design ideas

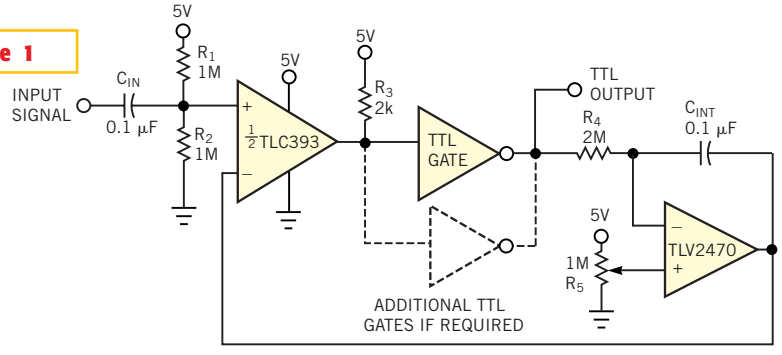
Edited by Bill Travis and Anne Watson Swager

Convert periodic waveforms to square waves

Ron Mancini, Texas Instruments, Bushnell, FL

Converting periodic waveforms to square waves is an integral part of extracting a clock signal from data, creating waveform generators, and making timing-pulse generators. Any square-wave-conversion circuit is more valuable when the square wave's duty cycle is variable and controllable. **Figure 1** shows a circuit that has these attributes and can drive several TTL-compatible loads. C_{IN} couples the input signal onto a dc level set by R_1 and R_2 (the level is $V_{CC}/2$ when $R_1=R_2$). Thus, the periodic signal at the noninverting comparator input rises above and falls below $V_{CC}/2$. The parallel value of R_1 and R_2 (R_P) and C_{IN} form a highpass filter with a −3-dB frequency of $1/(2\pi R_P C_{IN})$. Increasing R_P or C_{IN} lowers the cutoff frequency for low-frequency applications. If high-frequency noise riding on the signal causes problems, add a capacitor in parallel with R_2; this addition eliminates the high-frequency noise by creating a low-frequency filter. If the input signal is a square wave, the added capacitor integrates the square wave, thus increasing its rise and fall times. The longer rise and fall times give the circuit more control range.

Figure 1



You can obtain a square wave with 2 to 98% duty cycle with this simple circuit.

When the input signals are symmetrical, setting the dc level at $V_{CC}/2$ produces the maximum pulse-width control and duty-cycle range. Asymmetrical input signals require a different dc level, because the time durations of the positive and negative portions (with respect to $V_{CC}/2$) of the coupled signal are not equal. R_P's value must be low to prevent input-bias current from developing an appreciable offset voltage. The comparator in this design is a CMOS TLC393 version of the industry-standard LM393. The comparator weighs the dc-referenced input signal against a reference voltage from the integrator output. The comparator's output waveform is a square wave. The comparator drives a gate (or several gates if you need more output drive) through R_3. R_3 must have a low value to quickly charge the gate input during the low-to-high transition. The current the comparator can sink limits R_3's lower value.

The TLV2470 integrator integrates and inverts the output square wave and feeds it back as the reference voltage for the input comparator. If the voltage on the positive-integrator input is $V_{CC}/2$, the output square wave must be symmetrical for its average value to be $V_{CC}/2$. Adjusting R_5 to its center point yields a 50%-duty-cycle square wave. Adjusting R_5 close to ground yields a square wave that is low for most of the period, and adjusting R_5 close to V_{CC} yields a square wave that is high for most of the period. The integrator pole is at $f_P=1/(2\pi R_4 C_{INT})$. With the values shown in **Figure 1**, the 0-dB crossover frequency is 0.8 Hz. The gain of the integrator circuit is unity at 0.8 Hz, and the gain rolls off at 20 dB per decade, so the comparator's small-signal gain is not high enough to cause oscillation. The selection of the integrator pole is a trade-off between stability and control-response time. The circuit in **Figure 1** does not oscillate or multiple-switch under any conditions. It produces a square wave that's adjustable from 2 to 98% duty cycle, and it responds to 20-mV input signals.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Circuit improves on first-event detection

