

例如,要从图 6-3 中  $16 \times 8$  的 DRAM 中读出超单元(2, 1), 内存控制器发送行地址 2, 如图 6-4a 所示。DRAM 的响应是将行 2 的整个内容都复制到一个内部行缓冲区。接下来, 内存控制器发送列地址 1, 如图 6-4b 所示。DRAM 的响应是从行缓冲区复制出超单元(2, 1)中的 8 位, 并把它们发送到内存控制器。

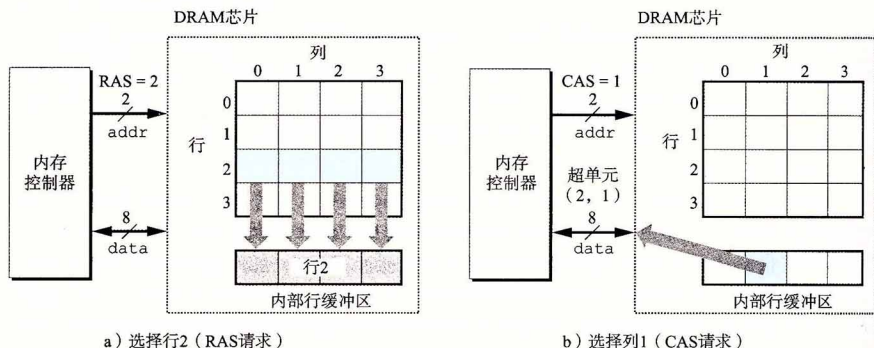


图 6-4 读一个 DRAM 超单元的内容

电路设计者将 DRAM 组织成二维阵列而不是线性数组的一个原因是降低芯片上地址引脚的数量。例如, 如果示例的 128 位 DRAM 被组织成一个 16 个超单元的线性数组, 地址为 0~15, 那么芯片会需要 4 个地址引脚而不是 2 个。二维阵列组织的缺点是必须分两步发送地址, 这增加了访问时间。


#### 4. 内存模块

DRAM 芯片封装在内存模块(memory module)中, 它插到主板的扩展槽上。Core i7 系统使用的 240 个引脚的双列直插内存模块(Dual Inline Memory Module, DIMM), 它以 64 位为块传送数据到内存控制器和从内存控制器传出数据。

图 6-5 展示了一个内存模块的基本思想。示例模块用 8 个 64 Mbit 的  $8 \times 8$  的 DRAM 芯片, 总共存储 64MB(兆字节), 这 8 个芯片编号为 0~7。每个超单元存储主存的一个字节, 而用相应超单元地址为  $(i, j)$  的 8 个超单元来表示主存中字节地址  $A$  处的 64 位字。在图 6-5 的示例中, DRAM 0 存储第一个(低位)字节, DRAM 1 存储下一个字节, 依此类推。

要取出内存地址  $A$  处的一个字, 内存控制器将  $A$  转换成一个超单元地址  $(i, j)$ , 并将它发送到内存模块, 然后内存模块再将  $i$  和  $j$  广播到每个 DRAM。作为响应, 每个 DRAM 输出它的  $(i, j)$  超单元的 8 位内容。模块中的电路收集这些输出, 并把它们合并成一个 64 位字, 再返回给内存控制器。

通过将多个内存模块连接到内存控制器, 能够聚合成主存。在这种情况下, 当控制器收到一个地址  $A$  时, 控制器选择包含  $A$  的模块  $k$ , 将  $A$  转换成它的  $(i, j)$  的形式, 并将  $(i, j)$  发送到模块  $k$ 。

 **练习题 6.1** 接下来, 设  $r$  表示一个 DRAM 阵列中的行数,  $c$  表示列数,  $b_r$  表示行寻址所需的位数,  $b_c$  表示列寻址所需的位数。对于下面每个 DRAM, 确定 2 的幂数的阵列维数, 使得  $\max(b_r, b_c)$  最小,  $\max(b_r, b_c)$  是对阵列的行或列寻址所需的位数中较大的值。