

DRAM 结构 自底向上

总体结构

目前常见的DRAM模块是双列直插式内存模块(DIMM)，存储器控制器(MC)通过通道与DIMM通信，每个通道由命令总线、地址总线和数据总线组成。一个DIMM有一个或多个rank(一般看内存芯片有几面)。一个rank有一组内存芯片，每个芯片有若干个bank。在数据总线为64位宽的典型情况下，一个rank有8个芯片，一个芯片上有8个bank来响应数据总线的请求。一个bank由多个子阵列和一个全局行缓冲区组成。每个子阵列是一个具有局部行缓冲区的二维单单元格数组，存储最近访问的特定行单元格的数据。一行中的单元格通过 Wordline 水平连接。列中的单元格通过 Bitline 垂直连接到本地行缓冲区。本地行缓冲器通过全局位线与全局行缓冲器相连。一个单元由一个作为开关的存取晶体管和存储单个比特的电容器组成。

每个bank中的数据按行和列组织。在下图中，每个bank有256K行，每行有1K列。尽管每行有1K列，但DDR4协议规定列必须以8或4为对齐单位进行访问，其中8是最常见的配置。在读或写时，对齐单位中的8列以8个连续的数据突发进行读/写。因此，每行中只有 $1K/8 = 128$ 个可寻址的对齐数据单位。

