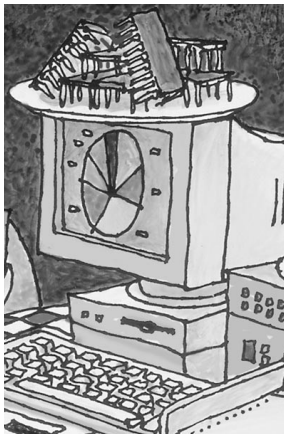
SLDRAM: HIGH-PERFORMANCE, OPEN-STANDARD MEMORY

Peter Gillingham

MOSAID Technologies

Bill Vogley

Texas Instruments



Synchronous-link DRAM—the highest rung yet on DRAM's evolutionary ladder—is a high-bandwidth interface standard meeting memory requirements into the 21st century.

The primary objective of DRAM—dynamic random access memory—is to offer the largest memory capacity at the lowest possible cost. Designers achieve this by two means. First, they optimize the process and the design to minimize die area. Second, they ensure that the device serves high-volume markets and can be mass-produced to achieve the greatest economies of scale.

SLDRAM—synchronous-link DRAM—is a new memory interface specification developed through the cooperative efforts of leading semiconductor memory manufacturers and high-end computer architects and system designers. SLDRAM meets the high data bandwidth requirements of emerging processor architectures and retains the low cost of earlier DRAM interface standards. These and other benefits suggest that SLDRAM will become the mainstream commodity memory of the early 21st century.

Benefits

In developing SLDRAM (see Origins box for details), we had several goals beyond high bandwidth and low cost. It was also important that system designers could adopt the new interface smoothly, that it would work for a wide variety of applications, and that its manufacturing yields would be high. Finally, it was important for SLDRAM to be an open standard.

Evolutionary solution. Over more than 20 years encompassing nine generations of DRAMs (from 1 Kbit to 64 Mbits), the market has always embraced evolutionary solutions. With evolutionary changes, system designers can adapt existing technology and migrate to higher performance solutions. As shown in Figure 1 next page, SLDRAM represents the next step in DRAM's evolution, after EDO, SDRAM, and DDR (extended data out, synchronous, and double-data-rate DRAM).

SDRAM includes several important architectural features over standard EDO, including multiple internal banks, a clocked synchronous interface, terminated small-swing signaling, and programmable data bursts. These changes decouple internal DRAM address and control paths from the data interface to achieve higher bandwidth. The emerging standard for DDR adds data clocking on both clock edges and a return clock, allowing higher speed operation with an improved system timing margin.

SLDRAM builds on the features of SDRAM and DDR by adding an address/control packet protocol, in-system timing and signaling optimization, and full compatibility from generation to generation. Command packets in the SLDRAM protocol include spare bits to accommodate addressing for the 4-Gbit generation and beyond.

The first SLDRAM samples will be 64-Mbit devices operating at 400 Mbps/pin. Interfaces with bandwidths of 600 and 800 Mbps/pin, and eventually greater than 1 Gbps/pin, will come on the market when they become cost-effective. The protocol allows us to mix interfaces of different speeds. So, for example, we can plug an 800-Mbps/pin device into a 400-Mbps/pin system, and it will operate correctly at 400 Mbps/pin.

Application variety. The standards working group specified SLDRAM as a general-purpose, high-performance DRAM for a wide variety of applications. As computer main memory—in desktop, mobile, and high-end servers and workstations—SLDRAM offers high sustainable bandwidth, low latency, low power, user upgradability, and support for large hierarchical memory configurations. For video, graphics, and telecommunications applications, SLDRAM provides multiple independent banks, fast read/write bus turnaround, and the capabil-

0272-1732/97/$10.00 © 1997 IEEE

Origins of SLDRAM

Only recently, the computer industry made the transition from fast-page-mode DRAM to EDO (extended data out) DRAM. Now, it has accepted SDRAM (synchronous DRAM) as the standard for mainstream computer applications.

In the early days of SDRAM, as the standards were being developed, both manufacturers and users were concerned about cost. They did not know how much silicon the new functions would consume and how much additional complexity it would take to test and verify them. With SDRAM now fully standardized, these issues have been resolved, and the technology has achieved volume production, commodity-level pricing.

During the development of SDRAM in 1989 and 1990, the SCI working group (IEEE Std. P1596—Scalable Coherent Interface) was developing a memory interface as a subgroup. RamLink, which became IEEE Std. 1596.4 for memory interfaces, is a point-to-point interface using the SCI protocol but with a reduced command set that makes it memory specific. During RamLink's development, the working group realized that the interface would have to support large memory configurations. The protocol allowed this, and so did the architecture. The drawback was the serial delay from one device to the next in a point-to-point system. This increased latency from 4 to 6 ns per device on the interface. If a configuration incorporated 64 devices, the additional latency would be intolerable—around 300 to 400 ns.

It made sense to stick with the RamLink protocol, but we needed a parallel interface. Because the new devices would fundamentally be synchronous DRAMs linked in a different manner, the group chose the name SyncLink. A little over a year ago, we discovered that Synclink had already been registered and used as a trademark. Thus, we instead adopted the name SLDRAM, for synchronous link DRAM. We retained the RamLink protocol but made the interface parallel.

To determine the most efficient method of exchanging address, control, and read and write data between the controller and memory devices, we simulated various bus configurations with actual traces of processor memory requests. In real systems, main memory address requests are virtually random, and the read/write ratio averages about 4:1.

We rejected several configurations, including one with a single, bidirectional bus for address, control, and data. Because of the need to interrupt data flow to issue a command in this configuration, effective data bandwidth was a small fraction of theoretical peak bandwidth. The single bus sacrificed the benefits of hidden bank activation and deactivation.

We also rejected a configuration having one unidirectional bus from controller to memory for command, address, and write data, and another from memory to controller for read data. Although this was an improvement over the single bus, effective bandwidth still suffered from the collision of commands with write data.

The configuration we chose consists of one unidirectional bus for command and address and a bidirectional bus for read and write data. In this optimal configuration, the data bus bandwidth could be fully utilized. Placement of the command and address in a packet reduces pin count, allows future address expansion without pin increments, and enables higher pipelining capability.

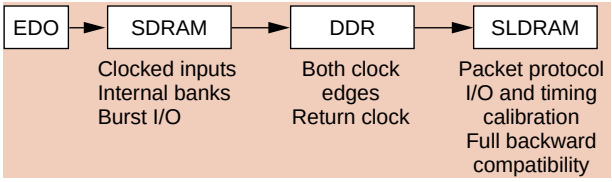


Figure 1. Evolution from EDO to SLDRAM, by way of SDRAM and DDR.

ity for small, fully pipelined bursts. SLDRAM addresses the requirements of all major high-volume DRAM applications.

