HIGH-SPEED ELECTRICAL SIGNALING:
Overview and Limitations

Mark Horowitz

Chih-Kong Ken Yang

Stefanos Sidiropoulos

Stanford University

As IC speeds and integration levels increase with advances in fabrication technology, so must off-chip data bandwidth. Although this goal is challenging, circuit design techniques will enable bandwidth to scale.

Advances in IC fabrication technology, coupled with aggressive circuit design, have led to exponential growth of IC speed and integration levels. For these improvements to benefit overall system performance, the communication bandwidth between systems and ICs must scale accordingly. Currently, communication links in various applications approach Gbps data rates. These applications include computer-to-peripheral connections,[1] local area networks,[2] memory buses,[3] and multiprocessor interconnection networks.[4]

Designers are concerned that these links will soon reach the fundamental limits of electrical signaling. In this article, we examine the limitations of CMOS implementations of high-speed links and show that the links' performance should continue to scale with technology. To handle the interconnects' finite bandwidth, however, requires more sophisticated signaling methods.

CMOS circuits, typically slower than circuits implemented in nonmainstream technologies, are particularly attractive for common applications because of their lower cost. The overall system cost is further reduced when signaling components are implemented as macro cells, integrated on the same die with a microprocessor or signal processing block. For this reason, we do not address bipolar or GaAs Gbps links.

Signaling issues

Figure 1 shows the components of a signaling system: transmitter, channel, and receiver. The transmitter converts digital information to a signal (waveform) on the transmission medium, or communication channel. This channel is commonly a board trace, coaxial cable, or twisted-pair wire. The receiver on the other end of the channel restores the signal, by sampling and quantizing it, to the original digital information.

Clock generation and timing recovery are tightly coupled to signal transmission and reception. The timing recovery, often embedded in the receiving side, adjusts the phase of the clock that strobes the receiver. The receiver samples the signal waveform at the optimal position.

Before discussing the performance of these components, we first need a metric that indicates how a CMOS circuit's performance scales with technology.

CMOS performance metric. Basic circuit speed improves as technology scales. Fortunately, all CMOS circuit delays scale roughly the same way; thus, the ratio of a circuit's delay to a reference circuit remains comparable. We exploit this with a metric called a fan-out of four (FO-4) delay. A FO-4 delay is the delay through one stage in a chain of inverters, in which each inverter drives a capacitive load (fan-out) four times larger than its input capacitance.

Figure 2a illustrates the normalized delay of various circuit structures versus technology and voltage scaling, demonstrating a 20% worst-case prediction accuracy for a relatively complex circuit structure. Figure 2b shows the actual FO-4 inverter delay for various technologies. The FO-4 delay for these processes can be roughly approximated by 500-ps/micron (gate length). In a 0.5-micron technology, a 1-Gbps data stream has bit widths that are 4 FO-4 delays, while in a 0.25-micron technology, the bit widths are 8 FO-4 delays.

Link performance metrics. Data bandwidth largely characterizes link performance. In many applications, latency, power, and die area are also crucial issues. This is especially the case in intrasystem parallel links in which cost is multiplied by the number of wires. Data bandwidth or bit rate is not equivalent to symbol rate because symbols may contain multiple bits of information. For example, a binary NRZ (non-

return-to-zero) signal has the same symbol rate as the bit rate. In contrast, a four-level pulse-amplitude-modulated (PAM) signal, comprising 2 bits per symbol, has a bit rate twice that of its symbol rate.

Another link performance metric, the bit error rate (BER), measures how many bit errors are made per second. BER is important because it reduces the effective system bandwidth and because, in many systems, applying error correction techniques can prohibitively increase the system cost.

System noise and imperfections cause errors. Intrinsic noise sources are the random fluctuations due to the inherent thermal and shot noise of the passive and active system components. However, especially in VLSI applications, other nonfundamental noise sources can limit the link performance. These sources include coupling from other channels, switching activity from other circuits integrated with the link circuitry, and reflections induced from channel imperfections. These noise types typically have a nonwhite frequency spectrum and exhibit strong data dependencies. Moreover, their overall power is often proportional to the power of the transmitted signals.

Link applications. In subsequent sections we examine circuits optimized for two different applications: medium-long serial links and short parallel links. In medium-long serial links, the designer's goal is to achieve maximum data rate because the wire is the critical resource. Since there is only one transmitter and receiver, the area and power costs are not paramount. In addition, latency of the link's active circuits is not a large concern because the channel delay already dominates overall system latency. To maximize performance, the

Figure 1. Signaling system components: transmitter, channel, and receiver.

link operates until the BER is on the order of 10^{-9} to 10^{-11}, and then relies on an error-correcting code to increase the system robustness.

Short parallel links are different. Intrasystem interconnects (for example, multiprocessor networks, CPU-memory links) typically exhibit much lower channel delays. Hence, the link circuitry's incremental latency has a much more important impact on the overall system performance. Furthermore, the tens to hundreds of transmitter and receiver circuits require modest area and power for each circuit. To further reduce latency and complexity, these systems target aggregate BERs that are practically zero (or much less than the projected mean-time-between-failures of the overall system), so no error correction is needed.

Signaling circuits

A natural limitation of link bandwidth is the on-chip data rate, dictated by the speed of the logic that processes the transmitted/received data and by the clock speed. A clock must be buffered and distributed across the chip. Figure 3 shows the
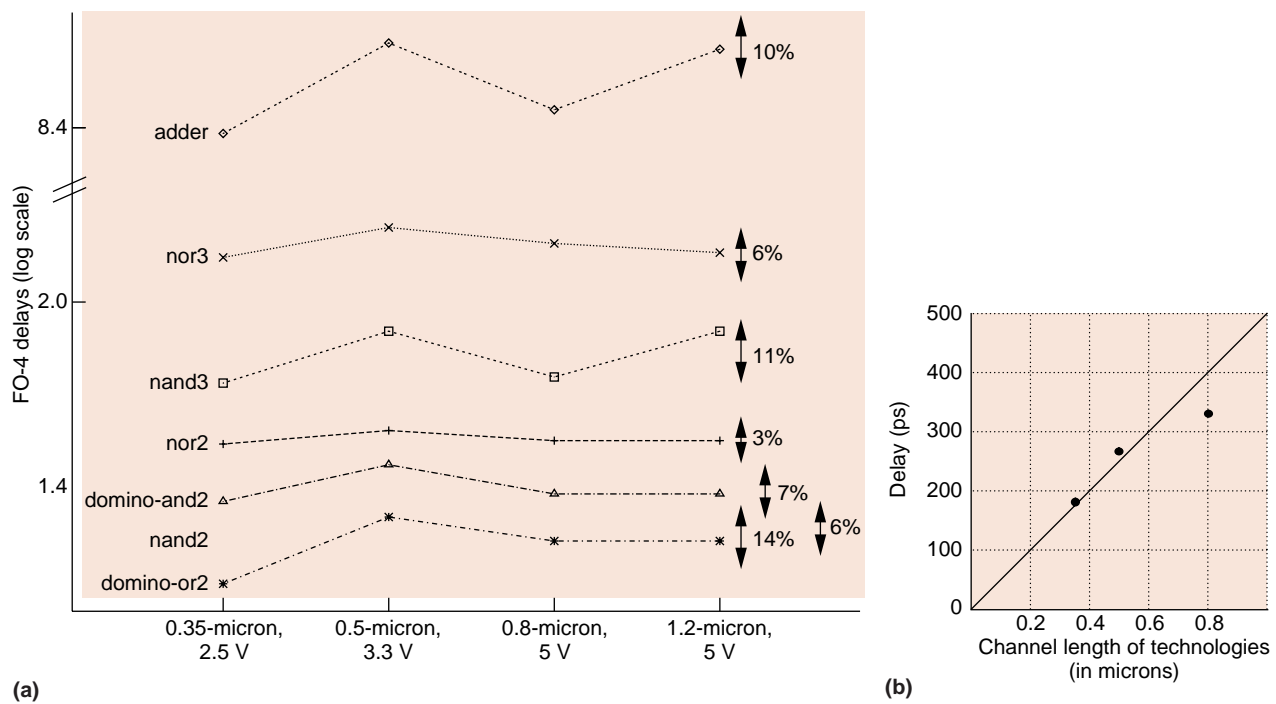