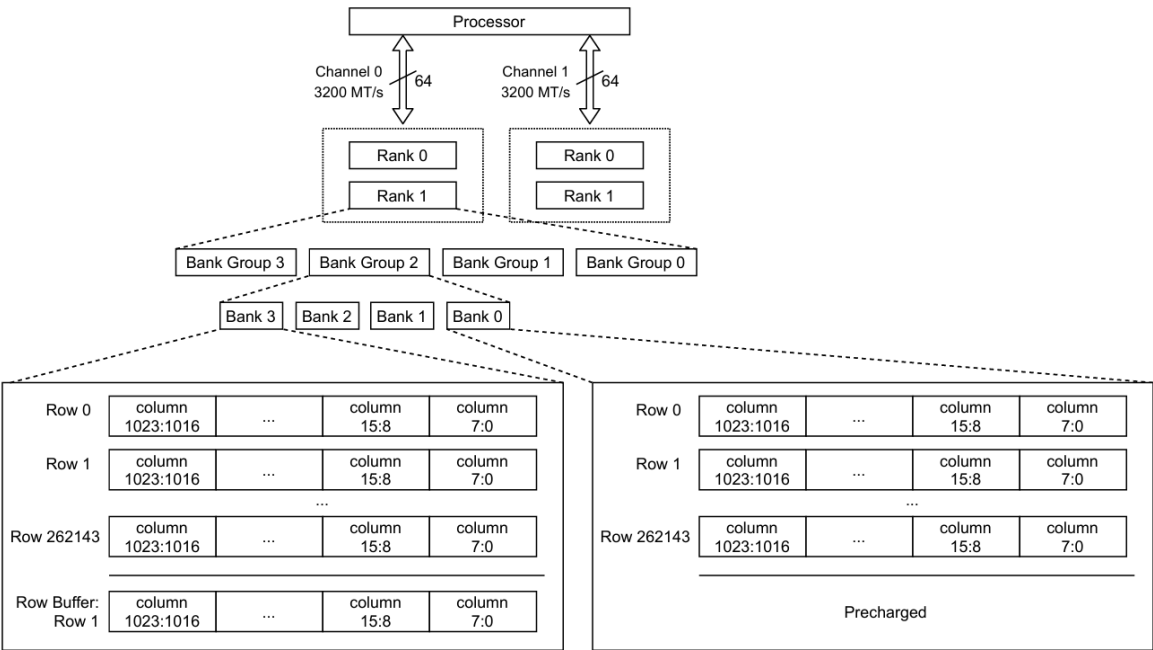
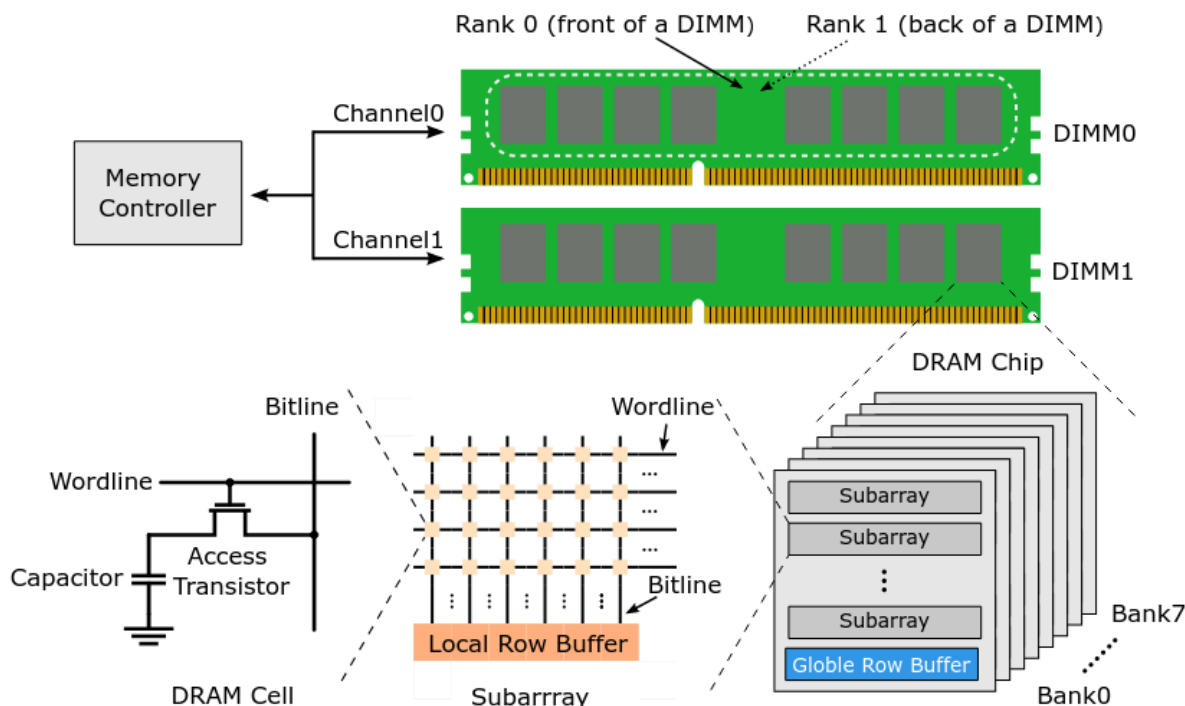
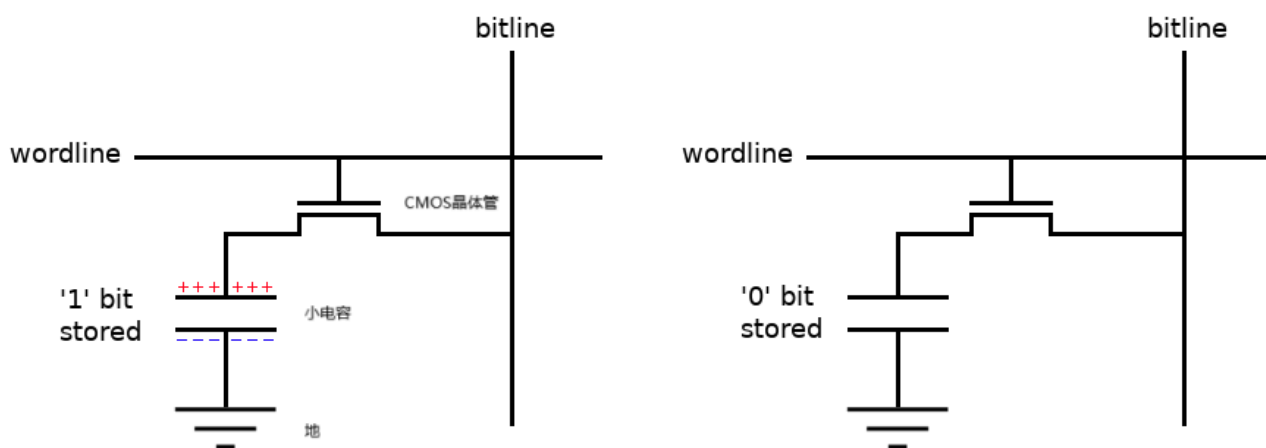


Figure 2.1: DRAM hierarchical organization.



cell单元

DRAM的基础存储单元是 `cell`，`cell` 中的小电容是存储信息的关键，小电容可以存储电荷，现在规定当电容存有电荷，`cell` 存储比特信息“1”；当电容不存有电荷，存储比特信息“0”。其基本结构如下

