<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Circuit clamps video signals</td>
<td>3</td>
</tr>
<tr>
<td>Circuit monitors RS-232C communications</td>
<td>30</td>
</tr>
<tr>
<td>High-gain amp yields low dc output offset</td>
<td>4</td>
</tr>
<tr>
<td>Monitor thermocouples electronically</td>
<td>31</td>
</tr>
<tr>
<td>Op amp provides accurate level shifter</td>
<td>6</td>
</tr>
<tr>
<td>C program finds words that sound alike</td>
<td>33</td>
</tr>
<tr>
<td>AGC circuit uses an analog multiplier</td>
<td>7</td>
</tr>
<tr>
<td>Simple routines provide calendar date</td>
<td>35</td>
</tr>
<tr>
<td>Arbiter lets two µPs use common RAM</td>
<td>8</td>
</tr>
<tr>
<td>Resistors provide nonlinear pot tapers</td>
<td>37</td>
</tr>
<tr>
<td>Charger extends lead-acid-battery life</td>
<td>10</td>
</tr>
<tr>
<td>Sync separator flags odd fields</td>
<td>39</td>
</tr>
<tr>
<td>MOSFET circuit debounces power relay</td>
<td>11</td>
</tr>
<tr>
<td>Digital AGC is fast, stable, and accurate</td>
<td>42</td>
</tr>
<tr>
<td>Converters yield droop-free S/H circuit</td>
<td>13</td>
</tr>
<tr>
<td>68020 adapter upgrades 68000 systems</td>
<td>44</td>
</tr>
<tr>
<td>Diodes and capacitors imitate transformer</td>
<td>14</td>
</tr>
<tr>
<td>Lamps improve solenoid-driver efficiency</td>
<td>46</td>
</tr>
<tr>
<td>Step-up converter produces 5V from 1.5V</td>
<td>16</td>
</tr>
<tr>
<td>Circuit measures op-amp settling time</td>
<td>47</td>
</tr>
<tr>
<td>V/I converter has zero I_B error</td>
<td>18</td>
</tr>
<tr>
<td>Program converts Gray to binary and back</td>
<td>50</td>
</tr>
<tr>
<td>Direction detector doubles as decoder</td>
<td>20</td>
</tr>
<tr>
<td>Control duty cycle with up-down counters</td>
<td>51</td>
</tr>
<tr>
<td>Simple circuit suits quadrature detection</td>
<td>22</td>
</tr>
<tr>
<td>Circuit monitors duty cycle</td>
<td>52</td>
</tr>
<tr>
<td>Build a µP-controlled tone generator</td>
<td>24</td>
</tr>
<tr>
<td>Video IC extracts timing waveforms</td>
<td>53</td>
</tr>
<tr>
<td>Modulator offers 12-bit precision</td>
<td>25</td>
</tr>
<tr>
<td>Perform a Z80 reset without memory loss</td>
<td>54</td>
</tr>
<tr>
<td>TV sync generator acts as clock timebase</td>
<td>27</td>
</tr>
<tr>
<td>Program computes log magnitude and phase</td>
<td>55</td>
</tr>
<tr>
<td>Capacitor maintains data during power loss</td>
<td>29</td>
</tr>
<tr>
<td>Pulse generators offer improved performance</td>
<td>57</td>
</tr>
<tr>
<td>Pulse-delay circuit has dual-edge trigger</td>
<td>60</td>
</tr>
<tr>
<td>Subroutines handle speech companding</td>
<td>61</td>
</tr>
<tr>
<td>IC emulates many types of logic gates</td>
<td>64</td>
</tr>
<tr>
<td>Multiplier serves as a high-speed switch</td>
<td>66</td>
</tr>
<tr>
<td>Square-root algorithm is fast and simple</td>
<td>68</td>
</tr>
<tr>
<td>Reset circuit solves brownout problems</td>
<td>69</td>
</tr>
<tr>
<td>Convert square waves to sine waves</td>
<td>70</td>
</tr>
<tr>
<td>Timer IC provides low-power voltage monitor</td>
<td>71</td>
</tr>
<tr>
<td>DAC lends digital control to phase shifter</td>
<td>73</td>
</tr>
<tr>
<td>Enlarge Z80 memory space to 512k bytes</td>
<td>75</td>
</tr>
<tr>
<td>Battery powers simple IR detector</td>
<td>77</td>
</tr>
<tr>
<td>Biquad filter offers adjustable f_0 and Q</td>
<td>78</td>
</tr>
<tr>
<td>Storage-time tester uses two ICs</td>
<td>79</td>
</tr>
<tr>
<td>MOSFET provides glitchless power backup</td>
<td>81</td>
</tr>
<tr>
<td>Talking meter gives dc-voltage readings</td>
<td>82</td>
</tr>
<tr>
<td>Improved tachometer eliminates backlash</td>
<td>84</td>
</tr>
</tbody>
</table>
INTRODUCTION