Figure 2. Normalized delay of circuit structures versus technology and voltage scaling (a), and FO-4 inverter delay for different CMOS technologies (b).
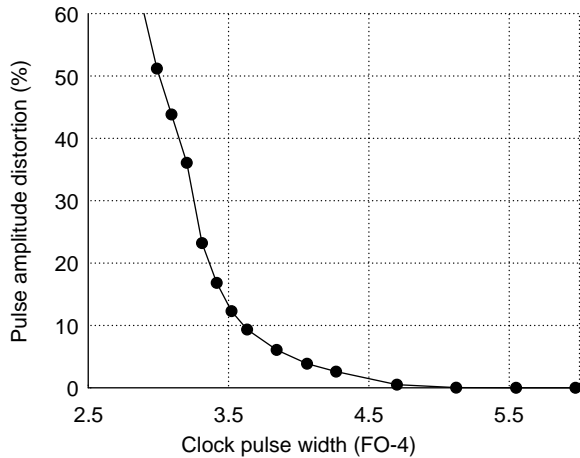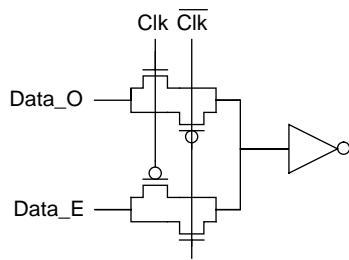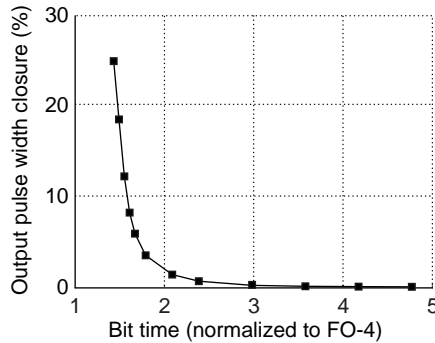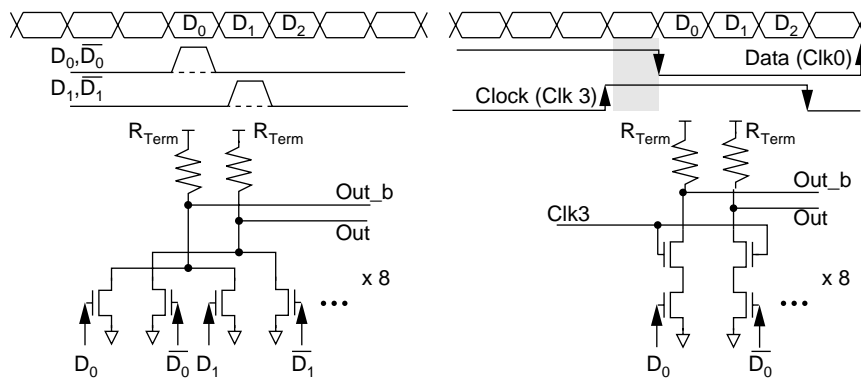
Figure 3. Pulse amplitude reduction as a percentage of pulse width.

Figure 4. Multiplexing predriver in a transmitter that is implementing parallelism.

Figure 5. A 2:1 multiplexer design in CMOS (a), and the effect of bit time on output pulse width of a simple 2:1 multiplexer (b).

Figure 6. Higher bandwidth multiplexing at the output pin, implemented on a pseudodifferential current-mode output driver (a) and in an improved design that uses two control signals to generate the output current pulse (b).
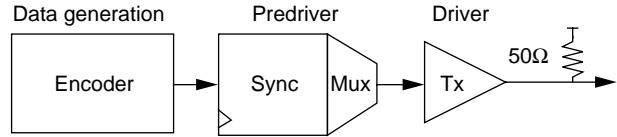
effect of driving different pulse widths (expressed in the FO-4 delay metric) through a clock buffer chain with a fan-out of 4 per stage. As the pulse width lessens, the clock pulse amplitude significantly lessens too. The minimum pulse that makes it through the clock chain is around 3 FO-4 delays.

Because a clock is actually a rising pulse followed by a falling pulse, the minimum clock cycle time will be 6 FO-4 delays. Accounting for margins, it is unlikely to see clocks faster than 8 FO-4 cycles. With one bit per cycle, this would give only 500 Mbits/s in a 0.5-micron technology. Designers can overcome this limitation with parallelism, as we discuss next, which lets the on-chip processing circuits operate at a lower frequency than the off-chip data rate.

Transmitter design. Figure 4 shows the block diagram of a transmitter implementing parallelism by multiplexing on-chip parallel data into a single serial bitstream. The primary bandwidth limitation here stems from either the multiplexer or the clocks. When the bit times are long, the multiplexer output achieves full CMOS swing.

Clipping the multiplexer output voltage creates an inherent memoryless medium; that is, the previous bit values do not affect the waveform of the bit being transmitted. However, when bit time is shorter than multiplexer settling time, the output no longer swings fully, and the values of the previously transmitted bits affect the current bit's waveform. This interference, called intersymbol interference (ISI), reduces the transmitted signal's timing and voltage margins.

Figure 5 shows the effect of bit time on the pulse width of the multiplexer output signal. It assumes that a fan-out of 2 buffer chain generates the clock driving the 2:1 multiplexer. The multiplexer is fast enough for a bit time of 2 FO-4s. However, a less aggressive clock buffering—a fan-out of 3 or 4 per stage—would increase the time to at least 3 FO-4s.

A larger fan-in multiplexer could get around the clock problem, but its larger capacitance would decrease its performance to about 3 FO-4 delays. This 2:1 multiplexer solution is simple and cheap, and is used in short parallel links.

Rather than rely on the on-chip multiplexer's bandwidth, we can use the high bandwidth available at the chip output pins. As the chip's output drives a low (25–50 ohm) impedance, high bandwidth is available even with larger output capacitance. Figure 6a shows the idea behind this technique implemented on a pseudodifferential current-mode output driver. To achieve parallelism through multiplexing, additional driver legs are connected in parallel. Each driver leg is enabled sequentially by well-timed pulses of a width equal to a bit time. In this scheme, on-chip voltage pulses generate the current pulses at the output. The minimum pulse width that can be generated on the chip limits the minimum output pulse width.[5]

An improved scheme, illustrated in Figure 6b, eases that limitation.[6] Two control signals, both at the slower on-chip clock rate, generate the output current pulse. In this design, either the transition time of the predriver output or the output bandwidth determines the maximum data rate.

Figure 7 shows the amount of pulse time closure with decreasing bit width for an 8:1 multiplexer using this technique. The minimum bit time achievable for a given technology is less than a single FO-4 inverter delay. The cost of this scheme is a more complex clock source, since it requires precise phase-shifted clocks and increased latency (measured in bit times). The highest speed links use this technique, achieving bit times on the order of 1 FO-4.

Receiver design. Designers face a more challenging problem in the receiver front end. The typically small swing signal on the channel must be restored to full CMOS swings. If a conventional amplifier restores the signal, its output bandwidth would limit the achievable bit rate, causing intersymbol interference. Again, the use of parallelism will relax the bandwidth requirements of any single receiver component, increasing the maximum achievable data rate. Designers can attain this by first sampling the data through multiple parallel samplers, each followed by an amplifier. Each amplifier now has a bandwidth requirement that is a fraction of the original single front-end amplifier bandwidth.[7] Ideally, an amplifier with a high gain–bandwidth product optimizes performance. An effective design is a regenerative amplifier, whose gain is exponentially related to the bandwidth due to the positive feedback.

Commonly, in low-latency parallel systems,[3] two receivers are used on each input, one of them triggered by the positive and one by the negative edge of the clock, as Figure 8a shows. Each receiver has 1/2 cycle to sample (while resetting the amplifier). Another 1/2 cycle—one full bit time—can be allocated for the regenerative amplifier to resolve the sampled value. The receiver's minimum operating cycle time determines the bit rate achievable by this simple demultiplexing
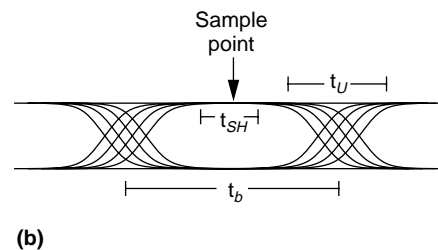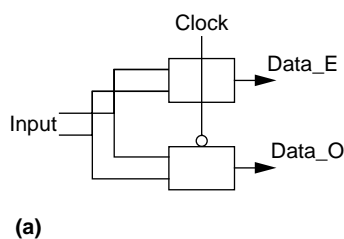
structure. Representative receiver designs can achieve bit widths on the order of 4 FO-4, which match the clock limitations.[3,6]

Similar to the transmitter, for a higher degree of demultiplexing, designers can employ multiple clock phases with well-controlled spacing. In such a system as Figure 9 shows, the demultiplexing occurs at the input sampling switches.[5,6] In this parallelized architecture, three items limit data bandwidth. These are the accuracy in generating the finely spaced clock phases (discussed later), the sampling aperture (sampling bandwidth) of CMOS transistors, and the input capacitance. The sampling aperture is the time required for the sampler to capture the input value. An NMOS transistor sampler can easily be sized for low-enough resistance to have a sampling aperture less than 1/3 FO-4. This performance is sufficient to robustly recover data with a FO-4 bit time. However, an often more severe limitation results from the input capacitance, which depends on the sampling network design and the degree of demultiplexing. In a well-balanced design, the maximum achievable data rate is the product of the inverse of the minimum cycle time of the front-end receiver, multiplied by the degree of demultiplexing. The demultiplexing degree is limited such that the input RC time constant is small enough to not cause additional ISI on the input signal.

