

(3) 串行访问存储器

如果对存储单元进行读/写操作时,需按其物理位置的先后顺序寻找地址,则这种存储器称为串行访问存储器。显然这种存储器由于信息所在位置不同,使得读/写时间均不相同。例如,磁带存储器,不论信息处在哪个位置,读/写时必须从其介质的始端开始按顺序寻找,故这类串行访问的存储器又称为顺序存取存储器。还有一种属于部分串行访问的存储器,如磁盘。在对磁盘读/写时,首先直接指出该存储器中的某个小区域(磁道),然后再顺序寻访,直至找到位置。故其前段是直接访问,后段是串行访问,称为直接存取存储器。

3. 按在计算机中的作用分类

按在计算机系统中的作用不同,存储器主要分为主存储器、辅助存储器、缓冲存储器。

主存储器(简称主存)的主要特点是它可以和 CPU 直接交换信息。辅助存储器(简称辅存)是主存储器的后援存储器,用来存放当前暂时不用的程序和数据,它不能与 CPU 直接交换信息。两者相比,主存速度快、容量小、每位价格高;辅存速度慢、容量大、每位价格低。缓冲存储器(简称缓存)用在两个速度不同的部件之中,例如, CPU 与主存之间可设置一个快速缓存(有关内容将在 4.3 节中讲述),起到缓冲作用。

综上所述,存储器分类如图 4.1 所示。

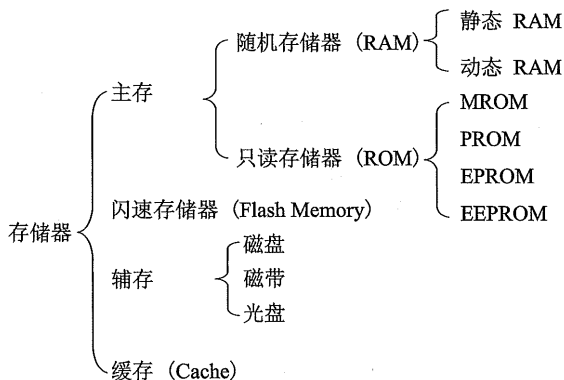


图 4.1 存储器分类

4.1.2 存储器的层次结构

存储器有 3 个主要性能指标:速度、容量和每位价格(简称位价)。一般来说,速度越高,位价就越高;容量越大,位价就越低,而且容量越大,速度必越低。人们追求大容量、高速度、低位价的存储器,可惜这是很难达到的。图 4.2 形象地反映了上述三者的关系。图中由上至下,位价越来越低,速度越来越慢,容量越来越大, CPU 访问的频度也越来越少。最上层的寄存器通常都制作在 CPU 芯片内。寄存器中的数直接在 CPU 内部参与运算, CPU 内可以有十几个、几十个寄存器,它们的速度最快,位价最高,容量最小。主存用来存放将要参与运行的

程序和数据,其速度与 CPU 速度差距较大,为了使它们之间速度更好地匹配,在主存与 CPU 之间插入了一种比主存速度更快、容量更小的高速缓冲存储器 Cache,显然其位价要高于主存。以上三类存储器都是由速度不同、位价不等的半导体存储材料制成的,它们都设在主机内。现代计算机将 Cache 也制作在 CPU 内。磁盘、磁带属于辅助存储器,其容量比主存大得多,大都用来存放暂时未用到的程序和数据文件。CPU 不能直接访问辅存,辅存只能与主存交换信息,因此辅存的速度可以比主存慢得多。

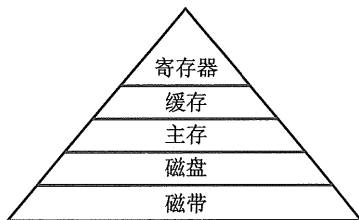


图 4.2 存储器速度、容量和位价的关系

实际上,存储系统层次结构主要体现在缓存-主存和主存-辅存这两个存储层次上,如图 4.3 所示。显然,CPU 和缓存、主存都能直接交换信息;缓存能直接和 CPU、主存交换信息;主存可以和 CPU、缓存、辅存交换信息。

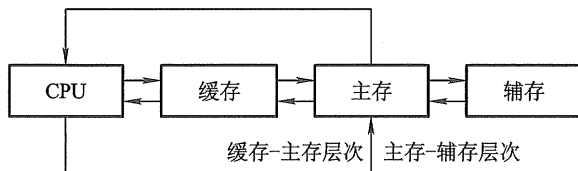


图 4.3 缓存-主存层次和主存-辅存层次

缓存-主存层次主要解决 CPU 和主存速度不匹配的问题。由于缓存的速度比主存的速度高,只要将 CPU 近期要用的信息调入缓存,CPU 便可以直接从缓存中获取信息,从而提高访存速度。但由于缓存的容量小,因此需不断地将主存的内容调入缓存,使缓存中原来的信息被替换掉。主存和缓存之间的数据调动是由硬件自动完成的,对程序员是透明的。

主存-辅存层次主要解决存储系统的容量问题。辅存的速度比主存的速度低,而且不能和 CPU 直接交换信息,但它的容量比主存大得多,可以存放大量暂时未用到的信息。当 CPU 需要用到这些信息时,再将辅存的内容调入主存,供 CPU 直接访问。主存和辅存之间的数据调动是由硬件和操作系统共同完成的。

从 CPU 角度来看,缓存-主存这一层次的速度接近于缓存,高于主存;其容量和位价却接近于主存,这就从速度和成本的矛盾中获得了理想的解决办法。主存-辅存这一层次,从整体分析,其速度接近于主存,容量接近于辅存,平均位价也接近于低速、廉价的辅存位价,这又解决了速度、容量、成本这三者的矛盾。现代的计算机系统几乎都具有这两个存储层次,构成了缓存、主存、辅存三级存储系统。

