Edited by Bill Travis

Input filter prevents instrumentation-amp RF-rectification errors

Charles Kitchin, Lew Counts, and Moshe Gerstenhaber, Analog Devices, Wilmington, MA

Instrumentation amplifiers serve in a variety of applications that need to extract a weak differential signal from large common-mode noise or interference. However, designers often overlook the potential problem of RF rectification inside the instrumentation amplifier. The amplifier's common-mode rejection normally greatly reduces common-mode signals at an instrumentation amplifier's input. Unfortunately, RF rectification still occurs, because even the best instrumentation amplifiers have virtually no common-mode rejection at frequencies higher than 20 kHz. The amplifier's input stage may rectify a strong RF signal and then appear as a dc-offset error. Once the input stage rectifies the signal, no amount of low-pass-filtering at the instrumentation amplifier's output can remove the error. Finally, if the RF interference is intermittent, measurement errors may go undetected. The best practical solution to this problem is to provide RF attenuation ahead of the instrumentation amplifier by using a differential lowpass filter. The filter needs to remove as much RF energy as possible from the input lines, preserve the ac signal's "balance" between each line and ground (common), and maintain a high enough input impedance over the measurement bandwidth to avoid loading the signal source. **Figure 1** provides a basic building block for a wide range of differential RFI filters.

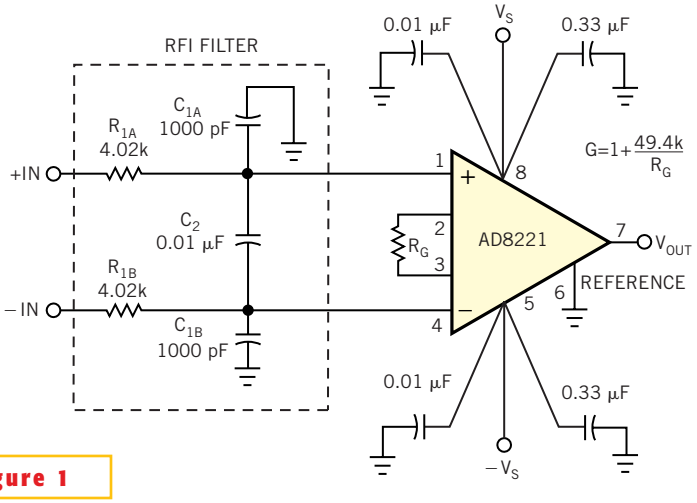


Figure 1

This lowpass-filter circuit prevents RF-rectification errors in instrumentation amplifiers.

The component values are typical of those for the latest generation of instrumentation amplifiers, such as the AD-8221, which has a typical −3-dB bandwidth of 1 MHz and a typical voltage noise level of 7 nV/$\sqrt{\text{Hz}}$. In addition to RFI suppression, the filter also provides additional input-overload protection; resistors R_{1A} and R_{1B} help isolate the instrumentation amplifier's input circuitry from the external signal source. **Figure 2** shows a simplified version of the RFI circuit. It reveals that the filter forms a bridge circuit whose output appears across the instrumentation amplifier's input pins. Because of this connection, any mismatch between the time constants of C_{1A}/R_{1A} and C_{1B}/R_{1B} unbalance the bridge and reduce high-frequency common-mode rejection. Therefore, resistors R_{1A} and R_{1B} and capacitors C_{1A} and C_{1B} should always be equal. C_2 connects across the "bridge output" so that C_2 is effectively in parallel with the series combination of C_{1A} and C_{1B}. Thus connected, C_2 effectively reduces any ac common-mode-rejection errors from mismatching. For example, making C_2 10 times larger than C_1 provides a 20-times reduction in common-mode-rejection errors arising from C_{1A}/C_{1B} mismatch. Note that the filter does not affect dc common-mode rejection.

The RFI filter has differential and common-mode bandwidths. The differential bandwidth defines the frequency response of the filter with a differential input signal applied between the circuit's two inputs, +IN and −IN. The sum of the two equal-value input resistors, R_{1A} and R_{1B}, and the differential capacitance,

Publish your Design Idea in *EDN*. See the What's Up section at www.edn.com.

which is C_2 in parallel with the series combination of C_{1A} and C_{1B}, establish this RC time constant. The −3-dB differential bandwidth of this filter is equal to $BW_{DIFF}=[1/(2\pi R(2C_2+C_1))]$. The common-mode bandwidth defines what a common-mode RF signal "sees" between the two inputs tied together and ground. C_2 does not affect the bandwidth of the common-mode RF signal, because this capacitor connects *between* the two inputs, helping to keep them at the same RF-signal level. Therefore, the parallel impedance of the two RC networks (R_{1A}/C_{1A} and R_{1B}/C_{1B}) to ground sets common-mode bandwidth. The −3-dB common-mode bandwidth is equal to $BW_{CM}=1/(2\pi R_1 C_1)$.