Lawrence Arendt, Oak Bluff, MN, Canada

THE CIRCUITS in **figures 1** and **2** exhibit certain advantages over the circuit shown in the Design Idea in *EDN*, "Circuit detects first event," May 3, 2001, pg 89. The n-player first-event-detection circuit offers several improvements:

- It has fewer passive components. It needs only n diodes instead of $(n^2+n)/2$ for three or more players. And, excluding the LEDs' current-limiting resistors, the circuit needs only n+1 resistors instead of 5n.
- The circuit uses less expensive ICs. The 74F74 or 4013 costs only 25% of the price of a LMC6762 (DigiKey catalog).
- The circuit offers inexpensive and simple scalability. You can easily add any number of additional player-event-detection channels to an event-detection configuration. All that's needed is that you connect the additional circuits to a common five-wire bus consisting of V_{CC}, ground, the Reset, the SwitchBus signal, and the CaptureInhibit signal. Thus, the wiring complexity is independent of n; in other words, it is O(1). Expanding the number of players for the original event-detection circuit requires additional diode-connected reset signals from each channel to all other channels, resulting in a wiring complexity that scales as (n^2-2n), or $O(n^2)$.
- The improved circuit uses D flip-flops having reset and clear pins: either 74F74s for regulated 5V supplies or, with minor circuit changes, 4013s for unregulated 9V-battery supplies. For a 74F74 implementation, the D input of flip-flop FF0 connects to logic 1. The Q output of this flip-flop drives the SwitchBus signal. The Q outputs of FF! through FFn have a diode-OR connection to the CaptureInhibit signal, which clocks the clock input of FF0. All the $\overline{\text{Set}}$ inputs for FF1 through FFn are connected through resistors to logic 1. Upon power-up or after you press the reset button, all the flip-flops' Q outputs are at logic 0 because of a pulse on the flip-flops' $\overline{\text{Reset}}$ inputs. The reset forces the SwitchBus signal to logic

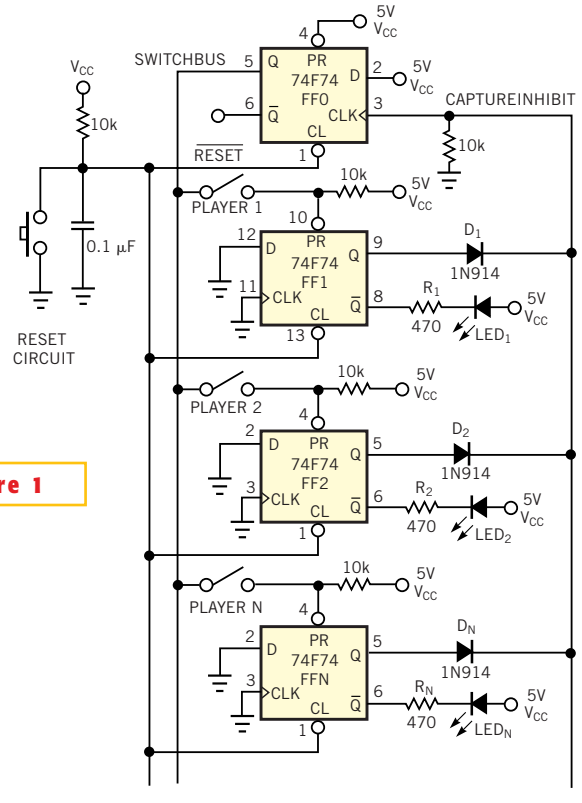


Figure 1

This first-event-detection circuit uses standard 74F74 flip-flops with a 5V supply.

