DELAY CIRCUITS

There are two forms in which delays can appear in circuits. First, there are inevitable delays associated with wiring and physical devices, which are not at the invitation of the designer. However, delays can also be included for very different purposes and with distinct applications. In this article, we shall describe the circuits or systems employed for generating these intentional delays. We will refer to them with the generic term of delay circuits.

A delay circuit shifts an input signal in time by a specific magnitude. In other words, the output of a delay circuit is a replica of the input, occurring a specific length of time later. In many situations arising in practice, the specifications (coming from magnitude or bandwidth, for example) are better met by cascading identical delay circuits. A delay line is so obtained. Other applications require generating a number of shifted replicas at arbitrary intervals. This is generally done by taps placed at the output of every stage of a delay line, and then a tapped delay line is obtained.

According to this definition, the class of circuits which must be considered ranges from the most simple resistance-capacitance (RC) stages to finite impulse response (FIR) or infinite impulse response (IIR) filters able to delay a discrete-time signal by a magnitude which is not a multiple of the sampling interval. Given this wide circuit scope, there is in consequence a possible overlap with the contents of other articles in this Encyclopedia. In order to minimize this overlap, the design and implementation of some of these circuits will be more extensively treated than others.

The article is structured in two well-differentiated sections, as the continuous-time and the discrete time approaches are dealt with separately. Each section has been organized in several subsections. In both cases we address mathematical modelling, system implementation, and circuit-level implementation issues. Continuous amplitude signals (analog signals) and discrete amplitude signals (digital signals) have a very distinct nature and it is well established that depending on the signal type, the implementation of delay elements follows very different circuit approaches. In consequence, we differentiate the analog and digital domain when required.

CONTINUOUS-TIME APPROACH

Delay Models

The building of a device which delays a continuous-time signal, $x_c(t)$, by an amount, t_D, as shown in Fig. 1, is conceptually simple. There is nothing physically unreasonable with such a device if t_D is positive (the response occurs after the excitation). If we only require that the response be a scaled (by k) replica without distortion of the excitation occurring t_D time units later, we can define a linear operator, L_c, which yields its output, $y_c(t)$, as:

$$y_c(t) = L_c\{x_c(t)\} = kx_c(t - t_D) \tag{1}$$

The delayed signal output response must be zero for $0 \le t < t_D$ because we analyze the behavior from $t = 0$ onward. In Eq.

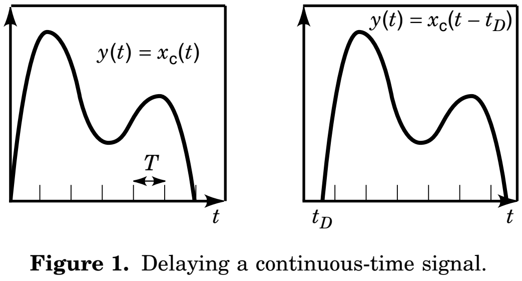

Figure 1. Delaying a continuous-time signal.

(1), k is a constant which represents amplification or attenuation, and perhaps polarity reversal.

Delaying continuous-time signals can be considered in a suitable transform domain provided by the Laplace transform. The ideal transfer function of such a device can be easily derived as:

$$\begin{aligned} Y_c(s) &= \int_0^\infty y_c(t)e^{-st}\,dt = k\int_0^\infty x_c(t-t_D)e^{-st}\,dt \\ Y_c(s) &= k\int_0^\infty x_c(\mu)e^{-s(\mu+t_D)}\,d\mu \quad \text{where} \quad \mu = t - t_D \\ Y_c(s) &= ke^{-st_D}\int_0^\infty x_c(\mu)e^{-s\mu}\,d\mu = ke^{-st_D}X_c(s) \\ H_{\text{id}}(s) &= ke^{-st_D} \end{aligned} \tag{2}$$

From Eq. (2), we obtain for ideal distortionless transmission that the transfer function $H_{\text{id}}(s)$, is $H_{\text{id}}(s) = ke^{-st_D}$. This condition is frequently more useful when expressed in the frequency domain (ω-domain) by setting $s = j\omega$. It gives $H_{\text{id}}(j\omega) = (ke^{-st_D})_{s=j\omega} = ke^{-j\omega t_D}$, which expressed in terms of modulus $|H_{\text{id}}(j\omega)|$, and argument $\arg\{H_{\text{id}}(j\omega)\} = \Theta_{\text{id}}(\omega)$ allows us to obtain two properties for the transfer function: first, a constant modulus for all frequencies is required, and second, a phase shift depending linearly on the frequency, $\Theta_{\text{id}}(\omega)$, is needed in order to provide a frequency-independent group delay $\tau_g(\omega)$:

$$\begin{aligned} &|H_{\text{id}}(j\omega)| = k && \text{constant} \\ &\begin{cases} \Theta_{\text{id}}(\omega) = -\omega t_D & \text{linear with } \omega \\ \tau_g(\omega) = -\dfrac{\partial}{\partial\omega}\Theta_{\text{id}}(\omega) = t_D & \text{constant} \end{cases} \end{aligned}$$

Two models of delays are widely used for digital signals. Pure delays shift the input signal in time by a specific magnitude t_D. This model corresponds to the mathematical operator introduced at the beginning of this section. A second useful model is the inertial delay. An inertial delay of magnitude t_D shifts the input signal in time by t_D and filters out pulses (both positive and negative) of duration less than t_D.

Delay Circuits for Analog Signals

A device implementing distortionless transmission condition cannot be a finite, lumped linear constant element network, because its transfer function is transcendental and not rational in s. Ideal transmission lines have such transfer functions and they are described by partial differential equations. Hence physical realizations of ideal delay-transfer functions do exist, although not as networks.

Good approaches for devices implementing distortionless transmission, however, can be obtained, for example, with all-pass filters (1). Table 1 shows the transfer and group delay functions for first- and second-order all-pass filters. Both filters satisfy the amplitude condition: they have an amplitude of 1 for all frequencies. The group delay condition, is only partially fulfilled: the associated group delay is only approximately constant if $(\omega/\omega_0)^2 \ll 1$ in a first-order all-pass filter, with the specific properties of this function being controlled by Q and ω_0 in a second-order all-pass filter. Hence, for band-limited signals, all-pass filters can perform as delay sections. Two or more of these sections can be cascaded to obtain delay lines. Delay lines using low-pass filters (first- or second-order follower-integrator sections) have also been proposed for specific applications (2). For such lines, the response at high frequencies will decrease sharply, in a very steep way, but in the range of application in which they are used, the group delay is approximately constant in the same conditions as the all-pass filters are.

Table 1. Transfer and Group Delay Functions for First- and Second-Order All-Pass Filters

Order	Transfer Function	Group Delay Function
First	$H(s) = \dfrac{s-\omega_0}{s+\omega_0}$	$\tau_g(\omega) = \dfrac{2/\omega_0}{1+(\omega/\omega_0)^2}$
Second	$H(s) = \dfrac{s^2 - \dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + \dfrac{\omega_0}{Q}s + \omega_0^2}$	$\tau_g(\omega) = \dfrac{2}{Q\omega_0}\dfrac{1+(\omega/\omega_0)^2}{[1-(\omega/\omega_0)^2]^2 + \dfrac{1}{Q^2}(\omega/\omega_0)^2}$

The field of filter design and implementation is the subject of many other articles in this work, so we do not deal with the circuit and physical level implementation issues here. We will just mention that time constants depend on parameters such as capacitors and transistors which are temperature- and process-dependent; therefore, some extra circuitry is required to control the delay time. Solutions to this problem resort either to control the delay time by an external voltage or, more commonly, to locking it to an external reference frequency.

Delay Circuits for Digital Signals

A different point of view is taken in many applications that require delaying digital signals. These delay circuits are a key component of phase locked loops (PLLs) and delay locked loops (DLLs), which find wide application in wireless and communication circuits, high-speed digital circuits, disk drive electronics, and instrumentation, because a number of design problems can be efficiently solved with them. The main problems that can be solved are jitter reduction, skew suppression, frequency synthesis, and clock recovery. A different and familiar application area is the control of the timing of a data sampling or of a generation process. Very fine resolution is often required in these types of applications. Very large-scale integration (VLSI) automated test equipment and time measurement systems for nuclear instrumentations are some examples. Finally, another interesting application of these delay circuits is that of self-timed circuits. Following, a brief review of the relevant issues and options for these delay circuits is presented.

