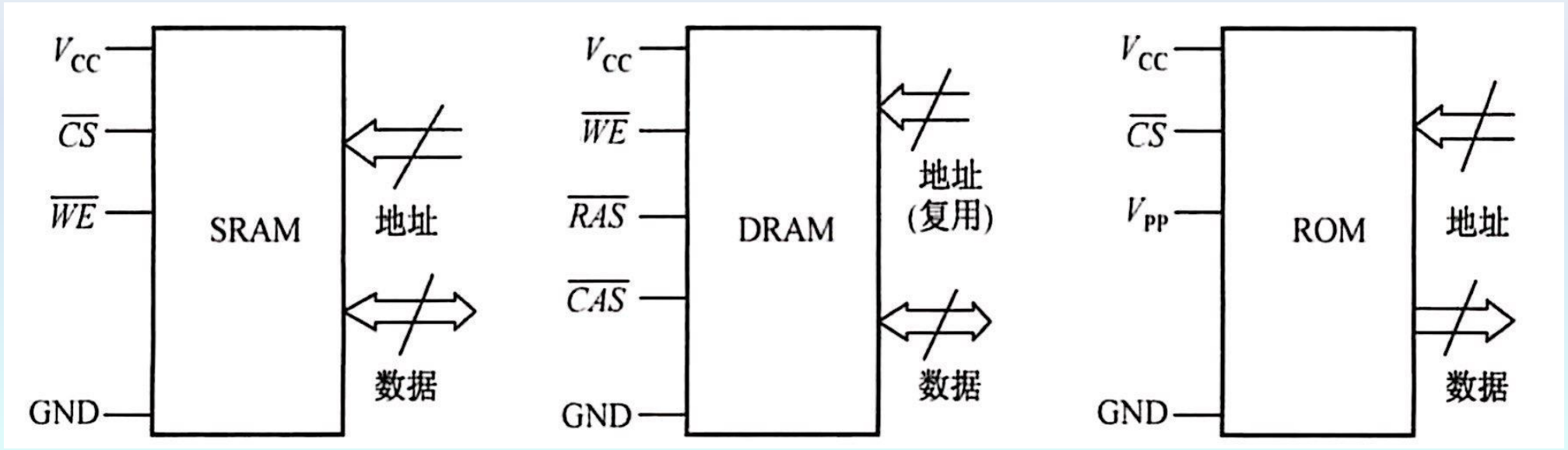
3．DRAM的刷新（了解）

9.3 存储器芯片的扩展（了解）

存储器芯片根据存储单元类型的不同，有SRAM存储器芯片、DRAM存储器芯片和ROM类型存储器芯片等，存储器芯片的容量有限，它在字数或字长方面与实际存储器的要求都有很大差距，因此可以将多片存储芯片连接起来，组成一个具有足够存储容量的内存条，这就需要在字方向和位方向上进行扩展。



9.3.1 存储器芯片介绍（了解）



9.3.2 存储容量的扩展方法（理解其基本含义）

1. 位扩展

位扩展的目的是增加同一个地址的存储单元的位数。例如用2片1K×4位的SRAM芯片组成1K×8位的存储器。

2. 字扩展

字扩展是指增加存储器中字的数量，即容量扩展。例如，用2片1K×8位的SRAM芯片组成2K×8位的存储器。

3. 字位扩展

字位扩展是指按字扩展的方法增加存储器字的数量，同时按位扩展的方法增加存储字长。

9.4 高速缓冲存储器Cache

现代计算机系统中，主存储器速度的提高始终跟不上CPU的发展，据统计，CPU的速度平均每年可以改进约60%，而组成主存储器的DRAM的速度平均每年只能改进7%左右，因此CPU和主存储器之间的速度间隙平均每年增大约50%。

为了解决CPU与主存之间速度不匹配的问题，可以在CPU与常规主存储器之间增设了一级或两级高速小容量存储器，称之为高速缓冲存储器Cache，它由高速的SRAM组成。把小容量、高速度的Cache存储器和大容量、低速度的主存储器组合起来，就可以得到速度与高速度存储器相当，且价格适中的存储器。