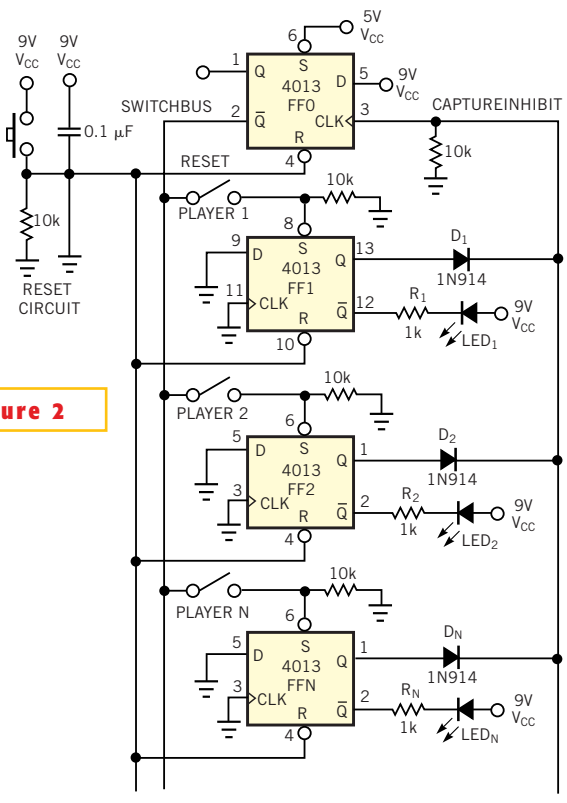


Figure 2

You can use 4013 flip-flops in a 9V-battery-powered system.

0. When you press player-event switch m, the logic-0 SwitchBus signal connects to the $\overline{\text{Set}}$ of the mth flip-flop, forcing Q_M to logic 1. Q_M now clocks FF0, forcing its Q output (SwitchBus) to logic 1. Because SwitchBus is now at logic 1, and applying logic 1 to the $\overline{\text{Set}}$ input of a 74F74 has no effect, any further switch closures by player m or any other player now have no effect.

For a 4013 implementation, the flip-flop connections are the same as for the 74F74 circuit. Upon power-up, or after you press the reset button, all the flip-flops' Q outputs are at logic 0, because of a pulse on the flip-flops' Reset inputs. The Reset signal forces the SwitchBus signal to logic 1. When you press player-event switch m, the logic-1 SwitchBus signal connects to the Set input of the mth flip-flop, forcing Q_M to logic 1. Q_M now clocks FF0, forcing its $\overline{Q}$ output (SwitchBus) to logic 0. Because SwitchBus is now at logic 0, and applying a logic 0 to the Set input of a 4013 has no effect, any further switch closures by player m or any other player now have no effect.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