Neither pure nor inertial delays can be perfectly realized in the real world. A real inertial delay of magnitude t_D, when stimulated with pulses of width varying from $t_D + e$ to $t_D - e$ for some small value $e > 0$, produces a continuum of waveforms between a pulse of width $t_D + e$ to no pulse at all. Real delays are often better modelled as combinations of the two types (3). The introduced models produce output waveforms similar to the input waveforms. This is because both rising and falling edges are propagated with the same delay. In practice, delay circuits which have different delays for each transition are useful for many applications. Also, there is another type of application for which only one transition polarity is significant. The delay is then used to initiate events at arbitrary times. The response to an input edge or pulse is a fixed-width pulse after a given time.

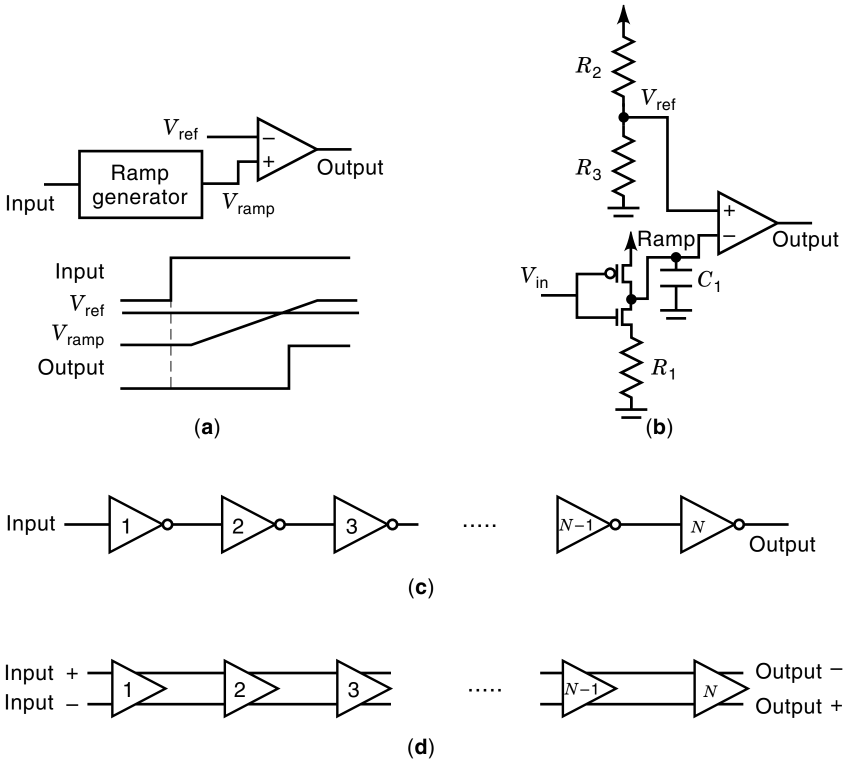

Figure 2. Generic schemes for delaying digital signals: (a) The ramp and comparator approach; (b) conventional *RC* delay element; (c) single-ended gate chain; and (d) differential gate chain.

Because of the impossibility of ideal delay elements, a set of figures of merit (4) are used to specify delay circuits in addition to nominal delay:

1. Bandwidth or maximum usable input signal frequency (In many cases it is not limited by the functional failure of the delay but by the accuracy degradation due to what is called history effects or pulse memory. The delay of a signal edge is perturbed by the presence of other edges in the recent past.)
2. Tolerance of nominal delay
3. Temperature coefficient of nominal delay
4. Voltage coefficient of nominal delay
5. Jitter or random variation in the delay of different edges due to noise (It can be evaluated by maximum error or by the standard deviation of errors.)

Basically, continuous digital signals can be delayed using passive transmission lines which behave roughly as a pure delay element, *RC* delays or logic gates. Transmission lines can be used at the PC board and hybrid levels. Here, we focus on delay circuits to be included with other elements within an integrated circuit (IC). Figure 2 shows different generic ways of realizing delays together with common ways of implementing their building blocks. The ramp and comparator approach is depicted in Fig. 2(a). A transition of the input signal makes a voltage ramp, V_{ramp}, start from a stable initial level. The ramp and a control voltage, V_{ref}, are applied to the inputs of a high-speed comparator, which switches at a time proportional to the control level. Fig. 2(b) shows a conventional *RC*-delay circuit for implementing the approach just described. When the input signal V_{in} rises, the node Ramp starts to discharge through R_1. The simplest method for delaying a digital signal, an *RC* stage with input and output buffers, can be viewed as a particular case of this generic scheme. The reference voltage is now the threshold voltage of the output buffer. Another variation substitutes the comparator for a monostable circuit.

Logic gates are an obvious and almost universally available delay medium. Chains of logic gates are widely applied to delay signals as shown in Figs. 2(c) and 2(d). The nominal delay depends both on the delay of each cell and on the number of stages. When single-end gates are the basic cell of the chain, inverting gates are usually used in order to balance the propagation time of rising and falling edges which can affect the accuracy of the delay element. Differential circuits techniques [Fig. 2(d)] are extensively used for several reasons. First, differential gates achieve a high rejection of noise; secondly, they reduce edge dependency. In general, reduced swing differential circuit techniques are a good choice because they also allow maximize bandwidth. Finally, the functionality of the basic delay cell can be other than that of a buffer, depending on the application. For example, exclusive OR (XOR) gates and latches have been used in pattern generation and time measurements systems (5,6).

Programmable or adjustable delays, that is, delays depending on one or more control inputs, are interesting for many reasons. Devices that can satisfy a wide range of applications and that can be of manual or automatic calibration are the main ones. Control inputs can be digital or analog. The specification of these delays requires additional variables such as range of available delays, resolution, tolerance and stability of range, and linearity (4).

There are three different strategies for realizing controllable delays which are summarized in Fig. 3. The first one consists of selecting one of several fixed delay paths between in-

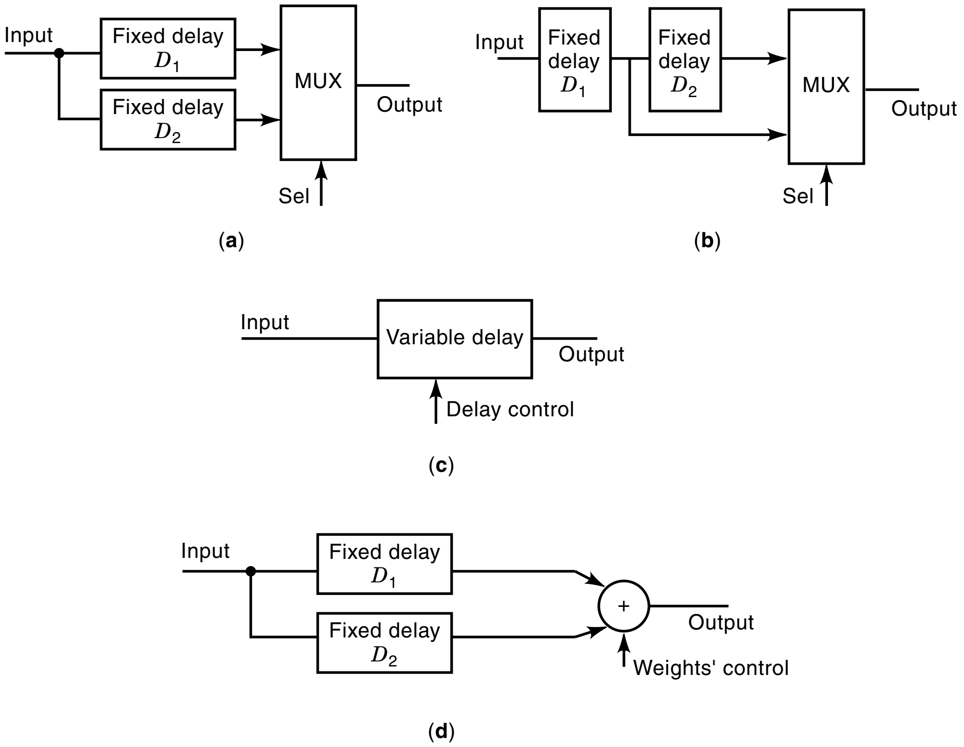

Figure 3. Generic methods for realizing variable delays: (a) Selectable path approach, parallel implementation; (b) selectable path approach, serial implementation; (c) variable delay fixed path approach; (d) delay interpolation.

put and output. Figures 3(a) and 3(b) show two possibilities. Care must be taken to match the delays in all paths in the selector logic. This can be critical if selectable delays differ little or in certain applications. In the second approach [Fig. 3(c)], the physical path of the signal remains constant but its delay is varied. A third approach, called delay interpolation, uses two paths with different delays in parallel. The total delay is an adjustable weighted sum of the delays of the two paths as shown in Fig. 3(d).

