



第3章 存储系统

存储系统由赋予计算机记忆能力的部件组成。随着以存储器为中心的系统结构的建立，尤其是共享主存储器的多处理机的出现，存储系统的特性，已经成为影响整个系统最大吞吐量的决定性因素。

3.1 主存储器原理

现代主存储器一般被分为只读存储器 (read only memory, ROM) 和随机存储器 (random access memory, RAM) 两部分。如图 3.1 所示，ROM 和 RAM 还可以有不同的类型。

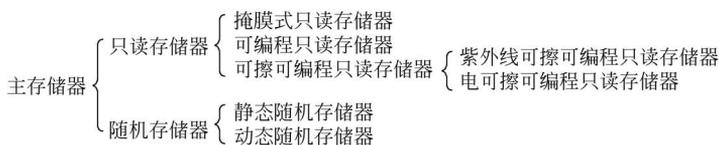


图 3.1 主存储器的基本分类

3.1.1 只读存储器原理

只读存储器是一种在机器运行过程中只能读出、不能写入信息的无源存储器，由非易失性器件组成，主要用于存储经常要用的一些固定信息，如基本输入输出系统 (basic input/output system, BIOS)。

1. 掩膜式只读存储器

掩膜式只读存储器 (mask read only memory, MROM) 采用由生产商进行编程的 ROM 元件，用户只能将自己的要求提供给生产商进行制作，自己无能为力。其优点是可靠性高、集成度高，可以在生产线上生产，大批量应用价格便宜；缺点是少量应用成本很高。

2. 可编程只读存储器

可编程只读存储器 (programmable read only memory, PROM) 在出厂时内部并没有存储任何数据，用户可以用专用编程器将自己的数据写入，但是这种机会只有一次，一旦写入也无法修改，若是出了错误，已写入的芯片只能报废。PROM 元件有多种形式，其中一种是熔丝型的，其原理如图 3.2 所示。它在出厂时各处熔丝都是完好的，用户在使用前可以将要存 0 的位用大电流将熔丝烧断，没有烧断的位就表示 1。这种元件，一经写好，存 0 的位便不可再改为 1。读出时，选中 w_0 (简单地认为其为高电平)，则在位线 $b_0b_1b_2$ 上

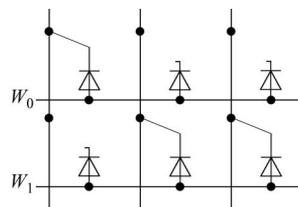


图 3.2 熔丝型 PROM

分别输出 100；选中 W_1 ，则在位线 $b_0b_1b_2$ 上分别输出 011。

3. 可擦可编程只读存储器

可擦可编程只读存储器（erasable programmable read only memory, EPROM）的特点是可以改写。它们在出厂时，全写为 1，用户可以根据需要将某些位改写为 0。当需要变更时，还可以将每位都擦除——恢复全 1，重新写入新内容。按照擦除方式，EPROM 可以分为两种：紫外线可擦可编程只读存储器（UVEEPROM）和电可擦可编程只读存储器（EEPROM）。

UVEEPROM 的存储单元由金属-氧化物-半导体场效应晶体管（metal-oxide-semiconductor field effect, MOSFET）构成。如图 3.3 所示，它在控制栅 G_2 和 N 沟道间有一个称为浮空栅 G_1 。浮空栅利用氧化膜使栅极与基板绝缘，可以使存储于此处的电荷不能被轻易释放，以持续保存记忆。通常，浮空栅中未存储电荷时，为高电平状态；有存储电荷时，为低电平状态。

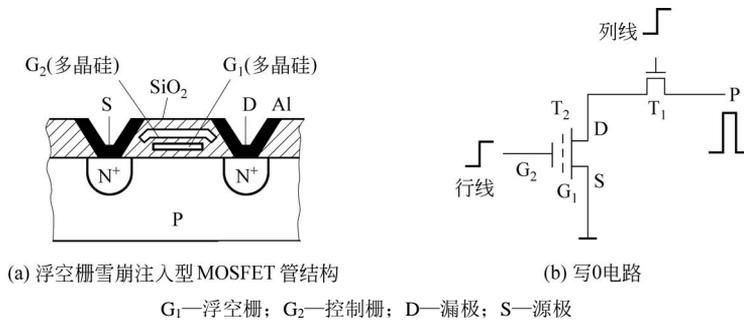


图 3.3 浮空栅雪崩注入型 EPROM

若需要对芯片内容进行改写，首先要将已存的内容擦除。如图 3.4(a)所示，在 UVEEPROM 芯片正面的陶瓷封装上，开有一个玻璃窗口，即石英窗，透过该窗口可以看到其内部的集成电路。如图 3.4 (b) 所示，当 40W 紫外线透过该孔照射到内部芯片中的 G_1 几分钟， G_1 中的电子即可获得能量，穿过氧化层回到衬底中。 G_1 电荷的消失，相当于抹去信息，源极与漏极之间导通，存储器中又都成为存 1 状态。鉴此，UVEEPROM 芯片在写入数据后，要以不透光贴纸或胶布把窗口封住，以免受到意外紫外线照射而使数据受损。

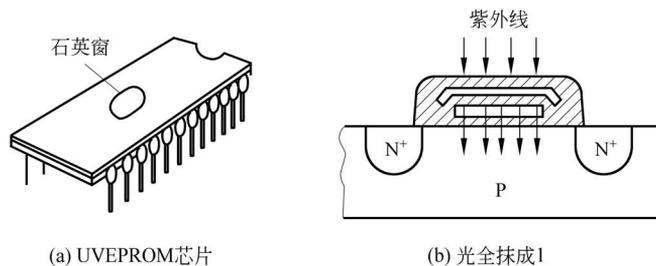


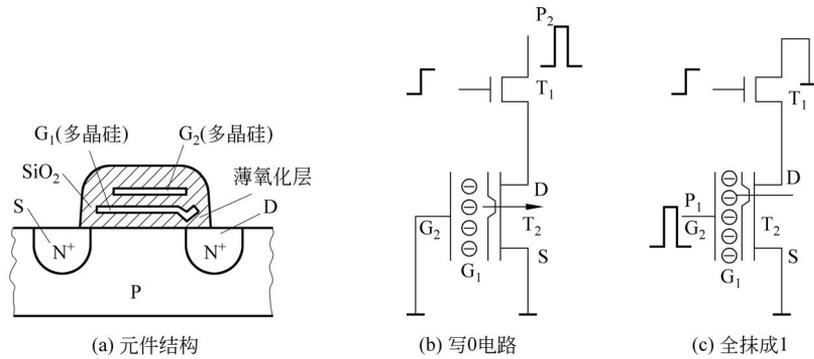
图 3.4 UVEEPROM 元件的原理

UVEEPROM 虽然使用很广泛，但也存在两个问题。

(1) 紫外线擦除信息需很长时间（与紫外线光的照射强度有关）；

(2) 不能把芯片中个别需要改写的存储单元单独擦除和重写。

如图 3.5 (a) 所示, EEPROM 是在 EPROM 基本单元电路的控制栅 G_1 的上面再生成一个抹去栅 G_2 , 可给 G_2 引出一个电极, 使其接某一电压 V_G ; G_1 与漏极 D 之间有薄氧化层。如图 3.5 (b) 所示, 若 V_G 为正电压, G_1 与 D 之间产生隧道效应, 使电子注入控制栅, 即编程写入 0。如图 3.5 (c) 所示, 若 V_G 为负电压, 将使控制栅的电子散失, 即擦除——全抹成 1。擦除后可重新写入。



G_1 —浮空栅 (控制栅); G_2 —抹去栅; D —漏极

图 3.5 EEPROM 元件的原理

3.1.2 随机存储器原理

按照工作原理, 随机存储器分为静态随机存储器 (static random access memory, SRAM) 和动态随机存储器 (dynamic random access memory, DRAM)。

1. SRAM 原理

1) SRAM 元件

SRAM 记忆元件有双极型和金属-氧化物-半导体 (metal-oxide-semiconductor, MOS) 开关两种。下面以 MOS 开关元件为例说明 SRAM 的工作原理。

MOS 开关元件是由金属 (M)、氧化物 (O) 和半导体 (S) 组成的场效应管。它有三极 (源极 S、漏极 D 和栅极 G)、两态 (导通与截止): 当栅极电压达到某个阈值 (槛压) 时, 源极与栅极导通; 低于这个阈值时源极与栅极截止。图 3.6 为 MOSFET 的简化符号。

