
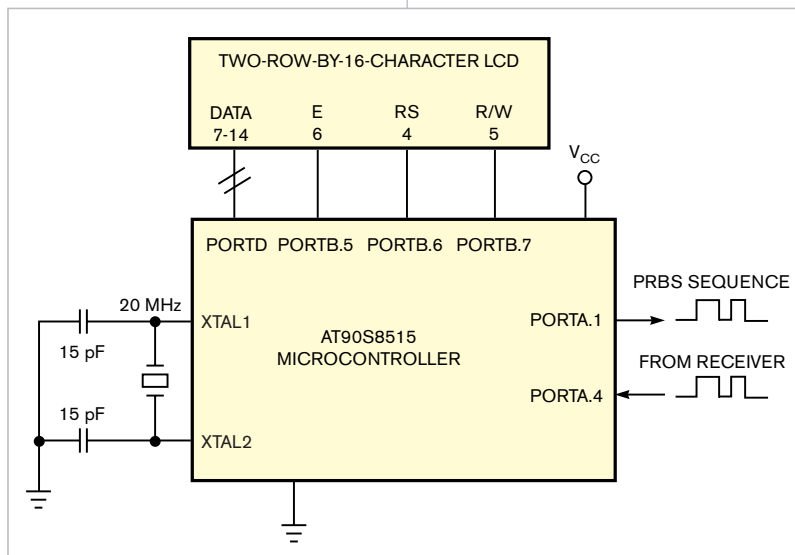


## Low-cost BER tester measures errors in low-data-rate applications

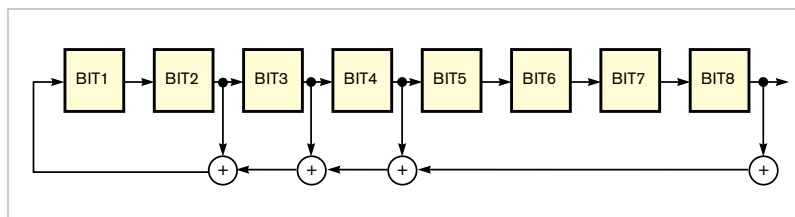
Cedric Mélange, Johan Bauwelinck, Jo Pletinckx, and Jan Vandewege, Ghent University, Ghent, Belgium

 You don't need an expensive pattern generator to produce a PRBS (pseudorandom-bit-sequence) signal for making elementary BER (bit-error-rate) measurements in low-data-rate continuous-transmission systems (Reference 1). You also need not spend time programming on a computer to

compare sent and received data patterns. Moreover, most professional BER-measurement equipment doesn't cover lower bit rates. This Design Idea offers a simple, low-cost alternative that can accommodate data rates as high as 20 kbps. The system tests a 10-kbps transceiver in low-power sensor net-



**Figure 1** You can assemble a bit-error-rate-measurement circuit from only a few components.



**Figure 2** A linear-feedback-shift register generates a pseudorandom bit sequence.

### DIs Inside

**124** DMA eases CPU's workload for waveform generation

**128** Bipolar current source maintains high output impedance at high frequencies

► What are your design problems and solutions? Publish them here and receive \$150! Send your Design Ideas to [edndesignideas@reedbusiness.com](mailto:edndesignideas@reedbusiness.com).

works. The pattern generator, a Hewlett-Packard ([www.hp.com](http://www.hp.com)) E1401B, can produce PRBS streams of only 150 kbps and higher.

An Atmel ([www.atmel.com](http://www.atmel.com)) AVR microcontroller creates a PRBS signal and compares the generated output stream with received data bits (Figure 1). After sending 1 million bits, the system displays the number of badly received bits on a two-row-by-16-character LCD. You can program the unit to transmit longer sequences of bits; however, doing so significantly increases the measurement time. Many low-cost or free development tools are available for AVR microcontrollers. This Design Idea uses an assembler and a serial programmer (references 2 and 3).