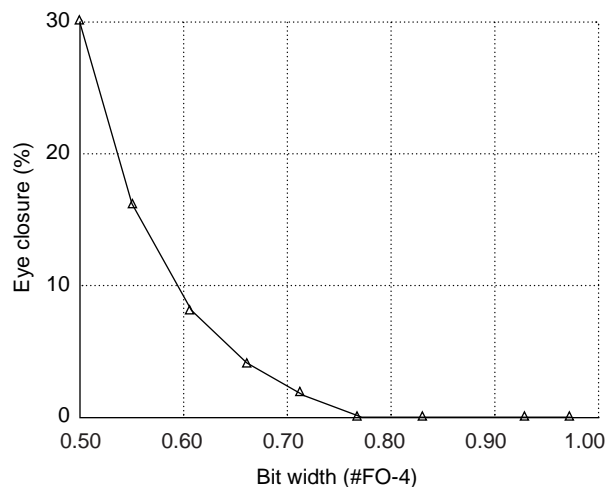Even with sufficient sampling bandwidth, sampling uncer-

Figure 7. Effect of bit time on pulse width of the higher bandwidth 8:1 multiplexer.

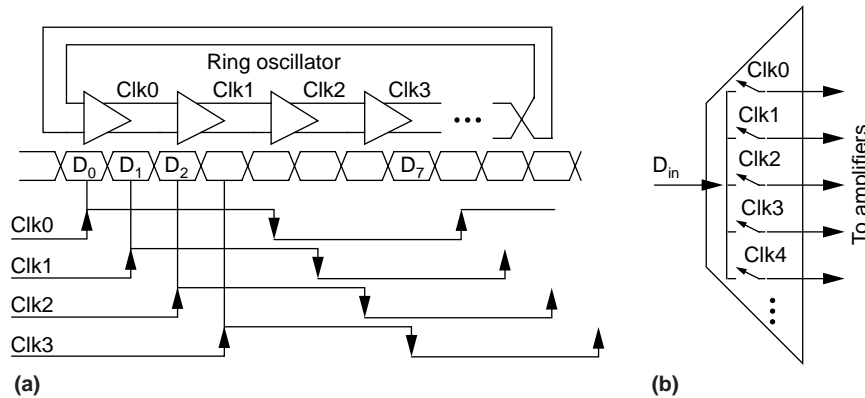Figure 8. Simple 1:2 demultiplexing (a) and timing margin (b).

Figure 9. A 1:8 demultiplexing receiver. Clock signals on the left (a) are used in the circuit block diagram (b).

tainty (aperture uncertainty) can degrade performance. For NRZ signaling, the sampling occurs in the center of the data bit for maximum timing margin. Figure 8b illustrates that because of data uncertainty t_U, the sampling uncertainty cannot exceed t_{SU}; otherwise, errors occur. Sampling uncertainty is the sum of phase uncertainty in the sampling clock, static input offset of the amplifier, and sampling noise. Phase uncertainty, both static and dynamic, is the dominant source, which we discuss later.

Another noise issue arises in many parallel links. Due to cost, these links typically use pseudodifferential signaling.[3,4] In these systems, receivers at each pin share a reference voltage. Reference-voltage sharing creates an imbalance on the load of the reference and input lines. This causes on-chip power supply and substrate noise to be more heavily coupled on the shared reference line. The high-frequency noise manifests in the time domain as reference voltage "spikes" that can be detrimental in operating a high-bandwidth sampling/regenerative receiver.

To reduce reference noise effects, the receiver should filter the high-frequency reference noise. An effective filter resembles the traditional communications integrate-and-dump filter.[8] Because the input signal in low-latency parallel links is valid for longer than the brief instant of the sampling aperture, capacitors can integrate current based on the input voltage difference. At the end of the integrating period, a regenerative amplifier resolves the integrated voltage. Since input differential voltage "spikes" are averaged over the bit time, the final polarity of the integrated signal is unaffected, improving the overall system robustness.[9]

Transmitter and receiver summary. Transmitters and receivers have similar limitations. A simple 2-1 multiplexing/demultiplexing transmitter/receiver pair can easily achieve bit times of 3 to 4 FO-4 inverter delays and thus should continue to scale with technology. Further improved data rates are possible with wider transmitter multiplexers and receiver demultiplexers. By employing greater parallelism, these systems can achieve bit times of approximately 1 FO-4 inverter delay. These systems mainly require precise timing, placing more stringent requirements on the

link synchronization circuits, which we discuss next.

Synchronization circuits

To properly recover the bit sequence at the channel's receiver end, the receiver's sampling clock phase must have a stable, predetermined relationship to the incoming data's phase. This maximizes timing margins. In higher bandwidth systems, the deterministic phase relationship is even more stringently required. In these systems, the bit rate is a multiple of the on-chip clock. This requires either an explicitly faster bit clock or multiple phases of lower frequency clocks with a well-controlled phase relationship between them.

Clock quality can be characterized by phase offset and jitter. Phase offset is a static (dc) quantity that equals the difference between a clock's ideal average position and the actual average position. This offset can refer to the phase relationship between clock and data, as well as to intraphase offset in multiplexing systems with multiple clock phases.

Jitter is the dynamic (ac) variation of phase, dominated by on-chip power supply and substrate noise. Jitter is specified in terms of both short- and long-term variations. Cycle-to-cycle jitter describes the short-term uncertainty on the clock period. Long-term jitter describes the uncertainty in the clock position with respect to the system clock source.

In conventional digital design, the most important requirement is minimizing cycle-to-cycle jitter. In high-speed links, however, both quantities can be equally important. Imperfections on the system clock source and slow temperature and operating voltage variations cause low-frequency jitter. With a phase-locked loop, designers can track this type of jitter reasonably well. On-chip supply and substrate noise are major concerns because they cause medium frequency and cycle-to-cycle jitter.

Clock generation architectures. A reliable, flexible method to resolve the synchronization problem uses on-chip active phase-aligning circuits. Generally, these circuits are known as phase-locked loops. These control systems use negative feedback to align the phase of the on-chip receive or transmit clock to the phase of an external reference. For parallel links, this external reference is the clock distributed along with the data. For serial links, the reference is more often extracted from the serial data stream.

In a parallel link, the process of buffering this reference clock to drive multiple receivers alters the timing relationship between clock and data. With the delay of the clock amplification/buffering appropriately embedded in the feedback path, phase-locked loops can cancel out the skew. This fixes the phase relationship between the internal clock and external reference. Similarly, in a serial link, the timing information extracted from the data must be fed back to control the sampling phase.

To improve the system timing margin, designers must min-

imize the additional fixed and time-varying phase uncertainty—offset and jitter—introduced by the phase-aligning blocks. This means minimizing the effect of supply and substrate noise.

Figure 10 shows two alternative control loop topologies for high-speed signaling systems: voltage-controlled oscillator (VCO)-based phase-locked loops (PLLs), and delay-line-based phase-locked loops or delay-locked loops (DLLs). Both circuits try to drive the phase of their periodic output signal to have a fixed relationship with the phase of their input signal.

A PLL employs a VCO to generate its output clock. The phase detector compares the clock's phase with that of the reference. The loop filter filters the phase detector's output, generating the loop control voltage that drives the VCO's control input.

Because a VCO integrates frequency to generate the phase of its output clock, a PLL is inherently a higher order control system. The system's transfer function contains two poles at the origin. The first pole is due to the VCO's phase-integrating nature. The second is due to the integrator usually embedded in the loop filter to achieve zero static-phase error. To counteract the effect of these two poles, the loop transfer function must contain a stabilizing zero. Designers usually implement the zero in the loop filter by employing a resistor in series with the integrating capacitor.