Using the circuit of **Figure 1**, with a C_2 value of 0.01 µF, the −3-dB differential-signal bandwidth is approximately 1900 Hz. When operating at a gain of 5, the circuit has measured dc-offset shift over a frequency range of 10 Hz to 20 MHz of less than 6 µV referred to the input. At unity gain, there is no measurable dc-offset shift. Some instrumentation amplifiers are more prone to RF rectification than others and may need a more robust filter. A micropower instrumentation amplifier, such as the AD627, with its low-input-stage operating current, is a good example. The simple expedient of increasing the value of the two input resistors, R_{1A}/R_{1B}, that of capacitor C_2, or both can provide further RF attenuation at the expense of reduced signal bandwidth. Some steps for selecting RFI-filter component values follow:

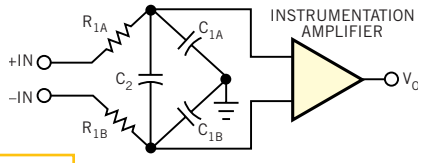


Figure 2

Capacitor C_2 shunts C_{1A}/C_{1B} and reduces ac common-mode-rejection errors arising from component mismatch.

1. Decide on the value of the two series resistors and ensure that the previous circuitry can adequately drive this impedance. With typical values of 2 to 10 kΩ, these resistors should not contribute more noise than that of the instrumentation amplifier itself. Using a pair of 2-kΩ resistors adds Johnson noise of 8 $nV/\sqrt{Hz}$. This figure increases to 11 $nV/\sqrt{Hz}$ with 4-kΩ resistors and 18 $nV/\sqrt{Hz}$ with 10-kΩ resistors.

2. Select an appropriate value for capacitor C_2, which sets the filter's differential (signal) bandwidth. Set this value as low as possible without attenuating the input signal. A differential bandwidth of 10 times the highest signal frequency is usually adequate.

3. Select values for capacitors C_{1A} and C_{1B}, which set the common-mode bandwidth. For decent ac common-mode rejection, these capacitors should have values 10% or lower of the value of C_2. The common-mode bandwidth should always be less than 10% of the instrumentation amplifier's bandwidth at unity gain.

You should build the RFI filter using a pc board with a ground plane on both sides. All component leads should be as short as possible. Resistors R_1 and R_2 can be common 1% metal-film units. However, all three capacitors need to be reasonably high-Q, low-loss components. Capacitors C_{1A} and C_{1B} need to be ±5%-tolerance devices to avoid degrading the circuit's common-mode rejection. Good choices are the traditional 5% silver micas, miniature micas, or the new Panasonic ±2% PPS film capacitors (Digi-key part number PS1H102G-ND).□

White-LED driver backlights LCD and keypad

Fabien Franc, Catalyst Semiconductor, Sunnyvale, CA

Designers widely use white LEDs to backlight color LCDs and keypads in handheld devices, such as cell phones, MP3 players, GPS navigators, and PDAs. Their spectrum and brightness represent near-ideal light sources. One possible configuration for a phone or a phone/PDA combination is to have a group of three LEDs to light the display and six LEDs for the keypad. **Figure 1** shows a method for driving all the LEDs with a single driver IC from Catalyst Semiconductor (www.catsemi.com), the CAT32. Power comes from a single lithium-ion battery cell. A FET switch can independently turn off the group of six LEDs. The shutdown input, $\overline{SHDN}$, on the CAT32 turns off all the LEDs. LED brightness is a direct function of the current running through

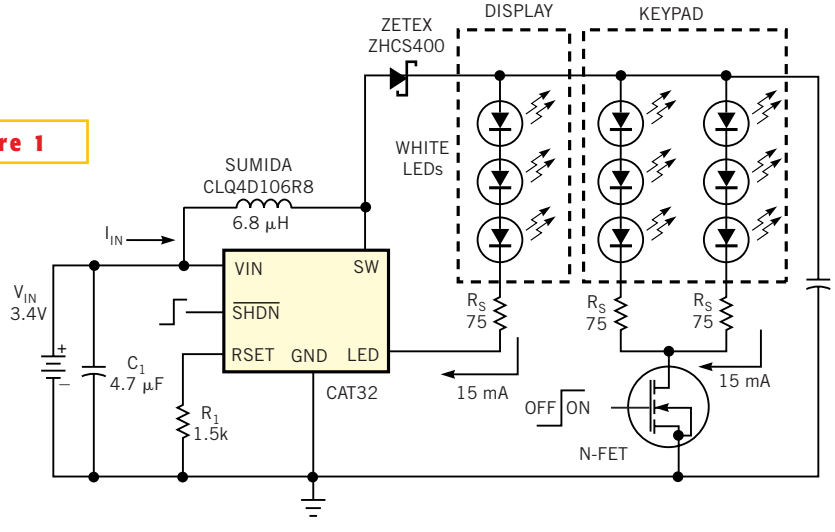
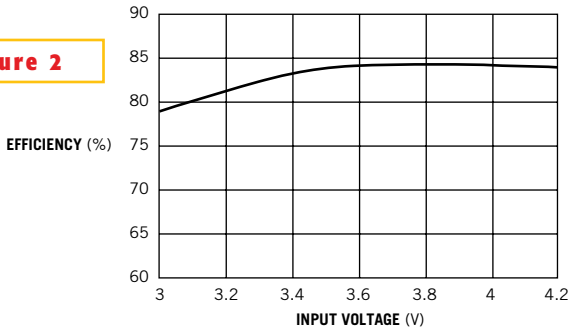


Figure 1

A single IC boosts the battery voltage to drive a total of nine LEDs.

the LED, and typical values are 10 to 20 mA for white LEDs. The CAT32 regulates a constant current through the LED pin. Resistor R_1 connected to the RSET pin adjusts the LED current.