Every batch of Reader Service cards we receive contains many comments about our Design Ideas section. Most often, readers tell us they’ve found a particularly useful circuit that helped them solve a tricky design problem. But readers also want to know how they can get a complete set of past circuits and software. Alas, such a compendium doesn’t exist. But these Design Ideas Special Issues represent the next best thing—each is a collection of 50 of the best Design Ideas published in EDN from 1985 through June 1988.

This collection of designs reflects your needs and not the preferences of our technical editors. In fact, you and your colleagues voted for these winning ideas by circling the “bingo card” numbers that appear at the end of each Design Idea we publish. Of the circuit-design tips in this Design Issue, 29 garnered Best-of-Issue honors based on reader preference. The remaining Design Ideas received the second-highest number of reader votes over the 1985 to 1987 time frame.

In planning these Design Ideas Special Issues, EDN hoped to be able to present a good mixture from the standpoint of design disciplines. Happily, our readers’ choices made it easy to achieve this goal. This issue contains 21 analog circuit designs and 23 digital designs, and the remaining six ideas highlight programming tips. To achieve some semblance of order, we’ve divided the ideas into some generic categories and provided an index that lists design and circuit categories. Although the index is a handy place to look for a special circuit, we’ll bet that most readers will want to read the ideas one by one, just to see what circuits are available.

As you read these Design Ideas, remember that each one came from a
reader who thought someone else might benefit from his or her work. So, if you find this issue helpful, thank your fellow engineers who submitted the ideas we’ve published over the years. And the next time you have an interesting and useful circuit, consider sharing it with other engineers by submitting it for publication in *EDN*’s regular Design Ideas section.

Besides seeing your name in print and gaining an extra $100, you might find your idea has been selected as the issue winner. Each issue winner collects an extra $100. Keep in mind, too, that *EDN*’s editors also choose a grand-prize winner each year. The grand prize includes a check for $1500. Your idea may also be selected to appear in a future Design Ideas Special Issue. You’ll find an entry form in the Design Ideas section in most issues of *EDN*.

Our commitment to short, comprehensive design solutions includes one more Design Ideas Special Issue in 1988; you’ll receive it with your first December showcase issue. If you enjoy this Design Ideas Special Issue and find it useful, we’d like to hear from you. If you have suggestions for improvements and changes, we’d like to hear those, too. Just send us a note or give us a call. We’d also like to thank all the readers who have submitted Design Idea entries over the years.

Enough introduction. Here are the ideas you’ve been clamoring for. Good reading.

---

Tom Ormond
Senior Editor
Circuit clamps video signals

Richard Andelfinger
Hughes Aircraft Co, Tucson, AZ

The circuit shown in Fig 1 uses a track-and-hold amplifier in a closed-loop configuration to clamp the back-porch voltage of a standard video waveform to 0V. The circuit’s outputs include a clamped composite-video signal and a TTL-level horizontal-blanking pulse.