Cost-effectiveness. As was the case with SDRAM, industry will adopt a new, high-performance DRAM interface once its cost premium over the conventional alternative falls below 5% (see Figure 2, next page). Designers achieve high performance by improving the interface while leaving the DRAM core relatively unchanged. We could easily have improved core performance with reduced bit-line, word-line, and data-bus loading. However, this would have generated a significant die area penalty because of increased array fragmentation, grossly exceeding the 5% cost target.

SDRAM, DDR, and SLDRAM all use a DRAM core that has a page-mode cycle time of roughly 10 ns. To maintain an efficient die layout and obtain an interface rate higher than the core cycle time, the device must fetch several words in parallel. For example, a 16-bit I/O, 200-Mbps/pin, DDR device must read or write two I/O words over a 32-bit internal data path in one 10-ns core cycle.

The need to widen the internal data path also leads to a die cost penalty. For the 16-Mbit generation of SDRAM (100-Mbps/pin, 16-bit I/O), this penalty has recently fallen below 5%. DDR running at 200 Mbps/pin will become cost-effective for 64-Mbit devices, which have a sufficient number of active memory subarrays to support a 32-bit data path without substantial area penalty. SLDRAM devices will first become available with a 400-Mbps/pin, 16-bit I/O interface employing a 64-bit internal data path. These devices will be cost-effective in the 256-Mbit density. Later SLDRAM devices will run at 800 Mbps/pin and employ 128-bit internal data paths. SLDRAM or any other memory technology using a 128-bit internal data path will not be cost-effective until the 1-Gbit generation.

Manufacturability. DRAM's low cost is due to high manufacturing yields, but the tight timing specifications required for high-frequency operation are incompatible with high yield. Therefore, we defined SLDRAM so that virtually any functional part will meet the specification in one speed grade

or another. On power-up, a system can ascertain the speed performance capabilities of all SLDRAMs present and then make the appropriate adjustments. The SLDRAM packet protocol permits the system to adjust and match setup and hold time, data delay, and output drive levels of individual memory devices for consistent system operation. Periodically during operation, the system recalibrates the devices to account for system drift. (We will discuss this in more detail later in the article.) The flexibility afforded by the SLDRAM packet protocol ensures high yield and low cost.

Low system cost. SLDRAM achieves low system cost through conventional packaging and printed-circuit-board technology. The SLDRAM devices themselves are packaged in standard, 0.5-mm pitch TSOPs (thin, small-outline packages) or in 0.8-mm, staggered-pitch vertically mounted packages (VSMPs). With buffered modules, an SLDRAM controller must support only 33 high-speed signals to accommodate gigabyte memory configurations. Wide modules can add 16-bit data channels without additional control overhead for increased memory bandwidth. For the SLDRAM interface, we recommend conventional low-cost PCB material with 5-mm tracks, using two of four layers for interconnect.

Open standard. SLDRAM is an open standard that will be formalized by IEEE (Std P1596.7) and JEDEC specifications. Open standards allow manufacturers to develop differentiated products that address emerging applications and niche´ opportunities. Open competition will ensure rapid development of DRAM technology at the lowest possible cost.

Architectural and functional overview

In this section we describe how SLDRAM works in a system, covering the basics of SLDRAM protocol, timing, and signals. For a full description of SLDRAM functionality, we encourage readers to refer to the SLDRAM data sheet, which is available on the SLDRAM Consortium Web page (www.sldram.com).

Bus topology. The SLDRAM multidrop bus has one memory controller and up to eight loads. A load can be either a single SLDRAM device or a buffered module with many SLDRAM devices. Command, address, and control information from the memory controller flows to the SLDRAMs on the unidirectional CommandLink. Read and write data flow between controller and SLDRAM on the bidirectional DataLink. Both CommandLink and DataLink operate at the same rate (400-Mbps/pin, 600-Mbps/pin, 800-Mbps/pin, and so on). Figure 3 illustrates SLDRAM signals and data flow. In actual SLDRAM modules, all connections are on one side.

Signal names and definitions. The CommandLink comprises signals CCLK, CCLK*, FLAG, CA[9:0], LISTEN, LINKON, and RESET. (See Table 1 for descriptions.) Commands consist of four consecutive 10-bit words on CA[9:0]. A 1 on the FLAG bit indicates the first word of a command. The SLDRAMs use both edges of the differential free-running clock (CCLK/CCLK*) to latch command words. For a 400-Mbps/pin SLDRAM, the clock frequency is 200 MHz, and bit period *N* is equal to 2.5 ns—half the clock period. While the LISTEN pin is high, the SLDRAMs monitor the CommandLink for commands. When LISTEN is low, there can be no commands on the CommandLink, so the SLDRAMs enter a power-saving standby mode. When LINKON is low, the SLDRAMs enter a shutdown mode, in which CCLK can be turned off to achieve zero power on the link. A RESET signal puts the SLDRAMs in a known state on power-up.

The DataLink is a bidirectional bus for the transmission of write data from controller to the SLDRAMs and read data from the SLDRAMs back to the controller. It consists of DQ[17:0], DCLK0, DCLK0*, DCLK1, and DCLK1*. Read and write data packets of a minimum burst length of four are

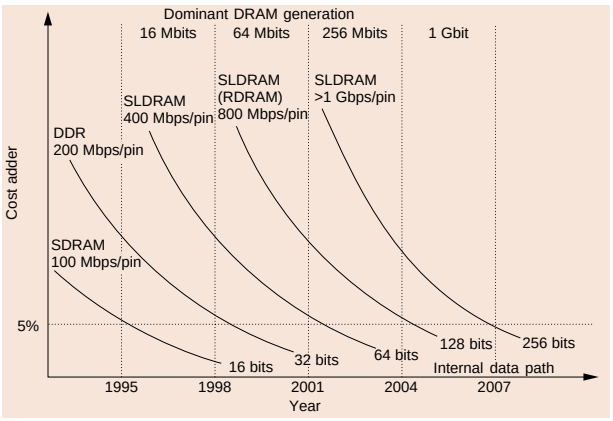


Figure 2. DRAM bandwidth cost penalty for 16-bit I/O.