The design uses an 8-bit Fibonacci-type LFSR (linear-feedback-shift register) to produce the PRBS stream. The basic design includes a serial-shift register with modulo-2 addition using XOR instructions (Figure 2). You select the feedback taps' position to obtain a maximal-length sequence that has a period of  $2^8 - 1$  bits. Additional LFSR designs of different lengths and optimal feedback taps are also available (Reference 4). You can easily adapt the software in Listing 1, which is available for downloading at [www.edn.com/](http://www.edn.com/)

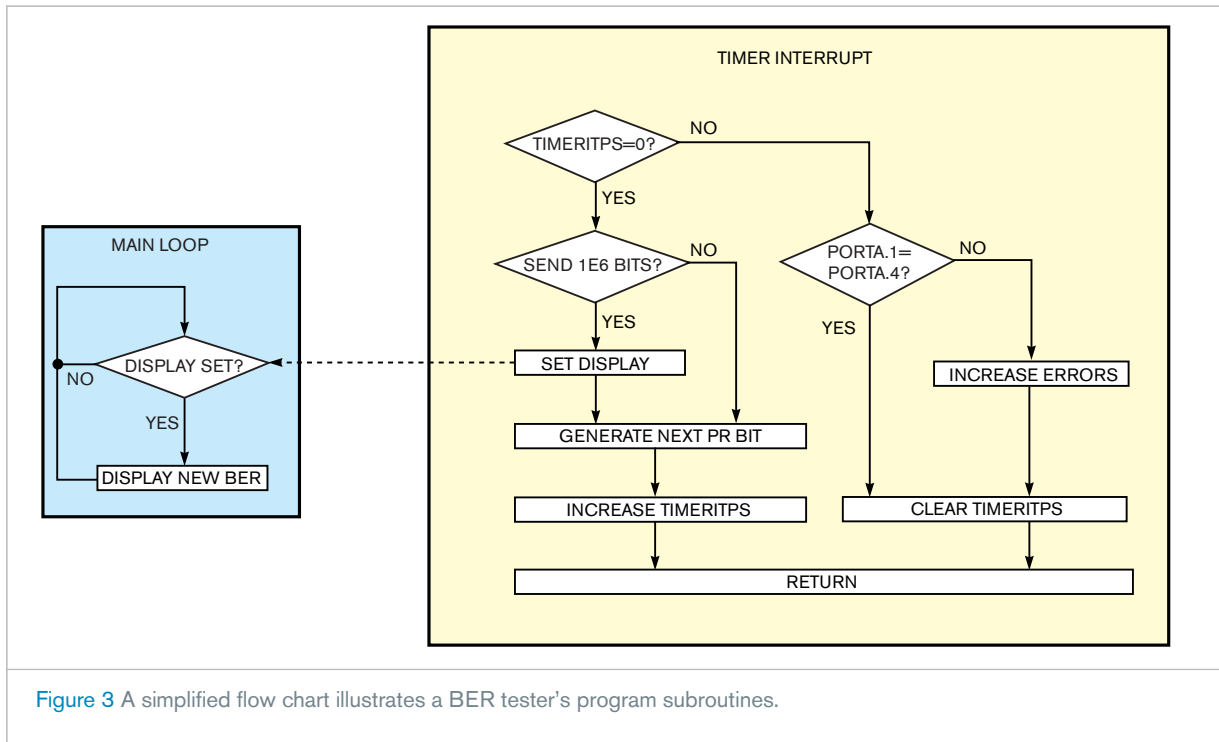


Figure 3 A simplified flow chart illustrates a BER tester's program subroutines.

051205di1 to produce PRBS signals with longer periods. A simplified flow chart of the assembler program is written for the AVR microcontroller (Figure 3).

The generated bit sequence appears at Pin Port A1, which you connect to a transmitter that's suitable for the system under test. Connect the digital out-

put of a convenient receiver to Pin Port A4. The processor compares the received input with the output at Port A1 between two "send" bits. When the bits sent and received don't match, the number of displayed errors increases. If the system exhibits throughput delay, you need to modify the software to cope with the delay. **EDN**

## REFERENCES

- 1 <http://intec.ugent.be/design/>
- 2 AVRStudio 4, [www.atmel.com](http://www.atmel.com).
- 3 SP12, [www.xs4all.nl/~sbolt/e-spider\\_prog.html](http://www.xs4all.nl/~sbolt/e-spider_prog.html).
- 4 [www.newwaveinstruments.com/resources/articles/m\\_sequence\\_linear\\_feedback\\_shift\\_register\\_lfsr.htm](http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm).

