EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

Spice simulators provide behavioral sources to mere mortals

William H Morong, Paoli, PA

Spice simulators include three types of tools for voltage and current sources: independent, dependent, and behavioral. Independent sources have two terminals and provide a specified number of volts or amperes you enter as a constant, just as a bench supply generates a set voltage or current. These simulators, like signal generators, also provide waveforms.

Figure 1a shows two independent sources. Voltage source V_1 provides 5V dc, and V_2 provides a 1V-peak sine wave. Dependent sources have at least four terminals. A first terminal pair is a voltage or current output. Another terminal pair is an input that attaches in shunt or in series with two circuit nodes. The output responds to the voltage across or current through the input nodes. Some dependent sources have multiple input-terminal pairs. The output responds to the inputs according to some linear mathematical rule. For example, a voltage-dependent voltage source set to the constant 100, such as E_1 in **Figure 1a**, is an ideal voltage amplifier with a gain of 100.

Behavioral, or arbitrary, sources are the least-used but most powerful of these sources. They have only an output-terminal pair, but they are more powerful than their simpler counterparts. They can implement a set of mathematical functions that roughly correspond to those available on scientific calculators. Their outputs can be independent or dependent. In the simplest case, a behavioral voltage source set to V=5 produces 5V, just as a power supply does.

A behavioral source can also respond to a designated voltage or current somewhere in the simulated circuit. In **Figure 1b**, B_1 responds to the voltage at the input node. Applying

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

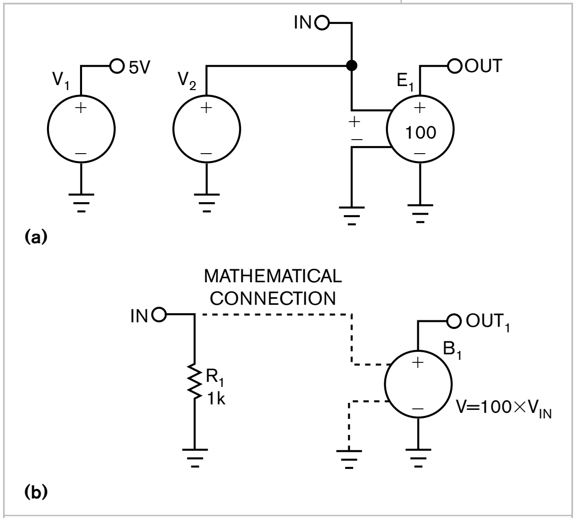


Figure 1 Independent voltage source V_1 provides 5V dc, and V_2 provides a 1V-peak sine wave (a). Behavioral source B_1 responds to the voltage at the input node (b).

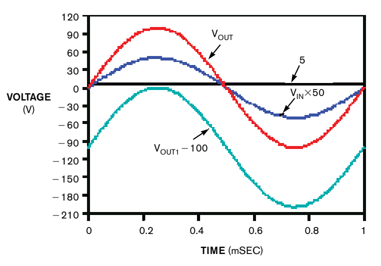


Figure 2 You can set behavioral sources to become simulated electronic components that embody mathematical expressions.

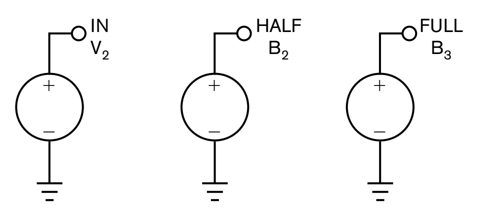


Figure 3 Behavioral source B_2 amplifies and half-wave-rectifies the voltage at the input node. Behavioral source B_3 also amplifies and full-wave-rectifies the same voltage.

the **equation** beside B_1 sets the behavioral source to $V=100\times V_{IN}$ and amplifies the voltage across R_1 by 100V.

In this case, the behavioral source behaves as the dependent source when the input node and ground drive it, but the behavioral source's input connection is purely mathematical. **Figure 2** shows several possible plots for output signals. You can set behavioral sources to become simulated electronic components that embody mathematical expressions. In the preceding example, if you set the behavioral source to $V=\text{uramp}(100\times V_{IN})$, the amplified output is ideally half-wave-rectified. If you substitute the abs function for uramp, you get full-wave rectification.