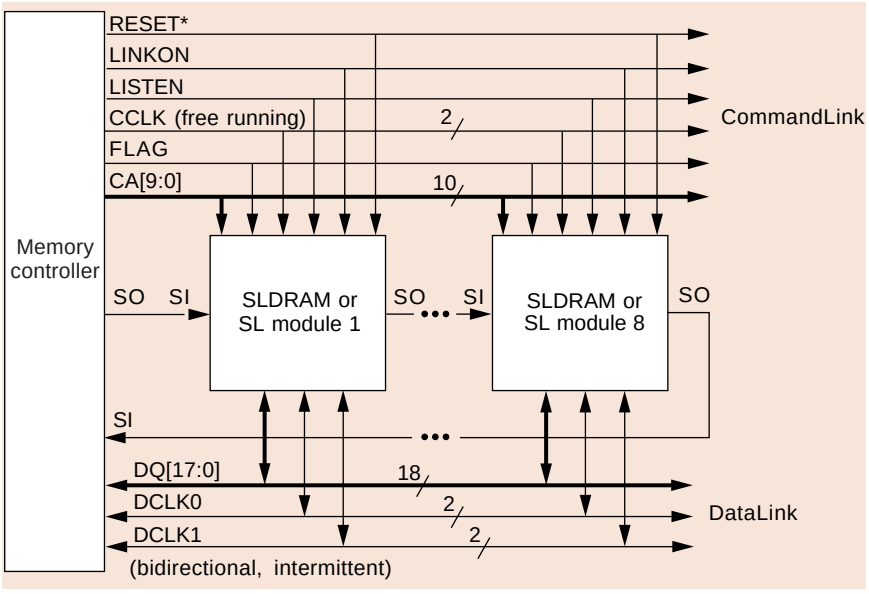


Figure 3. SLDRAM bus topology.

Table 1. SLDRAM signal names and descriptions. MC: memory controller; SSTL_2: series stub terminated logic for 2.x-V generation.

Bus	Signal	Description	Direction	Level
CommandLink	CCLK/CCLK*	Command clock	MC → SLDRAM	SLIO
	CA[9:0]	Command address bus	MC → SLDRAM	SLIO
	LISTEN	Standby mode	MC → SLDRAM	SLIO
	LINKON	Shutdown mode	MC → SLDRAM	LVTTL
	RESET	Hard reset	MC → SLDRAM	LVTTL
DataLink	DCLK0/DCLK0*	Data clock 0	MC fl↔ SLDRAM	SLIO
	DCLK1/DCLK1*	Data clock 1	MC fl↔ SLDRAM	SLIO
	DQ[17:0]	Data bus	MC fl↔ SLDRAM	SLIO
Serial	SI	Serial input	MC → SLDRAM, ´ SLDRAM → SLDRAM	LVTTL
	SO	Serial output	SLDRAM → SLDRAM, SLDRAM → MC	LVTTL

accompanied by either differential clock—DCLK0/DCLK0* or DCLK1/DCLK1*. The two sets of clocks allow control of the DataLink to pass from one device to another with the minimum gap. On power-up, a daisy-chained serial bus with input SI and output SO on each device synchronizes the SLDRAMs and assigns unique IDs to each one.

Pipelined transactions. The timing diagram in Figure 4 shows a series of page read and page write commands issued by the memory controller to the SLDRAMs. For purposes of illustration, all burst lengths are 4*N*, although the controller can dynamically mix 4*N* and 8*N* bursts. The read access time to an open bank, also known as page read latency, is shown here as 12*N* (30 ns). The first two commands are page reads to SLDRAM 0 to either the same or differnt banks. SLDRAM 0 drives the read data on the data bus along with DCLK0 to provide the memory controller the necessary edges to strobe in read data. Since the first two page read commands are for the same SLDRAM, it is not necessary to insert a gap between the two 4*N* data bursts. The SLDRAM itself ensures that DCLK0 is driven continuously without glitches.

The data burst for the next page read—to SLDRAM 1—must be separated by a 2*N* gap. This allows for settling of the DataLink bus and for timing uncertainty between SLDRAM 0 and SLDRAM 1. A 2*N* gap is necessary any time control of the DataLink passes from one device to another. This occurs in reads to different SLDRAMs and in read-to-write and write-to-read transitions between SLDRAMs and the memory controller. The memory controller creates the gap between data by inserting a 2*N* gap between commands. SLDRAM 1 begins driving the DCLK lines well in advance of the actual data burst.

The next command is a write command in which the controller drives DCLK0 to strobe write data into SLDRAM 2. The page write latency of the SLDRAM is programmed to equal page read latency minus 2*N*. To create a 2*N* gap between Read1 and Write2 data on the DataLink, the Write2 command must be delayed 4*N* after the Read1 command. Programming write latency in this manner creates an open 4*N* command slot on the CommandLink, which could be used for nondata commands such as row open or close, register write, or refresh, without affecting DataLink utilization. The subsequent read command to SLDRAM 3 does not require any additional delay to achieve the 2*N* gap on the DataLink. The final burst of three consecutive write commands shows that the 2*N* gap between data bursts is not necessary when the system is writing to different SLDRAM devices. This is because all write data originates from the memory controller.

Data clocks. When control of the DataLink passes from one device to another, the bus remains at a midpoint level for nominally 2*N*. This results in indeterminate data and possibly multiple transitions at the input buffers. This is acceptable for the data lines themselves, but not the data clocks, which strobe data. To solve this problem, the data clocks

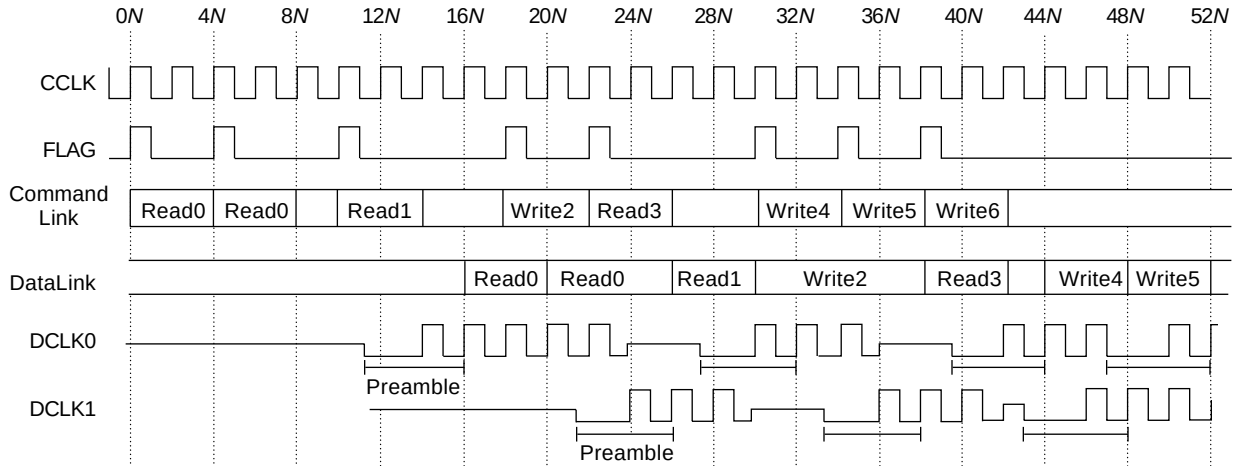


Figure 4. Timing diagram, SLDRAM bus transactions.

have a 00010 preamble before the transition associated with the first bit of data occurs. The device receiving data can enable the DCLK input buffer at any time during the first 000 period. The preamble also includes dummy transition 10 to remove pulse-width-dependent skew (also known as intersymbol interference, or ISI) from the DCLK signal. The receiving device ignores the first rising and falling edge of DCLK and begins clocking data on the second rising edge. There are two data clocks, so the device can accommodate gapless 4*N* write bursts to different SLDRAMs and 4*N* read bursts from different SLDRAMs. The controller indicates in each command packet which DCLK is to be used.

