

# design ideas

Edited by Bill Travis and Anne Watson Swager

## Voltage reference sets current limit

Joe Engle, Burr-Brown Corp, Tucson, AZ

Power op amps have a real need for active output-current limiting. Most power-amplifier designs rely on the voltage drop across a user-supplied sense resistor to turn on an internal transistor. This method has several drawbacks, notably, an inability to change the current-limit point under program control. The current-limit circuit in **Figure 1** allows you to establish the setpoint by applying a voltage to one of the amplifier's pins. With this design, it is possible to set the current-limit point with the output of a DAC, possibly under the control of an embedded  $\mu$ C.

The OPA547 is a true op amp; thus, it does not need a connection to power ground. The current-limit-setting voltage for this IC uses the negative supply as a reference. For single-supply applications in which the negative supply is ground, this referencing technique presents no problem, but for circuits that use a negative supply below ground potential, you need a different technique. The circuit in **Figure 1** shifts the reference potential for the control signal from ground to the negative supply. For simplicity, **Figure 1** shows the OPA547 as an inverting amplifier, but you

can use any op-amp application circuit. The circuit uses an OPA340 for reference shifting because it is capable of rail-to-rail operation on both input and output.

To understand the operation of the reference-shifting circuit, first recognize that the  $R_3$ -to- $R_4$  voltage divider sets the voltage at  $IC_2$ 's Pin 3. Thus, the intermediate voltage ( $V_i$ ), as measured from the negative supply, is given by

$$V_i = V_C \frac{R_4}{R_3 + R_4} \quad (1)$$

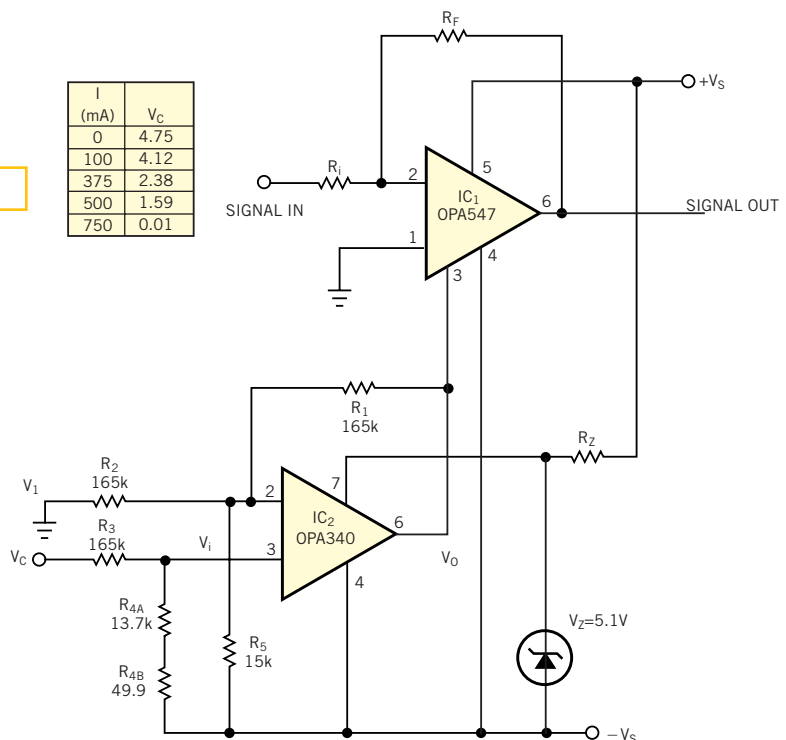
To find the voltage at  $IC_2$ 's pin 2, note that the current through  $R_3$  equals the sum of the currents in  $R_1$  and  $R_5$ , leading to the following expression:

$$V_1 - V_i = \frac{V_i - V_O}{R_1} + \frac{V_i}{R_5} \quad (2)$$

As long as op amp  $IC_2$  operates in the linear region, the voltage at Pin 2 equals the voltage at Pin 3, so the value of  $V_i$  in each of the expressions is equal. When you substitute the first term into the sec-

Figure 1

I (mA)	$V_C$
0	4.75
100	4.12
375	2.38
500	1.59
750	0.01



You can use a difference amplifier with wide common-mode range to control a power amplifier's current limit.

