CONSTANTLY INCREASING SYSTEM-PERFORMANCE EXPECTA-TIONS ARE OUTSTRIPPING THE CAPABILITIES OF TRADITION-AL SYNCHRONOUS INTERFACES. SOURCE-SYNCHRONOUS-INTERFACE ALTERNATIVES PROVIDE ADDITIONAL BANDWIDTH HEAD ROOM BUT REQUIRE MORE CAREFUL DESIGN ANALYSIS AND IMPLEMENTATION.

DDR-SDRAM, high-speed, source-synchronous interfaces create design challenges

TRADITIONAL INTERFACES LIMIT interconnect speed to less than 250 MHz and pc-board-interconnect length to approximately 5 in. Designers are increasingly turning to source-synchronous interconnects that demonstrate transfer rates of 1 billion transitions/sec at distances of 5m and greater.

Several examples of source-synchronous technology exist. Their implementations affect design complexity and overall performance. Within memory subsystems, major examples include double-datarate (DDR) SRAM, DDR synchronous DRAM (SDRAM), synchronous-graphics RAM, and Direct Rambus DRAM.

For networking and I/O, examples include the scalable coherent interface (SCI), Silicon Graphics' (www.sgi.com) CrayLink, and High Performance Parallel Interface (HIPPI)-6400-PH.

The following case study reviews a practical source-synchronous link technology with respect to both the signal integrity of the interconnect and the required interface logic. The case study also compares the design trade-offs between synchronous and source-synchronous interconnects at the ASICand wire-implementation levels.

SOURCE-SYNCHRONOUS INTERFACES

With source-synchronous interfaces, data and clock transport from a transmitter to a receiver, and the receiver interface uses the clock to latch the accompanying data. Source-synchronous interfaces offer many advantages over traditional synchronous interfaces. Both time of flight between chips and system-level clock skew limit bus-clock frequencies between chips for standard synchronous interfaces. Meanwhile, the achievable pin count of IC packag-



Double-data-rate interfaces transition data on both clock edges.

ing has not dramatically increased over time.

Source-synchronous interfaces remove the time-of-flight limit on interconnection between ICs and require no controlled clock skew (Table 1). Another advantage of source-synchronous interfaces is dramatically increased I/O frequencies. With increased bandwidth per I/O driver, the number of pins per interface better matches the capabilities of available IC-packaging technologies. I/O-driver frequencies can reach five to 10 times the core-logic frequency.

However, source-synchronous interfaces create new design-analysis challenges. Interface latency is not necessarily predictable; if your design requires predictable latency, overall interface latency increases. Increases in I/O speeds require more robust IC-package electrical performance. Because the I/O frequency can be much higher than that of the core logic, I/O-interface-logic com-



address/control and data.

The DDR-SDRAM interface contains separate clocks for



Delaying the data clock creates necessary setup-and-hold times at the receiver.

TABLE 1-COMPARISON OF STANDARD AND SOURCE-SYNCHRONOUS INTERFACES

Synchronous interface	Source-synchronous interface
Limits the time of flight between two ICs	Has no time-of-flight limit between two ICs
to a clock period	
Requires clock-skew control	Requires no clock-skew control between ICs
Presents no interface-synchronization	Presents an interface-synchronization challenge
challenge for multiple-RAM interface	for interface with two or more RAMs
Increases pin count for interface to increase	e Increases frequency to increase total bandwidt
total bandwidth for interface	for interface

plexity must grow to handle the frequency multiplication. Data bit-to-bit timing skews and "eye patterns" define overall link-operation frequencies, whereas you may have previously ignored these effects.

IMPLEMENTING INTERFACES

DDR interfaces transmit data on both edges of the clock, or "strobe." These types of interfaces offer a straightforward way to increase the bandwidth to various

> memory subsystems, such as levels 2 and 3 cache, main memory, and frame-buffer memory, and build on the foundation of the previous-generation single-data-rate interface. The trade-off, however, is often a more complex interface-agent RAM port, and latency prediction becomes more difficult because of the asynchronous nature of the data reception (Figure 1).

