Design Ideas from EDN

January 2000 pag. 6
Low-power PWM circuit is simple, inexpensive
Manchester co-decoder fits into 32-macrocell PLD
Input-protection scheme tops other approaches
Level-shifting nixes need for dual power supply
Synchronize asynchronous reset
Circuit resolves 0.1-fF change from 100 pF
Dual-voltage supply powers SIM card
Design formulas simplify classic V/I converter
Rail-to-rail op amp provides biasing in RF amp
Circuit multiplexes automotive sensors
Analog switch acts as dc/dc converter
Circuit provides message on disabled phone line
Optocoupler isolates shift registers
Tack a log taper onto a digital potentiometer

February 2000 pag. 21
Low-cost isolation amplifier suits industrial applications
Unused µC ADC pins find second life
AC-couple analog signals without a dc bias source
Simple active-matrix-LCD bias supply operates from battery input
µC generates a frequency burst
Keyboard data-acquisition system is cheap and simple
MOSFET switch provides efficient ac/dc conversion
Passive circuit monitors AES data
Microcontroller multiplexes DIP switches to I/O port
Switched-capacitor IC controls feedback loop
Simple circuit disconnects load
Follow the debouncing flip-flops
Inductorless converter provides high efficiency

March 2000 pag. 36
Simple BER meter is easy to build
Software avoids interrupt overhead
Delay line eases Spice dead-time generation
Comparison macro for PIC processors
Bridging enhances filter close-in selectivity
Notch filter is insensitive to component tolerances
Instrumentation amp works from one supply
Amplifier requires no dc bias

Halogen light dimmer provides infinite control
Cascoded stack yields multiple voltages
BIOS interrupt performs A/D conversion
Generate frequencies with arbitrary relationships
Push-pull driver provides isolated 5V at 1A
Circuit programs Atmel AVR microcontrollers
Circuit adds latch-off current limit to regulator
Dual power supply delivers 8A with no heat sinks
Power switch provides soft start
Circuit eliminates PC echoes
Clamping circuit dissipates minimal power
Piezo crystal monitors liquid level
Switch intelligently controls current

April 2000 pag. 58
FPGA implements X.50 Division 3 recommendation
Concurrent processing produces fast priority selector
Proportional thermoregulator synchronizes to line
Spread-spectrum method identifies audio path
5-to-1.8V converter works without magnetics
Band-reject filter includes compensation
Low-cost active load draws constant battery power
Microcontroller visualizes hex code
Simple circuit times bathroom fan
Termination supply tracks one-half core voltage
Switch debouncer isolates input and output
Circuit detects reset source
Interface LCD with ease

May 2000 pag. 73
Potentiometer tames current-feedback op amp
Cable finder seeks out wall conduits
Electronic transformer dims halogen lamp
Low-cost, programmable oscillator operates at 2 GHz
Simple circuit produces a less-than-25-nsec pulse
Microcontroller multiplexes six digital potentiometers
Filter design uses image parameters
Circuit efficiently drives inductive loads
Use a PC to record four-channel waveforms
Pulse generator has low top-side aberrations
Circuit provides ADSL frequency reference
ActiveX control brings bit manipulation to Windows
Circuit breaker handles voltages to 32V

Data-acquisition circuit measures almost everything
Full-wave rectifier has programmable gain
Method synchronizes slaves in power-line communications
Audio amp makes efficient fan controller
Inverted bipolar transistor doubles as a signal clamp
Amplitude-stable oscillator has low distortion, low cost
Transistor latch improves on/off circuitry
High-side driver feeds IGBTs and MOSFETs
Economical circuit drives white LEDs
Real-time-clock chip makes low-power oscillator
Everything you wanted to know about wideband, low-frequency transformers
Microcontroller makes effective frequency counter
Software provides interrupt system for 8051
Look-up table facilitates bit flipping

December 2000 pag. 194
Free-line indicator stops interruptions
Generate CID/CIDCW analog signals
Bipolars provide safe latch-off against opto failures
Simple logic analyzer pushes microcontroller to its limit
PIC debugging routine reads out binary numbers
Switching regulator charges NiMH batteries
Circuit provides reverse-battery protection
One-shot circuit is programmable
BIOS interrupt does eight-channel frequency counting
RTDs provide differential temperature measurement
Transistor junctions monitor temperature zones

design ideas

Edited by Bill Travis and Anne Watson Swager

Low-power PWM circuit is simple, inexpensive

Anthony Smith, Scitech, Biddenham, England

A common technique for implementing PWM involves comparing a triangular waveform of fixed amplitude and frequency with a variable dc voltage level. Although this approach results in a PWM signal of precise frequency and with duty cycle variable from 0 to 100%, the need for a reference triangle waveform and a suitable fast comparator can be prohibitively expensive in low-cost applications. Furthermore, if an application requires a high-frequency PWM signal, the power consumption may be unacceptable in power-sensitive applications, such as high-efficiency, low-power switch-mode regulators.

The circuit in **Figure 1** is a relatively simple alternative to the triangle/comparator approach. Although the frequency of the output waveform is not stable and varies with input voltage, the circuit is inexpensive, requires only a handful of readily available parts, and exhibits a linear relationship between input voltage and output duty cycle. The circuit lends itself to applications that enclose a simple PWM section within a feedback loop. Also, the excellent dynamics—the duty cycle responds to an input step change within one cycle of the output waveform—make the circuit ideally suited to switch-mode-regulator applications.

In the circuit, the dc input voltage, V_I, varies the duty cycle of the rectangular signal at the output of Schmitt inverter, IC_{1A}. Q_1 and Q_2 function as switched-current sources. These sources charge and discharge timing capacitor C_1 at a rate that their base voltages and, hence, the voltage at the junction of R_2 and R_3 determine. When the output of IC_{1A} is high, C_1 charges through R_6 and Q_1 (Q_2 is cut off) with a charge current set by R_6 and the emitter voltage of Q_1. Similarly, when the output of IC_{1A} is low, C_1 discharges via Q_2 and R_6 (Q_1 is cut off) with a discharge current set by R_6 and the emitter potential of Q_2. Adjusting the input voltage changes the emitter potentials and thus varies the charge and discharge currents so that the duty cycle of the output waveform varies in direct linear proportion to V_I.

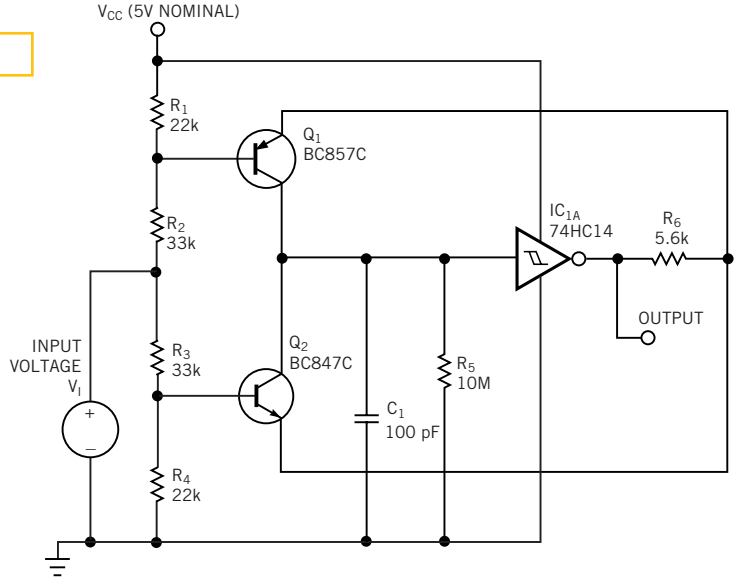
Figure 2 shows the relationship between V_C, which is the voltage on C_1, and the output waveform. V_{TU} and V_{TL} are the upper and lower thresholds of the Schmitt inverter, V_H is the Schmitt trigger's hysteresis, and V_{OH} and V_{OL} are the high and low output levels, respectively, of the inverter.

If you assume that $V_{OH}=V_{CC}$ and $V_{OL}=0V$ and taking the base-emitter voltages of Q_1 and Q_2 to be roughly equal and denoted by V_{BE}, you can derive the following first-order expressions for T_1 and T_2, where $K_1=R_2/(R_2+R_3)$, and $K_2=R_4/(R_3+R_4)$:

$$T_1=\frac{C_1 \bullet R_6 \bullet V_H}{V_{CC}(1-K_1)+V_I(K_1-1)-V_{BE}},$$

and

Figure 1



In this PWM circuit, adjusting the input voltage, V_I, changes the emitter potentials of Q_1 and Q_2 and thus varies the charge and discharge currents of C_1 so that the duty cycle of the output varies in direct linear proportion to V_I.

$$T_2 = \frac{C_1 \bullet R_6 \bullet V_H}{K_2 \bullet V_I - V_{BE}}.$$

Defining the output duty cycle as equal to $100\% \times T_1/(T_1+T_2)$, you can combine the expression for T_1 and T_2 to yield

$$\text{DUTYCYCLE} = \frac{K_2 \bullet V_I \bullet -V_{BE}}{V_{CC}(1-K_1)+V_I(K_1+K_2-1)-2V_{BE}} \times 100\%.$$

If the R_1-to-R_4 divider network is symmetrical, or $R_1=R_4$ and $R_2=R_3$, this expression simplifies to

$$\text{DUTYCYCLE} = \frac{K_2 \bullet V_I - V_{BE}}{V_{CC}(1-K_1)-2V_{BE}} \times 100\%.$$

Taking the values for R_1 to R_4 in **Figure 1**, the equation reduces to

$$\text{DUTYCYCLE} = \frac{0.4\,V_I - V_{BE}}{0.4\,V_{CC}-2V_{BE}} \times 100\%.$$

This expression shows that the duty cycle is directly proportional to the input voltage and that V_I must be greater than $V_{BE}/0.4$ for the circuit to work. If $V_{BE}=0.6$V, this equation suggests that V_I must be at least 1.5V, although, in breadboard tests, the circuit produced low duty cycles with V_I as low as 1V.

You select C_1 and R_6 according to the required operating-frequency range. **Figure 3** illustrates the results of breadboard tests with $R_6=5.6$ kΩ and $C_1=100$ pF. The circuit exhibits linear performance with V_I at approximately 1.2 to 3.6V with a corresponding duty-cycle range of approximately 2 to 95%. This **figure** also shows that the output frequency varies by as much as 15 to 1 over this range; the output frequency peaks when V_I is approximately equal to $V_{CC}/2$.

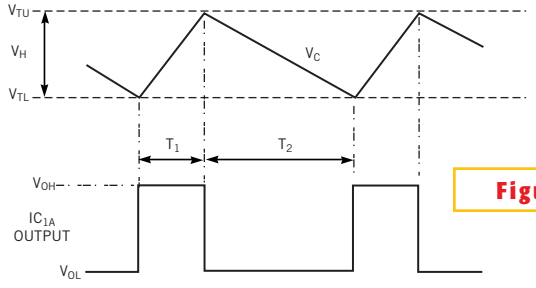


Figure 2

The changing voltage, V_C, across C_1 and the hysteresis, V_H, of IC_{1A} determine the duty cycle, $T_1/(T_1+T_2)$, of the output waveform. V_{TU} and V_{TL} are IC_{1A}'s upper and lower thresholds, respectively.

You need to observe a few caveats when selecting R_1 to R_4 and IC_{1A}. To ensure that the duty cycle is variable from near zero to near 100%, the charge and discharge currents through Q_1 and Q_2 must be able to approach zero. You can meet this requirement simply by ensuring that V_{E1}, or Q_1's emitter potential, can approach V_{CC} and that V_{E2}, or Q_2's emitter potential, can approach ground.

You can make V_{E1} approach V_{CC} when V_I is a maximum by the suitable selection of R_1 and R_2, provided that you choose R_3 and R_4 so that V_{E2} can go a few hundred millivolts below the minimum lower threshold voltage, V_{TL} (minimum), of IC_{1A} when V_I is a maximum. This feature is necessary to ensure that Q_2 does not saturate when V_C approaches V_{TL} (minimum) as C_1 discharges.

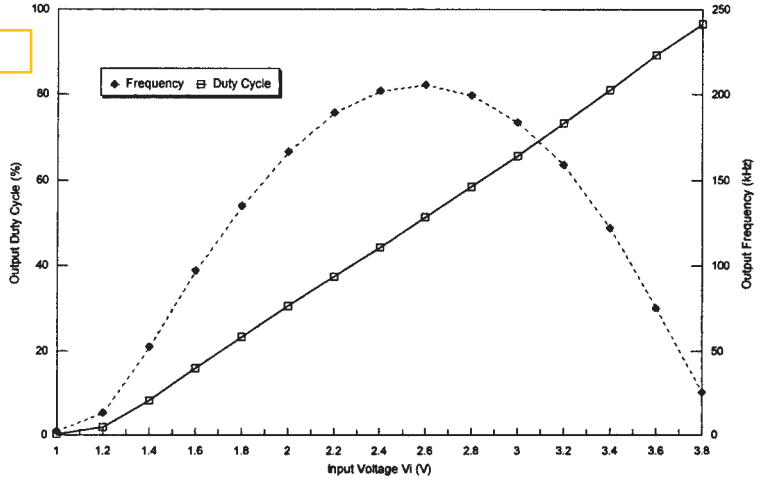
Similarly, by suitably selecting R_3 and R_4, you can make V_{E2} approach zero when V_I is a minimum, provided that you choose R_1 and R_2 so that V_{E1} can go a few hundred millivolts above the maximum upper threshold voltage, V_{TU} (maximum), of IC_{1A} when V_I is a minimum. This feature is necessary to ensure that Q_1 does not saturate when V_C approaches V_{TU} (maximum) as C_1 charges.

The values $R_1=R_4=22$ kΩ and $R_2=R_3=33$kΩ meet these requirements and provide an optimum range for V_I. These values should provide reliable operation for $V_{CC}=5\text{V}\pm5\%$ and IC_{1A} and $IC_{1A}=$ 74HC14, but you may need to recalculate the values if you use a different supply voltage or a different inverter.

Two possible devices to use for IC_{1A} are the 74HC14 and the 4093. The 74HC14 is preferable because the minimum to maximum variation in its hysteresis voltage is only about 3.3 to 1, whereas the variation in V_H for the 4093 is approximately 6.7 to 1. However, the 4093 allows operation at supply voltages greater than 5V, but take care to avoid base-emitter breakdown of Q_1 and Q_2 at higher supply voltages.

Power consumption is low. For example, with $C_1=100$ pF, the maximum current draw is 570 µA at the point of maximum frequency, which is approximately 200 kHz. The maximum practical operating frequency is limited to around 500 kHz ($C_1=10$ pF, $R_6=5.6$kΩ), where the relationship between V_I and the duty cycle starts to become noticeably nonlinear. (DI #2461)

Figure 3



Although the frequency of the output waveform varies with the input voltage, the PWM circuit exhibits a linear relationship between input voltage and output duty cycle.

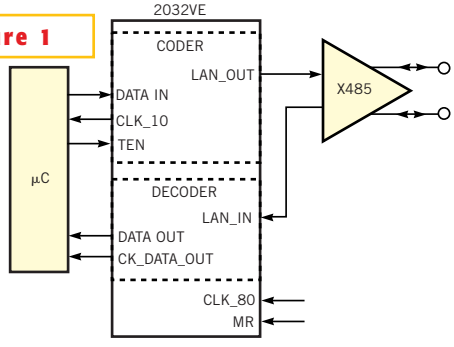
To Vote For This Design, Circle No. 458

Manchester co-decoder fits into 32-macrocell PLD

Antonio Di Rocco, Siemens ICN Spa, L'Aquila, Italy

Manchester encoding is common, and this scheme erases the dc-spectrum component present in an NRZ signal in baseband transmissions. An important application is in Ethernet-interface adapters, in which several kinds of media-attachment units interface with OSI layers. Many commercial transceivers work on all physical layers of the IEEE 802.3 standard. **Figure 1** and the corresponding source code realize a customized version of the 10BaseT standard in which the physical layer is a coupled stripline in a backplane. **Figure 1** shows the simple schematic of the LAN controller.

Figure 1



A 32-macrocell PLD implements a complete Manchester co-decoder at a 10-MHz bit-speed rate.

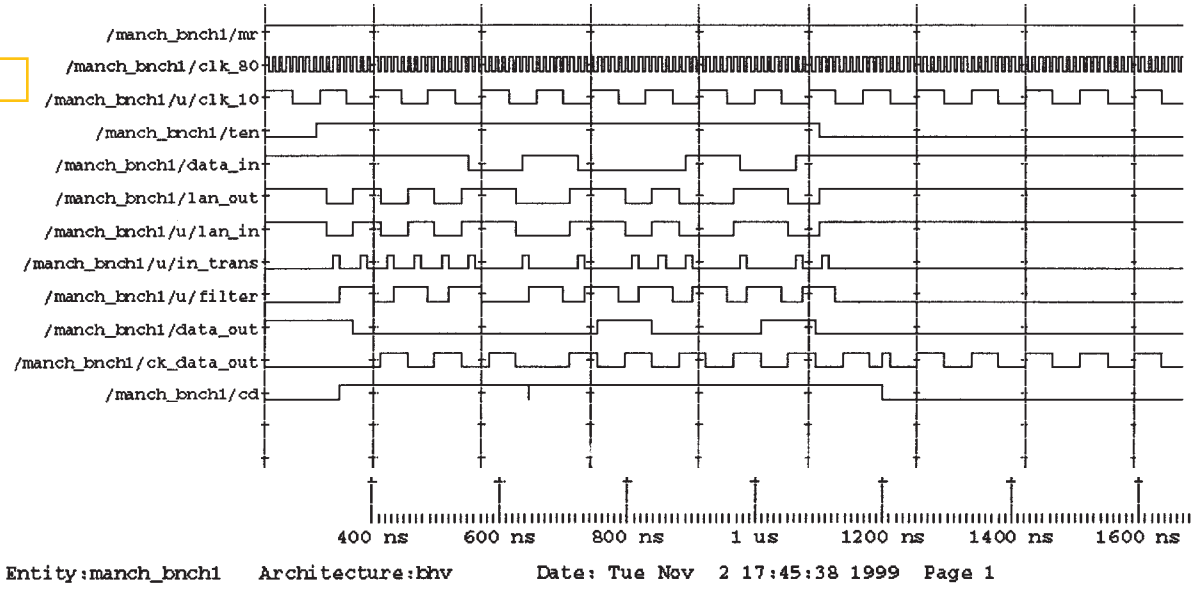
With an 80-MHz external clock, the 32-macrocell PLD implements a complete Manchester co-decoder at a 10-MHz bit-speed rate. You can download the VHDL source code from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2462.

The Manchester coder comprises an XOR gate between the transmitted data from the µC data_in) and the internal 10-MHz clock. Both the data_in and coded output lan_out signals are synchronous with the 10- and 80-MHz clocks, respectively. Asserting a high at the "10" input enables the coder.

The decoder's operation is more complicated than that of the encoder. A behavioral simulation (**Figure 2**) shows the internal signals that are involved in the decoding process. Note that the spike on the "cd" signal is not a true spike; it appears only in the behavioral simulation and disappears in postlayout simulation. The signal "in_trans" is a short trigger pulse that occurs at every positive and negative "lan_in" transaction. These pulses trigger a filter maker that generates an impulse signal called filter, and each pulse of this filter signal lasts 75% of the bit interval. The end of each filter pulse marks the start of a pulse of a 10-MHz recovered clock. The design generates decoded data by sampling the data stream with the rising edge of the recovered clock. After a bit violation, or when "data_in" remains a one or a zero for more than 100 nsec, the system deasserts the carrier-detect signal, "cd." Many µC families require that five or six recovered clock pulses are present after the system deasserts the carrier-detect signal. To conserve space in the PLD, this design roughly multiplexes the recovered clock and the 10-MHz system clock. (The 68360 µP tolerates one pulse with no aspect of duty cycle.) The carrier-detect signal is the multiplexer controller. The 80-MHz clock has no stability requirements, and the system tolerates jitter on 10-MHz Manchester-coded signals. (DI #2462)

To Vote For This Design, Circle No. 459

Figure 2



A behavioral simulation of the decoder's operation shows the internal signals involved in decoding.

Input-protection scheme tops other approaches

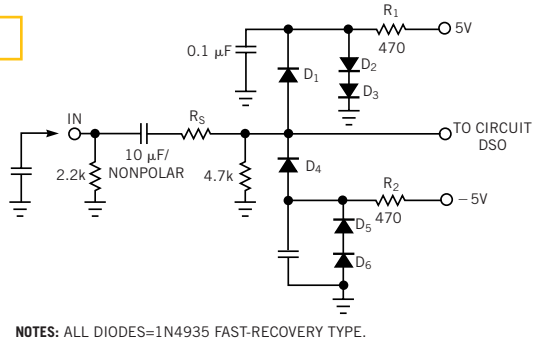
Kannan Natarajan, Mediatronix Private Limited, Kerala, India

You typically accomplish overvoltage or surge protection at circuit inputs by connecting diodes to the supply rails, connecting zener diodes to ground, or connecting transzorbs to ground. Unfortunately, for high-energy surges at the inputs, connecting diodes to the supply rails results in surges in supply lines and affects other components because of the inductance of supply rails, regulator shutdown, and so on. Zener diodes have limited surge capability, and transzorbs have large capacitance and are therefore suited only for low-bandwidth applications.

The circuit in **Figure 1** has many advantages over these approaches: wide bandwidth and low capacitance; high surge-energy handling because the diodes can carry 50A peak; 1A continuous current; and fast response. Also, the circuit doesn't affect the supply rails and is suitable for protecting multiple I/O lines because the lines can share the bias voltage. You can further improve the response time by using faster diodes; a ground plane; low-inductance, short connections; and close, high-frequency decoupling.

The circuit reverse-biases D_1 and D_4 to bias voltages of ±1.2V, respectively. R_1 and R_2 bias two pairs of diodes, D_2/D_3 and D_5/D_6, respectively, to generate the ±1.2V. R_1 and R_2 prevent input surges from reaching the supply rails. The surge shunt path consists of D_1, D_2, and D_3 to ground or D_4, D_5, and D_6 to ground, depending on the surge's polarity. Because of the ±12V bias-voltage settings, the circuit works with maximum input signals of ±1V. Above this level, D_1 and D_4 start to leak and distort the signal. The circuit was tested using a 100-µF/50V test capacitor charged to 30V and then discharged to the input. A DSO captured the results (**Figure 2**). In **Figure 2a**, with R_S= 100Ω, the peak is approximately 3.5V, and settling to around 2V occurs within 15 nsec. **Figure 2b** shows the same response as **Figure 2a** but with a horizontal scale of 1 msec/div. **Figure 2c** is also the response under the same conditions but shows the long-term response and the coupling-capacitor recovery. If you let R_S=0, the peak rises to 10V and settles within 500 nsec. Thus, some small resistance, such as 100Ω, is necessary for R_S. (DI #2463)

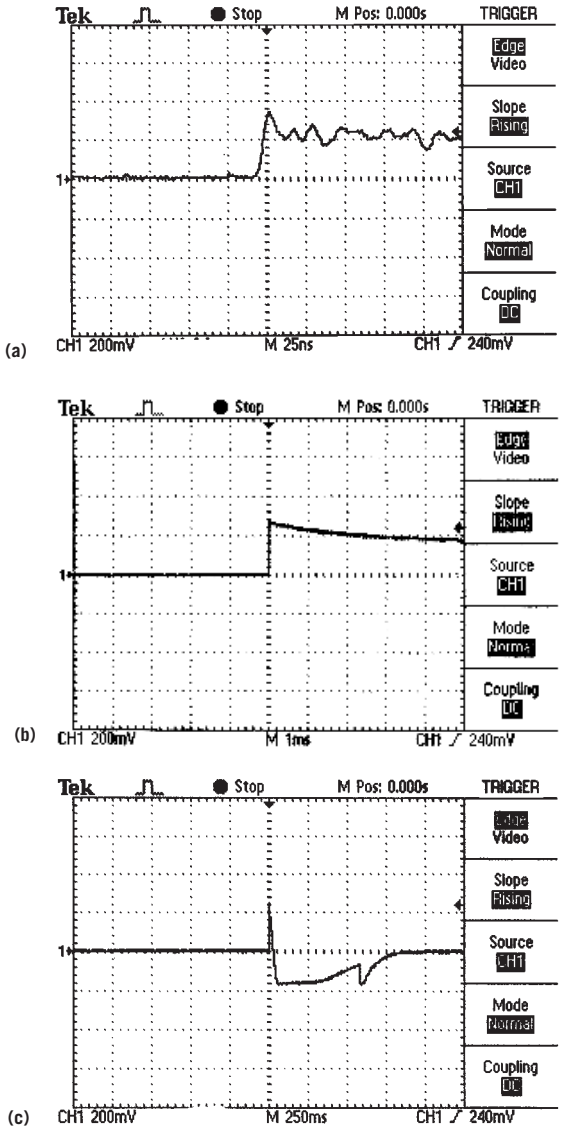
Figure 1



NOTES: ALL DIODES=1N4935 FAST-RECOVERY TYPE.

Two surge shunt paths, consisting of D_1, D_2, and D_3 to ground or D_4, D_5, and D_6 to ground, provide overvoltage protection.

Figure 2



Tests with a 30V charged capacitor at the input show the circuit's response with a horizontal scale of 25 nsec/div (a) and 1 msec/div (b). The long-term response shows the recovery of the coupling capacitor (c).

Level-shifting nixes need for dual power supply

Ron Olmstead, Westcor, Sunnyvale, CA

The AD736 true-rms-to-dc converter is useful for many applications that require precise calculation of the rms value of a waveform. This converter can determine the true rms value, the average rectified value, or the absolute value of a myriad input waveforms. Basically, all applications require both a positive and a negative power supply. According to the data sheet, you can use the device with a single supply by ac-coupling the input signal and biasing the common pin above ground. However, the ability to process only ac signals is a major performance limitation. You can lift this limitation by using a level-shifting approach (**Figure 1**). This approach requires more circuitry, but it removes the ac-only input-waveform restriction.

The circuit consists of three sections. The first is a differential amplifier that adds the level-shifting offset, V_{REF}, to the input waveform. This amplifier's primary function is to level-shift the waveform, but it can also provide gain and filtering if necessary. The output of the op amp needs to swing to the value of V_{REF} minus the peak negative swing of the input waveform times the gain of the op amp ($V_{REF}-(A\cdot V_{IN})$) and to the value of V_{REF} plus the peak positive swing of the input voltage times the gain of the op amp ($V_{REF}+(A\cdot V_{IN})$). By adjusting the value of V_{REF} and the gain of the op amp, you can eliminate the need for an expensive rail-to-rail op amp and can then use any single-supply op amp. All three sections use the same level-shifting offset, V_{REF}.

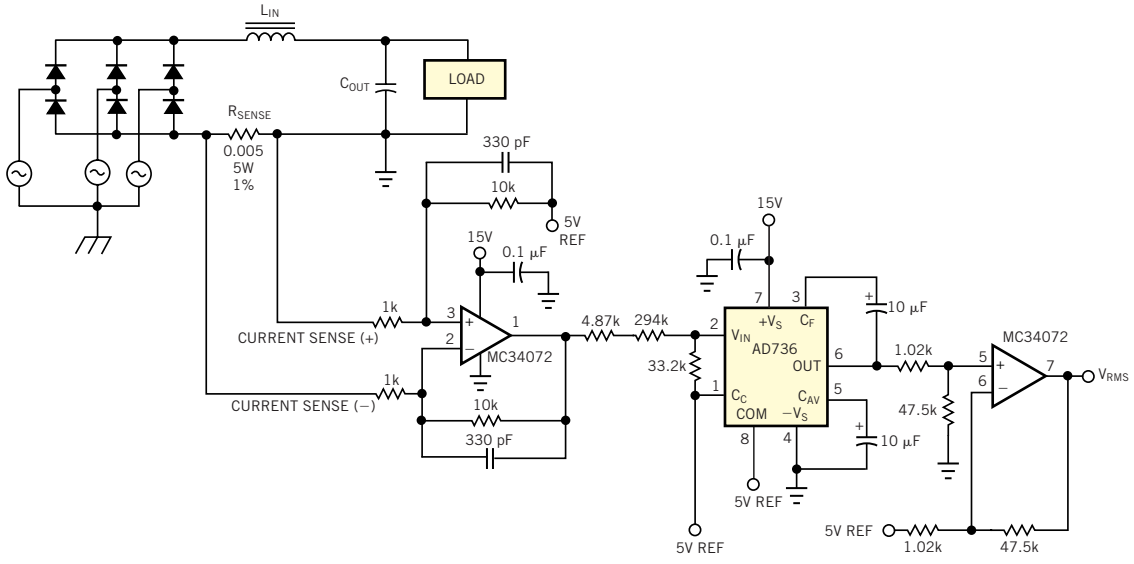
The second section is the rms-to-dc-converter stage. The output of this stage is the dc (rms) value of the input waveform plus the offset value (V_{REF}). The input voltage divider reduces the amplitude of the input waveform. For successful rms-to-dc conversion, the circuit must keep the voltage going into the AD736 within the specified range, which is 1V rms for a V_{CC} of ±5 to ±16V. If amplitude reduction is unnecessary, you can eliminate these resistors and simply ground Pin 1 of the AD736. The offset voltage needs to connect to the AD736 (**Figure 1**). This connection provides a reference for the circuit that is above ground. The AD736 cannot provide accurate calculations for inputs that go below or even equal the converter's negative rail, $-V_S$. V_{REF} should be greater than the peak negative swing of the input waveform. V_{CC} should be greater than V_{REF} plus the peak positive swing of the input voltage.

The third section of the circuit is a level-shifting circuit, which subtracts V_{REF} from the output of the AD736. The last-stage differential amplifier can provide any necessary gain, and you can use this gain to eliminate the need for a rail-to-rail op amp.

The application of the circuit in **Figure 1** is to measure the current draw of a power supply and detect overcurrent conditions. For this application, only a positive power supply was available. The input op amp raises the amplitude of the input signal and filters out any noise greater than 5 kHz. The power-supply input is a three-phase 60-Hz signal, so the ripple frequency is 360 Hz. By providing gain in this first stage and a 5V level shift, any single-supply op amp is suitable. Also, a rail-to-rail op amp is unnecessary. The circuit divides down the output of the first stage to be sure not to exceed the input voltage range of the AD736. The output amplifier provides gain to the dc signal and level-shifts the signal back to a ground-referenced signal. Again, the gain of this op amp produces a signal with an amplitude suitable for use with any single-supply op amp. (DI #2466)

To Vote For This Design,
Circle No. 461

Figure 1



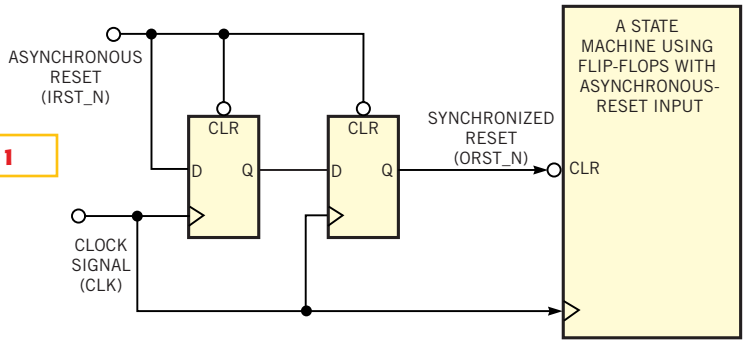
Level-shifting the input to an rms-to-dc converter allows you to use the converter with only positive supply voltages.

Synchronize asynchronous reset

Willy Tjanaka, Philips Semiconductors, Sunnyvale, CA

Synchronous reset and asynchronous reset are both common reset mechanisms for state machines, and the reset circuit in **Figure 1** combines the advantages of each. Synchronous reset has the advantage of synchronization between clock and reset signals, which prevents race conditions from occurring between the clock and the reset signal. However, synchronous reset does not allow a state machine to operate down to a dc clock because reset does not occur until a clock event occurs. In the meantime, uninitialized I/O ports can experience severe signal contention.

Figure 1



A simple circuit combines the advantages of asynchronous and synchronous resets.

Asynchronous reset has the advantage of allowing state machines to operate down to dc clock. This operation is possible because asynchronous reset immediately initializes the state machine when a reset signal occurs independently of the clock. Unfortunately, asynchronous reset may cause a race condition between the reset signal and the clock. Race conditions can cause problems, including metastability or wrong-state initialization.

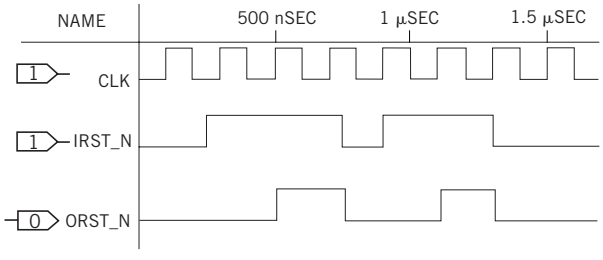
The reset circuit in **Figure 1** asserts the reset signal immediately after detecting the asynchronous reset signal. However, the circuit also synchronizes the reset release with the clock. The circuit uses this synchronized asynchronous-reset signal to drive a state machine that uses flip-flops and the asynchronous-reset input.

LISTING 1—VERILOG DESCRIPTION OF THE SYNCHRONIZED ASYNCHRONOUS-RESET CIRCUIT

```
module reset (clk, irst_n, orst_n);
// Willy Tjanaka
// Rev. 1.0, 17 October 1999

input  clk, irst_n;
output orst_n;
reg    orst_n, mrst_n;

always @ (posedge clk or negedge irst_n)
begin
  if (!irst_n)
  begin
    mrst_n <= 1'b0;
    orst_n <= 1'b0;
  end

  else
  begin
    mrst_n <= irst_n;
    orst_n <= mrst_n;
  end
end

endmodule
```

The reset circuit consists of two back-to-back D flip-flops that synchronize the asynchronous reset signal. In addition, the asynchronous reset causes the D flip-flop outputs to immediately go low. **Figure 1** also shows the corresponding signal names for the Verilog description of the circuit (**Listing 1**), which you can immediately incorporate into a design or simulation. **Figure 2** shows the simulation waveform from the Verilog code in **Listing 1** using Altera Max+PlusII. Observe that the circuit immediately asserts the output-reset signal (orst_n) when the system asserts the input reset signal, irst_n. Also notice that the reset release is synchronous with the clock within two cycles. (DI #2465)

Figure 2



The simulation waveform shows that the circuit asserts the output reset signal, orst_n, immediately after the system asserts the asynchronous input signal, irst_n, and shows that the reset release is synchronous with the clock signal within two cycles.

To Vote For This Design, Circle No. 462

Circuit resolves 0.1-fF change from 100 pF

Derek Redmayne, Linear Technology Corp, Milpitas, CA

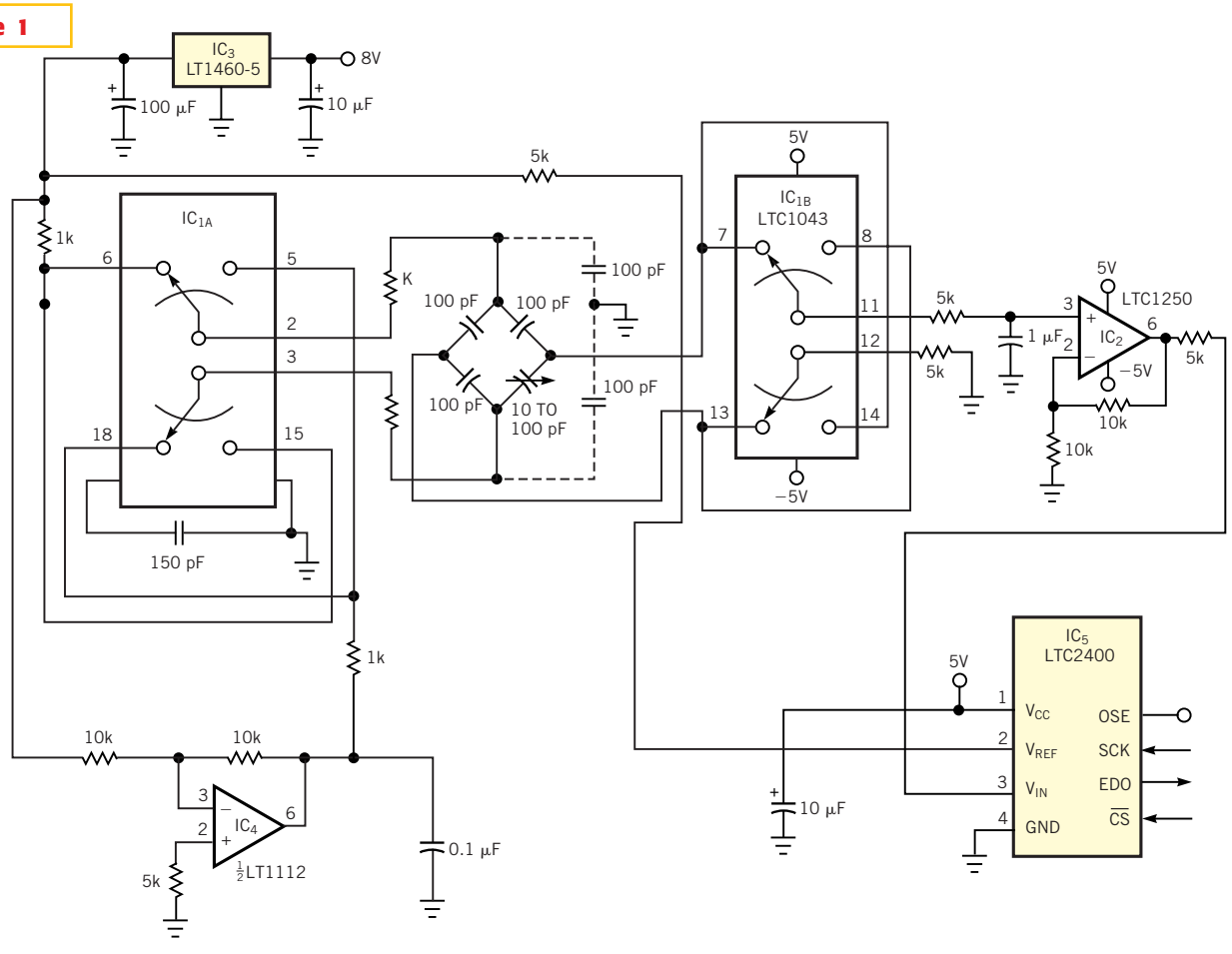
The circuit in **Figure 1** can resolve 0.1-fF changes in a 100-pF bridge element and can accommodate large-scale changes in the bridge without adjustment. You can use changes in capacitance to measure applied pressure, rotation, torque, liquid level, the water content of toast, and a host of other things. Many variants of the circuit are possible.

IC_1, an analog switch, provides both bridge excitation and synchronous rectification. A chopper-stabilized amplifier, IC_2, which the circuit configures for a gain of 2, buffers and amplifies the output of the synchronous rectifier, IC_{1B}. No amplification occurs before the rectification stage. IC_1's internal oscillator and an external capacitor determine the frequency of the square-wave excitation signal—in this case, 20 kHz—that the circuit delivers to the bridge via IC_{1A}.

If, as in this case, the excitation waveform is essentially a square wave, the system is not oversensitive to oscillator frequency and thus not oversensitive to the supply voltage. This circuit reduces the slew rates of the excitation to reduce EMI and to prevent transient load changes from disturbing the reference and buffer, IC_3 and IC_4, respectively. Further significant reductions in the slew rate cause the frequency of commutation in IC_1 to affect the output. A delta-sigma ADC, IC_5, resolves the output of amplifier IC_2 to approximately 1 ppm.

You can use a capacitance change of this magnitude to measure subtle changes in dielectric constant, such as

Figure 1



Using an analog switch, IC_1; a chopper-stabilized amplifier, IC_2; a reference, IC_3; a buffer, IC_4; and a delta-sigma ADC, IC_5, this circuit can resolve 0.1-fF changes in a 100-pF bridge element.

those that may occur in oil due to contamination. For example, if you create a capacitor using 5×5-in. plates that are 1/4 in. apart, the dielectric constant, K, of the media between the plates could be resolvable over the range of 1 to 4.5 (22.48 to 101.2 pF). A change in K of as little as 0.000004 would be measurable. The rigidity and separation between these plates would have to be constant and stable because movement of as little as 0.3 µm would produce the same 0.1-fF change. The use of low-thermal-coefficient materials would be necessary to maintain this separation, but this measurement is practical with good mechanical design.

Other capacitor geometries are possible, of course. For example, the plates of the capacitor could be coplanar interleaved fingers etched onto an insulator, and the unknown dielectric could either touch the surface or be distanced with an insulator. Also, many configurations of bridges are possible. For example, you could devise bridges to compare two substances. You could also construct bridges to deflect the field toward the plates of one capacitor or another, depending on the K of some substance running through channels—for example, to compare the dielectric constant of two liquids. Assuming good sensor design, E-field (ac) measurements can be comparatively free of the effects, including drift, hysteresis, creep, nonlinearities, thermocouple effects, self-heating, leakage, and electromigration that compromise dc measurements.

The circuit in **Figure 1** is usable, but you can improve the circuit's long-term drift and temperature stability by deriving a timing signal from a quartz oscillator. Note that resolving small capacitance changes requires diligent attention to parasitics. If a single variable capacitor, as in this example, sits remotely from the other bridge elements, it is recommended that you use shielded cable with the shield driven from either the other bridge arm or even a third arm (see the dashed line in **Figure 1**). If this situation occurs, you should route the lower end of the bridge separately to the external capacitor. If you plan to bundle these cables, you should use the upper arm of the excitation to shield the excitation to the lower end of the unknown capacitor. This cable capacitance loads and hence attenuates the bridge drive, and you should perhaps use a separate synchronized analog switch to sense these loads to provide a reference signal for ratiometric operation.

Alternatively, you can ground the shield if the bridge is symmetrical about the midpoint. If the bridge is asymmetrical, the inputs to IC_1 see a substantial ac component. You can potentially drive an asymmetrical bridge with a transformer and ground the midpoint of the third arm to reduce the common mode seen in the taps. (DI #2464)

To Vote For This Design,
Circle No. 463

design ideas

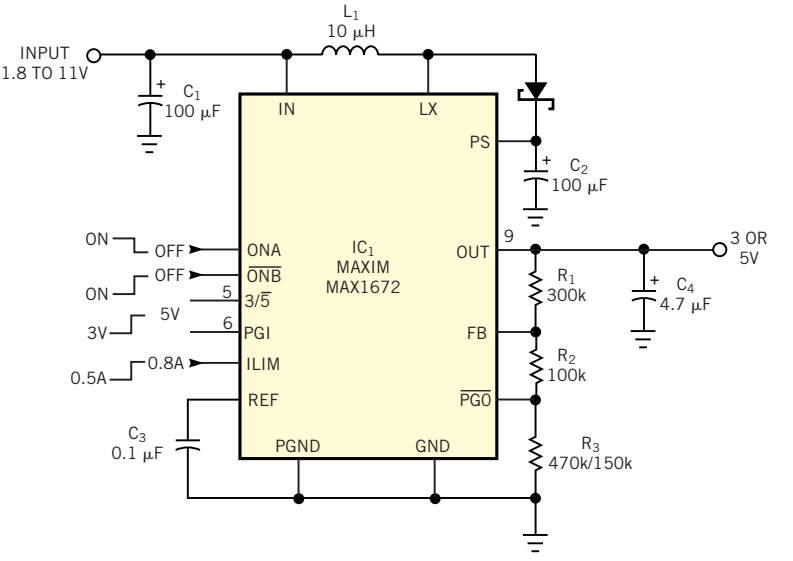
Edited by Bill Travis and Anne Watson Swager

Dual-voltage supply powers SIM card

Larry Suppan, Maxim Integrated Products, Sunnyvale, CA

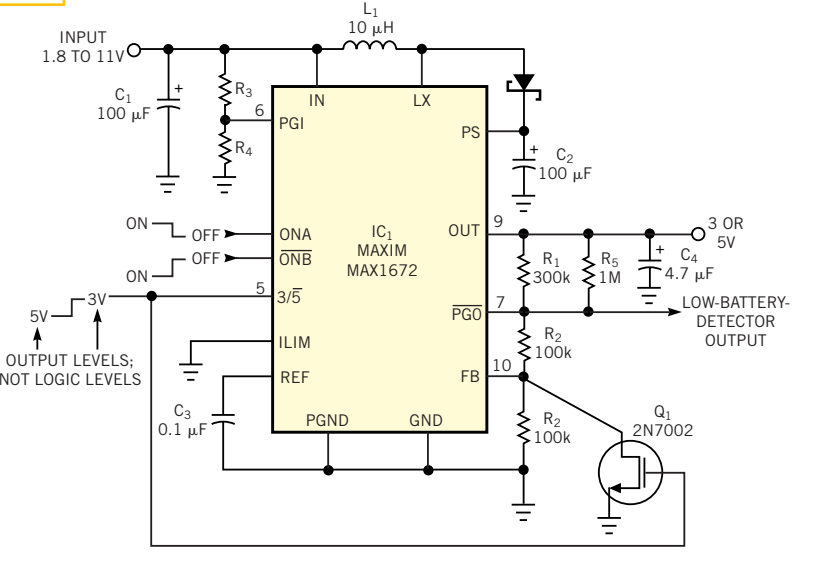
Global-system-for-mobile-communication phones have a subscriber-identification module (SIM) that allows local wireless providers to recognize the user and his or her billing information. Although most SIMs are changing to 3V operation, they also accommodate 5V as well during the transition. IC_1 in **Figure 1** combines a step-up dc/dc converter with a linear regulator, allowing it to regulate up or down for a range of input voltages. It offers hardware-selectable fixed outputs of 3.3 and 5V; however, 3.3V is out of spec for a 3V SIM card. With properly chosen $R_1/R_2/R_3$ values, you can switch the regulated output between 3 and 5V (or any other two outputs within the allowed range) by applying digital control to the power-good input (PGI). The power-good output (PGO), the output of an internal comparator, then changes the IC's feedback by grounding the node between R_2 and R_3. If the power-good comparator is in use, you can implement the digital control using the $3/\overline{5}$ input and an external MOSFET (**Figure 2**). (DI #2468)

Figure 1



You can obtain a regulated 3 or 5V output, according to digital control applied to the power-good input (PGI).

Figure 2



This circuit provides the same outputs as the circuit in Figure 1 without tying up the internal power-good comparator.

To Vote For This Design, Circle No. 315

Design formulas simplify classic V/I converter

Dudley Nye, Nye Engineering Co, Fort Lauderdale, FL

FIGURE 1 SHOWS a classic voltage-to-current(V/I) converter. You can select the resistor values such that the output current in the load, R_L, varies only with the input voltage, V_{IN}, and is independent of R_L. The circuit is widely used in industrial instruments for supplying a 4- to 20-mA signal. The circuit has its limitations, however, because the resistor values must be quite accurate to obtain a true current source. The literature describing the circuit provides design methods that are for special cases or are for approximate designs. This Design Idea gives two simple design formulas you can use to determine the component values that produce a true current source. It also provides a general formula for the output current, I_L, for any selection of resistor values, not just the constant-current selection.

For a true current output, I_L, as a function of the input voltage, V_{IN}, you must satisfy the following two equations:

$$I_L = \left(\frac{V_{IN}}{R_1}\right)\left(\frac{R_2}{R_X}+1\right). \quad (1)$$

$$R_3 = (R_2 + R_X)\left(\frac{R_4}{R_1}\right). \quad (2)$$

In **Equation 1**, you can arbitrarily select any four of the terms and then determine the fifth term by solving the resulting equation. In **Equation 2**, you can arbitrarily select either R_3 or R_4 and then determine the unselected resistor after substituting the applicable terms from **Equation 1**. For example, you can solve **Equation 1** for R_2 when I_L=20 mA, R_1=100 kΩ, R_X=0.1 kΩ, and V_{IN}=4V yields R_2=49.9 kΩ. Now, let R_4=100 kΩ and, with **Equation 2**, solve for R_3 as follows: R_3=(49.9 kΩ+0.1 kΩ)=50 kΩ. This example configures a design for the popular current source of 4 to 20 mA. In a second example, if R_X changes from 100 to 400Ω, the feedback changes fourfold, and you would expect that the output current would change fourfold, to 1 to 5 mA. You can check the result by substituting in the general formula for the output current:

$$I_L = V_{IN}(KR_2 + R_X)/\left\{R_L\left[R_1 + R_2 + R_X\left(\frac{R_1+R_2}{R_2}\right) - R_1 \bullet \left(\frac{KR_2+R_X}{R_2}\right)\right] + R_X(R_1+R_2)\right\}, \quad (3)$$

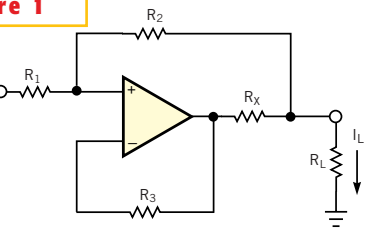
where $K = 1 + \frac{R_3}{R_4}$.

When the complete coefficient (the terms inside the square brackets) of R_L equals zero, a true current source results, and **equations 1** and **2** are valid. Note that substituting the values from the first example above forces the coefficient to zero. Substituting the values from the second example above results in the following expression:

$$I_L = 75.25\frac{V_{IN}}{0.06R_L + 59.96}. \quad (4)$$

With R_L=0.2 kΩ and V_{IN}=4V, IL=5.019 mA. Then, with V_{IN}=0.8V, I_L=1.003 mA. Thus, after changing the feedback resistor by 4-to-1, you still have currents close to the 1- to 5-mA standard. Note also that I_L=5.02 mA when R_L=0Ω; thus, the circuit is still almost a perfect current source. This result is unique, as you can convert from 4 to 20 ma to 1 to 5 mA by changing only one resistor. You can configure the less used standard of 10 to 50 mA by making R_X=100/2.5=40Ω. (DI #2471)

Figure 1



Design formulas make this classic V/I converter easy to use.

TO VOTE FOR THIS DESIGN, CIRCLE NO. 316

Rail-to-rail op amp provides biasing in RF amp

Frank Cox, Linear Technology Corp, Milpitas, CA

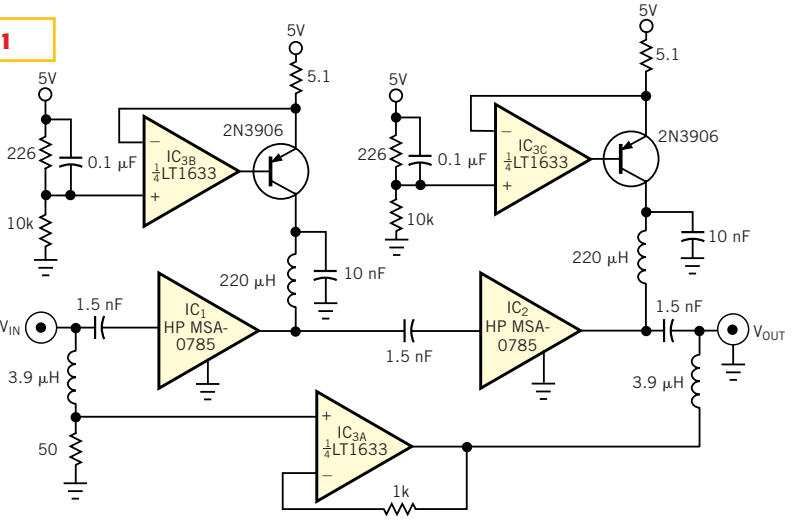
IT IS OFTEN USEFUL to monitor the dc level of an RF signal. However, most RF systems use capacitive coupling; thus, the dc information is lost. The circuit in **Figure 1** is an RF amplifier comprising two monolithic microwave integrated circuits (MMICs), IC_1 and IC_2, and a quad rail-to-rail op amp (IC_3, an LT1633). IC_{3A} restores the dc level at the output. Inductors at both the input and the output of the op amp isolate the amplifier from the RF signal. The isolation is good practice, because frequencies higher than the bandwidth of the op amp can undergo rectification in the amplifier's input stages, thereby introducing offset. MMICs IC_1 and IC_2 are Hewlett-Packard HP MSA-0785 devices, which have an inverting gain of 13 dB; the result is a total gain of approximately 26 dB and a noninverted signal. IC_1 and IC_2 have a 3-dB bandwidth of approximately 2 GHz. The 1.5-nF blocking capacitors set the low-frequency cutoff at 2 MHz.

IC_1 and IC_2 have a 1-dB compression point of 4 dBm, or 1V p-p, into 50Ω, allowing for an input level as high as 18 mV

rms. The maximum output current of IC_{3A}, typically 40 mA with a single 5V supply, limits the dc level on the output to 2V into 50Ω. The output saturation (low) voltage of the LT1633, typically 40 mV, sets the minimum pedestal voltage. IC_1 and IC_2 use constant-current bias sources to stabilize their gain with respect to temperature. Two other sections of the quad op amp, IC_{3B} and IC_{3C}, form active 22-mA current sources. You can make the voltage dividers on the noninverting inputs of IC_{3B} and IC_{3C} adjustable to trim the gain of the RF amplifier. The rail-to-rail inputs of IC_3 allow the circuit to operate to within 110 mV of the positive rail. (DI #2467)

To Vote For This Design, Circle No. 317

Figure 1



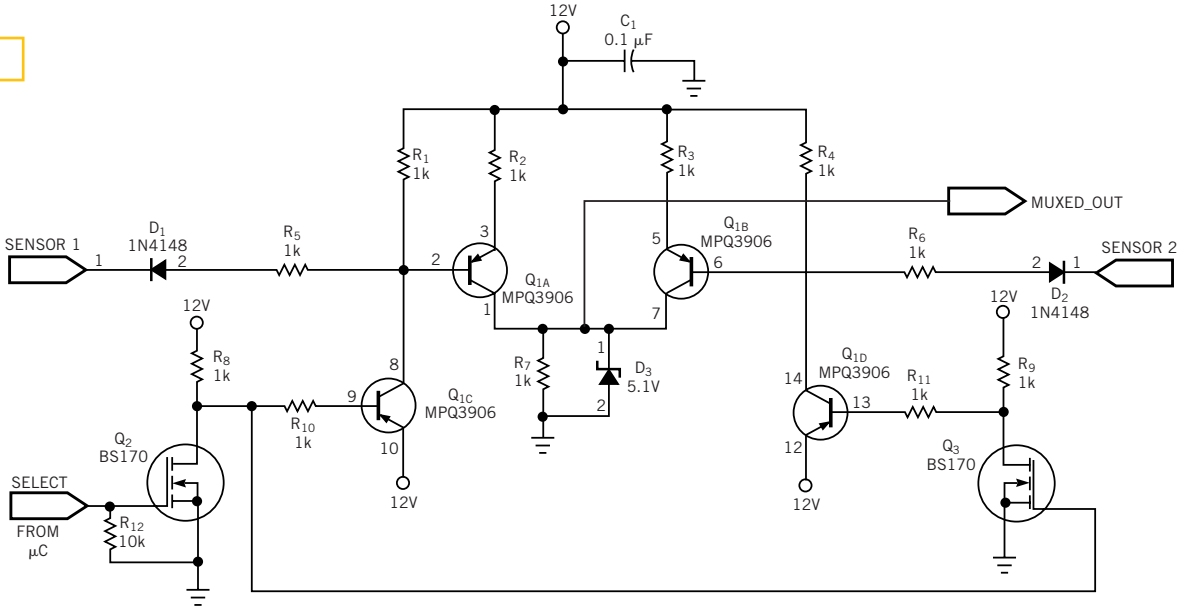
A simple op-amp-follower circuit with the aid of inductive blocking restores the dc level of an RF signal.

Circuit multiplexes automotive sensors

Adil Ansari, Delphi-Delco Electronics, Kokomo, IN

Often, a µC limits the number of input-capture lines to accommodate the various types of automotive sensors with pulsed outputs, such as vehicle- and engine-speed sensors. The circuit in **Figure 1** uses discrete components to multiplex two sensors with open-collector outputs into a single output, thereby sharing one input-capture line of the µC. The µC selects the sensor whose output you will measure. You can apply this approach to sensors whose outputs are

Figure 1



You can multiplex the output signals from two sensors into one input-capture line in a µC.

amenable to time-sharing and do not require continuous monitoring, such as position sensors. In **Figure 1**, Sensor 1 and Sensor 2 are outputs from two sensors using npn transistors with open-collector outputs. To enable Sensor 1 or Sensor 2, Q_{1A} or Q_{1B}, respectively, must turn on. A logic-low signal from the µC on the Select input turns off Q_2 and Q_{1C}. When Sensor 1 input goes low, D_1 forward-biases, and Q_{1A} turns on, providing a high signal on MUXED_OUT. When Sensor 1 input turns off (high-impedance state), Q_{1A} turns off, providing a low signal on MUXED_OUT. Therefore, when the Select input is low, MUXED_OUT produces pulses that are inverted but synchronized with the Sensor 1 pulses. At the same time, Q_3 and Q_{1D} are on, turning off Q_{1B} and disabling the Sensor 2 input.

Similarly, when Select goes high, Q_2 and Q_{1C} turn on, turning off Q_{1A} and disabling the Sensor 1 input. At the same time, Q_3 and Q_{1D} turn off, allowing the Sensor 2 signal to turn Q_{1B} on and off when Sensor 2 switches on (low) and off (high-impedance state), respectively. Therefore, MUXED_OUT produces pulses synchronized with the Sensor 2 input. You can change the values of R_1, R_4, R_5, and R_6 to meet the sensors' requirements. D_3 clamps MUXED_OUT to CMOS/TTL levels. The use of the MPQ3906, containing four pnp transistors in one package, minimizes the number of components. Similarly, you can obtain arrays of 1-kΩ resistors in a single package. (DI #2469)

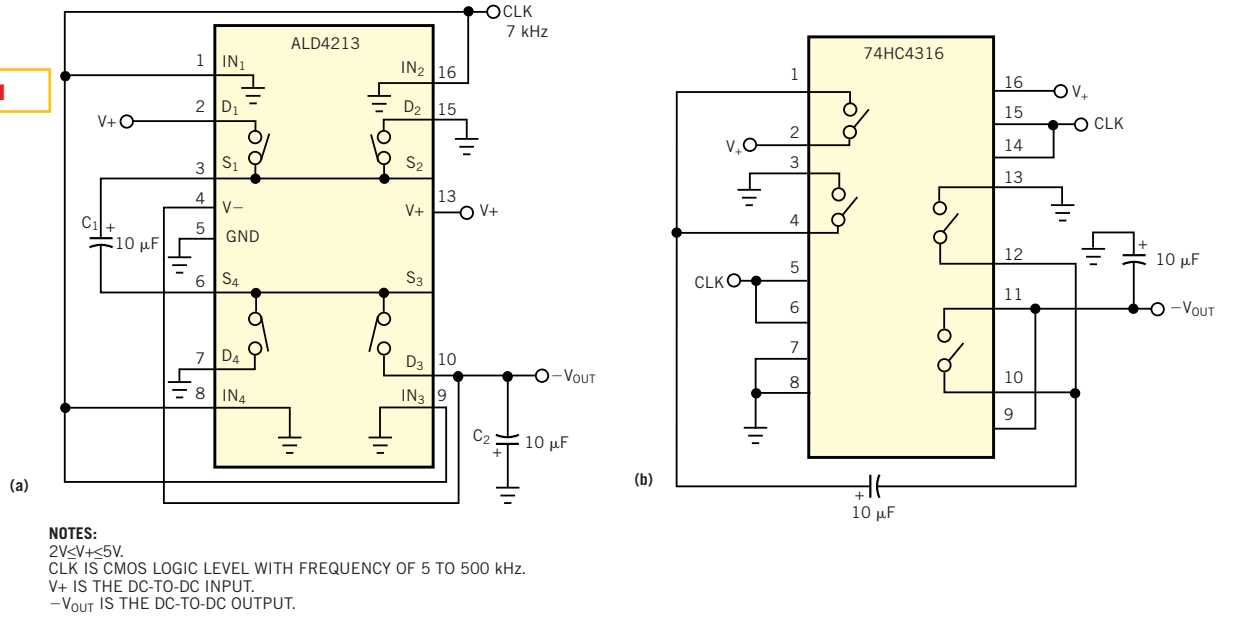
Analog switch acts as dc/dc converter

John P Skurla, Advanced Linear Devices Inc, Sunnyvale, CA

Many low-current devices that require 65V supplies can operate reliably in a single 5V power-supply environment if you use an appropriate localized dc/dc converter to generate the −5V bias. Often, the capabilities and advantages of these 5V ICs far outweigh the minor inconvenience and added costs of an additional −5V-converter function. Many companies manufacture dc/dc-converter ICs and modules in a variety of power ratings and footprints. However, these typical dc/dc converters can be overkill for simple, single-chip applications that require only a negative bias voltage with low operating currents. For these applications, typical negative- voltage requirements range from −4 to −6V with a supply current of 1 mA, and requirements for the −5V supply are generally noncritical.

A lower cost alternative to conventional dc/dc converter modules for generating negative dc voltages from a posi-

Figure 1



NOTES:
2V≤V+≤5V.
CLK IS CMOS LOGIC LEVEL WITH FREQUENCY OF 5 TO 500 kHz.
V+ IS THE DC-TO-DC INPUT.
$-V_{OUT}$ IS THE DC-TO-DC OUTPUT.

Using an analog switch with two external capacitors and an external clock is a viable way to produce 25V from a 5V input for low-power, −5V needs. One approach uses only one phase of the clock (a); a second approach requires both phases (b).

tive supply uses a low-cost quad- semiconductor analog switch and an onboard system clock (**Figure 1a**). This type of voltage converter generates a low-power, negative bias voltage from a 5V input. This circuit emulates charge-pump dc/dc converters, which are suitable for generating an output voltage whose polarity is opposite that of the input voltage. Two charge-storage capacitors are also necessary, as with conventional converters. Unlike the conventional self-contained dc/dc converter approach, this circuit requires a single external clock input to sequence the switches on and off and approximately the same amount of pc-board space. You can tap this clock from any 5V logic-gate output with continuous, regular periods of 5- to 500-kHz signals.

Charge-pump converters operate by first charging up one capacitor and alternately transferring that charge to another capacitor using a switching circuit. The switching circuit in **Figure 1a** alternately charges and discharges C_1 and C_2 to generate a −5V output from a 5V input. Integrated level translators and logic gates inside the ALD4213 analog switch provide the logic translation to convert a single 5V input to a ±5V logic swing.

The circuit closes two switches, S_1 and S_4, under clock control. During the first half of a clock cycle, C_1 charges up to a voltage equal to the input voltage, V+. The next half-cycle of the clock control opens S_1 and S_4 and closes S_2 and S_3. C_1 now connects across C_2 through S_2 and S_3, and the charge on C_1 subsequently transfers to C_2 until the voltage across both C_1 and C_2 is equal. Notice the "inverted" polarity across C_2, which forces the output voltage on C_2 to be V−, or the opposite of V+.

Each subsequent clock cycle, which again begins with the closing of S_1 and S_4, causes C_1 to charge up from the previous voltage to V+. After many repeated clock cycles, the voltage on C_2 remains charged to a value equal to the negative of V+, or close to it; it performs the function of a voltage inverter, which is more commonly called a converter.

An alternative analog-switch-based converter uses the industry-standard 74HC4316 quad analog switch with level translator (**Figure 1b**). The circuit is similar to the circuit in Figure **1a** but has different pin connections. This circuit also requires both phases of the clock. You can use an additional inverting logic gate to generate both clock phases if necessary. The recommended input is a logic clock that has a useful frequency range of 5 to 500 kHz.

Figure 1a's single-phase design costs less than $1 in large quantities. The cost of the circuit in **Figure 1b** can be less than half the cost of the circuit in **Figure 1a** provided that both clock phases exist and that you don't have to add an external logic-gate inverter. You can also integrate analog-switching inverters with other analog functions in a custom ASIC; the ALD4213 and ALD500A are compatible with the company's library of standard cells. (DI #2476)

To Vote For This Design,
Circle No. 319

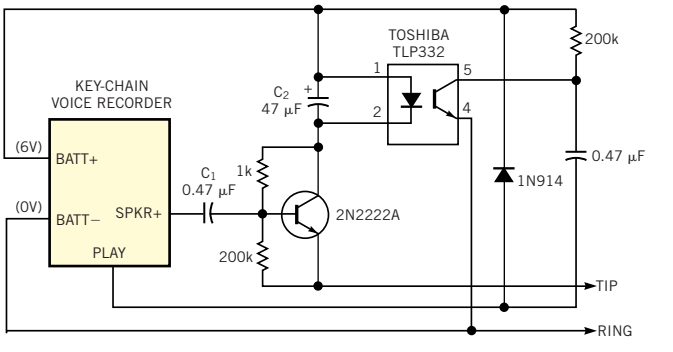
Circuit provides message on disabled phone line

Kevin Kelley, BAE Systems, Greenlawn, NY

Phone companies often disconnect a misbehaving phone line from a complainant's residence for troubleshooting purposes. With the problem between the residence and the central office, the residence is left with a dead phone line and no visible repairman while the line is under repair. The circuit in **Figure 1** adapts a small key-chain voice recorder to the Tip and Ring lines of a phone line that has been disconnected from the central office. The purpose is to play a prerecorded message into any phone on the line when its receiver goes off-hook. A Radio Shack key-chain voice-memo recorder (part number 63-945) or a similar device provides solid-state voice-message storage and playback in a small package and also powers the phone line with its internal 6V batteries. You open and modify the recorder to bring four signals out to the external circuit: Battery (+), Battery (−), Speaker (+ or −), and the Play button contact. You can disconnect the internal speaker to save power.

With all phones on-hook, phone-line

Figure 1



Dead phone line? You can send a prerecorded message to any phone on the defunct line.

current is near zero, keeping the optocoupler off and its transistor open with the voltage at Pin 5 at the battery voltage. The Tip and Ring lines are at 6 and 0V, respectively, to power the phones on the line (Most phones operate on as little as 3V.) Battery drain in this condition is minimal. When a receiver goes off-hook, the line impedance drops, and several milliamps flow through the saturated transistor. The transistor provides a high ac impedance between C_1 and the battery, allowing audio-signal transfer to the line, and provides a low dc resistance to maximize the low battery voltage to the phones. The transistor current turns the optocoupler on, and the voltage on Pin 5 drops to near 0V. This negative edge generates a low pulse into the Play contact, as if you had pressed the Play button. The message plays once in its entirety every time a receiver goes off-hook. C_2 prevents any clicks at the end of the message from restarting the sequence if the receiver goes on-hook before the message ends. (DI #2472)

To Vote For This Design, Circle No. 320

Optocoupler isolates shift registers

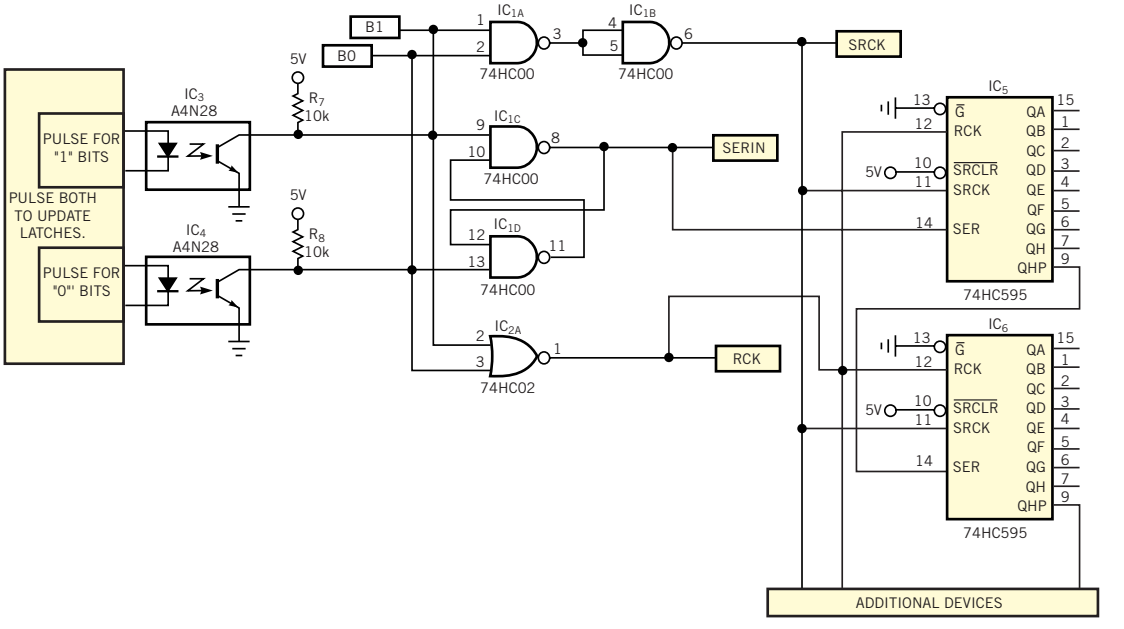
Jim Hartmann, Silent Knight LLC, Maple Grove, MN

Conventional shift registers, such as the 74HC595, require data-, clock-, and strobe-logic signals. The circuit in **Figure 1** needs only two logic signals to isolate and control shift-register devices. For each transmitted bit and one of the two optocouplers receives a short drive pulse: one optocoupler for a high transmitted bit and the other for a low bit and After pulsing all the bits, the circuit a final concurrent 1 and 0 pulse strobes the data into the output registers. Two logic-gate packages on the isolated side of the circuit decode the two negative pulse signals back into data, clock, and strobe. Two NAND gates form an RS latch that captures the data state for the serial input (SERIN). Two more NAND gates form an AND to combine the two pulse sources into the SRCK shift clock. Finally, a NOR gate (or four more NAND gates) produces the RCK strobe. You can cascade the shift-register devices as necessary.

You have no timing constraints on the signals other than observing the maximum data rate of the optocouplers and ensuring an off period between pulses. The final latch pulse also generates an extra rising SRCK edge that you can use to load the first bit of the next sequence. In this case, the optocoupler that turns off last determines the RS latch state for the first bit. You can also ignore the extra clock; it has no effect on the output. Low power consumption is possible by keeping the pulses as short as possible by limiting the LED current and the updating rate. For example, with 40-µsec pulses and 1-msec period, the average drive current is 80 µA. (DI #2470)

To Vote For This Design, Circle No. 321

Figure 1



Optocouplers allow you to isolate and control shift registers with only two logic signals.

Tack a log taper onto a digital potentiometer

Hank Zumbahlen, Analog Devices, Campbell, CA

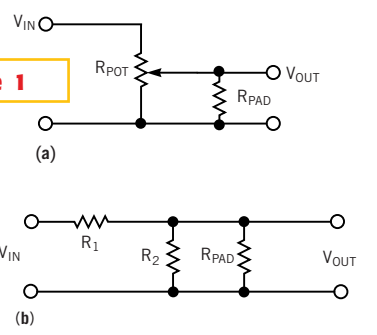
It's sometimes convenient to have digital control of the volume level in an audio system. The use of multiplying DACs (MDACs) is problematic because of the switching noise of the ladder network. This noise comes from the bit switches injecting charge into the signal when they turn on and off. Audio engineers have dubbed this noise "zipper noise" from the sound that results from dynamically adjusting the volume (gain riding). An alternative to an MDAC in this application is a digital potentiometer, such as the Analog Devices AD52XX, AD84XX, or AD7376. You can think of the digital potentiometer as a tapped resistor string. It generates less noise because fewer switches change state. In addition, you can connect the three terminals of the potentiometer anywhere within the common-mode range of the circuit (the supply-voltage range), unlike an MDAC, which generally uses ground as reference.

The primary drawback with using the digital potentiometer for volume control is that it currently comes with only a linear taper. With a linear taper, if the "wiper" is at the midpoint, the signal is only 6 dB less than the maximum. Thus, most of the adjustment range occurs within a small percentage of the range of the potentiometer. This constraint limits the adjustability of the volume setting. The ear responds logarithmically; the volume control should respond similarly. The primary reason for having only a linear taper is the manufacturing problems that the large range of resistance values for a log taper cause. By adding a pad resistor from the wiper of the potentiometer to one end (**Figure 1a**), you can to simulate a log taper. If you split the potentiometer into two resistors, R_1 and R_2, you can redraw the circuit as in **Figure 1b**. The output voltage then depends on the parallel combination of R_2 and R_{PAD}. You define a ratio, r, which is R_{POT}/R_{PAD} ($R_{POT}=R_1+R_2$). By adjusting the value of R_{PAD}, you can modify r, which adjusts the taper, or the attenuation-versus-digital-input code to suit the application. The following expression gives the transfer function of the potentiometer:

$$\frac{V_{OUT}}{V_{IN}}=\frac{R_2 \| R_{PAD}}{R_1+R_2 \| R_{PAD}}$$

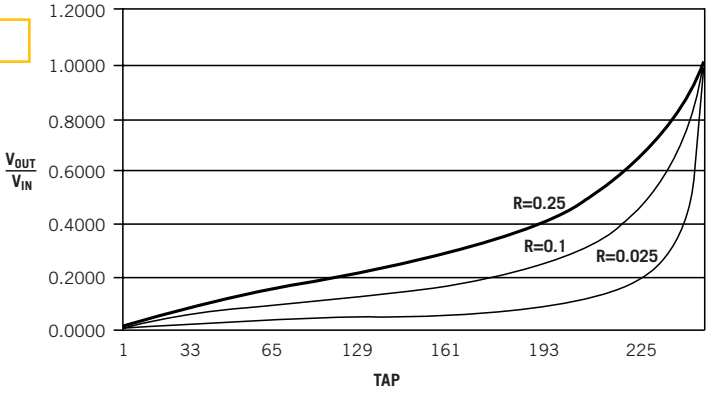
Figure 2 shows the attenuation curves for three values of a pad resistor. As you can see, this trick doesn't give a taper that is so many decibels per step, but it does allow for better low-level settability. You must address a couple of issues. The first is that the end-to-end resistance of the potentiometer changes with the digital code. It varies from the potentiometer resistance at one end (with the wiper at the lower end) to the value of the pad resistance in parallel with the potentiometer resistance at the other end. If you configure the circuit as a typical attenuator and drive it from a low-impedance source, the low pad resistance should not present a major problem. If, however, you are trying to obtain a set resistance value to determine a time constant (or any other application in which the resistor value is critical), this approach may not work well. The second issue involves overvoltage. The three terminals of the potentiometer can be anywhere within the supply range of the IC, which is 5V for the AD52XX and ±15V for the AD72XX family. If you apply overvoltage to one of the pins, even in a transient condition, the IC could latch up because of a parasitic substrate SCR. (DI #2473)

Figure 1



Adding a pad resistor to a digital potentiometer imparts a logarithmic-like taper to the device.

Figure 2



It's not log, but it's close. These curves approximate what you can obtain from an audio-taper potentiometer.

To Vote For This Design, Circle No. 322

design ideas

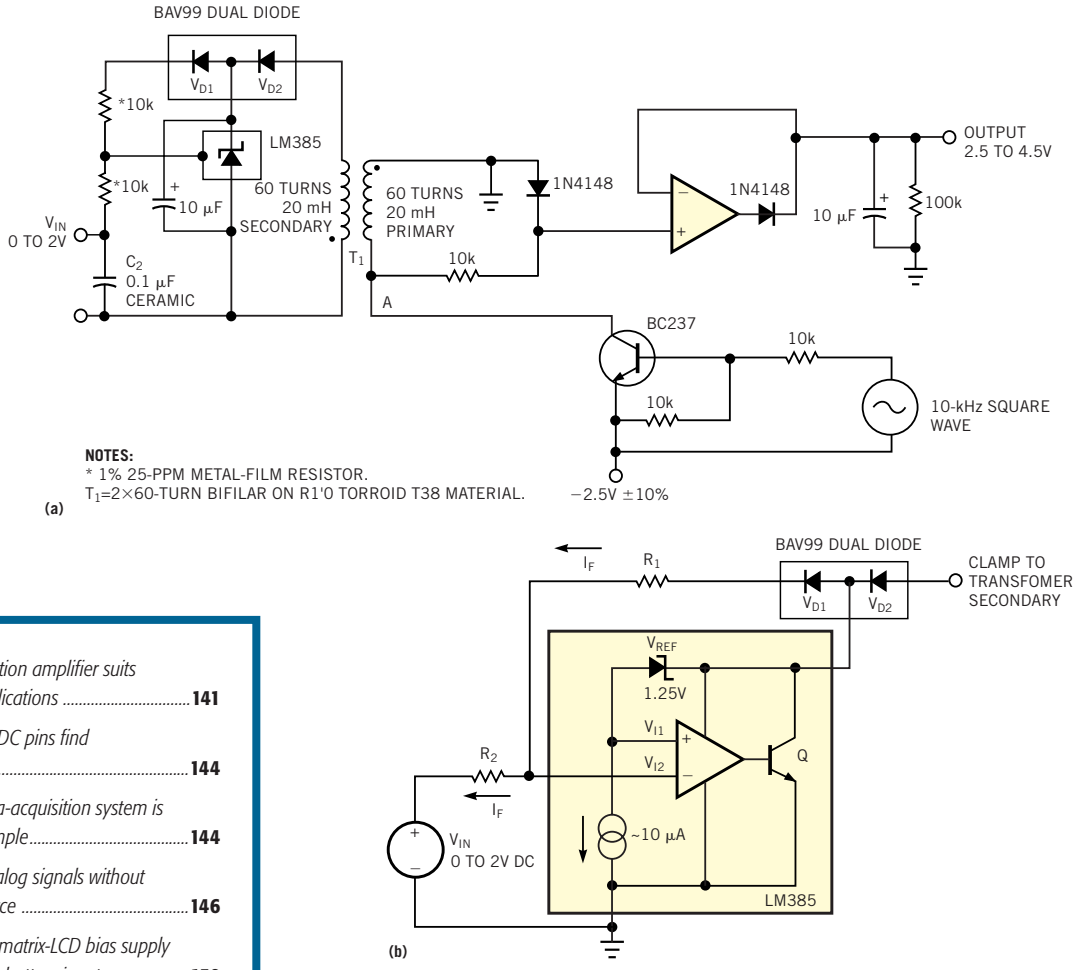
Edited by Bill Travis and Anne Watson Swager

Lost-cost isolation amplifier suits industrial applications

Andrew Russell, Philips Semiconductors, Hazel Grove, England

THE CIRCUIT IN **Figure 1a** is a low-cost isolation amplifier for instrumentation applications that provides as much as 500V of galvanic isolation between input and output. The amplifier uses only one small, low-cost transformer and with little modification lends itself to cost-effective multichannel applications. Input-to-output linearity is around 0.05% for a 2V input signal. The LM385 (National Semiconductor) low-power programmable reference diode, which operates in the shunt mode, and the dual planar BAV99 (Philips Semiconductors) diode are the major circuit components.

Figure 1

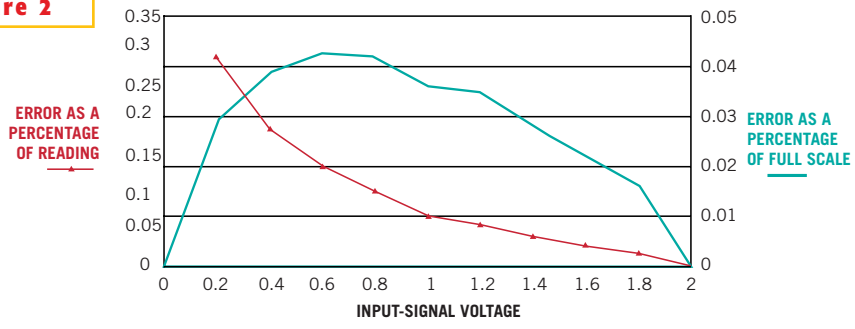


An isolation amplifier for instrumentation applications provides as much as 500V of galvanic isolation and uses only one low-cost transformer (a). The clamp circuit includes dc blocking, which V_{D2} provides (b).

To understand the circuit's operation, you have to first look at the clamp circuit (**Figure 1b**). The LM385 is a shunt regulator that consists of a control amplifier, a current-shunt transistor, and an internal precision 1.25V reference. Two external feedback resistors, R_1 and R_2, set the output voltage. In conventional LM385 applications, the collector of Q is the clamp, or current shunt point. However, in this design, the clamp has to include dc blocking, which V_{D2} provides because the circuit feeds V_{D2} with an ac signal from the transformer secondary. V_{D1}, which is inside the LM385 feedback loop, compensates for V_{D2}. Because V_{D1} and V_{D2} are thermally coupled inside a single package, changes in the forward voltage drop across V_{D2} due to temperature are mirrored across V_{D1}. The transfer function of the clamp circuit is $V_{CL}=2V_{REF}+V_{IN}$.

In the overall circuit, a symmetrical, 10-kHz square wave drives a low-cost BC237 npn transistor, which in turn drives the primary of T_1. In the forward mode, no secondary current flows due to the dc blocking action of V_{D2}. During this phase, primary magnetizing current, which the circuit converts to magnetic field energy, ramps from 0 to −650 µA. When the BC237 turns off due to the drive voltage on its base switching low, the inductive energy in the core dumps into the secondary, causing V_{D2} to be forward-biased and current to flow through the LM385 and back to the other side of the secondary winding. The anode voltage of the LM385 clamps at precisely $V_{IN}+2V_{REF}-V_{D1}$. However, the addition of V_{D2} into the clamp voltage at the anode of V_{D2} compensates for V_{D1}, resulting in a clamp voltage of $2V_{REF}+V_{IN}$. Note that the voltage drop across the diodes during clamping is different due to the large discrepancy in current between the two diodes; V_{D2} carries the peak clamp current, and V_{D1} conducts only the feedback current. However, the circuit largely compensates for the temperature-induced changes in forward voltage drop, which can be a major source of error. Although some mismatch in the thermal tracking of the diodes does occur due to the different forward currents in the diodes, this mismatch is small enough given the accuracies of the circuit that you can consider it a second-order effect.

Figure 2



Linearity performance of the circuit is measurable in terms of percentage of reading and as a percentage of full scale.

The response time of the circuit at Point A is less than 3 msec for 10 to 90% and 90 to 10% input-signal steps. Note that the input signal must be capable of sinking the feedback current, I_F, which for the values in the circuit is approximately 65 µA. With 100-kΩ feedback resistors, the feedback current drops to approximately 8 µA. Drift is largely a function of feedback-resistor stability, LM385 temperature stability, and the thermal tracking of the diodes within the dual-diode package. Average current consumption of the circuit, excluding the peak-detector op amp, is approximately 150 µA. The noise and stability of the -2.5V supply that drives the transformer are not critical, and a simple zener regulator suffices. **Figure 2** depicts the linearity performance as both a percentage of reading and as a percentage of full scale.

In multichannel-isolation applications, you can delete the peak detector stage and feed Point A, or the primary winding, directly into a high-speed, multichannel ADC. For a 10-kHz drive frequency, the clamped waveform tops are typically approximately 20 to 40 µsec long. Sampling should take place at some fixed time after the rising edge, such as 25 to 30 µsec, because the amount of current that shunts through the LM385 decreases during the clamping period as the magnetic energy in the core decays. The LM385 and the associated rectifier diode, V_{D2}, have a dynamic resistance that depends on the current that each device is conducting. The secondary winding resistance times the clamp current also gives rise to a further error term. These two errors combine and are reflected as a slope on the clamped waveform tops appearing on the primary winding when you view the signal at Point A with a scope. For this reason, you need to sample this waveform at some fixed point after the rising edge of the clamped portion of the waveform. Otherwise, errors can result in the readings taken from one sampling event to the next.

In this design, slight overshoot on the rising edge is too fast to cause any problems on the peak detector, but this overshoot would cause a problem at the input to a high-speed ADC. Again, sampling at some fixed period after the rising edge will obviate any problems.

Note that the output clamp voltage measured at Point A varies from 2.5 to 4.5V because of the initial 2.5V offset that stems from the $2V_{REF}$ term in the transfer equation. You have to remove this offset through a subsequent offset removal circuit or by simply subtracting the offset value from the reading in software when using a high-speed ADC. The use of software calibration techniques makes this a viable option in a production environment. (DI #2474)

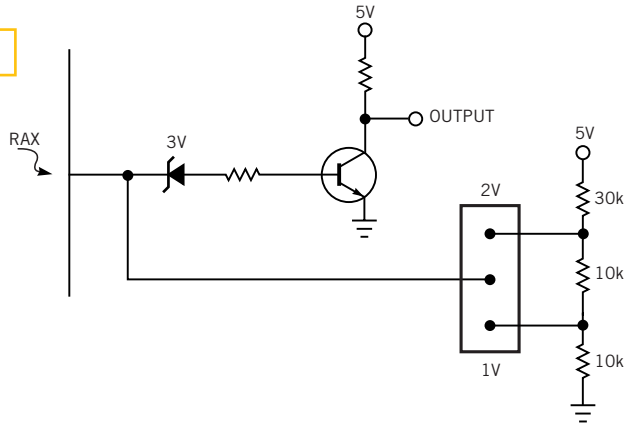
Unused μC ADC pins find second life

Kannan Natarajan, Mediatronix Private Limited, Kerala, India

Some PIC μCs, such as the 12C67X and 16C7X, have more than one analog input channel. If you don't need all the available channels, you can use the unused channels as general-purpose I/O. For example, you can use unused ADC pins for power-on status reading and as an output in normal operation.

In **Figure 1**, the jumper selects a mode—the battery charge voltage, for example—by PIC software at power-on. At power-on, software configures the RAX line as an analog input and reads the voltage—for example, 1V at Digit 0 and 2V at Digit 1. The same RAX line also drives the base of a transistor through a 3V zener diode. At power-on, the voltage at the RAX input is insufficient to turn on the transistor. However, after using power-on-initialization software, you configure RAX as a digital output. Now, the low-impedance high/low voltage at RAX can override the bias voltage at the jumper to turn the transistor on and off.

Figure 1



An unused PIC μC input acts as an input for power-on status reading and as an output to drive an external transistor on and off.

You can easily extend this method to multiple jumpers and BCD switches. The only condition is that the voltage that the jumpers determine should be less than 3V and that impedance should be high. (DI #2475)

To Vote For This Design, Circle No. 302

Keyboard data-acquisition system is cheap and simple

Tom Lyons Fisher, Juniata College, Huntingdon, PA

The most important criteria of a data-acquisition system for college science laboratories are simplicity and price rather than precision or speed. The data-acquisition system in **Figure 1** offers adequate precision of less than 0.5% and speed of 1 Hz to replace the outdated laboratory chart recorder in student laboratories. You can install the system in 5 sec, operate it with a single toggle switch, and construct it for approximately $40. The only additional equipment necessary is a computer running Excel and an ATbus (not Universal serial bus), keyboard.

The system is simple to operate. After you install it between the keyboard and the PC, the keyboard functions normally until you close the toggle switch, which puts the circuit into "acquire" mode. The system then bypasses the keyboard and "types" data into an Excel spreadsheet column at the rate of 1 point/sec. When you switch off the system, the circuit finishes sending the current data before returning control to the keyboard. The slow sampling rate gives Excel time to replot an entire column of data and thus appear to be charting in real time.

The central IC in the circuit is the PIC12C671-04 μC, IC_2, which has an onboard 8-bit ADC. The circuit configures this μC to receive an analog voltage through the A/D pin. Because laboratory instruments output 1V full scale and the ADC's internal reference is set for 5V full scale, a rail-to-rail single-supply op amp, IC_1, provides a gain of 5. The op amp's feedback circuit also acts as a low-pass filter. The system has acceptable offset of −1.2 bits and displays excellent linearity; the coefficient of determination, R^2, equals 0.99998.

Because no pins are available for external clocking, the circuit allows the PIC μC to run at approximately 4 MHz using its internal RC oscillator. However, this oscillator is not a sufficiently accu-

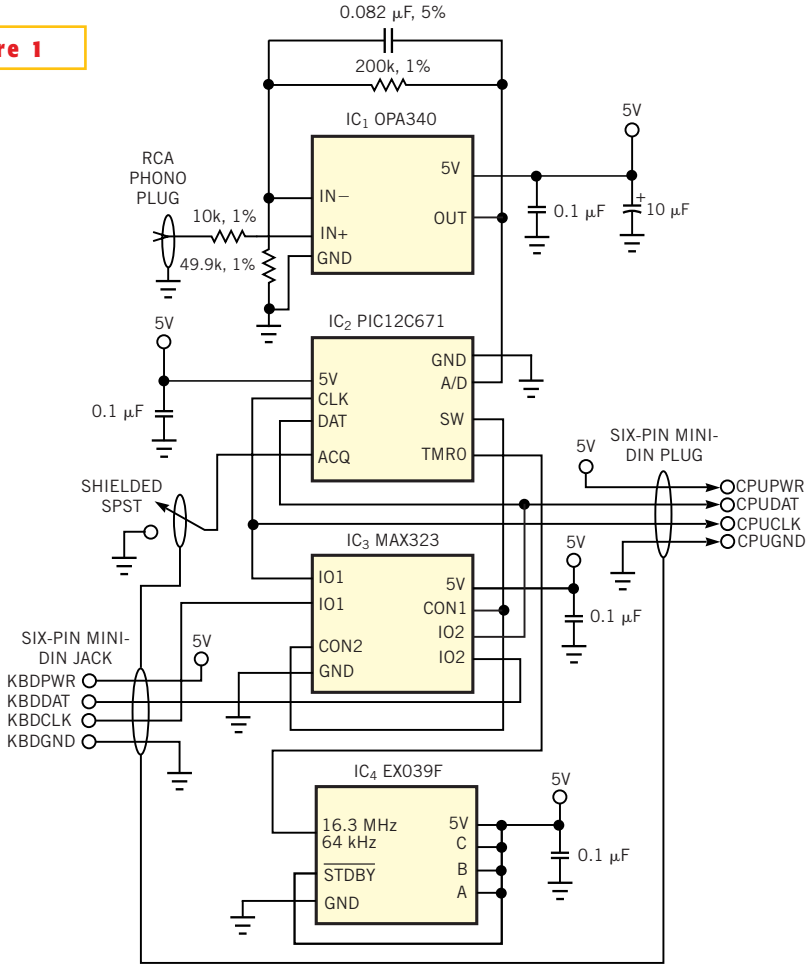
rate timebase for even 8-bit precision, so an external 16.384-MHz oscillator-divider, IC_4, produces a 64-kHz waveform that feeds into the Tmr0 pin of the PIC. The combination of a divide-by-256 prescaler and the appropriate period loaded into Timer 0 provide an accurate 1-sec interrupt.

The PIC µC "types" to the computer by outputting signals that emulate the keyboard via the Clk and Dat lines of IC_2. These pins duplicate the wired-OR electrical characteristics of the keyboard interface. When the data-acquisition system is active, the keyboard must not connect to the computer. The circuit fulfills this requirement using analog switches inside IC_3 in the keyboard clock and data lines. The µC controls these switches using the Sw signal.

The keyboard line powers the entire circuit, and the circuit shields the handle of the spst switch as a protection from static electricity. You can download the source program for the PIC from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2478. You can then compile under MPLAB 4.12 or use the .hex file. (DI #2478)

To Vote For This Design, Circle No. 303

Figure 1



When the switch is in the acquire position, this data-acquisition-system circuit bypasses the keyboard and puts data into an Excel spreadsheet column at the rate of 1 point/sec.

AC-couple analog signals without a dc bias source

Joseph Luis Sousa, Linear Technology Corp, North Chelmsford, MA

A conventional ac-coupling circuit in a single-supply system comprises a series capacitor and a shunt resistor to ground. Unfortunately, the negative peaks of the input signal can exceed the −0.3V input operating-range limits of an ADC, such as the LTC1402 serial 12-bit sampling ADC. To avoid going below ground, the circuit must return the shunt input resistor to a midsupply voltage source. This problem is classic with all single-supply ADCs. You can use fully differential analog inputs, such as those of the LTC1402 2.2M-sample/sec, 12-bit, serial ADC, to ac-couple an analog signal without this midsupply bias voltage.

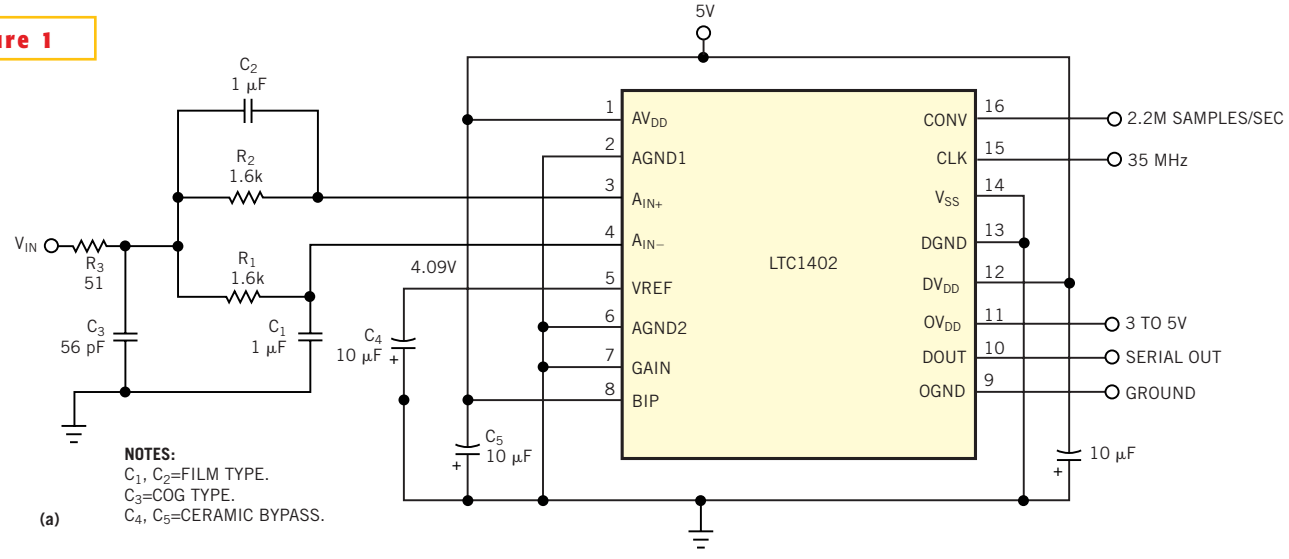
The ADC inputs derive the common-mode dc operating voltage directly from the input signal. The circuit has two requirements: The analog input signal must remain between 0V and the 5V supply voltage, and the ac transients must remain below the ±2V bipolar input range of the ADC. In **Figure 1a**, R_1 and the

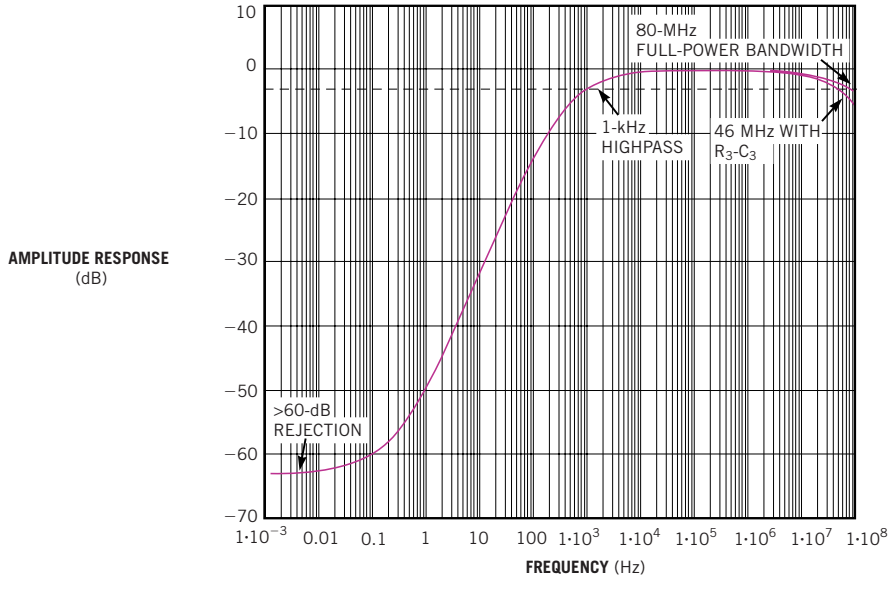
grounded C_1 at the A_{IN-} input of the ADC cancel the low-frequency signals and provide the basic ac-coupling function. R_2 and its shunt capacitor, C_2, at the ADC's A_{IN+} input cancel the sampling current bias offset. The optional C_3-R_3 46-MHz lowpass network isolates the ADC input from sampling-glitch-sensitive circuitry.

The frequency response for the values in the circuit has a low-cutoff pole at 1 kHz and low-frequency stopband rejection in excess of −60 dB, as set by the common-mode-rejection specification of the ADC, independent of RC-component-match accuracy (**Figure 1b**). The LTC1402 accepts wide bandwidth, full-scale, 4V p-p signals as great as 80 MHz. This ac-coupling circuit adds no distortion to the input signal. You can couple a 1.1-MHz Nyquist frequency sine wave into the ADC while keeping the THD below −82 dB.(DI #2479)

To Vote For This Design,
Circle No. 304

Figure 1





You can use fully differential analog inputs, such as those of the LTC1402 ADC, to ac-couple an analog signal without this midsupply bias voltage (a). The circuit's frequency response includes a low-cutoff pole at 1 kHz and low-frequency rejection of 260 dB (b).

Simple active-matrix-LCD bias supply operates from battery input

Michael Shrivathson, National Semiconductor Corp, Santa Clara, CA

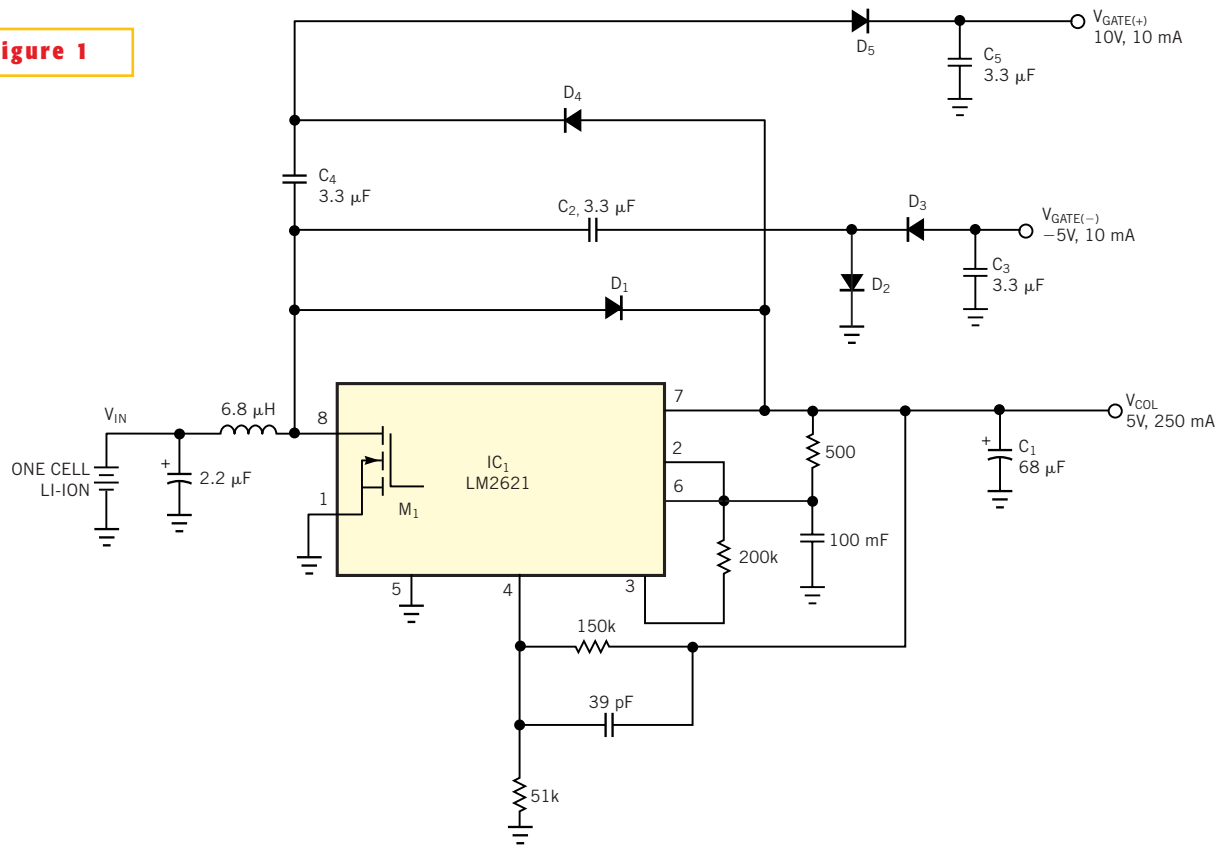
Many active-matrix-LCD applications need multiple voltages for thin film-transistor (TFT) bias. Typically, three voltages are necessary: 5V for the column driver; a positive voltage, such as 10V; and a negative voltage, such −5V, for the TFT gate drive, or row driver. For handheld electronic devices, a battery must produce these voltages. The most popular batteries in these devices are two-cell NiCd alkaline or one-cell lithium-ion batteries.

Figure 1 shows a simple, cost-effective way of providing these bias voltages. A step-up regulator, IC_1, forms the heart of the circuit. This regulator switches at a constant frequency of 1 MHz and a fixed duty cycle of 70%. IC_1 steps up the input voltage to 5V by storing the energy in the inductor when the internal MOSFET, M_1, is on and transferring this energy to C_1 when M_1 is off. IC_1's hysteretic gated-oscillator control scheme achieves the regulation.

C_2, C_3, D_2, and D_3 form a charge-pump inverter to provide an output of approximately −5V. When M_1 is off, C_2 connects in parallel with C_1 through D_1 and D_2. Thus, C_2 charges to V_{COL}, or 5V. When M_1 turns on, C_2 connects in parallel with C_3 through M_1 and D_3. Because of the polarity of this connection, C_3 charges to approximately $-V_{COL}$, or −5V.

C_4, C_5, D_4, and D_5 form a charge-pump doubler that provides an output of 10V. When M_1 is on, C_4 connects in parallel with C_1 through D_4 and M_1. Thus, C_4 charges to V_{COL} (5V). When M_1 turns off, C_1 and C_4 connect in series through D_1 and D_5, and this series pair connects in parallel with C_5. Thus, C_5 charges to approximately two times V_{COL}, or 10V.

Figure 1



A step-up regulator, IC_1; a charge-pump inverter comprising C_2, C_3, D_2, and D_3; and a charge-pump doubler comprising C_4, C_5, D_4, and D_5 produce the three voltages necessary for active-matrix-LCD applications.

This circuit provides 250 mA at the 5V output, V_{COL}, with 3% accuracy. The ac ripple is less than 100 mV. The circuit regulates the 10V output, $V_{GATE(+)}$, with 5% accuracy, and this output can provide 10 mA. The ac ripple at the 10V output is approximately 30 mV. The circuit regulates the −5V output, $V_{GATE(-)}$, with 6% accuracy and provides as much as 10 mA of output current. The ac ripple voltage at this output is 40 mV. A minimum load of 25 mA at the V_{COL} output ensures sufficient charge-pump action and thus maintains $V_{GATE(+)}$ and $V_{GATE(-)}$ at their nominal values. The efficiency of this circuit varies from 75 to 82% when operating from a one-cell lithium-ion battery. (DI #2477)

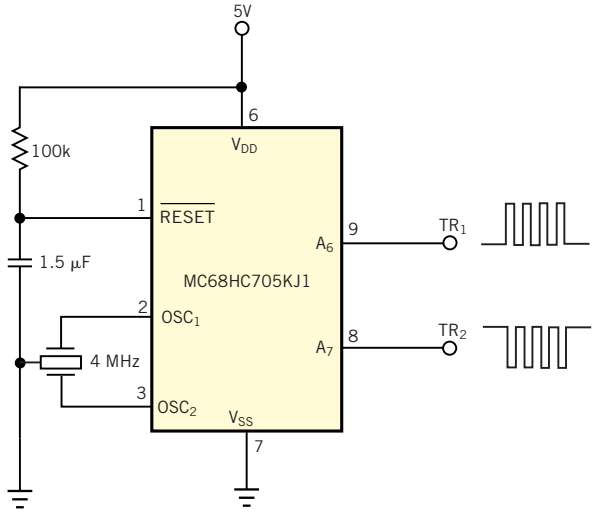
To Vote For This Design,
Circle No. 305

µC generates a frequency burst

Abel Raynus, Armatron International Inc, Melrose, MA

Pulse-sonar applications require generating bursts of a given frequency, duration, and repetition rate. Traditionally, the burst generator comprises a crystal oscillator with pulse modulation. But the easiest and cheapest way to generate the bursts is by using an inexpensive 8-bit µC, such as the 68HC705KJ1 and 68HC705J1A (Motorola) and do the whole job using software. You can get additional benefits by outputting two signals in opposite phase to feed the ultrasonic transducer directly or via a push-pull buffer (**Figure 1**). Note that only two µC pins are necessary for burst generation. You can use the rest of the pins for different purposes.

Figure 1



The easiest way to generate bursts for pulse sonar applications is to use a single µC and do the whole job in software.

The highest frequency that the µC can generate depends on the value of the highest oscillator frequency, f_{OSC}, that the manufacturer specifies and the structure of the instruction set, namely the quantity of machine cycles the µC takes to execute an instruction. With f_{OSC}=4.00 MHz, the mentioned µCs can generate a maximum frequency of 58.8 kHz. This value is a good match for sonar projects because most of the ultrasonic transducers, working in an air medium, have a standard resonant frequency of 40 kHz. To lower the frequency from 58.8 to 40.0 kHz requires a simple delay of 4 µsec using nop and brn instructions.

The constant value in the counter "Number" determines the burst duration. With one 8-bit counter, the burst duration can range from 0.1 to 3.2 msec. If a longer burst is necessary, you can add one or two more counters. If you choose a duration of 1 msec, as in this case, the value to put into the counter is

$$\text{NUMBER} = \frac{\text{BURST DURATION}}{\text{HALF OF PERIOD}} = \frac{1000\ \mu\text{SEC}}{12.5\ \mu\text{SEC}} = 80.$$

How you program the burst, repetition rate depends on the timer structure of the µC. For µCs with 16-bit programmable timers, the best way is to use either timer-overflow or output-compare functions. For µCs with multifunction timers, only the first eight timer stages are usable. Thus, timer overflow occurs every 0.51 msec, which is too short for a repetition period. So, you can use either real-time interrupt or, as in this case, organize a pacemaker based on the timer-overflow-interrupt. This design generates a burst every time the counter T rolls over from $FF to $00 with a repetition period of 131 msec. You can download the accompanying programs from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the files for Design Idea #2480. (DI #2480)

To Vote For This Design,
Circle No. 306

design ideas

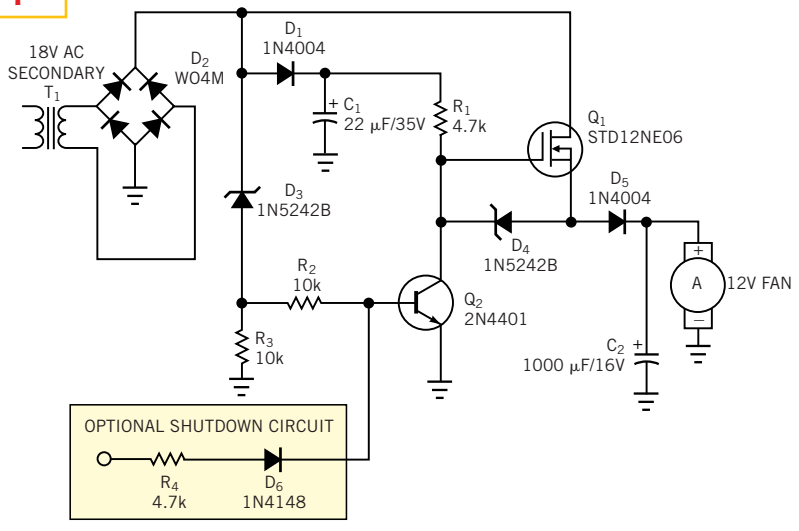
Edited by Bill Travis and Anne Watson Swager

MOSFET switch provides efficient ac/dc conversion

Spehro Pephany, Trexon Inc, Toronto, ON, Canada

Occasionally, you have access to a transformer for powering a dc circuit, but its output voltage is much higher than that required for the dc voltage. The full-wave-rectified and filtered output of an ac input voltage V_X, is $V_{DC} = 1.414V_X - 2V_F$, where V_F is the forward drop in the rectifier (approximately 0.7V). For example, if you require 12V dc to power a small cooling fan drawing 100 mA and the ac voltage is 18V, a full-wave rectifier and filter results in a 24V-dc output. Although you can regulate the voltage down to 12V dc by using a simple three-terminal regulator (such a μA7812), the result is wasted power of approximately 1.3W. This waste means that you must provide for heat removal, somewhat defeating the purpose of including the cooling fan. If you use a typical 100 × 100-mm, 12V-dc fan rated at 0.45A, the typical heat loss is approximately 2.5W, increasing to 5W at full load. In many applications, this level of loss is unacceptable, so you'd have to use an extra transformer secondary, a dc/dc converter, or a switching regulator. The circuit in **Figure 1** uses a MOSFET switch and diode to effectively draw current from the transformer when the voltage is close to the desired level of 12V dc.

Figure 1



Using a MOSFET circuit, you can efficiently convert the too-high voltage of a leftover transformer to a lower dc level.

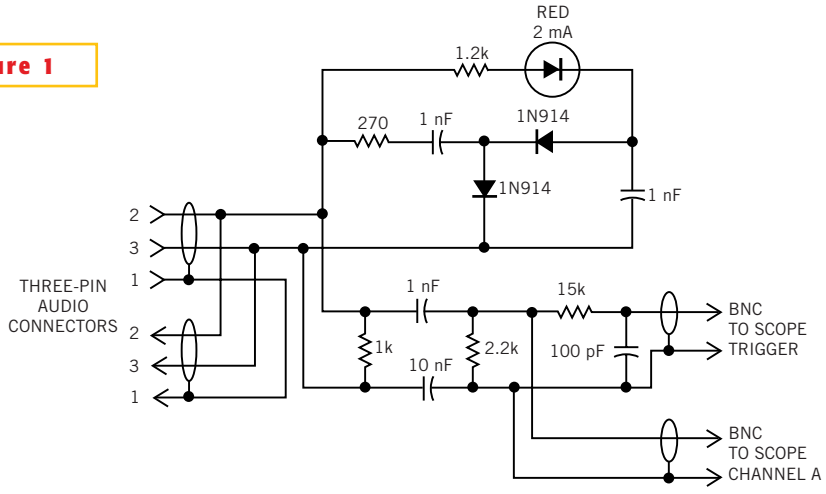
The full-wave bridge, D_2, rectifies the 18V-ac signal. The diode, D_1, and C_1 provide a gate bias voltage of approximately 24V dc. This voltage drives the gate of Q_1 through R_1, shunted by D_4, which maintains the gate voltage at a maximum of 12V relative to the source, even during transient conditions. As the bridge-rectifier output increases from 0V to the peak of approximately 24V each half-cycle, the bias voltage holds the MOSFET on until the input voltage reaches the breakdown voltage of D_3 (12V) plus the $V_{BE(ON)}$ of Q_2, or approximately 12.7V. At that point, Q_2 turns on, turning Q_1 off. The output filter capacitor, C_2, charges through D_5. As the rectifier output voltage decreases from 24 to 0V, Q_2 again turns off at approximately 12.7V, allowing Q_1 to turn on and provide another pulse of current to charge C_2. C_2 provides power for the load between the pulses, which occur at 240 Hz with a 60-Hz input. Thus, power drain from the transformer occurs in short pulses, much in the manner of a typical bridge-rectifier/output-filter arrangement but at double the frequency. If you want to turn the fan off with a logic signal, you can add R_4 and D_6. When you apply a logic-high signal to the input, Q_2 conducts, turning the MOSFET off. (DI #2484)

To Vote For This Design, Circle No. 311

Passive circuit monitors AES data

Wayne Sward, Bountiful, UT

Figure 1



A passive circuit gives a good indication of data activity on digital-audio lines.

The circuit in **Figure 1** efficiently monitors common digital-audio signals. One format for such signals is the Audio Engineering Society (AES) 44.1- or 48-kHz standard. Typically, the data consists of a serial data stream with a data rate of approximately 1 Mbps. A lower frequency pulse interspersed in the data stream synchronizes data frames every 16 to 20 data bits. The amplitude of the data and sync pulses is 3 to 12V p-p, with one cycle of an ac wave representing each bit. The signals are on a two-wire cable that you can isolate from ground by using signal transformers or capacitors. Several other data-transmission standards use a similar data format. An oscilloscope does not reliably trigger on such a waveform; thus, troubleshooting and signal tracing such systems is difficult. The circuit in **Figure 1** is passive and small and uses no power supplies. You can keep it in a toolbox, ready for instant use.

The two diodes and associated capacitors form a voltage doubler to provide a bias voltage of approximately −2 to −10V. The 2-mA LED connects between one of the signal lines and this bias voltage. Whenever the signal-line voltage exceeds approximately 1.5V, the LED turns on. At a data rate of nearly 1 MHz, the LED appears to continuously glow when good data is present. The LED's intensity is proportional to the peak-to-peak voltage level, so you can easily observe low or high voltage levels. Intermittent levels, crosstalk, or interference on the signal causes the LED to flicker. The LED circuit is differential and measures the voltage levels between the two signal lines. Common-mode ground noise or hum do not affect the LED's display. The voltage doubler is an efficient way to increase the sensitivity of the LED without an additional power supply.