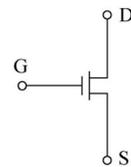


图 3.6 MOSFET 的简化符号

图 3.7 为 6 管静态 MOS 存储单元。其中, T_1 和 T_2 反向耦合组成一个双稳态触发器 (当 T_1 导通时, T_2 就会截止; 当 T_1 截止时, T_2 就会导通), T_3 和 T_4 作为阻抗, T_5 和 T_6 作为记忆单元的选中开关 (读写控制门)。当记忆单元未被选中 (字线保持低电平) 时, T_5 、 T_6 截止, 触发器与位线隔开, 原来保存的状态不改变。当字线加上高电平时, T_5 、 T_6 导通, 该记忆单元被选中, 可进行读写操作。

字线 W 是一个存储单元的一字节 (面向字节的计算机) 或一个字 (面向字的计算机) 中的所有位是共有的。选中字线, 就是该存储单元中的所有记忆元件都被激活, 便可以进行读写操作了。

写过程：选中字线 W ， T_5 、 T_6 导通，即读写控制门打开。写 1 时，位线 b' 上送高电平，使 T_2 导通，位线 b 上送低电平，使 T_1 截止。这种状态不因写脉冲的撤离而改变。因为 T_2 、 T_1 成反向耦合，只要 V_{CC} 上有 +5V 的电位，就能保持这一状态。写 0 时，位线 b' 上送低电平，位线 b 上送高电平，使 T_2 截止、 T_1 导通。

读过程：选中字线 W ，位线 b' 和 b 分别与 A 点和 B 点相通。记忆单元若原存 1，A 点（即位线 b' ）为高电平（读 1）；若原存 0，B 点（即位线 b ）为高电平（读 0）。

静态 MOS 存储单元具有非破坏性读出的特点，抗干扰能力强，可靠性高。但是存储单元电路所用管子数目较多，占硅片面积大，功耗大，集成度不高。

双极型一般可分为晶体管-晶体管逻辑（transistor-transistor logic, TTL）和射极耦合逻辑（emitter coupled logic, ECL）两种。它们的电路驱动能力强，存取速度快，一般用作高速缓冲存储器。近年来还出现了新型的双极型存储单元电路，如集成注入逻辑（integrated injection logic）电路，简称 I²L 电路。它的特点是集成度高，工作电压低，功耗小，可靠性高，速度较快。双极型存储单元电路样式繁多，这里不再介绍。

2) 一维地址译码 RAM 中的信号及其时序

图 3.8 为字结构、一维译码方式的 RAM 逻辑结构，通常用于小容量的 SRAM。它在读写过程中，要使用如下一些信号。

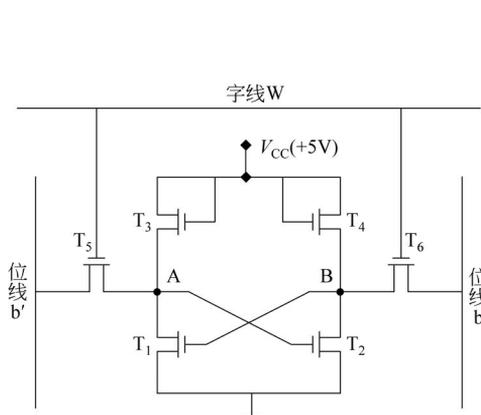


图 3.7 6管静态 MOS 存储单元

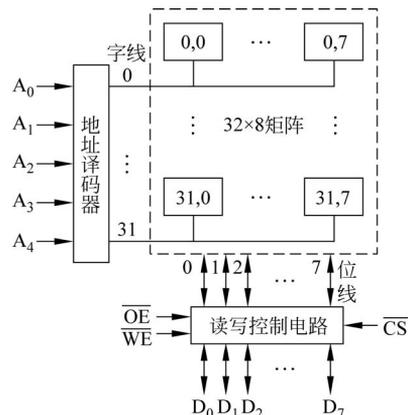


图 3.8 字结构、一维译码方式的 RAM 逻辑结构

- (1) 数据总线上的地址信号。
- (2) 片选信号 \overline{CS} 。
- (3) 写入控制信号 \overline{WE} 。
- (4) 读出使能信号 \overline{OE} 。
- (5) 数据总线上的数据信号。

图 3.9 所示的读写时序，形象地描画了这类 SRAM 在读写过程中，各种信号之间的时序关系。

如图 3.9 所示，不管是读还是写，都是按照下面的顺序发送有关信号。

- (1) 送地址信号，指定要读写的单元，此信号一直保留到有效读写完成。
- (2) 发送片选信号 \overline{CS} ，选中一片存储器。

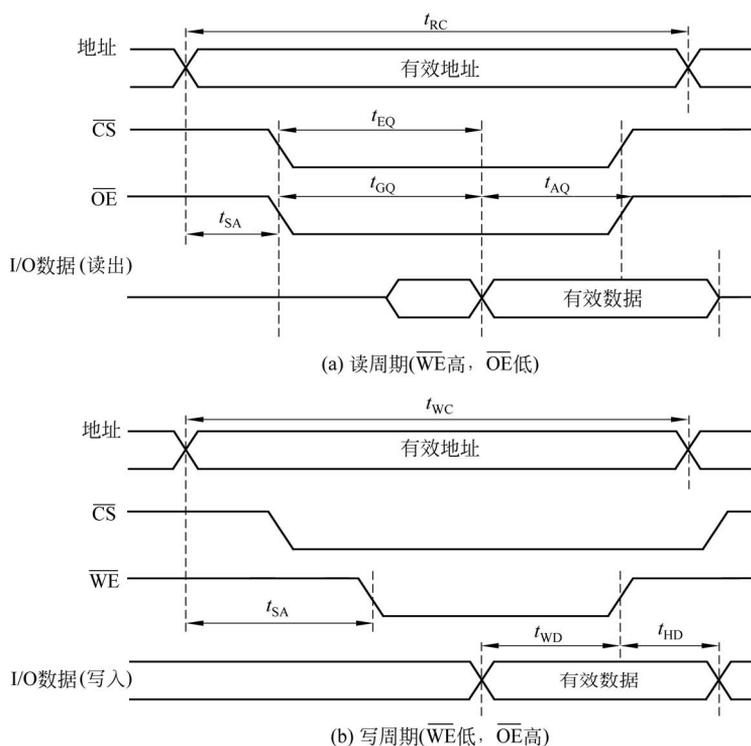


图 3.9 一维地址译码 RAM 的读写时序

(3) 发送读 (\overline{WE} 高、 \overline{OE} 低——有效) 和写 (\overline{WE} 低、 \overline{OE} 高——有效) 信号。

注意：数据线上的信号因读写而异。读出是从有关单元向数据线上输出数据信号，一般是在 \overline{CS} 信号和 \overline{OE} 信号稳定后即可有数据输出到数据线上；写入是由数据线向有关单元输入数据信号，一般对于送入数据的时间没有要求，但是 \overline{WE} 信号和地址信号不能撤销太早，以保证有一个写入的稳定时间。

3) SRAM 的有关时间参数

在对于一维地址译码 RAM 进行读写的过程中，为了保证读写质量，要关注如下一些时间参数。

(1) t_{SA} ：地址建立时间。从 CPU 发送地址信号到地址信号稳定时的时间，以保证地址正确。

(2) t_{EQ} ：片选有效时间。从 CPU 发送 \overline{CS} 到其稳定的有效时间，以保证选片正确。

(3) t_{GQ} ：读控制有效时间。从 CPU 发送 \overline{OE} 到其稳定的有效时间，以保证稳定的读操作。

(4) t_{AQ} ：有效读出时间，即 \overline{CS} 和 \overline{OE} 稳定到数据线上数据信号稳定的有效时间，在这段时间内 \overline{CS} 和 \overline{OE} 信号不可撤销，以保证读出信号正确。

(5) t_{WD} ：有效写入时间，以保证数据信号在 \overline{WE} 信号之前撤销，否则可能造成写入数据错误。

(6) t_{WC} ：写入周期。

(7) t_{RC} : 读出周期。通常 $t_{RC} = t_{WC}$, 通称存取周期。

(8) t_{HD} : 可靠写入数据信号需要的维持时间。

需要强调, 在不同的技术中, 读写的时序可能不同。关键是如何保证可靠地读写。

2. DRAM 原理

1) DRAM 元件

如图 3.10 所示, DRAM 的存储元件靠栅极电容上的电荷保存信息, 也称电荷存储型记忆元件。此外, 每个 RAM 还有行地址线和列地址线, 以支持二维地址译码。CPU 使用行选与列选信号, 使电容与外界传输电路导通, 电容充电(写入)与放电(读出)。

