maintain the memory contents. Requiring a battery increases the complexity of the system and also begs the question of what happens when the battery wears out. In the 1980s, it was common for a PC's BIOS configuration to be stored in battery-backed CMOS SRAM. This is how terms like "the CMOS" and "CMOS setup" entered the lexicon of PC administration.

SRAM is implemented not only as discrete memory chips but is commonly found integrated within other types of chips, including microprocessors. Smaller microprocessors or *microcontrollers* (microprocessors integrated with memory and peripherals on a single chip) often contain a quantity of on-board SRAM. More complex microprocessors may contain on-chip data caches implemented with SRAM.

4.6 ASYNCHRONOUS DRAM

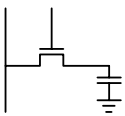

FIGURE 4.9 DRAM bit structure.

SRAM may be the easiest volatile memory to use, but it is not the least expensive in significant densities. Each bit of memory requires between four and six transistors. When millions or billions of bits are required, the complexity of all those transistors becomes substantial. Dynamic RAM, or DRAM, takes advantage of a very simple yet fragile storage component: the capacitor. A capacitor holds an electrical charge for a limited amount of time as the charge gradually drains away. As seen from EPROM and flash devices, capacitors can be made to hold charge almost indefinitely, but the penalty for doing so is significant complexity in modifying the storage element. Volatile memory must be both quick to access and not be subject to write-cycle limitations—both of which are restrictions of nonvolatile memory technologies. When a capacitor is designed to have its charge quickly and easily manipulated, the downside of rapid discharge emerges. A very efficient volatile storage element can be created with a capacitor and a single transistor as shown in Fig. 4.9, but that capacitor loses its contents soon after being charged. This is where the term *dynamic* comes from in DRAM—the memory cell is indeed dynamic under steady-state conditions. The solution to this problem of solid-state amnesia is to periodically refresh, or update, each DRAM bit before it completely loses its charge.

As with SRAM, the pass transistor enables both reading and writing the state of the storage element. However, a single capacitor takes the place of a multitransistor latch. This significant reduction in bit complexity enables much higher densities and lower per-bit costs when memory is implemented in DRAM rather than SRAM. This is why main memory in most computers is implemented using DRAM. The trade-off for cheaper DRAM is a degree of increased complexity in the memory control logic. The number one requirement when using DRAM is periodic refresh to maintain the contents of the memory.

DRAM is implemented as an array of bits with rows and columns as shown in Fig. 4.10. Unlike SRAM, EPROM, and flash, DRAM functionality from an external perspective is closely tied to its row and column organization.

SRAM is accessed by presenting the complete address simultaneously. A DRAM address is presented in two parts: a row and a column address. The row and column addresses are multiplexed onto the same set of address pins to reduce package size and cost. First the row address is loaded, or strobed, into the row address latch via *row address strobe*, or RAS*, followed by the column address with *column address strobe*, or CAS*. Read data propagates to the output after a specified access time. Write data is presented at the same time as the column address, because it is the column strobe that actually triggers the transaction, whether read or write. It is during the column address phase that WE* and OE* take effect.

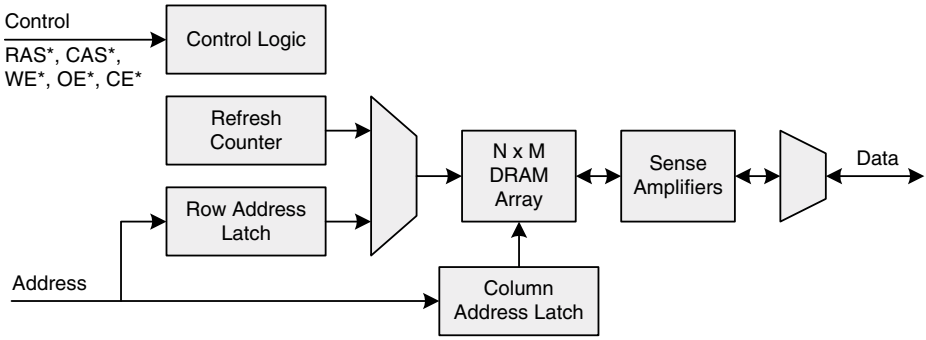

FIGURE 4.10 DRAM architecture.

Sense amplifiers on the chip are necessary to detect the minute charges that are held in the DRAM's capacitors. These amplifiers are also used to assist in refresh operations. It is the memory controller's responsibility to maintain a refresh timer and initiate refresh operations with sufficient frequency to guarantee data integrity. Rather than refreshing each bit separately, an entire row is refreshed at the same time. An internal refresh counter increments after each refresh so that all rows, and therefore all bits, will be cycled through in order. When a refresh begins, the refresh counter enables a particular memory row. The contents of the row are detected by the sense amplifiers and then driven back into the bit array to recharge all the capacitors in that row. Modern DRAMs typically require a complete refresh every 64 ms. A 64-Mb DRAM organized as 8,388,608 words × 8 bits (8 MB) with an internal array size of 4,096 × 2,048 bytes would require 4,096 refresh cycles every 64 ms. Refresh cycles need not be evenly spaced in time but are often spread uniformly for simplicity.

The complexity of performing refresh is well worth the trouble because of the substantial cost and density improvements over SRAM. One downside of DRAM that can only be partially compensated for is its slower access time. A combination of its multiplexed row and column addressing scheme plus its large memory arrays with complex sense and decode logic make DRAM significantly slower than SRAM. Mainstream computing systems deal with this speed problem by implementing SRAM-based cache mechanisms whereby small chunks of memory are prefetched into fast SRAM so that the microprocessor does not have to wait as long for new data that it requests.

Asynchronous DRAM was the prevailing DRAM technology until the late 1990s, when synchronous DRAM, or SDRAM, emerged as the dominant solution to main memory. At its heart, SDRAM works very much like DRAM but with a synchronous bus interface that enables faster memory transactions. It is useful to explore how older asynchronous DRAM works so as to understand SDRAM. SDRAM will be covered in detail later in the book.

RAS* and CAS* are the two main DRAM control signals. They not only tell the DRAM chip which address is currently being asserted, they also initiate refresh cycles and accelerate sequential transactions to increase performance. A basic DRAM read works as shown in Fig. 4.11. CE* and OE* are both assumed to be held active (low) throughout the transaction.

A transaction begins by asserting RAS* to load the row address. The strobes are falling-edge sensitive, meaning that the address is loaded on the falling edge of the strobe, sometime after which the address may change. Asynchronous DRAMs are known for their myriad detailed timing requirements. Every signal's timing relative to itself and other signals is specified in great detail, and these parameters must be obeyed for reliable operation. RAS* is kept low for the duration of the transaction. Assertion of CAS* loads the column address into the DRAM as well as the read or write status