The two coupling capacitors sample the high-frequency data waveform but reject any low-frequency common-mode noise or hum. You can display the data waveform on an oscilloscope to more closely inspect the wave shape. The 15-kΩ resistor and 100-pF capacitor form a simple but effective filter to detect the sync bit in the data stream. You feed this sync bit to the external sync input of the oscilloscope, resulting in a stable display of the data frame. The coupling capacitors avoid creating a ground loop between the signal lines and the grounded, shielded input of the oscilloscope. You can readily monitor data amplitude, waveshape, and activity of individual bits with the oscilloscope. (DI #2482)

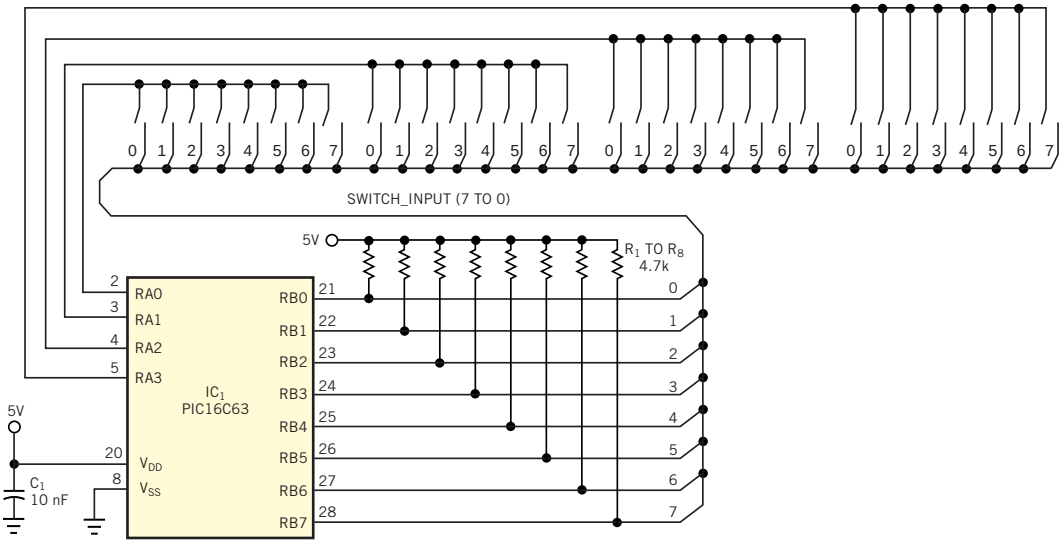
To Vote For This Design,
Circle No. 312

µC multiplexes DIP switches to I/O port

Gregory Willson, ACS Defense Inc, Warrenton, VA

At times, a µC must read a large number of DIP switches, such as for system identification, bus-address setup, manual configuration, or other purposes. However, the available number of I/O lines is sometimes not enough to assign a switch to each one. You can use multiplexer ICs to share one I/O port with multiple switches, but they complicate the circuit, dissipate additional power, and consume precious board real estate. **Figure 1** shows a method of multiplexing 32 DIP switches using only 12 I/O pins and eight pullup resistors. Four 8-bit DIP switches connect in parallel to a single 8-bit I/O port. A pullup resistor on each port pin defaults the input to a high state; a switch closure pulls the input to

Figure 1



Using only 12 I/O pins, a µC can read 32 DIP switches.

a low state. The key to multiplexing the DIP switches is to ground each set of eight switches in turn using output pins from a second I/O port.

To deselect a set of switches, the controlling-port pin acts as an input, rendering it a high-impedance port. In this way, 12 I/O pins can read 32 switches, and 16 I/O pins can read 64 switches. Select the values of the pullup resistors to limit the total current into the controlling-port pin to less than the maximum sink current. Some µCs, such as the Microchip PIC16C6x family, provide the ability to enable weak internal pullups on I/O port pins. By using this feature, you can eliminate the eight external pullup resistors. The code fragment in **Listing 1** illustrates reading the four 8-bit DIP switches and storing the results, using a Microchip PIC16C63 µC. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Design Center to download the file for Design Idea #2483. (DI #2483)

To Vote For This Design,
Circle No. 313

LISTING 1—MULTIPLEXING DIP SWITCHES TO SINGLE I/O PORT

```
;Microcontroller Multiplexes Multiple DIP Switches to Single I/O Port
;Author: Gregg Willson, ACS Defense, Inc.

;Define constants
      list  p=16c63
#include    "c:\mpasm\P16C63.inc"

;Define RAM storage locations for DIP switch values
TMP0  equ   h'20'
TMP1  equ   h'21'
TMP2  equ   h'22'
TMP3  equ   h'23'

;Main Routine
Read_32_DIP_Switches
      call  Initialize_Ports  ;Setup ports for multiplexing
      movlw b'11111110' ;Get ready to make port A, bit 0 an output
      call  Read_DIP_Sw ;Read the first DIP switch
      movwf TMP0        ;Save in temporary register
      movlw b'11111101' ;Get ready to make port A, bit 1 an output
      call  Read_DIP_Sw ;Read the second DIP switch
      movwf TMP1        ;Save it
      movlw b'11111011' ;Get ready to make port A, bit 2 an output
      call  Read_DIP_Sw ;Read the third DIP switch
      movwf TMP2        ;Save it
      movlw b'11110111' ;Get ready to make port A, bit 3 an output
      call  Read_DIP_Sw ;Read the fourth DIP switch
      movwf TMP3        ;Save it
      return
; DONE

;Subroutines
Initialize_Ports
      bcf   STATUS, RP0 ;Select register Bank 0
      movlw b'00000000' ;Default port A outputs to 0 to ground switches
      movwf PORTA
      bsf   STATUS, RP0 ;Select register Bank 1
      movlw b'11111111' ;Make port B all inputs
      movwf TRISB
      movwf TRISA       ;Make port A all inputs initially
      bcf   OPTION_REG, 7     ;Option: Enable weak internal pullups on port B
      return

Read_DIP_Sw
;Enable DIP Switch based on W register value
      bsf   STATUS, RP0 ;Select register Bank 1
      movwf TRISA       ;DIP switch is now selected
;Read the data
      bcf   STATUS, RP0 ;Select register Bank 0
      movf  PORTB, W    ;Put switch values in W register
      return

      END
```

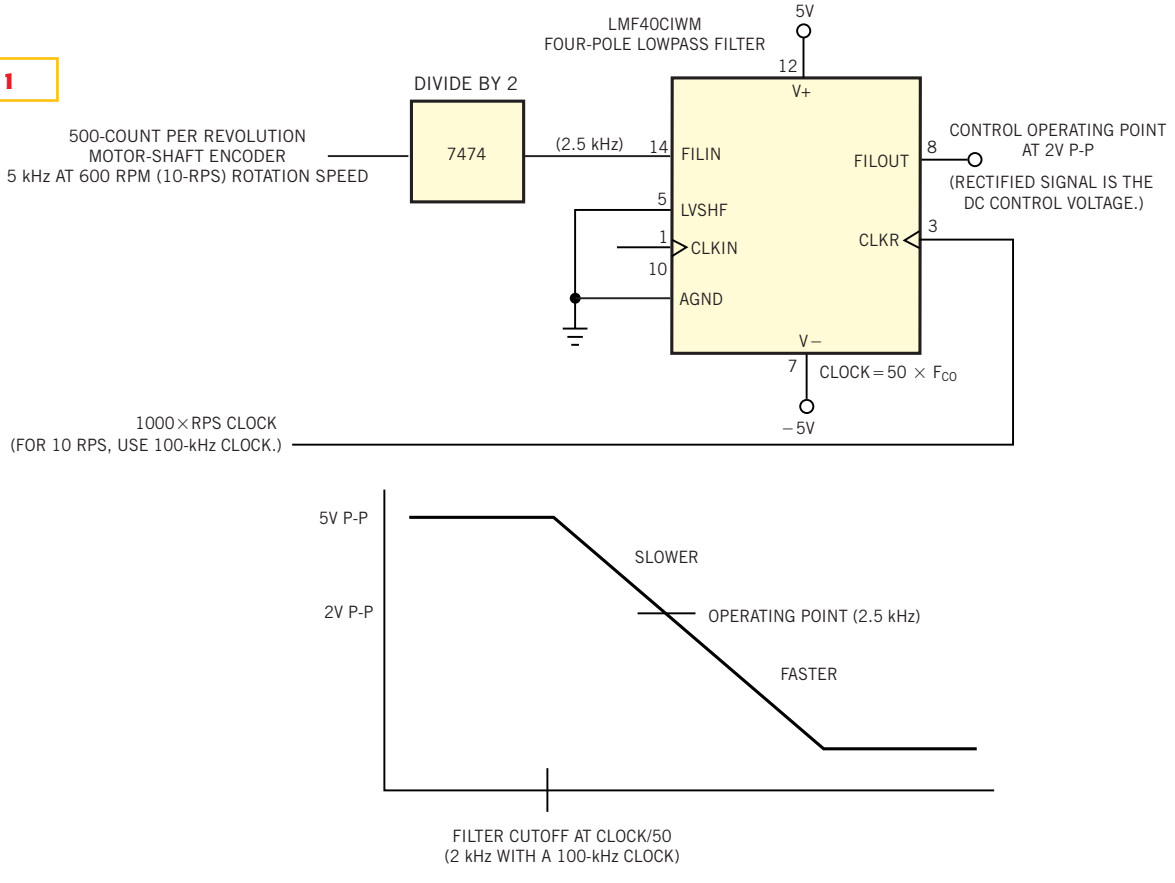
Switched-capacitor IC controls feedback loop

Dave Sargent, IBM Research, San Jose, CA

You can implement a simple control loop with a constant setpoint over a wide range of control by using a switched-capacitor filter. The circuit controls motor speed over 1 to 200 Hz or 60 to 12,000 rpm. In **Figure 1**, a National LMF40CIWM four-pole lowpass filter is the heart of the design. This filter has a cutoff frequency defined by the clock divided by 50. Consider this filter as a difference amplifier that compares the difference between two frequencies. The relationship between the clock divided by 50 and the 500-count encoder divided by 2 is such that when the clock rate is 1000 times the revolutions per second of the motor, the signal frequency from the encoder is at approximately the midpoint of the filter response. This midpoint is the point of zero error and is approximately 2V p-p. The rectified output serves as the dc setpoint voltage for the control loop. If the motor speed increases, the voltage decreases, and if the motor slows down, the voltage increases. As simple as the method is, it can drive a motor-control chip and provide good speed regulation.

You can use the method to control other servo loops that provide a feedback frequency within the range of the filter. The big advantage of this scheme is that the setpoint remains constant with a range of speed settings. Another advantage is that the clock provides direct speed calibration thanks to the 1000-to-1 relationship between the clock and the rotational speed of the motor. **Figure 2** shows some circuit details for enhanced operation. To cover the higher speed range, you need an additional divide by 10 to stay within the frequency range of the filter. An example is given for 6- and 60-Hz rotational speeds. Conventional op-amp circuits buffer and rectify the output of the filter. The application uses one of many full-wave op-amp-rectifier circuits that follow a buffer with a gain of 2. After rectification, the 75-kΩ resistors and 0.1-µF capacitor provide some filtering and time-constant conditioning. You set the gain of op amp 1 so its output dc voltage is 2.5V at the operating point. Op amp 2 offsets this voltage and moves the operating point to 0V when no speed error is present. The offset-adjust trimmer allows for minor variations and calibrates the actual rotational speed to the clock signal. Op

Figure 1

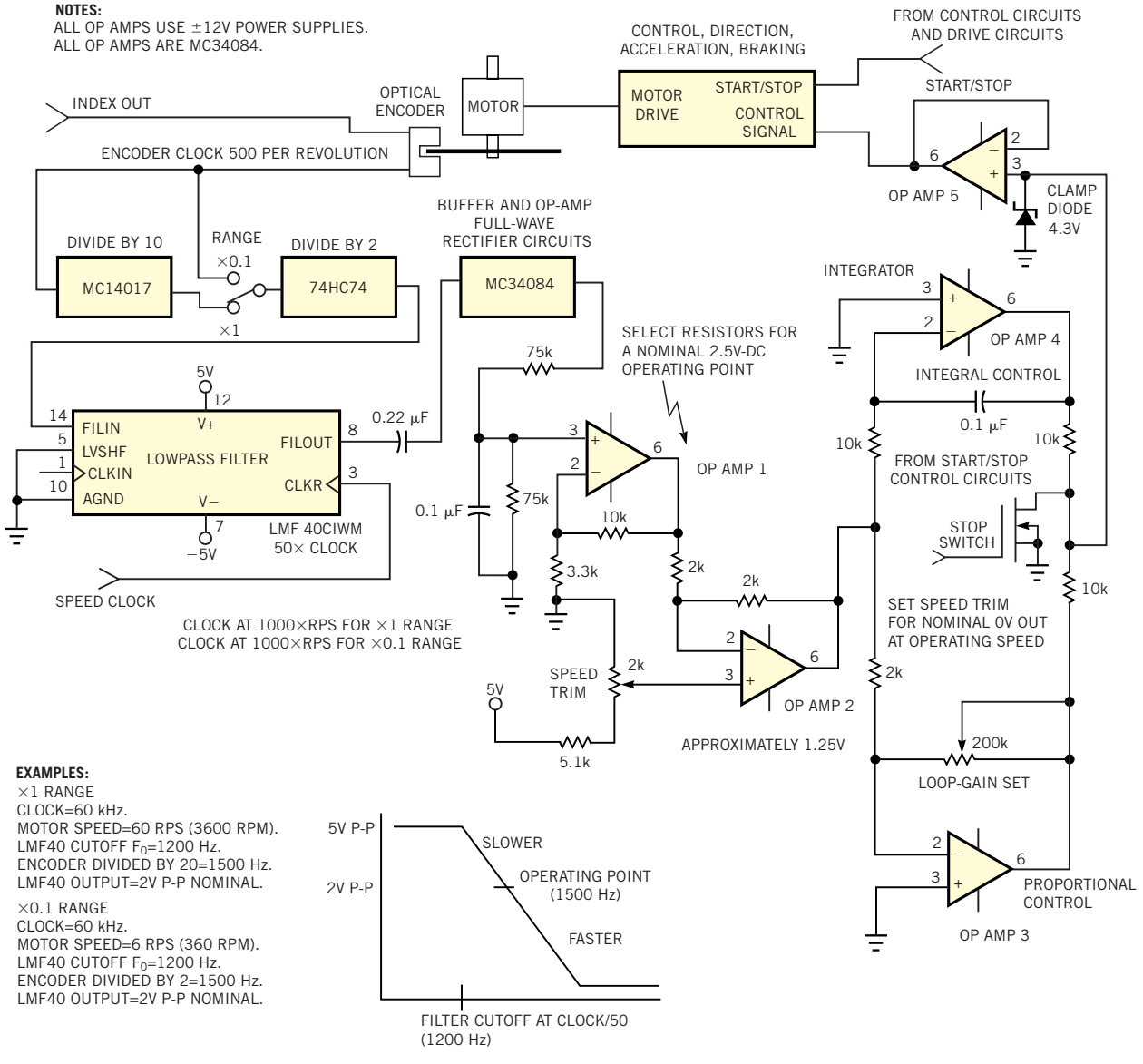


A lowpass filter is the heart of a wide-range control loop.

amp 3 provides gain for the proportional signal.

You can use a fixed resistor in place of the 200-kΩ trimmer. Op amp 4 is an integrator circuit that provides the classical integral control for the loop. The integrator makes up for errors in the following control circuits and motor characteristics throughout the control range. The integrator control-point output voltage therefore changes at different speed settings. Buffer op amp 5 sums the proportional and integral signals at its input. A clamp diode limits the positive drive voltage and prevents any negative excursions from driving the loop to a latch-up condition. In an application, the clamping limits the output to 4.3A, because the motor-control circuit has a drive characteristic of 1A per volt. When the motor stops, the FET stop switch clamps the control signal to zero. Additional circuits control acceleration rate, braking rate, and direction. The speed accuracy for the system is a nominal 0.002% throughout the range. The speed clock comes from a DDS chip, and all the above functions are under control of a PC or front-panel switch settings. (DI #2486)

Figure 2



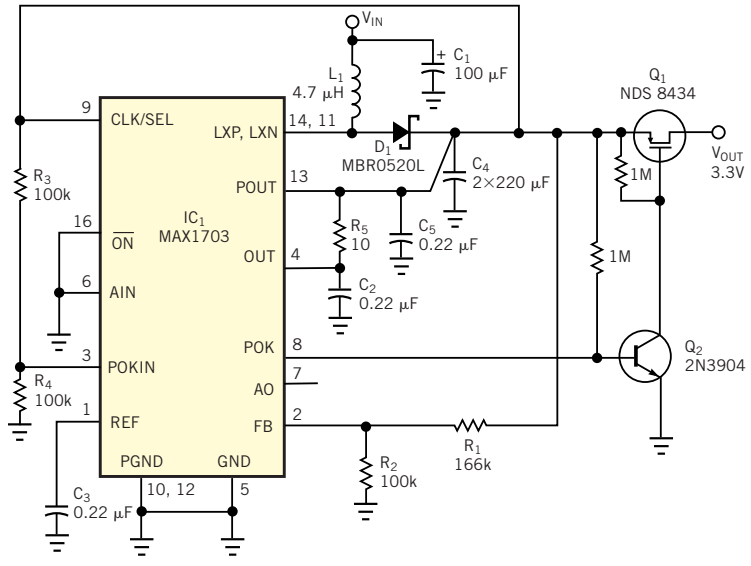
A wide-range control circuit uses two ranges for maximum resolution.

Simple circuit disconnects load

Larry Suppan, Maxim Integrated Products, Sunnyvale, CA

Placing a load-disconnect circuit on the output of a bootstrapped step-up regulator allows the regulator to start with load currents much higher than would otherwise be possible (**Figure 1**). During shutdown, the disconnect completely isolates the battery from the load. The circuit boosts a single NiMH-cell output to 3.3V and delivers output currents to 600 mA. Step-up regulators are excellent for portable applications because they exhibit high efficiency, low supply current (120 µA operating, 20 µA in shutdown), and ample current once started. Many, however, cannot start with maximum load from low supply voltages, such as those from single-cell batteries. This problem arises because most low-voltage CMOS boost regulators derive power from their own outputs, which equal V_{IN} minus a diode drop at start-up. Low values of input voltage don't allow the switching transistor to become fully enhanced at start-up, so the transistor presents a high impedance that limits the peak inductor current. As a result, the circuit cannot produce enough current to simultaneously supply the load and charge the output capacitor.

Figure 1

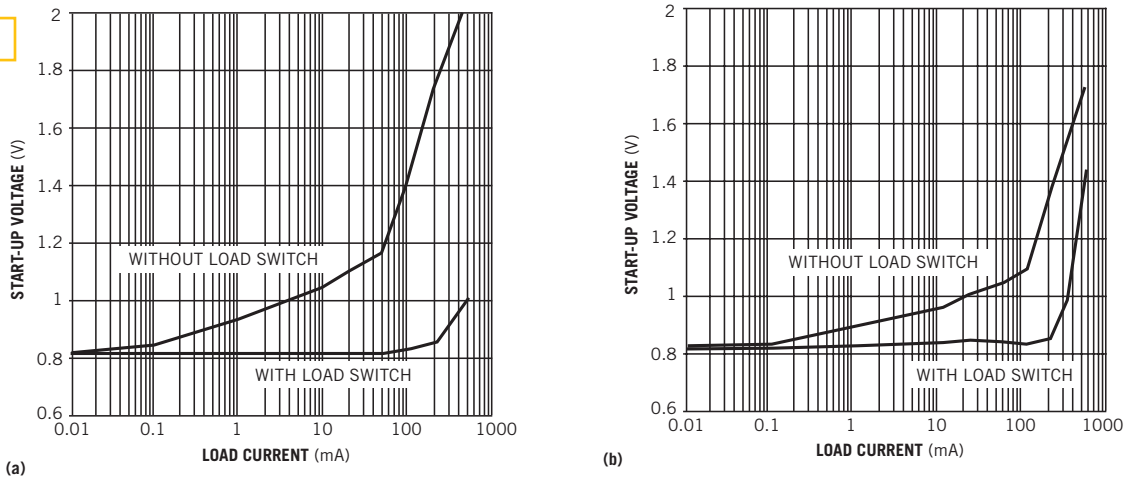


NOTE:
HEAVY LINES INDICATE HIGH-CURRENT PATHS.

The addition of a couple of transistors enables a switching regulator to start with full load and low input voltages.

To get around this problem and ensure reliable start-ups, most regulator ICs incorporate an undervoltage lockout (UVLO). IC_1, for example, is a synchro-

Figure 2



The load-disconnect switch in Figure 1 allows the regulator to start up with heavy loads and low input voltages (a). A slight modification of the circuit in Figure 1 provides 5V-output operation (b).

nous boost converter whose bootstrapped operation cannot start until its output voltage exceeds the internal UVLO threshold of 2.3V. You can overcome this start-up limitation with an external power MOSFET, Q_1, operating as a load-disconnect switch, and by using the power-OK (POK) comparator built into many low-voltage switching regulators. R_3 and R_4 set the POK threshold at 2.5V, allowing V_{IN} to rise above the UVLO threshold. Q_2 inverts the POK output before driving Q_1. Q_1 disconnects the load, allowing V_{OUT} to rise to a level (above UVLO) that ensures full enhancement of Q_1 when it turns on. As a result, the circuit can start under full load with input voltages as low as 0.8V (**Figure 2a**). Because the circuit takes the regulator feedback before this switch, the MOSFET you choose for a given application depends on the load current and minimum acceptable level of load regulation. The MOSFET shown is a low-threshold device. Connecting the FB terminal (Pin 2) to ground and removing R_1 and R_2 produces a 5V regulated output, whose performance is similar to that of the 3.3V version (**Figure 2b**). (DI #2487)

To Vote For This Design, Circle No. 315

Follow the debouncing flip-flops

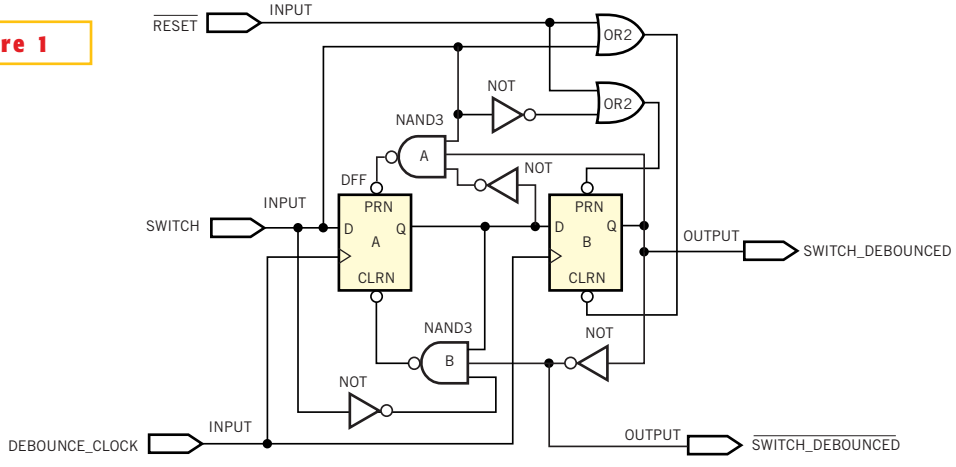
Ray Scott and John Stanley, Airport Systems International, Overland Park, KS

During a recent development effort, we could not find literature detailing how to debounce an spst momentary switch using only logic (no capacitors, Schmitt triggers, or other components). Our application placed the spst switches several feet from the logic board, and both noisy switches and line transients caused false triggers. Many methods simulate a debounce by checking the state of the switch on clock edges and summing the checks over time, but our application required no transitions during the qualification time before acknowledgment of a key press. Thus, the switches can work effectively in noisy environments over reasonably long distances. **Figure 1** illustrates a means of debouncing a momentary switch for both the make and the break operations. Designers often use programmable logic to debounce momentary switches used in keypads, in keyboards, or as configuration inputs. Flip-flops are usually precious commodities in programmable logic, whereas logic gates are available in greater abundance. The design in **Figure 1** minimizes the use of flip-flops.

The circuitry monitors the state of the Switch input. Once the circuit detects a transition, a "qualifying" time of two Debounce_Clock periods begins. If at any time during the qualifying time the Switch input returns to its original state, indicating switch bounce or an electrical transient, the circuitry returns to its starting state and begins looking for another transition. The Switch input must be completely stable for two positive transitions of the Debounce_Clock input before the Switch_Debounced output will change. A frequency of approximately 15 Hz (or a period of 66 msec) for the Debounce_Clock input works well, even for low-cost, "noisy" switches. You can delete the reset logic if you are unconcerned with the power-on state of the Switch_Debounced output. Following power-on, the output will be correct after two clock periods. (DI #2481)

To Vote For This Design, Circle No. 316

Figure 1



A debouncing circuit using programmable logic makes frugal use of flip-flops.

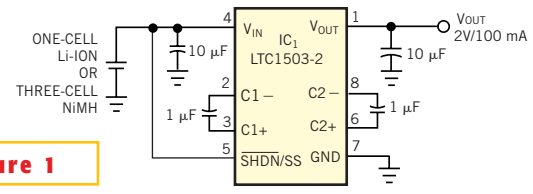
Inductorless converter provides high efficiency

Sam Nork, Linear Technology Corp, Milpitas, CA

Two common methods exist for generating a regulated dc output voltage that is lower than the input voltage. The first approach is to use a low-dropout (LDO) regulator. LDO regulators are small, easy to use, and inexpensive, but all the output current must also flow through the input; hence, they exhibit low efficiency. The second approach is to use an inductor-based switching regulator. Inductor-based switchers can be efficient, but they tend to be more complex, costly, and area-consuming than their LDO-regulator counterparts. A third option retains the simplicity and size of an LDO regulator but enjoys the high efficiency usually reserved for inductor-based circuits. The circuit in **Figure 1** uses switched-capacitor techniques to achieve high-efficiency step-down conversion without an inductor.

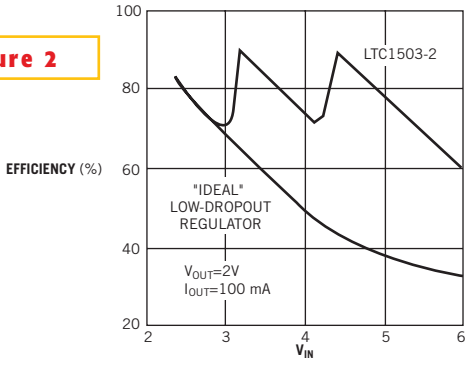
The circuit produces a regulated 2V output with as much as 100 mA of load-current capability. IC_1, an LTC1503-2, has an input range of 2.4 to 6V, allowing the IC to take power from either a single Li-ion cell or a three-cell NiMH battery. IC_1 uses fractional-conversion techniques to achieve efficiencies typically more than 25% higher than that of an LDO regulator (**Figure 2**). Internal control circuitry ensures that the device operates with the optimal step-down ratio as the input voltage and load conditions vary. You need only four small ceramic capacitors to make a complete step-down supply. Quiescent current of 25 µA typical and the small MSOP-8 package make the circuit ideal for handheld devices. (DI #2485)

Figure 1



Eschew bulky inductors, using switched-capacitor step-down conversion.

Figure 2



Switched-capacitor conversion yields higher efficiency than LDO regulators.

To Vote For This Design, Circle No. 317

design ideas

Edited by Bill Travis and Anne Watson Swager

Simple BER meter is easy to build

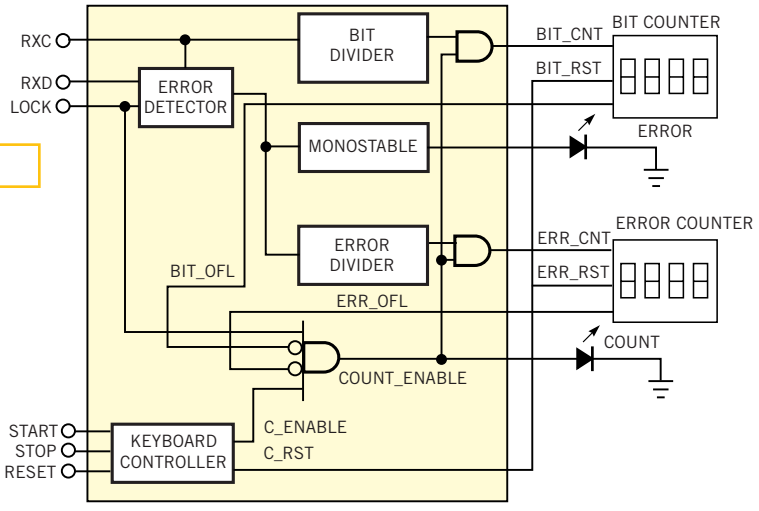
Luis Miguel Brugarolas, Sire Sistemas y Redes Telmáticas, Tres Cantos, Madrid, Spain

A BIT-ERROR-RATE (BER) tester is a basic tool for digital-communications measurements. Although many commercial BER testers are available, you can easily design and build an inexpensive version. The scheme in **Figure 1** has performance similar to that of a commercial tester but requires you to perform a manual calculation based on displayed data. The tester displays received bits and received erroneous bits, and you must calculate the BER data using a handheld calculator, for example.

You can build the tester in **Figure 1** from a piece of programmable logic, such as an FPGA or a CPLD, and two counter modules. You can buy the counter modules in kit form or in built form from numerous suppliers. The counters are available in LCD or LED-display formats with four or more digits. The counter modules must have overflow indicators, and they must allow pulse widths that are as narrow as half the data-clock period.

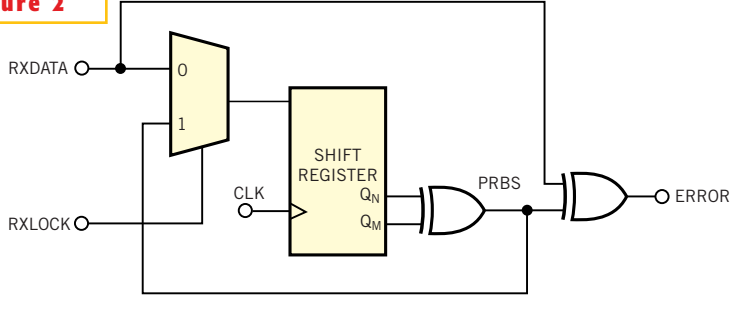
Figure 2 shows the core of the error detector. This detector uses the same pseudorandom-bit-sequence (PRBS) generator as the transmitter does but adds a trick. When the demodulator under test is not in lock, the shift register loads the receive data, and no error count takes place. In every demodulator, except for special burst-mode units, the BER decreases to some nominal rate before you declare the system "locked." Thus, the shift register is self-synchronized to the incoming sequence with great probability. When the demodulator is in lock, the Lock signal switches the multiplexer out-

Figure 1



With some programmable logic and two counter modules, you can design and build a simple bit-error-rate tester.

Figure 2



In the error-detector core, the Lock signal controls whether the shift register loads with received data or the locally generated pseudorandom bit sequence (PRBS).

put so that locally generated data, which should be the same data as the transmitted data, shifts into the register while the measurement is running. Any divergence between the received data and the locally generated data constitutes a bit error. As long as the counter counts pulses, the error signal must combine with the clock in an RZ format so that the tester does not count two consecutive errors as only one.

An error that occurs just before the demodulator is in lock causes incorrect initialization of the shift register, and the local and received sequences is highly uncorrelated. Thus, the BER in this case is close to 0.5. You can easily detect this erroneous condition and restart the measurement.

The architecture in **Figure 1** allows you to divide the number of errors and bits in the error-divider and bit-divider blocks. You can divide errors by 1, 10, 100 and 1000 and divide bits by 10^4, 10^5, 10^6, and 10^7. This division feature allows the tester to measure of a range of BERs from poor ones for which the tester must divide the error count to a low value to situations that require bit division or that entail long measurement periods. Two switches for each block can control the division rate in a simple way. To avoid mistakes, division control should also control a decimal point in the display, and an indication label under the displays should show the multiplication factor for the actual configuration.

Another feature is related to the overflow. When either counter unit overflows, the scheme in **Figure 1** immediately stops the error count using the BIT_OFL and ERR_OFL flags so that the tester does not display erroneous data when taking unattended measurements or when taking measurements for long periods. When active, the BIT_OFL and ERR_OFL flags turn off the COUNT_ENABLE signal.

The Start, Stop, and Reset keys control the unit. They drive a finite state machine, which produces the variables C_ENABLE and C_RESET. The first variable controls the bit and error count, and the second controls the counter reset. (DI #2488)

To Vote For This Design,
Circle No. 406

Software avoids interrupt overhead

Hans-Herbert Kirste, WAGO Kontakttechnik GmbH, Minden, Germany

You can service peripheral ICs connected to a µC by polling or via interrupts. The polling method can be time-consuming, so interrupt handling is often preferable because the µC has to take care of the peripheral only on request. However, each separate interrupt causes the µC to stop normal program execution, save its current state on the system stack, and vector to the interrupt's processing function. The first instructions in the interrupt function normally push some or all registers used onto the stack.

Peripherals, such as the 16550 UART, that have more than one interrupt source, may require that the µC process more than one request at a time. Fortunately, you can write software that allows the µC to process more than one request in one interrupt cycle. Thus, the interrupt

LISTING 1–STANDARD INTERRUPT-HANDLING SOFTWARE

```
static interrupt (0x1F) void UartIrq ( void ) /* EX7 = vector 0x1f, address 0x7c */
{
    switch ( (uiPortBase + IIR) & 6 ) /* read interrupt identification register */
    {
            case 0:
            /* do something to handle the ID 0 */
                    break;
            case 2:
            /* do something to handle the ID 2 */
                    break;
            case 4:
            /* do something to handle the ID 4 */
                    break;
            case 6:
            /* do something to handle the ID 6 */
                    break;
    }
}
```

LISTING 2–MORE EFFICIENT INTERRUPT-HANDLING SOFTWARE

```
static interrupt (0x1F) void UartIrq ( void ) /* EX7 = vector 0x1f, address 0x7c */
{
UCHAR ucIIR;
    while ( 0 == (ucIIR = (uiPortBase + IIR) & 1 ) ) /* D0=0 --> interrupt pending */
    {
            switch ( ucIIR & 6 )    /* identify the interrupt source */
            {
                    case 0:
                    /* do something to handle the ID 0 */
                            break;
                    case 2:
                    /* do something to handle the ID 2 */
                            break;
                    case 4:
                    /* do something to handle the ID 4 */
                            break;
                    case 6:
                    /* do something to handle the ID 6 */
                            break;
            }
    }
}
```

overhead occurs only once. The result is improved system performance.

Listing 1 is an example of standard interrupt-handling software. Standard practice involves vectoring, pushing, and popping registers for each interrupt request. A more sophisticated function in **Listing 2** tries to handle as many interrupt requests as the peripheral requires. This function processes multiple reads to the identification register, and the process repeats until the μC has serviced all sources. You can download both listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2489. (DI #2489)

Delay line eases Spice dead-time generation

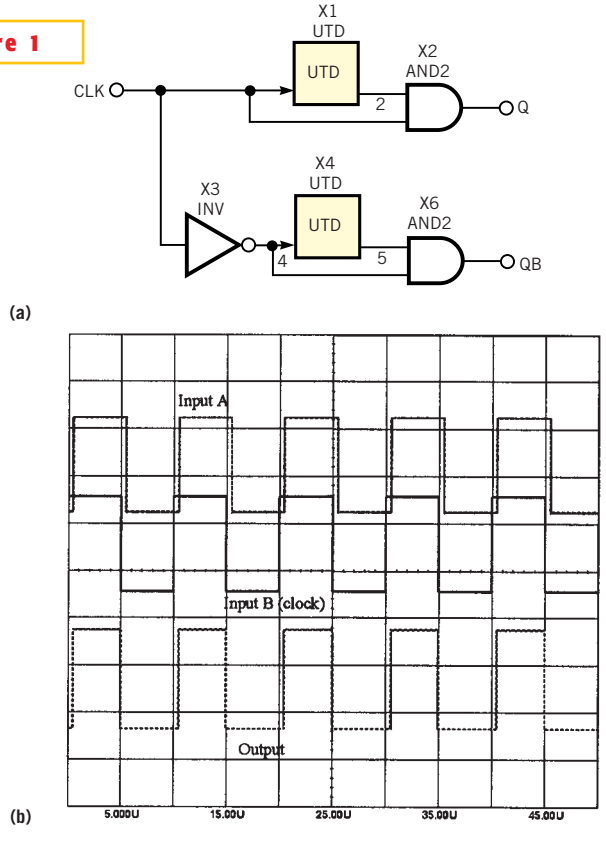
Christophe Basso, On Semiconductor, Toulouse, Cedex, France

Generating complementary clock signals in a Spice simulation is an easy task. However, this task gets much harder if you need to introduce some dead time into the signals. This difficulty is especially true when you're dealing with a variable-pulse-width-modulated switching cycle. In fact, you need to insert a dead-time interval between the switching of any two power devices in series, such as bridge or half-bridge designs that use MOSFETs and switch-mode power supplies and that implement synchronous rectification. The dead time prevents any cross-conduction, or shoot-through, between both switches and helps to reduce the associated losses.

The circuit in **Figure 1a** overcomes this typical Spice problem. The input clock drives two delay lines that feature the same specifications. When the clock goes high, one input to X2's AND gate is also high. However, because of the delay line, the other input stays low for the given dead time. When both inputs are high, the output is a logic one (**Figure 1b**).

When you generate models in a proprietary syntax, the translation process to another platform is usually painful. However, thanks to common Spice3 primitives, such as the delay line, T, the trans-

Figure 1



Two AND gates and two delay lines generate a dead-time element in Spice (a). When both inputs are high, the output is a logic one (b).

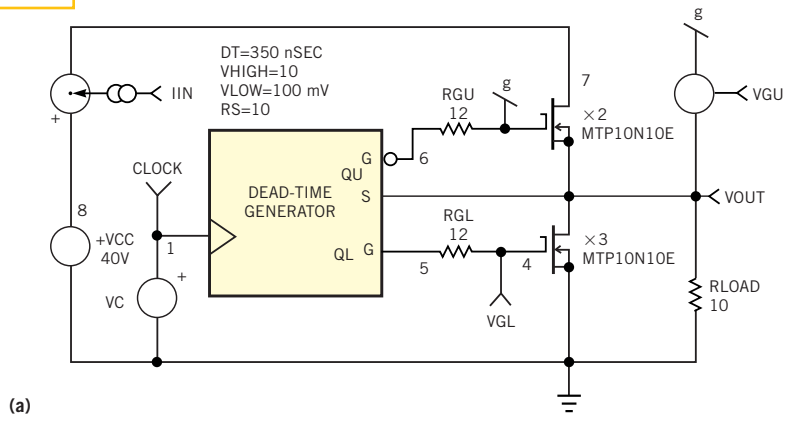
LISTING 1—HALF-BRIDGE DRIVER IN ISSPICE4

```
Listing 1 for DI #2490

.SUBCKT NEWDT CLK GU SU QL {DT=500N VHIGH=10V VLOW=100M RS=1
* Clock_In GateUpper SourceUpper GateLower
*
* DT: Dead time in seconds
* VHIGH: Output level when high
* VLOW: Output level when low
* RS: Driver's output resistance
*
BU1 1 0 V=(V(CLK)>800M) & (V(TD1)>800M) ? {VHIGH} : {VLOW}
BU2 4 SU V=V(1)
RSU 4 GU {RS}
RFLO SU 0 1G
BL 2 0 V=(V(CLKB)>800M) & (V(TD2)>800M) ? {VHIGH} : {VLOW}
RSL 2 QL {RS}
X1 CLK TD1 UTD PARAMS: TD=DT
X2 CLKB TD2 UTD PARAMS: TD=DT
X3 CLK CLKB INV
.ENDS
*INCLUDE DEAD.LIB
**********
.SUBCKT UTD 1 2 {TD=???}
*
*Parameters K=GAIN TD=DELAY
RIN 1 0 1E15
E1 3 0 1 0 1
T1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
.ENDS
**** 1 INPUT INVERTER ****
.SUBCKT INV 1 2
B1 4 0 V= V(1)>800M ? 0 : 5V
RD 4 2 100
CD 2 0 10P
.ENDS INV
```

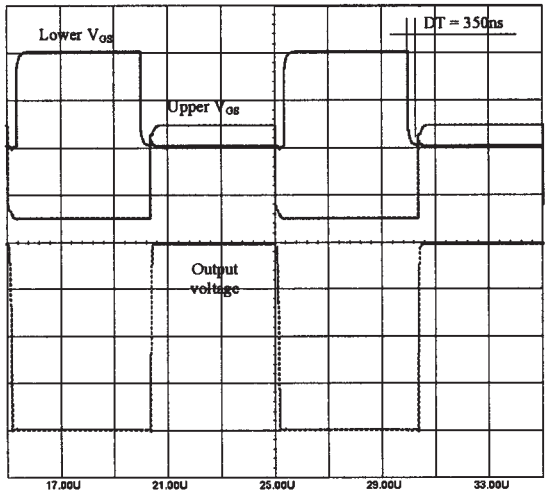
LISTING 2–HALF-BRIDGE DRIVER IN PSPICE

```
.SUBCKT NEWDT CLK GU SU QL PARAMS: DT=500N VHIGH=10V VLOW=100M
RS=10
* Clock_In GateUpper SourceUpper GateLower
*
* DT: Dead time in seconds
* VHIGH: Output level when high
* VLOW: Output level when low
* RS: Driver's output resistance
*
EBU1 1 0 VALUE = { IF ( (V(CLK)>800M) & (V(TD1)>800M), {VHIGH}, {VLOW} ) }
EBU2 4 SU VALUE = { V(1) }
RSU 4 GU {RS}
RFLO SU 0 1G
EBL 2 0 VALUE = { IF ( (V(CLKB)>800M) & (V(TD2)>800M), {VHIGH}, {VLOW} ) }
RSL 2 QL {RS}
X1 CLK TD1 DL PARAMS: TD={DT}
X2 CLKB TD2 DL PARAMS: TD={DT}
X3 CLK CLKB INV
.ENDS
**********
.SUBCKT DL 1 2 PARAMS: TD=500n
*
RIN 1 0 1E15
E1 3 0 1 0 1
T1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
.ENDS DL
**** 1 INPUT INVERTER ****
.SUBCKT INV 1 2
EB1 4 0 VALUE = { IF ( V(1)>800M, 0, 5V ) }
RD 4 2 100
CD 2 0 10P
.ENDS INV
```

Figure 2



(a)



(b)

A typical application for the Spice dead-time generator is for simulating the operation of a half-bridge driver (a). IsSpice4 waveforms show a dead time of 350 nsec (b).

lation of this generator is easy to implement. The netlists in **listings 1** and **2** implement a half-bridge driver with a floating upper output in IsSpice4 (Intusoft) and Pspice (OrCAD), respectively. The BL (**Listing 1**) and EBL (**Listing 2**) inline equations implement the AND gates of **Figure 1a**, which saves you from using a subcircuit arrangement. Typical applications include half-bridge drivers and synchronous rectifiers. You can easily tailor any output polarity by reversing the corresponding Spice element. For instance, if you want to reverse the upper generator, BU1, in **Listing 1**, simply replace the line V=(V(CLK)>800M) & (V(TD1)>800M) ? {VHIGH} : {VLOW} with V=(V(CLK)>800M) & (V(TD1)>800M) ? {VLOW} : {VHIGH}.

Figure 2a portrays a typical application of the dead-time generator in a simplified half-bridge driver, and **Figure 2b** shows the corresponding IsSpice4 waveforms. You can download both listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2490. (DI #2490)

To Vote For This Design, Circle No. 408

Comparison macro for PIC processors

Dennis M Nagel, Dennis M Nagel Inc, Delray Beach, FL

If you ever get tired of trying to remember the subtleties of the "carry" status bit every time you want to use the subtract instruction to perform a comparison, the macro in **Listing 1** can help. (You can download a copy of the listing from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2493.) The macro contains all of the nuances, once and forever. The macro reads like a sentence: branch to target if ram-register is [comparison condition] a literal value. The comparison conditions available are "equal-to," "not-equal-to," "below," and "above-or-equal." The words "below," "above," and others adhere to the Intel/Microsoft assembly-language convention of referring to "unsigned" comparisons for which the byte value can range only from 0x00 (decimal zero) to 0xFF (decimal 255).

LISTING 1—COMPARISON MACRO

```
#define eq              1               ; if equal
#define ne              2               ; if not equal
#define bl      3               ; if below (unsigned comparison)
#define ae              4               ; if above or equal (unsigned)
#define WREG    0xFFFD          ; if immediate-value specified as the W reg

b2              MACRO                   target, ram_reg, cond, lval
   nolist
   if lval != WREG                      ; if lval spec'd as "WREG", don't move lval to W
           movlw lval
   endif
; subtract will set carry if ram_reg >= lval, clear it if reg < lval (unsigned)
           subwf  ram_reg, W    ; ram_reg minus W -> W, ram_reg unchanged
   if cond == eq                ; if cond is "eq", skip if non 0, branch if 0
           skpnz
   else
    if cond == ne               ; if cond is "ne", skip if 0, branch if non 0
           skpz
    else
    if cond == bl               ; if cond is "bl", skip if carry, branch if not
           skpc
    else
    if cond == ae               ; if cond is "ae", skip if not carry, else branch
           skpnc
    else                        ; if condition none of the above, error
           error    "b2 macro condition argument not a valid choice"
endif                   ; cond = ae
endif                   ; cond = bl
   endif                ; cond = ne
   endif                        ; cond = eq
           b        target  ; skip this if condition not true, take it if true
   list
           ENDM ; b2
```

Although you can use this macro for "equal" and "not-equal" comparisons, its real power comes in examining a value in a range or in a window of values. As an example of the macro's use, suppose you want to execute some code, but only if some ram-register is 0x6C or more but not greater than 0x93 (0x94 to 0xFF). You would use:

```
b2 continue_label1, ram-register, b1,
   0x6C
.
source code here to execute only if ram-register = 0x6c to 0x94 inclusive.
.
   continue_label1:
```

A variation of usage is available if you want to perform a comparison between a ram-register and the "W" register instead of a literal value. An example of this usage is:

```
b2 continue_label2, ram-register, b2, WREG
.
source code here to execute only if ram-register is below (less than and not equal to) the current value in the W register.
.
   continue_label2:
```

Note that these examples destroy the value in the "W" register. (DI #2493)

To Vote For This Design, Circle No. 409

Bridging enhances filter close-in selectivity

Richard M Kurzrok, RMK Consultants, Queens Village, NY

General filters are bandpass filters that usually employ bridging couplings between nonadjacent interstage couplings (**Reference 1**). This class of filters also includes bridging coupling across the filter input and output ports. The implementation of input-to-output bridging already exists for a one-pole filter (**Reference 2**). For a two-pole filter, dielectric resonators help achieve input-to-output bridging coupling (**Reference 3**).

You can also implement a two-pole

LC-bandpass filter using input-to-output bridging (**Figure 1**). This general filter provides enhanced close-in selectivity by adding a single bridging inductor across the input and output of a conventional bandpass filter. This relatively obscure passive filter may be useful in some applications, and you can probably realize similar filter performance using an active circuit.

You can compare the performance of this general filter to the performance of a conventional two-pole bandpass filter that has a convenient center frequency of 20 MHz. The design parameters are for a lossless 0.01-dB Chebyshev response with a 3-dB bandwidth of 2 MHz and input and output impedances of 50Ω. **Table 1** shows the relative measured amplitude response data. Center-frequency insertion loss was 0.8 dB, corresponding to inductor unloaded Q's of about 150. The measured relative 3-dB bandwidth was 2.1 MHz. The filter response is asymmetrical due to the frequency sensitivity of the capacitive input, output, and interstage couplings.

The general filter adds a bridging inductor from the input to the output of the conventional bandpass filter (**Figure 1**). The measured center-frequency insertion loss was 0.7 dB, and **Table 2** shows the relative amplitude-response data. The general two-pole filter provides nearby rejection peaks with degraded far-out selectivity. The filter exhibits this type of behavior for upper and lower stopbands. You adjust the variable capacitors with the enclosure cover removed. As filter bandwidth becomes smaller, alignment-tool access holes in the cover become necessary. (DI #2491)

TABLE 1–RESPONSE OF A TWO-POLE CONVENTIONAL BANDPASS FILTER

Frequency (MHz)	Insertion loss (dB)
15	40.3
18	17.8
19	6.2
19.2	4.0
19.4	2.0
19.6	0.8
19.8	0.2
20	0
20.2	0
20.4	0
20.6	0.2
20.8	0.7
21	1.7
21.2	3.0
21.4	4.4
22	8.6
23	14.2
24	17.8
30	26.2

TABLE 2–RESPONSE OF A TWO-POLE GENERAL FILTER

Frequency (MHz)	Insertion loss (dB)
13	9.8
15	11.2
17	14
18	22
18.2	29.2
18.3	>34 (peak)
18.4	28.6
18.6	18.2
18.8	12.6
19	7.3
19.2	4.0
19.4	1.6
19.6	0.5
19.8	0.2
20	0
20.2	0
20.4	0.2
20.6	0.8
20.8	1.8
21	3.6
21.2	6.6
21.4	8.0
21.6	10.4
21.8	13.1
22	16.2
22.5	24.6
23	>39 (peak)
23.5	29.8
24	25.2
25	22
30	20.4

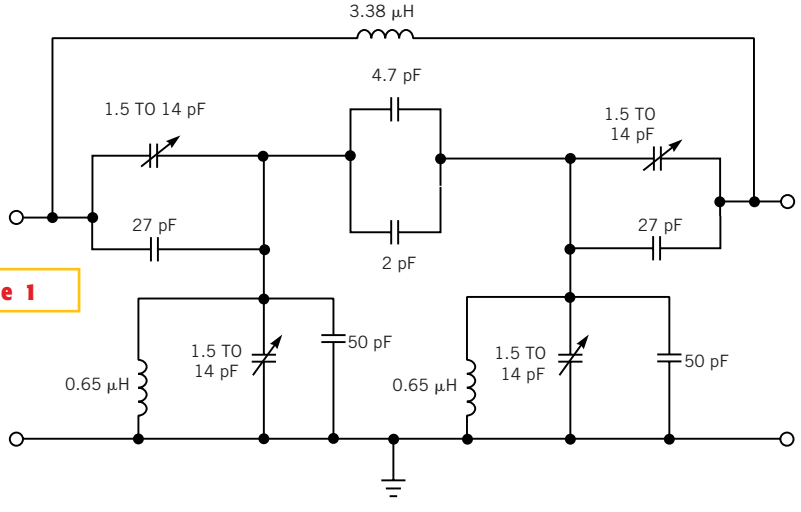


Figure 1

A bridging inductor across the input and output of a conventional two-pole bandpass filter creates a general filter that provides nearby rejection peaks with degraded far-out selectivity.

References

1. Kurzrok, RM, "General three-resonator filters," *EDN*, May 1966, pg 92.
2. Kurzrok, RM, "Single component changes bandpass into general filter," *Electronics*, April 18, 1966 pg 95.
3. Cohn, SB, "Microwave filters containing high-dielectric resonators," presented at Clearwater, FL, May 5 to 7, 1965, and printed in *G-MTT Digest*, pg 49.

To Vote For This Design, Circle No. 410

Notch filter is insensitive to component tolerances

John Guy, Maxim Integrated Products, Sunnyvale, CA

Many approaches for creating notch filters, which reject a narrow band of frequencies and pass all others, are unsatisfactory because they allow the component tolerances to interact. The circuit in **Figure 1a** overcomes this limitation and enables easy calculation of the component values for a desired notch frequency.

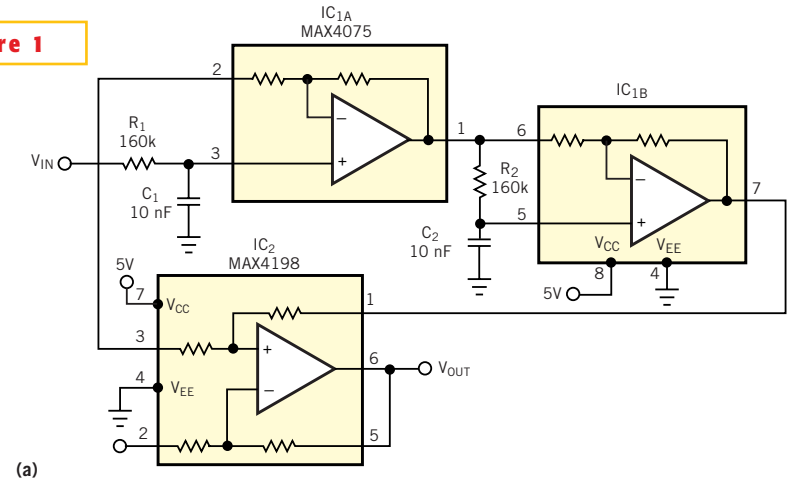
Two allpass filter stages, IC_{1A} and IC_{1B}, create a dc-accurate, 180° phase shift at the cutoff frequency. Each op amp in IC_1 includes gain resistors that match to within 0.1%. This tight tolerance eliminates the need for trimming in most applications. Summing this phase-shifted signal with the input produces a cancellation that produces the notch.

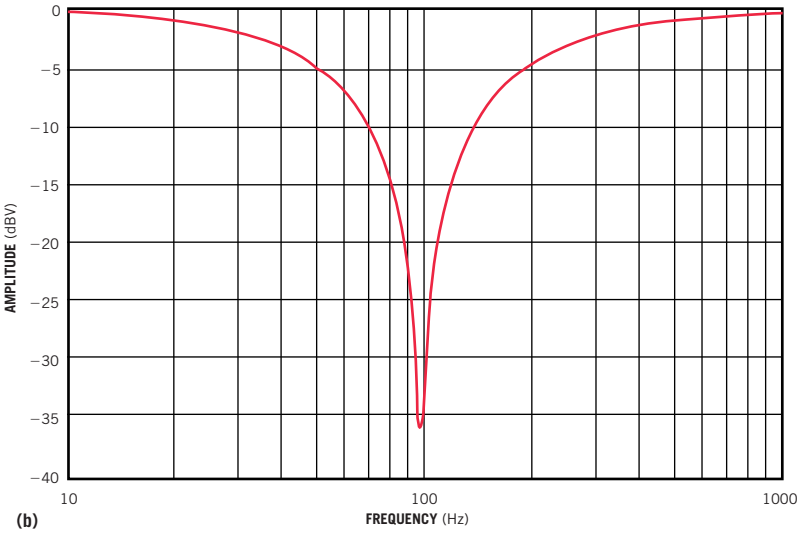
At low frequencies for which the impedance of C_2 is negligible, the circuit forms a voltage follower and produces no phase inversion. For high frequencies, however, this capacitor acts as a short circuit that causes the amplifier to act as a unity-gain inverter with the associated 180° phase shift. Phase behavior for the resulting allpass filter is identical to that of a single RC pole and produces 90° of phase shift at the resonant frequency, which is equal to $1/2\pi R_1C_1$ and $1/2\pi R_2C_2$.

R_1, R_2, C_1, and C_2 affect only the notch frequency and not its depth. Conversely, the integrated resistors in IC_1 affect only the depth of the notch and not its frequency. If you require a highly accurate notch frequency, specify R_1, R_2, C_1, and C_2 accordingly or simply trim one of the two resistors. IC_2 is a precision differential amplifier that the circuit uses as a matched summing amplifier. Note that the inverting input is left unconnected.

Figure 1b shows the circuit's performance with 5% resistors and 20% capacitors, all unmatched. To produce a deeper notch, you can trim the circuit by adding a 100Ω resistor in series with Pin 3 of IC_2. You can also add a 200Ω potentiometer in series with Pin 1 of IC_2 and adjust this potentiometer for maximum rejection at the desired frequency. (DI #2492)

Figure 1





Summing V_{IN} with the output of IC_1's phase-shifting allpass filter results in a notch response (a). Operating the circuit with 5% values for R_1 and R_2 and 10% values for C_1 and C_2 produces a notch frequency of approximately 99 Hz (b).

To Vote For This Design, Circle No. 411

design ideas

Edited by Bill Travis and Anne Watson Swager

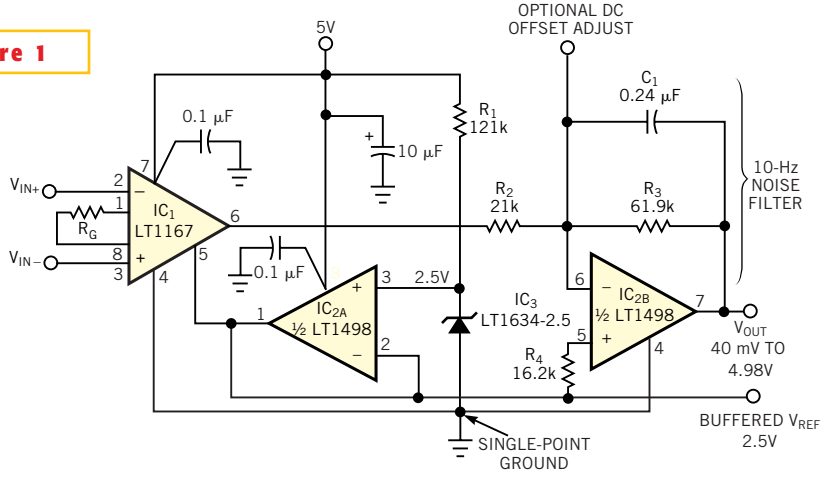
Instrumentation amp works from one supply

Adolpho Garcia, Linear Technology Corp, Milpitas, CA

Many single-supply applications need precision amplifiers that can operate from 5V or lower. Although many precision, single-supply op amps are currently available for configuring as 2- and 3-amplifier instrumentation amplifiers (IAs), these designs require great attention to detail to achieve accuracy and precision. Furthermore, although single-supply IA ICs are available, these products trade off dc and ac performance for low-supply-voltage and -current operation. Dual-supply IAs still offer the best performance.

Achieving high-precision performance in a single-supply application is practical, because the majority of sensor applications provide an output signal centered about the midpoint of a circuit's supply or reference voltage. Examples include strain gauges, load cells, and pressure transducers. In these applications, the signal-conditioning circuitry is not required to operate near the sensor's or circuit's positive supply voltage or ground. Even though the signal-conditioning circuitry need not operate at the extremes of the input-voltage range, the output-voltage swing of the circuitry should be as large as possible to achieve maximum dynamic range. The circuit in **Figure 1** achieves high-precision performance while operating from a 5V supply.

Figure 1



A 2.5V reference IC provides a stable supply midpoint to configure a single-supply instrumentation amplifier.

The trick here is to reference the dual-supply IA's inputs to a stable supply midpoint, then follow the IA with a single-supply precision op amp with a rail-to-rail output swing. This "composite" IA uses IC_1, an LT1167 high-performance IA, for the input stage, and IC_2, an LT1498 high-speed, rail-to-rail input/output dual op amp for the output stage. IC_3, an LT1634 micropower 2.5V precision shunt reference, provides a stable 5V-supply midpoint. The output of IC_3 connects to the input of IC_{2A}, configured as a voltage follower. The output of IC_{2A} provides a low-impedance source for IC_1's reference pin 5, which exhibits 20-kΩ input resistance and input current to 50 µA maximum. A low-impedance source is necessary to maintain IC_1's high common-mode rejection. In addition, IC_{2A}'s output stage can provide load currents to 20 mA for additional external circuitry without affecting IC_3's accuracy.

IC_{2B} is a gain-of-3 inverter whose out-

TABLE 1–SUMMARY OF STATIC AND DYNAMIC CHARACTERISTICS

Circuit gain	RG (Ω)	V*OS (µV)	Bandwidth (kHz)	0.1- to 10-Hz noise* (µV p-p)
10	20.5k	1300	900	2
30	5.36k	450	850	0.7
100	1.5k	160	500	0.4
300	487	100	160	0.3
1000	147	90	40	0.3

*** Referred to input**

put can swing ±2.5V (rail-to-rail) with only ±0.82V drive from IC_1. The primary reason for choosing an inverting-amplifier configuration for the output stage is to make system dc-offset adjustments available. You can connect trim networks to the inverting terminal of IC_{2B} without affecting the static or dynamic behavior of the circuit. However, you should design the trim range so as to not adversely affect the output dynamic range of the circuit.

IC_1 maintains its high-linearity performance with a 5V supply because its front end is configured to operate from dual supplies, and the circuit in **Figure 1** relaxes its output drive level. Because IC_3 level-shifts the entire circuit above ground, you measure the circuit's final output voltage with respect to 2.5V, not 0V. An expression for the gain of this composite IA combines the gain equations of IC_1 and the gain-of-3 inverter:

$$\text{GAIN} = \left[1 + \left(\frac{49.4\,\text{k}\Omega}{R_G}\right)\right] \times \left(\frac{R_3}{R_2}\right)$$

As shown in **Figure 1**, choosing R_G = 1.5 kΩ yields a gain-of-100 composite configuration. You can obtain other gain values with different values of R_G, as shown in **Table 1**. Even though it's not necessary that the inputs to the circuit operate at the positive rail or ground, wide input common-mode operation is always beneficial.

In this configuration, IC_1's input stage can accept signals as high as 3.7V (common mode plus differential mode) with no loss of precision. In fact, at low circuit gains, the circuit's common-mode input-voltage range spans 2.25 to 3.45V. This wide common-mode range allows room for the full-scale differential input voltage to drive the output ±2.5V about the reference point (V_{REF}). Another application hint regarding this circuit: Though IC_1's input bias currents are lower than 1 nA, the circuit's differential-input terminals must have a dc return path to the power supply.

Table 1 summarizes the static and dynamic performance of the composite IA. Nonlinearity for all gain values is lower than 0.006%. The transient response of the circuit as a function of gain and load is well-behaved and is attributable to IC_2's wideband rail-to-rail output stage. Note that measurements of small/large-signal transient response and circuit bandwidth reflect the absence of C_1. The circuit's 10-MHz gain-bandwidth product and 6V/μsec slew rate ensure that the small-signal performance is primarily a function of IC_1's characteristics. Capacitor C_1 is beneficial in low-frequency applications (signal bandwidth lower than 20 Hz), to eliminate or significantly reduce noise pickup. Noise can also sneak into the circuit via the input terminals of IC_1, especially if the sensor is located some distance from the signal-conditioning circuitry. This type of noise can cause a shift in the input offset voltage of IC_1, thereby producing errors. This effect is commonly termed RF rectification. You can easily add a differential filter to IC_1's input terminals to reduce this effect. (DI #2498).

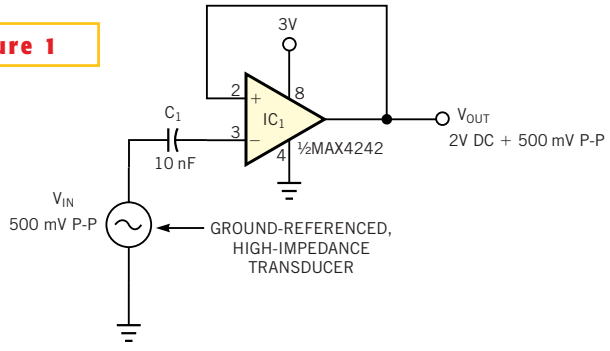
To Vote For This Design,
Circle No. 301

Amplifier requires no dc bias

John Guy, Maxim Integrated Products, Sunnyvale, CA

Intrinsically capacitive transducers and other high-impedance signal sources usually require ac coupling and a buffer amplifier to condition the signal for further processing. Buffers take many forms, but most of them compromise signal quality through the use of external resistors that provide a dc path for the input bias current. Recent improvements in op-amp technology allow ac-coupled inputs without the need for bias resistors. The new op amps have inputs that operate within the supply-voltage span. Some are CMOS, but many use complementary bipolar transistors. For the latter, each input connects to both a npn-difference pair and a pnp-difference pair. Combining these parallel input stages ensures that the second stage has a wide input common-mode range. Thus, some op amps exhibit an input-voltage level for which bias current to the npn pair exactly equals that from the opposite-polarity pnp pair, causing the input bias current to go to zero (**Figure 1**).

Figure 1



A self-biasing amplifier buffers a high-impedance signal without the need for bias resistors.

For the amplifier shown, this bias-canceling effect self-biases the inputs at a level consistently close to $2V_{CC}/3$. To avoid the distortion effects that increase with signal deviations from $2V_{CC}/3$, you should limit the input signal swings to 500 mV p-p. If necessary, downstream circuitry can remove the dc offset while providing gain and filtering. Even with small, low-cost coupling capacitors, this circuit's ultrahigh input impedance and absence of bias resistors allow operation at frequencies well below 1 Hz. (DI #2496).

To Vote For This Design,
Circle No. 302

Halogen light dimmer provides infinite control

Suded Emmanuel, Emmanuel's Controls, Auckland, New Zealand

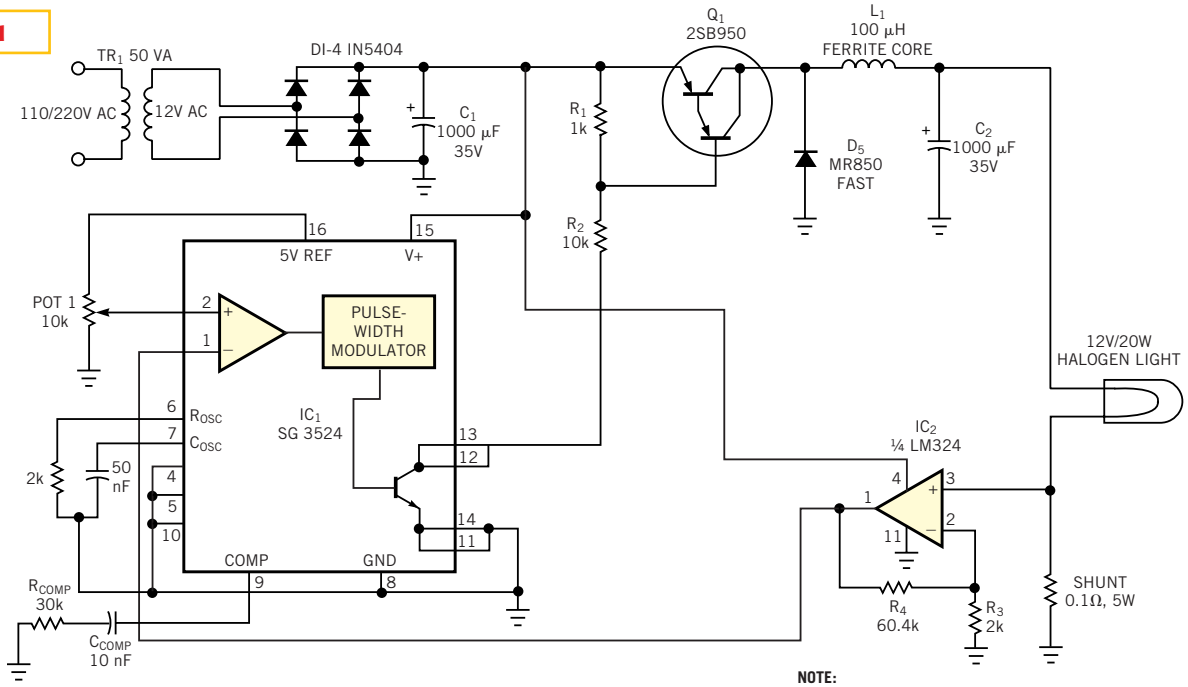
Modern lighting systems use halogen lamps, most of which run on 12V ac from a transformer. The dimmer circuit in **Figure 1** can change the intensity of the light from zero to maximum. The dimmer operates at approximately 12V, unlike the usual ones that function by adjusting the firing angle of the 110 or 220V mains supply.

The dimmer works to inject a constant current into the halogen lamp and to regulate that current using pulse-width modulation (PWM) according to a potentiometer-controlled input, or a 0 to 5V signal, or even an analog output from a µC. 12V ac from the transformer, converted to 16.8V dc, powers the SG3524 PWM circuit (IC_1). An RC circuit sets the approximately 10-kHz operating frequency. The output of the PWM IC drives the power transistor (Q_1), a pnp Darlington. The collector of Q_1 connects to a 10-turn ferrite-core inductor; a 1000-µF capacitor affords filtering to provide the bulb with dc current. Op-amp IC_2 amplifies the drop across the shunt resistor and feeds the amplified signal back to IC_1. IC_1 compares the feedback signal with the desired input level from the potentiometer, 0 to 5V, then controls and regulates the current in the bulb.

Current regulation is important, not only because it makes dimming possible, but also because it protects the bulb at start-up (when the bulb is cold). The constant current gives the filament longer life and makes the bulb immune to line-voltage disturbances. Some designs use "electronic transformers," which are basically switching power supplies that drop the mains voltage from 110/220V ac to 12V dc that's pulsed at high frequency. These systems generate higher RFI than the design in **Figure 1**. In this design, because the controlled variable is current, not voltage, you could use supplies higher than 12V ac to compensate for the drop in the connecting wires in case you wish to place the halogen bulb and dimmer at some distance from the transformer. (DI #2497).

To Vote For This Design, Circle No. 303

Figure 1



NOTE:
FOR A 20W LIGHT, MAXIMUM CURRENT IS 1.6A; THAT IS, A 0.16V DROP ACROSS THE SHUNT. THIS DROP EQUALS 5V INPUT TO THE PWM IC. THE GAIN OF THE NONINVERTING OP AMP IS 31.2.

Current, not voltage, controls halogen-lamp dimming in this simple scheme.

Cascoded stack yields multiple voltages

Clayton Grantham, National Semiconductor, Tucson, AZ

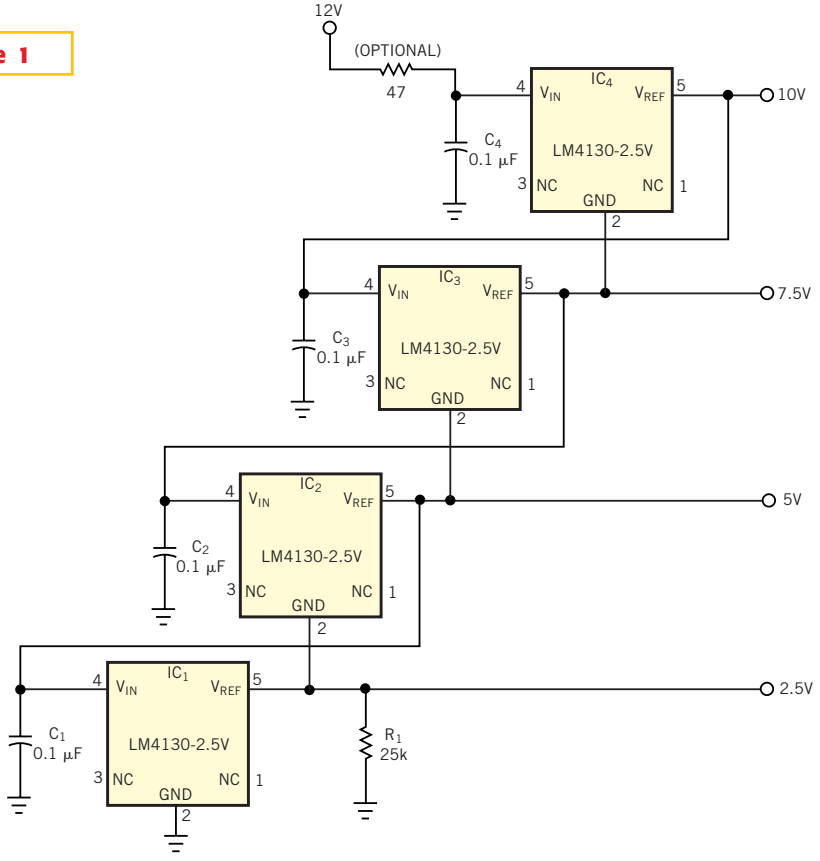
Although the cascode voltage-reference configuration in **Figure 1** may seem obvious, the choice of R_1 and the bypass capacitors is critical. At first glance, stacking up references of the same voltage to produce a collection of voltages seems straightforward. However, nothing comes without precautions and an understanding of limitations.

As a case in point, the CMOS references used in this circuit have a 5.5V supply-voltage limitation. The circuit operates at 12V, which exceeds the limitation; thus each of the individual references must proportionally scale down the 12V input. IC_2's 5V output powers IC_1, IC_3's 7.5V output powers IC_2, and IC_4's 10V output powers IC_3. Similarly, the references take their ground potentials from references beneath them in the cascode, to keep the total supply span for each IC below 5.5V.

The line-regulation errors of IC_1, IC_2, and IC_3 are near perfect, because their individual V_{IN} potentials come from a solid voltage source. The input source of 12V has a range of 10.2 to 14V, but if it dropped below 10.2V, each output voltage would be accurate until the input source dropped to within 200 mV above each stack voltage. For example, if the input source were at 5.2V, the 2.5 and 5V outputs would be within specification and the 7.5 and 10V outputs would be close to 5.2V. In this way, as a 12V battery collapses, external circuitry dependent on the lower voltages would still be functional.

Let's examine R_1 and its limits. The LM4130 sources current very well (to 20 mA), but it sinks only 10 µA. Thus, R_1 is a resistive pull-down for IC_2's quiescent current (50 µA). R_1 must be a maximum of 25 kΩ to keep IC_2 biased for worst-case specs over temperature. This 100-µA bias current also keeps IC_3 and IC_4 biased. The ICs of the stack roughly share a single quiescent current, as opposed to a parallel configuration that would draw four times the quiescent current. It is also true that each output source current has a ripple effect from previous ICs in the stack.

Figure 1



If you understand its limitations, this circuit provides an easy way to obtain accurate multiple voltages.

Thus, cumulatively, the voltage outputs (2.5, 5, 7.5, and 10V) can source as much as 20 mA (5 mA from each output). Load currents of the lower references have effects on the voltages of the ICs stacked above. If the output impedance (0.075Ω) of the LM4130 were not very small, then its effect would create a large crosstalk error, with one output causing another to vary. For example, with the 2.5V output loaded with 20 mA and accounting for the Z_{OUT} effects of both IC_2 and IC_3, the worst-case change in the 10V output is 3 mV, or 0.03%.

The bypass capacitors C_1 through C_4 have a secondary function, other than input bypassing, that overcomes another limitation of the cascode configuration. They compensate the internal LM4130 output. This output stage is a common-source PMOS FET with local feedback that reduces the output impedance beyond 100 kHz. Ceramic, tantalum, or aluminum-electrolytic capacitors work to keep the stack of voltage references from start-up instabilities and oscillations. Another limitation of the cascode arises if the lower voltages (2.5, 5, and 7.5V) become grounded. Any continuous short circuit would produce excessive power dissipation in the LM4130. The optional 47Ω series resistance in the 12V line protects the ICs under worst-case conditions. Note that despite the limitations, the accuracy of the stack voltages tracks the LM4130's accuracy. And, as with accuracy, the temperature-coefficient errors do not degrade up the stack. (DI #2501).

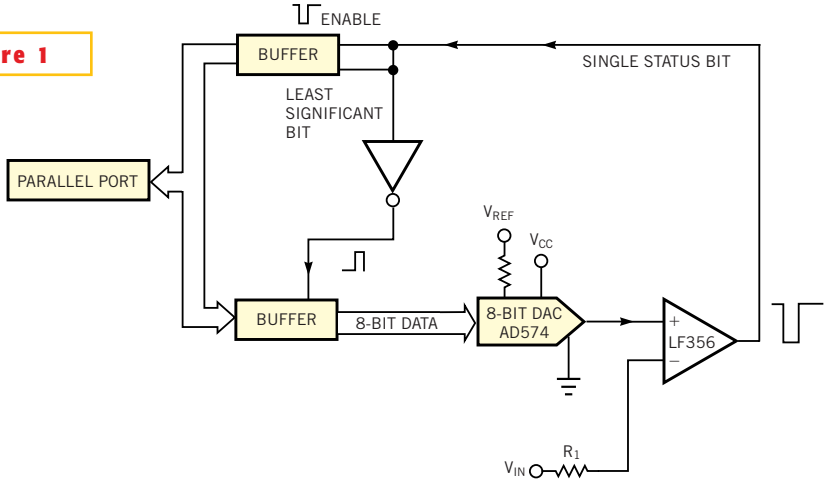
To Vote For This Design,
Circle No. 304

BIOS interrupt performs A/D conversion

J Jayapandian, IGCAR, Tamil Nadu, India

Most A/D-conversion techniques use dedicated hardware, usually a single-chip IC. The flexible conversion technique allows you to use successive-approximation, ramp-type, or other converters by writing the appropriate control software. The design in **Figure 1** uses a PC's parallel port for interface to a DAC and a special BIOS interrupt (INT 1CH) for the conversion process. The INT CH hardware interrupt is available in all PCs. It automatically occurs 18.2 times per second; the BIOS-timer interrupt invokes the interrupt after the interrupt updates the time-of-day count. The INT 1CH handler routine activates the hardware in **Figure 1**. The analog input V_{IN} connects to the inverting input of the LF356 op amp; the noninverting input connects to the output of the AD574 8-bit DAC. The output current of the DAC follows the digital pattern from the PC's port (you can use the LPT port for 8-bit applications).

Figure 1



You can use a PC's special BIOS interrupt to implement an A/D converter.

The interrupt-handler routine (**Listing 1**) for INT 1CH sets the required 8-bit digital image for the DAC's input and monitors the Status bit from the op amp's output. You can write the software for any conversion technique. Here, the INT 1CH handler routine, written in Turbo C, implements a counter-ramp conversion technique. The routine in **Listing 1** uses INT 1CH and the LPT2 port. The vari-

LISTING 1—HANDLER ROUTINE FOR A/D CONVERSION

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#define OUT_PORT  0X378   /* Out port address of LPT2 */
#define CTRL_PORT 0X37A   /* Control port address of LPT2 */
#define INTRTIMER 0x1C    /* BIOS Timer (INT 1CH)Interrupt */
/*-------------------------------------------------------------------*/
/*-------------------------GLOBAL VARS-------------------------------*/
static int ADC_value,STATUS;
static int TICKER;
int i = 0;
void interrupt (*timerhandler)();
void interrupt ADCHANDLER();
/*-------------------------Handler routine for INT 1C----------------*/
void interrupt ADCHANDLER()
{
        disable();
             ++TICKER;
             for(i=0; i<256; i++)
             {
             outportb(OUT_PORT,i);
             ADC_value = i;
             }
             STATUS = inportb(OUT_PORT) & 0x01;
            enable();
} /* END OF ADCHANDLER */

void INSTALLADCHANDLER()
{
        disable();
        timerhandler = getvect(INTRTIMER);
        setvect(INTRTIMER,ADCHANDLER);
        enable();
}

void CLEARADCHANDLER()
{
        disable();
        setvect(INTRTIMER,timerhandler);
*/
        enable();
}

void main(void)
{
        clrscr();
        outportb(CTRL_PORT,0x01);
            INSTALLADCHANDLER();
            while (STATUS != 0x01);
            printf("ADC value is %x\n",ADC_value);
            CLEARADCHANDLER();
          //getch();          /* For testing this can be included */
        return;
}
```

able Ticker recognizes the occurrence of INT 1CH. For every Ticker, the handler routine writes the incremental data from 0 to 255 to the LPT2 port, and thus to the 8-bit DAC. The handler checks for the zero-status bit by reading the LPT2 port. If it reads a high-to-low transition in the Status bit, the handler writes the final bit pattern (a value between 0 and 255) to the DAC. The op amp compares the value of V_{IN} with the DAC's output. When the output reaches V_{IN}, the op amp's output (status) becomes 0V, and the handler stops incrementing the bit pattern presented to the DAC.

This final digital value corresponds to the analog input V_{IN}. For each occurrence of INT 1CH (Ticker), the design completes a conversion cycle. Make sure that the handler routine does not exceed the time of occurrence of the interrupt. You can implement a successive-approximation converter in the same way with this design by writing an appropriate handler routine. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2495. (DI #2495).

Generate frequencies with arbitrary relationships

Sanjay Gupta, NIIT Ltd, New Delhi, India

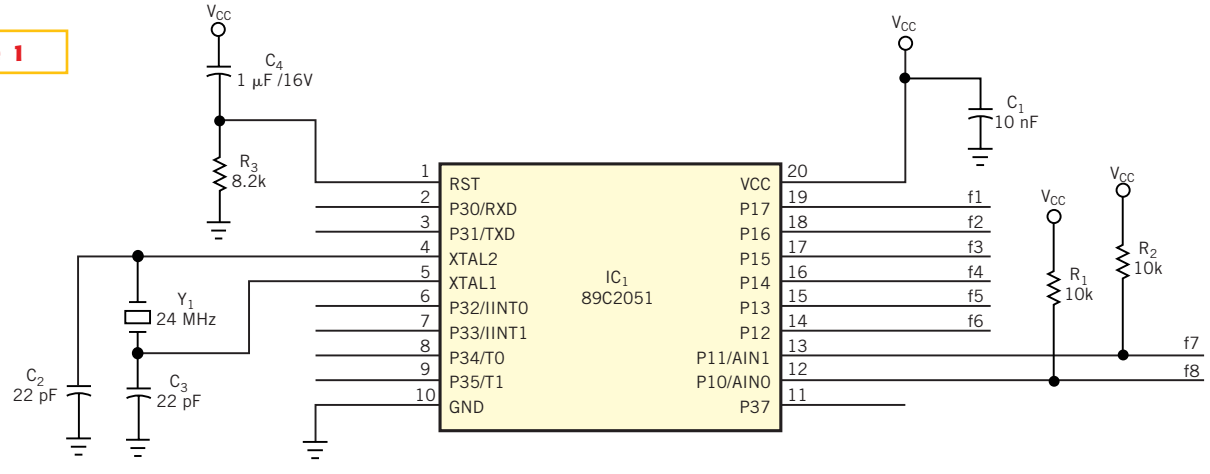
The circuit in **Figure 1** and the program in **Listing 1** use an Atmel 89C2051 µC to generate a variety of frequencies that have no specific relationship to each other. The example given here generates the following eight frequencies: 500, 700, 1000, 1050, 1100, 1500, 1700, and 2000 Hz. The program delivers these frequencies on eight I/O pins (P17 to P10, respectively) of the µC. The program associates each of these I/O pins with an internal-RAM address (f1 through f8).

The interrupt routine in **Listing 1** takes 492 oscillator periods to complete. At 24 MHz, this interval corresponds to 20.5 µsec. The internal timer T0 of the µC generates an interrupt every 25 µsec. During each interrupt cycle, each of the counts (f1 through f8) decrements by one. Whichever count decreases to zero, the corresponding output pin toggles and that count value reinitializes. Average ac-

TABLE 1–PERIOD AND COUNT DATA FOR REPRESENTATIVE FREQUENCIES

Frequency	Period of half-cycle (µsec)	Closest half-cycle time period obtainable with 25-µsec interrupt interval	Error (%)	Count
500 Hz	1000	100	0	1000/25=40
700 Hz	714	700	1.96	700/25=28
1000 Hz	500	500	0	500/25=20
1050 Hz	476	475	0.21	475/25=19

Figure 1



You can generate arbitrary, unrelated frequencies using an 89C2051 µC.

curacy improves as the magnitude of the required output frequency decreases (**Table 1**). You can also obtain more accuracy by using the following techniques:

- Increase the processor-clock frequency, thereby reducing the time it takes to execute the interrupt routine.
- Reduce the number of clock cycles in the interrupt routine.

If you need to generate fewer frequencies, for example, the last six from the preceding list, then the time taken to complete the interrupt cycle reduces to 372 cycles (15.5 µsec). So you can program timer T0 to generate an interrupt every 18 µsec instead of 25 µsec. **Table 2** shows the new accuracy figures with fewer frequencies. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2500. (DI #2500).

TABLE 2–ACCURACY FIGURES FOR SIX FREQUENCIES

Frequency (Hz)	Period of half-cycle (µsec)	Closest half-cycle period obtainable with 18-µsec interrupt interval	Error (%)	Count
1000	500	504	0.8	504/18=28
1050	476	468	1.68	468/18=26
1100	455	450	1.11	450/18=25
1500	333	324	2.7	324/18=18
1700	294	288	2.04	288/18=16
2000	250	252	0.8	252/18=14

To Vote For This Design, Circle No. 306

LISTING 1–ROUTINE FOR GENERATING ARBITRARY FREQUENCIES

```
C1 EQU 40    ; Count for 500Hz
C2 EQU 28    ; Count for 700Hz
C3 EQU 20    ; Count for 1000Hz
C4 EQU 19    ; Count for 1050Hz
C5 EQU 18    ; Count for 1100Hz
C6 EQU 14    ; Count for 1500Hz
C7 EQU 12    ; Count for 1700Hz
C8 EQU 10    ; Count for 2000Hz

F1 EQU 21    ; Internal RAM locations for storing the
F2 EQU 22    ; above count values
F3 EQU 23
F4 EQU 24
F5 EQU 25
F6 EQU 26
F7 EQU 27
F8 EQU 28

org 0
ajmp main;

;----------Timer T0 service routine --------------

org 00BH            ; Timer 0 to service vector

cpl p3.4
djnz F1,CF2 ; 24 osc. periods
cpl p1.7    ; 12 osc. periods
mov F1,#C1  ; 24 osc. periods;
          ; Total Osc.cycles = 60 * 8 + 12 = 492
            ; with 24MHz crystal this amounts to
            ; 20.5 microseconds

CF2:
djnz F2,CF3 ; decrement counter for freq. f2
cpl p1.6    ; if it reduces to 0 then complement the
mov F2,#C2  ; logic on the corresponding pin and reload
            ; the count value

CF3:
djnz F3,CF4
cpl p1.5
mov F3,#C3

CF4:
djnz F4,CF5
cpl p1.4
mov F4,#C4

CF5:
djnz F5,CF6
cpl p1.3
mov F5,#C5

CF6:
djnz F6,CF7
cpl p1.2
mov F6,#C6

CF7:
djnz F7,CF8
cpl p1.1
mov F7,#C7

CF8:
djnz F8,exit
cpl p1.0
mov F8,#C8

exit:
reti        ; End of timer interrupt routine

main:
call init;

LOOP: JMP LOOP;


;------ Initialization Routine -----------
init:
        mov F1,#C1      ;  Initialize count values
        mov F2,#C2
        mov F3,#C3
        mov F4,#C4
        mov F5,#C5
        mov F6,#C6
        mov F7,#C7
        mov F8,#C8

; The following code sets timer T0 to 25 microseconds
; and enables the timer intterupt

        mov TMOD, #02H   ; set timer mode to mode 2
        mov tl0,#0ceh    ; (8-bit auto-reload mode)
        mov th0,#0ceh    ; set reload value
        setb ET0         ; enable timer interrupt
        setb EA          ; master interrupt enable
        setb TR0         ; start timer
ret

