

Edited by Bill Travis and Anne Watson Swager

## Model fixed-point DSP arithmetic in C

Roger Maher, SSL Ltd, Dublin, Ireland

You can run initial high-level simulations of custom numerical algorithms, such as digital filters, using floating-point numbers in an environment such as C or Matlab. Unfortunately, you won't see include fixed-point effects, such as truncation due to limited precision and register overflow, until you use a Hardware Design Language (HDL), such as Verilog or VHDL. However, a technique that models these effects in C—the function “bit\_limit” in Listing 1—provides faster execution and better portability than HDLs and allows early exploration of the trade-off between bus width and performance.

Figure 1 shows a flow diagram that calculates the output  $E=A/4+A\cdot B$ . Table 1 shows the effect of the chosen bus width. For example, the output of the multiplier truncates to 10 bits, and the output of the adder truncates to 9 bits. In this example, when input  $A=201.8$  and input  $B=0.19$ , the output  $E=87.5$ ; an ideal model without truncation would give 88.792.

Listing 1, the ANSI C code for the bit\_limit function, shows how the code truncates. The code first converts floating-point numbers a\_fl and b\_fl to justi-

### LISTING 1—FUNCTION “BIT\_LIMIT” C CODE

```
#include <stdio.h>

/* Set the precision, ie. the maximum number of bits to the right
of the 'binary' point */
#define PREC 8

/*****
Function : bit_limit

Implements a precision limit on integers by regarding them as fixed
point numbers with 'l' bits allowed to the left of the 'binary' point
and 'r' bits to the right. If 'sgn_ext' is on then the MSB of the
resulting truncated number is sign-extended.

*****/

int long bit_limit(
int long l,      /* Bits left of the 'binary' point */
int long r,      /* Bits right of the 'binary' point */
int long in_val /* Input Value */
)
{
int long mask;

/* Build a mask which has 1's where data is valid */
mask = ((int long)1 << (l+r)) - 1 << (PREC-r);

/* Mask the input data */
return mask & in_val;
}

/*****
Function : main

Sample code to display the effect of limited precision fixed-point
binary arithmetic.

*****/

int main(void)
{
float a_fl, b_fl, e_fl;
int long A, B, C, D, E;

/* Pick some arbitrary input values */
a_fl = 201.8;
b_fl = 0.19;

/* Convert a_fl and b_fl to justified integers and limit to 8 bits
and 4 bits respectively */
A = bit_limit(8, 0, a_fl*(1<<PREC));
B = bit_limit(0, 4, b_fl*(1<<PREC));

/* Each multiplication needs to be re-scaled */
C = bit_limit(8, 2, (A*B)>>PREC);
/* Truncate D to 7 bits total */
D = bit_limit(6, 1, A>>2);

/* Truncate the output to 9 bits */
E = bit_limit(8, 1, C+D);

/* Re-scale to convert back to float */
e_fl = ((float)E)/(1<<PREC);

/* Display output and full_precision output for comparison */
printf("e_fl = %f\n", e_fl);
printf("e_fl(full_precision) = %f\n", a_fl/4 + a_fl*b_fl);
}

```

Model fixed-point DSP arithmetic in C .....	125
Two ADC channels double sensor precision .....	126
Transistor pair lowers PWM IC's start-up current .....	128
Autozero a position-sensing detector .....	130
OTP $\mu$ C controls Boomer amplifier .....	132
LED driver displays standing-wave ratio .....	134
Comparator provides stable hysteresis .....	136