## DMA eases CPU's workload for waveform generation

Mike Mitchell, Texas Instruments, Houston, TX

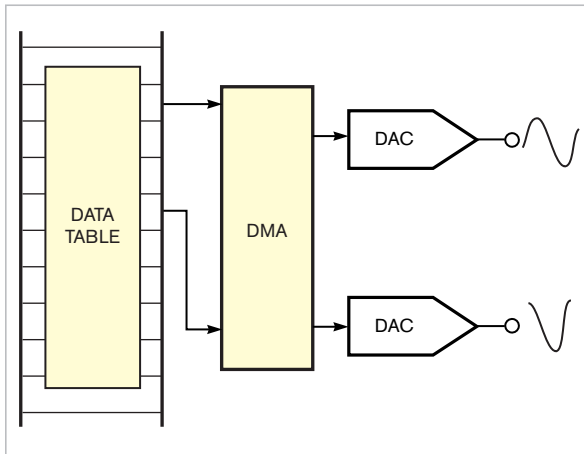
To generate analog voltages and waveforms, embedded systems often require one or more embedded or external DACs. To produce an analog voltage, the CPU must write the desired output value to the DAC at the appropriate time, a task that a timer-generated interrupt applied to the CPU usually initiates. In applications in which the DAC generates a periodic waveform, the CPU reads the next value from the table, sends it to the DAC, increments a table pointer, and

checks for table boundaries to determine when to reset the table pointer.

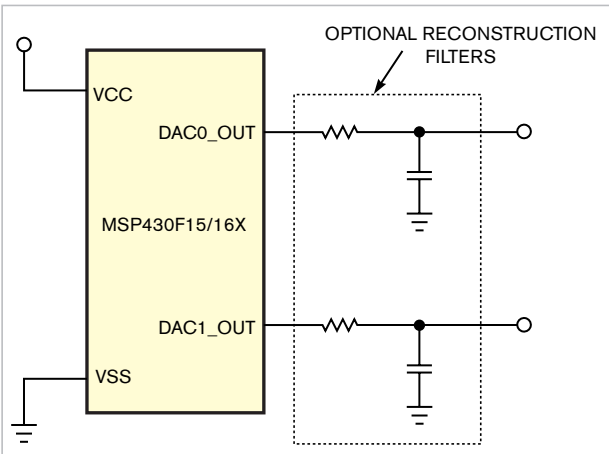
Writing the periodic values to the DAC to maintain the output waveform requires CPU overhead, which varies depending on the data table's length, the output waveform's frequency, and the CPU's operating frequency. For example, using 32 data points per period to generate a 1-kHz sine wave requires the CPU to service 32,000 interrupts/sec. If the application requires a second analog output wave-

form, the CPU's loading increases, and updating both DACs within the required interrupt-service time may be impossible.

To calculate CPU loading, you need to know the length and the context-switching overhead of the ISR (interrupt-service routine). For the MSP430 processor, the ISR's overhead consumes 11 cycles, but the ISR's length depends upon how it is written. The assembly code in Listing 1, available at the Web version of this Design Idea at [www.edn.com/051205di2](http://www.edn.com/051205di2), uses the fewest cycles to implement periodic waveform generation using one or two DACs. For a typical 1-MHz MSP430 CPU-instruction rate, serving 32,000 interrupts/sec leaves 1 million/32,000=



**Figure 1** Using DMA, tabulated data values on their way to waveform-generating DACs avoid handling by a system's processor.

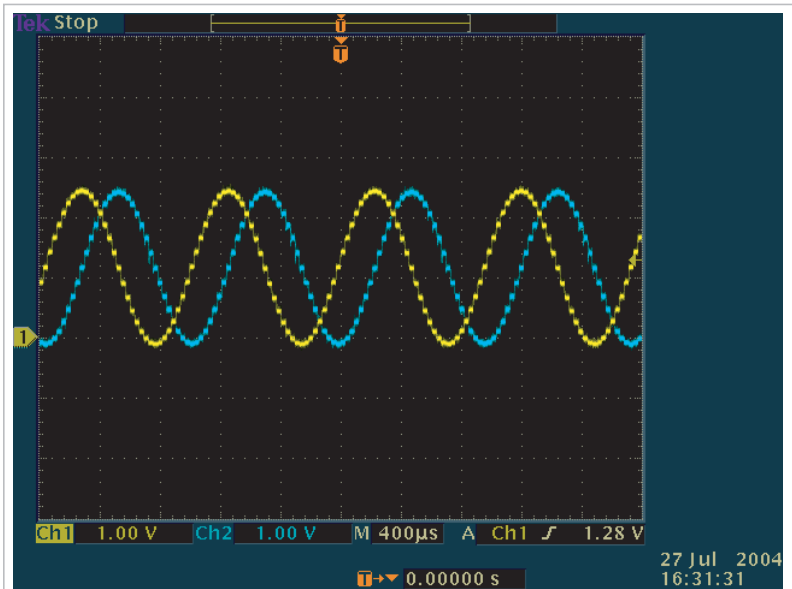


**Figure 2** Optional lowpass filters on the DACs' outputs remove switching artifacts from the sine and the cosine waveforms.