end
```

design ideas

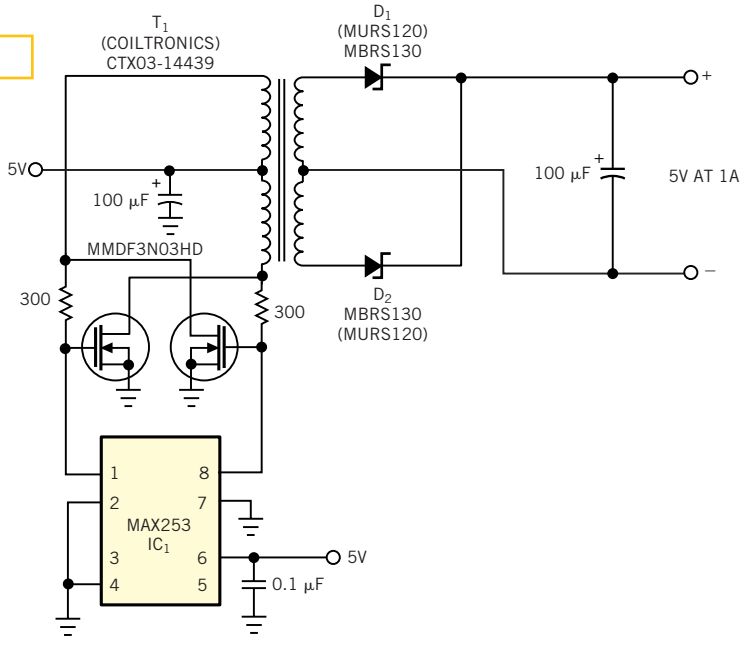
Edited by Bill Travis and Anne Watson Swager

Push-pull driver provides isolated 5V at 1A

Ron Young, Maxim Integrated Products, Sunnyvale, CA

The circuit in **Figure 1** converts a regulated 5V input to an isolated 5V output with 1A current-output capability. IC_1, a push-pull transformer driver, powers a pair of cross-coupled power MOSFETs in a flip-flop-like configuration. In turn, the MOSFETs toggle the primary winding of a forward transformer. The transformer's secondary output, after rectification and filtering, provides the isolated 5V supply. Because the output voltage is unregulated, its voltage tolerance depends on the input-voltage range and the range of load current. With Schottky rectifiers, such as the MBRS130 for D_1 and D_2, the circuit delivers 5V±10% at 700 to 1000 mA from a 5V±5% input with 80% efficiency (**Figure 2**). Using ultrafast-recovery silicon rectifiers, such as the MURS120, the circuit delivers 5V±10% at 200 to 500 mA from a 5V±5% input, with 77% efficiency. (DI #2502)

Figure 1



A simple circuit produces a 5V, 1A isolated output from a 5V regulated input.

To Vote For This Design, Circle No. 366

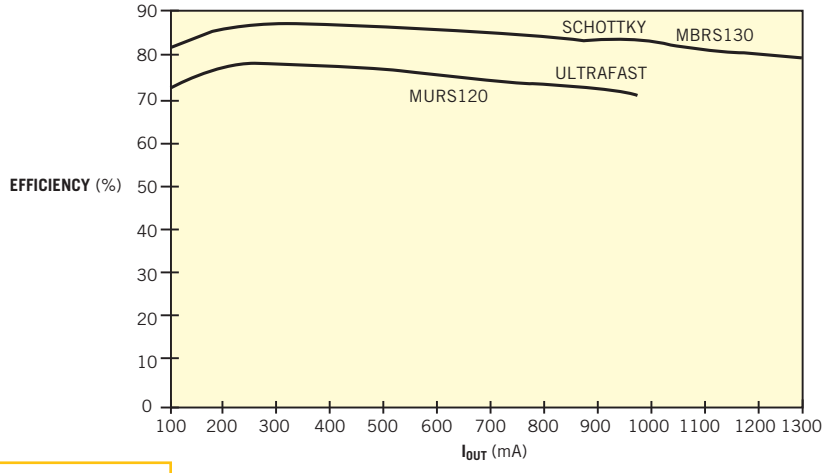


Figure 2

The efficiency of the circuit in Figure 1 depends directly on the forward drops of the output rectifiers.

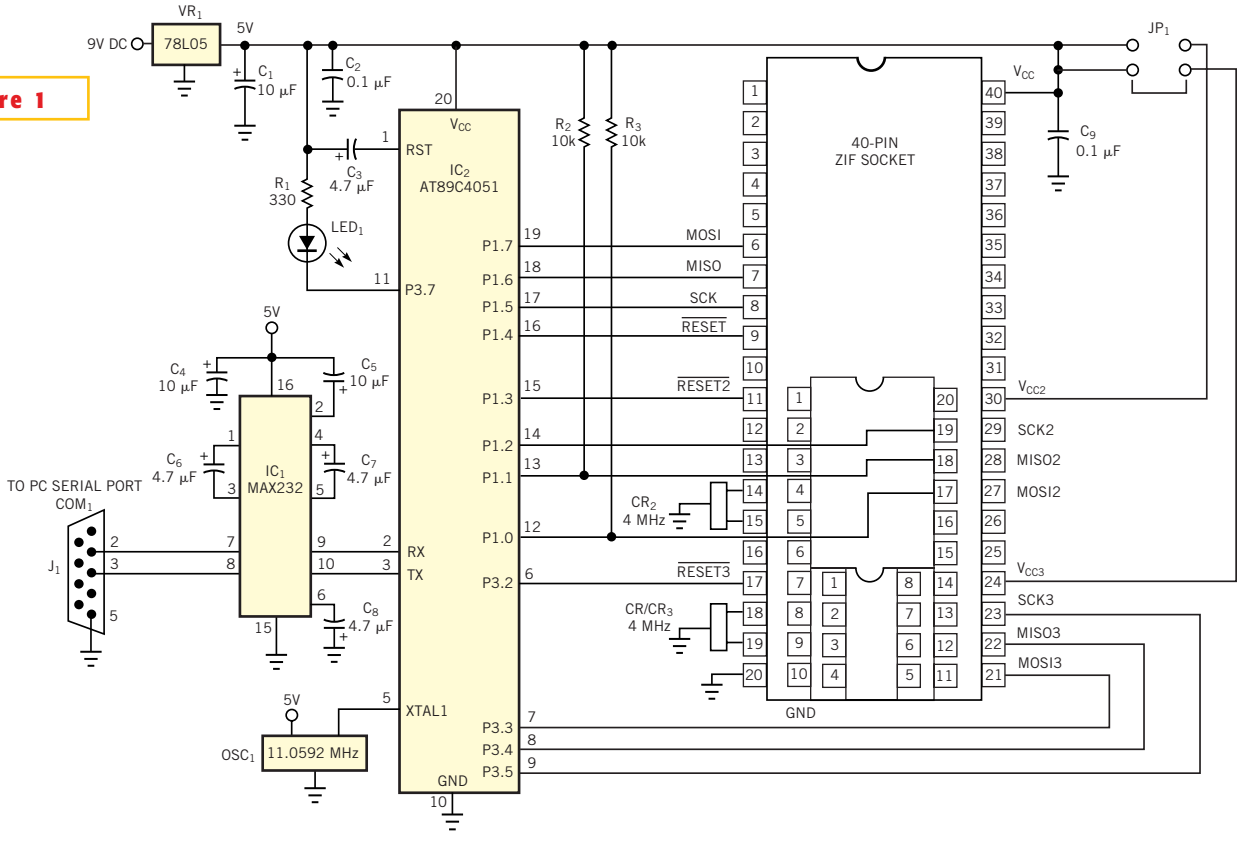
Circuit programs Atmel AVR µCs

Guo-Yin Xu, XuMicro, Houston, TX

Atmel AVR µCs feature an enhanced RISC architecture that purportedly offer the highest MIPS-per-milliwatt capability in the 8-bit µC market. **Figure 1** shows an easy-to-build AVR µC-programming circuit that can program the 40-pin AT90S4414/8515, the 20-pin AT 90S1200/2313, and the eight-pin AT 90S2323/2343. The programmer uses only three chips. It connects to the host PC's serial port via a MAX232 RS-232 transceiver, IC_1. Power comes from a 9V wall cube and the 78L05 linear regulator, VR_1. The AT89C4051 µC (IC_2) works with the 11.0592-MHz oscillator, and controls all programming tasks. LED_1 indicates the programmer's status. The circuit exploits the fact that all AVR µCs have a built-in SPI (serial-peripheral-interface) port that you can use to effect serial programming. The SPI port uses only the system-clock (SCK), master-output/slave-input (MOSI), and master-input/slave-output (MISO) pins. The AVR data book requires that, to place a µC in serial-programming mode, you must first pull the Reset and SCK pins low (**Reference 1**). Then, the µC must execute a programming-enable instruction before it can execute any program of erase instructions.

Hence, you need four pins to control the programming of a µC. For instance, the control signals from the AT89C4051 port pins P1.4 to P1.7 are for 40-pin µCs, the signals from P1.0 to P1.3 are for 20-pin µCs, and the signals from P3.2 to P3.5 are for eight-pin µCs. Note that you need pullup resistors for the AT89C4051 port pins P1.0 and P1.1, because these pins normally serve as analog-signal-input lines. Jumper JP_1 controls the 5V power supply for 20- and eight-pin µCs. You should remove the jumper when programming 40-pin µCs, which use a hard-wired connection to the 5V power. The circuit uses two 4-MHz ceramic resonators: CR_2 for 20-pin µCs, and CR/CR_3 for 40- or eight-pin µCs. Because the wires for the resonators should be as short as possible, the circuit uses no switching or jumping mechanisms. Instead, it uses one more resonator, CR_2 that's hard-wired to pins 4 and 5 of the

Figure 1



Exploit the power of Atmel's AVR µCs, using this easy-to-build programmer.

20-pin µC pins 14 and 15 of the ZIF socket. This connection does not disturb the programming of 40-pin µCs.

In addition to the hardware in **Figure 1**, the programmer also needs associated software. A binary file, AVRP1.BIN, holds the finished AVR-programmer software, burned into the AT89C4051 µC by using an 8X51 EPROM/flash µC programmer (**Reference 2**). A DOS file, AVRP1.EXE, contains the host-PC communication program. You can download these routines from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2504. (DI #2504)

References

1. "8-bit RISC Microcontrollers Data Book," Atmel Corp, August 1999.

2. Xu, Guo-Yin, "8X51 EPROM/flash microcontroller programmer," *Circuit Cellar Magazine*, April 1998.

To Vote For This Design, Circle No. 367

Circuit adds latch-off current limit to regulator

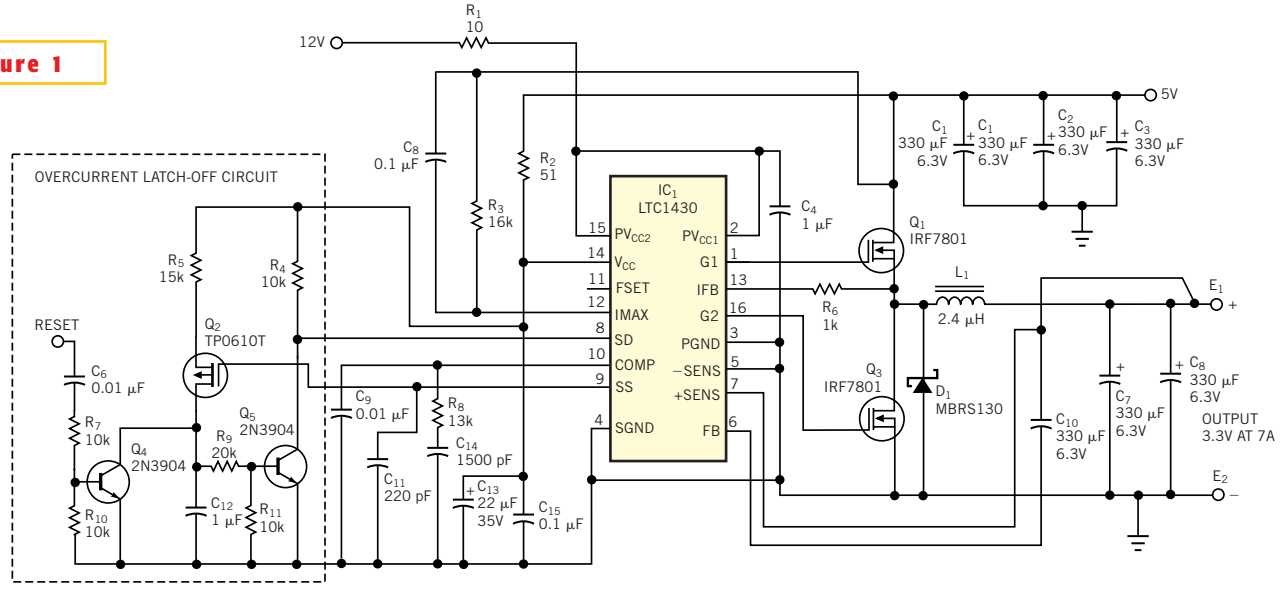
Craig Varga, Linear Technology Corp, Milpitas, CA

In many applications, forcing a high-current power supply to latch off if a sustained fault condition exists can minimize the likelihood of damage to the pc-board traces and the power devices in the supply. Pulse-width-modulation (PWM)-control circuits provide no latch-off feature, but the circuit in **Figure 1** does. The circuit is based on IC_1, an LTC1430 PWM controller. The current-limiting feature of the IC operates by sensing the voltage across the high-side MOSFET and compares it with a threshold voltage developed across R_3. If an overcurrent condition exists, an internal current source starts to discharge the soft-start capacitor, C_9. At this point, the latch-off circuit begins to take over. When the voltage across C_9 decreases to a couple of volts below V_{CC}, Q_2 turns on and begins to source current, charging C_{12}.

After a time interval depending on the values of R_5 and C_{12}, Q_5 turns off and pulls the shutdown pin low, thereby turning off the controller. Because this action internally grounds the soft-start pin (to allow a normal soft-start cycle at turn-on), the circuit remains latched in the off state. You can initiate a reset by applying a fast logic high to the reset line. C_6 and C_7 provide a differential pulse to the base of Q_4, which discharges C_{12}, allowing the regulator to restart. If you don't need the reset function, you can eliminate C_6, R_7, R_{10}, and Q_4. You can then initiate a restart by recycling the 5V input power. The only timing requirement is that the latch-off delay be greater than the soft-start rise time at turn-on. Otherwise, the regulator can never start. You can modify the circuit to work with any other controller, such as the LTC1553, having the soft-start function. (DI #2503)

To Vote For This Design, Circle No. 368

Figure 1



You can add a latch-off current-limiting feature to a simple pulse-width-modulation controller by adding a few external components.

Dual power supply delivers 8A with no heat sinks

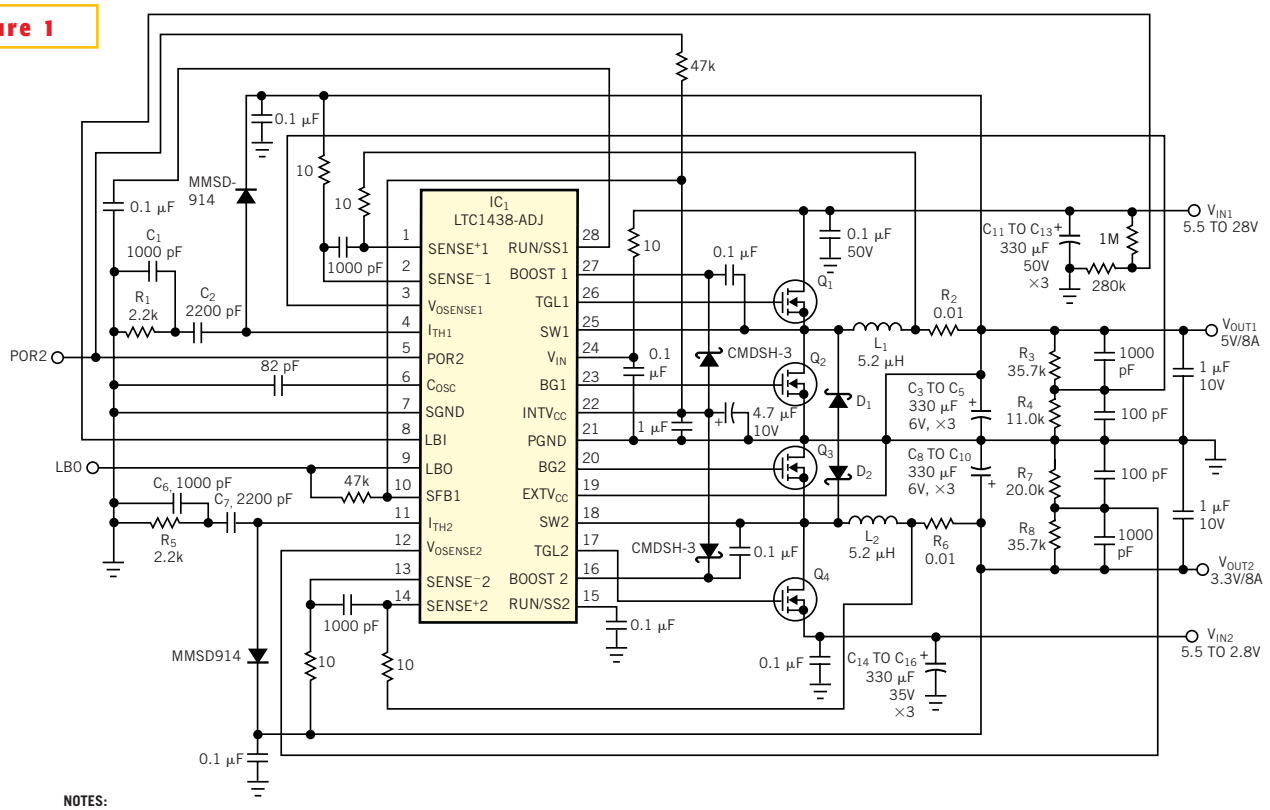
John Seago, Linear Technology Corp, Milpitas, CA

THE CIRCUIT IN **Figure 1** is a high-current dual supply that provides 5 and 3.3V at currents as high as 8A. The circuit uses a fixed-frequency, two-output, current-mode synchronous-buck-controller, IC_1, to regulate both 5 and 3.3V outputs. The circuit uses separate regulator circuits for each output voltage. However, both circuits are identical except for the lower feedback resistors, R_4 and R_7, which determine the 5 and 3.3V output voltages, respectively.

IC_1 regulates the 5V output, V_{OUT1}, by controlling the duty cycle of the top MOSFET for V_{OUT1}, Q_1, so that average input voltage to the buck inductor, L_1, is equal to the output voltage. The buck inductor and output capacitors C_3 to C_5—three 330-µF capacitors in parallel—integrate and filter the energy pulses from Q_1 to generate the dc output. After Q_1 turns off, the bottom MOSFET for V_{OUT1}, Q_2, turns on to conduct inductor current to the load. To avoid shoot-through current, a short dead time occurs before each MOSFET turns on. During this dead time, inductor current flows through the commutating diode, D_1, to the load. Feedback resistors R_3 and R_4 connect IC_1's internal error amplifier to the output. Loop-compensation components R_1, C_1, and C_2 control the frequency response of the error amplifier. The internal current comparator senses inductor current by the voltage developed across the current-sense resistor, R_2.

The 3.3 regulator, which produces V_{OUT2}, functions exactly like the 5V regulator. Q_3 and Q_4 are the top and bottom

Figure 1



NOTES:
C_{11} TO C_{13}, C_{14} TO C_{16}=SANYO 35CV330GX.
C_3 TO C_5, C_8 TO C_{10}=KEMET T495X337M006AS.
D_1, D_2=MOTOROLA MBRD835L.
L_1, L_2=PULSE ENGINEERING PE-53700.
Q_1, Q_4=SILICONIX SUD50N03-10.
Q_2, Q_3=SILICONIX Si4420DY.
R_2, R_6=IRC LRF2512-01-R010-J.

A single IC regulates both 5 and 3.3V outputs; each output delivers 8A.

MOSFETs, respectively, and L_2 is the buck inductor. D_2 is the commutating diode. Feedback resistors R_7 and R_8 connect the error amplifier to the output. R_5, C_6, and C_7 are the loop-compensation components, and R_6 is the sense resistor. C_8, C_9, and C_{10} make up the output capacitor.

The circuit in **Figure 1** has some features that add versatility. The low-battery comparator in IC_1 flags a low-input-voltage condition. Normally, the LBO pin is high but goes low when the input voltage is low. IC_1 includes a complete power-on-reset circuit. At startup, the POR2 pin is low. This pin goes high 65,536 oscillator cycles after Channel 2's output voltage reaches 95% of its programmed value. The POR2 pin goes low if the output voltage falls 7.5% from nominal. Each output has a RUN/SS pin that provides output-voltage delay, output-current soft-start, and on/off control. The value of the capacitor connected to the RUN/SS pin determines the output voltage delay and the inductor-current ramp time, both at a rate of 0.5 sec/µF. Pulling a RUN/SS pin low turns off that output voltage. Pulling both RUN/SS pins low shuts down IC_1, turns off all internal circuitry, and limits the input current to 16 µA. (DI #2494)

To Vote For This Design, Circle No. 369

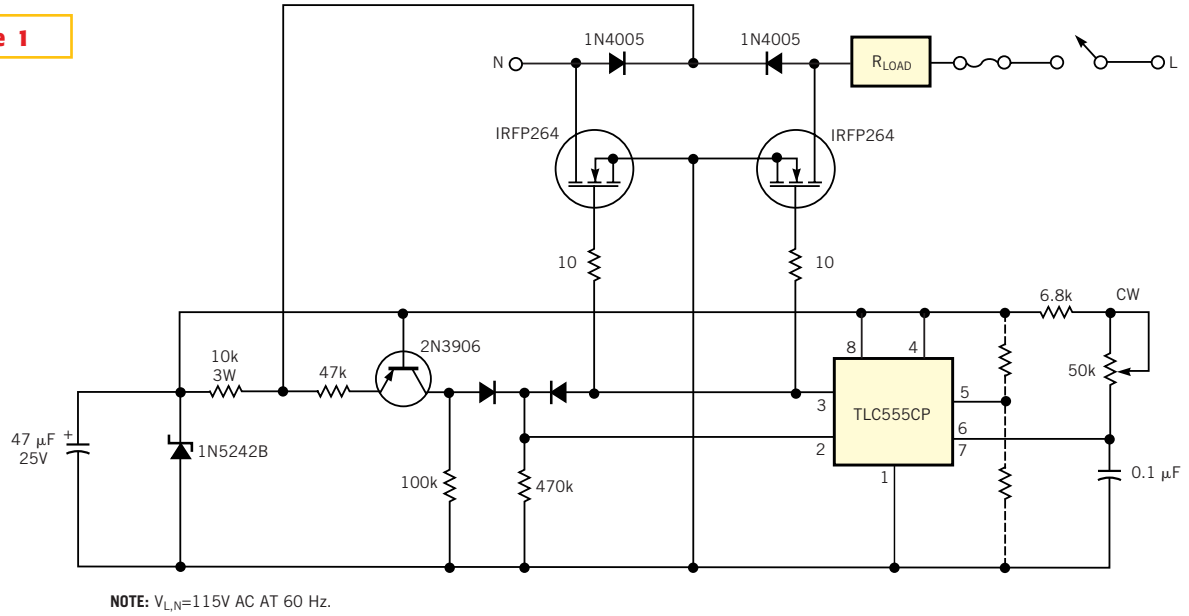
Power switch provides soft start

John Haase, Colorado State University, Fort Collins, CO

In the circuit in **Figure 1**, series-connected MOSFETs turn on the line voltage near the zero-crossing point and off when the 555-timer delay lapses. That delay ranges from 1 to 7 msec. The MOSFETs' body diodes and the 1N4005 diodes form a full-wave bridge rectifier that provides a floating 12V-dc level via the 10-kΩ, 3W resistor. The bridge simultaneously delivers the crossing signal to 2N2906 common-base comparator. When the emitter current approaches zero, the collector voltage falls to 4V and triggers the delay timer, initiating gate drive to the switches. Positive-going pin 3 of the 555 also removes the trigger threshold by coupling to the diode OR gate. Because the circuit generates the 12V operating voltage only during off-time, the limit for maximum delay is approximately 7 msec. You trim the delay by selecting a resistance value from Pin 5 to raise (pins 5 to 8) or lower (pins 5 to 1) the upper comparator threshold at Pin 6. Thus, load power is from 0 to 90% of maximum (600W). You must use a heat sink for the power MOSFETs. (DI #2506).

To Vote For This Design, Circle No. 370

Figure 1



NOTE: $V_{L,N}$=115V AC AT 60 Hz.

This smart switch provides a small initial current for loads with a normally high inrush current.

Circuit eliminates PC echoes

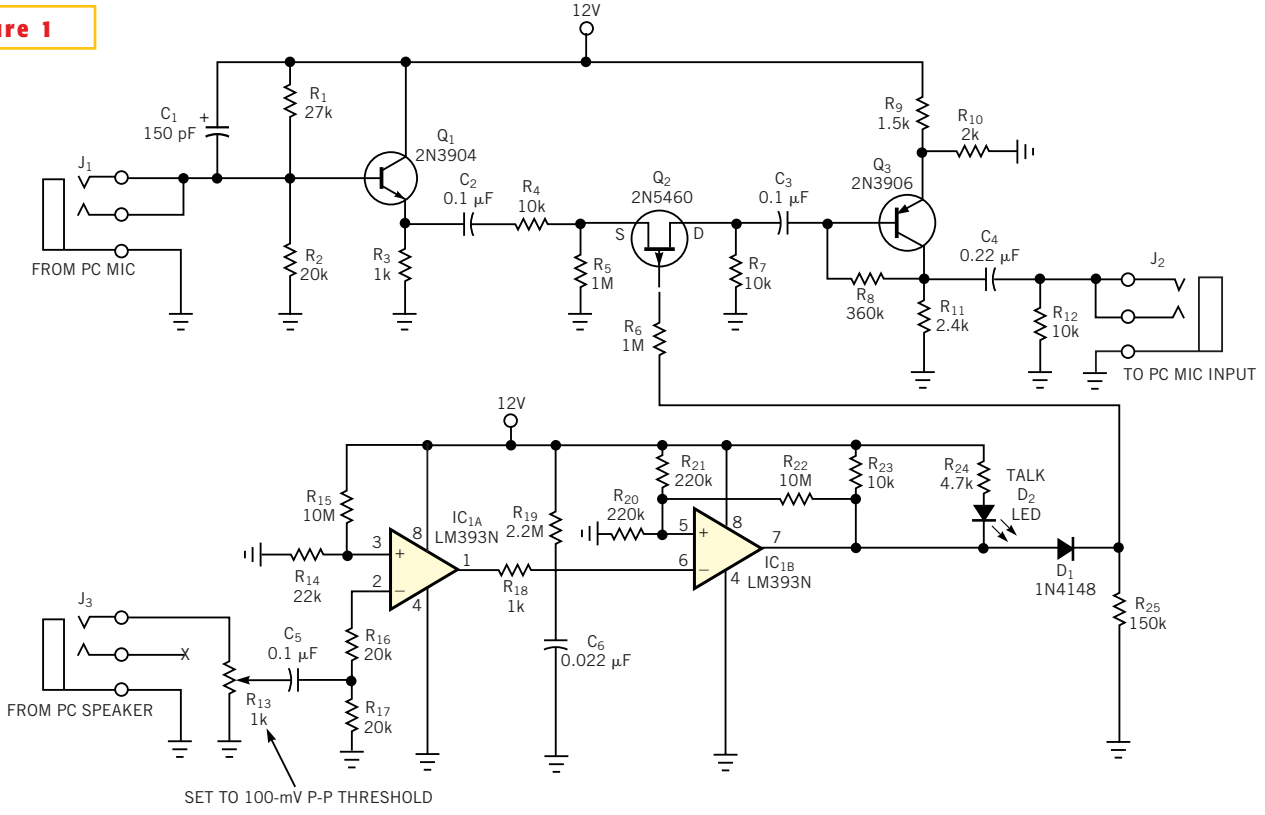
Hans Krobath, EEC, Nesconset, NY

Long-distance-telephone services available via the Internet often require the PC user to wear headphones of a headset to prevent echo caused by the microphone's picking up the loudspeaker outputs. The circuit in **Figure 1** eliminates the echo while using the existing PC microphone and speakers for a comfortable conversation. The interface is between a standard electret condenser microphone and the microphone input of the PC. The loudspeaker output of the PC serves to mute the microphone input. R_1 and R_2 provide biasing for both the electret microphone and the Q_1 emitter follower. Q_2, a p-channel FET, acts as a switch that opens with the application of a gate voltage greater than 6V and closes with a gate voltage of 0V. Q_3 compensates for the switching-circuit losses and buffers the output. R_9 and R_{10} provide an appropriate input impedance to Q_3 and limit the output to 5V p-p, thus preventing any possible damage to the PC's microphone input.

IC_1, which acts as low-level retriggerable monostable multivibrator, controls the Q_2 FET switch. Loudspeaker voltage levels as low as 15 mV from the PC cause comparator IC_{1A}'s open-collector output to discharge C_6 via R_{18}. The falling voltage of C_6, passing the threshold of comparator IC_{1B}, produces a high output that turns off Q_2. Any input from the PC's loudspeaker output discharges C_6. The absence of an input allows C_6 to charge within approximately 40 msec to the IC_{1B} threshold, producing a low-level output and turning on Q_2. LED D_2 lights whenever no loudspeaker output is present, and the microphone input to the PC becomes enabled. D_1 reduces the Q_2 gate voltage to 0V when the IC_{1B} output saturates. You should set R_1 such that approximately a 100-mV p-p microphone input just triggers IC_1, as indicated by the LED's extinguishing. This level prevents any noise from the PC's loudspeaker output from falsely triggering the monostable multivibrator. (DI #2508).

To Vote For This Design,
Circle No. 371

Figure 1



Eliminate annoying echoes from loudspeaker-microphone feedback by using this simple circuit.

Clamping circuit dissipates minimal power

Carlisle Dolland, Allied Signal Aerospace, Torrance, CA

THE CIRCUIT IN **Figure 1** is a quasi-linear regulator. It functions as a source follower for input voltages greater than a preset level, determined by VR_3. For input voltages lower than the preset level, the pass element, Q_2, operates as a saturated switch. The circuit comprises an oscillator, a charge pump, and a linear regulator. The linear regulator, consisting of Q_1, R_1, VR_1, and R_2, drives a charge pump comprising C_2, D_2, D_3, and C_3. The charge pump generates a voltage equal to the output of the linear regulator for input voltages greater than the breakdown voltage of VR_1. For input voltages lower than VR_1's breakdown voltage, the output voltage is $V_{OUT} = V_{IN} - I_{OUT} \times (R_{ON}$ of Q_2). For input voltages that exceed VR_3's breakdown voltage by approximately 3V, R_6 and VR_3 dissipate the energy the charge pump supplies. In this mode, the circuit functions as a source follower, and the output voltage is approximately 3V lower than VR_3's breakdown voltage. The circuit dissipates minimal power. During transients, the load current determines the dissipation in Q_2. (DI #2499)

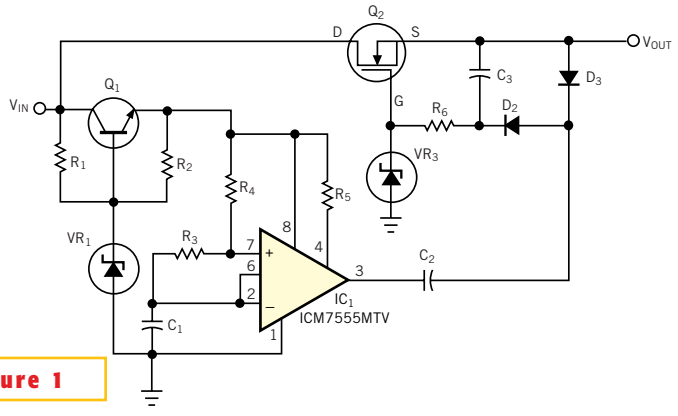


Figure 1

This circuit clamps transient voltages and dissipates minimal power.

To Vote For This Design, Circle No. 372

Piezo crystal monitors liquid level

J Jayapandian, IGCAR, Tamil Nadu, India

THE SIMPLE AND INEXPENSIVE circuit in **Figure 1** monitors the liquid level in a container. The piezo crystal, carefully mounted at the bottom surface of the container, receives it activation from the 74HCT14 hex Schmitt trigger. The crystal generates stable clock pulses according to its specification (for example, 6.14 MHz) when it is in free air. The crystal-based clock drives the 0th counter in an 8254 programmable-counter/timer chip, programmed in Mode 0 as an event counter. The first counter of the 8254, programmed in Mode 1 as a retriggerable one-shot whose time period is 1 sec, controls the gate of the 0th counter. The first counter allows the 0th counter to count for a period of 1 sec. The counts in the air medium serve as a reference. Depending on the height of the liquid level, the pressure acting on the surface of the crystal increases, thereby reducing reference-crystal clock frequency. The variation in clock frequency with respect to the air-medium reference gives you the height of the liquid level. (DI # 2507)

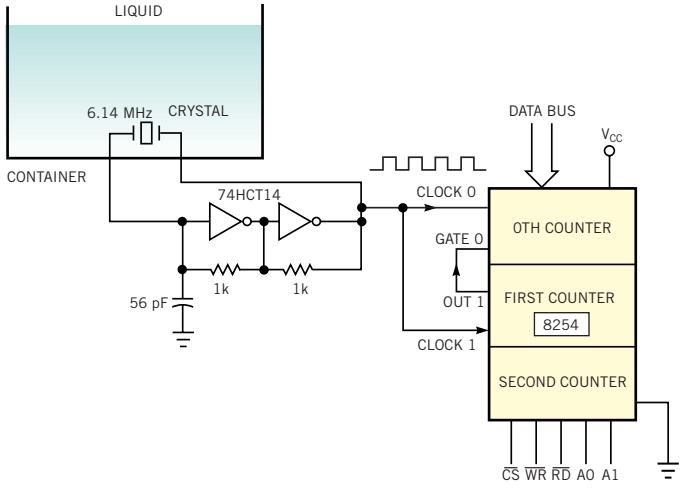


Figure 1

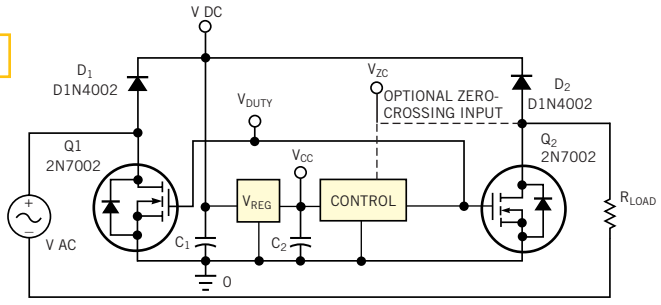
By measuring frequency shifts, this circuit provides a measure of liquid level.

To Vote For This Design, Circle No. 373

Switch intelligently controls current

Jim Hartmann, Silent Knight LLC, Maple Grove, MN

Figure 1



With a control circuit of your choice, you can obtain intelligent control of ac or dc current.

The circuit in **Figure 1** can intelligently control ac or dc current when connected in series with a load. The circuit "steals" its power by turning off the load at a low duty cycle. The switch uses the MOSFETs' parasitic body diode to its advantage. While the MOSFETs are off, the body diodes, along with D_1 and D_2, serve as two legs of a diode bridge. Current flows through the load and the bridge, charging C_1 to the peak ac or dc voltage. The relatively small control-block supply current continues to flow through the load when the load is turned off. The circuit has low insertion loss because of the MOSFET's bidirectional nature. The control block connects power to the load by turning MOSFETs Q_1 and Q_2 on. On alternating cycles, either Q_1 or Q_2 becomes reverse-biased, but current does not flow through the body diode because the MOSFET can conduct in either direction. The insertion loss is equivalent to the loss in two times the MOSFET's on-resistance.

While the load is turned on, the control block draws current from C_1. The circuit must periodically recharge C_1 by briefly turning off the load . You can allow the duty cycle to go as high as 99.99% with a high-current load and a micropower control circuit. The maximum duty cycle is approximately $I_{LOAD}/(I_{CONTROL}+I_{LOAD})$. For example, with a 1A load and a control-circuit current of 1 mA, the maximum duty cycle is 99.9%. By choosing MOSFETs and diodes with higher current ratings, you can adapt the circuit to control high-power loads. Many applications are possible—lamp dimmers and thermostats, for example. The controller can optionally synchronize to the ac zero-crossing point as shown. (DI # 2505)

To Vote For This Design,
Circle No. 374

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02458

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____

E-mail _____ **Fax** _____

Company _____

_____

Address _____

_____

Country _____ **ZIP** _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to *EDN*, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit text and listings by e-mail to b.travis@cahners.com or send a disk.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

design ideas

Edited by Bill Travis and Anne Watson Swager

FPGA implements X.50 Division 3 recommendation

Andres Martinez, Alcatel, Ramirez de Prado, Spain

THE SCHEME IN **Figure 1a** uses five delay cells and an XOR gate to configure the data stream for the X.50 Division 3 recommendation of ITU-T. The X.50 recommendation defines the fundamental parameters of a multiplexing scheme for interworking data networks using different envelope structures. Division 3 applies to the interworking between two networks, both of which use the 8-bit envelope structure. X.50 recommends a pattern of 19+1 bits, which the following primitive polynomial generates: $1+x^2+x^5$. With this polynomial and the initial conditions, **Table 1** generates the pattern 01110. The polynomial does not directly generate the first value, A, in the table because this value depends on the rest of the system. The scheme in **Figure 1a** is typical of a scramble, and is one part of a transmitter/receiver system for data communications. You can implement **Figure 1a** in a 30,000-gate FPGA.

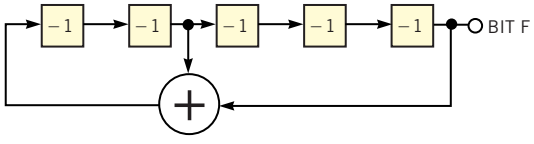
TABLE 1—GENERATING THE 0110 PATTERN

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Value	A	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0

It is important to set the delay cells to the correct initial value. For this design to achieve the correct data stream, the initial data should be 0010110. The use of a 7-bit word for the initial data enables you to change the design to X.50 Division 2 by changing only the initial word and the position of the XOR gate. The process for achieving the polynomial is as follows:

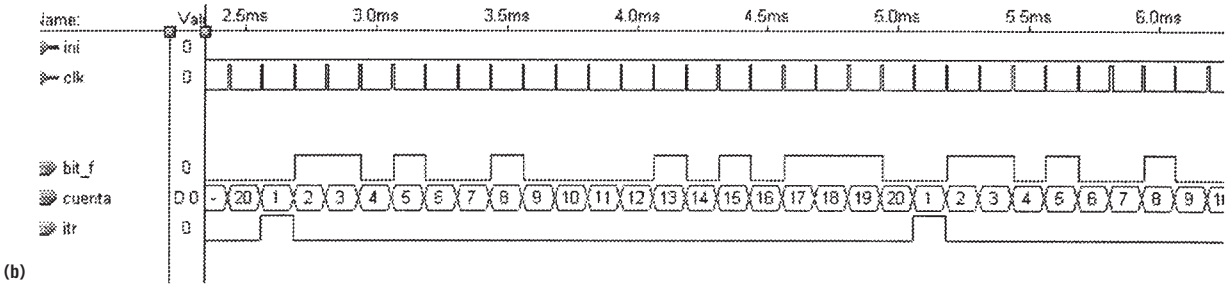
```
gen_scr:process(clk, count,scr, aux, ini)
 begin
   if (ini or aux)= '1' then
        scr_new <= "0010110";
   elsif clk'event and clk = '0' then
        for i in 1 to 6 loop
          scr_new(i-1) <= scr(i);
        end loop;
          scr_new(4) <= scr(0) xor scr(3);
   end if;
end process;
```

The clk signal is not exactly a true clock signal. This design uses a 244-nsec-wide



(a)

Figure 1



(b)

Five delay cells and an XOR gate (a) configure the data stream (b).

clock pulse and a 125-µsec period. Rising edges generate the interval number, and falling edges generate the data out. Because 244 nsec is much less than 125 µsec, data out is present in all intervals. You can download the source file from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2509. The timing diagram in **Figure 1b** shows the resultant data. (DI #2509)

To Vote For This Design,
Circle No. 313

Concurrent processing produces fast priority selector

Itamar Elhanany, Ben-Gurion University, Beer-Sheva, Israel

Priority-encoding circuits are common in high-speed digital applications, such as interrupt controllers and task schedulers, for selecting the highest priority set bit of all set bits in a given binary vector. Typically, the bit index corresponds to its priority level. A project required that several logic-function blocks, as opposed to one, would be enabled concurrently at any given time. Accordingly, the design goal was to identify the M highest priority admissible logic blocks. A conventional 1-bit priority selector receives as its input an arbitrary binary vector and outputs a vector of the same size, where the location of the set bit corresponds to the position of the highest priority set bit in the input vector. Such a selector may be described by the following Boolean expressions:

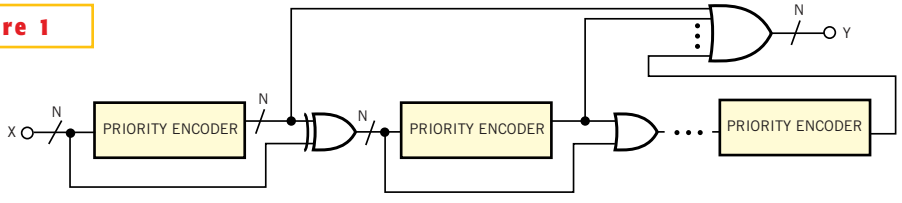
$$Y_i = \begin{cases} X_1 \\ \left(\overline{\sum_{j=1}^{i-1} X_j}\right) \bullet X_i = \left(\overline{\sum_{j=1}^{i-1} X_j}\right) \bullet X_i \end{cases} \begin{matrix} i = 1 \\ i > 1. \end{matrix}$$

Figure 1 depicts a straightforward realization of an N-bit priority selector by means of cascading 1-bit priority selectors. Despite its simplicity, the design introduces extensive delay, which grows linearly with M.

An alternative implementation (**Figure 2**) is based on processing elements of the input vector concurrently as opposed to sequentially. The architecture comprises a bit-count function, which counts the number of set bits in its input vector, and a comparator. Parallel bit counts are deployed to concurrently determine the number of set bits in subsets of the input vector. A logical 1 at the comparator's output indicates that the bit count is smaller than or equal to M. The result is a binary mark signifying the range of bits in the input vector, which contains the M highest priority set bits. Note that, if N is the input vector length, then only N-M paths are necessary because the first M bits of the input vector contain by definition at most M set bits. You obtain the final output as a product of the input bits and the corresponding mask bits. The complexity of the bit-count function is O(logN); hence, the proposed scheme offers a high-speed invariant to M. (DI #2515)

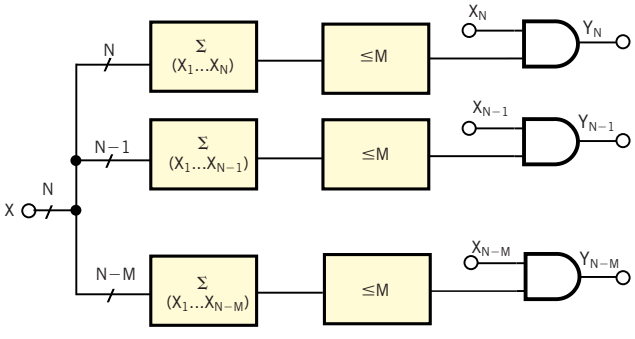
To Vote For This Design,
Circle No. 314

Figure 1



A straightforward realization of an N-bit priority selector involves cascading a series of 1-bit priority selectors.

Figure 2



An alternative priority selector processes elements of the input vector concurrently.

Proportional thermoregulator synchronizes to line

Jordan D Dimitrov, N Poushkarov Institute of Soil Science and Agroecology, Sofia, Bulgaria

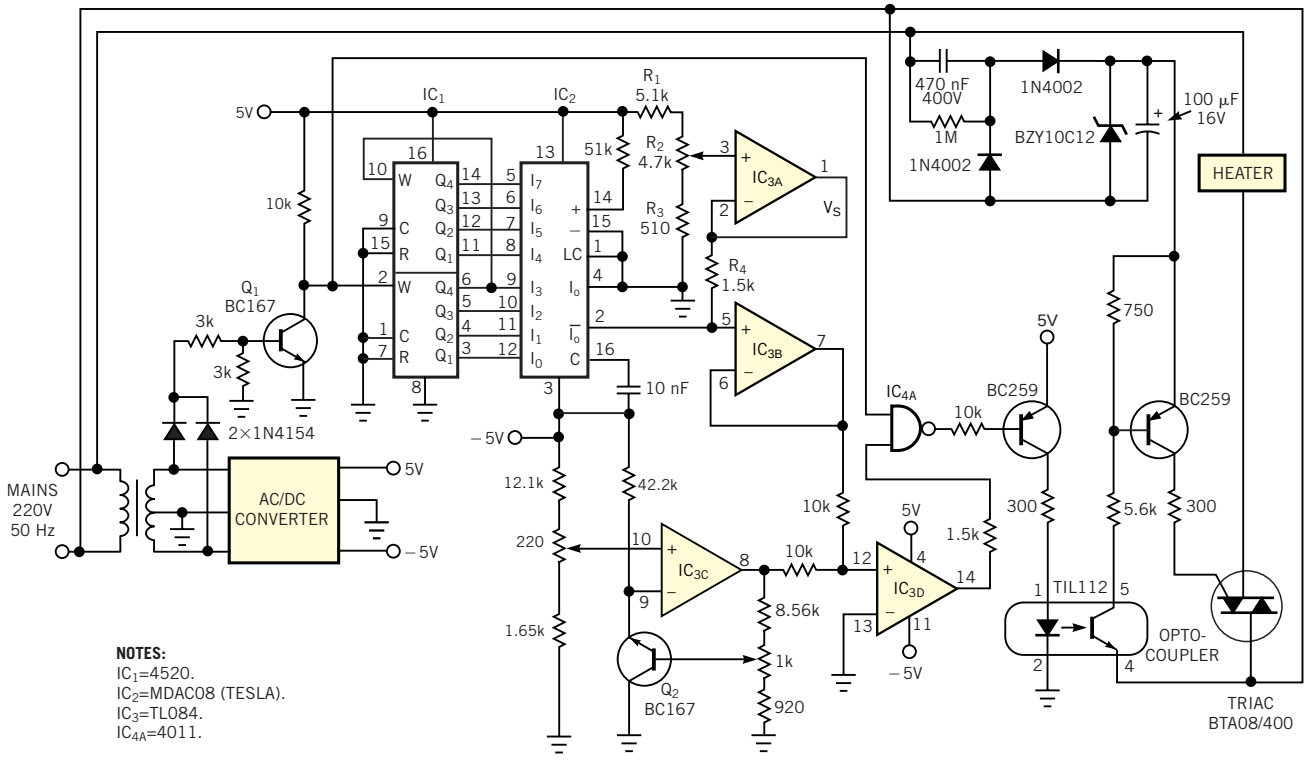
The circuit in **Figure 1** is a proportional thermoregulator in which the switching of the triac occurs when the mains voltage crosses the zero level. As a result, the circuit generates no RFI. The circuit can easily achieve linear regulation and perfect isolation between the control and power sections.

Whenever the mains voltage crosses zero, a 1-msec positive pulse appears at the collector of Q_1. From the resultant series of pulses, dual binary counter IC_1, DAC IC_2, and R_4 generate a sawtooth voltage with a 150-mV amplitude and a 2.56-sec period. Through R_4, the circuit subtracts this voltage from the constant setpoint voltage, V_S, that R_1, R_2, R_3, and follower IC_{3A} create. Simultaneously, sensing transistor Q_2 cooperates with IC_{3C} and associated resistors to create a voltage proportional to the object's current temperature with a sensitivity of 25 mV/°C.

If the difference between the desired and the real temperature of the object exceeds 6°C when the regulator is turned on for the first time, the output of the comparator IC_{3D} is continuously positive. The 4011 gate, IC_{4A}, passes all the triggering pulses from Q_1 through the TIL112 optocoupler to the triac's control electrode, and the heater delivers it full power. When the difference in temperatures becomes less than 6°C, the circuit starts working proportionally. In this condition, IC_{4A} passes only part of the triggering pulses generated within one period of the sawtooth voltage to the triac. The higher the object's temperature, the less time that the gate and the lower the average power delivered by the heater. When the real temperature reaches the setpoint temperature, the circuit always keeps the gate closed, and the heater is constantly off.

Thus, during the final stage of regulation, the temperature rises smoothly toward the desired value and is free from the overshoots and oscillations around the setpoint that on/off regulators exhibit. The resolution in the region of proportional regulation is as low as 0.4% of the nominal heater power. If necessary, you can change the width of this zone by simply changing R_4. (DI #2510)

Figure 1



Depending on the object's temperature, which Q_2 senses, IC_{4A} passes all, part, or none of the triggering pulses to the triac that controls the heater.

Spread-spectrum method identifies audio path

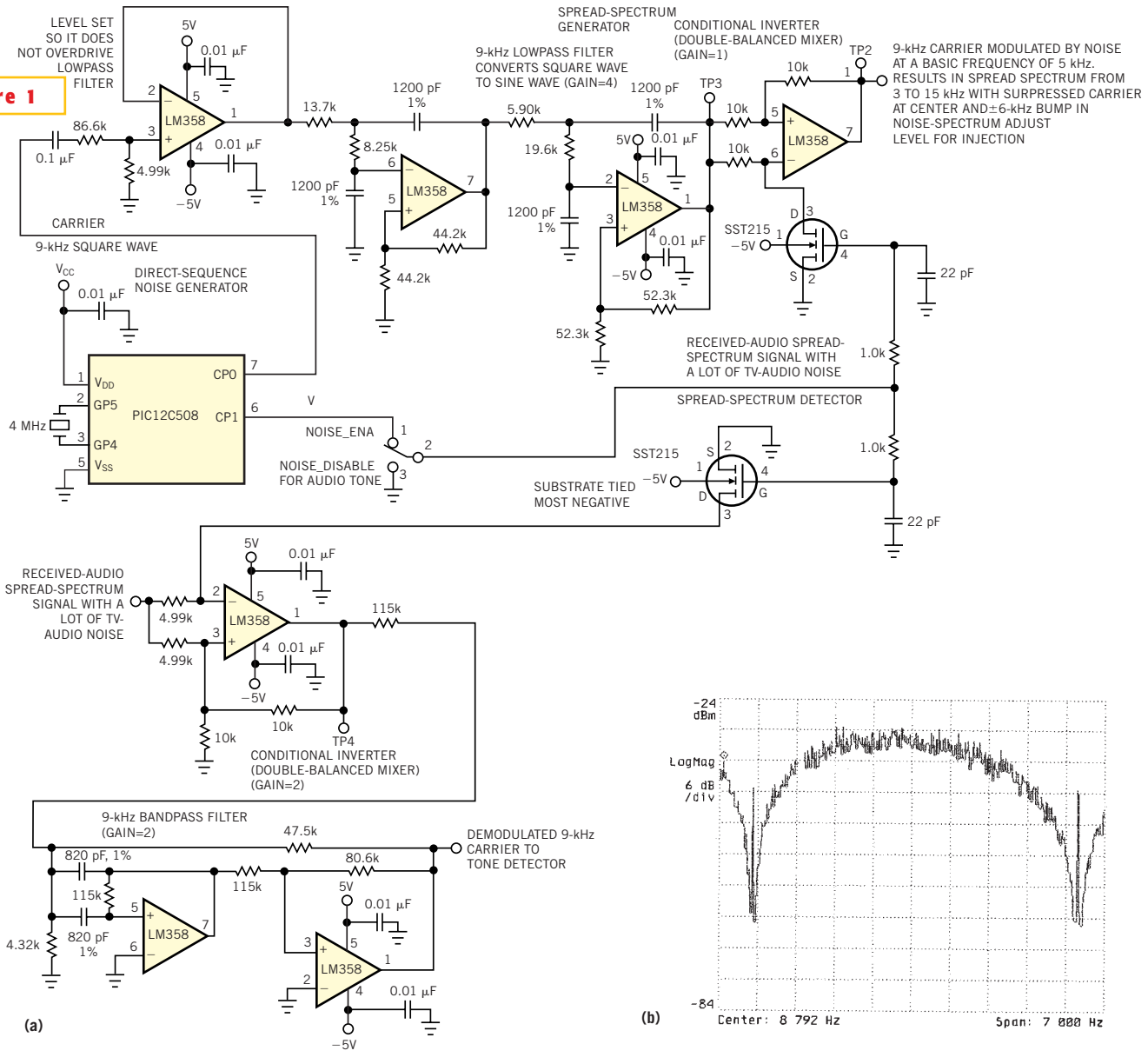
Bob Dougherty, Nielsen Media Research

AN UNUSUAL METHOD of audio spread spectrum can identify the audio path that's currently in use through a consumer-electronic device. The design in **Figure 1a** uses an injected audio spread-spectrum tag signal because a listener does not notice the low-level noise in the audio bandpass, whereas the listener would hear a steady-tone tag signal. The design injects the tag into the various inputs of the possible paths and detects the tag at the output, or speaker.

A direct-sequence noise generator running at 6 kHz furnishes the local oscillator, and a 9-kHz square-wave, lowpass filter that is filtered for its fundamental furnishes the sine-wave carrier. The circuit derives both of these signals from an 18-kHz clock. A PIC12C508A, running at 4 MHz, generates the local oscillator and the carrier. You can download the firmware from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2513.

The local oscillator appears as sinx/x noise in the frequency domain, which the circuit then mixes with a clean, 9-kHz audio sine wave in a double-balanced mix-

Figure 1



An audio-path identifier (a) injects an audio spread-spectrum tag signal (b).

er (DBM). Transformer and diode DBMs are unsuitable for the audio range. This circuit's DBM is simply an analog conditional complementor, and the local oscillator flips the output. The output of the DBM is a suppressed-carrier, double-sideband amplitude-modulated audio signal. The circuit bandpass-filters this output to 3 to 15 kHz to bandlimit the audio spread-spectrum signal (**Figure 1b**). Most of the audio "talk power" is lower than 3 kHz. The circuit then sets the level and buffers the output, which is the inject-tag signal. Note that, if the local-oscillator input to the DBMs is held either high or low, the DBMs become simple amplifiers and the tag signal is simply a 9-kHz tone.

Detection of a direct-sequence spread spectrum requires knowing that the signal, which sounds like noise, is present and that a synchronized-sequence local oscillator that's identical to the modulator side is present and requires a synchronized-sequence local oscillator that's identical to the modulator to demodulate the signal. In this case, the modulator and demodulator are in the same box, so the circuit knows when the signal is being generated, and it uses the same direct-sequence noise generator for both the injector and the detector. Thus, you eliminate the two biggest problems with spread spectrum.

The detector uses the same type of local oscillator and DBM, an analog conditional complementor, as does the injector. The detector then narrowly bandpass-filters the demodulated signal at the frequency of the modulated tag signal, or 9 kHz. This scheme separates the tag signal from the much louder audio. The audio has sporadic components in this narrow bandpass, so averaging is necessary to verify that the tag signal is present. (DI #2513)

To Vote For This Design,
Circle No. 316

5-to-1.8V converter works without magnetics

Dan Christman, Maxim Integrated Products, Sunnyvale, CA

To derive 1.8V from 5V, you might think of using a switch-mode regulator. Switchers are highly efficient but also complicated and expensive. Linear regulators, too, are out of the question unless your design can tolerate 40% efficiencies. The circuit in **Figure 1**, on the other hand, is more than 70% efficient (**Figure 2a**), sources as much as 100 mA, costs less than a switch-mode regulator, and requires less space.

IC_1 is a CMOS charge-pump voltage converter that the circuit configures as a voltage inverter. With its output grounded and 5V at its V+ pin, IC_1 generates V+/2, or approximately 2.5V at Pin 3. This nominal 2.5V output, which sags as the device sources current, drives linear-regulator IC_2, which regulates the 2.5V input to 1.8V. IC_2 is can source 100 mA before its sagging input voltage falls below the dropout level (**Figure 2b**). Using larger values for C_1 and C_2 enables IC_1 to maintain its output voltage with heavier load currents. (DI #2511)

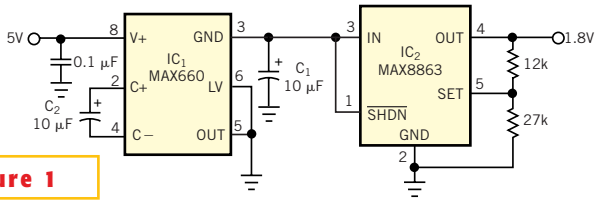
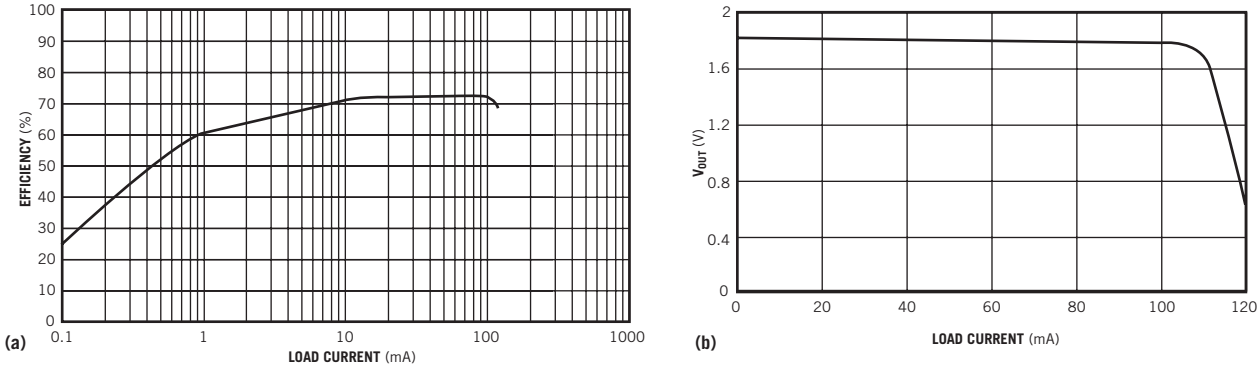


Figure 1

A 5-to-1.8V converter uses a charge-pump IC to lower the input voltage and then uses a linear regulator to achieve the desired output level.

To Vote For This Design,
Circle No. 317

Figure 2



The circuit provides maximum efficiency for load currents of 10 to 100 mA (a) and maintains its output level for load currents as high as 100 mA (b).

Band-reject filter includes compensation

Richard M Kurzrok, RMK Consultants, Queens Village, NY

Some band-reject LC filters employ a reactance to decouple individual resonators. One such filter uses relatively small shunt inductors to decouple series resonators, known as a top-L coupling. This type of coupling readily achieves narrow filter bandwidths while degrading amplitude response symmetry about the frequency of peak rejection.

Filter peak rejection occurs at a transmission pole that the shunt resonance of the coupling inductor and the effective capacitance (below resonance) of the series resonator create. A transmission zero occurs on the filter's high-frequency skirt due to series resonance. **Table 1** shows the amplitude response data of an uncompensated single-resonator band-reject filter with a center, or peak-rejection, frequency, of 14 MHz, a nominal 3-dB bandwidth of 1 MHz, and a 50Ω source/load impedance. The data in the **table** reveals appreciable asymmetry.

You can add compensation to the band-reject filter by connecting a grounded shunt capacitor to the midpoint of the coupling inductance, which this circuit realizes using two toroidal inductors in series (**Figure 1**). The measured 3-dB bandwidth of 0.9 MHz demonstrates good symmetry about the 14-MHz center frequency. Peak rejection of 21.4 dB corresponds to effective resonator unloaded Q of approximately 180. **Table 2** shows the measured amplitude response for the compensated one-pole band-reject filter.

The compensating capacitor and the bisected coupling inductor form a tee section. This section is a three-pole lowpass filter with 0.01-dB of passband ripple and a 3-dB cutoff frequency of 47 MHz, which provides supplementary lowpass selectivity superimposed upon the basic band-reject behavior. For multisection band-reject filters using artificial quarter-wave connecting lines, additional lowpass selectivity is available (**Reference 1**). Also note that the compensated top-L coupled band-reject filter is the LC counterpart of microwave band-reject filters. (**Reference 2**).

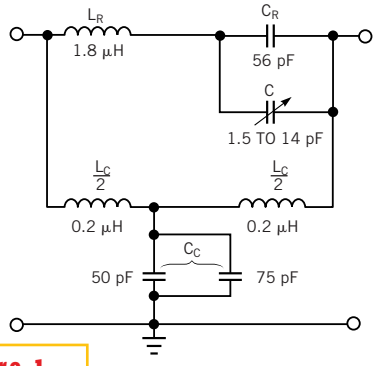


Figure 1

A compensated one-pole band-reject filter connects a grounded shunt capacitor to the midpoint of the coupling inductance.

You can avoid the need for compensation by tapping down on shunt resonators connected in series. This technique is useful for band-reject filters that use low-frequency pot-core inductors with many turns. The technique is unattractive for band-reject filters at higher frequencies using toroidal inductors with few turns. The compensated LC band-reject filter (**Figure 1**) is practical and provides substantial partial correction of filter-response-shape symmetry and can provide additional lowpass filtering. (DI #2514)

TABLE 1–UNCOMPENSATED AMPLITUDE RESPONSE

Frequency (MHz)	Insertion loss (dB)
11	0.6
11.5	0.7
12	0.9
12.5	1.2
12.8	1.6
13.1	2.2
13.4	4.0
13.7	6.3
14	21.8 (peak)
14.3	4.4
14.6	1.2
14.9	0.4
15.2	0.2
15.5	0.1
15.8	0.1
16.5	0.2
17	0.2
17.5	0.3
18	0.4
19	0.5
20	0.6
25	1.0

TABLE 2–COMPENSATED AMPLITUDE RESPONSE

Frequency (MHz)	Insertion loss (dB)
11	0.15
11.5	0.2
12	0.2
12.5	0.4
12.8	0.6
13.1	0.9
13.4	1.9
13.56	3.0
13.7	3.9
14	21.4 (peak)
14.3	5.6
14.46	3.0
14.6	2.1
14.9	1.1
15.2	0.75
15.5	0.5
15.8	0.3
16.5	0.2
17	0.2
18	0.15
19	0.15
20	0.1
25	0.25
30	0.55
35	1.3
40	3.1
45	5
50	7.2
60	14.2

References

1. Kurzrok, R, M, "Band reject filter provides supplementary low pass filtering", *RF Design*, October 1999, pg 54, plus errata, November 1999 , pg 16.
2. Kurzrok, R, M "Trimming improves response of waveguide band-reject filter", *Electronic Design*, Nov 8, 1967, pg 116.

To Vote For This Design, Circle No. 318

Low-cost active load draws constant battery power

Doug Farrar, Los Altos, CA

Unlike nickel-metal-hydride and lithium-ion rechargeable batteries, the discharge voltage of alkaline batteries is not constant and tends to vary from 1.5 to 0.8V per cell. Alkaline-battery manufacturers specify the discharge life of their batteries under constant *power* loads. Because of the varying discharge voltage, you can't simply connect a load resistor to the cells to verify or measure power profiles without incurring a lot of error in your measurements. The circuit in **Figure 1** draws a constant power from the battery pack. Thus, as the battery voltage decreases, the load current increases. A single potentiometer allows a user to set the desired wattage draw. The circuit is simple and inexpensive to build.

The heart of the load is an analog multiplier/divider circuit based on a cheap transistor array, IC_1. The five transistors have closely matching V_{BE}s, which is crucial to the accuracy of the circuit. The circuit does not use the one transistor, Q_{1E}, whose emitter ties to the chip's substrate.

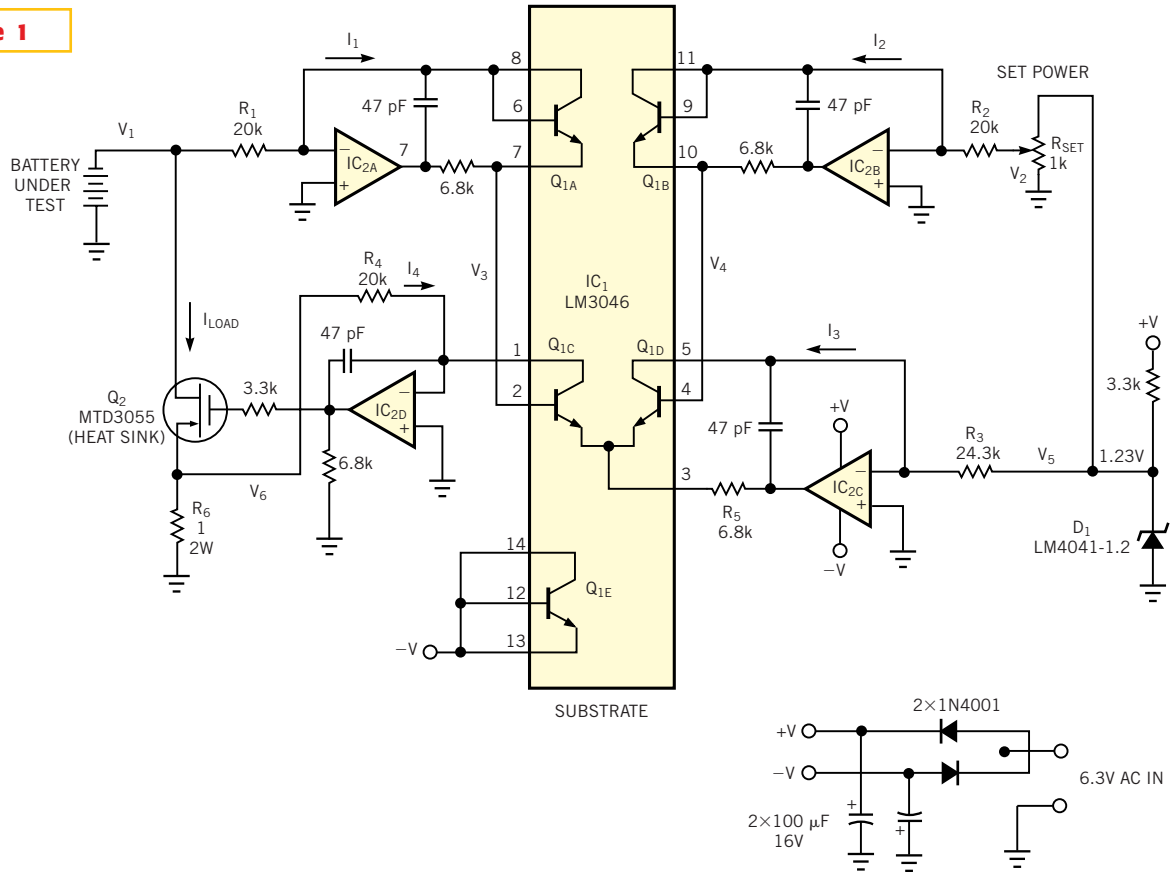
The circuit applies the battery voltage between V_1 and circuit ground. Op amp IC_{2A} forces current I_1 to flow through R_1; thus, $I_1 = V_{BATTERY}/R_1$. Likewise, current I_2 flows through R_2 and is equal to V_2/R_2. The setting of potentiometer R_{SET} determines I_2. Q_{1A} and Q_{1B}, which the circuit wires as diodes, are within the feedback paths of the op amps, so their currents equal I_1 and I_2, respectively. Because of the logarithmic nature of V_{BE}s, the voltage difference between the two diodes is as follows:

$$V_{Q1A} - V_{Q1B} = 0.026 \times \ln\left(\frac{I_2}{I_1}\right) = V_4 - V_3.$$

The circuit applies this voltage differential to the bases of Q_{1C} and Q_{1D}. Because all four transistors closely match one another, the ratio of currents I_3/I_4 equal the ratio I_2/I_1, or

$$0.026 \times \ln\left(\frac{I_2}{I_1}\right) = 0.026 \times \ln\left(\frac{I_3}{I_4}\right).$$

Figure 1



This active-load draws constant power; load current is inversely proportional to the applied voltage.

I_3, like I_1 and I_2, is set by an op-amp feedback network: $I_3 = V_5/R_3$. Because I_3 must flow through Q_{1D}, IC_{2C} will supply enough current to the differential pair Q_{1C}/Q_{1D} to force $I_4 = V_6/R_5$. IC_{2D} will then force its output voltage, which drives current sink Q_2, such that V_6 equals $I_4 \times R_4$. All you need to do now if finish the math:

$$\frac{I_2}{I_1} = \frac{V_1}{R_1} \div \frac{V_2}{R_2} = \frac{I_3}{I_4} = \frac{V_5}{R_3} \div \frac{V_6}{R_4},$$

or

$$V_6 = (V_5 \times V_2) \div V_1 \times \frac{R_1}{R_2} \times \frac{R_4}{R_3}.$$

Note that the circuit forces V_6 across the current-setting resistor, R_6, so that the current forced through transistor Q_2 is $I_{LOAD} = V_6/R_6$. Thus,

$$I_{LOAD} = (V_5 \times V_2) \div V_1 \times \left(\frac{R_1}{R_2}\right) \times \left(\frac{R_4}{R_3}\right) \div R_6.$$

The load current is inversely proportional to the applied battery voltage, which is precisely what you want. V_2 and V_5 are scaling voltages, allowing you to trim the current source value. In this case, the reference diode, D_1, sets V_5 at 1.225V, and a potentiometer sets V_2. The resistor ratios in **Figure 1** allow DMM measurements of V_2 to correspond to a power-dissipation level of 1V = 1W from a four-cell battery pack. You should try to keep I_1 through I_4 at 10 to 500 µA for best linearity. Using an op amp with a lower input offset voltage also helps improve accuracy.

R_1 draws a small amount of power from the battery as well, adding to the preceding equation. However, this error term is small, and you can ignore it. The current-source bias voltage, namely the load current times the sum of R_6 and Q_2's on-resistance, or R_{FET}, limits the minimum battery voltage. Doing the math, you'll find that $V_{IN(MIN)} = \sqrt{V_2 \times (R_6 + R_{FET})}$, where V_2 is the desired power draw.

A small power supply that outputs approximately 6.3V ac powers the circuit. Two 1N4001 diodes half-wave-rectify the positive and negative voltages, and two 100-µF capacitors provide filtering. The extremely light load of the circuit means that postfilter regulation is unnecessary. If portability is necessary, you could power the circuit from a pair of series-wired 9V batteries. You can also use asymmetrical power supplies, but you need a positive voltage of around 8V to allow Q_2 to turn fully on. If you substitute a different op amp, make sure that it can withstand the power-supply voltages. Q_2 dissipates most of the power, so it needs an attached heat sink. (DI #2512)

To Vote For This Design, Circle No. 319

design ideas

Edited by Bill Travis and Anne Watson Swager

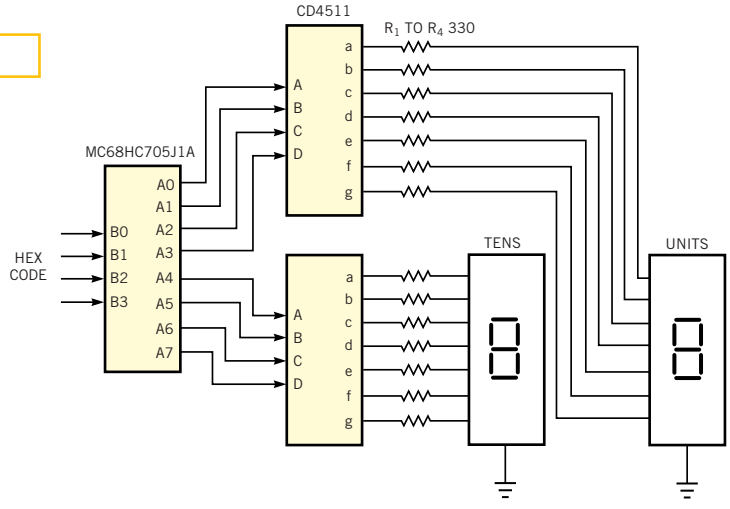
µC visualizes hex code

Abel Raynus, Armatron International, Melrose, MA

In µC systems, information exchanges usually use the hexadecimal 1-2-4-8 format; output data also appear in this format. Reading the hex code is not a problem; several LEDs connected to the output lines can display the answer. The problem arises when you wish to observe the output data. Many engineers are unfamiliar with hex code and prefer to observe data in the common decimal format. If the value of the output data is less than 10, you can use an ordinary BCD-to-seven-segment decoder to visualize the hex code on an LCD or an LED display. But what do you do if the value of the output data is greater than 10? Unfortunately, no decoders can transform hex code into two seven-segment codes. Of course, you could configure such a decoder using a number of logic gates, but another simple and inexpensive option is available. The key to this option is using a low-end µC to transform the hex code into two BCD codes. In **Figure 1**, the data displayed ranges from 0 to 15. Thus, you need only 4-bit hex code, using four input and eight output µC lines.

Figure 1 uses the approximately $1 Motorola MC68HC705J1A, with 14 I/O pins. The input lines Pin B0 to Pin B3 re-

Figure 1



You can visualize hex code in BCD format by using this simple circuit.

LISTING 1—HEX-TO-BCD CONVERSION

```
                          1  * HEX CODE TO 2 BCD CODE TRANSFER*
 0000                     2  $pagewidth 160
 0000                     3  $include "std-jla.asm" ;standard frame for
MC68HC705J1A
                          4  *VARIABLES
 00C0                     5          org     RAM
 00C0                     6  M1      rmb     1
                          7  *INITIALIZATION
 07F1                     8          org     MOR;
 07F1       20            9          fcb     %00100000;resistor osc
 0300                    10          org     ROM
 0300 [02] A600          11  init    lda     #%00 ; prtB as input
 0302 [04] B705          12          sta     ddrB
 0304 [02] A6FF          13          lda     #$ff ; prtA as output
 0306 [04] B704          14          sta     ddrA
 0308 [05] 3F00          15          clr     prtA ; 0 -> prtA
                         16  ************************************************
 030A [03] B601          17  main    lda     prtB
 030C [02] A40F          18          and     #%00001111 ; extract data from pB0-pB3
 030E [04] B7C0          19          sta     M1
 0310 [02] A109          20          cmp     #$9         ; data > 9?
 0312 [03] 220A          21          bhi     h1
 0314 [02] A600          22          lda     #$00        ;set 00 to pA4-pA7 (tens)
 0316 [04] B700          23  h2      sta     prtA
 0318 [03] BAC0          24          ora     M1
 031A [04] B700          25          sta     prtA        ;set data to pA0-pA3
(units)
 031C [03] 20EC          26          bra     main
 031E [02] A00A          27  h1      sub     #$a         ;(data-10) -> Acc
 0320 [04] B7C0          28          sta     M1
 0322 [02] A610          29          lda     #$10        ;set 1 to pA4-pA7 (tens)
 0324 [03] 20F0          30          bra     h2
                         31
*************************************************************
 07FE                    32          org     VECTORS+6
 07FE       0300         33          fdb     init
```

ceive the 4-bit hex code to be displayed. The µC-assembly program in **Listing 1**, converts the hex code into two BCD codes, which appear on lines Pin A0 to Pin A3 (units) and Pin A4 to Pin A7 (tens). These outputs drive the two standard BCD-to-seven-segment decoders, which, in turn, drive the common-cathode LED displays. You can use the same method for an expanded data range, but you need more I/O lines, decoders, and displays. For example, the 8-bit hex code covers the data range 0 to 255, but it needs eight input lines, 12 output lines, and three decoders and displays. You can download **Listing 1** and the "include" file in line 3 from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2518. (DI #2518)

To Vote For This Design, Circle No. 301

Simple circuit times bathroom fan

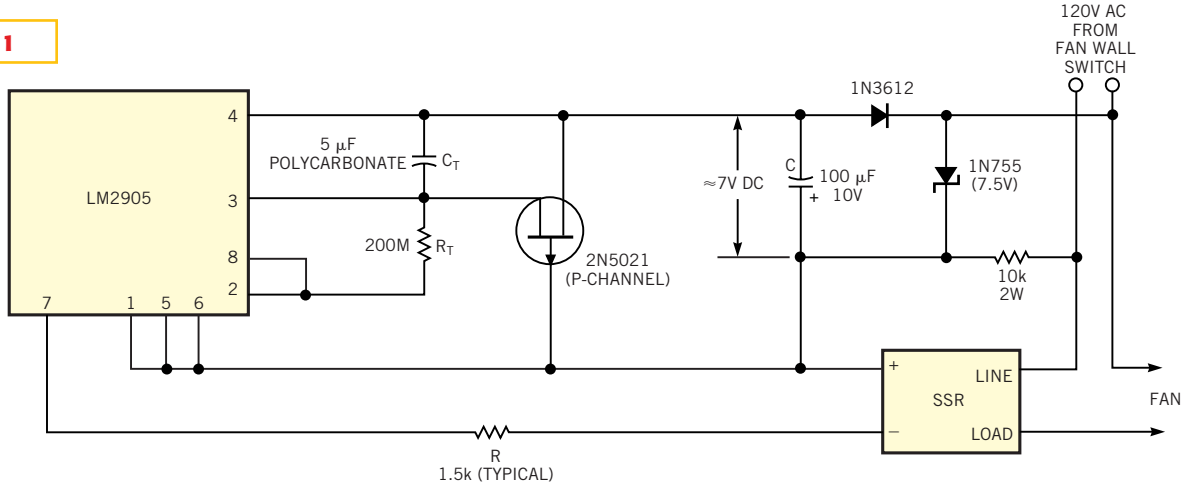
Maxwell Strange, Goddard Space Flight Center, Fulton, MD

Forget to turn off the ceiling fan in your bathroom? Install the simple timer in **Figure 1**. It's located out-of-sight in the fan unit, and you turn it on via the wall switch. The circuit costs virtually nothing, using "junk" parts. When ac power appears, a simple rectifier develops approximately 7V across filter capacitor C. This voltage powers the LM2905 analog timer and simultaneously triggers it via trigger-input Pin 1; output Pin 7 then goes low, turning on the solid-state relay and fan. Low-leakage capacitor C_T and resistor R_T set the time delay; for the values shown, the delay is 1000 sec, or approximately 17 minutes. At time-out, Pin 7 goes high, turning off the relay. For most fans, this relay can be a small, pc-mountable unit rated at as little as 0.5A ac load current. These relays typically turn on reliably at 3V. You can use resistor R to drop excess voltage, thereby reducing loading on the power supply.

The p-channel JFET turns on when the power switches off, rapidly discharging C_T to allow immediate recycling. The 10-kΩ series resistor makes the circuit inherently safe; in a worst-case failure, line voltage appears across this resistor and develops a harmless 1.4W. You can easily adapt this circuit to other applications. You can change the timing or make it linearly adjustable, and you can program the solid-state relay to turn on instead of off at time-out, by connecting Pin 8 of the timer to Pin 4 instead of Pin 2. You can also reverse the action of the solid-state relay by reversing the control inputs. (DI #2520)

To Vote For This Design, Circle No. 302

Figure 1



Gain control of your bathroom fan by using this simple time-out circuit.

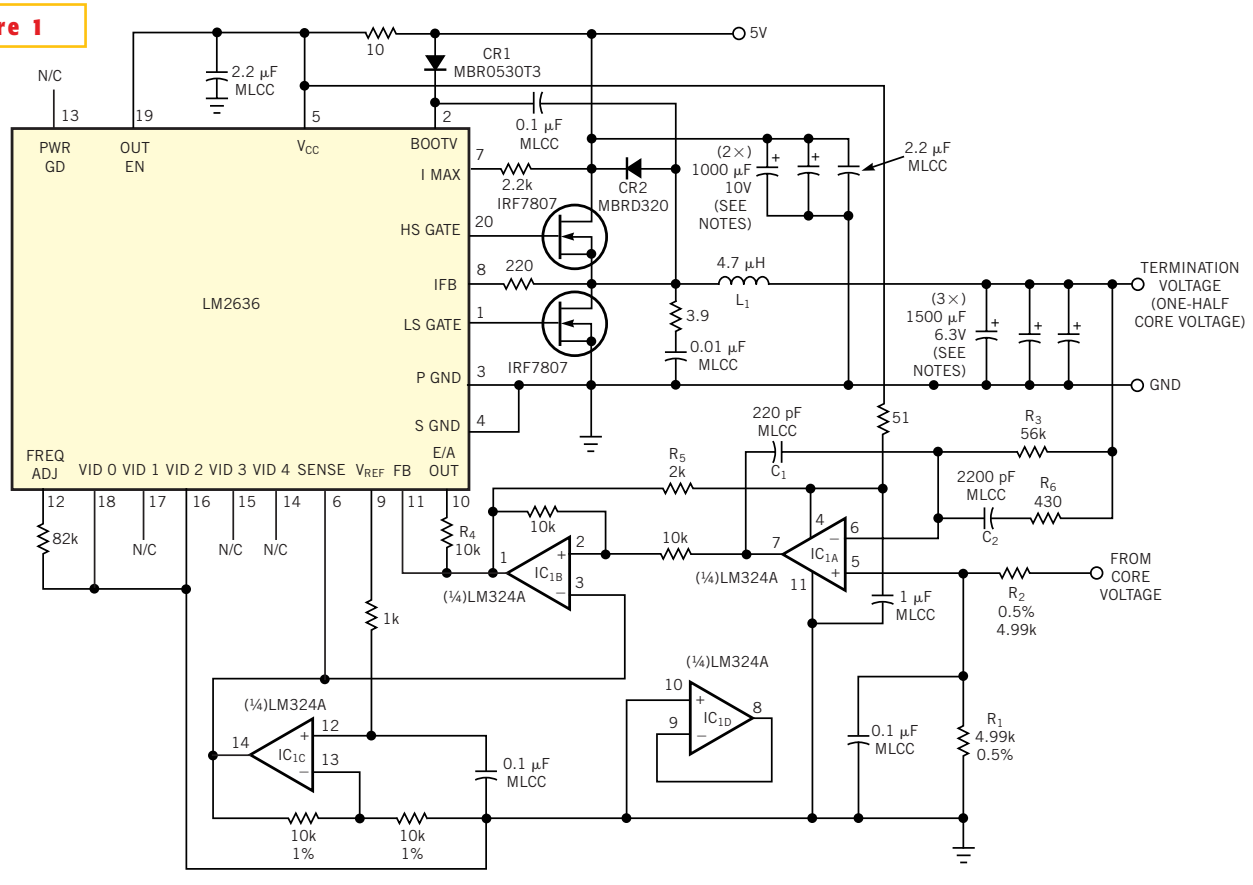
Termination supply tracks one-half core voltage

Chester Simpson, National Semiconductor, Santa Clara, CA

Modern µPs typically require core voltages of 2 to 3V. They also require "termination" voltages that, for best performance, should equal half the core voltage. The problem is, core voltages vary among CPUs, and some systems even incorporate variable-core-voltage supplies that allow systems to adjust the voltage on the fly, thereby increasing or reducing CPU operating speed to optimize power consumption. A termination-voltage power supply that maintains an output of one-half V_{CORE} over a range of core voltages is highly desirable, as the core voltage can vary without upsetting the termination-voltage set point. The circuit in **Figure 1** is a 6A power-supply design that generates a termination voltage regulated to 1/2 V_{CORE}. The circuit targets applications in which the core voltage is approximately 1.8 to 3.6V.

An LM2636 synchronous-rectifier controller switching at 300 kHz provides an efficient power converter that operates from a 5V input. Because the LM2636 is designed to operate at a fixed output voltage (as determined by control bits 14 through 18), the circuit in **Figure 1** uses a different control scheme to force the regulated output to track at one-half V_{CORE}. Resistors R_1 and R_2 halve the core voltage, and this voltage serves as the reference in error amplifier IC_{1A}. The amplifier compares the one-half V_{CORE} reference with the termination output voltage obtained through R_3 and adjusts its output to lock the termination voltage at one-half V_{CORE}. In this way, the core-voltage signal sets the termination voltage. IC_{1B} is a unity-gain inverter that corrects the phase of the feedback signal that goes to the input of the LM2636's inter-

Figure 1



NOTES:
USE LOW-ESR ALUMINUM ELECTROLYTIC CAPACITOR SUITABLE FOR 300-kHz SWITCHING APPLICATIONS.
UNLESS OTHERWISE SHOWN, RESISTOR TOLERANCES ARE 5%.
MLCC DESIGNATES A CERAMIC CAPACITOR WITH AN X7R OR X5R TEMPERATURE RANGE.

Optimize your CPU's performance by feeding it a termination voltage that's exactly half the core voltage.

nal error amplifier. R_4 sets the gain of the internal amplifier to unity. R_5 forces a soft start on turn-on and also eliminates overshoot. IC_{1C} amplifies the 1.23V internal reference to approximately 2.5V, which sets the operating point for the error amplifier, IC_{1A}.

The 2.5V also connects to the sense input of the LM2636, which would normally sense the regulated output voltage. Because the termination output voltage must be variable (to track the core voltage), a fixed 2.5V goes to the sense pin, and the control pins 16 and 18 are grounded. These connections program the internal DAC for a 2.5V output. This scheme prevents the LM2636's internal error-detection circuitry from shutting down the part in response to an undervoltage or overvoltage condition. R_3, R_6, C_1, and C_2 provide loop compensation. (DI #2517)

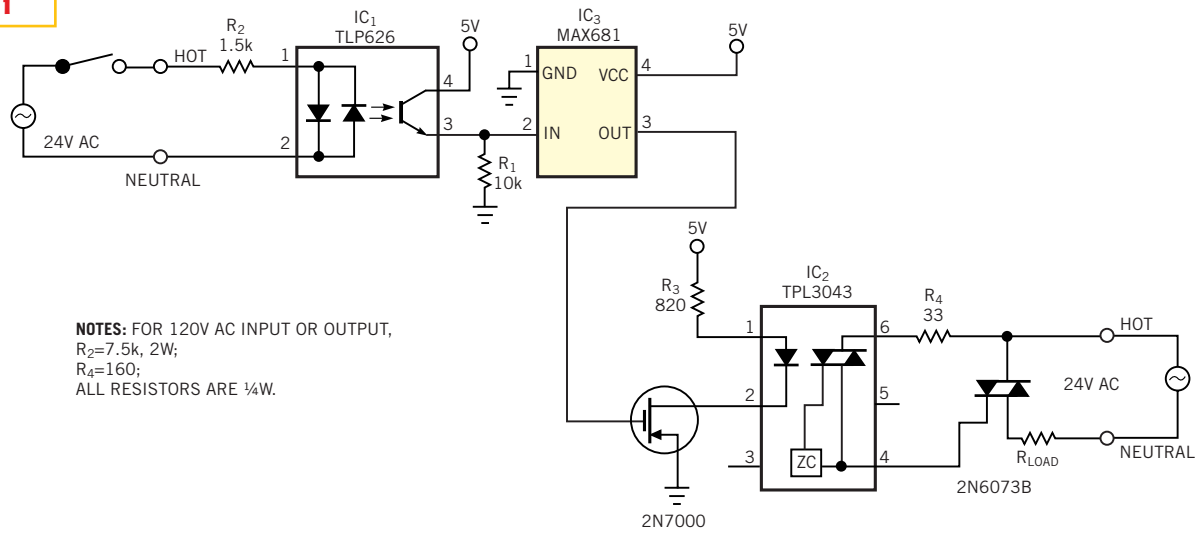
Switch debouncer isolates input and output

Phill Leyva, Maxim Integrated Products, Sunnyvale, CA

Modern programmable-logic controllers (PLCs) for automated process-control systems have either 16 or 32 inputs and accept ac voltages of 24 to 120V. A single circuit (a relay for isolation and an RC network with a Schmitt trigger to debounce signals to the processor) can debounce all the PLC signals in sequence. However, this approach slows real-time data processing. Such debounce circuits also produce delay times that change with relay wear and capacitor aging. In the PLC program, you can use a debounce timer for each input, but this technique increases the program-scan time and ties up valuable timers. The solid-state, electrically isolated circuit in **Figure 1** debounces single inputs without slowing the PLC module. Optoisolators IC_1 and IC_2 provide electrical isolation for the ac sources at the input and output. IC_3 is a CMOS switch debouncer whose output (a 4V logic high) appears following a fixed 40-msec delay. A 63-kΩ pullup resistor, connected internally between IN and V_{CC}, forms a voltage divider with R_1. R_1's value ensures a logic low of less than 0.8V at IN when IC_1's optotransistor (an emitter follower) is off.

The two LEDs in IC_1, which illuminate the phototransistor on alternate half cycles of the ac-input current, rectify this current. Most optoisolator applications set the current-transfer ratio (CTR) to more than 10 to ensure an accurate reproduction of the input signal. The circuit in **Figure 1**, however, sets the CTR to less than 1, which ensures that the emitter follower does not turn off as the ac current goes to zero twice in each cycle. R_2 biases the emitter follower such

Figure 1

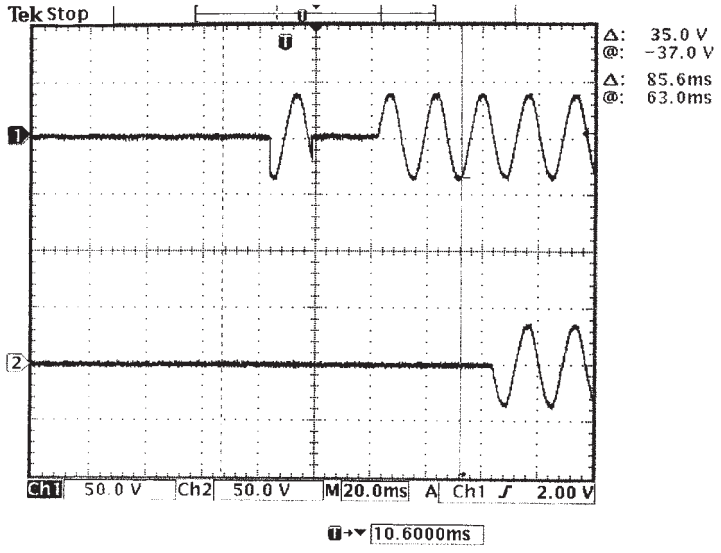


This debouncer circuit allows an isolated ac voltage to control a separately isolated ac source.

that IC_3's IN signal remains greater than the high level, 2.4V, during these zero crossings. This action eliminates the capacitor normally found in debouncing circuits. IC_3's OUT pin drives the n-channel MOSFET, and R_3's value is such that the resulting current flow (in the MOSFET and LED) is approximately 5 mA. When the MOSFET turns on, the LED activates IC_2's zero-crossing triac driver. Thus, when the power triac turns on, an ac source connected to the output drives R_{LOAD} with as much as 4A (**Figure 2**). Turning on the triac at zero crossings eliminates EMI and reduces the turn-on stress in the triac. R_4 limits current into the triac driver (IC_2) to 1A. Each source can be either 24V ac (as shown) or 120V ac. (DI #2521)

To Vote For This Design, Circle No. 304

Figure 2



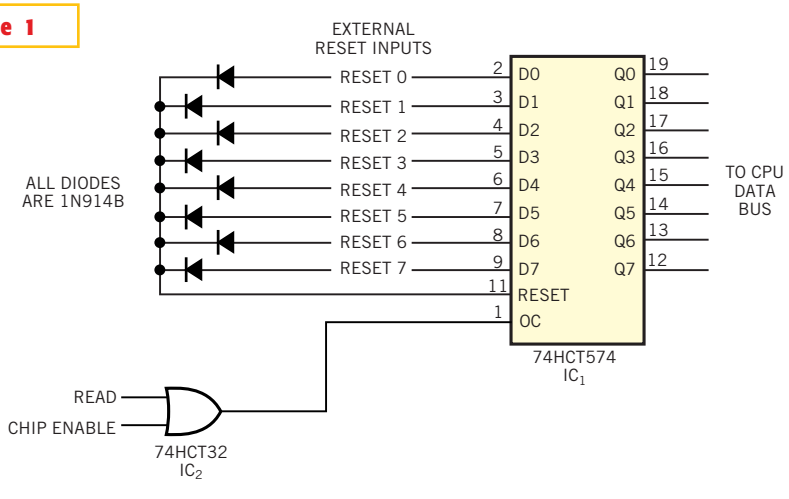
An ac switch (top trace) turns on and then briefly bounces off; the output of the associated debouncer circuit turns on cleanly (bottom trace).

Circuit detects reset source

Shyam Tiwari, Sensors Technology Pvt Ltd, Gwalior, India

Embedded single-board computers do not rely only on power-on resets; they often use multiple sources to reset CPUs to prevent CPUs from being locked into endless loops. However, in most cases, CPUs start from the starting points of memories to fetch their first code, because they have no way of knowing what generated the reset. The circuit in **Figure 1** allows a CPU to know the source of a reset. It stores in an 8-bit latch reset-control input data that the CPU can read. If an input frequently resets the CPU, the CPU can then report the error source to the user, using LEDs or other indicators. In the circuit of **Figure 1**, an experiment combined eight independent reset-signal sources into a wired-OR single output. The 74HCT574 stores the 8-bit reset data at the rising edge of the signal. The latch records no other signal if the signal appears after the rising edge of the first signal. If the CPU finds data that resembles a reset signal in the latch that the signal does not reset, then the reset pulse is too narrow to effect a reset. The CPU recognizes this signal after reading the information from the latch. The output of the latch is a tristate structure; the CPU reads the output using active chip-enable and read-input signals. (DI #2516)

Figure 1



This circuit detects the source of a CPU-reset signal.

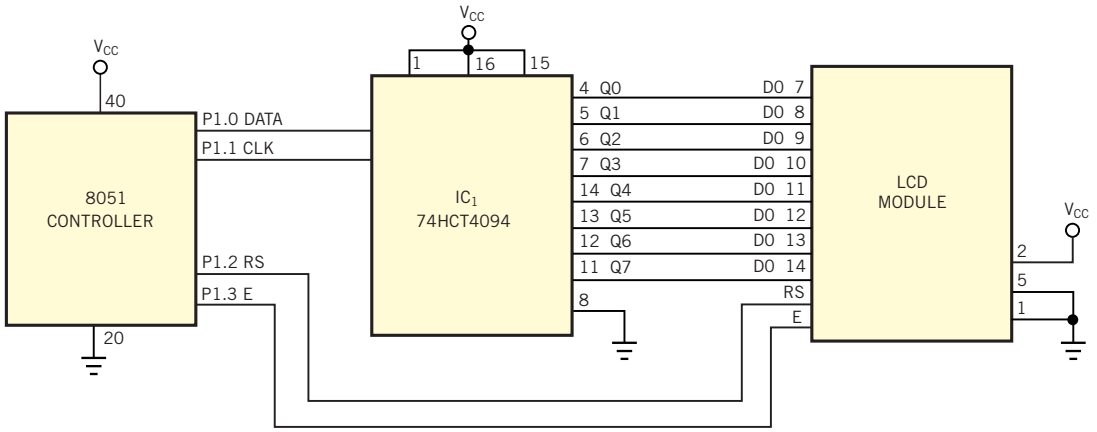
To Vote For This Design, Circle No. 305

Interface LCD with ease

Bharat Mehta, Space Applications Center, Ahmedabad, India

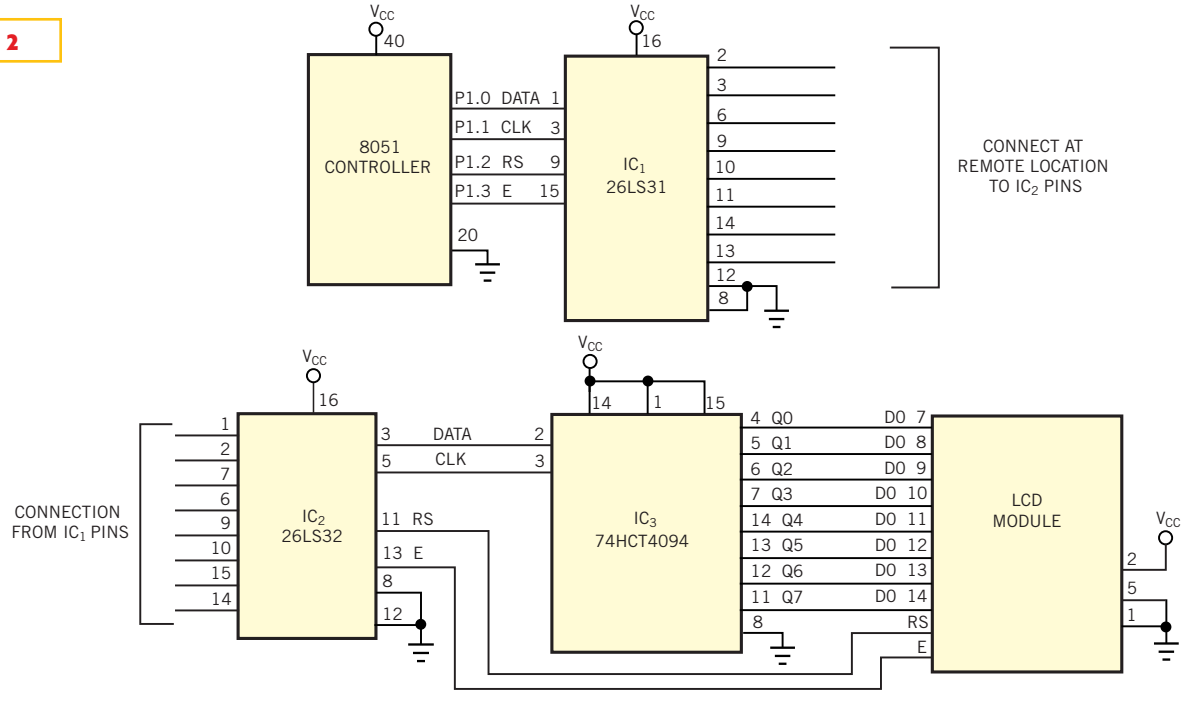
Alphanumeric LCDs find use in various embedded systems. In these systems, µCs often lack I/O resources, so you must make optimum use of the available resources. LCDs have a bus-oriented interfacing structure, which consumes 11 I/O lines (eight data lines and three control lines). You cannot always afford to spare this many I/O lines. Most applications need to write the data to the LCD and then read from it. In these situations, you can omit one control pin (of three) for reading data and save one additional I/O-port pin. The circuit in **Figure 1** saves I/O pins. It uses an 8051

Figure 1



Save µC I/O pins by effecting a serial-to-parallel conversion with a shift register.

Figure 2



Extend Figure 1's scheme to long distances by using line-driver and -receiver ICs.

µC, and only four I/O pins (P1.0 to P1.3) drive the LCD, instead of the 11 pins other displays require. Data transfers in a serial mode through the P1.0 port of the µC. The µC interfaces to Pin 2 of IC_1, a 74HCT4094 8-bit shift register. Serial data advances on every clock pulse and transfers to the shift register. The register converts the serial data to parallel data, available on the output pins Q0 to Q7. The P1.1 port of the µC provides the clock.

The data bus, D0 to D7, of the LCD module connects to the shift register's outputs. Software carries out the data transformation and displays the result on the LCD (**Listing 1**). The same design can drive various types of alphanumeric LCDs—for example, single-line-to-multiline types, with different character lengths on each line. You can configure the circuit to send data over a long distance for remote LCD readouts (**Figure 2**). You can transmit serial data through a differential line-driver IC such as the 26LS31 (IC_1). You can feed the output of IC_1 to either twisted-pair wires or a parallel pair of wires for transmission to a remote location. At the other end of the remote location, you can retrieve the data through a differential line receiver such as the 26LS32 (IC_2). The output of IC_2 drives IC_3, which drives the LCD as in **Figure 1**.

Listing 1, written in 8051 assembly code, provides the sample text "Hello, EDN Reader" on a single-line, 16-character LCD. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2519. (DI #2519)

To Vote For This Design,
Circle No. 306

LISTING 1—SAMPLE DATA DISPLAY

```
temp1:  reg     20h
Dat:    reg     P1.0
Clk:    reg     P1.1
RS:     reg     P1.2
ENA:    reg     P1.3

;;;;;;;;;;;;;;;;  Main progarm   ;;;;;;;;;;;;;;;

        org     0000h
        jmp     start

        org     30h

start:  call    INT
        mov     dptr,#title
        mov     r0,#10h         ;counter for  data transfer

loop1:  clr     a
        movc    a,@a+dptr
        mov     temp1,a
        djnz    r0,loop2
        call    dly
        jmp     start

loop2:  call    sr_tx
        setb    RS      ;This part transfer data to LCD.
        setb    ENA
        call    delay   ;small delay
        clr     ENA
        clr     RS
        inc     dptr
        jmp     loop1

title:  db "Hello,EDN Reader"

;;;;;;;;;;;;;;;;  Main program End  ;;;;;;;;;;;

;------  Sub-Routine  programs  --------
;;; initialization of LCD screen.

INT:    mov     sp,#60h
        clr     RS
        clr     ENA
        mov     temp1,#38h
        call   disp2
        mov     temp1,#01h
        call    disp2
        mov     temp1,#02h
        call    disp2
        mov     temp1,#06h
        call    disp2
        ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;

disp2:  call    sr_tx


        setb    ENA
        mov     r3,#0ffh
        djnz    r3,$
        clr     ENA

        ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; program to transfer data serially from I/O

sr_tx:  mov    r1,#08h          ;counter
s1:     mov     a,temp1
        clr     c
        rlc     a
        mov     temp1,a
        mov     Dat,c
        setb    Clk
        nop
        nop
        clr     Clk
        djnz    r1,s1

        ret
;;;;;;;; Delay Sub-Routine   ;;;;;;;;;;;;;;;
; this delay program is a flexible loop , any time delay can
; be adjusted by changing the value of registers

dly:    mov     r5,#0fh
d1:     acall   mdly
        djnz    r5,#d1
        ret

mdly:   mov     r6,#0ffh
mdly1:  call    delay
        djnz    r6,mdly1
        ret

delay:  mov     r7,#0ffh
        djnz    r7,$

  ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

end
```

Edited by Bill Travis and Anne Watson Swager

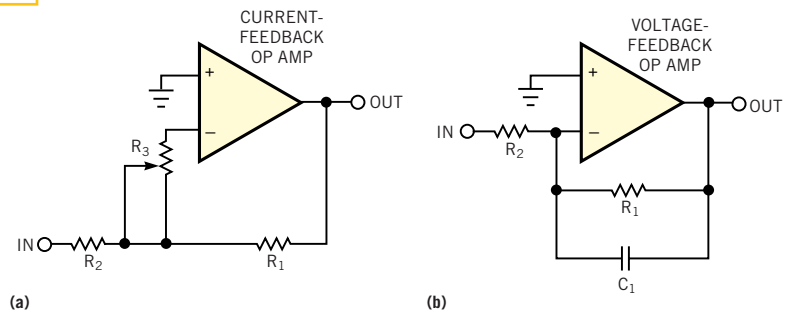
Potentiometer tames current-feedback op amp

Leslie Green, Gould-Nicolet Technologies, Ilford, Essex, UK

THE CIRCUIT IN **Figure 1a** includes a potentiometer in series with the inverting input of a current-feedback op amp. With a voltage-feedback op amp, a potentiometer in series with the inverting input would serve no useful function. However, with a current-feedback amplifier, the potentiometer controls the bandwidth without changing the gain. The function of this potentiometer is similar to a capacitor, C_1, in parallel with the feedback resistor of a voltage-feedback op amp (**Figure 1b**).

In **Figure 1a**, feedback resistor R_1 strongly governs the bandwidth and pulse-response characteristics of the current-feedback amplifier. However, changing R_1 also changes the gain. Thus, you have to change both R_1 and R_2 to adjust the pulse response while keeping the gain constant, which makes the design considerably difficult to adjust. The inclusion of the potentiometer allows you to adjust the pulse response independently of the gain, which makes the potentiometer a useful control.

Figure 1



A potentiometer, R_3, allows you to adjust the bandwidth of a current-feedback op amp without changing the gain (a). R_3 performs the same bandwidth-control function as C_1 does for a voltage-feedback op amp (b).

Current-feedback amplifiers are available with closed-loop bandwidths much greater than 500 MHz, and they oscillate or give poor pulse responses if you wire them with 2-in. leads on all components and fail to use proper decoupling. You need skill to properly lay out these devices. However, using normal UHF circuit skills and the trick in **Figure 1a**, you can easily control these devices.

A typical value for the feedback resistor when using a current-feedback amplifier is approximately 150 to 900Ω. The nominal value of the potentiometer, R_3, is 50 to 500Ω.

If you look at the manufacturers' specified time-domain, or pulse, response of current-feedback amplifiers, you may feel that the response is not flat enough for your application. Some data sheets show poor figures for overshoot and ringing. However, using **Figure 1a**'s circuit, you can tune the response of the amplifier within a system to give acceptable performance. You can also put a capacitor across R_3 to modify the response of the amplifier. These modifications are sometimes necessary to get the best response from an amplifier or a system. It is also not unusual to have to correct for an error that occurs in another stage.

The main signal path of an oscilloscope is a demanding application for an op amp due to the requirement for high bandwidth and excellent pulse response. However, the scheme in **Figure 1a** was useful in the design of the main ADC buffer amplifier of the 150-MHz Gould DataSYS 840 DSO. The design uses an HFA1130 current-feedback op amp and an R_3 value of 100Ω to adjust the overall bandwidth of the oscilloscope. (DI #2524)

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 343

Cable finder seeks out wall conduits

Luis Miguel Brugarolas and Manuel Ramón,
Sire Sistemas y Redes Telemáticas, Tres Cantos, Madrid, Spain

When you debug home electrical wiring, a tool that determines the location of cable conduits comes in handy, especially when problems arise. Some tools, such as nylon or glass fiber guides, contain small magnets and compasses that act as sensor elements. These short-range systems are useful for brick walls, but not for other wall-mounting systems that include iron elements. Some electronic aid would be helpful.

The circuit in **Figure 1** uses an antenna in the wall conduit and a receiver to locate a cable conduit. The combination can accurately locate the antenna placement. The simplest and cheapest approach is to use a portable AM radio, which includes a ferrite antenna with excellent directivity and sufficient range of a few centimeters for good resolution.

The circuit uses a relaxation oscillator with an adjustable frequency to apply current pulses to a twisted-pair cable. This oscillator has the AM-broadcast range of 530 to 1600 kHz. A lower frequency oscillator switches these pulses on and off. The envelope detectors of AM radios can easily detect the pulses. However, the circuit requires an additional refinement because pure tones are unpleasant. Thus, some modulation is necessary. This circuit allows for dual tones, such as a police siren, or sweeping tones, such as an ambulance siren, using a switch. The circuit provides dual-tone operation using a bipolar transistor that shorts a capacitor on and off in the capacitive section of the oscillator path. The sweeping tone is a mixed AM/FM format that the circuit produces by biasing the relaxation oscillator with a triangular signal from a lower frequency oscillator. This bias changes the threshold level of the modulating tone, which produces modulation of the fundamental frequency and the aspect ratio of the output.

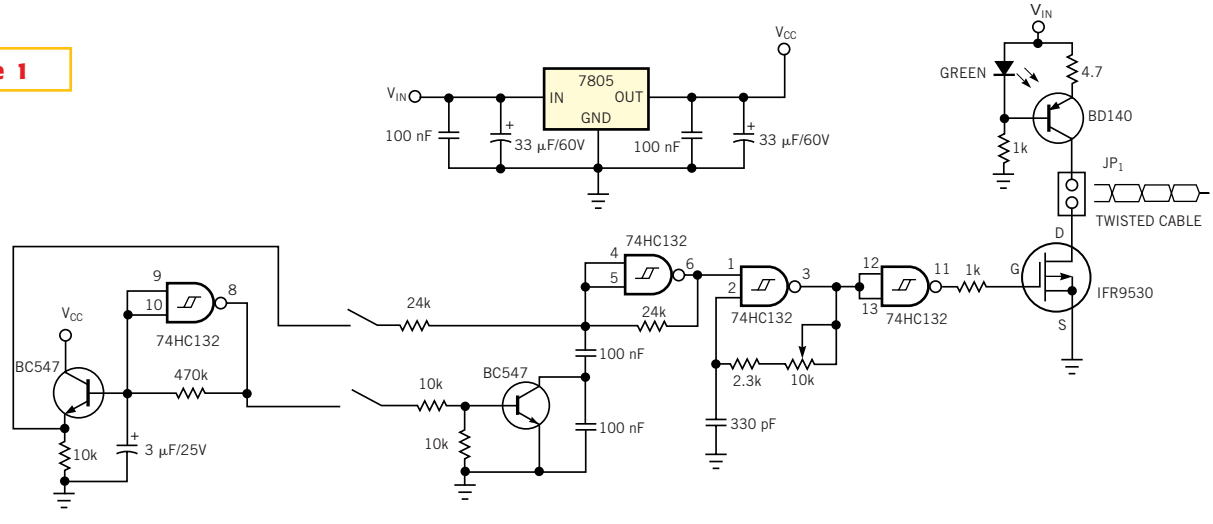
A regulated power source biases the oscillator section to minimize the high-frequency oscillator's voltage sensitivity. A current source, which is insensitive to supply voltage, controls the final stage. This control, instead of a direct connection to the supply voltage, provides some protection in the event of a short circuit and provides similar output independently of the antenna resistance. The LED, which the circuit uses as a voltage reference, also indicates circuit activity. The supply voltage can vary from 7 to 12V or higher if you take care of the BD140 transistor's dissipation requirements.

To use the circuit, switch on an AM receiver and find a position without broadcast stations. Place the receiver close to the generator, switch on the circuit, and adjust the variable resistor until the tone is audible. Move the receiver to find the direction of better sensitivity. Then, use the radio receiver to find where the twisted-pair cable goes.

Beware that this circuit produces EMI not only in the AM-broadcast range, but also over a wide frequency range due to the high harmonic content of the generated signals. The use of twisted-pair cable reduces the amount of interference in the far-field region. Therefore, use the circuit cautiously, and disconnect the circuit as soon as you locate the cable. (DI #2523)

To Vote For This Design,
Circle No. 344

Figure 1



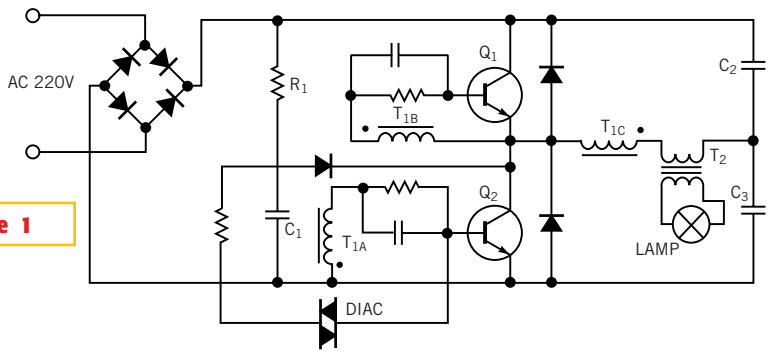
To discover the location of cable conduits, a relaxation oscillator applies current pulses to a twisted-pair cable, and an AM radio functions as a receiver.

Electronic transformer dims halogen lamp

Robert Magdziak, Warsaw, Poland

A typical, switched-mode power supply for a halogen lamp, commonly known as an electronic transformer, is a clever and simple device. However, it has some limitations, such as its inability to dim the lamp. **Figure 1** shows a simplified schematic of a typical low-voltage halogen-lamp transformer without the protection circuits and EMI filter. Q_1 and Q_2 with C_2 and C_3 comprise a classic, half-bridge topology that works in self-oscillating mode. The circuit provides positive feedback by placing the primary windings of transformer T_1 in series with the bridge output. To achieve a high power-factor value, a rectified but unfiltered mains voltage supplies power to the circuit. The working frequency is approximately 30 to 40 kHz.

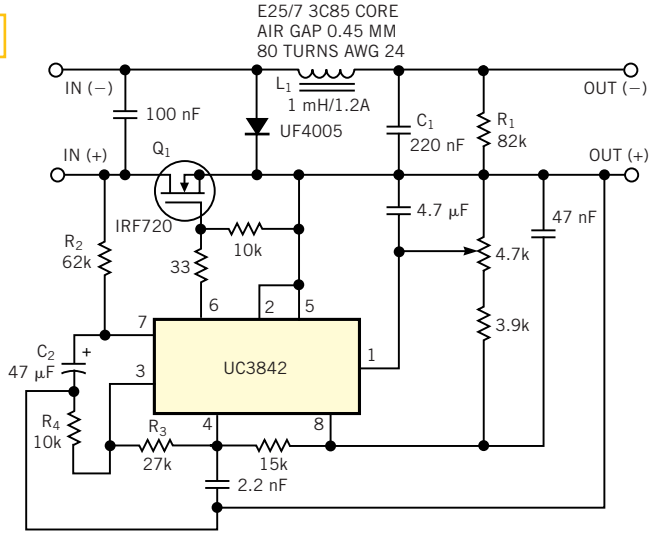
Figure 1



A typical low-voltage halogen-lamp transformer provides no lamp-dimming capability.

This simple circuit does not allow for correctly and conveniently dimming the lamp because any attempts to manipulate the time constant of the R_1/C_1 start-up circuit worsen the power factor. A possible solution that involves making taps in the primary winding of transformer T_2 is neither elegant nor convenient.

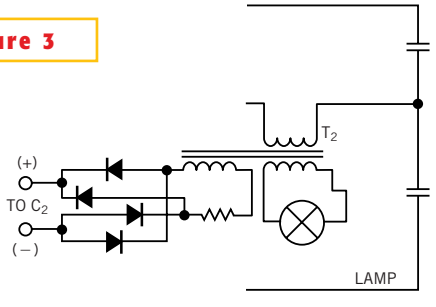
Figure 2



A simple PWM IC controls the duty cycle of Q_1, which dims or brightens the lamp.

The circuit in **Figure 2** provides a way to dim the lamp. In this circuit, a popular PWM IC controls a modified buck regulator. One modification is a change in the position of Q_1, which lets you control the regulator without using a pulse transformer, for example. A second modification significantly reduces the capacity of C_1 because the circuit should act similarly to an autotransformer to keep a high power factor. Therefore, the rectified mains voltage powers the regulator, which sits behind the bridge rectifier. The regulator produces the same output-voltage shape but of lower and regulated amplitude. C_1 has a low value, so, in conjunction with L_1, C_1 removes any switching products from the output voltage without filtering out 100-Hz ripples of the mains voltage.

Figure 3



To keep the UC3842 IC running requires extra power to C_2, which secondary windings on T_2 and a bridge rectifier can provide.

Thus, the circuit is halfway between a buck regulator and a current source feeding an H-bridge converter. Using a classic current source to feed the bridge converter would be inconvenient because of the lamp power supply's self-oscillating mode and variable switching frequency.

Changing the duty cycle of Q_1's pulses accomplishes the lamp dimming. You can use any one of many integrated PWM controllers to control the switching of Q_1. The only important point is that the PWM IC needs to work with the maximum pulse duty cycle, which is important if you want to keep the brightness of the

halogen lamp close to maximal. The prototype uses a cheap controller, the UC3842, which operates in voltage mode because of the presence of the R_3/R_4 divider.

To operate the circuit, you need to supply the controller. On power-up, C_2 charges through R_1 and R_2, which starts the switching. To keep the controller running requires extra power to C_2. The additional, secondary windings on T_2 and the bridge rectifier support this function (**Figure 3**). Unfortunately, the core choke, L_1, can have no secondary windings because those windings can't supply enough power to the controller under maximum-duty-cycle conditions.

After adding an EMI filter and bridge rectifier to the regulator input, the circuit can work in stand-alone mode. With the circuit in this mode, you can connect to the output more than a few parallel ready-made electronic transformers without making any changes. In such a case, you need to recalculate the values and types of power components according to the power load. (DI #2525)

To Vote For This Design,
Circle No. 345

Low-cost, programmable oscillator operates at 2 GHz

A Navarro, M Rostami and J N Matos, Instituto de Telecomunicações, Aveiro University, Portugal

All wireless communication systems need a local oscillator. **Figure 1** shows a low-cost PLL circuit for the 2-GHz band for such applications as industrial, scientific, and medical (ISM) and satellite personal-communication systems (S-PCS). The central IC in the circuit is the LMX2325 (National Semiconductor, www.national.com) frequency synthesizer, which features RF operation to 2.5 GHz. This IC incorporates an N-programmable divider and a phase comparator. The JTOS-3000P (Mini-Circuits) VCO constrains the oscillation frequency to 2.3 to 2.5 GHz. The VCO's tuning-voltage range of 0 to 5V is compatible with the internal charge-pump D0 output of the LMX2325. The COP8SAA7 µC programs the synthesizer's internal registers. You can calculate the register values using the program LMX2325.EXE (National Semiconductor). The values depend on the desired oscillation frequency, the external crystal, and the reference-frequency values. R_1, C_1, and C_2 implement a second-order loop filter, and you can determine the component values using the LOOPFILT.EXE (National Semiconductor) program. The values depend mainly on the oscillation frequency and the VCO tuning sensitivity.

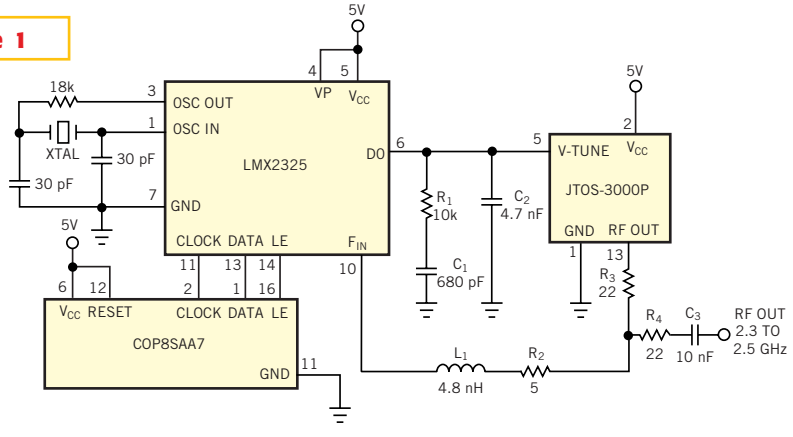
R_2, R_3, R_4, and L_1 form a T-network. The design in **Figure 1** assumes that a 50Ω load connects to ac-coupling capacitor C_3. The T-section matches the synthesizer F_{IN} pin's input reactive impedance of 40-jω to the 50Ω output impedance of the VCO's RF OUT pin and output load. If f is the oscillation frequency in hertz, X is approximately equal to $(-36\times10^{-9})f+162$. Therefore, the following inductor value for L_1 cancels the reactance:

$$L=\frac{9}{\pi}\left(\frac{9}{f}-2\times10^{-9}\right).$$

You must place the matching network components close to each other on the pc board. A power supply with good regulation characteristics is necessary, and, therefore, you should use a 7805 IC on the pc board.

A prototype of **Figure 1**'s circuit operating at f=2.45 GHz has an output power of 7 dBm and a phase noise of −75 dBc/Hz at a 10-kHz offset. You can download the source program for the µC from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2526. You can then use the previously mentioned support software (www.national.com/cop8/) or use the .hex file. (DI #2526)

Figure 1



An LMX2325 frequency-synthesizer IC, a JTOS-3000P VCO, and a COP8SAA7 µC create a low-cost, programmable oscillator for 2-GHz wireless applications.

To Vote For This Design,
Circle No. 346

Ring-tone generator uses high-Q filter

Jonathan Celani, Linear Technology Corp, Milpitas, CA

Modern telephone applications need to generate a high-voltage sine wave to use as a ring-tone signal. Several familiar design alternatives generally involve either an IC-controlled flyback-type scheme or a ringer-module-"brick" approach. Unfortunately, module approaches are expensive, and flyback circuits are generally complicated and have inherent harmonic-suppression issues. The design parameters of both these approaches are generally fixed and offer little flexibility for users.

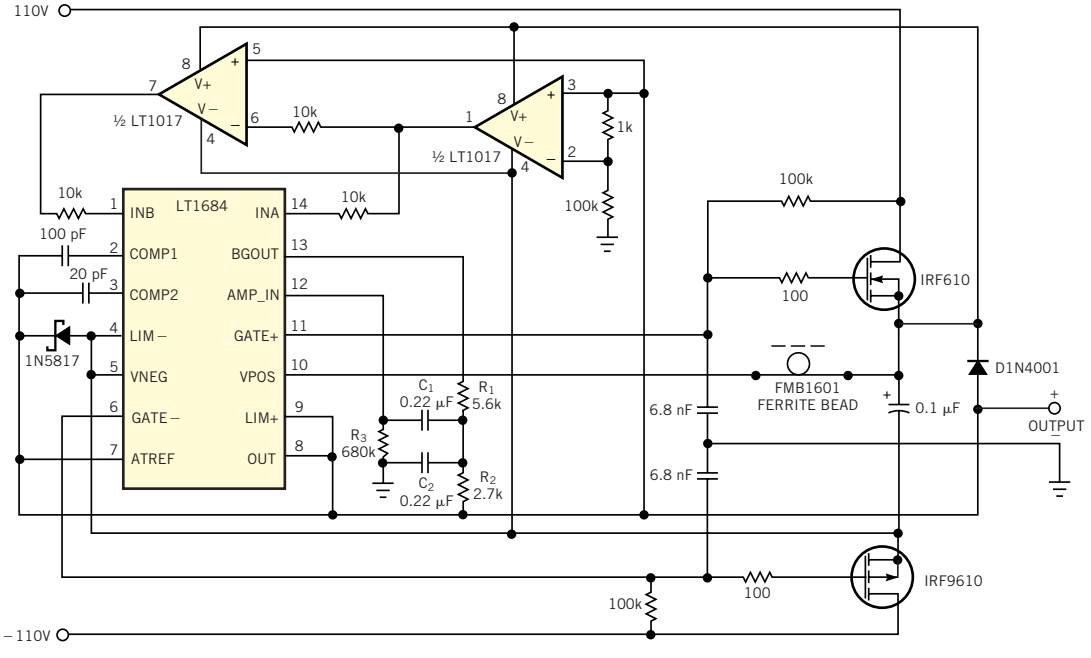
A simple yet effective method for producing a high-quality sine wave is to place a high-Q bandpass filter in a positive-feedback loop with a hard-limited gain element. This scheme produces a sine wave at the bandpass frequency, ω_O, at the filter output. This signal drives the input of the hard-limiter, which produces a square wave at the same frequency. Limiting voltage levels control the amplitude of this square wave. The limiter's output feeds back to the input of the bandpass filter, which completes the loop. The overall circuit produces a sine wave of frequency ω_O with an amplitude that corresponds to the filter gain, $|H_O|$, times the amplitude of the fundamental component of the square wave. This type of circuit is easy to produce using a couple of inexpensive ICs. The trick, however, is translating the IC-generated sine wave into a high-voltage signal suitable for ringing telephones.

The ring-tone generator in **Figure 1** configures an active-filter IC, the LT1684, as a high-gain, high-Q bandpass filter in a positive-feedback loop with a controlled-output comparator. This configuration produces a high-voltage, high-quality sine wave at the filter output. You can adjust amplitude and frequency characteristics by simply changing a few resistor values. The output of the LT1684 is internally current-limited to a minimum of ±100 mA, so load capacity is variable based on programmed peak-output amplitude. The LT1684 employs an active-tracking supply technique that provides a versatile and simple approach to the ring-tone problem. This technique comprises two high-voltage MOSFET source-followers that provide power for the IC. The MOSFETs function as power-supply level-shifters, bootstrapping the supply pins of the IC about the output of the active filter element. This scheme effectively suspends the IC between the MOSFETs, allowing the IC to output a voltage far exceeding its own supplies. The voltages across the MOSFETs themselves are the only limit to the IC's output-voltage capability. This supply-bootstrapping technique enables linear generation of a high-voltage output signal and takes advantage of the intrinsic supply-noise immunity of a linear amplifier, reducing the need for large high-voltage filtering elements.

The ring-tone generator's hard-limited gain element uses the LT1684's controlled-output receiver/ buffer to limit the output of a comparator. The circuit bootstraps an LT1017 dual-comparator IC along with the LT1684 to provide voltage gain and to provide the differential control signals that the LT1684 receiver/ buffer needs.

The active-bandpass-filter circuit uses

Figure 1



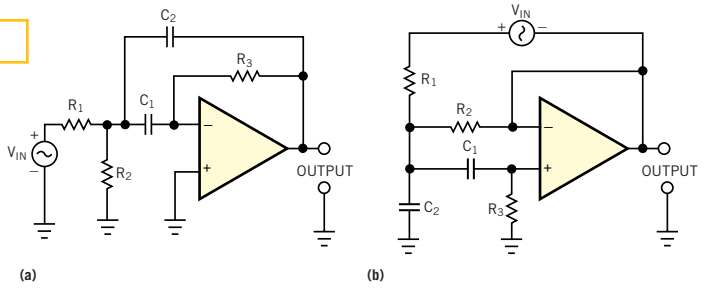
An active-filter IC, the LT1684, and a pair of MOSFETs produce a high-voltage, high-quality sine wave at the output.

a basic single-amplifier, multiple-feedback configuration using R_1, R_2, R_3, C_1, and C_2. However, the active-tracking supply technique of the LT1684 requires some modifications to the standard topology (**Figure 2a**). The modifications include swapping the amplifier's signal polarity, referencing all signals to the output, and referencing all feedback elements to ground. **Figure 2b** shows the modified multiple-feedback bandpass filter. The transformed multiple-feedback bandpass-filter design equations are the same as those of a standard topology. If both capacitors are the same value ($C_1=C_2$), the design equations that define the filter resistor values are as follows:

$$R_1 = \frac{Q}{\omega_0 \bullet C \bullet |H_0|}.$$

$$R_2 = \frac{Q}{(2Q^2 - |H_0|)(\omega_0 \bullet C)}.$$

Figure 2



The active-tracking multiple-feedback bandpass filter in Figure 1 (b) involves some modifications to the standard version (a).

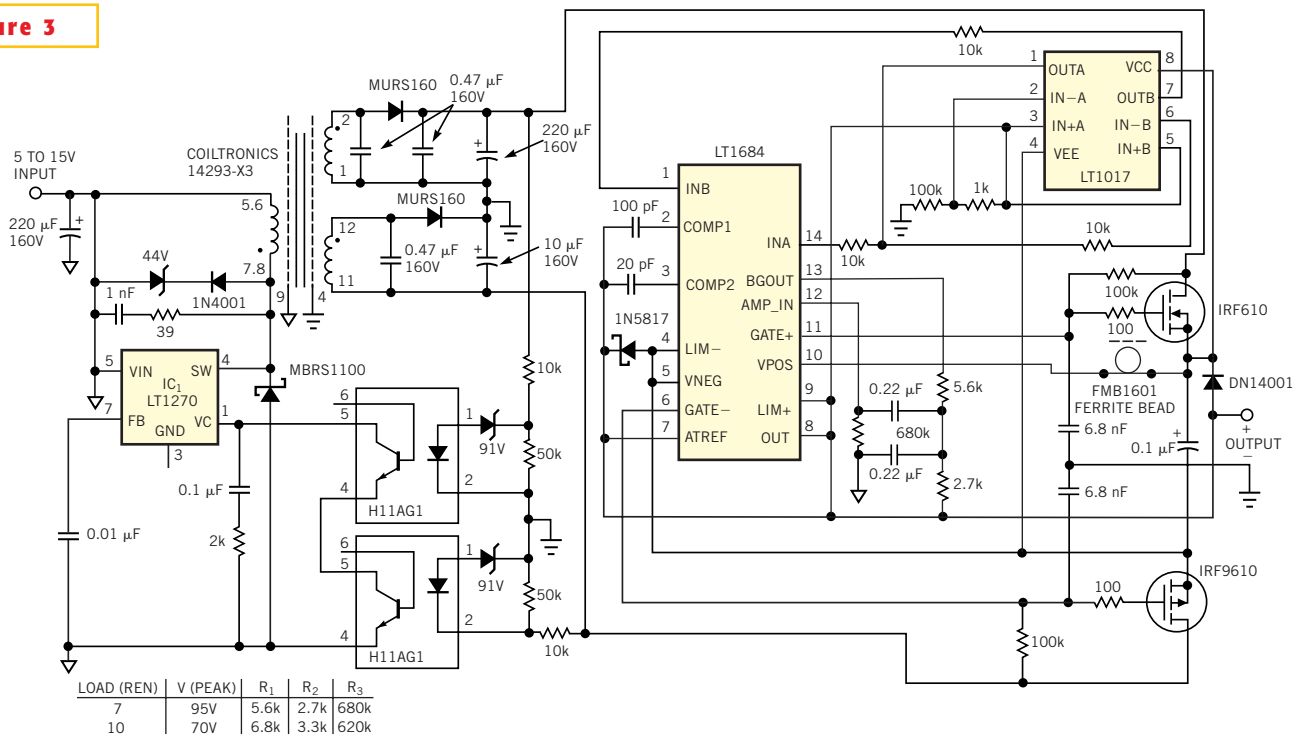
$$R_3 = \frac{2Q}{\omega_0 \bullet C}.$$

As a design example, consider that you want to construct a ringer circuit that yields a sine wave with a peak amplitude of 95V (7-REN load capability), a ring frequency of 20 Hz, and a bandpass Q of 9.4. A square wave with peak amplitude A has a fundamental component with an amplitude of $4A/\pi$. This value corresponds to an LT1684 buffer/receiver output, which has a limit of ±1.25V. Thus, A=1.25V. Now, the desired filter's bandpass gain is

$$|H_0| = \frac{95}{(4 \bullet 1.25/\pi)} \approx 60.$$

Given capacitor values of 0.22 µF, Q=9.4, $|H_O|=60$, and $\omega_O=2\pi(20\ Hz)$,

Figure 3



LOAD (REN)	V (PEAK)	R_1	R_2	R_3
7	95V	5.6k	2.7k	680k
10	70V	6.8k	3.3k	620k

A dc-to-ring-tone converter combines the circuit from Figure 1 with a high-voltage power supply.

you can calculate R_1=5.6 kΩ, R_2=2.7 kΩ, and R_3=680 kΩ.

You can describe many telephone-ringer modules as dc-to-ring-tone converters. A dc power supply provides power, and the output is a high-voltage sine wave. Most of these modules also offer ground-path isolation between the supply input and the ring-tone output. You can construct a circuit that mimics these attributes by coupling an isolated, high-voltage, dual-output power supply with the LT1684/LT1017 resonant ringer circuit of **Figure 1**.

Due to the ripple rejection inherent in an op amp, the power supply requires little output filtering. Thus, you can use a relatively simple supply. The ringer circuit generates a supply load that is unbalanced, however, because the circuit imposes the low-frequency ring signal across the system load. Thus, the supply must have well-controlled cross-regulation to prevent excessive output-voltage variations during normal operation. The dc-to-ring-tone-converter circuit in **Figure 3** uses a high-voltage power supply built around IC_1. This supply uses a dual-feedback scheme that regulates on the loaded output and uses capacitor-coupled secondary windings to limit the voltage excursions at the supply outputs. (DI #2527)

To Vote For This Design,
Circle No. 347

Simple circuit produces a less-than-25-nsec pulse

Andy Brandenberger, National Instruments, Austin, TX

In many testing situations, a circuit that can produce a short pulse is useful. You can use a short pulse to test trigger or interrupt-pulse-width requirements and to mimic glitches or noise on inputs, for example. In some cases, short pulses with long delays between them are desirable. For these cases, a function generator is inadequate because you can reduce the duty cycle low enough to produce low-nanosecond pulse widths only at relatively high frequencies. Although purely analog circuits can generate short pulses, input and output loading and buffering propagation delays create problems with interfacing them to TTL-compatible circuitry.

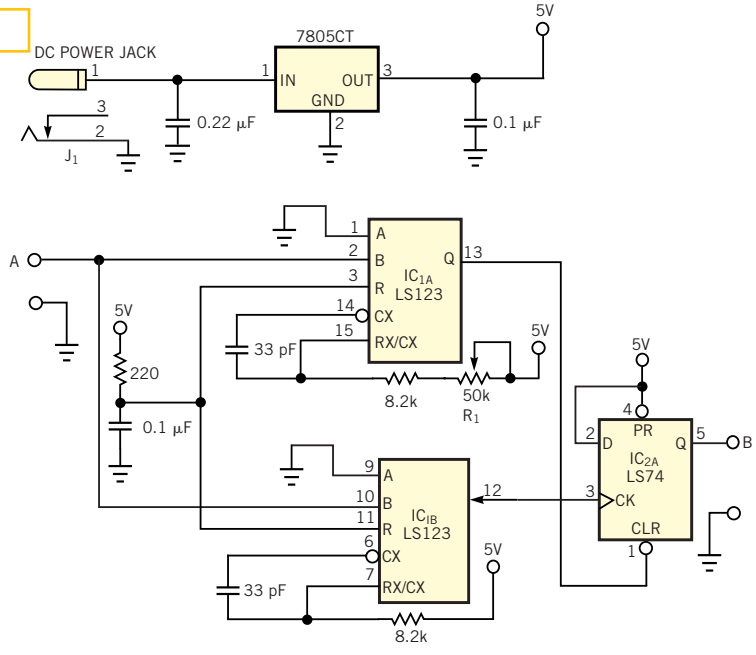
The circuit in **Figure 1** uses two one-shot timers and a D flip-flop to create a short, TTL-compatible positive pulse when triggered with a TTL-compatible positive edge. Because the circuit is positive-edge-triggered, the only requirement for the duty cycle of the triggering device is that it meets the minimum input pulse width for the one-shot timers, which is 40 nsec for the 74LS123, IC_1.

IC_{1B} provides a fixed pulse width of approximately 150 nsec. Via R_1, IC_{1A} provides a variable pulse width of 150 to 1000 nsec. A positive pulse at Input A triggers each timer. This trigger inactivates IC_{2A}'s CLR input and makes its CK signal low. When IC_{1B} times out, its positive-going edge clocks the logic 1 on the D input of IC_{2A} to Output B. When IC_{1A} times out, it's output clears the Q output of IC_{2A}, taking Output B low. Thus, you can express the pulse width at Output B as follows: Pulse width of IC_{2A}=pulse width of IC_{1A}−pulse width of IC_{1B}.

The output pulse width for this circuit is variable from approximately 5 to 850 nsec. You can produce a shorter pulse width by using faster logic. The circuit also allows for easy modification should negative trigger inputs, output pulses, or both be necessary. (DI #2528)

To Vote For This Design,
Circle No. 348

Figure 1



Two one-shots and a D flip-flop comprise a pulse generator that can produce pulses of 5 to 850 nsec with a low repetition rate.

µC multiplexes six digital potentiometers

Ted Salazar, Maxim Integrated Products, Sunnyvale, CA

In Figure 1's circuit, a µC lets you adjust as many as six digital potentiometers. Like mechanical potentiometers, digital potentiometers can adjust regulator outputs or speaker volumes and act as rheostats. Available in resistances of 50, 100, and 200 kV, they let you throw away your mechanical potentiometers and the little flat-headed screwdrivers that go with them.

A DIP switch, two pushbutton switches, and IC_6's µC control the six digital potentiometers, IC_0 through IC_5 in the circuit. All of the potentiometers have a nominal end-to-end temperature coefficient of resistance of 50 ppm/°C. At power-up, each potentiometer assumes its midrange value of resistance. Each wiper connects to one of 32 equally spaced taps along the linear-taper resistance between the H and L terminals for that potentiometer.

TABLE 1–DIGITAL-POTENTIOMETER SELECTION

A_4	A_3	A_2	Digital potentiometer
0	0	0	IC_0
0	0	1	IC_1
0	1	0	IC_2
0	1	1	IC_3
1	0	0	IC_4
1	0	1	IC_5
1	1	0	IC_5 (default)
1	1	1	IC_5 (default)

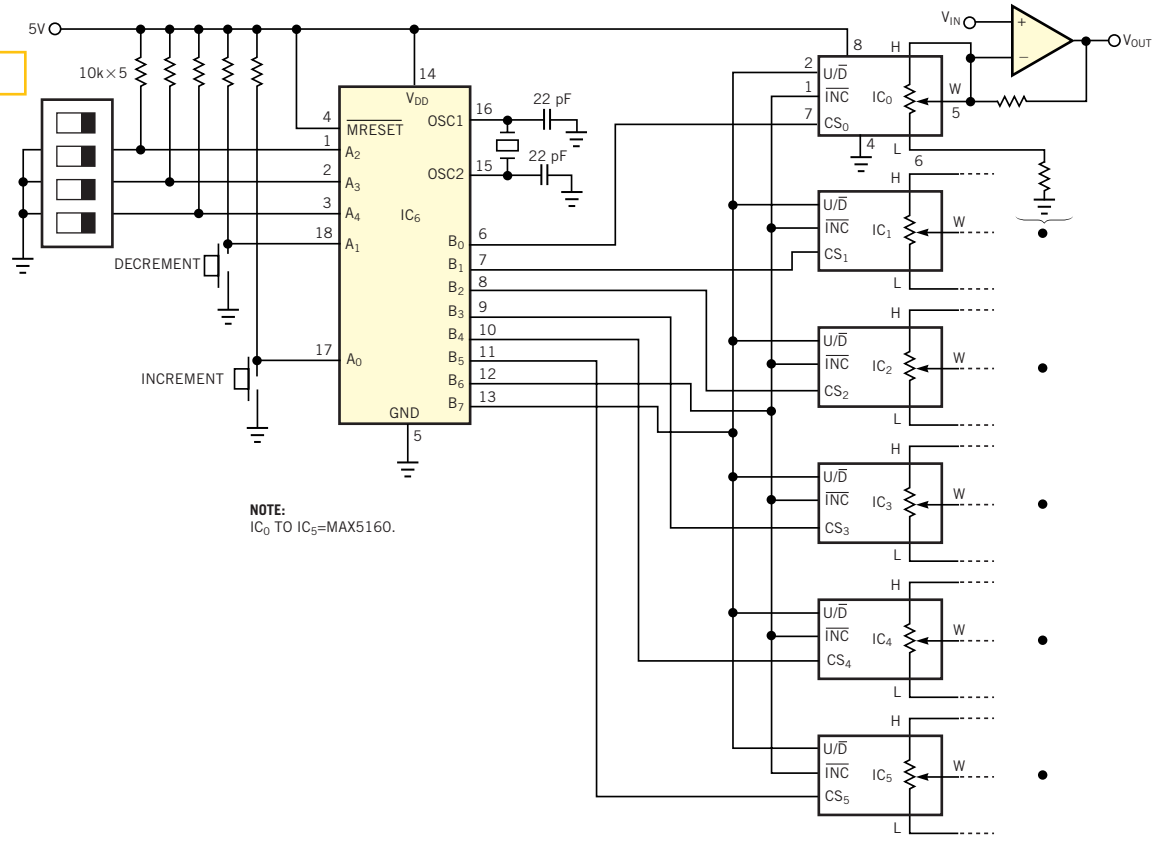
TABLE 2–INCREMENT/DECREMENT CONTROL

A_1	A_0	Operation
0	0	No change
0	1	Decrement selected potentiometer
1	0	Increment selected potentiometer
1	1	No change

Of the 13 I/O ports on IC_6, Port A (A_4 through A_0), comprises the inputs. Port B (B_7 through B_0) comprises the outputs. To operate this manually controlled digital system, you select the desired potentiometer by setting a code via the DIP switches (**Table 1**) and then increment or decrement the potentiometer via pushbutton switches that connect to A_1 and A_0 (**Table 2**). You can obtain the µC's assembly-language program from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2529. (DI #2529)

To Vote For This Design, Circle No. 349

Figure 1



A single µC controls six digital potentiometers.

design ideas

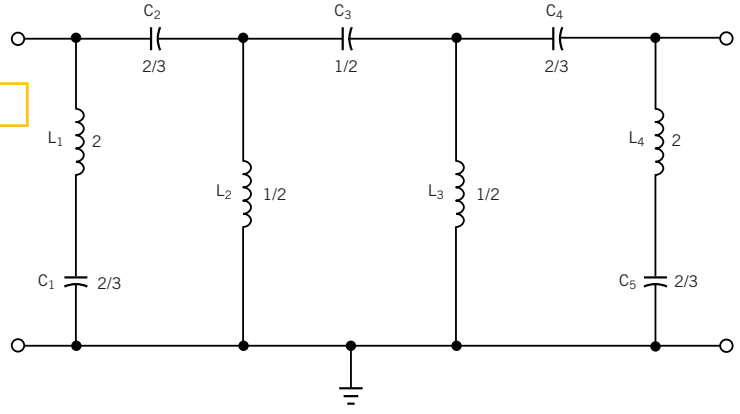
Edited by Bill Travis and Anne Watson Swager

Filter design uses image parameters

Richard Kurzrok, RMK Consultants, Queens Village, NY

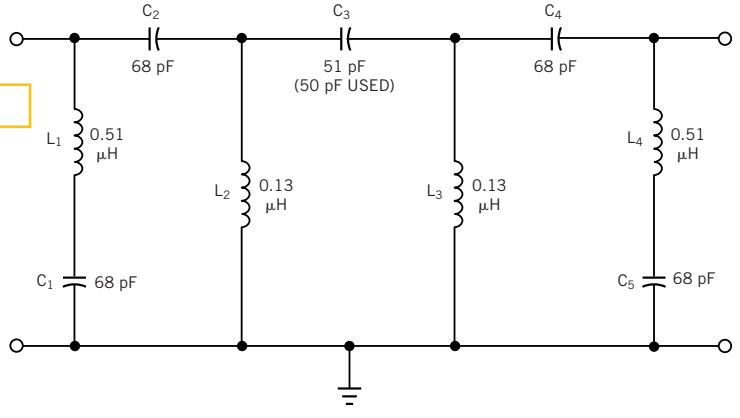
Reference 1 gives low-cost image-parameter design techniques for LC lowpass filters. Filter design using a low number of circuit elements results in reduced costs for both parts procurement and manufacturing. The technique applies to highpass filters. You derive a composite highpass filter by using m-derived terminating half-sections with one or more constant-k interior full sections. Classic image-parameter design used m-derived half-sections with m=0.6 for best passband impedance matching (in other words, high input and output return losses). The design uses a value of m=0.5 for the terminating half-sections. This value provides sharper close-in selectivity while maintaining passband return losses that are satisfactory for most applications. **Figure 1** shows the normalized schematic for the composite highpass filter. It uses midseries, m-derived, terminating half-sections with m=0.5, plus two interior constant-k full sections. The 3-dB cutoff frequency, f_0, is 31.2 MHz, and source and load impedances, Z_0, are 50Ω. Reference levels of filter inductance

Figure 1



Using image parameters results in a low number of circuit elements in a filter.

Figure 2



Multiplying the normalized filter in Figure 1 by the reference inductance and capacitance values yields this 31.2-MHz, 50V filter.

TABLE 1–FILTER PARTS LIST

Function	Value	Type	Quantity
L_1, L_4	0.51 µH	Micro-Metals T 25-10 14T- #26	Two
L_2, L_3	0.13 µH	Micro-Metals T 25-10 14T- #26	Two
C_1, C_2, C_4, C_5	68 pF	CD-15 Series dipped mica	Four
C_3	50 pF	DC-15 Series dipped mica	One
Connectors	BNC female	Pomona 2447 panel receptacle	Two
Enclosure	Aluminum box	Hammond 1590A/Bud CU-123	One
Board	Cut by hand	Vector board 169P44C1	One
Standoffs	Male/female	Amatom 9794-SS-0440	Four

Note: all fixed capacitors have ±15% tolerance.

and capacitance are as follows:

$$L_0 = \frac{Z_0}{2\pi f_0} = 0.255\ \mu H;$$

$$C_0 = \frac{10^6}{2\pi f_0 Z_0} = 102\ pF.$$

You obtain the actual inductance and capacitance values for the highpass filter by denormalization; in other words, by multiplying the normalized inductances and capacitances in **Figure 1** by L_0 and C_0, respectively. **Figure 2** shows the actual component values for a dissipationless highpass filter. **Table 1** gives the parts list for the filter. **Table 2** gives the measured amplitude response for the composite highpass filter. The results indicate inductor unloaded Qs of approximately 100. As the passband frequency approaches 100 MHz, some modest shape degradation occurs. You can reduce the degradation by using microstrip construction with surface-mount components. You can trim the filter's cutoff frequency by spreading or squeezing the turns of the toroidal inductors. (DI #2533)

TABLE 2–MEASURED AMPLITUDE RESPONSE

Frequency (MHz)	Insertion loss (dB)
29	23.7
30	12.8
31	3.7
31.5	1.8
32	1
33	0.6
35	0.5
40	0.5
45	0.4
50	0.2
55	0.2
60	0.2
70	0.4
100	0.5
130	0.6

REFERENCE

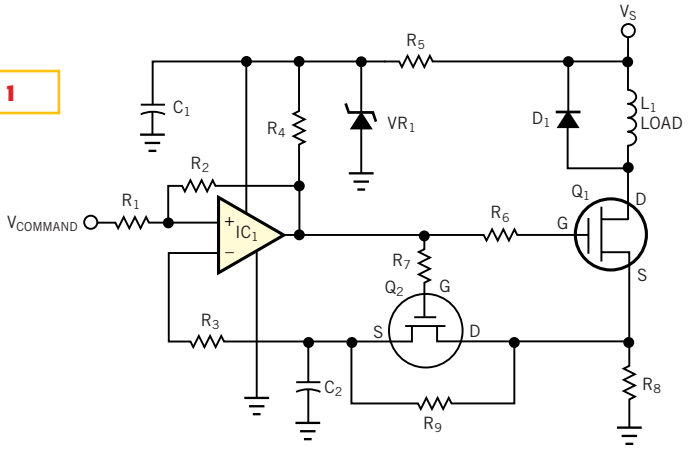
1. "Low Cost Lowpass Filter Design Using Image Parameters," *Applied Microwave & Wireless*, February 1999, pg 72, plus correction May 1999, pg 12.

Circuit efficiently drives inductive loads

Carlisle Dolland, Honeywell Engines and Systems, Torrance, CA

IN THE DRIVER CIRCUIT in **Figure 1**, the system controller provides the $V_{COMMAND}$ signal. $V_{COMMAND}$ equals the desired load current multiplied by R_8. When the controller applies this voltage to R_1, the output of IC_1 goes high, applying voltage to the gates of Q_1 and Q_2. These transistors turn on, allowing load current to flow to ground through Q_1 and R_8. The current in the load ramps up, and a voltage proportional to the load current, sensed by R_8, feeds back to the inverting input of the comparator IC_1. When this voltage exceeds the voltage at the noninverting input, the output of IC_1 goes to ground. Q_1 and Q_2 then switch off. The load current now circulates around the loop comprising D_1 and L_1. During this time, the slope of the load current becomes negative because of the dissipation in D_1 and the load resistance. The duration of this phase of the circuit's operation is a function of the hysteresis (set by R_1, R_2, and R_4) and the decay of the voltage across C_2 (essentially a function of R_9). C_2 and R_9 determine the ripple current in the load. The circuit cannot use a power MOSFET for Q_2, because of the intrinsic drain-to-source diode. You must use a device without the intrinsic diode, such as a 3N71. (DI #2535)

Figure 1



Inductive loads are tricky to drive. This circuit provides efficient drive to relays and solenoids.

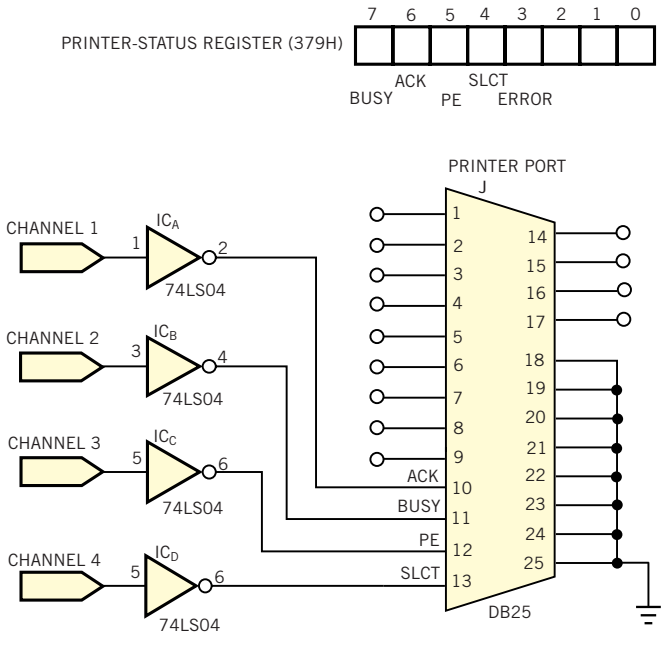
Use a PC to record four-channel waveforms

Dean Chen, Dycam Inc, Chatsworth, CA

THIS DESIGN IDEA is a sequel to a previous one, "Use a printer port to record digital waveforms," *EDN*, June 18, 1998, pg 136. Both ideas are similar: Use the PC's printer port to sample waveforms, and use the PC's memory to store data. The technique presented here expands the capability to four channels. The advantage is that you can see the relationships of the waveforms in the four channels. **Figure 1** depicts the sampling circuit. It uses printer-port pins ACK, BUSY, PE, and SLCT to record signals. The 74LS04 is a buffer between the sampled signals and the printer port. **Listing 1** is the sampling program, written in assembly language. Because there are four channels, every sample needs 4 bits (one nibble) to record. One byte can store two samples: odd and even samples. To accurately record signals, the sampling program needs exclusive access to the CPU.

Execution of the program must take

Figure 1



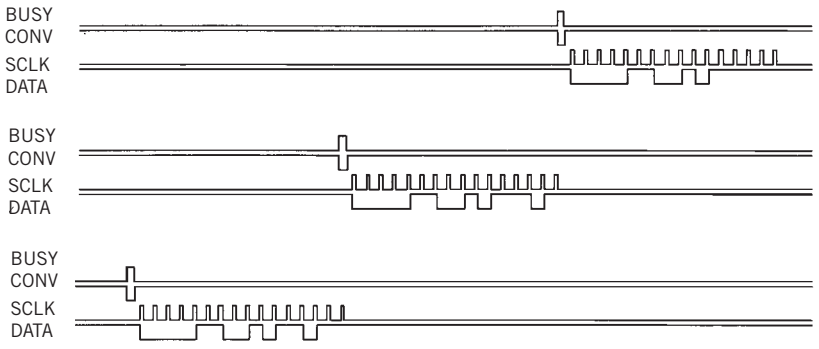
Use your PC's printer port to record four-channel waveforms.

LISTING 1—FOUR-CHANNEL PC-PORT WAVEFORM-SAMPLING ROUTINE