The driver is a step-up converter using an inductor to boost the voltage, such as that from a lithium-ion battery, and bias multiple LEDs in series. The R_S resistors in series with the LEDs balance out the current between the groups of LEDs. The LED forward voltage, typically around 3.5V, experiences some variation from one LED to the other for a given bias current, and the series resistors also compensate for that situation. By using series resistors with values of 75Ω, a current of approximately 15 mA flows in all LEDs. The display uses Nichia (www.nichia.com) NSCW335 side-view LEDs, and top-view Nichia NSCW100 LEDs backlight the keypad. You define efficiency as the ratio of the power dissipated in the LEDs, including the power in the series resistors but not including the loss in the Schottky diode, to the total input power. **Figure 2** shows the efficiency of the circuit for a load of 15-mA current in the LEDs and for supply voltage ranging from 3 and 4.2V, corresponding to the charge-discharge cycle of lithium-ion batteries. The expression for efficiency is as follows:



This curve shows the efficiency of the circuit in Figure 1 for 15-mA LED current and 3 to 4.2V battery voltage.

$$\text{EFFICIENCY} = \frac{P_{OUT}}{P_{IN}} = \frac{\sum_{i=1}^{9} V_{LEDi} \times I_{LEDi} + \sum_{j=1}^{3} R_{Sj} \times I_i^2}{V_{IN} \times I_{IN}},$$

where V_{LED} is the voltage across the LED, I_{LED} is the LED current, R_S is the series resistor, and I_{IN} is the current from the input supply. The efficiency is approximately 84% for a 3.6V supply. If you now consider only the power the LEDs dissipate, excluding all the other losses, the following formula gives the net efficiency (approximately 75% with a 3.6V supply):

$$\text{NET EFFICIENCY} = \frac{P_{OUT}}{P_{IN}} = \frac{\sum_{i=1}^{9} V_{LEDi} \times I_{LED_i}}{V_{IN} \times I_{IN}}.$$

Microcontroller or DSP circuit controls on/off function

Dirk Gehrke, Texas Instruments, Freising, Germany, and Frans Ravn, ChemoMetec A/S, Alleroed, Denmark

In many applications, a single on/off button switches the power supply on or off. Usually, the system switches off regardless of the processing function or workload the microcontroller or DSP circuit is currently handling. The small circuit in **Figure 1** is intended to make the microcontroller or DSP circuit the master over this on/off function. The circuit allows the microcontroller or DSP circuit to take ownership over the on/off function. Thus, the system can do whatever has to be done, such as processing data, storing data, and so forth, before it issues the command to shut down the supply. **Figure 2** shows the timing diagram. The switch connects directly to the dc supply or battery, though the end application is disconnected from the supply. Thus, all the capacitors discharge. The most difficult task is to get the two D flip-flops in **Figure 1** into the desired "off" configuration.

The inverter, IC_{1B}, in **Figure 1** resets the D flip-flop, IC_{2B}, via the diode, D_4. The RC network comprising R_{10}, R_{11}, and C_4, connected to the input of the inverter, IC_{1B}, generates a delay time of approximately 4.7 msec. This interval ensures that this D flip-flop is released from reset after a delay time of approximately 4.7 msec. When the delay time has passed, reset-input pin R of IC_{2B} changes from high to low potential because R_7 connects to ground. At the same time, output Q of IC_{2B} switches to low level, and $\overline{Q}$ switches to high level. The D input node of IC_{2B} changes as well from high to low potential because it connects to the output pin, $\overline{Q}$. The RC network comprising R_1 and C_3, in conjunction with the inverter, IC_{1A}, generates a delay of approximately 47 msec. This delayed output connects to the set pin of the D flip-flop, IC_{2A}. The set pin holds high for 47 msec before it goes low. The RC network, comprising $R_1$and C_3, along with the inverter, IC_{1A}, generate a 47-msec delay. The delayed output connects to the set pin of the D flip-flop, IC_{2A}. The set pin holds high for 47 msec before it goes low.