31.25 CPU-instruction cycles between interrupts. An ISR requiring 18 cycles—that is,  $18/31.25=57.6$ —represents a 57.6% CPU load. Supporting two DACs requires 23 cycles—that is,  $23/31.25=73.6$ —and imposes a 73.6% CPU load. Increasing the MSP430's clock rate to its maximum 8 MHz reduces the CPU loading to 7.2 and 9.2%, respectively.

The required CPU load imposes limits not only on other tasks that the application may demand, but also on the waveform's maximum frequency. For example, a CPU operating at 100% CPU loading and an instruction rate of 1 MHz can generate a single waveform with a maximum frequency of approximately 1.73 kHz or two waveforms with a maximum frequency of approximately 1.35 kHz each. Raising the instruction rate to 8 MHz increases the respective maximum frequencies to approximately 13.9 and 10.9 kHz.

However, the MSP430F15x/16x family of devices includes a multi-channel-DMA controller that can move data from one location to another without CPU intervention (Figure 1). In a waveform-generation application, the DMA controller moves data from the data table to the two DACs, significantly reducing the necessary CPU overhead to produce the waveforms. You can configure each of the



**Figure 3** A phase shift occurs between the sine and the cosine output waveforms; removing the lowpass filters in Figure 2 reveals switching artifacts.

DMA controller's three separate and independent channels to move a value from any address to any other address. In this example, one data table contains values for both the sine and the cosine waves, and two of the DMA channels simply access different parts of the table to form the sine and the cosine outputs. In addition, each DMA channel can independently increment its source or destination address. For this application, each DMA channel increments

its source address, but the destination addresses of the respective DAC data registers always remain the same.

You can reconfigure each controller's preset number of DMA transfers. When either DMA channel has transferred its programmed number of data values, it begins the next data transfer from its originally programmed source address. In effect, each DMA channel treats its portion of the data table as a circular buffer to gener-

ate a periodic waveform. Although DMA transfers do not involve the CPU, each transfer does consume two CPU clock cycles, which delays CPU code execution and thus introduces overhead. For the single-waveform example, using DMA transfers consumes two clock cycles for each DAC update instead of the 18 cycles necessary when using only the CPU. Thus, for a CPU clock rate of 1 MHz, using DMA reduces the effective CPU loading from 57.6% to 6.4% and increases the possible maximum output frequency from approximately 1.73 kHz to approximately 15.6 kHz. For an 8-MHz clock rate, using DMA reduces single-waveform CPU loading from 7.2% to 0.8%.

Generating two waveforms requires two DMA transfers or four clock cycles. For the two-waveform example, DMA

reduces loading from 73.6% to 12.8% for a 1-MHz instruction rate, and from 9.2% to 1.6% for an 8-MHz rate. For the 1-MHz instruction rate, using DMA increases the possible maximum frequency for two waveforms from approximately 1.35 kHz to approximately 7.8 kHz.

After initialization, each DMA controller simply performs its duties with no further intervention other than receiving a trigger to move the data value. In this example, each DAC's interrupt flag serves as a trigger for its respective DMA channel. When you use dual DACs, you can load each DAC with the next value of waveshape data before it's required and then simultaneously trigger both DACs using a timer to avoid introducing delays that manifest themselves as output har-

monic distortion. **Listing 2**, also available at [www.edn.com/051205di2](http://www.edn.com/051205di2), contains software that generates sine and cosine waves and illustrates the DMA channels' independent operation apart from the CPU. Note that, after initialization of DMA channels and other device-specific peripherals, no further CPU activity occurs.