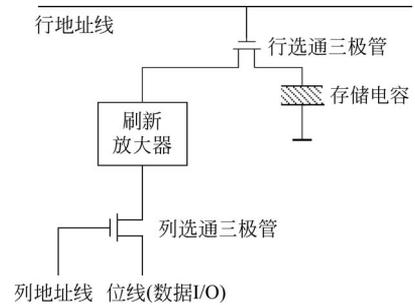


图 3.10 DRAM 的原理示意图

表 3.1 为 SRAM 与 DRAM 的比较。

表 3.1 SRAM 与 DRAM 的比较

比较内容	SRAM	DRAM
存储体中的记忆元件	触发器	电容
是否破坏性读出	否	是
是否需要再刷新	否	是
读写周期	短	长
集成度	低	高(约是 SRAM 的 4 倍)
每位成本	高	低(约是 SRAM 的 1/4)
所需功率	大	小(约是 SRAM 的 1/6)
主要应用场合	高速缓冲存储器	主存储器

2) DRAM 的逻辑结构与读写信号

图 3.11 为 $1M \times 4b$ 的 DRAM 芯片组成的存储阵列的逻辑结构图。

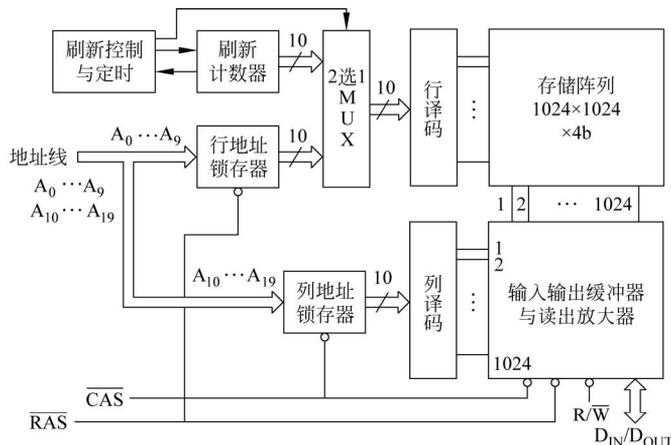


图 3.11 $1M \times 4b$ 的 DRAM 芯片组成的存储阵列的逻辑结构图

从这个逻辑结构图可以看出, 它的地址总数为 20, 分为两部分: 行地址 ($A_0 \cdots A_9$) 和

列地址 ($A_{10}\cdots A_{19}$)。此外, 还有一个刷新计数器及其控制线路。这样, 它在读写时涉及如下一些信号。

- (1) 行地址。
- (2) 行地址选通信号 $\overline{\text{RAS}}$ 。
- (3) 列地址。
- (4) 列地址选通信号 $\overline{\text{CAS}}$ 。
- (5) 数据总线 (D_{IN} 和 D_{OUT})。
- (6) 读写控制脉冲 $\text{R}/\overline{\text{W}}$ 。

由于 DRAM 中的信息要靠电容上的电荷保存, 而存储单元中的电容容量很小, 所以一个存储体中要有一个读出放大器 (sense amplifier, S-AMP), 用来放大/驱动数据信号, 以保证可识别性。

此外, 由于电容电荷总是会有泄漏的, 为了保持数据, DRAM 必须隔一段时间进行一次电荷的补充——刷新 (refresh)。刷新周期视泄漏速度而定, 如 2ms、4ms 等。为此 DRAM 中需要一个刷新放大器。不过这个刷新放大器现在已经被并入 S-AMP 中。

3) DRAM 读写操作时的时序

图 3.12 形象地描画了这类 DRAM 在读写过程中各种信号之间的时序关系。可以看出, 不管是读还是写, 都是按照下面的顺序发送有关信号。

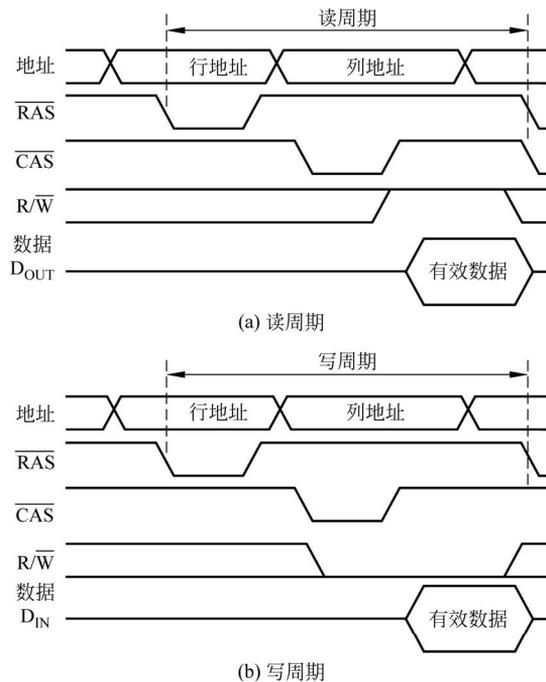


图 3.12 二维地址译码 DRAM 的读写时序

- (1) 发送行地址和列地址。
- (2) 行地址稳定后, 发送行地址选通信号 $\overline{\text{RAS}}$, 将行地址选存。
- (3) 发读写命令 $\text{R}/\overline{\text{W}}$ 。读时高电平, 并保持到 $\overline{\text{CAS}}$ 结束之后; 写时低电平, 在此期间,

数据线上必须送入欲写入的数据并保持到 $\overline{\text{CAS}}$ 变为低电平之后。

(4) 发送列地址选通信号 $\overline{\text{CAS}}$ ，将列地址选存。在此期间，若 $\text{R}/\overline{\text{W}} = 1$ ，则将数据输出到数据线上；当 $\text{R}/\overline{\text{W}}$ 、 $\overline{\text{RAS}}$ 和 $\overline{\text{CAS}}$ 都有效时，数据线上的数据被写入有关单元。

4) DRAM 的工作模式

DRAM 在发展过程中，不断改进，先后形成了一些不同的工作模式。

(1) PM RAM、FPM DRAM 与 EDO DRAM。PM RAM (page mode RAM, 页模式随机存储器)、FPM DRAM (fast page mode DRAM, 快速页模式随机存储器) 与 EDO DRAM (extended data output RAM, 扩展数据输出随机存取存储器) 都是异步 DRAM, 即 RAM 与 CPU 采用不同的时钟频率。另外, 它们还各有一些特点。

早期的 DRAM 称为 PM RAM。其特点是每写入一位数据, 就必须送出列(页)和行地址各一次, 以决定该位的位置; 并且每个地址都必须有一段稳定的时间, 才能读写有效数据。在地址稳定之前, 写入或读取的数据都是无效的。

FPM DRAM 是一种改良型 PM RAM。其特点是若需要读写的前后数据在同一列或同一页(page)内, 则内存控制器不必重复送出列地址, 只需指出下一行地址即可, 以此来提高读写效率, 可以做到每 3 个时钟周期输出一次数据。

EDO RAM 取消了主板与内存两个存储周期之间的时间间隔, 即不必等到数据完整地读取或写入, 只要一到有效的时间, 即可准备送出下一个地址, 实现了每隔两个时钟脉冲周期输出一次数据, 使存储速度提高 30%。

(2) SDRAM。SDRAM (synchronous DRAM, 同步动态随机存储器) 的基本原理是将 CPU 与 RAM 通过一个相同的时钟锁在一起, 使得 RAM 与 CPU 共享同一时钟, 以相同的速度同步工作, 避免了系统总线对异步 DRAM 进行操作时所需的一个时钟周期的额外等待时间, 每个时钟脉冲的上升沿便开始传递数据, 速度比 EDO 内存提高 50%。

(3) DDR SDRAM。DDR SDRAM (double data rate SDRAM, 双倍速率同步动态随机存储器, 简称 DDR) 是 SDRAM 的更新换代产品, 允许在时钟脉冲的上升沿和下降沿传输数据, 这样不需要提高时钟的频率就能加倍提高 SDRAM 的速度。现在 DDR 已经进一步升级为 DDR2、DDR3、DDR4、DDR5 等标准系列。