In **Figure 3**, behavioral source B_2 amplifies and half-wave-rectifies the voltage at the input node. Behavioral source B_3 also amplifies and full-wave-rectifies the same voltage. **Figure 4** shows the results. Engineers commonly generate ABMs (analog behavioral models) for Spice simulations to represent entire blocks of analog functions. Sometimes you need simple models when none are available, but you can use behavioral models to solve those problems. For example, you can set $I=\text{uramp}(V_{FB}-1.25)$ as a behavioral current source to draw 1 mA/mV at the FB (feedback) node that exceeds 1.25V (**Figure 5**). The behavioral source looks like an idealized three-terminal shunt regulator (**Figure 6**). If you need more refinement, you can set the feedback node to act as an RC filter or another function.

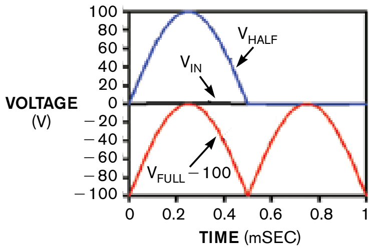


Figure 4 Behavioral source B_2's amplification and half-wave-rectification of the voltage at the input node and B_3's amplification and full-wave rectification of the same voltage yield these results.

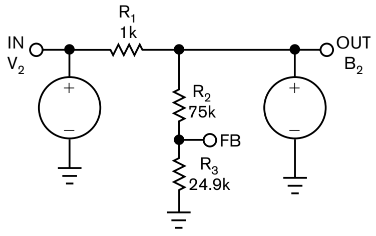


Figure 6 The behavioral source looks like an idealized three-terminal shunt regulator.

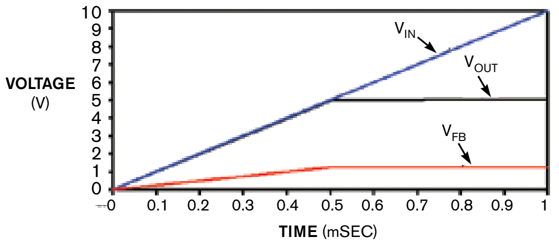


Figure 5 You can set $I=\text{uramp}(V_{FB}-1.25)$ as a behavioral current source to draw 1 mA/mV at the FB (feedback) node that exceeds 1.25V.

Because behavioral sources can embody mathematical functions, you can use them to test the mathematical viability of a design before simulating the circuit. Long mathematical expressions can be difficult to read, so you should break the idea into small blocks and set a behavioral source to implement each block. Just because your idea works in math, though, doesn't mean that its behavioral model is a practical circuit. In Spice, it is easy to make a model of a current source with 1 MV of near-perfect compliance, sometimes without intending to do so, but most practical engineers would rather defer the realization of such a circuit.

The caution to be realistic applies to Spice in general and to behavioral sources in particular. When you draw two equal resistors in your Spice schematic, they will be perfectly matched in simulation. The resistors in your stockroom are not that good. **EDN**

Multidecade BCD DAC uses resistors of only six values

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

A previous Design Idea uses a three-decade BCD (binary-coded-decimal) DAC to precisely set the output current of a current source (**Reference 1**). The circuit acts as a code-to-conductivity converter. The values of resistors of this DAC are staggered by powers of two within any of the decades, and the values of resistors at corresponding bits of the decades are staggered by powers of 10. Thus, the circuit needs 12 values of resistors, ranging from 125Ω to 100 kΩ.

In comparison, the circuit in this Design Idea enables you to construct a BCD DAC using only six resistor values, regardless of the number of decades. Moreover, these six resistor values vary within a relatively narrow 1:8 range. The voltage-output DAC operates ratiometrically. That is, if the temperature coefficients of the resistors are approximately the same—and you can assume they will be within this narrow range of values—then the variation of resistance with temperature has almost no detrimental effect on accuracy. This situation is not true, however, for code-to-conductivity DACs, in which the temperature coefficient of the resistors directly influences the temperature coefficient of the DAC.