After the set pin falls to a low level, the D flip-flop, $IC_{2A,}$ changes its output levels at Q to high and $\overline{Q}$ to low, and feedback from $\overline{Q}$ connects to the D input. The high level at Q ($\overline{SHTDN}$), connected to the enable pin of the dc/dc converter or low-dropout regulator, keeps the system off. The D flip-flop, IC_{2A}, is now in an off condition. From this point on, both D flip-flops are in a known state. The DSP I/O pin is low during this initialization, because the DSP circuit receives no power. R_{15} ensures a low level during the power-up sequence at the DSP I/O pin. The $\overline{SHTDN}$, after its initialization phase, assumes an active-high level. If you connect it to the enable pin of the system dc/dc converter or low-dropout regulator, it keeps the system in an off condition. IC_{1B}'s input changes its level from high to low when you press the pushbutton, because the switch shorts the pullup resistor at the pushbutton node to ground. IC_{1B}'s output changes its level according-

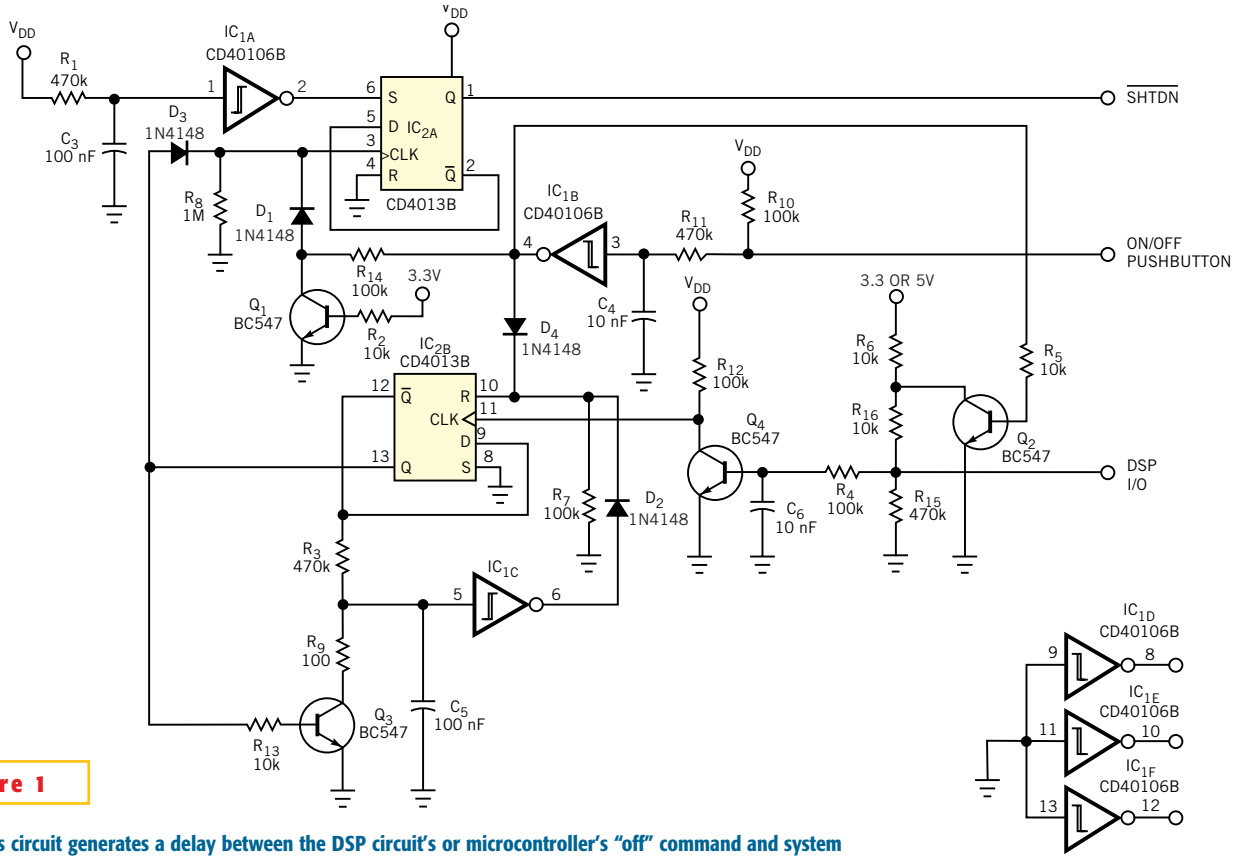


Figure 1

This circuit generates a delay between the DSP circuit's or microcontroller's "off" command and system shutdown.

ly from low to high for that period.

The CLK input of D flip-flop IC_{2A} triggers via R_{14} and D_1, and output Q changes its status from low to high. This state enables the low-dropout regulator or dc/dc converter to start operation. The 3.3 or 5V connected to R_2 supplies transistor Q_1 to change the logic level at the CLK input of D flip-flop IC_{2A}. This action ensures that the system disregards glitches when you press the on/off pushbutton. The DSP I/O pin of the circuit connects to one of the DSP circuit's or microcontroller's I/O pins. You should configure the I/O pin of the DSP circuit or microcontroller as an input pin after power-up and reset release. As long as you press the on/off pushbutton, transistor Q_2 remains on, driving the DSP circuit's I/O pin low. You should program the DSP circuit or microcontroller such that the DSP circuit stops executing code before you release the button, and the DSP I/O pin changes its level from low to high. D flip-flop IC_{2B} again resets via D_4, but this reset does not alter the output because the application is running.