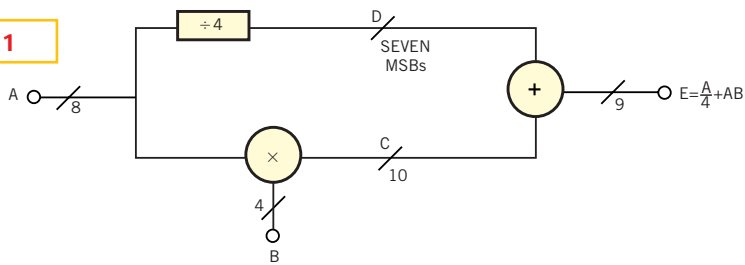
fied integers by scaling. The macro PREC sets the number of bits that the code uses to store the fractional part of the input, which are the bits to the right of the “binary” point. The bit\_limit function truncates a\_fl to 8 bits on the left of the binary point and to 0 bits on the right; the respective figures for b\_fl are 0 and 4 bits. Because the code stores the numbers as normal integers, the standard operators can perform addition and multiplication. However, the code must rescale the result of a multiplication because both operands are justified. The bus widths are set at points A, B, C, D and E; the code displays the bit-limited value of E and compares this value to the result from full-precision arithmetic.

You can handle signed arithmetic using 2’s complement numbers by extending the bit\_limit function to sign-extend the MSB after each truncation. You can perform rounding using the following statement:

bit\_limit(8, 0, A + 0.5\*(1<PREC)).

The number of bits in the “int long” data type on the simulation platform sets

**Figure 1**



A simple flow diagram implements the output  $E=A/4+A\cdot B$ .

**TABLE 1—TRUNCATION EFFECTS BASED ON BUS WIDTH**

								Binary point					
A	1	1	0	0	1	0	0	1	1	1	0	0	201
B	0	0	0	0	0	0	0	0	0	0	1	1	0.1875
C=AB	0	0	1	0	0	1	0	1	1	0	1	1	37.5
D	0	0	1	1	0	0	1	0	0	1	0	0	50
E=C+D	0	1	0	1	0	1	1	1	1	0	0	0	87.5

the major limitation of this technique. Normally, the number is 32 bits, so all numbers in the simulation, including the unscaled result of a multiplication, must be less than  $2^{31}$ . You can slightly extend this range by using “unsigned int,” or you can double the range by using “int long” when this type maps to 64 bits.

Listing 1 is available for downloading from EDN’s web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2328. (DI #2328)

To Vote For This Design,  
Circle No. 360

## Two ADC channels double sensor precision

Luke J Barker, Reinke Manufacturing Co Inc, Deshler, NE

The accuracy of the on-chip ADCs of numerous small and inexpensive 8-bit  $\mu$ P is well-suited for many applications. However, some situations benefit from just a little more accuracy from a resistive position sensor, for example. The resistive circuit in Figure 1 uses two ADC pins on a  $\mu$ P to double the precision of a resistive position sensor. In effect, the 8-bit ADC becomes a 9-bit ADC. The resistive part of the circuit shown in Figure 1 costs as little as \$3; the cost is higher if you use a precision potentiometer as the position sensor.

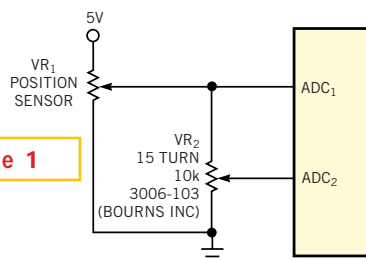
The concept behind using two ADC channels is simple. One ADC input takes direct measurements of the position sensor. The second ADC input measures the voltage out of a second potentiometer that is a  $1/2$  bit behind the first input.

**TABLE 1—ADC “PHASING”**

ADC <sub>1</sub>	0	1	1	2	2	3	.	.	.	253	254	254	255	255
ADC <sub>2</sub>	0	0	1	2	2	3	.	.	.	253	253	254	254	255
9-BIT RESULT	0	1	2	3	4	5	.	.	.	506	507	508	509	510

This scheme creates a “phasing” of the two analog-to-digital results (Table 1). Adding the two results produces a 9-bit answer that is limited to a value of 510.

**Figure 1**



Using two ADC pins on a  $\mu$ P doubles the precision of a resistive position sensor.