> The current standard DDR SDRAM includes

both an address/control interface and a data interface (Figure 2). Data transfers for reads and writes on both edges of a DQS (data-I/O) bidirectional strobe. Address and control signals transmit at half the data frequency and latch on only the rising edge of the transmit clock. Several design issues complicate the analysis of this interface. Any timing skews or uncertainties, such as pulse-width distortion and jitter on CLK and DQS, cause data- and address-timing problems at the SDRAM input and at the memory-agent IC's synchronization flip-flop. DQS' bidirectional and random nature further worsens its jitter component. In contrast, the CLK signal is unidirectional and of constant frequency.

For this interface, data and DQS synchronously and inphase exit the SDRAM (Figure 3). You must delay DQS to create data setup-and-hold time at the synchronization flip-flop. Possible delay techniques include using a digital-delaylocked loop (DLL) or PLL within the interface agent or using a pc-board etchdelay line. All of these techniques work, but none is flexible; once you implement these techniques, they lock the interface into an operating-frequency range. In addition, the DLL or PLL may be boardspace-prohibitive for designs requiring multiple SDRAMs. Each SDRAM would require two DLLs or PLLs on the interface-agent IC (Figure 4).

Target data rates for DDR SDRAMs are 250 Mbps and greater, translating to clock frequencies in excess of 125 MHz. At these speeds, poorly terminated or unterminated lines exhibit signal-integrity effects that increase settling time. Lines that approach the tuned resonant or quarter- and half-wavelengths of the clock frequencies are key factors in settling-time jitter for poorly terminated lines. For a 125-MHz DDR SDRAM, the tuned-resonant lengths in FR4 stripline etch for the 250-Mbps data line are 5.71 and 11.43 in., not accounting for package delays. At these lengths, driver and receiver reflections superimpose on rising and falling edges of the next data bits, changing the measured rising- and falling-edge settling times (**Figure 5**).

Another example of settling-time jitter is a signal that does not stabilize to V_{OH} (output high voltage) or V_{OL} (output low voltage) before the next transition occurs. Such effects are eye patterns, or "intersymbol interference" (Figure 6a). As line length and topology become more complex, network termination becomes crucial in limiting jitter and its effects. What's the "eye"? As an example, a 200-MHz data bus has a maximum data-toggle rate of 1 bit every 5 nsec. Take a look at the voltage in the time domain at the receiver input, and you can see rising and falling edges with highs and lows.

Now, take 10-nsec slices of the time domain, and take those 5-nsec partitions, and pile them up like a deck of cards. The edges cross, and the ends are the dc high and low voltages. The area where no signal trace exists between the rising and falling edges and the highest low and lowest high is the eye. If you place the clock edge so that it rises in the middle of the eye, you can latch settled data, assuming that the rising/falling edge before the clock meets setup time and that the following edge meets hold time. Terminated lines increase the eye size, thereby increasing setup-and-hold time, allowing your interface to run more reliably and enabling you to increase its speed (Figure 6b).

DDR-SDRAM-DESIGN ANALYSIS

Interface-design analysis consists of signal quality, interface timing, and interface synchronization. Signal-line topology, pc-board routing and construction, and IC-package electrical parasitics all influence signal quality. Using a pseudorandom pattern sequence, you can characterize overshoot, eye-pattern jitter, and eye-pattern closure for a given signal topology (**Figure 7**).

You can determine appropriate line termination by examining operationalfrequency targets. The DDR-SDRAM interface does not lend itself to parallel



DDR data-write (a) and -read (b) waveforms are identical, although the receiver and transmitter are different in each case.

TABLE 2-DATA-WRITE TIMINGS

Data-write interface-transmitter timing			
Skew component	Setup time (psec)	Hold time (psec)	
Data/strobe clock jitter (t _{cut})	50	50	
Clock skew (t _{cskew})	50	50	
Buffer-delay mismatch (t _{BDM})	25	25	
Buffer pulse-width distortion (t _{RPWD})	150	150	
Data source-synchronous-output (SSO) jitter (t _{psso}) 300		
Strobe SSO jitter (t _{ssso})		300	
Total timing uncertainty	575	575	
Data-write interconnect timing			
Skew component	Setup time (psec)	Hold time (psec)	
PC-board-trace skew (t _{pcsk})	25	25	
Package-trace skew (t _{PKSK})	25	25	
Data eye-pattern jitter (t _{DEYE})	0	500	
Strobe eye-pattern jitter (t _{seve})	50	0	
Strobe-placement error (t _{STP})	125	125	
Total timing uncertainty	225	675	
Data-write receiver timing			
Skew component	Setup time (psec)	Hold time (psec)	
DDR-SDRAM data setup (t _{DDRS})	800	0	
DDR-SDRAM data hold (t _{DDRH})	0	400	
Total timing uncertainty	800	400	
Data-write end-to-end timing			
Skew component	Setup time (psec)	Hold time (psec)	
Transmitter-timing uncertainty	575	575	
Interconnect-timing uncertainty	225	675	
Receiver-timing uncertainty	800	400	
Timing margin at 125 MHz	400	350	