在主存-辅存这一层次的不断发展中,逐渐形成了虚拟存储系统。在这个系统中,程序员编程的地址范围与虚拟存储器的地址空间相对应。例如,机器指令地址码为 24 位,则虚拟存储器存储单元的个数可达 16 M。可是这个数与主存的实际存储单元的个数相比要大得多,称

这类指令地址码为虚地址(虚存地址、虚拟地址)或逻辑地址,而把主存的实际地址称为物理地址或实地址。物理地址是程序在执行过程中能够真正访问的地址,也是实实在在的主存地址。对具有虚拟存储器的计算机系统而言,程序员编程时,可用的地址空间远远大于主存空间,使程序员以为自己占有一个容量极大的主存,其实这个主存并不存在,这就是将其称为虚拟存储器的原因。对虚拟存储器而言,其逻辑地址变换为物理地址的工作是由计算机系统的硬件和操作系统自动完成的,对程序员是透明的。当虚地址的内容在主存时,机器便可立即使用;若虚地址的内容不在主存,则必须先将此虚地址的内容传递到主存的合适单元后再为机器所用。有关这些方面的内容,读者可在“计算机体系结构”和“操作系统”课程中学到。

4.2 主存储器

4.2.1 概述

主存储器(简称主存)的基本结构已在第1章介绍过,如图1.11所示。实际上,根据MAR中的地址访问某个存储单元时,还需经过地址译码、驱动等电路,才能找到所需访问的单元。读出时,需经过读出放大器,才能将被选中单元的存储字送到MDR。写入时,MDR中的数据也必须经过写入电路才能真正写入被选中的单元中。可见,主存的实际结构如图4.4所示。

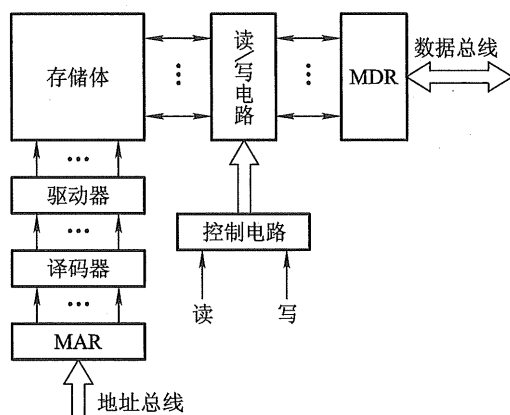


图 4.4 主存的基本组成

现代计算机的主存都由半导体集成电路构成,图中的驱动器、译码器和读写电路均制作在存储芯片中,而 MAR 和 MDR 制作在 CPU 芯片内。存储芯片和 CPU 芯片可通过总线连接,如图 4.5 所示。

当要从存储器读出某一信息字时,首先由 CPU 将该字的地址送到 MAR,经地址总线送至主存,然后发出读命令。主存接到读命令后,得知需将该地址单元的内容读出,便完成读操作,将该单元的内容读至数据总线上,至于该信息由 MDR 送至什么地方,这已不是主存的任务,而是由 CPU 决定的。若要向主存存入一个信息字时,首先 CPU 将该字所在主存单元的地址经 MAR 送到地址总线,并将信息字送入 MDR,然后向主存发出写命令,主存接到写命令后,便将数据线上的信息写入对应地址线指出的主存单元中。

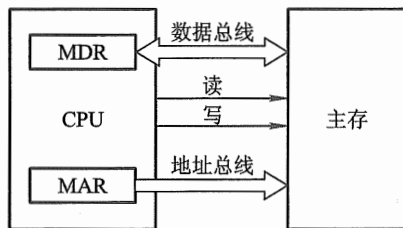


图 4.5 主存和 CPU 的联系

1. 主存中存储单元地址的分配

主存各存储单元的空间位置是由单元地址号来表示的,而地址总线是用来指出存储单元地址号的,根据该地址可读出或写入一个存储字。不同的机器存储字长也不同,为了满足字符处理的需要,常用 8 位二进制数表示一个字节,因此存储字长都取 8 的倍数。通常计算机系统既可按字寻址,也可按字节寻址。例如 IBM 370 机的字长为 32 位,它可按字节寻址,即它的每一个存储字包含 4 个可独立寻址的字节,其地址分配如图 4.6(a) 所示。字地址是用该字高位字节的地址来表示,故其字地址是 4 的整数倍,正好用地址码的末两位来区分同一字的 4 个字节的位置。但对 PDP-11 机而言,其字长为 16 位,字地址是 2 的整数倍,它用低位字节的地址来表示字地址,如图 4.6(b) 所示。

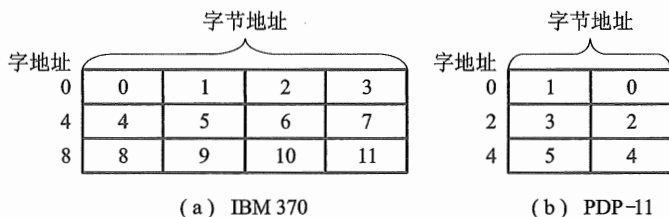


图 4.6 字节寻址的主存地址分配

由图 4.6(a) 所示,对 24 位地址线的主存而言,按字节寻址的范围是 16 M,按字寻址的范围为 4 M。由图 4.6(b) 所示,对 24 位地址线而言,按字节寻址的范围仍为 16 M,但按字寻址的范围为 8 M。

2. 主存的技术指标

主存的主要技术指标是存储容量和存储速度。

(1) 存储容量

存储容量是指主存能存放二进制代码的总位数,即

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长}$$

它的容量也可用字节总数来表示,即

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长} / 8$$

目前的计算机存储容量大多以字节数来表示,例如,某机主存的存储容量为 256 MB,则按字节寻址的地址线位数应对应 28 位。

(2) 存储速度

存储速度是由存取时间和存取周期来表示的。

存取时间又称为存储器的访问时间(Memory Access Time),是指启动一次存储器操作(读或写)到完成该操作所需的全部时间。存取时间分读出时间和写入时间两种。读出时间是从存储器接收到有效地址开始,到产生有效输出所需的全部时间。写入时间是从存储器接收到有效地址开始,到数据写入被选中单元为止所需的全部时间。