```
;                  Printer Status Register
;
;                  7  6  5  4  3  2  1  0
;                   -------------------------
;                   |  |  |  |  |  |  |  |
;                   -------------------------
;                   |  |  |  |--------------- ERROR
;                   |  |  |------------------ SLCT
;                   |  |--------------------- PE
;                     |  |---------------------- ACK
;                     |------------------------- BUSY
;
;
;sta_reg            equ   0379h
;mask_reg           equ   021h
;
code               segment para    public 'code'
                   assume  cs:code
                   org     100h
begin:               jmp main
file_name    db      'samsig.dat',0
msg          db      'Sample Signal is saved in samsig.dat ! $'
;
main               proc near
                lea di,buffer
                mov dx,sta_reg
pe_l:           in al,dx                        ; PE is as Trigger Signal
                and al,20h
                jz  pe_l                        ; When PE from 0 to 1,
pe_h:           in al,dx
                and al,20h                      ; Start Sampling.
                jz  pe_h
                mov dx,mask_reg
                in al,dx
                or al,01h               ; Mask Time Interrupt
                out dx,al
                mov dx,sta_reg
                mov cx,0f000h                   ; Sample Size
sam_lp:
                in al,dx                        ; Even Sample
                shr al,1
                shr al,1
                shr al,1
                shr al,1
                shr al,1
                xchg al,ah              ; high nibble save in ah
                nop
                nop
                nop
                in al,dx                ; Odd Sample
                and al,0f0h             ; get low nibble
                or al,ah                ; 2 nibble form 1 byte
                xor al,077h                   ; correct bits polarity
                stosb


                loop sam_lp
                mov dx,mask_reg
                in al,dx
                and al,0feh                     ; Allow Time Interrupt
                out dx,al
create_file:
                lea dx,file_name                ; memory is full,
                mov cx,0                        ; save data to 'samsig.dat' file
                mov ah,3ch
                int 21h
                mov bx,ax
write_data:
                mov cx,0f000h
                lea dx,buffer
                mov ah,40h
                int 21h
close:                mov ah,3eh
                int 21h
                lea dx,msg
                mov ah,9
                int 21h
                int 20h
main                endp
;
buffer:
code                ends
                    end begin
```

place in pure MS-DOS mode, and not in a Windows multitasking environment. Second, it does not allow interrupts to occur during sampling. You must thus mask interrupts during the sampling procedure. Moreover, you need to equalize the odd and even sampling periods. Because the even sampling period is shorter then the odd one, the routine adds three nonoperation (NOP) instructions in the even sampling period. When the sampled data attains approximately 60 kbytes, the program restores the interrupt-mask register and generates a file named samsig.dat. **Listing 2** is a QBasic program for displaying the recorded waveforms. The program reads and then displays the samsig.dat file. **Figure 2** provides an example, a recording of the command and data signals from an Analog Devices AD7896 A/D converter. You can increase the sampling period by inserting some NOP instructions in the sampling routine. You can download **listings 1** and **2** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2536. (DI #2536)

Figure 2



Four channels of data from an AD7896 and the timing relationships thereof are visible.

To Vote For This Design, Enter No. 364 at www.ednmag.com/infoaccess.asp

LISTING 2–DISPLAY PROGRAM FOR SAMPLED WAVEFORMS

```
KEY 20, CHR$(0) + CHR$(72):     ON KEY(20) GOSUB UpLine
KEY 21, CHR$(0) + CHR$(80):     ON KEY(21) GOSUB DownLine
KEY 15, CHR$(0) + CHR$(73):     ON KEY(15) GOSUB UpPage
KEY 16, CHR$(0) + CHR$(81):     ON KEY(16) GOSUB DownPage
KEY 22, CHR$(0) + CHR$(75):     ON KEY(22) GOSUB Left
KEY 23, CHR$(0) + CHR$(77):     ON KEY(23) GOSUB Right
KEY 17, CHR$(0) + CHR$(1):      ON KEY(17) GOSUB Finish

SCREEN 12
DIM chr AS STRING * 1
DIM prv(3) AS INTEGER
DIM ptr AS LONG
OPEN "samsig.dat" FOR BINARY AS #1
 GET #1, 1, chr: lo% = ASC(chr) MOD 16
 FOR k% = 0 TO 3
  prv(k%) = lo% MOD 2: lo% = lo% \ 2
 NEXT k%
 d% = 12: dd% = 16: fl% = 0: ptr = 0
 KEY(17) ON

WHILE fl% = 0
 KEY(15) ON: KEY(16) ON: KEY(20) ON
 KEY(21) ON: KEY(22) ON: KEY(23) ON
 LOCATE 1, 36: PRINT ptr
 FOR i% = 0 TO 255: FOR j% = 0 TO 128: NEXT j%
 NEXT i%
 KEY(15) STOP: KEY(16) STOP: KEY(20) STOP
 KEY(21) STOP: KEY(22) STOP: KEY(23) STOP
 FOR i% = 0 TO 4
  y% = i% * 96 + 32
  FOR j% = 1 TO 320
   GET #1, ptr + j% + i% * 320, chr
   lo% = ASC(chr) MOD 16: hi% = ASC(chr) \ 16
   x% = 2 * j%
   FOR k% = 0 TO 3
    IF prv(k%) <> lo% MOD 2 THEN
     LINE (x%, y% + k% * dd%)-(x%, y% + k% * dd% + d%)
    ELSE
     IF (lo% MOD 2) THEN
      PSET (x%, y% + k% * dd%)
     ELSE PSET (x%, y% + k% * dd% + d%)
     END IF
    END IF
    prv(k%) = lo% MOD 2: lo% = lo% \ 2
   NEXT k%
   x% = x% + 1
   FOR k% = 0 TO 3
    IF prv(k%) <> hi% MOD 2 THEN
     LINE (x%, y% + k% * dd%)-(x%, y% + k% * dd% + d%)
    ELSE
     IF (hi% MOD 2) THEN
      PSET (x%, y% + k% * dd%)
     ELSE PSET (x%, y% + k% * dd% + d%)
     END IF
    END IF
    prv(k%) = hi% MOD 2: hi% = hi% \ 2
   NEXT k%
  NEXT j%
 NEXT i%
WEND
 KEY(15) OFF: KEY(16) OFF: KEY(20) OFF
 KEY(21) OFF: KEY(22) OFF: KEY(23) OFF
CLOSE #1
END

UpLine:
 IF ptr < 61120 THEN
  CLS 1: ptr = ptr + 320
 END IF
RETURN
Left:
 IF ptr < 61440 THEN
  CLS 1: ptr = ptr + 1
 END IF
RETURN
UpPage:
 IF ptr < 59840 THEN
  CLS 1: ptr = ptr + 1600
 END IF
RETURN
DownLine:
 IF ptr >= 320 THEN
  CLS 1: ptr = ptr - 320
 END IF
RETURN
Right:
 IF ptr >= 1 THEN
  CLS 1: ptr = ptr - 1
 END IF
RETURN
DownPage:
 IF ptr >= 1600 THEN
  CLS 1: ptr = ptr - 1600
 END IF
RETURN
Finish:
 fl% = 1
RETURN
```

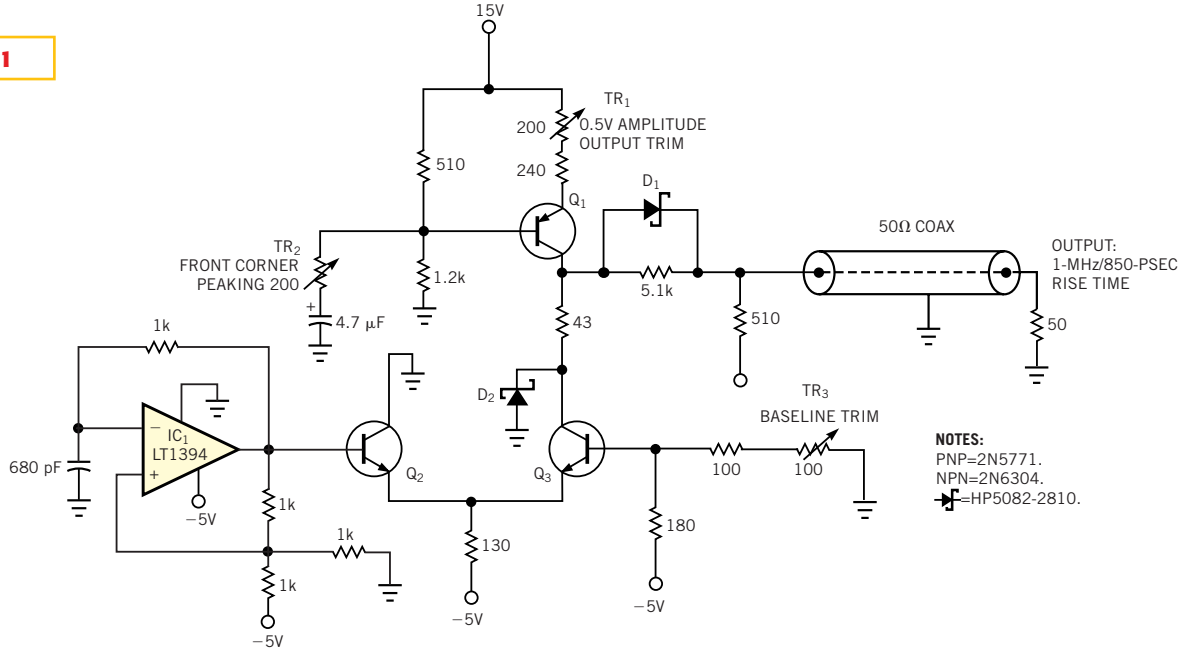
Pulse generator has low top-side aberrations

Jim Williams, Linear Technology Corp, Milpitas, CA

Impulse-response and rise-time testing often require a fast-rise-time source with a high degree of pulse purity. These parameters are difficult to achieve simultaneously, particularly at subnanosecond speeds. The circuit in **Figure 1**, derived from oscilloscope calibrators (**Reference 1**), meets the speed and purity criteria. It delivers an 850-psec output with less than 1% pulse-top aberrations. Comparator IC_1 delivers a 1-MHz square wave to current-mode switch Q_2-Q_3. Note that IC_1 obtains power between ground and −5V to meet the transistors' biasing requirements. Q_1 provides drive to Q_2 and Q_3. When IC_1 biases Q_2, Q_3 turns off. Q_3's collector rises rapidly to a potential determined by Q_1's collector current, D_1, and the output resistors combined with the 50Ω termination resistor. When IC_1 goes low, Q_2 turns off, Q_3 turns on, and the output settles to 0V. D_2 prevents Q_3 from saturating.

The circuit's output transition is extremely fast and singularly clean. **Figure 2**, viewed on a 1-GHz real-time-bandwidth oscilloscope, shows 850-psec rise

Figure 1



Need a fast, clean pulse? This simple circuit provides 500-mV, 850-nsec pulses with a high degree of purity.

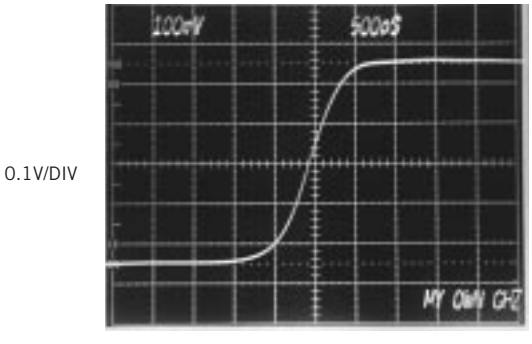


Figure 2

Viewed with 1-GHz bandwidth, pulses are free of discontinuities and anomalies.

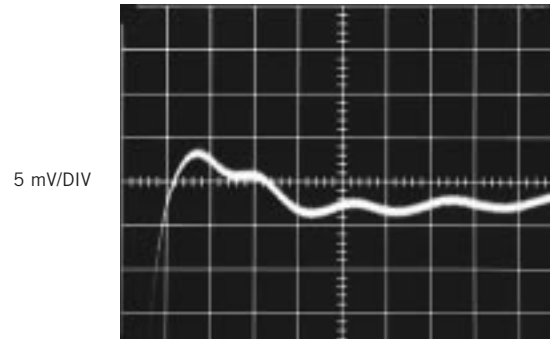


Figure 3

Pulse tops exhibit less than 64 mV, or less than ±1%, aberrations.

time with exceptionally pure pretransition and post-transition characteristics. **Figure 3** details the pulse-top settling. The photo shows the pulse-top region immediately following the positive 500-mV transition. Settling occurs within 400 psec of the edge's completion with all activity within ±4 mV. The 1-mV, 1-GHz ringing undoubtedly stems from breadboard-construction limitations; you can probably eliminate it by using stripline-layout techniques. The level of performance of this circuit requires some trimming. The oscilloscope you use should have at least 1-GHz bandwidth. You adjust trimmers TR_2 and TR_3 for the best pulse presentation. TR_1 sets the output amplitude at 500 mV across the 50Ω termination. The trims are somewhat interactive, although not unduly so, and converge quickly to give the results described. (DI #2530)

REFERENCE

1. 485 Oscilloscope Service and Instruction Manual, "Calibrator," pg 3 to 15, Tektronix Inc, 1973.

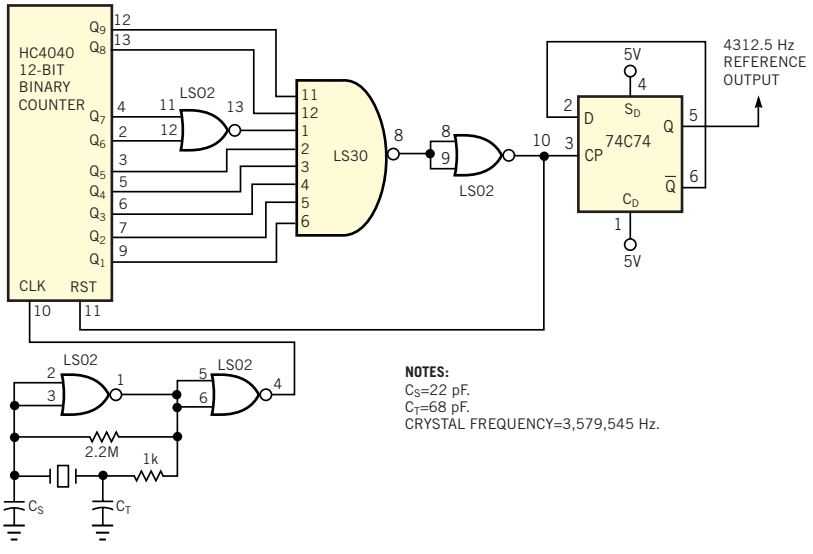
Circuit provides ADSL frequency reference

Bert Erickson, Fayetteville, NY

THE DISCRETE-MULTITONE (DMT) frequencies that asymmetrical-digital-subscriber lines (ADSL) use are integral multiples of a common frequency, and the symbol period is the inverse of this frequency. Integration over the symbol period allows the sine and cosine orthogonal waveform products to vanish for all multiples of the common frequency except for those having the same frequency. As the ADSL standards (TI.413) specify, the 256 channels are separated by 69/16 kHz. You can generate the midchannel frequencies with a PLL, but the reference frequency differs from that of crystals for computers and clocks. However, by using the circuit in **Figure 1**, you can generate the frequency by using a 3.58-MHz crystal to control the horizontal scanning rate in television sets. A typical 3.58-MHz crystal has a tolerance of ±50 ppm and a load capacitance of 18 pF. This tolerance provides a frequency of 3.579366 to 3.579724 MHz. If you multiply this common DMT frequency by 830, the result is 830×69/16 kHz, or 3.579375 MHz, which is 9 Hz above the crystal's lower tolerance limit. Assuming that you can select the C_S and C_T capacitors at either side of the crystal to tune the frequency near the lower tolerance limit, you can also select them for the desired frequency.

In other words, reduce the oscillator frequency with bistable flip-flops and combine the outputs in a NAND gate to divide by 830. For the 3.58-MHz crystal, design values for C_S and C_T were 23.6 and 75.7 pF, respectively. We chose 22 pF for C_S and 68 pF for C_T. A trimmer capacitor in parallel with C_T reduces the frequency. When C_T increased from 22 to 90 pF, the frequency decreased by 448 Hz and handily bridged the 3.579545- and 3.579375-MHz frequencies. Tests showed that the lower frequency was more than 100 Hz below 3.579357 MHz, but the exact number depends on the calibration of the counter. Because 830 is a 10-bit binary number, the circuit divides by 415 first to permit combining with an eight-input NAND gate. A strobe applied to a flip-flop then creates a square wave for the reference-frequency output. (DI #2531)

Figure 1



Using a common TV crystal, you can generate the reference frequency for ADSL systems.

ActiveX control brings bit manipulation to Windows

Steve Hageman, Agilent Technologies, Santa Rosa, Ca

Nothing compares with the C language for working with bits. C provides a rich set of signed and unsigned number formats, along with many intrinsic bit-manipulation operators. However, most of the popular rapid-application-development Windows languages lack C's ability to easily work with bits. Visual Basic is such a language. Although it's hard to find a faster language to develop a small to midsized application in Windows, Visual Basic starts to show its weakness when it comes time to talk to hardware. Hardware programming is usually bit-oriented. That is, it's necessary to turn bits on and off or shift out serial streams to get the hardware to operate correctly. The ActiveX control serves just these types of bit-manipulation needs (**Figure 1**). The control includes functions for changing binary strings to numbers, a hex-output function, the ability to set and clear bits in a word, and the ever-needed shift-left and -right functions. As an example, many of the three-wire serial devices need to have a setup word shifted to them. Suppose you need to shift the setup word 0111 1101 first to an A/D converter to initiate a conversion on some channel. You can use the functions in the ActiveX control to easily effect the shift operation, as follows:

```
Setup_word = Bits ("01111101")
`Returns 125
For i = 0 to 7
    Val = ShiftRight_8(setup_word,0)
    `write val to the A/D here
next i
```

In the above example, val has the values 1, 0, 1, 1, 1, 1, 1, 0 during each iteration of the loop. The routine can then clock these bits to the A/D converter as required by the hardware. If the operation requires MSB first, you can use the ShiftLeft function. The SetBit and ClearBit functions are useful when using a port as clock and data lines, because you can set individual bits as needed instead of doing entire port writes. Any modern programming language that can use ActiveX controls, such as Agilent VEE, Visual Basic, Delphi, and others, can use the functions given here. You can download the ActiveX control from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2534. The routine includes all the functions listed in **Figure 1**, plus a few more, with application examples. (DI #2534)

Figure 1

```
Function GetBit(ByVal x As Long, ByVal n As Integer) As Integer
Returns the value of bit n in input value x. Returns 1 or 0 if bit
is set or not. x = 1 to 16 bit, n = 0 = LSB.
Example: GetBit(16,5) returns 1.

Function Bits(ByVal inval As String) As Long
Given a  representation of a binary string, returns the
value. inval may be any length from 1 to 16 bits
Example: Bits("101") returns 5.

Function BitsStr(ByVal inval As Long, ByVal sizeof As Integer) As String
Given a number, returns with a representation of a binary string.
sizeof is the width of the return field (1 to 16 bits).
Example: BitsStr(82,8) returns "01010010"

Function HexStr(ByVal inval As Long, ByVal sizeof As Integer)
Given a number, returns with a representation of a hex string.
sizeof is the width of the return field (1 to 16 bits).
Example: HexStr(179,8) returns "B3"

Function ClearBit(ByVal x As Long, ByVal n As Integer) As Long
Clears bit position n in input x. Returns new x value.
x may be 1 to 16 bits, n = 0 = LSB
Example: ClearBit(16,4) returns 0.

Function SetBit(ByVal x As Long, ByVal n As Integer) As Long
Sets bit n in input value x. Returns new  x. x may be any
width 1 to 16 bits, n = 0 = LSB.
Example: SetBit(0,4) returns 16

Function ShiftRight_8(ByRef x As Integer, ByVal y As Integer) As Integer
Shifts the 8 bit value x right by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftRight_8(129,1) Returns 1 and the new value for x (was 129)
is 192.

Function ShiftRight_16(ByRef x As Long, ByVal y As Integer) As Integer
Shifts the 16 bit value x right by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftRight_16(1,1) Returns 1 and the new value for x (was 1)
is 32768.

Function ShiftLeft_8(ByRef x As Integer, ByVal y As Integer) As Integer
Shifts the 8 bit value x left by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftLeft_8(1,0) returns 0 and the new value for x (was 1)
is 2.

Function ShiftLeft_16(ByRef x As Long, ByVal y As Integer) As Integer
Shifts the 16 bit value x left by 1 place. Bit shifted in is y.
Returns bit shifted out.
Example: ShiftLeft_16(32768,1) returns 1 and the new value for x
(was 32768) is 1

Function RotateRight_8(ByVal x As Integer) As Integer
Rotates the 8 bit value x right by 1 place. Returns new value.
Example: RotateRight_8(1) returns 128.

Function RotateRight_16(ByVal x As Long) As Long
Rotates the 16 bit value x right by 1 place. Returns new value.
Example: RotateRight_16(1) returns 32768.

Function RotateLeft_8(ByVal x As Integer) As Integer
Rotates the 8 bit value x left by 1 place. Returns new value.
Example: RotateLeft_8(64) returns 128

Function RotateLeft_16(ByVal x As Long) As Long
Rotates the 16 bit value x left by 1 place. Returns new value.
Example: RotateLeft_16(32769) returns 3.

End.....
```

An ActiveX control offers many handy functions for bit manipulation.

Circuit breaker handles voltages to 32V

Greg Sutterlin and Craig Gestler, Maxim Integrated Products, Sunnyvale, CA

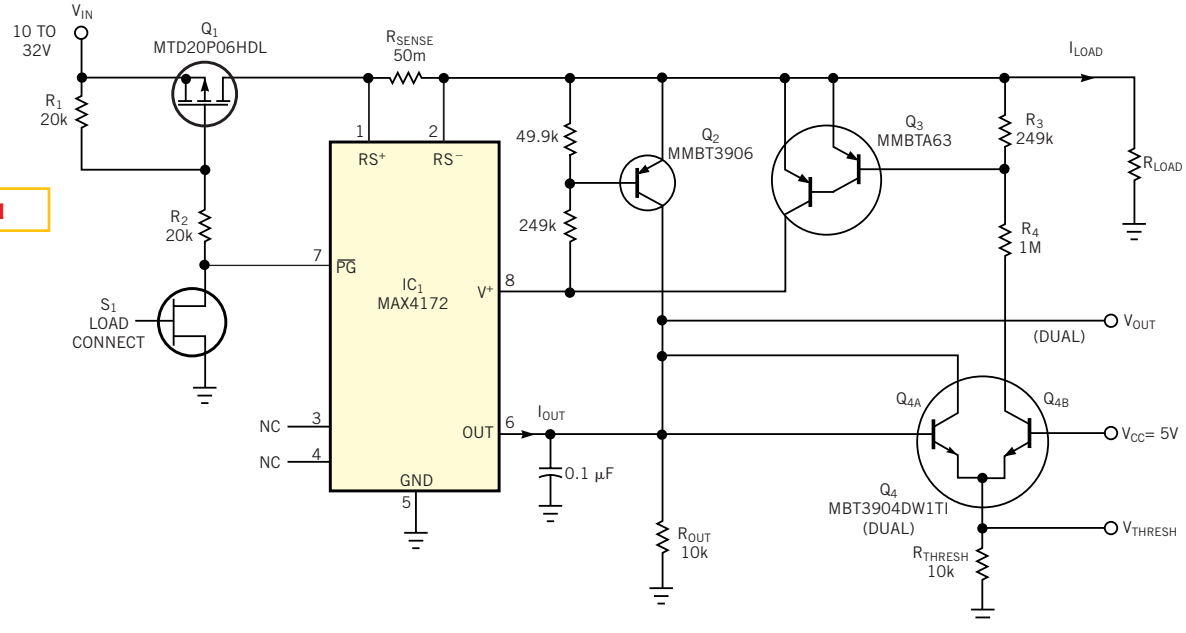
The simplicity of low-side current monitoring can mask the advantages of a high-side approach. You can monitor load currents in a power supply, a motor driver, or another power circuit on either the high or the low side (ground). However, don't let the ease of low-side monitoring cause you to overlook its dangers or the advantages of a high-side approach. Various fault conditions can bypass the low-side monitor, thereby subjecting the load to dangerous and undetected stresses. On the other hand, a high-side monitor connected directly to the power source can detect any downstream failure and trigger the appropriate corrective action. Traditionally, such monitors required a precision op amp, a boost power supply to accommodate the op amp's limited common-mode range, and a handful of precision resistors. Now, the MAX4172 IC can sense high-side currents in the presence of common-mode voltages as high as 32V (**Figure 1**). IC_1 provides a ground-referenced current-source output proportional to the high-side current of interest. This output current, equal to the voltage across an external sense resistor divided by 100, produces a voltage output across a load resistor.

IC_1 and a few external parts form a low-cost circuit breaker. R_{SENSE} senses load currents, and Q_1 controls the currents. The design accepts inputs of 10 to 32V; you can easily modify it to operate from voltages as low as 6.5V. The initial application of V_{IN} and V_{CC} places the breaker in its trip state. Pressing S_1 resets the breaker and connects power to the load, thereby activating Q_1, Q_3, and Q_{4B}. Q_3 powers IC_1, and Q_{4B} establishes the overcurrent threshold, $V_{THRESH}=V_{CC}-V_{BE(4B)}$. Because V_{CC} (2.7 to 5.5V typical) equals 5V and the base-emitter voltage of Q_{4B} is approximately 0.7V, V_{THRESH} is typically 4.4V. The circuit trips at a nominal load current of 1A. The values for R_{SENSE}, R_{THRESH}, and R_{OUT} are functions of the system's accuracy and power-dissipation requirements. First, select $R_{SENSE}=50\ m\Omega$ and $R_{THRESH}=10\ k\Omega$. Then, calculate $R_{OUT}=V_{CC}/I_{LOAD}R_{SENSE}G_m$, where I_{LOAD} is the trip point (1A) and G_m (IC_1's typical transconductance) equals 0.01A/v. Thus, $R_{OUT}=10\ k\Omega$.

Applying power to Q_3 and Q_{4B} causes Q_{4B} to conduct, which establishes V_{THRESH} and activates Q_3 to power IC_1. A fraction of the load current through R_{SENSE} mirrors to the IC_1 output and appears as a voltage, V_{OUT}, across R_{OUT}. Q_{4B} turns off when V_{OUT} increases above $(V_{THRESH}+V_{BE(4BA)})$, turning off Q_3 and causing a drop in V^+ (IC_1, pin 8). When V^+ reaches 2.67V (typical), $\overline{PG}$ goes high, thereby tripping the breaker by turning off Q_1. Q_2 adds feedback to ensure a clean turn-off at the trip level. Current draw in the tripped state is minuscule and equals the V_{CC} load current, 0.5 mA typical. Press S_1 to reset the breaker. The design is intended for low-cost applications in which the absolute accuracy of the trip current is not critical. The accuracy, which depends on variations in V_{CC} and the base-emitter voltages of Q_{4A} and Q_{4B} and on the error current through R_4, is approximately ±15% at a trip current of 1A. (DI #2532)

To Vote For This Design,
Enter No. 368 at
www.ednmag.com/infoaccess.asp

Figure 1



A current-sense amplifier and a few transistors form a low-cost circuit breaker.

Edited by Bill Travis and Anne Watson Swager

Model a nonideal transformer in Spice

Vittorio Ricchiuti, Siemens ICN, L'Aquila, Italy

Designers often use transformers as voltage, current, and impedance adapters. Transformers usually comprise two inductively coupled coils, wound around a ferrite core. The coupling between the windings is never perfect. Spice provides a model (**Figure 1a**) of the coupled inductors using the k parameter, which is the coefficient of coupling between the windings. The model takes into account self and mutual inductances. With nonideal transformers, the problem is to determine k. **Figure 1b** shows a proposed equivalent circuit of a nonideal transformer, in which the conduction losses in the windings and the core losses are assumed to be negligible. L_S is the equivalent leakage inductance of the transformer, L_P is its magnetization inductance, and T is an ideal transformer (k=1) with transformation ratio equal to n. To obtain equivalence between the two circuits in **Figure 1**, we consider the equations describing these circuits. For the circuit in **Figure 1a**, the expressions are

$$V_1 = j\omega L_1 I_1 + j\omega M I_2;$$

$$V_2 = j\omega M I_1 + j\omega L_2 I_2.$$

For the circuit in **Figure 1b**, the equations are

$$V_1 = j\omega(L_S + L_P) I_1 + j\omega L_P \frac{I_2}{n};$$

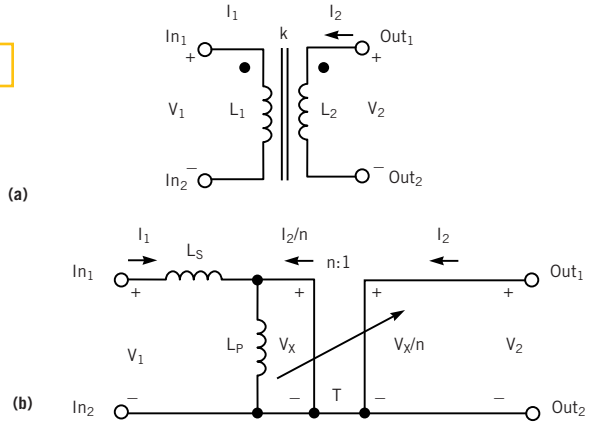
$$V_2 = j\omega \frac{L_P}{n} I_1 + j\omega \frac{L_P}{n^2} I_2.$$

Comparing the two systems and considering $M = k(L_1 L_2)^{1/2}$, we obtain

$$L_1 = L_S + L_P;$$

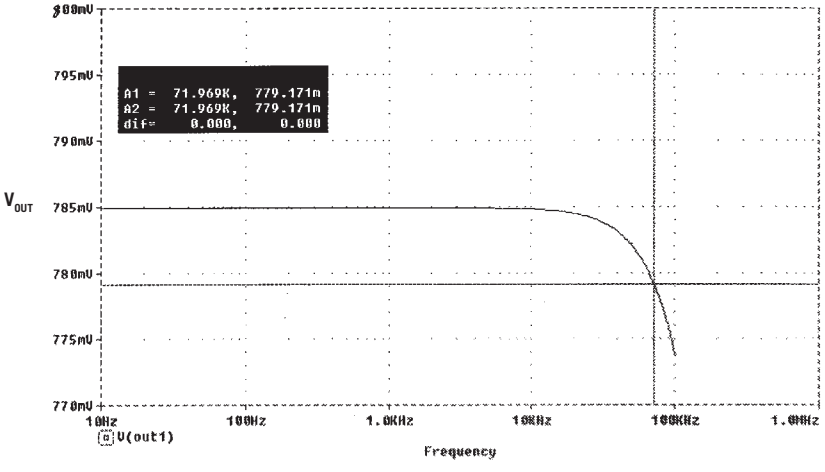
$$L_2 = \frac{L_P}{n^2};$$

Figure 1



Spice models simulate an ideal transformer (a) and a nonideal one (b).

Figure 2



A Spice simulation yields the transfer function of the nonideal transformer described in the text.

$$L_S = L_P\left(\frac{1}{k^2}21\right);$$

$$n = k\left(\sqrt{\frac{L_1}{L_2}}\right).$$

Then, if you know the L_P and L_S values, you also know the coupling factor, k. L_1 is the inductance measured at the operating frequency between terminals In_1 and In_2 with no load connected between Out_1 and Out_2. Similarly, L_2 is the inductance measured at the operating frequency between terminals Out_1 and Out_2 with no load connected between In_1 and In_2. L_S is the inductance measured at the operating frequency between terminals In_1 and In_2 with Out_1 and Out_2 short-circuited. From these values, using the previous **equations**, we obtain the parameters of the equivalent circuit in **Figure 1b**. **Listing 1** shows the PSpice subcircuit that represents the behavioral model of a nonideal transformer. You can use the subcircuit for both transient and ac analysis.

LISTING 1—SPICE SIMULATION OF NONIDEAL TRANSFORMER

```
.SUBCKT transformer in1 in2 out1 out2 params: L1=1u L2=1u Ls=1u
R_Rs       in1 1  1n
L_Ls       1 2 {Ls}
L_Lp       in2 2  {L1-Ls}
E_E1       3 out2 VALUE { (sqrt(L2/(L1-Ls)))*V(2, in2) }
V_Vsense   out1 3 0V
F_F1       4 0 V_Vsense 1
R_Rload    4 0  0.1n
G_G1       2 in2 VALUE { ((10G)*sqrt(L2/(L1-Ls)))*V(4, 0) }
.ENDS    transformer
```

The input parameters of the subcircuit are the measured values of inductances L_1, L_2, and L_S. You obtain the ideal transformer, T, by means of the voltage-controlled voltage source, E_1, and the voltage-controlled current source, G_1, connected back to back (**Reference 1**). The current source, G_1, senses the current, $I(V_{sense})$, and provides the current $I(V_{sense})/n$. The transformation ratio n is a function of inductances L_1, L_2, and L_S.

As an example, consider a transformer that provides an impedance transformation of 46 to 75Ω at 72 kHz. It uses an RM8 ferrite core with inductance factor A_L=1600 nH. The measured inductances are L_1=4.2 mH, L_2=2.6 mH, and L_S=20 μH. **Figure 2** shows the simulated transfer function of the transformer. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2539. (DI #2539).

REFERENCE

1. Coelho, J, "A Spice model for the ideal transformer," *Electronic Design*, June 28, 1999.

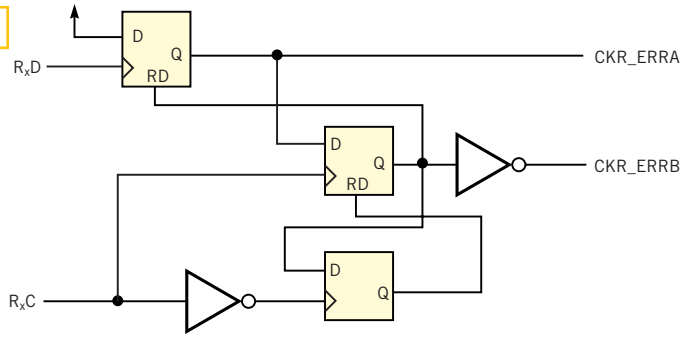
TO VOTE FOR THIS DESIGN, ENTER NO. 500

Clock-recovery scheme suits low-SNR systems

Luis Miguel Brugarolas, SIRE, Madrid, Spain

A CLOCK-RECOVERY ARCHITECTURE can operate with NRZ digital signals, even at low SNRs. A clock-recovery subsystem is based on a PLL comprising a phase comparator, a loop filter, and a voltage-controlled oscillator (VCO). If you place the phase comparator after the demodulator block, a typical criterion for comparison is the zero crossings of the received signal (the edges of a TTL-compatible signal). The phase comparator must provide a voltage proportional to the phase difference between the incoming signal and the local-

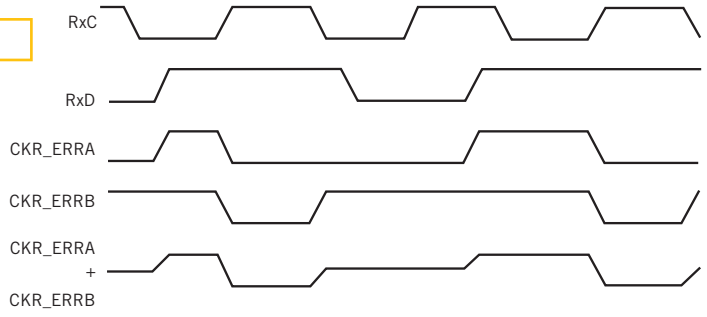
Figure 1



A programmable-logic IC forms the phase comparator for the clock-recovery system.

ly generated clock (the VCO output). But, in comparison with a synthesizer-based phase comparator, the comparison must take place only when information exists; that is, at data-level transitions. The recovery system should disable the phase comparator in the absence of data-level transitions to minimize comparator noise. **Figure 1** shows a phase comparator you can implement in a CPLD or an FPGA. Tests show that the comparator works successfully in a 2-Mbps V_{SAT} demodulator with SNR of 6 dB and lower. The comparator can easily acquire and track the signal.

Figure 2



The last waveform represents the absolute value of the difference between CKR_ERRA and CKR_ERRB.

The phase comparator uses three D-type flip-flops with asynchronous reset inputs. The comparator has two inputs: RxD and RxC, and two outputs: CKR_ERRA and CKR_ERRB. These inputs and outputs form a three-level output, both high for increasing frequency, both low for decreasing frequency, and both at different levels for no change in frequency. They form a state machine:

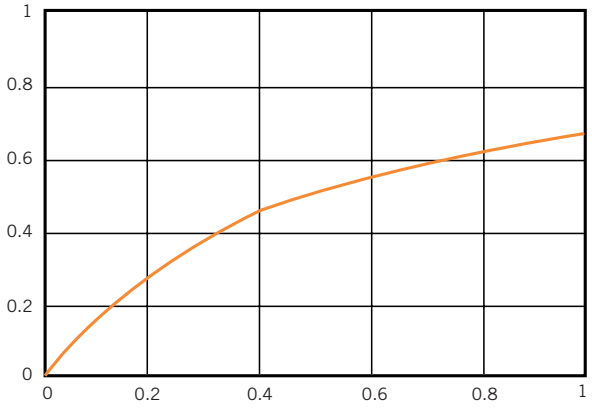
1. Data rising edge triggers CKR_ERRA.
2. Once triggered, next RxC rising edge triggers CKR_ERRB and resets CKR_ERRA.
3. Next RxC falling edge (thus, a half-clock period later) resets CKR_ERRB.

Figure 2 shows the operation of the phase comparator. The transfer function is not linear (**Figure 3**). It corresponds to the expression $V_{OUT} = t/(1+T/2)$, where t is the time from RxD's rising edge to RxC's rising edge, and T is the period of RxC. The nonlinear transfer function is not a limitation, because the system operates around the point at which t=T/2. Moreover, the presence of noise effectively reduces the comparator's gain and smoothes its operation. Thus, the PLL's design must accommodate variations in the comparator's transfer function. **Figure 4** shows a simplified final PLL implementation. The programmable-logic device uses buffering to attenuate the digital switching noise of the device. The op amp's noninverting reference input comes from the filtered symmetrical clock signal; thus, the reference level is exactly centered between the low and high logic-level voltages. The system is thus voltage-independent.

(DI #2540)

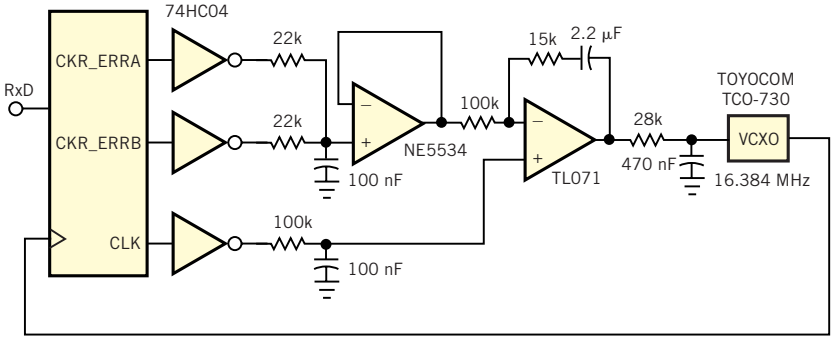
For This Design, Circle No. 501

Figure 3



The phase comparator's output is nonlinear; however, the nonlinearity is inconsequential to the clock-recovery function.

Figure 4



The complete clock-recovery circuit uses a small handful of components.

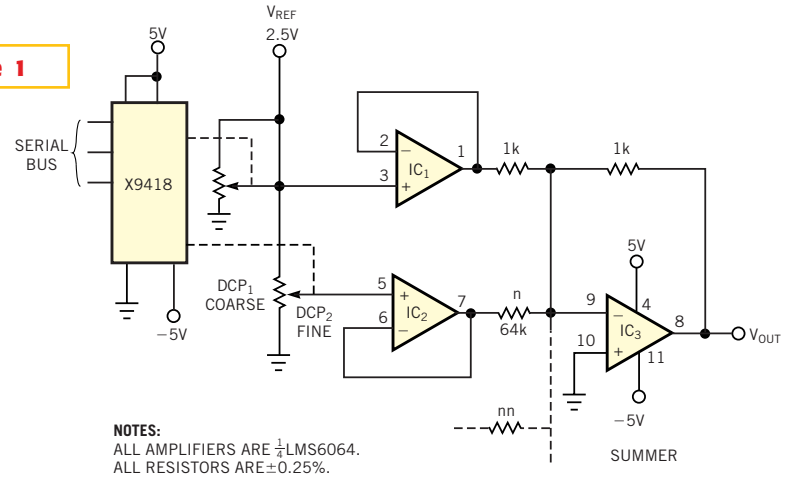
Op amp increases potentiometer's resolution

Chuck Wojslaw, Xicor Inc, Milpitas, CA, and P Gareth Lloyd, Elab Ltd, Stoke-on-Trent, UK

A KEY PARAMETER of digitally controlled potentiometers (DCPs) is the number of taps (n) programmable positions of the wiper. This parameter establishes the resolution in programmable-voltage and -current applications. A number of circuit techniques exist for improving the resolution using one or more DCPs with a given number of taps. The circuit in **Figure 1** has no theoretical limit on increasing the resolution in programmable-voltage applications. The IC_3 amplifier circuit is an inverting summer with weighted input-resistor values. The input voltages to the summer are the programmable output voltages of the DCPs. To reduce the effects of loading, IC_1 and IC_2 buffer the signals from the potentiometer wipers. For an n-tap potentiometer, the input resistors of the summing amplifier are R and nR, providing a programmable output voltage of $0V > V_{OUT} > -V_{REF}(1+1/n)$, with a resolution of $V_{OUT}(\text{smallest}) = -V_{REF}/[n(n-1)]$.

For the dual 64-tap Xicor X9418 DCP and the circuit values shown, the output voltage, V_{OUT}, has n^2, or 4096, programmable values. The full-scale value is 2.5391V, and the smallest programmed voltage is 0.62 mV. You can program the coarse DCP_1 from 0 to 2.5V with a resolution of 39 mV, and you can program the fine DCP_2 from 0 to 39 mV with a resolution of 0.62 mV. This circuit provides the same resolution as a 12-bit D/A converter. Measured data fell within 2 LSBs of calculated values. Adding more potentiometers, buffers, and input resistors provides theoretically unlimited resolution. If you add a third section, the resolution increases to one part in 262,144 (18 bits). You can implement a similar scheme using a noninverting summer circuit. You can use the circuit as a substitute for expensive D/A converters in any application that requires a precise, high-resolution programmable voltage. (DI #2537).

Figure 1



The sky's the limit on resolution in this programmable-voltage circuit.

To Vote For This Design, Circle No. 502

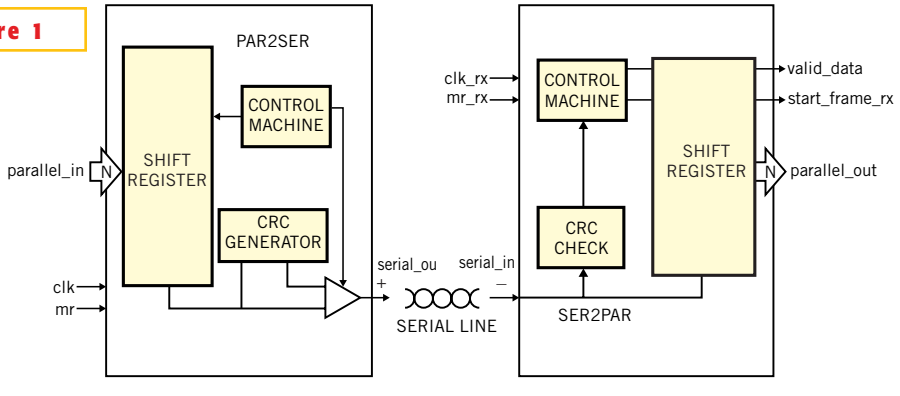
VHDL customizes serializer/deserializer

Antonio Di Rocco, Siemens ICN, L'Aquila, Italy

MANY APPLICATIONS require a multiple-signal exchange among cards through a backplane. Several solutions are available to serialize/deserialize data—from the classic UART to newer low-voltage differential-signaling components. It is sometimes important to have hardware flexibility in

Using VHDL, you can customize a PLD to perform serialization and deserialization.

Figure 1



transferring signals; for example, you can use a PLD to implement a UART-like function to perform a parallel-serial-parallel conversion (**Figure 1**). **Figure 2** shows the frame structure. The idle "1" bits exactly fill the time between the start and stop bits. Assuming $f_{BIT}=1/T_{BIT}$ as the speed of the serial link, the sampling frequency, f_{SH}, for each parallel-input channel is $f_{SH}=f_{BIT}[(N+3)/2]$. With an internal state machine working with a system clock (clk), each eight-clock period corresponds to one bit period, T_{BIT}. The relationship between sampling frequency f_{SH} and the PLD clock is $f_{SH}=[f_{CLK}(N+3)]/4$.

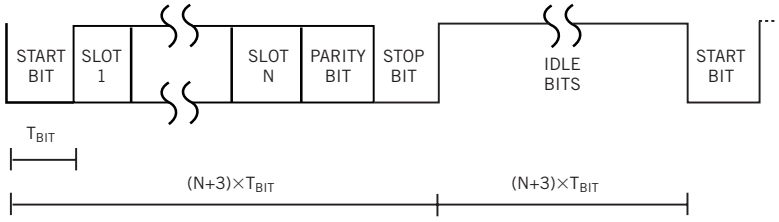
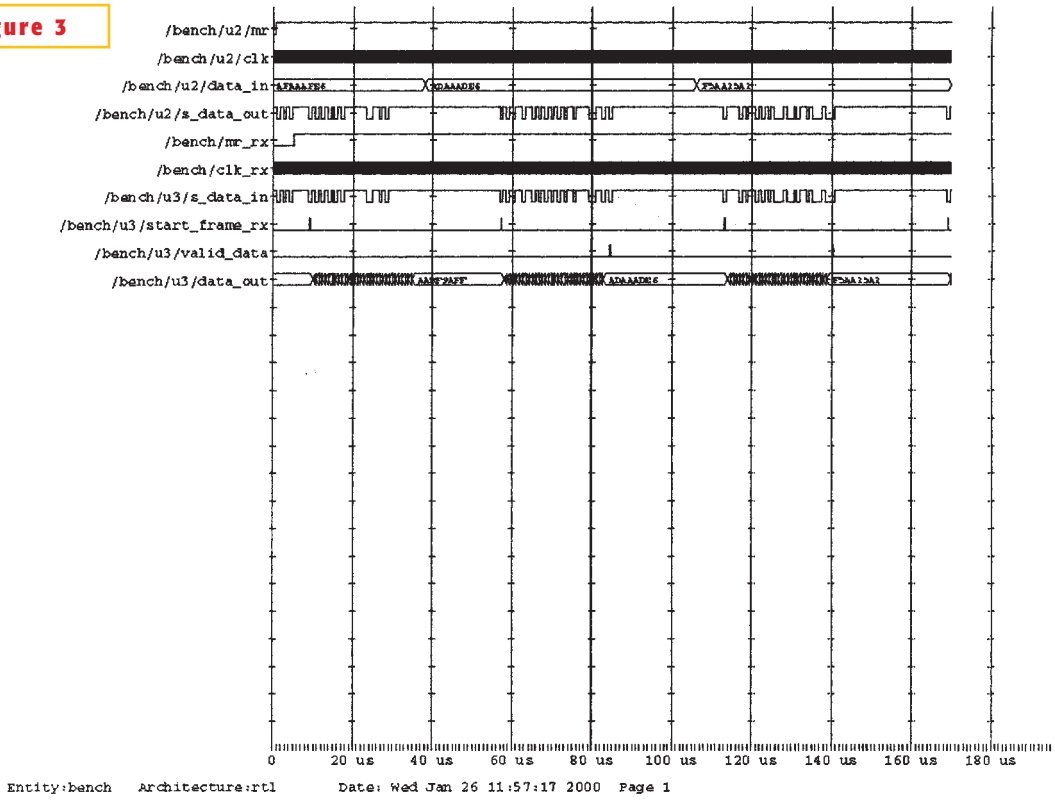


Figure 2 The system in Figure 1 produces this frame structure.

Figure 3



Simulation yields this serialization pattern for 32 parallel-input channels.

The deserializer does not perform a clock-recovery function but works with the nominal clock frequency of the transmitter side. Jitter tolerance on the serial line is related to the number of parallel-input channels. During the time between the start bit and the stop bit, the system tolerates a delay time of $^3/_8T_{BIT}$. Often, system features are related to the maximum speed the serial link allows. To improve the quality of transmission, you can insert a more complex CRC function. The timing simulation shows a serialization of N=32 signals using a sampling frequency f_{SH}=17.8 kHz, which corresponds to a PLD clock f_{CLK}=10 MHz with a speed of 1.25 Mbps on the serial link (**Figure 3**). Note that the serializer's start bit occurs just as the master reset (mr_rx) is deasserted. On the receiver side, the deserializer has its master reset (mr_rx) deasserted while it receives a frame; thus, it starts to sample a wrong frame. Between the two first start_frame_rx pulses, no valid_data pulse exists. The frame-acceptance pulse appears at the end of the second received frame. You can easily customize the VHDL code. It uses the ser2par.vhd and par2ser.vhd component source files. Another VHDL listing has the package source file, in which the constant N designates the number of parallel channels. Finally, a "bench" routine, bench.vhd, runs simulations. You can download the VHDL listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2542. (DI #2542).

To Vote For This Design, Enter No. 503

Sometimes, one capacitor is better than two

Robert LeBoeuf, National Semiconductor, Salem, NH

Many A/D converters use an internal resistor ladder as a two-point differential voltage reference in the conversion. This method demands that these two nodes remain steady. The higher the resolution, the stronger the demand for quiet voltages. **Figure 1** depicts a simplified schematic of the LM985XX reference ladder. **Figure 1a** shows the traditional decoupling scheme; **Figure 1b** shows a proposed scheme. Typically, designers use two capacitors to decouple each reference node—one low-value capacitor and one of higher value because the effective series inductance (ESL) of the smaller capacitor is much lower than that of the larger one. Contrary to tradition, you can eliminate these larger capacitors and replace them with one differential capacitor if you choose the values wisely. Because the difference in the reference voltages, ΔV_{REF}, is important in conversion, this is the delta that is of interest.

Figure 1b shows two common-mode decoupling capacitors, C_1 and C_2, and the differential capacitor, C_3. The current sources, I_1 and I_2, represent the average currents pushing or pulling on the ladder. These currents are generally proportional to the input voltage, V_{IN}, and the sampling frequency, f. If the input voltage is periodic or at least quasiperiodic, then you can choose the decoupling capacitors on the basis of the maximum ripple voltage allowed to appear differentially on the nodes. This ripple specification is based on the permissible output error. The poles and zero of the transfer function are, respectively,

$$\text{pole}_1 = \frac{-1}{2} \bullet \frac{(C_2 \bullet R_{P3} \bullet R_{P1} + C_3 \bullet R_{P2} \bullet R_{P1} + C_1 \bullet R_{P2} \bullet R_{P3})}{[(C_2 \bullet C_3 + C_1 \bullet C_3 + C_1 \bullet C_2) \bullet R_{P2} \bullet R_{P3} \bullet R_{P1}]} \bullet \delta;$$

$$\text{pole}_2 = \frac{1}{2} \bullet \frac{(C_2 \bullet R_{P3} \bullet R_{P1} + C_3 \bullet R_{P2} \bullet R_{P1} + C_1 \bullet R_{P2} \bullet R_{P3})}{[(C_2 \bullet C_3 + C_1 \bullet C_3 + C_1 \bullet C_2) \bullet R_{P2} \bullet R_{P3} \bullet R_{P1}]} \bullet (\delta - 2);$$

$$(\delta << 1)$$

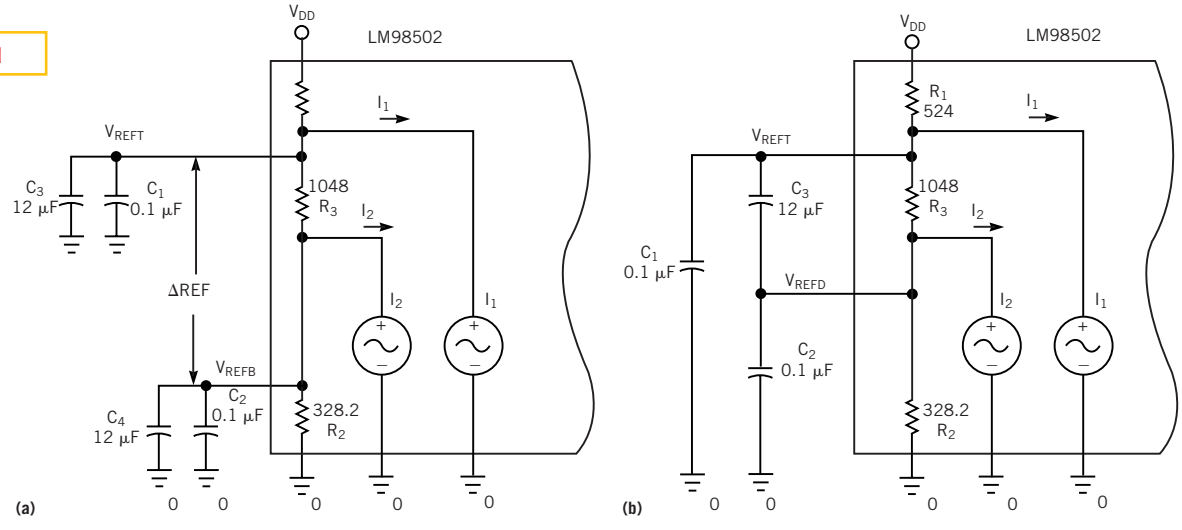
$$\text{ZERO} = \frac{R_2 + R_1}{R_2 \bullet R_1 (C_1 + C_2)}, \text{ and } R_{PK} = \frac{R_n \bullet R_m}{R_n + R_m},$$

where k, n, and m are cyclic permutations of 1, 2, and 3. You can easily see that the pole in the first **equation** is the dominant pole. As long as the zero is sufficiently far from pole_1, then this pole (hence, C_3) determines the roll-off. If C_1 and C_2 are small enough, the zero finds itself well away from pole_1. Using the values in **Figure 1b**, the poles and zero become $\text{pole}_1 = -28.50$ Hz; $\text{pole}_2 = -3.948 \times 104$ Hz; and zero $= 2.48 \times 105$ Hz.

Typically, the current sources have a fundamental frequency equal to the frequency of the analog clamp. The most pessimistic assumption is that the input signal contains all white pixels. This scenario causes the maximum swing in the current sources and the smallest duty cycle. Using this assumption and a current amplitude of 0.6 mA, the capacitor values shown in **Figure 1b** would produce $\Delta V_{REF} = 1.989$ mV. Instead, if you used a pair of 12-µF decoupling capacitors, as in **Figure 1a**, the circuit would produce $\Delta V_{REF} = 3.975$ mV. (DI #2543)

To Vote For This Design, Enter No. 504

Figure 1



More is not always better; the circuit in b provides better decoupling than the one in a.

Inverter offers design flexibility

Nihal Kukaratna, Arthur C Clarke Institute, Katubedda Moratuwa, Sri Lanka

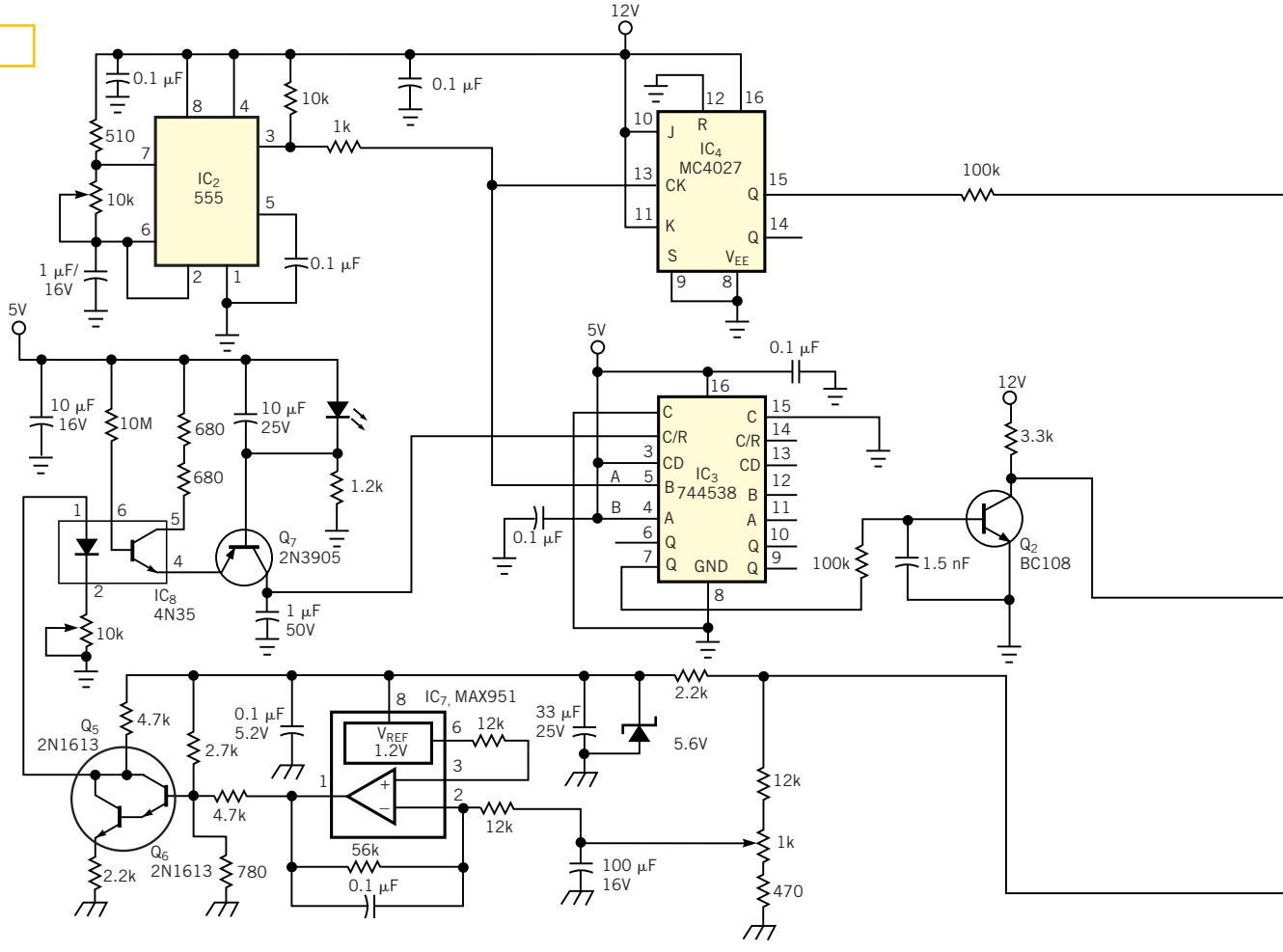
You may occasionally need a substitute for a commercial dc/ac inverter. A typical application is in an uninterruptible power supply (UPS). The circuit in **Figure 1** is a flexible, low-component-count inverter with closed-loop voltage regulation. The advantages of the circuit are that it works from a 12V car battery (or from higher battery voltages with minor modifications), it offers closed-loop voltage regulation, and phase locking with a commercial power supply is possible.

The circuit is designed around a MOS-gate driver family, such as International Rectifier's (www.irf.com) IR215X family (IC_5). This IC drives the gates of power MOSFETs Q_3 and Q_4 through NAND gates (IC_6). A 555-timer-based 100-Hz oscillator (IC_2) feeds the MOS-gate driver's frequency-generation block through a divide-by-two circuit (IC_4). The MOS-gate drivers' low and high outputs drive the power MOSFETs' gates through IC_6. The combination of the MOS-gate driver and the IC_6 NAND gates maintains the necessary deadband to prevent simultaneous conduction of the power-MOSFET pair. A voltage-feedback sample, compared with the 1.2V reference source in a MAX951, IC_7, provides a closed-loop feedback to vary the value of the constant-current source comprising Q_7 and the optoisolator, IC_8. This variable constant-current source varies the monostable output of IC_3, which feeds the IC_6 NAND gates. The feedback system thus maintains the proper pulse width in the gate drivers.

You can easily modify the (square-wave) circuit for a sinusoidal output by

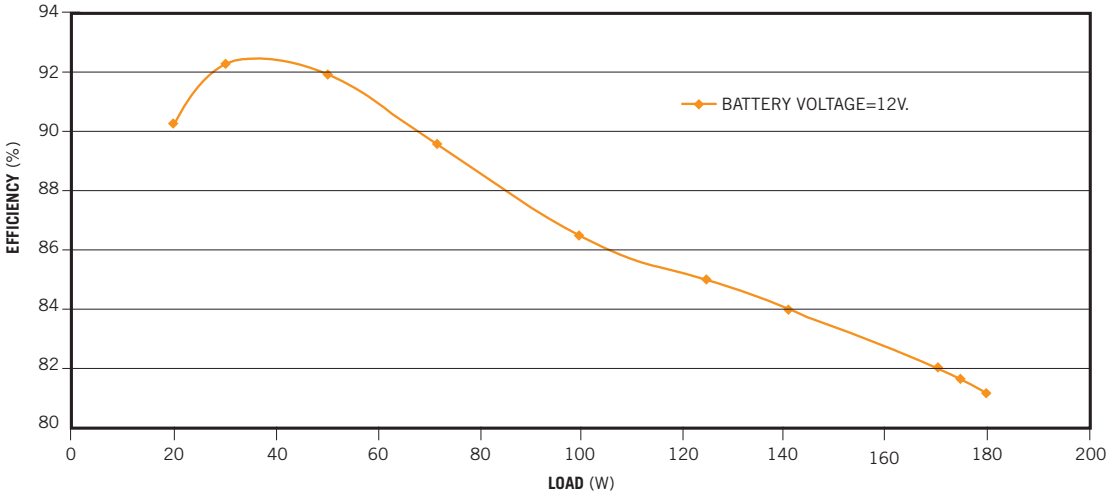
(text continued on pg 174)

Figure 1

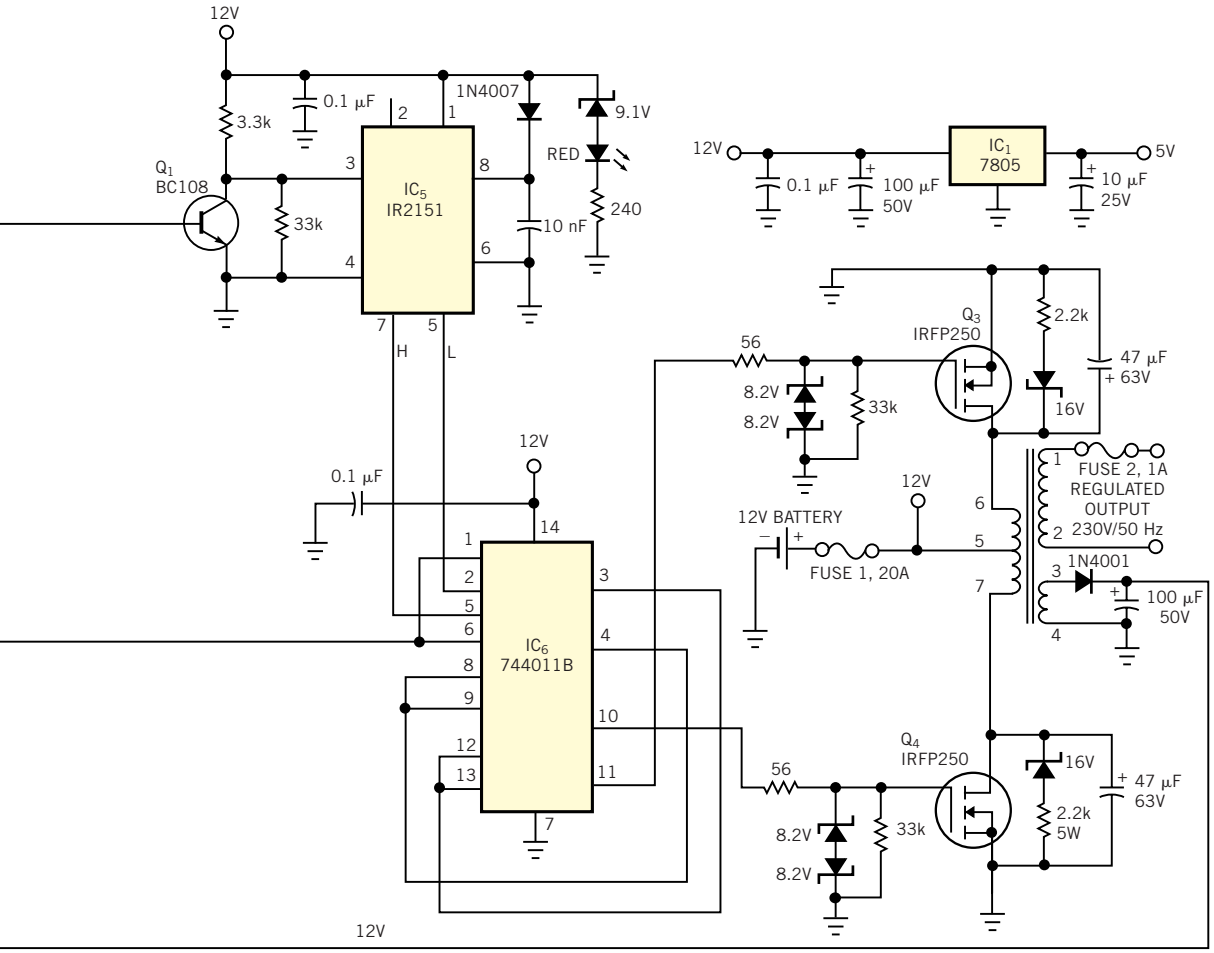


Roll your own dc/ac inverter, using a MOS-gate driver IC.

Figure 2

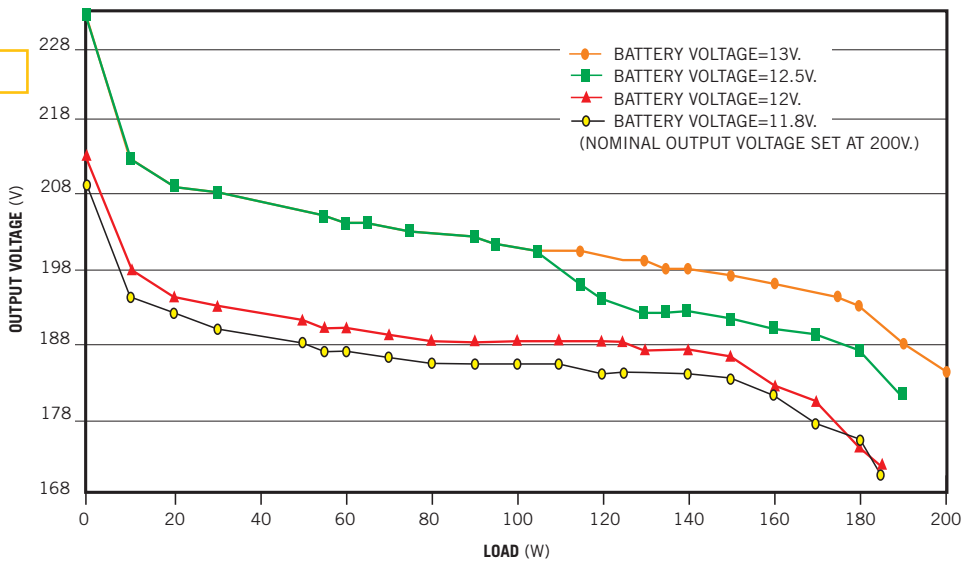


The circuit in Figure 1 maintains reasonably tight regulation for a wide range of loads.



adding a few components between IC_3 and IC_6 (**Reference 1**). For higher outputs, you need change only the battery voltage and the power MOSFETs. The circuit in **Figure 1** is a 200-VA unit. **Figure 2** shows the output-regulation curve for different battery-voltage inputs. **Figure 3** shows the efficiency-versus-load curve. The inverter circuit has 81 to 93% efficiency for loads of 15 to 180W. Using a tape-wound, powdered-iron-core transformer as T_1, you can package the unit in a 100-in.^{3} volume for a 230V, 50-Hz emergency power source. (DI #2538).

Figure 3



Efficiency peaks at nearly 93% for moderate loads.

References

1. Kularatna, N, and P Silva, "New Approach to Sine-Wave Inverters for UPS and Emergency Power Supplies Using Low-Cost MOS Gate Drivers," Proceedings of Power Systems World Conference (PCIM 98), pg 332.

2. N Kularatna and M de Silva, "DSP-Based Sine-Wave Inverters for UPS and Emergency Power Supplies with MOS Gate Drivers," Proceedings of Power Systems World Conference (PCIM 99), pg 296.

To Vote For This Design, Circle No. 505

Single-button lock provides high security

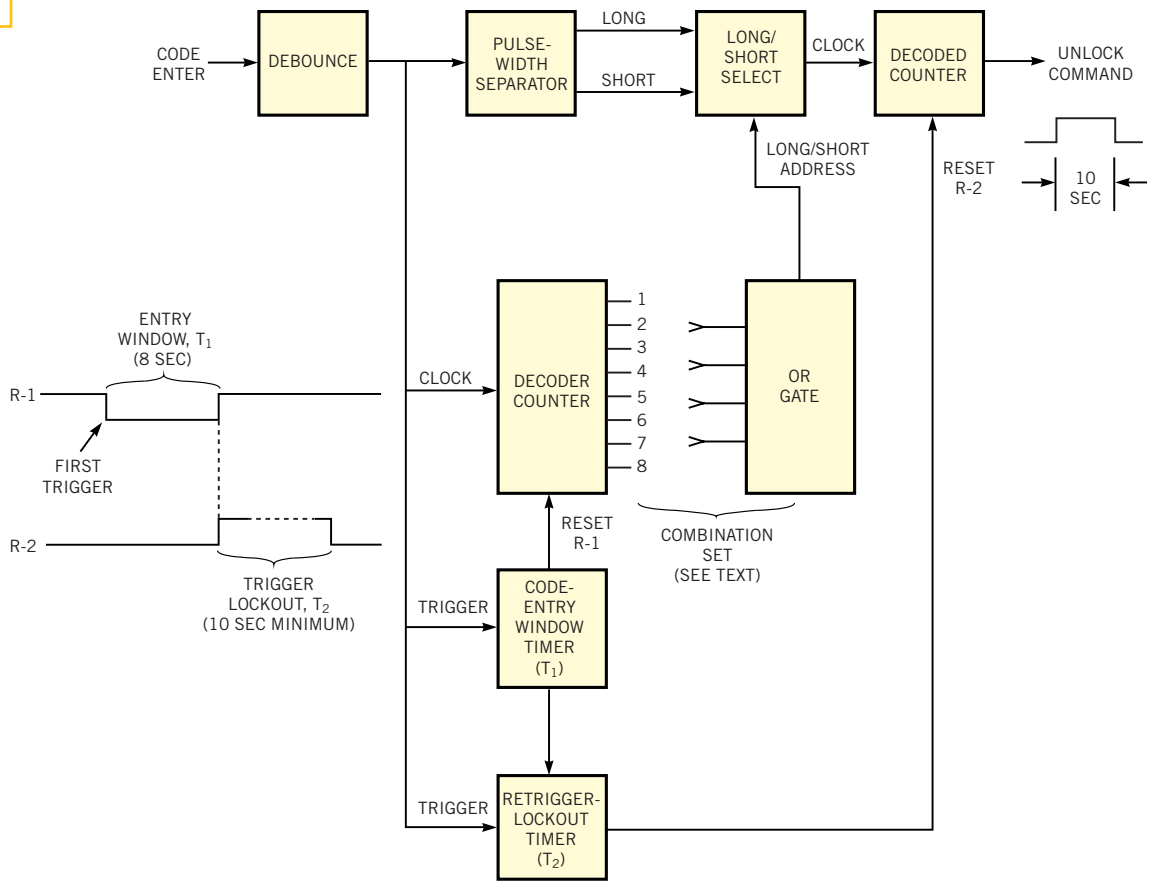
Maxwell Strange, Fulton, MD

FIGURE 1 IS the block diagram of an easily programmed, single-button combination lock. You operate the lock by using a series of short and long pulses from a momentary switch that masquerades as a doorbell button. The circuit uses inexpensive CMOS logic. The retriggerable timer, T2, locks out entries made after the T1 code-entry window, thereby greatly enhancing security. The circuit in Figure 2 operates as follows: The Schmitt-trigger quad NAND gate, IC1, debounces the code-entry switch and, with the aid of simple analog circuitry, produces separate outputs for activation times of less than and more than 0.3 sec. These outputs connect to the select gate, IC5. The initial entry also sets timer T1 to enable the decoded decade counter, IC3. Each entry clocks IC3.

As IC_3 steps through its counts, certain of its output positions represent "short" and connect to IC_4's inputs; unconnected lines represent "long" positions. This coding arrangement sets the combination. Short pulse positions change the address of IC_5 to select the short input pulse; otherwise, IC_5 selects the long pulse input. The short and long inputs, if present in the programmed sequence, produce an output from IC_5. IC_6 counts the outputs and produces an unlock command only if it counts all pulses. The power-on-reset circuit ensures that no compromise of security arises under any conditions after a power outage. The timers are crucial to the high security of the system. You must enter the code within the 8-sec T_1 window. If you make a mistake, you must wait at least 10 sec for T_2 to time out before you make another attempt. If entries occur continuously and less than 10 sec apart, as an intruder might try, T_2 continuously inhibits counter IC_6.

The lock proves to be reliable over several years of use. The circuit in **Figure 2** uses an eight-character combination,

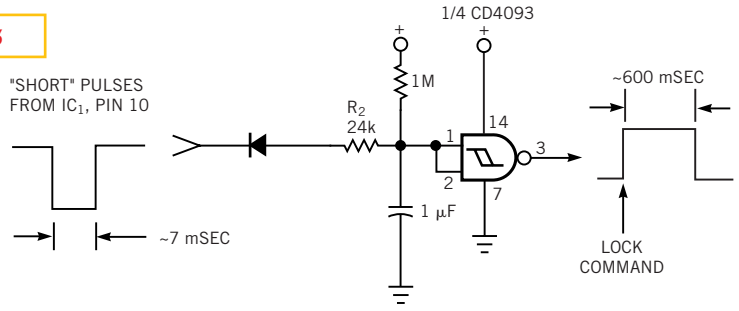
Figure 1



A handful of timers and counters configures a highly secure, single-button combination lock.

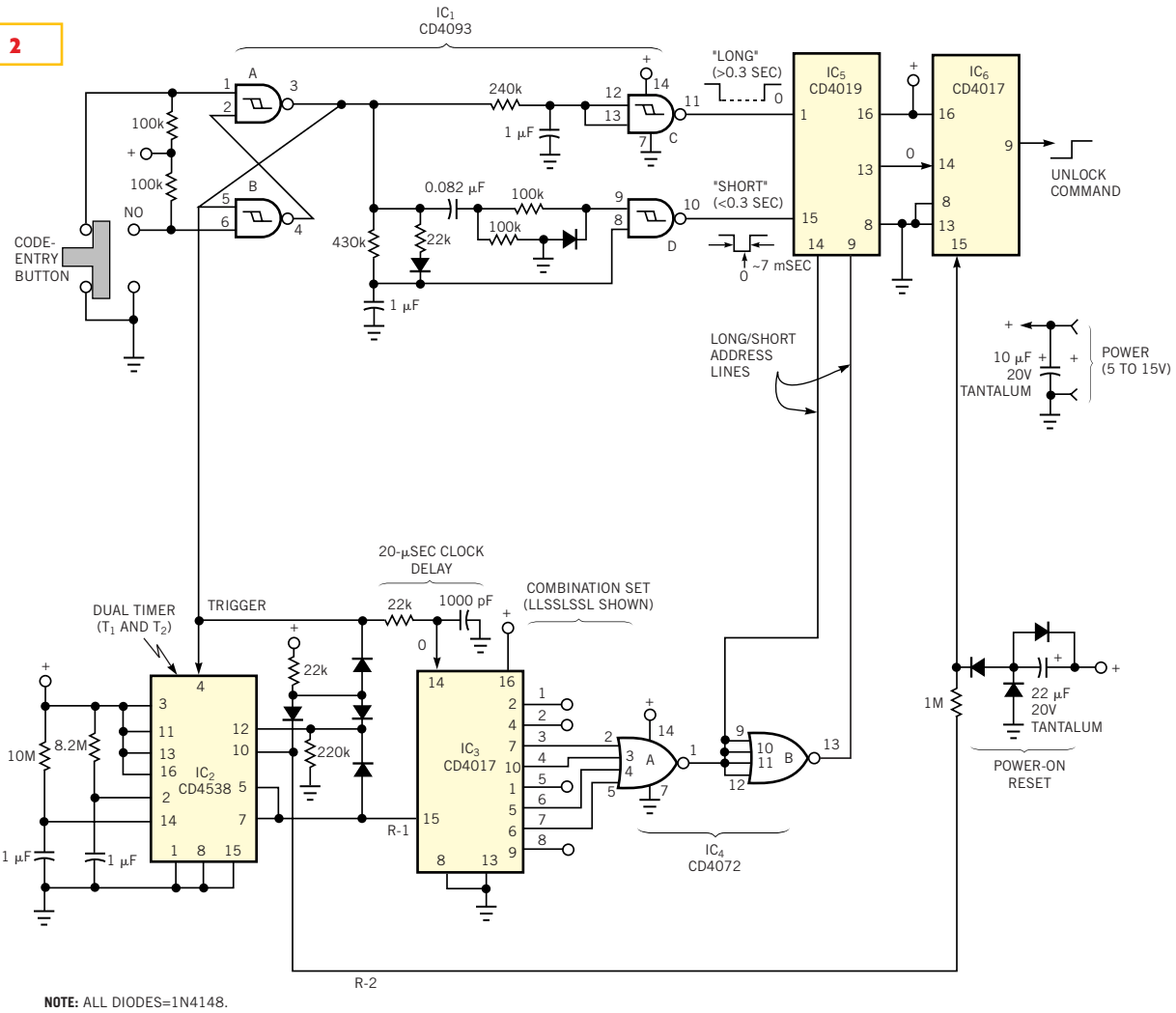
which you can quickly enter. A short pulse is a quick jab to the button; a long pulse is only slightly longer. A shorter sequence would also be secure; you can implement a shorter code by simply taking the unlock pulse from a lower count on IC_6. IC_6's output returns low after 10 sec when T_2 resets. If desired, you can generate a lock command, which need not be secure, by adding the simple circuit in **Figure 3**. (DI #2327).

Figure 3



You can generate a lock command with this additional circuit by rapidly entering four or more short pulses.

Figure 2



NOTE: ALL DIODES=1N4148.

You program your combination by hard-wiring the IC_3-IC_4 output-to-input connections, LLSSLSSL, where L and S are long and short inputs, respectively, in this example.

design ideas

Edited by Bill Travis and Anne Watson Swager

Current source has high output impedance

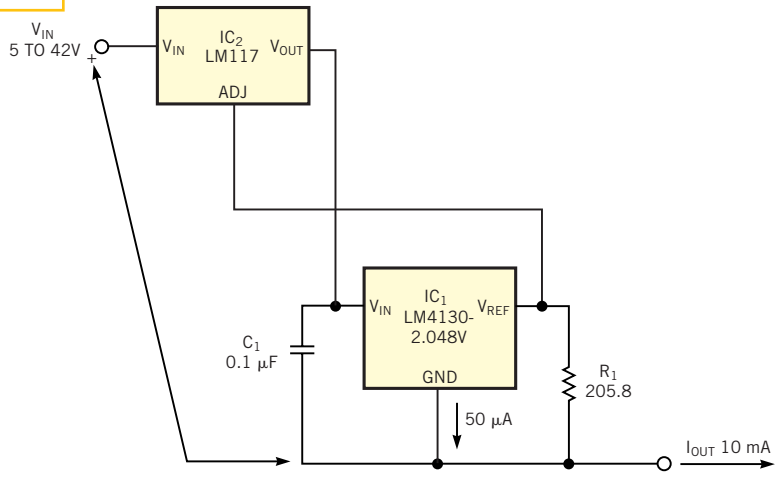
Clayton B Grantham, National Semiconductor, Tucson, AZ

A reference current source needs high accuracy, low temperature drift, and high output impedance. Available IC current sources come with some of these features. However, at current levels greater than 1 mA, their output impedance decreases to less than 10 MΩ. **Figure 1** shows a composite 10-mA current-source configuration that has a compliance voltage of 5 to 42V, a set-current error of less than 1%, a temperature drift of less than 45 ppm/°C, and an output impedance of greater than 100 MΩ. One application of this accurate 10-mA current source with high impedance is as a 4- to 20-mA current-loop calibration reference that has a maximum loop voltage of 40V and that operates over the −40 to +85°C industrial-temperature range.

IC_1's V_{REF} output and R_1 set I_{OUT}. I_{OUT} equals V_{REF}/R_1 plus IC_1's bias current, which is typically 50 µA. This bias current is a small error of less than 0.05% at I_{OUT}=10 mA and changes by only 10 µA over −40 to +85°C. Changes in V_{REF} and R_1 over temperature more directly contribute to I_{OUT}'s accuracy and temperature coefficient. Inexpensive resistors with 0.1% tolerances and 25 ppm/°C drift over temperature are common. The LM4130 has a V_{REF} grade of 0.05% and 20 ppm/°C over −40 to +85°C. Thus, the worst-case current-setpoint error is within 0.15% and 45 ppm/°C, which results in an upper limit on the current error over temperature of 0.45%, or 45 µA.

The circuit's high performance would degrade if its output impedance were not very high. The circuit's output impedance is an undesirable parasitic in parallel with I_{OUT}. The product of IC_1's line regulation times IC_2's line regulation determines the current source's output impedance. IC_2's 1.2V output sets IC_1's input-minus-output to a near constant. IC_1's V_{REF} over R_1 sets I_{OUT}, which is twice removed from V_{IN}. V_{REF} of IC_1 has an overtemperature line regulation of 500 ppm/V, and IC_2's output has an overtemperature line regulation of 350 ppm/V. Output impedance is greater than 300 MΩ, which is good, if you calculate it using only the line-regulation effects. Although line regulation is the dominant source of output impedance, other thermal errors beyond line regulation degrade the potential of keeping very high output impedance over temperature. Bench measurements made on the composite showed output impedance greater than 300 MΩ at 25°C and 100 MΩ over −40 to +85°C. (DI #2544)

Figure 1



This composite configuration of a voltage reference and a voltage regulator has high output impedance.

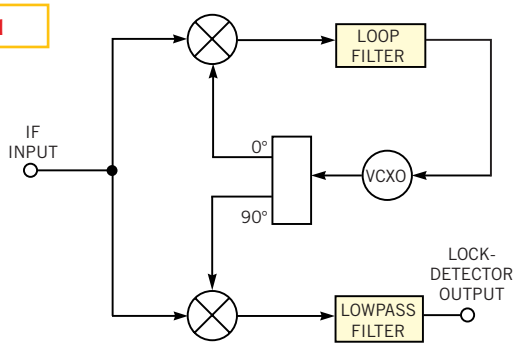
To Vote For This Design, Enter No. 364

Novel method detects lock in Costas loops

MR Raghavendra, ISRO Satellite Centre, Bangalore, India

IN THE WELL-ESTABLISHED lock-detection scheme for conventional PLLs, the VCO or VCXO local oscillator splits the output into 0 and 90° signals (**Figure 1**). The incoming IF signal mixes with the 0 and 90° signals to perform phase locking and lock detection, respectively. This method gives unambiguous results even under noisy conditions.

Figure 1



A conventional PLL uses 0 and 90° signals for phase locking and lock detection.

For Costas-loop systems for binary-phase-shift-keying demodulation, the demodulation process uses both the 0 and the 90° signals. Thus, these signals are unusable for lock detection. Instead, simple Costas-loop systems generally use level detection as the lock-detection method (**Figure 2a**). This straightforward method involves detecting and comparing the demodulated I and Q signal levels. In the locked condition, the I-data level is more than the Q-data level. In this condition, the Q-data level equals 0. In the unlocked condition, the two data levels are equal. The scheme detects this difference to indicate lock. Unfortunately, this method can give ambiguous results under noisy conditions.

You can adopt a similar method in modified Costas loops for QPSK (quadrature-phase-shift-keying) demodulation (**Figure 2b**). In the unlocked condition, a beat-frequency component is present along with the data. In the lock condition, this beat-frequency component disappears. Hence, the detected I and Q data levels are higher in the un-

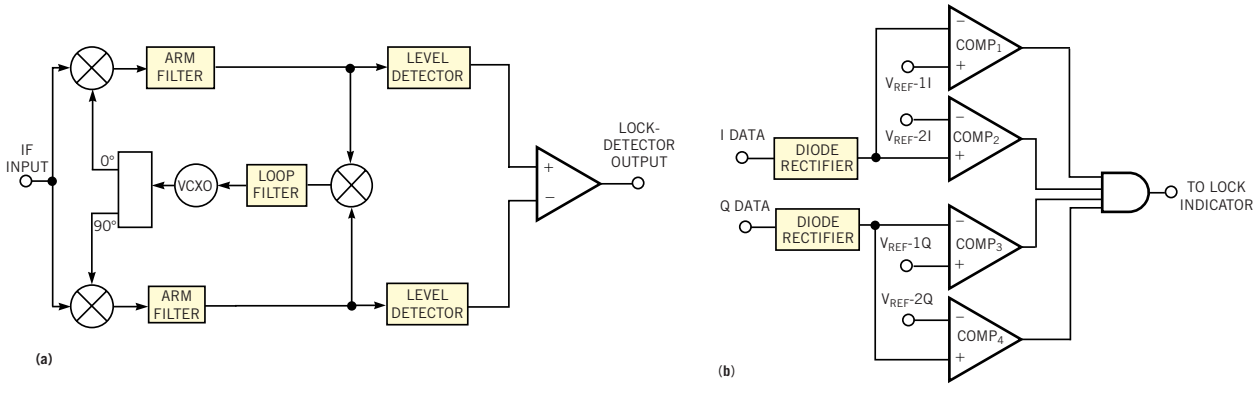
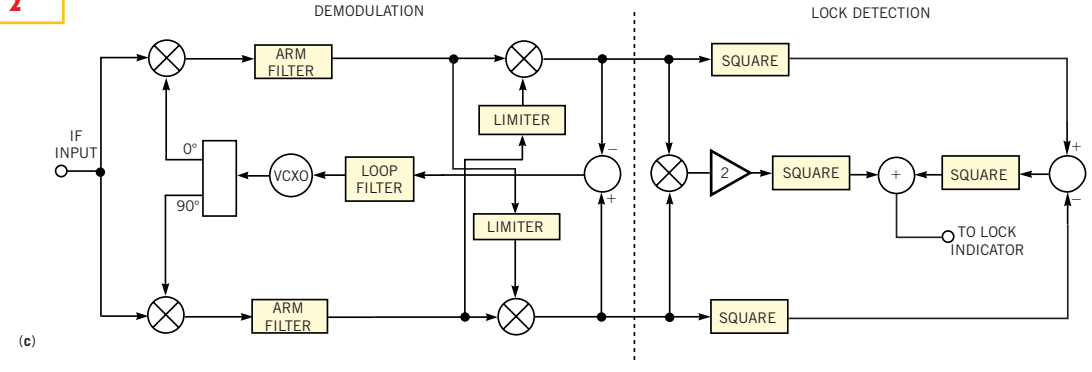


Figure 2



Various lock-detection schemes for Costas-loop systems suffer from problems. Level-detection (a) and a similar method (b) can give ambiguous results under noisy conditions. A coherent method (c) gives unambiguous results but is complex.

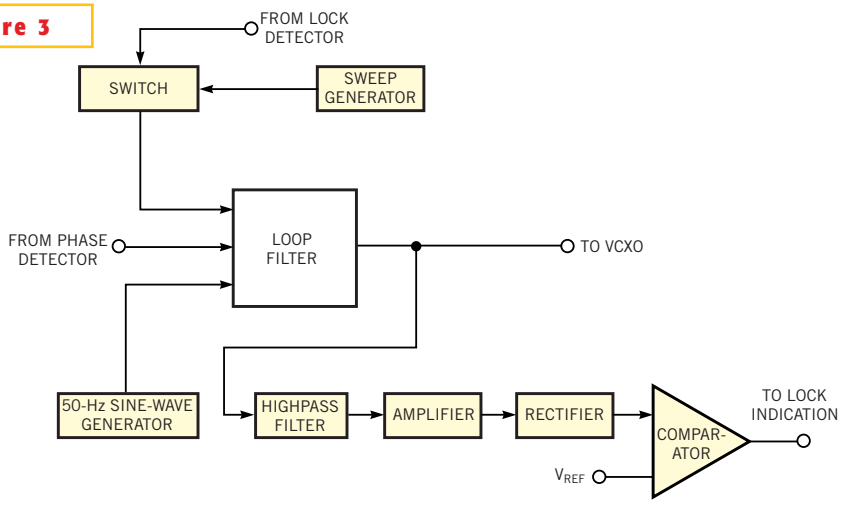
locked condition than in the locked condition. By detecting this difference in levels, you can determine the locked condition. This method is not free of ambiguities under noisy conditions because the detected levels change with levels of noise. Thus, reference-voltage settings for level comparison are critical. At some levels of noise, level comparison becomes almost impossible; under such conditions, this method of lock detection fails.

The coherent method illustrated in **Figure 2c** gives ambiguity-free lock detection for modified Costas-loop systems. Unfortunately, this method is complex, and the hardware realization is as complex as the demodulation process itself.

Figure 3 presents a new method of lock detection with much less hardware complexity. You can adopt this level of detection for any PLL system. The underlying principle of this method is that when the PLL is in lock, it tracks and neutralizes all of the low-frequency modulations within the loop. Therefore, if the system introduces a low-frequency, low-level modulation, such as at the loop-filter input, into the loop, then when the loop is locked, the low-frequency signal at the output of loop filter disappears. The presence or absence of the signal at the output of the filter provides unambiguous lock detection.

The circuit in **Figure 3** implements this method in a 375-MHz QPSK demodulator with a loop bandwidth of 10 kHz. This method involves injecting a low-level, 50-Hz sine wave at the input of the loop filter, which is an active filter, and injecting a sweep signal of 2 Hz, which speeds lock acquisition. If you use a passive filter in place of the active filter, you can apply the signal at the input of VCXO after the loop filter, that is at the input of the amplifier after the passive loop filter.

Figure 3



An improved lock-detection scheme introduces a low-frequency, low-level modulation into the loop. When the loop is in lock, this low-frequency signal at the output of the loop filter disappears.

The scheme taps the output of the loop filter and filters out the sweep signal. The scheme then amplifies the 50-Hz signal to compensate for attenuation in the loop filter and rectifies the signal to give a dc voltage to indicate its presence. When the loop is in lock, the 50-Hz signal disappears, and the rectifier output is zero. The lock-indication output also cuts off the sweep. The 50-Hz signal remains connected to the loop filter even during lock. The presence of this signal at the input of the loop filter does not degrade the demodulator performance because the level is low. You choose 50 Hz as the injected signal frequency because 50 Hz is considerably higher than the sweep-signal frequency and considerably lower than the loop bandwidth.

The performance of this method is effective even under very noisy conditions, such as when the input's Eb/No (energy-per-bit-to-noise) figure is 3 dB. Previous methods give ambiguity-free results only at Eb/No levels of 7 or 8 dB. (DI #2545)

To Vote For This Design,
Enter No. 365

Coax connectors make low-cost test pieces

Richard M Kurzrok, RMK Consultants, Queens Village, NY

You can readily construct low-cost test pieces using coaxial panel jacks without pc boards or enclosures (**Figure 1**). Some engineers and technicians occasionally use this construction technique, but the versatility of the technique is not well-known. You can design and construct these low-cost test pieces for a variety of passive circuits using tee, pi, el, or bridged-tee networks. Other simple circuits are also amenable to this type of construction. These circuits include highpass filters, minimum-loss pads, transformers, amplitude equalizers, and feedthrough terminations. The technique applies to all commercial, industrial, and educational breadboard units that require a quick-and-dirty implementation. The test pieces are useful for low-cost laboratory experiments on passive circuits at reasonably high frequen-

cies. The quasi-open "enclosure" for the test pieces provides good visibility of components and solder joints.

The 50Ω BNC is a popular coaxial panel connector, and significant cost differences exist between military and commercial versions. Some panel jacks come with number 3-48 tapped holes. You can drill these holes out for number 4-40 clearance. This application accommodates minor differences in panel-jack hole spacing.

Using a single panel jack with a solder lug, you can construct a one-port circuit, such as a termination or standard mismatch. You can fabricate a two-port circuit using two panel jacks mechanically secured to each other with four 4-40 stainless-steel machine screws and ancillary hardware, such as hex nuts, lock washers, and solder lugs. The four screws provide electrical-ground continuity between the two panel jacks. You create simple passive circuits by soldering components, such as resistors, inductors, and capacitors, to the panel jacks' center conductors and the ground lugs. You assemble three screws to the panel jacks before soldering and add the fourth screw after soldering.

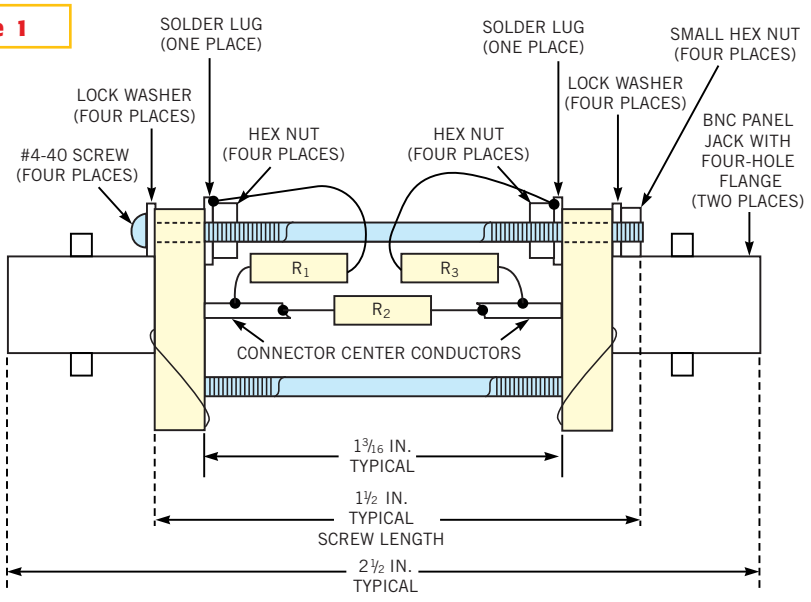
The test piece in **Figure 1** is a pi-section fixed attenuator. This 6-dB, 50Ω attenuator uses ¼W composition and carbon-film resistors with 5% tolerances. The nominal value of series resistor R_2 is 36Ω, and the shunt resistors, R_1 and R_3, are nominally 150Ω. From 2 to 150 MHz, the measured attenuation is 6±0.2 dB. At 50 MHz, the measured attenuation is 6.4 dB.

You can also create a dc block using this construction technique by placing a series capacitor between the panel-jack center conductors. The 0.1-µF CK05 capacitor has a tolerance of 10%. From 300 kHz to 30 MHz at a 50Ω impedance, the measured insertion loss is less than or equal to 0.1 dB. At 50 MHz, the measured insertion loss is 0.2 dB.

You can also achieve a pi-section low-pass filter with a 9.6-MHz cutoff frequency at a 50Ω impedance level using a series inductor and two shunt capacitors. You can characterize the filter as a single constant-K section using image parameters or as a three-pole Butterworth unit using modern network theory. Shunt input and output capacitors are 330-pF polypropylene units with 5% tolerances. A 1.66-µH series inductor comprises 18 turns of number 26 magnet wire wound on a Micro Metals T37-2 toroid. The estimate of the inductor's unloaded Q exceeds 100. **Table 1** shows the measured test data.

When the frequency exceeds 50 MHz, alternative design techniques are more appropriate (**Reference 1**). Similar designs are achievable using other panel jacks, such as threaded-N-connector and N types. The use of nut plates can extend the technique to three- and four-port circuits. When radiation is a problem, wrapping copper-foil adhesive tape around the four machine screws provides partial shielding. (DI #2548)

Figure 1



NOTES:
R_1 AND R_3 SOLDERED TO ONE CONNECTOR CENTER CONDUCTOR AND ONE SOLDER LUG.
R_2 SOLDERED TO TWO CONNECTOR CENTER CONDUCTORS.

You can easily construct a pi-circuit fixed attenuator using two BNC panel jacks, series resistor R_2, and shunt resistors R_1 and R_3.

TABLE 1–MEASURED TEST DATA

Frequency (MHz)	Insertion loss (dB)
2.5	Less than 0.1
4	0.1
5	0.2
6	0.35
7	0.65
8	1.25
9	2.2
9.6	3.0
10.5	4.2
12	6.7
15	11.8
20	19.5
25	25.5

REFERENCE

1. Kurzrok, RM, "Low-cost prototyping for VHF Circuits," *Electronic Design*, Oct 18, 1999, pg 91.

TO VOTE FOR THIS DESIGN, ENTER NO. 366

Transistor junction monitors remote temperature

Matt Smith, Analog Devices, Limerick, Ireland

THE CIRCUIT IN **Figure 1** uses a regular, general-purpose transistor as the sensing element and a thermal-diode monitor IC to measure temperature. The ADM1021 usually connects to an on-chip thermal diode on a CPU die itself, but in this application the sensor connects to a regulator low-cost discrete npn transistor, the 2N3904. The circuit ties the transistor base and collector together to form a two-wire sensor.

A technique known as ΔV_{BE} sensing forces two currents through the base-emitter junction. By measuring the differential voltage across the junction, you can accurately determine the junction temperature. This technique avoids the problems associated with thermal measurement using thermistors or thermocouples. The output from the ΔV_{BE} sensor is approximately 2.2 mV/°C. The ADM1021 amplifies this signal and converts it to digital format.

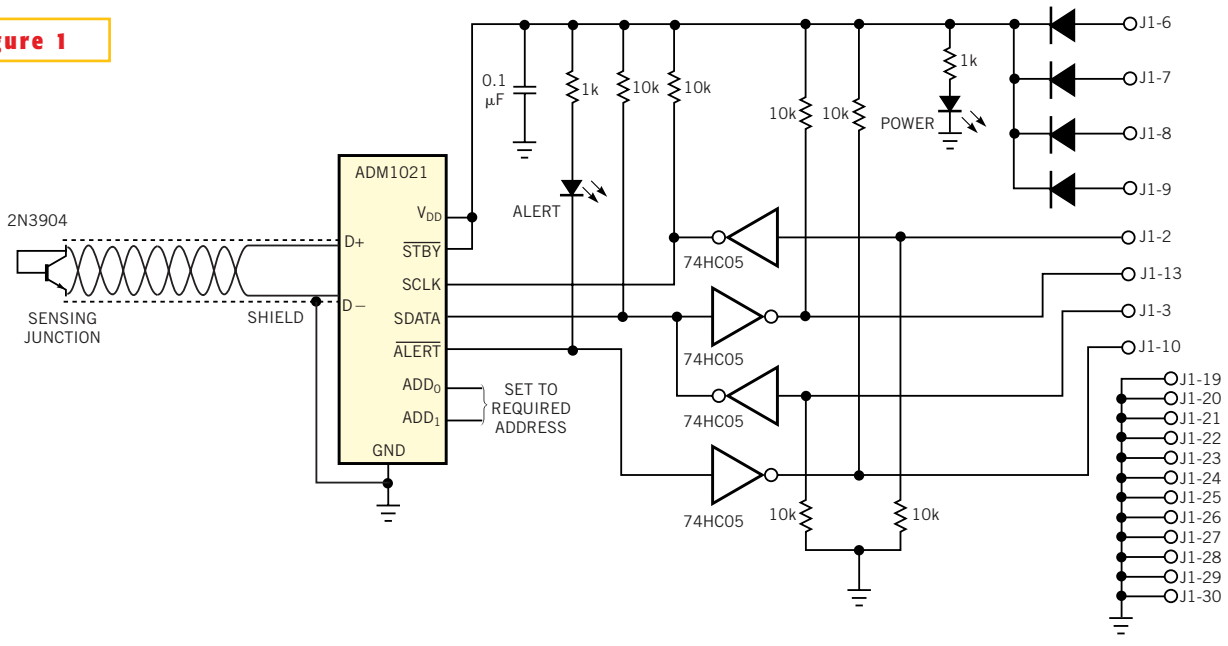
The circuit features a simple two-wire SMBus or I^2C interface, which enables simple communication with a µC or µP. This case achieves communication using three wires on a standard PC Centronics printer port, which makes for a simple and convenient, PC-based temperature-measuring system. A further advantage of the circuit is its inherent power dissipation of 500 µA. This low level enables the circuit to obtain all its power directly from the Centronics port. A graphical-user-interface software application reads the data over the Centronics port and displays the temperature in a strip-chart format or logs readings to an Excel spreadsheet. This application is useful for environmental temperature recording.

Four unused lines on the parallel port power the circuit. Diodes provide isolation to prevent bus contention if any of the lines inadvertently drive low. A 74HC05 buffers the data and clock lines. The clock line is unidirectional, and the data line is bidirectional.

The thermal-diode monitor contains all the necessary circuitry to force differential currents through the sensing transistor. It also contains the amplifiers, scaling circuitry, and a precision ADC to convert the small differential voltages into temperature data. The remote-sensing transistor connects via a shielded twisted pair. The shielded twisted pair is necessary only in electrically noisy environments. The circuit can accommodate temperatures of −128 to +128°C, but in practice the usable range is more limited. The ADM1021 also contains both high- and low-limit registers and has an alert, or alarm, output. If the circuit senses a temperature that exceeds a programmed temperature limit, the IC activates the alarm output, which can drive a buzzer or warning beacon via a buffer.

Analog Devices (www.analog.com) manufactures the ADM1021 and provides PC software for Win95 or Win98 that you can use to monitor and graphically plot the recorded temperature in real time in a moving-strip-chart format. The ADM1021 also monitors its own local temperature as well as the remote-transistor temperature. You can also record temperatures and store them in an Excel format for later analysis. The recording frequency is programmable from a maximum rate of eight recordings per second. (DI #2547)

Figure 1



Using a regular, general-purposes transistor and a thermal-diode-sensor IC, you can monitor remote temperature and display it on your PC.

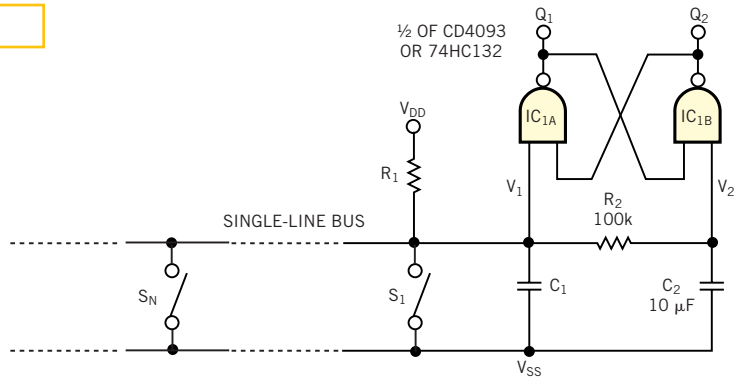
Multiple remote points control on/off switch

Tom Hornak, Portola Valley, CA

Sometimes, you need to remotely turn on or off a two-state system, such as a light, from multiple points. You could connect simple pushbutton switches in parallel to a single-line bus. However, if the bus simply controls a toggle flip-flop, the system must know its current state to positively ensure the new, desired state. For example, if you want to make sure the light is on, you must have visual or electrical feedback via a second line before deciding whether to toggle the flip-flop. Also, bounce-free pushbutton switches are necessary.

The circuit in **Figure 1** achieves positive state control according to the length of time you activate any pushbutton switch from S_1 to S_N. An activation of a few milliseconds, for example, guarantees that the system is set to one state. An activation of 1 sec, for example, guarantees that the system is set to the other state. No visual or electronic feedback is necessary.

Figure 1

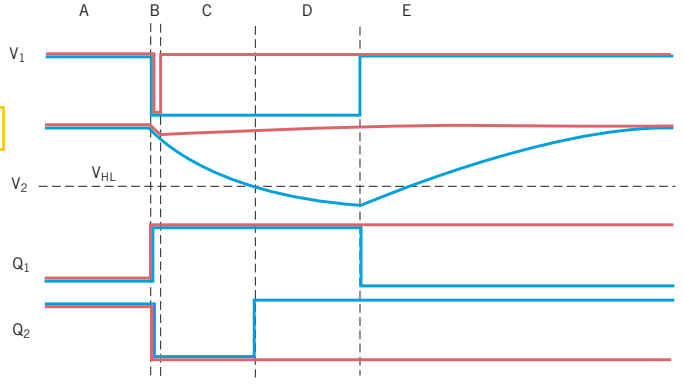


NOTE: THE VALUES OF R_1 AND C_1 DEPEND ON THE BUS CHARACTERISTICS.

A short activation of any one of the pushbutton switches, S_1 to S_N, locks the output state to Q_1 high and Q_2 low. A long activation sets the circuit to the opposite state: Q_1 low and Q_2 high.

The circuit is a NAND R/S latch comprising two simple CMOS Schmitt trigger NAND gates. You can use half of a CD4093 or a 74HC132. **Figure 2** depicts the signal on the circuit's individual nodes. The red lines correspond to a pushbutton's "short" activation; the blue lines correspond to a "long" activation.

Figure 2



The red line corresponds to a short-activation time in section B; the blue line corresponds to a long activation time.

In **Figure 2**, in time section A, all pushbutton switches are off, both V_1 and V_2 are high, and the Q_1 and Q_2 outputs are holding the last entered state. The **figure** shows this arbitrarily as Q_1 low and Q_2 high, but the reverse state could also be true. When one of the switches closes (section B), V_1 goes low, but the delaying action of R_2 and C_2 keeps V_2 high. This action turns Q_1 high and Q_2 low, regardless of the previous state in section A.

If you push the button for a short time (red line in section C), V_1 returns high before V_2 can approach gate IC_{1B}'s high-to-low hysteresis threshold, V_{HL}. The circuit remains locked with Q_1 high and Q_2 low (red lines in sections C to E). If you push the button for a long time, V_1 stays low keeping Q_1 high (blue lines in sections C and D). C_2 gradually discharges through R_2 (section C) until V_2 crosses IC_{1B}'s low-hysteresis threshold, V_{HL}, at the end of section C. As a result, Q_2 goes high. While the push button is still depressed (section D), C_2 discharges even more, keeping Q_2 high. When you release the pushbutton (section E), V_1 returns high. Now, because Q_2 is also high, Q_1 goes low. C_2 charges up again via R_2, preparing the circuit for the next activation.

Thus, a short pushbutton activation locks Q_1 high and locks Q_2 low. A long activation locks Q_1 low and Q_2 high but only after you release the pushbutton. It is important that contact bounce or reflections from the nonterminated bus line do not influence Q_1 and Q_2's final state.

Because the flip-flop's loop delay is negligible, the time required to completely discharge C_1 and the bus line's stray capacitance via the bus line's series resistance determine the minimum pushbutton-activation time. This time never exceeds 1 msec. The maximum short-activation time is the time over which C_2 discharges to 90% of V_{DD}, which still keeps V_2 safely above V_{HL}. For the values of R_2 and C_2 in **Figure 1**, this time is 110 msec. The minimum long-activation time is the time necessary to discharge C_2 below VHL. For a worst case of

$V_{HL}=V_{DD}/2$ and for the given values of R_2 and C_2, this minimum time is 690 msec. There is no maximum activation time. Thus, a wide tolerance exists for both the short and the long activation times, and a large margin exists between the two.

C_1 bypasses any noise that the bus may pick up from its environment. You choose the value of R_1 to keep the bus line high in spite of leakage. R_1 also needs to recharge C_1 and the bus-line capacitance to V_{DD} within a few milliseconds after you release the pushbutton. Thus, the values of R_1 and C_1 depend on the particulars of the bus. (DI #2549)

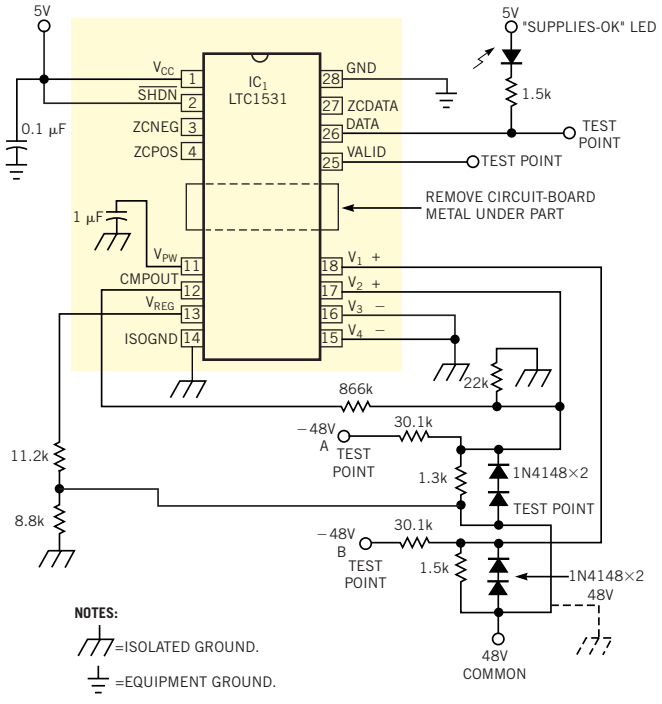
To Vote For This Design,
Enter No. 368

Comparator detects failed telecomm supplies

Richard Markell, Linear Technology Corp, Milpitas, CA

You often need to know when a telecomm power supply's output drops below its nominal value. The dropout generally indicates a failure and may dictate replacement of the supply or some other form of system maintenance. The circuit in **Figure 1** uses an isolated comparator to monitor two 48V telecomm supplies (of either polarity). The comparison occurs on the 48V side of the isolation barrier. The data travels across the barrier inside the package to the output of the comparator, where a µP or another system monitor can check it. IC_1, an LTC1531 self-powered, isolated comparator, performs the isolated-comparator function. The IC has an internal capacitive barrier that provides 3000V rms of isolation between the comparator's inputs and outputs. The part provides UL-rated comparisons without an isolated supply or cumbersome optoisolators. The comparator's power and output data traverse the capacitive barrier.

Figure 1



A comparator with an isolation barrier forms the heart of a telecomm power-supply monitoring system.

The two power supplies to be monitored connect to the −48 A and −48 B points; the 48V returns connect to the "Common" input (not to isolated ground). Resistor dividers attenuate the −48V inputs; the attenuated voltages connect to the dual comparator at V_1 and V_2. The V_{REG} pin of IC_1 provides a 2.5V regulated output, and the voltage divider consisting of the 11.2- and 8.8-kΩ resistors provides approximately 1.1V to the Common point for the 48V supplies. Connecting V_3 and V_4 to isolated ground makes the trip point a negative voltage set by the voltage divider at approximately −1.1V. The series-connected 1N4148 diodes act as crude clamps on inputs V_1 and V_2. Clamping the inputs is necessary because the comparator function is $V_1+V_2>V_3+V_4$. If the inputs were not clamped, a high voltage on one input would allow a low voltage on the other input to go undetected. The 866- and 22-kΩ resistors provide a small amount of hysteresis to stabilize the output for slow-moving inputs.

When the inequality $V_1+V_2<V_3+V_4$ is false (that is, the sum of the power-supply voltages when attenuated is greater than the sum of the reference voltages), the comparator sends a signal across the isolation barrier such that the Data output goes low and the LED turns on. (Note that the sense of the inequality is reversed, because you are sensing negative voltages.) The voltages in the circuit are such that when the sum of the two voltages at −48 A and −48 B are approximately −72V, the inequality is false, and the "Supplies OK" LED turns on. Thus, if one supply is "good" at −48V, the other supply is considered "bad" if it falls below approximately −24V. (DI #2541)

To Vote For This Design,
Enter No. 369

design ideas

Edited by Bill Travis and Anne Watson Swager

Two buttons provide safe start

Vincent Himpe, Alcatel Microelectronics, Desselgem, Belgium

THE CIRCUIT IN **Figure 1** provides a safety interlock that checks the actuation of two pushbuttons before enabling a relay. When you push both buttons, the circuit actuates the relay. At that point, you can release one of the switches without the relay's switching off. The circuit was intended to lock out the engine of an underwater propulsion unit. When handling such units on the surface, a person might accidentally press the actuator switch, which is mounted inside the handles of the propulsion unit. ANDing two switches makes the unit safer but requires two hands, a situation that is sometimes unsuitable. Using the circuit in **Figure 1**, you need two hands to start the engine; once the engine is running, one hand is sufficient to keep it running. When you press no switch, the engine shuts off. The circuit uses a simple 555 timer, IC_1 in a unique way. In this circuit, the 555 serves as a window comparator, followed by a memory element.

Actuating one pushbutton puts the input voltage between the two levels of the comparator and thus has no effect on the state of the 555's internal flip-flop. Only when you press two buttons does the voltage go above the high trigger level and set the flip-flop in the 555. Releasing one button brings the voltage back to a level inside the window and has no effect on the state of the flip-flop. Releasing both buttons brings the input level to ground, thus below the low trigger level, and resets the output. Because the 555 derives its internal levels from a resistive divider, the supply voltage has no influence on the behavior of the circuit. The reset input of the 555 connects to an RC circuit, so the IC resets the output upon power-on. By playing with the resistor values, you can obtain many operating conditions. The high output drive of the 555 can actuate almost any type of relay. The diode at the output protects the 555 from the back-EMF of the relay coil when the relay shuts off.

The detection levels of the 555 are one-third and two-thirds of the supply voltage. R_1 and R_2 are of equal value. When you push no button, the input connects to ground via R_3. When you push one button, the input voltage rises to $V_{CC}R_3/(R_1+R_3)$. When you push both buttons, the input voltage rises to $V_{CC}R_3/(0.5R_1+R_3)$. If $R_3=1.5\ k\Omega$, and $R_1=R_2=1\ k\Omega$, you have the following conditions:

- No buttons: $V_{IN}=0V$; thus, the output is off.
- One button: $V_{IN}=0.6V_{CC}$—below the $0.66V_{CC}$ high level; thus, no change is in state.
- Two buttons: $V_{IN}=0.75V_{CC}$—above the $0.66V_{CC}$ high level; the relay switches on.

The circuit lends itself to modifications. You could add an emergency cutoff by connecting a switch across C_1, providing a permanent reset of the output. Or, for delayed action, you could connect capacitors across R_1 and R_2. You can also manipulate the detection levels by connecting a resistor between Pin 5 and V_{CC} or ground. (DI #2555)

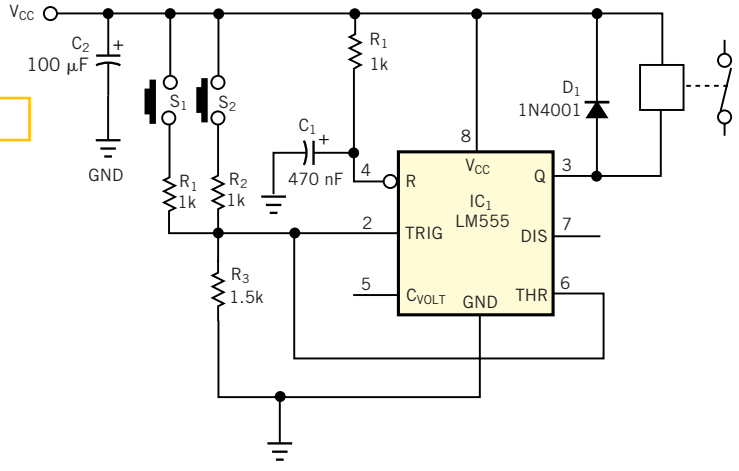


Figure 1 Two buttons turn on the relay; one buttons keeps it going.

TO VOTE FOR THIS DESIGN, ENTER NO. 321

PWM circuit uses one op amp

Ferran Bayes, Electronica Digital, Barcelona, Spain

A previous design idea is reminiscent of a similar but somewhat simpler circuit (see "Low-power PWM circuit is simple, inexpensive," *EDN*, Jan 6, 2000, pg 119). This circuit delivers a rectangular signal with duty cycle varying between 0 and 100% in response to an input signal varying from 0 to 5V dc (**Figure 1**). As with the above-mentioned circuit, the frequency is not constant (**Figure 2**), but the circuit is so simple that it can be useful in certain applications. In response to the hysteresis R_2 provides and the time constant R_3C_1, the comparator delivers the rectangular wave (**Figure 3**). The voltage V− at the inverting input swings between the two threshold levels, V_{TH} and V_{TL}. If you assume that $R_2 >> R_1$, then V+ is always very close to V_{IN}. R_3C_1 averages the signal at V_{OUT}, and the dc voltage at V− is proportional to the duty cycle of V_{OUT}. The closed feedback loop tries to make V− equal to V+; therefore, the duty cycle at V_{OUT} is proportional to V_{IN}.

The voltage at V_{OH} determines both the output signal's high level and the full-scale range of V_{IN}. It can have any value, insofar as it does not surpass the common-mode input range of the comparator. The mathematical analysis of the circuit is easy if we assume that, because $V_{TH}-V_{TL}$ is small, we can approximate the exponential charge and discharge of C_1 to assume the characteristic stemming from a constant-current source/sink. During the charging phase, the current is approximately $(V_{OH}-V_{IN})/R_3$, so:

$$V_{TH}-V_{TL}=\frac{(V_{OH}-V_{IN})T_1}{R_3C_1}.$$

Similarly, during the discharge phase, we can assume the current is $V_{IN}/3$, and

$$V_{TH}-V_{TL}=\frac{V_{IN}T_2}{R_3C_1}.$$

Matching the two equations yields

$$\frac{T_2}{T_1}=\frac{V_{OH}-V_{IN}}{V_{IN}},$$

and the duty cycle is

$$\text{DUTY CYCLE}=100\frac{T_1}{T_1+T_2}=$$

$$100\frac{1}{1+\frac{T_2}{T_1}}=100\frac{V_{IN}}{V_{OH}}.$$

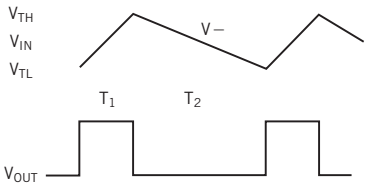
You can see that the duty cycle is directly proportional to V_{IN}: 0% for $V_{IN}=0V$ and 100% for $V_{IN}=V_{OH}$. Moreover, the duty cycle is essentially independent of the component values, with the constraint that $R_2 >> R_1$ to keep hysteresis small. An inverse relationship between duty cycle and V_{OH} can be useful in some applications, so consider V_{OH} as an additional input. The output frequency follows the relationship

$$f=\frac{(R_1+R_2)(V_{OH}V_{IN}-V_{IN}^2)}{R_3C_1V_{OH}^2R_1},$$

reaching its maximum at $V_{IN}=V_{OH}/2$.

Tests with a TLC393 CMOS comparator and a bipolar LM393 reveal that the TLC393 performs better at low values of V_{IN}, because of its lower V_{OL}. Avoid loading the comparator's output; buffer it if necessary, because the loading can degrade the switching levels. (DI #2552)

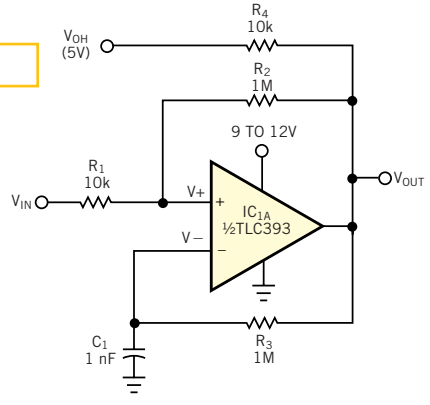
Figure 3



The voltage at the inverting input follows a linear ramp.

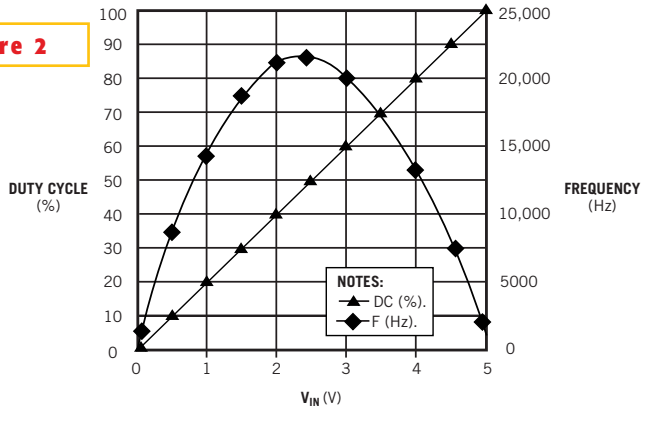
To Vote For This Design, Enter No. 322

Figure 1



This voltage-controlled PWM circuit is simplicity personified.

Figure 2



Output frequency is a nonlinear function of the input voltage.

Use 8051's power-down mode to the fullest

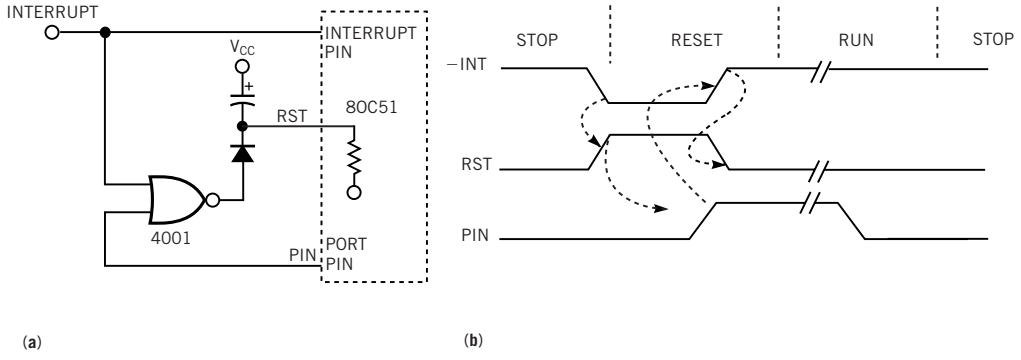
Jerzy Chrzaszcz, Warsaw University of Technology, Poland

The 8051 series µCs offer idle and power-down modes, both of which you invoke by setting appropriate control bits from the code. For instance, a 5V, 12-MHz Atmel AT89C2051 consumes approximately 9 mA in active mode, 1.8 mA in idle mode, and only 12 µA in power-down mode. Although power-down savings outperform those of the idle mode by two orders of magnitude, the only way to restore the AT89C2051's operation from a power-down state is to reset, which constitutes a serious drawback for system designers. This disadvantage holds true for most 8051-compatible processors; only high-end models wake from power-down state by interrupt. However, a simple method exists for using an ordinary 8051 in power-down mode with interrupt recovery. You can easily adapt the proposed solution for various requirements; the only assumption is that an external device, such as a keyboard or sensor, asserts an interrupt request, which is negated when the processor takes a specific action, such as reading the status register. When the µC has nothing to do, the program switches the external pin on the µC low, and the µC enters power-down mode.

The incoming interrupt request passes through the NOR gate (**Figure 1a**), causing a processor reset (**Figure 1b**). When the port pin is automatically high, the gate closes, negating the reset signal. Such a scenario requires an initialization program to distinguish between cold restart (power-up) and warm restart (wake-up). You can easily accomplish this function by checking whether locations in data memory match a predefined pattern (signature), which is set just before entering power-down mode. For details on the warm-restart concept, refer to application note AN424, "8051 family warm boot determinations" from Philips. Another problem is restoring special-function registers' contents—unlike internal data RAM, which remains unchanged, Special-function registers are automatically initialized upon reset. Depending on the application, some registers may always be set to predefined values, and some must be stored in data memory before entering power-down mode and reloaded during warm restart. You can download the necessary software listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2557. (DI #2557)

To Vote For This Design, Enter No. 323

Figure 1



A simple NOR gate (a) allows interrupt-driven wake-up from power-down mode (b).

Current source allows measuring three-wire RTD

John Wynne, Analog Devices, Limerick, Ireland

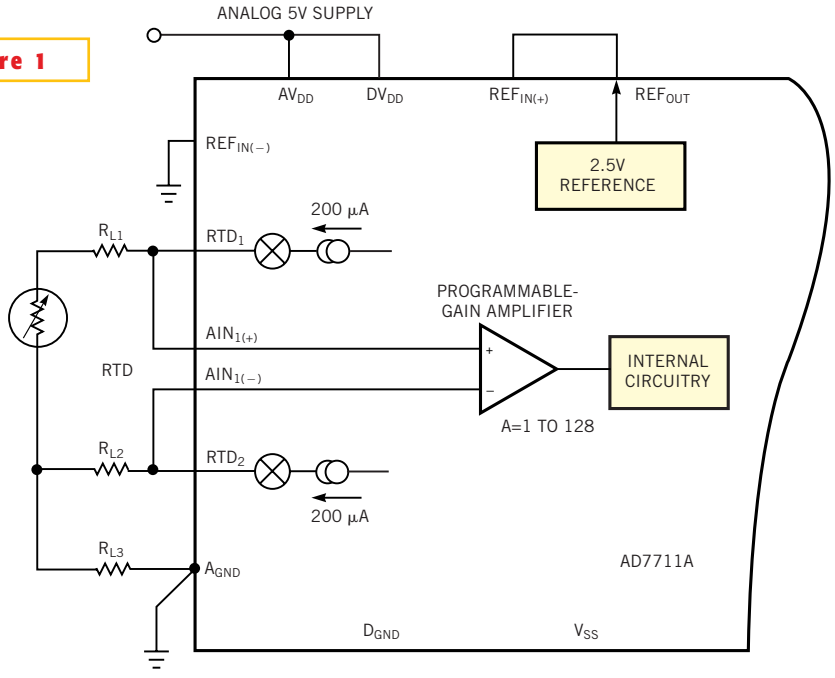
In applications in which you need remote measurement of temperature with a three-wire RTD (resistance-temperature detector), it is important to eliminate the ohmic errors caused by the excitation current flowing through the wiring resistance. You can locate the RTD more than 1000 ft from the ADC with wiring resistance in the tens of ohms. Normally, you would remove the ohmic errors by using two identical current sources that convert the wiring drops to a common-mode signal that the differential input of the ADC rejects. This technique is based on the not-unreasonable

assumption that the wiring resistances of the three-wire RTD are equal. **Figure 1** shows a typical circuit based on this assumption. The two current sources are assumed to be identical and to track each other closely over temperature and supply-voltage changes. However, a finite level of mismatch exists. In applications in which accuracy is paramount, it may be wise to use only one excitation-current source for the RTD and thus avoid any potential mismatch between two sources. However, the single-current-source approach for exciting a three-wire RTD complicates the effort to reject the ohmic drops, because you can no longer eliminate the wiring drops as a common-mode signal.

Nevertheless, you can still eliminate the wiring drops by using a two-channel ADC and a little extra software computation. You take two conversions, and software subtracts the error term stemming from the wiring resistance. In **Figure 2** the wiring resistances are represented by lumped elements R_{L1}, R_{L2}, and R_{L3}. Assume that the wiring resistance of all three leads is equal ($R_{L1}=R_{L2}=R_{L3}=R_L$). In fact, it is necessary only that R_{L2} and R_{L3} be equal, because R_{L1} appears in both equations. The circuit uses the RTD in an "upside-down" fashion. Two FET switches, SW_A and SW_B, direct the excitation current through the appropriate legs of the RTD. To avoid interruptions in the current flow, make-before-break switching is advisable. Starting with both switches closed, SW_A opens, and the AD7711A takes a measurement. The measured voltage is $2I_1R_L$, which represents the out-and-back wiring drop. Next, SW_A closes, and SW_B opens. The ADC takes a measurement on Channel 2. The measured voltage is $2I_1R_L+V_{RTD}$. This signal represents the out-and-back wiring drop plus the desired signal. The V_{RTD} term is the first result subtracted from the second. The onboard, 400-µA current source of the AD7711A serves as the excitation current in **Figure 2**. (DI #2556)

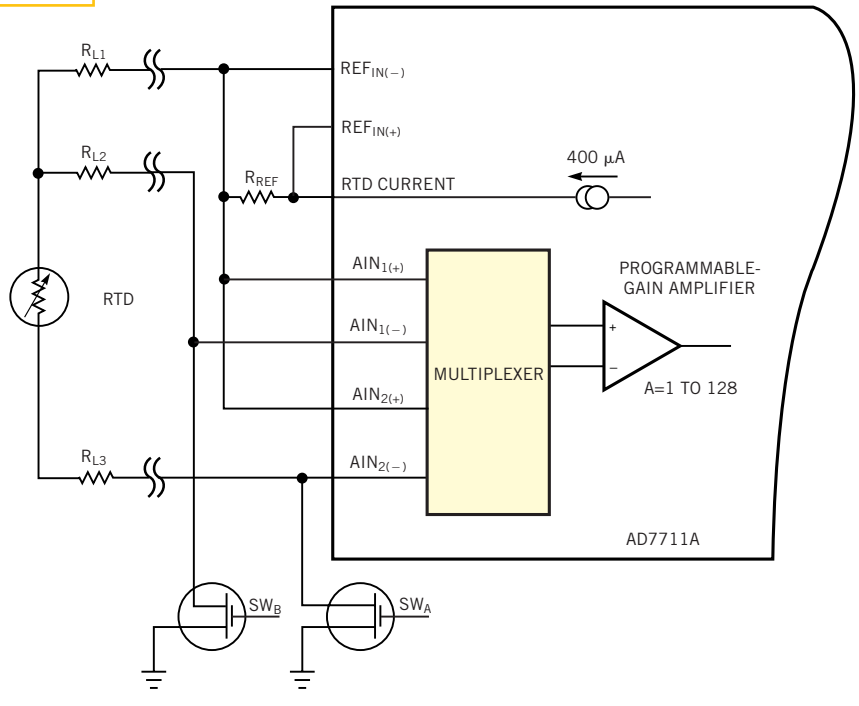
To Vote For This Design, Enter No. 324

Figure 1



Two current sources turn wiring drops into a common-mode signal.

Figure 2



One current source and a software subtraction eliminate wiring-drop errors.

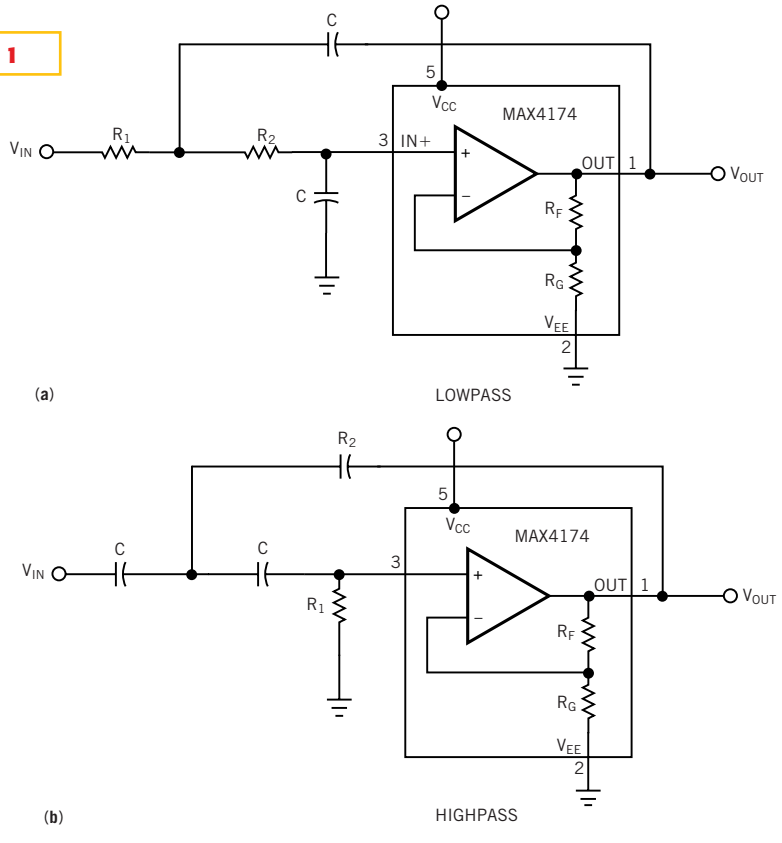
Fixed-gain op amps simplify filter design

Dan Christman, Maxim Integrated Products, Sunnyvale, CA

Simple second-order filters meet many filtering requirements. A low-order lowpass filter, for example, is often adequate for antialiasing in ADC applications or for eliminating high-frequency noise in audio applications. Similarly, a low-order highpass filter can easily remove power-supply noise. When you design such filters with built-in gain, fixed-gain op amps can save space, cost, and time. **Figure 1** illustrates the use of fixed-gain op amps in building second-order lowpass and highpass Sallen-Key filters. Filter "cookbooks" are useful in designing these filters, but the cookbook procedures usually break down for a given response, such as Butterworth, if the gain set by R_F and R_G is greater than unity. What's more, the cookbook component-value formulas can yield unrealistic values for the capacitors and resistors.

Butterworth filters, for example, offer the flattest passband. They also provide a fast initial falloff and reasonable overshoot. You can easily design such filters using **Table 1** with the following equations: $R_2=1/(2\pi f_C C\sqrt{X})$ and $R_1=XR_2$. For a gained filter response, the use of a fixed-gain op amp reduces cost and component count. It also reduces sensitivity, because the internal, factory-trimmed, precision gain-setting resistors provide 0.1% gain accuracy. To design a second-order Butterworth lowpass or highpass filter using a fixed-gain op amp, follow these steps:

Figure 1



Sallen-Key filters use fixed-gain op amps to realize a second-order Butterworth response.

TABLE 1—BUTTERWORTH-FILTER-DESIGN CRITERIA

Gain	Lowpass X	Highpass X
1.25	*	1.372
1.5	2	1.072
2	0.5	0.764
2.25	0.404	0.672
2.5	0.343	0.602
3	0.268	0.5
3.5	0.222	0.429
4	0.191	0.377
5	0.15	0.305
6	0.125	0.257
7	0.107	0.222
9	0.084	0.176
10	0.076	0.159
11	0.07	0.146
13.5	0.057	0.121
16	0.049	0.103
21	0.038	0.08
25	0.032	0.068
26	0.031	0.066
31	0.026	0.056
41	0.02	0.043
50	0.017	0.035
51	0.017	0.035
61	0.014	0.029
81	0.011	0.022
100	0.009	0.018
101	0.009	0.018

*** A gain of 1.25 is impossible to obtain with matched capacitors for the lowpass case.**

1. Determine the corner frequency f_C.
2. Select a value for C.
3. For the desired gain value, locate X under the proper column heading in **Table 1**.
4. Calculate R_1 and R_2 using the equations.

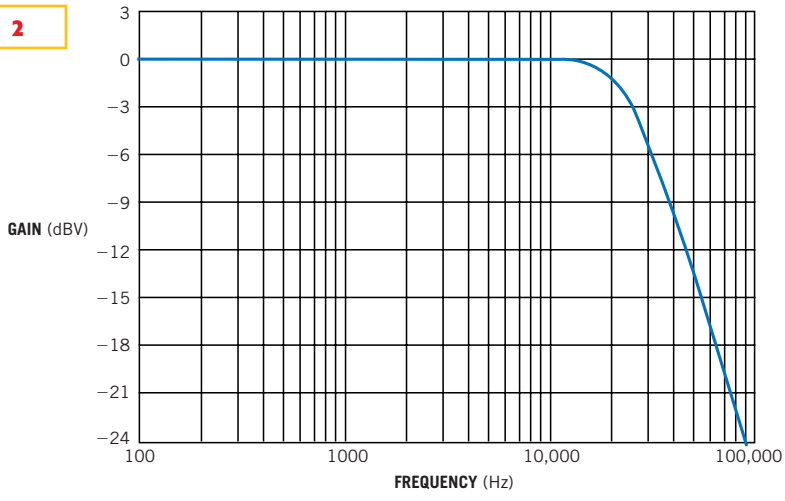
Choosing C and then solving for R_1 and R_2 lets you optimize the filter response by selecting component values as close to the calculated values as possible. C can be lower than 1000 pF for most corner frequencies and gains. Fixed-gain op amps come optimally compensated for each gain version and provide excep-

tional gain-bandwidth products for systems operating at high frequencies and high gain. Suppose, for example, you must design a lowpass filter with a 24-kHz corner frequency and a gain of 10. Step 1 is complete (f_C=24 kHz). Next, complete Step 2 by selecting a value for C, say 470 pF. In **Table 1**, note that X=0.076 for a lowpass filter with a gain of 10. Substitute these values in the equations:

$R_2 = 1/(2\pi f_C C\sqrt{X} = 1/(2\pi \times 24\ \text{kHz} \times 470\ \text{pF} \times \sqrt{0.076} = 51\ \text{k}\Omega$, and $R_1 = XR_2 = 0.076 \times 51\ \text{k}\Omega = 3.9\ \text{k}\Omega$. With these component values, the circuit in **Figure 1** yields the second-order Butterworth lowpass response in **Figure 2**. (DI #2551)

To Vote For This Design, Enter No. 325

Figure 2



Using the circuit values in the text, the circuit in Figure 1a produces this Butterworth response.

Use your printer port as a high-current ammeter

K Suresh, Indira Gandhi Centre for Atomic Research, Kalpakkam, India

With a few inexpensive components and INT1Ch, you can turn the printer port of your PC into a high-current ammeter. This design's goal is to make remote high-current measurements, but you can use this technique to remotely measure any other similar analog electrical quantity, such as voltage and charge, at moderate speeds without going for expensive, PC-based general-purpose or tailor-made data-logging add-on cards.

In **Figure 1a**, a low temperature-coefficient manganin element senses the high current of the remote module. The manganin element gives an output of 0 to 500 mV, which is V_{IN} to the circuit, for an output-current range of 0 to 100A. Instrumentation amplifier IC_1 amplifies this output voltage by 2. Voltage-to-frequency converter IC_2 digitizes this amplified voltage to a resolution of 13 bits. The values of R_1, R_2, and C_1 give a serial output-pulse train at a rate of 10 kHz/V according to $F_{OUT}=V_{IN}/10(R_1+R_2)C$.

The converter's output linearity of less than 0.01% ensures a linear conversion of the sensed voltage/current to frequency throughout the current range.

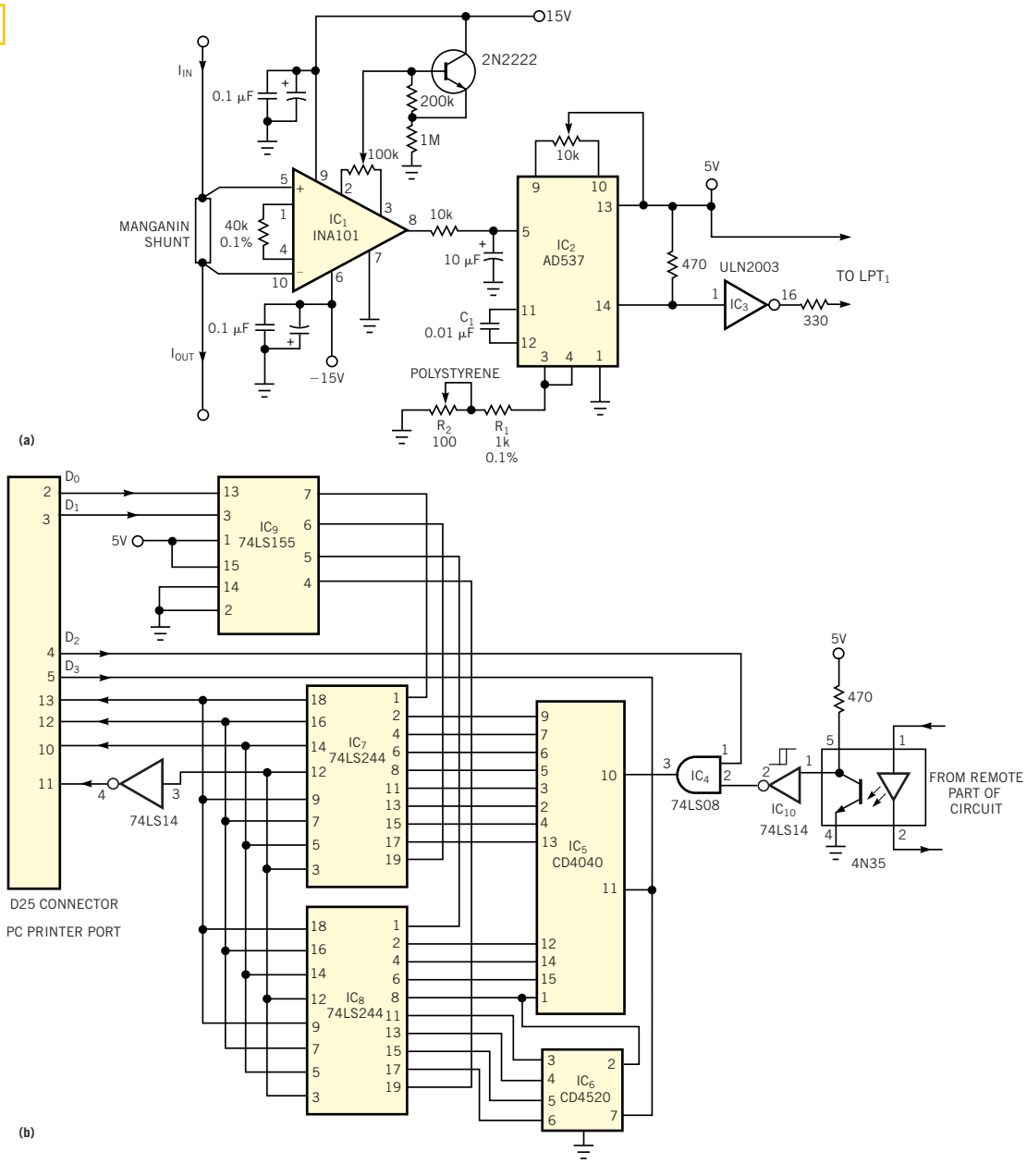
Figure 1b shows the other part of the circuit that attaches to the PC's LPT printer port. This circuit couples the converter's output pulses through an optocoupler. It also conditions and counts the pulses using a 16-bit counter, IC_4 and IC_5, whose output bits IC_7 and IC_8 buffer. The circuit hooks the buffer outputs to the input port, STATUS port at 0x379h, of the printer adapter. The circuit inhibits or allows the pulses to the counter by controlling the output bit D_2 (DATA port at 0x378h) of the printer port to enable or disable AND gate IC_4. The PC reads the counter output a nibble at a time by controlling the address inputs of a two-to-four decoder (IC_9) using D_0 and D_1 bits of the DATA port. The decoder outputs in turn control the buffer outputs.

A simple Turbo C program controls the remote current measurement. (You can download the program from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2550). The timer-tick interrupt, 0x1Ch, which occurs 18.2 times/sec and whose only task is to keep track of the time of day, generates the timebase. The timer-tick interrupt executes the TIMEBASE() routine to update the TIMER variable. To make a measurement, the printer port first disables the pulses to the counter by setting D_2 of the data port to logic 0 and clearing the counter contents by setting D_3 to logic 1 and resetting it to logic 0. To enable the

pulses to the counter, D_2 bit of data port is a logic 1. For each 0x1C interrupt, the TIMER increments. When the TIMER reaches 18 (in approximately 0.989 sec), disabling the D2 bit to logic 0 inhibits the pulses. Also at this time, the program corrects the timebase to 1 sec by applying correction factor CF=18.2/18. The ISENSE() routine sequentially reads the counter-output nibbles starting from LSNIBBLE through the status port by applying addresses 0x00 to 0x03 to the decoder. The program applies the correction factor to the values and displays the value of the sensed current CURRENT. The program sets the TIMER variable to zero and clears the counters. The unit is now ready for another measurement cycle. (DI #2550)

To Vote For This Design, Enter No. 326

Figure 1



A low-temperature-coefficient manganin element senses the high current of the remote module, and IC_2's voltage-to-frequency converter digitizes the resulting amplified voltage (a). This result, in turn, attaches to the PC's LPT printer port through an optocoupler, counters, and buffers (b).

design ideas

Edited by Bill Travis and Anne Watson Swager

Scheme autodetects baud rate

Alexander Eisen, State University of New York, Buffalo, New York

THE POPULARITY AND easy access of RS-232 ports lend them to many communication projects. You can use a port "as is" or as a tiny parallel port when the exchange uses only control lines. Before the asynchronous serial-data transfer between two devices can take place, you must ensure that both devices are configured to the same data format and transmission rate. Usually, the OS utility or application program on the host computer performs this task, which the OS or a combination of switches selects in the peripheral.

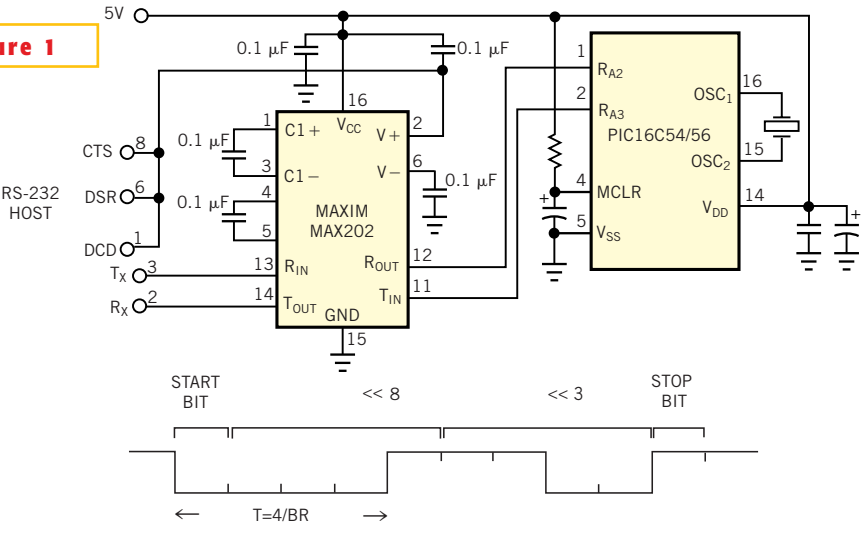
The design in **Figure 1** and the accompanying software realize the requirement of automatically equalizing the speed of the parties by adapting the bit rate of a µC's serial port to that of a host computer. The host sends a known ASCII code to a µC that estimates the bit rate of the transmission. The µC samples the receiving end of a transmission line. As soon as the line goes low, indicating a start bit, the µC clears the timer. After the line again goes high, the µC reads the timer's value and uses this value to calculate the bit rate of a transmission. The µC uses the calculated bit rate in a timing-delay loop for clocking out the data from the µC to the host computer during transmission and for sampling the incoming data during receiving mode. The time between those falling and rising edges on a transmission line depends, of course, on the transmitted ASCII symbol. Its good practice to use a symbol with 2^k-1 zeros on the low side. Then, when you include the start bit, there are 2^k trailing zeroes. One timing unit is easier to derive by substituting a division with k right shifts.

The application in **Figure 1** uses an 8-bit RISC µP and an ASCII code of 8 (38 hex) to establish a serial communication. Any symbol that ends with 8H also works. The timer increments its value every $4P/F_{OSC}$ seconds, where P is a prescaler factor and F_{OSC} is the µP's clock frequency. Thus, the timer's value after 4 bits have transmitted during the time between the falling and rising edge, is $T=4/BR$, is $N=F_{OSC}/(BR\times P)$, where BR is the bit rate.

Now you can derive the 1-bit transmission time, which is a reciprocal of a transmission speed, as $(N/4)(4P)/F_{OSC}=N\times P/F_{OSC}$.

Because the µP's timer is only 8 bits long, it is important to pick up the right prescaler factor. N must not exceed the timer's capacity of 255 but should be large enough to cover a certain range with good resolution. For the worst case, which is when the bit rate is at its minimum, P should be $P>F_{OSC}/(N_{MAX}\times BR_{MIN})$.

Figure 1



Using its internal timer, a µP calculates the bit rate of transmission from a host computer and adapts its serial port's bit rate to match.

After you determine all of these timing relationships, you need to design a software loop for a 1-bit delay. One machine cycle for the µP in **Figure 1** is $4/F_{OSC}$, so the total cycle count for the execution time of a loop should be NP/4. The loop for a bit_clk is 4(k+2) cycles long (**Listing 1**). Therefore, k=(NP/16)−2. If you choose P=16, k then conveniently becomes N−2. At F_{OSC}=8 MHz, these parameters reliably cover speeds of 2400 to 38,400 bps, or baud rate, which in the case of the serial port is the same thing. (DI #2546)

To Vote For This Design, Enter No. 328

LISTING 1—AUTOMATIC BIT-RATE-DETECTION ROUTINE