This higher order nature of the PLL creates some design challenges. For example, the effects of varying process and environmental condition on the stabilizing zero position might be detrimental to the loop stability.[10,11] On the other hand, a VCO has important advantages. First, the jitter of the reference signal only indirectly affects the output clock jitter, because the loop acts as a low-pass filter. Second, the oscillator allows the internal clock frequency to be a multiple of the reference. This frequency multiplication property primarily underlies the widespread adoption of PLLs in applications such as microprocessor clock generation.[12] Moreover, since the VCO inherently generates a periodic clock signal, PLLs using appropriate phase detector designs are common in clock and data recovery applications.[13]

Delay-locked loops, in contrast, rely on the fact that in many applications the reference signal is already a clock of the right frequency.[14] Instead of generating their output clock with a VCO, DLLs use a voltage-controlled delay line. The VCDL generates the output clock by delaying its input clock by a controllable time. The phase detector compares the VCDL output clock's phase with that of the reference clock. The loop filter filters the phase detector output, generating control voltage V_C. This control voltage drives the VCDL control input, closing the negative feedback loop.

As the VCDL in this system is simply a delay gain element, the loop filter does not need a stabilizing zero. Designers can implement the filter with a single integrator (for exam-
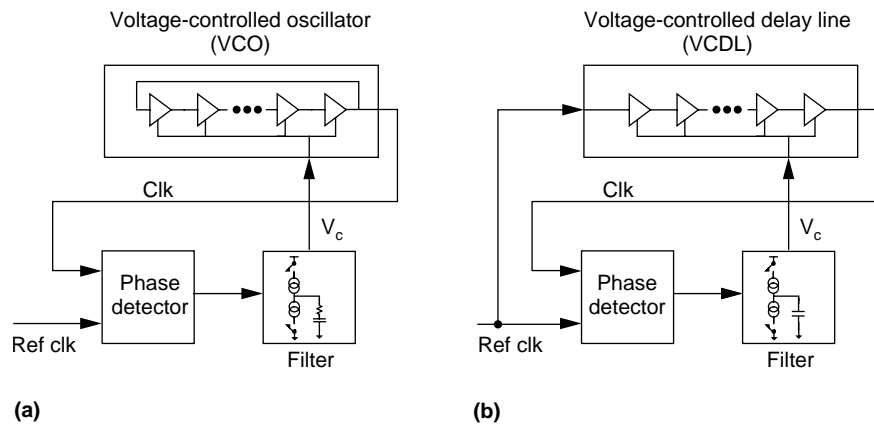

Figure 10. PLL (a) and DLL (b) loop topologies.

ple, a charge pump and a capacitor). This system is unconditionally stable and is easier to design. Additionally, a DLL can be easily implemented as a bang-bang control system. The phase detector output in this system is simply a binary up-down phase error indication rather than a voltage proportional to the instantaneous phase error. In this case, the phase detector can be a replica of the input pin receiver. The receiving clock edge placement thus compensates for the input receiver's setup time, which is especially important in guaranteeing timing margins at high bit rates. In contrast, due to frequency acquisition constraints, PLLs usually rely on a state-machine-based phase-frequency detector, which results in suboptimal placement of the receiving clock edge.

In the noisy environment of a digital IC, the most important difference between PLLs and DLLs is in the way they react to noise. The delay elements within the delay line or VCO have a sensitivity to supply or substrate noise. This performance measure is best expressed in a normalized *percentage of delay change* per *percentage of supply or substrate change* (%delay/%volt).

The sensitivity varies considerably with the elements' design. For example, a simple CMOS inverter has a supply sensitivity of 1-%delay/%supply. A well-designed differential buffer, as shown in Figure 11 (next page), has a supply sensitivity of 0.2-%delay/%supply. Typically, a PLL will have higher supply or substrate noise sensitivity than a DLL comprising identical delay elements.

A change on the supply or substrate of a VCO results in a change on its operating frequency. This frequency difference causes an increasing phase error that accumulates until the loop feedback's correcting action takes effect. In contrast, the change on the supply of a VCDL causes a delay change through the delay line. Because the VCDL does not recirculate its output clock, the resulting phase error does not accumulate. Instead, it decreases with a rate proportional to the loop bandwidth. Figure 12 shows this performance difference and illustrates the simulated phase error of a PLL and a DLL under a supply voltage step. Both the PLL and the DLL are locked to a reference clock with a frequency of 250 MHz.

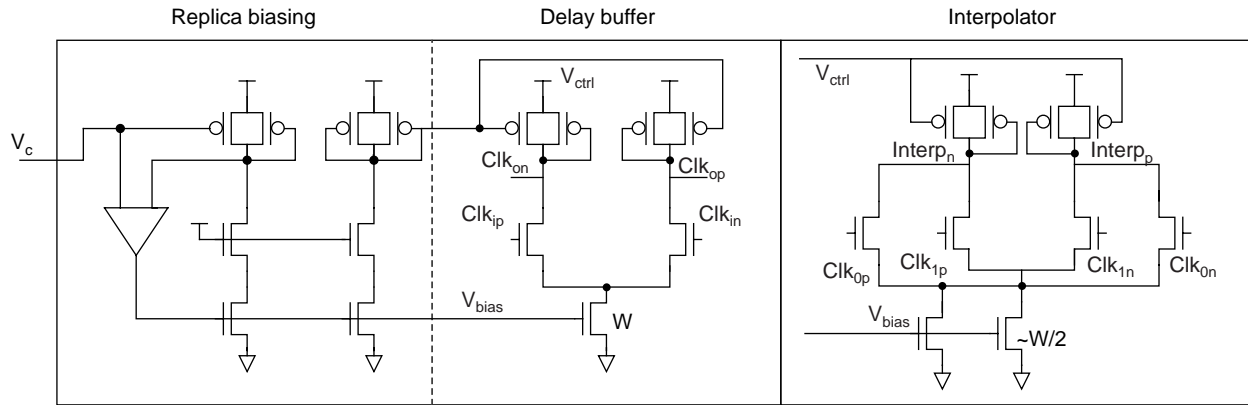The VCO and the VCDL comprise six voltage-controlled
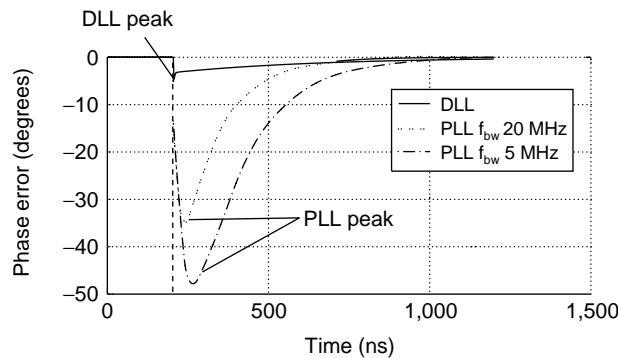
Figure 11. Delay element and interpolator.

Figure 12. PLL and DLL transient response to supply step.

delay elements, each with a supply sensitivity of 1.8°/volt in a 3.3-V supply environment. That is, a 1-V change in the supply of the VCO or VCDL with a 4-ns cycle time changes the delay through each element by 20 ps, corresponding to 0.2-%delay/%supply. A 300-mV supply step is applied on both the VCO and the VCDL.

As Figure 12 shows, the PLL peak phase error is generally larger than 6.5° ($12 \times 1.8° \times 0.3$). The magnitude of this error depends on both the delay element supply sensitivity and the loop bandwidth. A larger loop bandwidth results in less phase error accumulation, thus minimizing the peak phase error.

In contrast, the DLL phase error depends only on the delay elements' supply sensitivity. Its peak occurs during the first clock cycle after the supply step. Even in the best case, where the PLL bandwidth is 20 MHz, the peak phase error is approximately a factor of 6 larger than that of the DLL. (Increasing the PLL bandwidth further than 1/10 of the operating clock frequency compromises loop stability.)

In applications, therefore, such as high-speed parallel signaling that do not require clock frequency multiplication, DLL use maximizes the timing margins. This is because it exhibits lower supply- and substrate-induced phase noise, and because it can more readily use the input pin receiver as a phase detector.

PLL and DLL design trade-offs. Note that other factors such as the system clock quality and the final on-chip clock buffer supply sensitivity can affect design trade-offs. For example, the earlier comparison does not include the supply sensitivity of the final on-chip clock buffer, which typically comprises CMOS inverters. Because an inverter's supply sensitivity is approximately five times worse than that of the delay elements, a long buffer chain can contribute to significant jitter. Thus the difference between a system with a PLL and that with a DLL is often smaller than the factor of 6 cited earlier. This smaller noise difference and the frequency multiplication capability of PLLs make them a better choice in many applications.