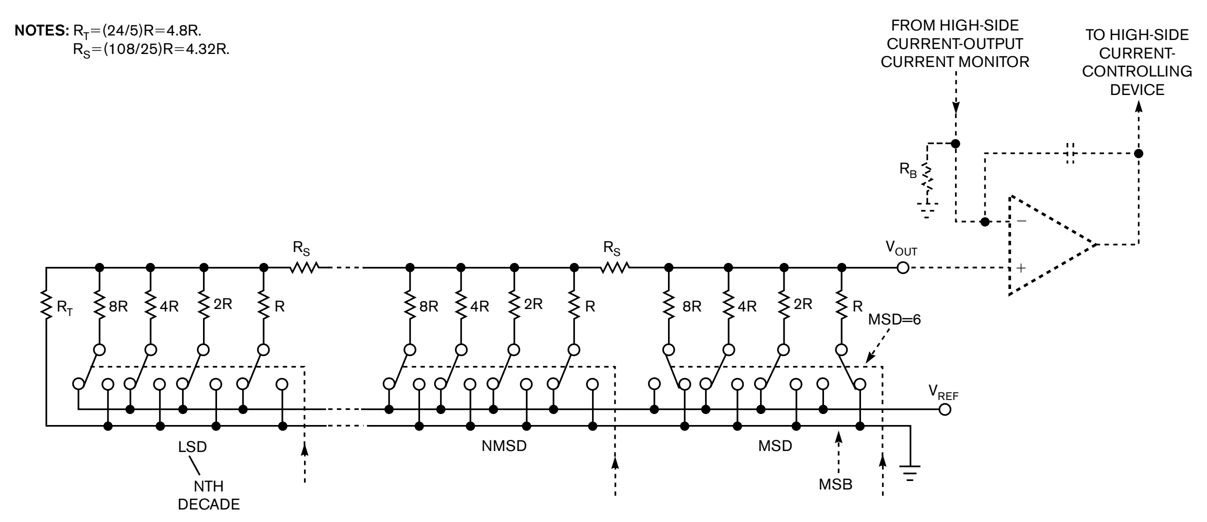


Figure 1 A voltage-output BCD DAC needs just six precision-resistor values. Resistors connect either to ground or to the reference voltage.

Figure 1 shows the voltage-output BCD DAC. The values of resistors are staggered by powers of two within each decade. The values of resistors at corresponding bits of the decades are of equal values. The switched ends of the resistors connect to either ground at logic zero or to the reference voltage at logic one. The voltage-output DAC thus has an advantage in that all the resistors' ends are at a defined potential. In a code-to-conductivity converter, on the other hand, one end of the resistor remains open at logic zero, and these open ends might act as capacitive sensors or even antennas, which could introduce additional errors. The common ends of four resistors in four bits of the MSD (most-significant decade) form the output. The common outputs of resistor quads in the less-significant decades successively connect to the main output through the series resistors, R_S, which all have the same value. Thus, R_S=(108/25)R=4.32R, where R is the value of the resistor at the MSB (most-significant bit) of any of the decades.

The common ends of bit resistors in the LSD (least-significant decade) connect to ground through terminating resistor R_T. This resistor represents the theoretically infinite number of decades having weight lower than the actual LSD, whereas these hypothetical decades are all set to zero. Thus, they contribute no voltage at the output. They do, however, influence the properties of the resistive network. R_T sets this influence and is equal to (24/5)R, or 4.8R. The full-scale output of the voltage-output BCD DAC is $3/5\times(1-10^{-N})V_{REF}$, where N is the number of decades—in this case, three.

To exploit a voltage-output BCD DAC in a single-supply programmable-current source, connect the output of the voltage-output BCD DAC to the noninverting input of an op amp, which accepts input voltages as low as 0V. The inverting input of the op amp connects to ground through resistor R_B, which has the value of $(3/5)\times(V_{REF}/10^{-2}\text{A})$. **EDN**

REFERENCE

[1] Diószegi, Gyula, and János Nagy, "Current-sense monitor and MOSFET boost output current," *EDN*, May 28, 2009, pg 44, www.edn.com/article/CA6659409.

