nous DRAMs have associated pulse width and signal-to-signal delay specifications that are tied closely to the characteristics of their internal memory arrays. When maximum bandwidth is desired at high clock frequencies, these specifications become difficult to meet. It is easier to design a system in which all interfaces and devices run synchronously so that interface timing becomes an issue of meeting setup and hold times, and functional timing becomes an issue of sequencing signals on discrete clock edges.

Synchronous DRAM, or SDRAM, is a twist on basic asynchronous DRAM technology that has been around for more than three decades. SDRAM can essentially be considered as an asynchronous DRAM array surrounded by a synchronous interface on the same chip, as shown in Fig. 8.1. A key architectural feature in SDRAMs is the presence of multiple independent DRAM arrays—usually either two or four banks. Multiple banks can be activated independently and their transactions interleaved with those of other banks on the IC's synchronous interface. Rather than creating a bottleneck, this functionality allows higher efficiency, and therefore higher bandwidth, across the interface. One factor that introduces latency in random accesses across all types of DRAM is the row activation time: a row must first be activated before the column address can be presented and data read or written. An SDRAM allows a row in one bank to be activated while another bank is actively engaged in a read or write, effectively hiding the row activation time in the other bank. When the current transaction completes, the previously activated row in the other bank can be called upon to perform a new transaction without delay, increasing the device's overall bandwidth.

The synchronous interface and internal state logic direct interleaved multibank operations and burst data transfers on behalf of an external memory controller. Once a transaction has been started, one data word flows into or out of the chip on every clock cycle. Therefore, an SDRAM running at 100 MHz has a theoretical peak bandwidth of 100 million words per second. In reality, of course, this number is somewhat lower because of refresh and the overhead of beginning and terminating transactions. The true available bandwidth for a given application is very much dependent on that application's data transfer patterns and the capabilities of its memory controller.

Rather than implementing a DRAM-style asynchronous interface, the SDRAM's internal state logic operates on discrete commands that are presented to it. There are still familiar sounding signals such as RAS* and CAS*, but they function synchronously as part of other control signals to form commands rather than simple strobes. Commands begin and terminate transactions, perform refresh operations, and configure the SDRAM for interface characteristics such as default burst length.

SDRAM can provide very high bandwidth in applications that exploit the technology's burst transfer capabilities. A conventional computer with a long-line cache subsystem might be able to fetch 256 words in as few as 260 cycles: 98.5 percent efficiency! Bursts amortize a fixed number of overhead cycles across the entire transaction, greatly improving bandwidth. Bandwidth can also be improved by detecting transactions to multiple banks and interleaving them. This mode of operation

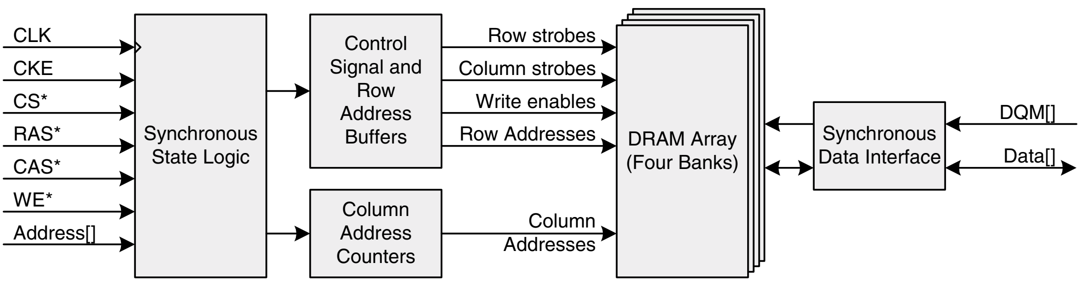

FIGURE 8.1 Basic SDRAM architecture.

allows some new burst transfers to be requested prior to the current burst ending, thereby hiding the initial startup latency of the subsequent transaction.

Most of the input signals to the state logic shown in Fig. 8.1 combine to form the discrete commands listed in Table 8.1. A clock enable, CKE, must be high for normal operation. When CKE is low, the SDRAM enters a low-power mode during which data transactions are not recognized. CKE can be tied to logic 1 for applications that are either insensitive to power savings or require continual access to the SDRAM. Interface signals are sampled on the rising clock edge. Many SDRAM devices are manufactured in multibyte data bus widths. The data mask signals, DQM[ ], provide a convenient way to selectively mask individual bytes from being written or being driven during reads. Each byte lane has an associated DQM signal, which must be low for the lane to be written or to enable the lane's tri-state buffers on a read.

TABLE 8.1 Basic SDRAM Command Set

Command	CS*	RAS*	CAS*	WE*	Address	AP/A10
Bank activate	L	L	H	H	Bank, row	A10
Read	L	H	L	H	Bank, column	L
Read with auto-precharge	L	H	L	H	Bank, column	H
Write	L	H	L	L	Bank, column	L
Write with auto-precharge	L	H	L	L	Bank, column	H
No operation	L	H	H	H	X	X
Burst terminate	L	H	H	L	X	X
Bank precharge	L	L	H	L	X	L
Precharge all banks	L	L	H	L	X	H
Mode register set	L	L	L	L	Configuration	Configuration
Auto refresh	L	L	L	H	X	X
Device deselect	H	X	X	X	X	X

Some common functions include activating a row for future access, performing a read, and precharging a row (deactivating a row, often in preparation for activating a new row). For complete descriptions of SDRAM interface signals and operational characteristics, SDRAM manufacturers' data sheets should be referenced directly. Figure 8.2 provides an example of how these signals are used to implement a transaction and serves as a useful vehicle for introducing the synchronous interface. CS* and CKE are assumed to be tied low and high, respectively, and are not shown for clarity.

The first requirement to read from an SDRAM is to activate the desired row in the desired bank. This is done by asserting an activate (ACTV) command, which is performed by asserting RAS* for one cycle while presenting the desired bank and row addresses. The next command issued to continue the transaction is a read (RD). However, the controller must wait a number of cycles that translates into the DRAM array's row-activate to column-strobe delay time. The timing characteristics of the underlying DRAM array is expressed in nanoseconds rather than clock cycles. Therefore, the integer number of delay cycles is different for each design, because it is a function of the clock period and the internal timing specification. If, for example, an SDRAM's RAS* to CAS* delay is 20 ns, and the clock period is 20 ns or slower, an RD command could be issued on the cycle immediately