sometimes written as E^2PROM, is more expensive to manufacture per bit than EPROM or flash, because individual bytes may be erased randomly without affecting neighboring locations. Because of the complexity and associated cost of making each byte individually erasable, EEPROM is not commonly manufactured in large densities. Instead, it has served as a niche technology for applications that require small quantities of flexible reprogrammable ROM. Common uses for EEPROM are as program memory in small microprocessors with embedded memory and as small nonvolatile memory arrays to hold system configuration information. Serial EEPROM devices can be found in eight-pin DIP or SOIC packages and provide up to several kilobytes of memory. Their serial interface, small size, and low power consumption make them very practical as a means to hold serial numbers, manufacturing information, and configuration data.

Parallel EEPROM devices are still available from manufacturers as the 28xx family. They are pin and function compatible (for reads) with the 27xxx EPROM family that they followed. Some applications requiring reprogrammable nonvolatile memory may be more suited to EEPROM than flash, but flash is a compelling choice, because it is the more mainstream technology with the resultant benefit of further cost reduction.

Serial EEPROMs, however, are quite popular due to their very small size and low power consumption. They can be squeezed into almost any corner of a system to provide small quantities of nonvolatile storage. Microchip Technology is a major manufacturer of serial EEPROMs and offers the 24xx family. Densities range from 16 bytes to several kilobytes. Given that serial interfaces use very few pins, these EEPROMs are manufactured in packages ranging from eight-pin DIPs to five-pin SOT-23s that are smaller than a fingernail. Devices of this sort are designed to minimize system impact rather than for speed. Their power consumption is measured in nanoamps and microamps instead of milliamps, as is the case with standard flash, parallel EEPROM, and EPROM devices.

Microchip's 24LC00 is a 16-byte serial EEPROM with a two-wire serial bus. It requires only four pins: two for power and two for data communication. Like most modern flash devices, the 24LC00 is rated for one million write cycles. When not being accessed, the 24LC00 consumes about 250 nA! When active, it consumes only 500 μA. For added flexibility, the 24LC00 can operate over a variety of supply voltages from 2.5 to 6.0 V. Speed is not a concern here: writes take up to 4 ms to complete, which is not a problem when writing only a few bytes on rare occasions.

4.5 ASYNCHRONOUS SRAM

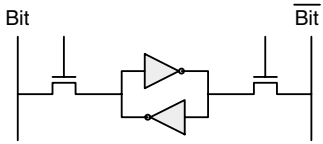

FIGURE 4.7 SRAM bit feedback latch.

Static RAM, or SRAM, is the most basic and easy to use type of volatile memory and is found in almost every computer in one form or another. An SRAM device is conceptually easy to understand, consisting of an array of latches along with control and decode logic to resolve the address that is being read or written at any given time. Each latch is a feedback circuit that traps and maintains a particular logic state. A typical SRAM bit implementation is shown in Fig. 4.7.

An SRAM latch is created by connecting two inverters in a loop. One side of the loop remains stable at the desired logic state, and the other remains stable at the opposite state. Inverters are used rather than noninverting buffers, because an inverter is the simplest logic element to construct. The two pass transistors on either side of the latch enable both writing and reading. When writing, the transistors turn on and force each half of the loop to whatever state is driven on the vertical bit lines. When reading, the transistors also turn on, but the bit lines are sensed rather than driven. Typical

SRAM implementations require six transistors per bit of memory: two transistors for each inverter and the two pass transistors. Some implementations use only a single transistor per inverter, requiring only four transistors per bit.

Discrete asynchronous SRAM devices have been around for decades. In the 1980s, the 6264 and 62256 were manufactured by multiple vendors and used in applications that required simple RAM architectures with relatively quick access times and low power consumption. The 62xxx family is numbered according to its density in kilobits. Hence, the 6264 provides 65,536 bits of RAM arranged as 8k × 8. The 62256 provides 262,144 bits of RAM arranged as 32k × 8. Being manufactured in CMOS technology and not using a clock, these devices consume very little power and draw only microamps when not being accessed.

The 62xxx family pin assignment is virtually identical to that of the 27xxx EPROM family, enabling system designs where either EPROM or SRAM can be substituted into the same location with only a couple of jumpers to set for unique signals such as the program-enable on an EPROM or write-enable on an SRAM. Like an EPROM or basic flash device, asynchronous SRAMs have a simple interface consisting of address, data, chip select, output enable, and write enable. This interface is shown in Fig. 4.8.

Writes are performed whenever the WE* signal is held low. Therefore, one must ensure that the desired address and data are stable before asserting WE* and that WE* is removed while address and data remain stable. Otherwise, the write may corrupt an undesired memory location. Unlike an EPROM, but like flash, the data bus is bidirectional during normal operation. The first two transactions shown are writes as evidenced by the separate assertions of WE* for the duration of address and data stability. As soon as the writes are completed, the microprocessor should release the data bus to the high-impedance state. When OE* is asserted, the SRAM begins driving the data bus and the output reflects the data contents at the locations specified on the address bus.

Asynchronous SRAMs are available with access times of less than 100 ns for inexpensive parts and down to 10 ns for more expensive devices. Access time measures both the maximum delay between a stable read address and its corresponding data and the minimum duration of a write cycle. Their ease of use makes them suitable for small systems where megabytes of memory are not required and where reduced complexity and power consumption are key requirements. Volatile memory doesn't get any simpler than asynchronous SRAM.

Prior to the widespread availability of flash, many computer designs in the 1980s utilized asynchronous SRAM with a battery backup as a means of implementing nonvolatile memory for storing configuration information. Because an idle SRAM draws only microamps of current, a small battery can maintain an SRAM's contents for several years while the main power is turned off. Using SRAM in this manner has two distinct advantages over other technologies: writes are quick and easy, because there are no complex EEPROM or flash programming algorithms, and there is no limit to the number of write cycles performed over the life of the product. The downsides to this approach are a lack of security for protecting valuable configuration information and the need for a battery to

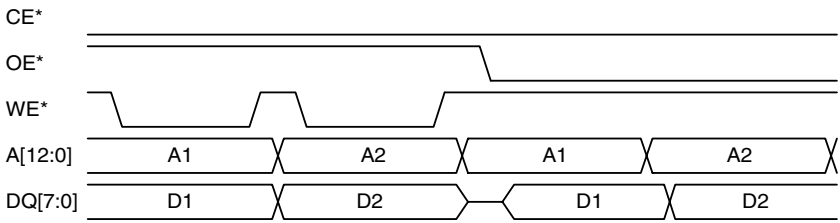

FIGURE 4.8 62xxx SRAM interface.