Converter translates Bayer raw data to RGB format

Yu-Chieh Chen and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

CMOS image sensors include the color filters of an RGB (red/green/blue) Bayer array, which lets the sensor detect colors. The image data, the output from the image sensor, is Bayer raw data (**Figure 1**). Unfortunately, most consumer-grade image-displaying devices require an RGB-image-data format with red, green, and blue in each pixel's data. Therefore, you often need a Bayer-raw-data-to-RGB converter between an image sensor and a displaying device. This Design Idea describes such a converter in Verilog HDL (hardware-description language). You can implement the code, available from the online version of this Design Idea at www.edn.com/100204dia, into a CPLD or an FPGA.

To make the design easy to understand, the RGB data is only 24 bits deep. A 320×240-pixel test-bench pattern verifies the design (**Figure 2**). The image data for red, green, and blue are 88h, 66h, and 22h, respectively. **Figure 3** shows the timing of the converter, and **Figure 4** shows the flow

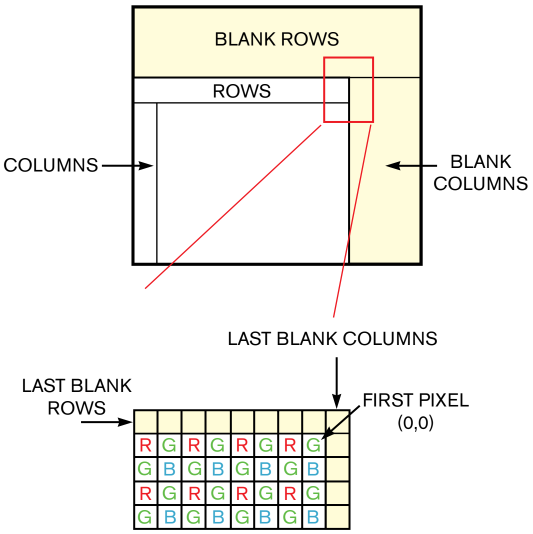


Figure 1 The Bayer raw data from an image sensor represents a 320×240-pixel array.

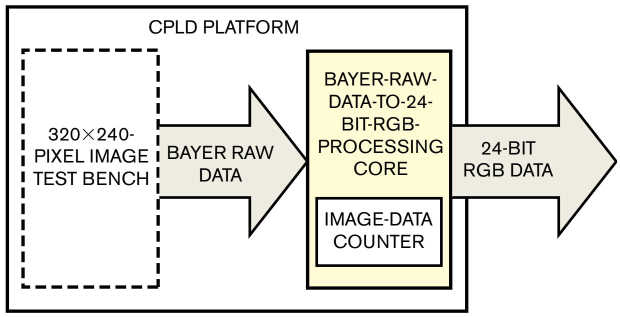


Figure 2 A CPLD processes Bayer raw data into 24-bit RGB data.

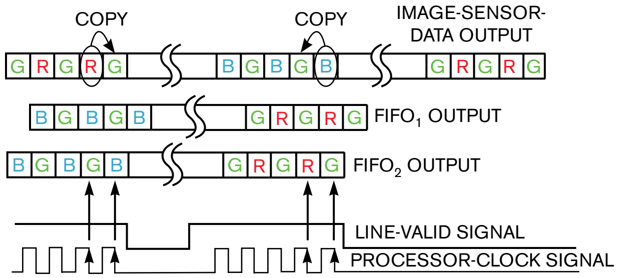


Figure 3 Rising edges of a clock latch Bayer raw data from FIFOs into RGB data.