存取周期(Memory Cycle Time)是指存储器进行连续两次独立的存储器操作(如连续两次读操作)所需的最小间隔时间,通常存取周期大于存取时间。现代 MOS 型存储器的存取周期可达 100 ns;双极型 TTL 存储器的存取周期接近于 10 ns。

(3) 存储器带宽

与存取周期密切相关的指标为存储器带宽,它表示单位时间内存储器存取的信息量,单位可用字/秒或字节/秒或位/秒表示。如存取周期为 500 ns,每个存取周期可访问 16 位,则它的带宽为 32 M 位/秒。带宽是衡量数据传输率的重要技术指标。

存储器的带宽决定了以存储器为中心的机器获得信息的传输速度,它是改善机器瓶颈的一个关键因素。为了提高存储器的带宽,可以采用以下措施:

- ① 缩短存取周期。
- ② 增加存储字长,使每个存取周期可读/写更多的二进制位数。
- ③ 增加存储体(详见 4.2.7 节)。

4.2.2 半导体存储芯片简介

1. 半导体存储芯片的基本结构

半导体存储芯片采用超大规模集成电路制造工艺,在一个芯片内集成具有记忆功能的存储矩阵、译码驱动电路和读/写电路等,如图 4.7 所示。

译码驱动能把地址总线送来的地址信号翻译成对应存储单元的选择信号,该信号在读/写电路的配合下完成对被选中单元的读/写操作。

读/写电路包括读出放大器和写入电路,用来完成读/写操作。

存储芯片通过地址总线、数据总线和控制总线与外部连接。

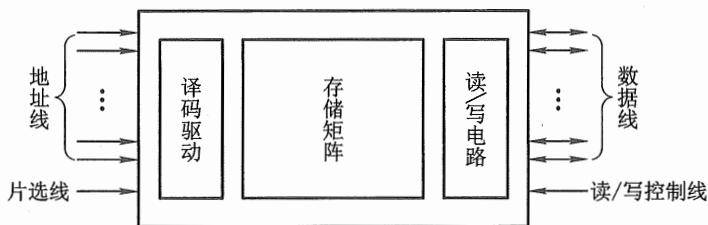


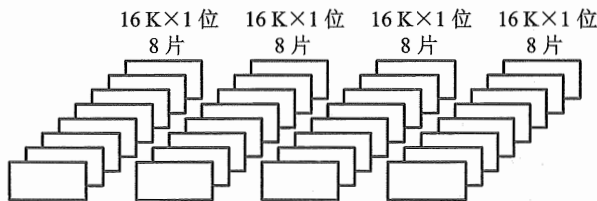
图 4.7 存储芯片的基本结构

地址线是单向输入的,其位数与芯片容量有关。

数据线是双向的(有的芯片可用成对出现的数据线分别作为输入或输出),其位数与芯片可读出或写入的数据位数有关。数据线的位数与芯片容量有关。

地址线和数据线的位数共同反映存储芯片的容量。例如,地址线为 10 根,数据线为 4 根,则芯片容量为 $2^{10} \times 4 = 4 \text{ K}$ 位;又如地址线为 14 根,数据线为 1 根,则其容量为 16 K 位。

控制线主要有读/写控制线与片选线两种。不同存储芯片的读/写控制线和片选线可以不同。有的芯片的读/写控制线共用 1 根(如 2114),有的分用两根(如 6264);有的芯片的片选线用 1 根(如 2114),有的用 2 根(如 6264)。读/写控制线决定芯片进行读/写操作,片选线用来选择存储芯片。由于半导体存储器是由许多芯片组成的,为此需用片选信号来确定哪个芯片被选中。例如,一个 $64 \text{ K} \times 8$ 位的存储器可由 32 片 $16 \text{ K} \times 1$ 位的存储芯片组成,如图 4.8 所示。但每次读出一个存储字时,只需选中 8 片。

图 4.8 $64 \text{ K} \times 8$ 位的存储器

2. 半导体存储芯片的译码驱动方式

半导体存储芯片的译码驱动方式有两种:线选法和重合法,如图 4.9 和图 4.10 所示。

图 4.9 是一个 16×1 字节线选法存储芯片的结构示意图。它的特点是用一根字选择线(字节线),直接选中一个存储单元的各位(如一个字节)。这种方式结构较简单,但只适于容量不大的存储芯片。如当地址线 $A_3A_2A_1A_0$ 为 1111 时,则第 15 根字节线被选中,对应图 4.9 中的最后一行 8 位代码便可直接读出或写入。

图 4.10 是一个 $1 \text{ K} \times 1$ 位重合法结构示意图。显然,只要用 64 根选择线(X、Y 两个方向各 32 根),便可选择 32×32 矩阵中的任一位。例如,当地址线为全 0 时,译码输出 X_0 和 Y_0 有效,矩阵中第 0 行、第 0 列共同选中的那位即被选中。由于被选单元是由 X、Y 两个方向的地