The differential input buffer (IC₁) and the summing amplifier (IC₂) isolate the input video signal. Clipper IC₄ removes the video signal, leaving only the synchronization information. Differentiator IC₅ detects the edges of the horizontal blanking pulses and produces pulses that correspond to the leading and trailing edges of the horizontal blanking pulses. IC₆ clips these pulses and converts them...
to a TTL level.

IC7 uses these clipped pulses to generate a TTL-level window that, when combined with the horizontal pulse generated by IC8, forms a TTL representation of the original horizontal pulse. This representation is synchronized to the input waveform. IC5 uses the trailing edge of this reconstructed waveform to generate the track pulse for track-and-hold amplifier IC10. IC11 filters IC10’s dc output and, after gain adjustment, feeds it back to IC3’s summing node.

This circuit will reliably acquire and clamp video signals with large dc offsets and superimposed ac waveforms. Capacitor C1 determines the circuit’s ac bandwidth. The 1-µF value chosen for the capacitor in this case makes the circuit suitable for handling very noisy signals, such as those you might encounter in third- and fourth-generation videotapes. If you eliminate this capacitor, the circuit will be able to clamp signals with frequencies as high as 120 Hz, but with reduced noise-rejection capabilities.

**DESIGN IDEAS**

**HIGH-GAIN AMP YIELDS LOW DC OUTPUT OFFSET**

Christopher Paul
*Coherent Communications Systems Group, Dix Hills, NY*

The digitally controlled audio amplifier shown in Fig 1 features high gain and low output-offset-voltage variations. Using the components shown, the circuit can provide a gain of 50 dB with an error as low as ±0.2 dB for a frequency range of 30 to 3000 Hz. Its output-offset-voltage variation is less than 15 mV.

Other digital-control schemes either can’t achieve high gain or generate objectionably large output-offset voltages. For example, a circuit comprising a D/A converter and an op
amp (Fig 2) features a low offset voltage output but has a maximum gain of only 6 dB.

On the other hand, the circuit shown in Fig 3, which uses a D/A-converter ladder network in the feedback path of an op amp, yields a high gain but generates output offset voltages as high as several volts. This problem arises because the circuit multiplies the input offset voltage by the op amp’s audio gain. Also, this circuit demands a high-performance op amp. To keep gain errors low in this circuit, the op amp’s noise gain at the highest frequency of operation must equal the noise gain of the rest of the circuit. Thus, the op amp must have a high open-loop gain.

Fig 1’s circuit overcomes the dc offset problem of Fig 3’s circuit by using ac coupling (via C1) rather than dc coupling to connect the D/A converter’s ladder network to ground. As a result, the circuit’s input offset voltage is no longer multiplied by the op amp’s audio gain. With the values shown, this circuit can operate at frequencies as high as 30 kHz. The circuit operates at lower frequencies if you use capacitors with higher values. Note also that because C1 has virtually 0V across it at all times, it can have a small voltage rating.

The circuit’s only drawback is that it requires an op amp with a high open-loop gain. The Signetics NE5534, which provides an open-loop gain of 80 dB at 10 kHz, is a good choice.EDN
Op amp provides accurate level shifter

Using an op amp and a few resistors and capacitors, you can build an accurate and adjustable level shifter. Fig 1’s circuit uses a positive or negative voltage to shift a pulse train’s level and is insensitive to a pulse’s rise time, repetition rate, and duty-cycle variations. The circuit lets you interface logic families with dissimilar logic levels—for example, ECL to TTL.

An external power supply or a resistive voltage divider provides the reference voltage, $V_s$, for the circuit. Given that the pulse train has a dc component described by

$$V_{DC} = \frac{1}{T_0} \int_{0}^{T_0} V_i(t) \, dt,$$

where $T_0$ is the period of the lowest frequency component ($f_0$) in the pulse train. Because the pulse train also has an ac component, you must sum both the pulse train’s dc and ac components with $V_s$ to the desired offset. By setting $R_3 = R_1 + R_2$ and $R_4 = R_5$, you configure the op amp as a summing amplifier with inputs of

$$V^+ = \frac{V_{DC} + V_s}{2}$$

and

$$V^- = V_{OA}/2.$$

Because $V^+ = V^-$, $V_{OA} = V_{DC} + V_s$ where $V_s$ is the control voltage by which $V_i$ (the pulse train) is to be shifted.