To set up the circuit for operation, you will need to accurately measure  $V_{REF}$ , which is 5.000V in this case, and set  $ADC_1$  ( $VR_1$ ) to a known voltage, 3.000V in this case. Then, adjust  $ADC_2$  ( $VR_2$ ) according to the following equation:

$$ADC_2 = ADC_1 - \frac{V_{REF}}{255} \cdot \frac{1}{2}$$

$$= 3.000 - \frac{5/255}{2} = 2.990V.$$

Thus, an  $ADC_2=2.990V$  sets input-channel  $ADC_2$  to a  $1/2$ -bit lag. (DI #2332)

To Vote For This Design,  
Circle No. 361



# Autozero a position-sensing detector

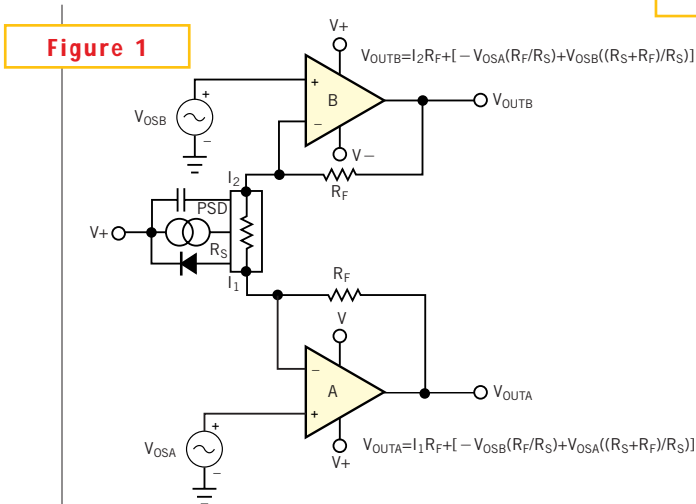
James Zannis, Renishaw S A, Champs-sur-Marne, France

**A**utozeroing schemes can be necessary to minimize the input-offset voltages of position-sensing detectors (PSDs), particularly when you use these detectors with high dc gain. PSDs are useful optical transducers for accurately measuring displacement. In practice, their typical configuration is as a differential current-to-voltage circuit (Figure 1). The ratio of the photocurrents  $I_1$ -to- $I_2$  linearly divides between the electrodes, proportional to the incident light beam. The magnitude of the photocurrents is a function of the light intensity.

Although PSDs are photodetectors, using them with high dc gain can quickly get you into trouble. Input-offset voltages can easily multiply due to the relatively low sheet resistance between pins 1 and 2, which typically ranges from 5 to 100 k $\Omega$ . Consider the following nodal equations:

$$V_{OUTB} = I_2 \cdot R_F + \frac{\emptyset}{\text{CE}} V_{OSA} \cdot \frac{R_F}{R_S} + V_{OSB} \cdot \frac{(R_S + R_F)}{R_S} \frac{\emptyset}{\text{CE}}$$

$$V_{OUTA} = I_1 \cdot R_F + \frac{\emptyset}{\text{CE}} V_{OSB} \cdot \frac{R_F}{R_S} + V_{OSA} \frac{(R_S + R_F)}{R_S} \frac{\emptyset}{\text{CE}}$$



**Figure 1**  
A differential current-to-voltage circuit is the typical configuration for a position-sensing detector (PSD).

An offset multiplication occurs because of the low intrinsic diode impedance,  $R_S$ , of the PSD, and the high values of  $R_F$ . For normal photodetectors,  $R_S$  is very high, and, therefore, no multiplication occurs. FET-input op amps are usually the choice for photodetector circuits because they have a higher input impedance and lower current-noise density at high-impedance levels than their bipolar counterparts.