**Figure 2** shows a partial schematic of the DACs' outputs. Depending on the application, you may need to add optional resistance-capacitance lowpass filters at the DACs' outputs. Select values for the resistor and capacitor in each filter to produce a pole in the filter response at the desired output frequencies. Note that the oscilloscope photo in **Figure 3** was taken with filters removed to show the DAC outputs' unfiltered waveforms. **EDN**

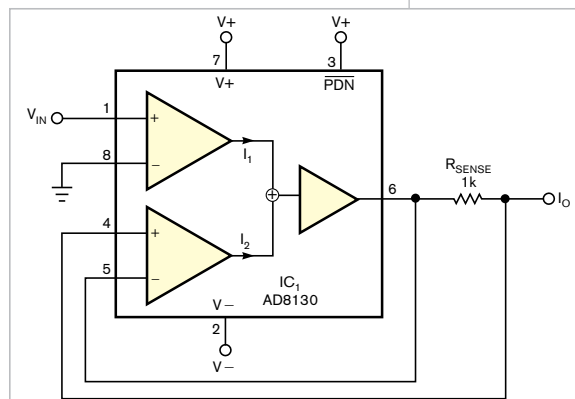
## Bipolar current source maintains high output impedance at high frequencies

Alex Birkett, University College London, Hospital National Health Service, London, UK

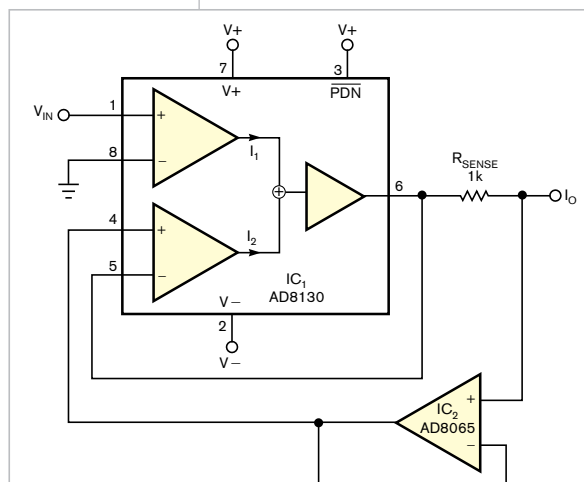
Traditional current sources and voltage-to-current converters based on instrumentation and operational amplifiers offer high output

impedances at low frequencies because of the amplifiers' good low-frequency CMRR (common-mode-rejection ratios). At higher frequencies, decreasing CMRR, inherent output capacitances, and slew-

rate limitations prevent realization of high-quality current sources. Two 200-MHz line-receiver/amplifier ICs from Analog Devices, the AD8129 and AD8130, offer differential inputs and outstanding CMRR, making them strong candidates for building high-frequency constant-current sources. Al-



**Figure 1** The 200-MHz AD8130 differential-input line receiver/amplifier can serve as a basic building block in a high-frequency-capable current source.

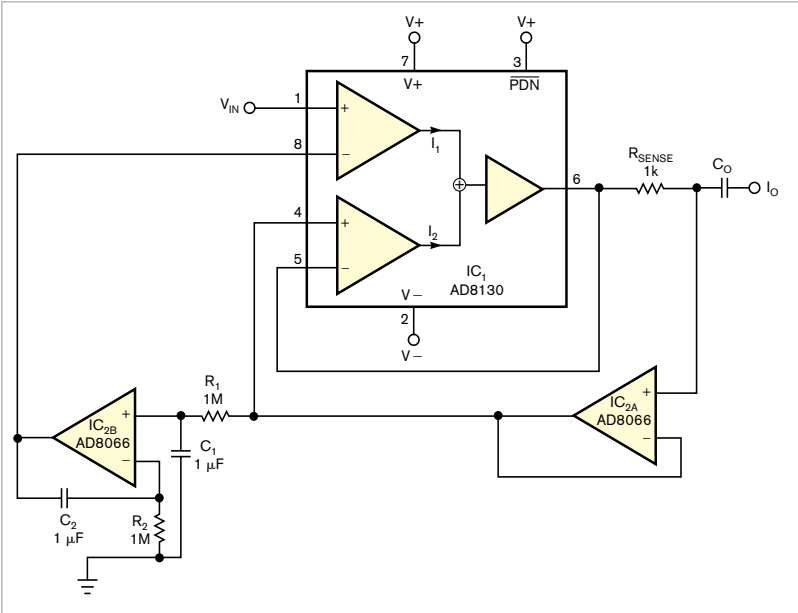


**Figure 2** Adding an Analog Devices AD8065 buffer amplifier, IC<sub>2</sub>, isolates the current-sampling resistor R<sub>SENSE</sub> and reduces errors that IC<sub>1</sub>'s input-bias current contributes.