Voltage reference sets current limit .....	129
Accelerometer output gives temperature info.....	130
Voltage comparator forms pulse demodulator.....	132
Watchdog-reset catcher aids embedded-system debugging .....	132
Easily display bit-map images on small-graphic LCDs .....	134
CMOS inverter VCO tunes octave to UHF.....	138
Pushbutton or logic controls nonvolatile DAC .....	140

ond, set  $R_1$  equal to  $R_2$ , and combine terms, the resulting expression is

$$V_O = V_C \left( \frac{R_1}{R_5} + \frac{R_4}{R_3 + R_4} \right) V_1 \quad (3)$$

In **Figure 1**,  $V_1$  connects to ground, and you obtain unity gain by setting the coefficient of  $V_C$  in **Equation 3** to 1. If you expand and combine terms, the expression becomes

$$1 + \frac{R_1}{R_5} = \frac{R_3}{R_4} \quad (4)$$

To change the scalar relationship between the controlling voltage applied to the power op amp, simply set the coefficient term to the desired value and solve **Equation 3**. To determine the resistor values, consider the worst-case common-mode voltage that  $IC_2$  can encounter. OPA547 allows a maximum supply differential of 60V. In an extreme case, the positive supply of the OPA547 connects to ground and the current-limit set voltage is +5V. **Equation 1** becomes

$$5 = 65 \frac{R_4}{R_3 + R_4} \quad (5)$$

which reduces to  $R_3 = 12R_4$ . Applying this ratio to **Equation 4** and setting  $R_3$  equal to  $R_1$  produces  $R_1 = 11R_5$ . Selecting from a list of standard 1% resistor values yields the values in **Figure 1**. Note that the stage operates with a common-mode voltage that equals the negative supply. Errors in the resistor values can produce a significant offset shift. With this circuit, it is possible to set the current limit of the power op amp to a known, repeatable value under program control. (DI #2270).

To Vote For This Design,  
Circle No. 406

## Accelerometer output gives temperature info

Harvey Weinberg, Analog Devices Inc, Cambridge, MA

The adxl202 dual-axis micromachined accelerometer from Analog Devices (Norwood, MA) is appropriate for high-resolution applications. In these applications, you sometimes need to know the ambient temperature for control purposes or for circuit-drift compensation. The scheme in **Figure 1** offers a novel way to convey temperature information to the system  $\mu C$  without the need for an A/D converter or any additional I/O pins. The ADXL202 delivers two PWM signals that are proportional to the acceleration in its X and Y axes. Current in the  $R_{SET}$  resistor sets the period of the PWM signals.

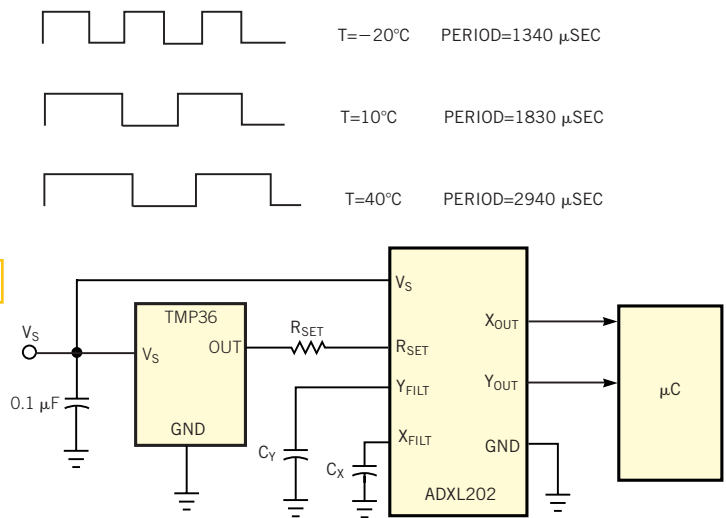
You can use a thermistor in series with or instead of the  $R_{SET}$  resistor to vary the PWM period with temperature. However, because of the grossly nonlinear response of thermistors, the PWM period is also grossly nonlinear with temperature. In addition, the thermistor's poor sensitivity at high temperatures may be unacceptable. Although  $R_{SET}$  normally connects to ground, you can connect it to any noise-free voltage source ranging from 0 to approximately 1.2V (at which voltage the internal current source runs out of compliance). By connecting  $R_{SET}$  to the  $V_{OUT}$  pin of a TMP36 temperature sensor, the PWM-

period set current varies fairly linearly (within  $\pm 5^\circ C$ ) with temperature from  $-20$  to  $+40^\circ C$ . Therefore, the PWM period varies linearly with temperature. You can easily extract temperature information from the PWM signal, because you nor-

mally measure the period to determine duty cycle. (DI #2271).

To Vote For This Design,  
Circle No. 407

**Figure 1**



**NOTES:**  
 $R_{SET} = 127k$ .  
PWM OUTPUT = 60% IN ALL CASES SHOWN.

An accelerometer can do double duty, by supplying both acceleration and temperature information.