data-bus termination because it is bidirectional. Series termination, ideally within the driver to eliminate separate passive components on the pc board, is a more appropriate scheme (**Figure 8**). However, the tolerance of the series output resistance limits the effectiveness of series termination within the driver. Typical process limitations are $\pm 22\%$, a wider tolerance than the process variation on discrete resistors. As

operational speed increases in the future to more than 500 Mbps per I/O buffer, series-resistor tolerance will become a strong definer of eye-pattern jitter and closure.

Three main paths require analysis for



Proper line termination minimizes signal-integrity effects that increase settling time.

the interface, and each of these paths further breaks down into three sections. Each timing path contains transmitter-, interconnect-, and receiver-timing components. Transmitter timing consists of all possible components of timing jitter



Unlike misterminated or unterminated alternatives (a), correctly terminated lines increase the setup-and-hold time (b).

and skews within the transmitting IC that would subtract from either setup or hold at the synchronizing latch within the receiving IC. Interconnect timing comprises all jitter and skew components of the signal trace, and receiver timing comprehends these same components within the receiving IC itself.

The goal of timing analysis is to achieve non-negative setup-and-hold margins using a summation of all worstcase effects. If robust system-level error detection and correction allow for an occasional bit error, you can employ statistical timing analysis. For DDR-SDRAM timing, pay attention to the data-write, data-read, and address signal paths. Robust data timing is typically the hardest to achieve because of the dual-edge latching and high-speed nature of these signals. Good driver design and proper signal topology often solve challenging multiload-address-bus timing problems.

The transmitter-timing parameters for the memory-controller ASIC in the following design example come from a design that TriCN Associates did with Nvidia (www.nvidia.com), modified to guardband results. DDR-SDRAM data comes from multiple DRAM vendors' specifications and Spice models; **tables 2** through **4** report the worst-case results. Interconnect-timing parameters are the results of worst-case analysis of all timing paths that use multiple SDRAM vendors and one memory-controller ASIC as a baseline.

The results consolidate into a worstcase analysis of timing for setup-andhold data with both the ASIC and the SDRAM driving the interface. Using faster SDRAMs results in interfaces with improved timing margin, but this analysis demonstrates that any SDRAM vendor can provide a DDR interface that meets the operational-frequency target. All setup-and-hold-timing data in **tables 2** through **4** comes from extracted pcboard layouts that were then simulated in Spice using 3 sigma error margin.

DATA-WRITE TIMING

Write timing includes interface-agent output-drive timing, interconnect timing, and DDR-SDRAM input-receive timing (Table 2). The interface agent must minimize the overall skew and jitter between the data bits (DQ) and the strobe. Skew components come from CLK-to-data and t_{PD} delay (propagation-delay) differences in the flip-flops, boundary-scan components, and output driver. Jitter can come from the PLL or oscillator, as well as from ac fluctuations on power supplies due to core and output switching events.

The interconnect-timing components originate with trace-length and dielectric-constant differences between data lines in the pc board and package. If you use a delay line to push out the strobe, strobe-centering errors occur due to dielectric-constant variations over all manufacturing-tolerance ranges. The final component of timing error for the interconnect is eye-pattern jitter on both the data and the strobe. This error arises from signal-integrity variations for random pattern sequences on either terminated or unterminated lines.

Receiver timing is DDR-SDRAM-vendor-specific. In this design example, the SDRAM places 800-psec-setup- and 400psec-hold-time requirements on the data with respect to the strobe.

DATA-READ TIMING

Read timing breaks down into interface-agent receive timing, interconnect timing, and DDR-SDRAM output-drive timing (**Table 3**). The DDR-SDRAM data-output drive skews with respect to the data strobe, and you should replace the typical output skews in this example with more exact numbers from your DRAM vendor. Interconnect-timing components are identical in cause and resolution to the data-write timings.