(4) RDRAM。RDRAM (rambus DRAM, 存储器总线式动态随机存储器) 是 RAMBUS 公司开发的新型 DRAM。它的主要特点是非常快, 速度是 DRAM 的千倍以上。这个特高速度源自 3 项技术: 采用超高频时钟频率 (800~1200MHz), 采用时钟双沿 (上升沿和下降沿) 传输数据 (每个 RDRAM 晶片的传输峰值可达 6.4GB/s), 采用串行模块结构 (各个芯片用一条总线串接起来, 前面芯片写满后, 后面的芯片才开始读入数据, 简化了产品设计)。

3.2 主存储体组织

3.2.1 内存条

主存储器相当于计算机中用于记忆的大脑, 要记忆大量信息就要有足够的存储单元。随着计算机应用领域的不断扩大, 计算机功能越来越强大, 主存储器的容量越来越大。为

了便于设计、安装、维修，主存储器都以内存条形式进行规格化生产制造。如图 3.13 所示，内存条一般由内存颗粒、SPD 芯片、引脚、印制电路板（printed-circuit board, PCB）（俗称金手指）及一些电容和电阻组成。

1. 内存颗粒

主存储器的核心是 IC 存储晶片。每个 IC 存储晶片被生成后，必须进行封装，使其与外界隔离，防止空气中的杂质对芯片电路的腐蚀而造成电学性能下降，也防止其他原因造成的损害。每个被封装的 IC 存储晶片就称为一个内存颗粒。

IC 存储晶片是非常精密、娇贵的产品。不仅制造需要高级技术，封装也需要高级技术。目前，业界普遍采用的封装技术有 TSOP（thin small outline package，薄型小尺寸封装）和 BGA（ball-grid-array，球栅阵列）封装技术。TSOP 的典型特征是引脚由四周引出，如 SDRAM 的 IC 为两侧有引脚，SGRAM 的 IC 四面都有引脚。BGA 封装的引脚是由芯片中心方向引出的，从而有效地缩短了信号的传导距离，可以减少信号的衰减。

在每个封装后的内存颗粒上，一般都印刷有生产厂家、产品编号，以及颗粒的容量、数据宽度、存取速度、工作电压等重要参数，如图 3.14（a）所示；也有的在内存条统一给出这些参数，如图 3.14（b）所示。各个厂家的参数标识各不相同。

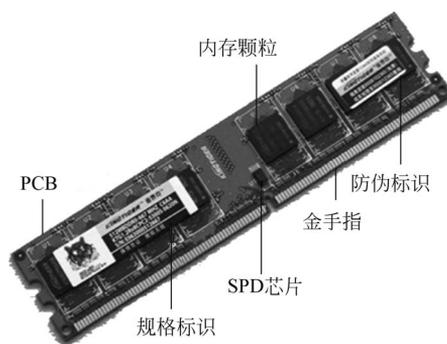


图 3.13 内存条结构



(a) 标识在各内存颗粒上



(b) 统一标识在内存条上

图 3.14 内存颗粒的参数标识

2. SPD 芯片

SPD（serial presence detect，系列参数预置检测）是一片小型的 EEPROM 芯片，里面记录了内存条出厂时预先存入的速度、工作频率、容量、工作电压、行 / 列地址带宽、传输延迟、SPD 版本等基本参数。当计算机开机工作时 BIOS 就会自动读取内存 SPD 中的记录信息，来对内存进行设置，使其运行在规定的工作频率上，工作在最佳状态，实现内存的“超频”。



图 3.15 内存条与插槽

3. 引脚

如图 3.15 所示，为便于安装与维修，内存条不是焊接在主板上，而是插装固定在主板上的内存插槽中使用。引

脚就是内存条中用来插入插槽进行内存条固定、并用于与计算机总线进行电气连接的部分。每个引脚都是 PCB 上镀金的覆铜条，所以俗称金手指。

内存条与其插槽都有一定的规格，要配套使用。引脚的基本参数是引脚数——线数。引脚数根据需要并行传递的数据位数而定。此外，为了防止插反，在引脚中间还设有槽口。有的规格设有一个槽口，有的设有两个槽口。下面介绍内存条的基本类型。

1) SIMM

SIMM(single in-line memory module, 单列直插式内存模块)是早期 FPM 和 EDD DRAM 内存条，这种内存条两面的引脚传输相同的信号，相当于只有一面有引脚。最初的 SIMM 一次只传输 8b 数据，后来逐渐发展到 16b、32b，形成 30 线、72 线和专用内存条 3 类。

2) DIMM

DIMM (dual in-line memory module, 双列直插式内存模块)的两面都有可独立传输信号的引脚，并且可单条使用。DIMM 又可以分为如下 4 类。

(1) SDRAM-DIMM 也称标准 DIMM，每面 84 线，双面共 168 线，故常称为 168 线内存条。工作时钟为 60MHz、67MHz、75MHz、83MHz。常见的容量有 8MB、16MB、32MB 等。主要用于 SDRAM 芯片，也称为 SDRAM 内存条。这种内存条有两个引脚槽口。

SDRAM-DIMM 的新产品：200 线的双面内存条，其工作时钟为 77MHz、83MHz、100MHz，数据宽度为 72b 或 80b，分缓冲型和非缓冲型两种，主要用于工作站和大型计算机。

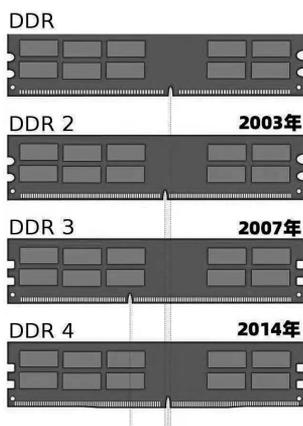


图 3.16 DDR、DDR2、DDR3、DDR4 内存条的结构

(2) DDR-DIMM。尽管起名 DDR-DIMM，但对于每代 DDR，其插槽还是不同。例如，DDR 内存条引脚数量为 184 针（每面 92 针）、DDR2 内存条为 240 针、DDR3 内存条有 240 针，而 DDR4 内存条则是 288 针，如图 3.16 所示，它们在结构上也有所区别。

(3) SO-DIMM (small outline dual in-line memory module, 小型双列直插式内存模块)是尺寸非常小的存储模块，同样 SO-DIMM 也根据 SDRAM 和 DDR 内存规格不同而不同：SDRAM 的 SO-DIMM 只有 144 针，而 DDR 的 SO-DIMM 拥有 200 针，DDR3 的 SO-DIMM 拥有 204 针；DDR4 的 SO-DIMM 拥有 260 针。

(4) 有 MicroDIMM 和 Mini Registered DIMM 两种接口。MicroDIMM 接口的 DDR 为 172 针，DDR2 为 214 针；Mini Registered DIMM 接口为 244 针，主要用于 DDR2 内存条。

3) RIMM

RIMM (rambus inline memory module) 是 Rambus 公司生产的 RDRAM 内存所采用的接口类型。RIMM 内存与 DIMM 的外形尺寸差不多，引脚同样也是双面的。RIMM 有 184 针的引脚，在引脚的中间部分有两个靠得很近的卡口。RIMM 非 ECC 版有 16 位宽，ECC 版则都是 18 位宽。

注意：内存条插槽也有与内存条对应的型号。

4. PCB、电阻和电容

内存条的 PCB 采用的是 4 层或 6 层的电路板。采用多层电路板的目的是：一是紧凑，减少电磁辐射；二是分层，屏蔽电路的电磁辐射。

内存条上的电阻有 10Ω 和 22Ω 两种，使用 10Ω 的电阻的内存条信号很强，对主板的兼容性较好，但其阻抗低，经常因信号过强导致系统死机。使用 22Ω 电阻的内存条，优缺点与前者正好相反。内存条厂家往往从成本考虑使用 10Ω 电阻。电容用于滤除高频干扰。

3.2.2 基本的存储体扩展方式

从逻辑结构上看，一个主存储体可以看成由一些芯片扩展成的存储体阵列。所采用的芯片的集成方式以及计算机要求的字长不同，扩展方式也不同。本节从逻辑角度介绍几种基本的存储体扩展方式。

1. 字扩展方式

字扩展方式是位数（字长）不变，字数扩展。图 3.17 为用 4 片 $16\text{K}\times 8\text{b}$ 的芯片扩展为 $64\text{K}\times 8\text{b}$ 的存储器。每个芯片内部都有自己的地址译码电路和数据缓冲电路。