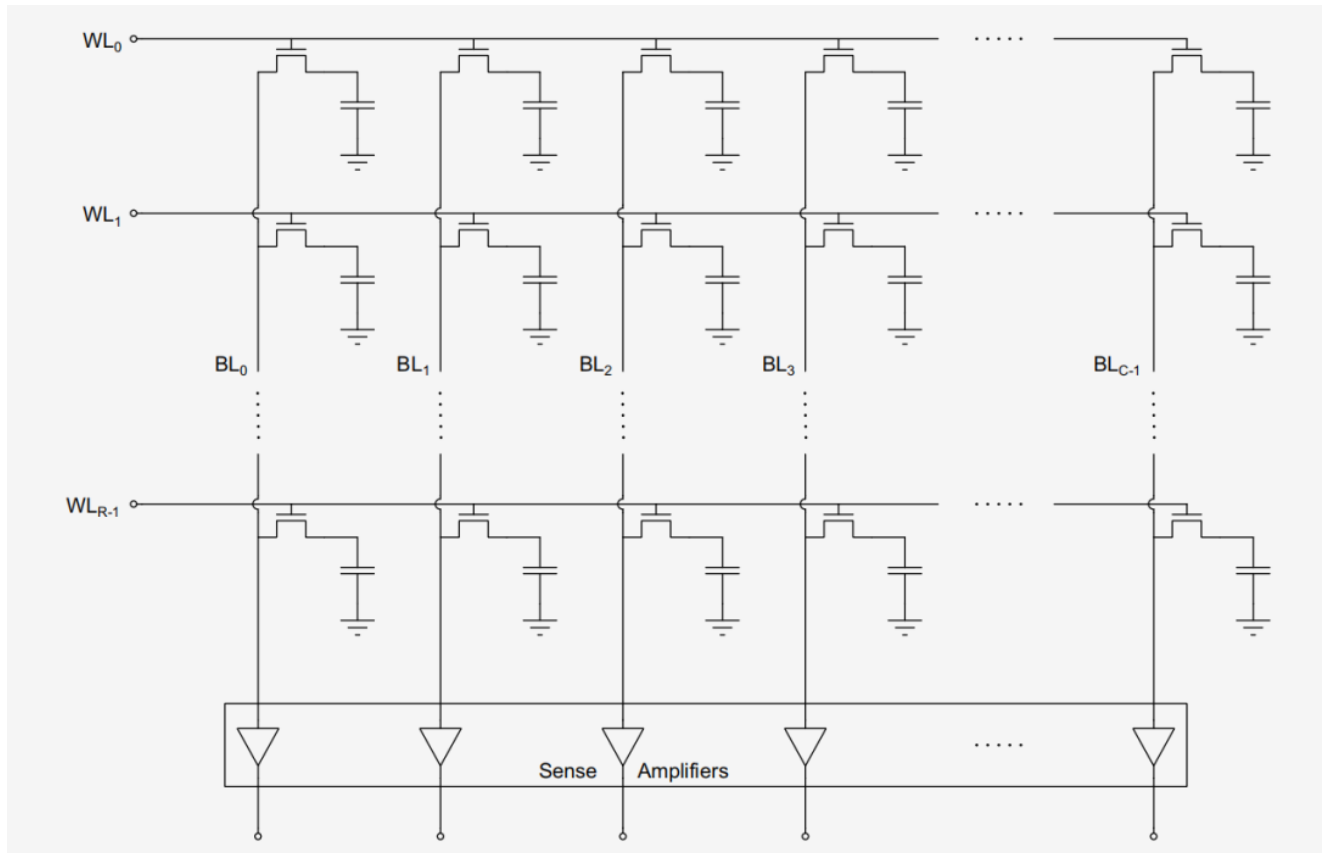


当要读取 `cell` 的存储值，首先打开电子开关（即晶体管），然后根据导通后的电容是否会进行充放电信息获得存储值。如果 `cell` 保存“1”，即电容存有电荷，那么当打开开关，电容就会放电；如果 `cell` 保存“0”，即电容不保存电荷，那么打开开关之后电容不会放电。

当要向 `cell` 中写入值，仍然先打开电子开关，然后在电子开关的另一侧施加电压。如果要写入“1”，则施加高电压，此时电流会通过晶体管向电容充电；如果要写“0”，则让电子开关另一端接地。施加电压一段时间后即可断开开关，此时 `cell` 已经保存好写入值，因为电容很小，所以施加电压的时间会很短。

cell阵列(subarray)

一个 cell 只能存储一比特信息，即“0”和“1”，为了存储大量信息，需要构建起 cell 阵列，这些存储单元按照行和列的方式排列。就像一个巨大的表格，每个单元格都是一个存储cell。每行 cell 的晶体管的栅极都是连在一起的，即都连在字线上，这意味着给字线施加电压，字线对应的一行cell都会被打开。当一行 cell 被打开，**cell 电容就会向位线充放电**，一行中的每个 cell 都与一条位线直接相连，读取位线的电压变化，即可知道 cell 的存储信息。