The interface agent must minimize the overall skew and jitter between the DQ and the strobe in the receiving block. Skew components come from t_{PD} differences in the boundary-scan components, input receiver, and strobe-routing skew. The setup-and-hold times of the latching flip-flop directly contribute to the timing budget, and you should also minimize them.

ADDRESS TIMING

Address timing, like data-write timing, includes interface-agent output-drive



Analysis of pseudorandom data patterns enables characterization of various signal-quality measures.

timing, interconnect timing, and DDR-SDRAM input-receive timing (**Table 4**). Receiver timing comes from the DDR-SDRAM vendor. This example places 2000-psec-setup- and 1000-psec-hold-time requirements on the data with respect to CLK.

conditions for silicon process, pc-board process, voltage, and temperature in this case study show that you can implement a DDR-SDRAM interface with no less than 7% of performance margin for all timing paths. As DDR-SDRAM vendors improve input and output timing specifications, this analysis shows that per-

All paths analyzed under three sigma

TABLE 3-DATA-READ TIMINGS		
Data-read transmitter timing		
Skew component	Setup time (psec)	Hold time (psec)
Data/strobe clock jitter (t _{cur})	150	150
DDR-SDRAM output skew (t _{DDRSKEW})	500	500
Total timing uncertainty	650	650
Data-read interconnect timing		
Skew component	Setup time (psec)	Hold time (psec)
PC-board-trace skew (t _{pcsk})	25	25
Package-trace skew (t _{PKSK})	25	25
Data eye-pattern jitter (t _{neve})	0	625
Strobe eye-pattern jitter (t _{seve})	75	0
Strobe-placement error (t _{stp})	125	125
Total timing uncertainty	250	800
Data-read receiver timing		
Skew component	Setup time (psec)	Hold time (psec)
Strobe/data-receiver pulse-width distortion (t _{sprpwp})	35	35
Strobe-routing skew (t _{spskew})	50	50
Flip-flop setup (t _{rs})	200	
Flip-flop hold (t _{en})		200
Total timing uncertainty	285	285
Data-read end-to-end timing		
Skew component	Setup time (psec)	Hold time (psec)
Transmitter-timing uncertainty	650	650
Interconnect-timing uncertainty	250	800
Receiver-timing uncertainty	285	285
Timing margin at 125 MHz	815	265

formance for these interfaces will rapid-ly approach 500-Mbps bandwidth. □

Authors' biographies

Hansel Collins is a consultant with TriCN Associates LLC (San Francisco). He received a master's degree in computer engineering from Boston University (Boston). Ronald Nikel, also a consultant with TriCN Associates LLC, received his master's degree in electrical engineering from Cornell University (Ithaca, NY). Both Collins and Nikel previously worked at Digital Equipment Corp and Silicon Graphics.



You should use series—not parallel—termination for DDR-SDRAM stub-series terminatedlogic networks.

TABLE 4–ADDRESS TIMINGS

Address transmitter timing		
Skew component	Setup time (psec)	Hold time (psec)
Data/strobe clock jitter (t _{cut})	50	50
Clock skew (t _{cskew})	50	50
Buffer-delay mismatch (t _{BDM})	25	25
Buffer pulse-width distortion (t _{BPWD})	150	150
Data source-synchronous-output jitter (tpsso)	300	
Strobe SSO jitter (t _{ssso})		300
Total timing uncertainty	575	575
Address interconnect timing		
Skew component	Setup time (psec)	Hold time (psec)
PC-board-trace skew (t _{DCSK)}	200	200
Package-trace skew (tPKSK)	25	25
Data eye-pattern jitter (t _{deye})	900	900
Total timing uncertainty	1125	1125
Address receiver timing		
Skew component	Setup time (psec)	Hold time (psec)
DDR-SDRAM data setup (t _{DDRs})	2000	0
DDR-SDRAM data hold (t _{DDRH})	0	1000
Total timing uncertainty	2000	1000
Address end-to-end timing		
Skew component	Setup time (psec)	Hold time (psec)
Transmitter-timing uncertainty	575	575
Interconnect-timing uncertainty	1125	1125
Receiver-timing uncertainty	2000	1000
Timing margin at 125 MHz	300	1300