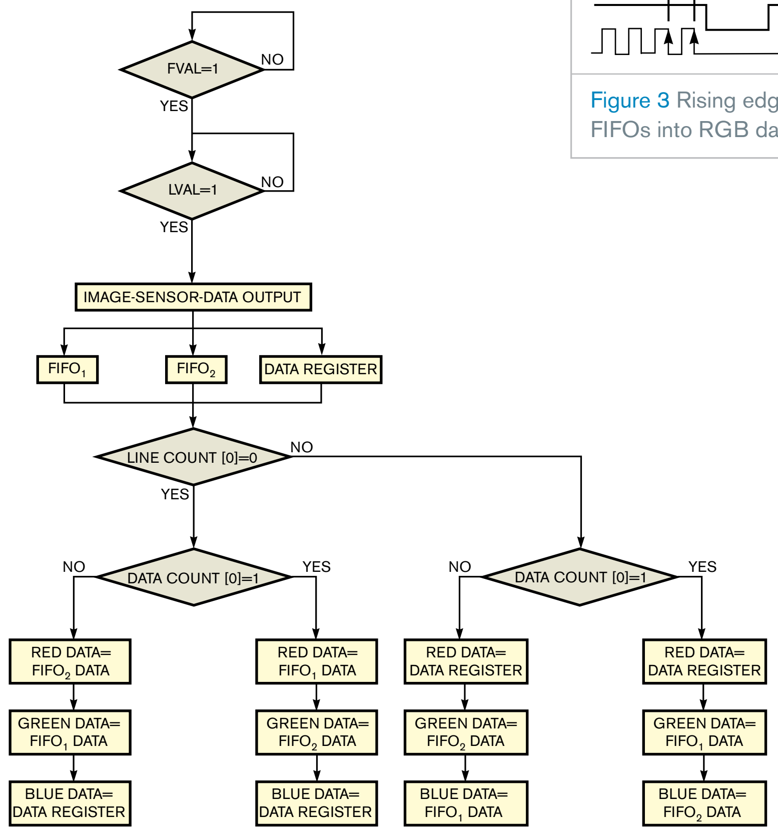


Figure 4 RGB data represents red, green, and blue for display.

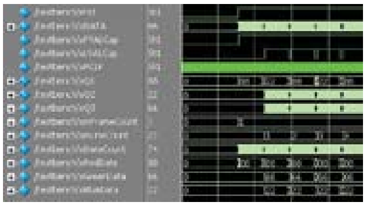


Figure 5 The values 88h, 66h, and 22h, respectively, represent red, green, and blue data.

chart. In **Figure 3**, the PCLK (processor-clock) signal's rising edge latches the 24-bit RGB data. The LVAL (line-valid) cap signal is the synchronized line-valid signal for reading the data. The process begins when the FVAL (frame-valid)-data signal goes high. When that action occurs, sensor data begins writing to FIFO$_1$ (first-in/first-out), FIFO$_2$, and the data register. After saving the data, FIFO$_1$ reads out the data. FIFO$_2$ and the data register read their data at the same time. Those readouts occur one clock cycle after FIFO$_1$ starts reading. The next 24-bit RGB data remains the same in the data register, and it combines the data from FIFO$_1$ and FIFO$_2$, which read out at the same rising edge of the clock. The line-count signal shows whether the data is even or odd, which influences the combinational sequences of the data that reads out from FIFO$_1$, FIFO$_2$, and the data register.