# Voltage comparator forms pulse demodulator

Abel Raynus, Armatron International, Melrose, MA

To process low-level ultrasound or radio-range pulses, you need a signal-conditioning amplifier followed by a pulse demodulator to translate the signals to dc pulses. Traditionally, you would use a diode-demodulator configuration (for example, the circuit in **Figure 1a**) with one stage of a single-supply op amp. The circuit in **Figure 1b** does the same job but uses a voltage comparator instead of a diode demodulator. The key to the method is choosing a threshold voltage ( $V_{TH}$ ) on the negative comparator input that is slightly higher than the dc level of the amplifier output, which is equal to or close to  $V_{CC}/2$ . The  $R_4$ -to- $R_6$  resistive divider determines the difference between the op-amp bias and the threshold voltage. This difference, calculated to yield an acceptable signal-to-noise voltage, is  $V_{CC}R_5/(R_4+R_5+R_6)$ .

Or, assuming  $R_4=R_5+R_6$ , the difference is  $V_{CC}R_5/2R_4$ .  $R_3$  and  $C_2$  make up a lowpass filter. This pulse demodulator has some advantages: First, its sensitivity is higher than that of a diode demodulator. A 25-mV, 40-kHz, 1-msec input pulse produces a 0.1V output pulse in **Figure 1a**'s circuit, and a 2V output pulse in **Figure 1b**'s circuit. Second, it's convenient and economical to use one more stage of the dual or quad op amp instead of adding discrete components. (DI #2273).

To Vote For This Design,  
Circle No. 408

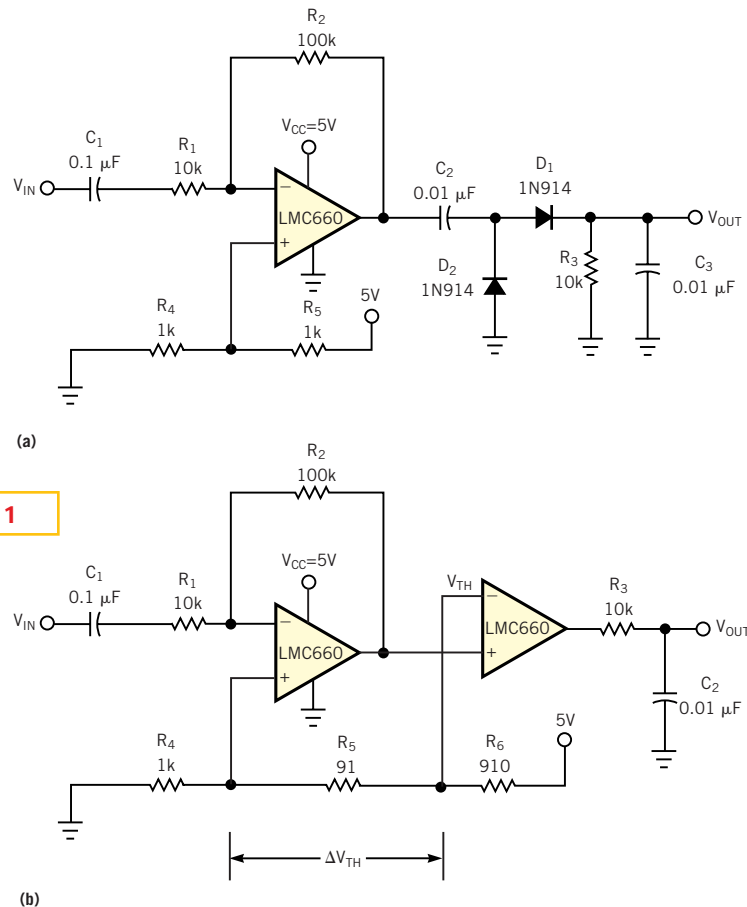


Figure 1

The passive, diode-based approach to pulse demodulation (a) provides a lower sensitivity than the active approach (b).

# Watchdog-reset catcher aids embedded-system debugging

Scott Newell, PCSI, Fort Smith, AR

A simple “junk-box” circuit uses a 4013 CMOS flip-flop and a handful of passive components to determine whether random resets are the result of a blown

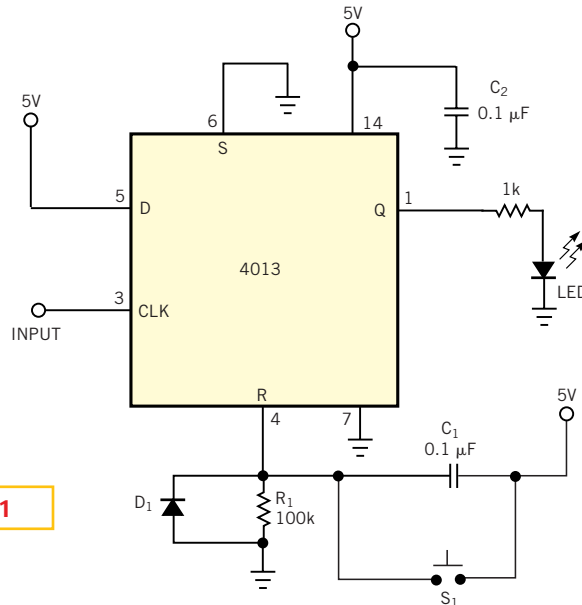
stack or the result of the watchdog-reset circuit tripping (**Figure 1**). You can also use this circuit to “grab” and hold other logic level edges like memory or I/O accesses.