址决定的,故称为重合法。当欲构成 $1\text{K} \times 1$ 字节的存储器时,只需用 8 片如图 4.10 所示的芯片即可。

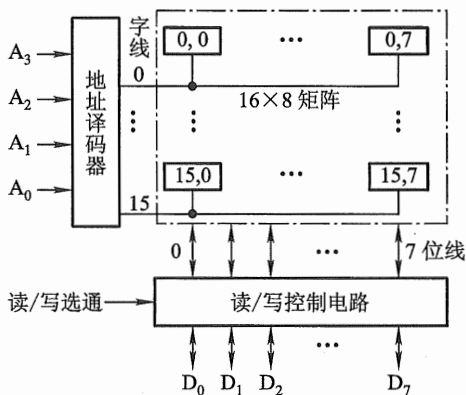


图 4.9 16×1 字节线选法结构示意图

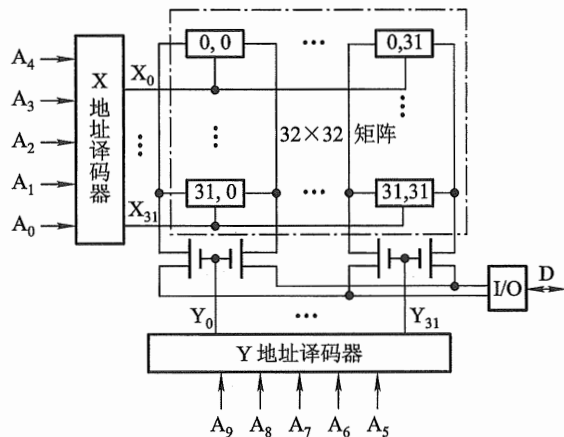


图 4.10 $1\text{K} \times 1$ 位重合法结构示意图

4.2.3 随机存取存储器

随机存取存储器按其存储信息的原理不同,可分为静态 RAM 和动态 RAM 两大类。

1. 静态 RAM (Static RAM, SRAM)

(1) 静态 RAM 基本单元电路

存储器中用于寄存“0”和“1”代码的电路称为存储器的基本单元电路,图 4.11 是一个由 6 个 MOS 管组成的基本单元电路。

图中 $T_1 \sim T_4$ 是一个由 MOS 管组成的触发器基本电路, T_5 、 T_6 犹如一个开关,受行地址选择信号控制。由 $T_1 \sim T_6$ 这 6 个 MOS 管共同构成一个基本单元电路。 T_7 、 T_8 受列地址选择控制,分别与位线 A' 和 A 相连,它们并不包含在基本单元电路内,而是芯片内同一列的各个基本单元电路所共有的。

假设触发器已存有“1”信号,即 A 点为高电平。当需读出时,只要使行、列地址选择信号均有效,则使 T_5 、 T_6 、 T_7 、 T_8 均导通, A 点高电平通过 T_6 后,再由位线 A 通过 T_8 作为读出放大器的输入信号,在读选择有效时,将“1”信号读出。

由于静态 RAM 是用触发器工作原理存储信息的,因此即使信息读出后,它仍保持其原状态,不需要再生。但电源掉电时,原存信息丢失,故它属易失性半导体存储器。

写入时不论触发器原状态如何,只要将写入代码送至图 4.11 的 D_{IN} 端,在写选择有效时,经两个写放大器,使两端输出为相反电平。当行、列地址选择有效时,使 T_5 、 T_6 、 T_7 、 T_8 导通,并将 A

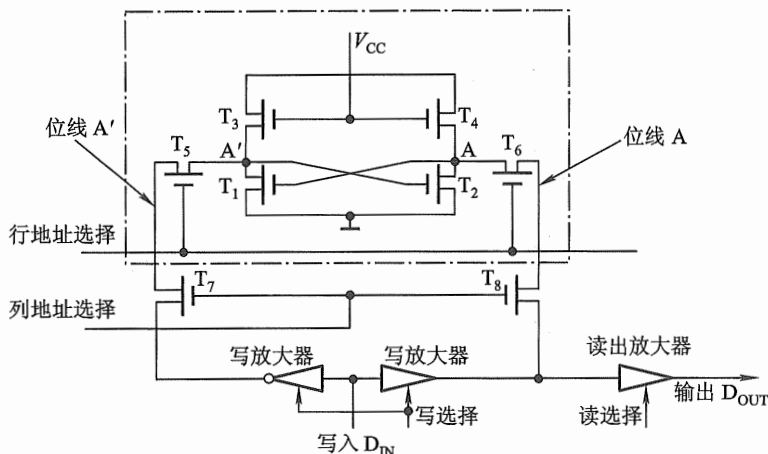


图 4.11 静态 RAM 的基本单元电路

与 A' 点置成完全相反的电平。这样,就把欲写入的信息写入该基本单元电路中。如欲写入“1”,即 $D_{IN}=1$,经两个写放大器使位线 A 为高电平,位线 A' 为低电平,结果使 A 点为高, A' 点为低,即写入了“1”信息。

(2) 静态 RAM 芯片举例

Intel 2114 芯片的基本单元电路由 6 个 MOS 管组成,图 4.12 是一个容量为 1 K×4 位的 2114 外特性示意图。

图中, $A_9 \sim A_0$ 为地址输入端; $I/O_1 \sim I/O_4$ 为数据输入输出端; \overline{CS} 为片选信号(低电平有效); \overline{WE} 为写允许信号(低电平为写,高电平为读); V_{CC} 为电源端; GND 为接地端。

2114 RAM 芯片的结构示意图如图 4.13 所示。图中存储矩阵由 64×64 个基本单元电路组成,列 I/O 电路即读/写电路。10 根地址线分为行地址 $A_8 \sim A_3$ 和列地址 A_9, A_2, A_1, A_0 , 4 根数据为 $I/O_4 \sim I/O_1$, 它们是受输入输出三态门控制的双向总线。当 \overline{CS} 和 \overline{WE} 均为低电平时,输入三态门打开, $I/O_4 \sim I/O_1$ 上的数据即写入指定地址单元中。当 \overline{CS} 为低电平、 \overline{WE} 为高电平时,输出三态门打开,列 I/O 电路的输出经片内总线输出至数据线 $I/O_4 \sim I/O_1$ 上。

2114 RAM 芯片内的存储矩阵结构如图 4.14 所示。其中每一个小方块均为一个由 6 个 MOS 管组成的基本单元电路,排列成 64×64 矩阵,64 列对应 64 对 T_7, T_8 管。又将 64 列分成 4 组,每组包含 16 列,并与一个读/写电路相连,读/写电路受 \overline{WE} 和 \overline{CS} 控制,4 个读/写电路对应 4 根数