In Fig. 3(c), the delay is tuned by the control input. The V_{ref} input of the ramp and comparator delay circuit can be used as the control input in order to build a variable delay element. This approach is popular because of its flexibility and naturally linear response. A different strategy is used in the circuit shown in Fig. 4(a) (2), where the control input, V_{ctrl}, affects the slew rate at node V_1 instead of the triggering voltage. Starting with V_{out} low, when V_{in} becomes active, current I_{in} charges C_1 until V_{out} triggers. The delay is inversely proportional to I_{in} which depends on V_{ctrl}. This element has been used in an adaptative delay block which exhibits a high pull-in frequency range partially because the transistor M_1 operates in the subthreshold region (7).

In gate-based delay generators, control can be accomplished through capacitive tuning or resistive tuning of the basic delay stages which form the chain. Figures 4(b)–4(e) show different methods of implementing capacitive-tuned variable-delay elements. These techniques vary the capacitance at the output node. The generic circuit shown in Fig. 4(b) uses a voltage-controlled resistor to control the amount of effective load capacitance seen by the driving gate. Figure 4(c) shows a metal oxide semiconductor (MOS) implementation of this approach (8,9). Transistor M_2 has its source and drain shorted together forming an MOS capacitor. The effective capacitance is larger as V_{ctrl} increases and the resistance of transistor M_1 is reduced. It can exhibit a poor linearity if a wide delay range is required. Figure 4(d) employs a voltage-controlled capacitor. This capacitor can be a reversed biased p–n junction diode. A different option with digital control (10) is depicted in Fig. 4(e). Node OUT is loaded by m pairs of p–n load devices. When the ith enable line is low, the capacitance load that the pair of devices present to node OUT is minimal because inversion layer cannot be formed for any voltage on OUT. When the ith enable line is high, the capacitive load that the pair presents is maximal, because an inversion layer can be formed under the gate of one or both of the p–n devices.

Resistive tuning approaches use variable resistances to control the current available to charge and discharge the load capacitance. A classical example is the current-starved inverter shown in Fig. 4(f). Voltage V_{ctrl} controls the ON resistance of transistor M_1 and through a current mirror, the transistor M_2. Delay decreases as V_{ctrl} increases, allowing a large current to flow. A Schmitt trigger followed by a buffer can be included to achieve fast rising and falling outputs (11). If a simple current mirror is used, the delay is a very nonlinear function of the control voltage. Moreover, its high gain coefficient (steep slope of delay characteristic) makes it sensitive to noise on the control voltage line. The linearity of the delay characteristic in the current-starved inverter can be improved by using more complex current mirror configurations (12).

Resistive tuning of differential delay elements is also possible. Other parameters such as the logic swing or dc gain are controlled in addition to the effective load resistance. Figure 4(g) depicts the generic structure of a number of reported circuits (5,13–15). Clearly, the delay of the generic differential gate can be changed with the voltage V_{c1}, since the effective resistance of the load changes with this control voltage. Also,

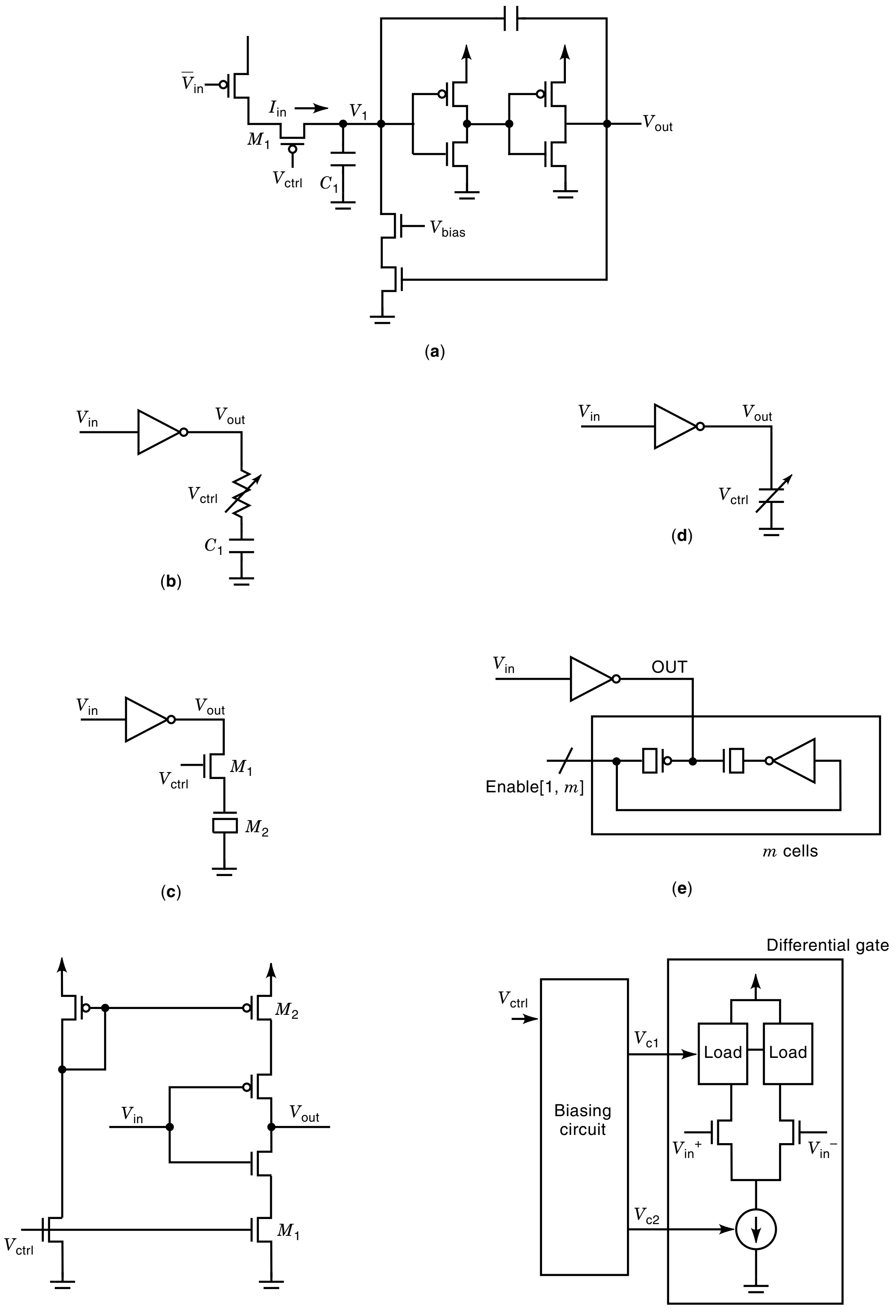

Figure 4. Voltage-controlled variable delay elements: (a) Variation of the ramp-and-comparator technique; (b) capacitive tuning using voltage-controlled resistor; (c) MOS implementation for (b); (d) capacitive tuning using voltage-controlled capacitance; (e) digitally controlled MOS implementation for (d); (f) conventional resistive tuning circuit or current-starved inverter; (g) resistive tuning of differential gates.

V_{c2} can vary the delay of the gate as it adjusts the tail current. The biasing circuit generates V_{c1} and V_{c2} from V_{ctrl}. One of the two voltages, V_{c1} or V_{c2}, may be nominally equal to V_{ctrl}. The biasing block produces the appropriate value for the other bias in order to control the parameters previously mentioned. This can be done by implementing the replica biasing concept in which a copy of the delay cell, a differential amplifier, and feedback are used. Also, the noise insensitivity of the delay cell is improved with this technique, as the appropriate bias voltage values are generated independently of supply voltage variations. Finally, resistive tuning allows fully differential approaches. That is, the control path is also differential. Partially because of this feature, resistive tuning has been identified as the most suitable for implementing voltage-controlled oscillators (VCOs) (16).

An important consideration in designing accurate delay elements is to compensate for variations in process, temperature, and supply voltage. Some of the delay circuits described use an *RC* time charge constant and generate delays almost independent of MOS transistor characteristics. The delay deviations due to these ambient and process conditions are lower than those of a chain of single-ended conventional inverters (17). This sensitivity has been further reduced by making the charging current proportional to the reference voltage, V_{ref} (18). Thus, even if the reference voltage fluctuates as the result of supply-voltage, temperature, and device parameter variations, the current charging the capacitor compensates it, so the delay is constant.