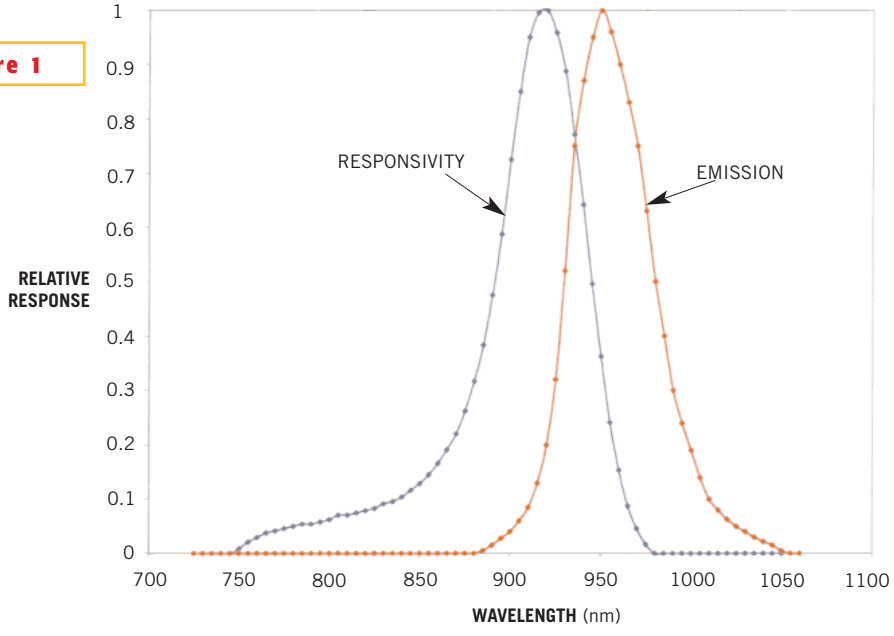
LED doubles as emitter and detector

Kyle Holland, LI-COR Inc, Lincoln, NE

Every junction diode exhibits some degree of photosensitivity when it receives light comprising an appropriate range of wavelengths. The spectral response of a junction diode depends on a variety of factors, including material chemistry, junction depth, and packaging. The packaging of most devices aims to inhibit sensitivity to radiant flux to maintain the intended function of the device. However, some devices' packaging and construction techniques allow convenient exposure to light. The most common light-sensitive devices, photodiodes and phototransistors, sense and measure light from a variety of sources. Other light-sensitive diodes, which don't usually come to mind for light-sensing applications, are LEDs. LEDs, packaged to emit radiant flux, can serve as narrowband photodetectors. The devices lend themselves to applications in which they serve as spectrally selective photodetectors or to applications in which they act as transducers. **References 1** through **4** provide further information on optoelectronic devices.

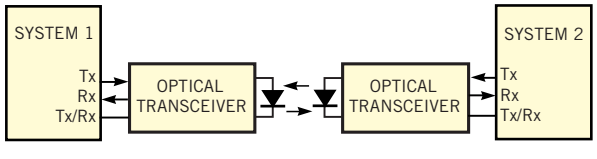
An LED's sensitivity to light and particularly to its emission wavelength depends primarily on the device's bulk material absorption and junction depth. For LEDs that have low bulk absorption,

Figure 1



Enough overlap exists between the responsivity and emission curves to make a LED useful for both transmission and reception.

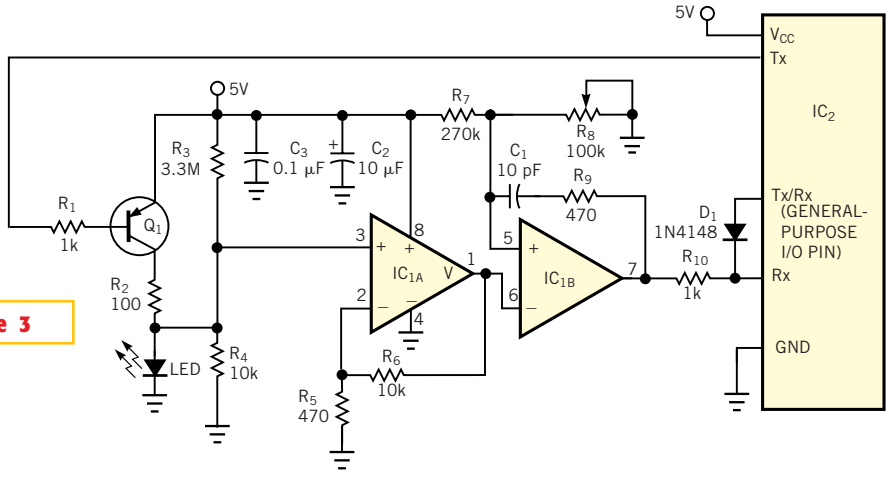
Figure 2



This half-duplex application uses a LED for both transmission and reception.

photosensitivity at or near peak wavelength is low, and, as a result, the creation of hole-electron pairs is low. GaAs-based emitters with emission wavelengths of 940 nm have relatively good sensitivity at or near their peak emission wavelength, thanks to high bulk absorption. **Figure 1** shows the relative responsivity and emission spectra for an Infineon (www.infineon.com) SFH409 LED. This infrared GaAs LED has peak emission at 940 nm with a half-peak bandwidth of 50 nm. In detection mode, it peaks at 920 nm with a half-peak bandwidth of 55 nm. As **Figure 1** shows, the wavelength of peak responsivity is shorter than the peak-emission wavelength, and a fair amount of overlap exists between the two curves. Thanks to this overlap, the LED is useful as a transducer. **Figure 2** shows a half-duplex application that exploits the LED.