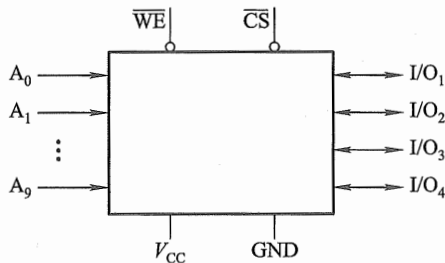


图 4.12 Intel 2114 外特性示意图

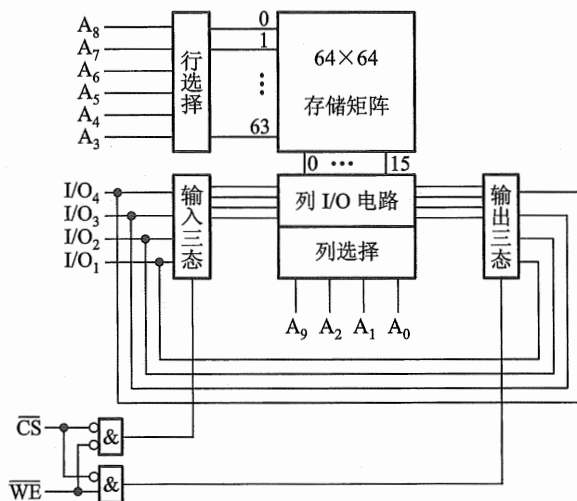


图 4.13 2114 RAM 芯片结构示意图

据线 $I/O_1 \sim I/O_4$ 。由图中可见,行地址经译码后可选中某一行;列地址经译码后可选中4组中的对应列,共4列。

当对某个基本单元电路进行读/写操作时,必须被行、列地址共同选中。例如,当 $A_9 \sim A_0$ 为全0时,对应行地址 $A_8 \sim A_3$ 为000000,列地址 A_9, A_2, A_1, A_0 也为0000,则第0行的第0、16、32、48这4个基本单元电路被选中。此刻,若做读操作,则 \overline{CS} 为低电平, \overline{WE} 为高电平,在读/写电路的输出端 $I/O_1 \sim I/O_4$ 便输出第0行的第0、16、32、48这4个基本单元电路所存的信息。若做写操作,将写入信息送至 $I/O_1 \sim I/O_4$ 端口,并使 \overline{CS} 为低电平、 \overline{WE} 为低电平,同样这4个输入信息将分别写入第0行的第0、16、32、48这4个单元之中。

(3) 静态 RAM 读/写时序

1) 读周期时序

图 4.15 是 2114 RAM 芯片读周期时序,在整个读周期中 \overline{WE} 始终为高电平(故图中省略)。读周期 t_{RC} 是指对芯片进行两次连续读操作的最小间隔时间。读时间 t_A 表示从地址有效到数据稳定所需的时间,显然读时间小于读周期。图中 t_{CO} 是从片选有效到输出稳定的时间。可见只有当地址有效经 t_A 后,且当片选有效经 t_{CO} 后,数据才能稳定输出,这两者必须同时具备。根据 t_A 和 t_{CO} 的值,便可知当地址有效后,经 $t_A - t_{CO}$ 时间必须给出片选有效信号,否则信号不能出现在数据线上。