For a typical pair of FET op amps mismatched by  $\pm 3$  mV and with an  $R_F$  of 1 M $\Omega$  and sheet resistance of 20 k $\Omega$ ,  $V_{OUTB}$  would be  $-0.303$ V, and  $V_{OUTA}$  would be 0.303V. For a theoretical pair of op amps mismatched by  $\pm 10$   $\mu$ V, an  $R_F$  of 1 M $\Omega$ , and a sheet resistance of 20 k $\Omega$ ,  $V_{OUTB}$  would be  $-0.001$ V and  $V_{OUTA}$  would be 0.001V. Ultimately, it would be nice to use a low-noise, low-bias-current amplifier with a very closely matched front end. However, it is a much more difficult process to match the input FET differential pair than that of a bipolar transistor. Nevertheless, this circuit shortcoming due to the sheet resistance is also the key to its success; you can force the offset voltage to be equal.

You can accomplish this goal by sam-

pling and integrating the difference of the two input-offset voltages and forcing the positive input of Amplifier A with the result (Figure 2). The two amplifier outputs converge to the input offset voltage of Amplifier B with the difference in the two output voltages equal to the offset of amplifier C.  $V_{OUTA}$  reduces to

$$V_{OUTA} = I_1 \cdot R_F + V_{OSB} + V_{OSC}$$

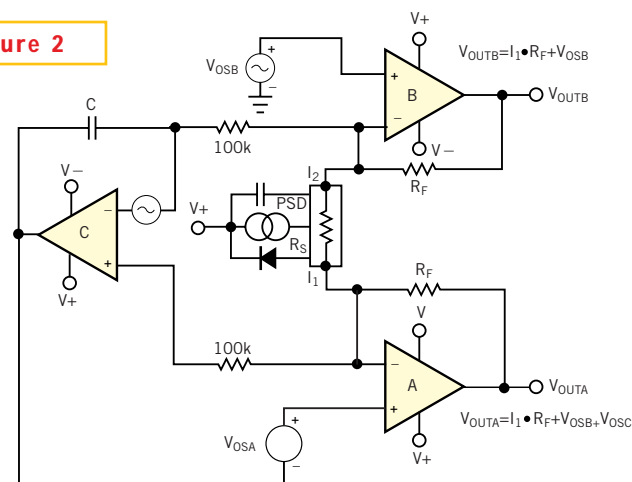
and  $V_{OUTB}$  reduces to

$$V_{OUTB} = I_1 \cdot R_F + V_{OSB}$$

Note that biasing the wrong op amp causes the two outputs to diverge, and you must choose an op amp for Amplifier C on the merits of its dc specifications.

The second approach to forcing the offset voltage to be equal is to use a potentiometer to adjust one of the op amps at its offset-adjustment pins. This circuit converges or diverges in a manner similar to that depicted in Figure 2. (DI #2331)

**Figure 2**



**Figure 2**  
Sampling the difference of the two input-offset voltages, integrating these voltages, and using the result to drive the positive input of Amplifier A causes  $V_{OUTB}$  and  $V_{OUTA}$  to converge to the input offset voltage of Amplifier B and to differ by the offset of Amplifier C.