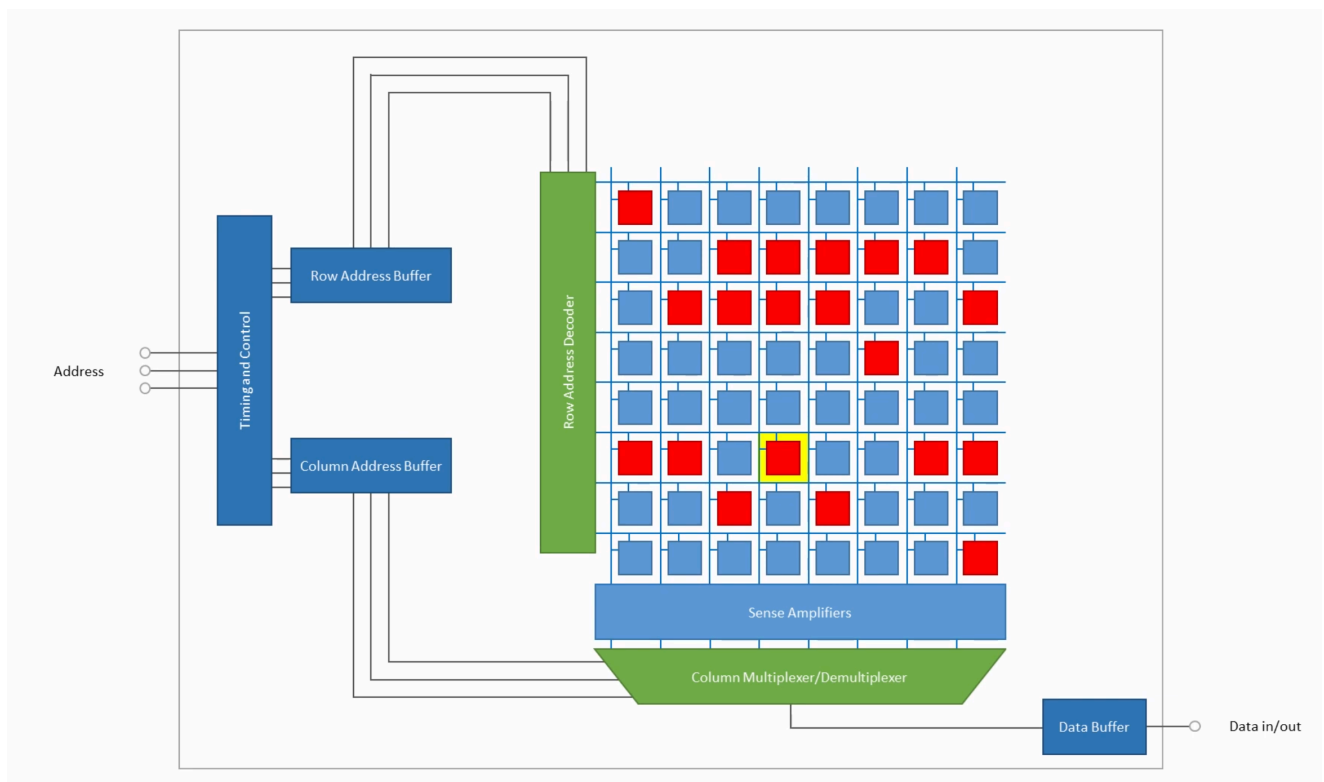
这里有两种线，用于读取存储的值：

- word line：字线（用来控制读取哪一个字，一个字是有若干个字节组成，由若干*8个位组成）。之所以叫字线，是因为给这根线通电，**一行 cell 都会被打开**，在计算机里八位等于一个字节，多个字节等于一个字，因此多个 cell 组合起来就是多个字，因为这根线可以打开多个字，所以叫字线
- bit line：位线。之所以叫位线，是因为在读取信息时，每一根线上的电压波动都代表一位比特信息，一根线代表一位，所以叫做位线

cell 的读取依靠小电容充放电，电容充放电导致位线产生电压波动，通过读取位线电压波动即可获取信息。小电容充放电所产生的电压波动是很微弱的，充放电所造成的电压波动的时间也是很短的，因此很难直接读取充放电信息，为此 cell 阵列的读取使用到了“**sense amplifier**”，即**读出放大器**。

读出放大器可以捕捉到微弱的电压波动，并根据电压波动的情况在本地还原出 cell 的电容电压，而且放大器内还有锁存器，可以把还原出来的电容电压值保存起来，这样一来 cell 保存的信息就从 cell 电容转移到了放大器。

在读取时地址会被解码成 Row Address 和 Column Address , 在经过地址解码器后 指定的 Row 中的数据会被全部读取出来并存在 **感应放大器**中, 随后再根据 Column 选出一个 bit 的数据输出.



