design ideas

The best of
design ideas

Check it out at:
www.edn.com

Edited by Bill Travis

Force/sense connection eliminates multiplexer on-resistance error

Stephen Woodward, Marine Sciences, Chapel Hill, NC

DIGITALLY CONTROLLED GAIN is an everyday analog-design element. You frequently find this element in an op-amp-based, transimpedance current-to-voltage converter. When you design digital gain control into such a converter, the usual scheme is to arrange things such that a digital multiplexer selects the appropriate feedback resistor for each gain figure. In **Figure 1**, op amp IC_1 is connected in a typical topology but with a twist. The normal way to arrange

the gain-setting multiplexer would be to take the converter's output directly from the op amp's output pin (IC_1, Pin 6). The trouble with this method is that the on-resistance of the multiplexer would then be effectively in series with the selected feedback resistance. In cases such as this one, in which the feedback resistance ranges as low as a few hundred ohms or less, the resulting gain error can be large. For example, the on-resistance of the HC4052 in **Figure 1** can exceed 100Ω.

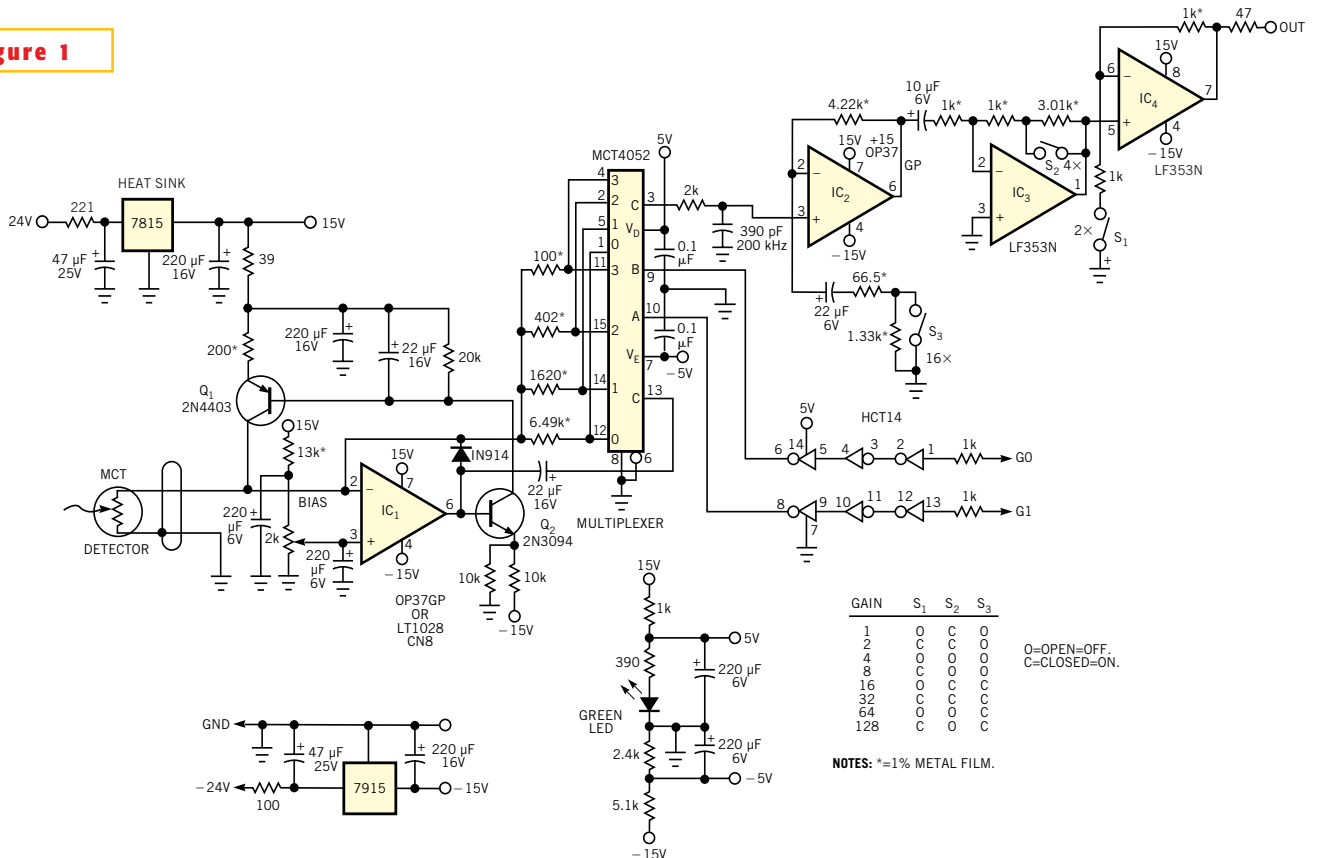Force/sense connection eliminates
multiplexer on-resistance error **79**