To Vote For This Design,  
Circle No. 363

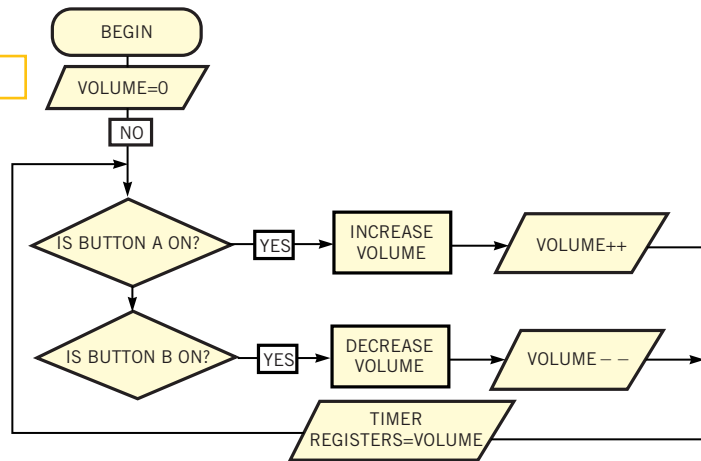
# OTP $\mu$ C controls Boomer amplifier

Wallace Ly, National Semiconductor Corp, Santa Clara, CA

**F**IGURE 1 shows a circuit that uses a one-time-programmable (OTP)  $\mu$ C in an unusual way. A COP8SGR7  $\mu$ C uses digital signals to “bit-bang” an LM4835 (dubbed “Boomer” by National) amplifier. Although the amplifier is designed for potentiometer control, you can modify it to make it a fully digitally controlled part. **Figure 2** shows the flow chart for the control process. A PWM signal and a lowpass filter allow you to use digital signals to control the amplifier. The technique sets the  $\mu$ C in processor-independent mode. You load the values affecting the duty cycle into the appropriate timer registers. When a control bit goes high, the  $\mu$ C delivers a PWM signal.

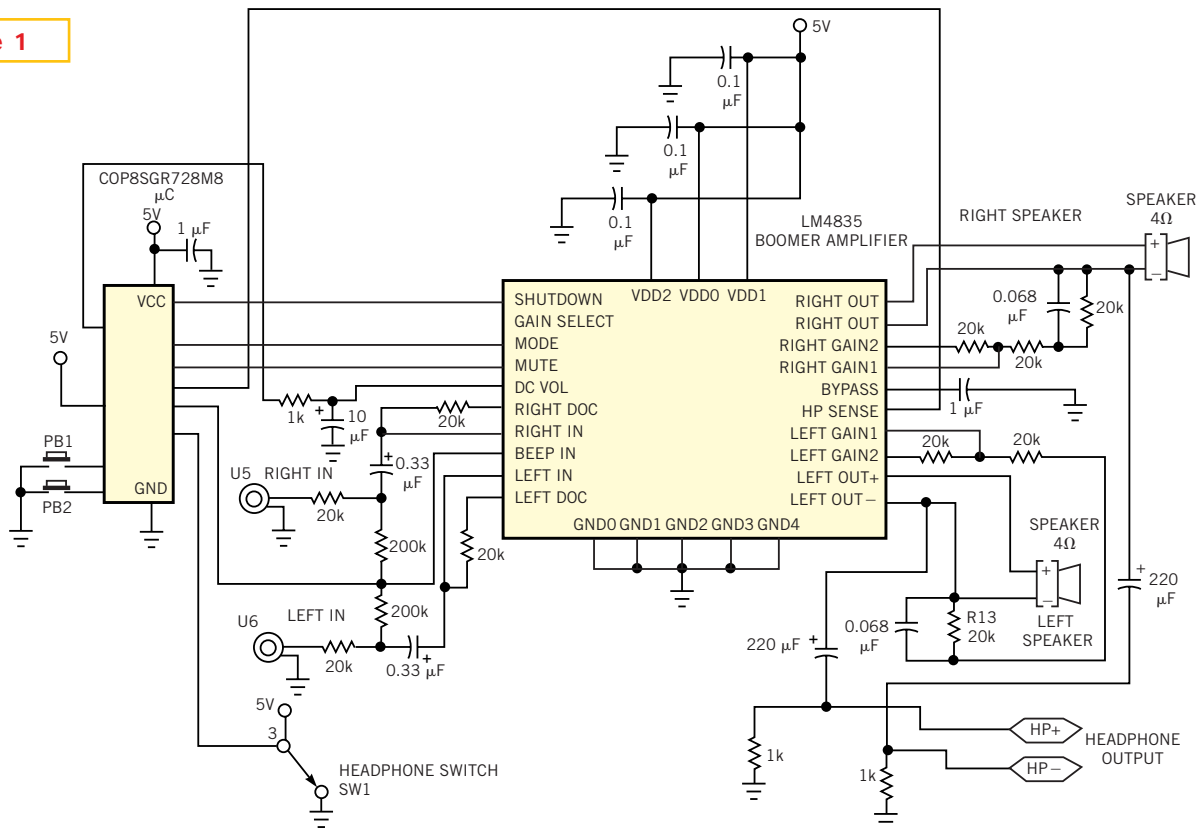
The PWM signal goes to a first-order lowpass filter. The output from the filter