9.4.1 程序访问的局部性（重点）

对大量典型程序运行情况的分析结果表明，在较短的时间间隔内，程序访问的地址往往集中在存储器一个很小范围内，这种现象称为程序访问局部性，包括时间局部性和空间局部性。

时间局部性是指若某个存储器单元被访问，则在较短时间内仍有可能被访问。产生时间局部性的典型原因是程勋中存在着大量的循环操作。

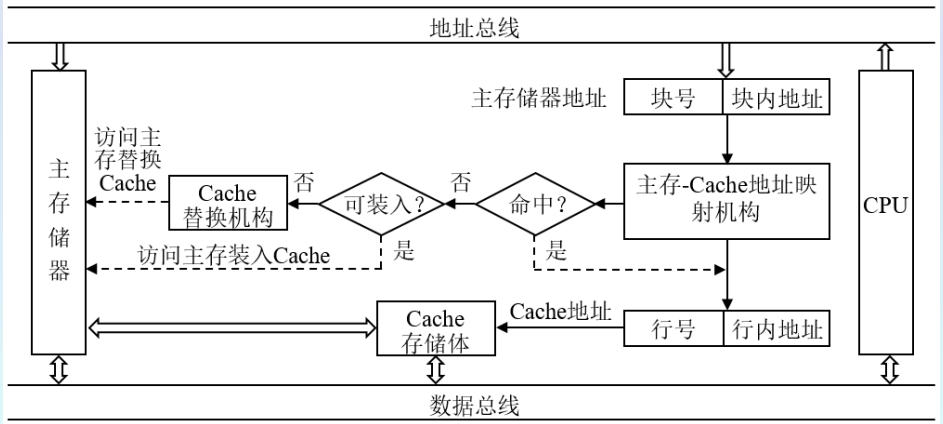
空间局部性是指若某个存储器单元被访问，则其邻近单元在较短时间内很可能被访问。空间局部性的典型情况便是程序的顺序执行。

根据程序访问的局部性原理，正在使用的主存储器单元临近的单元将被访问的可能性很大，因此把Cache介于CPU和主存储器之间，Cache中存储频繁使用的指令和数据，整个处理过程中，如果CPU绝大多数存取主存储器的操作都能被Cache所替代，那么计算机系统的处理速度就能明显提高。 注意的是，Cache不是主存储器容量的扩充，它保存的内容是主存储器中某些单元的副本。

9.4.2 Cache的工作原理

1．Cache的工作原理

在设计Cache系统时，主存和Cache都被分成若干个大小相等的数据块，每个数据块由若干个字节组成。为了便于区分，Cache中的数据块也称为行，行与块是等长的。由于Cache的容量远小于主存储器，所以Cache的数据行数远小于主存储器的数据块数，其保存的信息只是主存储器中最活跃的若干数据块的副本。把正在执行的指令代码单元及其附近的一部分指令代码或数据装入Cache中，由于程序访问的局部性原理，在一段时间内，CPU需要访问的指令或数据在Cache中的可能性较大，这样可以做到CPU大部分取指令代码及数据的读写操作都只需要访问Cache，而不需要访问主存储器，从而提高程序的运行速度。



2．Cache的命中率（了解）

从CPU 的角度看，增加Cache的目的就是为了在效果上使主存储器的速度尽可能接近CPU 速度。为了达到这个目的，在所有的存储器访问中由Cache满足CPU需要的部分应占很高的比例，即 Cache 的命中率应接近于1。

在一个程序执行期间，设N_c表示Cache完成存取的总次数（即命中次数），N_m表示主存储器完成存取的总次数，h定义为命中率，命中率指CPU访问主存数据时，命中 Cache的次数占全部访问次数的比率，则有：

$$h = \frac{N_c}{N_c + N_m}$$

9.4.3 Cache与主存储器的地址映射（了解）

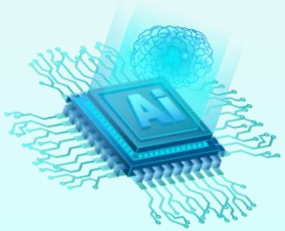
与主存储器相比，Cache的容量很小，它保存的内容只是主存储器内容的一个子集。为了把主存储器中的数据块放到Cache中，必须应用某种方法把主存储器地址定位到Cache 中，称为地址映射。主存地址与Cache地址间存在着对应关系，这一关系称为映射函数。地址映射完全由硬件实现，由于采用硬件，这个地址变换过程很快，软件人员丝毫感觉不到Cache的存在，这种特性称为Cache的透明性。

主存地址与Cache的地址映射方式有3种：直接映射、全相联映射和组相联映射。

9.4.4 Cache中数据行的替换算法（了解）

9.5 虚拟存储器

由于软件的规模不断扩大，有可能一个进程比现有可用主存大，如何运行该程序需要解决；另外，由于程序运行的局部性，如果一开始就分配所有内存空间，就会浪费空间。为了解决在有限内存空间下，运行存储在辅助存储器（辅存）上的规模大的程序，引入虚拟存储器概念，从逻辑上扩大内存空间。