Multiphasic clock generation. For the higher order multiplexing and demultiplexing systems we described, designers use precisely spaced clock phases to determine the bit width. Several techniques can generate these phases. The simplest is a ring oscillator (or a delay line with its phase input locked to its output phase) that taps the output of each stage. For example, a six-stage oscillator employing differential stages generates 12 edges evenly spaced in the 0 to 360° interval. Figure 11 showed an example of a robust differential delay element[15] with low supply sensitivity.

For even finer phase spacing than a single buffer delay, phase interpolators can generate a clock edge occurring halfway between two input phases. An interpolator design uses two buffers—each with different phases as inputs and each with a fraction of the drive strength. The sum of the drive strengths equals that of a normal buffer. The buffer with the earlier phase input drives the output for the period of the phase difference between inputs. Then the full drive strength drives the output. The resulting signal, compared with normally buffered versions of the inputs, has an intermediate phase that depends on the drive-strength ratio of the interpolating buffers.

Figure 11 showed an interpolator design using the same type of buffer as the delay element. Because these elements are not perfect integrators, the interpolator does not interpolate linearly with the current ratio. The quality of phase spacing generated by a ring oscillator and interpolators has errors measurably less than ±8% of the ideal phase spacing.

Alternative techniques exist for generating even more finely spaced clock phases, such as coupled oscillators and delay verniers.[16,17]

Tracking bandwidth considerations. Regardless of whether the clock generation employs a PLL or a DLL, a high loop tracking bandwidth can reduce jitter's effect. Tracking the data's phase variation can improve the system timing margin, provided that the phase variation is not a random cycle-to-cycle variation. Ongoing research is determining the actual on-chip noise spectrum. If the phase noise is correlated from one cycle to the next, a high tracking bandwidth is desirable. Achieving a PLL bandwidth of the same order as the operating clock frequency is virtually impossible, because of loop stability constraints.

A promising alternative used in UARTs[18] is data oversampling, in which multiple clock phases sample each data bit from the data stream at multiple positions. This technique detects data transitions and picks the sample furthest away from the transition. By delaying the samples while the decision is made, this method essentially employs a feed-forward loop, as Figure 13 shows. Because stability constraints are absent, this method can achieve very high bandwidth and track phase movements on a cycle-to-cycle basis. However, the tracking can occur only at quantized steps depending on the degree of oversampling, and the phase-picking decision incurs significant latency.

Jitter and phase error scaling. The basic clock speed scales with technology because each buffer's delay will scale. Each buffer has a minimum delay of less than a FO-4 inverter, so it's easy to use a 4-to-6 buffer ring and generate the 8 FO-4 clock. Designers can expect jitter and phase error to also scale with technology. A buffer's supply and substrate sensitivity is typically a constant percentage of its delay on the order of 0.1–0.3 %delay/%supply for differential designs with replica feedback biasing.[15] This implies that the supply- and substrate-induced jitter on a DLL scales with operating frequency—the shorter the delay in the chain, the smaller the phase noise.

This argument also holds for the jitter caused by the buffer chain that follows the DLL. As technology scales, the delay—and resulting jitter—of the buffer chain scales. The PLL's jitter scales with frequency if the loop bandwidth (and thus the input clock frequency) scales, too. If the reference clock does not scale due to cost constraints, designers can effectively use phase-picking to increase the tracking bandwidth. However, this improvement in the link's robustness is at the expense of increased latency.

Imperfections in the static phase spacing would cause bit-width variations, thus reducing the aggregate data eye. Errors result primarily from mismatches in the delays of the ring oscillator stages and corresponding buffering and interpolating paths. Layout matching errors can cause differences in coupling and load capacitance of multiple parallel clock paths, resulting in delay mismatches.
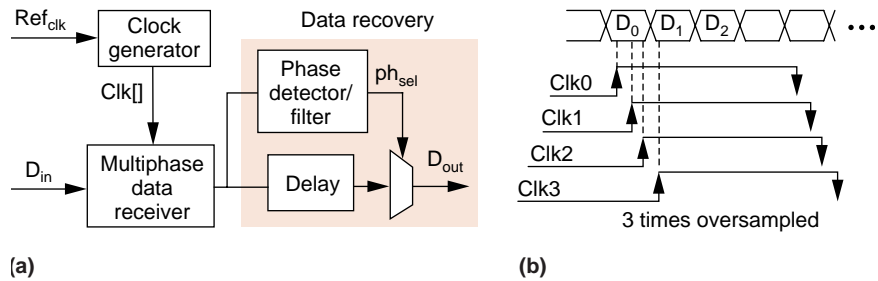

Figure 13. Phase-picking data recovery. The block diagram illustrates phase picking (a); the clock waveforms are oversampling the data stream (b).

Because careful design minimizes these systematic errors, a more important source of phase offset is the random mismatches between nominally identical devices such as transistor thresholds and widths. This timing-error source becomes more severe with technology scaling, as the device threshold voltage becomes a larger fraction of the scaled voltage swing. Designers can thus expect phase-spacing errors as a percentage of bit widths to increase with decreasing transistor feature sizes. Static timing calibration schemes, however, can cancel these errors because of the errors' static nature. For example, designers can augment interpolating clock generation architectures[16] with digitally controlled phase interpolators[19] and digital control logic to cancel device-induced phase errors.

In addition to random device mismatches, decreasing bit times in low-latency parallel interfaces will magnify the effect of mismatches on the electrical length of parallel interconnects. Per-channel timing adjustments can also mitigate this problem.

Link performance examples

Our research group built two different links that explore some of these design issues. The first is a parallel, low-latency (1.5 cycles) link, targeting a 4 FO-4 bit time in a 0.8-micron process and employing pseudodifferential signaling. To improve the reception robustness, the chip uses a current integrating receiver.[9] To minimize jitter we used a DLL.[15] The chip included PRBS testing and supply-noise monitoring circuits. The design achieved a BER less than 10^{-14} and a minimum bit time of 3.3 FO-4s. This yielded a maximum transfer rate of 900 Mbps per parallel pin operating from a 4-V supply. We measured the DLL's supply jitter sensitivity at 0.7-ps/mV, 60% of which can be attributed to the final clock buffer.

The second link we built[20] is a serial link transceiver targeting a data rate of 4 Gbps in a 0.6-micron (drawn) process with a 1 FO-4 bit width. The chip uses an input regenerative amplifier based on Yukawa's design[21] as part of the 1:8 demultiplexing receiver, and the 8:1 multiplexing transmitter of Figure 6b. A three-times oversampled phase-picking method performs timing recovery, similar to work by Lee.[5] Data recovery latency is 64 bits.

Our second design had a considerable area penalty (3 × 3 mm²), resulting from the phase-picking decision logic. The chip achieved a BER of less than 10^{-14} with the transmitter
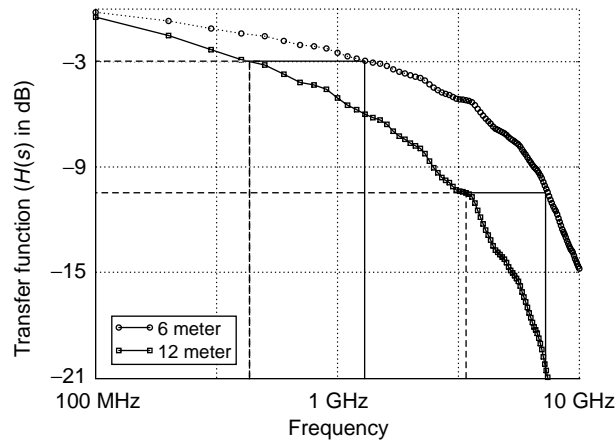
Figure 14. Transfer function of 12-meter and 6-meter RG-55U cable. The cable bandwidth is usually set by the 3-dB point (top line). By attenuating the low-frequency components, the effective bandwidth can be extended (bottom line). $H(s)$ is the transfer function, where s is frequency.

Figure 15. Time domain response of a square pulse into the 12-meter RG-55U cable.

output fed back to the receiver operating at 3.3-V supply. Combined static phase-spacing errors and jitter degrade the transmitted data eye by only one-fourth of the bit time. Due to the small delay of the on-chip clock buffers, the supply sensitivity of the clocking circuits is only 0.6 ps/mV.

Channel limitations

We have thus far focused on the implementation and technology limitations of the transmit, receive, and timing recovery circuits without addressing the effects of the channel. We have shown that circuits should continue to scale with technology. For example, Chang ported the integrative receiver to a 0.25-micron technology and achieved a 2-Gbps data rate.[22] Unfortunately, the wires' bandwidth is not infinite and will limit the bit rate of simple binary signaling. The 4-Gbps rate of the serial link example that we built is already higher than the bandwidth of copper cables longer than a few meters.