When you again press the on/off pushbutton, the DSP I/O pin assumes a low level. The DSP circuit or microcontroller should now detect this input change and generate an interrupt. This interrupt should initiate a shutdown procedure. D flip-flop IC_{2B} changes, via D_4, to reset mode, so that the toggle signal valid at CLK has no impact on the output status. You now release the on/off pushbutton. D flip-flop IC_{2B} releases from reset after

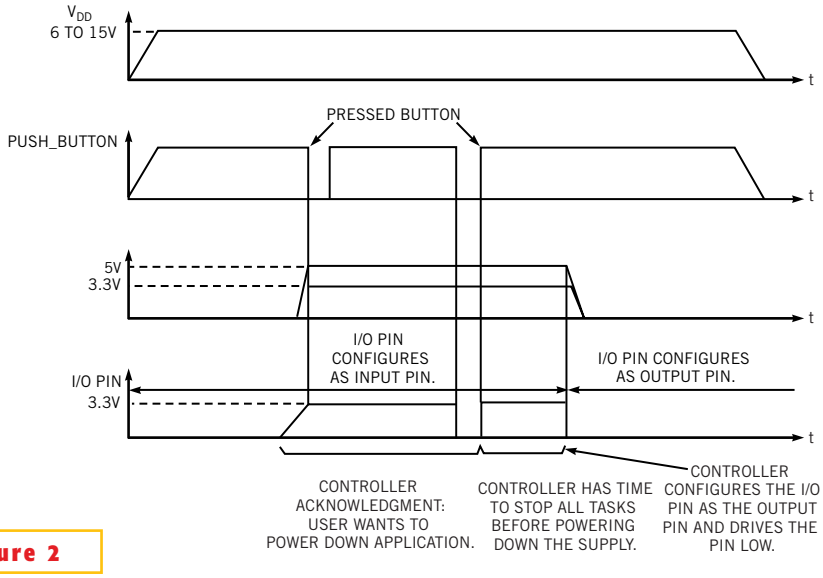


Figure 2

This timing diagram shows that the DSP circuit or microcontroller takes ownership over the system's on/off function to allow time for performing crucial tasks.

approximately 4.7 msec, and Q_2 and Q_4 change levels and maintain a low signal at the CLK pin before the reset delay time passes. The microcontroller or DSP circuit sees a high signal at the I/O pin after this time-out and can start the shutdown procedure.

The DSP circuit or microcontroller now has the time to store any critical data. You must program the microcontroller's or DSP circuit's I/O pin as an output pin set low. Q_4 loses its drive voltage, and the CLK pin of D flip-flop IC_{2B} changes its status from low to high. D flip-flop IC_{2A} changes its output status via the output, Q, and D_3. Q and $\overline{Q}$ change the status of the $\overline{SHTDN}$ pin to low level, and the system shuts down. At the same time, D flip-flop IC_{2B} resets via Q_3 and the comparator IC_{1C}. This reset brings D flip-flop IC_{2B} to the initial state described above before you first pressed the on/off pushbutton.□

Simple power supply fits into small spaces

Raymond Zheng, Linear Technology, Milpitas, CA

THE DEMAND FOR negative-voltage power supplies is increasing with the popularity of applications for portable devices. It can be expensive and relatively complicated to generate a negative supply from a positive input source, especially when the design requires both positive and negative outputs. **Figure 1** shows a simple and cost-effective solution that combines a voltage inverter and a voltage doubler in a single charge-pump circuit. It produces a regulated −5V output and an unregulated 10V output from a 5 to 6V input. The circuit requires only five small, ceramic, surface-mount capacitors and two diodes in addition to the charge-pump IC in an SOT-23 package.

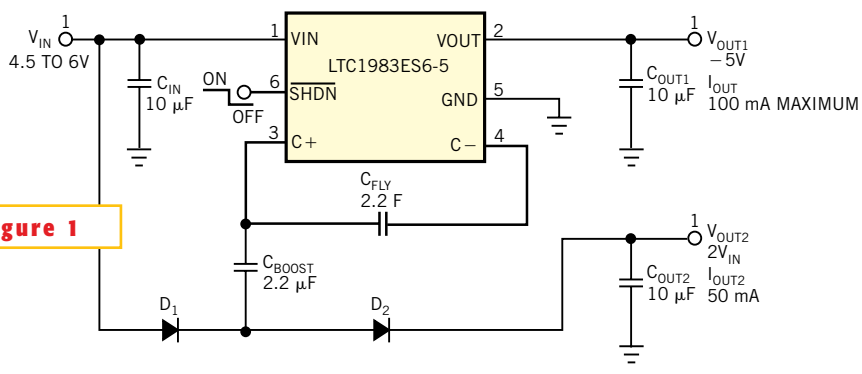


Figure 1

NOTES: C_{IN}, C_{OUT1}, C_{OUT2}: TAIYO YUDEN JMK316BJ106ML.
C_{FLY}: TAIYO YUDEN LMK212BJ225MG.
C_{BOOST}: TAIYO YUDEN EMK316BJ225ML.

This circuit combines a regulated inverter and a voltage doubler.

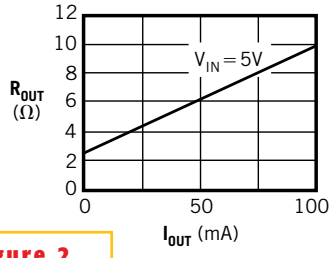


Figure 2

This graphic shows R_{OUT} versus I_{OUT} for the circuit in Figure 1.