The controller transmits CCLK edges coincidentally with edges of CA[9:0] and FLAG data. DCLK edges originating from the controller are also coincident with DQ[17:0] data. The SLDRAMs add fractional delay to incoming CCLK and DCLKs to sample commands and write data at the optimum time. (We discuss this in greater detail in the following section and in the section on synchronization.) The controller can program the SLDRAMs to add fractional delay to outgoing DCLKs during read operations. This allows the controller read data input registers to directly strobe in read data using the received DCLK without internal delay adjustments.

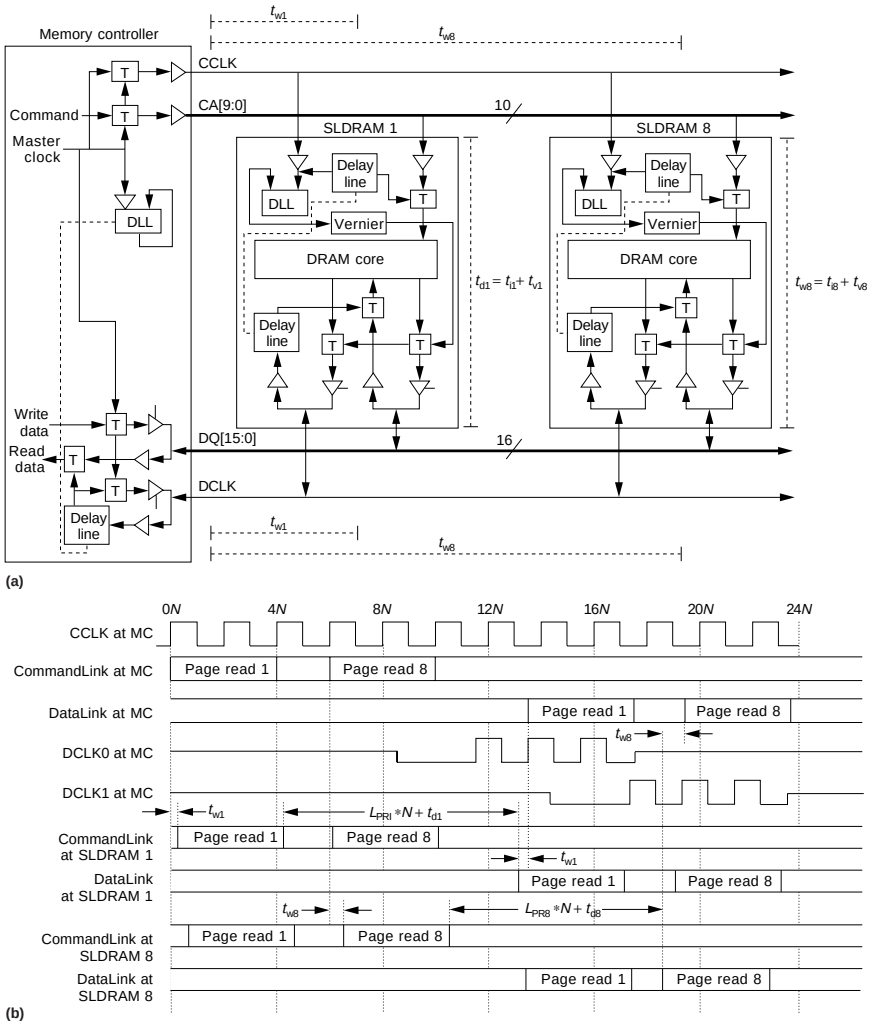


Figure 5. SLDRAM bus timing adjustment system block diagram (a); waveforms (b).

Timing adjustment. The controller programs each SLDRAM with four timing latency parameters: page read, page write, bank read, and bank write. We define latency as the time between the command burst and the start of the associated data burst. For consistent operation of the memory subsystem, each SLDRAM should be programmed with the same values. On power-up, the memory controller polls the status registers in each SLDRAM to determine minimum latencies, which may vary by manufacturer and speed grade. The controller then programs each SLDRAM with the worst-case values.

Read latency is adjustable in coarse increments of unit bit intervals and fine increments of fractional bit intervals. The controller programs the coarse and fine read latency of each SLDRAM so that read data bursts from different devices, at different electrical distances from the controller, all arrive back at the controller with equal delay from the command packet. Write latency is only adjustable in coarse increments. The write latency values determine when the SLDRAM begins looking for transitions on DCLK to strobe in write data. Since this can occur any time during the 000 portion of the DCLK preamble, it does not require fine adjustment.

Figure 5a shows a conceptual block diagram of the SLDRAM system with major timing components. A DLL (delay-locked loop) in both the controller and SLDRAM locks to the free-running clock to provide a stable reference for the input sampling delay lines. Input and output latches acting on both edges of the clock are labeled T.

Figure 5b shows the timing of two consecutive page read commands, the first addressed to the SLDRAM nearest the controller and the second to the SLDRAM at the far end of the bus. Total latency observed by the controller on the first read includes the programmed page read latency (L_{PR1}), the clock-to-data propagation delay through the SLDRAM itself (t_{d1}), and wire delay to the SLDRAM and back ($2t_{w1}$). The clock-to-data propagation delay is composed of intrinsic delay (t_{i1}) and the programmed fine vernier delay (t_{v1}). When an increment vernier command issued by the controller causes the fine read vernier to overflow past 1*N* delay, digital latency L_{PR} is incremented by one and fine vernier setting t_v is reduced to 0. If a decrement vernier command causes the

Table 2. SLDRAM command format.

Flag	CA9	CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
1	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	CMD5
0	CMD4	CMD3	CMD2	CMD1	CMD0	Bank 2	Bank 1	Bank 0	Row 9	Row 8
0	Row 7	Row 6	Row 5	Row 4	Row 3	Row 2	Row 1	Row 0	0	0
0	0	0	0	Col 6	Col 5	Col 4	Col 3	Col 2	Col 1	Col 0

fine read vernier to underflow below 0 delay, digital latency L_{PR} is decremented by one and fine vernier setting t_v is set to 1*N*.

The wire delay to the SLDRAM at the far end of the bus is significantly greater, and the intrinsic delays of the two devices may differ considerably. However, we can equalize the timing seen by the controller by enforcing the following relationship between the verniers on each device:

$$t_{v8} = t_{v1} - t_{i8} + t_{i1} - N(L_{PR8} - L_{PR1}) - 2(t_{w8} - t_{w1})$$

In this way the separation between bursts originating from different devices can be controlled, minimizing bus turn around delay.