In general, there are several ways to improve the stability of delay circuits. Actions can be taken at different levels in order to achieve the desired stability. In the architectural domain, a useful approach is to use adjustable delay elements and continuously control them with a feedback mechanism. Phase and delay locked loop techniques have been widely used. For example, a DLL can be used to maintain the accuracy of a chain of delay elements through which a periodic signal (clock) is propagating (5,6,8,9). The effect of the DLL approach is that two taps of the voltage-controlled delay line (VCDL) driven by a clock reference are examined, and the delay element control voltage (V_{ctrl}) is adjusted until the two taps are in phase. Different delay values within the range of the delay elements can be maintained with different clock frequencies and different selections of the taps. This concept has been applied to tuning in production, calibration, and active delay regulation. In some cases, a pair of matched VCDLs that depend on a common V_{ctrl} are used. A feedback circuit can make the delay of a reference chain match an external time signal. The second line is the functional delay generator which is also stabilized. Physical design techniques which can reduce the effect of process and temperature gradients within the chip, on-chip voltage, and temperature regulation and optimization of the gates for delay insensitivity can also be considered.

Noise is common to all electrical systems and it appears in digital circuits primarily as timing jitter. Jitter can be reduced by careful application of the standard noise decoupling and isolation techniques: guard rings, for example. Also, the use of differential circuits and replica biasing circuits helps reduce the sensitivity to noise.

Delays can be combined in various ways in order to extend the range or the resolution. A serial connection of a selected-path delay and a constant-path variable-delay stage may have a wide range and fine control of rising and falling delays. Other schemes can be used to improve the resolution of a chain of delay elements, which is limited to the basic delay of one or two (if single-ended, inverting gates are used) of its stages. They include delay interpolation performing an analog sum of consecutive taps. Precise delay interval generators with subgate resolution have been proposed based on a series of coupled ring oscillators (19) and using an array of similar DLLs with a small phase shift between them (12).

DISCRETE-TIME APPROACH

Delay Models

Discrete-time signals are obtained by sampling a continuous-time signal at discrete times [Fig. 5(a)] or they are directly generated by a discrete-time process. Delaying a uniformly sampled bandlimited (baseband) signal presents several major differences when compared with the continuous time addressed previously. If we simply convert Eq. (1) into discrete time by sampling the continuous signal at time instants $t = nT$, where n is an integer and T is the sampling interval, then we obtain:

$$y[n] = L\{x[n]\} = kx[n - D] \tag{3}$$

If D is an integer (when t_D is a multiple of the sampling interval), the output value is one of the previous signal samples,

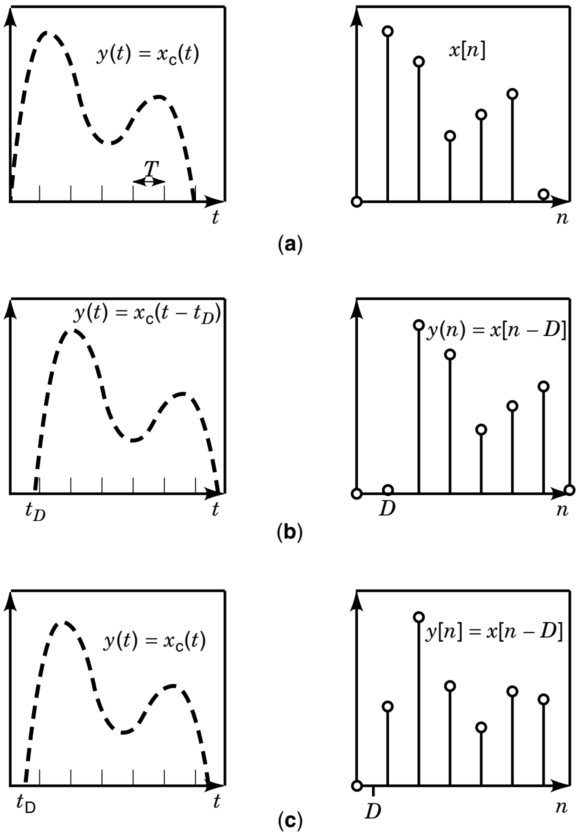

Figure 5. Delaying a discrete-time signal: (a) Sampling a continuous-time signal at discrete times; (b) delaying a discrete-time signal by an integer D; (c) delaying a discrete-time signal by a noninteger D.

and consequently, we have a delay of D samples [Fig. 5(b)]. But if D is not an integer, Eq. (3) has no formal meaning because the output value would lie somewhere between two samples, and it is impossible [Fig. 5(c)]. Other important differences with the continuous time problem are related to the necessity of clocking and the occurrence of aliasing effects.

In a similar way to the continuous-time case, delaying discrete-time signal can be considered in a suitable transform domain: the z-domain. An ideal transfer function in this domain can be obtained formally as:

$$\begin{aligned} Y(z) &= k\sum_{n=-\infty}^{\infty} x[n-D]z^{-n} \\ Y(z) &= k\sum_{m=-\infty}^{\infty} x[m]z^{-(m+D)} \qquad \text{where}\, m = n - D \\ Y(z) &= kz^{-D}\sum_{m=-\infty}^{\infty} x[m]z^{-m} = kz^{-D}X(z) \\ H_{\mathrm{id}}(z) &= kz^{-D} \end{aligned} \tag{4}$$

which strictly holds only for integer values of D. The term kz^{-D} represents an ideal discrete-time delay system in the z-domain, which performs the bandlimited delay operation at the specified sampling rate.

As the specifications are usually given in the frequency domain, it is interesting to obtain the response frequency (Fourier transform) of the ideal delaying system we are concerned with. It is determined from Eq. (4) by setting $z = e^{j\omega}$, where $\omega = 2\pi fT$ is the normalized angular frequency, which give us $H_{\mathrm{id}}(e^{j\omega}) = ke^{-j\omega D}$. This system has constant magnitude response, linear phase, and constant group delay:

$$\begin{array}{ll} |H_{\mathrm{id}}(j\omega)| = k & \text{constant} \\ \Theta_{\mathrm{id}}(\omega) = -\omega D & \text{linear with}\, \omega,\ |\omega| < \pi \\ \tau_g(\omega) = -\dfrac{\partial}{\partial\omega}\Theta_{\mathrm{id}}(\omega) = D & \text{constant in the whole frequency band} \end{array}$$

with periodicity 2π in ω assumed.

The inverse Fourier transform of $H_{\mathrm{id}}(e^{j\omega})$ is the impulse response. In case of a delay D taking an integer value, the impulse response is a single impulse at $n = D$: that is, $h_{\mathrm{id}}[n] = k\delta[n - D]$, where $\delta[\cdot]$ is the Kronecker delta function. The system simply shifts (and scales by k) the input sequence by D samples:

$$y[n] = x[n] * h_{\mathrm{id}}[n] = x[n] * k\delta[n-D] = kx[n-D] \tag{5}$$

When D is a noninteger value, appropriate values of $y[n]$ on the sampling grid must be found via bandlimited interpolation. This problem has a straightforward interpretation as a resampling process: the desired solution can be obtained by first reconstructing the bandlimited signal, shifting it, and finally resampling it.

To obtain the impulse response corresponding to a system able to give us the frequency response required, we use the inverse discrete-time Fourier transform:

$$\begin{aligned} h_{\mathrm{id}}[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{\mathrm{id}}(e^{j\omega})e^{j\omega n}\, d\omega \quad \text{for all}\, n \\ h_{\mathrm{id}}[n] &= \frac{k}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega D}e^{j\omega n}\, d\omega \end{aligned}$$

and so, the ideal impulse response is obtained as:

$$h_{\mathrm{id}}[n] = k\frac{\sin[\pi(n-D)]}{\pi(n-D)} \qquad \text{for all}\, n \tag{6}$$

The impulse response in Eq. (6) is now an infinitely long, shifted, and sampled version of the sinc function. We have here a fundamental difference with the continuous-time approximation problem. Causal continuous-time delays are always causal and bounded input-bounded output (BIBO) stable whereas in the discrete-time problem for fractional sample delays, neither of these properties hold: $h_{\mathrm{id}}[n]$ is noncausal and is not absolutely summable. This noncausality makes it impossible to implement it in real-time applications.