```
; Bit rate automatic detection routine
;*********************************************************
ser_out         =       ra.3            ;transmit pin
ser_in          =       ra.2            ;receive pin
;
        org     8
;
k                       ds      1       ;derived time constant
clk_cnt                 ds      1       ;counter for serial timing
bit_num                 ds      1       ;number of received/transmitted bits
xmt_b                   ds      1       ;byte to transmit
;
        device pic16c54,xt_osc,wdt_off,protect_off
        reset start
;
        org     0
;
start           mov     !ra,#4          ;ra.2 - input, ra.3 - output
                mov     option,#3       ;set prescaler to divide by 16
;
;******************** serial in ***************************************
;
st_by           snb     ser_in          ;skip next instruction if Rx goes low
                jmp     st_b            ;go back and wait till start bit
                clr     rtcc            ;reset timer, start to count ticks
                jb      ser_in,st_by    ;go back if it was only a glitch
;
;******************** start is good ***********************************
;
roll            jnb     ser_in,roll     ;wait till line goes high
                mov     k,rtcc          ;record timer's value
                dec     k               ;make k = N-2
                dec     k               ;decrement
;
;******************** success *****************************************
;
                mov     xmt_b,#'O'      ;output "OK" using calculated
                call    out             ;Bit Rate
                mov     xmt_b,#'K'
                call    out
                mov     xmt_b,#13       ;carriage return
                call    out
                mov     xmt_b,#10       ;line feed
                call    out
                goto    do_something    ;continue with your application
                nop
;
;****************** serial out ****************************************
;
out             mov     bit_num,#8      ;number bits to transmit
                clrb    ser_out         ;ready to send start bit
                call    bit_clk         ;send start bit
xmit            rr      xmt_b           ;move LSB to carry
                movb    ser_out,c       ;move carry to Tx
                call    bit_clk         ;send
                djnz    bit_num,xmit    ;decrement and go on if not done yet
                setb    ser_out         ;load stop bit
                call    bit_clk         ;and send
                ret                     ;done
                nop
;
bit_clk         mov     clk_cnt,k
loop            nop
                djnz    clk_cnt,loop
                ret
                nop
```

Circuit computes first derivative

Richard Panosh, Vista, Bolingbrook, IL

The circuit in **Figure 1** computes the derivative of an input signal as the integral of the input signal minus the signal itself. The response of the circuit is

$$V_{OUT} = -\frac{R_2}{R_1} V_{IN} \frac{R_{EQ} C_1 s}{1 + R_{EQ} C_1 s},$$

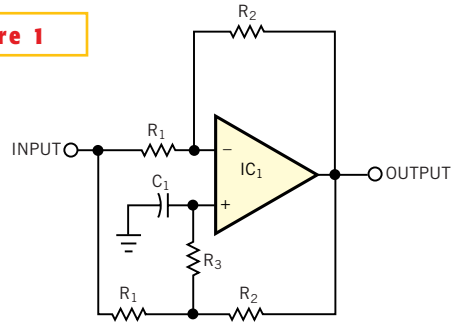
where R_{EQ} is the parallel equivalent resistance of R_1 and R_2 plus the resistance of R_3, or

$$R_{EQ} = \frac{R_1 R_2 + R_1 R_2 + R_2 R_3}{R_1 + R_2}.$$

This response is identical to that of the classic inverting differentiator, in which the response is $V_{OUT} \simeq -R_{EQ} C_1 s V_{IN}$ for input frequencies lower than $1/2\pi R_{EQ} C_1$. If R_3 is much greater than the parallel combination of R_1 and R_2, then $R_{EQ} \simeq R_3$. On the other hand, if R_3 is set at 0Ω, the bias currents of the op amp balance to minimize voltage offsets, and

$$R_{EQ} = \frac{R_1 R_2}{R_1 + R_2}.$$

Figure 1



This circuit produces less noise than the classic inverting differentiator.

A major problem with the classic inverting differentiator is high noise. By its nature, a differentiator must exhibit increasing gain with frequency, and this increasing gain amplifies the inherent amplifier noise. An equivalent input-noise voltage, V_N in the classic inverting differentiator, produces output noise of magnitude $RCsV_N$. In the case of the differentiator in **Figure 1**, the equivalent input noise produces an output magnitude of only $(1+R_2/R_1)V_N$. (DI #2522)

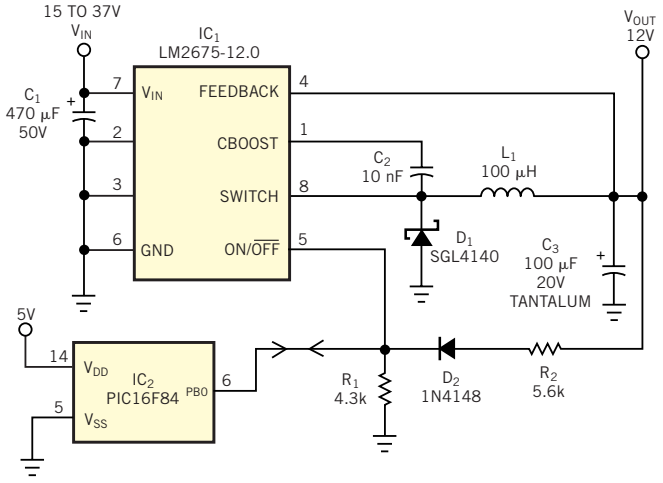
To Vote For This Design, Enter No. 329

Practice safe dc/dc converter

Eugene Kaplounovski, Nautilus International, Vancouver, Canada

Short-circuit protection is an obvious requirement for a power supply, especially when its load connects to a cable that's subject to damage. Many modern power-converter ICs include some means of protection, such as thermal shutdown, against the overload condition, but, in some cases, the built-in protection may be inadequate. **Figure 1** shows a step-down dc/dc converter for two videocameras installed on a remotely controlled vehicle. Because the vehicle operates in rather harsh environments, the cables can frequently short-circuit during the system's installation and normal operation. The cameras require 12V and consume approximately 250 mA each. The converter is based on National Semiconductor's LM2675 chip, which by itself includes good protection against overloads. However, in the case of a shorted output, the catch diode, D_1, must withstand the maximum current from the IC. This current, according to the data sheet, can be as high as 2.2A, calling for an oversized diode. Also, waiting for the thermal protection to kick in assumes that you are willing to allow the device's temperature to rise significantly, heating adjacent components. This situation is undesirable, given the long periods of nonsupervised operation, during which someone should notice the problem. Ideally, someone should immediately report the faults to an operator.

Figure 1



This dc/dc converter provides flexible overload protection and diagnostics.

The system in **Figure 1** uses a Microchip Technology PIC16F84 µC, which receives its power from a separate dc source. As usual, the I/O pins in the µC are at a premium, because of the multitude of other tasks the controller performs. However, you can obtain reliable short-circuit protection and on/off control of the power supply using just one I/O pin. When you first apply power, the µC starts up. In this condition, all of its I/O lines are high-impedance inputs, so the LM2675 cannot start; resistor R_1 ties its On/$\overline{\text{Off}}$ pin low. After the initialization routine, when it is time to turn on the camera, the µC makes its pin PB0 pin (or any pin that has a totem-pole driver) an output and sends a high level to that output. The dc/dc converter starts up. After a short delay, the µC again makes its PB0 pin an input, but the power supply keeps itself on because its output voltage connects to control input via R_2 and D_2. This condition prevails while the load is normal. When the output short-circuits, the bias voltage disappears, and the chip shuts itself down. The level at PB0 goes low, notifying the µC of this condition. (PB0 is especially useful in this situation, because it can generate an interrupt request.) The µC can then alert the user of the failure, try to restart the converter after a delay in a "hiccup" mode, or both.

The duration of the worst-case short-circuit condition with this scheme is a function of the length of the start-up pulse from the µC. This pulse should be long enough—usually, approximately 10 msec—to allow the normal load (with its own input capacitors and power converters) to start. The LM2675 with a catch diode in **Figure 1** withstands short circuits for several seconds without overheating or any other problem, so the short-circuit mode is perfectly safe. An added benefit is that the µC can at any time shut down the dc/dc converter, making it possible to save battery power and reduce heat dissipation. For shutdown, the µC again makes PB0 an output but sends a zero to the I/O pin. The powerful driver in the PIC16F84 easily overcomes the bias from R_2, D_2, and R_1 and shuts down the LM2675. The divider R_2, D_2, R_1 provides a voltage—5V in this case—close to the µC's V_{DD} when the output voltage is normal. Slight voltage variations cause no harm to the µC, thanks to the controller's input-diode protection and the fact that R_2 limits the current to the PB0 pin to a safe level. However, you should keep the values in the divider low enough to not create a significant voltage drop in R_1 by the bias current from the LM2675's On/$\overline{\text{Off}}$ pin (37 µA maximum). With the values shown, the largest bias current creates a drop that does not exceed 20% of the lowest possible threshold (0.8V) for the On/$\overline{\text{Off}}$ pin. (DI #2562)

To Vote For This Design, Enter No. 330

Single-sideband demodulator covers the HF band

Israel Schleicher, Bakersfield, CA

THE CIRCUIT IN **Figure 1** complements a previous Design Idea (**Reference 1**). The modulator used a phasing network to split a low-frequency (audio) signal into in-phase and quadrature (orthogonal) components. The phasing network has an advantage over other phasing circuits in that it delivers a phase error of only 0.15° and has low sensitivity to component tolerances (**Reference 2**). By reversing the direction of the network—in other words, feeding the output with two orthogonal signals and tapping the input—the network functions as a detector. Feeding the two signals one way may produce a signal at the input, but if you interchange the two signals, no signal goes through. Because the network in the modulator circuit has two floating differential inputs, for the demodulator you must feed them from two floating sources.

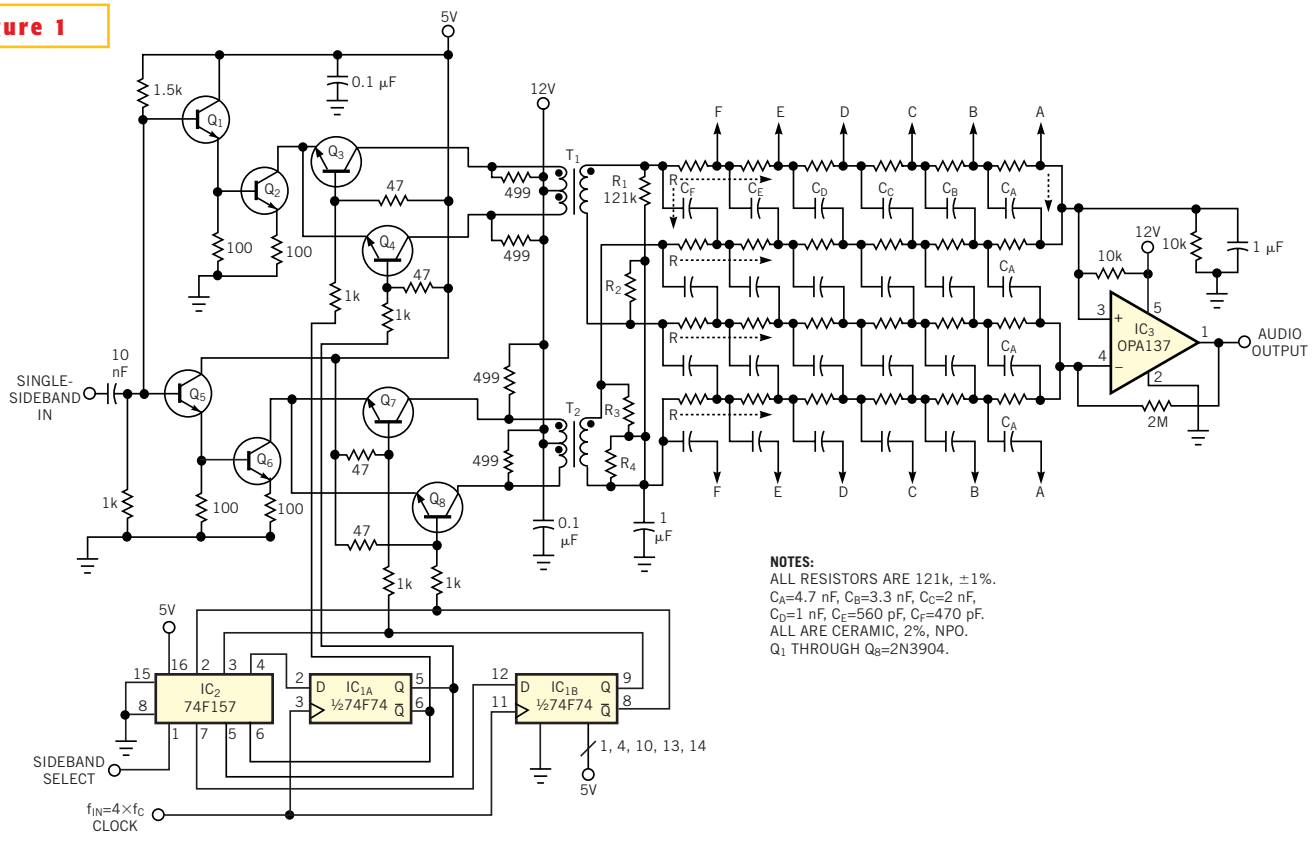
The simplest way to obtain the feed signals is to use transformers. T_1 and T_2 are 600Ω, 1-to-1 telephone-coupling transformers with center-tapped bifilar primary winding. It is important to minimize the capacitance between the primary and secondary. Q_1 through Q_4 and Q_5 through Q_8 function as balanced mixers. They provide a wide dynamic range to the circuit. The circuit forms part of a direct-conversion receiver. IC_1 provides two quadrature local-oscillator signals. IC_1 requires a drive signal with four times the carrier frequency. IC_2 allows upper or lower sideband selection. Measurements on the prototype circuit show 37 dB of unwanted-sideband rejection for a 1-kHz modulated carrier and 32 dB of rejection for a 3-kHz modulating signal. You must use a sharp-roll-off, 3-kHz lowpass filter with the circuit. (DI #2563)

REFERENCES

1. Schleicher, Israel, "SSB modulator covers the HF band," *EDN*, Sept 30, 1999, pg 122
2. Zavrell, Robert Jr, "New low-power single-sideband circuits," Philips Semiconductor, Application Note AN1981.

TO VOTE FOR THIS DESIGN, ENTER NO. 331

Figure 1



This single-sideband demodulator provides sharp rejection of the unwanted sideband.

2.8-GHz prescaler keeps cost down

Neil Eaton, Emsys Engineering, Peterborough, ON, Canada

THE PRESCALER in **Figure 1** inexpensively extends the range of a frequency counter by dividing the input signal's frequency by a factor of 1000. The guaranteed input-frequency range of the input prescaler, IC_1, is 250 MHz to 2.8 GHz, although typical values are 100 MHz to 3.5 GHz. The prototype operates at frequencies well below 100 MHz, but its fastest generator goes only to 1.7 GHz, so you cannot confirm the upper range. The input-voltage range is 400 to 1000 mV p-p from 250 to 500 MHz and 100 to 1000 mV p-p for higher than 500 MHz. IC_1 serves as a divide-by-128 prescaler, whose output is a 1.6V p-p square wave. The RC network level-shifts the output of IC_1 to ensure that the top of the square wave is above the 2V input threshold of IC_{5A}. The output of IC_{5A} is a 5V, CMOS-compatible square wave with a frequency of $^1/_{128}$ of the input frequency. Most frequency counters can handle these frequencies, but the submultiple is inconvenient for an operator. A further division by a factor of 7.8125 (1000/128) produces a scaling factor of 1000.

Fortunately, the frequency counter averages its input over many cycles, so the output of the prescaler need not be exactly $^1/_{1000}$ of the input frequency for every input pulse. The 0.8125 figure is 13 divided by 16. The average frequency ratio is therefore 7.8125 if you divide 13 output pulses of 16 by eight and the remaining three by seven. For best results, the divide-by-seven pulses should be as evenly spaced as possible. The result is a repeating sequence 16 output pulses long with the following pattern:

Pulse number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Divide by	8	8	8	8	7	8	8	8	8	7	8	8	8	8	7	8

IC_2 divides the prescaler's output by seven or eight, depending on the state of the B input. IC_3 and IC_5 both connect to the output of IC_2, so they count output pulses, not prescaler pulses. IC_5 divides the output by five to generate the divide-by-seven periods. IC_4 divides the output by 16 to reset the cycle upon completion. Without IC_4, the cycle would continue to divide by seven at every fifth output pulse for a ratio of 7.8. The construction of the circuitry inside the dashed lines in **Figure 1** is critical. The MC12079 is available only as a surface-mount device. In the design, it and its associated passive components is mounted on a Surfboard (Capital Advanced Technologies Model 9081, available from Digikey). It is then fastened, component side up, to a bare copper-clad board to create a ground plane. All connections to ground from the high-

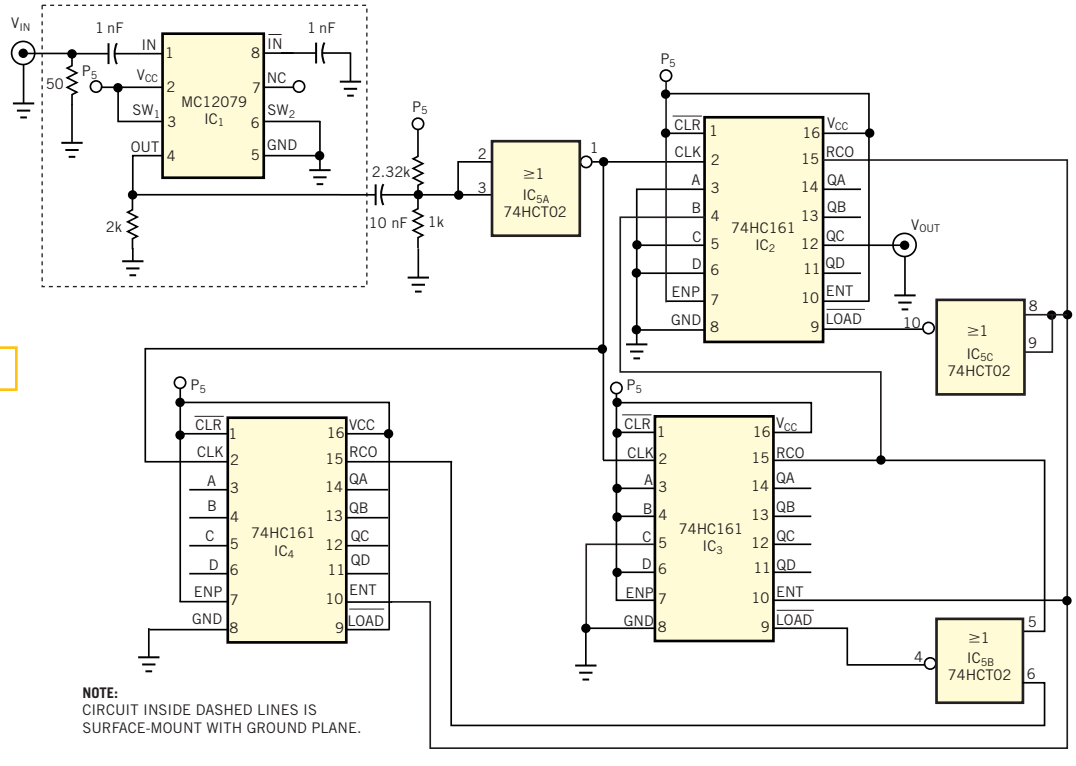


Figure 1

Extend your frequency-measurement capability with this inexpensive divide-by-1000 prescaler.

speed circuitry go directly to the ground plane using short lengths of copper braid. (Desoldering wick is ideal.) The input uses a BNC chassis-mount connector with its shell soldered directly to the ground plane, and the center pin connected to the Surfboard with the shortest possible wire. The rest of the circuit is noncritical. To avoid clutter, **Figure 1** shows no bypass capacitors, but you should place them near every IC. (DI #2564)

Add harmony to your system

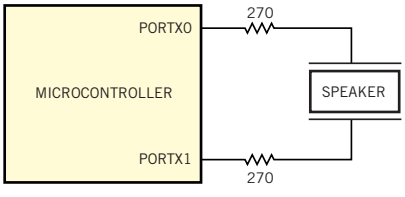
Eugene O'Bryan, Food and Drug Administration, Rockville, MD

Have you ever wished you could distinguish one device's operating state from another's by the sounds they make or that error states would sound harsh while normal operations would sound harmonious? By combining the NCO technique from a previous Design Idea (**Reference 1**) with digital mixing, you can obtain musical chords or intervals with a minimal amount of hardware and software. Any µC system can thus produce a variety of sounds. In **Figure 1**, a piezoelectric speaker, Radio Shack Model 273-091, and two 270Ω resistors transduce a pulse stream from a µC. Differential drive to this transducer increases the volume by doubling the effective voltage. The capacitance of the piezoelectric speaker reacts with the 270Ω resistors to integrate and smooth the pulse stream. The software (**Listing 1**), a tight loop, comprises a square-wave generator and an NCO's (numerically controlled oscillator's) summing part. After it sets up some registers, the sound-generating loop establishes an output level for each of two or more square waves.

The pseudocode example in **Listing 1** demonstrates a two-note generator in which the output levels of two square waves are established in registers r_vol1 and r_vol2. The frequency of each square wave is a function of the values set for variables first_note and second_note and by the cycle rate of the loop. Note that a half-cycle of a square wave concludes when the corresponding counter register (r_cnt1 or r_cnt2) reaches zero. The summing part of the loop uses the NCO technique to generate an output level for the digital mixture of the square waves generated in the first part of the loop. The mixing of two square waves with frequencies of 880 Hz, the C above middle C, and 988 Hz, the second D above middle C, in the equal-tempered scale produces the oscilloscope waveform in **Figure 2**. An AVR AT90S Series µC runs through the sound-generating loop in 18 clock cycles. Thus, to produce an 880-Hz square wave with this type of µC running at 8 MHz, the first_note or second_note value is set at 253, which is equal to 8 MHz/18/880/2.

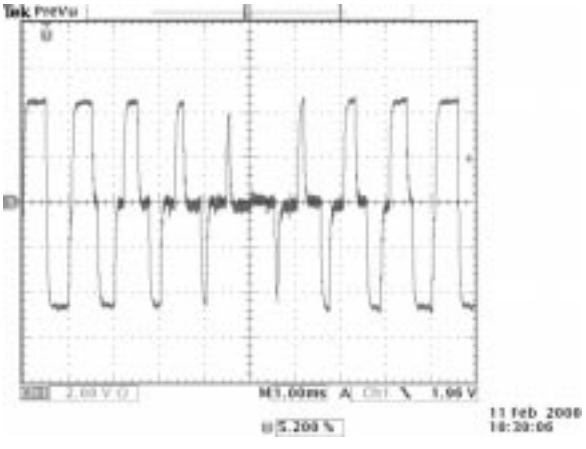
A note of caution is in order in calculating this value in an assembly program: Be careful of truncation issues; the musical intervals sound wrong if the frequencies are off. Also, when you use another type of µC, you must adjust the loop software to compensate for differences between the µC's instruction timing and the assumed instruction timing of the pseudocode. All instructions should complete in one clock cycle except for jump or branch executions, which take two clock cycles. You can create chords by expanding the software loop to include a third square-wave generator and modifying the mixer to add in the third note. A by-product of this sound-generation scheme is that it lets you control volume by changing the volume variable in **Listing 1**. The loudest possible volume occurs when the volume variable equals 80hex for a two-note generator. Setting the volume variable lower than 80hex reduces the voltage output from each square wave. With this scheme, you can produce 10 distinct volume levels, using hex values 80, 6A, 60, 58,

Figure 1



Produce harmonious sounds in your system, using this simple scheme.

Figure 2



Mixing square waves of 880 and 988 Hz produces this waveform.

50, 48, 40, 38, 30, and 28. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2561. (DI #2561)

REFERENCE

1. Ploss, Steve, "NCO technique helps μC produce clean analog signals," *EDN*, April 15, 1999, pg 129.

To Vote For This Design, Enter No. 333

LISTING 1—ASSEMBLY CODE FOR SOUND GENERATION

```
;****************************************
; Setup
;****************************************
  ldi  r_cnt1,first_note  ; load counter1 register with first_note value
  ldi  r_cnt2,second_note ; load counter1 register with second_note value
  ldi  r_vol,volume       ; load volume register with a value (0-80hex)
  ldi  r_lout,1           ; load output register with low value
  ldi  r_hout,2           ; load output register with high value

;****************************************
; Square Wave Generator
;****************************************
line1:
  dec  r_cnt1             ; decrement counter1 register
  breq line5              ; branch if result of decrement equals zero
  nop                     ; no operation (waste one instruction time)
  jmp  line7              ; jump to line7
line5:
  ldi  r_cnt1,first_note  ; load counter1 register with first_note value
  eor  r_vol1,r_vol       ; flip state of first_note intensity (0 or volume)

line7:
  dec  r_cnt2             ; decrement counter2 register
  breq line11             ; branch to line 11 if result of decrement equals zero
  nop                     ; no operation (waste one instruction time)
  jmp  line13             ; jump to line13
line11:
  ldi  r_cnt2,second_note ; load counter1 register with second_note value
  eor  r_vol2,r_vol       ; flip state of second_note intensity (0 or volume)

;****************************************
; NCO Summing
;****************************************
line13:
  add  r_acc,r_vol1       ; add first_note intensity to accumulator register
  brcs  line20            ; branch if result of add overflows
  add  r_acc,r_vol2       ; add second_note intensity to accumulator register
  brcs  line22            ; branch if result of add overflows
  nop                     ; no operation (waste one instruction time)
  out  portx,r_lout       ; output a low level to enunciator
  jmp  line1              ; jump to line1
line20:
  nop                     ; no operation (waste one instruction time)
  adc  r_acc,r_vol2       ; add second_note int. and overflow bit to accumulator
line22:
  out  portx,r_hout       ; output a high level to enunciator
  jmp  line1              ; jump to line1
```

Instrumentation amp provides unipolar and bipolar outputs

David Rathgeber, Alles Corp, Toronto, ON, Canada

Intelligent sensors and signal conditioners for plant-automation systems must produce outputs compatible with standard data-acquisition systems, such as PCs, programmable logic controllers, and remote terminals for supervisory and control systems. Standard analog inputs are 0 to 1V, ±10V, 0 to 1 mA, and 4 to 20 mA. The circuit in **Figure 1** makes all these signals available, using only three packages in addition to the DAC. The voltage-output section requires a thin-film resistor package, Beckman Series 668/698, which tracks with temperature, and two FET-input op amps. The current-output section requires the addition of a four-resistor package, Beckman Series 664/694, and two more op amps. A reference voltage of −1.000V yields voltage outputs of 0 to 1V and ±1V and either 0 to 1 mA with R_X=1 kΩ and JP_2 and JP_4 connected or 4 to 20 mA with R_X=83.33Ω and JP_1 and JP_3 connected. A reference of −10.00V and appropriate values of R_X yield corresponding voltage and current outputs. All R_1 and R_2 values in **Figure 1** are 100 kΩ±0.1%. You should select Q_1, and possibly use a heat sink, taking into consideration the supply voltage and output-current range. (DI #2560)

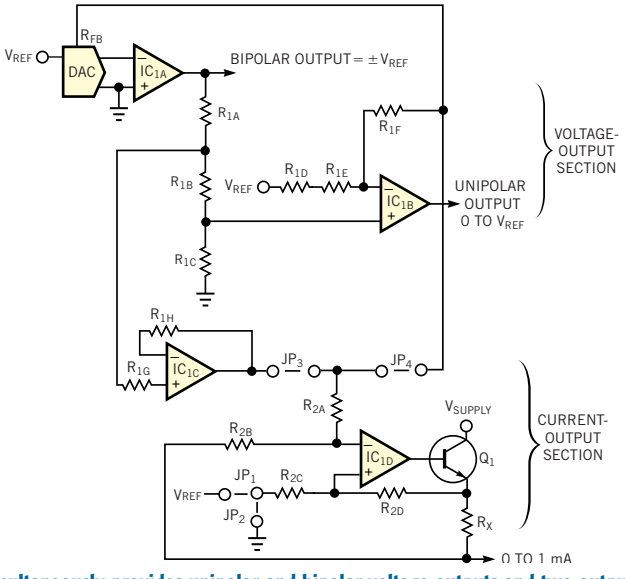


Figure 1

This circuit simultaneously provides unipolar and bipolar voltage outputs and two output-current ranges.

To Vote For This Design, Enter No. 334

design ideas

Edited by Bill Travis and Anne Watson Swager

Ride the logic thresholds

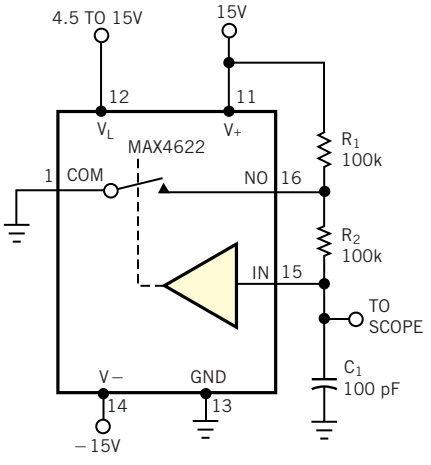
Roger Griswold, Maxim Integrated Products, Sunnyvale, CA

Most logic devices have a 5V supply rail. However, you often need to know the upper and lower switching thresholds for a device operating at supply levels other than 5V. In some cases, it is also important to know the variation of these thresholds with supply voltage. A simple test circuit allows you to make these measurements (**Figure 1**). The basic idea is to create an oscillator by feeding an output signal back to the control input and then monitor the control input with an oscilloscope. Then, you can easily monitor the upper and lower transition points on the scope. To slow the frequency of oscillation and reduce the effects of propagation delay, it's sometimes desirable to add an RC network.

The circuit examines the threshold voltages at the control input of an analog switch, but the scheme applies to any simple logic device. Switching thresholds for the IC depend on the logic-supply voltage you apply to V_L. (The analog-switch channel COM-NO handles voltages of ±15V.) The data sheet specifies IN thresholds for the most common usage (V_L=5V), but the chip accommodates V_L from 4.5V to the positive rail (15V in this case). The circuit characterizes threshold voltages over the full operating range of V_L (**Figure 2**). At power-up, the switch is open. V+ charges C_1 through R_1 and R_2 until IN reaches its upper threshold, closing the switch. C_1 then discharges to ground (through R_2 and the switch) until IN reaches its lower threshold, thereby opening the switch and repeating the cycle.

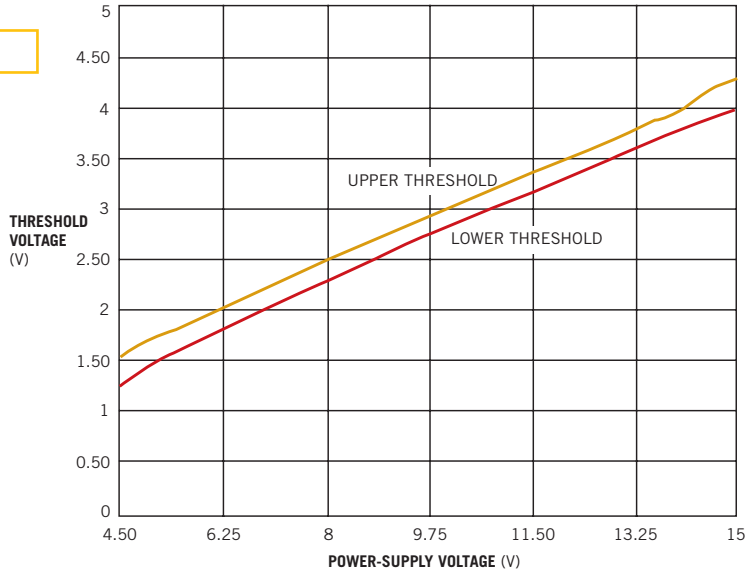
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 321 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



You can use an oscilloscope to measure switching thresholds of any simple logic device as a function of supply voltage.

Figure 2



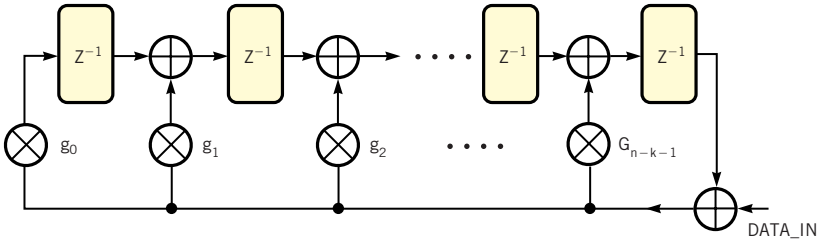
For the circuit in Figure 1, the thresholds vary with supply voltage as shown.

VHDL produces CRC checker

Antonio Di Rocco, Siemens ICN, L'Aqila, Italy

CRC (cyclic-redundancy checking) allows you to verify the integrity of transmitted data frames. It finds use in many transmission protocols, both bit-oriented (High Level Data Link Control, Advanced Data Communications Control Procedure, CRC-CCITT, for example) and frame-oriented (asynchronous transfer mode, Ethernet, Fiber Distributed Data Interface, for example). CRC protection constitutes adding redundancy bits (n−k bits) to a frame (k bits) of data to transmit a coded frame of n bits. Express the data as a polynomial data frame of k bits:

Figure 1



The linear-feedback shift registers implement the division of $G_{n-k}(D)$.

$$X_k=\{x_{k-1}, x_{k-2},\ldots, x_0\}, \quad (1)$$

which is written as:

$$X_k(D)=x_{k-1}D^{k-1}+x_{k-2}D^{k-2}+\ldots+x_1D+x_0, \quad (2)$$

where D is the variable and bits x_i (1,0) are the coefficients of the polynomial. Starting from $X_k(D)$ to generate a code word of n bits (n>k), you use a so-called generator polynomial:

$$G_{n-k}(D)=g_{n-k-1}D^{n-k-1}+g_{n-k-2}D^{n-k-2}+\ldots+g_1D+g_0, \quad (3)$$

Multiplying $x_k(D)$ by D^{n-k}, you obtain a polynomial of grade n. This operation shifts the data $X_k(D)$ to the left by n−k bits, attaching n−k zeros to the least significant bits. Now, consider:

$$D^{n-k-1}X_k(D)=Q_k(D)\cdot G_{n-k}(D)+R_{n-k-1}(D), \quad (4)$$

where the original data multiplied by D^{n-k} is divided by the generator polynomial, yielding the quotient $Q_k(D)$ and the remainder $R_{n-k-1n}(D)$, which always has the grade n−k−1 (n−k bits). The transmitted frame consists of the original data followed by n−k bits of remainder. The polynomial arithmetic, in this case, is cast in a binary algebraic field. Addition and subtraction are identical and equivalent to an XOR operation without the need of a carry. Thus, it is possible to rewrite **Equation 4** as follows:

$$D^{n-k}X_k(D)+R_{n-k-1}(D)=Q_k(D)\cdot G_{n-k}(D). \quad (5)$$

Equation 5 shows that the transmitted frame is divisible by the generator polynomial. At the receiving side it is divided by the same generator polynomial, and the quotient,

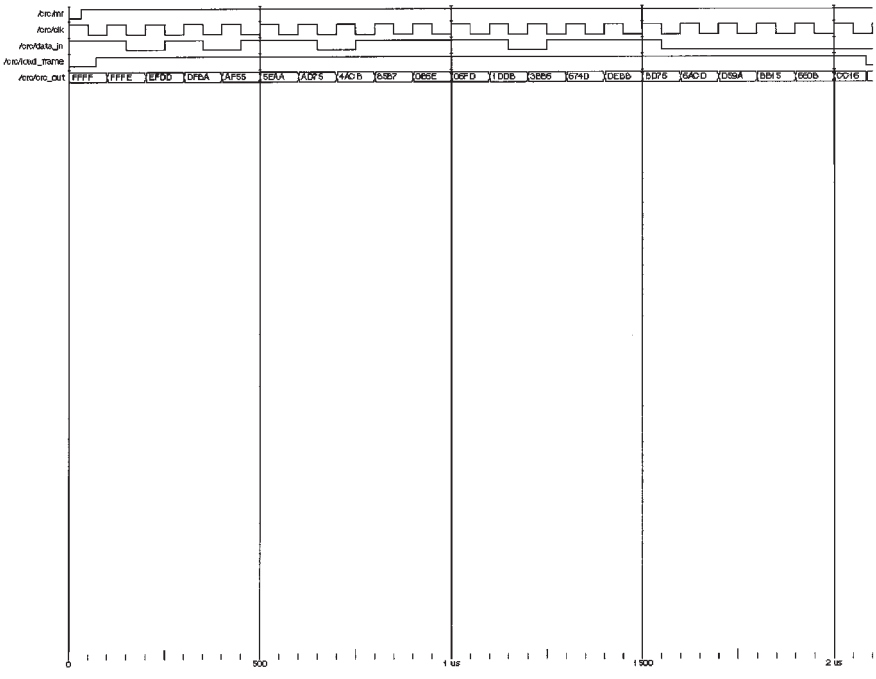


Figure 2 This simulation has a data-frame size of 20 bits.

$Q_k(D)$ is discarded. If no errors exist in the transmitted data, the remainder should always be zero. Both the transmitter and the receiver side must perform the same division by $G_{n-k}(D)$. The linear-feedback shift registers in **Figure 1** implement the division. When the k^{th} bit loads in, the shift-register value is the remainder. For more information about how the CRC scheme works, see **Reference 1**. An AND function performs multiplication by one or zero, and an XOR function performs the sum (subtraction) operation. **Listing 1** shows simple synthesizable VHDL code that generates the divider schematic.

The code includes constants that allow division by the generator polynomial. Note that **Listing 1** includes common polynomials from **Listing 2**, used as examples (see commented Constants) and also includes the relative initial value of the CRC shift register. Some standards dictate reversing the incoming bit stream (LSB first), the calculated CRC, or both. The signal load_frame, active high, defines the dimension of the serially loaded data frame. For example, the simulation in **Figure 2** has a data-frame size of 20 bits. At the 20th bit, load_frame goes low, reloading the initial value for the CRC shift registers in readiness for the next frame. The calculated CRC value is CC16. You can download **listings 1** and **2** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2553.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess. asp, enter No. 322 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Reference

1. Ross, N, "A painless guide to CRC error detection algorithms," ftp://ftp.rocksoft.com/papers/crc_v3.txt.

LISTING 1—SERIAL CRC CHECKER/GENERATOR

```
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE work.crc_pkg.ALL ;
--
ENTITY crc IS
      PORT (
            mr           : IN std_logic;
            clk          : IN std_logic;
              load_frame   : IN std_logic;
            data_in      : IN std_logic;
              crc_out      : OUT std_logic_vector((width_poly-1) DOWNTO 0)
              );

END crc;

ARCHITECTURE rtl OF crc IS

SIGNAL crc_shift          :std_logic_vector((width_poly-1) DOWNTO 0);
SIGNAL mr_and_load_frame  :std_logic ;

BEGIN

mr_and_load_frame <= mr AND load_frame ;

shift_register: PROCESS(mr_and_load_frame,clk)
 VARIABLE xor0  :std_logic;
 BEGIN
     IF (mr_and_load_frame = '0') THEN
        xor0 := '0';
        crc_shift <= crc_init ;
     ELSIF ((clk = '1') AND (clk'EVENT)) THEN
         xor0 := (data_in XOR crc_shift(width_poly-1));
         IF (load_frame = '1') THEN
            IF (xor0 = '0') THEN
                crc_shift  <= crc_shift(width_poly-2 DOWNTO 0) & xor0 ;
            ELSE
                crc_shift  <= (crc_shift(width_poly-2 DOWNTO 0) XOR
poly_gen(width_poly-1 DOWNTO 1)) & xor0 ;
            END IF;
        END IF;
     END IF;
END PROCESS shift_register;

crc_out      <= crc_shift;

END rtl;
```

LISTING 2—POLYNOMIALS AND RELATIVE INITIAL VALUE OF CRC SHIFT REGISTER

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL ;

PACKAGE crc_pkg IS

CONSTANT  width_poly : integer := 16;
--
-- (comment) CRC-16 CITT X25 standard :G(D) = D^16 + D^12 + D^5 + D^0 = (1-0001-
0000-0010-0001)
CONSTANT  poly_gen   : std_logic_vector((width_poly-1) DOWNTO 0) := X"1021" ;
CONSTANT  crc_init   : std_logic_vector((width_poly-1) DOWNTO 0) := X"FFFF" ;
-- (comment) CRC-16                    :G(D) = D^16 + D^15 + D^2 + D^0 = (1-1000-
0000-0000-0101)
--CONSTANT  poly_gen   : std_logic_vector((width_poly-1) DOWNTO 0) := X"8005" ;
--CONSTANT  crc_init   : std_logic_vector((width_poly-1) DOWNTO 0) := X"0000" ;
--
--CONSTANT  width_poly : integer := 32;
-- (comment) CRC-32  :G(D) = (1-0000-0100-1100-0001-0001-1101-1011-0111)
--CONSTANT  poly_gen   : std_logic_vector((width_poly-1) DOWNTO 0) :=
X"04C11DB7" ;
--CONSTANT  crc_init   : std_logic_vector((width_poly-1) DOWNTO 0) :=
X"FFFFFFFF" ;

END;
```

Data-acquisition setup measures everything

Matt Smith, Analog Devices, Limerick, Ireland

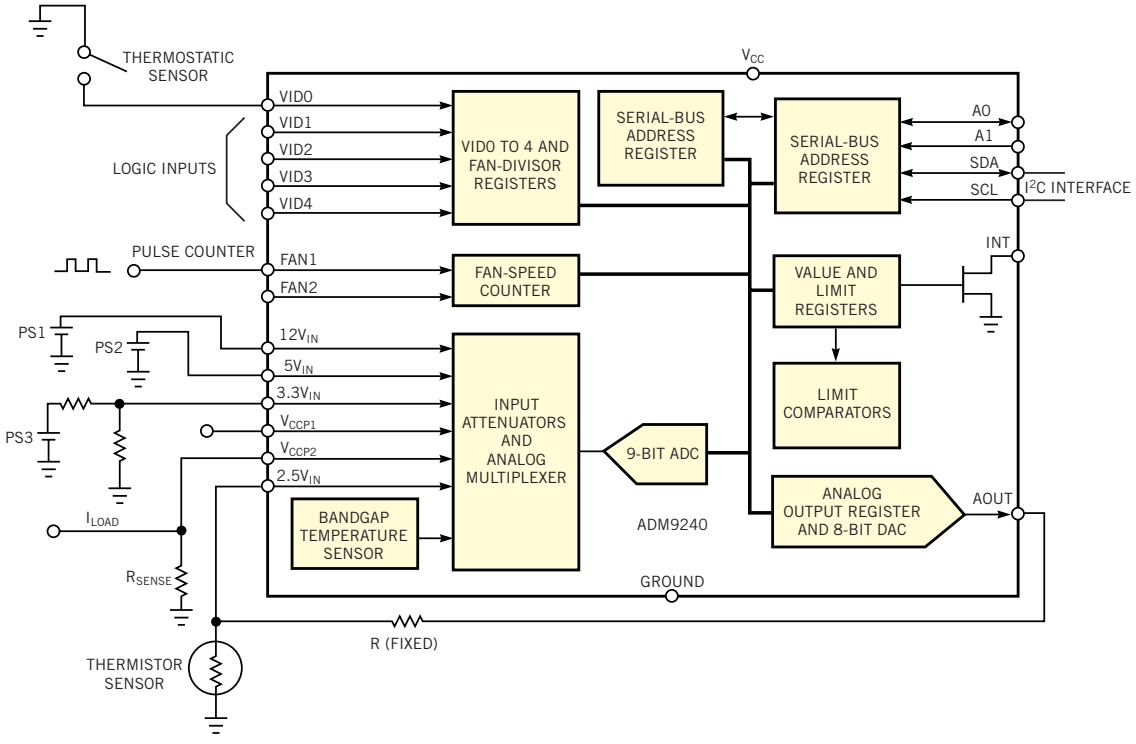
Using a product originally developed for PC-motherboard environmental monitoring, you can configure a low-cost, general-purpose DAS (data-acquisition system). The DAS in **Figure 1** can directly monitor multiple voltage channels, as well as temperature and frequency. It can also monitor digital sensors. Using only a few additional components, the system can accommodate other sensor and transducer elements. **Figure 1** shows one possible configuration. You can expand the scheme to cover additional input channels if necessary. For voltage sensing, the ADM9240 contains a multichannel ADC that can directly monitor as many as six input channels. The original intent of the IC was to monitor power supplies on PC motherboards, but it is flexible enough for general-purpose use. The maximum input-voltage ranges on the channels are 3.3, 3.6, 4.4, 6.64, and 16V. **Figure 1** shows the DAS monitoring two power supplies: PS1 and PS2. You can measure voltages greater than the channel range by using a simple voltage divider, as illustrated with PS3.

You can use the onboard DAC in the DAS, originally intended as a fan-speed controller, as a programmable, precision voltage reference. This configuration, for example, would facilitate measuring resistance-type sensors with the voltage-sensing channels. You could also use it as a bridge-excitation voltage source for accurate bridge-sensing applications. You can determine an unknown resistance value by setting the DAC's output voltage to a known level and using a known fixed resistance (**Figure 1**). You can implement current sensing by placing an accurate series resistor, R_{SENSE}, in the ground line and then monitoring the voltage drop across the resistor. An on-chip bandgap silicon sensor in the DAS provides temperature monitoring over the IC's −40 to +85°C operating range. You can use two frequency-monitoring channels in the DAS to monitor the pulsed digital output from a tachometer. The channels can also serve as general-purpose frequency counters.

The original intent of the five digital-input lines in the DAS was to monitor digital voltage-identification lines. You can use these lines as general-purpose input lines. They can sense high- or low-level status signals from digital sensors or from alarm channels. **Figure 1** shows a thermostatic sensor. You control and read the DAS using a simple two-wire SMBus or I²C interface to a µC or µP. If a dedicated I²C controller is unavailable, you can use a port "bit-banging" technique. Easy expansion is also possible by selecting a different device address. Using a different device-address bus needs no additional communication lines, because multiple devices may reside on the same bus.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 323 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and "Submit."

Figure 1



You can configure a DAS to simultaneously measure multichannel voltages, temperature, resistance, current, and frequency.

Algorithm tests for point location

Lawrence Arendt, Manitoba HVDC Research Centre, Winnipeg, Canada

A recent software project approximated the phase-space trajectory (also known as a strange attractor) of a certain dynamic system by using several nonoverlapping triangles. It became necessary to determine whether particular operating points were on or off that attractor. Determining whether a circle or a rectangle contains a point is trivial. A circle centered at point C and having a radius R contains a point P(x,y) only if $||CP|| < R$. A rectangle ABCD with opposite vertices $B(x_1,y_1)$ and $D(x_2,y_2)$ contains a point P(x,y) only if $x_1 < x < x_2$ and $y_1 < y < y_2$. However, it is not such a trivial task to determine whether triangle ABC, which has vertices $A(x_1,y_1)$, $B(x_2,y_2)$, and $C(x_3,y_3)$, contains point P(x,y) (**Figure 1**). You can simplify the task by calculating the triangle's area using the formula:

$$\text{AREA}(P_1P_2P_3) = \frac{1}{2}\begin{Vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{Vmatrix},$$

where P_1, P_2, and P_3 are the vertices. If you use the absolute value of the determinant, you need not be concerned whether $P_1P_2P_3$ are labeled in a counterclockwise fashion. To determine whether ABC contains P, calculate the area of ABC using **Equation 1** and then define three new triangles, each having as its vertices point P and two vertices of ABC.

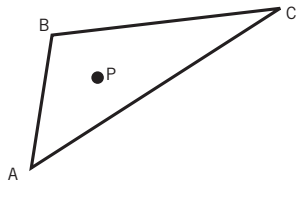
This operation results in three unique triangles: ABP, BCP, and CAP. You can then calculate the areas of triangles ABP, BCP, and CAP. If ABC contains P (**Figure 2**), then Area (ABP)+Area (BCP)+Area (CAP)=Area (ABC). If ABC does *not* contain P (**Figure 3**), then Area (ABP)+Area (BCP)+Area (CAP)>Area (ABC).

The beauty of the algorithm in **Listing 1** is that you need no squares, square roots, or trigonometric functions. You can extend the methodology to triangles in 3-D space. You can also adapt it for finding the area of irregular-shaped polygons. The C++ program in **Listing 1** is for a TTriangle class having a Contains(...) function similar to that of the Windows TRect class. TFPoint is the floating-point equivalent of the integer TPoint class. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and enter the Software Center to download the file for Design Idea #2566. **Is this the best Idea in this issue?** Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 324 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

LISTING 1–HIT TEST FOR TRIANGULAR REGIONS

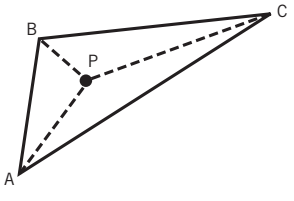
```
class TTriangle
{
    public:
        TTriangle (TPoint &PntA, TPoint &PntB, TPoint &PntC);
        BOOL Contains (TPoint &PntX);
        double Area (void) { return area; }
    private:
        TPoint PntA, PntB, PntC;   // vertices
        double area;
} ;
TTriangle::TTriangle(TPoint &pntA, TPoint &pntB, TPoint &pntC)
{
    PntA = pntA;
    PntB = pntB;
    PntC = pntC;
    area = 0.5*( ((PntB.x*PntC.y) - (PntC.x*PntB.y))
               -((PntA.x*PntC.y) - (PntC.x*PntA.y))
               +((PntA.x*PntB.y) - (PntB.x*PntA.y)));
    area = fabs(area);
}
BOOL TTriangle::Contains (TPoint &PntX)
{
    TTriangle ABX (PntA, PntB, PntX);
    TTriangle BCX (PntB, PntC, PntX);
    TTriangle CAX (PntC, PntA, PntX);
    double NewArea = ABX.Area()+BCX.Area()+CAX.Area();
    if (fabs(NewArea - area) > 1e-6)  return FALSE;
    return TRUE;
}
// an example
void main (void)
{
    TTriangle ABC (TPoint(1,1),TPoint(8,2), TPoint(5,9));
    TPoint X(4,4);
    if (ABC.Contains(X))   printf ("ABC contains X\n");
    else                   printf ("ABC does not contain X\n");
}
```

Figure 1



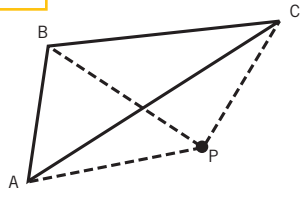
It is not a trivial task to determine whether a point P is within a triangle ABC.

Figure 2



If P is within triangle ABC, the three new triangles will be within ABC, too.

Figure 3



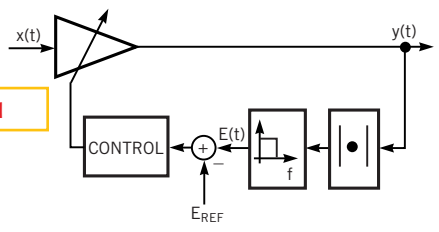
If P is outside triangle ABC, the three new triangles will extend beyond ABC's borders.

Digital potentiometer controls AGC circuit

Miguel Tavares, Goncalo Tavares, and Moises Piedade, INESC/IST, Lisbon, Portugal

AGC (automatic-gain-control) circuits are useful in applications ranging from automation to digital communications. **Figure 1** shows a typical AGC implementation with a feedback structure. The purpose of the circuit is to maintain a constant energy level at the output y(t). The output signal feeds a full-wave rectifier and then a filter to produce an estimate, E(t), of the signal energy. A subtracter weighs this energy signal against a preselected reference value. The difference signal causes the control circuit to vary the amplifier gain, to keep E(t) as close as possible to E_{REF}. **Figure 2** shows an AGC implementation, which uses a first-order lowpass filter with cutoff frequency f_C. The PGA (programmable-gain amplifier) uses a Xicor X9C103 digital potentiometer, a 100-tap, 10-kΩ device. Digital transitions on control-signal pin INC control the potentiometer's resistance, provided that $\overline{CS}$ is low. Transitions in control signal INC produce increments/decrements in the resistance value between pins RL and RW, depending on the value of the control signal U/$\overline{D}$.

Figure 1

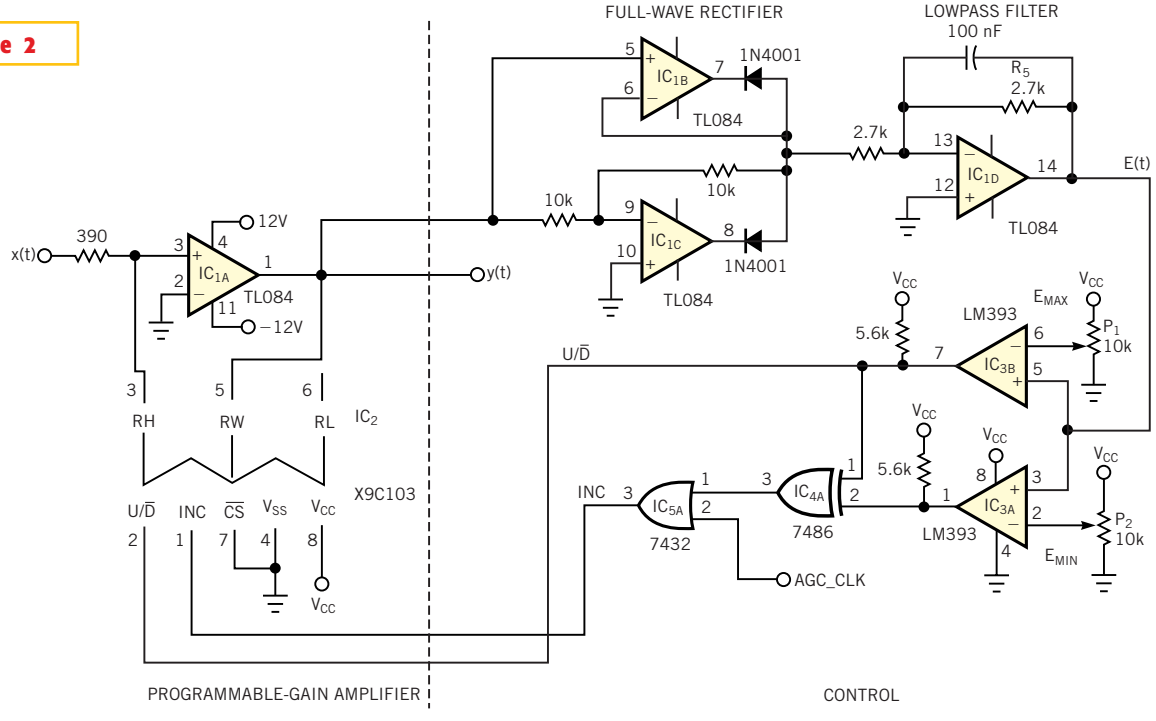


A classic AGC circuit keeps the output-energy level constant.

The control block in **Figure 1** works as follows: The potentiometers P_1 and P_2 establish two energy levels, E_{MAX} and E_{MIN}, which are slightly greater and lower, respectively, than the desired level E_{REF}. If the energy of y(t) is greater than E_{MAX}, the control circuit starts to reduce the resistance of the X9C103, causing the gain of the PGA to decrease. The inverse occurs when the energy of y(t) is lower than E_{MIN}. If the output signal's energy is between E_{MAX} and E_{MIN}, the control circuit ceases to alter the gain of the PGA, and the circuit operates in open-loop fashion. For the programming pulses (INC), use a fixed-frequency square-wave signal, which defines the adaptation rate of the PGA. **Figure 3** shows the experimental results. The input signal (**Figure 3a**) is a sinusoid with a frequency of 3840 Hz, amplitude-modulated by a 50-Hz square-wave signal. The maximum and minimum levels of the input signal are 1V and 200 mV, respectively. The cutoff frequency of the filter is 589 Hz. The programming pulses consist of a 7.68-kHz square-wave signal (readily available in the application at hand).

To keep the output signal's amplitude in the interval 0.9 to 1V, adjust the E_{MAX}

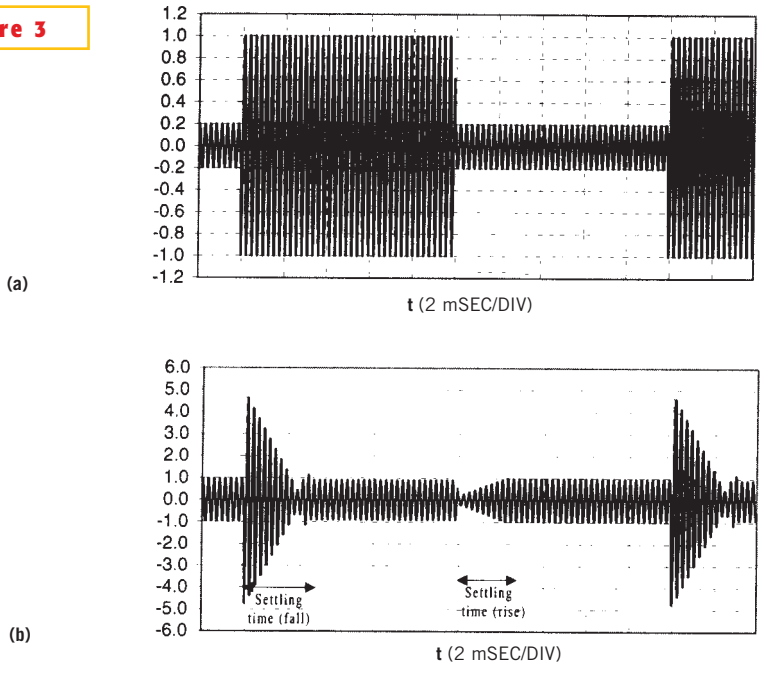
Figure 2



A digital potentiometer adjusts a PGA's gain to provide an AGC function.

and E_{MIN} levels to 0.64 and 0.57V, respectively. In a full-wave rectifier, with a sinusoidal input signal, the energy relates to the amplitude by $E=2A/\pi$. **Figure 3b**, the output of the AGC circuit, shows that the circuit behaves as expected. Note that the settling time of the gain changes depends not only on the frequency of the adaptation signal (INC) but also on the energy level of the input signal. The circuit obtained good performance for an input-signal amplitude in the interval 0.05 to 2.2V. This interval represents approximately 33 dB of dynamic range. The circuit is well-suited for implementation in systems using a DSP. You insert an ADC after the PGA, and the DSP performs all computations on the right side of the dotted line in **Figure 2**—full-wave rectifier, lowpass filter, and control. The DSP provides the digital control signals (INC and U/$\overline{D}$) to the PGA. **Is this the best Design Idea in this issue?** Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 325 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 3



The input signal (a) to Figure 2's circuit is a 3840-Hz sine wave modulated by a 50-Hz square wave. You can see the settling characteristics in (b).

Speed C functions for C16x µCs

Hans-Herbert Kirste, WAGO, Minden, Germany

The Infineon C16x Series of µCs use an internal 16-bit-register architecture. The architecture can also process byte-wide functions. The µC uses byte moves to implement library functions such as memset() or memcopy(). This process makes the library code independent of count values and address values; both may be odd or even. The disadvantage is the loss of speed because of the need to process loop counters for every byte. Memory access also cuts the speed, because two cycles are necessary to write a word in 2 bytes. The code in **Listing 1** implements a fast_memset() function that takes care of the byte count as well as the start access. Both may be odd or even. The function uses as many word moves as possible to preset the memory area. You can easily change the routine to process the copy function, memcopy(), with the same advantages. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2569. **Is this the best Idea in this issue?** Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 326 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

LISTING 1—FUNCTION-ACCELERATION CODE FOR C16X µCS

```
void far * fast_memset (  void far *pvDest, UCHAR ucVal, UINT uiSize )
{
UINT uiVal;
UINT far *p;
  if ( uiSize ) /* ignore function call if count is 0 */
  {
    p = pvDest;   /* keep a copy of the pointer */
    uiVal = ((UINT) ucVal << 8) + ucVal;  /* create a word to fill the
memory */
    if ( _pof ( p ) & 1 )/* if the start address is odd */
    {
      *(UCHAR far*)p = ucVal;   /* write a byte to the start address */
      p = (UINT far*) ((UCHAR far*) p + 1);   /* move pointer to next even
address */
      --uiSize;       /* count one byte written*/
    }
    USR0 = uiSize & 1;   /* USR0 = 1 if remaining count is odd */
    uiSize >>=1;         /* convert remaining byte count to word count */
    while ( uiSize )     /* write to all words */
    {
      *p = uiVal;          /* write a word */
      p += 1;              /* point to the next word */
      --uiSize;            /* decrement the word counter */
    }
    if ( USR0 )            /* if one byte is left*/
    {
      *(UCHAR far*)p = ucVal;       /* write to the last byte */
    }
  }
  return ( pvDest );     /* return the pointer to start of memory area   */
}
```

Instrumentation amp provides unipolar and bipolar outputs

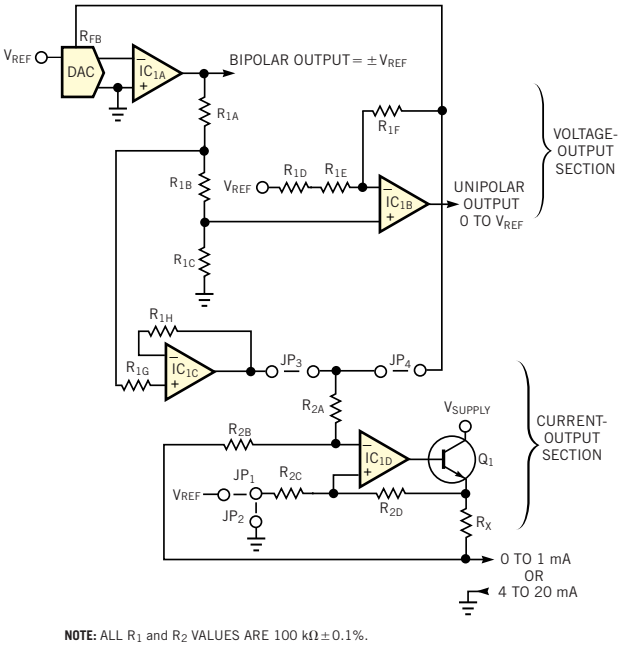
David Rathgeber, Alles Corp, Toronto, ON, Canada

Due to a translation error, many subscripts and mathematical symbols were typeset incorrectly in this Design Idea, which was published in the last issue (July 20, pg 150). We apologize to Mr Rathgeber and present the corrected version here.

Intelligent sensors and signal conditioners for plant-automation systems must produce outputs compatible with standard data-acquisition systems, such as PCs, programmable logic controllers, and remote terminals for supervisory and control systems. Standard analog inputs are 0 to 1V, ±10V, 0 to 1 mA, and 4 to 20 mA. The circuit in **Figure 1** makes all these signals available, using only three packages in addition to the DAC. The voltage-output section requires a thin-film resistor package, Beckman Series 668/698 (which tracks with temperature), and two FET-input op amps. The current-output section requires the addition of a four-resistor package, Beckman Series 664/694, and two more op amps. A reference voltage of −1.000V yields voltage outputs of 0 to 1V and ±1V and either 0 to 1 mA with R_X=1 kΩ and JP_2 and JP_4 connected or 4 to 20 mA with R_X=83.33Ω and JP_1 and JP_3 connected. A reference of −10.00V and appropriate values of R_X yield corresponding voltage and current outputs. All R_1 and R_2 values in **Figure 1** are 100 kΩ±0.1%. You should select Q_1 and possibly use a heat sink, taking into consideration the supply voltage and output-current range.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess. asp, enter No. 327 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



NOTE: ALL R_1 and R_2 VALUES ARE 100 kΩ±0.1%.

This circuit simultaneously provides unipolar and bipolar voltage outputs and two output-current ranges.

design ideas

Edited by Bill Travis and Anne Watson Swager

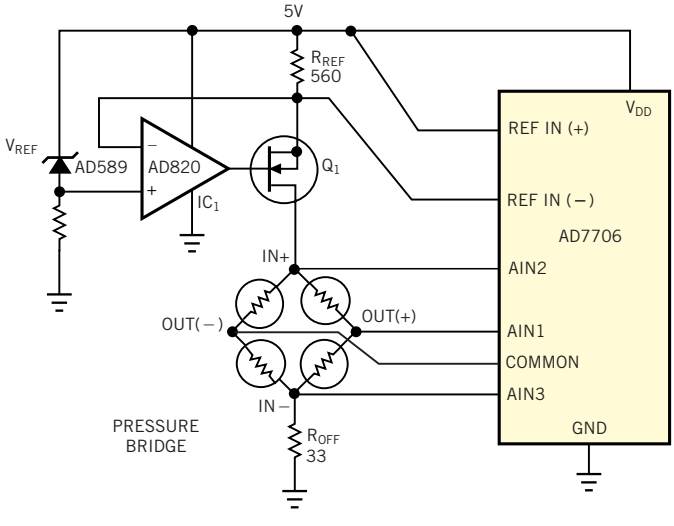
Bridge-temperature measurement allows software compensation

John Wynne, Analog Devices Inc, Limerick, Ireland

Bridge transducers are notoriously temperature-sensitive. When the temperature changes, almost everything else varies with some parameters increasing and some decreasing. The TCS (temperature coefficient of span) is generally negative for piezoelectric pressure sensors, and the TCR (temperature coefficient of resistance) is positive. In other words, as the temperature rises, the sensitivity decreases, and the resistance of the bridge increases. In general, the TCS and TCR are close to each other. This fact has in the past motivated designers to add both active and passive external components to achieve some measure of temperature compensation. However, the calibration of such systems can be tedious; the resulting performance, problematic. Piezoelectric-bridge manufacturers have tried to ease the problem by equalizing the coefficients in their manufacturing processes. Another approach is to simply measure the temperature of the bridge and then use a μC to compensate in software, given certain basic properties, such as the bridge resistance at 25°C and the TCR. This approach is effective, but concerns that the temperature you measure may not be the real temperature of the bridge hamper it. For instance, placing the temperature sensor relative to mechanical attachment of the strain gauge has a crucial bearing on the accuracy of the reading. It's not unusual to see errors of 1°C or more in such situations. The idea presented here (**Figure 1**), suggested in **Reference 1**, is to determine the temperature of the bridge by measuring the voltage across the bridge as a result of a known excitation current flowing through it.

R_{REF}, reference voltage V_{REF}, and op amp IC_1 determine the excitation current, which equals V_{REF}/R_{REF}. The differential bridge output, seen between terminals OUT(+) and OUT(−), feeds directly into channel AIN1 of the AD7706 ADC. This signal is a pseudodifferential input with respect to the ADC's Common input and can accept full-scale signals as low as 20 mV while providing full 16-bit performance. The AD7706 has three pseudodifferential inputs, all having the Common input as reference. The Common input connects to the midpoint of the bridge and serves a reference point from which to make all

Figure 1



You can provide software compensation for bridge-temperature coefficients through accurate temperature measurements.

the necessary measurements. Computing the voltage across the bridge entails two additional measurements. Input channel AIN2 measures the voltage from the top of the bridge to its midpoint (Common terminal), and input channel AIN3 measures the voltage from the midpoint to the bottom of the bridge. You can then compute the voltage across the bridge. Assume that the resistance of the bridge is independent of the pressure under measurement at least to the extent that the error is small compared with the measured results.