Figure 3



One dual op amp and a handful of components turn a LED into a dual-purpose device.

Here, the LED links two embedded systems via a fiber-optic cable or a short-distance, line-of-sight coupling path. **Figure 3** shows the transceiver circuit used in **Figure 2**'s application. The circuit can send half-duplex data between two embedded systems at rates as high as 250 kbps. The circuit comprises the LED driver, the preamplifier, and the output comparator. The LED driver drives the LED during data transmission and unhooks the Tx pin from the LED during data reception. The Tx pin connects to transistor Q_1 via base resistor R_1. When the Tx pin is in the idle state (logic 1), the quiescent current of the LED driver is zero, because Q_1 is off. Activating the Tx pin (logic 0) causes Q_1 to turn on. Resistor R_2 sets the LED's output-power level. You should set the power level to compensate for transmission losses through the communication medium and to minimize pulse-distortion phenomena in the communication link. R_2 should be 50 to 220Ω when the circuit operates from a 5V supply.

The preamplifier consists of resistor R_4 acting as a shunt current-to-voltage converter and a high-speed, noninverting voltage amplifier (IC_{1A}). Resistor R_3 provides a slight bias of a few millivolts to R_4 to keep IC_{1A} in its linear region. The op amp in this design is the high-speed dual OPA2350 from Texas Instruments' Burr-Brown division (www.ti.com). The device has rail-to-rail inputs and outputs and a gain-bandwidth product of 38 MHz, and it can operate from a single 2.7 to 5V supply. The transconductance gain of the preamplifier is a function of the values of R_4, R_5, and R_6. In the circuit of **Figure 3**, the resistor values produce a transconductance gain of approximately 220,000. The output comparator, IC_{1B}, converts the preamplifier's output signals to logic-level voltages. You set the input threshold of the comparator by adjusting resistor R_8. By properly adjusting the threshold, you can obtain good pulse symmetry for a variety of input-power conditions. The combination of R_9 and C_1 provides ac hysteresis and additional overdrive for improved comparator switching. Moreover, R_9 limits the switching current on input Pin 5 of IC_{1B} for logic one-to-zero transitions.

During data transmission, the transmitter circuitry drives the preamplifier and comparator. To prevent locally transmitted data from causing UART overrun errors, the Tx/Rx line should be at a logic-1 level. The combination of R_{10} and D_1 blocks any transition occurring on the Rx line from activating the UART's receiver. When switching between transmitting and receiving modes, the software should include a time delay to allow for preamplifier recovery. The preamplifier-recovery delay is typically less than 10 to 20 µsec. Note that for 8051-class microcontrollers with depletion-mode pullup resistors on their I/O pins, you should replace D_1 with a pnp transistor and an associated base resistor.

References

1. "The Radiometry of Light Emitting Diodes," Technical Guide, Labsphere Inc, North Sutton, NH.
2. "GE/RCA Optoelectronic Devices," GE/RCA Corp, 1987.
3. Gage, S, D Evans, M Hodapp, and H Sorensen, *Optoelectronics Applications Manual*, McGraw-Hill, 1977.
4. Sze, S, *Semiconductor Devices—Physics and Technology*, Wiley, 1985.