The output of the system for an input $x[n]$ can be formally obtained as:

$$\begin{aligned} y[n] &= x[n] * h_{\mathrm{id}}[n] = x[n] * \left(k\frac{\sin[\pi(n-D)]}{\pi(n-D)}\right) \\ y[n] &= k\sum_{l=-\infty}^{\infty} x[l]\,\frac{\sin[\pi(n-l-D)]}{\pi(n-l-D)} \end{aligned} \tag{7}$$

that is, input samples spread over all the discrete-time values weighted by appropriate values of the sinc function. Results obtained in Eq. (7) have important consequences: ideal fractional delays are impossible to implement and any system intending to do an emulation of this delay must be alike to the ideal response in some meaningful sense. Ideal fractional delays can be approached by using finite-order causal FIR or IIR filters. An excellent tutorial on fractional delays can be found in Ref. 20.

Unit Delay Circuits

This section is mainly devoted to the implementation of the z^{-1} term, identified in the previous section with the bandlimited unit delay operation at the sampling rate we are interested in. This term is a basic block in the realization of any discrete delay. In the case of integer delays, z^{-N} can be implemented by cascading N unit delay elements. In case of a fractionary delay, this must be approximated by a filter whose realization also needs these integer delays (besides arithmetic elements).

Analogous to the continuous-time case, approximations to the implementation of the z^{-1} term depend on the type of application we are interested in. Thus, digital and analog approximations will be treated separately.

Digital Implementations. Delays are realized using digital storage devices or memory cells to store data during a sampling period. There are different ways of implementing these delay operators depending on both architectural and circuit choices.

From an architectural point of view, a widely used approach for implement a delay line of N clock cycles employs a shift register. A shift register is a linear array of storage devices, such as flip-flops or latches, with the capability of exe-

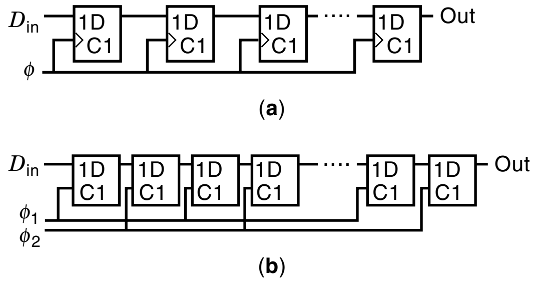

Figure 6. Implementation of z^{-N} with shift register: (a) one-phase clock; (b) two-phase clock.

cuting right shifts of one position. Figure 6 shows shift register structures for different clocking strategies. The one in Fig. 6(a), employs flip-flops as basic units in a one-phase clock scheme. In Fig. 6(b), the architecture when using a two-phase clock scheme and latches is shown. Data present at the D_{in} input of the registers (Fig. 6) will be available in the output OUT after N cycles and so OUT$[n] = D_{in}[n - N]$ as required.

We briefly summarize the different available approaches to the circuit realization of memory cells. An excellent treatment can be found in Ref. 21. The memory cells can be implemented as static or dynamic circuits. The first approach uses positive feedback or regeneration. That is, one or more output signals are connected to the inputs. A second approach uses charge storage as a means of storing signal values. This approach, which is very popular in MOS designs, has the disadvantage that the charge tends to leak away in time. Thus, there are restrictions on the sampling frequency used: it must be high enough so that the state is not lost. Figure 7 shows several CMOS memory cells suitable for register architectures. Note that flip-flops suitable for one-phase register architectures can be realized by cascading two latches operating on complementary clocks, in what is called a master–slave configuration.

The circuit depicted in Fig. 7(a) is a static latch. It consists of a cross-coupled inverter pair. The extra transistors are used to store the value of D_{in} when the clock ϕ is high. Let us consider the case when Q is high and D is zero: in this situation with ϕ high, and the appropriate sizing of transistors M_1, M_2, and M_3, Q is brought below the threshold of the inverter M_5–M_8. Then, the positive feedback forces Q to be zero. Although these latches have reduced noise margins and require careful design, they are small and can be very fast.

In Fig. 7(b) a pseudostatic latch is shown. The feedback loop is closed when ϕ is high. In this mode, the circuit behaves as a biestable element. When the clock ϕ goes high, the loop opens and the input value is stored in the internal capacitor. It is called pseudostatic because frequently ϕ_1 and ϕ_2, as shown in the figure, are used to control the pass transistors in order to avoid overlapping of both phases even if clock routing delays occur. During ϕ_{12} the circuit employs dynamic storage.

A fully dynamic approach is less complex, as illustrated in Fig. 7(c). Only three transistors are required to implement a latch. Possible variants for circuits in Figs. 7(b) and 7(c) include using complementary transmission gates instead of NMOS pass transistors. Also, versions of these latches can be built adding level-restoring devices, as illustrated in Fig. 7(d).

Figure 7(e) shows the C^{2}MOS latch. This circuit operates in two modes. With ϕ high, it is in the evaluation mode because it acts as an inverter (transistors M_3 and M_4 are ON). With ϕ low it is in the hold or high-impedance mode and so $\overline{Q}$ retains its previous value stored in the output capacitor, C_L. This structure presents advantages over both the pseudostatic and the fully dynamic latches. These two require the availability of two nonoverlapping clocks (four if complementary transmission gates are used) for correct operation of a cascaded configuration. Ensuring the nonoverlapping condition might involve making ϕ_{12} large, which has a negative impact on circuit performance, or generating the required clocks locally, which increases area. The operation of a cascaded pair of C^{2}MOS latches controlled by ϕ_1 and ϕ_2, respectively, is insensitive to overlap as long as the rise and fall times of the clock edges are small enough. The C^{2}MOS latch is useful for high speed as, in that case, it is hard to avoid clock overlap.

Finally, memory elements with a single clock have also been proposed. Figure 7(f) shows a single clock version of the circuit depicted in Fig. 7(e). With ϕ high, it corresponds to a cascade of two inverters and so it is transparent. With ϕ low, no signal can propagate from its input to its output. This circuit is called a true single-phase clock latch (TSPC latch) and is the basis for the TSPC logic design methodology. Figure 7(g) depicts a positive edge-triggered flip-flop built using p versions of the TSPC latch in Fig. 7(f).

Another approach to implement a delay line is based on a multiport random access memory (RAM) which is used to simulate a shift register. The selection of the most convenient technique (shift-register or multiport RAM memory) depends on the application.

Analog Implementations. We must essentially consider two approaches to analog discrete-time signal processing: switched-capacitor (SC) and switched-current (SI) techniques. Switched-capacitor techniques are extensively used in mixed-mode designs and SC delay-stages circuits have found applications in the implementation of sampled analog filters based on digital filter architectures as well as in the realization of interpolators and decimators. The high-quality capacitors needed are generally implemented using two polysilicon layers. Recently, the SI technique has appeared as an alternative to SC techniques that is fully compatible with digital CMOS standard processes. More details about these techniques can be found in Refs. 22 and 23.

Switched-Capacitor Techniques. If SC techniques are employed, delay elements can be realized in a simple way: by cascading two sample-and-hold (S/H) elements provided there are complementary clocking phases. If the output is sampled at the clock phase ϕ_1 and the input signal at the beginning of the phase ϕ_1, then it is possible to use only one S/H element.

Figure 8(a) shows the simplest configuration of an S/H element in which a voltage signal v_{in} is sampled and held in a linear capacitor C_h through the switch controlled by clock ϕ. Noise and unbalanced charge injection are the major sources of error in this configuration, and some compensatory techniques can be employed to reduce the switch-induced error.