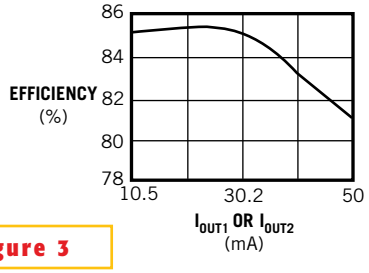


Figure 3

This curve shows efficiency versus either output current.

From a 6V input, the inductorless dc/dc inverter can deliver 100 mA at a regulated −5V (±5%), and the voltage doubler can provide 50 mA at 10.5V with ±7% variation. The inverter's output-voltage regulation follows the relationship $(V_{IN}-5)>(I_{OUT}\times R_{OUT})$; you can determine the values of R_{OUT} and I_{OUT} for V_{IN}=5V from the graph in **Figure 2**. (The R_{OUT} and I_{OUT} values for other V_{IN} values are available in the LTC-1983's data sheet.) If the variables don't meet this inequality condition, the part runs in open-loop mode and acts as a low-output-impedance inverter in which the output voltage is $V_{OUT1}=-[V_{IN}-(I_{OUT}\times R_{OUT})]$. You can define the output voltage of the voltage doubler as $V_{OUT2}=2V_{IN}-2V_D$, where V_D is the forward voltage drop across the diodes. **Figure 3** shows the efficiency of the circuit, which exceeds 81% and peaks at approximately 85%. **Figure 4** shows the inverter's output-voltage regulation versus inverter output current. The IC includes short-circuit and thermal protection.□

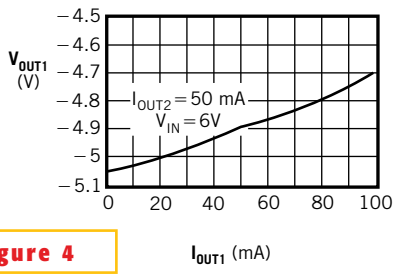


Figure 4

This curve shows output-voltage regulation versus output current.

Amplifier and current source emulate instrumentation amplifier

Dobromir Dobrev, Jet Electronics, Sofia, Bulgaria

THE CLASSIC three- or two-op-amp instrumentation-amplifier circuits are standard ways to amplify a small-amplitude differential signal contaminated with high common-mode noise. In some applications, the signal source is floating with high-series-output impedance and thus requires an appropriate high-input-impedance amplifier. This Design Idea proposes an alternative approach using a simplified amplifier circuit (**Figure 1**). The basic principle is to sense the current in the amplifier input B (I_B) and inject a current of the same value in the amplifier input A (I_A), by combining a virtual-ground transimpedance amplifier (A_1) with a voltage-controlled current source (G_1). Thus, G_1 balances the common- mode interference currents. In addition, the voltage at input B is at virtual-ground potential.

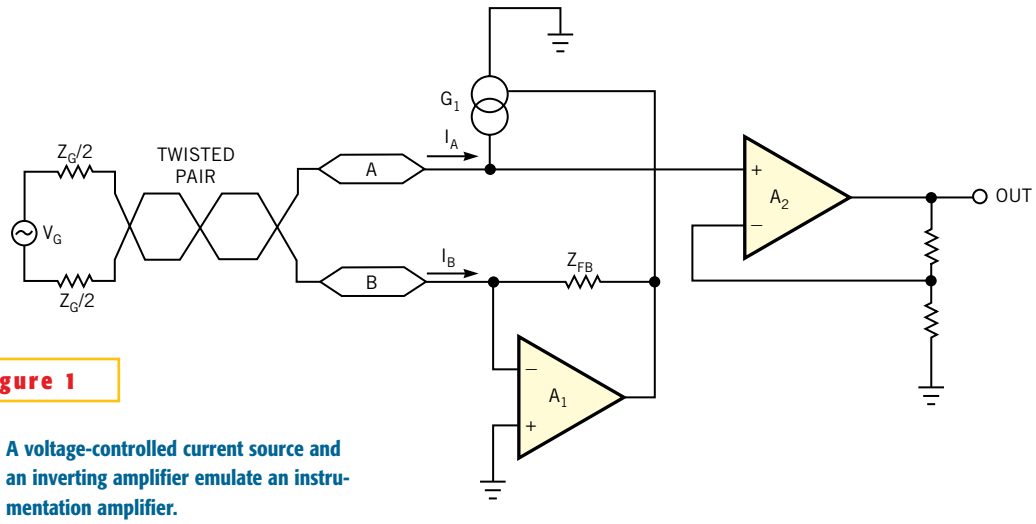


Figure 1

A voltage-controlled current source and an inverting amplifier emulate an instrumentation amplifier.