The AD7706 is a complete, 16-bit (with 14-bit maximum integral nonlinearity) delta-sigma ADC intended for dc and low-frequency ac measurements. Because its power consumption is 1 mW maximum at 3V, you can use the device in loop-powered, battery-powered, and local applications. The on-chip programmable-gain amplifier has gain settings of 1 to 128 to accommodate both low- and high-level analog inputs without the need for external signal-conditioning hardware. You can find additional information about the IC at http://products.analog.com/products/info.asp?product=AD7706. (DI #2576)

REFERENCE

1. Paillard, Bruno, "Temperature compensating an integrated pressure sensor," *Sensors*, January 1998, pg 36.

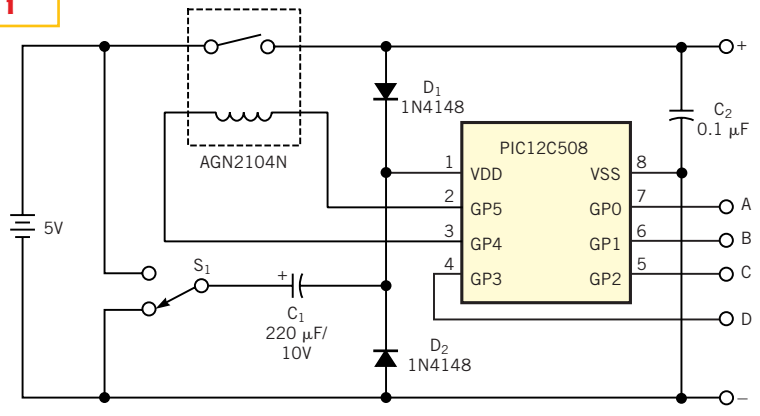
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 440 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Timer automatically shuts off

Yongping Xia, Teldata Inc, Los Angeles, CA

THE CIRCUIT IN **Figure 1** is a programmable power switch that offers on-state periods of 5 sec to three hours. It consumes zero power in the off state and has zero voltage drop in the on state. S_1 is a power-on switch. When you press S_1, the battery powers an eight-pin PIC12C508 µC. Because the value of C_1 is fairly large, it powers the µC long enough for the processor to send out a pulse through GP5 and GP4 to turn on a latching relay. Once the relay is on, the µC derives power through D_1. C_1 discharges through D_2 when you release the button. After turning on the relay, the µC reads the four inputs GP0 through GP3 to determine the delay time (**Table 1**). Once the time is over, an inverted pulse appears on GP5 and GP4 and turns off the latching relay. Once the relay turns off, the circuit consumes no power. You can download the PIC12C508 assembly program for the timer from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2571. (DI #2571)

Figure 1



This timer consumes no power in the off state and has zero voltage drop in the on state.

TABLE 1-DELAY TIMES

DCBA	Time
0000	10 sec
0001	15 sec
0010	20 sec
0011	30 sec
0100	45 sec
0101	One minute
0110	Two minutes
0111	Three minutes
1000	Five minutes
1001	10 minutes
1010	15 minutes
1011	30 minutes
1100	45 minutes
1101	One hour
1110	Two hours
1111	Three hours

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 441 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

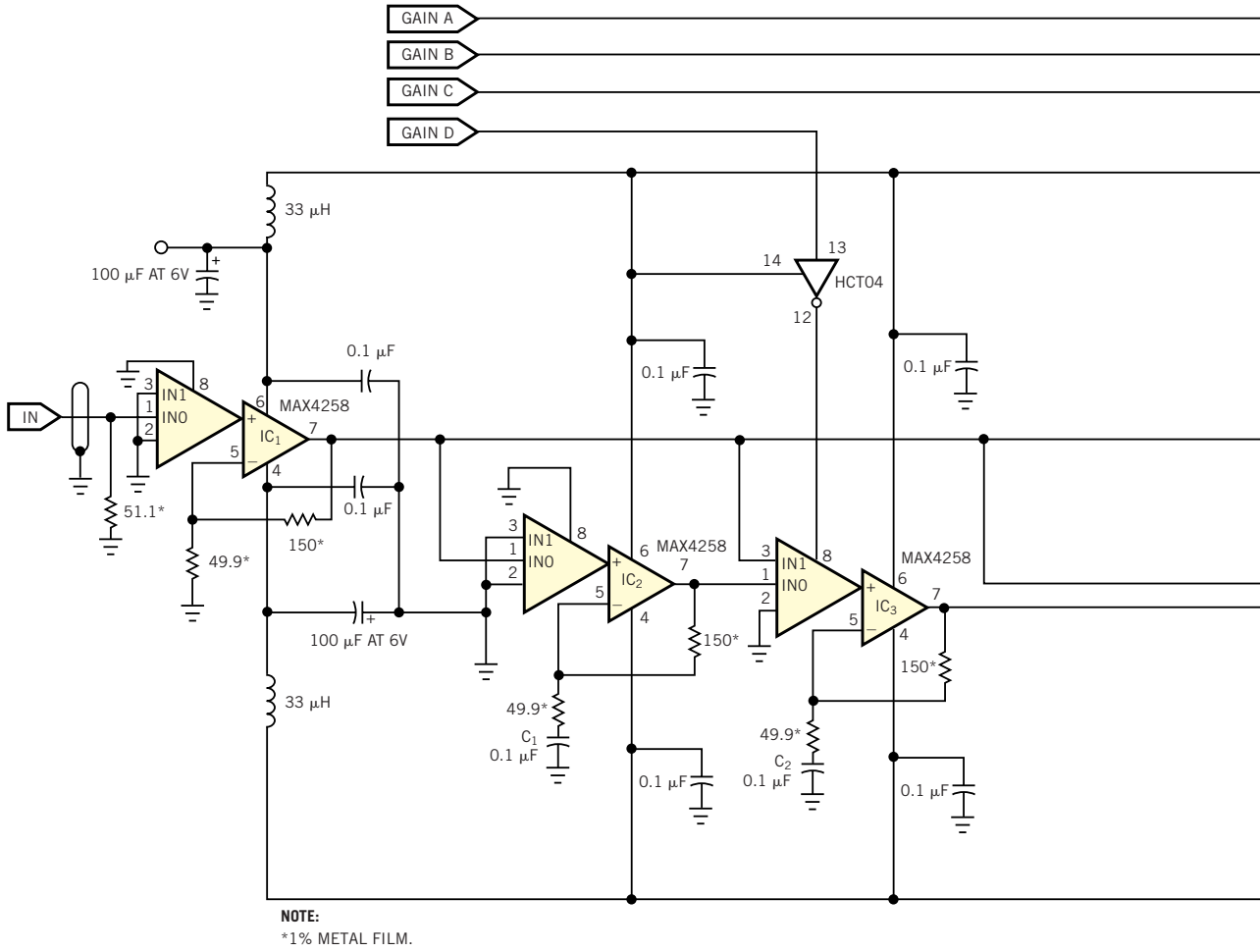
Video amplifier provides digital gain control

Stephen Woodward, University of North Carolina, Chapel Hill, NC

Programmable-gain amplifiers, many of them monolithic, are continuously improving, with lower noise and higher gain-bandwidth products. But for high-end applications, multichip circuits are still necessary. For example, the amplifier in **Figure 1** originally served as the photomultiplier preamplifier in a Doppler system intended for eventual operation in the Martian atmosphere. The design is based on a switchable array of six Maxim MAX4258 dual-channel video multiplexer-amplifiers, IC_1 through IC_6. Under control of a 4-bit input word, you can set the number of stages in the gain cascade to three, four, five, or six. Because each multiplexer-amp has a fixed gain of four, each unit change in the stage count changes the overall gain by 12 dB. In addition, you can interpose a 6-dB attenuator between the last and the next-to-last gain stages. The attenuator adds a factor-of-two resolution to the programmed gain and eight distinct gain settings (**Table 1**).

Gain-change settling time is lower than 30 nsec. The greater-than-100-MHz bandwidth of each stage results in an overall passband for all gain settings of greater than 60 MHz. The five lowpass-bandpass capacitors, C_1 through C_5, determine the lower end of the passband. In the original application, the frequen-

Figure 1



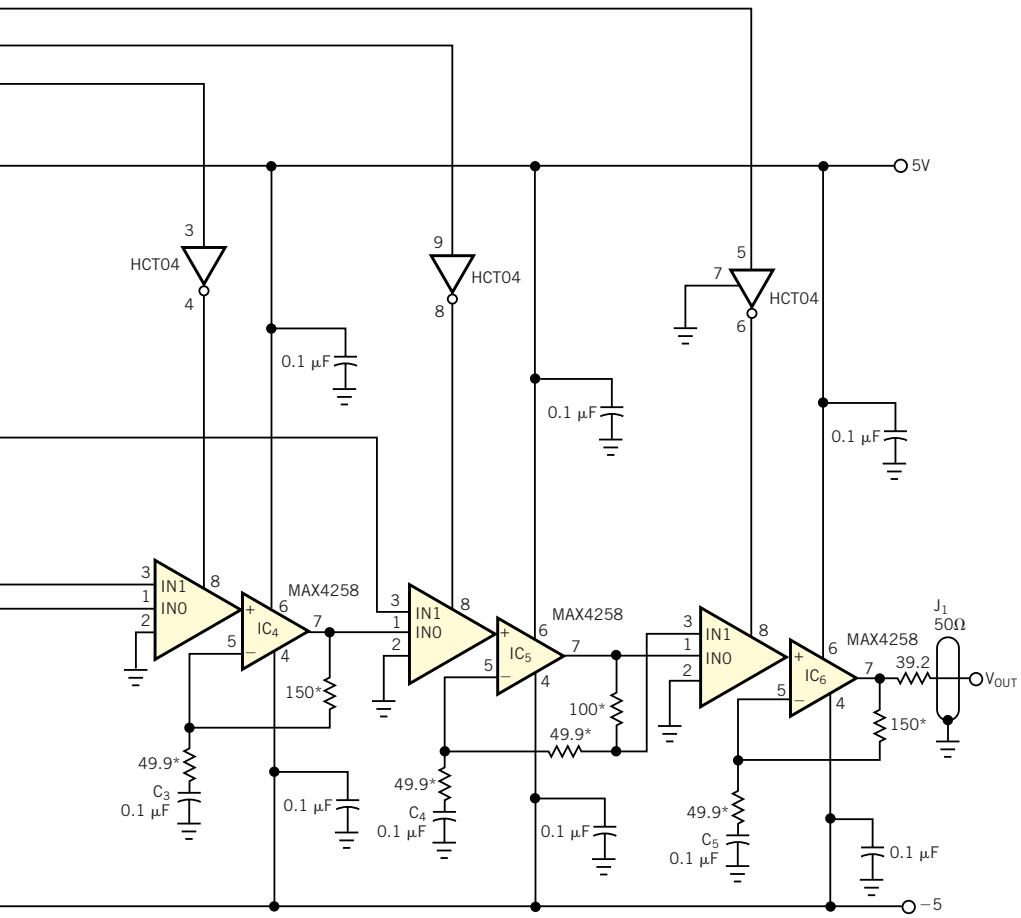
NOTE:
*1% METAL FILM.

You can obtain eight distinct gain settings with this low-noise, high-bandwidth video amplifier.

cies of interest don't extend much below 100 kHz. So, using 0.1-µF capacitors results in a roll-off of approximately 50 kHz. Larger values for these capacitors would reduce this figure. Omitting the capacitors for a dc response is not recommended, however, because the resulting amplification of IC_1's input offset would produce an output offset in the order of volts at high gain settings. You can generate the gain-programming word using any TTL-compatible, 8-bit parallel-I/O port, such as a PC's parallel printer port or an EIA-1284-compatible port. The HCT04 CMOS inverter chip in the gain-control pathway blocks any possible noise entry in the gain-control lines.

TABLE 1–DIGITAL GAIN SETTINGS

Code	D	C	B	A	Gain (into 50Ω, in decibels)
1	0	0	0	0	×16=24.1
2	0	0	0	1	×32=30.1
3	0	0	1	0	×64=36.1
4	0	0	1	1	×128=42.1
5	0	1	1	0	×256=48.2
6	0	1	1	1	×512=54.2
7	1	1	1	0	×1024=60.2
8	1	1	1	1	×2048=66.2



The overall input-referred voltage noise is approximately 2 nV per the square root of hertz, equivalent to the Johnson noise of a 250Ω resistor. Maximum output level is 15 dBm (3.6V p-p) into 50Ω and twice that into high-impedance loads. The combination of extreme gain and high-frequency response (gain-bandwidth products approaching 200 GHz) of this circuit mandates careful attention to issues of ground-plane and power-supply-bypass integrity. In addition, you must make every effort to minimize stray capacitance around the feedback-pin (Pin 5) components of all gain stages. (DI #2559)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 442 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

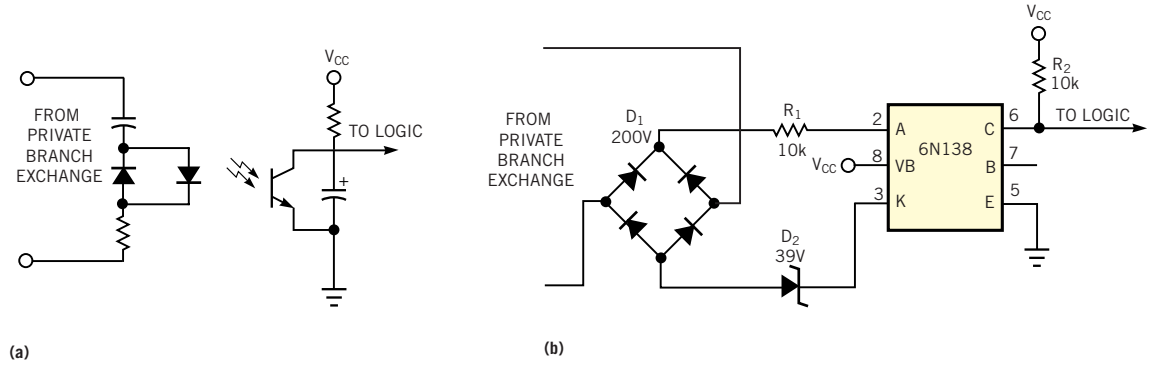
Simple circuit detects dc ringing

Jerzy Chrzaszcz, Warsaw Institute of Technology, Poland

Ring detection is a common task in telephony-related digital design. The goal is to sense an ac ringing voltage applied to the telephone line by a central office or PBX (private branch exchange) and then pass an appropriate signal to the µC or µP. **Figure 1a** shows a typical ac-coupled ring-detection circuit. The serially connected capacitor closes the ac path and the optoisolator provides galvanic isolation between the line and the logic. The output capacitor integrates the pulse train to form a ring envelope signal. More sophisticated designs may use the unfiltered output for ring-frequency discrimination. Unfortunately, some PBX systems signal ringing to extension phones by raising the dc line voltage. A traditional ac-coupled detector is thus useless. The circuit in **Figure 1b** consists of a full-wave rectifier, a current-limiting resistor, a zener diode, and a high-gain optoisolator with output pullup. By changing component values, you can easily meet circuit requirements for circuit isolation, input-voltage range, threshold voltage, and line load. (DI #2567)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 443 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



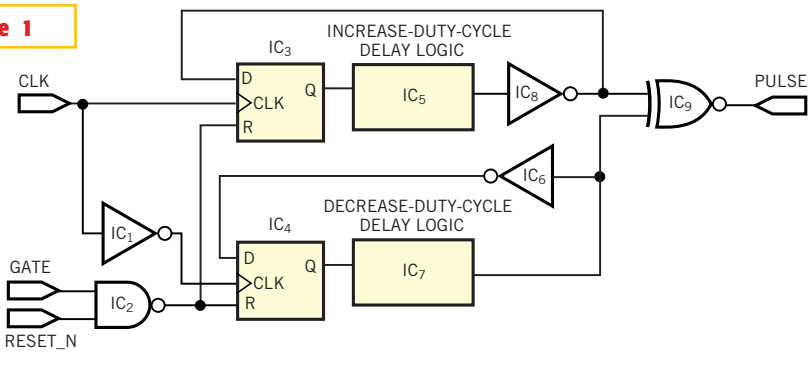
The ac-coupled circuit in (a) is useless in some PBX systems; the configuration in (b) takes over for PBX-generated dc ringing signals.

Gated clock has duty-cycle control

Paul Kemp, University of Colorado, Colorado Springs, CO

The circuit in **Figure 1** produces clock pulses with variable duty cycle from a gated clock. The output of the circuit, pulse, is always 180° out of phase with the clock input. When the delay-logic elements, IC_5 and IC_7, have the same propagation delays, the duty cycle of the circuit's output is 50%. The circuit produces gated clock pulses when the gate input, gate, is high and the active-low reset, reset_n, is high. Increasing the propagation delays of the increase-duty-cycle delay-logic element, IC_5, causes an increase in the duty cycle of the gated-clock output, pulse. The de-

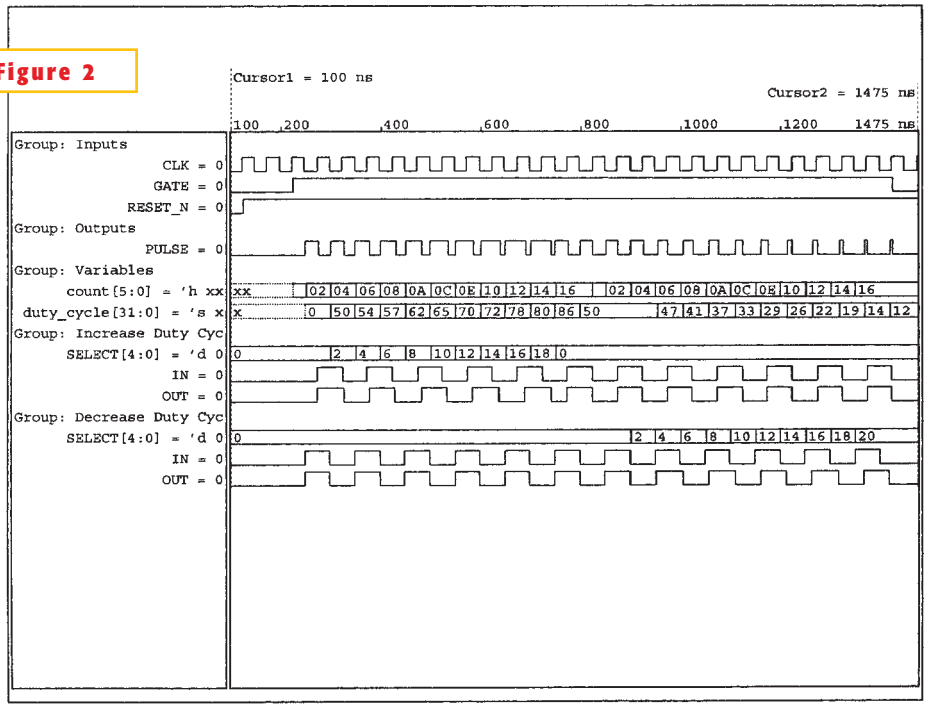
Figure 1



You can control the duty cycle of pulses from a gated clock.

crease-duty-cycle delay-logic element, IC_7, reduces the duty cycle of the gated-clock output. **Listing 1** is a Verilog model of the circuit; a Verilog testbench verifies the circuit's function. A pair of delay-logic elements is modeled with 20 delay lines of 1 nsec each. The testbench instantiated this operation. A simulation (**Figure 2**) used Cadence Verilog-XL. A 20-MHz clock serves as the input clock. The select signals in **Figure 2** for the increase-duty-cycle delay-logic element, IC_5, and the decrease-duty-cycle delay-logic element, IC_7, correspond to the number of 1-nsec delay elements inserted into the delay paths. Thus, a select value of 12 corresponds to a 12-nsec delay. You can download **Listing 1** and the Verilog testbench from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2554. (DI #2554)

Figure 2



Verilog simulates the function of the circuit in Figure 1.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 444 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

LISTING 1—GATED-CLOCK DUTY-CYCLE CONTROLLER

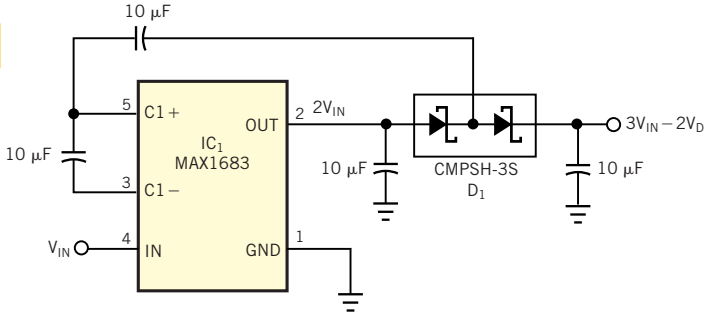
```
`timescale 1ns/1ps   // 1ns time units with 1ps resolution

module pulse_shaper (
                     CLK,           // I
                     GATE,          // I
                     GATE_DEC_DLY,  // I
                     GATE_INC_DLY,  // I
                     RESET_N,       // I
                     GATE_DEC,      // O
                     GATE_INC,      // O
                     PULSE          // O
                    );

//   Input Declarations
input        CLK;           // System clock
input        GATE;          // Active high signal produces variable duty cycle
pulses
input        GATE_DEC_DLY;  //
input        GATE_INC_DLY;  //
input        RESET_N;       // Active low asynchronous reset


//   Output Declarations
output       PULSE;         // Variable duty cycle output pulse
output       GATE_DEC;      //
output       GATE_INC;      //


//   Register Declarations
reg          GATE_DEC;      //
reg          GATE_INC;      //

//   Parameter Declarations
parameter    dly = 1;

// Generate a pulse from the system clock (CLK) and GATE signals.
assign #dly PULSE = GATE_INC_DLY ^ GATE_DEC_DLY;

always @(posedge CLK or negedge RESET_N or negedge GATE) begin
  if (~RESET_N) begin
    GATE_INC <= #dly 1'b0;
    end
  else if (~GATE) begin
    GATE_INC <= #dly 1'b0;
    end
  else begin
    GATE_INC <= #dly ~GATE_INC_DLY;
    end
  end

always @(negedge CLK or negedge RESET_N or negedge GATE) begin
  if (~RESET_N) begin
    GATE_DEC <= #dly 1'b0;
    end
  else if (~GATE) begin
    GATE_DEC <= #dly 1'b0;
    end
  else begin
    GATE_DEC <= #dly ~GATE_DEC_DLY;
    end
  end

endmodule
```

Tripler converts 5 to 15V

Ken Yang, Maxim Integrated Products, Sunnyvale, CA

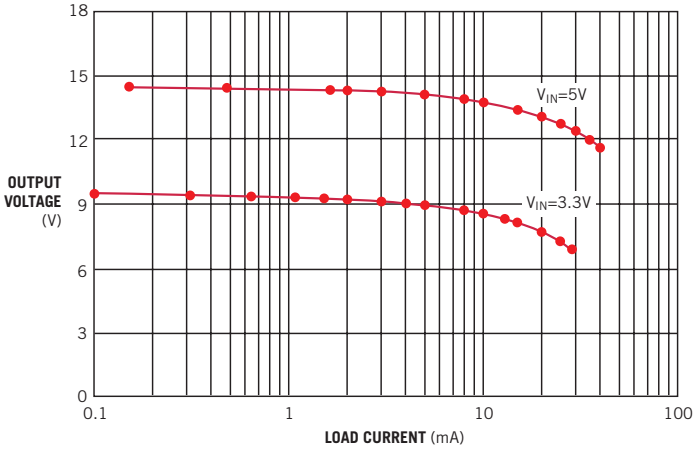
By configuring a charge-pump voltage doubler as a tripler, you can readily derive 15V from 5V (**Figure 1**). A 15V rail is useful for powering op amps, LCD-bias circuits and other low-current applications. The connections shown configure the IC_1 voltage doubler as a tripler. The no-load output voltage of the circuit is approximately $3V_{IN}-2V_D$, where V_D is the voltage drop across one diode. Use Schottky diodes as shown to minimize V_D and its effect on output voltage. Because the circuit's finite output impedance causes the output voltage to drop with load current (**Figure 2**), a practical limit for load current is approximately 30 mA. (DI #2570)

Figure 1



This circuit (almost) triples the input voltage for low-current applications.

Figure 2



Finite output impedance causes a decline in voltage with increasing load current.

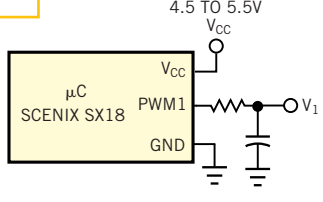
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 445 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Generate stabilized PWM signals

Martin Ossmann, FH, Aachen, Germany

A standard technique for generating analog voltages using µCs is to use a PWM output and filter the signal with a simple RC filter (**Figure 1**). The voltage of the PWM signal is directly proportional to the µC's supply voltage, so it is not necessarily clean or stable. To overcome this problem, you can use the circuit in **Figure 2**. Here, a 74HC14 Schmitt-trigger array serves as an output stage for three PWM signals from an SX18 µC. (The idea applies to other µCs, too.) The 74HC14 derives its supply voltage, V_{CCA}, from the stabilization circuit comprising the inexpensive shunt regulator, SR_1. You can adjust V_{CCA} by trimming R_1. The test circuit used V_{CCA}=4.096V. The PWM signals now have a stable amplitude that varies less than 0.1% when the µC's supply varies from 4.5 to 5.5V. Resistors R_3 to R_5 limit the current flowing from the µC through the 74HC14's input-protection diodes when V_{CCA} is too low. The values

Figure 1

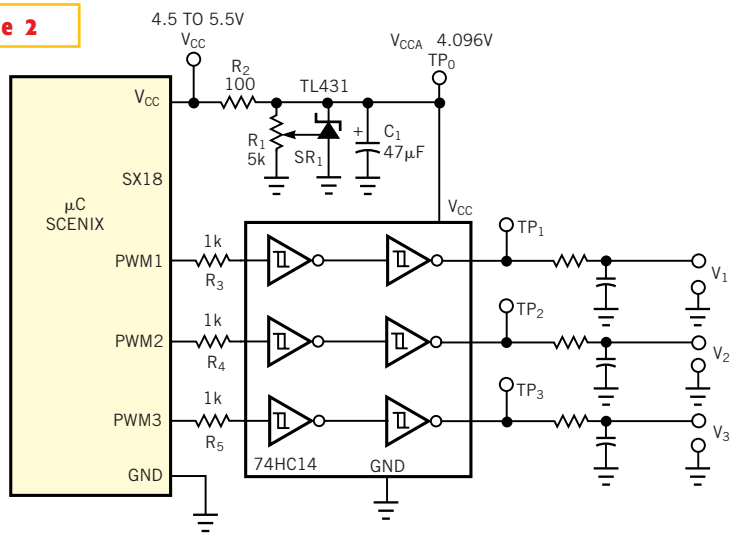


This classic PWM circuit suffers from supply-voltage sensitivity.

of R and C depend on the application. The test circuit uses 10 kΩ and 4.7 µF. If you feed multiple analog (or PWM) signals through a single IC, you usually encounter crosstalk. To characterize the circuit in **Figure 1** for internal crosstalk and unmatched delays, conduct the following tests.

Generated three PWM signals with different frequencies and 1-to-1 duty cycles. With a 0-dB reference-level square-wave signal at test point TP_1, crosstalk to test points TP_2 and TP_3 measures −70 dB. The first harmonic of the PWM signal (theoretically zero for a 1-to-1 duty cycle) is down 65 dB at test point TP_1. At test point TP_0, spurs are down 75 dB. So, the circuit in **Figure 1** has very good crosstalk characteristics. Also, if the duty cycle of one PWM channel changes, the influence on the voltage that other channels generates is less than 0.1%. You must take care to ensure that the switching delays of the 74HC14 do not change with varying V_{CC} applied to the µC. If the switching delays change with V_{CC} because of the changing levels of the driving signals, V_{CC} influences the generated output voltage, even if V_{CCA} is constant. You can use the circuit for precise generation of voltages, thanks to the temperature stability of the TL431. You can also use it for inexpensive implementations of sigma-delta converters, or to generate voltage-stabilized rectangular waveforms. (DI #2573)

Figure 2



Schmitt triggers with a stabilized supply make the PWM signals insensitive to supply voltage.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 446 in the "Circle Number" field and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Switched-capacitor converter suits portable applications

Clinton Jensen, National Semiconductor, Santa Clara, CA

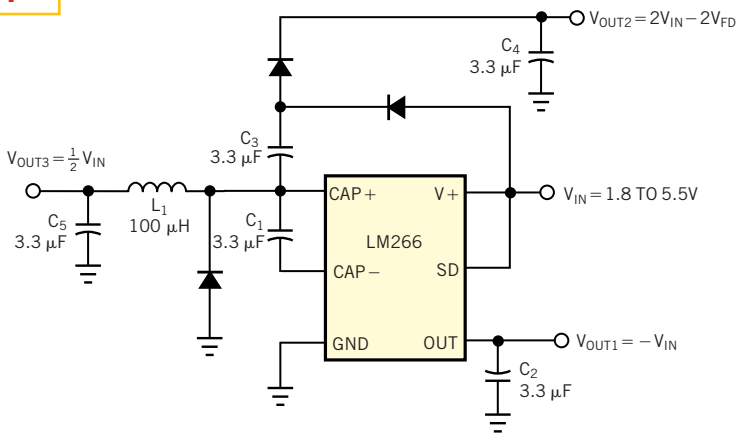
Basic switched converters generally provide one simple voltage conversion. The most common application for these circuits are to double, invert, and sometimes halve the input voltage. Because they are unregulated and have no stability problems, you can configure them to perform multiple conversions as well. However, because they have no inherent regulation, it is a good idea to use a regulated voltage as the input source. Multiple converters exhibit some voltage change at the output as a function of loading (because of the output resistance), but this drop is often acceptable. The circuit in **Figure 1** is a triple-output switched-capacitor circuit. You can configure the circuit using any switched-capacitor IC that's capable of inversion. This example uses the LM2664, but you can use larger ICs, such as the LM2661 or LM2663, to obtain higher current. The circuit in **Figure 1** can simultaneously invert, double, and halve the input voltage over an input range of 1.8 to 5.5V. The combined current capability of the three outputs equals the maximum load current of the IC in either of the standard topologies: inverting or doubling (40 mA for the LM2664 and 200 mA for the LM2663).

The output resistance for any output is equal to or less than the typical output resistance of the basic doubler or inverter. This last statement holds true only if you realize that the doubling output is actually $2V_{IN} - 2V_{FD}$, where V_{FD} is the forward drop of the diodes used. The doubling output simply uses two diodes to make a discrete charge pump in conjunction with the switch that connects CAP+ to V_{IN} during one cycle and to ground during the next cycle. The extra diode drops may be a problem in some applications, but they would be insignificant if you connected a linear, low-dropout regulator to the doubled output. The halving output uses the concept of an unregulat-

ed inductive step-down switcher with a constant 50% duty cycle. At no load, the output from a 50%-duty-cycle pulse filtered through the inductor and output capacitor is half the peak voltage of the square wave. The same switch used for the doubling function produces the square wave. In the topology of **Figure 1**, you always obtain the inverting function that is inherent to the IC, and you can use or omit the other two outputs as needed. (DI #2578)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 447 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



One switched-capacitor IC simultaneously inverts, doubles, and halves the input voltage.

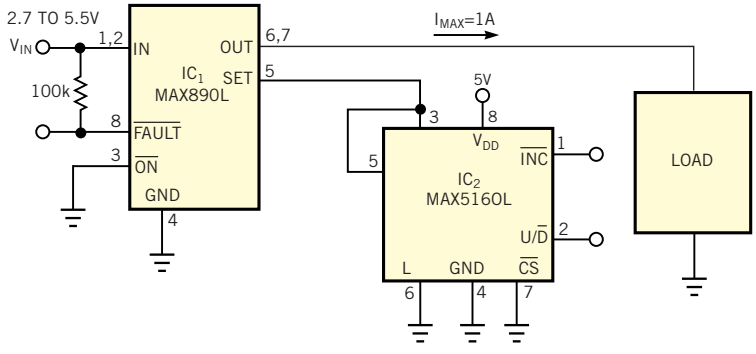
Current-limit switch is digitally programmable

Budge Ing, Maxim Integrated Products, Sunnyvale, CA

CURRENT-LIMIT SWITCHES are virtually ubiquitous in system controls. They provide a safe means for regulating the current delivered to a load circuit. The switches allow the load current to increase to a programmed limit but no higher. Typically, the current limit is a function of the voltage across an external resistor, produced by the current from a fixed source internal to the switch IC. This voltage serves as the reference for an internal current-limiting amplifier. By replacing the resistor with a digital potentiometer, you can easily program the current limit (**Figure 1**). IC_1 is a current-limit switch with a maximum programmable limit of 1A. The limit equals $1380/R_{SET}$, where R_{SET} is the resistance between pins 5 and 6 of IC_2. IC_2 is a 50-kΩ digital potentiometer whose resistance is programmable in 32 equal increments. With $\overline{CS}$ held low, high-to-low transitions at $\overline{INC}$ (Pin 1) increments IC_2's internal counter.

These transitions increase the resistance between W and L when U/$\overline{D}$ is low and reduce it when U/$\overline{D}$ is high. IC_1 includes a thermal-shutdown capability that turns the load current completely off when the chip temperature exceeds 135°C. It restores the load current when the temperature cools by 10°C. If the short-circuit fault is still present, the switch cycles off and on, yielding a pulsed load current. An open-drain Fault output (Pin 8) switches low when the load demands current beyond the programmed limit, enabling an external system to monitor the condition of the current switch. (DI #2577)

Figure 1



You can program a current limit to 1A in 32 equal increments by using a digital potentiometer.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 448 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

design ideas

Edited by Bill Travis and Anne Watson Swager

Current source has high compliance

Frank Vitaljic, Bellingham, WA

Current sources are useful in many areas of electronics, such as voltage and current division, current transmitters, excitation of thermistors, RTDs, bridges, potentiometers, and circuit biasing. Current sources are either fixed- or adjustable-current types. For the dual-op-amp current source in **Figure 1**, the load current, I_L, is adjustable by varying V_{IN} (0 to ±1.2V), according to the following equation:

$$I_L = \underbrace{\left[-\frac{V_{IN}}{R_{SENSE}}\right]}_{\text{IDEAL}} + \underbrace{\left[\frac{V_{OS1}+V_{OS2}}{R_{SENSE}}\right]}_{\text{OFFSET ERROR}}.$$

The maximum variation in load current is ±6 mA for a 200Ω current-sensing resistor, R_{SENSE}. This variation yields a maximum load voltage of ±12V (3V less than the supply voltages). The maximum current-offset error is 0.01 mA for V_{OS1} and V_{OS2} of 1 mV. This amount of offset is normal for lower grade op amps. Note that you connect one end of the load to virtual ground (VG). Amplifier IC_1 acts as a voltage-error driver, and the transconductance amplifier, IC_2, senses the load current and converts it to V_A. The control equations are:

$$(V_A - V_{IN})\,G_A = Z_L I_L \qquad (1)$$

$$\left(1+\frac{1}{G_B}\right)V_A = -R_{SENSE} I_L, \qquad (2)$$

where G equals open-loop gain; and

$$\frac{G_0}{1+(G_0/W)\,s}, \qquad (3)$$

where G_0 = dc gain, W = unity-gain bandwidth, and s = jω complex frequency.

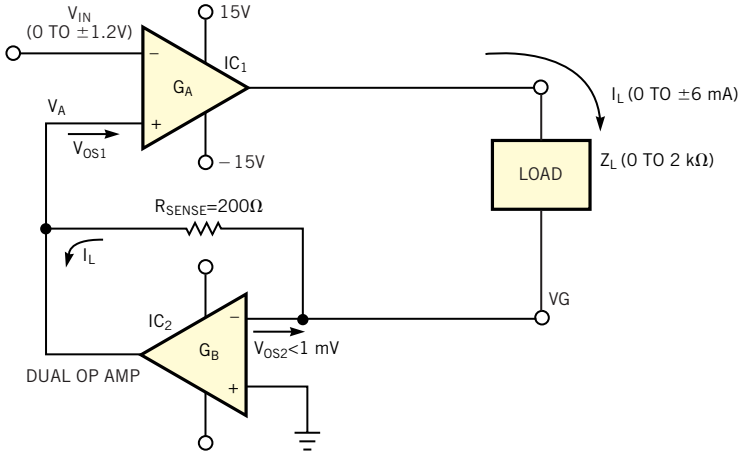
Solving **equations 1** and **2** for I_L and substituting G in **Equation 3** for $G_A G_B$, the load current becomes:

$$I_L = \frac{-WV_{IN}}{R_{SENSE}W + Z_L s} \approx \frac{-V_{IN}}{R_{SENSE}} \text{ for } R_{SENSE}W >> Z_L W,$$

which demonstrates the load current's insensitivity to the load, Z_L. (DI #2579)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 435 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



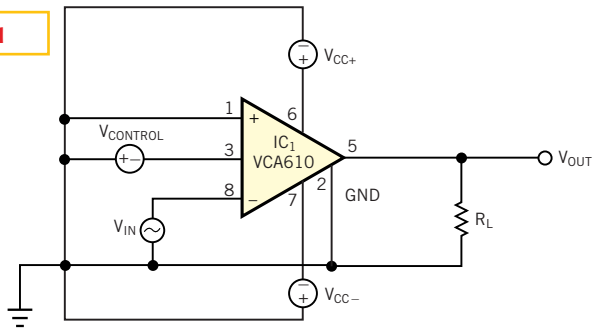
A ±6-mA current source has a ±12V compliance range.

Modification improves VCA's Spice simulation

Alex Rysin, RSVI Acuity CiMatrix, Canton, MA

THE BURR-BROWN VCA610 is a good wideband voltage-controlled amplifier (VCA) that you can use successfully for your application. The Spice model from Burr-Brown simulates normally when the VCA uses dual power supplies (**Figure 1**). However, when you attempt to use the Spice model in a single-supply application, which is equivalent to moving the GND symbol to the negative V_{CC} pin of IC_1 (**Figure 2**), you get an error message indicating that the circuit fails to converge. To find the source of the problem, restore the original schematic used for the Spice-model creation of the VCA610 and carefully analyze it. The model has several problems that prevent you from using it in single-supply simulations. The main problem arises from the faulty usage of "0" nodes in the model description. Spice considers subcircuit nodes as "local" and assigns them new names for simulations. This trait of Spice holds true for all subcircuit nodes except for ground nodes named "0," which are global for the entire simulation file. You can use the ground node "0" in subcircuit descriptions, but you must be careful that the node does not create unwanted dc references in a circuit that is intended to be floating. To avoid problems, do not to use "0" nodes in the model description but rather use different names, such as GND or AGND. Otherwise, you must carefully separate them the "0" nodes from dc connections that may affect the internal voltages and currents of the subcircuit. You can use control sources to separate the nodes. **Listing 1** is Burr-Brown's model of the VCA610.

Figure 1



With dual supplies for the VCA610, the Spice simulation converges.

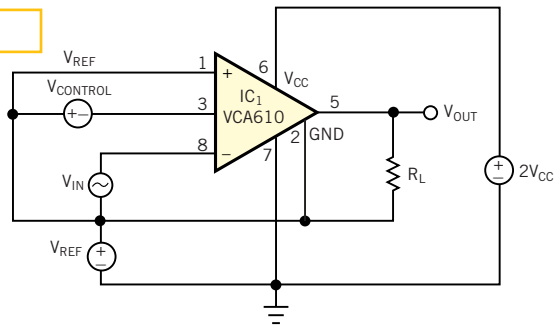
In **Listing 1**, all entries with problematical usage of the global node "0" appear in boldface. The line "**E1 11 7 POLY (1)(3,0)0.45 -0.11911**" is a polynomial voltage-controlled voltage source con-

LISTING 1—VCA610 SIMULATION FOR DUAL SUPPLIES

```
* CONNECTIONS:                NON-INVERTING INPUT
*                          | GROUND
*                                  |   | GAIN CONTROL, VC
*                              |   |   | OUTPUT
*                              |   |   |   | POSITIVE SUPPLY VOLTAGE
*                              |   |   |   |   | NEGATIVE SUPPLY VOLTAGE
*                              |   |   |   |   |   | INVERTING INPUT
*                              |   |   |   |   |   |   |
.SUBCKT  VCA610M/BB             1 2 3   5   6   7   8
* CONTROL VOLTAGE
Q1   7   3   13   P
C1   3   7   1E-12
Q2   7   2   13   P
I1   6   13   384E-6

Q3   10   11   7   N
R2   6   10   2

E1   11   7   POLY(1) (3,0) 0.45   -0.11911
G3   12   0   POLY(1) (10,6) 0 1
R3   12   0   139
C3   12   0   1.145E-9
G1   6   7   POLY(1) (6,10) 13.5102E-3 -0.489
G2   0   7   POLY(1) (6,10) 1.7958E-3 2.939E-3

* INPUT STAGE
Q01   20   1   26   N
C01   1   0   1E-12
Q02   21   8   26   N
C02   8   0   1E-12
R01   20   27   1E3
D01   29   27   DX
D03   6   29   DX
R02   21   28   1E3
D02   24   28   DX
D04   6   24   DX
IS   26   7   2.32E-3

* GAIN STAGE 1
R31   31   0   1E6

* GAIN STAGE 2
R41   41   44   20E3
C41   41   44   265.25E-15
G41   41   44   0   31   1E-3
D41   41   43   DX
E41   44   43   POLY(1) (3,0) 100.2 14.87

R42   41   45   20E3
C42   41   45   265.25E-15
G42   41   45   0   31   1E-3
D42   42   41   DX
E42   42   45   POLY(1)   (3,0) 100.2 14.87

E43   44   0   6   0   20
E44   0   45   0   7   20

* OUTPUT STAGE
E51   55   0 41 0 50E-3
D53   55   51   DX
D54   52   55   DX
D55   6   53   DX
D56   6   54   DX
D57   7   53   DZ
D58   7   54   DZ
G54   53   7   5   55   50E-3
G53   54   7   55   5   50E-3
V53   51   5   0.1833
V54   5   52   0.1833
G51   5   6   6   55   50E-3
G52   7   5   55   7   50E-3
R53   6   5   20
R54   7   5   20
.MODEL N NPN (IS=1E-12 BF=193)
.MODEL P PNP (IS=1E-12 BF=96)
.MODEL   DX   D(IS=1E-15 BV=200)
.MODEL   DZ   D(IS=1E-15 BV=50)
.ENDS
```

trolled by the voltage applied between nodes 3 (the gain-control input Pin 3) and the global "0." If you use the subcircuit in the dual-supply circuit of **Figure 1**, and Pin 2 (GND) connects to the common-point ground, then the voltage at the node (Pin 3) creates the correct output signal in the controlled source, E1. If you use the model for single-supply applications (**Figure 2**), then Pin 3 receives a significant dc-bias component, and E1 produces erroneous results. A similar problem occurs with other components and their nodes, designated by boldface type in **Listing 1**. To resolve the problem, give the "0" nodes different names. In this case, you can replace them with the other ground node of the model—the node "2." Note that not every "0" node creates a problem. You need take into account only those "0" nodes that have a dc reference to external connections. Other "0" replacements are unnecessary. After the cited corrections, the VCA610 model in **Listing 2** yields acceptable results in simulation. Unfortunately, it still has a convergence problem when you include it in a large schematic. You can significantly improve convergence by adding a 0.1 to 0.5Ω series resistor in the DX and DZ diode models. This addition makes the model more realistic and helps the Newton-Raphson algorithm that Spice uses to find an initial solution for the simulation. You can download **Listings 1** and **2** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2572.

Figure 2



With single-supply operation, the normal Spice model has convergence problems.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 436 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

LISTING 2–VCA SIMULATION FOR SINGLE-SUPPLY OPERATION

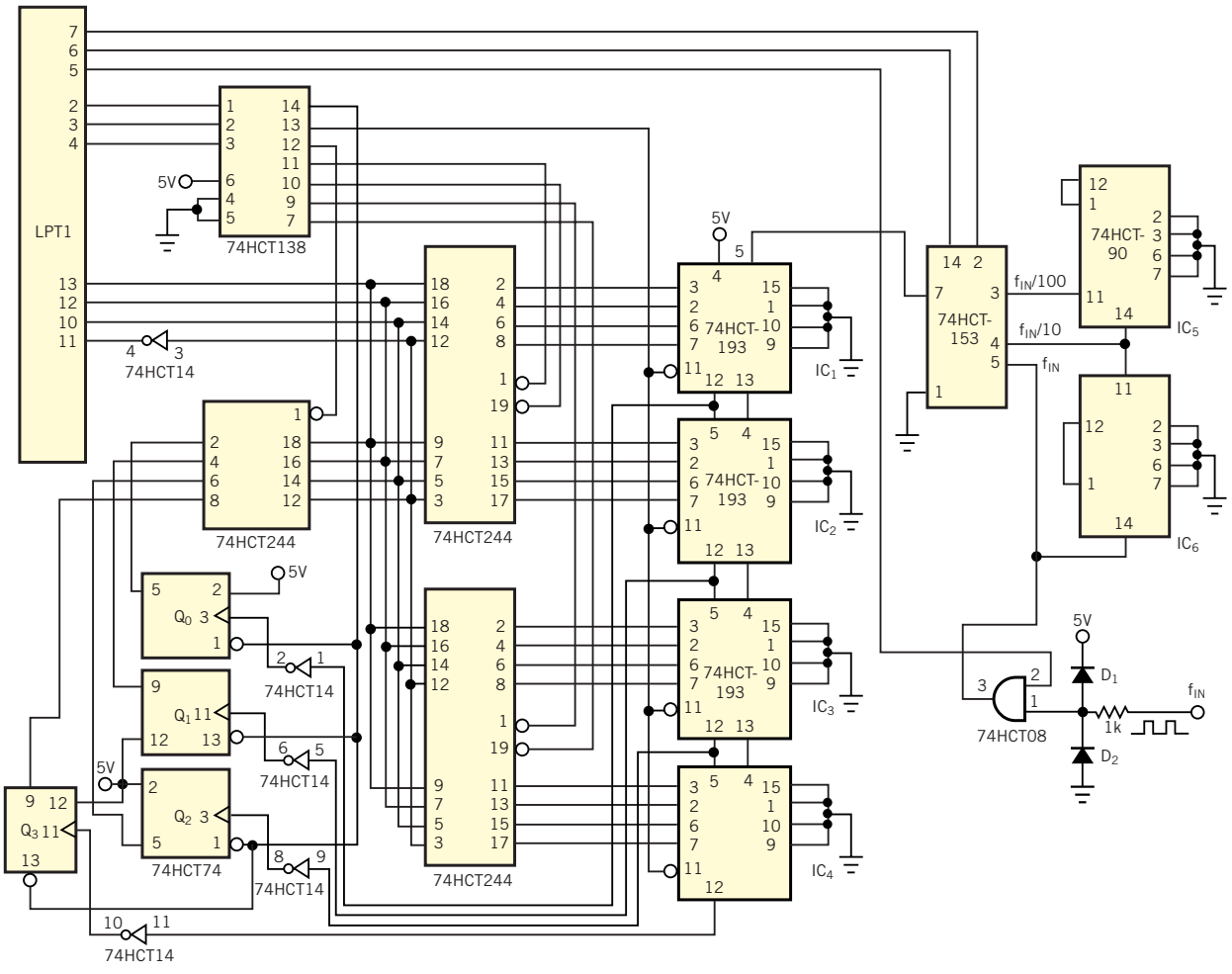
```
* CONNECTIONS:                NON-INVERTING INPUT
*                            | GROUND
*                            |    | GAIN CONTROL, VC
*                            |    |    | OUTPUT
*                            |    |    |    | POSITIVE SUPPLY VOLTAGE
*                            |    |    |    |    | NEGATIVE SUPPLY VOLTAGE
*                            |    |    |    |    |    | INVERTING INPUT
*                            |    |    |    |    |    |    |
.SUBCKT  VCA610M/BB                1 2 3  5  6  7  8
* CONTROL VOLTAGE
Q1   7   3   13   P
C1   3   7   1E-12
Q2   7   2   13   P
I1   6   13   384E-6
Q3   10   11   7   N
R2   6   10   2
E1   11   7   POLY(1) (3,2) 0.45   -0.11911
G3   12   0   POLY(1) (10,6) 0 1
R3   12   0   139
C3   12   0   1.145E-9
G1   6   7   POLY(1) (6,10) 13.5102E-3 -0.489
G2   0   7   POLY(1) (6,10) 1.7958E-3 2.939E-3

* INPUT STAGE
Q01   20   1   26   N
C01   1   0   1E-12
Q02   21   8   26   N
C02   8   0   1E-12
R01   20   27   1E3
D01   29   27   DX
D03   6   29   DX
R02   21   28   1E3
D02   24   28   DX
D04   6   24   DX
IS   26   7   2.32E-3

* GAIN STAGE 1
R31   31   0   1E6
G31   31   0   POLY(2) (8,1) (12,0) 0 0 0 0 1E-6 0

* GAIN STAGE 2
R41   41   44   20E3
C41   41   44   265.25E-15
G41   41   44   0   31   1E-3
D41   41   43   DX
E41   44   43   POLY(1) (3,2) 100.2 14.87
R42   41   45   20E3
C42   41   45   265.25E-15
G42   41   45   0   31   1E-3
D42   42   41   DX
E43   44   0   6   2   20
E42   42   45   POLY(1)   (3,2) 100.2 14.87
E44   0   45   2   7   20

* OUTPUT STAGE
E51   55   2 41 0 50E-3

D53   55   51   DX
D54   52   55   DX
D55   6   53   DX
D56   6   54   DX
D57   7   53   DZ
D58   7   54   DZ
G54   53   7   5   55   50E-3
G53   54   7   55   5   50E-3
V53   51   5   0.1833
V54   5   52   0.1833
G51   5   6   6   55   50E-3
G52   7   5   55   7   50E-3
R53   6   5   20
R54   7   5   20

.MODEL N NPN (IS=1E-12 BF=193)
.MODEL P PNP (IS=1E-12 BF=96)
.MODEL   DX   D(IS=1E-15 BV=200 Rs=.1)
.MODEL   DZ   D(IS=1E-15 BV=50 Rs=.1)
.ENDS
```

BIOS INT1Ch measures frequency

K Suresh, Indira Ghandi Centre for Atomic Research, Tamil Nadu, India

THE DESIGN IN **Figure 1** shows how you can exploit BIOS INT1Ch of a PC, which occurs 18.2 times/sec, along with a few inexpensive components and LPT1, to measure frequencies from 1 Hz to into the megahertz range. The design requires no expensive general-purpose or tailor-made data-acquisition card/logger. It also overcomes the disadvantage of continued interrupt of the PC in the interrupt-based frequency-measurement method, in which the speed of the PC decides the maximum frequency. The design exploits a few resources of the PC—for example, BIOS INT1Ch (without affecting its normal interrupt-service routine) and the Data and Status ports of LPT1—to measure frequency independently of the speed of the PC. The input pulses, with frequency f_{IN}, drive a 16-bit up/down counter, comprising four 4-bit counters (IC_1 to IC_4, 74HCT193) in cascade. The drive path includes a 74HCT08 AND gate and a 74HCT153 4-to-1 digital multiplexer. The multiplexer feeds any one of the frequency inputs, f_{IN}, $f_{IN}/10$, or $f_{IN}/100$, from the decade counters (IC_5

Figure 1



By using BIOS INT1Ch, you can measure frequency, using few resources of the host PC.

and IC_6, 74HCT90) to the counter.

The output of the counters and the flip-flops' Q outputs connect to the Status port (at 0x379h) of the LPT1 port through the 74HCT244 buffers. A 74HCT138 3-to-8 encoder and the digital multiplexer, controlled from the Data port (at 0x378h), provide various control signals, as dictated by the software in **Listing 1**. This simple Turbo C program controls the frequency measurement. The BIOS INT1Ch generates two timing windows—SWINDOW of approximately 100 msec, and MWINDOW of 1 sec—by incrementing the SINTR and MINTR variables (set to zero initially), respectively.

At the end of every SWINDOW, the software determines the range of the input frequency f_{IN}—hertz, kilohertz, or megahertz. The routine then selects a range, so that the number of input pulses per second nearly equals, but stays within, the counter's capacity. Once the routine selects the range, the program executes the MWINDOW routine to measure the frequency. At the start, disabling the AND gate inhibits the input pulses, and the routine clears the counters and flip-flops. The program sets the digital multiplexer to apply the input frequency, f_{IN}, to the counters. The next INT1Ch enables the AND gate and increments SINTR and MINTR by executing the SWINDOW routine. When each counter's output crosses from 0x0Fh to 0x00h during count-up, its Carry output sets the D flip-flop (74HCT74) to logic 1. When SINTR reaches 2 (SWINDOW=109 msec), the routine inhibits the input pulses and reads the SNIBBLE at the flip-flop's outputs ($Q_3Q_2Q_1Q_0$). If SNIBBLE equals 0x07h or greater, the program sets the digital multiplexer to apply $f_{IN}/10$ to the counters. The routine then generates SWINDOW. The routine then again reads SNIBBLE to determine whether to apply $f_{IN}/100$ (if SNIBBLE≥0x07h) or $f_{IN}/10$ (if SNIBBLE<0x07h) to the counters. The loop continues until SNIBBLE, read at the end of SWINDOW, becomes less than 0x07h.

At any stage, if SNIBBLE is less than 0x07h, MWINDOW generates a timing

LISTING 1—TURBO C ROUTINE FOR FREQUENCY MEASUREMENT

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<process.h>
#include<math.h>
#define INT1C   0x1C                /*INT 1Ch*/

int MINTR=0,SINTR=0,RANGE=0,MF=1;   /* Sample and Measure windows Variable*/
float  FREQ=0.0,CF=18.2/18;         /* Measured frequency& correction factor*/
int CUWORD=0,MCR,DP,SP,a,b=0x10;
float y1 = 0.0,y2=0.0;


void interrupt (*oldvect)();                /* INT1Ch pointer*/
void interrupt TIMER();                     /*Routine for INT1Ch*/
unsigned int  FREQREAD();       /*Routine declaration for reading the counter output*/
int SWINDOW();                  /*Routine declaration for generation of 100msec*/
void MWINDOW(void);             /*Routine declaration for generation of 1 sec*/

void interrupt  TIMER()   /*Our ISR for INT1Ch*/

{
    disable();
    outp(DP, (b+0x08));  / *Enable AND gate and feed input pulses*/
    SINTR++;             /*Increment Continues till SINTR  becomes 2*/
    MINTR++;             /*Increment continues till MINTR becomes 18*/
    oldvect();
    enable();
}

int  SWINDOW()          /*Routine  for generation of 100msec*/

{
 oldvect=getvect(0x1C);             /*store vector address of INT1Ch*/
 outp(DP,(b+0x00));                 /*Disable AND and inhibit pulses to counter*/
 outp(DP,b);                        /*set the digital multiplexer*/
 outp(DP,(b+0x02));                 /* Load  the counters with 0x00h*/
 outp(DP,(b+0x01));                 /*Clear all Flip Flops*/

 while(SINTR<=2)                    /*Wait for  generation of app 100msec*/
                    {
                     setvect(0x1C,TIMER);
                    }

 outp(DP,(b+0x00)); /*Disable AND*/

 RANGE++;
 setvect(0x1C,oldvect);
 SINTR=0;
 MINTR=0;
 outp(DP,b+0x03);
 a=((inp(SP)>>4) & 0x0F); /*Read the SNIBBLE*/
 return a;
}

void MWINDOW()                  /*Routine for generation of 1 sec*/

{
 oldvect=getvect(0x1C);             /*store vector address of INT1Ch*/
 outp(DP,(b+0x00));                 /*Disable AND and inhibit pulses to counter*/
 outp(DP,(b+0x02));                 /* Clear  the counters */
 outp(DP,(b+0x01));                 /*Clear all Flip Flops*/
 outp(DP,b);                        / *set the digital multiplexer*/

while(MINTR<=18) /*Wait for generation of  0.989sec*/
                    {
                     setvect(0x1C,TIMER);
                    }
outp(DP,(b+0x00));                  /*Disable AND*/
setvect(0x1C,oldvect);
FREQREAD();
SINTR=0;
MINTR=0;
return 0;
}


unsigned int FREQREAD()
{
    unsigned char nib0,nib1,nib2,nib3,byte1,byte2;
    int temp,i;
    y1=0.0;
    y2=0.0;
    byte1=0;
    byte2=0;
    outp(DP,b+0x03);
    a=((inp(SP)>>4) & 0x07);        /*Read the SNIBBLE*/

    switch(a)
      {
        case 0: /*Counter 1 alone counted, read nibble 0*/
        outp(DP,(b+0x04));
        nib0=inp(SP);           /*Read nibble 0*/
        nib0=nib0>>4;
        byte1=nib0|0x00;
        byte2=0;
```

(continued on pg 156)

window of 0.989 sec by incrementing MINTR with each INT1Ch from 0 to 18. During this incrementation, the routine counts the frequency pulses from the digital multiplexer. The program then corrects the timing window to 1 sec by applying a correction factor (CF=18.2/18). At the end of MWINDOW, the program inhibits the pulses and again reads SNIBBLE. SNIBBLE indicates the progress of the counters during the MWINDOW interval. This operation relieves the PC from reading all the counters; instead, it reads only the counters that actually did the counting. The routine thereby reduces the number of read operations and subsequent manipulations. The FREQREAD routine reads the counters' output, a nibble at a time, starting with Counter 1 (IC_1, nib0), depending on the last-read SNIBBLE, to Counter 4 (IC_4, nib3). The program manipulates the nibbles and displays the frequency in hertz. At the end of a measurement cycle, the routine sets SINTR and MINTR to zero and clears the counters and flip-flops to make them ready for another measurement cycle. You can use the design for frequencies of 1 Hz to 7 MHz. However, you can extend the range to 25 MHz or greater by increasing the counter capacity to 20 bits or more (for example, by using two 74HCT4040s in cascade) and by using appropriate FAST logic devices. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2575.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 437 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

LISTING 1—TURBO C ROUTINE FOR FREQUENCY MEASUREMENT (*Continued*)

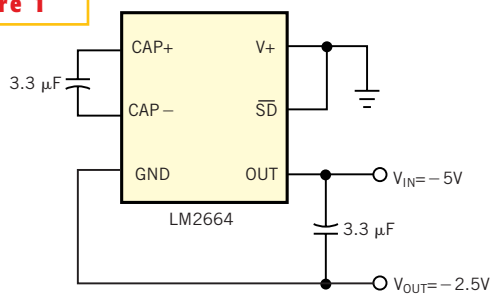
```
        break;

 case 1:/*Counter 1 & 2 involved in counting*/
        outp(DP,(b+0x04));
        nib0=inp(SP)>>4;          /*Read nibble 0*/
        outp(DP,(b+0x05));
        nib1=inp(SP)&0xF0;        /*Read nibble 1*/
        byte1=nib1|nib0;
        byte2=0;
        break;

 case 3: /*counters1,2 & 3 involved in counting* /
        outp(DP,(b+0x04));
        nib0=inp(SP)>>4;          /*Read nibble 0*/
        outp(DP,(b+0x05));
        nib1=inp(SP)&0xF0;        /*Read nibble 1*/
        outp(DP,(b+0x06));
        nib2=inp(SP)>>4;          /*Read nibble 2*/
        byte1=nib1|nib0;
        byte2=nib2&0x0F;
        break;

 case 7:  /*all counters involved in counting* /
        outp(DP,b+0x04);
        nib0=inp(SP)>>4;          /*Read nibble 0*/
        outp(DP,b+0x05);
        nib1=inp(SP)&0xF0;        /*Read nibble 1*/
        outp(DP,b+0x06);
        nib2=inp(SP)>>4;          /*Read nibble 2*/
        outp(DP,b+0x07);
        nib3=inp(SP)&0xF0;        /*Read nibble 3*/
        byte1 = nib1|nib0;
        byte2 = nib3|nib2;
        break;

default:
        printf("\n Error ! ...Check the flip flops,....exiting");
        exit(1);
    }
  /* Convert binary words byte1,byte2 to decimal*/

        for(i=8; i<16;i++)
            {
                temp=byte2;
                byte2=byte2 &(0x01);
                y1=y1+byte2*pow(2,i);
                byte2=temp;
                byte2=byte2>>1;
            }
        for(i=0; i<8;i++)

            {
                temp=byte1;
                byte1=byte1 &(0x01);
                y2=y2+byte1*pow(2,i);
                byte1=temp;
                byte1=byte1>>1;
            }
        CUWORD=(y1+y2);
        FREQ= CUWORD*CF*MF;
        gotoxy(10,10);
        printf("\nFREQ in Hz=%.1f", FREQ); /*Display freq in Hz*/
        return 0;
}

int main()          /*Main Program starts here*/
{
clrscr();
printf("\n\tBeyond its time of day, INT1Ch measures frequency");
printf("\n\t\t\t\t  by");
printf("\n\tK.Suresh,MSD,IGCAR,Kalpakkam,Tamil Nadu,India");
DP = peek(0x40,8); /*check up for availability of  printer port */
if  (DP==0)
            {
                printf("\n\n\n LPT NOT AVAILABLE! EXITING");
                exit(1);
            }
printf("\n\n\n LPT1 address =0x%X",DP);/* Address of DATAPORT*/
SP = DP+1;                  /*Address of  STATUS PORT*/

while(!kbhit())
    {

        SWINDOW();
        MF=1;
        if (a >=0x07 && RANGE==1)
                {
                  b = 0x20;       /*Apply fin/10 to the counters*/
                  MF=10;
                  SWINDOW();

                      if(a>= 7 && RANGE==2)
                        {
                            b=0x30;       /*Apply fin/100 to the counters*/
                            MF=100;
                            SWINDOW();

                              if (a>=0x07 && RANGE>2)
                                {
                                    printf ("\n over range!.... exiting");
                                    setvect(0x1C,oldvect);

                                    exit(1);
                                }

                        }

                }
        MWINDOW();
        RANGE=0;

    }     /*Freq measurement continues till a key is pressed*/
setvect(0x1C, oldvect);
return 0;
}        /*End of Main Program*/
```

Circuit variations produce negative voltages

Clinton Jensen, National Semiconductor Corp, Santa Clara, CA

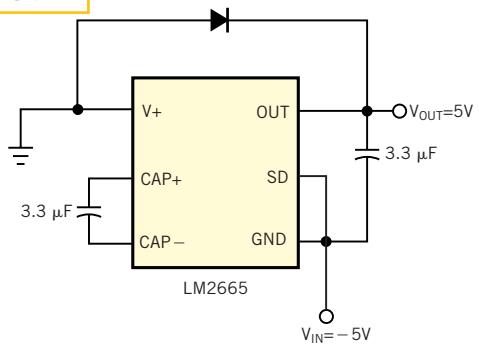
Basic switched-capacitor converters generally provide one simple conversion. They commonly double, usually invert, and sometimes halve a positive input voltage. Because they are not regulated converters and do not have stability problems, you can easily configure them to also do some negative conversions. However, given that they are unregulated, it is a good idea to use a regulated input voltage to obtain a predictable output. All these converters suffer from some voltage change on the output as a result of loading, but this voltage drop is often acceptable. The potential space and cost savings are often worth the trade-off if the circuit fits the application needs. **Figures 1** and **2** present two application circuits. The first is useful in systems that require multiple negative voltages. You can set up a basic switched-capacitor converter capable of inversion to halve a negative voltage. **Figure 1** shows the schematic of the circuit, using an LM2664 as an example. The circuit is capable of the full rated load (40 mA, in this case). The operating-voltage range is −3.6 to −11V. The circuit can be useful for creating multiple bias voltages for op amps, power amplifiers, or displays.

Figure 1



A switched-capacitor converter handily splits a negative voltage in half.

Figure 2



You can use a switched-capacitor converter to derive a positive voltage from a negative one.

The second application is useful when a negative voltage is present and you need an equal but inverted (positive) voltage. In **Figure 2**, an LM2665 produces a 5V output from a −5V input. The topology shown works with any switched-capacitor circuit designed for the doubling function. This circuit can also supply the full rated load current and works with an input-voltage range of −2.5 to −5.5V. The output resistance of the circuit equals that of the basic positive doubler. The circuit can be useful for op-amp biasing when only a negative voltage is present. Another benefit is that, if you have a loosely regulated negative supply for an op amp, the positive supply tracks the negative supply and keeps the output dc-biased near to or at ground. Another possible application is in systems using two-cell lithium-ion batteries or four alkaline batteries and needing ±5V supplies. (DI #2580)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 438 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

µC takes control of comparator

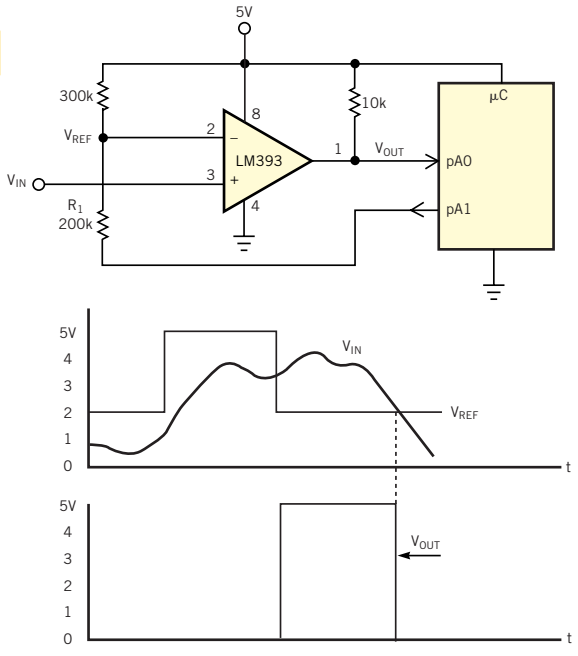
Abel Raynus, Armatron International, Melrose, MA

In many µC applications, a signal conditioner modifies an analog input signal from a sensor and passes it to the µC for processing. The last stage of the signal conditioner is always a device that transforms the signal level to a value acceptable for the µC's input. Usually, it is a comparator. Often, this signal channel is open or closed for some programmable time period. For this purpose, you can use a variety of analog switches. **Figure 1** shows an alternative: a cost-effective technique that needs no additional switches. The circuit exploits the fact that the µC's output pin can work as a pro-

grammable SPDT switch, connecting the lower end of R_1 either to 5V or to ground. When you program the µC's output pin (pA1 in **Figure 1**) to a low level, R_1 connects to ground, and the predetermined reference voltage connects to Pin 2 of the comparator. In **Figure 2**, V_{REF}=2V. The channel is open, and for input signals greater than V_{REF}, the comparator provides 5V to the µC's Pin pA0. When the output pin pA1 is high, the reference voltage becomes 5V, and the comparator's output switches to 0V. The channel is then closed. (DI #2582)

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/infoaccess.asp, enter No. 439 in the "Circle Number" field, and hit the Search bar. Put a tick in the "Select" box and hit "Submit."

Figure 1



You can use a µC to control a comparator, thus avoiding the need for analog switches.

design ideas

Edited by Bill Travis and Anne Watson Swager

Low-cost bias circuits serve HF and VHF bands

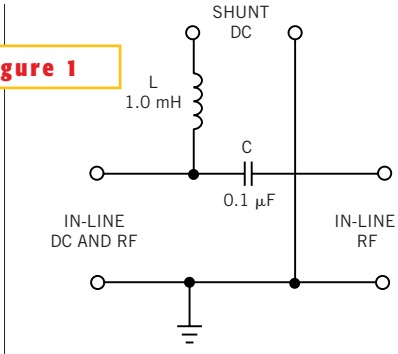
Richard M Kurzrok, RMK Consultants, Queens Village, NY

Bias circuits are passive networks that you use to apply dc excitation to various active circuits. The monitor tee, which is also known as the bias tee, has been commercially available for more than 40 years at microwave frequencies. The original products provide useful frequency ranges of two to one through five to one. More recent cost-effective bias tees cover 0.1 MHz to more than 4 GHz. Other designs are available that extend well into higher microwave-frequency bands (Reference 1). Another bias circuit is the bias-passing attenuator, which is also commercially available at microwave frequencies.

You can realize simple bias circuits at HF and VHF frequencies with minimal engineering and at much lower cost than those that must operate at microwave frequencies. You can obtain usable performance over a frequency range exceeding two decades. You can optimize the cost of these bias circuits by integrating them into subsystems and systems using surface-mount fabrication.

The monitor tee is a three-port network (**Figure 1**). One inline port handles both dc and RF. The second inline port handles only RF. The shunt port passes only dc. With values of L=1 mH and C=0.1 µF, the measured insertion loss from 0.5 to 100 MHz for inline transmission is less than 0.2 dB in a 50Ω test setup. The shunt port terminates in 50Ω.

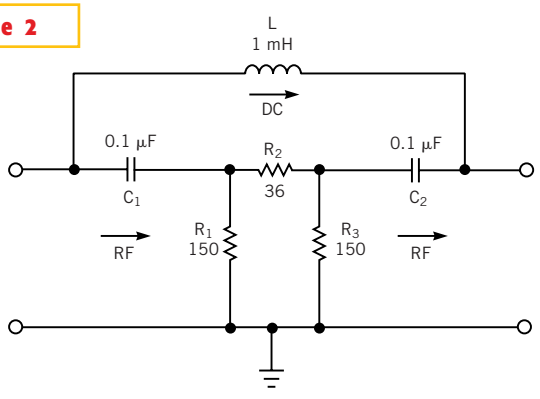
Figure 1



With L and C values of 1 mH and 0.1 µF, respectively, this monitor-tee network exhibits a measured insertion loss of 0.5 to 100 MHz for inline transmission of less than 0.2 dB in a 50Ω test setup.

The bias passing attenuator is a two-port network containing a fixed pad attenuator, input and output dc blocking capacitors, and a bridging RF choke (**Figure 2**). The values for L, C_1, C_2, and R_1 to R_3 in the **figure** create a nominally 6-dB attenuator. From 0.5 to 100 MHz, measured insertion loss is 6±0.5 dB in a 50Ω test setup.

Figure 2



This bias-passing attenuator has a measured insertion loss of 0.5 to 100 MHz of 6±0.5 dB in a 50Ω test setup.

Other bias-passing circuits include lowpass, highpass, and bandpass filters. The LC lowpass filter has inherent bias-passing capability via the cascaded series inductors. The LC highpass filter needs additional circuit elements for bias passing. Sometimes, this bias-passing circuit substantially degrades the stopband selectivity of the highpass filter. You can design the LC bandpass filter for bias passing using coupled shunt resonators. The filter input and output couplings must all be inductive. Also, the shunt inductors of the individual resonators must be floating with series RF bypass capacitors for ground returns.

Reference

1. Andrews, JR, "Broadband Coaxial Bias Tees," Application Note AN-1d, Picosecond Pulse Labs, Copyright February 1998.

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

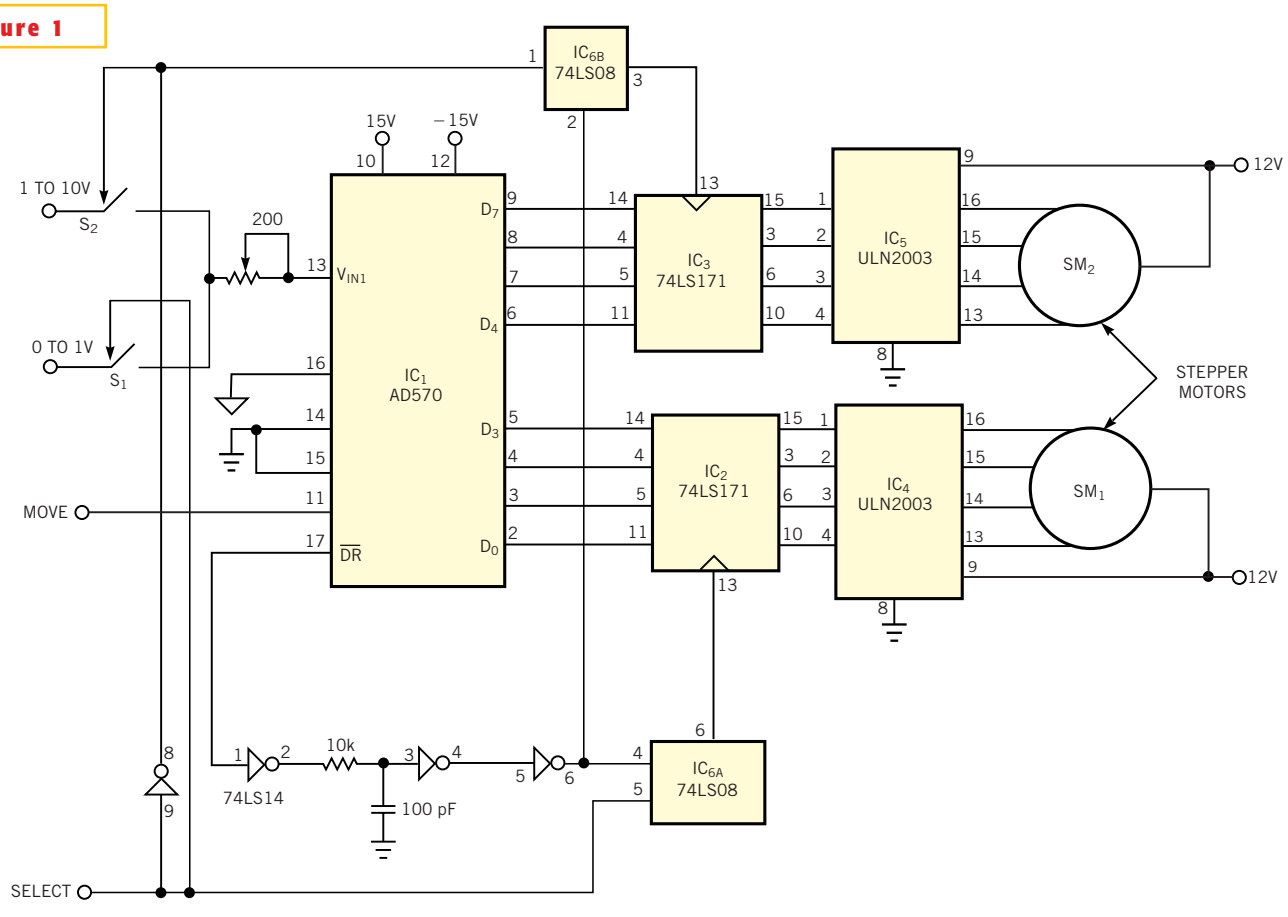
ADC controls multiple stepper motors

K Suresh, Indira Gandhi Centre for Atomic Research, Kalpakkam, India

Typical stepper-motor control circuits use either logic gates and flip-flops or shift registers to generate the proper sequences of binary codes that produce bidirectional stepper-motor movement. A conventional stepper-motor-control circuit uses a square-wave generator, a sequence generator, or a shift register and current translators to control one stepper motor apart from the logic circuit necessary for producing a known and valid binary code at start-up. When you need to control more than one stepper motor, as is the case of 2- and 3-D position control, the conventional type of control circuit becomes voluminous, complicated, and expensive due to the increased number of identical stages, the increased power dissipation, and the larger pc board.

Figure 1 shows a multiple-stepper-motor-control circuit that uses an ADC to control multiple stepper motors. The heart of this stepper-control circuit is an 8-bit, successive-approximation type ADC, (IC_1), whose 8-bit output forms two nibbles: LNIBBLE, D_0 to D_3, and UNIBBLE, D_4 to D_7. Each of these two nibbles carries valid 4-bit binary codes and drives the stepper-motor coils through a corresponding quad latch, IC_2 and IC_3, and buffer, IC_4 and IC_5. A set of four discrete analog voltages of 0 to 1V at V_{IN1} in the proper sequence control the stepper motor, SM_1, and another set of discrete analog voltages of 1 to 10V control stepper motor SM_2. A timing waveform at the MOVE input controls the start and end of each A/D conversion. The $\overline{DR}$ output of the ADC and AND gates in IC_6 generate enable signals to latch the ADC output nibbles to the

Figure 1



A simple and inexpensive ADC controls multiple stepper motors.

buffers. The SELECT input determines the selected stepper motor. A logic 1 at the SELECT input enables IC_{6A} and closes S_1, and the ADC's LNIBBLE latches in IC_2 to drive SM_1. A logic 0 at the SELECT input enables IC_{6B} and closes switch S_2, and the UNIBBLE latches in IC_3 to drive SM_2.

Tables 1 and **2** list the discrete analog voltages you must apply to the circuit in sequence and the corresponding valid ADC-generated codes. **Figure 2** shows the necessary timing waveforms for the stepper-control process. Initially, the MOVE input is at logic 1, which keeps the ADC in low-power mode and the ADC output in an open state so that both the motors are on hold. Next, you select SM_1 or SM_2 by applying a logic 1 or 0 at SELECT, and you apply any one of the four discrete analog voltages to the ADC. Then, pulling MOVE to logic 0 initiates a conversion. After approximately 42 µsec, the ADC generates a valid binary code, which the $\overline{DR}$ output of the ADC latches to the latch. The corresponding motor coils receive power according to the generated binary code through the buffer that the SELECT input enables. The MOVE input then returns back to logic 1. In this way, the circuit generates the successive valid codes by reading the corresponding analog voltages in a sequence, each time performing the A/D conversion. To rotate the selected motor in the opposite direction, you simply reverse the sequence of applied voltages.

The circuit in **Figure 1** is one way of using an ADC to control multiple stepper motors, and you can modify the circuit to suit individual requirements. Any low-cost, low-speed ADC is suitable for this circuit because the minimum time between application of successive codes, which the LR characteristics of the stepper-motor coils determine, is usually on the order of tens of microseconds. Because each motor uses only 4 bits of the ADC output, an 8-bit ADC can control two stepper motors. You can extend this concept to control three stepper motors using a 12-bit, low-speed ADC. You can apply the analog voltage to the ADC either through an analog multiplexer and a 2-bit up/down counter or through a DAC.

TABLE 1—SM_1 CONTROL VOLTAGES

Voltage to V_{IN1} (V_{REF}=10V)	ADC-generated LNIBBLE D_4	D_3	D_1	D_0
0.118V	0	0	1	1
0.352V	1	0	0	1
0.469V	1	1	0	0
0.235V	0	1	1	0

Notes:
- SELECT=logic 1
- IC_{6A} enabled
- S_1 closed
- IC_2 enabled

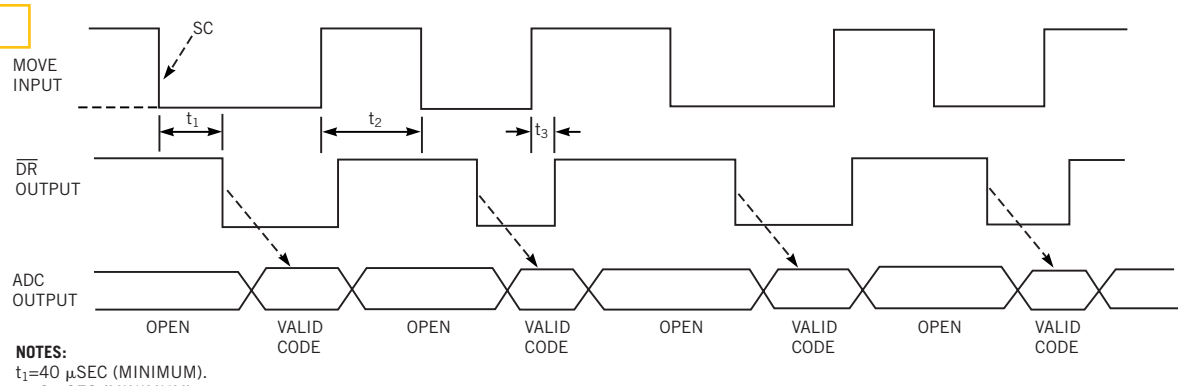
TABLE 2—SM_2 CONTROL VOLTAGES

Voltage to V_{IN1} (V_{REF}=10V)	ADC-generated UNIBBLE D_7	D_6	D_5	D_4
1.875V	0	0	1	1
5.625V	1	0	0	1
7.500V	1	1	0	0
3.750V	0	1	1	0

Notes:
- SELECT=logic 0
- IC_{6B} enabled
- S_2 closed
- IC_3 enabled

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

Figure 2



NOTES:
t_1=40 µSEC (MINIMUM).
t_2=2 µSEC (MINIMUM).
t_3=1.5 µSEC.

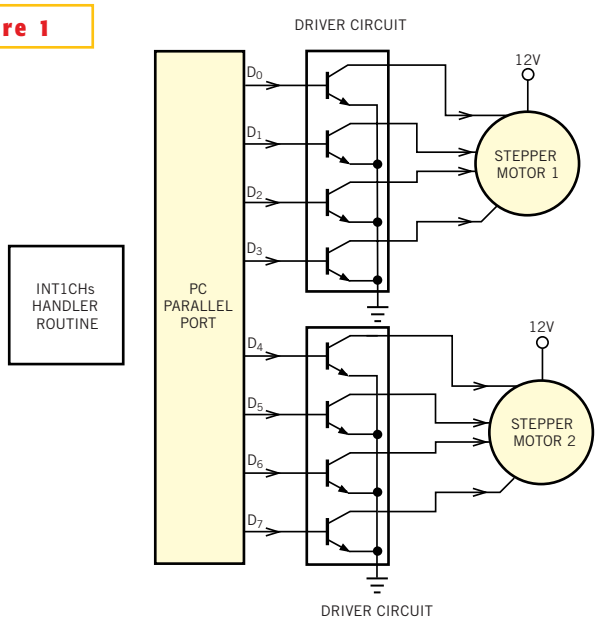
Timing waveforms show the MOVE input's control of the ADC output.

PC's BIOS interrupt drives twin stepper motors

J Jayapandian, Indira Gandhi Centre for Atomic Research, Kalpakkam, India

A simple and low-cost design achieves a twin stepper drive using a PC's special BIOS interrupt, INT1Ch, through the PC's parallel port (**Figure 1**). Turbo C control software programs the parallel port for the task of independently running two stepper motors. Users can write the appropriate control software for the required movement of two independent steppers. **Listing 1** is simple program to run the twin steppers in full step-clockwise mode. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2590. This program can run the steppers independently through the PC's LPT2 port using a handler routine for the interrupt INT1Ch.

Figure 1



A block diagram shows the control of two stepper motors through the PC's LPT2 port.

Figure 1 shows a block diagram of the required hardware, which the INT1Ch-handler routine can activate according to user requirements. The variable "TICKER" in the handler routine recognizes the occurrence of INT1Ch. For every occurrence of INT1Ch and thus every occurrence of TICKER, the handler routine writes the DATA necessary for the sequential step to move the stepper clockwise or counter-clockwise to the PC's LPT2 port. Every occurrence of TICKER causes the routine to go through the stepper cycle. The routine sends all the sequential DATA that the routine's SWITCH operator selects to the LPT2 port.

Full-step clockwise movement of the stepper requires four steps for the stepper coil using the following data: 1100, 0110, 0011, and 1001. The corresponding data in hex code are 0x0C, 0x06, 0x03, and 0x09, respectively, for a single stepper. For a twin stepper, the sequential-data pattern is 0xCC, 0x66, 0x33, and 0x99, respectively. The first 4 bits (least significant bits) in the 8-bit data control

LISTING 1—STEPPER-MOTOR-CONTROL PROGRAM

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

#define OUT_PORT  0X378   /* Out port address of LPT2 */
#define CTRL_PORT 0X37A   /* Control port address of LPT2 */
#define INTRTIMER 0x1C    /* BIOS Timer (INT 1CH)Interrupt */
/*-------------------------------------------------------------------------*/

/*-----------------------GLOBAL VARS---------------------------------------*/
static int TICKER;
static int  DATA;
void interrupt (*timerhandler)();
void interrupt STEPPERHANDLER();
/*---------------------------Handler routine for INT 1C---------------*/
void interrupt STEPPERHANDLER()
{
   disable();
   switch(TICKER % 16)  /* Reminder in  No. of TICKER divide by 16  sets four cases
                           for  setting  four DATA type.  This  method of TICKER
                           Processing sets the ON time of the  stepper  of 220 milli
                           seconds. For various ON time tuning  this division factor is
                           to be changed  and the corresponding reminder is to be
                           included in the case field. */
   {
     case  0: DATA = 0xCC; break; /* First step for clockwise full step i.e., 1 1 0 0 */
     case  4: DATA = 0x66; break; /* Second step for clockwise full step i.e., 0 1 1 0 */
     case  8: DATA = 0x33; break; /* Third  step for clockwise full step i.e., 0 0 1 1 */
     case 12: DATA = 0x99; break; /* Fourth  step for clockwise full step i.e., 1 0 0 1 */
    }
   outportb(OUT_PORT,DATA);
   ++TICKER;
   enable();
} /* END OF STEPPERHANDLER */

void INSTALLSTEPPERHANDLER()
{
        disable();
        timerhandler = getvect(INTRTIMER);
        setvect(INTRTIMER,STEPPERHANDLER);
        enable();
}

void CLEARSTEPPERHANDLER()
{
        disable();
        setvect(INTRTIMER,timerhandler);
        enable();
}

void main(void)
{
        clrscr();
        outportb(CTRL_PORT,0x01);
        INSTALLSTEPPERHANDLER();
        while (TICKER <=100)  /* This loop is only to test whether the  Sequential
                                 DATA  is properly sent to LPT2 port. This can be
                                 removed in the original program*/
        printf("TICKER No: %d  Value %X sent to port number  %x\n", TICKER,
                                                        DATA,OUT_PORT);
         CLEARSTEPPERHANDLER();
        return;
}
```

the first stepper, and the second nibble (most significant bits) control the second stepper. You can properly assign the least significant bits and most significant bits in the data field independently for the different mode of rotation of the steppers.

The handler routine shows identical movement for both steppers, and thus the data pattern for the least significant bits and most significant bits are the same. This sample program writes the variable "DATA" necessary for the step movement of the stepper once for four occurrences of INT1Ch. Because INT1Ch occurs once in 55 msec, the delay is 220 msec. Hence, the on-time of the stepper is fixed at 220 msec. To vary the on-time of the stepper, you can write the handler routine accordingly to change the necessary interrupt occurrences between writing new data for the stepper.

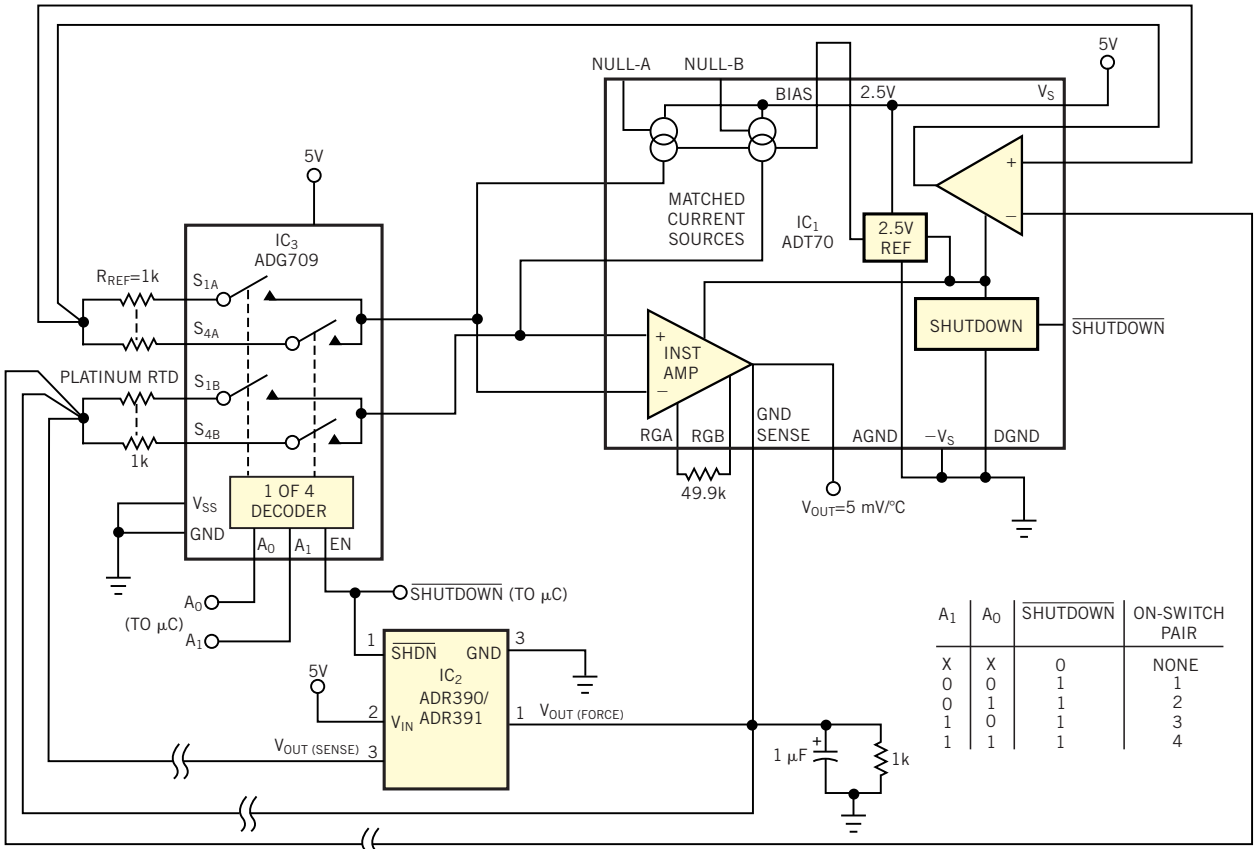
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Low-error platinum RTD circuit has shutdown capability

Reza Moghimi, Analog Devices, Santa Clara, CA

Resistor-temperature detectors (RTDs), are the most stable and popular temperature sensors. Platinum RTDs allow for a much wider range of temperatures than silicon-based sensors. In many cases, platinum RTDs sit far from the measurement circuitry, which adds a great deal of error into the measurement system. The circuit in **Figure 1** eliminates error by using a general-purpose amplifier and Kelvin-connected voltage references. The circuit also allows

Figure 1



A1	A0	SHUTDOWN	ON-SWITCH PAIR
X	X	0	NONE
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

A general-purpose amplifier in IC_1, Kelvin-connected voltage references, and platinum RTDs enable this circuit to accurately detect temperatures of −200 to +400°C.

for single-supply operation and can detect temperatures of −200 to +400°C with an output-voltage-scaling factor (sensitivity) of 5 mV/°C. To reduce errors due to self-heating, the circuit uses the largest RTD—value, in this case 1 kΩ—that results in an acceptable response time. The larger the RTD, the longer the response time.

IC_1 provides excitation and signal conditioning for the RTD, and internal current sources provide a matching excitation of 1 mA to the platinum RTD and reference resistor, R_{REF}. The instrumentation amplifier compares the voltage drop across the platinum RTD to the drop across R_{REF} and provides an amplified output signal that is proportional to temperature.

The lead resistance of wires connecting the RTD and R_{REF} can add inaccuracy to the temperature measurement. Voltage reference IC_2 creates a pseudo ground for IC_1 to overcome this inaccuracy. IC_2 has good temperature stability and low noise and can provide 5 mA of drive current. IC_2 also has a sense pin for sensing the drop on the line and compensating for the drop. Thus, the circuit provides stable and identical voltages at the bottom of the platinum RTD and R_{REF}. The circuit also buffers this voltage using the internal amplifier of IC_1. A 1-kΩ resistor in parallel with a 1-μF capacitor at the output of IC_2 provides a path for the current to flow to ground.

IC_3 makes it possible for the μC to address the platinum RTD. The circuit in **Figure 1** can accommodate four platinum RTDs, but you can increase this number by using other differential multiplexers. IC_3's low on-resistance match between channels of 0.4Ω does not introduce large errors into the system.

Another feature of this circuit is that it allows a programmer to put the circuit into shutdown mode. Initiating shutdown disables the enable pin of IC_3 so that none of the switch pairs are on. Also, IC_1 and IC_2 shut-down to conserve power.

The tracking of the current sources of IC_1 is 2 μA, and, as stated, the matching on-resistance of IC_3 is 0.4Ω . Thus, the worst-case mismatch resulting from the current sources and switches is 0.4Ω×2 μA=0.8 μV. If higher precision matching among current sources is necessary, you can connect a 50-kΩ potentiometer between the NULL-A and NULL-B pins and connect the center tap of the potentiometer to 5V.

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

Software provides three-priority-level interrupt for 8051

Deng Yong, Department of Instrumentation Engineering, Shanghai Jiaotong University, Shanghai, China

By using a pseudo-RETI instruction, the program in **Listing 1** provides an 8051 μC with a three-level-priority interrupt system. Among the three interrupt sources in the **listing**, External Request 0 (INT0) has the highest priority, and Internal Time/Counter 0 (IT0) has the lowest priority. The IT0 interrupt-service routine, before the pseu-

LISTING 1—THREE-PRIORITY-LEVEL INTERRUPT

```
        ORG   0000H
        LJMP  START
        ORG   0003H
        LJMP  INT0
        ORG   000BH
        LJMP  IT0
        ORG   0013H
        LJMP  INT1
START:  MOV   SP, #60H
        MOV   IP,#01H          ;INT0 has high priority
        MOV   TMOD,#01H
        MOV   TH0, #00H
        MOV   TL0, #00H
        SETB  EA               ;enable INT0,INT1,IT0
        SETB  EX0
        SETB  EX1
        SETB  ET0
        SETB  TR0
          ...
          ...
INT0:     ...    .....;INT0 interrupt service program
          ...
        RETI
INT1:     ...                  ;INT1 interrupt service program
          ...
        RETI
IT0:    CLR   TR0              ;IT0 interrupt service program
        PUSH  DPL
        PUSH  DPH
        MOV   DPTR, #GO_ON
        PUSH  DPL
        PUSH  DPH
        RETI                   ;"pseudo-RETI"
GO_ON:  NOP
          ...
          ...
        MOV   TH0, #00H
        MOV   TL0, #00H
        POP   DPH
        POP   DPL
        SETB  TR0
        RETI
```

do-RETI instruction, pushes the address of the first instruction behind the pseudo-RETI instruction onto the stack. The code clears the internal nonaddressable flip-flop of IT0 to acknowledge a higher interrupt after the pseudo-RETI instruction executes, and the IT0 interrupt-service routine continuously executes until the RETI instruction. You can download the listing from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2589.

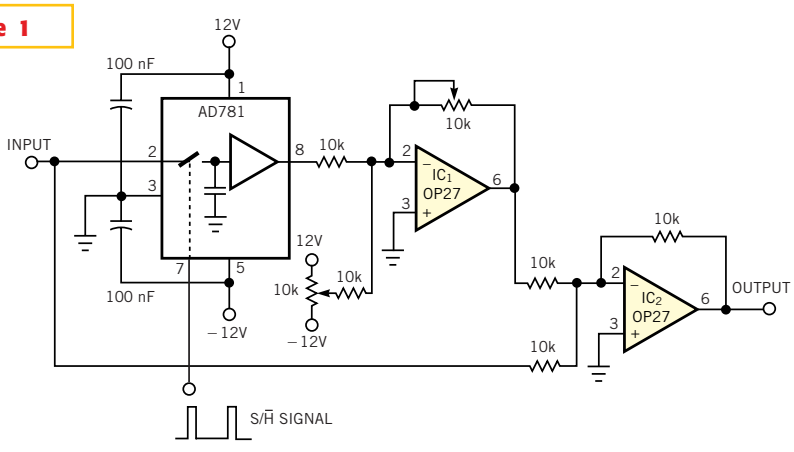
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Circuit samples derivative of a waveform

Joaquin Garcia and Jose Carrasco, Universidad Miguel Hernandez, Elche, Spain

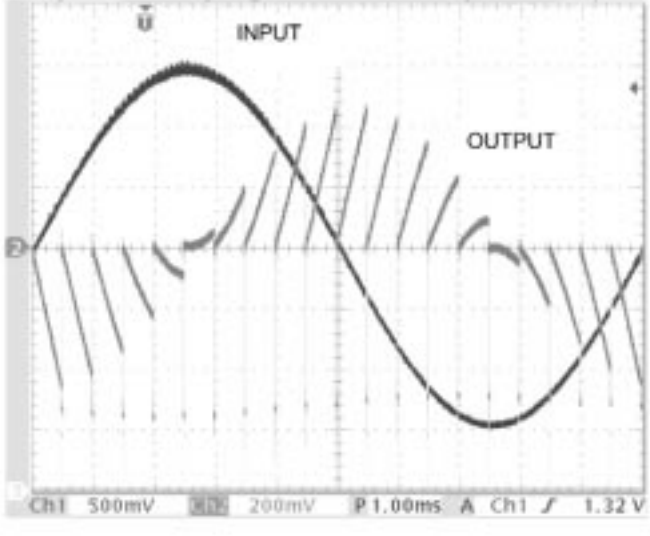
Basic derivative (differentiating) circuits use a lowpass filter or an active operational-amplifier implementation. These circuits need a large-value capacitor when differentiating a slowly varying signal. Moreover, the circuits are not useful for nonperiodic waveforms. The circuit in **Figure 1** offers a simple way of differentiating a waveform, even if it changes slowly or is nonperiodic. The circuit uses an AD781 sample/hold circuit and a subtracting circuit. By sampling the value of the input waveform at a given instant and then subtracting it from itself, op amp IC_2 produces a voltage proportional to the rate of change of the original waveform and to the duration of the hold time (the time the sample/hold input is held low). You can use a µC to transform the circuit's output to the derivative or use the output as is. IC_1 simplifies the subtraction operation and compensates the voltage shift of the sample/hold circuit during its sampling period. **Figure 2** shows a sinusoidal voltage and its derivative at the hold instants. Note that the sign of the derivative is inverted. To optimize the circuit, you should adjust the gain of the IC_2 subtraction circuit as a function of the input frequency.

Figure 1



This circuit differentiates slowly varying signals without the need for large capacitors.

Figure 2



At the hold instants of the circuit in Figure 1, the circuit produces the inverted derivative of the input waveform.

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

Edited by Bill Travis and Anne Watson Swager

Simple circuits provide nonlinearity

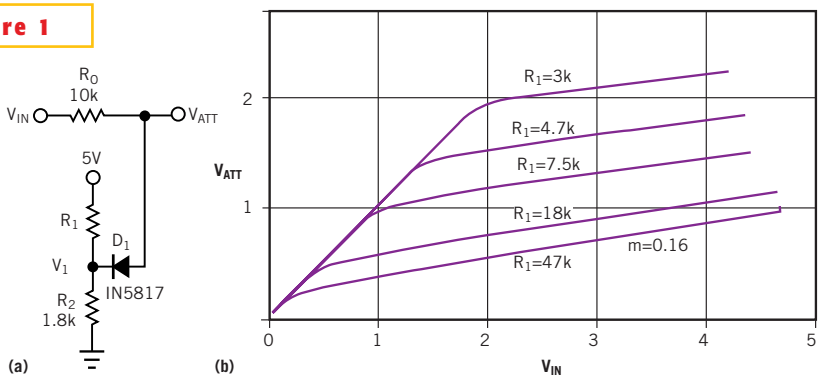
Abel Raynus, Armatron International, Melrose, MA

IN ANALOG DESIGN, you might need to create an amplifier with nonlinear dynamic characteristics—for example, logarithmic, exponential, or square-law. Usually, such amplifiers are complicated. However, the project often does not require mathematical precision. For example, you might just need to increase the dynamic range of an amplifier, or to eliminate saturation for an extended input-voltage range. The Design Idea is based on the nonlinear voltage attenuator with the attenuation ratio $m=V_{ATT}/V_{IN}$, controlled by the input voltage (**Figure 1a**). When V_{IN} is small enough to hold off D_1, $m=1$. When the input voltage increases and attains a certain threshold voltage, V_{TH}, the diode conducts, and the attenuation ratio decreases. The new value of m depends on the values of R_0 and R_2. R_1 and R_2 determine the threshold level, V_{TH}. Hence, you can estimate the resistors R_1 and R_2 for a given R_0, V_{TH}, and m as:

where V_D is the drop across diode D_1, and V_R is the dc voltage applied to R_1 and R_2.

$$R_2 = R_0 \frac{m}{1-m};$$

Figure 1

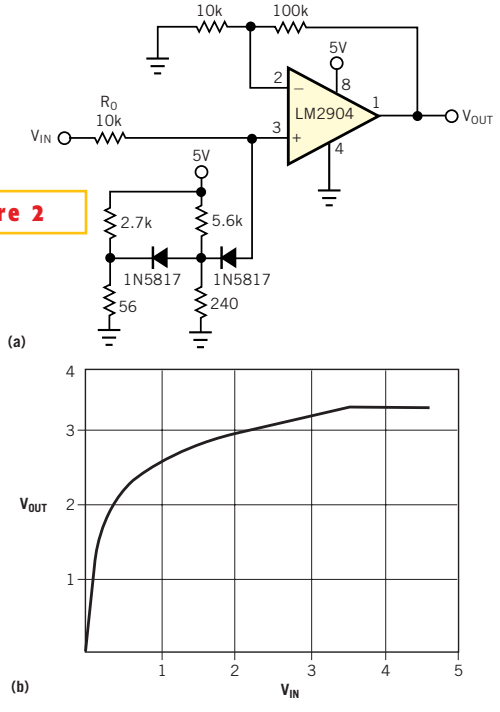


Manipulating the resistor values in an attenuator (a) changes the threshold for nonlinearity (b).

$$R_1 = R_0\left(\frac{V_R}{V_{TH}-V_D}-1\right),$$

Note that you can create any characteristic by choosing the proper ratio, m, and the threshold voltage for each fragment of the resulting characteristic. Also, the linear approximation is good for calculation purposes, but the real ratio changes smoothly near the threshold voltage. Finally, use Schottky diodes to increase the voltage range of regulation. **Figure 1b** shows the dynamic response of the attenuator for the constant ratio $m=0.16$ but for different threshold voltages. The measured voltage is $V_1 = V_{TH}-V_D$. To increase the dynamic range of an amplifier, you should put the nonlinear attenuator at the input of the amplifier. To

Figure 2

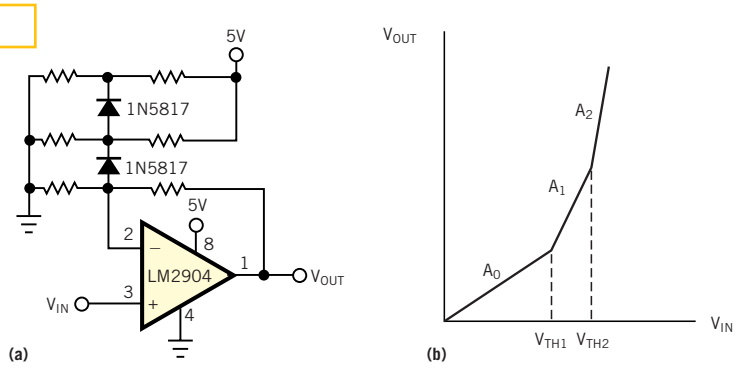


A multistage nonlinear attenuator (a) can increase the dynamic range of an amplifier (b).

widen the range, you can use a two-stage attenuator. **Figure 2** shows such an amplifier and its recorded characteristic. The applications of the nonlinear attenuator are not limited to increasing dynamic range. You can obtain a square-law response, for example, by putting the attenuator in a feedback circuit (**Figure 3**).

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

Figure 3



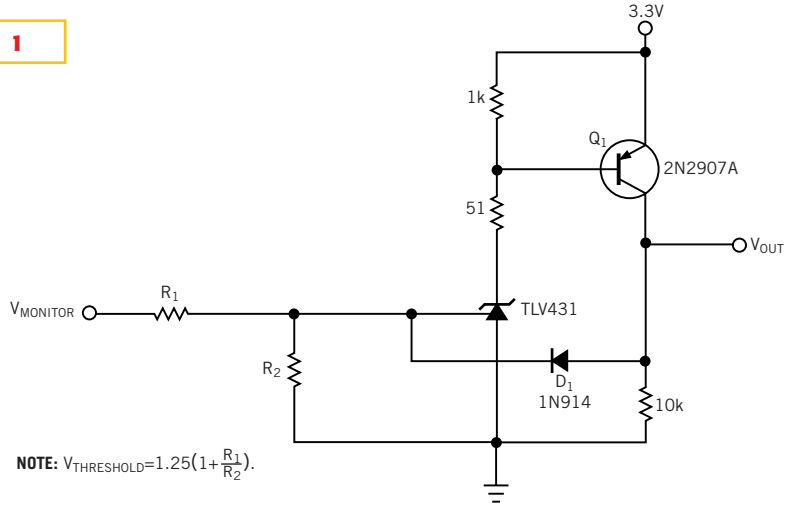
A nonlinear attenuator in the feedback loop (a) results in a square-law characteristic (b).

Regulator IC forms convenient overvoltage detector

Robert Bell, On Semiconductor, Phoenix, AZ

Figure 1 shows a simple, standalone overvoltage detector. The intent of the circuit is to monitor a voltage, $V_{MONITOR}$, and set the output, V_{OUT}, high when the monitored voltage exceeds a preset threshold. The minimum allowable threshold for this circuit is 1.25V. The operation of the circuit revolves around the TLV431 shunt regulator. This IC is based on the popular TL431 shunt regulator. The difference is that the TLV431's internal reference is 1.25V, as opposed to 2.5V for the TL431. When the voltage at the control input is less than 1.25V, the regulator's cathode current is essentially zero. If the control input exceeds 1.25V, the cathode conducts and turns Q_1 on to produce a high output at V_{OUT}. The trip threshold, determined by resistors R_1 and R_2, is $V_{THRESHOLD}=1.25(1+R_1/R_2)$. D_1, the diode between V_{OUT} and the control input, provides hysteresis and latches the overvoltage fault condition. If you don't need latching operation, you can add a resistor in series with the diode to lower the hysteresis value and prevent the circuit from latching.

Figure 1



A shunt regulator makes an inexpensive overvoltage detector.

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

Circuit provides effective LCD drive

Luo Ben Cheng, Chinese Academy of Science, Beijing, China

LCDs find wide use in portable instruments, thanks to their attractive display and low power consumption. The circuit in **Figure 1** is an effective driver for LCDs. The circuit comprises two main sections—the ICM7211 drivers (IC_2 and IC_3) and the YN06 display itself (IC_1). The Intersil (www.intersil.com) ICM7211 is a 4-bit LCD driver that needs no external components. It contains three basic sections: a reference signal-generator circuit, an input and display-channel section, and a digit-selection and drive circuit. It contains a complete pulse-generator unit and an oscillator-divider clock-drive circuit. When you disconnect the BP pin (Pin 5), the IC produces a 125-Hz pulse signal. YN06 is a six-bit character LCD, which uses 5 decimal bits and 2 column bits. To control the display, you need a 4-bit BCD driver.

In **Figure 1**, an AT89C51 µC controls the two ICM7211 drivers. The drivers in turn drive the 6-bit YN06. Pin 5 of IC_2 and IC_3 connect to the COM pin (Pin 1) of IC_1. The reference signal-generator circuit works in open-loop mode. This mode results when you disconnect the OSC pin (Pin 36) of IC_2 and connect the OSC pin (Pin 36) of IC_3 to ground. The result is a 125-Hz pulse train, which serves as the LCD's drive clock. The chip-enable signals $\overline{CS1}$ of IC_2 and IC_3 to connect to the µC's pins P2.5 and P2.6, and $\overline{CS2}$ connects to Pin P3.6, which serves as a read or write port. In addition, data-input ports B0 to B3 and digital-selection input ports DS1 and DS2 connect to the data bus through the D0 to D5 lines. To control the LCD, you need only provide 4-bit BCD codes through the µC. Unfortunately, in some cases, the display needs decimal bits. The normal method of providing these bits is to add another LCD decimal driver, such as a CD4056.

Note that the LCD in **Figure 1** needs only 6 bits, whereas the drivers can provide 8 bits. That fact means that two more seven-segment output ports go unused. You can take advantage of the unused ports of IC_2 and IC_3 to solve the decimal-bit problem. Connect DP1 (Pin 5 of IC_1) to Pin 25 of IC_3, DP2 (Pin 9 of IC_1) to Pin 23 of IC_3, DP3 (Pin 13 of IC_1) to Pin 21 of IC_3, and DP4 (Pin 17 of IC_1) to Pin 25 of IC_2. Also, connect COL1 (Pin 33 of IC_1) to Pin 23 of IC_2, and COL2 (Pin 42 of IC_1) to Pin 24 of IC_2. With the help of some µC software, you can control the LCD in a flexible fashion. **Listing 1** shows the AT89C51 assembly code for controlling the LCD. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2574.

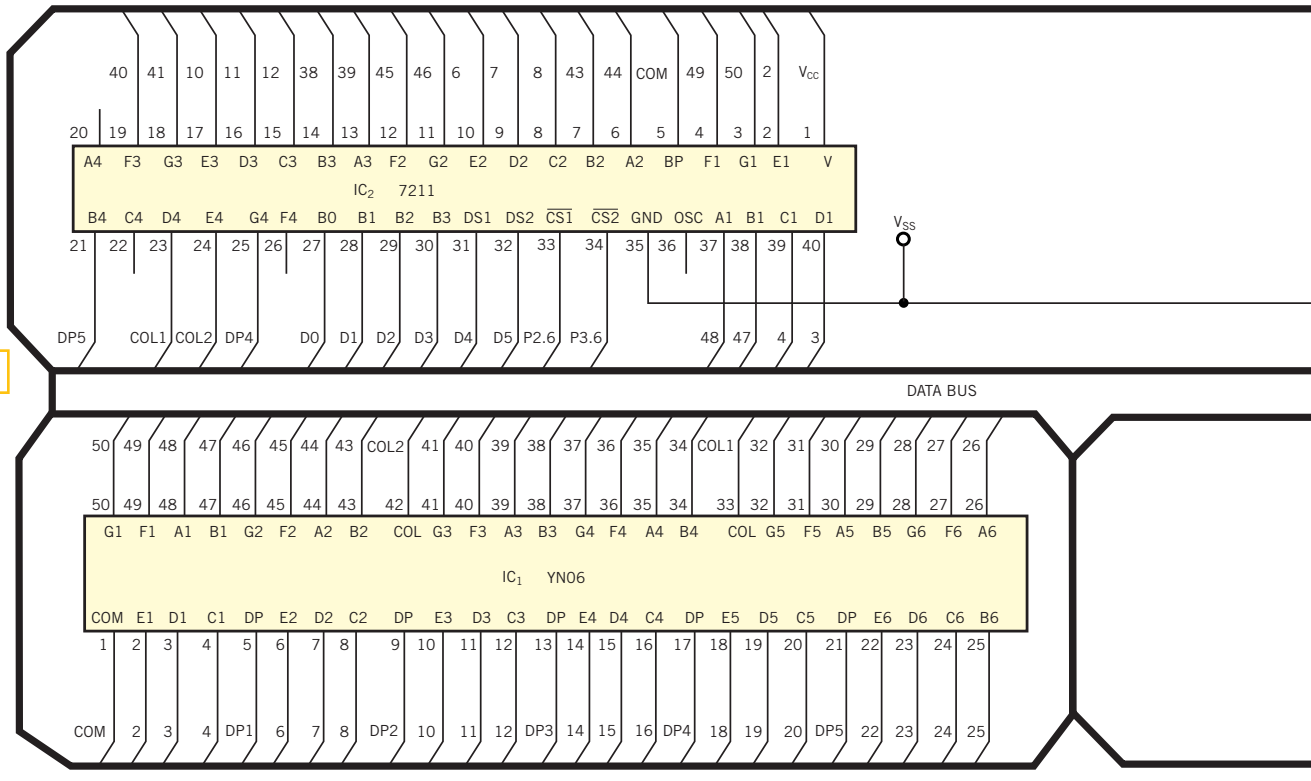
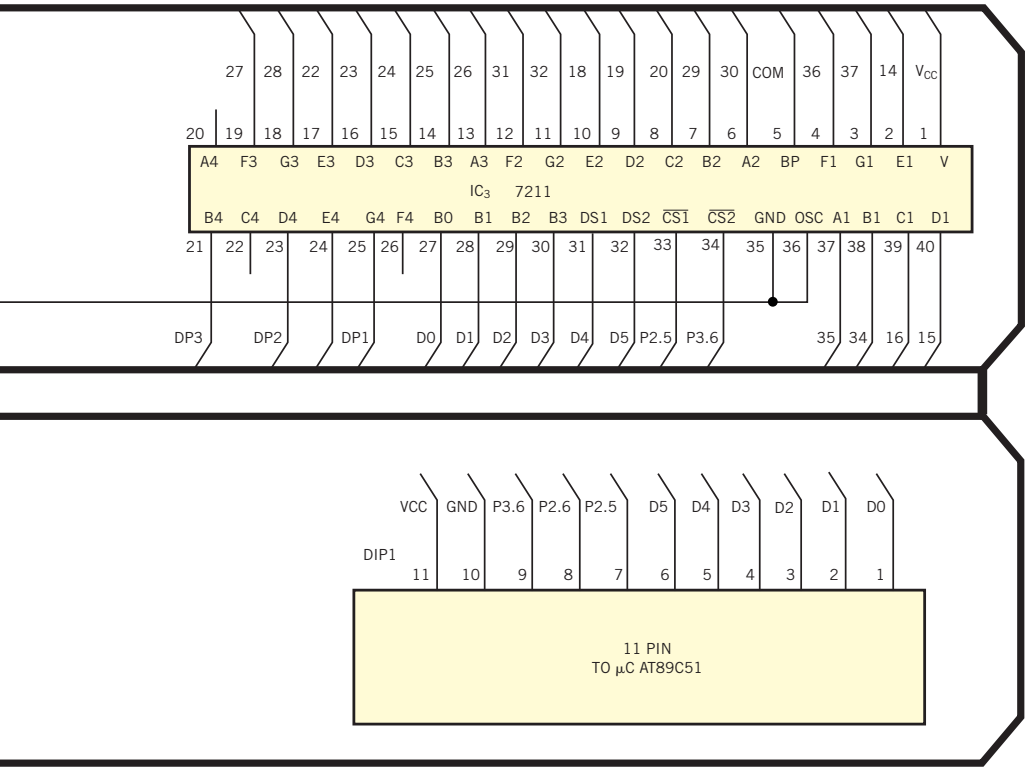


Figure 1

The unused driver outputs take care of the LCD's need for decimal-bit inputs.