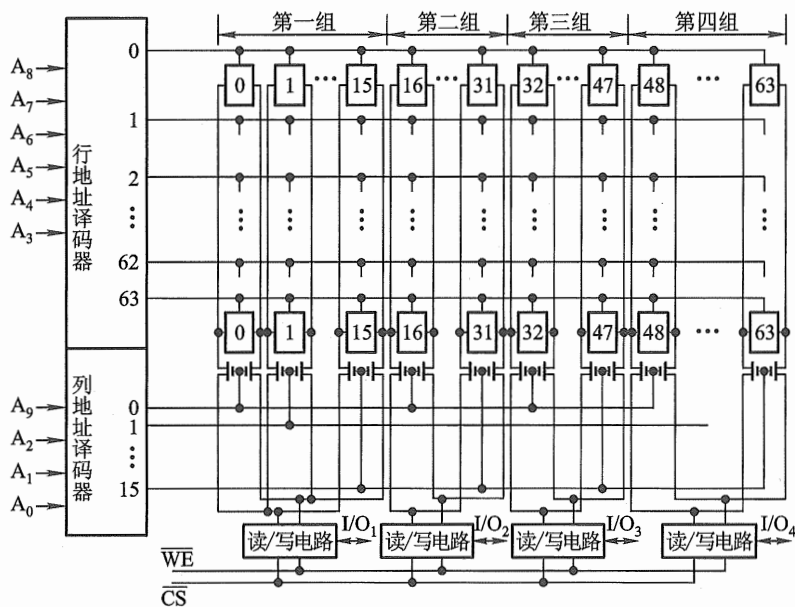


图 4.14 2114 RAM 矩阵结构示意图

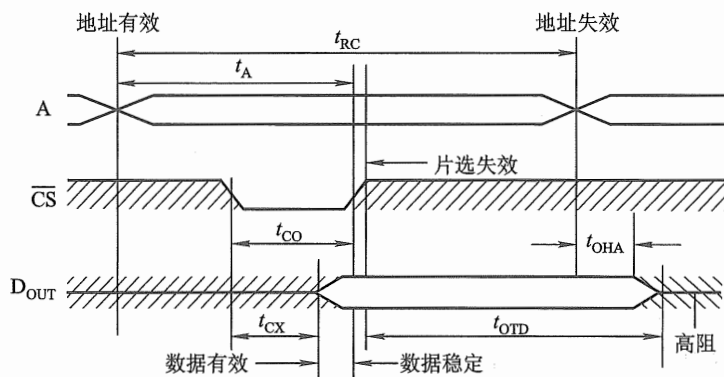


图 4.15 2114 RAM 的读周期时序

需注意一点,从片选失效到输出高阻需一段时间 t_{OTD} ,故地址失效后,数据线上的有效数据有一段维持时间 t_{OHA} ,以保证所读的数据可靠。

2) 写周期时序

图 4.16 是 2114 RAM 写周期时序。

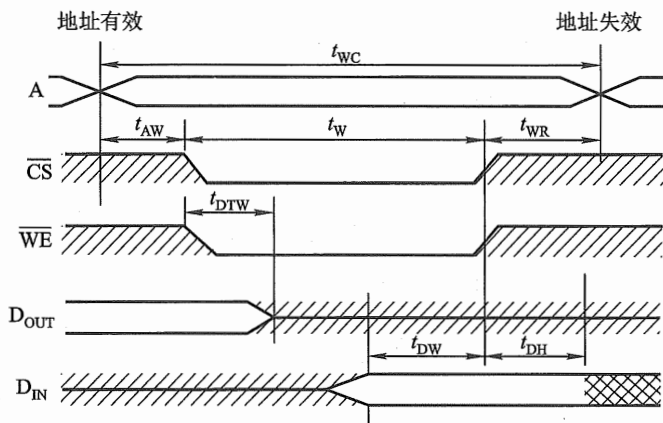


图 4.16 2114 RAM 的写周期时序

写周期 t_{WC} 是对芯片进行连续两次写操作的最小间隔时间。写周期包括滞后时间 t_{AW} 、写入时间 t_W 和写恢复时间 t_{WR} 。在有效数据出现前, RAM 的数据线上存在着前一时刻的数据 D_{OUT} (如图 4.15 所示的维持时间), 故在地址线发生变化后, \overline{CS} 、 \overline{WE} 均需滞后 t_{AW} 再有效, 以避免将无效数据写入 RAM 的错误。但写允许 \overline{WE} 失效后, 地址必须保持一段时间, 称为写恢复时间。此外, RAM 数据线上的有效数据 (即 CPU 送至 RAM 的写入数据 D_{IN}) 必须在 \overline{CS} 、 \overline{WE} 失效前的 t_{DW} 时刻出现, 并延续一段时间 t_{DH} (此刻地址线仍有效, $t_{WR} > t_{DH}$), 以保证数据可靠写入。

已制成的 RAM 芯片读写时序关系已被确定, 因此, 将它与 CPU 连接时, 必须注意它们相互的时序匹配关系, 否则 RAM 将无法正常工作。具体 RAM 芯片的读/写周期时序可查看相关资料。

值得注意的是, 不论是对存储器进行读操作还是写操作, 在读周期和写周期内, 地址线上的地址始终不变。

2. 动态 RAM (Dynamic RAM, DRAM)

(1) 动态 RAM 的基本单元电路

常见的动态 RAM 基本单元电路有三管式和单管式两种, 它们的共同特点都是靠电容存储电荷的原理来寄存信息。若电容上存有足够多的电荷表示存“1”, 电容上无电荷表示存“0”。电容上的电荷一般只能维持 1~2 ms, 因此即使电源不掉电, 信息也会自动消失。为此, 必须在 2 ms 内对其所有存储单元恢复一次原状态, 这个过程称为再生或刷新。由于它与静态 RAM 相比, 具有集成度更高、功耗更低等特点, 目前被各类计算机广泛应用。

图 4.17 示意了由 T_1 、 T_2 、 T_3 这 3 个 MOS 管组成的三管 MOS 动态 RAM 基本单元电路。

读出时, 先对预充电管 T_4 置一预充电信号 (在存储矩阵中, 每一列共用一个 T_4 管), 使数据线达高电平 V_{DD} 。然后由读选择线打开 T_2 , 若 T_1 的极间电容 C_g 存有足够多的电荷 (被认为原

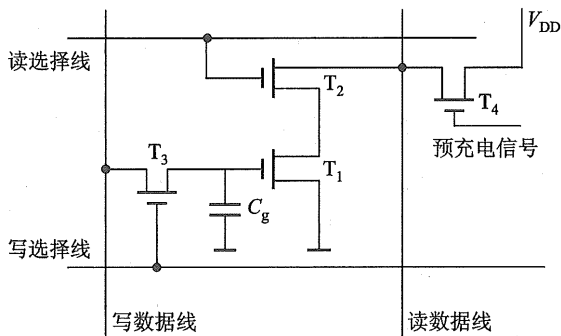


图 4.17 三管 MOS 动态 RAM 基本单元电路

存“1”)，使 T_1 导通，则因 T_2 、 T_1 导通接地，使读数据线降为零电平，读出“0”信息。若 C_g 没有足够电荷(原存“0”)，则 T_1 截止，读数据线为高电平不变，读出“1”信息。可见，由读出线的高低电平可区分其是读“1”，还是读“0”，只是它与原存信息反相。

写入时，将写入信号加到写数据线上，然后由写选择线打开 T_3 ，这样， C_g 便能随输入信息充电(写“1”)或放电(写“0”)。

为了提高集成度，将三管电路进一步简化，去掉 T_1 ，把信息存在电容 C_s 上，将 T_2 、 T_3 合并成一个管子 T ，便得到单管 MOS 动态 RAM 基本单元电路，如图 4.18 所示。

读出时，字线上的高电平使 T 导通，若 C_s 有电荷，经 T 管在数据线上产生电流，可视作读出“1”。若 C_s 无电荷，则数据线上无电流，可视作读出“0”。读操作结束时， C_s 的电荷已释放完毕，故是破坏性读出，必须再生。

写入时，字线为高电平使 T 导通，若数据线上为高电平，经 T 管对 C_s 充电，使其存“1”；若数据线上为低电平，则 C_s 经 T 放电，使其无电荷而存“0”。

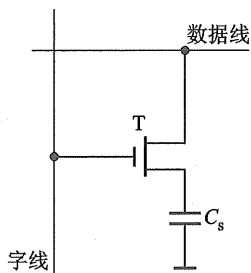


图 4.18 单管 MOS 动态 RAM 基本单元电路

(2) 动态 RAM 芯片举例

1) 三管动态 RAM 芯片

三管动态 RAM 芯片结构的示意图如图 4.19 所示。

这是一个 1 K×1 位的存储芯片，图中每一小方块代表由 3 个 MOS 管组成的动态 RAM 基本单元电路。它们排列成 32×32 的矩阵，每列都有一个刷新放大器(用来形成再生信息)和一个预充电管(图中未画)，芯片有 10 根地址线，采用重合法选择基本单元电路。