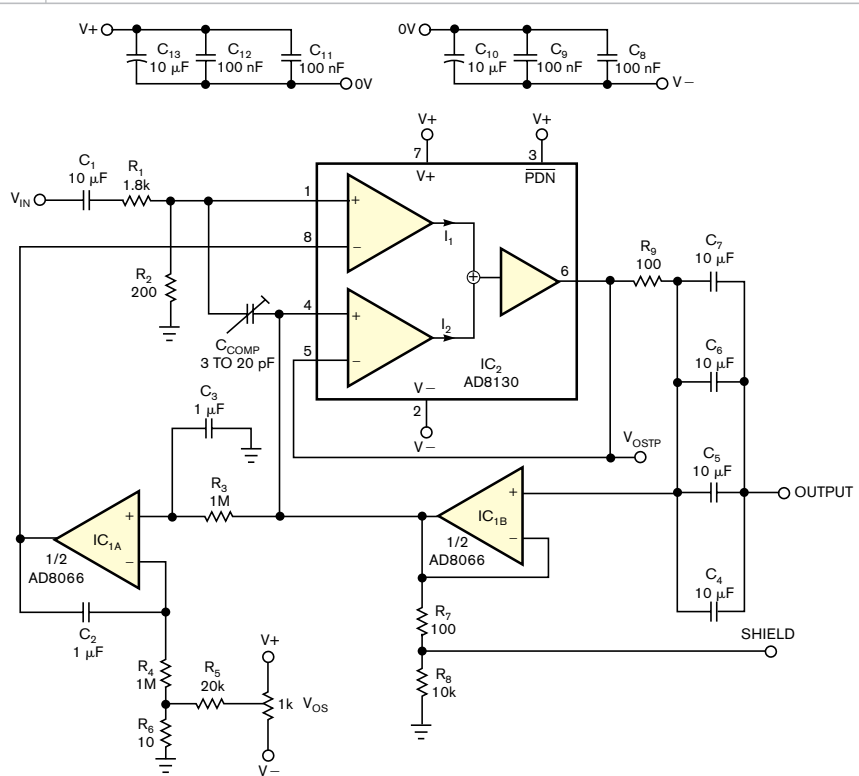
though the circuit in **Figure 1** provides a good starting point, the AD8130's relatively high input bias current can affect output-current accuracy at low current levels.

To overcome the problem, you can add a unity-gain buffer,  $IC_{2A}$ , to isolate the current-sense resistor (**Figure 2**). In addition, you can use the buffer amplifier to measure the load voltage and bootstrap the output cable's capacitance. The circuit presents an output impedance of about 500 k $\Omega$  at 1 MHz and a current-compliance range of 0 to  $\pm 3V$  using  $\pm 5V$  power supplies.

Current sources that have capacitance-coupled loads benefit from a dc servo loop to stabilize the circuit's operating point (**Figure 3**). The value of output-coupling capacitor  $C_O$  depends on the desired low-frequency roll-off characteristic. Further improvements of the basic circuit enable compensation of output capacitance and increase the circuit's output impedance. A small, adjustable feedback capacitor,  $C_{COMP}$ , that's approximately one-half of the output's stray capacitances provides feedforward compensation and further reduces the effects of stray capacitance at the output (**Figure 4**). To prevent oscillation, the cable's shield-driver circuit's gain should be slightly less than unity. Note that reducing the output-current-sense resistor,  $R_S$ , to 100 $\Omega$  compensates for the input attenuator formed by  $R_1$  and  $R_2$  and maintains a 1-mA/V characteristic. This voltage-to-current source's frequency range spans 20 Hz to 10 MHz. For best results, use high-frequency circuit-layout and power-supply-bypassing methods.**EDN**



**Figure 3** For an ac-coupled current output, add a dc-stabilization loop,  $IC_{2A}$  and  $IC_{2B}$ .



**Figure 4** The complete circuit includes trimmer capacitor  $C_{COMP}$ , which compensates for stray capacitances in the circuit's packaged layout. Also, note wideband treatment of power-supply bypass capacitors.

**MORE AT EDN.COM**

**+** We now have a page exclusively devoted to Design Ideas on our Web site. Check it out at [www.edn.com/designideas](http://www.edn.com/designideas).