A practical circuit is a two-electrode biosignal amplifier for electrocardiogram signals (**Figure 2**). IC_{2B} is the transimpedance amplifier. The feedback capacitor, C_{FB}, ensures circuit stability. The INA134 difference amplifier, IC_1, and the op amp, IC_{2A}, make up a high-quality, bidirectional voltage-controlled current source. You could use many similar ICs, such as INA132, 133, 152, 154, 105, or AMP03, for IC_1. The remaining part of the circuit comprises two conventional noninverting stages. The proposed circuit can be useful in many two-wire or two-electrode applications, in which you need to maintain high amplifier input-impedance values.□

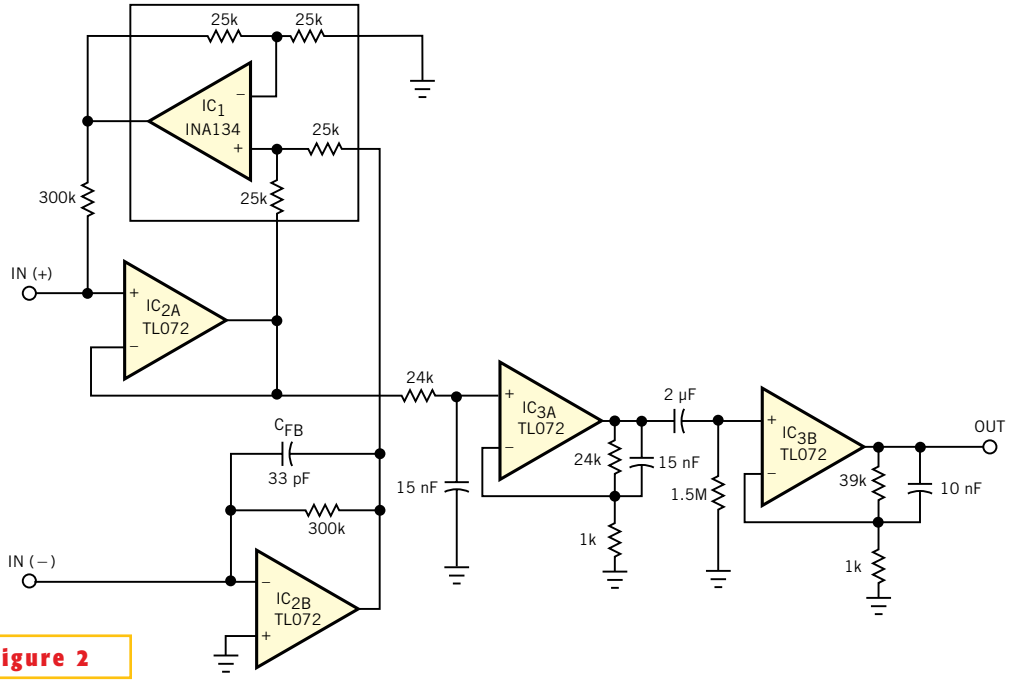


Figure 2

This biosignal amplifier has the high input impedance that medical applications require.

Hardware histogram speeds ADC test

Tony Zizzo Jr, Texas Instruments, Tucson, AZ

Accumulating enough samples to effectively and accurately measure linearity errors in pipeline converters at production test and minimizing test time are difficult enough. Add to that multiple sites, multiple channels, and the expense of multiple custom-instrument resources during periods of low capital expenditure, and engineers must think creatively to increase throughput without sacrificing yield. Many test approaches are beginning to rely more on FPGAs. These ICs' high speeds and flexible usage lend them well to ATE (automated-test-equipment) shortfall workarounds and pare down final test time, which could add up to thousands of dollars in savings, depending on time saved and volume. But development of FPGA designs can be time-consuming and expensive; outsourced designs can cost more than $100,000, depending on complexity. Also, the hardware-histogram capability of many ATE systems is expensive, and you must multiplex it to multiple sites, which reduces its time-saving effectiveness. This Design Idea presents a simple, discrete hardware-histogram circuit module (**Figure 1**), which uses off-the-shelf components, requires no software, is easy to debug, and is easy to multiply to many sites and channels.