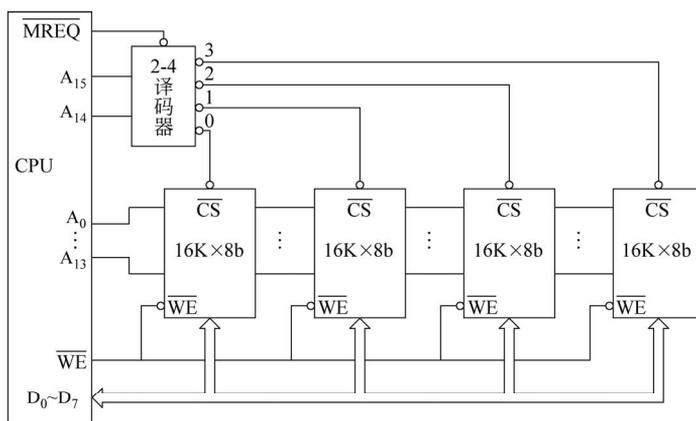


图 3.17 用 4 片 $16\text{K}\times 8\text{b}$ 的芯片扩展为 $64\text{K}\times 8\text{b}$ 的存储器

说明：

(1) CPU 地址总线的 A_{14} 、 A_{15} 连接到片选译码器， A_{14} 、 A_{15} 为不同值时，在片选译码器的 4 条输出中，只有一条被选中（输出低电位）。由于它们分别连接各芯片的 $\overline{\text{CS}}$ （片选端），因而只有一片被驱动（选中）。 $\overline{\text{CS}}$ 上的短线表示低电平驱动。

(2) 各片的 14 条地址线都连接到 CPU 总线的 $A_0\sim A_{13}$ 端。当 CPU 的地址总线输出一个 16 位的地址码时， A_{14} 、 A_{15} 选中某片，而 $A_0\sim A_{13}$ 选中该片中的某字。14 条地址线的寻址范围为 16K。

(3) $\overline{\text{WE}}$ 为读写控制。该端为高电平（ $\overline{\text{WE}} = 1$ ）时，被选中的字将被读出；该端为低电平（ $\overline{\text{WE}} = 0$ ）时，被选中的字将被写入。

(4) 每个芯片的 8 位数据引脚并联在 8 条 CPU 数据总线上。

2. 位扩展方式

位扩展方式是字数不变，位数（字长）扩展。图 3.18 为用 8 片 $8\text{K}\times 1\text{b}$ 的芯片扩展成 $8\text{K}\times 8\text{b}$ 的存储器。

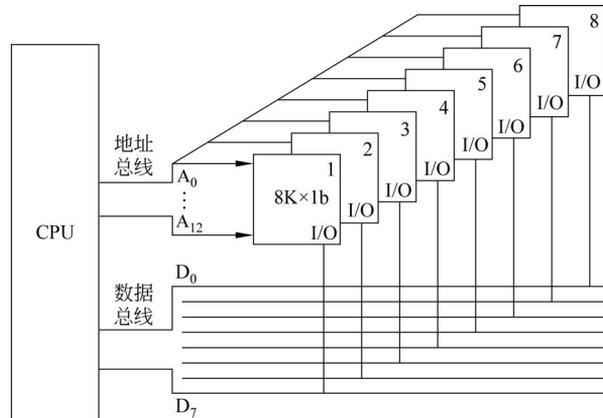


图 3.18 用 8 片 $8\text{K}\times 1\text{b}$ 的芯片扩展成 $8\text{K}\times 8\text{b}$ 的存储器

注意：

(1) 由于每个字的各位分布在所有芯片中，不同各片的片选端 $\overline{\text{CS}}$ 都并联在一起，连接到 CPU 的相应控制线上。当进行存储器读写时，该端应为低电平。同时，CPU 的 16 条地址线并联到各片的 16 条地址线上，以便同时选中各片中属于同一字的各位。

(2) CPU 中的每条数据线只与一个片中唯一的一条数据线相连。

3. 段扩展方式

段扩展方式是字向和位向都扩展。例如，由 4 片 $1\text{M}\times 4\text{b}$ 的芯片扩展成 $2\text{M}\times 8\text{b}$ 的存储器。

3.2.3 P-Bank 和 L-Bank

P-Bank 和 L-Bank 是从物理和逻辑两方面进行内存芯片设计和组织的两个术语。

1. P-Bank

通常，内存芯片与 CPU 交换数据时，为了保证 CPU 能正常工作，必须每发出一次片选信号，选中与 CPU 数据总线（以 SDRAM 系统为例，就是 CPU 到 DIMM 槽）的接口位宽相同的存储颗粒以便并行工作交换数据。这样的存储颗粒组就称为一个 P-Bank (physical bank, 也称 Rank, 因为 Intel 定义 P-Bank 为行)。这样的位宽称为物理层位宽。例如，一个 CPU 的数据总线位宽为 64b，就需要 64 个颗粒组成一个 P-Bank。从理论上说，为这样的 CPU 提供位宽为 64b 的存储芯片最为方便。但这对芯片制造有非常高的要求，会使芯片的成本效用都处于劣势。所以芯片会从自己的成本效用角度来确定自己的位宽。这样，一个 P-Bank 就需要由几个小位宽的存储芯片组合。例如，CPU 与内存之间的接口位宽是 64b，而芯片每个传输周期能提供的数据量是 8b，就需要 8 个内存颗粒并联。

内存条的带宽为其上各个内存芯片的带宽之和，基本条件是带宽之和等于 64b 或其倍

数。假如出现了各个内存芯片位宽之和等于 128b，则分成两个 64b，当读取一个 64b 部分时，另一个 64b 部分就不能读取。

2. L-Bank

从逻辑上看，一个内存就是如图 3.19 (a) 所示的一个由许多内存小方块 (cell) 组成的矩阵。显然，只要指定一个行 (row)，再指定一个列 (column)，就可以准确地定位到某个 cell。这个阵列称为内存芯片的 L-Bank (logical bank，逻辑 Bank，简称 Bank)。这里，cell 是根据由数据线条数决定的一次能同时读写的位数。

但是，由于技术、成本等原因，不可能只做一个全容量的 L-Bank。最重要的是，由于 SDRAM 的工作原理限制，单一的 L-Bank 将会造成非常严重的寻址冲突，大幅降低内存效率。所以，人们在 SDRAM 内部分割成多个 L-Bank，最多是 4 个；而 DDR-2 提高到了 8 个，RDRAM 则最多达到了 32 个。

由于主板芯片组本身设计时只允许一个时钟周期内对单个 L-Bank 进行操作，而不是主板芯片组对内存芯片内所有 L-Bank 同时操作。因此，如图 3.19 (b) 所示，在进行寻址时就要先确定是哪个 L-Bank，然后再在这个选定的 L-Bank 选择相应的行与列进行寻址。这样，单个 L-Bank 就决定了一次存储器寻址可以操作的位的数量。所以，有人将单个 L-Bank 内的 cell 数量称为数据深度。

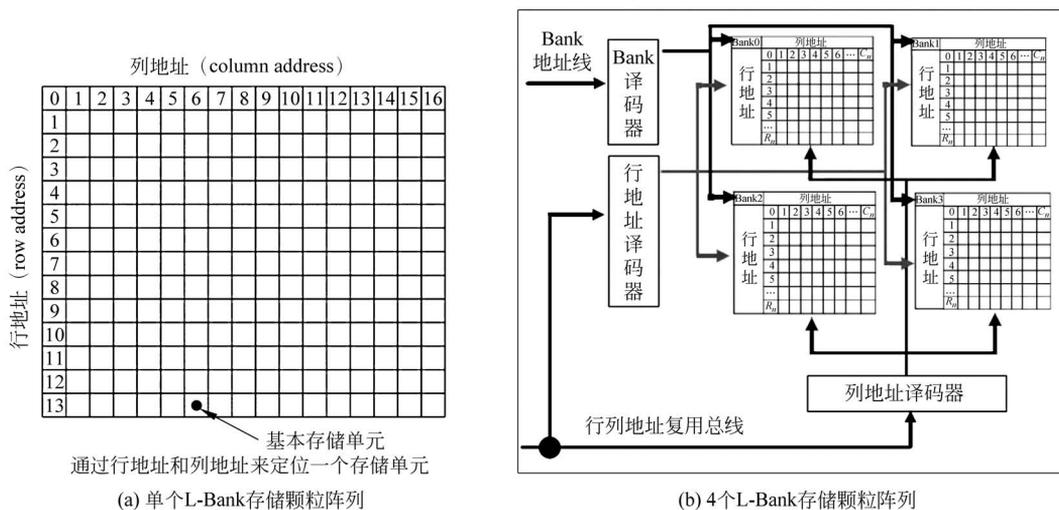


图 3.19 L-Bank 存储颗粒阵列示意图

3. 内存芯片容量的计算

内存芯片的容量可以用 L-Bank 计算，也可以用位宽计算。

(1) 用 L-Bank 计算。存储单元数量 = 一个 L-Bank 存储单元数量 × L-Bank 的数量。其中，一个 L-Bank 存储单元数量 = 行数 × 列数。