A logic-level rising edge at the clock input (Pin 3) of the 4013 clocks the flip-flop. Because the circuit holds the data input (Pin 5) high, the Q output (Pin 1) goes

high, which turns on the LED. Once the LED is on, the circuit ignores any further changes at the input.

$R_1$  and  $C_1$  are the power-up reset for the flip-flop. At power-up,  $C_1$  discharges, which holds the reset input (Pin 4) of the 4013 high, clears the Q output of the 4013, and turns off the LED.  $C_1$  charges up to the supply voltage through  $R_1$ , taking the R input (Pin 4) low to deassert the 4013 reset time.  $D_1$  discharges  $C_1$  quickly on power-down.  $S_1$  is an optional reset switch.  $C_2$  is a power-supply bypass capacitor. Don't forget to ground all unused inputs on the 4013. To reset the circuit, either momentarily close  $S_1$  or temporarily disconnect power.

You can solder all the



**Figure 1**

**NOTES:**  
 LED=HIGH-EFFICIENCY RED LED.  
 $D_1$ =IN4001 OR ANY OTHER SMALL-SIGNAL DIODE.  
 $S_1$ =OPTIONAL MOMENTARY PUSHBUTTON SWITCH.  
 TIE UNUSED INPUTS TO GROUND.

**A 4013 CMOS flip-flop and a handful of passive components monitor the activity of an embedded system's watchdog reset.**

parts onto a BNC, which makes it easy to connect a scope probe directly to the watchdog-reset catcher. You can use a clip lead for the power line and easily steal power from the device under test. You can then connect the output of the embedded system's watchdog-reset circuit through the scope probe to the clock input of the 4013.

None of the part values are critical, and many types of flip-flops can substitute for the 4013. A faster flip-flop may be necessary to watch fast signals. Adding an inverter to the input would allow you to catch falling edges, such as active-low reset signals. (DI #2293)

To Vote For This Design,  
 Circle No. 409

## Easily display bit-map images on small-graphic LCDs

Todd Fitzsimmons, Densitron Corp, Santa Fe Springs, CA

Combining a high-level language, such as Microsoft Windows, with low-level assembly code or C++ code allows you to display perfect bit-map pictures on your small-graphic LCD screen. You can use Microsoft Paint or any other bit-map-generating program to define and edit the picture. Unlike segmented and alphanumeric LCDs, small-graphic LCDs are fully graphical and can display logos, graphs, or any other image in addition to numbers and characters. The main hurdle to upgrading to this type of display is the software necessary to display the bit-map pictures.

To create your bit maps, you can use any

available bit-map program, such as Microsoft Paint. After opening this program, select the Attribute menu. At the prompt, you enter the LCD size you are using and then choose the monochrome-bit-map option. You can now close the Attribute window and start drawing, typing, or pasting the images you want. After saving the pictures, you need to attach the bit maps to the end of your assembled or compiled program in order of their intended use. You can attach them using the COPY/B command in DOS, which differs from the regular COPY command by copying directly as a binary format without adding a byte of data at the end.

To use these attached bit maps, you need a subroutine in your program, such as the 8051 assembly code in Listing 1, which can pull the data out of the lower memory and send it to the display. (You can download this listing 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, #2295.) This code clocks in bit maps to the SED1330 controller chip on a 240×320-pixel display. If your display doesn't use the SED1330 chip, you can still use this code with modifications to the WRCMD and WRDATA subroutines and possibly some alterations on the direction you clock in the data. The di-

rection in the SED1330 is the same as the T6963 controller chip. However, if you use the HD61830, you need to switch the direction of the data; D<sub>7</sub> becomes D<sub>0</sub>, D<sub>6</sub> becomes D<sub>1</sub>, and so on.

For a bit-map program, the first 62 bytes of data call out the protocol for the rest of the bit-map code, such as the type, size and layout of the bit map. Because you have selected the monochrome option and specified layout dimensions using a bit-map-generating program, the subroutine can skip the first 62 bytes. The 63rd byte defines the first 8 pixels in the lower left corner of the display. The following bytes go sequentially to the screen until you hit the right edge of your display. The next byte is either the first byte on the next row up on the left side or a padded zero that the bit-map program places there to maintain certain integers for row length.