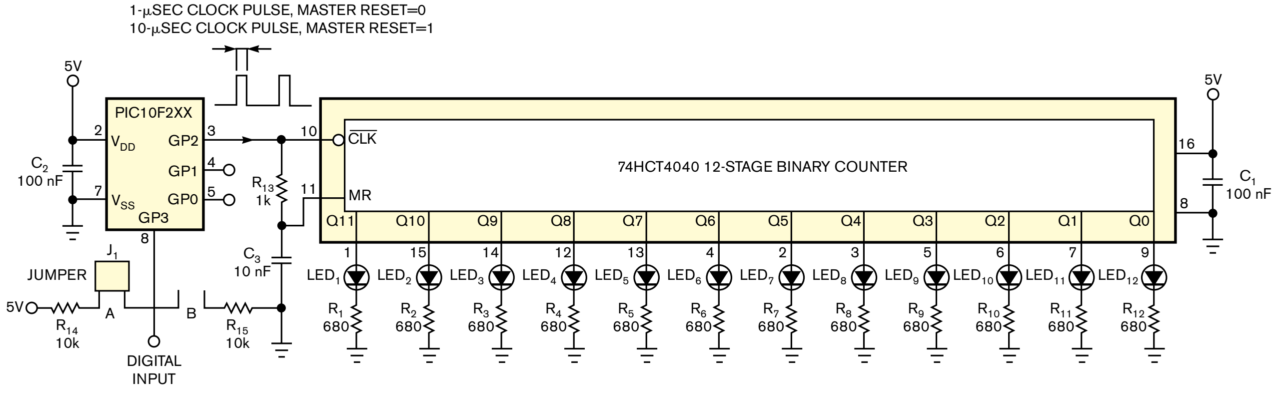
Figure 5 is the ModelSim simulation waveform of the converter. The 24-bit RGB output-data values 88h, 66h, and 22h are red, green, and blue data, respectively. The **figure** shows the 24-bit RGB data as having red, green, and blue values of 88h, 66h, and 22h, respectively, during every line-data period. The line-data period matches the default pixel value in the image data's test-bench pattern.EDN

Drive 12 LEDs with one I/O line

Charaf Laissoub, Maisons Alfort, France

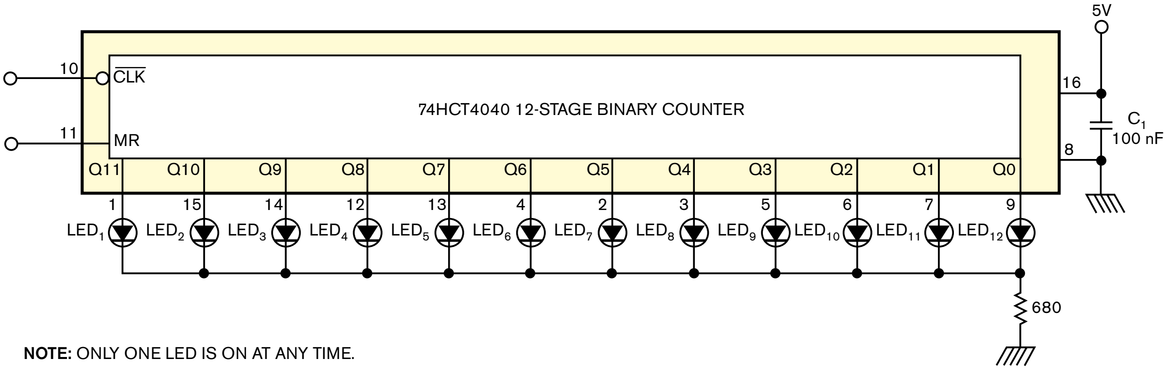
Several Design Ideas expand the I/O of a pin-limited microcontroller (**references 1** through **4**). The circuit in this Design Idea uses an alternative method (**Figure 1**). It limits external additional parts to one IC, and it drives as many as 12 LEDs in dot-bar or bar-graph mode. You can use the same technique in a dot-bar design (**Figure 2**). If you need a seven-segment LED display, you can use the circuit in **Figure 3**, which shows how to rearrange the circuit according to a classic multiplexed, four-digit common-cathode display. The prototype display uses Kingbright's (www.kingbright-led.com) SC52-11EWA high-efficiency LEDs, which emit 2000 to 5600 µcd at a forward current of 10 mA. The driver is a 12-stage NXP (www.nxp.com) 74HCT4040 binary counter or a 74HC4040 version for a lower power supply.

Listing 1, which you can download at the online version of this Design Idea at www.edn.com/100204dib, contains an assembly-language routine. It generates a precise quantity, Q, of high-frequency pulses, which deliver the number, N, that the outputs of the 74HCT4040 require. The relations are $Q=2^{N-1}$ in dot-bar mode and $Q=2^N-1$ in bar-graph mode. **List-**



NOTE: BEFORE POWER-ON, PUT JUMPER INTO A FOR BAR-GRAPH MODE OR PUT JUMPER INTO B FOR DOT-BAR MODE.