SLDRAM command format. Each SLDRAM command packet contains four 10-bit words. The command packet shown in Table 2 is for a 64-Mbit SLDRAM with eight banks, 1,024 row addresses, and 128 column addresses. The same 40 bits can accommodate many other organizations and densities. On power-up, the memory controller polls the SLDRAMs to determine how many banks, rows, and columns each device has. The controller can then include the appropriate number of address bits in the command packet for each SLDRAM individually.

Chip ID and multicasting. The first word of the command packet contains the chip ID bits. An SLDRAM ignores any command that does not match the local ID. On power-up, the controller uses the SI/SO signals to assign chip IDs. (We describe this in greater detail in the section on initialization.) This allows the controller to uniquely address every SLDRAM in the system without separate chip-enable signals or glue logic. The chip ID consists of nine bits, allowing up to 256 SLDRAMs on a single hierarchical DataLink. Multicasting allows the controller to address any group of two (or four, eight, sixteen, and so on) SLDRAMs with a single command (see Table 3, next page). For example, chip ID 100000001 addresses devices 0, 1, 2, and 3. This feature is useful for initialization, refresh, and multiple DataLink configurations.

SLDRAM commands. The command field consists of six bits, as shown in Table 4. When most significant bit CMD5 is 0, the device executes normal read or write commands. The selection of page or bank access, burst length, read or write, autoprecharge, and DCLK are independent. Bank access commands provide bank, row, and column addresses to the SLDRAM, which then schedules internal row open and page access activity. Page access requires the row to be previously opened. Setting CMD5 to 1 selects row operations, register accesses, events, or special synchronization commands.

For register operations, the bank, row, and column addresses are not required. These are replaced in the command packet by register address and register write data. Since register writes do not use the DataLink, they can occur before write synchronization has completed. Register read data appears on the DataLink with DCLK exactly like a normal read.

An event includes commands such as autorefresh, self-refresh, reset, and close all rows. Event codes are located in the unused address fields of the command packet. SLDRAM also implements as events calibration commands such as read data fine vernier adjustment, read data DCLK offset adjustment, and Voh/Vol output-level adjustment. These event commands, along with the special synchronization commands, allow the memory controller to initialize and maintain the very tight signal timing and voltage level requirements of an SLDRAM memory subsystem.

SLDRAM calibration. System-level calibration of individual SLDRAM timing and output drive levels is key to the components' high manufacturing yields and low cost. Individual devices are not required to meet tight ac and dc parametric specifications. Rather, these will be calibrated at the system level to compensate for wide variation in individual device parameters. Calibrated parameters include Voh and Vol levels, input setup and hold time, and read and write data latency.

Synchronization. In normal operation the controller drives CA[9:0], FLAG, and CCLK with simultaneous transitions. The SLDRAM must internally delay CCLK approximately *N*/2 relative to CA[9:0] and FLAG to properly latch the command packet. There are many sources of timing error in the system, which we can compensate for with appropriate adjustment of the input sampling clock. These include static errors such as variations in bus impedance, loading, or track length, and dynamic errors such as cross talk, power supply noise, and intersymbol interference (ISI). We can compensate for static errors with a one-time adjustment of the sampling clock. Cross talk and power supply noise symmetrically advance or delay an edge depending on the coupling, so they cannot be ameliorated using a static timing adjustment. We must instead minimize these effects through design. ISI results in a significant asymmetric skew between clock and data; we can therefore improve it by appropriate adjustment of the internal sampling delay.

As bus speeds approach the limit of the transmission media, ISI becomes a significant fraction of a bit interval. When the device transmits a signal with the highest edge rate—01010101—individual 0 and 1 levels will not reach their final steady-state values owing to the filtering effect of the limited bandwidth channel. The subsequent transition will

Table 3. Multicasting definition.

Chip ID8 to ID0	Destination device(s)
000000000	0
000000001	1
⋮	⋮
011111111	255
100000000	0 through 1
100000001	0 through 3
100000010	2 through 3
100000011	0 through 7
100000100	4 through 5
100000101	4 through 7
⋮	⋮
100000111	0 through 15
⋮	⋮
100001111	0 through 31
⋮	⋮
101111111	0 through 255
⋮	⋮

Table 4. SLDRAM command format.

CMD5 to CMD0	Command
0 0 x* x x x	Page access
0 1 x x x x	Bank access
0 x 0 x x x	Burst length = 4
0 x 1 x x x	Burst length = 8
0 x x 0 x x	Read
0 x x 1 x x	Write
0 x x x 0 x	Leave page open
0 x x x 1 x	Autoprecharge
0 x x x x 0	Use DCLK0
0 x x x x 1	Use DCLK1
1 0 0 0 0 1	Open row
1 0 0 0 1 0	Close row
1 0 0 0 1 1	Register write
1 0 0 1 0 x	Register read
1 0 0 1 1 1	Event
1 0 1 0 0 0	Read sync
1 0 1 0 1 0	Drive DCLKs = 0
1 0 0 0 1 1	Drive DCLKs =1
1 0 1 1 0 0	Write sync
1 0 1 1 1 0	Disable DCLKs
1 0 1 1 1 1	Toggle DCLKs
1 1 1 1 1 1	Command/write sync

* x indicates don't care.

have a lower amplitude starting point and will therefore cross the midpoint reference sooner than a 00110011 data pattern. Since CCLK has every possible transition, but data can have long periods with no transitions, data transitions range from no delay to significant delay with respect to clock transitions. Bus reflections complicate this simple explanation of ISI.

To properly center the input sampling clock in the received data eye during synchronization, the controller transmits inverted and noninverted versions of a 15-bit repeating pseudorandom SYNC sequence—111101011001000. This pattern creates every possible 4-bit data sequence with the exception of 0000, allowing for long impulse responses. An SLDRAM recognizes this sequence by two consecutive 1s on the FLAG bit. The SLDRAM then determines the optimum internal delay for CCLK to sample incoming bits of the known SYNC pattern at the center of the received eye.