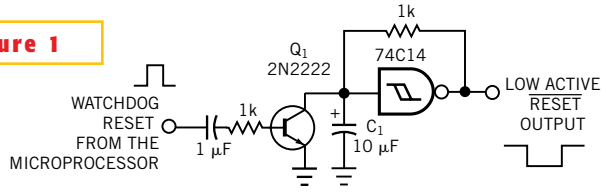
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Watchdog circuit uses ac triggering

Shyam Tiwari, Sensors Technology Private Ltd, Gwalior, India

A DC-TRIGGERED RESET of a watchdog circuit is prone to failure. If the watchdog program hangs up, then the reset signal becomes activated continuously, and the microprocessor has no way to escape the situation. We found that a simple solution uses an ac trigger to reset the watchdog circuit (**Figure 1**). We used an RC oscillator consisting of a 74C14 gate to generate active-low reset signals to the microprocessor at approximately 10-msec intervals. High-level pulses at the base of the transistor switch Q_1 reset the charging capacitor C_1 to a low level. If the watchdog trigger remains in a high state for a longer period than you want, the oscillator generates an active-low reset pulse. You may believe that a reset signal from a watchdog circuit to a microprocessor is equivalent to a power-on reset, but it is not. The warm-boot and cold-boot programs in embedded microprocessors significantly differ. Warm-boot watchdog signals are prone to hang-up. The circuit in **Figure 1** can activate the microprocessor even if the watchdog signal hangs up.

Figure 1



This watchdog circuit uses ac triggering to avoid watchdog-signal hang-up problems.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Use power line for baud-rate generation

Joseph Julicher, Microchip Technology Inc, Chandler, AZ

ONE COST-SAVING measure associated with 8-bit embedded microcontrollers is to use a resistor-capacitor oscillator. These RC oscillators are inexpensive, but the trade-off is low stability with temperature and voltage. In many applications, the low cost of an RC oscillator is alluring, but the application requires a stable clock source for baud-rate generation or event timing. In these cases, you can find a low-frequency, stable clock source and use it to calibrate a baud-rate generator or event timer. One source of a low-frequency, stable clock is the line voltage. This voltage is a good source 50- or 60-Hz frequency that you can easily interface to the microcontroller's 16-bit timer. By counting CPU cycles for a half cycle of the external clock, you can determine the frequency of the microcontroller's internal RC oscillator and calibrate the baud rate.

TABLE 1—SAMPLE NUMBERS FOR BAUD-RATE GENERATION

F_{osc} (MHz)	Reference frequency	Timer 1 prescale	Timer 1 counts	Calculated frequency	Desired baud rate	BRGH	SPBRG	Actual baud rate	Percent of error
4	60	16	4166.667	4,000,000	9600	1	25	9615.38	0.16
4.4	60	16	4583.333	4,400,000	9600	1	28	9482.75	−1.22
3.6	60	16	3750	3,600,000	9600	1	22	9782.6	1.90

The PIC16F627 flash microcontroller can benefit from this technique. This device has an internal 4-MHz RC oscillator. You can create a simple capacitor-coupled circuit to allow the microcontroller to see the pulses from the power line. You can power this PIC microcontroller from a separate circuit that includes voltage regulation (see Tech Brief 008 at www.microchip.com concerning a transformerless power supply). The power line supplies a solid 50- or 60-Hz reference frequency. You can use the 16-bit Timer 1 to time the internal oscillator. As the internal RC time constant drifts, the timer count changes, and you can use the value to determine new values for the baud-rate generator. If you adjust the baud-rate generator appropriately, you can maintain the baud rate to ±2% of the desired value. The incremental cost may be less than the cost of a crystal or ceramic resonator. You can also use this technique to periodically learn the value of the internal RC oscillator to calibrate time captures of external events. The following is an example of the technique: The base INTRC value is 4 MHz, and the power-line frequency is 60 Hz. Set BRGH for a baud clock 16 times the baud rate, and set Timer 1 prescale to 16. The desired baud rate is 9600.