The signal source can be isolated from the capacitor load by using an op-amp as a voltage follower. Avoiding any loading of the holding capacitor by an output circuit can be realized in a similar way. Configurations following this idea are sensitive to the offset voltages of the amplifiers. A feed-

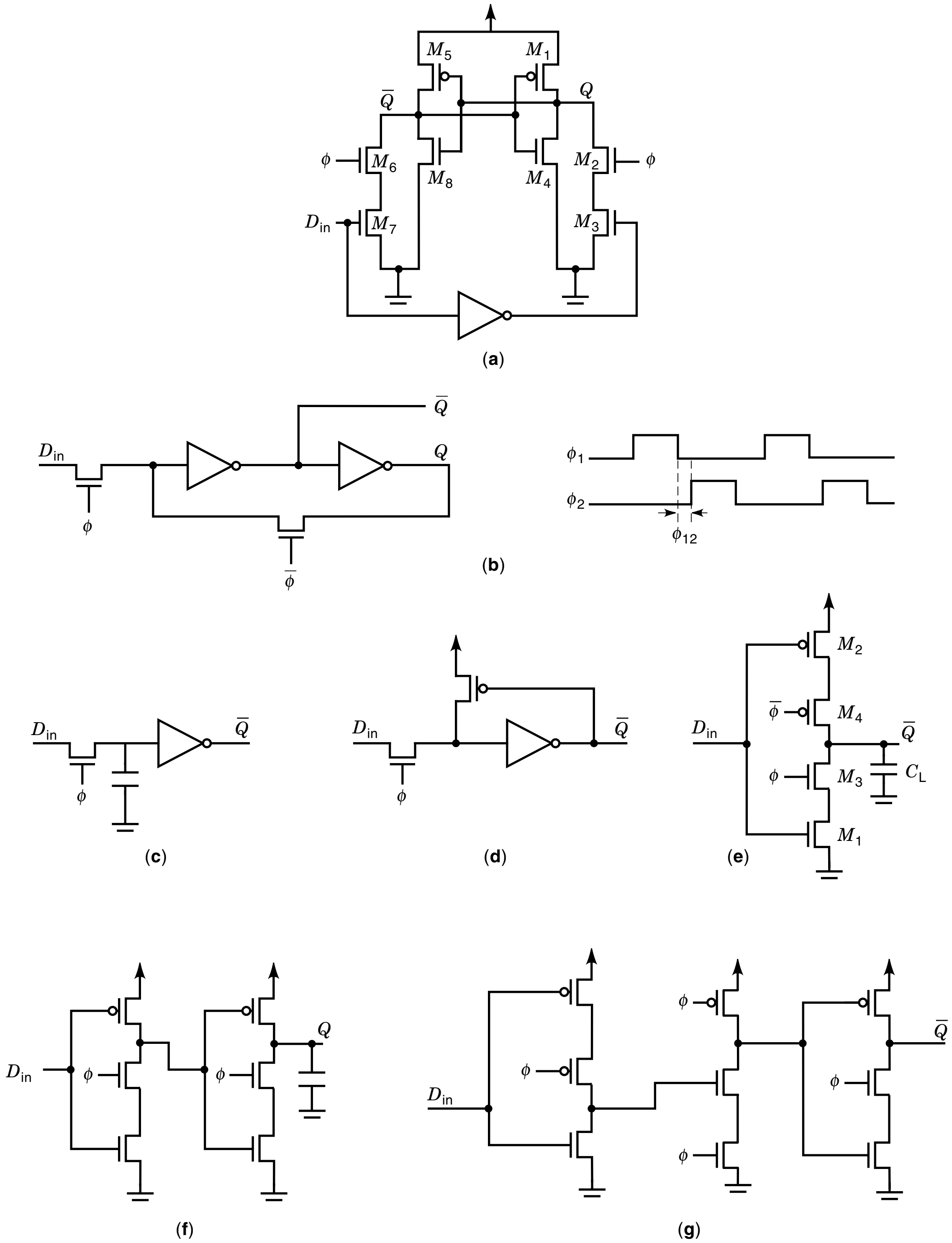

Figure 7. CMOS memory cells suitable for register architectures: (a) Static latch; (b) pseudostatic latch; (c) fully dynamic latch; (d) fully dynamic latch with level restoring device; (e) C^2MOS; (f) TSPC latch; (g) positive edge-triggered TSPC flip-flop.

back loop around the hold capacitors can be used to reduce this offset error, as shown in Fig. 8(b), where offset and common mode error of the output follower are reduced by the gain of the first op-amp. However, the offset of the first op-amp appears at the output. Further improvements in both speed and accuracy are obtained in the configuration shown in Fig. 8(c), when the second op-amp is connected as an integrator.

Figure 8(d) shows a configuration with an autozeroing feature which can be used to solve the problems related with the offset voltage. This S/H circuit also has unity-gain and is offset free and parasitic capacitance insensitive. Another inter-

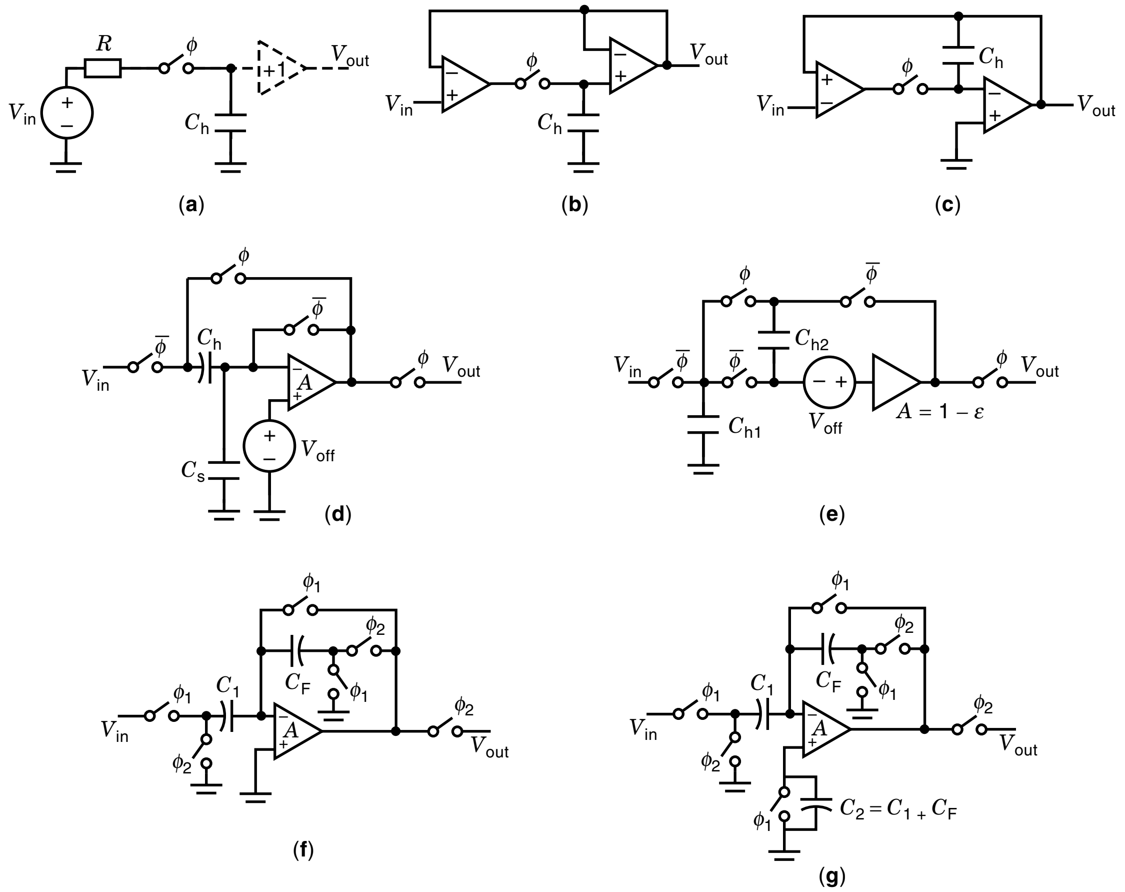

Figure 8. Switch capacitor configurations: (a) Elementary sample-and-hold (S/H) element; (b) S/H amplifier configuration with feedback loop; (c) S/H amplifier configuration with integrator; (d) offset and parasitic free unity gain S/H stage; (e) offset- and gain-compensated unity gain S/H stage; (f) basic configuration S/H gain stage; (g) switch-induced error compensated S/H gain stage.

esting configuration is shown in Fig. 8(e), where the voltage amplifier has approximate unity-gain (ϵ denotes the gain error) and an auxiliary hold capacitor C_{h2} is used to provide compensation of the gain and the offset voltage of the amplifier. Both structures use a technique usually known as correlated double sampling (CDS): the offset voltage of the op-amp is *measured* in one of the clock phases, stored in the capacitors, and then substracted in the subsequent signal amplification clock phase. This technique eliminates the offset voltage and additionally reduces the low-frequency 1/*f* noise and the power supply noise. Switch-induced errors and other parasitic effects such as power supply noise and common-mode signals can be drastically reduced by employing fully differential configurations.