Padded zeros are necessary when the number of bytes in a row are not divisible by four. If you have 16 bytes of data per row, no padded zeros are necessary. However, if there are 30 bytes per row, two padded zeros are necessary to bring the number of bytes to 32. Your internal program must disregard these zeros before going on with the 33rd byte of data (Table 1).

Consider the example of driving a

TABLE 1—CORRELATION BETWEEN BIT-MAP RESOLUTION AND PADDED ZEROS

Bit-map resolution (pixels)	Bytes per row	Padded zeros per row	Totals divisible by four
32×80	10	2	12
32×202	26	4	30
33×100	13	3	16
64×128	16	0	16
64×240	30	2	32
64×480	60	0	60
128×128	16	0	16
128×240	30	2	32
128×56	32	0	32
200×640	80	0	80
240×320	40	0	40

128×240-pixel display. You would set up your assembly code to strip off and discard the first 62 bytes of data from the bit-map file. The 63rd byte is then the first byte in the lower left of the LCD. Then, the next 29 bytes of data (240/8=30) appear directly in the display. The code must then discard the next 2 bytes of padded zeros. The next byte of data then appear in the next row up and over on the left. A user continues this process until all 128 lines are completed.

If you access the upper bit-map memory by using the data pointer address in your μP, then, when you paint the first page and

increment the data pointer, you see the first byte of the next picture in your list.

An important difference between a bit map and an LCD is that, in bit-map programs, a binary 1 is an off pixel, and a binary 0 is an on pixel. So, a user must perform an exclusive-OR with FFh to properly view the bytes. Without this operation, your picture would be the inverse image of your original picture. (DI #2295)

To Vote For This Design,  
Circle No. 410

LISTING 1—BIT-MAP-EXTRACTION SUBROUTINE

```

.....
; This is a base program for clocking in bitmaps to the SED1330 on a 240x320 LCD display
; using 8051 assembly code.
.....
START:      MOV     30H,#3EH      ;SETS UP THE INNER LOOP TO 62
            MOV     31H,#01H   ;SETS UP FOR NULL DATA
            LCALL  CK_DATA     ;GO GET DATA
            MOV     B,#050H    ;SETS UP INNER LOOP FOR 240 LINES OF DATA
            MOV     32H,#008H  ;SETS LOWER CURSOR ADDRESS (address where program ends/bitmap begins)
            MOV     33H,#02AH  ;SETS UPPER CURSOR ADDRESS
            PUSH  B           ;STORE VALUE
DO_OVER:   MOV     30H,#28H    ;SETS UP THE INNER LOOP TO 40 CHARACTERS PER LINE
            MOV     31H,#008H  ;SETS UP FOR VALID DATA
            LCALL  SET_CUR    ;SETS THE INITIAL ADDRESS ON DISPLAY
            LCALL  CK_DATA     ;GO GET DATA
            POP    B          ;RECALLS INNER LOOP
            DJNZ  B,CONTINUE  ;IF NOT ZERO, THEN REDO (moving up one line on display if doing over)
            RET
CONTINUE:  MOV     A,#00H     ;RECALL CARRY FLAG
            ANL   A,#7FH     ;CLEAR THE CARRY FLAG
            MOV     0D0H,A     ;STORE FLAG
            MOV     A,#32H    ;RECALL LOWER CURSOR ADDRESS
            SUBB  A,#28H     ;SUBTRACT 40 FROM THIS ADDRESS TO MOVE UP ONE LINE
            MOV     32H,A     ;RESTORE THE NEW LOWER ADDRESS
            MOV     A,#00H    ;RECALL CARRY FLAG
            ANL   A,#80H     ;STRIP OFF CARRY FLAG
            CJNE  A,#80H,NO_CARRY ;SEE IF FLAG IS SET (if set then must bump down higher address by 1)
            MOV     A,#33H    ;RECALL UPPER CURSOR ADDRESS
            SUBB  A,#01H     ;SUBTRACT 1 FROM UPPER
            MOV     33H,A     ;RESTORE THE VALUE
NO_CARRY:  LJMPL DO_OVER     ;DO INNER ROUTINE OVER AGAIN
CK_DATA:   MOV     A,#31H    ;RECALL BYTE FOR NULL OR VALID DATA
            CJNE  A,#00H,GET_DATA ;SEE IF NULL DATA OR NOT (selects appropriate subroutine)
            LCALL  DATA_ARM  ;GET THE NULL DATA
            LCALL  WRCHAR     ;SEND IT TO THE DISPLAY
            DJNZ  30H,DATA_IN ;COMPARE LOOP AND DECREMENT UNTIL ZERO
            RET
            ;
GET_DATA:  LCALL  DATA_ARM   ;GET THE NULL DATA
            DJNZ  30H,GET_DATA ;SEE IF INNER LOOP IS ZERO
            RET
SET_CUR:   MOV     B,#46H    ;SET ADDRESS POINTER
            LCALL  WRCMD      ;CLOCK IN
            MOV     B,#32H   ;RECALL THE LOWER CURSOR POSITION
            LCALL  WRDATA    ;CLOCK IN
            MOV     B,#33H   ;RECALL THE HIGHER CURSOR POSITION
            LCALL  WRDATA    ;CLOCK IN
            RET
DATA_ARM:  MOV     R2H,#4FH   ;MOV LOWER DATA POINTER ON THE STACK
            MOV     R3H,#5FH ;MOV HIGHER DATA POINTER ON THE STACK
            MOV     A,#00H   ;CLEAR OUT REGISTER FOR TRUE DATA POINTER
            MOVC  A,@A+DPTR  ;GET THE DATA POINTED TO BY THE DATA POINTER
            MOV     B,A      ;SWAP DATA TO "B" REGISTER
            INC   DPTR      ;INCREMENT THE DATA POINTER
            MOV     34H,#2H  ;RESTORE LOWER DATA (includes increment)
            MOV     35H,#3H  ;RESTORE HIGHER DATA
            RET
WRCHAR:   PUSH  B          ;STORE DATA
            MOV     B,#042H  ;SETS WRITE DATA COMMAND
            LCALL  WRCMD     ;CLOCK IN
            POP   B         ;RECALL DATA
            XOR   B,#0FFH   ;INVERSE DATA TO PREVENT INVERSE-VIDEO
            LCALL  WRDATA    ;GO CLOCK IN
            RET
WRDATA:   MOV     0C0H,#03H ;SETS WR AND RD HIGH AND A0 LOW
            MOV     0C0H,#01H ;STROBE WR LOW
            MOV     090H,B   ;PUTS DATA ON PORT 1
            MOV     0C0H,#03H ;STROBE WR HIGH
            RET
            ;
WRCMD:   MOV     0C0H,#07H ;SETS WR, RD, AND A0 HIGH
            MOV     0C0H,#05H ;STROBE WR LOW
            MOV     090H,B   ;PUTS COMMAND ON PORT 1
            MOV     0C0H,#07H ;STROBE WR HIGH
            RET
            ;
            END