Analog multiplexer uses
flying capacitors **80**

FPGA-configuration scheme
is flexible **82**

Add fault protection to
a 4- to 20-mA loop supply **84**

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

Figure 1

In this programmable-gain circuit, the on-resistance of the gain-setting multiplexer plays no role in the determination of gain.

www.edn.com   JANUARY 22, 2004 | EDN **79**

That figure is equal to that of the lowest gain-setting resistor and, therefore, the source of 100% error. The obvious approach, using larger resistors, works poorly when you need high-frequency performance. The resulting RC delay products can cause frequency peaking, ringing, and, sometimes, outright oscillation.

This Design Idea offers an approach that makes the current-to-voltage converter gain independent of multiplexer resistance. The idea is to use two multiplexers in a force/sense topology such that the output comes from the "force" end of the selected gain resistance, rather than directly from the op amp's output. Assuming that that the load presented to the "sense" output is reasonably high, the result is a gain product that is insensitive to on-resistance. The remainder of the circuit surrounding IC$_1$ comprises a high-performance bias and preamplifier circuit for a cryogenic (liquid-nitrogen-cooled), mercury-cadmium-telluride infrared detector. These broadband, photoconductive optical sensors are popular in IR spectrometers. They are particularly popular in Fourier-transform-type spectrometers. Their popularity stems from their low noise, wide optical-wavelength responsivity, and electrical response of faster than 1 MHz.

Notable features of the circuit in **Figure 1**, besides the force/sense gain-setting topology, include dynamic biasing (via Q$_1$ and Q$_2$) of 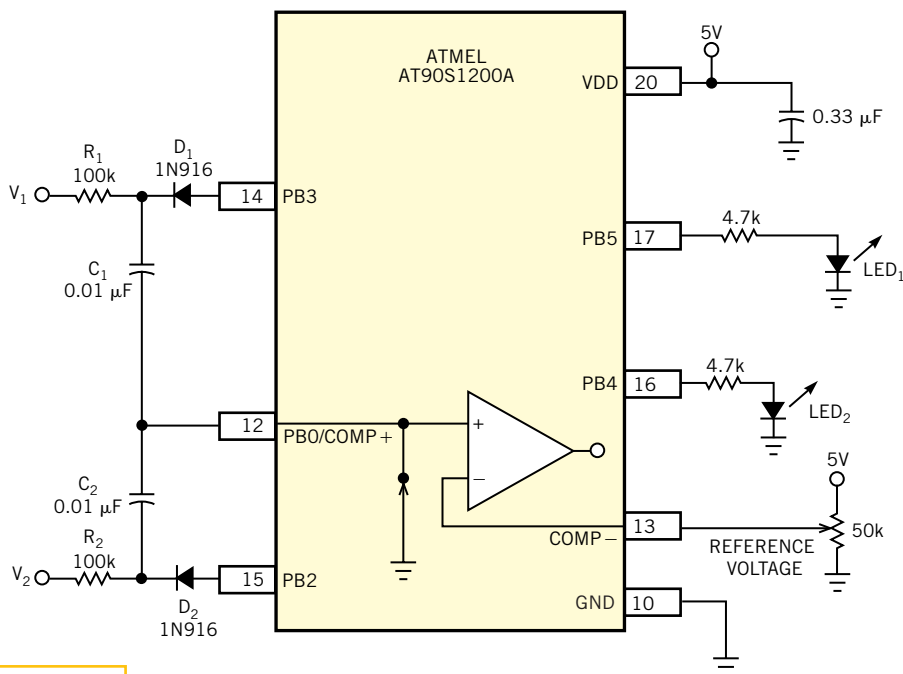the MCT detector, 64-to-1 (36-dB) digitally programmable gain, 128-to-1 (42-dB) manual-switch-settable gain, approximately 200-kHz bandpass response, and approximately 700-nV rms input-referred noise of less than 1 nV/$\sqrt{\text{Hz}}$). One trick that helps achieve this noise performance, other than the use of the superquiet LT1028 op amp for IC$_1$, is the cascaded-inverter HCT14 structure. The HCT14s serve no purpose other than to block entry of noise, which might be present on the digital gain-setting lines, into the gain-setting-multiplexer circuitry. Without these inverters, any such digital noise, a common cause of gremlins in high-gain, computer-controlled analog circuitry, could easily become capacitively coupled to the ac signal path.□