Capacitor $C_1$ passes $V_i$’s ac components, provided that $X_{C1} \gg R_6$ at $f_0$. The combination of $R_1$, $R_2$, and $C_2$ provides a lowpass filter that removes the ac component of $V_i$ and passes just its dc component. Choose values such that reactances of $C_2$ and $C_3$ meet the condition $X_{C2}, X_{C3} \gg R_6$ at $f_0$. $C_4$ provides an ac ground.

For a TTL-to-ECL interface, choose $R_6 = 50\Omega$ (to provide an ECL termination) and $V_s = -2.9V$. A prototype of this circuit has performed successfully for both positive and negative values of $V_s$ and pulse repetition rates to 100 MHz.

Fig 1—With an op amp and a handful of parts, you can build this level-shifting interface circuit.
AGC circuit uses an analog multiplier

Steve Lubs
Dept of Defense, Washington, DC

In the AGC circuit of Fig 1, a four-quadrant analog multiplier (IC₁), an amplifier stage (IC₂), an active, full-wave rectifier (D₁, D₂, R₅-R₉, and IC₃), and an integrator (IC₄) accomplish automatic gain control of Vᵢᵣ’s amplitude variations in the audio-frequency range.

The multiplier’s output is \(-VᵢᵣVᵧ/10\), where Vᵧ is a negative voltage generated by integrator IC₄. Together, the integrator and the rectifier extract the dc component (Vᵧ) of Vₒᵤᵣ for use as a feedback signal to the multiplier. The integrator sums signal current from the rectifier and control current from potentiometer R₉, which lets you adjust Vₒᵤᵣ’s signal level.
Circuit analysis yields the frequency-response equation

$$V_{OUT} = \frac{K_1AV_C}{10RC_3} \left( \frac{1}{s + \frac{10A}{RC_3}} \right),$$

or, in the time domain,

$$V_{OUT} = \left( \frac{K_1AV_C}{10RC_3} \right) \exp \left( -\frac{10At}{RC_3} \right).$$

In both equations, $K_1$ is the gain of amplifier IC2, $A$ is the peak amplitude of $V_{IN}$, and $R$ is the resistance between the integrator input and the rectifier output. (For this circuit, $R$ equals $R_6$ in parallel with $R_7$.)

This AGC circuit is suitable for controlling long-term variations of amplitude within a limited range. It doesn’t respond uniformly over a wide dynamic range, however, because the time response is inversely proportional to input-signal amplitude.

The arbiter circuit shown in Fig 1 allows two microprocessors to access a common RAM (not shown) through address buffers and data transceivers. When either µP alone seeks access to the RAM, the circuit introduces a maximum delay of only 6 nsec.

If either µP requests access to the common RAM (by asserting its RAM-enable signal) while the other µP has access, the circuit will put the requesting µP on hold by asserting the appropriate wait signal. This signal deactivates when the first µP finishes (its RAM enable goes high) and allows the second µP to complete its memory cycle.

The data input of each flip-flop is wired high, so $Q$ outputs are high and $\bar{Q}$ outputs are low when a RAM access is not in progress. Then, when either µP requests access, the corresponding flip-flop’s outputs change state in response to a low on that flip-flop’s CLR input. As a re-
sult, the four output signals grant access to the requester and force the other \( \mu P \) to wait.