```

# CMOS inverter VCO tunes octave to UHF

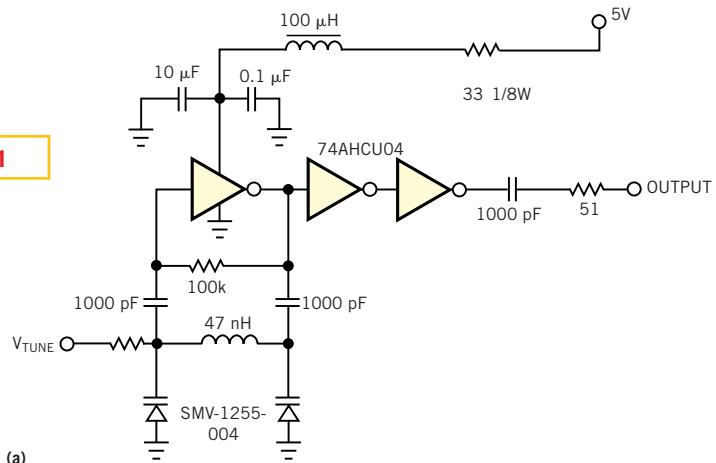
Shawn Stafford, AM Communications Inc, Quakertown, PA

**A** robust and versatile VCO provides a stable output to 300 MHz (Figure 1). The circuit's simplicity, unconditional stability, and consistent high-drive capability over an octave make the oscillator ideal for many applications, such as synthesized sources, local oscillators, and transmitters. The AHC logic family (Texas Instruments, www.ti.com) makes the circuit's performance possible. AHC is a relatively new line of CMOS logic whose high speeds and good noise performance allow oscillator operation into regions in which bipolar-junction-transistor and FET designs prevail.