Analog multiplexer uses flying capacitors

Dick Cappels, Mesa, AZ

THIS DESIGN IDEA describes a way to increase the number of analog inputs to your microcontroller for cases in which adding an analog-multiplexer chip or upgrading to a microcontroller with more inputs might be impractical. If the microcontroller you're using has some spare I/O pins and at least one of them is bidirectional or is amenable to tristating, you can configure a simple analog multiplexer using switched capacitors. **Figure 1** shows a two-input multiplexer. A typical switched-capacitor multiplexer completely disconnects the capacitor from the sensed voltage before sampling the voltage across the capacitor.

To use a typical microcontroller's I/O ports, one terminal of the capacitor remains connected to the input source through a resistor. During most of the operating time, pins 12, 14, and 15 are configured as output pins and are held at logic 0. Diodes D$_1$ and D$_2$ do not conduct, so capacitors C$_1$ and C$_2$ charge to the values of the input voltages V$_1$ and V$_2$, respectively. To sample the voltage stored in the capacitors, pin 12 becomes an input, and the pin associated with each channel switches high while the microcontroller's comparator compares the voltage on Pin 12 with the reference voltage. **Listing 1**, which is available on the Web version of

Figure 1

The LEDs in this circuit indicate when the sampled input voltages are above the reference voltage on Pin 13.

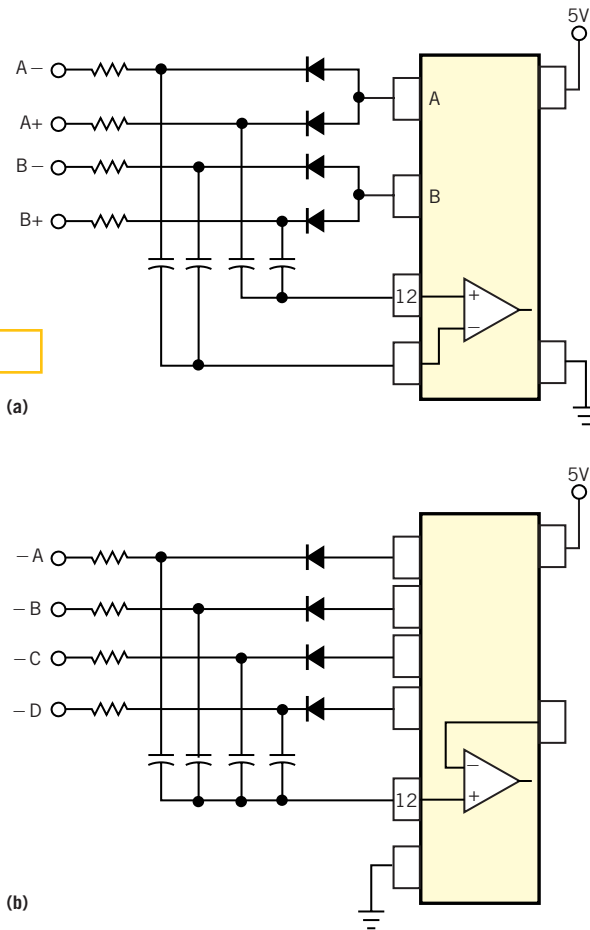this Design Idea at www. edn.com, gives the code fragment that samples the inputs.