**Figure 2**



Depressing the volume-control pushbuttons sets the duty cycle of the PWM signal the COP8SGR7  $\mu$ C generates.

**Figure 1**



Instead of tweaking potentiometers, you can use pushbuttons and a  $\mu$ C to control National's Boomer amplifier IC.

is a dc voltage, whose amplitude is proportional to the duty cycle of the PWM signal. The processor-independent mode allows the  $\mu\text{C}$  to perform other duties and calculations while it generates the PWM signals. With the COP8SGR7  $\mu\text{C}$ , three PWM outputs are available; there-

fore, the  $\mu\text{C}$  can control three amplifiers, or six channels of audio. The  $\mu\text{C}$  debounces the two (volume-up and -down) pushbuttons. The controller also handles the mute, shutdown, beep, volume, and headphone functions. You can download the C code for controlling the

$\mu\text{C}$  from EDN's Web site, [www.edn-mag.com](http://www.edn-mag.com). At the registered-user area, go into the Software Center to download the files from DI-SIG, #2326. (DI #2326).

To Vote For This Design,  
Circle No. 364

## LED driver displays standing-wave ratio

Richard Panosh, Vista, Bolingbrook, IL

The circuit in **Figure 1** uses an LM3914 LED driver to directly display standing-wave ratio (SWR) in a low-cost, rugged instrument. The SWR-sensing head is derived from the ARRL *Antenna Handbook* in an article that describes the tandem match. The forward voltage and reflected voltage ( $V_F + V_R$ ) signal drives Pin 6 ( $R_{HI}$ ), and the  $V_F - V_R$  signal drives the normal signal input at Pin 5. You can use this basic arrangement to display the ratio of two voltages in other applications. The internal circuit of the LM3914 comprises 10 voltage comparators that compare the input voltage at Pin 5 to an internal, 10-step, linear-voltage-divider string be-

tween Pin 6 ( $R_{HI}$ ) and Pin 4 ( $R_{LO}$ ). The voltage-divider levels,  $V_N$ , are  $V_N = V_{REF} \times n/10$ , where  $n$  is the voltage-divider step from 1 to 10, and  $V_{REF}$  is the voltage between pins 6 and 4.

When the signal voltage on Pin 5 satisfies the equality in the following equation, the  $n$ th LED energizes in the dot-mode display (with approximately 1 mV of hysteresis).