LISTING 1—ASSEMBLY CODE FOR AT89C51 LCD DRIVE

```
ORG  0000H
        LJMP  MAIN
        ...
ORG  0030H
     FIG3       EQU  23H
     ADR00      EQU  40H
     SDATA      EQU  50H
     BITD       EQU  53H
  ADRU2    EQU  3000H
  ADRU3    EQU  5000H
MAIN:           CLR EA                  ;CLOSE ALL THE INTERRUPTS
                ...
                LCALL DISPLAY           ;DISPLY DATUS
                ...
HERE:           NOP
                AJMP HERE               ;WAIT FOR INTERRUPTS
; ***************************************************************
ORG  0120H
DISPLAY:        MOV R1, #SDATA          ;INPUT THE DATUS FOR DISPLAY
                MOV A, BITD
     DIR:       MOV R6, #4              ;FOR ONE 4_BIT LCD DRIVER
                MOV R0, #ADR00          ;DATUS BUFFERS
                MOV FLG3, A             ;STORE THE DECIMAL BIT
                LCALL DOT               ; PROCESS THE DECIMAL BIT
                LCALL DXCH              ;FORMAT THE DATA FOR DISPLAY
     SDIR:      MOV DPTR,#ADRU2         ;CHOICE THE LCD DRIVER U2
                MOV A,@R0
                MOVX @DPTR,A
                INC R0
                DJNZ R6,SDIR
                MOV R6,#4
SDIRO:          MOV DPTR,#ADRU3         ;CHOICE THE LCD DRIVER U3
                MOV A,@R0
                MOVX @DPTR,A
                INC R0
                DJNZ R6,SDIRO
                MOV R6,#4               ;RECOVER THE INFORMATION
                MOV R0,#ADROO
                RET
; ***************************************************************
DOT:            MOV R0,#ADROO           ;DECIMAL BIT PROCESSING
                JB FLG3.0,D0
                JB FLG3.1,D1
                JB FLG3.2,D2
                JB FLG3.3,D3
                JB FLG3.4,D4
                JB FLG3.5,D5
                JB FLG3.6,COL
D0:        MOV @R0,#0FH
                MOV R0,#ADROO+7
                MOV @R0,#0FH
                AJMP DCOL
D1:        MOV @R0,#0FH
                MOV R0,#ADROO+7
                MOV @R0,#01H
                AJMP DCOL
D2:        MOV @R0,#0FH
                MOV R0,#ADROO+7
                MOV @R0,#0AH
                AJMP DCOL
D3:        MOV @R0,#01H
                MOV R0,#ADROO+7
                MOV @R0,#0FH
                AJMP DCOL
D4:        MOV @R0,#0DH
                MOV R0,#ADROO+7
                MOV @R0,#0FH
                AJMP DCOL
D5:        MOV @R0,#0AH
                MOV R0,#ADROO+7
                MOV @R0,#0FH

                AJMP DCOL
COL:            MOV @R0,#0FH
                MOV R0,#ADROO+7
                MOV @R0,#0DH
DCOL:           MOV R0,#ADROO+1         ;JUMP OVER THE DECIMAL BIT
                RET
; ***************************************************************
DXCH:                   MOV A,@R1               ;DATA PROCESSING MODULE
                        MOV B,A
                        ANL A,#0FH
                        ORL A,#10H
                        MOV @R0,A       ;CHOICE THE SIXTH BYTLE FOR DISPLAY
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#20H      ;CHOICE THE FIFTH BYTLE
                        MOV @R0,A
                        INC R0
                        DEC R1
                        MOV A,@R1
                        MOV B,A
                        ANL A,#0FH
                        ORL A,#30H
                        MOV @R0,A       ;CHOICE THE FOURTH BYTLE
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#10H
                        MOV @R0,A       ;CHOICE THE THIRD BYTLE
                        INC R0
                        DEC R1
                        MOV A,@R1
                        MOV B, A
                        ANL A,#0FH
                        ORL A,#20H
                        MOV @R0,A       ;CHOICE THE SECOND BYTLE
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#30H
                        MOV @R0,A       ;CHOICE THE FIRST BYTLE
                        MOV R0,#ADROO
                        INC R1
                        INC R1
                        RET
```



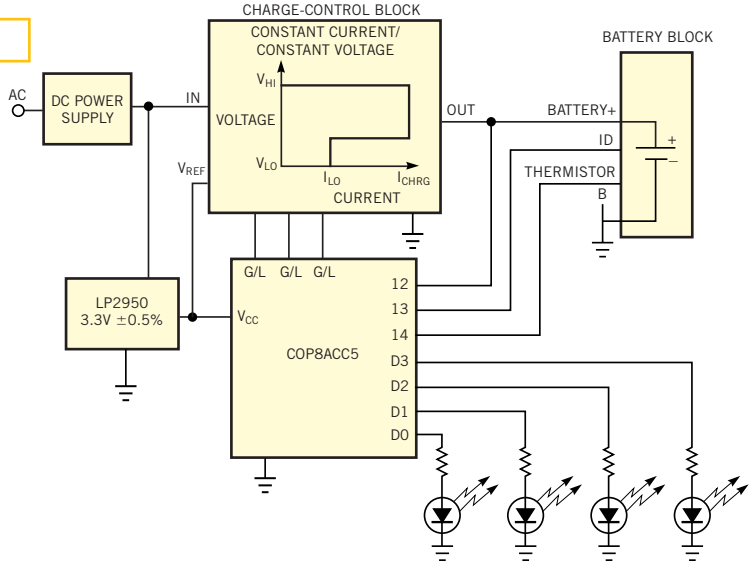
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

µC controls multichemistry battery charger

Kelly Flaherty, National Semiconductor, Santa Clara, CA

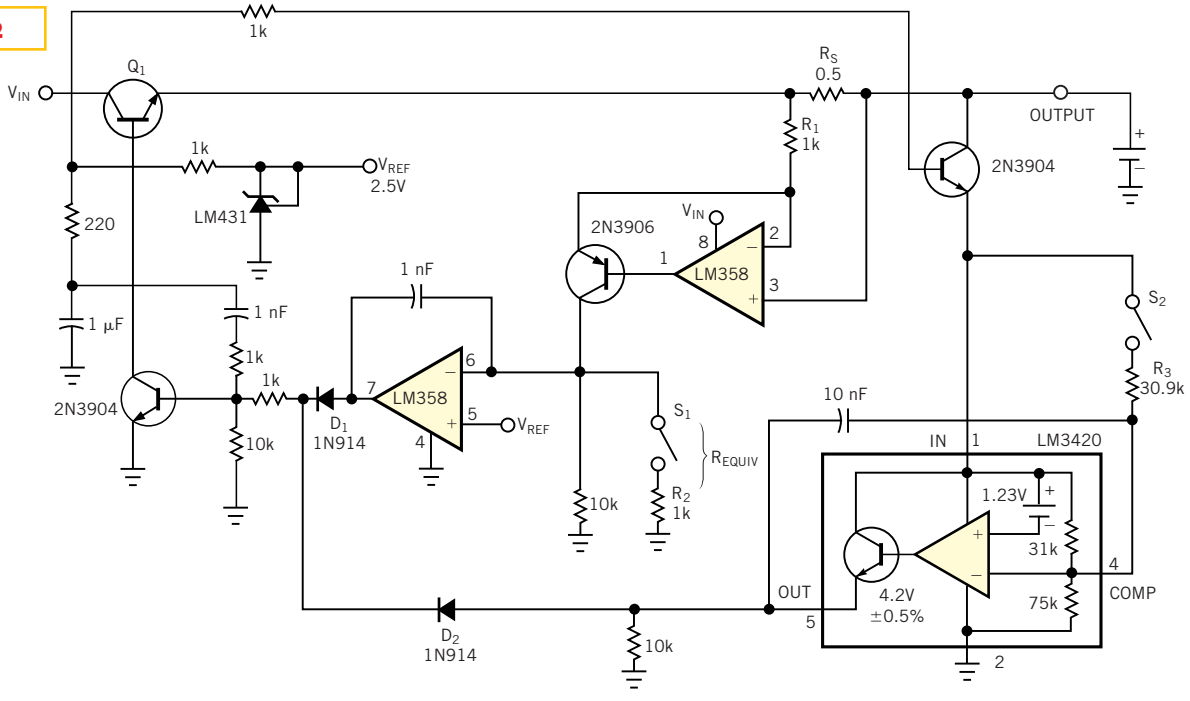
Figure 1 is a generalized block diagram of a multichemistry battery charger. A COP-8ACC5 µC handles its key charging features. The µC is available in a 20-pin (15 I/O pins) SOIC or a 28-pin (23 I/O pins) SOIC/DIP. It contains 4 kbits of internal ROM. The controller's A/D inputs monitor the battery-voltage pin, ID pin, and thermistor pin. For more complex charging systems, you can add external EEPROM via the Microwire serial interface. Such an external EEPROM might be useful for storing battery-specific charge history, a battery-specific look-up table for more accurate charging, or both. The LP2950 doubles as a low-dropout-voltage regulator for the µC and as a ±0.5% reference for the charge-control block. The charge-control block is basically a constant-voltage, constant-current regulator, as the voltage-versus-current curve in **Figure 1** shows. The µC reads the battery pack's ID pin and adjusts the circuit

Figure 1



This stand-alone battery charger handles multiple battery chemistries.

Figure 2



A battery charge-control block operates in both constant-current and constant-voltage mode.

accordingly. If the battery is a lithium-ion type, the charge-control block adjusts the fast-charge rate according to the battery's capacity rating and adjusts the constant voltage to the critical maximum voltage for lithium-ion. If the battery is nickel-based, similar adjustments take place. However, the voltage adjustment is to a level greater than the maximum battery voltage that is still low enough to accommodate the power dissipation of the pass transistor.

Figure 2 shows a possible implementation of the charge-control block. It is an adjustable constant-current, constant-voltage regulator under control of a µC. The switches can be analog switches, such as the CD4066; discrete transistors, such as the 2N3904; or FETs, such as the 2N7002. The default setting (switches open) is 4.2V and 0.5A. When S_1 closes, current regulation increases by the change in equivalent resistance (R_{EQUIV}): $I_{CHRG}=(V_{REF}\times R_1)/(R_{EQUIV}\times R_{SENSE})$. S_1 switches in R_2, resulting in the doubling (to 1A) of the default regulated current. Closing S_2 similarly increases the level of the regulated voltage from the LM3420 lithium-ion charge controller. The LM3420 contains an error amplifier, a precision voltage reference, and a trimmed voltage divider that sets the regulated voltage to within ±0.5%. The IC is available in five fixed voltage levels that correspond to 4.2V per cell for one, two, three, and four cells. The Comp pin of the LM3420 switches an external resistor, R_3, in parallel with one of the internal divider resistors, and results in a regulated voltage of 7.2V. Q_1 provides a disconnect between the battery and the LM3420 upon removal of the input voltage. D_1 and D_2 act as an exclusive-OR gate for current regulation of voltage regulation. When V_{REG} is reached, D_2 overrides D_1.

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

Ultracapacitor powers bicycle light

Dennis Eichenberg, Parma Heights, OH

THE CIRCUIT IN **Figure 1** represents a simple and inexpensive way to provide additional bicycling safety. A conventional bicycle-generator lighting system works with the circuit to provide safety lighting for several minutes after the bicycle has stopped. Energy storage uses an ultracapacitor rather than a battery to avoid the typical battery problems of limited life, critical charging rates, and intolerance to cold. The advent of inexpensive, compact ultracapacitors has made this approach practical. A standard headlight connects to the generator in a normal fashion, so that the headlight is on whenever the bicycle is moving. An ultracapacitor receives its charge from the generator and connects to an astable multivibrator to pulse the taillight.

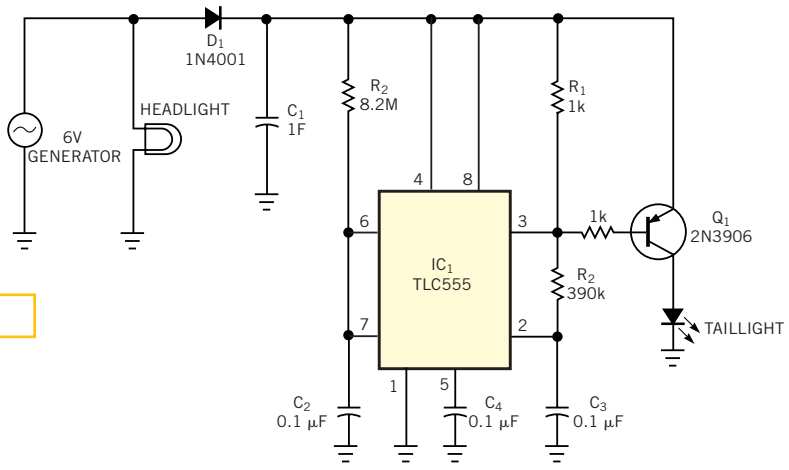


Figure 1

An ultracapacitor provides additional safety by flashing the taillight after the bike has stopped.

A typical bicycle generator is a 6V-ac device. The load regulation of the generator is poor, so connecting the headlight directly to the generator stabilizes the generator's output. Diode D_1 provides half-wave rectification of the generator's output to charge capacitor C_1 and to power the taillight circuit. D_1 also acts as a blocking diode to prevent C_1 from discharging back through the headlight and the generator. The CMOS 555 timer, IC_1, acts as an astable multivibrator tolerant of voltage variations as low as 2V. The circuit provides an off-time of 820 msec ($1.1\times R_1\times C_2$) and an on-time of 43 msec ($1.1\times R_2\times C_3$). IC_1 drives transistor Q_1 to pulse the grounded taillight. You must use an LED for the taillight, because it does not draw the extreme surge current of an incandescent lamp and thus provides several minutes of illumination. A 2V LED limited to 100 mA provides the greatest duration of light.

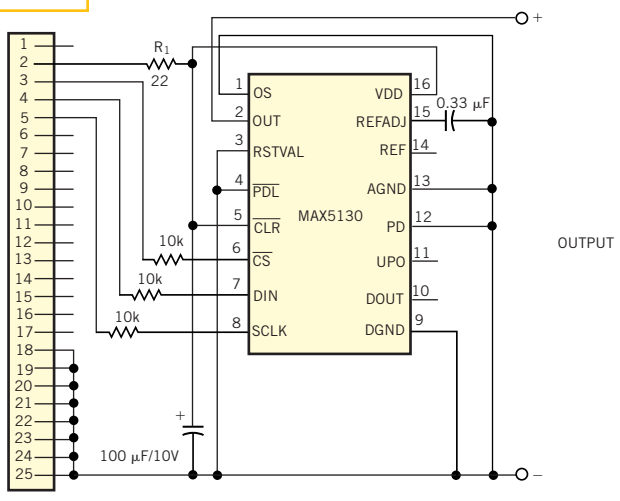
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Printer port controls reference generator

Yongping Xia, Teldata Inc, Los Angeles, CA

FIGURE 1 SHOWS a programmable, precision reference-voltage generator. A PC's printer port controls the generator. The voltage range is 0 to 4.0955V in 0.5-mV increments. Because the computer's hard drive saves the reference setting, when you restart the computer, the output voltage is exactly the same as the previous setting. The Max5130 is a 13-bit serial voltage-output DAC with an internal reference. It uses a three-wire serial interface. Because the IC has everything necessary for a programmable reference, the circuitry is simple. The printer port's Pin 2 powers the circuit. Pins 3, 4, and 5 provide chip select (CS), data (DIN), and clock (SCLK), respectively, to the Max5130. **Listing 1** is a C program for the generator. "U" and "D" keys speed the voltage setting, given that the DAC has 8192 steps. Push "U" or "D" for 100-step changes, equivalent to 650-mV steps. The "u" and "d" keys fine-tune the output with 0.5 mV per step.

Figure 1



Use a PC's printer port and a 13-bit DAC to configure a precision reference generator.

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

LISTING 1–C PROGRAM FOR PRECISION REFERENCE GENERATOR

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <math.h>

#define POWER_ON      0x01
#define CLK_HIGH      0x08
#define CLK_LOW       0xf7
#define CS_HIGH       0x02
#define CS_LOW        0xfd
#define DATA_HIGH     0x04
#define DATA_LOW      0xfb

typedef unsigned int WORD;
FILE  *data_file;

int i, out_port, in, out=0;
long data, step;
float ref_voltage;

void find_port(void)          /* find printer port address  */
{
  out_port=*(WORD far *)MK_FP(0x0040,8);
  out|=POWER_ON;
  outportb(out_port, out);              /* power on            */
  delay(100);
}

void show_ref(void)                     /* display voltage      */
{
  gotoxy(2,2);
  ref_voltage=(float)step/2000;
  printf("Voltage = %.4fV   ", ref_voltage);
}

void write_ref(void)          /* write data to c driver     */
{
  if ((data_file = fopen("c:\\ref_data", "wb")) == NULL)
    printf("File open failed.\n");
  fseek(data_file, 0, 0);
  fwrite(&step, 1, 2, data_file);
  fclose(data_file);
}

void read_ref(void)                   /* read data from c driver    */
{
  if ((data_file = fopen("c:\\ref_data", "rb")) == NULL)
    write_ref();
  fseek(data_file, 0, 0);
  fread(&step, 1, 2, data_file);
  fclose(data_file);
}

void send_clock(void)                 /* send sclk to MAX5130 */
{

  out|=CLK_HIGH;
  outportb(out_port, out);              /* clock high  */
  delay(2);
  out&=CLK_LOW;
  outportb(out_port, out);              /* clock low   */
  delay(2);
}

void set_ref()                         /* send data to MAX5130   */
{
  out|=CS_HIGH;
  outportb(out_port, out);               /* cs high  */
  delay(2);
  data=step+0x4000;
  out&=CS_LOW;
  outportb(out_port, out);               /* cs low   */
  delay(2);
  for (i=0; i<16; i++)
   {
     if (data>=(pow(2, (15-i))))
      {
      data-=(pow(2, 15-i));
      out|=DATA_HIGH;
      }
     else
      out&=DATA_LOW;
     outportb(out_port, out);            /* send 1 bit data      */
     delay(2);
     send_clock();
   }
  out|=CS_HIGH;
  outportb(out_port, out);               /* cs high        */
  delay(2);
  show_ref();
  write_ref();
}
```

(*continued on pg 134*)

LISTING 1—C PROGRAM FOR PRECISION REFERENCE GENERATOR (CONTINUED)

```
void main(void)
{
 clrscr();
 find_port();
 read_ref();
 show_ref();
 set_ref();
 do{
   do{
      } while(!kbhit());
   in=getch();
   if (in=='u')          /* if hit 'u', increase 0.5mV per step    */
    {
      if (step<8191)
      step++;
    }
   if (in=='U')          /* if hit 'U', increase 50mV per step     */
    {
      if (step<8191)
      step+=100;
      if (step>8191)
      step=8191;
    }
   if (in=='d')          /* if hit 'd', decrease 0.5mV per step    */
    {
      if (step>0)
      step--;
    }
   if (in=='D')          /* if hit 'U', decrease 50mV per step     */
    {
      if (step>0)
      step-=100;
      if (step<0)
      step=0;
    }
   set_ref();
 }while(in!='q' && in!='Q');  /* if hit 'q' or 'Q', quit    */
}
```

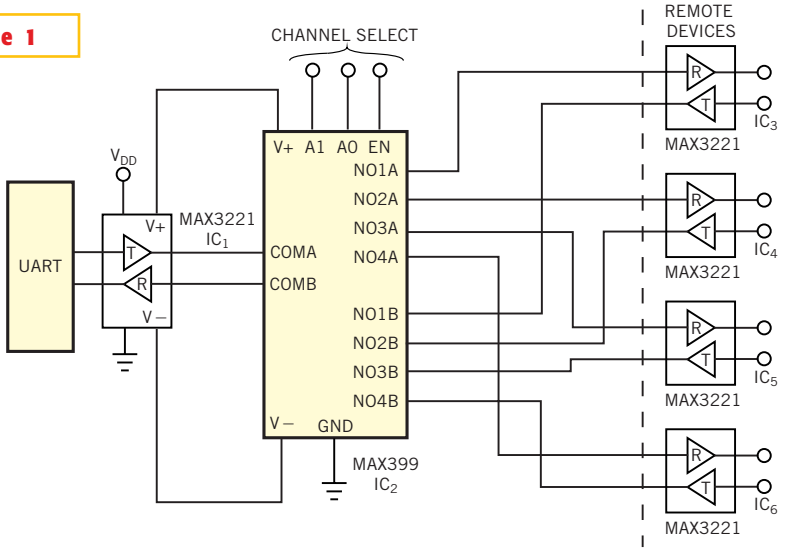
Multiplexer enables pseudomultidrop RS-232 transmission

Dan Christman, Maxim Integrated Products, Sunnyvale, CA

RS-232 communications with one µC and more than one remote system can be problematic, because most µCs contain only one UART, which provides an interface between synchronous and asynchronous ports. The multiplexer in **Figure 1,** IC_2, allows multiple channels (four, in this case) to share a single UART. The dual four-to-one multiplexer allows transceiver IC_1 to form a network with the four remote transceivers IC_3 to IC_6. **Table 1** defines the channel-selection codes. Selecting Channel 1, for instance, enables IC_1 to communicate with IC_3 without being loaded by IC_4 to IC_6. Pulldown resistors inside the remote transceivers force the outputs of unselected receivers to a known state.

The circuit's supply-voltage range (3 to 5.5V) makes it compatible with 3 and 5V logic. IC_2 receives its power directly from the V+ and V− terminals of IC_1, whose ±5.5V outputs come from an internal charge pump. The multiplexer handles rail-to-rail signals, so obtaining its power from IC_1 ensures that RS-232 signals pass directly through, regardless of amplitude. Each transceiver's charge pump requires four small capacitors (not shown), whose values depend on the V_{DD} range but do not exceed 0.47 µF. Note that pulling too much current from the charge-pump terminals of IC_1, V+ and V−, can cause these rails to droop and may pull the IC's RS-232 transmission levels out of specification.

Figure 1



One UART and one multiplexer enable one RS-232 transceiver to communicate with four others.

TABLE 1—CHANNEL SELECTION

A1	A0	EN	Selected channel
X	X	0	All channels disconnected
0	0	1	Channel 1 (IC3)
0	1	1	Channel 2 (IC4)
1	0	1	Channel 3 (IC5)
1	1	1	Channel 4 (IC6)

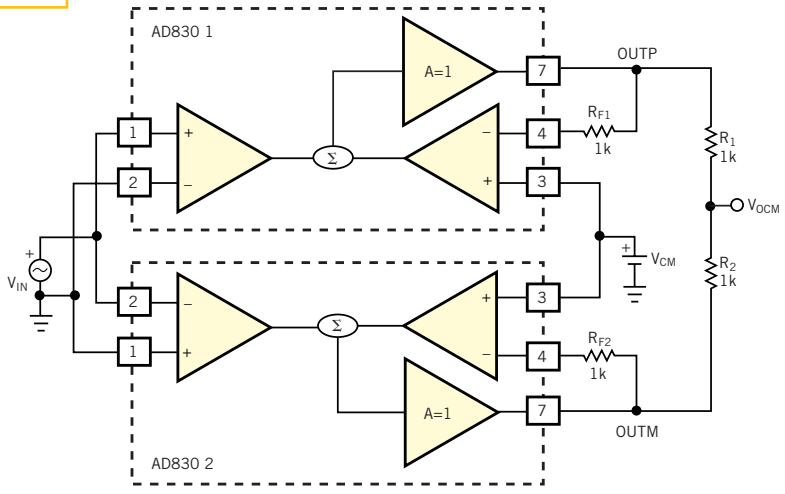
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Differential amp drives high-speed ADC

Moshe Gerstenhaber and Chau Tran, Analog Devices, Wilmington, MA

THE SCHEMATIC IN **Figure 1** depicts a differential-input/differential-output or single-ended-input/differential-output amplifier with a gain of two. You can use the low-distortion circuit to drive high-speed ADCs. You can also use it to drive precision delta-sigma ADCs. The circuit contains two active-feedback amplifiers, with input connections such that one amplifier acts as a voltage follower and the other acts as an inverter. You take the output differentially from the amplifiers' outputs. The ADC's reference output can connect to VCM to set the output common-mode voltage of the amplifier stages, or you can set this voltage by external means. Resistors R_{F1} and R_{F2} reduce the distortion of the system. We added R_1 and R_2 for displaying the effects of the amplifiers' mismatch. **Figure 2** is a performance photo at 10 MHz and a gain of two. The top trace is the single-ended input signal; the two bottom traces are the output signals, out of phase with each other. **Figure 3** demonstrates the gain error and the low distortion of the system. The bottom trace, at 10 mV/div, shows the effects of the amplifiers' mismatch at the common-mode node.

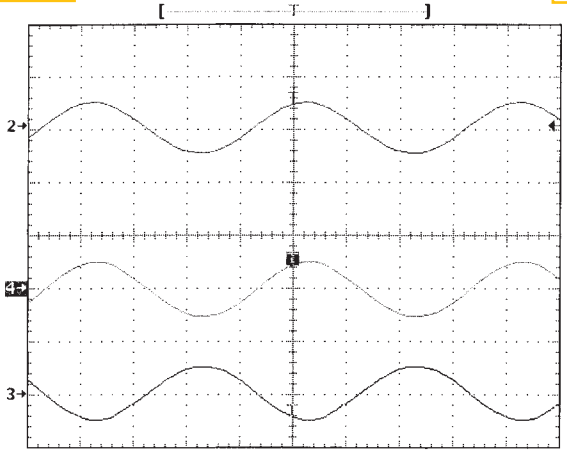
Figure 1



A single-ended-input/differential-output configuration drives high-speed ADCs with minimal common-mode error.

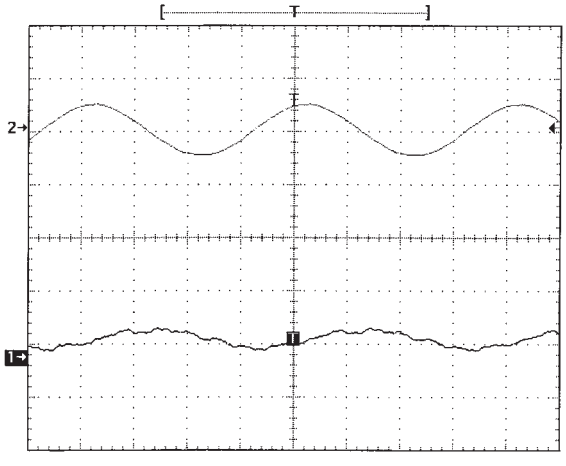
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 2



In the bottom traces, the differential outputs are 180° out of phase with each other.

Figure 3



The bottom trace shows the amplifiers' mismatch at 10 mV/div.

design ideas

Edited by Bill Travis and Anne Watson Swager

Keypad sequence activates relay

JB Guiot, DCS AG, Allschwil, Switzerland

Today, it seems that µPs assume even the easiest tasks, such as activating a relay (for an alarm or access control, for example) via a keyboard command. However, a simpler and less expensive solution is available (**Figure 1**). Moreover, the method does not entail programming and the concomitant, almost inevitable, debugging. A CD4016 CMOS analog switch, IC_1, forms the heart of the system. Depressing the key labeled "1st" closes the topmost switch (between pins 1 and 2), which latches itself via the diode connected between pins 2 and 13. The same topology repeats three times, with the keys labeled "2nd," "3rd," and "4th." Each switch provides power to the next one, so that depressing the key labeled "3rd" before the "1st" or "2nd" key does not activate the third switch. The Disable keys, connected in parallel, disable the entire chain by shorting the control input of the first switch to ground. The values of the power-supply voltage and the components are not critical; you can use almost anything you find in your drawer. Take care, however, that the output transistor has a rating that can handle the relay. The entire circuit fits onto a 50×50-mm board, which you can easily mount on the back of the keypad.

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

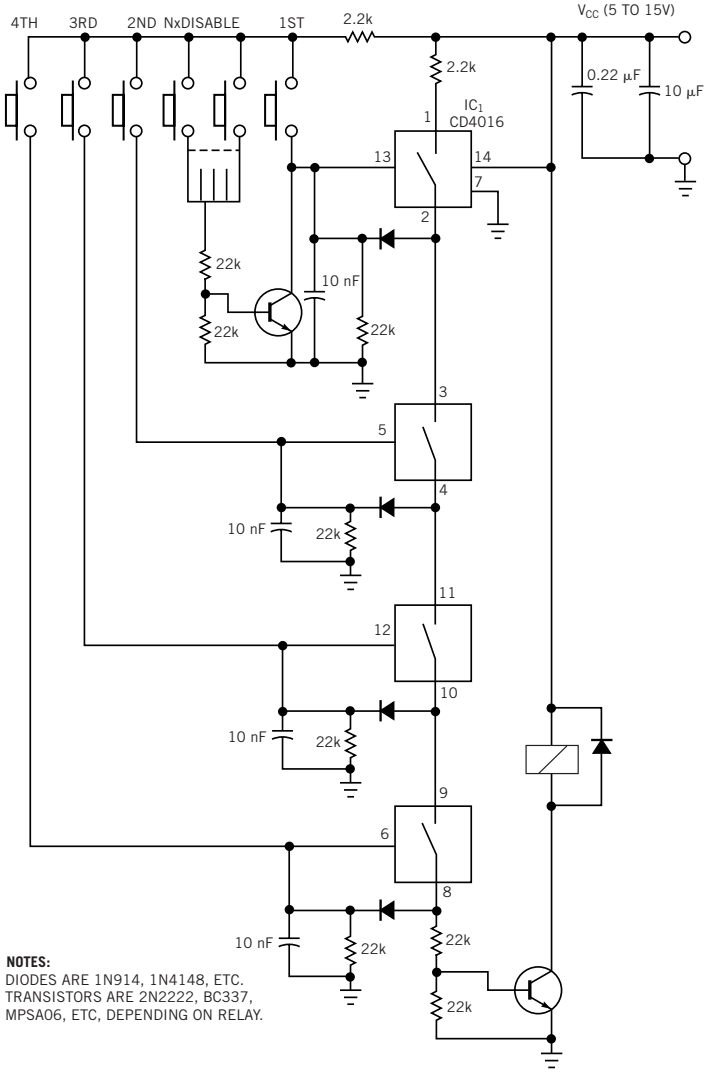


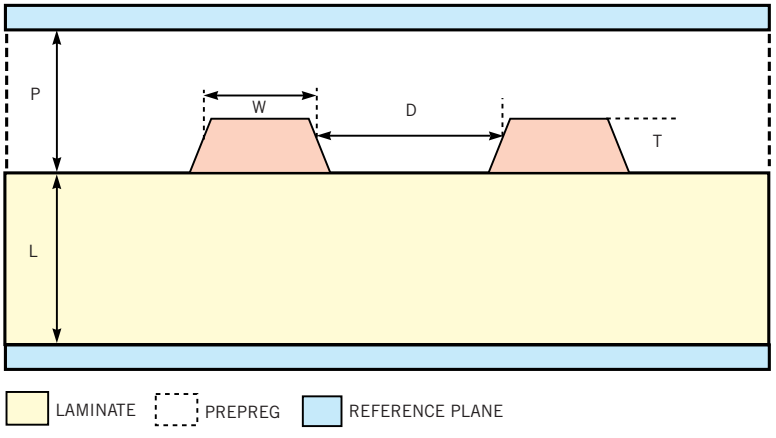
Figure 1 Forget µPs; this circuit uses a simple, sequential key press to activate a relay.

Spice models differential stripline

Vittorio Ricchiuti, Siemens ICN, L'Aquila, Italy

Differential signaling is a common technique for obtaining high noise immunity for critical signals in high-speed digital pc-board systems. The pc board's traces, carrying differential signals, are often edge-coupled offset striplines, with the traces sandwiched between two reference planes (**Figure 1**). Striplines are desirable because they offer good protection against EMI and ESD and because they allow you to tightly control tolerances in fabrication. Unfortunately, PSpice does not provide a geometry-characterized model of two edge-coupled striplines that carry differential signals. As a result, when it is necessary to simulate a differential stripline, you must use the PSpice Tline-coupled model with the line parameters (L, C, Lm, Cm) obtained by means of a 2-D field solver. **Listing 1** shows the PSpice subcircuit that represents the model of two lossless, differentially routed striplines, as in **Figure 1**.

Figure 1



PSpice can model edge-coupled offset striplines for impedance and propagation delay.

You can use the subcircuit for both transient and ac analysis. The model's input parameters are the geometrical dimension of the striplines in meters; the relative permittivity, ϵr, of the surrounding medium; the trace length in meters; and the correction factor, kc (**Reference 1**). The model calculates the odd-mode impedance of each stripline (in other words, the impedance of a single stripline when the two striplines carry differential signals) with a maximum error of 5 to 6%. The model also calculates the propagation delay. The odd-mode impedance of each line is the parallel combination of two odd-mode impedances, each one calculated with respect to each reference plane. The empirically determined correction factor, kc, takes into account the imperfection of the formulas in the model in determining the odd-mode impedance of each line. The correction factor is a function of the pc-board material and its supplier. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2599.

LISTING 1—PSPICE MODEL FOR DIFFERENTIAL STRIPLINE

```
.SUBCKT diff_stripline a1 a2 b1 b2 params: W=1u D=1u T=1u L=1u P=1 kc=1
+ er=4.6 LEN=1
.func Pi() {4*atan(1)}
.func A(x) {1+log(1+1/tanh(Pi()*D/(4*x+2*T)))/log(2)}
.func C(x) {2*log(2+T/(2*x))-(T/(2*x+T))*log(T*(4*x+T)/(4*x**2))}
.func Z1(x) {60*Pi()*x/((sqrt(kc*er))*(W+(2*x+T)*C(x)*A(x)/(2*Pi())))}
T_T1      a1 0 b1 0
+Z0={2*Z1(L)*Z1(P-T)/(Z1(L)+Z1(P-T))}
+TD={(sqrt(er)/3.0e8)*LEN}
T_T2      a2 0 b2 0
+Z0={2*Z1(L)*Z1(P-T)/(Z1(L)+Z1(P-T))}
+TD={(sqrt(er)/3.0e8)*LEN}
.ENDS diff_stripline
```


Reference

1. Riscchiuti, Vittorio, "Propagation of High-Speed Digital Signals on Printed Circuit Boards," *Future Circuits International*, Issue 5, June 1999.

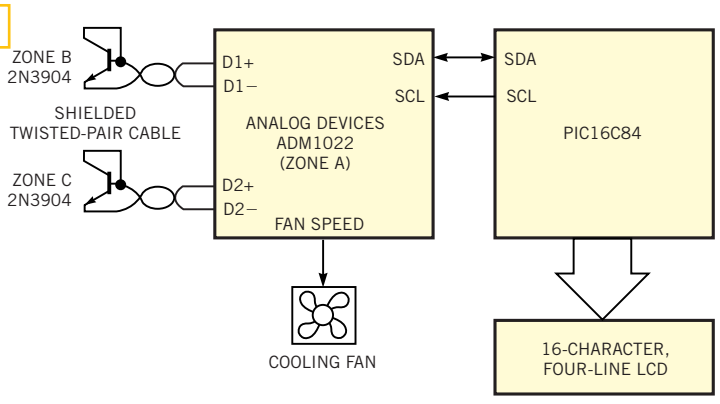
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

System monitors multiple temperatures, controls fan speed

David Hanrahan, Analog Devices Inc, Limerick, Ireland

THE BLOCK DIAGRAM in **Figure 1** represents a complete remote-temperature-sensing and fan-control system. The system uses an Analog Devices temperature-monitor and fan-control ASIC and a PIC16C84 µC from Microchip Technology. The ADM1022 allows you to measure the local temperature and two remote temperatures within a system. An on-chip, 8-bit DAC controls the speed of a cooling fan in response to the measured temperature. This circuit can form the basis of a central-heating/air-conditioning system with minimal component count and cost. The ADM1022 uses TDM (thermal-diode-modeling) techniques to accurately sense temperature. The use of readily available transistors, such as the 2N3904, eases temperature monitoring. The temperature-sensing elements remotely connect to the ADM1022, using a shielded twisted-pair cable. Zone A represents ambient temperature; the internal bandgap temperature sensor in the ADM1022 measures this temperature. Writing to the on-chip DAC controls the speed of the cooling fan.

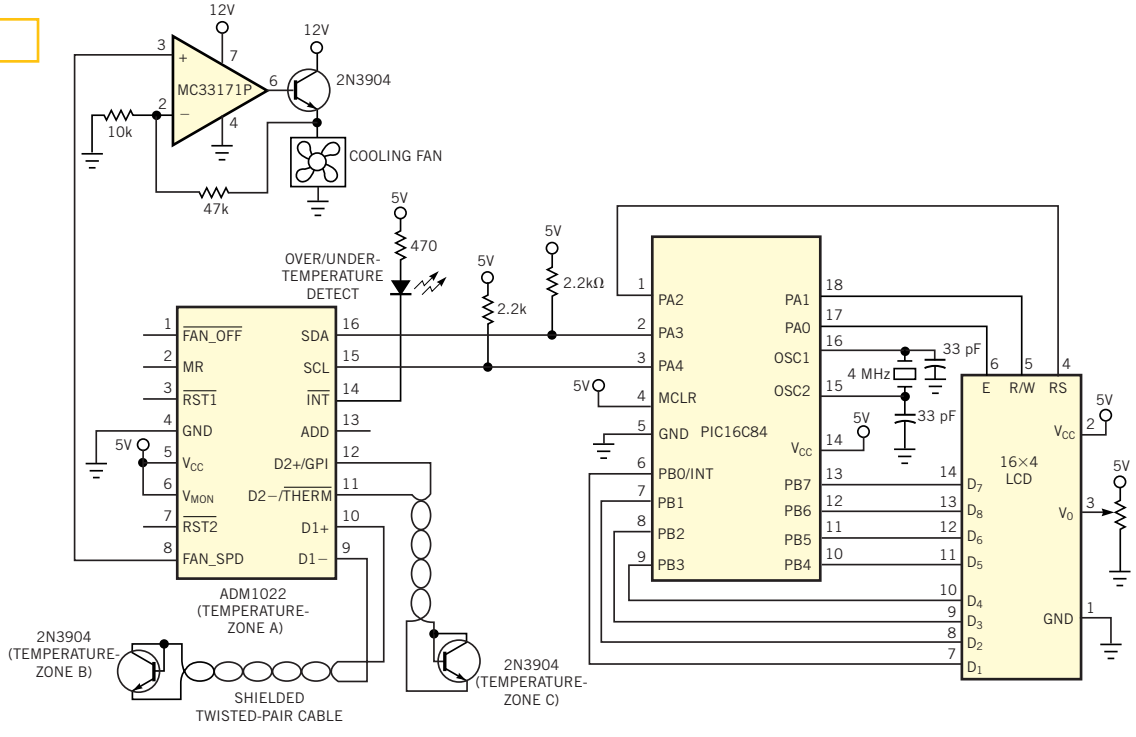
Figure 1



An ASIC and a µC monitor three temperature zones and control a cooling fan.

Figure 2 shows a complete schematic of the system. The heart of the unit is the

Figure 2



A temperature-monitor/fan-control circuit uses minimal components and space.

PIC16C84 µC. The I²C software-based communication system uses "bit-banging" of two of the pins. A 16-character-by-four-line LCD displays all measured values. The µC reads the temperature data from the ADM1022 via the I²C bus. Zone A represents the ambient temperature. Zones B and C represent temperatures at a distance from the system and use shielded twisted-pair cable. If any temperature goes outside the programmed limits, the over/undertemperature-detection LED lights up. In the schematic, the ADM1022 drives a 12V fan. You can substitute other fans or actuators by providing suitable drive circuitry. The three-state ADD pin can be high, low, or floating. Thus, as many as three ADM1022s can connect to one µC, allowing you to easily expand the system to monitor nine temperature zones.

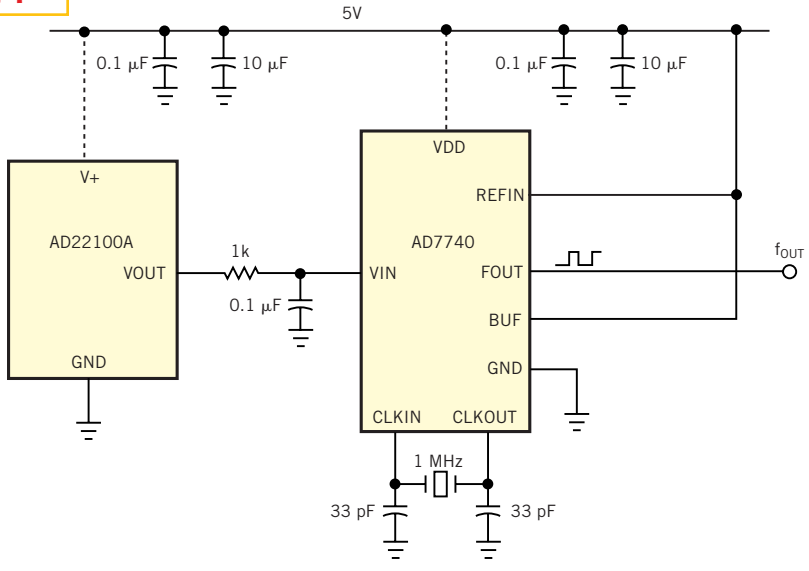
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Measure temperature in remote locations

Helen Stapleton, Analog Devices, Limerick, Ireland

THE COMPACT AND low-power temperature-to-frequency converter in **Figure 1** is ideally suited for measuring temperature in remote locations. An AD22100A temperature sensor generates a voltage proportional to the ambient temperature. This voltage then drives an AD7740 VFC (voltage-to-frequency converter), which in turn generates a digital pulse train. The duty cycle of this pulse train is directly proportional to the ambient temperature. A 1-MHz crystal defines the full-scale frequency on f_{OUT}. In noisy environments, you can use a single optocoupler to feed f_{OUT} back to the host computer. The optocoupler provides more than 2 kV of isolation between the transducer and the host. The host counts the f_{OUT} pulses. The resolution of this system is a function of the number of f_{OUT} pulses counted for each temperature reading. A count interval of $2^N/f_{OUT}$(maximum) corresponds to N-bit resolution. Hence, a trade-off exists between resolution and conversion time. The synchronous nature of the AD7740 produces a more temperature-stable transfer function than asynchronous VFCs, which are prone to errors introduced by an external capacitor.

Figure 1



This remote-temperature measurement system is immune to power-supply variations.

Both the temperature sensor and the VFC operate from the 5V supply. Connecting the REFIN of the AD7740 to the same supply eliminates the need for an external precision reference. Because the circuit is fully ratiometric, the outputs of both the temperature sensor and the VFC scale with the supply voltage, and any errors caused by supply variations cancel each other out. The AD22100A temperature sensor operates over −40 to +85°C. The corresponding output-voltage range is nominally 0.475 to 3.288V. The AD7740 converts this voltage range to a frequency range of 176 to 626 kHz. The transfer functions of these devices are as follows:

AD22100A:

$V_{OUT}-V_{DD}/5\times[1.375+(22.5\ mV/°C\times T_A)]$ V.

AD7740:

$f_{OUT}=CLKIN\times[0.1+(0.8\times V_{IN}/V_{DD})]$ Hz.

For the circuit in **Figure 1**, $f_{OUT}=(320+3.6T_A)$ kHz. The AD22100A is available in a TO-92 package, and the AD7740 comes in an eight-lead SOT-23 package. The ICs require minimal external components. When V_{IN} is buffered (BUF is at logic 1), the power consumption in the two ICs is typically 8 mW. This figure does not include the power consumed by the crystal, which is a function of the effective series resistance and the associated capacitor values.

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

PC's IRQ7 and INT1Ch measure currents, charges

K Suresh, IGCAR, Tamil Nadu, India

Figure 1 shows that you can use just two interrupts (IRQ7 and INT1Ch) of a PC and a few inexpensive components to make simultaneous measurements of two electrical quantities. Two examples are low currents and their associated charges, parameters important in ion implanters used in the semiconductor industry. The method shown here effects the simultaneous measurement of the interrelated parameters without the need for two dedicated, multiplexed analog-input channels of a data-acquisition system. Simple and simultaneous counting of the number of IRQ7 interrupts in two modes—one periodic and the other totalizing—gives the magnitude of the current and its associated charge. You can use this design as a low-current ammeter, a coulomb meter, or a current/coulomb meter with simple modifications to the software. The circuit in **Figure 1** provides simultaneous measurement of input currents of 0 to 100 µA and the corresponding charges. Because the currents under measurement are low, the circuit uses an electrometer amplifier with low bias current and noise current. The amplifier converts the input current to voltage as follows:

$$V_{OUT} = (I_{IN} \times R_F)V.$$

IC_2 inverts the amplifier's output and then undergoes digital integration, using a simple and inexpensive digital integrator, IC_3, whose output is a 10-kHz/V pulse train. Each pulse represents a fixed charge, Q, as follows:

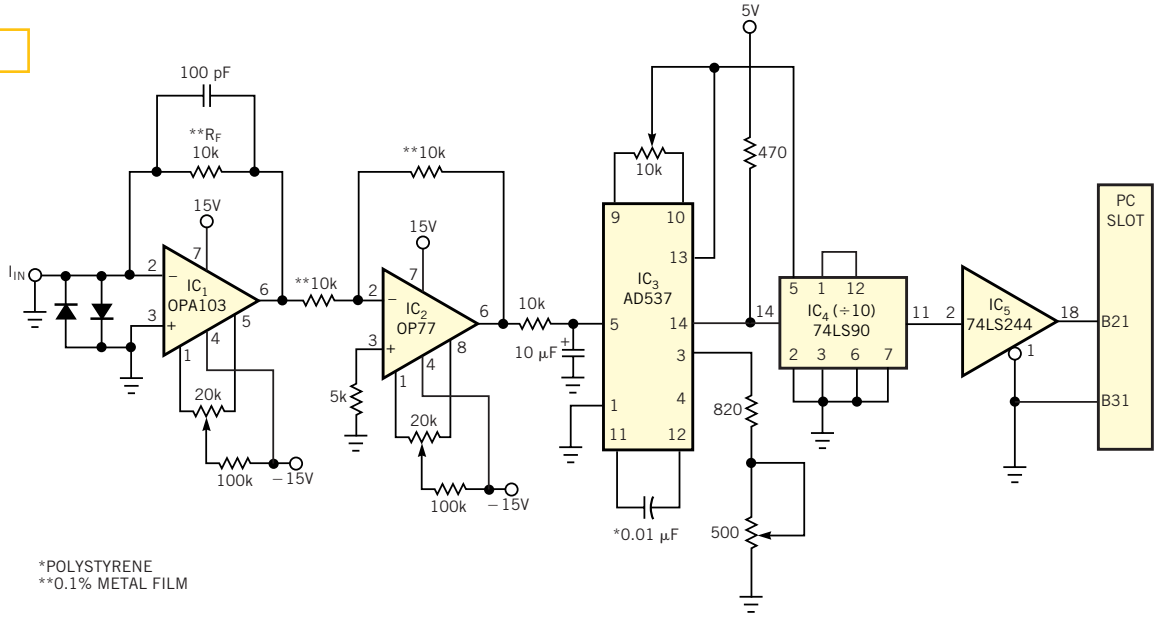
$$F_{OUT} = \frac{V_{OUT}}{10(R_1 + R_2)C}\text{Hz}.$$

The output pulses of the digital integrator, after a division by 10 in IC_4 and buffering by IC_5, interrupt the PC through IRQ7 to a maximum rate of 1000 pulses/sec. Access to IRQ7 is via the LPT printer port 10 ($\overline{ACK}$) for testing the design. However, you can also directly plug pin B21 into the PC's slot, as in **Figure 1**. Each IRQ7 interrupt represents a fixed, 100-nC charge when counted in totalizing mode over a period, T. The following equation gives the total charge associated with the current in period T:

$$\int_0^T v(t)dt = \int_0^T i(t)R_F R_F dt =$$
$$k\int_0^T [dN(t)/dt]dt = kN,$$

where N is the total count during the integration period, T, and k is the charge/count-conversion factor. However, if you periodically count the IRQ7 interrupts—say, at 1-sec intervals—you obtain the magnitude of the input current. Though the circuit shown here is designed for positive input currents, you can process bipolar input currents by using an absolute-value circuit before the digital integrator. The Turbo C program in **Listing 1** controls the entire measurement. BIOS interrupt INT1Ch, which occurs 18.2 times/sec (normally used only for time-of-day data), generates the 1-sec timebase for the periodic counting of the IRQ7 interrupts. At the start, the variables ITIMER, QTIMER, ICOUNT, QCOUNT, ETIME, and TEMP are set to zero. The INT1Ch signal causes execution of the TIMEBASE() routine to update the ITIMER and QTIMER variables, and the IRQ7 interrupt causes execution

Figure 1



You can use a PC's interrupts to measure low currents and their associated charges.

of IROUTINE(), which updates the QCOUNT, ICOUNT, and TEMP variables. Subsequently, when the ITIMER reaches 18 (approximately 0.989 sec) and a correction factor (CF) of 18.2/18 adjusts the timebase to 1 sec, the DISPLAY() routine manipulates ICOUNT and TEMP with Q and the CF to display the current magnitude and associated charge, as follows:

$$\text{CURRENT} = (Q \times \text{ICOUNT} \times CF)/1000\ \mu\text{A}.$$

$$\text{CHARGE} = (\text{TEMP} \times Q \times CF)/1000\ \mu\text{C}.$$

The routine then resets the variables ITIMER, ICOUNT, and TEMP to zero in preparation for another measurement cycle. However, the system continues to update QTIMER and QCOUNT in totalizing mode with each INT1Ch and IRQ7, respectively. At any instant, if you wish to know the total charge associated with the input current, you can press any key to obtain the total charge QTOTAL and the elapsed time, as given here:

$$\text{QTOTAL} = (\text{QCOUNT} \times Q)/1000\ \mu\text{C}.$$

$$\text{ETIME} = (\text{QTIMER} \times 54.95)/1000\ \text{SEC}.$$

You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2613.

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

LISTING 1—TURBO C ROUTINE FOR CURRENT AND CHARGE MEASUREMENT

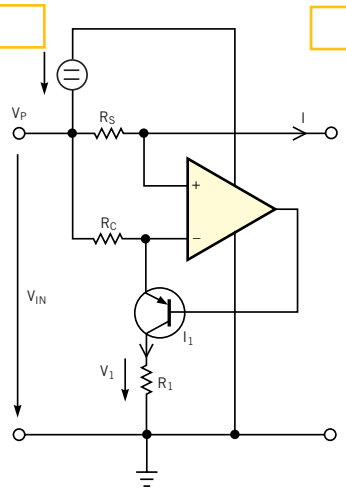
```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <math.h>

#define TIMERINTR 0X1C          /*Timer Interrupt*/
#define IRQ7 0X0F               /*IRQ7 interrupt*/

/*Global Variables*/
float Q =100.0;
int ITIMER=0,QTIMER,ICOUNT=0,QCOUNT=0;
float CF, CURRENT=0.0,ETIME=0.0,CHARGE=0.0;
long float QTOTAL=0.0
int TEMP=0;

static int EXISTINGIMR,OURIMR;

void interrupt (*oldfunc)(void); /* IRQ7 interrupt function pointer */
void interrupt IROUTINE();
void interrupt (*oldvect)(void); /*Timer Tick INT1Ch pointer */
void interrupt TIMEBASE();

void interrupt IROUTINE()   /* ISR  for IRQ7  */
    {
      disable();
      TEMP++;
      QCOUNT ++;
      ICOUNT++;
      outp(0x20,0x20);      /* End of Interrupt */
      enable();
     }

  void interrupt TIMEBASE() /* ISR for INT1Ch & Routine for 1Sec time
base
                                                    generation
*/
     {
       disable();
       ITIMER++;

       QTIMER++;
       enable();
      }

 void INITIALISEIRQ7()
   {
     disable();
     EXISTINGIMR=inportb(0x21);
     OURIMR=EXISTINGIMR & 0X7F;     /*Enable IRQ7 to 8259 PIC*/
     outportb(0x21,OURIMR);
     oldfunc=getvect(IRQ7);
     setvect(IRQ7,IROUTINE);
     enable();
   }

 void INITIALISETIMER()
   {
     disable();
     oldvect=getvect(TIMERINTR);
     setvect(TIMERINTR,TIMEBASE);
     enable();
   }

 void RESTOREALL()
  {
     disable();
    outportb(0x21,EXISTINGIMR);
    setvect(TIMERINTR,oldvect);
     setvect(IRQ7, oldfunc);
     enable();
  }

 void DISPLAY()
 {
   disable();
   CURRENT= (Q*ICOUNT*CF)/1000.0;   /*Calculate Current from ICOUNT*/
   Gotoxy(15,15);
   printf("\nCURRENT=%6.2f uA",CURRENT); /*Display the current
measured*/
   CHARGE= (TEMP*Q*CF)/1000.0;              /*Calculate Charge from
TEMP*/
   printf("\nCHARGE = %6.2f uC", CHARGE);/*Display the charge
measured*/
   enable();
}

void main(void)             /* Main program starts here */
{
    int c,a,lpt;
   float CF;
   CF=18.2/18;          /*Correction Factor for time base*/
   clrscr();

   lpt=peek(0x40,0x08);
   printf("\n\tPC'sIRQ7,INT1Ch provide simultaneous measurement of ");
   printf("\n\t \t\t Low Currents and Associated Charges");
   printf("\n\t\t\t\t by");
   printf("\n\t K.SURESH, MSD,IGCAR,Kalpakkam,TamilNadu,India 603
102");
   outp(lpt+2,(inp(lpt+2)|0x10));/*Enable IRQ7 to the printer adapter*/
   outp(0x20,0x20);
   INITIALISEIRQ7();
   INITIALISETIMER();

   while(!kbhit())
   {
     if (ITIMER>=18)
        {
          DISPLAY();
          ITIMER=0;
          ICOUNT=0;
          TEMP=0;
        }
    }    /*Measurement continues*/

 /*Display the total associated charge due to the input current*/
QTOTAL=(QCOUNT*Q)/1000.0;
printf("\n\n Total Associated Charge during
measurement=%.2lfuC",QTOTAL);
ETIME= ((QTIMER*54.945)/1000); /*Display the total elapsed time*/
printf("\n Total Elapsed Time= %7.3f sec ", ETIME);
RESTOREALL();
return ;
}                   /*          END OF MAIN       */
```

Simple circuit provides high-side current sensing

M Ossmann and B Kazay, Aachen University of Applied Sciences, Germany

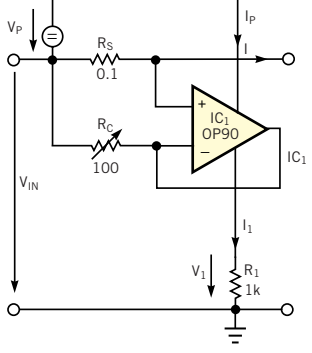
In power supplies or battery chargers, you often need information about the current flowing in the high-side rail. **Figure 1** shows a common circuit for obtaining this information. The circuit provides a ground-referenced voltage proportional to the current flowing through the high-side sense resistor, R_S. The circuit needs an additional high-side supply, V_P. If you use a low-voltage op amp, such as an OP90, V_P=1.5V is adequate, so you can derive this supply from a series-transistor voltage drop in many applications. The op amp keeps the voltage drop in R_S and the conversion resistor, R_C, equal, so the current through R_1 is $(R_S/R_C)I$, and the voltage across R_1 is $I(R_S/R_C)R_1$. **Figure 2** shows a circuit that needs no auxiliary transistor. The output stage of the op amp, in a sense, replaces the transistor. The current through R_C is the same as that in **Figure 1**. Now, this current flows through the negative-supply pin of the op amp and serves as the current source for R_1 as before. The circuit uses a low-power OP90 op amp, which draws a low supply current. The idea is that the supply current contributes a negligible offset to I_1. Measurements show that the positive-supply current, I_P, rises with current I_1. This situation leads to a higher current I_1 because of the current I_P. **Figure 3** shows the result. The dots in **Figure 3** indicate the measured values of V_1, and the straight line shows the value if I_P were zero.

Figure 1



This method for measuring high-side supply current requires an auxiliary transistor.

Figure 2



This circuit uses the negative-supply current of the op amp to measure the high-side supply current.

Fortunately, the supply, I_P, is nearly proportional to the op amp's output current. So the value of R_C in **Figure 2** is such to yield the right value at the maximum current of 2.5A. This simple, one-point adjustment corrects the gain error. **Figure 4** shows the result after this gain compensation. Now, the error between the measured V_1 and the desired value is within 5%, which is tolerable for many applications. Measurements show that the accuracy holds for input voltages of 5 to 25V and an additional supply voltage of 2 to 15V.

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

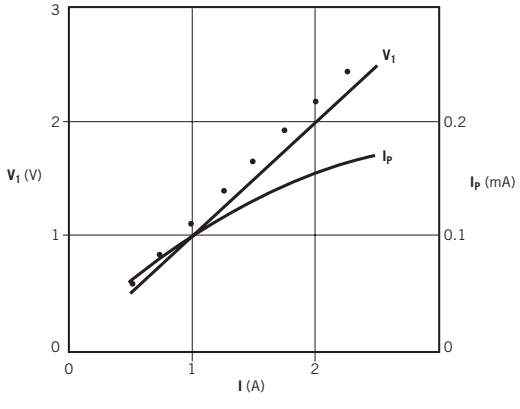


Figure 3 Because of the positive-supply current, the measured values (dots) do not agree well with the theoretical values (straight line).

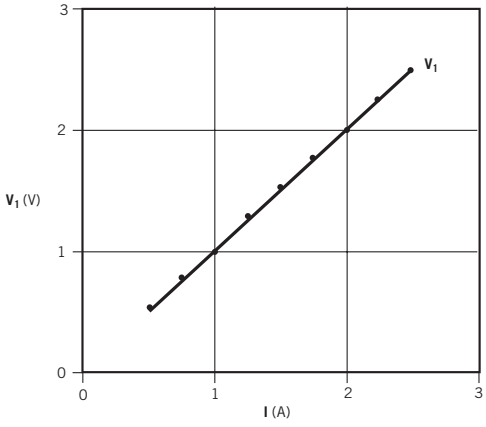


Figure 4 Gain compensation by adjusting R_C improves the accuracy of the measurement.

design ideas

Edited by Bill Travis and Anne Watson Swager

Cheap pushbutton replaces rotary switch

Abel Raynus, Armatron International Inc, Melrose, MA

The problem of setting any one of four modes of a µC-operated process has a straightforward solution: Have the µC check the states of some 4 bits of a µC register or an input port and then execute one of the four predetermined subroutines according to the result. The next question is how the operator sets these 4 bits or, specifically, selects a desirable mode of operation.

One approach is to use a rotary switch (**Figure 1a**). This circuit does the job but is rather expensive for consumer applications. You can also reject DIP switches because even a qualified person can choose a wrong setting when using this solution. Slide switches are another possibility, but those in the catalogs have no more than three positions (SP3T), and this application requires four.

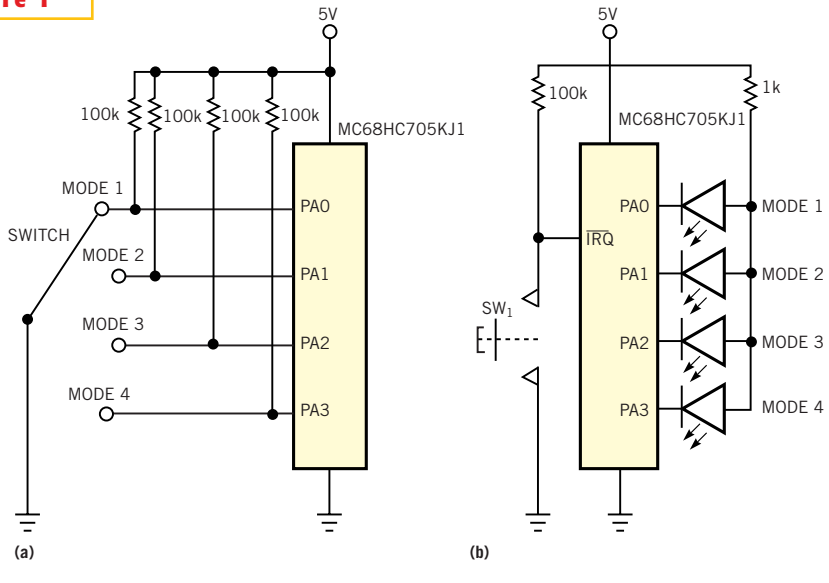
Figure 1b shows a simple and cost-effective solution that uses one pushbutton switch and four LEDs for mode indication. (You can download the corresponding µC program from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2605.)

The four µC bits PA0 through PA3 serve as a mode register and simultaneously as a mode indicator via the LEDs. After the µC resets, the initialization routine puts a low level on PA0, which sets MODE 1 and lights the corresponding LED. All other LEDs remain off, and all the rest of the bits of Port A stay unchanged. For the intended project application, it is more convenient to organize the mode-switching process as an external interrupt and let the main program perform its other functions. When an operator pushes the normally open switch, SW_1, the resulting negative edge at the IRQ pin of the µC causes an external interrupt. The interrupt-service routine shifts the low level to the next output pin consecutively from PA0 to PA3. When the routine reaches PA3, it goes back to PA0, and so on. The service routine has no effect on bits PA4 through PA7, and they remain unchanged. The debouncing delay is 54 msec.

The circuit in **Figure 1** uses an inexpensive µC, but the program contains the standard instruction set and therefore is applicable to any µC.

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

Figure 1



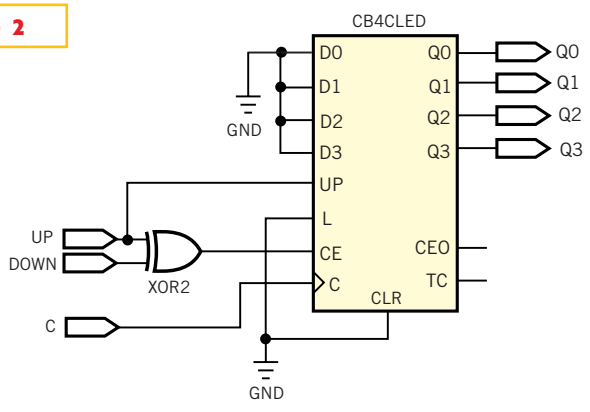
As an alternative to a rotary switch (a), you can use a pushbutton switch (b) to cycle through four modes of a µC-controlled process. Four LEDs indicate the selected mode.

FPGA makes simple FIFO

Luis Miguel Brugolaras, SIRE, Madrid, Spain

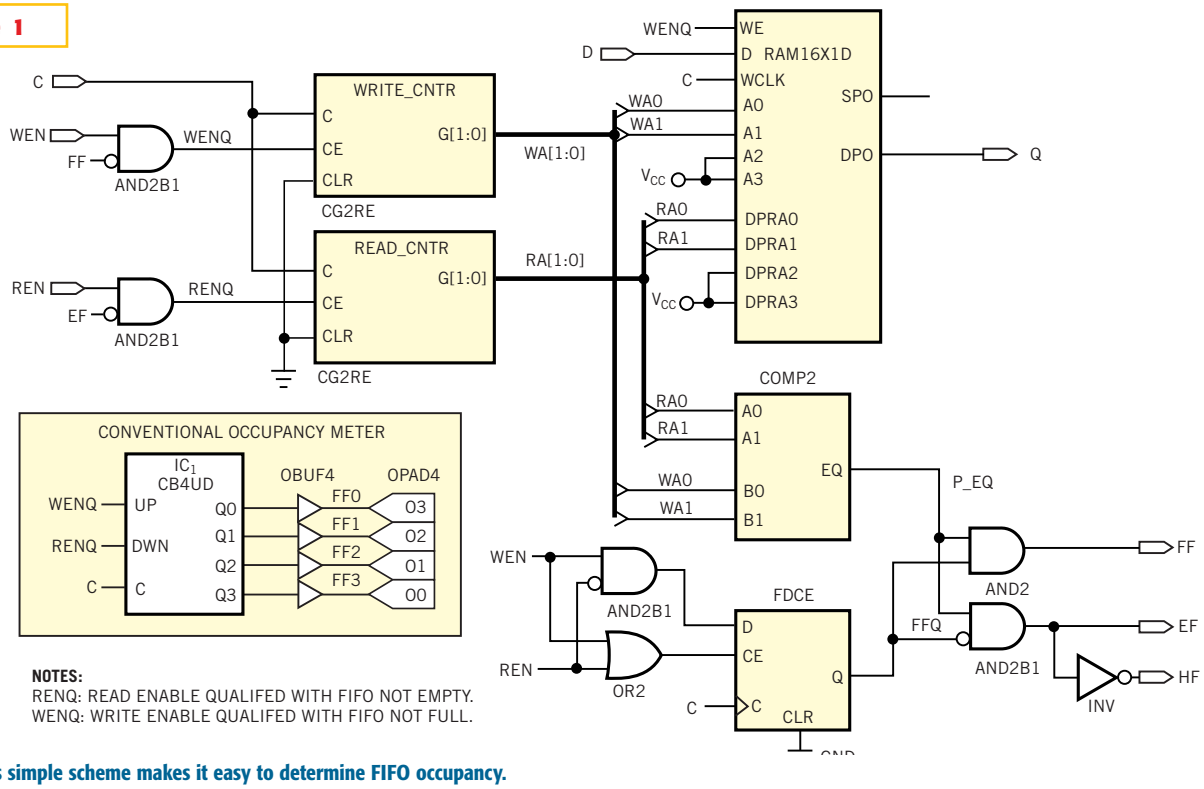
THE CIRCUIT IN **Figure 1** is an FPGA-based, synchronous FIFO that uses the same clock for read and write operations. The circuit can generate FIFO-occupancy flags with a minimum of logic. The boxed area in **Figure 1** shows a more conventional occupancy meter. The circuit is implemented in a demultiplexer that writes data in the FIFO when the data arrives and reads data according to FIFO occupancy. The circuit uses a Xilinx Spartan (XC4000 equivalent) FPGA. The method uses three main blocks: a 16-bit dual-port RAM macro, read- and write-address counters, and the flag processor. In this design, the FIFO is 4 bits deep but can be as great as 16 bits deep using the RAM16X1D macro. Read and write counters can take the form of any cyclic counter, a conventional binary counter, a Gray-code counter, or a linear-feedback shift register. FIFOs commonly use Gray-code counters. These counters have the property that only one bit changes from state to state. Thus, they have the advantage of not providing intermediate false states when the counter advances. For example, a binary counter moving from 0111 to 1000 changes all its bits but with different delays and thus can fool asynchronous comparison logic. Linear-feedback shift registers have the advantage of requiring modest logic resources and can work at high clock rates. They pose one small inconvenience, however: For an n-stage shift register, only 2^n-1 states exist.

Figure 2



The CB4CLED increments upon assertion of UP and CE and decrements upon assertion of CE and deassertion of UP.

Figure 1



This simple scheme makes it easy to determine FIFO occupancy.

You can read FIFO occupancy in an orthodox way, by using an up/down counter controlled by read- and write-enabled clocks (**Figure 2**). This method makes the counter advance up when the FIFO is written to but not read, advance down when the FIFO is read but not written to, and remain unchanged otherwise. Such an arrangement leads to full knowledge of FIFO occupancy, a valuable asset during debugging. For example, you can use a DAC to convert the occupancy data and monitor it with an oscilloscope. In this application (**Figure 1**), it was necessary to keep logic to a minimum. We simply needed to know whether the FIFO was full (to avoid loss of data in a write attempt), or empty; hence, the application needed just Full and Empty flags. The addresses of the read and write pointers are equal only when the FIFO is either full or empty. These states are distinguishable because, before the FIFO became full, a write operation occurred after no previous read operation. A flip-flop qualifies this status by latching data only when a read or write attempt occurs. The flip-flop latches a high level when a write operation (but not a read operation) occurs, indicating an attempt to increment the FIFO's occupancy.

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

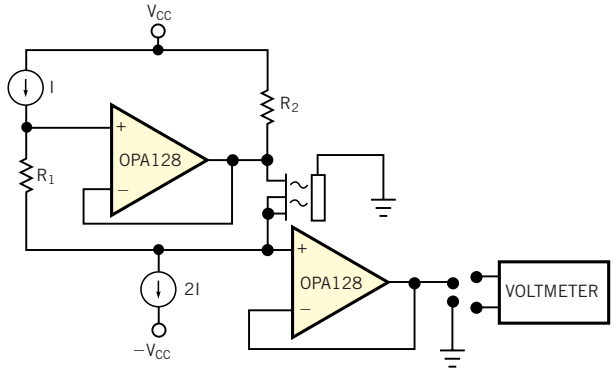
Circuit provides constant current for ISFETs/MEMFETs

S Casans, D Ramirez, and AE Navarro, University of Valencia, Burjassot, Spain

ISFETs and MEMFETs (ion-sensitive and membrane FETs) are solid-state chemical sensors that provide an electrical signal associated with the change of a chemical concentration in a solution. These types of sensors need a stable operating point. A common operating point is a drain-source current of 100 μA and a drain-source voltage of 500 mV. Under these conditions, the sensed chemical information is contained in the gate-source voltage. **Figure 1** shows a constant-current driver for ISFETs/MEMFETs. The current source, I, produces a voltage drop in resistor R_1; the voltage follower reflects this voltage to the drain-source terminals of the chemical sensor. In this example, VDS is equal to IR. The entire current, I, traverses resistor R_1, because the op amp is an electrometer type with femtoamp-range input-bias current. The sensor's drain-source current is the difference between the constant currents, 2I and I—in this case, I. A REF200 provides the constant currents in the circuit in **Figure 1**. The REF200 has two internal 100-μA current sources and a current mirror.

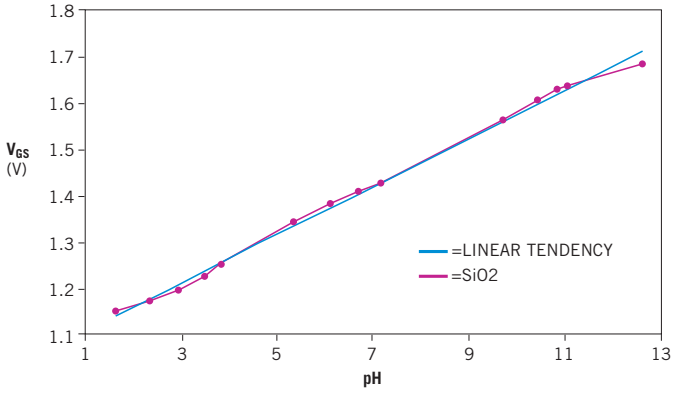
One of the 100-μA current sources provides I in **Figure 1**. The lower 2I current source uses the REF200's second

Figure 1



A constant-current driver provides the basis for a chemical-concentration test system.

Figure 2

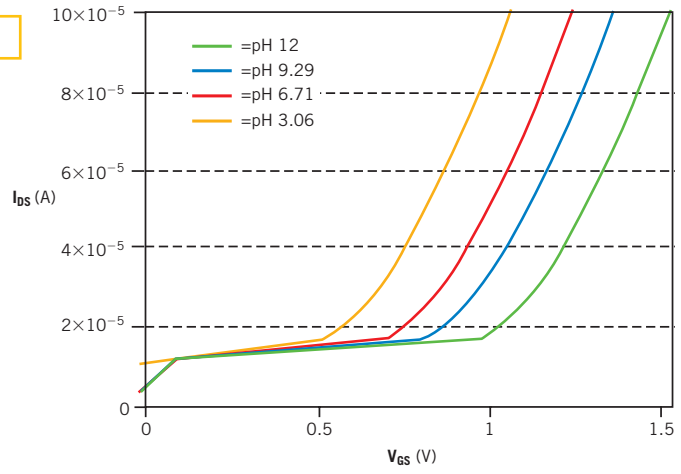


The gate-source voltage shows good correlation with the concentration of [H¹] ions.

100-μA current source and the internal current mirror. The second voltage follower measures the gate-source voltage of the sensor—hence, the chemical concentration of interest. **Figure 2** shows experimental results. You can see that the setup provides a linear relationship between gate-source voltage and the concentration of $[H^+]$ ions. The experiment used SiO_2- and Si_3N_4-based ISFET sensors. **Figure 3** shows the variation of the threshold voltage as a function if pH.

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

Figure 3



The gate-source voltage of an ISFET shifts with the degree of acidity.

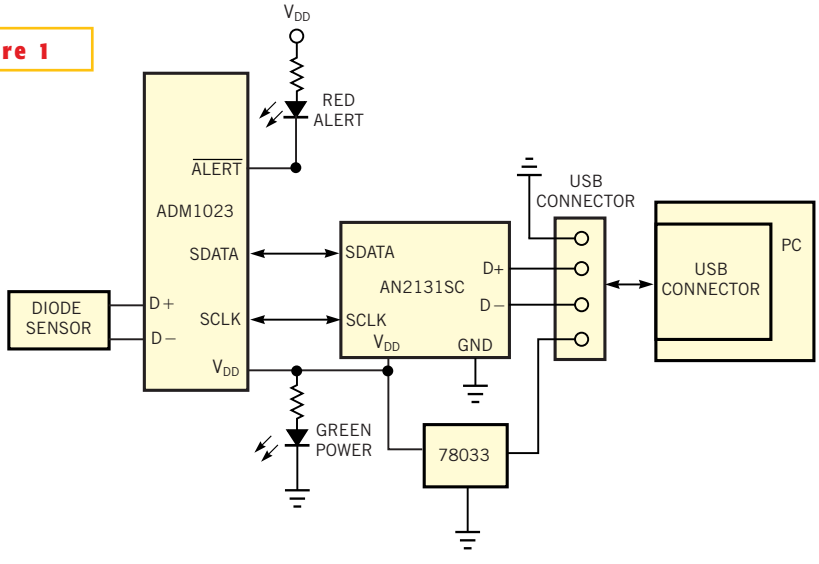
USB link neatly measures temperature

Mary Burke, Analog Devices, Limerick, Ireland

Figure 1 shows a convenient and neat way to measure temperature using the USB of a PC. An ADM1023 IC senses the temperature, and an Anchor Chips EZ-USB AN2131SC μC provides control. The ADM1023 is a low-cost device with an accuracy of 1°C and a resolution of 0.125°C. The IC is a two-channel device that can measure its own local temperature and the temperature of a remote location. You obtain remote-temperature sensing by connecting a remote thermal diode between the D+ and D− pins of the μC. You can locate this remote diode (for example, a diode-connected 2N3906) in a hot spot that can be as far as 100 ft from the ADB1023. However, for distances greater than 12 in., you should use twisted-pair cable. For distances greater than 12 ft, you should use shielded twisted-pair cable, such as microphone cable.

The EZ-USB μC includes an integrated I²C controller. This feature is useful in that the μC communicates via I²C, thereby simplifying both the circuit and the μC's firmware. The μC takes care of the lower level I²C signals. The firmware needs only to place the data destined for the I²C data register and to tell the μC to send it. A second advantage of this μC is the availability of on-chip RAM. You can write Windows drivers such that the firmware automatically downloads to the RAM when you plug in the board. The μC then simulates a disconnect and reconnects the board as a new device. This

Figure 1



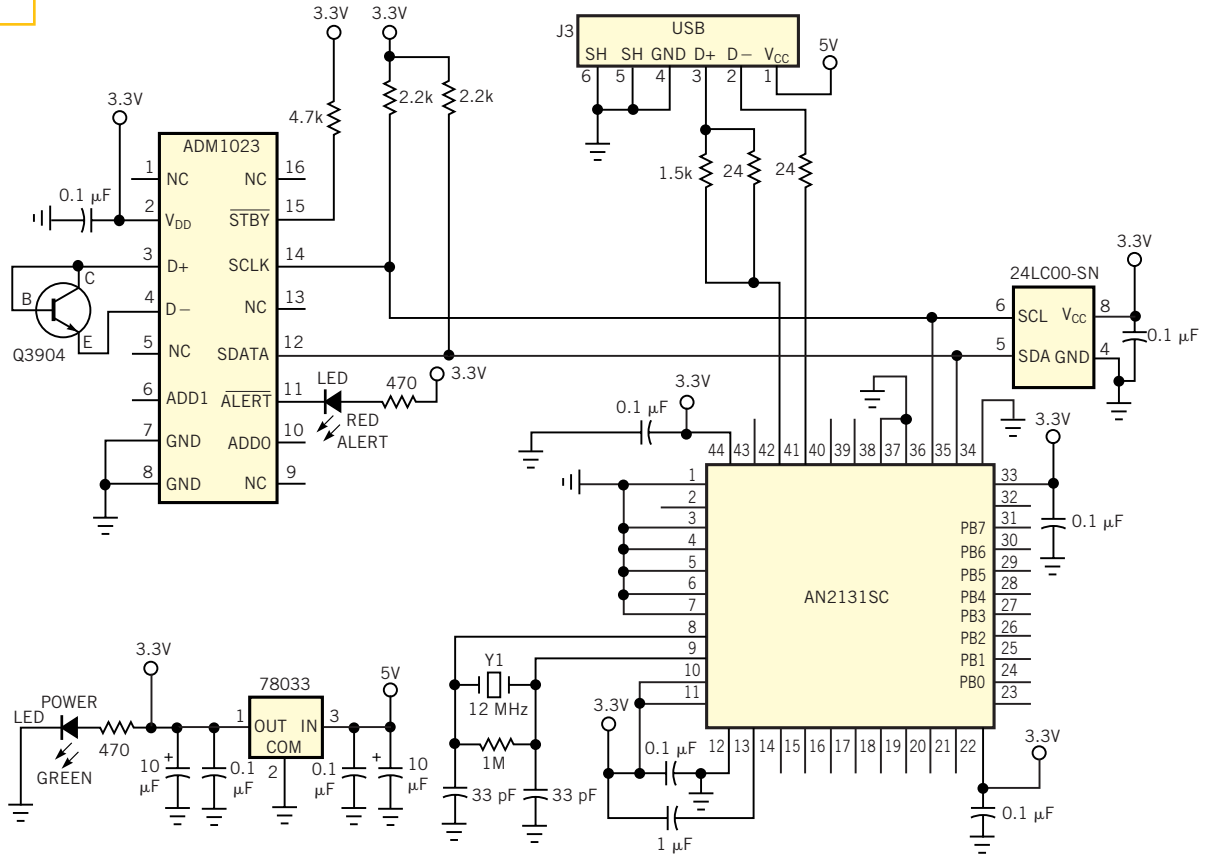
Use the USB port on your PC to make quick and easy temperature measurements.

feature of the µC is called renumeration. The circuit in **Figure 2** works as follows: The SDATA and SCLK pins of the ADM1023 connect to the corresponding pins of the µC. You need a 2.2-kΩ pullup resistor on both lines, because these pins have open-drain outputs. Also connected to these lines is a 24LC00, a 16-byte EEPROM. This ROM stores the board's device descriptors (the vendor and product IDs). Windows uses this information to identify which device driver to use.

The remote thermal diode (or diode-connected transistor) connects between the D+ and D− lines of the ADM1023. The ADM1023 can also signal an Alert. You can program the IC with high and low temperature limits for both the local and the remote channels. If any of the measured values of temperature are beyond the temperature limits, then the ADM1023 signals an Alert. The $\overline{\text{ALERT}}$ is active-low; an LED tied to the pin lights whenever an Alert signal arises. The USB port supplies power to the circuit. The port can supply as much as 500 mA at 5V. Because both chips in **Figure 2** operate at 3.3V, the circuit uses a 78033 regulator to generate the required 3.3V. The USB Master, which in this case is the application running on the PC, maintains control of the circuit. The master initiates all USB communications. The temperature-measurement circuit is the slave. It responds only when the master requests it to do so. When the master requests data (via the PC application), the request travels down the USB cable to the µC. This request consists of the device address of the ADM1023 and the address of the register that stores the data. It also tells the µC whether a read or a write is required. The µC then uses this information to interrogate the ADM1023 over the I^2C interface. You can download the software used in this project from *EDN*'s Web site. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2596.

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

Figure 2



This temperature-measurement system requires minimal components.

Single-sideband demodulator covers the HF band

Israel Schleicher, Bakersfield, CA

The circuit in **Figure 1** complements a previous Design Idea ("SSB modulator covers HF band," *EDN*, Sept 30, 1999, pg 122). The modulator employs a phasing network to split a low-frequency audio signal into in-phase and quadrature (orthogonal) components. This circuit delivers a phase error of only 0.15° and has a low sensitivity to component tolerances, which are advantages over other phasing circuits (**Reference 1**). By reversing the direction of the network, that is, feeding the output with two orthogonal signals and tapping the input, the network functions as a detector. Feeding the two signals one way may produce a signal at the input, but if you interchange the two signals, no signal will go through.

Because the network in the modulator circuit has two floating differential outputs, the demodulator network requires two floating sources. The simplest way to accomplish this requirement is by using transformers. T_1 and T_2 are 600Ω 1-to-1 telephone-coupling transformers with a center-tapped bifilar primary. It is important to minimize the capacitance between the primary and the secondary windings.

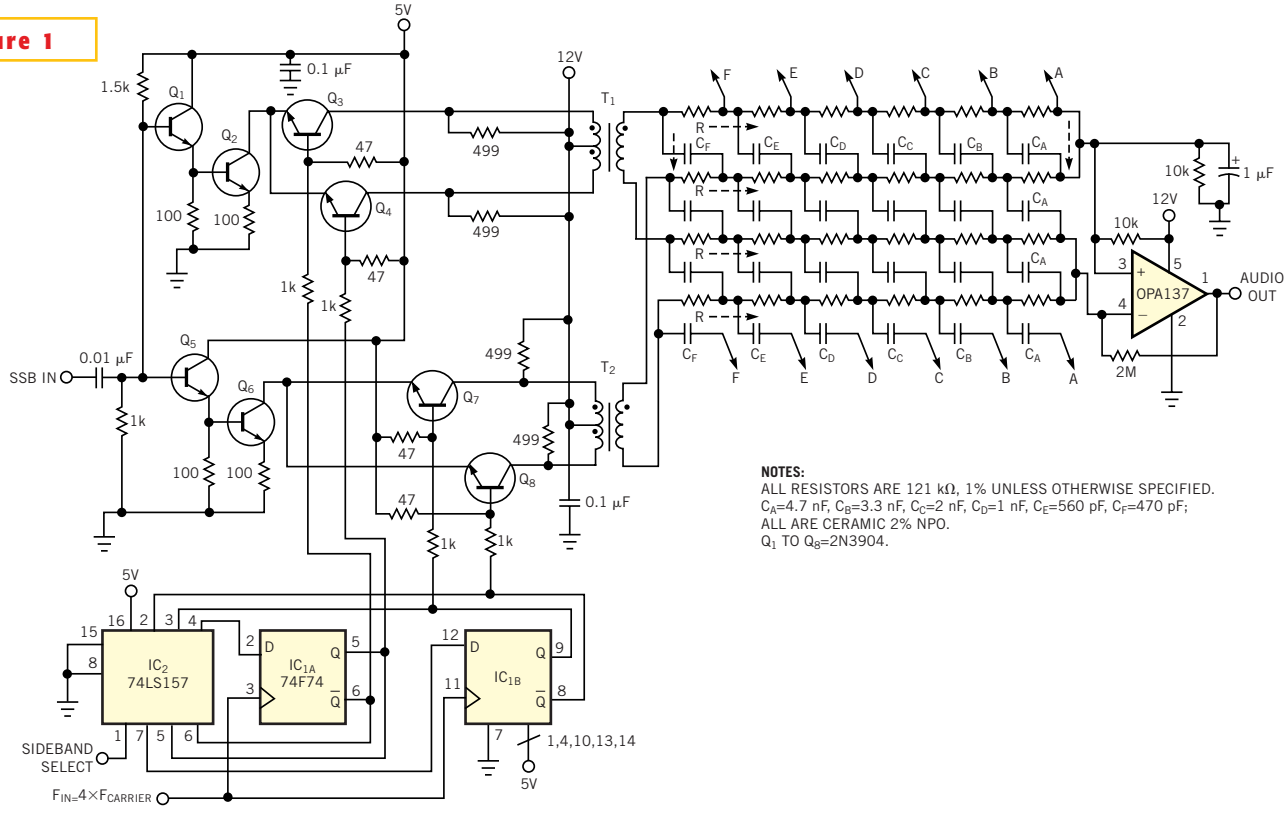
Q_1 to Q_4 and Q_5 to Q_8 function as balanced mixers. They provide a high dynamic range for the circuit, which is part of a direct-conversion receiver. IC_1 provides two, quadrature LO signals, and this IC requires a drive of four times the carrier frequency. IC_2 allows for upper or lower sideband selection. The prototype circuit measures 37 dB of unwanted sideband rejection for a 1-kHz modulated carrier and 32-dB rejection for a 3-kHz modulation. A sharp 3-kHz lowpass filter must follow the circuit.

Reference

1. Zavrell, Robert, J, "New low-power single sideband circuit," Philips Semiconductor, AN1981.

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

Figure 1



A single-sideband demodulator uses two transformers as floating sources.

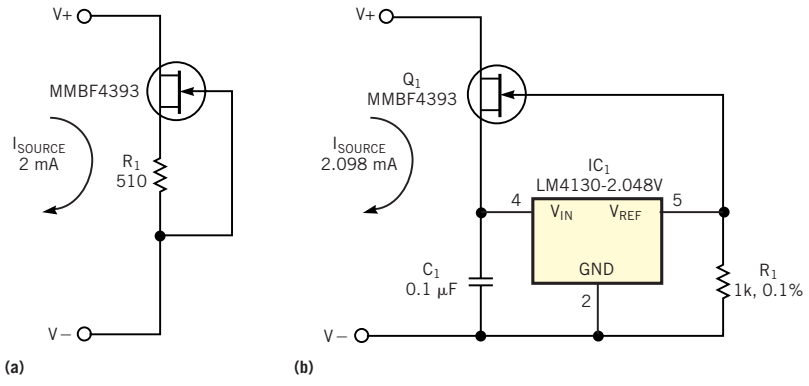
Voltage reference improves JFET

Clayton Grantham, National Semiconductor, Tucson, AZ

THE COMMON JFET current-source in **Figure 1a** has average output impedance and depends heavily on the JFET's V_P and I_{DSS} variations. These manufacturing variations limit the initial accuracy of the current setpoint that the circuit can obtain with a fixed R_1. Most users of JFET current-sources sample V_P and I_{DSS} for each lot of JFETs and then select R_1 by first solving the quadratic equation $I_D = I_{DSS}(1 - V_{GS}/V_P)2$ for V_{GS} and then solving the current-setpoint equation $I_{SOURCE} = V_{GS}/R_1$. After completing these lot-specific calculations, the accuracy error for I_{SOURCE} can be less than ±15%. In addition to initial accuracy error at ambient, the V_{GS} overtemperature performance limits the current temperature drift to 0.3%/°C. The temperature dependence of V_{GS} is a function of both the mobility variation of I_{DSS} with I_{DSS} negative-temperature coefficient and the built-in potential of V_P's positive coefficient. V_{GS} has one I_D operating point at which it has a zero temperature coefficient, but this one current setpoint is probably not the one that you wanted. With resistors commonly available at 0.1%, 25 ppm/°C grade, R_1 will be a small contributor to any errors in the current setpoint. Compliance voltage across the current source is V_{GS} to BV_{DSS}. The output impedance, R_O, for this current source is approximately 1 MΩ. You can improve R_O to approximately 10 MΩ by adding another JFET in a cascode configuration.

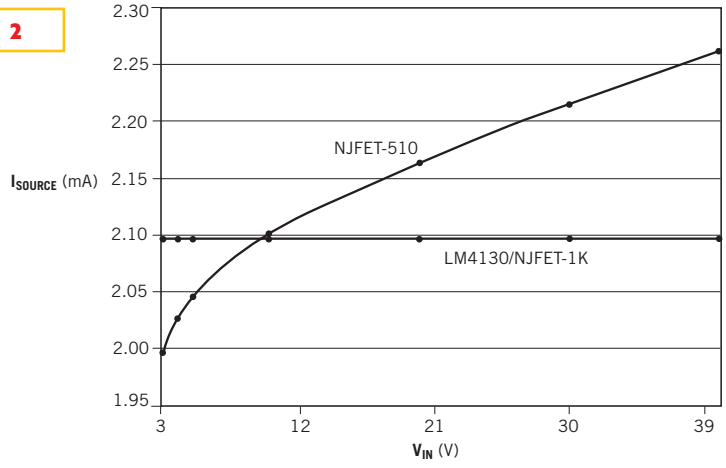
In contrast, the composite circuit in **Figure 1b** adds an IC voltage reference to the JFET that improves the output impedance. I_{SOURCE} is set by IC_1's output voltage divided by R_1 plus the small ground-current (50 µA) of IC_1. The good supply rejection of the LM4130 nulls any V_{GS} variation of the JFET. The current path is from V+ through the n-channel JFET (drain to source) into IC_1's supply input and then out V_{REF} and through R_1 to V−. The gate current of Q_1 is in the picoamp range. The JFET's V_{GS}, approximately 1.2V, keeps IC_1 biased well above its dropout level.

Figure 1



The common JFET transistor current source (a) has an average output impedance. A composite voltage reference and JFET circuit (b) features higher output impedance, high accuracy, and low temperature drift.

Figure 2



The composite configuration improves the variation of I_{SOURCE} with a change in V_{IN}.

Figure 2 shows I_{SOURCE} versus V_{IN} of both the JFET and the composite current sources. The slope of these two operating plots represents the inverse of output resistance. The composite circuit curve has an output resistance of greater than 200 MΩ as compared with the JFET-only output resistance of 0.2 MΩ.

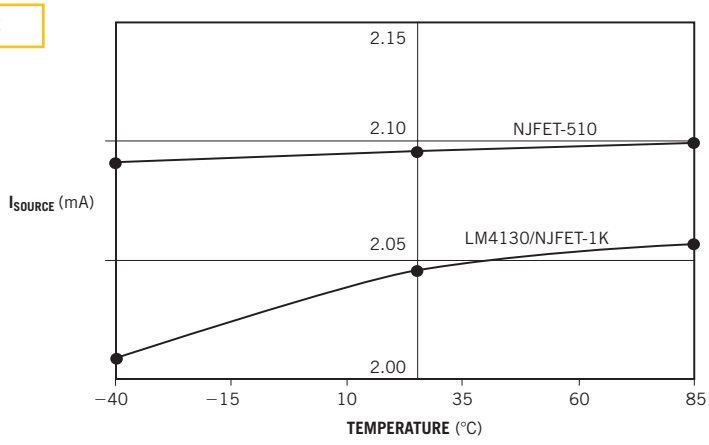
To a slighter degree than output impedance, the circuit in **Figure 1b** also improves the initial accuracy of I_{SOURCE}. The

high accuracy achievable with IC_1 and R_1 controls the setpoint. Both these components are available in at least 0.1% grades. However, IC_1's ground current of 50 µA introduces an offset to I_{SOURCE}, which you must include in the calculation. IC_1's ground-current variation of ±7 µA is the practical limit of I_{SOURCE}'s accuracy. **Table 1** summarizes the impact on the practical accuracy that you would attain for a range of I_{SOURCE} values.

Figure 3 compares the overtemperature performance. Again, IC_1's ground-current variation dominates for I_{SOURCE} set below 200 µA. IC_1's variation of ground-current overtemperature is ±5 µA.

The compliance voltage across the composite is V_{GS}+2.1V to BV_{DSS}+5.5V, or approximately 3.5 to 36V. Regarding overvoltage stress, when V_{IN} goes above BV_{DSS}+5.5V, the composite circuit's setpoint permanently goes into the microamp range due to the CMOS structures within the LM4130 that the overvoltage destroys. The JFET-only configuration may shift the setpoint by 20% but otherwise will be more forgiving of an overvoltage stress. You can further enhance the composite configuration by cascoding the JFET, which boosts the output resistance to approximately 1 GΩ and extends the V_{IN} range to more than 70V.

Figure 3



The composite configuration also offers a slight improvement in the variation of I_{SOURCE} with a change in temperature.

TABLE 1–IMPACT ON ACCURACY FOR I_{SOURCE}

I_{SOURCE} (mA)	R_2 (Ω)	Accuracy-error range (%)	Temperature-coefficient range (ppm/°C)
4.066	510	±0.4	±75
2.098	1k	±0.6	±95
1.074	2k	±1.0	±135
0.255	10k	±3.0	±380
0.126	27k	±6.0	±980

Note:
R_2=0.1%, 25 ppm/°C.

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

Wideband filter uses image parameters

Richard M Kurzrok, RMK Consultants, Queens Village, NY

You can design various lowpass and highpass filters using image parameters (**references 1** and **2**). By cascading a highpass and a lowpass filter using image-parameter designs, you can also realize a low-cost wideband filter (**Figure 1**). A different approach uses a composite design with terminating m-derived half-sections and two interior constant-k full sections. This approach results in viable performance with the use of relatively few components. For this approach, however, m=0.5, which is not the preferred value for classic-filter designs.

In **Figure 1**'s filter, the upper and lower cutoff frequencies of this overall bandpass filter can be independent design parameters. For this filter, the nominal highpass cutoff frequency is 3.3 MHz, and the lowpass cutoff frequency is 10 MHz. Impedance levels are 50Ω. A male-to-male BNC adapter connects the two separate filter units. At higher frequencies, total integration of the two filters in a single enclosure would be desirable. Use of surface-mount components might also be appropriate.

Table 1 shows the measured amplitude responses of the highpass and lowpass sections. **Table 2** summarizes the overall

bandpass filter's amplitude response. You can see that the highpass and lowpass cutoff frequencies are sufficiently removed from each other to realize a fairly wide passband of low insertion loss. If the cutoff frequencies of the highpass and lowpass filters are too close to each other, you can reduce interactions between the two filters by using a fixed-pass attenuator to connect the two filters.

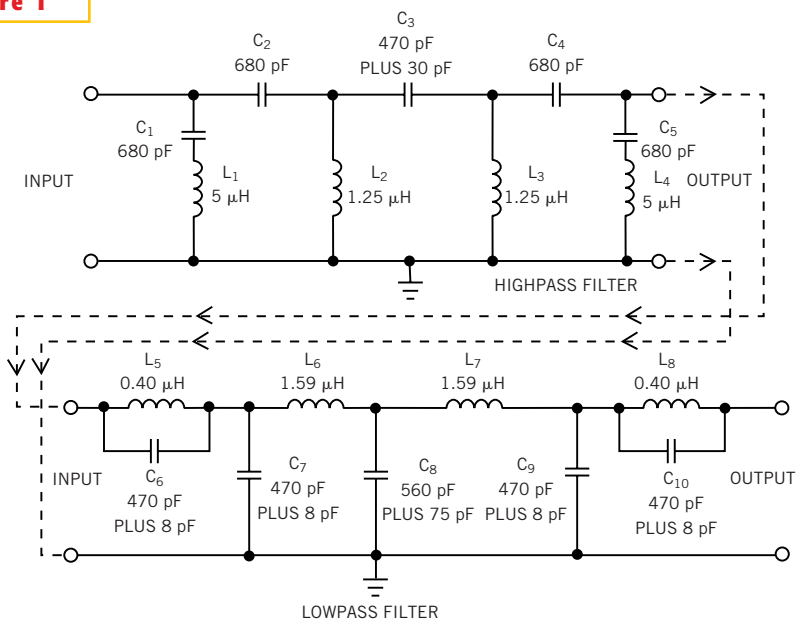
The design computations are simple, harking back to the days of slide-rule designs. You can gauge the theoretical performance of composite lowpass filters using modern computer-aided analysis (**Reference 3**).

References

1. Kurzrok, Richard M, "Low cost lowpass filter design using image parameters", *Applied Microwave & Wireless*, pg 72, February 1999, plus correction pg 12, May 1999.
2. Kurzrok, Richard M, "Filter design uses image parameters", *EDN*, May 25, 2000, pg 111, May 25, 2000.
3. Kurzrok, Richard M, "Update the design of image-parameter filters", *Microwaves & RF*, pg 119, May 2000.

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

Figure 1



NOTES:
C_1, C_2, C_4, C_5=POLYPROPYLENE.
C_3: 470 pF=POLYPROPYLENE, AND 30 pF=CM-15 DIPPED MICA.
L_1, L_4=MICRO METALS T37-2 32T-#26.
L_2, L_3=MICRO METALS T37-2 14T-#26.
C_6, C_7, C_9, C_{10}: 470 pF=POLYPROPYLENE, AND 8 pF=CERAMIC.
C_8: 470 pF=POLYPROPYLENE, AND 75 pF=CM-15 DIPPED MICA.
L_5, L_8=MICRO METALS T25-6 10T-#30.
L_6, L_7=MICRO METALS T25-6 23T-#30.
ADAPTER=BNC M-M (U6-491A/U, COMMERCIAL).
I/O CONNECTORS=BNC FEMALE.
ENCLOSURE=ALUMINUM BOX (HAMMOND 1590A/BND CU-123).
PC BOARD=VECTOR BOARD 169P44C1, CUT BY HAND.
STANDOFFS=MALE, FEMALE (AMATOM 9794-SS-0440).

A low-cost wideband filter results from cascading a highpass and a lowpass filter using image-parameter designs.

TABLE 1–PARTS LIST FOR WIDEBAND FILTER

Frequency (MHz)	Highpass insertion loss (dB)	Lowpass insertion loss (dB)
2.9	23.8	Less than 0.1
3.3	9	Less than 0.1
3.4	2.5	–
3.5	1.15	–
4	0.65	0.1
5	0.3	0.1
7	0.15	0.2
9	0.1	0.55
9.5	–	1
10	–	5
11	–	23.5
12	0.1	–

TABLE 2–AMPLITUDE RESPONSE

Frequency (MHz)	Insertion loss (dB)
2.9	24.4
3.3	7.9
3.4	2.7
3.5	1.2
4	0.8
4.5	0.6
5	0.4
7	0.3
8	0.5
9	0.8
9.5	1.1
10	4.8
12	Greater than 28

design ideas

Edited by Bill Travis and Anne Watson Swager

Test batteries without a voltmeter

Nam Phan, Pasadena, CA

THE CIRCUIT IN **Figure 1** is an easy approach to testing batteries without exiting the voltmeter. The battery holders in sizes AAA, AA, C, and D make this tester so much faster than a voltmeter. You just put the battery into the holder and look at the circuit meter instead of getting the voltmeter out of the case, plugging in the probe, and turning on the meter. Holding the tips of the probes to the tips of the battery is clumsy.

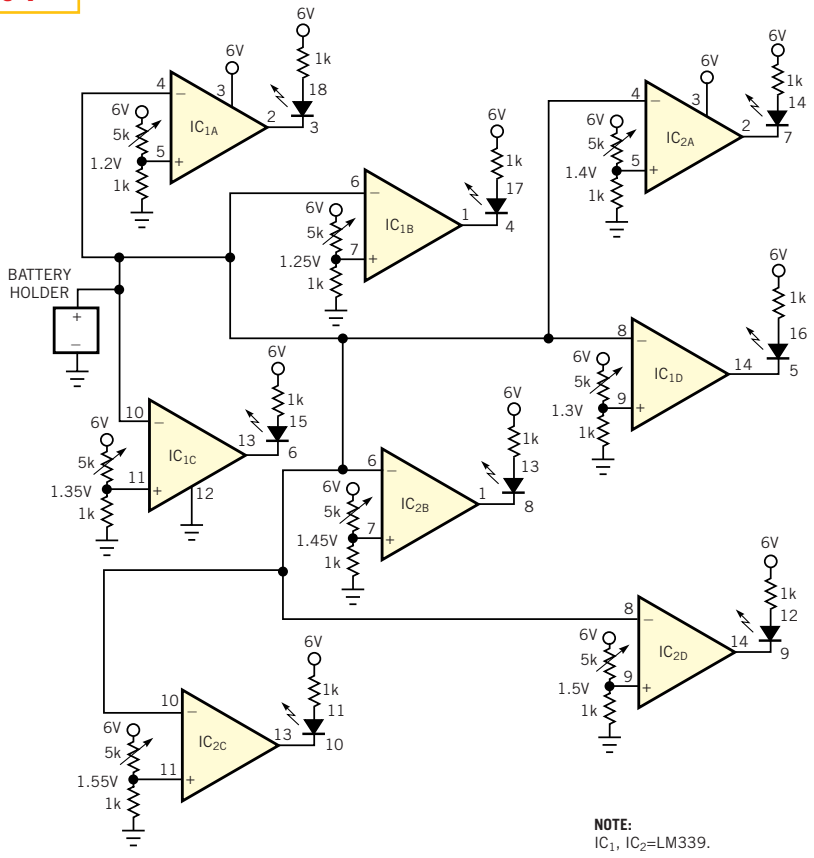
The heart of this circuit comprises op amps that the circuit configures as comparators. When the voltages at the plus (noninverting) inputs are higher than the voltages at the minus (inverting) inputs, the op-amp outputs are equal to V_{CC}. When the plus inputs are lower than the minus inputs, the outputs are equal to V_{DD}. Every plus input connects to a potentiometer that controls the voltages going into the plus-input pin. The minus inputs all connect to battery holders.

You adjust the potentiometers in increments of 0.05V starting at 1.2V and ending at 1.55V. You can change this adjustment to increments of 0.2 or 0.3V, depending on how accurate you want the tester to be.

The output of each op amp connects to a 20-pin LED bar, which you place vertically to look like a meter. The circuit uses only eight of the LEDs. If the battery voltage is higher than 1.4V, the bottom five LEDs will light up because the minus input is greater than the plus input on the bottom five op amps. The top three LEDs do not light up because 1.4 is not higher than 1.45, 1.5, or 1.55.

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

Figure 1



This battery tester configures two dual op amps as comparators.

Electronic fuse emulates fast- or slow-blow fuses

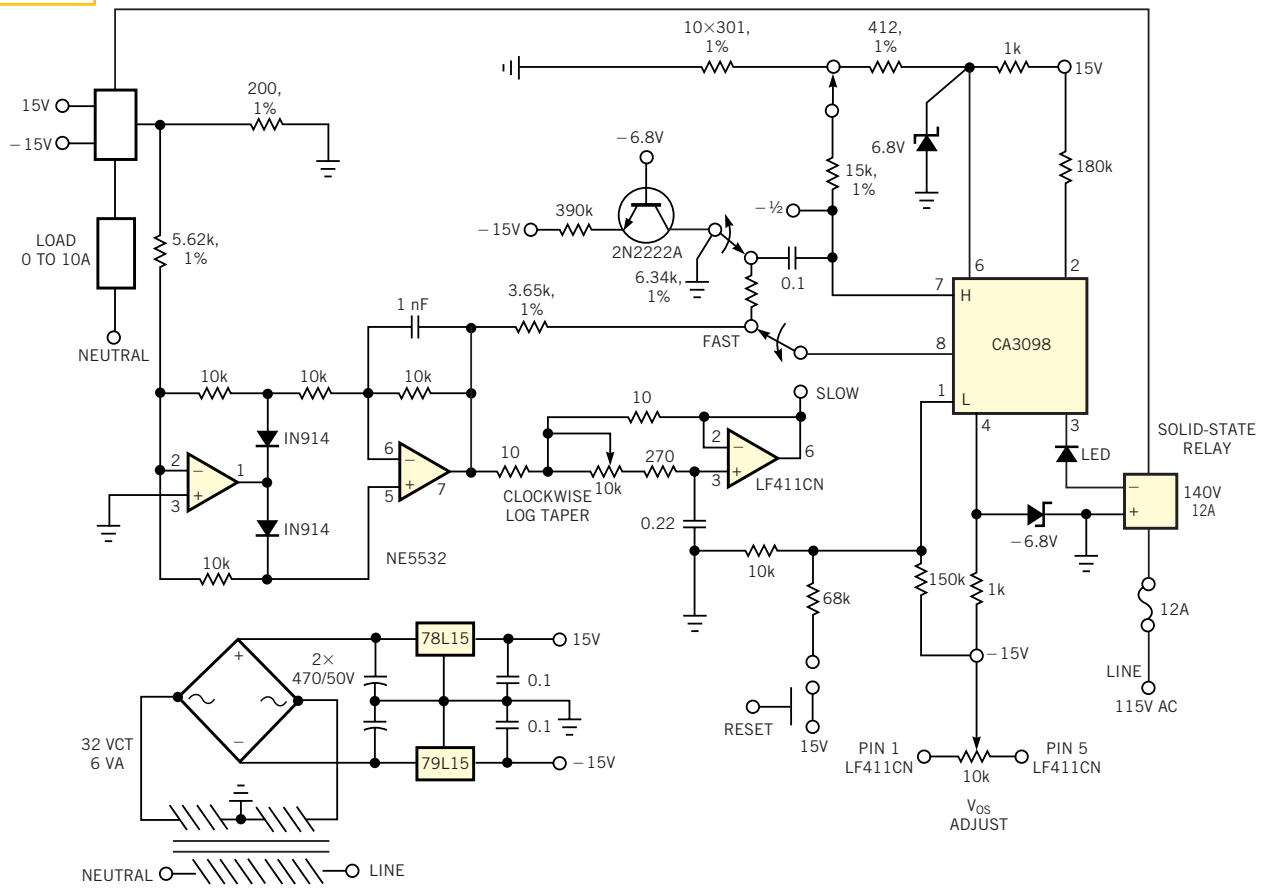
John A Hasse, Colorado State University, Fort Collins, CO

The electronic-fuse circuit in **Figure 1** combines the properties of a current transducer and a solid-state relay to disconnect low power at preset levels. Using this circuit lets you avoid the bother of stocking and replacing fusible links.

The circuit simulates fast- or slow-blow fuses as large as 10A in 1 or 2A increments using a convenient pushbutton reset. This device can bracket trip levels of functioning equipment or help locate chronic faults. The circuit full-wave-rectifies the output from the Lem current transducer and applies the result or with a variable delay to a window comparator. The reference steps are 600 mV/1A at Pin 7 as a high level. Signals greater than the H pin of the CA3098 set a flip-flop in the CA3098, which removes drive to the solid-state relay. Forcing Pin 1 of the CA3098 from −1V to 1V resets the flip-flop and restores load power. An offset current through the 15-kΩ, 1% resistor adds −300 mV to the set level, which is equivalent to −1 or −2A from the integral switch settings of 1 to 10 with standard 30° indexing. The circuit blocks two switch positions from use. You adjust the LF411CN with no ac load to zero voltage at Pin 6 relative to dc common.

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

Figure 1



An electronic fuse combines the properties of a current transducer and a solid-state relay.

Amplitude-stable oscillator has low distortion, low cost

Moshe Gerstenhaber, Chau Tran, and Mark Murphy, Analog Devices Inc, Wilmington, MA

THE MULTIVIBRATOR IS a common circuit that consists of an amplifier with both positive and negative feedback (**Figure 1a**). When the output is positive, the positive input terminal equals ½ V+, and the voltage at the negative input terminal changes toward V+. When this voltage exceeds ½ V+, the output voltage rapidly changes to V−. The positive input terminal becomes ½V−, and the negative input terminal changes toward V−. When the voltage at the negative input terminal is less than ½V−, the process repeats (**Figure 1b**).

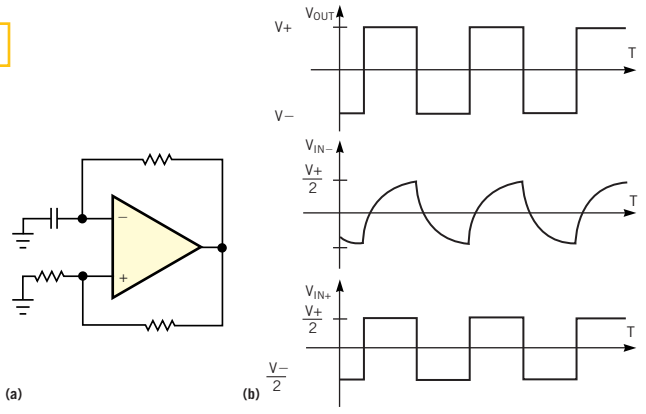
For the multivibrator to work, the bandwidth of the amplifier must be 10 times higher than the time constant of the passive network, and consideration of the high slew rate helps define the amplifier. The output is a square wave.

The circuit in **Figure 2a** is a sinusoidal oscillator. External compensation at Pin 5 forces the unity-gain bandwidth of the amplifier to be the same as the passive-network bandwidth.

Loop-gain analysis results in the following transfer function:

$$\frac{V_{OUT}}{V_{IN}} = -\frac{(1-2\pi f R_1 C_1 j)}{100\pi f \frac{1}{g_m} C_2 j(1+2\pi f R_1 C_1 j)}.$$

Figure 1



The common multivibrator has positive and negative feedback (a). When V_{OUT} is positive, V_{IN-} changes toward V+. When V_{IN-} exceeds V+/2, V_{OUT} changes to V− (b).

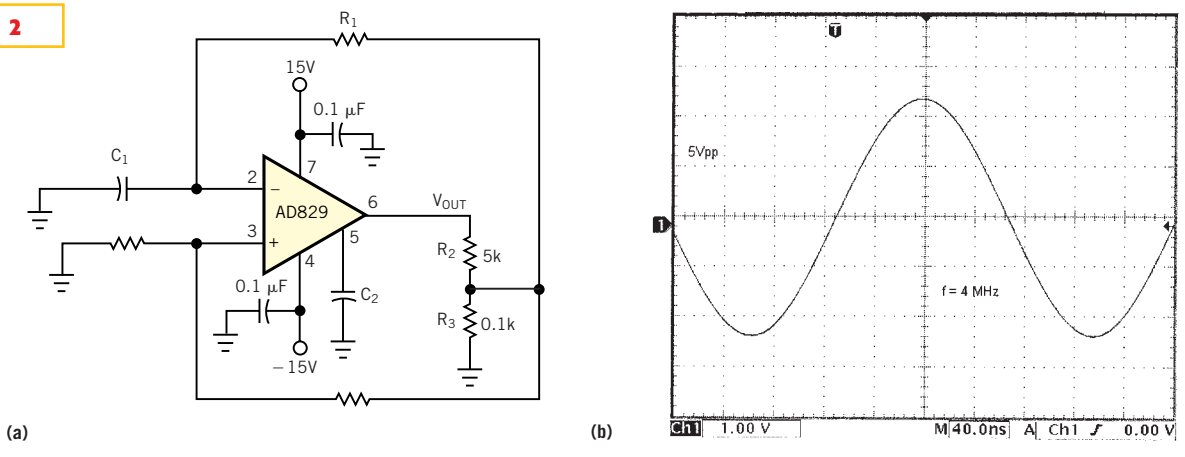
To meet the conditions necessary to sustain oscillation—loop gain equal to unity and phase equal to zero—choose $(1/g_m)\times C_2 = 1/(100\pi f)$ and $R_1\times C_1 = 1/(2\pi f)$.

The inverse transconductance, $1/g_m$, of the input stage, re, is equal to 52Ω. The design assures amplitude stability because re always increases with an amplitude increase, which reduces the loop gain. The ratio of the R_2/R_3 divider network sets the amplitude.

Figure 2b is a performance photo of the oscillator running at 4 MHz and 5V p-p. For better frequency stability, you can replace C_2 with a crystal of the desired frequency and low shunt capacitance.

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

Figure 2



This stable oscillator (a) has low distortion (b).

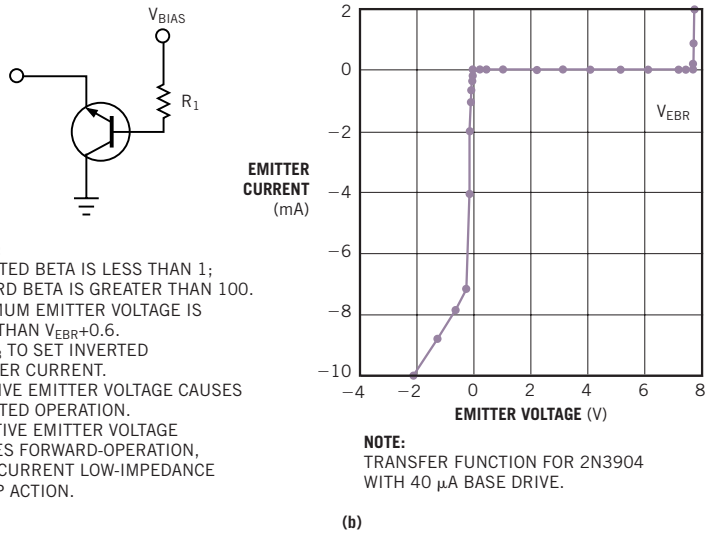
Inverted bipolar transistor doubles as a signal clamp

Art Hogrefe, Puma Instrumentation, State College, PA

A number of circuits, such as level detectors and AM demodulators, benefit from a rectifier with a low offset voltage. Silicon diodes have an offset of approximately 0.6V and do not work well in low-level circuitry. A Schottky diode is a bit better with an offset of approximately 0.4V. A few germanium diodes are still available, but they do not tolerate the temperature range of silicon. Also, you can't include a germanium diode in an IC. A superior configuration uses a bipolar transistor for these applications.

Figure 1 shows the bipolar-inverted-clamp circuit and a typical transfer function. The collector connects to ground or any other desired reference voltage. A fixed current drives the base. In the absence of any external drive, the emitter voltage is near zero. Driving the emitter with an external voltage produces the transfer function in **Figure 1**.

Figure 1

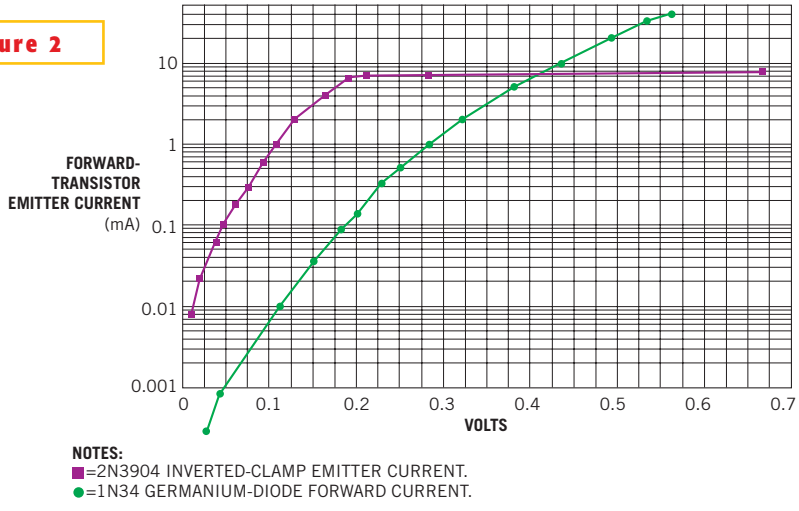


The bipolar inverted clamp (a) has an excellent rectification characteristic (b) because of the 2N3904's large forward-beta-to-reverse-beta ratio.

The circuit achieves this excellent rectification characteristic by using a transistor with a large forward-beta-to-reverse-beta ratio. Many of these transistors are still available. The 2N3904 provides excellent characteristics at a low cost. The reverse beta of the 2N3904 is only 0.25, so that for positive voltage on the emitter and, with 40 μA of base drive, the emitter current is around 10 μA. This current is sufficient in most level-detector applications for which the ac input amplitude changes slowly.

The emitter current at even small negative voltages is much greater than in the inverted region because the forward beta of the 2N3904 is greater than 100. Impedance is low up to the beta-limited forward current, at which point the impedance increases to approximately the value of R_1/beta. **Figure 2** shows the forward-transistor emitter current of the 2N3904 and the forward current of the 1N34 germanium point-contact diode. The logarithmic current scale shows the impressive response of the 2N3904 at small voltages.

Figure 2



A logarithmic scale of the 2N3904's forward-transistor emitter current and the forward current of the 1N34 shows the impressive response of the 2N3904 at small voltages.

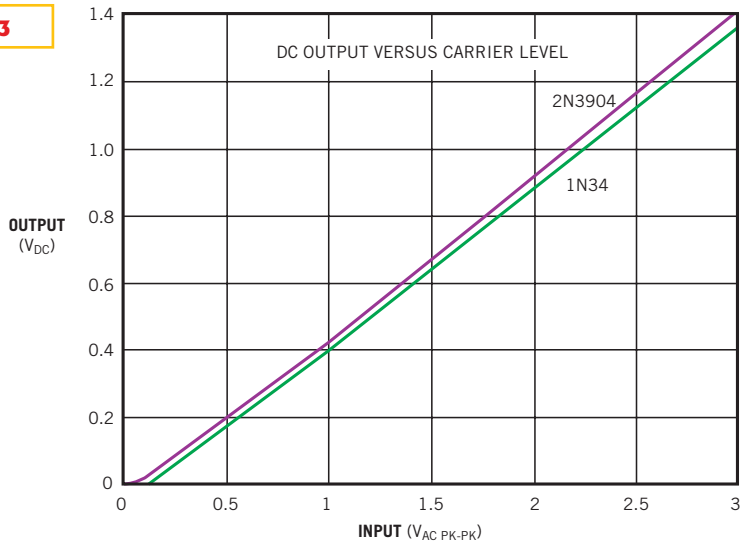
Figure 3 shows the output as a level detector for the two clamps. The transistor circuit that produced these results is similar to the demodulator in **Figure 4** except

the base drive is 40 µA. For the 1N34, the anode connects to grounded and the cathode connects to the input capacitor in place of the transistor's emitter. **Figure 3** shows that the two configurations have similar responses to input levels, and that the 2N3904 has a bit less offset, as you would expect from **Figure 2**. The output can drive a signal level meter or following electronics as part of an automatic-level-control or automatic-gain-control loop.

The transfer function in **Figure 1** also shows a sudden increase in inverted current at approximately 7.6V, which occurs at the reverse breakdown voltage for the emitter-to-base junction. Because you know in this case that the base is near 0.6V, the breakdown voltage for the tested part is near 7V. Production circuits would have an input limit of 6.6V p-p because of the minimum specified breakdown voltage of 6V. Note that, for a small production, such as for test equipment, it is practical to select individual transistors to slightly increase the dynamic range. A 6V p-p input dynamic range is sufficient in many applications.

The RF demodulator in **Figure 4** has a base drive current of 300 µA. This current is necessary to track the RF-modulation envelope and depends on the size of the input capacitor, modulation frequency, and maximum signal amplitude. The reverse current, which is I_{BASE} times the reverse beta, must be large enough to discharge the input capacitor at the highest modulation frequency and amplitude to prevent distortion in the output waveform. **Figure 5** shows the running demodulator with the upper trace at the emitter node and the lower trace at the output.

Figure 3

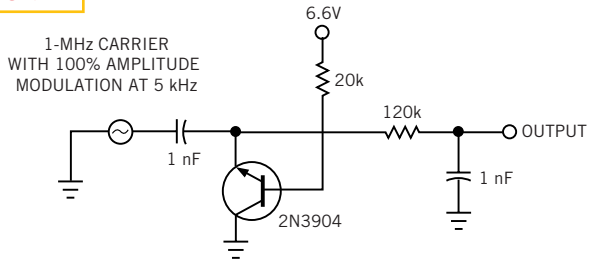


NOTES:
FOR 2N3904, $V_{DC}=0.5V_{AC}-0.08$, AND $I_{BASE}=41$ µA.
FOR 1N34, $V_{DC}=0.5V_{AC}-0.11$.

When operating as demodulators, the two configurations have similar input-level responses.

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

Figure 4



NOTES:
BASE-DRIVE CURRENT IS 300 µA, EMITTER REVERSE CURRENT IS APPROXIMATELY 75 µA.
2N3904 FORWARD BETA IS GREATER THAN 100; REVERSE BETA IS APPROXIMATELY 0.25.
EMITTER-BASE REVERSE-BREAKDOWN VOLTAGE IS GREATER THAN 6V.
AC-DRIVE VOLTAGE IS LESS THAN 6.6V P-P.

Using the inverted clamp as an AM demodulator requires a base current of 300 µA to track the RF-modulation envelope.

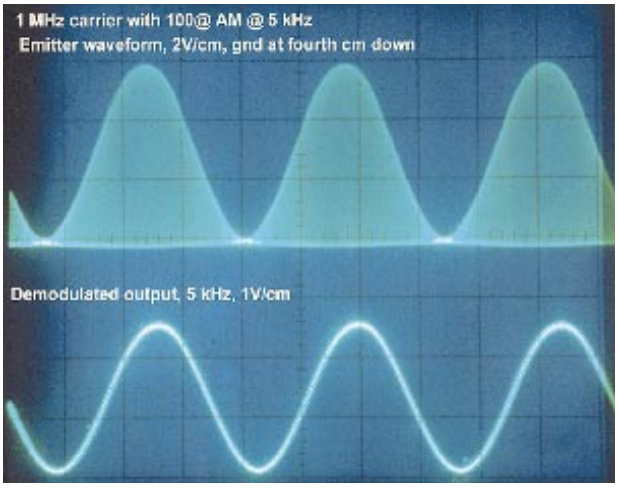


Figure 5 **A scope photo shows the running demodulator; the upper trace is the emitter node, and the lower trace is the output.**

Data-acquisition circuit measures almost everything

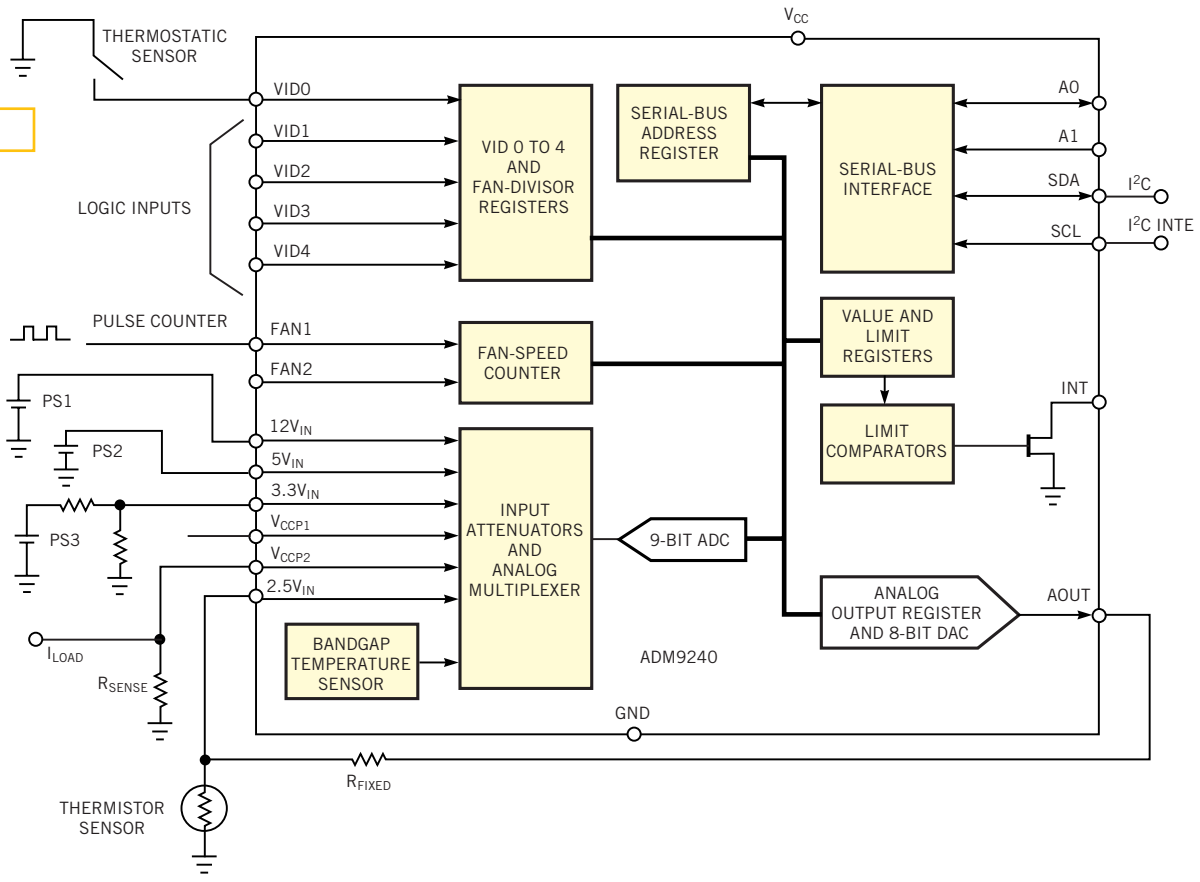
Matt Smith, Analog Devices, Limerick, Ireland

Using a product developed for PC-motherboard environmental monitoring, you can configure a low-cost, general-purpose DAS (data-acquisition system) (**Figure 1**). The DAS can directly monitor multiple voltage channels as well as temperature and frequency. It can also directly monitor digital sensors. Using only a few additional components, the system can accommodate other sensor and transducer elements. The flexibility exists to expand the scheme to cover additional input channels if necessary. For voltage sensing, the ADM9240 contains a multichannel ADC that can directly monitor as many as six input channels. The original intent of the ADC was to monitor power supplies on PC motherboards, but the converter is flexible enough for general-purpose use. The maximum input-voltage ranges for the channels are 3.3, 3.6, 4.4, 6.64, and 16V. **Figure 1** shows the system monitoring two power supplies: PS1 and PS2. The DAS can monitor voltages greater than the channel range by using a simple voltage divider at the front end, as illustrated with PS3.

The ADM9240's on-board DAC (originally intended as a fan-speed controller) can serve as a programmable, precision reference source. This function, for example, would facilitate measuring resistance-type sensors on the voltage-sensing channels. You could also use it as a bridge-excitation voltage source for accurate bridge-sensing elements. You can determine an unknown resistance value, such as a thermistor, by setting the DAC's output voltage to a known level with a known fixed resistance (**Figure 1**). You can implement current sensing by placing an accurate series resistor (R_{SENSE}) in the ground line and monitoring the voltage drop across the resistor. The DAS also provides temperature sensing by using an on-chip bandgap silicon sensor. The system can directly monitor temperatures over a −40° to +85°C range.

The DAS provides two frequency-monitoring channels. You can use them

Figure 1



A data-acquisition IC originally intended for PC motherboards can monitor a multitude of parameters.

to monitor pulsed digital output from a tachometer or as general-purpose frequency counters. Five digital-input lines were originally intended to monitor digital voltage-identification lines. You can use them for general-purpose input lines, whose inputs can sense high- or low-level status signals from digital sensors or from alarm channels. In **Figure 1**, the DAS monitors a thermostatic sensor. The DAS handles control and reading functions via a simple two-wire SMBus or an I^2C interface to a µP or µC. If a dedicated I^2C controller is not available, then you can use a port "bit-banging" technique. Easy expansion is also possible by selecting a different device address. Using a different device-address bus entails no additional communication lines, because multiple devices can reside on the same bus.

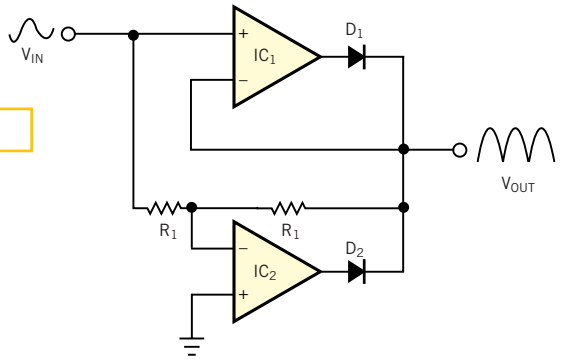
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Full-wave rectifier has programmable gain

Chuck and Chris Wojslaw, Xicor Inc, Milpitas, CA

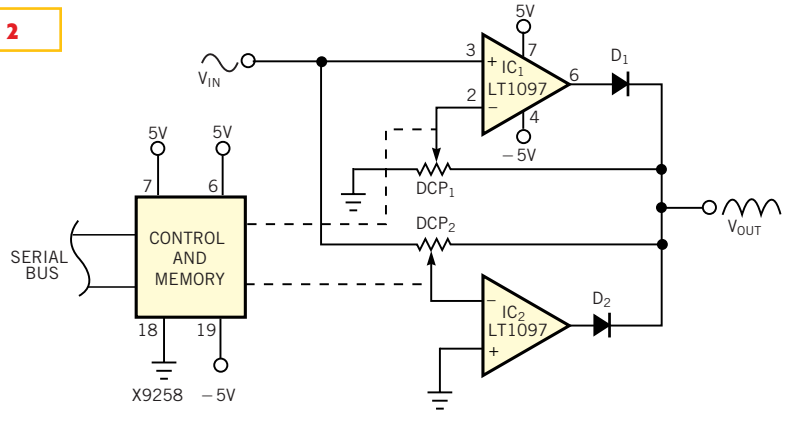
THE TRADITIONAL approach to the design of a full-wave rectifier (**Figure 1**) is to set the gains of IC_1 and IC_2 to 1 and use the steering diodes D_1 and D_2 to sum the sinusoidal half-cycles of the input to form the rectified output. For the positive half-cycle of the input signal, IC_1 is a noninverting amplifier with a gain of 1. For the negative half-cycle of the input signal, IC_2 is an inverting amplifier with a gain of −1. This full-wave-rectifier circuit often combines with a lowpass filter to form a low-cost ac/dc converter. If you need a full-wave rectifier with amplification, the combination of these two functions in one circuit can provide savings in cost, component count, and board space. The circuit in **Figure 2** combines rectification and programmable amplification. The two 256-tap Xicor digitally controlled potentiometers, DCP_1 and DCP_2, control the gains of the noninverting amplifer, IC_1, and the inverting amplifier, IC_2, respectively. The gain of IC_1 is ${}_1G_1=255/P1$, where P1 ($0 \le P1 \le 255$) is the programmed decimal value of DCP_1. Similarly, the gain of IC_2 is $G=-(255-P2)/P2$. The performance of this circuit takes advantage of the resistor matching inherent in the digital potentiometers. The measured data falls within 2% of calculated values.

Figure 1



This full-wave-rectifier circuit has a fixed gain of 1.

Figure 2



You can program the gain of this full-wave rectifier from unity to 255.

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

Method synchronizes slaves in power-line communications

Jose Sebastia and Diego Munoz, University of Valencia, Spain

In PLC (power-line-communication) applications, the communications system usually uses one master and a large number of slaves (for example, 64). The idea presented here is an easy method for the synchronization of slaves, using one µC and a few other components. The µC is a PIC16C7X, which has three important properties for this application: a watchdog timer, an external interrupt, and a sleep instruction. The watchdog timer is a free-running, on-chip RC oscillator that requires no external components. The watchdog timer continues running even if the µC clock stops in the event of a sleep instruction. During normal operation, a time-out from the watchdog timer generates a device reset. If the device is in sleep mode, a watchdog timer time-out causes the device to wake up and continue normal operation.

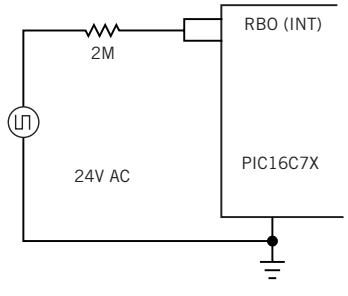
In PLC applications, the ac power-line voltage is 24V ac (for outdoor applications) or 220V ac (for household applications) at 50 or 60 Hz. With a period of 20 msec, the power-line voltage has a zero crossing every 10 msec. These zero crossings serve as a timer for the slaves. Each zero crossing activates the µC's interrupt when the slave is sleeping and wakes up and updates the timer/counter. In **Figure 1**, a single resistor connects the 24V-ac line to the µC. Each slave has a counter/timer, and all counter/timers count simultaneously. To synchronize the slaves, this method uses the watchdog timer, which has a normal time-out period of 18 msec. If this time elapses without activation of the interrupt, the µC wakes up and starts the counter/timers of all the slaves. At this moment, the slaves are synchronous with the master. When the output of pin RC0 of the µC in **Figure 2** is at 0V, the pin draws current, I_{OUT}. This current activates the optically coupled triac, enabling a 24V-ac power line for the slaves. When the RC0 output is high, the result is a short circuit in the ac power line. At this point, the slaves begin the synchronization subroutine (**Listing 1**). You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2602.

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

LISTING 1—SYNCHRONIZATION SUBROUTINE

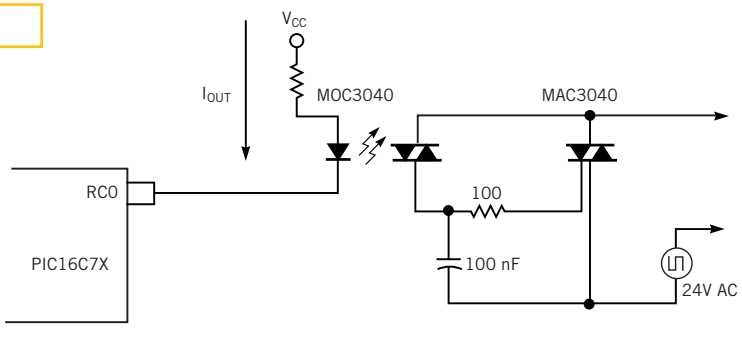
```
BUCLE
                clrw                    ; clear the W register
                SLEEP                   ; the microcontroller is sleeping
                clrwdt                  ; clear the watch-dog register
                btfsc   WAKEUP,5        ; the bit 5 of register WAKEUP show that the
                                        ; interruption is of synchronisation if is 1. But
                                        ; if is 0 the wake up of micro is due at WDG, and
                                        ; clear the TIMER1.
                 goto   por_RB0         ;
                goto    wdg_sincro      ;
wdg_sincro
                clrf    TMR1            ; clear the TMR1 register
                bsf     flag,SINCRO     ; the bit flag in the SINCRO register show what the
                                        ; synchronisation are make.
                goto    BUCLE;
por_RB0
                bcf     WAKEUP,5        ; clear the bit of show the synchronisation
                incf    TMR1            ; This is the internal counter for each slave
                goto    BUCLE;
; Subroutine of interruption
__INT
                bsf     WAKEUP,5        ; if the micro wakes up due to the interruption,
                                        ; the bit 5 of WAKEUP register is 1 for indicate
                                        ; this.
                RETFIE
```

Figure 1



The ac power line can directly drive the PIC16C7X interrupt pin.

Figure 2



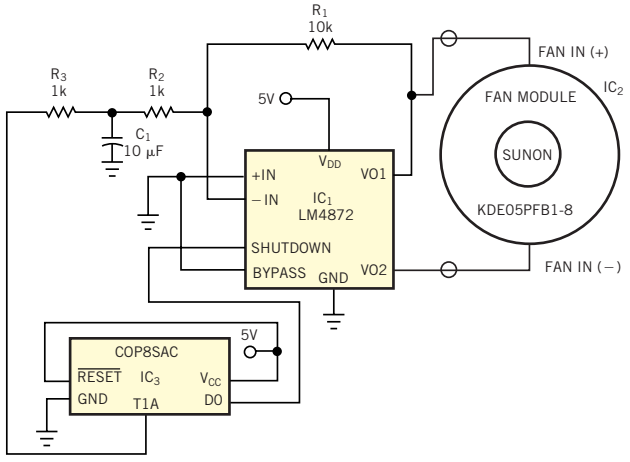
An optically coupled triac controls power for the slaves.

Audio amp makes efficient fan controller

Wallace Ly, National Semiconductor Corp, Santa Clara, CA

You can use discrete transistors to vary power to a fan to control its speed. However, with a simple modification, you can use an audio-amplifier IC to control a fan module (**Figure 1**). The LM4872 "Boomer" is an audio amplifier capable of delivering 1W maximum output power. A COP8SAC µC connects to the audio amplifier and the fan module. The µC's T1A output pin delivers a PWM signal to an RC network, which produces a dc signal. The dc output of the RC network is proportional to the duty cycle of the PWM signal. The dc signal drives the audio amplifier, which powers the fan via its bridge-configuration outputs. **Listing 1** demonstrates how to control the PWM signal from the µC. For standby operation, set the D0 bit; this operation puts the LM4872 in shutdown mode. We also recommend that you put the µC in shutdown mode. In shutdown mode, the total quiescent current is approximately 4 µA, so the circuit in **Figure 1** is ideal for battery-powered applications. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2608.

Figure 1



An inexpensive µC and audio-amplifier IC form an efficient fan controller.

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

Want to submit your own Design Idea? Turn the page for an entry blank.

LISTING 1—FAN-DRIVER C FILE