The voltage on Pin 12 is $V_{PIN12} = V_{DD} - V_{DIODE} - V_{IN}$, where V_{PIN12} is the voltage applied to the analog input of the comparator; V_{DD} is the power-supply voltage (5V in this example); V_{DIODE} is the voltage across the diode, and V_{IN} is the voltage applied to input of the RC filter. During the sampling of one input, the voltage on the positive terminals of the capacitors exceeds V_{DD}; thus, D_1 and D_2 are in series with microcontroller pins 14 and 15 to block voltages above V_{DD} and prevent C_1 and C_2 from discharging into the power supply. Also during sampling, C_1 and C_2 are in series with the filter resistor of the input undergoing sampling, causing the capacitors to discharge through the resistor. For this reason, it is important to keep the RC time constant with respect to the sampling period. The worst-case voltage error occurs in the second channel to be sampled, when both V_1 and V_2 are at 0V:

$$V_{ERROR} = (V_{DD} - V_{DIODE})\left(1 - e^{-\left(\frac{T_{SAMPLE}}{R_2 C_2}\right)}\right),$$

where T_{SAMPLE} is the time one of the diodes' anodes switches to V_{DD} (3 μsec in this example). This sampling time uses the assumptions $R_1 = R_2$, $C_1 = C_2$, and the fact that the sampling periods for each channel are the same.

Figure 1

(a)

(b)

The technique lends itself to variations, such as switching sets of inputs (a) and expanding to more than two inputs (b).

With the 1-MHz controller in **Figure 1**, sampling time is a total of 6 μsec for the two channels; using a 16-MHz controller, the total sampling time would be only 375 nsec. When you expand the circuit for more inputs (for example, using the four-input multiplexer in **Figure 2a**), you must take the extra sampling time into account. To maintain a low duty cycle and thus allow the RC filters to charge to the full input voltage, the software should infrequently call the sampling routine. An interrupt every 2048 clock cycles calls the sampling routine in this example. The voltage at Pin 12 in **Figure 2a** is inverted, and, because of the isolation diodes, the maximum input voltage is a diode drop below V_{DD} (approximately 4.4V). If you multiplex both inputs, the circuit compensates for both the polarity and the diode drop (**Figure 2b**). **Listing 2** in the Web version of this Design Idea (www.edn. com) gives the microcontroller assembly code for the multiplexer scheme. You can download the software from the Web version of this Design Idea at www.edn.com.□

FPGA-configuration scheme is flexible

Zhe Lou, Ghent University, Ghent, Belgium

FPGAs ARE POPULAR in circuit design because of their flexibility and efficiency. You need to program an FPGA by loading configuration data into designated configuration memory. Because most FPGAs have no internal non-volatile memory, you must store the configuration files in external devices. When you use many FPGAs in a design, it is in-advisable to put a large amount of external memories near the FPGAs. The memory consumes a lot of area and increases the difficulty of the pc-board layout. Consider Xilinx (www.xilinx.com) FPGAs. Xilinx offers daisy-chaining techniques to program multiple FPGAs from a single source. However, when you want to change only one FPGA's functions and keep others unchanged, it is unwise to reprogram all FPGAs, because it takes a lot of time and can cause unexpected problems in the related circuits. This Design Idea describes how to individually program multiple FPGAs with limited resources. It uses a serial port of the Analog Devices (www.ana log.com) ADSP21065L to arbitrarily

program four FPGAs (**Figure 1**).