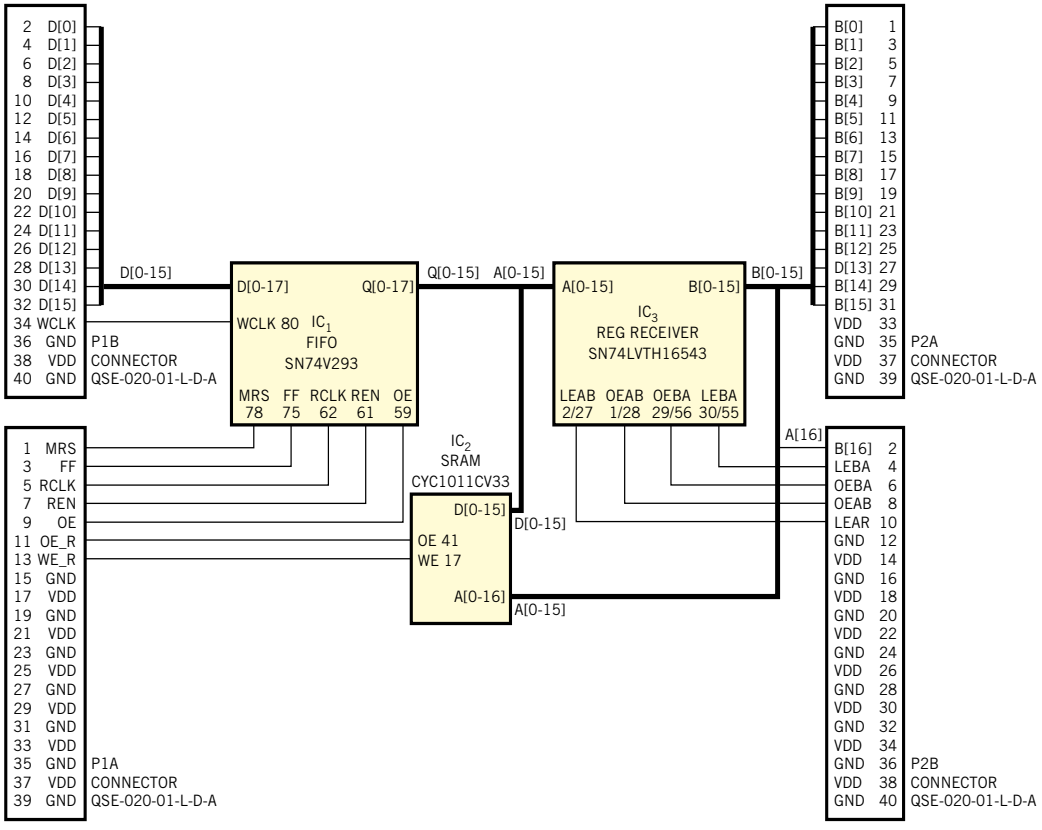


Figure 1

A hardware-based histogram system for testing ADCs uses off-the-shelf components.

The circuit of **Figure 1** shows how you can configure a FIFO memory, a static SRAM, and a bus register to generate fast hardware histograms for linearity tests and still allow straight data acquisition for other tests. The main purpose of the FIFO memory is to collect data at the speed of the DUT (device under test), because, with all the data-swapping that occurs during the histogram routine, the cycle is slower than the DUT. But it also reduces switching noise in the test circuit by keeping the outputs and the other devices "quiet" during the initial acquisition. And, because many high-speed converters require low-jitter sources, the use of the FIFO memory eliminates the need to synchronize the test head with external sources by using the system clock to run the rest of the circuit.

The SRAM stores each bin count in the address whose value is equal to the code being accumulated. The memory is generally underused, because part of the memory holds a look-up table of all practical bin counts. Rather than use an adder or a counter to implement the bin-count function, which would require an extra IC, this circuit implements data+1 by storing each look-up entry in the address whose value is one less than the value stored at that location, not including bit 16. Address-bit 16 keeps the bin counts and look-up table in different columns. When each sample routes from the FIFO memory to the address bits, the FIFO memory's output becomes disabled as the bin count for that code appears on the bus. The bus register's "bus-hold" function conveniently holds the current bin count on the address lines while the

SRAM's address-bit 16 is toggled to access the count look-up table, and the bin count+1 appears on the bus. The bus register latches in the new bin count while the FIFO memory reasserts the sample data.

The SRAM then accesses the bin count (address-bit 16 reset) in write mode when the bus register reasserts the new bin count with the FIFO memory disabled. This cycle plays out until the histogram is complete and the data is read through the bus register to the test head. **Figure 2** shows the timing scheme. Although the FIFO memory has a depth of only 64 kbytes, note that the FIFO memory can be filled multiple times and the histogram can be updated multiple times until the SRAM is cleared. You encounter no overflow, regardless of bus width, as long as no bin count exceeds 16,383. The look-up table can be written before any device is tested, and the bin counts can be reset between tests when the handler is binning. This circuit is particularly useful when your application requires multiple sites or multiple channels, because you can simply plug the inexpensive module into other sites or channels and operate it in parallel.□

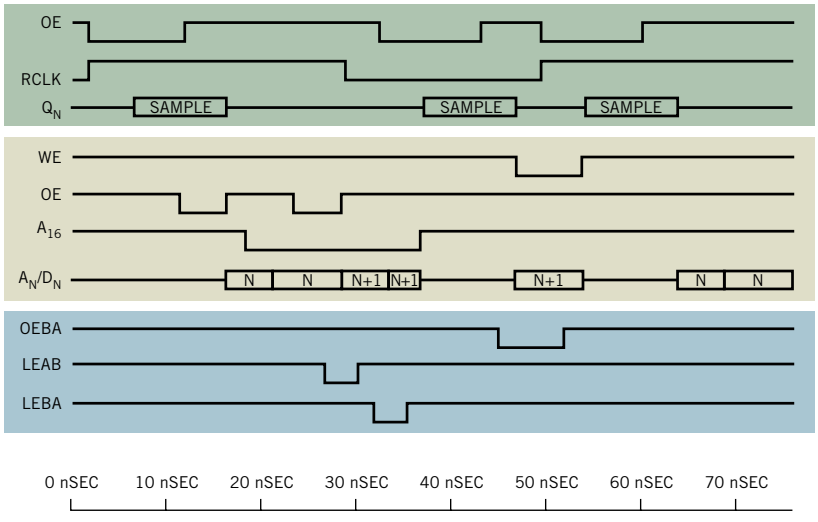


Figure 2 This graphic shows the histogram cycle timing for the circuit in Figure 1.