Power-up initialization. When the SLDRAM subsystem is powered up, the controller must take the following steps before normal memory operations can begin:

1. Power up—The controller applies V_{cc}, V_{ref}, and V_{ccq}. It follows later with V_{term}, the 1.25-V CommandLink and DataLink termination supply, to avoid latch up.
2. Reset—The controller holds the RESET* pin on each SLDRAM low. This clears the SLDRAM's internal synchronization indication, programs the internal device ID to 255, and sets all read and write latency registers to their minimum values.
3. Synchronization—The controller begins transmitting CCLK, drives both DCLKs with continuous transitions, and sets SO to 1. On DQ[17:0], CA[9:0], and FLAG, the controller repeatedly transmits the 15-bit SYNC sequence. Before the SLDRAM has synchronized, it sets SO to 0. Once the SLDRAM has synchronized and has programmed the appropriate delays for CCLK, DCLK0, and DCLK1, it sets SO equal to SI. The controller stops sending the SYNC pattern when SI is 1. It then resets SO to 0, which ripples through all SLDRAMs.
4. ID assignment—The controller sets SO to 1 once again and sends an ID register write command with write data 0. Only the SLDRAM with SI set to 1 and ID set to 255 responds to this command. This SLDRAM overwrites its ID register with the value 0 and sets SO to 1. The controller repeats the ID register write with write data 1, and so on, until it observes a high logic level on SI.
5. Voh, Vol calibration—To calibrate each SLDRAM's I/O levels, the controller sends each device drive DCLKs = 0 or drive DCLKs = 1 commands. It then issues increment/decrement Voh/Vol event commands until the levels match the controller's own reference.
6. Read synchronization—The controller issues a read sync command to each SLDRAM, which then transmits continuous transitions on both DCLKs and the 15-bit SYNC pattern on DQ[17:0]. The controller then adjusts the delay of DCLK edges relative to DQ edges, using the increment/decrement DCLK offset event command, to sample the synchronization pattern at the optimum point.
7. Read latency calibration—After RESET, the controller sets SLDRAM read and write latencies to their minimum values. For each SLDRAM, the controller issues a drive DCLKs = 0 command followed by a read command. To measure latency, the controller monitors the specified DCLK for the transitions that accompany the data burst. Once the controller has measured minimum latencies for each SLDRAM, it can program the worst-case read latency value

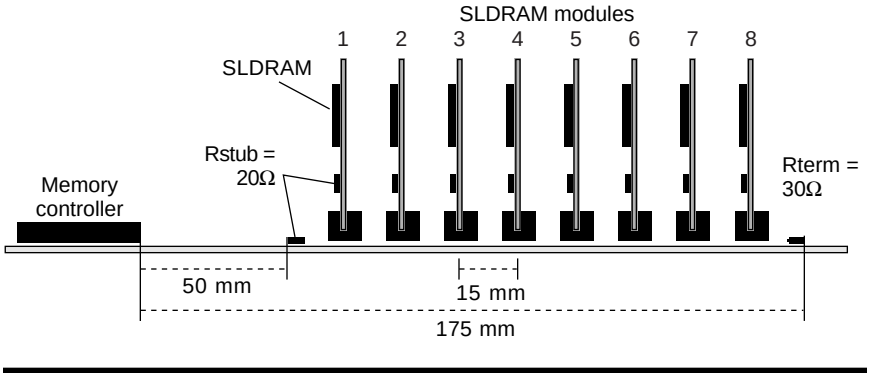


Figure 6. 400-Mbps/pin SLDRAM motherboard layout cross section. Not to scale.

into each device by writing to the read delay registers.

8. Write latency programming—The controller issues each SLDRAM a write command with zero write latency and an extended 32*N* burst of data. Data words in the write burst are incremented 0, 1, 2, 3, ..., 31. After the write command, the controller reads the location in the SLDRAM. The first word contains the minimum write latency. With this information, the controller can set the write latencies of all SLDRAMs to a common value by writing to the write delay registers.

The memory system is now fully calibrated and ready for normal operation.

Recalibration during operation. During normal operation, calibrated timing and voltage-level parameters may drift owing to changes in temperature and supply voltage. To ensure robust operation, the controller must periodically recalibrate the SLDRAMs. It hides recalibration operations during autorefresh and self-refresh periods so as not to affect system performance.

Physical design

In this section, we describe the physical design of a typical PC memory subsystem employing SLDRAM, including wire length and position of components and modules. We also discuss buffered modules supporting greater memory capacity and higher memory bandwidth for high-end server applications.

Board layout. The 400-Mbps/pin SLDRAM subsystem supports eight loads, as shown in Figure 6. The motherboard tracks are single-end terminated to a midpoint reference at the far end with 30Ω resistors, Rterm. Using conventional PCB material with 6-mil tracks and 20-mil pitch, the characteristic impedance of the wire falls between 60 and 65Ω. However, the effect of the closely spaced stubs lowers the effective impedance to half that of a single track. Series stub resistors Rstub of 20Ω isolate the point load of the SLDRAM chip, representing roughly 3 pF, from the bus. The controller is also isolated by series stub resistors, although these may be placed near the first module to ease wiring congestion near the controller.

Modules are 15 mm apart, and there is a 50-mm lead-in between the memory controller and the first module. Stub length through the connector and over the module to the SLDRAM chip is roughly 15 mm, although most of this length is on the other side of the series stub resistor. Overall, allowing for termination and series stub resistors, the bus is 175 mm end to end. To minimize the effects of cross talk, we recommend a shielded board layout with alternating signals and grounds.

Signaling. SLDRAM uses SLIO, a subset of SSTL_2 (series stub terminated logic for 2.x-V generation), the JEDEC standard for high-speed signaling. Output drive levels are more tightly specified than SSTL_2 to achieve faster bus settling and improved noise margin. The controller and SLDRAM devices employ a 2.5-V V_{DDQ} supply to power their output drivers. CommandLink and DataLink signals are single-end terminated to a midpoint reference level of 1.25 V at the far end of the bus. Single-end termination saves power over double-ended termination. The memory controller calibrates output drive levels on power-up to achieve 0.9-V SLIO and 1.6-V drive levels on the main bus.

Buffered modules. The SLDRAM protocol allows the addressing of hierarchical memory subsystems that have more devices than a single electrical interface can support. Systems can mix buffered and unbuffered modules. The initialization and synchronization sequence accommodates the additional delays added by the buffers.

Buffered modules permit deeper and/or wider memory configurations than the basic unbuffered configuration discussed earlier. An unbuffered configuration with eight loads can support a memory size of 64 Mbytes using first-generation 64-Mbit, 400-Mbps/pin devices, thus achieving 800-Mbytes/s bandwidth. By the time SLDRAM achieves significant market penetration with 256-Mbit, 800-Mbps/pin devices, an unbuffered configuration will support a memory size of 256 Mbytes and bandwidth of 1.6 Gbytes/s. This meets the requirements of desktop PC applications but not those of high-end workstations and servers.

Buffered CommandLink. A wide module with multiple DataLinks providing higher memory bandwidth, based on a buffered CommandLink, is also possible (see Figure 7). CommandLink overhead—including pins on the controller, motherboard tracks and termination networks, bus termination power dissipation, and module connectors—does not have to be duplicated for each DataLink. The CommandLink buffer adds approximately 5 ns of delay to SLDRAM read access time.

Each CommandLink buffer can drive eight individual SLDRAM devices on a module, supporting an aggregate DataLink width up to 128 bits. At 800-Mbps/pin, the wide module approach can deliver data bandwidth as high as 12.8 Gbytes/s with a total memory capacity of 2 Gbytes using 256-Mbit devices.