读出时，先置以预充电信号，接着按行地址 $A_9 \sim A_5$ 经行译码器给出读选择信号，同时由列地址 $A_4 \sim A_0$ 经列译码器给出列选择信号。只有在行、列选择信号共同作用下的基本单元电路才能将其信息经读数据线送到读/写控制电路，并从数据线 D 输出。

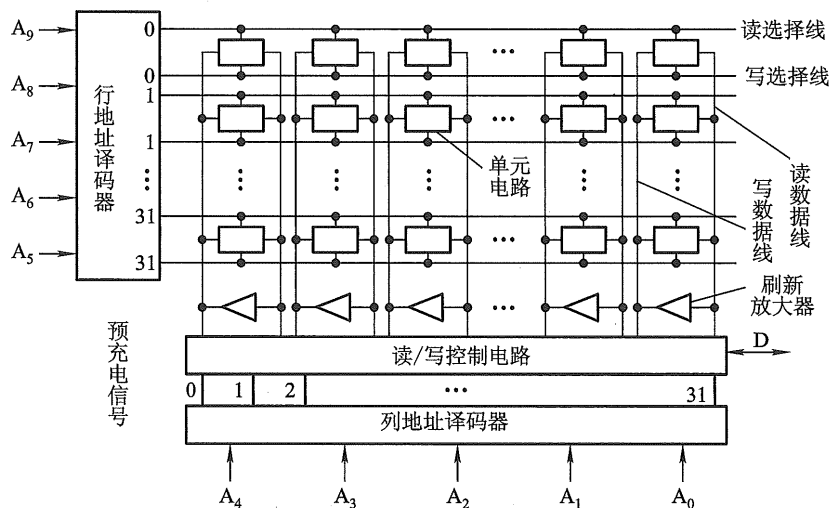


图 4.19 1K×1 位三管 MOS 动态 RAM 结构示意图

写入时,在受行地址控制的行译码器给出的写选择信号的作用下,选中芯片的某一行,并在列地址的作用下,由列译码器的输出控制读/写控制电路,只将数据线 D 的信息送到被选中列的写数据线上,信息即被写入行列共同选中的基本单元电路中。

2) 单管动态 RAM 芯片

单管动态 RAM 芯片结构的示意图如图 4.20 所示。这是一个 16 K×1 位的存储芯片,按理应有 14 根地址线,但为了减少芯片封装的引脚数,地址线只有 7 根。因此,地址信息分两次传送,先送 7 位行地址保存到芯片内的行地址缓存器内,再送 7 位列地址保存到列地址缓存器中。芯片内有时序电路,它受行地址选通 $\overline{\text{RAS}}$ 、列地址选通 $\overline{\text{CAS}}$ 以及写允许信号 $\overline{\text{WE}}$ 控制。

16 K×1 位的存储芯片共有 16 K 个单管 MOS 基本单元电路,它们排列成 128×128 的矩阵,如图 4.21 所示。图中的行线就是图 4.18 中的字线,列线就是图 4.18 中的数据线。128 行分布在读放大器的左、右两侧(左侧为 0~63 行,右侧为 64~127 行)。每根行选择线与 128 个 MOS 管的栅极相连。128 列共有 128 个读放大器,它的两侧又分别与 64 个 MOS 管相连,每根列线上都有一个列地址选择管。128 个列地址选择管的输出又互相并接在一起与 I/O 缓冲器相连,I/O 缓冲器的一端接输出驱动器,可输出数据;另一端接输入器,供数据输入。

读出时,行、列地址受 $\overline{\text{RAS}}$ 和 $\overline{\text{CAS}}$ 控制,分两次分别存入行、列地址缓存器。行地址经行译码后选中一行,使该行上所有的 MOS 管均导通,并分别将其电容 C_s 上的电荷反映到 128 个读放大器的某一侧(第 0~63 行反映到读放大器的左侧,第 64~127 行反映到读放大器的右侧)。读

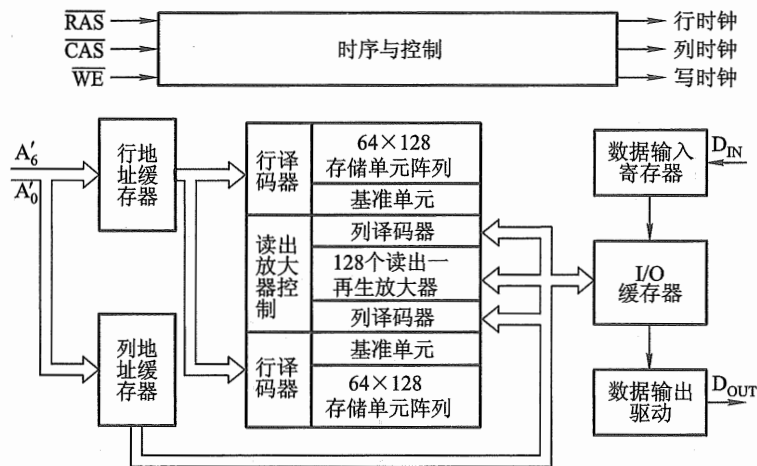


图 4.20 4116 动态 RAM (16 K×1 位) 芯片的结构

放大器的工作原理像一个跷跷板电路,类似于一个触发器,其左右两侧电平相反。此外列地址经列译码后选中某一列,该列上的列地址选择管导通,即可将读放大器右侧信号经读/写线、I/O 缓冲器输出至 D_{OUT} 端。例如,选中第 63 行、第 0 列的单管 MOS 电路,若其 C_s 有电荷为“1”状态,则反映到第 0 列读放大器的左侧为“1”,右侧为“0”,经列地址选择管输出至 D_{OUT} 为 0,与原存信息反相。同理,第 0~62 行经读放大器至输出线 D_{OUT} 的信息与原存信息均反相。而读出第 64~127 行时,因它们的电容 C_s 上的电荷均反映到读放大器的右侧,故经列地址选择管输出至 D_{OUT} 的信息均同相。