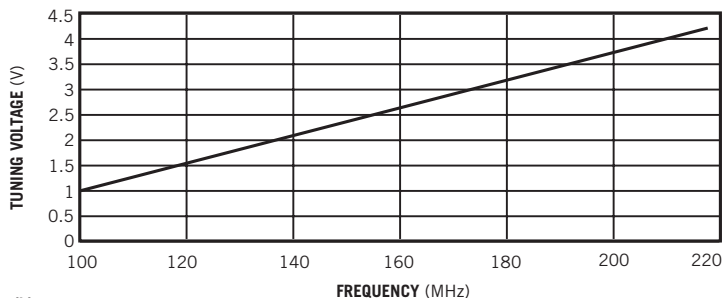
The oscillator topology is a modified Colpitts oscillator for which two hyperabrupt varactor diodes create the capacitive divider. The SMV-1255-004 (Alpha Industries, www.alphaind.com) encloses two varactors in one SOT-23 package (Figure 1a). The capacitance-voltage ratio of these varactors allows linear tuning over an octave with less than 4V (Figure 1b). You can substitute other varactors as long as the loaded Q of the resonant circuit is high enough to ensure start-up oscillation, but tuning characteristics may change. The inductor is a wound spring type chosen to maximize resonant Q. Oscillation is unstable when you use a low-Q, surface-mount-wound, chip-type inductor. The 100-k $\Omega$  resistor biases the gate to provide the gain and the 180° of phase shift necessary for oscillation. A lowpass filter with a low-frequency cutoff is highly recommended on the IC's power pin. Without this filter, incidental modulation from power-supply noise and pickup easily contaminate the oscillator signal. A dedicated voltage regulator is also recommended in noisy environments, but the filter is still necessary to keep the signal as clean as possible.

With a 5V supply, current consumption is approximately 25 mA $\pm$ 1 or 2 mA, depending on the frequency of oscillation. Using a 33 $\Omega$  series resistor can reduce the current to 18 mA and supply enough power for reliable oscillation. The cascaded gates provide extra buffering and drive; the output resistor improves match with additional buffering. If your design needs a known constant output imped-

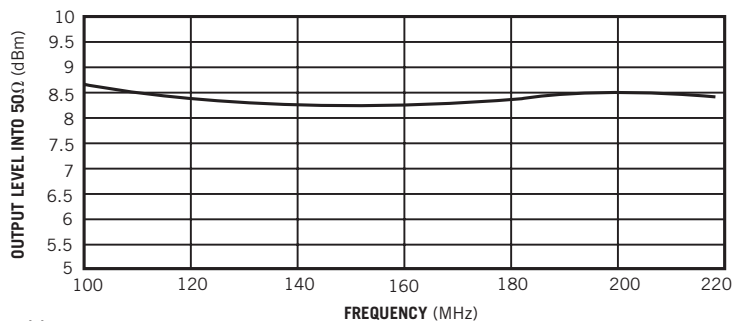
Figure 1



(a)



(b)



(c)

**A 300-MHz VCO (a) uses varactor diodes with a capacitance-voltage ratio that allows linear tuning over an octave with less than 4V (b). A high-drive capability over an octave (c) makes the oscillator ideal for many applications.**

ance, you can substitute a resistive match pad for the output resistor and maintain a considerable output level. Figure 1c shows the drive capability over frequency at mid-VHF, as well as level variation of less than 0.5 dB over the selected octave. Temperature effects on level are minimal with less

than 1-dB change over 0 to 75°C, and worst-case harmonics are always better than -12 dBc. (DI #2294)

To Vote For This Design,  
Circle No. 411

# Pushbutton or logic controls nonvolatile DAC

Stephen Woodward, University of North Carolina, Chapel Hill, NC

FOR MANUAL CONTROL of analog signals, it's hard to beat the venerable precision multiturn potentiometer's simplicity, resolution, and power-off nonvolatility. When digital control of an analog parameter is the design objective, a universe of DACs is available to the designer. The circuit in **Figure 1**, however, has manual-pushbutton and CMOS/TTL-compatible digital interfaces to a 10-bit, nonvolatile, two- or four-quadrant multiplying DAC. The heart of the circuit is the Xicor (Milpitas, CA) X9511 PushPot series of digitally controlled potentiometers. These devices implement a convenient up/down response to either ground-referenced contact closures (with built-in debounce and pullup provisions) or open-collector/drain digital signals.

Other useful features of these digital potentiometers include a  $\pm 5V$  analog-signal range and automatic storage and retrieval of settings with power-on/off

cycles via an on-chip EEPROM. The potentiometer's only shortcoming in this context that its resolution is inadequate for precision applications (only 32 distinct settings, equivalent to a mere five bits). To overcome this limitation, the circuit combines two PushPots with a summing op-amp buffer to achieve nearly 10-bit resolution. IC<sub>1</sub> provides a weighted sum of the wiper voltages of P<sub>2</sub> (coarse input) and P<sub>1</sub> (fine input) in the ratio of 25.5-to-1. This operation provides a composite resolution of  $32 \cdot (25.5 + 1) = 848$  distinct settings, equivalent to 9.7 bits.