Buffered CommandLink and DataLink. This configuration lets a controller with a single SLDRAM interface address a larger memory depth than eight individual

unbuffered SLDRAM devices would allow. Although shown as separate devices in Figure 8, the CommandLink and DataLink buffers may be combined in a single chip.

Each CommandLink and DataLink buffer can drive eight individual SLDRAM devices on a module, supporting a memory system with 64 chips. With SLDRAM devices from the 256-Mbit generation, the system can address a total memory of 2 Gbytes. By splitting the single DataLink into several parallel DataLinks, we can increase bandwidth from 1.6 Gbytes/s to 3.2 Gbytes/s, 6.4 Gbytes/s, and so on, to meet application needs. The SLDRAM protocol will support further hierarchical levels of buffering, such as an additional CommandLink buffer, to increase total memory capacity to 16 Gbytes and beyond.

Availability. SLDRAM is an open-standard, multisourced, commodity device that will be manufactured by the members of the SLDRAM Consortium. Virtually all DRAM manufacturers are members of the Consortium. A license to manufacture SLDRAM is available to any company by joining the Consortium. The first SLDRAM devices will be available for sampling in the first half of 1998. These initial samples will be 64-Mbit devices offering 400 Mbps/pin on 16-bit I/O. Volume production of 64-Mbit parts will be available with a 600 Mbps/pin interface, followed by fully backward-compatible, 128-Mbit and 256-Mbit, 800-Mbps/pin devices.

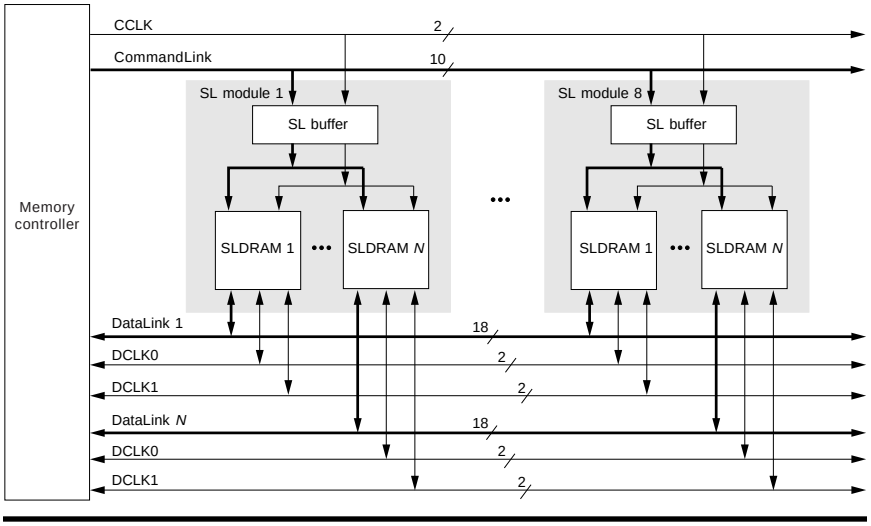


Figure 7. Wide configuration using buffered CommandLink modules.

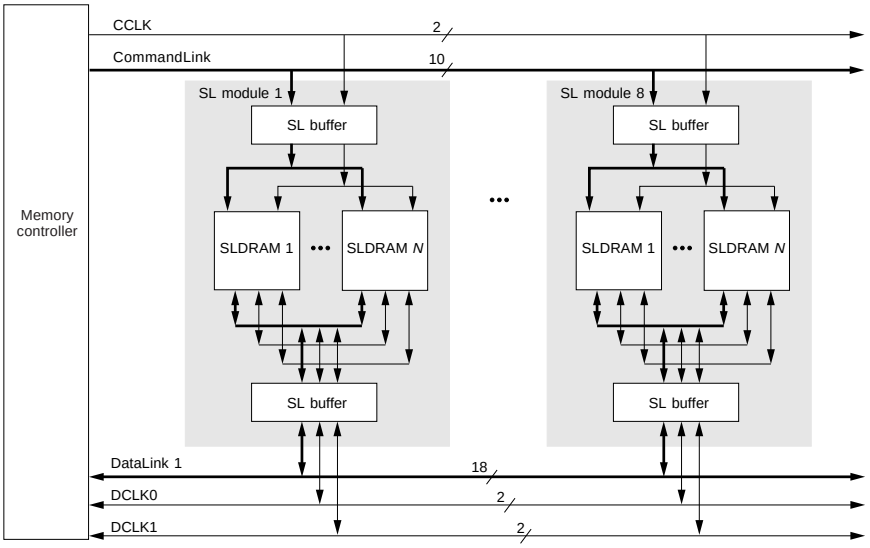


Figure 8. Deep configuration using buffered CommandLink and DataLink modules.

Internal architecture

Figure 9 (next page) shows a functional block diagram of a representative, 72-Mbit, 400-Mbps/pin SLDRAM. At the heart of the chip lies a conventional eight-bank DRAM array. A bidirectional data path shared among the eight banks provides 72 bits of read or write data at the 100-MHz core frequency. Most of the chip operates on a time base derived from CCLK. Commands/address packets are captured by ICLK, which the controller adjusts during synchronization to sample inputs in the center of the eye. The SLDRAM compares the chip ID field of incoming commands with its local ID register. If there is a match, the SLDRAM passes data or bank commands on to the address sequencer and immediately executes control and event commands. The address sequencer delays execution of commands depending on the programmed latency values, to implement just-in-time bank activation and data transfer. A new command can be queued in the address sequencer every 10 ns, but may execute in any order depending on command type and latency. The internal banks are not controlled directly from external pins and can therefore be highly pipelined.

A delayed clock generated by the fine vernier drives the final read data output latch to correctly position the read data burst. DCLK0 and DCLK1 are driven from latches controlled from the same delayed clock, so that DCLK and read data transitions can occur simultaneously. A read FIFO passes read data from the ICLK-driven core DRAM time domain to the delayed clock read data output domain.