Cell中的感应放大器有着如下的功能：

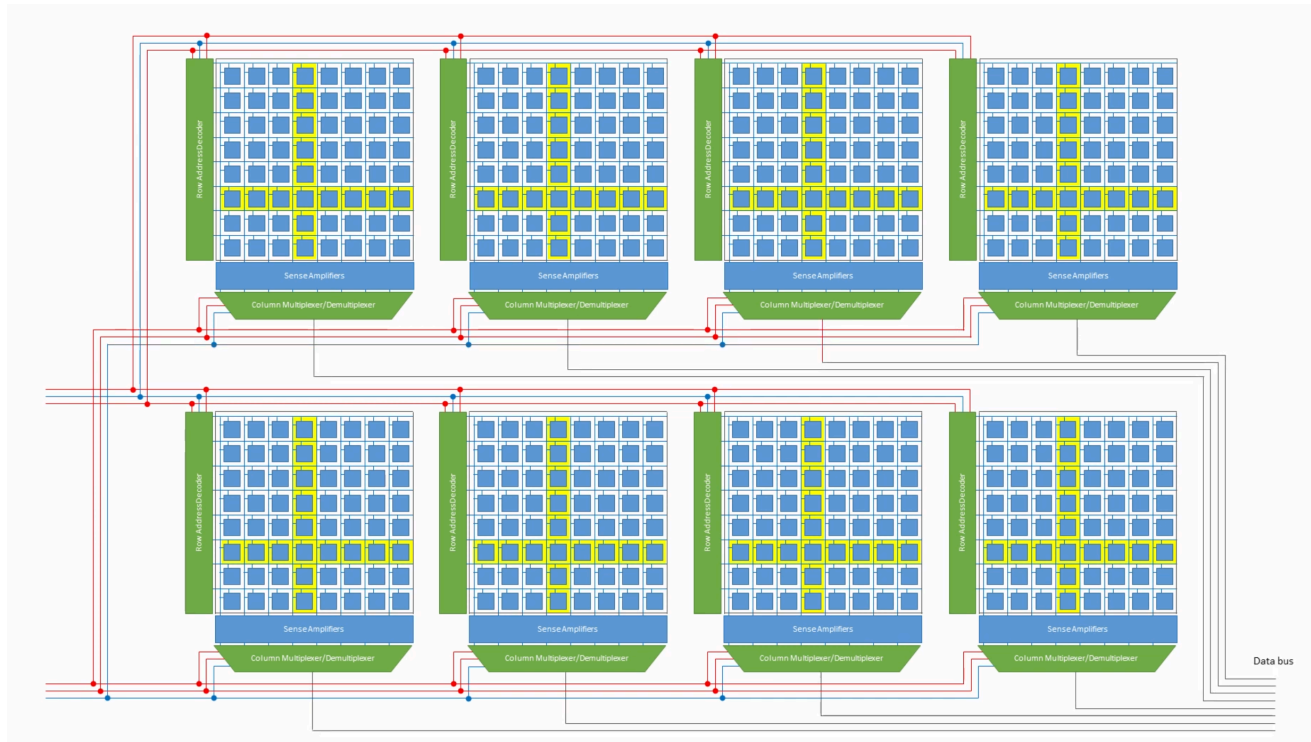
1. 感测: **放大器将该位线上的电压与另一根单独的位线上提供的参考电压进行比较, 并将电压差放大到极限, 使得存储值可以被解析为数字1或0。**
2. 临时存储: 在存储单元数据被感测和放大后, 感测放大器会持续驱动这些数据, 直到 DRAM 阵列预充电并准备好下一次存取。这样, 可以从感测放大器直接访问整行数据, 而无需重复行激活。感测放大器阵列在此过程中充当行缓冲器, 缓存并放大数据。因此, **感测放大器阵列也被称为行缓冲器(RowBuffer)**, 并需要管理策略来控制其行为。不同的管理策略决定了行缓冲器是保留数据直到下次刷新, 还是在数据恢复到存储单元后立即放电。
3. 恢复(刷新): 在位线电压被感测和放大后(**读取**), 感测放大器需恢复存储单元的原始值。读取单元行时, 存1的电容放电, 存0的电容充电, 导致信息丢失, 称为“**破坏性读出**”。感测放大器通过暂存单元行每个cell的电容电压, 在读取后将这些电压写回原单元行。根据锁存的读取值, 位线连接到供电电压或地, 使cell电容充电或放电, 结束后断开字线, 完成DRAM读取。

一般在 DRAM 芯片中, **读出放大器会把 cell 阵列分为两半**.

Bank

Bank结构

一个 cell 阵列一次可以提供一位，那么多个 cell 阵列就可以一次提供多位。假如 CPU 一次读写 8 个比特，那么我们就可以用 8 个 cell 阵列。8 个 cell 阵列可以共享一组行地址和列地址。**共享行、列地址的一组 cell 阵列被称作一个 bank**，下图展示了一个含有 8 个 cell 阵列的 bank。它们共用行地址、列地址和地址选通、写使能，**每个阵列提供一条输出线，8 个阵列最终组成 8 根输出线，可以输出 8 个比特。**



每一个 subarray 都有一个 Local Row Buffer，其中存储了该 subarray 在上次访问的整个 row 的内容，在下次访问相同 row 时就可以直接从其中读取相应的值而不用再去读取。

一个 bank 中所有的 subarray (可能有 128 个，来自[Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques]) 连接到一个 Global Row Buffer。