The formula we use is ((Timr1 value×prescaler×reference frequency×baud multiplier))−1=SPBRG value. We used a spreadsheet to run some sample numbers, and the results are in **Table 1**. To use the technique with a PIC microcontroller, simplify the math to (Timr1/160)−1=SPBRG. You will find that 9600/(16×60)=160. This simplification causes a slight error; the rounded off value of SPBRG becomes 27 instead of 28. The error at 4.4 MHz becomes 2.3% instead of −1.22%.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

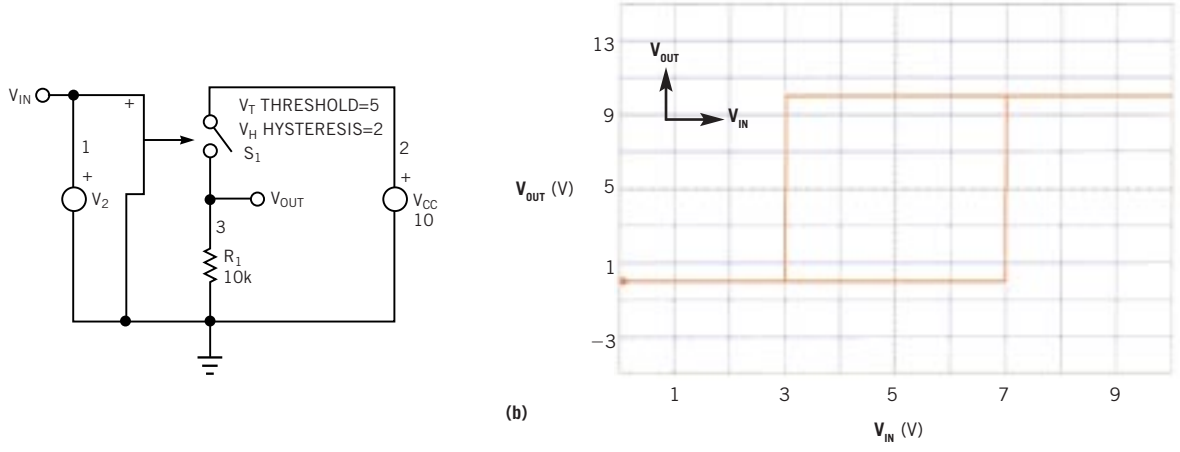
Inline equations offer hysteresis switch in PSpice

Christophe Basso, On Semiconductor, Toulouse, France

Smooth-transition switches are convenient devices in many Spice-based simulators. Their action can greatly ease the convergence process. Unfortunately, these devices lack inherent hysteresis, a helpful feature used to build UVLO (undervoltage-lockout) systems, oscillators, and other systems. Intusoft's (www.intusoft.com) IsSpice4 not only provides users with smooth-transition devices, but also adds the Berkeley Spice primitive switch featuring adjustable hysteresis. **Figure 1a** shows how to wire this component in a simple comparator architecture. **Figure 1b** shows the resulting V_{OUT} versus V_{IN} curve. In Cadence's (www.cadence.com) PSpice, the S_1 primitive switch implements a soft transition between R_{ON} and R_{OFF}, but V_{ON} and V_{OFF} are simply the final levels that reflect the specifications for R_{ON} and R_{OFF}. To incorporate some hysteresis, you just need to add some analog-behavioral-model sources to help tailor our switching events. **Figure 2** shows how you can derive the new device with adjustable hysteresis.

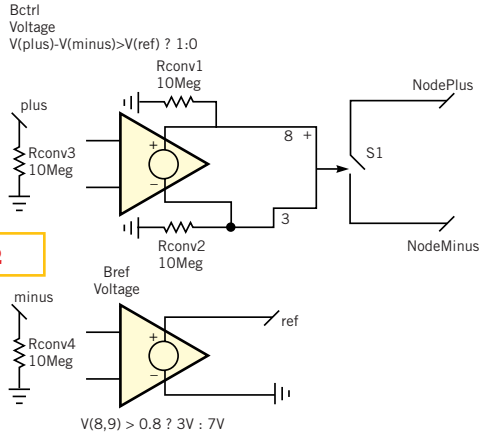
Listing 1 is the complete PSpice netlist for the switch. The inputs V(plus) and V(minus) route the switch-control signal to the Bctrl behavioral element. (A b element in IsSpice4 becomes a "value" E source in PSpice.) When the switch is open (Bctrl delivers logic 0), the reference node (node *ref*) becomes armed at the highest toggling point (7V in **Figure 2**'s