```
#include "8saa.h"                       // Include file for the COP8SLB
void init_params();                     // Initializtion routine
void shutdown(unsigned int state);      // Shutdown amplifier routine
void fan_speed(unsigned int state);     // The Speed of the fan
                                        // Alters the PWM duty cycle
                                        // In percent of (on) vs (off)
void main(){

        init_params(); // initialize the parameters

        // Make sure the amplifier is on
        shutdown(0);

        // Scale up the fan speed

        // Turn up the fan speed
        fan_speed(20);

        // Scale down the fan

        //fan_speed(10);


        // Shutdown the amplifier (off)
        shutdown(1);

        while(1);  // forever loop

}

// The shutdown routine

void shutdown(unsigned int state){

if (state==0)

   PORTDD.0=0;     // Turn the pin low
   else
      PORTDD.0=1;  // else turn the pin high

}
```

```
// The intialization parameters

void init_params(){

        // Turn PWM on out
        PORTGC.3=1;   // Set so the portgc pin is high
        PORTGD.3=1;   // Lower the portgd pin for PWM mode

        // Make the high byte zero

        T1RAHI=0;
        T1RBHI=0;

        // Set to PWM: TxA Toggle
        // Autoreload RA, RB

        CNTRL.T1C1=1;
        CNTRL.T1C2=0;
        CNTRL.T1C3=1;

}


// The change fan speed routine

void fan_speed (unsigned int state){


        // Detect if it is less or equal to 100 percent
        if (state<101){

           T1RALO=state;       // Initialize the state
           T1RBLO=state-100;   // Turn the lower half state-100
           CNTRL.T1C0=1;       // Set the timer enabled bit

          }

}
```

design ideas

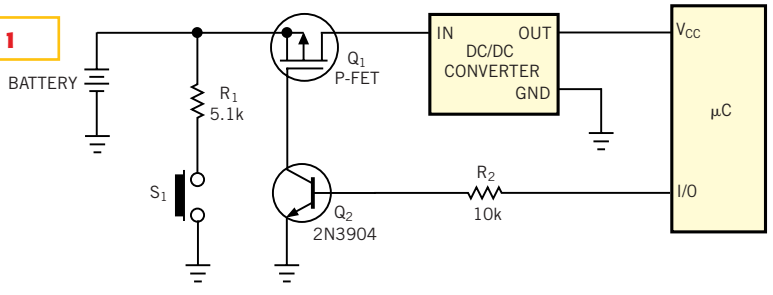
Edited by Bill Travis and Anne Watson Swager

Transistor latch improves on/off circuitry

Eugene Palatnik, SIMS-BCI, Waukesha, WI

Figure 1 shows an example of on/off circuitry commonly used in battery-operated devices. The p-channel MOSFET, Q_1, serves as a power switch. When you push the On button, S_1, Q_1's gate goes low. Q_1 turns on and supplies battery voltage to the dc/dc converter. Depending on the battery voltage in the device, the dc/dc converter might convert the voltage either up or down. In either case, it supplies V_{CC} to the µC. The µC goes through its power-up software sequence and programs one of its general-purpose I/O pins, setting it to logic one. This operation, in turn, causes saturation of the npn transistor, Q_2, which "confirms" the power-up state. Later, when the µC decides to power itself off, the µC simply sets its I/O output to logic zero, and Q_1 returns to its off state. The circuit is simple and reliable but has a significant disadvantage. It usually takes a fraction of a second for the dc/dc converter to reach its stable output voltage. Then, the µC's Reset pulse usually lasts 50 to 200 msec. After the release of Reset, the µC must go through its "housekeeping" start-up code before it has a chance to set its I/O pin to logic one. This delay in some portable systems may be user-unfriendly, because if you don't depress the On button long enough, the system will not power up. The circuit in **Figure 2** eliminates this uncertainty.

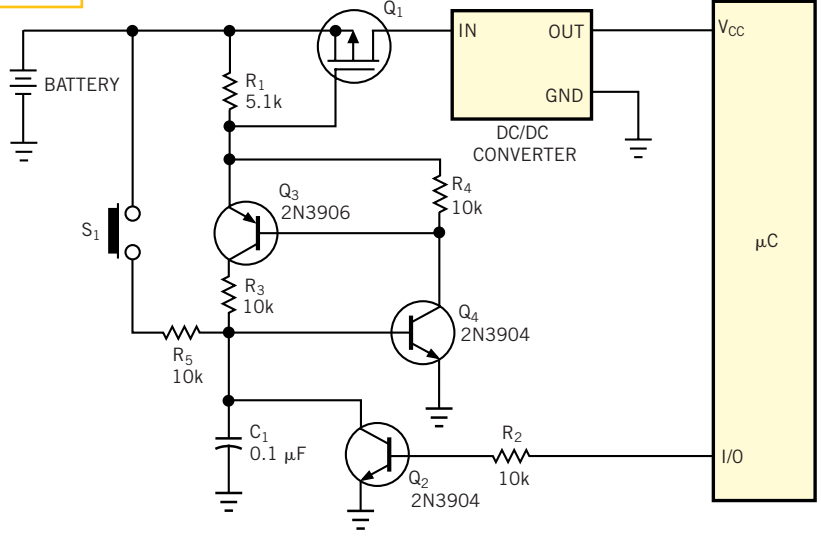
Figure 1



This on/off circuitry is effective but can suffer from turn-on ambiguity.

The circuit includes a simple two-transistor latch, which the On button switches to the on state. As in **Figure 1**, the p-channel MOSFET, Q_1, serves as a power switch. When you push the On button, S_1, it causes saturation of the npn transistor, Q_4, via the base-current-limiting resistor, R_5. The collector current of Q_4 flows through R_1 and the base-emitter junction of pnp transistor Q_3, thereby saturating Q_3. Q_3 redirects some current into the

Figure 2



A two-transistor latch provides positive turn-on when you push the button.

base-emitter junction of Q_4 and finishes the latching process. At this point, both Q_3 and Q_4 are saturated, and the voltage on the gate of Q_1 is a function of the voltage drop across the base-emitter junction of Q_3 and the saturation voltage of Q_4. This voltage is approximately 0.9V. The µC need not confirm the on state of the latch. When the µC powers up and finishes its housekeeping start-up code, it programs the I/O pin to logic zero.

Later, when the µC decides to power itself off, it programs the I/O pin to logic one and stops. Q_2 turns off Q_4, resetting the latch to its initial off state. R_4 lowers the equivalent input impedance of Q_3. This function improves EMI and ESD noise immunity and prevents the circuit from turning itself on in the presence of strong electromagnetic fields. Capacitor C_1 in combination with R_5 protects Q_4 and Q_2 from direct ESD into the pushbutton. Some portable devices use undervoltage-lockout circuitry. This circuitry usually uses a voltage comparator with a built-in voltage reference. If the battery voltage drops below the threshold, the output of the comparator (usually an open-drain type) switches low. If your portable system uses this type of circuitry, you can connect the open-drain output of the comparator in parallel with Q_2, thus preventing the latch from turning on if the battery voltage is too low.

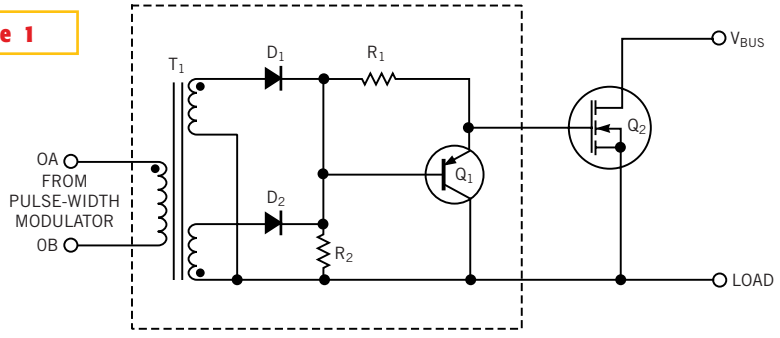
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

High-side driver feeds IGBTs and MOSFETs

Carlisle Dolland, Honeywell Engines and Systems, Torrance, CA

THE LOW- TO MODERATE-POWER system in **Figure 1** provides the interface between a pulse-width modulator and a high-side IGBT (insulated-gate-bipolar-transistor) or MOSFET switch. You can use it to interface TTL or CMOS circuitry to an H-bridge if you buffer it by a FET driver, such as an ICL7667 or an MIC4423. When OA is positive, D_1 conducts, charging the capacitance of the FET through R_1. The value of R_1 and the output impedance associated with the drive signal determine the turn-on time of the FET. After the capacitance is charged, the voltage across R_1 is essentially 0V, and Q_1 is off. During the PWM dead time, the gate capacitance discharges through Q_1 and R_2. R_2 and the h_{fe} of Q_1 determine the turn-off time of the FET. This circuit achieves turn-on and -off times of less than 150 nsec.

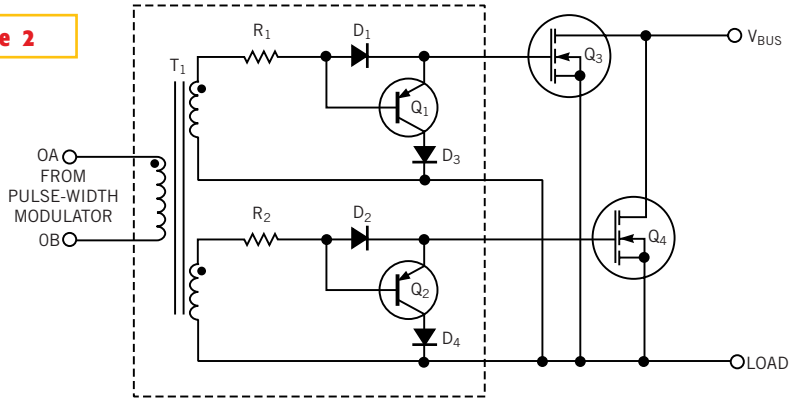
Figure 1



You can use this low-cost circuit to drive high-side FETs or IGBTs.

In systems that require higher power, you can use a dual-FET circuit (**Figure 2**). R_1 and R_2 and the output impedance of the PWM or FET driver determine the turn-on time. R_1 and R_2 and the h_{fe} values for Q_1 and Q_2 determine the turn-off time. The Schottky diodes, D_3 and D_4, prevent current flow through the collector-base junctions of Q_1 and Q_2 when the drive signal is negative. You can obtain switching speeds higher than 50 nsec at turn-on and 100 nsec at turn-off with this circuit, depending on the output impedance of the transformer drive signal.

Figure 2



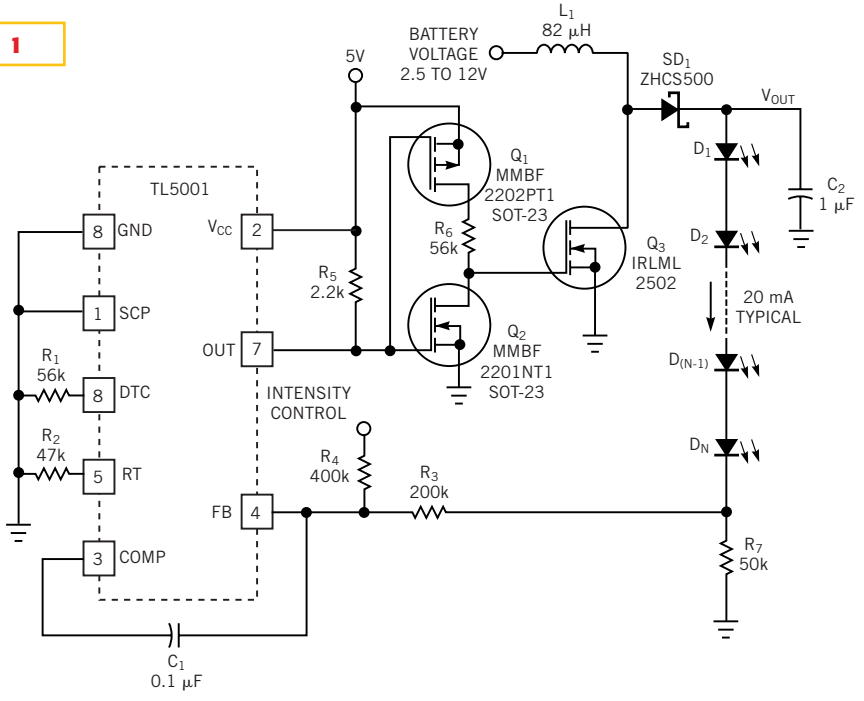
Two FETs in parallel provide more power than the circuit in Figure 1.

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

Economical circuit drives white LEDs

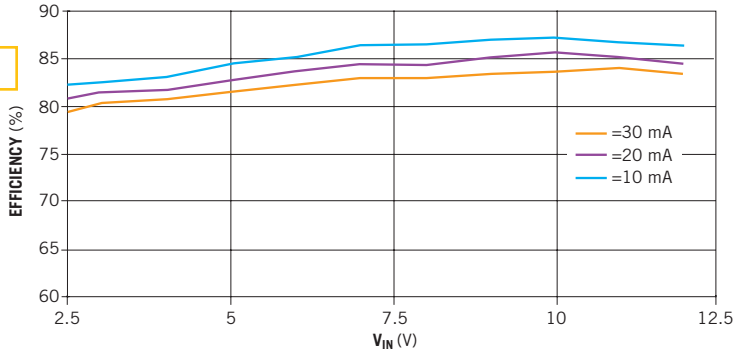
Eddy Wells, Texas Instruments

Figure 1



White LEDs provide ideal backlighting for small LCDs.

Figure 2



The circuit in Figure 1 provides more than 80% conversion efficiency.

Newly available white LEDs are replacing CCFLs (cold-cathode fluorescent lamps) in handheld applications using a backlit LCD. These applications include PDAs (personal digital assistants), digital cameras, and cellular telephones, to name a few. Advantages of white LEDs over CCFLs include longer life, higher efficiency, and significantly lower operating voltages. Regulating the current in the LED (typically 10 to 30 mA) controls the brightness; the forward voltage in each LED is approximately 3V. The circuit in **Figure 1** provides a means of efficiently controlling LED current in a series-connected string. The TL5001 PWM-controller IC is an older, industry-standard, inexpensive driver. The boost topology of the circuit allows operation from a single or dual lithium-ion cell. The ratings of Q_3, SD_1, and the maximum allowed duty cycle of the IC (programmed with pin DTC) determine the maximum output voltage of the circuit. V_{CC} comes from a separate 5V supply.

Trade-offs in the selection of inductor L_1 include size, dc resistance, and inductance value. An 82-µH inductor (with 200-mΩ dc resistance) results in continuous inductor current at higher LED currents, but the current becomes discontinuous at lower levels. The RT pin programs the oscillation frequency at approximately 200 kHz. Because the TL5001 has a relatively weak (20-mA) open-collector output driver and is intended to drive a buck-topology circuit, the circuit uses a low-cost inverter stage comprising Q_1 and Q_2 to efficiently drive Q_3. R_6 provides controlled turn-on and fast turn-off for Q_3. The reverse-breakdown voltage of SD_1 must be greater than the C_2-filtered V_{OUT}.

R_7 senses current in the white-LED string; the error amplifier at the FB pin of the IC controls the feedback signal at this pin to 1V. You can control the LEDs' intensity by summing in a control voltage via R_4. **Figure 2** shows the efficiency of a four-LED string. You could obtain approximately 2% higher efficiency by adding a gain-of-5 op-amp stage between R_7 and R_3, resulting in a lower voltage drop across R_7. Of course, this slight efficiency improvement adds to the system cost. For higher power applications, such as notebook computers, you can attach additional LED strings to V_{OUT}. To maintain uniform intensity in each string, you should add a dummy resistor of the same value as R_7 to each string.

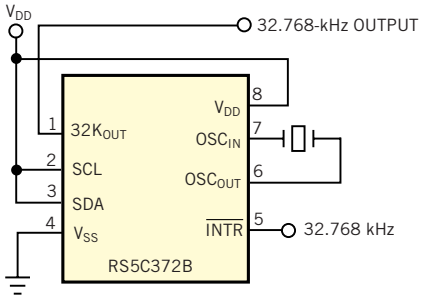
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Real-time-clock chip makes low-power oscillator

Yongping Xia, Teldata Inc, Los Angeles, CA

Many systems use watch-crystal-based, 32.768-kHz oscillators. In battery-powered designs, the 32-kHz oscillator may consume a fairly high percentage of the total power budget. Reduced power consumption equates to longer battery life, smaller batteries, and smaller products. Ricoh (www.ricoh.com) manufactures more than 10 types of real-time-clock chips, including the RS5C372B (**Figure 1**). This device is an eight-pin IC with a built-in oscillator, programmable periodic interrupts, and an I^2C interface to a µC. The only function the device in **Figure 1** uses is the 32-kHz oscillator. Using only the IC and the crystal, the circuit consumes low current over its 1.5 to 6V power-supply range, as the table in **Figure 1** shows. The CMOS-based output delivers a waveform with an amplitude of 0V to V_{DD}.

Figure 1



V_{DD} (V)	CURRENT (µA)
1.5	0.9
2	1.1
2.5	1.3
3	1.6
3.5	2.1
4	2.6
4.5	3.2
5	3.9
5.5	4.7
6	5.6

This ultra-low-power oscillator uses only the oscillator portion of a real-time-clock chip.

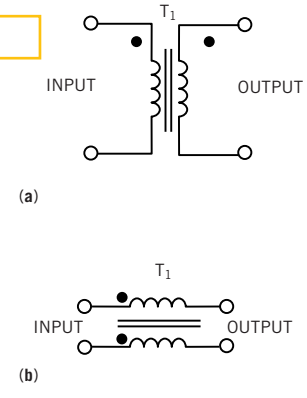
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Everything you wanted to know about wideband, low-frequency transformers

Richard Kurzrok, Queens Village, NY

Wideband, low-frequency transformers are useful components in various passive circuits, such as the return-loss bridge (**Reference 1**, **Figure 2**). **Figure 1a** shows a conventional transformer. If you connect the transformer windings differently, you can configure a transmission-line transformer (**Reference 2**, **Figure 1b**). These transformers use a magnetic core of modest size, and unit cost is reasonable. The basic transformer uses a Fair Rite (www.fair-rite.com) toroid (type 597700601), which has a nominal outer diameter of 0.825 in., a nominal inner diameter of 0.525 in., and a nominal thickness of 0.25 in. The toroid uses number 77 material and has an inductance factor (AL) of 1175. To obtain useful performance at audio frequencies, the transformer uses a 129-turn bifilar winding of number 26 magnet wire. To avoid the use of expensive commercial bifilar wire, you can twist together monofilar red and green windings using a hand drill before winding it on the toroid.

Figure 1



You can use a wideband transformer in conventional mode (a) or in a transmission-line configuration (b).

Table 1 gives the measured performance with 50Ω source and load impedances of the conventional transformer in **Figure 1a**. **Figure 1b** shows the schematic diagram of a one-to-one transmission-line transformer. **Table 2** gives the measured performance with 50Ω source and load impedances. This transformer provides bandwidth enhancement with useful behavior down to dc. You can use the conventional transformer in **Figure 1a** in a passive return-loss bridge (**Figure 2**) or for stand-alone dc isolation. **Table 3** gives the measured performance of the 50Ω

return-loss bridge. At 1 MHz, the return-loss bridge exhibits a forward insertion loss of 12 dB and an open-circuit-to-short-circuit ratio of 0.5 dB. You can use the conventional transformer to isolate a grounded signal from a balanced test piece. We built the circuits for the wideband transformers and return-loss bridge using single-clad vector board and enclosed them in die-cast aluminum boxes with BNC connectors.

References

1. Wetherhold, E, "Design and Construction of a 9-kHz Highpass Filter and Assembly of a Return Loss Bridge for Filter and PLISN," *Interference Technology Engineers' Manual*, pg 220, 1993.
2. Sevick, J "Transmission Line Transformers," American Radio Relay League, 1990.

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

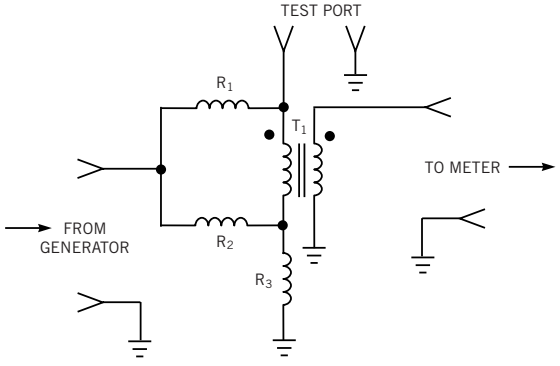
TABLE 1–PERFORMANCE OF A CONVENTIONAL TRANSFORMER

Frequency (MHz)	Insertion loss (dB)	Frequency (kHz)	Insertion loss (dB)
10	7.2	50	<0.1
8	3.8	20	<0.1
7	2.6	10	<0.1
6	1.9	5	<0.1
4	1	2	<0.1
2	0.5	1	0.15
1	0.3	0.5	0.5
0.5	0.15	0.2	1.7
0.2	<0.1	0.15	2.6
0.1	<0.1	0.1	4.1

TABLE 2–PERFORMANCE OF A TRANSMISSION-LINE TRANSFORMER

Frequency (MHz)	Insertion loss (dB)	Frequency (kHz)	Insertion loss (dB)
10	1.7	100	0.15
7	1.2	50	0.15
4	0.9	20	0.1
2	0.6	10	0.1
1.5	0.5	5	0.1
1	0.4	1	0.1
0.5	0.3	0.1	<0.1
0.2	0.2	0.01	<0.1

Figure 2



A return-loss bridge is one application for wideband, low-frequency transformers.

TABLE 3–DIRECTIVITY OF THE RETURN-LOSS BRIDGE

Frequency (kHz)	Directivity (dB)
1000	22
500	27
200	33.5
50	38
20	41.5
10	44.5
5	>46
2	>46
1	>46
0.5	>46

µC makes effective frequency counter

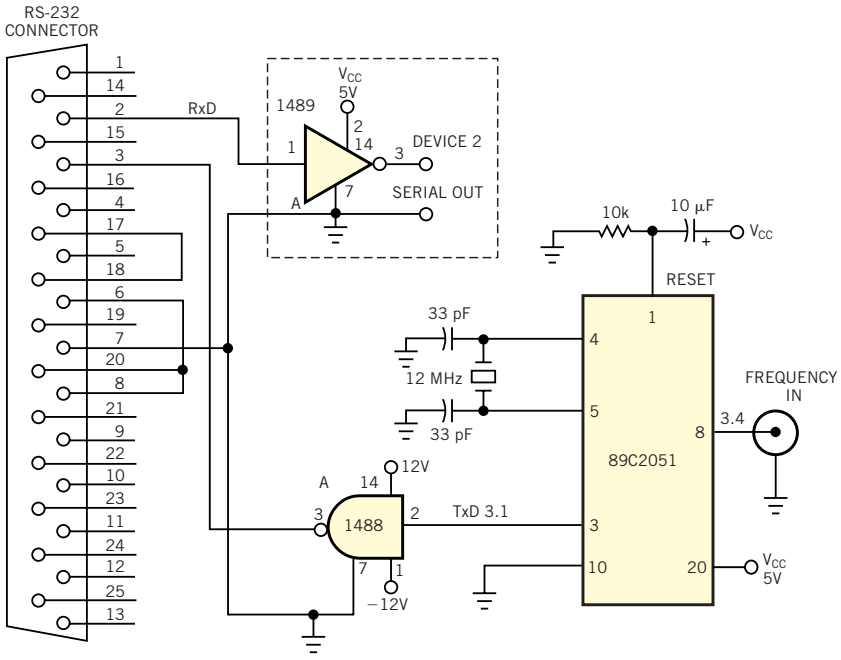
Fazal Pathan, Physical Research Laboratory, Ahmedabad, India

Figure 1 shows an efficient and cost-effective frequency counter using an Atmel 89C2051 µC (**Reference 1**). The design can use any µC of the 8051 family. The circuit counts frequency and sends the count to a PC via the serial port. The signal connects to pin 3.4 of the µC. The TTL-compatible output of the µC drives the 1488, which converts the output to RS-232 voltage levels. The output of the 1488 connects to the RxD pin on the serial port of the PC. In this design, only unidirectional communication exists between the µC and the PC. But you can use the TxD line to control another device. The assembly routine in **Listing 1** shows initialization of the timer, counter, and interrupts. The main program sets up T0 to count external pulses and T1 to count time in the autoreload mode 2 (**Reference 2**). The main program monitors the main-program flag, frqflg, until the flag is set. Then, the main program sends the counted frequency to the PC's port using mode 1. First, it sends the special character "L" to recognize the following 3 bytes as valid data. Then it sends the value of register R2, which is the most significant byte of the counter value. This value increments every time the counter

overflows. Then, the main program sends the values of registers TH0 and TL0 to the PC's port.

After sending the data, the program again jumps to the main routine, in which it clears the timer and counter and reinitializes them, and then starts counting again. The µC counts the value in hexadecimal format; the PC can then convert it to decimal format. In this example, we used a 12-MHz crystal and a baud rate of 2400 bps. However, you can use higher clock frequencies and baud rates, because the µC can operate to 24 MHz. We use the system in **Figure 1** for photon counting in astronomical instrumentation. We give thanks to the Department of Space, Government of India, for material support and BG Anandarao, PhD, for encouragement. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2616.

Figure 1



An 8051-family µC makes an efficient and cost-effective frequency counter.

References

1. Data Sheet for 89C2051, www.atmel.com.

2. Ayala, Kenneth J, *8051 Microcontroller: Architecture, Programming and Applications* (ISBN 0-314-77278-2), West Publishing Co, St Paul, MN.

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

LISTING 1—FREQUENCY-COUNTER ROUTINE FOR 89C2051

```
; Program counts the pulses and transmits through p3.1 pin only without
handshaking signal.

        .Equ Frqflg,0fh                 ;use a bit flag to signal main program
        .Equ Baudnum,0f3h               ;number loaded in TH1 for 2403.8 baud
Freq:
        .Org 0000h
         Mov SP, #30h                   ;set stack above register/bit area
         Sjmp over                      ;jump over T1 interrupt address

        .Org 000bh
         Inc r2
         Reti

        .Org 001bh                      ;T1 overflow flag interrupt to here
         Setb psw.3                     ;swich to register bank1
         Inc r0                         ;count R0 up until overflow at 00h
         Cjne r0, #00h,Checktime        ;check to see if time is up
         Inc r1                         ;or inc R1 when R0 rolls over
Checktime:
        Cjne r1, #27h, Goback           ;check R1 for terminal count
        Cjne r0, #10h, Goback            ;check R0 for terminal count
        Clr tr0                         ;stop T0
        Clr tr1                         ;stop T1 time before T0 stopped =
        Setb Frqflg                     ;signal main program that T0 =freq.
Goback:
        Clr psw.3                       ;return to bank 0 registers
        Reti                            ;return to main program

Over:
        Mov tcon, #00h                  ;all timers stopped - flags reset
        Set psw.3                       ;select register bank 1 and reset R0, R1
        Mov r0, #00h
        Mov r1, #00h
        Clr psw.3                       ;return to bank 0
        Mov tmod, #25h                  ;T1 a mode 2 timer, T0 mode 1 counter
        Mov tl1, #9ch                   ;start TL1 at 9ch
        Mov th1, #9ch                   ;TH1 = 156d, overflows in 100 clocks
        Mov tl0, #00h                   ;zero T0
        Mov th0, #00h
        Clr Frqflg                      ;reset the freq. measured flag
        Mov tcon, #50h                  ;start timer T1 and counter T0
        Mov ie, #88h                    ;enable global and T1 overflow interrupts
Simulate:
        Jbc Frqflg, Getfrq              ;have main program test "frqflg"
        Sjmp Simulate                   ;loop here until freq.is measured
Getfrq:
        Anl pcon, #7fh                  ;set SMOD bit to 0 for Baud * 32 rate
        Anl tmod, #30h                  ;alter timer T1 configuration only
        Orl tmod, #20h                  ;set timer T1 as an 8 bit autoload
        Mov th1, #Baudnum
        Setb tr1                        ;run T1
        Mov scon, #40h
        Mov sbuf, #'L'
Wait:
        Jbc ti, Next
        Sjmp Wait
Next:
        Mov sbuf, r2                    ;set UART to mode1
Wait1:
        Jbc ti, Next1
        Sjmp Wait1
Next1:
        Mov sbuf, th0                   ;transmit content of the th0
Wait2:
        Jbc ti, Next2                   ;wait for T1set before next transmission
        Sjmp Wait2                      ;else poll flag again
Next2:
        Mov sbuf, tl0                   ;now transmit tl0
                                        ;and wait until the ti flag is set
Wait3:
        Jbc ti, Over
        Sjmp Wait3

       . End
```

Software provides interrupt system for 8051

Deng Yong, Shanghai Jiaotong University, China

By using a "pseudo-RETI" instruction, the program in **Listing 1** provides a three-priority-level interrupt system for the 8051 µC. Among the three interrupt sources in the routine, External Request 0 (INTO) has the highest priority, and Internal Time/Counter 0 (ITO) has the lowest priority. In the ITO interrupt-service routine before the "pseudo-RETI" instruction, the address of the first instruction after the "pseudo-RETI" instruction goes back into the stack. The internal nonaddressable flip-flop of the ITO clears to acknowledge a higher interrupt after execution of the "pseudo-RETI" instruction, while the ITO interrupt-service routine executes continuously until the arrival of the RETI instruction. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2589.

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

LISTING 1—THREE-PRIORITY-LEVEL INTERRUPT SYSTEM FOR THE 8051

```
        ORG   0000H
        LJMP  START
        ORG   0003H
        LJMP  INT0
        ORG   000BH
        LJMP  IT0
        ORG   0013H
        LJMP  INT1
START:  MOV   SP,#60H
        MOV   IP,#01H           ;INT0 has high priority
        MOV   TMOD,#01H
        MOV   TH0,#00H
        MOV   TL0,#00H
        SETB  EA                ;enable INT0,INT1,IT0
        SETB  EX0
        SETB  EX1
        SETB  ET0
        SETB  TR0
        ...

INT0:   ...  .....;INT0 interrupt service program
        ...
        RETI
INT1:   ...                     ;INT1 interrupt service program
        ...
        RETI
IT0:    CLR   TR0               ;IT0 interrupt service program
        PUSH  DPL
        PUSH  DPH
        MOV   DPTR,#GO_ON
        PUSH  DPL
        PUSH  DPH
        RETI                    ;"pseudo-RETI"
GO_ON:  NOP
        ...
        ...
        MOV   TH0,#00H
        MOV   TL0,#00H
        POP   DPH
        POP   DPL
        SETB  TR0
        RETI
```

Look-up table facilitates bit flipping

Brad Bierschenk, High End Systems, Austin, TX

In certain instances in embedded software, it becomes necessary to flip the order of bits in a byte, so that B7:B0 becomes B0:B7. For example, this feature could be useful with a synchronous serial port that does not allow programmed selection of bit order (MSB first or LSB first) for its shift register. If a device to which the processor sends data expects one bit order but the serial port can provide only the other bit order, you must use a software method to translate the data. One solution to this problem is to provide a look-up table in ROM, in which the value of each byte in the table is offset into the table, but with bit order reversed. In other words, the first byte is offset 0 (00000000b), the second byte is offset 1 (10000000b), the third byte is offset 2 (01000000b), and so on.

LISTING 1—CODE SEGMENT FOR 80XC51 ARCHITECTURE

```
; Load value into accumulator (hex AA)
mov       A, #10101010b
; Load lookup table address into index
mov     DPTR,#InvertTable
; Load "flipped" value into accumulator
movc    A,@A+DPTR
; Accumulator should now hold hex 55
```

The program merely needs to load the value to be translated into a register that can serve as an offset, index the look-up table, and load the corresponding value from the index+offset location. This design uses the Philips 80xC51 architecture as an example. **Listing 1** shows a code segment for the µC. You can use the µC's 16-bit DPTR (data pointer) plus an 8-bit offset in its accumulator to load the accumulator with a byte value. This solution to bit flipping is dynamically more efficient than rotating a byte location through carry bits or other possible solutions. However, it's not the most statically efficient solution, because it requires 256 bytes of ROM for the look-up table. You can download **Listing 1** and the look-up table from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2621.

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

design ideas

Edited by Bill Travis and Anne Watson Swager

Free-line indicator stops interruptions

JM Terrade, Clermont-Ferrand, France

When only one phone line is available for two phones, each time you want to make a call, someone may be using the second phone. A simple circuit lights an LED, which indicates whether the line is free (**Figure 1a**). Batteries are unnecessary; the phone line powers the circuit, and an accumulator saves energy for an "in use" indication. A rectifier bridge ensures that the voltage is positive for the circuit. You can safely use this circuit on a private phone line, but you may need authorization before connecting it to your operator line.

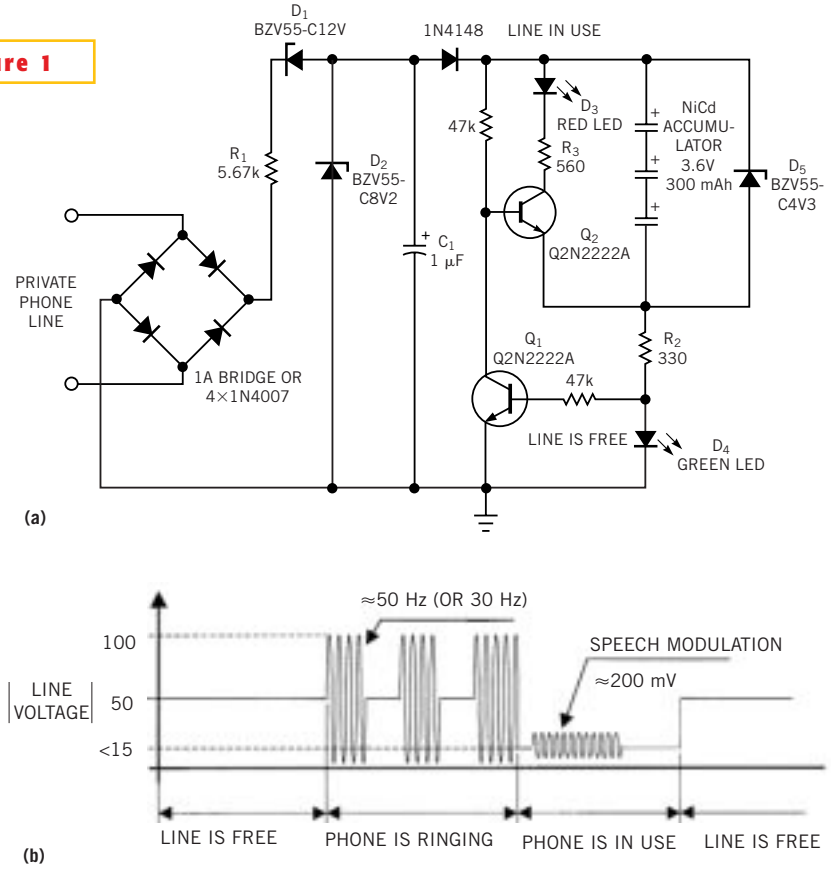
A phone line has different voltages between terminals depending on the line's availability. Three possible states exist (**Figure 1b**). **Figure 1b** shows the absolute value of the line voltage, because you can switch the line terminals.

In phase 1, the line is free, and the voltage is a continuous 50V dc voltage. The series zener diode, D_1, decreases the voltage by 12V, and R_1 and D_2 further limit the voltage to 8V. The current now flows through the NiCd accumulator, R_2, and D_4. The green LED, D_4, turns on, and the voltage across D_4 turns on Q_1. Q_2 is off, and there is no current for D_3. R_2 limits the current to 3 mA, which is enough to charge the accumulator. The green LED is a low-current model. D_5 protects the accumulator against overvoltage. Total line consumption is 5 mA because an extra 2 mA of current flows through D_2.

In phase 2, when the line is ringing, an extra ac voltage with an amplitude of 50V adds to the 50V dc voltage. In this case, the value of C_1 is critical. If C_1 is 1 µF, both LEDs will turn on because the 15V voltage value will vary. If C_1 is 47 µF, the voltage remains greater than 15V, and D_4 turns on.

In phase 3, when answering the call, the voltage falls to a value of about 10V. Voice modulation adds to this continuous voltage. The operator considers a phone line as "in use" if a current in the phone draws close to 30 mA through its 300Ω equivalent impedance. These current and impedance values are not criti-

Figure 1



D_4 lights to indicate that the phone line is free, and D_3 lights to indicate the line is in use (a), depending on the three possible phone-line voltage states (b).

cal. The line voltage, which is less than 15V, blocks D_1, and the voltage across D_2 is almost zero. Current discontinues its flow through R_2, and D_4 turns off. Q_1 is also off, and Q_2 conducts. Current travels from the accumulator through R_3, and D_3 turns on. R_3 limits the current to 3 mA, which is enough for a low-current LED. The 300 mAh, 3.6V accumulator is a phone type. If you unplug the circuit, D_3 remains on until the accumulator discharges.

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

Generate CID/CIDCW analog signals

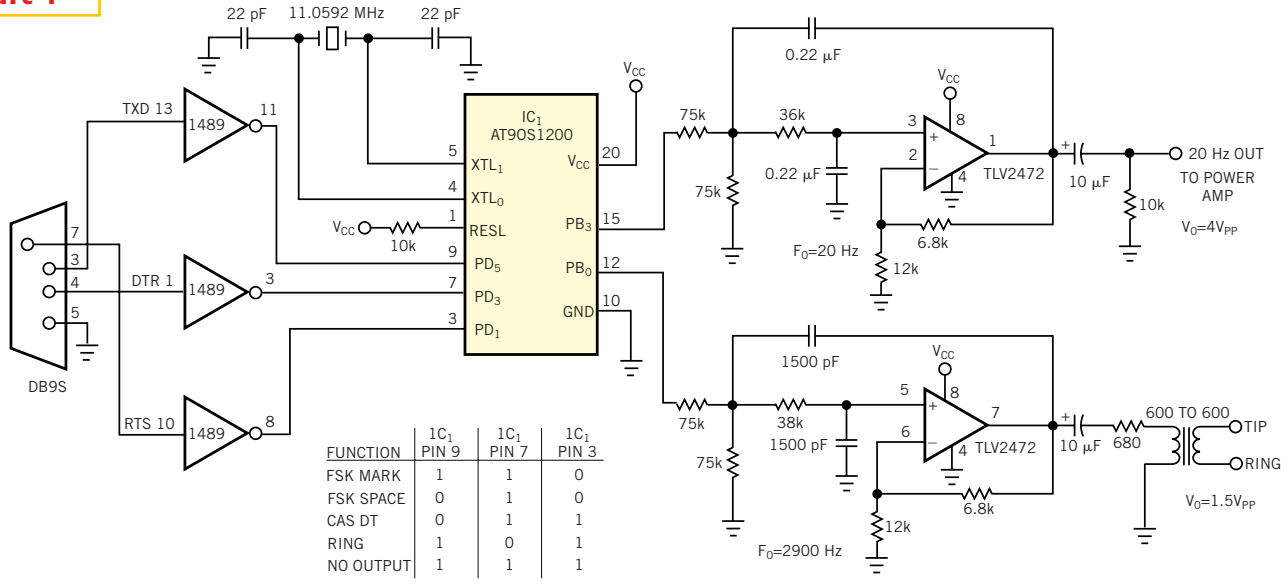
Hans Krobath, EEC, Nesconset, NY

A HIGH-SPEED µP and active lowpass filters can generate CID (caller-ID) and CIDCW (caller-ID-on-call-waiting) analog signals (**Figure 1**). CID data transmits at 1200 baud FSK between the first and second 20-Hz ring of an incoming call. CIDCW uses a CAS (CPE Alert Signal) dual tone, which consists of 2130 and 2750 Hz to initiate the FSK data transfer. You can produce these analog signals using software-generated PWM (pulse-width-modulated) outputs from a high-speed µP, such as an Atmel AVR or Scenix SX. You create these PWM outputs using a constant-sampling frequen-

LISTING 1—TABLE GENERATION FOR PWM MARKING AND SPACING SEQUENCE

```
PI = 3.1415926#
KM = 182                ' MAXIMUM RANGE OF PULSE WIDTH
DT = 576 / 11059200     ' TIME FOR EACH SAMPLE
DTI = DT
F1 = 19200 / 7          ' F1 = 2742.9 (2750)
F2 = 19200 / 9          ' F2 = 2133.3 (2130)
DTI = DT
FOR I = 1 TO 63         ' 63 SAMPLES THEN REPEAT
A = SIN(2 * PI * F1 * DT)   ' GENERATE F1 SINE WAVE
B = SIN(2 * PI * F2 * DT)   ' GENERATE F2 SINE WAVE
V = KM / 2 + (KM / 4) * A + (KM / 4) * B  'GENERATE F1 + F2 SINE WAVE
PRINT I, DT, V
DT = DT + DTI           ' NEXT SAMPLE
NEXT I
```

Figure 1



FUNCTION	IC_1 PIN 9	IC_1 PIN 7	IC_1 PIN 3
FSK MARK	1	1	0
FSK SPACE	0	1	0
CAS DT	0	1	1
RING	1	0	1
NO OUTPUT	1	1	1

Based on RS-232 inputs from a PC, pin 12 of IC_1 produces a PWM output proportional to FSK 1200-baud serial data or a dual-tone CAS signal. The output at pin 15 is a 20-Hz-ring signal. Subsequent lowpass filtering produces sine-wave outputs.

cy, and you select this frequency so that small, repetitive integer-number samples define the desired output. The 11.0592-MHz µP clock frequency and single-instruction cycle time are the same for IC_1. The design produces the 1200-baud FSK marking frequency of 1200 Hz by using 22 cycles of a constant-sampling frequency of 11.0592 MHz/419. The design produces the 2200-Hz spacing frequency by using 12 cycles of the same sampling frequency. At each sample, the µP produces a PWM TTL output proportional to the analog voltage. The µP obtains successive pulse widths from a look-up table that also provides pointers to allow phase-coherent transitions when switching between marking and spacing. A Basic program produces the tables for the marking and spacing PWM sequence (**Listing 1**).

A sampling frequency of 11.0592 MHz/576, or 19,200 Hz, produces the dual-tone 2750- and 2130-Hz CAS signal. Seven samples define 2742.9 Hz, and nine samples define 2133.3 Hz. These samples are well within the Bellcore specification. The sum of these waveforms repeats 7×9=63 times when you use the 19,200-Hz sampling frequency. Another Basic program produces the pulse-width outputs that generate the dual tone (**Listing 2**).

The design produces the 20-Hz-ring sine wave, which you can amplify and step up to the appropriate level in a similar manner using values to produce pulse widths that **Listing 3** provides.

An assembly-language program written for IC_1 uses the table-generating programs in **Listings 1**, **2**, and **3**. The program fits within the 512-word memory space to generate three PWM outputs, which comprise the analog signals for CID/CIDCW after lowpass filtering. You can download the program from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2622.

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

LISTING 2—TABLE GENERATION FOR DUAL-TONE PWM OUTPUTS

```
PI = 3.1415926#
KM = 182                ' MAXIMUM RANGE OF PULSE WIDTH
DT = 576 / 11059200     ' TIME FOR EACH SAMPLE
DTI = DT
F1 = 19200 / 7          ' F1 = 2742.9 (2750)
F2 = 19200 / 9          ' F2 = 2133.3 (2130)
DTI = DT
FOR I = 1 TO 63         ' 63 SAMPLES THEN REPEAT
A = SIN(2 * PI * F1 * DT)   ' GENERATE F1 SINE WAVE
B = SIN(2 * PI * F2 * DT)   ' GENERATE F2 SINE WAVE
V = KM / 2 + (KM / 4) * A + (KM / 4) * B  'GENERATE F1 + F2 SINE WAVE
PRINT I, DT, V
DT = DT + DTI           ' NEXT SAMPLE
NEXT I
```

LISTING 3—TABLE GENERATION FOR A 20-Hz-RING SINE WAVE

```
PI = 3.1415926#
KM = 246                ' MAXIMUM RANGE OF PULSE WIDTH
DT = 27648 / 11059200   ' TIME FOR EACH SAMPLE
DTI = DT
F = (11059200 / 27648) / 20 ' OUTPUT FREQUENCY
FOR I = 1 TO 20             ' 20 SAMPLES THEN REPEAT
V = KM / 2 + (KM / 2) * (SIN(2 * PI * F * DT))
PRINT I, DT, V
DT = DT + DTI           ' NEXT SAMPLE
NEXT I
```

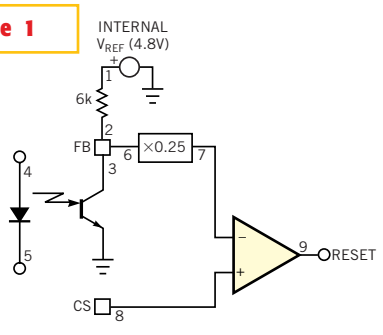
Bipolars provide safe latch-off against opto failures

Christophe Basso, ON Semiconductor, Toulouse Cedex, France

A typical SMPS (switch-mode power supply) uses a reference voltage and an optocoupler to regulate the output voltage. The optocoupler carries the isolated information from the secondary side to the primary nonisolated side and ensures proper output regulation. Current-mode architectures obtain the final feedback level by implementing the optocoupler as a variable resistor that internally connects to a pullup element (**Figure 1**). The voltage on the feedback pin (FB) directly fixes the power supply's peak-current setpoint. When the output power demand is low, V_{FB} and the peak current are low. When the output requires power, V_{FB} increases and authorizes a higher peak current.

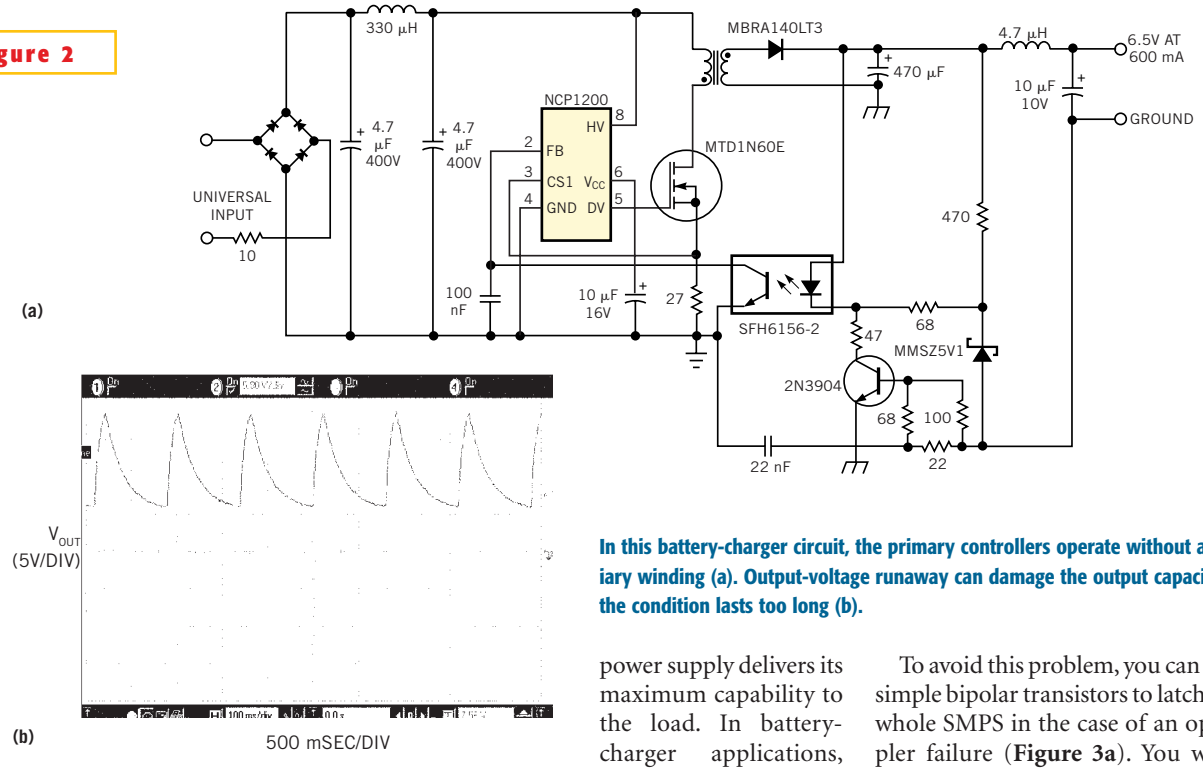
When an output short circuit is present, the LED optocoupler loses its bias, and the variable-resistor action disappears. The internal 6-kΩ pullup resistor forces V_{FB} to its maximum value, and the

Figure 1



A typical current-mode architecture uses an optocoupler as a variable resistor.

Figure 2



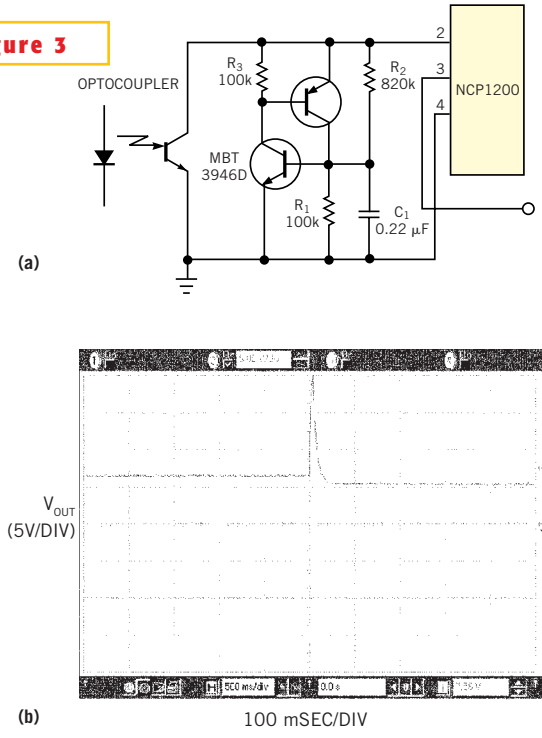
In this battery-charger circuit, the primary controllers operate without any auxiliary winding (a). Output-voltage runaway can damage the output capacitors if the condition lasts too long (b).

power supply delivers its maximum capability to the load. In battery-charger applications, however, short-circuit conditions do not cause optocoupler loss because the output permanently monitors the delivered current and forces the current to be constant (**Figure 2a**). In this case, the primary implementation is simple because of the lack of the auxiliary winding. If the optocoupler fails to open, the peak-current setpoint increases to its maximum for a given time until the burst-protection feature takes over. This situation repeats until the user switches off the SMPS. The worst case arises in unloaded situations: The output voltage runs away until the burst sequence ends (**Figure 2b**). As a result, this condition can quickly damage output capacitors if this situation lasts too long.

To avoid this problem, you can use two simple bipolar transistors to latch-off the whole SMPS in the case of an optocoupler failure (**Figure 3a**). You wire the bipolars in a thyristor manner using a dual-transistor device, such as the MBT3946D.

In normal operation, R_1 through R_3 ensure that neither the pnp nor the npn can start conducting. Furthermore, R_1 and R_2 form a voltage divider that monitors V_{FB}. When V_{FB} increases, the voltage over C_1 begins to rise until the npn transistor starts to pull the pnp transistor's base to ground. This action immediately fires the SCR, which locks V_{FB} to nearly zero. When V_{FB} is less than 1.4V, the NCP1200 IC stops delivering pulses until the SCR resets. You can reset the SCR by unplugging the charger from the main outlet. **Figure 3b** shows the results of this operation and that the operation is safe with an open optocoupler. When the optocoupler fails, the output voltage grows until the SCR stops the IC operation. V_{OUT} then slowly discharges toward ground. C_1 filters out any spurious noise that appears at power-on that could adversely fire the SCR.

Figure 3



You can use a dual npn+pnp to build a cheap thyristor (a). When the thyristor fires, the pulses permanently stop, leaving no voltage runaway (b).

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

Simple logic analyzer pushes µC to its limit

Tom Lyons Fisher, Inexpensive Systems, Huntingdon, PA, and Michael Deskevich, Juniata College, Huntingdon, PA

A simple logic-analyzer design is compatible with all versions of Windows and pushes the PIC 18C252 chip to its speed limit to achieve a 1-MHz sampling rate (**Figure 1**). The circuit can examine three channels of relatively low-speed logic signals that have infrequent, or sparse, transitions. The analyzer can record only 510 transitions per run, but a run can last as long as several minutes, if necessary. Applications include monitoring the I/O of an IBM keyboard or printer port, TI-calculator intercommunication, and serial (RS-232) signals.

Although the ability of the PIC 18C252 µC to use a 40-MHz clock input suggests that sampling rates in the megahertz region are easy to achieve, careful programming and unorthodox use of some of the PIC's features are necessary even to achieve a rate of 1 MHz. Nevertheless, the 18CXXX series works well for this application because it is the fastest PIC available, with 1536 bytes of RAM and an RS-232 port on-chip. Thus, this logic analyzer requires minimal external circuitry.

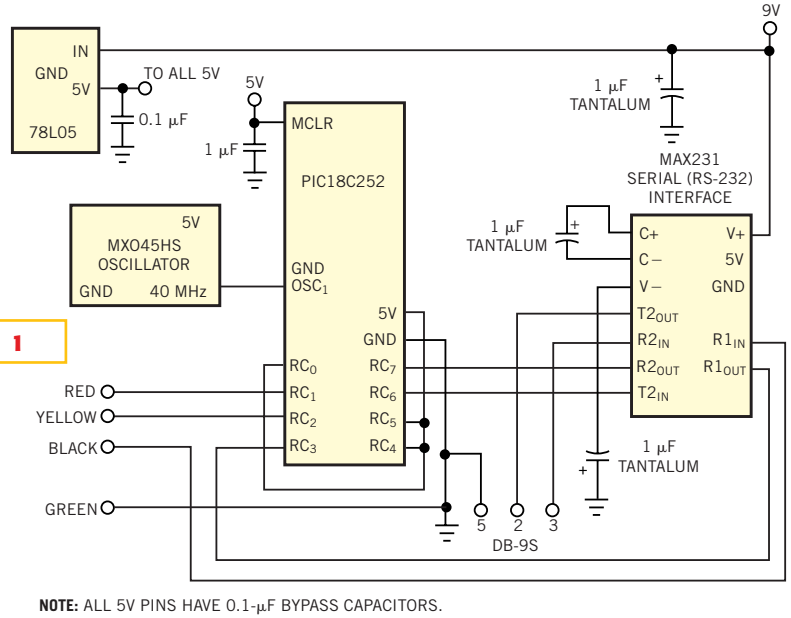
LISTING 1– CRITICAL CODE FOR THE LOOP IN MACRO FORM

```
Scan      MACRO
          LOCAL  Scn2,Scn3
;Has there been a transition on any channel?
  Scn3    movf   PORTB,W,A
          xorwf  Last,F,A
          movwf  Last,A
          bnz    Scn2
;If no transition, just increment the interval timer.
          tblrd  +*
          movff  TBLPTRU,INDF1
          goto   Scn3
;If transition, record the new logic states and the current value of interval timer.
  Scn2    iorwf  POSTINC1,F,A
          movff  TBLPTRH,POSTIC1
          movff  TBLPTRL,POSTINC1
          ENDM
```

Although the signal-input channels use the Schmitt-trigger inputs, which are available only on Port C, each channel has different capabilities. The red and yellow channels detect 5V logic signals. You can set the red channel to trigger on either a positive- or a negative-going transition. The black channel monitors only bipolar "external" RS-232 signals, which the MAX231 level translator converts into standard 5V logic (5V=logic 1).

With a 40-MHz clock, the PIC has a machine cycle of 10 MHz, or 100 nsec, so the sampling-code loop must use no more than 10 cycles for a sampling rate of 1 MHz. Because of stringent time constraints, a summary of the software strategy is "save fast and pick up the pieces later." This device records the time between transitions and the logic state of the three channels after the transition. **Listing 1** shows the critical code for the loop in macro form, before expansion. It takes exactly 10 µsec to determine whether there is a transition to record. Notice that this design uses the external address register, TBLPTR, as a timer/counter because a two-cycle command can increment the register's 21 bits. Because there is no time to store 3 bytes of a transition event within 10 cycles, the design stores the upper timer/counter byte when there is no transition, leaving only the logic states and the other 2 timer/counter bytes for you to store when there is a transition. The compiler copies the entire loop into program ROM 510 times to avoid a goto command, which requires two cycles.

When the loop is active, it records the current state of the timer/counter, which



NOTE: ALL 5V PINS HAVE 0.1-µF BYPASS CAPACITORS.

Figure 1

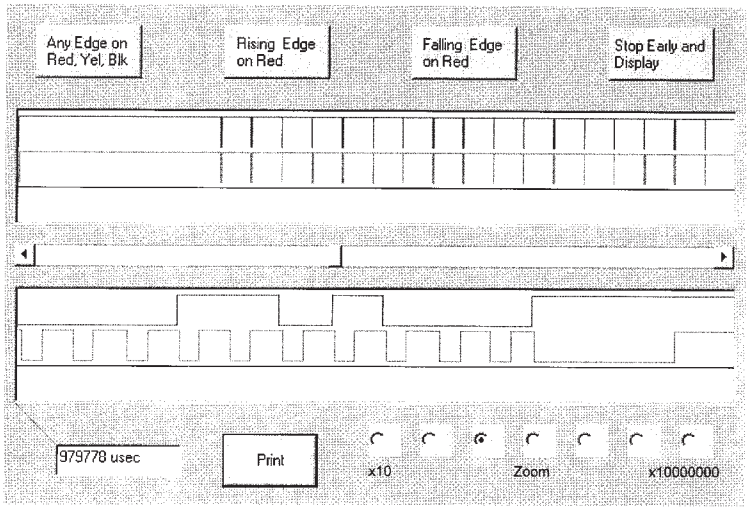
The external wiring is simple because the Schmitt triggers, memory, and UART are all within the µC.

may "roll over" if the intervals are long. After completing the run, the program makes a pass through the data to convert the recorded times to true intervals before sending the results to the PC. The resulting format is RYB-MMMMM IIIIIII LLLLLLL, where R, Y, and B are the logic states of the red, yellow, and black channels, and M, I, and L are the most-, intermediate-, and least-significant bits of the time interval between transitions in microseconds.

After the data from a run transmits through a PC communications port at 19,200 baud, a Visual Basic program displays the data. The screen displays the three traces in colors corresponding to the channel test leads. It also permits the user to expand the traces for detailed examination by using the mouse to manipulate a horizontal scroll bar and zoom buttons, which select the X-position and magnification of the expanded traces (**Figure 2**).

The Visual Basic program and both the PIC source and the object code are available from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2617.

Figure 2



The display shows the entire run in the upper plot and a magnified portion in the lower plot. The small window at the lower left reports the exact time of the leftmost displayed transition.

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

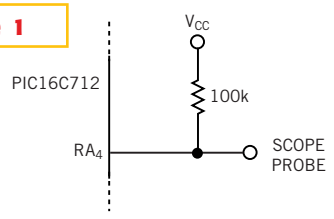
PIC debugging routine reads out binary numbers

Brad Peeters, Theta Engineering Inc, Costa Mesa, CA

The "burn and learn" method of firmware development excludes an in-circuit emulator and a serial port. With this method, it is common practice to use spare I/O pins on a µC as a debugging aid. By strategically placing instructions to set and clear these I/O pins in the code and then observing the pins with a scope, you can obtain limited real-time information about the execution of the code. An I/O pin serves as a 1-bit debugging port.

You can overcome this limitation somewhat by writing a function or subroutine that shifts data out serially on the port pin. Then you can use the scope to capture and observe several bits of information. However, setting the port pin high for a one bit and low for a zero bit results in a display that requires careful reading. Unless you know and accurately measure the timing of the bits, judging which bit corresponds to which position on the scope display presents a challenge. If the data includes several zeros or several ones in a row, no transitions exist with which to align the timing. This problem becomes particularly acute when attempting to capture more than 8 bits.

Figure 1



Attaching a large-valued pullup resistor on an open-drain I/O pin results in fast falling edges but slow rising edges.

A software routine for midrange PIC µCs (**Listing 1**) overcomes this difficulty by producing a scope display that you can read at a glance. The routine encodes zero as a short pulse and one as a long pulse. Using an open-drain I/O pin with a large-value pullup resistor results in fast falling and slow rising edges, which is due to the RC time constant of the pullup resistor and the capacitance of the pin and scope probe (**Figure 1**). Consequently, zeros show up as short spikes in the display, and ones appear as medium spikes. The separation between consecutive bytes appears as a tall spike or pulse (**Figure 2**). Each byte starts with a clean falling edge, which serves as a convenient trigger signal for the scope. A midrange PIC running at 4 MHz using a 100-kΩ pullup resistor produces the plot in **Figure 2**. The resistor value is not critical. To use another clock rate, you can scale the resistor approximately as the inverse of

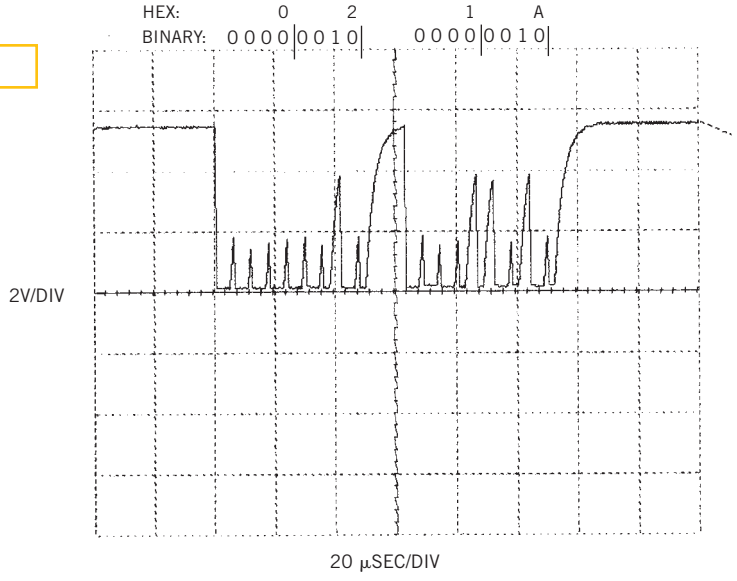
the clock rate. For example, at 8 MHz, a pullup resistor of 47 kΩ produces equivalent results.

The scope plot depicts a 16-bit value of 00000010 for the first byte and 00011010 for the second byte. Each invocation of the subroutine in **Listing 1** displays the most-significant bit of 1 byte first. By taking care to invoke the subroutine on the most-significant byte of a multibyte value first, the scope display naturally reads from left to right. Hence, the depicted value is 021A hex. By slowing the timebase of the scope, you can display a 32-bit value. The resolution of the scope is the only limitation on the amount of data the scope can display.

Because the subroutine preserves all registers and flags, except for the general-purpose register for the subroutine itself, you can safely insert a call to the subroutine at any point in your code to obtain visibility into the value of the W register. The limitation on the W register is not a severe restriction because in the PIC architecture, most operations pass through the W register. One additional instruction suffices to load any general-purpose register into W before calling the subroutine.

The code snippet in **Listing 2** shows the addition of a 16-bit value called *Result* to a 24-bit *Base* value. Inserting *call Debug* instructions at the points that the arrows indicate makes the 16-bit *Result* value visible (**Figure 2**). You can download the subroutine from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2594.

Figure 2



Inserting "call Debug" instructions into the code makes the 16-bit result visible. Zeros appear as short spikes, ones appear as medium spikes, and a tall spike indicates separation between consecutive bytes.

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

LISTING 2–CODE SNIPPET

```
        movfw   ResultHi
        addwf   BaseHi
        btfsc   C
        incf    BaseEx
        movfw   ResultLo
        addwf   BaseLo
        btfsc   C
        incfsz  BaseHi
        goto    $+2
        incf    BaseEx
```

LISTING 1–DEBUGGING SUBROUTINE

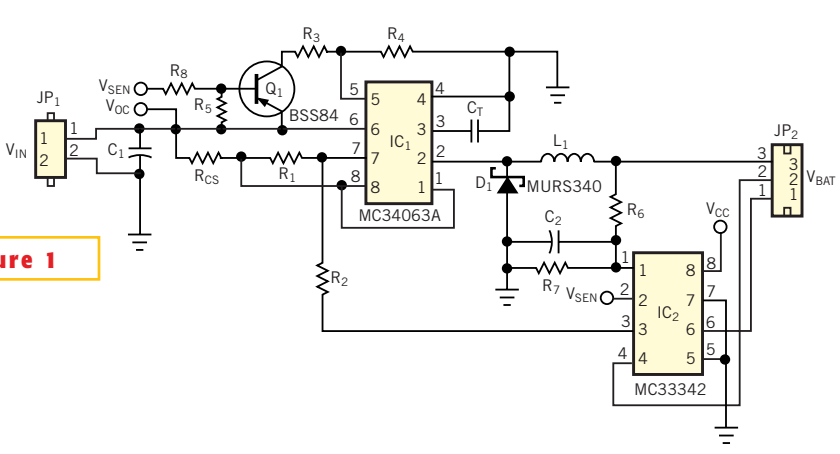
```
;Define the pin to be used as the test point:
#define  TP1        5,4         ;bit 4 of Port A.

;Allocate a register for use by the debug subroutine:
    cblock
    DebugReg
    endc

;-----------------------------------------------------------------
; This subroutine takes the contents of the W register and shifts
; it out, bit by bit, on the TP1 port pin for viewing on a scope.
; Long pulse is a one and short pulse is a zero.  W register and
; flags are preserved!
Debug
    movwf   DebugReg            ;Store value in reg.
    rlf     DebugReg            ;Shift MSB into CY.
    bcf     TP1                 ;"Start of byte" transition.
    btfsc   C                   ;If bit is set,
    bsf     TP1                 ;start pulse sooner,
    rlf     DebugReg
    bsf     TP1                 ;rather than later.
    bcf     TP1                 ;End of pulse.
    btfsc   C                   ;Test next bit and repeat…
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg
    bsf     TP1
    bcf     TP1
    btfsc   C
    bsf     TP1
    rlf     DebugReg            ;Restore CY flag.
    bsf     TP1
    bcf     TP1
    nop                         ;Optional to provide consistent timin
    bsf     TP1                 ;"End of byte" transition.
    return
```

Edited by Bill Travis and Anne Watson Swager

Switching regulator charges NiMH batteries

Jason Hansen and Jim Hill, On Semiconductor, Phoenix, AZ

Many NiMH (nickel-metal-hydride) fast chargers use linear regulators, such as the LM317, in a current-source configuration with a charge-monitor IC. This arrangement dissipates much heat and requires considerable heat sinking, adding weight and cost. In the fast-paced, portable-system world, users do not want to carry heavy battery chargers that feel hot. A switching regulator can alleviate the weight and heat problem. In **Figure 1**, an MC34063A switching regulator and an MC33342 NiMH battery-charger IC combine to generate from an unregulated power supply a 600-mA battery charger for one to four NiMH cells. Using some tricks with the current limit and the feedback network, you can make a simple dc/dc converter into a programmable current source for NiMH charging. IC_1 operates in continuous mode in the buck converter without the output capacitor. The elimination of the output filter causes the inductor to act as the current source to charge the batteries. The externally programmable current limit sets the peak current in IC_1. You should select the inductor size so the peak-to-peak ripple does not exceed a predetermined level.



A switching regulator eliminates much of the heat in an NiMH battery charger.

The equation $V_1 t_{ON} = V_2 t_{OFF}$ helps determine the minimum input voltage to the system.

V_1 is the input voltage minus the saturation voltage of the switch, minus the battery voltage. V_2 is the battery voltage summed with the forward drop of the Schottky diode. In continuous mode, t_{ON} plus t_{OFF} is the total switching period. Rearranging the above equation for input voltage yields

$$V_{IN} = V_{SAT} + V_{BAT} + (V_{BAT} + V_D)\left(\frac{T}{t_{ON}} - 1\right).$$

The minimum input voltage to the system is 5.6V if t_{ON} is 7 sec, the switching frequency is 100 kHz, V_D is 0.3V, and V_{SAT} is 1.2V. Selecting 6V as the input voltage and allowing 0.1A ripple in the charging current and 1.4V across the inductor at maximum charge, the calculated inductor value is 126 µH. The R_6/R_7 ratio sets the fast-charge window. With V_{BAT} at 3.05V, slightly higher than the maximum charge level for the two cells, the ratio is 0.525. With 10-µA bias current through the resistive divider, R_6 is 27 kΩ, and R_7 is 51 kΩ. C_2 has a value of 1 nF and provides stability during fast-charge battery-voltage monitoring. C_1 provides input stability. Its size depends on the input-voltage ripple. C_T is 180 pF for 100 kHz. R_{CS}, 0.5Ω, is a function of peak current and the turn-off threshold current.

The feedback circuit of IC_1 connects to IC_2's V_{SEN} pin for the fast-charge monitor shutdown. During the fast charge, this open-collector output pin pulses low every 1.38 sec. R_3, R_4, R_5, R_8, and Q_1 make up the fast-charge monitor circuit. During normal-charge operation, Q_1 is off, and R_4 returns the feedback pin to ground. R_5 ensures that the base is high. When the MC33342 needs to measure the battery voltage, the V_{SEN} pin pulls the base of Q_1 low. The R_3/R_4 ratio ensures that the feedback pin is at a level higher than its threshold. You must understand the peak-voltage detector to properly design the dc/dc converter. The detector samples the battery voltage every 1.38 seconds. The sample time is 33 msec with an 11-msec preset time at the beginning. During this preset time, the MC33342

toggles the MC34063A into 0%-duty-cycle mode. This action shuts down the current to the battery so the MC33342 can take an accurate voltage reading. The inductor current must reach 0A during the initial preset time to minimize errors in the voltage reading.

R_{CS}, R_1, and R_2 make up the trickle-charge circuit. To effect trickle charge, this circuit connects to the current-limit pin of the MC34063A to create an offset for the maximum current. The open collector of the Fast/Trickle Charge pin goes low to enter trickle-charge mode. When the collector goes low, R_1 creates a voltage offset. R_1 and R_2 are much larger than R_{CS}; therefore, the offset current is very low. Because the current-sense pin monitors a low differential voltage across R_{CS}, V_{CC} needs to be within a few hundred millivolts of its nominal value for charge-current accuracy. The desired trickle-charge current determines R_1 and R_2. The target peak current for the trickle charge is 100 mA. The voltage across R_{CS} is 30 mV at the threshold; therefore, the voltage across R_1 must be 270 mV. For a current of 1 mA, the value of R_1 is 270Ω. Because the saturation voltage of the Fast/Trickle Charge pin is 0.2V, R_2's value is the input voltage minus the current-sense threshold voltage minus the saturation voltage of the charge-control pin divided by the selected current. The calculation yields approximately 5.5 kΩ for R_2. The component values in **Figure 1** produce a 600-mA battery charger.

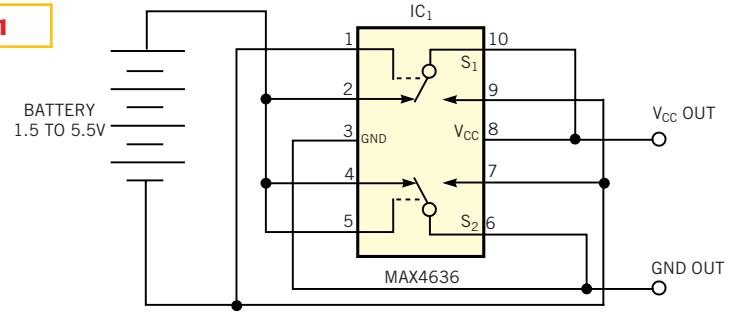
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Circuit provides reverse-battery protection

John Guy, Maxim Integrated Products, Sunnyvale, CA

A UNIVERSAL PROBLEM in battery-operated devices is the threat of damage when an end user (never an engineer) inserts the battery backward. You can avoid damage by inserting a single diode or by using a diode-bridge configuration, but those fixes waste power and reduce the supply voltage by adding one or two diode drops between the battery and the supply rail. An alternative solution not only protects against battery-reversal damage, but also automatically corrects the reversal (**Figure 1**). To eliminate the voltage drops associated with discrete diodes, a low-on-resistance, DPDT (double-pole, double-throw) switch serves as a full-wave rectifier. When you insert the battery with the correct polarity as shown, the upper switch, S_1, is in its normally closed state, because its control pin is in its low state. The resulting connection from Pin 2 to Pin 10 provides a low-impedance path from the battery to the V_{CC} terminal. Conversely, the lower switch, S_2, closes its normally open terminal (not as shown) because its control pin is in its high state. The resulting path from Pin 7 to Pin 6 connects the battery's negative terminal to ground.

Figure 1



This circuit senses battery polarity, then quickly connects the load or swaps the battery leads.

The ESD-protection diodes in IC_1 guarantee start-up and act as a full-wave rectifier. MOSFETs internal to the analog switch turn on when the battery voltage exceeds 1V. Their less-than-20-nsec turn-on time enables the circuit to maintain normal operation by quickly swapping the leads of a reversed-polarity battery connection. The circuit resistance is proportional to the battery voltage. When the circuit operates from four NiCd, NiMH, or alkaline cells, the resistance in each leg of the rectifier is 2.5Ω (5Ω total). Operation with a two-cell battery (2.4 to 3V) yields a total resistance of 10Ω. IC_1 is rated for operation to 5.5V with 30-mA continuous current, making the circuit useful for cordless phones, portable audio equipment, handheld electronics, and other light- to medium-current applications. IC_1's miniature 10-pin μMAX package takes less space than four through-hole signal diodes and is almost as small as two SOT-23 dual signal diodes.

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

Transistor junctions monitor temperature zones

Matt Smith, Analog Devices, Limerick, Ireland

A simple circuit uses transistor junctions to monitor multiple temperature zones (**Figure 1**). The temperature sensors are ordinary, general-purpose, low-cost, diode-connected transistors. The well-known diode equation $V_{BE}=(kT/q)\times\ln(I_C/I_S)$ shows that there is a temperature dependency of approximately 2.2 mV/°C for a base-emitter junction. By forcing a two-level current through the base-emitter junction and measuring the resultant voltage, you can accurately determine the junction temperature, a technique known as ΔV_{BE} sensing. To prevent self-heating with this technique, current levels must be low. IC_2 uses this approach and supplies a low-level switched current source on its D+ and D− pins. An on-chip ADC converts the voltage information on D+ and D− into digital data that IC_2 stores in a register.

To monitor multiple-channel temperatures, you need to multiplex the measurement channels. A four-channel differential multiplexer, IC_1, selects the transistor junction that the circuit measures. The differential multiplexer ensures that D+ and D− remain as differential signals to preserve noise immunity. By cycling through the A_0 and A_1 address lines of the multiplexer, the µC or µP can poll each channel in sequence. If extra channels are necessary, you can add multiplexer channels.

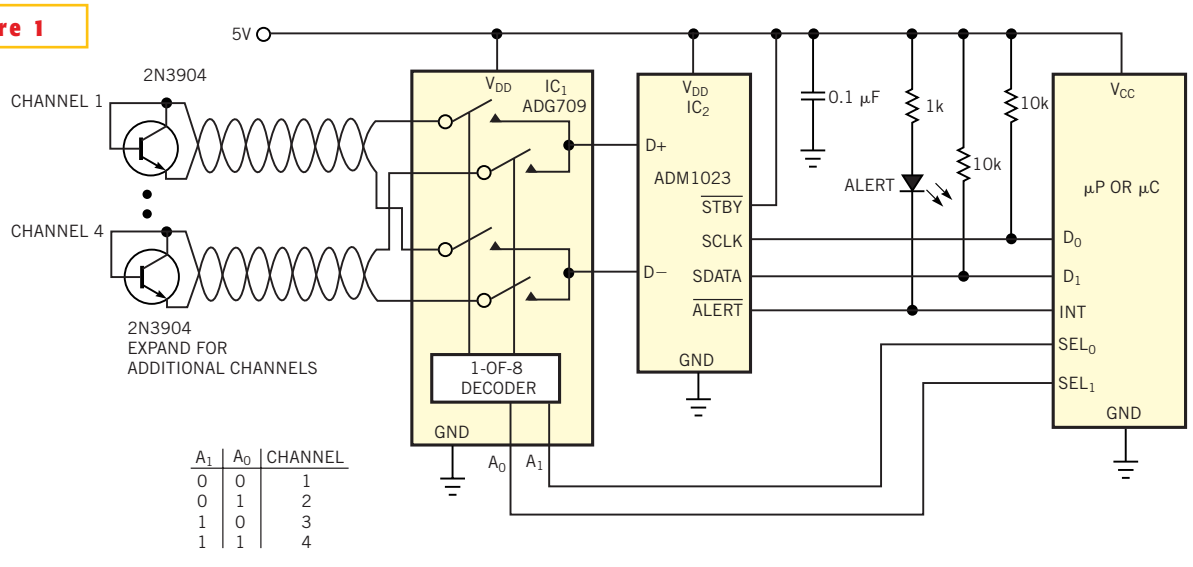
The on-resistance of the multiplexer channel results in a voltage drop across the channel. Therefore, you initially need to calibrate the circuit to remove this error. Fortunately, the error is constant because the channel resistance remains constant. You can use an offset register in IC_2 to store and automatically subtract the offset.

The remote-sensing transistors connect via a twisted-pair cable, and the cable can be as long as 50 ft. In extremely noisy environments, using a shielded twisted pair prevents the noise from interfering with the sensitive measurement. The circuit features a standard two-wire SMBus or I²C interface, enabling communication with a µC or µP.

The circuit can accommodate a theoretical temperature range of −128 to +128°C. However, the practical range is more limited than these temperatures because moisture causes leakage currents and, hence, temperature errors. IC_2 also contains high- and low-limit registers and has an alert output. Thus, you can use the circuit to ensure that temperatures remain within an allowable band. Any deviation outside the limits, either high or low, results in activation of the alert output. The alert line drives an interrupt line on the µC or µP. The circuit can also detect fault conditions, such as open or short circuits, on the sensing elements. Fault conditions generate an alert signal. An interrupt-service routine can then interrogate the status register to provide fault identification and initiate corrective action.

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

Figure 1



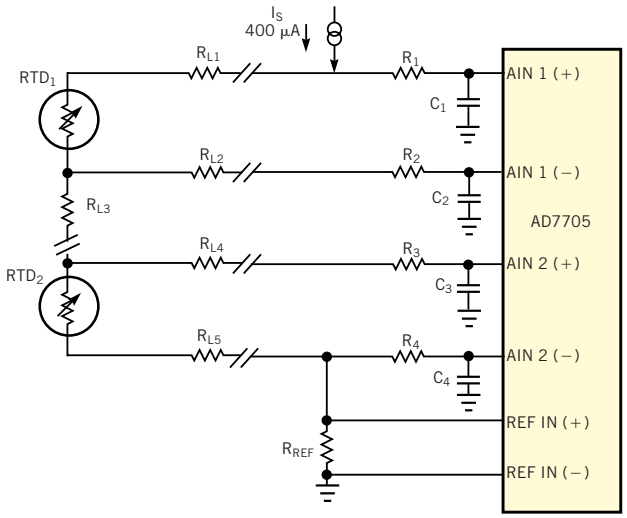
Diode-connected transistors monitor multiple temperature zones.

RTDs provide differential temperature measurement

John Wynne, Analog Devices Inc, Limerick, Ireland

You sometimes need to measure the differential temperature between two points in a system to a greater accuracy than that required in measuring the absolute temperature at either of the individual points. Differential-temperature measurement is necessary, for example, in monitoring heating-energy consumption in an apartment. Hot water enters from one pipe, circulates around the apartment through the radiators, and exits through a second pipe. Billing depends on the temperature differential between the entry and exit copper pipes, so absolute temperature is irrelevant. One way of measuring is to attach an RTD (resistance-temperature detector) to each pipe as it enters or exits the apartment and to take the voltage difference across the two RTDs. To ensure that the measurements are truly relative, you must wire the RTDs in series and excite them with the same current source, I_S (**Figure 1**). The same excitation current also flows through the reference resistor, R_{REF}, and generates the voltage reference for the ADC. Hence, the entire circuit is ratiometric. Therefore, both the current source, through, and the reference resistor need not be particularly stable over temperature for the circuit to operate properly. The circuit is also tolerant of ohmic drops in the connections to the three-wire RTDs.

Figure 1



This ratiometric differential-temperature measurement system eliminates ohmic wiring drops as sources of errors.

Channel 1 of the AD7705 reads an input voltage equal to $I_S R_{RTD1} + I_S R_{L1}$. Channel 2 reads an input voltage equal to $I_S R_{RTD} + I_S R_{L5}$. R_{L1} and R_{L5} represent the wiring resistances between the local electronics and the remote RTD elements. You should wire the RTDs such that $R_{L1} = R_{L5}$. Using software, subtract the ADC's Channel 1 reading from the Channel 2 reading. The ohmic drops cancel, leaving the differential temperature as the only remaining term. The inter-RTD wiring resistance, R_{L3}, does not appear in the equations and, therefore, has no effect. The input impedance of the AD7705 is very high, so essentially no current flows through R_{L2} and R_{L4}. The RC combinations act as lowpass filters that attenuate high-frequency noise that the wiring picks up. This filtering function is especially important with RTDs that are remote from the AD7705 and related measurement electronics. Choosing these components is straightforward, according to the data sheet at www.analog.com.

The RTDs give rise to a certain source of errors. Consider a common 100Ω platinum RTD with a resistance coefficient of 0.0038Ω/Ω/°C. This type of sensor, the European PRTD, is the most common RTD sensor. It is available in accuracy-tolerance classes A and B (or DIN A and DIN B), which specify both the initial accuracy at 0°C and the interchangeability over the operating range. Class A specifies $\pm(0.15 + 0.002|t|)$, and Class B specifies $\pm(0.3 + 0.005|t|)$, where t is the specified interchangeability temperature. You can buy two Class A, 100Ω, platinum RTDs from the same manufacturer and find that one is reading 0.2°C high at 25°C and the other is reading 0.2°C low at 25°C. This difference represents an apparent 0.4°C difference before you even commission the measurement system. To combat this initial error, you must either request a matched pair of RTDs from the manufacturer or calibrate out this difference at the time of installation. For instance, some sensor manufacturers sort PRTDs into tolerance groups with maximum Δt of ±0.05°C over 0 to 100°C. Alternatively, you can easily calibrate out the error by using the AD7705's separate gain and offset registers for the two channels.

The AD7705 specifies integral nonlinearity at 14 bits or better. However, the ADC measures the two inputs with 16 bits of peak-to-peak resolution. All this resolution is useful, because the ADC has the same linearity for either channel, whatever that resolution may be. This premise assumes that the gain of the ADC's internal PGA does not change between channels. Changing channels via the internal multiplexer does not contribute any additional error sources. Thus, differential-temperature measurements have a resolution of 14 bits or better.

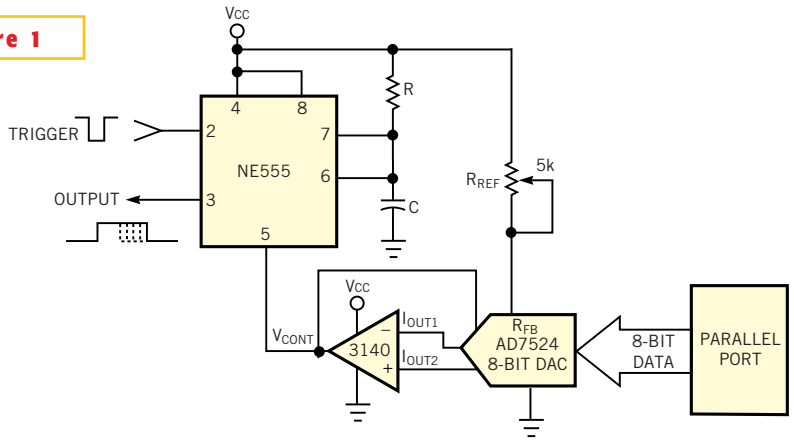
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

One-shot circuit is programmable

J Jayapandian, IGCAR, Tamil Nadu, India

Figure 1 shows how to digitally program the on-time of a one-shot multivibrator circuit. More and more, the Internet is playing a role in control operations in industrial and R&D endeavors and in household appliances. One-shot circuits are popular choices for the on/off control circuitry. You can interface the programmable one-shot design in **Figure 1** with any intelligent system, such as a PC, a µP, or a µC. The design uses a low-cost NE555 timer and an 8-bit AD7524 D/A converter. The timer IC is connected in a one-shot configuration with an on-time transfer function of t=1.1 RC. The control voltage on Pin 5 of the 555 can change the threshold of the comparator in the timer IC, thereby changing the on-time of the one-shot's output. In other words, by selecting the voltage on this pin, you can control the pulse width of the output waveform of the timer. The 8-bit DAC, with its MC 3104 op-amp buffer, provides programmable control of the one-shot's pulse width. You can send the required 8-bit word to the DAC via a PC's parallel port or from a µP or µC. The maximum on-time of the one-shot is a function of the R and C values. You can vary the pulse width from minimum to maximum by changing the bit pattern at the DAC's input from 00 to FF.

Figure 1



An 8-bit code applied to the DAC determines the on-time of this one-shot circuit.

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

BIOS interrupt does eight-channel frequency counting

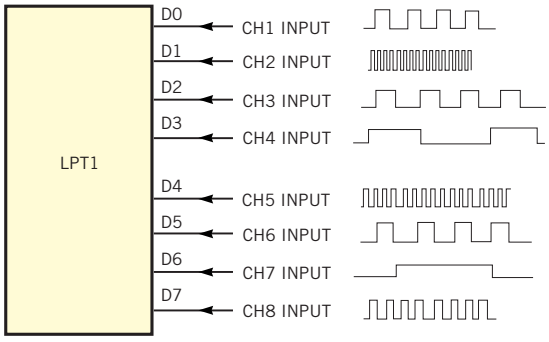
J Jayapandian, IGCAR, Tamil Nadu, India

The simple, no-cost connection in **Figure 1** provides frequency measurement for eight pulse trains through a PC's LPT1 port. The method uses a special BIOS interrupt and its handler routine without any hardware circuitry. The interrupt-handler routine, written in Turbo C, recognizes the external clock pulses coupled to the eight lines in the LPT1 port and determines their frequency, the number of pulses per second (**Listing 1**). The variable TICKER in the handler routine recognizes the occurrence of INT 1CH. The handler routine reads an 8-bit DATA word from the LPT1 port for every TICKER from the data received (the variable "COUNT"). The software extracts the 8 bits as eight-channel, single-bit data CH1, CH2, CH3, CH4, CH5, CH6, CH7, and CH8 by rotating right and by effecting a subsequent AND operation. For CH1 data, the 8-bit data received in "COUNT" is ANDed with 0x01, resulting in the first bit. For CH2, the "COUNT" data is ANDed with 0x02 and rotated right once. The remaining channel data undergoes an ANDing operation with 0x03, 0x04, and so on and rotation twice, thrice, and so on.

This method of converting data to a single bit for eight independent channels helps to monitor the variation in eight in-

dependent signals with respect to time. The handler routine immediately records any change in any one of the bits. The software increments variables I, j, k, l, m, n, p, and q if it detects a change in state of the bits in the corresponding channels 1 through 8. The completion of the 20th TICKER represents a time interval of 1 sec. The counts in variables I through p are a measure of frequency in channels 1 through 8, respectively. This method allows frequency measurement to 100 kHz. In the software example in **Listing 1**, the routine acquires 32,000 samples ("COUNT" from the LPT1 port); hence, for 1 sec and 20 TICKERs it is possible to acquire 6×10^5 samples with a 200-MHz Pentium system. The sampling variable, a, should not exceed the time of occurrence of INT 1CH. In this example, a is 32,000. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2620.

Figure 1



A PC's special BIOS interrupt measures eight independent frequencies.

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

LISTING 1—BIOS-INTERRUPT HANDLER ROUTINE

```
/* Program which Measures Eight independent frequencies
   through PC's LPT port using Special BIOS Interrupt.
Author: J.Jayapandian, Materials Science Division, IGCAR,
                Kalpakkam. Tamil Nadu.  INDIA. */

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <time.h>

#define INTRTIMER  0x1C  /* Timer Interrupt */
#define OUT_PORT   0X378   /* Out port address of LPT1 */
#define CTRL_PORT 0X37A   /* Control port address of LPT1

/*--------------------GLOBAL VARIABLES------------------/
static int COUNT,NEW_COUNT,TICKER,CH1,CH2,CH3,CH4,CH5,
                                        CH6,CH7,CH8;
static int PRE_COUNT;
unsigned int i,j,k,l,m,n,p,q;
int a;

void interrupt (*timerhandler)();
void interrupt COUNTHANDLER();

void interrupt COUNTHANDLER()
{
disable();
     ++TICKER;
      for (a = 0; a < 32000; a++) /* Input Sensing
                                  Loop for an occurrence
                                  of INT 1CH  interrupt */
          {
          COUNT = inportb(OUT_PORT);
          NEW_COUNT = (COUNT ^ PRE_COUNT);
          CH1 = NEW_COUNT & 0x01;
          CH2 = (NEW_COUNT & 0x02)>>1;
          CH3 = (NEW_COUNT & 0x04)>>2;
          CH4 = (NEW_COUNT & 0x08)>>3;
          CH5 = (NEW_COUNT & 0x10)>>4;
          CH6 = (NEW_COUNT & 0x20)>>5;
          CH7 = (NEW_COUNT & 0x40)>>6;
          CH8 = (NEW_COUNT & 0x80)>>7;
/* Monitoring for the change of state in each channel */
          if (CH1 != 0) i++;
                                                (P.T.O)
          if (CH2 != 0) j++;
          if (CH3 != 0) k++;
          if (CH4 != 0) l++;
          if (CH5 != 0) m++;
          if (CH6 != 0) n++;
          if (CH7 != 0) p++;
          if (CH8 != 0) q++;
          PRE_COUNT = COUNT;
          }
enable();
}              /* END OF COUNTHANDLER */

void INSTALLCOUNTHANDLER()
{
     disable();
     timerhandler = getvect(INTRTIMER);
     setvect(INTRTIMER,COUNTHANDLER);
     enable();
}
void CLEARCOUNTHANDLER()
{
     disable();
     setvect(INTRTIMER,timerhandler);
     enable();
}

void main(void)
{
     clrscr();
     outportb(CTRL_PORT,0x01);
     outportb(OUT_PORT,0xff); /* this command is required
                              for initializing all 8-bits
                              in the LPT to high for
                              sensing the change of state
                              from high-to-low */
     INSTALLCOUNTHANDLER();
      while(TICKER != 21)  /* 1 tick in 0.054945 sec;
                           20t =>1.098s;120t =>1min */
      printf("Freq. in Ch1: %d  Ch2: %d  Ch3: %d Ch4: %d
                        Ch5: %d Ch6: %d Ch7: %d Ch8: %d\n",
                                i,j,k,l,m,n,p,q);
     CLEARCOUNTHANDLER();
     getch();
     return;
}
      /* ----------------End of Program ----------------*/
```