(2) 用位宽计算。在很多内存产品介绍文档中，都会用 $M \times W$ 的方式来表示芯片的容量。 M 是该芯片中存储单元的总数，单位是 M； W 代表每个存储单元的容量，即 SDRAM 芯片的位宽，单位是 b。计算出来的芯片容量也是以 Mb 为单位，但用户可以采用除以 8 的

方法换算为兆字节（MB）。

3.2.4 并行存储器

随着计算机所处理信息量的增加，不断对存储器的速度及容量提出更高的要求；而随着 CPU 功能的增强和 I/O 设备数量的增加，主存储器的存取速度越来越成为计算机系统中的一个瓶颈。为了提高访问存储器的带宽，在致力于寻找高速元件的同时，也加紧从存储体系结构方面对存储器组织结构加以改进，发展并行存储结构。

从结构上拓宽存储器带宽技术主要有双端口（或多端口）存储器、单体多字系统和多体存储系统。

1. 双端口存储器

传统的存储器只有一个读写端口，要么进行写，要么进行读，读写不能同时进行。双端口存储器的基本特点：每个芯片都有两组数据总线、地址总线和控制总线，形成两个访问端口，只要不是同时访问同一个存储单元，就允许两个端口并行地进行独立的读写，而不互相干扰。如果两个端口同时访问同一个存储单元，就由片内仲裁逻辑决定由哪个端口访问。这样，在多处理机系统中，可以让两个 CPU 同时访问主存储器；或者设计成一个端口面向 CPU、一个端口面向 I/O 处理的系统。两个端口可以不受编址的限制，成倍地提高存取效率。

双端口存储器的常见应用有 Cache-主存储器结构中的主存储器、运算器中的通用寄存器等。此外，在多机系统中常采用双端口存储器甚至多端口存储器，实现多 CPU 之间的存储共享。

2. 单体多字系统

根据程序访问的局部性原理（详见 3.7.1 节），要连续使用的信息（数据，尤其是指令）大多是就近（多数是连续）存放的。以此为前提，可以用同一套地址系统按同一地址码，在一个读周期内取出多个字，如同时读出 4 条指令，然后把它们组织成队列，每隔 $1/4$ 主存周期（ T_m ）依次将一条指令送入指令寄存器去执行。

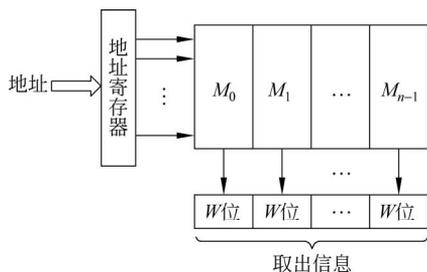


图 3.20 多模块单体存储器

典型的单体多字系统是如图 3.20 所示的多模块单体存储器。它是由字长为 W 位的 n 个容量相同的模块器 M_0, M_1, \dots, M_{n-1} 并行连接起来，构造字长为 $n \times W$ 的存储体。这样，每个存储周期中可以同时读出 n 个字（例如，同时读出 n 条指令），从而使存储器的带宽 $BM = W/T_m$ 提高到 n 倍。当然，若要访问的不是 n 个连续的字，如遇到转移指令或随机分布的数据时，会大大降低实际的带宽。

3. 多体存储系统

与单体存储系统不同，多体存储系统的各存储体都有自己的一套地址寄存器和地址译码、驱动、读数、时序电路，各自能以同等方式与 CPU 通信，形成分别独立的编址，又能

并行或交叉工作，具有 N 个容量相同的存储体。

1) 多体存储器的访问方式

分为 N 个存储体的主存储器称为“模 N ”的存储器，这 N 个存储体按统一规则，分别编址。它们的工作方式有两种。

(1) N 个存储体同时启动，完全地并行工作，即同时送进 N 个地址，同时读出 N 个字，在总线上分时传送。

(2) N 个存储体分时启动流水式工作。通常互相错开 $1/m$ 个存储周期 (m 为存储体的个数)，交叉地工作。图 3.21 为有 4 个存储体的交叉存储器中各存储体启动的时序关系 (负脉冲启动)。可以看出，采用 4 个存储体的交叉存储器，每隔 $1/4$ 主存周期 (T_m) 便启动一个存储体工作。这样，在一个存储周期内，就访问了 4 个存储单元，将存储带宽提高到了 4 倍。

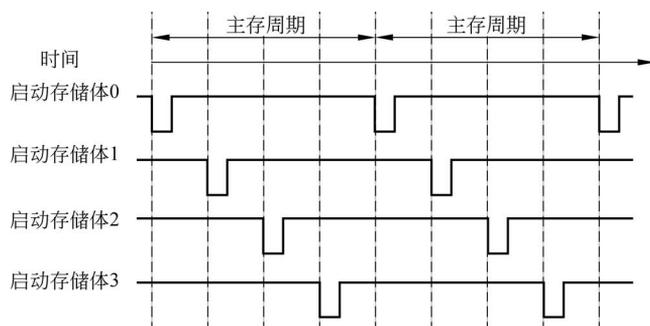


图 3.21 4 个存储体的交叉存储器中各存储体启动的时序关系

2) 多体存储器编址

访问多体存储器时，需要分别指定所访问的体号和体内地址，所以每个地址码由体号 + 体内地址 (字地址) 两部分组成，并需要分别进行地址译码。按照存储地址在各存储体内的分布关系，多体存储器的编址分为顺序 (高位交叉) 编址和交叉 (低位交叉) 编址两种。

如图 3.22 所示，顺序编址用高位地址表示体号，低位地址表示体内地址，所以每个存储体内的地址都是连续的。这样有利于存储器的扩充，可以任意增加一个或多个存储体 (低位交叉方式只允许按 2 的倍数增加)，并且当一个存储体出现故障时，不影响其他存储体的

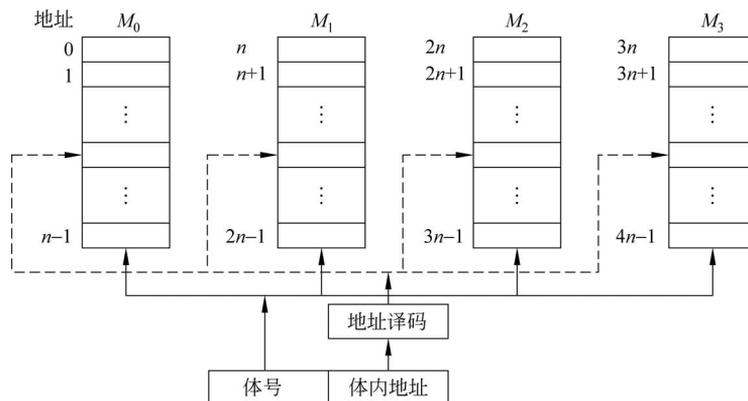


图 3.22 顺序编址的 4 个存储体的存储器的基本结构

工作，可靠性较高。但是，由于一个存储体在一个访存周期内只能读写一个字单元，而根据程序的局部性原理，多数程序代码是连续存放的，并且所访问的数据也是局部连续的。读写时，先选中一个存储体，再选其体内地址进行读写。这种编址方式一般不能实现多个存储体的并行工作，也不能实现多个存储体的分时启动（除非存储器采用的工作频率为 CPU 工作频率的 m 倍， m 为存储体的个数），只有在直接存储器访问（DMA）及通道控制方式等情形下才有可能实现并行或并发工作。

如图 3.23 所示，交叉编址用低位地址表示体号，用高位地址表示体内地址，所以存储地址是横向连续的，即连续地址分布在相邻的块内。读写时，先选中 4 个相同的体内地址，然后依次给出体号来选中具体单元，进行读写。这样，非常容易实现各存储体的分时流水以及并行工作，大大提高了存储体的带宽。这是目前多体存储器的主流。