$$\frac{N}{10} \cdot V_{REF} \leq V_{PIN5} < \frac{N+1}{10} \cdot V_{REF}$$

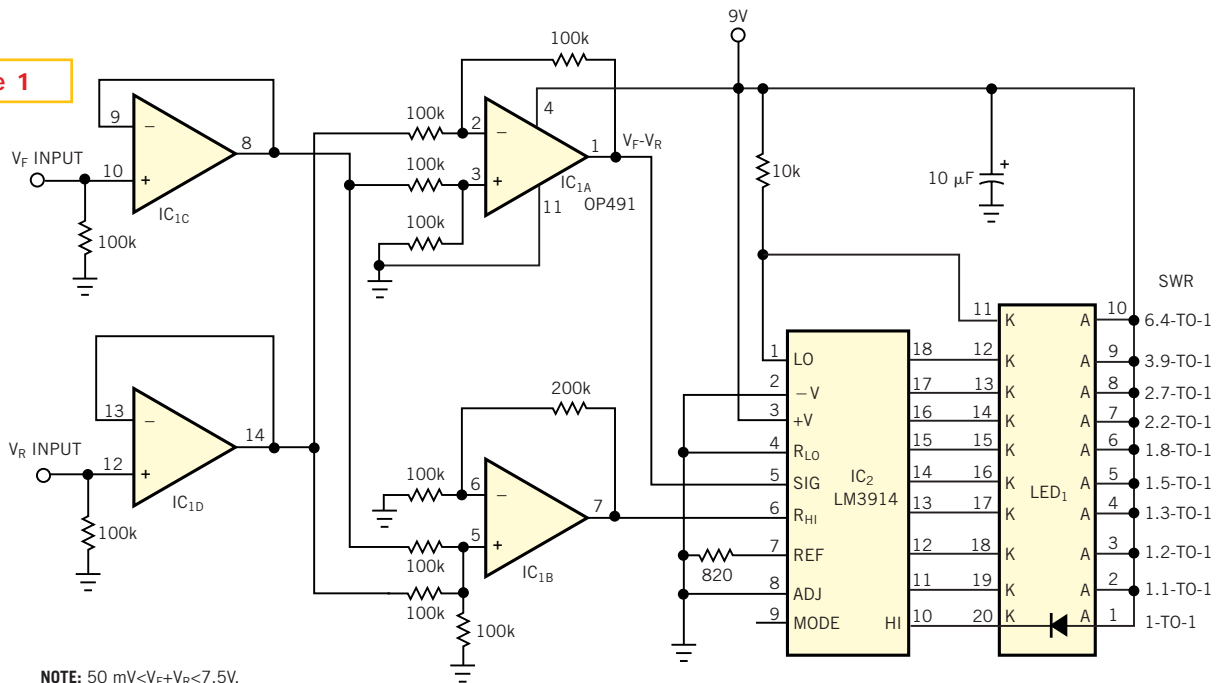
If you force  $V_{REF}$  to equal  $V_F + V_R$  and the voltage at Pin 5 to equal  $V_F - V_R$ , you can arrange the terms as in the following equation.

$$\frac{10}{N} \pm \frac{V_F + V_R}{V_F - V_R} = \frac{V_{REF}}{V_{PIN5}} > \frac{10}{N+1}$$

where  $V_F + V_R$ ,  $V_F - V_R$  is the definition of the SWR. Two sections of the quad op amp,  $IC_1$ , buffer the forward and reverse voltages developed in the SWR bridge. The remaining two sections serve as a difference amplifier to produce the voltage  $V_F - V_R$  and as a noninverting summing amplifier to produce the voltage  $V_F + V_R$ . Even though the circuit does not use the internal voltage regulator, you must properly terminate it to establish the LED current. (DI #2324).

To Vote For This Design,  
Circle No. 365

Figure 1



NOTE:  $50\text{ mV} < V_F + V_R < 7.5\text{ V}$ .

You can use an LED-driver IC to produce a thermometer-type indicator for the standing-wave ratio in RF systems.

# Comparator provides stable hysteresis

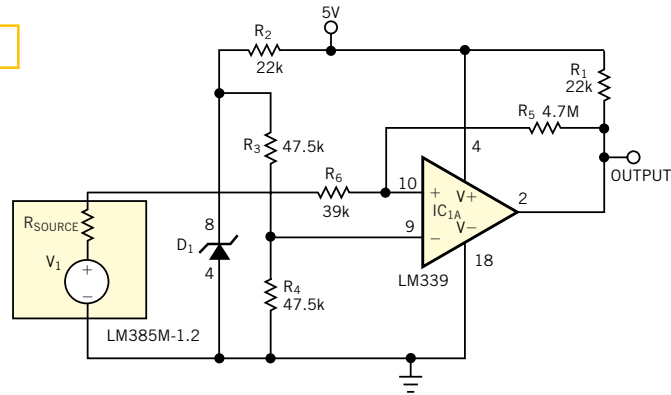
*Fernando Garcia, Lucent Technologies, Brownsville, TX*