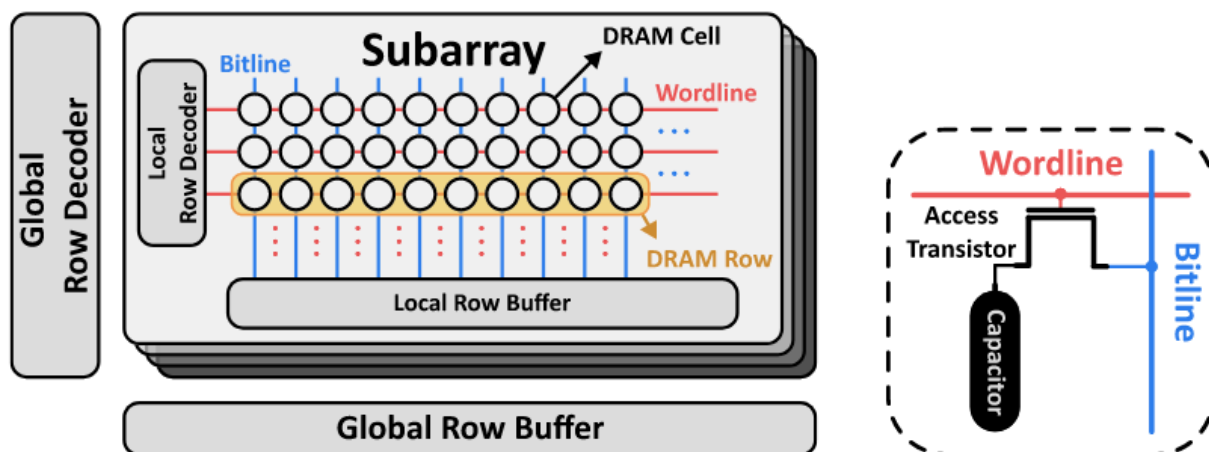


Figure 2: DRAM bank and cell.

与Row Buffer有关的时间

来自[内存系统: DRAM, DDR 与Memory Controller之二 - 知乎](#)

一条访存指令发到内存控制器，它的访存延时是存在不同的可能性的。

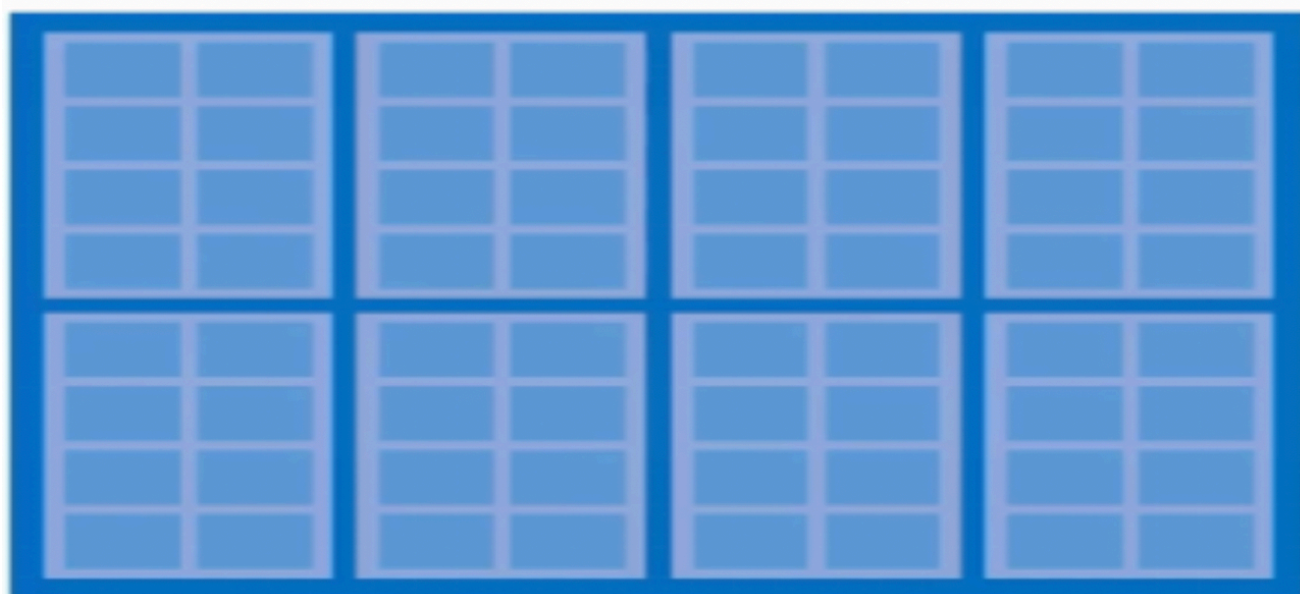
1. row buffer hit 就是说数据已经在row buffer中，这时延时主要来自于从row buffer到把数据放在数据总线上的时延，这个过程需要大约20ns的时间。
2. empty row buffer，即row buffer是空的，访存延时除了从row buffer到数据总线时间，还包括从电容到sense amplifier再到row buffer的时序，需要的延时大约40ns。
3. row buffer conflict，即当前row buffer存有别的row的内容，此时需要回写过程，即把row buffer中的比特刷回电容。否则电容上的内容可能会与row buffer上的内容不同。再加上2的延时，总共约60ns。