A DSP processor, the AD-SP21065L, serves as a micro-controller to program the FP-GAs. The configuration bus consists of the Clk, Data, Program, Init, and Done signals. The output data from the ADSP21065L is synchronous with the Clk signal, and the Program (output), Init (input), Done (input), and two control signals (output) are the ADSP21065L's I/O flags. The rest of the circuit comprises four FPGAs from Xilinx. The arrows to the FPGAs represent the configuration bus. The trick is in the so-called switchboard, which traces the configuration bus to an FPGA according to the ADSP-21065L's control signals. At first thought,
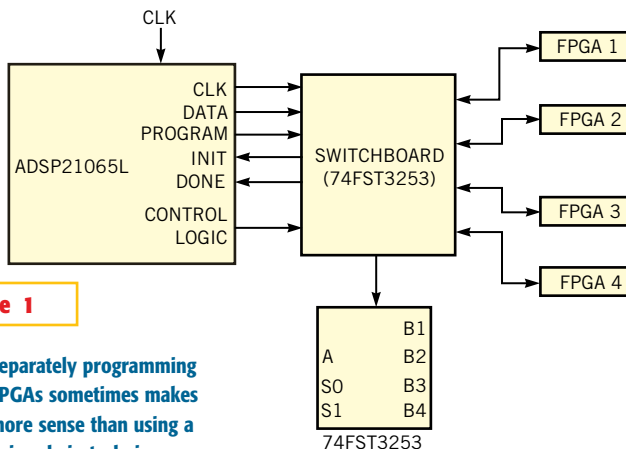
Figure 1

Separately programming FPGAs sometimes makes more sense than using a daisy-chain technique.

some bidirectional buffers, for example, 74LVT16245s, would seem suitable for this requirement by linking the control signals to OE and T/R pins of the buffers.

But after taking a closer look at the situation, this approach would be difficult

because the Init and Done are output signals from the FPGAs, which you cannot merge together. Therefore, the "buffer" you are looking for should have multiplexing or demultiplexing capabilities. This design uses the 74FST3253 dual 4-to-1 multiplexer/demultiplexer bus switch from On Semiconductor (www. onsemi.com) to implement this function. By connecting two control signals to the two select inputs, S0 and S1, you can cause I/O Signal A to connect to I/O lines B1, B2, B3, or B4, respectively, if the value of the two control signals are 00, 01, 10, or 11.□

Add fault protection to a 4- to 20-mA loop supply

Mark Pearson, Maxim Integrated Products, Sunnyvale, CA

A 4- TO 20-mA CURRENT LOOP consists of a power source and a current-measuring device at the control end and a field transmitter that senses process-variable information, such as temperature or pressure, and converts it to a current (**Figure 1**). Most such industrial current loops are powered by 24V dc, but that voltage can range from 12 to 36V. The loop voltage in older systems can be even higher. Many such applications require current limiting, fault protection, or both. For example, a short circuit or another high-current fault in one of several loops powered by a single source can produce a power-supply failure that disables all transmitters powered by that source. Intrinsically safe loops, on the other hand, include a barrier module that limits current and voltage to the transmitter. Fault-protected sources can add another level of system safety. Setting a current limit on each loop lets you accurately size the power supply without over-specifying it. **Figure 2** shows one form of flexible fault protection for the 24V power-
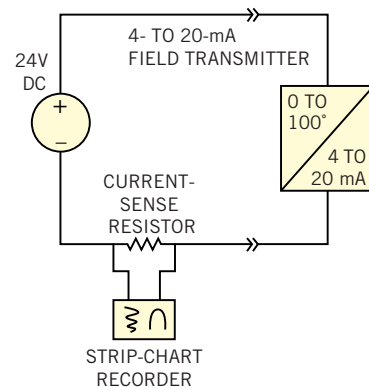
Figure 1