While finite wire bandwidth is an issue that designers must deal with, it will not fundamentally limit signaling rate, at least not for a while. The question of how to maximize the number of bits communicated through a finite bandwidth channel is an old one. It is the basis of modem technology, which transmits at a rate of 30 to 50 Kbps through the limited bandwidth (4 kHz) of phone lines.[23]

To counteract channel limitations, designers have developed complex schemes that equalize the channel's attenuation to extend—and more fully utilize available—bandwidth. We quickly review these techniques here; see Proakis and Salehi[8] for a more complete discussion.

Cable characteristics. A cable's bandwidth limitation depends on the size and construction of the cable's conductor and shield, and the dielectric material.[24] The cable's conductor thickness determines the surface area on which the current can flow. Along with the metal's conductivity, this area determines the conductor's resistance per meter.
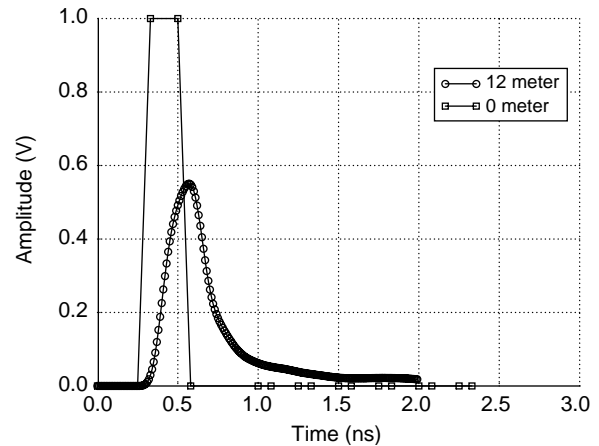
Signal current on the conductor requires a return path to close the circuit; the loop size and the proximity to the return path determine inductance and capacitance (L, C). These characteristics determine the cable frequency response, formally expressed by a transfer function.

For better frequency response, cables are designed with a shield as the return path, isolated from the signal by a fixed distance with a dielectric material. With this construction, the signal sees a distributed LC. The LC behaves as temporary energy storage that propagates a lossless signal down the cable as a wave. The effect of an ideal cable is just a signal delay, depending on the cable length and no signal energy loss at any frequency. At the end of the line, a resistor with value $\sqrt{L/C}$ terminates the line and absorbs the propagated energy.

In reality, such an ideal medium propagating all frequency components of the signal does not exist. The transfer function of a 6-meter and 12-meter RG55U cable, shown in Figure 14, illustrates increasing attenuation with frequency. The attenuation's main source is the cable's series resistance. Due to the skin effect, this resistance increases at higher frequencies because higher frequency currents travel closer to the conductor surface, reducing the area of current flow. The increase in resistance is proportional to $\sqrt{frequency}$. A lesser cause of attenuation is energy loss through the imperfect dielectric that isolates the signal from the shield.

Figure 15 demonstrates the time-domain effect of the frequency-dependent attenuation. A single square pulse is injected into the 12-meter cable. The attenuation above the pulse frequency reduces the pulse amplitude by more than 40%. Moreover, lower frequency attenuation results in the signal's long settling tail. The net effect on a transmitted pseudorandom pattern is a significant closure of the resulting data eye. As a result, the cable quality and length ultimately limit the NRZ data bandwidth, unless the cables are extremely high quality, which increases the overall system cost.

The simplest way to achieve higher data rate, even with the lossy channels, is to actively compensate for the uneven channel transfer function. Designers can accomplish this
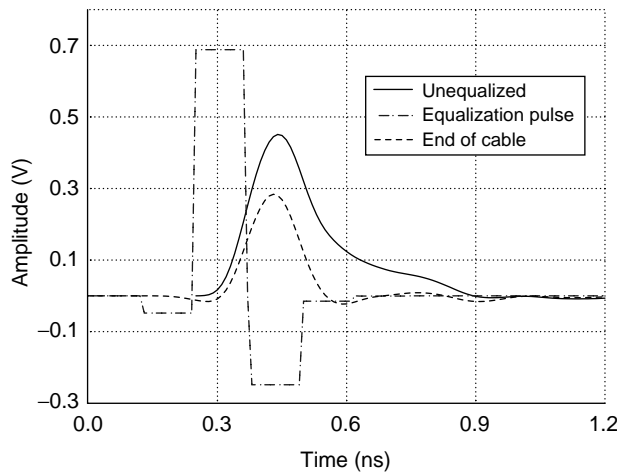
Figure 16. Pulse response from a predistorted transmitter.

Figure 17. Simulated 4-PAM data eye.

either through a predistorting transmitter[25,26] or an equalizing receiver.[27] Predistortion and equalization have a similar effect: the system transfer function is multiplied by the predistorting/equalizing transfer function, which ideally is the inverse of the channel transfer function.

Unfortunately, this flattened channel frequency response comes at the expense of reduced signal-to-noise ratio (SNR). This is because equalization attenuates the lower frequency signal components so that they match the high-frequency channel-attenuated components (shown in the horizontal lines in Figure 14). This reduces the overall signal power, and hence degrades the system SNR. The advantage of these simple equalization techniques is that they can be implemented with little latency and power overhead.

Predistorting transmitters. Transmitter predistortion uses a filter that precedes the actual line driver. Filter inputs are the current, past, and future bits being transmitted. Filter coefficients depend on the channel characteristics. The optimal filter length depends on the number of bits affecting the currently transmitted bit. Effectively, the FIR filter output is no longer a binary signal but a multilevel analog signal. The output driver behaves as a high-frequency digital-to-analog converter (DAC) that operates at the bit rate. In the simplest case, the FIR filter effectively suppresses the power of low-frequency components by reducing the amplitude of continuous strings of same-value data on the line. Simultaneously, it keeps the power of high-signal-frequency components the same by increasing the signal amplitude during transitions. Figure 16 shows the predistorted pulse and the resulting pulse at the end of the cable as compared with the original pulse.

Because the transmitter has no information about the signal shape at the channel's receiver end, obtaining the appropriate FIR filter coefficients can be formidable. To date, predistorting high-speed transmitters use static filter coefficients relying on a fixed channel transfer function. In the future, information from the receiver will probably need to be sent to the transmitter through a separate channel to "train" the transmitter filter. This will ensure robust operation under varying channel conditions.
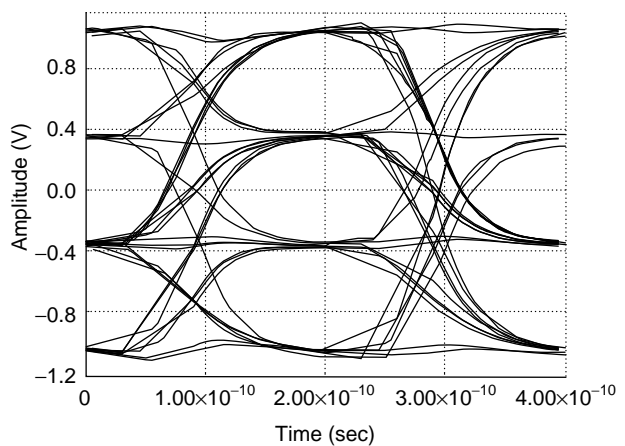
Receiver equalization. Receiver-side equalization relies essentially on the same mechanism as predistortion, by using a receiver with increased high-frequency gain to flatten the system response.[27] Designers can implement this filtering as an amplifier with the appropriate frequency response. Alternatively, designers can implement the filter in the digital domain by feeding the input to an analog-to-digital converter (ADC) and postprocessing the ADC's output with a high-pass filter. Like the predistorting transmitter, this technique reduces the overall system SNR, but it does so because the high-pass filter also amplifies the noise at higher frequencies. The usual technique is to use an ADC at the input and build the filters digitally, since it also lets designers implement more complex and nonlinear receive filters. While this approach works well when the data input is bandwidth-limited to several MHz, it becomes more difficult with GHz signals, which require Gsamples/s converters.

Trade-offs. Equalization in either the transmitter or receiver is the simplest design technique and is effective for extending the wires' bit rate. Its cost is modest, especially if done at the transmitter. However, equalization does not take full advantage of the channel, as it simply attenuates the low-frequency signals resulting in smaller signal amplitudes. If SNR is large enough to detect these small signals, rather than attenuating all the signals, a more efficient scheme is to send many small signals at once (multibit signals) and detect them.