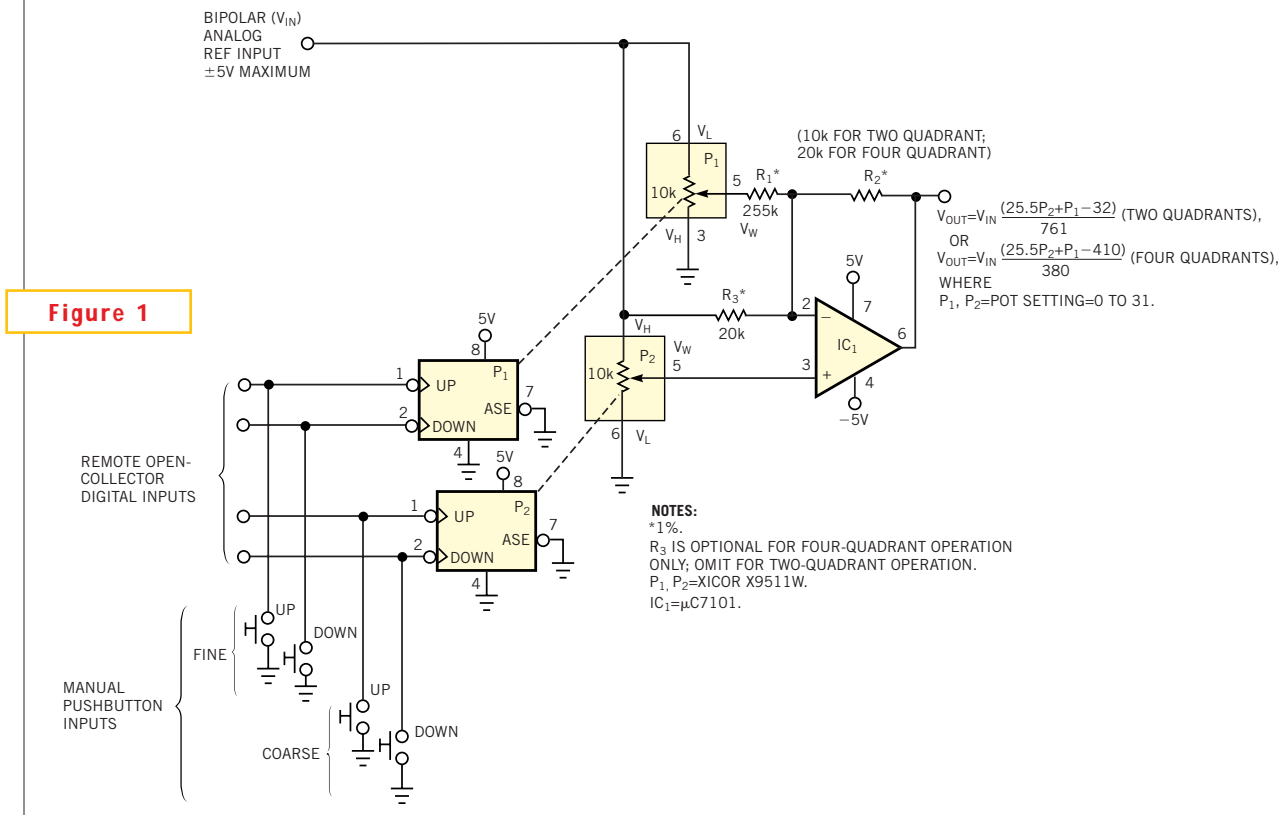
The missing 0.3 bits are lost to the good-but-still-only-finite differential linearity of the X9511 (Xicor specifies  $\pm 0.2$  LSBs) and the consequent need to give a less-than-ideal weight ( $32 \times 0.8$  instead of 32) to P<sub>2</sub> to guarantee overall DAC monotonicity. The resultant two-quadrant ( $R_2 = 10\text{ k}\Omega$ ,  $R_3$  omitted) gain equation

is  $V_{OUT}/V_{IN} = (25.5 \times P_2 + P_1 - 31) / 761$ . Thus, two-quadrant gain runs from  $-0.04$  to  $1.04$  in steps of  $0.0013$ , as P<sub>1</sub> and P<sub>2</sub> settings vary from (0,0) to (31,31).

Optionally, you can obtain four-quadrant multiplication by adding one resistor to the circuit, with the value  $R_3 = R_2 = 20\text{ k}\Omega$ . Gain then becomes  $V_{OUT}/V_{IN} = (25.5 \times P_2 + P_1 - 410) / 380$  and ranges from  $-1.08$  to  $1.08$  in steps of  $0.0026$ , as P<sub>1</sub> and P<sub>2</sub> vary from 0 to 31. The loading of P<sub>1</sub> by R<sub>1</sub> is light enough to produce a negligible effect on linearity. Connecting Pin 7 (automatic store enable) of P<sub>1</sub> and Pin 7 of P<sub>2</sub> to ground enables automatic storage of potentiometer settings to internal EEPROM upon power-down. The circuit then automatically retrieves the settings on power-up. (DI #2269).

To Vote For This Design,  
Circle No. 412

Use a pushbutton or provide a digital signal to choose a nonvolatile analog output with nearly 10-bit resolution.



## 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 02158

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 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.**