Bank Burst Mode

TODO

存储芯片/chip

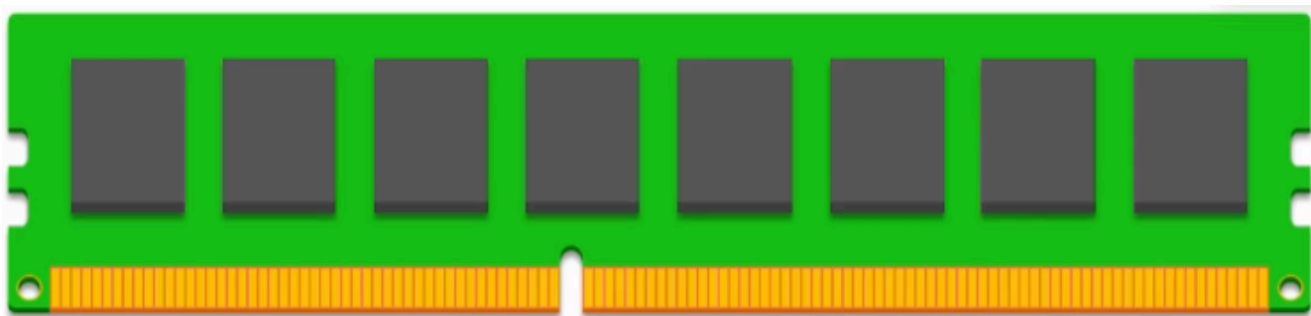
一个 8 阵列的bank 一次读写 8 个比特，一颗存储芯片上一般含有多个 bank，下图是一颗含有 8 个 bank 的存储芯片的示意图。芯片每次读写都只针对一个 bank，因此读写地址必须包含一个 bank 号，**bank 号用于开启目标 bank，目标 bank 之外的 bank 是不工作的。**