Multilevel signaling. Transmitting multiple bits in each transmit time decreases the required bandwidth of the channel for a given bit rate. The simplest multilevel transmission scheme is pulse amplitude modulation. This requires a digital-to-analog converter (DAC) for the transmitter and an analog-to-digital converter (ADC) for the receiver. An example is four-level pulse amplitude modulation (4-PAM), in which each symbol time comprises 2 bits of information. Figure 17 shows a 5-GSym/s, 2-bit data eye.[28] The predistortion and equalization just discussed can still be applied to improve the SNR if the symbol rate approaches or exceeds the channel bandwidth.

Shannon's capacity theorem determines the upper bound on maximum bit rate. Equation $C/f_{bw} = \log (1 + S/N)$ formu-
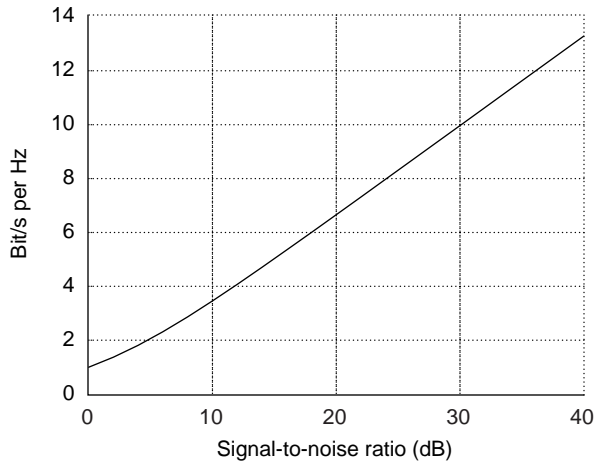
Figure 18. Shannon's capacity theorem determines the maximum capacity per hertz of channel bandwidth.

Figure 19. Link performance versus technology for existing links.

lates the maximum capacity (bits/s) per Hertz of channel bandwidth as a function of the SNR (see Figure 18). The larger the SNR, the more information transmitting multilevel signals can transfer for a given bandwidth.

Once the receiver has to deal with multilevel signals, designers can further improve link performance by letting symbols overlap and accounting for it in the detector. When the input consists of the sum of previous bits, a sequence detector (convolutional coders or Viterbi detectors[29]) determines the transmitted bits. Instead of deciding the transmitted values on a bit-by-bit basis, these detectors receive an entire bit sequence prior to making a decision. Thus they can use the signal energy that arrives late to help determine the bit value and improve the effective SNR. The downside to this technique is that the receiver input has effectively more values than the transmitted levels, thus requiring a higher resolution ADC.

DAC and ADC requirements. All these techniques require higher analog resolution for the transmitter and the receiver. The trade-off for higher resolution transmitter DACs and receiver ADCs is higher data throughput and improved SNR. The intrinsic noise floor for these converters is thermal noise $\sqrt{4kTRf}$. In a 50-ohm environment, the noise is about 1–nV/$\sqrt{Hz}$ (at 4GHz, 3σ is roughly 200 μV), which is negligible for 2 to 5 bits resolution even at GHz bandwidth.

Implementing high-speed DACs with less than 8-bit resolution is less challenging than implementing similar speed and resolution ADCs. This is so because transmitter devices are large and thus less sensitive to device mismatches. Instead, these DACs are limited by circuit issues: primarily transmit clock jitter and the settling of output transition-induced glitches. Both of them relate to the intrinsic technology speed and therefore will scale with reduced transistor feature size. High-speed, low-resolution DACs are rare; the nearest comparable product is found in video display RAMDACs achieving 8-bit linearity operating at 320 MHz on a 0.8-micron process.[30]

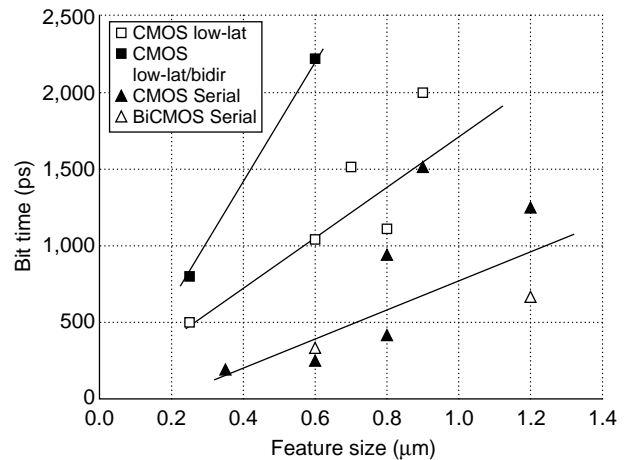The receive-side ADC is more difficult. It requires accurate sampling and signal amplification on the order of 30 mV (5 bits of resolution for 1-V signal) at a very high rate. Such fast sampling requires a high sampling bandwidth (small sampling aperture), which depends on the on-resistance of the sampling field-effect transistors and the slew-rate of the clock driving the sample-and-hold network. Since both of these quantities scale well with technology, sampling aperture will not limit ADC performance.

To maintain good resolution, the aperture uncertainty must affect the sampled value by less than one least significant bit. For an ADC sampling a sine wave, the aperture uncertainty must be less than $1/(\pi f 2^m)$,[31] where m is the number of bits and f is the input frequency.

The aperture uncertainty window is dominated by the sampling clock's jitter. It now also includes an additional component from any nonlinear voltage dependency of the sampling. Since jitter scales with technology, and the sampling nonlinearity can be characterized as a percentage error of the sampling aperture, we can expect the aperture uncertainty to scale. The only factor not scaling well with technology is random device mismatch. Shrinking device dimensions decrease the total area over which the random mismatches can be averaged, thus increasing the total error.[32] In addition, keeping the receiver input capacitance low requires smaller device sizes, which increases the effect of random device mismatch. Using active offset cancellation circuitry will be necessary to mitigate these effects.[31]

Examples of high-speed ADCs can be found in disk-drive read-channels, where a 6-bit flash ADC operating at 200 MHz has been demonstrated in a 0.6-micron process technology.[33] Unfortunately, this is still far below the 1- to 10-GHz sample rate that a high-speed link will require. Although there are no fundamental limitations preventing these devices, significant research is needed before they are practical.

AS TECHNOLOGY SCALES, WE CAN EXPECT the electronics capability to scale along with the FO-4 metric. This scaling has been reflected in the progressively increas-

ing bit rates in non-return-to-zero signaling. Figure 19 shows the published performance for both serial and parallel links fabricated in different technologies. Low-latency parallel links have maintained bit times of 3 to 4 FO-4 inverter delays. Meanwhile, through high degrees of parallelism, designers have attained off-chip data rates of serial links that exceed on-chip limitations, achieving bit times of 1 FO-4 inverter delay.

To track technology scaling, designs require additional complexity (for example, offset cancellation) to handle the increased device mismatches and on-chip noise levels. Although we expect jitter to scale with technology, designers must pay more attention to the clocking architecture. This is necessary to minimize phase noise and phase offset of the on-chip clock and to maintain high bit rates.

The fundamental limitation of bit-rate scaling stems from the communication channel rather than the circuit fabrication technology. Frequency-dependent channel attenuation limits the bandwidth, by introducing intersymbol interference. As long as sufficient signal power exists, the simplest technique to extend the bit rate beyond the cable bandwidth is transmitter predistortion.

Similar but slightly more complex is receive-side equalization. Also, with sufficient signal power, systems can transmit more bits per symbol by employing various modulation techniques. By limiting total signal power, designers can apply complex techniques such as sequence detection and coding to best use the available bandwidth. As Shannon's theorem predicts, the bps limit on information that can be transmitted per hertz of channel bandwidth depends entirely on the amount of expended signal power.

The cost of applying these methods is more digital signal processing, longer latency, and higher analog resolution. Digital processing complexity is a strength of CMOS technology. Therefore, it does not impose a limitation, other than the required increase in die area and power consumption. Much of the processing can be pipelined and parallelized to meet speed requirements. Also, as technology improves, the processing ability scales. However, the longer latency from applying these more elaborate techniques will limit their application to longer serial links. For low-latency intrasystem communication, only simpler techniques such as transmitter predistortion can be applied without significant latency penalty. Lastly, higher analog resolution is a critical requirement in both applications.