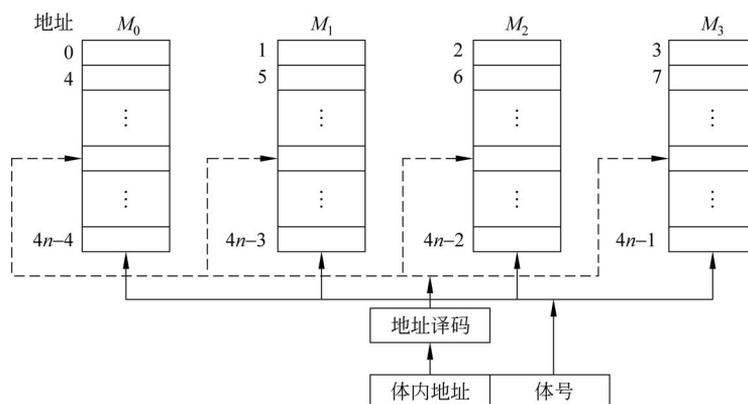


图 3.23 交叉编址的 4 个存储体的存储器的基本结构

由于指令、数据、向量等的存储和执行基本上是顺序的，所以采用交叉编址时，同一主存周期中取出的是要连续执行的指令或数据，因而有利于减少存储冲突。但是可靠性较低，一个存储体故障将会导致所有程序无法执行。采用顺序编址方式时，一个存储体内的地址是连续的，但多个存储体仅扩大了存储容量，对提高吞吐量并没有作用。

3.2.5 并行处理机的主存储器

并行处理机一般是指一台计算机中有一个指令部件（取指令）和多个执行部件。在这种计算机中，处理机与存储器间通过互连网络（interconnection network, ICN）交换信息。如图 3.24 所示，其存储结构大体可以分为两大类：共享存储器结构和分布存储器结构。

1. 共享存储器结构

在具有共享存储器的并行处理机中，只有一个统一的内存，经 ICN 为全部处理元素（processing element, PE）共享。I/O 设备、外存都可以通过 I/O 通道与共享存储器交换信息。

2. 分布存储器结构

在具有分布存储器的并行处理机中，存储系统由两部分组成：一部分集中在专做管理的监控机（supervisory control, SC）中，为常驻操作系统使用；另一部分称为处理元素存储器或局部存储器（processing element memory, PEM），分布在各处理单元中，用于存放程

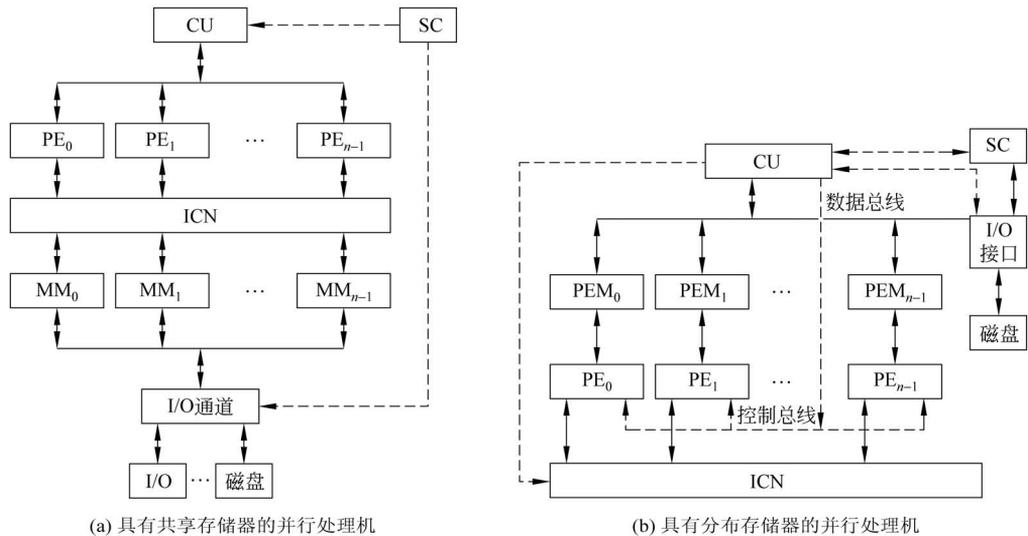


图 3.24 并行处理机系统中处理机与存储器的互连网络

序和数据。高速磁盘是存储的后援，通过 I/O 接口与 SC 及 PEM 交换信息。为了有效地进行高速处理，要使每个 PE 都可以依靠自己的 PEM 中的数据进行运算，为此要合理分配各处理单元中的数据。各处理单元之间可以通过两条途径相互联系：一条通过 ICN；另一条通过控制部件（control unit, CU），即数据从 PEM 读至 CU，然后通过公共数据总线“播送”到全部 PE 中。

3.2.6 存储器控制器

DRAM 的刷新需要有硬件电路的支持，包括刷新计数器、刷新/访存裁决、刷新控制逻辑等。这些控制线路形成 DRAM 控制器，它将 CPU 的信号变换成适合 DRAM 芯片的信号。

- (1) 地址多路开关：刷新时不需要提供刷新地址，由多路开关进行选择。
- (2) 定时发生器：提供行地址选通信号 RAS、列地址选通信号 CAS 和写信号 WE。
- (3) 刷新定时器：定时电路用来提供刷新请求。
- (4) 刷新地址计数器：只用 RAS 信号的刷新操作，需要提供刷新地址计数器。
- (5) 仲裁电路：对同时产生的来自 CPU 的访问存储器的请求和来自刷新定时器的刷新请求的优先权进行裁定。

有些芯片将刷新控制电路集成在芯片内部，具有自动刷新功能；有些芯片需要外加刷新控制电路。

3.3 SDRAM 内部操作与性能参数

SDRAM 在集成度和成本两方面的优势，使其得到广泛应用和快速发展，并在发展中形成了一些极有特色的技术。这些技术特色主要表现在 SDRAM 的内部操作上。

3.3.1 SDRAM 的主要引脚

为了说明 SDRAM 工作时其内部对技术性能有重要影响的内部操作，需要了解内部的

重要信号，这些信号主要由引脚提供。表 3.2 为关于这些信号的说明。

表 3.2 SDRAM 的重要引脚

引 脚	名 称	描 述
CLK	时钟	芯片时钟输入
CKE	时钟使能	片内时钟信号控制
$\overline{\text{CS}}$	片选	禁止或使能 CLK、CKE 和 DQM 外的所有输入信号
BA ₀ , BA ₁	组地址选择	用于片内 4 个组的选择
A ₁₂ ~A ₀	地址总线	行地址: A ₁₂ ~A ₀ , 列地址: A ₈ ~A ₀ , 自动预充电标志: A ₁₀
$\overline{\text{RAS}}$	行地址锁存	行、列地址锁存和写使能信号引脚
$\overline{\text{CAS}}$	列地址锁存	
$\overline{\text{WE}}$	写使能	
LDQM, UDQM	数据 I/O 屏蔽	在读模式下控制输出缓冲; 在写模式下屏蔽输入数据
DQ ₁₅ ~DQ ₀	数据总线	数据输入输出引脚
V _{DD} /V _{SS}	电源/地	内部电路及输入缓冲电源/地
V _{DDQ} /V _{SSQ}	电源/地	输出缓冲电源/地
NC	未连接	未连接

3.3.2 SDRAM 的读写时序

SDRAM 的读写操作是从对一个 L-Bank 中的阵列发出激活 (active) 命令开始的, 其过程如图 3.25 所示: 先是行有效操作, 再是列读写。两者之间的时间间隔称为 t_{RCD} (RAS to CAS Delay, RAS 信号到 CAS 信号之间的延迟)。最后才是读写操作。

1. 行有效

1) 行有效及其过程

行有效就是确定要读写的行, 使之处于激活——有效状态。一般行有效之前要进行片选和 L-Bank 定址, 但它们与行有效可以同时进行。行有效的时序如图 3.26 所示。操作过程如下。