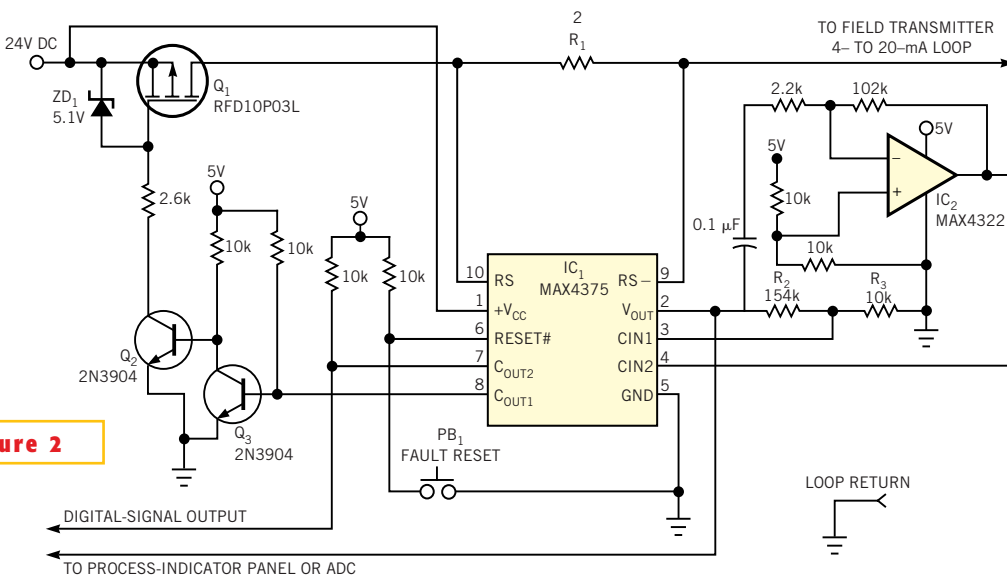**Industrial applications widely use the basic structure of a 4- to 20-mA current loop.**

er supply of a 4- to 20-mA loop. It also includes circuitry for recovering a digital signal superimposed on that loop. IC_1, a high-side current-sense amplifier with comparator and reference, senses the loop current in R_1 as an 8- to 40-mV voltage and amplifies it by 100, producing an output-voltage range of 0.8 to 4V. That out-

put, V_{OUT}, can directly drive external meters, strip-chart recorders, and A/D-converter inputs.

The R_2-R_3 voltage divider sets the selected fault-current trip point for IC_1's first internal comparator at 0.6V. Setting the trip point for a 50-mA fault, for instance, establishes the following relationship between R_1 and R_2: $R_2/(R_1+R_2) = 0.6V/(R_1 \times 100 \times I_{FAULT})$, so $R_1 = 15.67 \times R_2$. When faults occur, the C_{OUT1} output assumes a high-impedance state and is pulled high by R_3. The noninverting cascaded-transistor pair Q_2-Q_3 provides an interface to the high loop voltage and preserves a proper logic polarity for controlling the gate of Q_1. Q_1 is held in the off state until pushbutton PB_1 or another reset signal resets IC_1's first comparator. (To disable this comparator's latched output, tie the Reset# pin to ground.) Zener diode ZD_1 protects Q_1's gate-source junction from overvoltage.

IC_2 and its associated circuitry can recover any digital information imposed on the 4- to 20-mA loop current by modu-

lation. The High-way-Addressable Remote Transducer Protocol, for instance, typically uses FSK (frequency-shift keying) of 1200 to 2400 Hz to modulate the loop current between the ±0.5mA levels. (For this circuit, the modulated signal at V_{OUT} (Pin 2 of IC_1) is ±0.1V.) V_{OUT} from IC_1 is capacitively coupled to IC_2 and amplified by that device to recover such digital signals. IC_1 includes a second comparator with inverting input, which you can use to cancel the inversion in IC_2's digital-signal output. Though not essential, this comparator output (C_{OUT2}) can also present the recovered digital signal as a clean rectangular waveform for driving external circuitry.□

Figure 2

This circuit provides fault protection and digital-signal recovery for a 4- to 20-mA current loop.