Unless a voltage-comparator circuit is sampling an extremely clean signal, the comparator always requires some hysteresis. Traditional comparator circuits obtain the required hysteresis by using positive feedback derived from the ratio of two resistors. The voltage at the noninverting input is the superposition of a fraction of the input and output voltages, each divided by its respective resistor ratio. With an inverting comparator, the noninverting input usually connects to a voltage reference with reasonably low impedance, and you can choose the hysteresis-setting resistors without concern for loading. However, for the noninverting comparator, the input voltage comes from the actual source the circuit is sampling. This source has a series impedance,  $R_{SOURCE}$ , which can be a significant fraction of the input resistance  $R_6$  (Figure 1). This impedance may not be repeatable or may change value with changing circuit conditions; therefore, it may result in hysteresis errors.

Although you can change the feedback and input resistor values to minimize the source-impedance impact, you face a practical limit on the increases, because the value of the feedback resistor that the resistor requires to maintain the proper resistor ratios can become excessively high. Traditionally, you could use an op amp configured for unity gain to buffer the high-impedance source from the comparator. However, in some applications, cost, board-area, or current-consumption constraints may preclude adding an op amp. The circuit variant in Figure 2 eliminates the source-impedance problem. The voltage source under comparison connects directly to the noninverting input. Hysteresis does not come from resistor feedback, but rather from MOSFET  $Q_1$ . If the voltage you are sampling is less than the threshold, the comparator's output is low, and  $Q_1$  turns off.

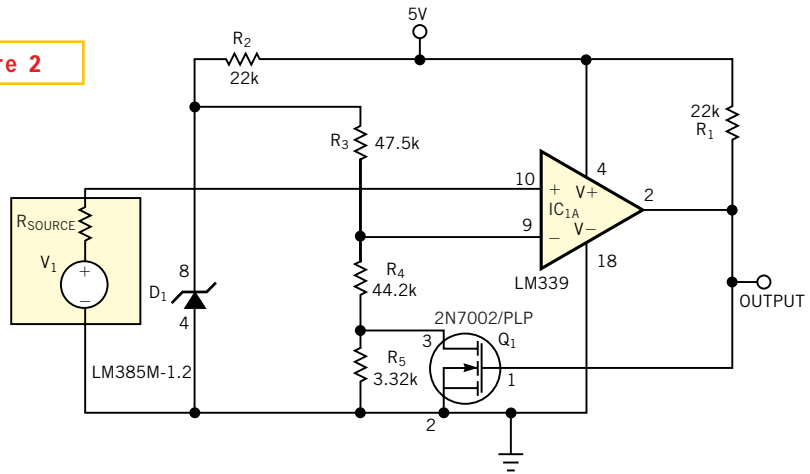
The comparator's inverting input essentially sees a reference voltage identical to the reference voltage in Figure 1.

**Figure 1**



Finite source impedance can provoke large errors in the hysteresis that you carefully calculate using resistor feedback ratios.

**Figure 2**



Source impedance has virtually no effect in this circuit; hysteresis comes from manipulating the thresholds at the inverting input.

However, when the voltage exceeds the threshold, the comparator's output goes high, turning on  $Q_1$ . The MOSFET shorts out the lower portion ( $R_5$ ) of the resistor divider. This action has the net effect of lowering the reference voltage to the comparator's inverting input. The differential voltage thus increases, providing the required hysteresis. The source impedance basically sees only the comparator's input impedance. This impedance is extremely high, so the source-impedance

impact on offset is low. Most small-signal MOSFETs can work in this application, provided that the  $R_{DS(ON)}$  is at least one order but preferably two orders of magnitude lower than the resistor it must shunt. This application requires a logic-level FET that turns completely on with  $V_{GS}=5V$ . (DI #2335).

To Vote For This Design,  
Circle No. 366

## 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 software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Business Information 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.**