Figure 1 A 12-LED dot-bar/bar-graph display uses a 12-stage binary counter.



NOTE: ONLY ONE LED IS ON AT ANY TIME.

Figure 2 Using one current-limiting resistor can reduce parts count.

ing 2, which is also available at www.edn.com/100204dib, is a full example of using this routine with Microchip's (www.microchip.com) PIC10F202, a member of the PIC10F series, which is the company's most pin-limited microcontroller family.

Although the PIC's internal unique clock frequency is 4 MHz, you'll notice little flicker effect. You can reduce the flicker by using a midrange pin-limited PIC microcontroller, such as the PIC12F629, which has an internal clock frequency of 20 MHz. **Listing 3**, also available at www.edn.com/100204dib, uses a look-up table to convert the desired number into seven-segment code to replace the 12 LEDs with a four-digit display.**EDN**

REFERENCES

[1] Anonymous, "Microcontroller provides low-cost analog-to-digital conversion, drives seven-segment displays," *EDN*, May 10, 2007, pg 80, www.edn.com/article/CA6437954.

[2] Raynus, Abel, "Squeeze extra outputs from a pin-limited microcontroller," *EDN*, Aug 4, 2005, pg 96, www.edn.com/article/CA629311.

[3] Jayapal, R, PhD, "Microcontroller's single I/O-port line drives a bar-graph display," *EDN*, July 6, 2006, p 90, www.edn.com/article/CA6347254.

[4] Lekic, Nedjeljko, and Zoran Mijanovic, "Three microcontroller ports drive 12 LEDs," *EDN*, Dec 15, 2006, pg 67, www.edn.com/article/CA6399101.