9.5.1 虚拟存储器的基本概念

虚拟存储器技术是一种通过软、硬件结合的方法，在逻辑上扩大内存容量的技术。所谓**虚拟存储器**是指具有请求调入和置换功能，能够从逻辑上对内存容量进行扩充的一种存储器系统。程序预先放在外存储器（一般是硬盘）中，当计算机需要用到这一段程序时，程序调入内存，被CPU执行。从CPU角度看到的是一个速度接近内存却具有外存容量的虚构的存储器，虚拟存储器由此得名。

虚拟存储器技术的基本原理是：当程序运行时，不是将它的全部信息装入内存，而是将其中一部分装入内存，另一部分暂时留在辅存中，当需要运行留在辅存中的信息时，再由操作系统将它们装入内存，保证程序的正常运行。

在采用虚拟存储器的计算机系统中，编程空间对应整个**虚拟地址**空间或称为**逻辑地址空间**。虚拟地址由编译程序生成，内存空间称为**实地址空间**或称为**物理地址空间**，物理地址由CPU地址引脚送出。虚存空间远远大于物理地址空间。

9.5.2 虚拟存储器的实现方式（了解）

虚拟存储器的实现方式有3种：页式、段式和段页式。

1．页式虚拟存储器

页式虚拟存储器中，内存和辅存换进、换出的基本单位是定长的页面，页的大小都取2的整数幂个字。

2．段式虚拟存储器

段式虚拟存储器中，内存和辅存换进、换出的基本单位是段。段是指按照程序的逻辑结构划分成的多个相对独立的部分，例如，过程、子程序、一个数组或一张表等都可以作为一个段处理。

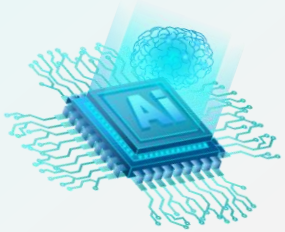
3．段页式虚拟存储器

页式虚拟存储器能够有效地提高内存利用率，而段式虚拟存储器则能很好地满足用户需要。将两者结合起来，取长补短，形成一种新的虚拟存储器的实现方式，其既具有段式虚拟存储器中分段共享、分段保护等优点，又如页式虚拟存储器般能够很好地解决存储“碎片”问题，这种虚拟存储器管理方式就称为段页式管理。

9.6 Flash存储器

1990年代，日本东芝公司和美国英特尔公司开始推出一种称之为Flash存储器的非易失存储器，简称闪存。它具有密度高、非易失性，兼有RAM和ROM的优点，且功耗低、集成度高。目前广泛应用的U盘和存储卡等都属于Flash存储器，MCU内部大多也集成了Flash存储器，用于存放程序及参数。

从2000年以来，MCU内部Flash开始逐步支持在线擦除与写入，即在线编程模式，使得本书的通用计算机GEC概念得以实现，把程序分成了BIOS及User独立又关联的两部分，方便了编程与调试。



9.6.1 Flash在线编程的通用基础知识（掌握基本概念）

Flash存储器具有固有不易失性、电可擦除、可在线编程、存储密度高、功耗低和成本较低等特点。

对于MCU内部的Flash存储器，擦除与写入模式有：写入器编程模式、在线编程模式。通过编程器将程序写入Flash存储器中的模式被称为写入器编程模式；通过运行Flash内部程序对Flash其他区域进行擦除与写入，称为Flash在线编程模式。

Flash编程的基本操作有两种：擦除（Erase）和写入（Program）。擦除操作的含义是将存储单元的内容由二进制的0变成1，而写入操作的含义是将存储单元的某些位由二进制的1变成0。

Flash在线编程的擦除操作包括整体擦除和以m个字为单位的擦除。这m个字在不同厂商或不同系列的MCU中，其称呼不同，有的称为“块”，有的称为“页”，有的称为“扇区”等等。它表示在线擦除的最小度量单位。

9.6.2 Flash驱动构件知识要素分析

序号	函数			形参	
	简明功能	返回	函数名	英文名	中文名
1	flash 初始化	无	flash_init		
2	扇区檫除	uint_16	flash_erase	sect	扇区号
3	向指定扇区写数据	uint_8	flash_write	sect	扇区号
				offset	偏移量
				N	写入数据长度
				buff	写入数组
4	向指定地址写数据	uint_8	flash_write_physical	addr	指定物理地址
				cnt	写入数据长度
				*buff	写入数据

9.6.3 Flash驱动构件的使用方法（编程实践）

样例工程： Flash

（实际运行）

Thank you