These configurations provide unity gain. If an S/H stage with arbitrary positive gain is required, we can resort to the circuit shown in Fig. 8(f), which also uses the CDS technique. Assuming there is an infinity op-amp gain, the circuit operation is as follows: During clock phase ϕ_1, it operates as a unity-gain voltage follower (both inverting input and output are short circuited). Capacitors C_F and C_1 are charged to the offset voltage and to the input voltage minus the offset voltage, respectively. Next, during clock phase ϕ_2, the capacitor C_1 is discharged through C_F, giving an output voltage which is independent of the op-amp input offset voltage. Improvements in eliminating the switch-induced voltage error at the expense of doubling the total amount of required capacitance can be obtained with the configuration shown in Fig. 8(g), where CDS technique has been again applied. It adds an appropriate network (capacitor C_2 and switch controlled by ϕ_1) to the noninverting input of the op-amps in order to cancel the signal injected at the inverting terminal by the clock feedthrough. However, a switch-induced voltage error remains, which is determined by the mismatching of the switches and capacitors and the common mode rejection ratio (CMRR) of the op-amp.

To obtain an analog delay line of *N* clock periods we only need to connect in cascade *N* delay elements. The cascading of delay elements transfers errors due to such effects as gain mismatch, offset voltage, or clock feedthrough from stage to stage, accumulating them and limiting the maximum possible number of S/H stages in the cascade.

Another approach employs a parallel of *N* S/H elements rather than a cascade implementation, as shown in Fig. 9. It is composed of *N* channels, each one containing an S/H stage with a unity gain buffer and an array of switches controlled by the clock sequence shown in the figure. The S/H stages sequentially sample the input signal and hold

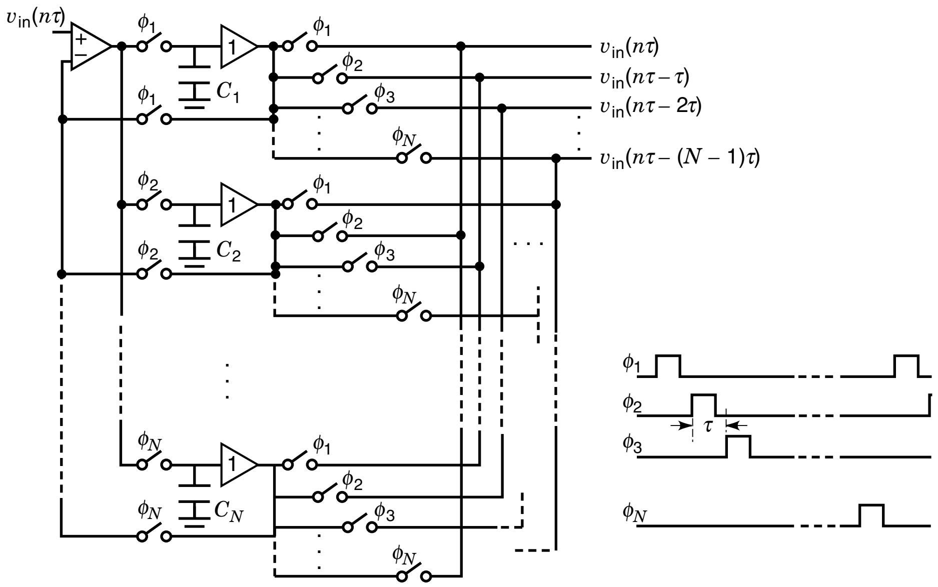

Figure 9. SC delay line and clock sequence controlling it.

it for the next N clock cycles: thus, the errors are added only once. Errors caused by the unity gain buffer are minimized by connecting the S/H stages in a feedback loop of a single time-sharing op-amp. Errors in the S/H stages are greatly reduced because they are divided by the gain of the op-amp. Errors due to the offset voltage and the finite gain of the op-amp are not compensated but they likewise affect all the outputs.

Switched-Current Techniques. In SI techniques, delay elements are simply made by cascading memory cells. Topologies used for delay elements are included in one of two categories: the current-mode track-and-hold (T/H) and the dynamic current mirror. The current-mode T/H delay is shown in Fig. 10(a). A digital clock signal switches ON and OFF switch S, which, when ON, shorts the gates of transistors T_1 and T_2, and the circuit functions as a current mirror: with an input i_{in} applied to the drain of T_1, the output i_{out} tracks such input current. When the switch is turned off, the gates of T_1 and T_2 are disconnected, and the gate voltage of T_1, corresponding to the input current value in this moment, is sampled on C_{gs2}. Voltage V_{gs2} remains constant while the switch is open, and so the output current is held at a constant value which is the input current value in the instant when the switch was opened.

An important drawback of this circuit refers to the exact reproduction of the input current at the output: it depends on the matching of the two transistors T_1 and T_2 and the two bias current sources J_1 and J_2. This disadvantage is solved by the second generation of memory cells, the dynamic current mirror or *current copier,* by using only one transistor for both input and output of current, as explained in the following.

The conventional SI memory cell is shown in Fig. 10(b). It can achieve current memory in the transistor T_1 when driven by the clock waveforms of Fig. 10(c). Its operation is as follows: on phase ϕ_1, switch S_1 is closed and current i_{in} adds to the bias current J flowing into the circuit. Current $J + i_{in}$ begins to charge the initially discharged capacitor C_{gs1}. As C_{gs1} charges, the gate-source voltage V_{gs} of T_1 increases and when it exceeds its threshold voltage, T_1 conducts. Eventually, when C_{gs1} is fully charged, all of the current $J + i_{in}$ flows in the drain of T_1. On phase ϕ_2, switch S_1 is opened and the end value of V_{gs} when ϕ_1 finishes is held on capacitor C_{gs1} and it sustains the current $J + i_{in}$ flowing in the drain of T_1. As the input switch is open and the output one closed, there is a current imbalance which forces an output current, $i_{out} = -i_{in}$, to flow throughout phase ϕ_2.

A delay cell comprises two cascaded current memory cells with the phase reversed on alternate memory cells. A delay line of N clock periods could be generated by cascading N delay cells ($2N$ memory cells), as shown in Fig. 10(d). Another approach uses an array of $N + 1$ memory cells in parallel, as shown in Fig. 10(e). By using the clock sequence shown in the figure, on clock phase ϕ_i, memory cell M_i acquires the input current and memory cell M_{i+1} releases its output, for $i = 0, \ldots, N - 1$. On phase ϕ_N, cell M_N receives its input signal and cell M_0 delivers its output.

Actual operation of the basic memory cell deviates from the ideal behavior due to transistor nonidealities which degrade the performance of the cell. Previous structures for the delay line inherit these problems: if the memory cell has transmission errors (which occur through conductance ratio errors and charge injection errors) or signal-to-noise ratio errors, then the serial solution increases both by a factor of $2N$; in the other solution, errors are the same as those of the memory cell, but two extra problems arising from the parallel nature of the structure can be found. One problem comes from unequal path gains and the other from nonuniform sampling. The degree of importance of both problems is different: very close path gains are obtained as transmission accuracy is not achieved by component matching. Nonuniform sampling could be carefully considered if the cell is used for the sample and hold function. An additional drawback of that structure results from leakage which discharges C_{gs} during the N clocks between sampling and output.

Some of these error sources can be controlled by a precise choice of transistor sizes and currents, in particular those coming from mismatching, charge injection, conductance ra-