During a write, a delayed version of DCLK0 or DLCK1 latches data into the chip. The controller adjusts the delay during synchronization to sample write data inputs in the

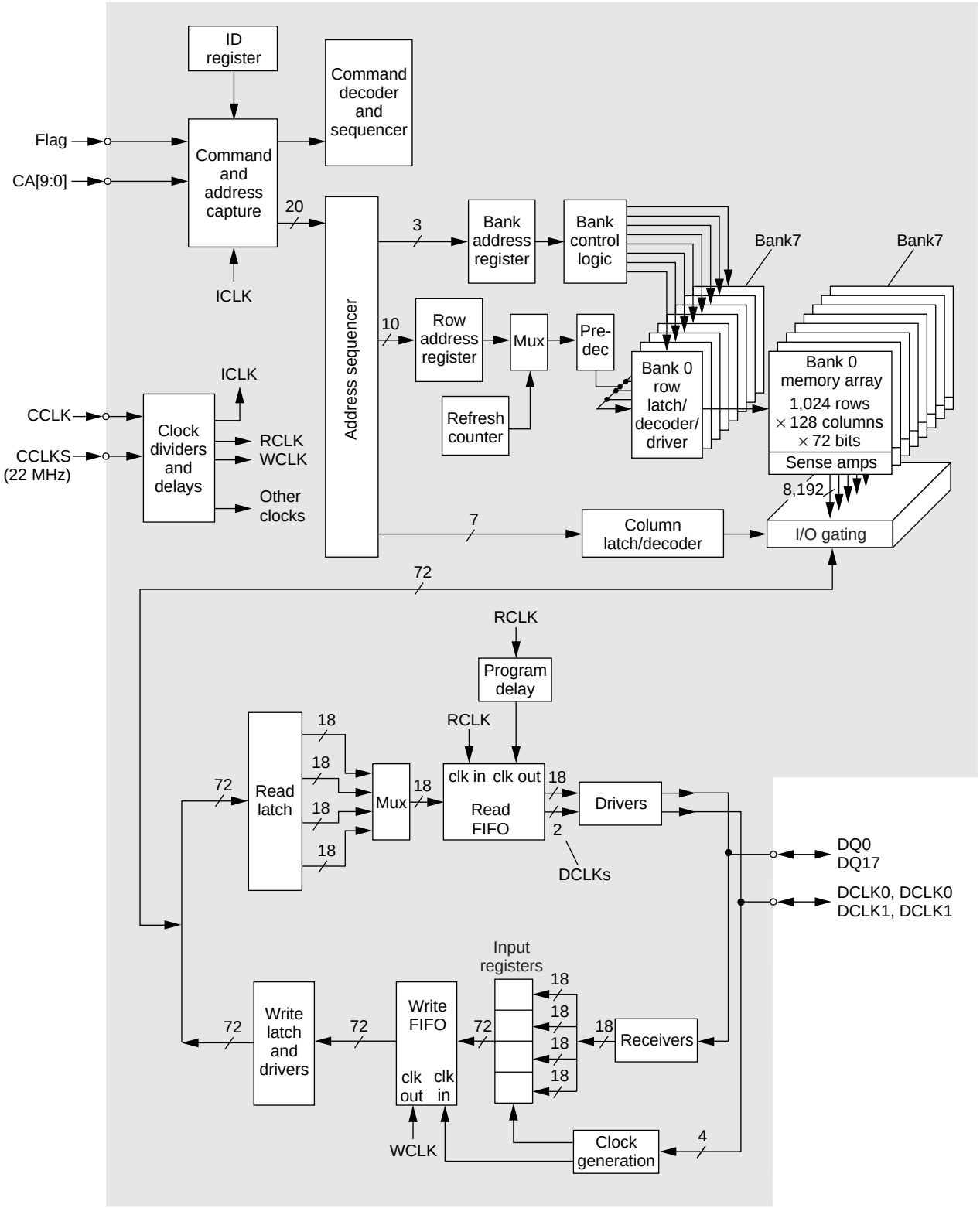


Figure 9. SLDRAM functional block diagram.

center of the eye, in the same way that ICLK is generated from the received CCLK. Write data inputs are enabled during the DCLK preamble to eliminate the possibility of clock glitches on the input latches. A write FIFO passes write data from the delayed DCLK input capture time domain to the ICLK-driven core DRAM time domain.

SLDRAM REPRESENTS THE LATEST STAGE in DRAM interface evolution, meeting the bandwidth requirements of next-generation processors. SLDRAM offers a cost-effective solution for a wide range of applications, from large servers to palmtops, desktop PCs to custom telecommunications and video graphics processors. Most importantly, SLDRAM is an open standard that can be adapted for specialty niche applications and will evolve to meet changing system requirements. To keep abreast of the progress of the SLDRAM Consortium, and to obtain the technical documentation required for your system implementation, look us up on the Web at www.sldram.com.

Acknowledgments

SyncLink is a trademark of MicroGate Corporation, and is not related to SLDRAM.

Suggested readings

Foss, R.C., et al., "Fast Interfaces for DRAM," *IEEE Spectrum,* Oct. 1992, pp. 54-57.

IEEE Std. 1596—Scalable Coherent Interface (SCI), IEEE, Piscataway, N.J., 1992.

IEEE Std. 1596.4—High-Bandwidth Memory Interface Based on Scalable Coherent Interface (SCI) Signaling Technology (RamLink), IEEE, 1996.

IEEE Proposed Std. P1596.7—High-Speed Memory Interface (SyncLink), IEEE, draft available on SLDRAM Web site, www.sldram.com.

"MOSAID to Design Next Generation Memory Technology," MOSAID, Ottawa, Ont., Canada and Santa Clara, California, http://www.mosaid.com/, 1997.

Prince, B., *High Performance Memories: New Architecture DRAMs and SRAMs, Evolution and Function,* John Wiley & Sons, New York, 1996.

Prince, B., *Semiconductor Memories: A Handbook of Design, Manufacture and Application,* 2nd ed., John Wiley & Sons, 1996.

Przybylski, S., *New DRAM Technologies,* MicroDesign Resources, Sebastopol, Calif., 1994.

"SLDRAM Datasheet," SLDRAM Consortium, www.sldram.com, 1997.

Vogley, B., "SyncLink: High-Speed DRAM for the Future," *IEEE Micro,* Vol. 16, No. 6, Dec. 1996, pp. 74-75.

Peter Gillingham is vice president of strategic and technical marketing for MOSAID Technologies Inc. His earlier positions at MOSAID include DRAM Project Manager—leading the development of advanced 1-Mbit, 4-Mbit, and 16-Mbit DRAMs—and director of strategic marketing. Before joining MOSAID, he worked for Mitel Corporation, where he was involved in the design of mixed analog-digital products for telecommunication applications. Prior to that, he was a research assistant in the area of switched capacitor circuits at L'École Polytechnique Fédérale in Lausanne, Switzerland.

A member of the IEEE, Gillingham serves on the Memory Subcommittee of the ISSCC Technical Program Committee. He is an associate editor of the *IEEE Journal of Solid-State Circuits,* and he served as guest editor for that journal's November 1995 and November 1996 special issues on logic and memory.

Bill Vogley, a systems architect for Texas Instruments, chairs the IEEE P1596.7 working group for the SLDRAM memory interface. He has been working on high-speed interfaces for over 15 years—with supercomputers beginning at Control Data, and advancing through ETA System with the ETA 10. Vogley's many credits include his work on synchronous DRAM. He is on the board of the SLDRAM Consortium.

Address questions or comments about this article to Peter Gillingham, VP Strategic and Technical Marketing, MOSAID Technologies Inc., PO Box 13579, Kanata, Ontario, Canada K2K 2K9; gillingham@mosaid.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Interest Card.

Low 159 Medium 160 High 161