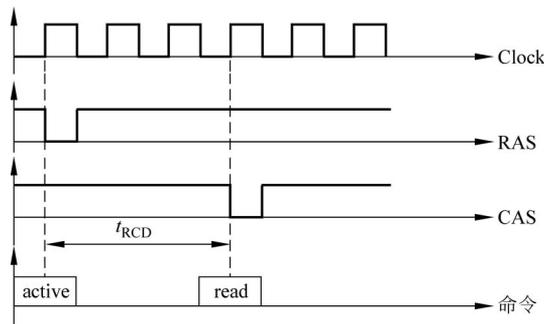


图 3.25 RAS 信号到 CAS 信号之间的延迟

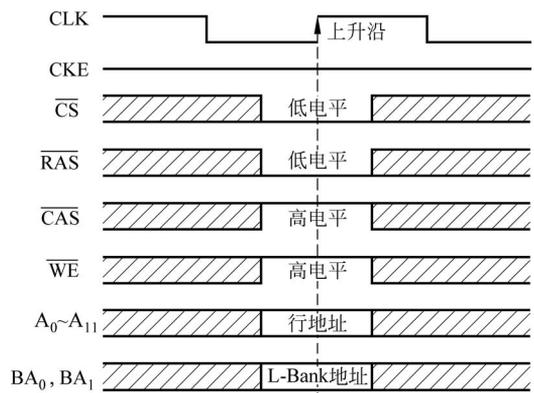


图 3.26 行有效的时序

(1) 行地址通过地址总线传输到地址引脚。

- (2) $\overline{\text{RAS}}$ 引脚被激活，行地址被放入行地址选通电路（row address latch）。
- (3) 行地址解码器（row address decoder）选择正确的行然后送到传感放大器（S-AMP）。
- (4) 此时 $\overline{\text{WE}}$ 引脚未被激活，所以 DRAM 知道它们不是进行写操作。

2) 行有效过程中的时间参数

在行有效过程中，主要涉及 3 个时间关系： t_{RAS} 、 t_{RC} 和 t_{RP} 。它们之间的时序关系如图 3.27 所示。

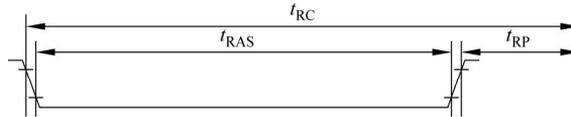


图 3.27 行有效过程中的 3 个时间关系

- (1) t_{RAS} : 行地址选通（row address strobe, RAS）信号的有效时间。
- (2) t_{RC} : 行周期（row cycle）时间。在一个 L-Bank 中，两个相邻的 active 命令之间的时间间隔。而在同一 Rank 不同 L-Bank 中，执行两个连续 active 命令之间的最短的时间间隔被定义为行地址间延迟（RAS to RAS Delay） t_{RRD} 。
- (3) t_{RP} : 行地址选通预充电（RAS precharge）时间。如前所述，它用来设定在另一行能被激活之前 RAS 需要的充电时间（详见 3.3.6 节）。

2. 列读写

行地址确定之后，就要对列地址进行寻址。在 SDRAM 中，行地址与列地址线在 $A_0 \sim A_{11}$ 中一起发出。列地址选通（column address strobe, CAS）信号则可以区分行与列寻址的不同。

在发出列寻址信号的同时发出读写命令，并用 $\overline{\text{WE}}$ 信号的状态区分是读还是写：低电平（有效）时是写命令，高电平（无效）时是读命令。图 3.28 为列读写的时序。

在发送列读写命令时必须要与行有效命令有一个间隔，这个间隔被定义为 t_{RCD} 。广义的 t_{RCD} 以时钟周期为单位，如 $t_{\text{RCD}}=2$ ，代表延迟周期为两个时钟周期。

在选定列地址后，就确定了具体的存储单元，剩下的事情就是将数据通过 I/O 通道输出到内存总线上了。但是并非 CAS 信号一发出就会有数据输出，往往要等几个时钟周期才可以进行读取动作。这段时间被定义为列地址选通潜伏期（CAS latency, CL）。由于 CL 只在读取时出现，所以 CL 又被称为读取潜伏期（read latency, RL）。

需要注意，潜伏（latency）与延迟（delay）是两个不同的概念。延迟指一个信息或一个事件被推迟的时间量，而潜伏指已经发生但还没有到达一定水平。因此说，CL 造成了一

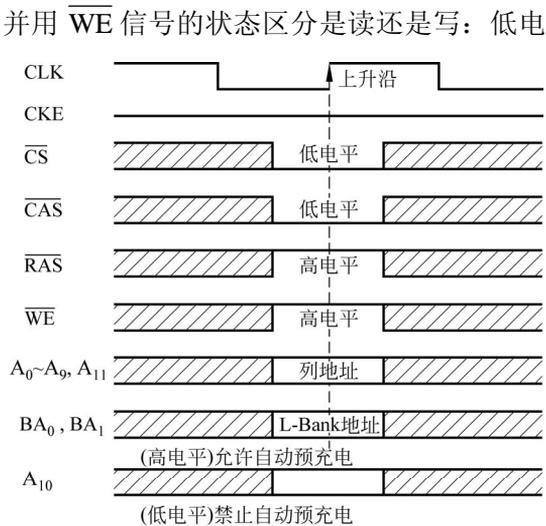


图 3.28 列读写的时序

些输出延迟，但 CL 并不是列地址选通延迟（CAS Delay, CD）。

一般来说，CL 值越小越好。不过还要看内存颗粒的原始设定值，不能让 CL 值超过芯片的设计规范（电子工程设计发展联合会（Joint Electron Device Engineering Council, JEDEC）定义的《JESD79》），否则会导致内存不稳定，甚至开不了机。此外，还要注意 CL 值不能在数据读取前临时修改，只能在初始化过程中的模式寄存器设置（mode register set, MRS）阶段设置，可用的选项为 1、2、4、8、全页（full page），通常选 4 或 8。

3. SDRAM 读周期中的时序细节

图 3.29 为 SDRAM 读周期中的时序细节。

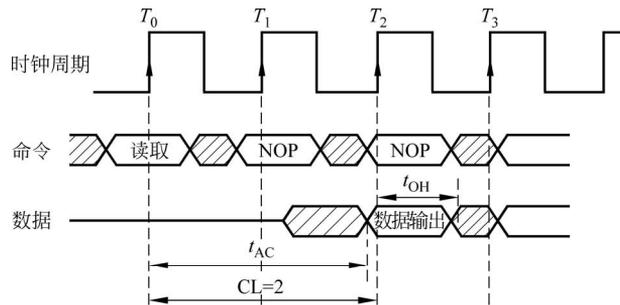


图 3.29 SDRAM 读周期中的时序细节

图中， t_{OH} 为数据逻辑电平保持周期。时钟触发后的访问时间 t_{AC} 则由如下原因引起：S-AMP 的放大驱动要有一个准备时间才能保证信号的发送强度（事前还要进行电压比较以进行逻辑电平的判断）。简单地说，从数据 I/O 总线上有数据输出之前的一个时钟上升沿开始，数据就已传向 S-AMP，经过一定的驱动时间最终传向数据 I/O 总线进行输出，这段时间被定义为 t_{AC} 。 t_{AC} 的单位是纳秒（ns），必须小于一个时钟周期。

需要注意的是，每个数据在读取时都有 t_{AC} ，包括在连续读取中，只是在进行第一个数据传输的同时就开始了第二个数据的 t_{AC} 。

3.3.3 突发传输

突发传输（burst transmission）是指在短时间内进行相对高带宽的数据传输。为此在对一组连续内存单元进行顺序读写时，要采取的与早期的 FPE/EDO 不同的技术策略。就此将 FPE/EDO 所采取的技术称为非突发传输技术。

非突发传输技术的基本思想：当要对一组连续内存单元顺序读写操作时，考虑到行地址相同，一开始只送一次行地址，以后每访问一个单元，就只送一个列地址和读写命令。这样就可以把送行地址的时间节省下来。以读操作为例，CAS 发出之后，经过一个 CL 就会在数据总线上有数据流出。由于这几个单元的操作类似，CL 相同，所以在数据总线上的数据是连续的。图 3.30（a）为 $CL = 2$ （个时钟周期）时的非突发传输时序图。这种非突发传输技术，节省了发送行地址的时间，但是它在数据进行连续传输时无法输入新的命令，效率很低（内存就是以这种方式进行连续的数据传输）。为此，人们开发了突发传输技术。

如图 3.30（b）所示，采用突发传输技术，只要指定起始列地址与突发长度（burst length, BL，即连续传输所涉及存储单元的数量），内存就会依次地自动对后面相应数量的存储单元