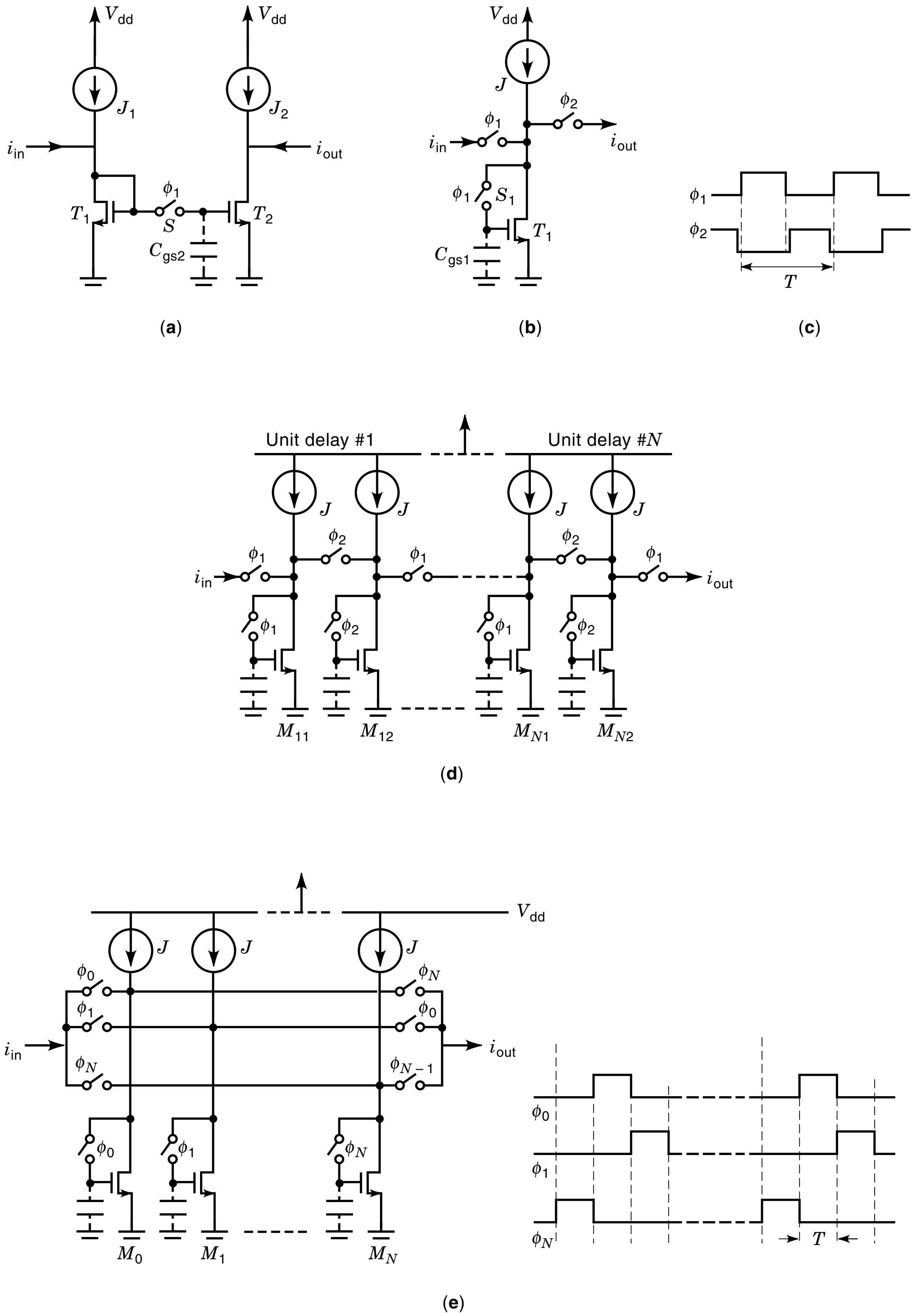

Figure 10. Switch current (SC) configurations: (a) Track-and-hold delay; (b) memory cell with a single transistor; (c) clock waveforms; (d) serial delay line; (e) parallel delay line.

tios, settling, and noise. However, if we are interested in achieving a performance in terms of precision, dynamic range, and linearity, which is competitive with state-of-the-art SC circuits, a different approach must be taken by resorting to circuit techniques.

At present, main circuit techniques use either negative feedback techniques or fully differential structures. Feedback techniques are specially indicated to reduce conductance ratio errors and increase dynamic range. Two groups can be considered depending on how negative feedback is applied in the memory cell. The first one includes the op-amp and the grounded-gate active memory cells, which use feedback to increase the input conductance by the creation of a *virtual earth* at the input. Op-amp option can make monotonic settling difficult to achieve, and this behavior is improved if the op-amp is substituted by a grounded-gate amplifier. Conductance error improvement is similar but the dynamic range is a little better than that of the basic memory cell. In the second group, negative feedback is used to decrease the output conductance. Monotonic settling is also achieved in simple and folded cascodes, but it may be necessary to use of compensation in regulated cascode memory cells. More details can be found in Ref. 22.

Fully differential structures are able to reduce errors coming from charge injection and improve noise immunity. Compared with basic cells of the same supply voltage and current, much lower charge injection errors and similar bandwidth, dynamic range, and chip area are obtained with fully differential cells designed with half-width transistors. Additionally, any of the cascode variations previously cited may be used with this approach.

Finally, if quiescent power consumption is a basic concern, then the class AB SI technique can be indicated, because an important reduction of it is obtained. Additionally, delay cells using class AB memory cells are generated without duplicating the entire cell; instead the output of the first stage is simply cross-coupled to a second stage of memory cells. Simulation results in some applications have shown how charge injection can produce significant errors.

BIBLIOGRAPHY

1. K. Bult and H. Wallinga, A CMOS analog continuous-time delay line with adaptative delay-time control, *IEEE J. Solid-State Circuits,* **23**: 759–766, 1988.
2. C. Mead, *Analog VLSI and Neural Systems,* Reading, MA: Addison-Wesley, 1989.
3. S. Unger, *The Essence of Logic Circuits,* 2nd ed., Piscataway, NJ: IEEE Press, 1997.
4. R. Feldman and D. Rosky, A step-by-step guide to programmable delays, *Electron. Design,* **39**: 97–98, 100, 102–104, 107, 1991.
5. G. C. Moyer et al., The delay vernier pattern generation technique, *IEEE J. Solid-State Circuits,* **32**: 551–562, 1997.
6. T. E. Rahkonen and J. T. Kostamovaara, The use of stabilized CMOS delay lines for the digitization of short time intervals, *IEEE J. Solid-State Circuits,* **28**: 887–894, 1993.
7. S.-Ch. Liu and C. Mead, Continuous-time adaptive delay system, *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing,* **43**: 744–751, 1996.
8. M. Bazes, A novel precision MOS synchronous delay line, *IEEE J. Solid-State Circuits,* **20**: 1265–1271, 1985.
9. M. G. Johnson and E. L. Hudson, A variable delay line PLL for CPU-coprocessor synchronization, *IEEE J. Solid-State Circuits,* **23**: 1218–1223, 1988.
10. M. Bazes, R. Ashuri, and E. Knoll, An interpolating clock synthesizer, *IEEE J. Solid-State Circuits,* **31**: 1295–1301, 1996.
11. D. K. Jeong et al., Design of PLL-based clock generation circuits, *IEEE J. Solid-State Circuits,* **22**: 255–261, 1987.
12. J. Christiansen, An integrated high resolution CMOS timing generator based on an array of delay locked loops, *IEEE J. Solid-State Circuits,* **31**: 952–957, 1996.
13. J. G. Maneatis, Low-jitter process-independent DLL and PLL based on self-biased techniques, *IEEE J. Solid-State Circuits,* **31**: 1723–1732, 1996.
14. M. Mizuno et al., A GHz MOS adaptive pipeline technique using MOS current-mode logic, *IEEE J. Solid-State Circuits,* **31**: 784–791, 1996.
15. I. A. Young, J. K. Greason, and K. L. Wong, A PLL clock generator with 5 to 110 MHz of lock range for microprocessors, *IEEE J. Solid-State Circuits,* **27**: 1599–1607, 1992.
16. B. Razavi, Design of monolithic phase-locked loops and clock recovery circuits—A tutorial, in B. Razavi (ed.), *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design,* Piscataway, NJ: IEEE Press, 1996.
17. Y. Watanabe et al., A new CR-delay circuit technology for high-density and high-speed DRAMs, *IEEE J. Solid-State Circuits,* **24**: 905–910, 1989.
18. T. Tanzawa and T. Tanaka, A stable programming pulse generator for single power supply flash memories, *IEEE J. Solid-State Circuits,* **32**: 845–851, 1997.
19. J. G. Manneatis and M. Horowitz, Precise delay generation using coupled oscillators, *IEEE J. Solid-State Circuits,* **28**: 1273–1282, 1993.
20. T. I. Laakso et al., Splitting the unit delay, *IEEE Signal Process. Mag.,* **13**: 30–60, 1996.
21. J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective,* Upper Saddle River, NJ: Prentice-Hall, 1996.
22. C. Tomazou, J. B. Hughes, and N. C. Battersby (eds.), *Switched-Currents, An Analogue Technique for Digital Technology,* London: Peregrinus, 1993.
23. R. Unbenhauen and A. Cichocki, *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems,* Berlin: Springer-Verlag, 1989.

JOSÉ M. QUINTANA
MARÍA J. AVEDILLO
Universidad de Sevilla

DELAY LINE. See DELAY CIRCUITS; PHASE SHIFTERS.
DELAYS. See CLOCK DISTRIBUTION IN SYNCHRONOUS SYSTEMS.