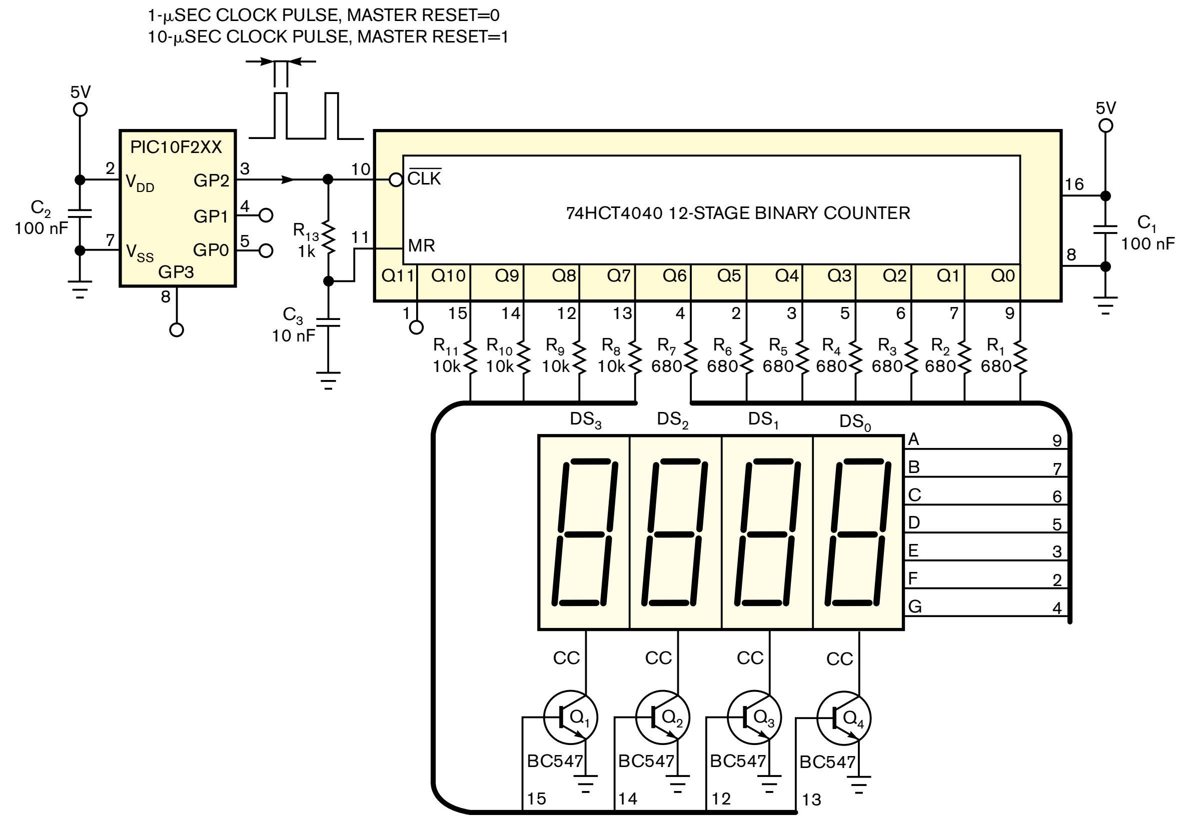


Figure 3 The same technique works with seven-segment displays.

Circuit precludes common-mode conduction

Ken Herrick, Oakland, CA

When driving an H-bridge or a similar circuit, you usually must ensure that two or more transistors are not on at the same time. Eliminating multiple transistors from turning on reduces power consumption and lowers EMI (electromagnetic interference). Crossover-delay circuits solve that problem. **Figure 1** shows a simple, two-phase design that lets you adjust the crossover delays equally by changing the value of one component with minimal phase delay.

Each Schmitt trigger inverter is driven during one half-cycle through

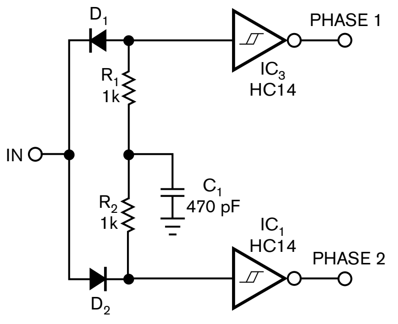


Figure 1 Each Schmitt trigger inverter is driven during one half-cycle through a diode. The RC delay occurs during the alternate half-cycle. Equal-value resistors R_1 and R_2 serve alternatively as delay elements and gate-coupling resistors.

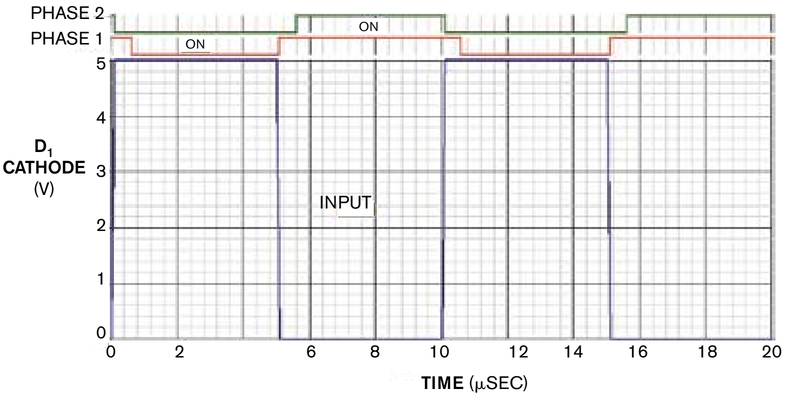


Figure 2 For the two out-of-phase half-cycles, leading edges are delayed equally with respect to the input transition, and trailing edges are coincident with the transition within about one gate delay.

a diode. The RC delay occurs during the alternate half-cycle. Equal-value resistors R_1 and R_2 serve alternatively as delay elements and gate-coupling resistors. The waveform in **Figure 2** shows the result. For the two out-of-phase half-cycles, leading edges are delayed equally with respect to the input transition, and trailing edges are coincident with the transition within about one gate delay. If you need equal polarity “on” half-cycles, insert an inverter in one of the two phase outputs. Alternatively, if biphase drivers, such as those for driving coupling transformers, will follow this circuit, merely interchange the outputs of one of those drivers to effect the inversion.**EDN**