These techniques have been applicable to lower rates primarily because high-quality (low jitter and phase offset) clocks, along with high-resolution DACs and ADCs, are available. Current research concentrates on low-resolution and high-sampling bandwidth DA/AD converters, and timing offset cancellation both for serial and parallel links. Besides Shannon's limit, there is no fundamental limit to signaling rates, though many implementation challenges certainly exist. ⊔

References

1. I.J. Wickelgren, "The Facts about FireWire [serial communication bus]," *IEEE Spectrum*, Vol. 34, No. 4, Apr. 1997, pp. 19-25.
2. G.O. Thompson, "Work Progresses on Gigabit Ethernet," *Computer*, Vol. 30, No. 5, May 1997, pp. 95-96.
3. N. Kushiyama et al., "A 500-Megabyte/s Data Rate 4.5 M DRAM," *IEEE J. Solid-State Circuits*, Vol. 28, No. 4, Apr. 1993, pp. 490-498.
4. R. Mooney et al., "A 900 Mb/s Bidirectional Signaling Scheme," *IEEE J. Solid-State Circuits*, Vol. 30, No. 12, Dec. 1995, pp. 1,538-1,543.
5. K. Lee et al., "A CMOS Serial Link for Fully Duplexed Data Communication," *IEEE J. Solid-State Circuits*, Vol. 30, No. 4, Apr. 1995, pp. 353-364.
6. C.K.K. Yang et al., "A 0.8-mm CMOS 2.5 Gb/s Oversampling Receiver and Transmitter for Serial Links," *IEEE J. Solid-State Circuits*, Vol. 31, No. 12, Dec. 1996, pp. 2,015-2,023.
7. B. Kim et al., "A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2-mm CMOS," *IEEE J. Solid-State Circuits*, Vol. 25, No. 6, Dec. 1990, pp. 1,385-1,394.
8. J. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice Hall, Englewood Cliffs, N.J., 1994.
9. S. Sidiropoulos et al., "A 700-Mb/s/pin CMOS Signaling Interface Using Current Integrating Receivers," *IEEE J. Solid-State Circuits*, Vol. 32, No. 5, May 1997, pp. 681-690.
10. F.M. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Trans. Comm.*, Vol. COM-28, No. 11, Nov. 1980, pp. 1,849-1,858 .
11. R. Best, *Phase-Locked Loops*, 3rd ed., McGraw Hill, New York, 1997.
12. D.K. Jeong et al., "Design of PLL-Based Clock Generation Circuits," *IEEE J. Solid-State Circuits*, Vol. SC-22, No. 2, Apr. 1987, pp. 255-261.
13. B. Razavi, ed., *Monolithic Phase Locked Loops and Clock Recovery Circuits*, IEEE Press, Piscataway, N.J., 1996.
14. M.G. Johnson et al., "A Variable Delay Line PLL for CPU-Coprocessor Synchronization," *IEEE J. Solid-State Circuits*, Vol. 23, No. 5, Oct. 1988, pp. 1,218-1,223.
15. J.G. Maneatis et al., "Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques," *IEEE J. Solid-State Circuits,* Vol. 31, No. 11, Nov. 1996, pp. 1,723-1,732.
16. J.G. Maneatis et al., "Precise Delay Generation Using Coupled Oscillators," *IEEE J. Solid-State Circuits*, Vol. 28, No.12, Dec. 1993, pp. 1,273-1,282.
17. G.C. Moyer et al., "The Delay Vernier Pattern Generation Technique," *IEEE J. Solid-State Circuits*, Vol. 32, No. 4, Apr. 1997, pp. 551-562.
18. J.E. McNamara, *Technical Aspects of Data Communication,* 2nd ed., Digital Press, Bedford, Mass., 1982.
19. S. Sidiropoulos et al., "A Semi-Digital DLL with Unlimited Phase Shift Capability and 0.08-400 MHz Operating Range," *1997 IEEE Int'l Solids-State Circuits Conf.*, *Digest of Technical Papers*, pp. 332-333.
20. C.K.K. Yang et al., "A 0.6-μm CMOS 4-Gbps Transceiver with Data Recovery using Oversampling," *1997 IEEE Symp. VLSI Circuits*, *Digest of Technical Papers*, IEEE Press, Piscataway, N.J., 1997, pp. 71-72.
21. A. Yukawa et al., "A CMOS 8-Bit High Speed A/D Converter IC," *Proc. 10th European Solid-State Circuits Conf.,* IEEE Press, Piscataway, N.J., pp. 193-196.
22. K.Y. Chang et al., "A 2Gb/s Asymmetric Link for High-Bandwidth Packet Switches," *Hot Interconnects V Symp. Record*, Aug. 1997, IEEE Computer Society Press, Los Alamitos, Calif., pp. 171-179.

23. M. Combe et al., "A Second-Generation Modem Analog Front-End," *Proc. IEEE Int'l Solid-State Circuits Conf.*, *Digest of Technical Papers*, IEEE Press, Piscataway, N.J., 1992, pp. 224-225.

24. R. Matick, *Transmission Lines for Digital and Communication Networks*, 3rd ed., IEEE Press, Piscataway, N.J., 1997.

25. A. Fiedler et al., "A 1.0625 Gbps Transceiver with 2x-Oversampling and Transmit Signal Pre-Emphasis," *Proc. IEEE Int'l Solids-State Circuits Conf.*, *Digest of Technical Papers*, IEEE Press, Piscataway, N.J., 1997, pp. 238-239.

26. W.J. Dally et al., "Transmitter Equalization for 4-Gbps Signaling," *IEEE Micro*, Vol.17, No.1, Jan.-Feb. 1997, pp. 48-56.

27. B-S. Song et al., "NRZ Timing Recovery Technique for Band Limited Channels," *Proc. IEEE Int'l Solid-State Circuits Conf.*, *Digest of Technical Papers*, IEEE Press, Piscataway, N.J., 1996, pp. 194-195.

28. R. Farjad-Rad et al., "An Equalization Scheme for 10Gb/s 4-PAM Signaling over Long Cables," *Proc. Mixed Signal Conf.*, IEEE Press, Piscataway, N.J., 1997.

29. P.J. Black et al., "A 1-Gb/s, Four-State, Sliding Block Viterbi Decoder," *IEEE J. Solid-State Circuits*, Vol. 32, No. 6, June 1997, pp. 797-805.

30. D. Reynolds, "A 320 MHz CMOS Triple 8 bit DAC with PLL," *IEEE J. Solid-State Circuits*, Vol. 29, No.12, Dec. 1994, p. 1,545.

31. R. Van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic, Boston, 1994.

32. M.J. Pelgrom, "Matching Properties of MOS Transistors," *IEEE J. Solid State Circuits*, Vol. 24, No. 10, Dec. 1989, pp. 1,433-1,440.

33. J. Spalding et al., "A 200M Sample/s 6b Flash ADC in 0.6mm CMOS," *Proc. IEEE Int'l Solid-State Circuits Conf.*, *Digest of Technical Papers*, IEEE Press, Piscataway, N.J., 1996, pp. 320-321.

Stefanos Sidiropoulos is a member of the technical staff at Rambus Inc. and is also a PhD candidate in electrical engineering at Stanford University. His research interests are in high-speed-circuit design and CAD tools. Sidiropoulos earned BS and MS degrees in computer science from the University of Crete, Greece.

Direct questions or comments about this article to Mark Horowitz at Gates Computer Science 3A, Rm. 306, M/C 9030, Stanford University, Stanford, CA 94305-9030; horowitz@chroma.stanford.edu.

Reader Interest Survey

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

Low 153   Medium 154   High 155

Call for Articles

IEEE Micro seeks general-interest submissions for publication in upcoming 1998 issues. These works should discuss the design, performance, or application of microcomputers and microprocessor systems. Of special interest are articles on embedded systems.

Summaries of work in progress and descriptions of recently completed works are most welcome; readers particularly appreciate practical articles that help them on the job. Readers always welcome tutorials.

Interested authors should submit a 150-word abstract or an outline to the Editor-in-Chief at the address below. Include your full contact information (all authors' names, postal and e-mail addresses, and phone and fax numbers). *Micro* does not accept submissions under consideration elsewhere.

Check *Micro*'s home page at http://computer.org/micro/ for author guidelines.

Submit to:

Stephen L. Diamond
IEEE Micro Editor-in-Chief
Picosoft, Inc.
PO Box 5032
San Mateo, CA 94402-0032
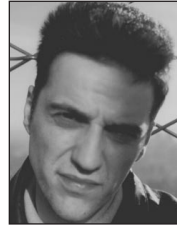s.diamond@computer.org

Mark Horowitz, the Yahoo Founders professor of electrical engineering and computer science at Stanford University, took leave in 1990 to help establish Rambus Inc. His research interest is in digital system design.

Horowitz received BS and MS degrees in electrical engineering from MIT, and a PhD from Stanford. His awards include a Presidential Young Investigator Award, an IBM Faculty Development award, and a best-paper award at the International Solid-State Circuits Conference.

Chih-Kong Ken Yang is a PhD candidate in electrical engineering at Stanford University. He received BS and MS degrees in electrical engineering from Stanford. Yang is a member of Tau Beta Pi and Phi Beta Kappa.