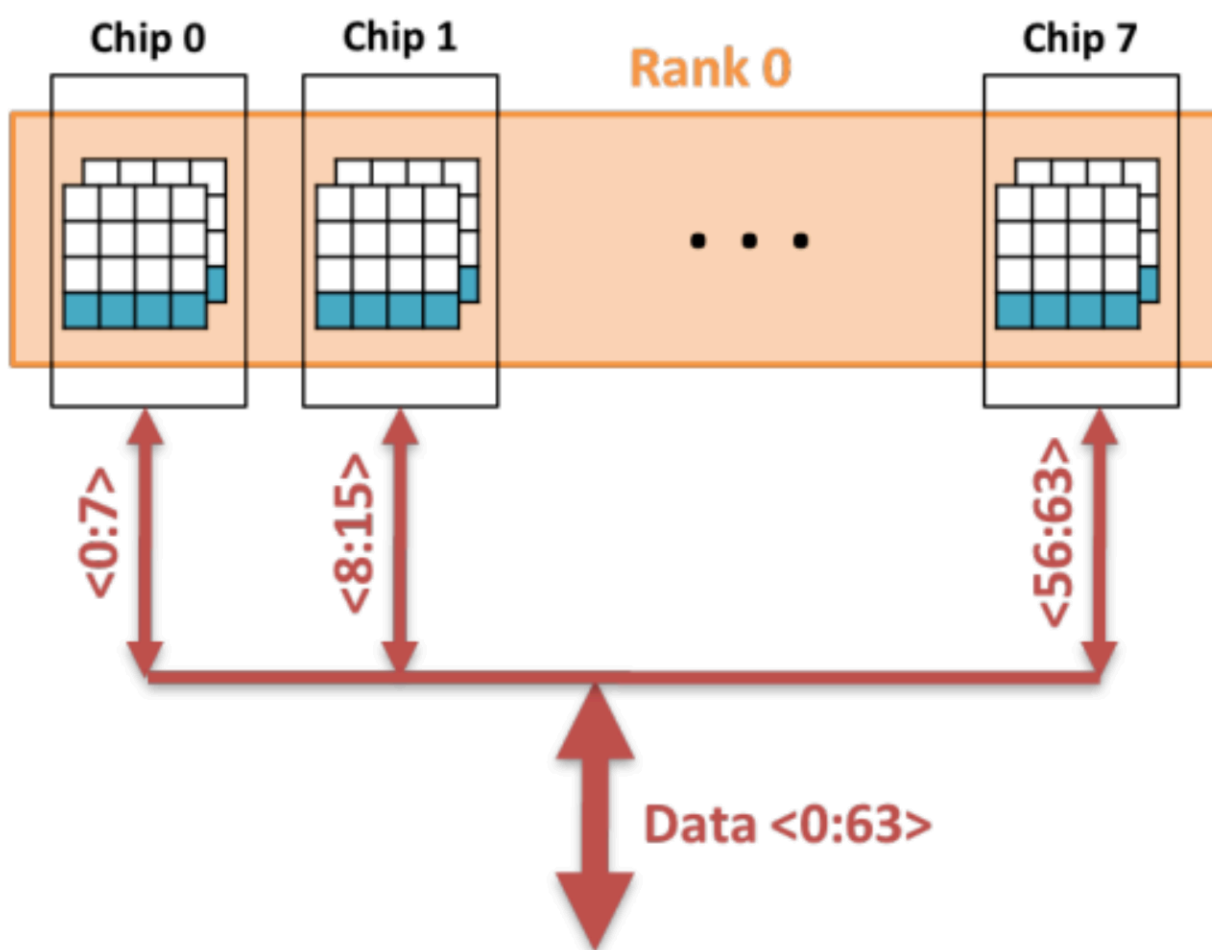
rank和DIMM

电路板加板上的芯片，这就是所谓的内存条，也称为 DIMM 条（全称 Dual-Inline-Memory-Modules，中文名叫双列直插式存储模块）。内存条通过“内存通道”连接到内存控制器，一组可以被一个内存通道同时访问的芯片称作一个 rank。一个 rank 中的每个芯片都共用内存通道提供的地址线、控制线和数据线，同时每个芯片都提供一组输出线，这些输出线组合起来就是内存条的输出线。

下图是一个包含 8 颗芯片的 DIMM 条。这 8 颗芯片被一个内存通道同时访问，所以它们合称为一个 rank。有的 DIMM 条有两面，即两面都有内存芯片，这种 DIMM 条拥有两个 rank。



假设上图中的每个芯片都包含 8 个 bank，每个 bank 都包含 8 个阵列，那么这条内存条就可以一次读写 $8 \times 8 = 64$ 比特，其中第一个 8 是指每个芯片输出 8 位（即为每一个 bank 是由 8 个阵列构成的可以一次输出 8bit），第二个 8 是指这个 rank 总共有 8 颗芯片（即为上图的 8 个黑方块，每一个芯片都可以输出 8bit），因为这 8 颗芯片被同一个内存通道访问，所以其被访问的 bank 和 bank 内的行地址、列地址都是完全一致的。下图是一个描述这个过程的简图：显然，我们在读写 8 颗芯片同一个 bank 同一个位置的 cell，注意，图中没有显示不在工作状态的 bank。



对一个 rank 读写，即同时读写 rank 内 8 个存储芯片内的同一位置的 bank



电脑有时候可以插入多个内存条，多个内存条有助于提升电脑的内存容量，但是未必能提高电脑的速度。电脑的速度受“内存通道”数限制，如果电脑有四个插槽，却只有一个内存通道，那么 CPU 仍然只能一次访问一个 rank；但如果电脑有四个插槽的同时还有四个内存通道，那么 CPU 就可以一次访问四个 rank，很显然，四并行访问明显比串行访问快，假设每个 rank

可以输出 64 比特，那么四通道就可以一次访问 $4 \times 64 = 256$ 比特，而单通道只能访问 64 比特。

OpenPage Policy和Close Page Policy

DRAM访问有两种模式，一个Open Page 一个是Close Page。前者在完成一次访存后保留row buffer的内容，如果下一个访存命令恰好也在同一个row上，就会row buffer hit，节省访问时间，但如果后一个访问地址不在同一个row上，就可row buffer conflict，增加了访存时间。后者在完成一次访存后立即执行precharge命令，即将row buffer的内容写回cell，这种情况下下一个访存一定是empty row buffer状态。

TMP

 现代计算设备使用DRAM作为主存。为了提高内存带宽，DRAM通常被划分为多个通道。每个通道可能与多个双列直插内存模块（DIMM）相关联。每个DIMM有一个或多个rank（例如，现代DIMM可以是单/双/四/八rank），每个rank有多个bank（例如，DDR3通常有8个bank，DDR4有16个bank）。如图1所示，每个bank可以看作是一个二维的内存字数组，按行和列组织。内存字的大小取决于数据总线宽度，并决定存储其内容需要多少个单元（例如，存储一个64位内存字需要64个单元）。每个单元由一个电容器和一个晶体管组成，电容器充电或未充电以表示二进制值1，晶体管用于访问电容器。在每个bank中，还有一个行缓冲区，可以保存单行的内容。要访问一个单元，必须先激活相应的行，将行的内容放入行缓冲区，然后从行缓冲区提供访问。激活的行保持在行缓冲区中，直到被内存控制器关闭，在此之前，对该行的连续访问将直接从行缓冲区提供。根据使用的内存控制器策略，激活的行可能由于不同原因被关闭：如果内存控制器使用开放页策略，激活的行将不会关闭，直到同一bank中的不同行被访问；这种因果事件通常称为行冲突。另一方面，如果使用关闭页策略，内存控制器将主动关闭行[25]，[42]。  Leveraging EM Side-Channel Information to Detect Rowhammer Attacks