Figure 1



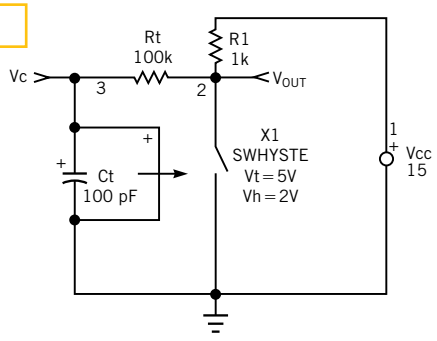
This switch circuit (a) toggles at 7V and resets at 3V, yielding 4V hysteresis (b).

Figure 2



Some analog-behavioral-model sources and a primitive S switch yield a switch with adjustable hysteresis.

Figure 3



These results clearly show the 4V hysteresis in Figure 2's subcircuit.

example). When the input voltage increases, it crosses the *ref* level, and Bctrl switches high. Switch S_1 closes and authorizes the current to flow in its terminals through R_{ON}. At this time, the Bref source has detected that S_1 is closed and now modifies its reference node to the second level (3V in the example). When the input voltage drops, it crosses the 7V level, but no action takes place, because *ref* has changed to 3V. When the input voltage finally crosses 3V, S_1 opens and applies R_{OFF} between its terminals. **Figure 3** shows the PSpice simulation results, confirming the 4V hysteresis. **Figure 4** shows how to build a simple RC test oscillator, using the PSpice-derived hysteresis. You can download the PSpice netlist from the Web version of this article at www.ednmag.com.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

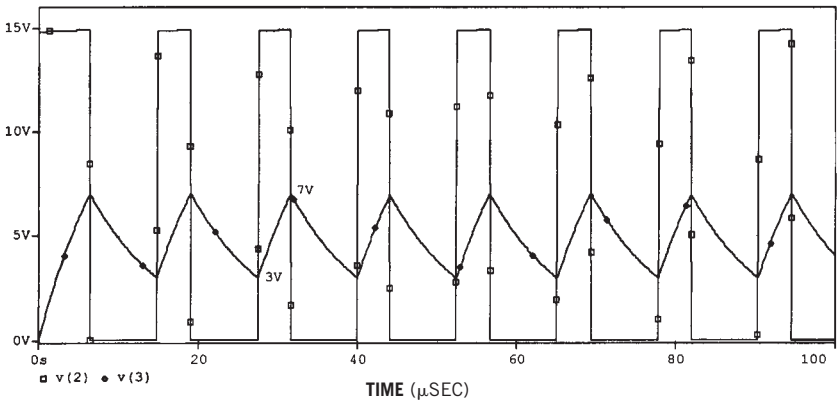


Figure 4 A switch with hysteresis and a few passive elements produce an RC oscillator.

LISTING 1—PSPICE NETLIST FOR SWITCH WITH HYSTERESIS

```
.subckt SWhyste NodeMinus NodePlus Plus Minus PARAMS: RON=1 ROFF=1MEG VT=5 VH=2
S5 NodePlus NodeMinus 8 0 smoothSW
EBcrtl 8 0 Value = { IF ( V(plus)-V(minus) > V(ref), 1, 0 ) }
EBref ref1 0 Value = { IF ( V(8) > 0.5, {VT-VH}, {VT+VH} ) }
Rdel ref1 ref 100
Cdel ref 0 100p  IC={VT+VH}
Rconv1 8 0 10Meg
Rconv2 plus 0 10Meg
Rconv3 minus 0 10Meg
.model smoothSW VSWITCH (RON={RON} ROFF={ROFF} VON=1 VOFF=0)
.ends SWhyste
```