写入时,行、列地址也要分别送入芯片内的行、列地址缓存器,经译码可选中某行、某列。输入信息 D_{IN} 通过数据输入器,经 I/O 缓冲器送至读/写线上,但只有被选中的列地址选择管导通,可将读/写线上的信息送至该列的读放大器右侧,破坏了读放大器的平衡,使读放大器的右侧与输入信息同相,左侧与输入信息反相,读放大器的信息便可写入选中行的 C_s 中。例如,选中第 64 行、第 127 列,输入信息为“1”,则第 127 列地址选择管导通,将“1”信息送至第 127 列的读放大器的右侧。虽然第 64 行上的 128 个 MOS 管均导通,但唯有第 64 行、第 127 列的 MOS 管能将读放大器的右侧信息“1”对 C_s 充电,使其写入“1”。值得注意的是,写入读放大器左侧行的信息与输入信息都是反相的,而由读出过程分析又知,对读放大器左侧行进行读操作时,读出的信息也是反相的,故最终结果是正确的。

(3) 动态 RAM 时序

由图 4.20 可知,动态 RAM 的行、列地址是分别传送的,因此分析其时序时,应特别注意 \overline{RAS} 、 \overline{CAS} 与地址的关系,即

- 先由 \overline{RAS} 将行地址送入行地址缓存器,再由 \overline{CAS} 将列地址送入列地址缓存器,因此,

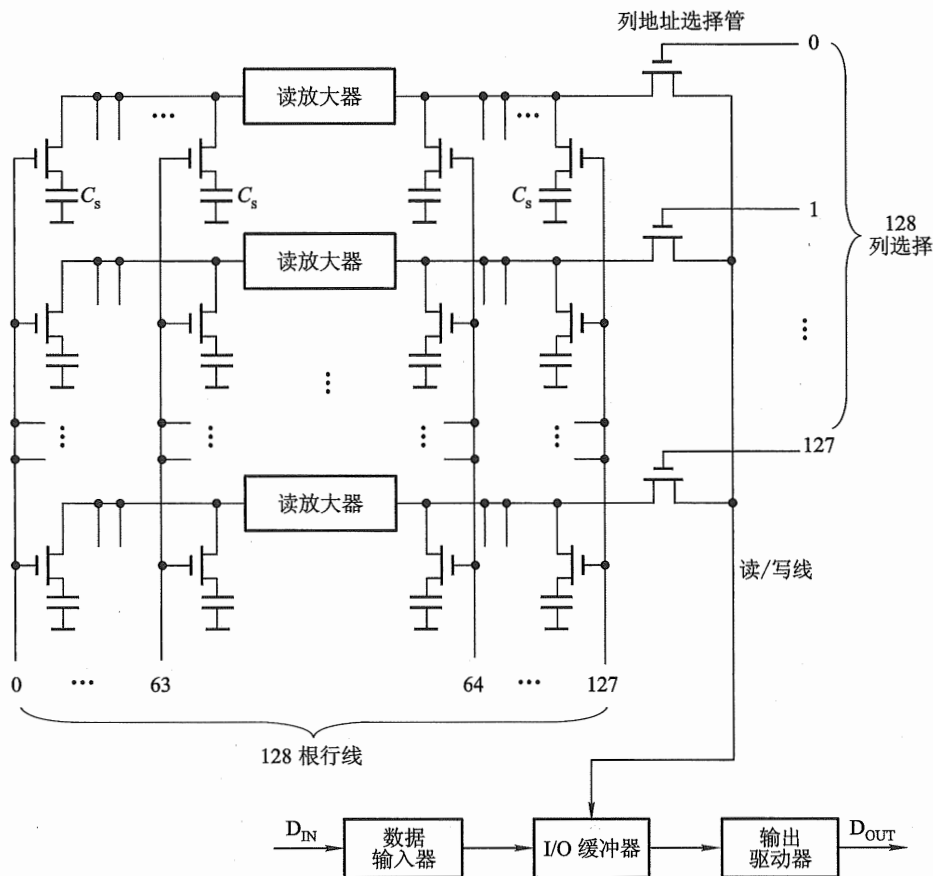


图 4.21 16 K×1 位 4116 动态 RAM 存储矩阵示意图

$\overline{\text{CAS}}$ 滞后于 $\overline{\text{RAS}}$ 的时间必须要超过其规定值。

- $\overline{\text{RAS}}$ 和 $\overline{\text{CAS}}$ 正、负电平的宽度应大于规定值,以保证芯片内部正常工作。
- 行地址对 $\overline{\text{RAS}}$ 的下降沿以及列地址对 $\overline{\text{CAS}}$ 的下降沿应有足够的地址建立时间和地址保持时间,以确定行、列地址均能准确写入芯片。

1) 读时序

在读工作方式时(写允许 $\overline{\text{WE}}=1$),读工作周期是指动态 RAM 完成一次“读”所需的最短时间 t_{CRD} ,也是 $\overline{\text{RAS}}$ 的一个周期。如图 4.22 所示,为了确保读出数据无误,必须要求写允许 $\overline{\text{WE}}=1$ 在列地址送入前(即 $\overline{\text{CAS}}$ 下降沿到来前)建立,而 $\overline{\text{WE}}=1$ 的撤除应在 $\overline{\text{CAS}}$ 失效后(即 $\overline{\text{CAS}}$ 上升沿后);还要求读出数据应在 $\overline{\text{RAS}}$ 有效后一段时间 t_{aRAS} 且 $\overline{\text{CAS}}$ 有效后一段时间 t_{aCAS} 时出现,而数据有效的撤除时间应在 $\overline{\text{CAS}}$ 失效后一段时间 $t_{\text{bCAS-OUT}}$ 。