To prevent the circuit from forcing both processors to wait in response to near-simultaneous requests for access, you connect OR gate IC\(_{2A}\) to give one processor priority. To give priority to \( \mu P_1 \), for example, you connect the OR gate’s output to IC\(_{3A}\)’s PR input as shown. To give priority to \( \mu P_2 \), you’d make the connection shown by the dotted line. In either case, you connect the PR input of the low-priority \( \mu P \) to 5V.

Fig 1—This circuit arbitrates between two microprocessors requesting access to a common RAM. In the event of simultaneous requests, IC\(_{2A}\) grants priority according to a connection made by the designer.
Charger extends lead-acid-battery life

Fran Hoffart
National Semiconductor Corp, Santa Clara, CA

A circuit that properly charges sealed lead-acid batteries ensures long, trouble-free service. **Fig 1** is one such circuit; it provides the correct temperature-compensated charge voltage for batteries having from one to as many as 12 cells, regardless of the number of cells being charged.

The **Fig 1** circuit furnishes an initial charging voltage of 2.5V per cell at 25°C to rapidly charge a battery. The charging current decreases as the battery charges, and when the current drops to 180 mA, the charging circuit reduces the output voltage to 2.35V per cell, floating the battery in a fully charged state. This lower voltage prevents the battery from over-
charging, which would shorten its life.

The LM301A compares the voltage drop across $R_1$ with an 18-mV reference set by $R_2$. The comparator’s output controls the voltage regulator, forcing it to produce the lower float voltage when the battery-charging current passing through $R_1$ goes below 180 mA. The 150-mV difference between the charge and float voltages is set by the ratio of $R_3$ to $R_4$. The LEDs show the state of the circuit.

Temperature compensation helps prevent overcharging, particularly when a battery undergoes wide temperature changes while being charged. The LM334 temperature sensor should be placed near or on the battery to decrease the charging voltage by 4 mV/°C for each cell. Because batteries need more temperature compensation at lower temperatures, change $R_5$ to 30Ω for a TC of −5 mV/°C per cell if your application will see temperatures below −20°C.

When the circuit charges more than six cells, the additional voltage across the LM334 increases self-heating, so use a small heat sink and increase the the resistance of $R_5$. Likewise, use higher resistances in series with the LEDs to avoid overloading the LM301A.

The charger’s input voltage must be filtered dc that is at least 3V higher than the maximum required output voltage: approximately 2.5V per cell. Choose a regulator for the maximum current needed: LM371 for 2A, LM350 for 4A, or LM338 for 8A. At 25°C and with no output load, adjust $R_7$ for a $V_{OUT}$ of 7.05V, and adjust $R_8$ for a $V_{OUT}$ of 14.1V. EDN

**MOSFET circuit debounces power relay**

Carlo Vendetti
The Charles Stark Draper Laboratory Inc.
Cambridge, MA

You can eliminate turn-on bounce in an electro-mechanical relay by connecting an n-channel power MOSFET across the relay terminals (Fig 1). The MOSFET’s high peak-current rating (108A for the unit shown) allows it to carry the load current when the relay contacts bounce open. As an additional benefit, the low on-resistance of the relay carries most of the current, eliminating the need for a heat sink on the relay.

Fig 1—This circuit operates a power MOSFET ($Q_4$) in parallel with the terminals of an electromechanical relay. $Q_4$ debounces the relay by conducting load current when the relay terminals bounce open.
MOSFET.

When the relay is off ($S_1$ open), a trickle of current through the relay coil biases $Q_3$ on, which in turn holds $Q_2$, $Q_1$, and MOSFET $Q_4$ off. Closing $S_1$ quickly toggles the state of all four transistors, turning on $Q_4$ before the first contact bounce occurs. The 15V zener diode ($D_1$) protects the MOSFET by clamping $V_{GS}$ at a level below its maximum rating.

Fig 2a shows the sharing of the load current between the MOSFET and the relay for a resistive load, $Z_L$, of 2Ω. A second scope photo shows the total load current through $Z_L$ (Fig 2b) for the same operating conditions. Note the absence of discontinuities in this waveform.EDN

![Fig 2](image)

**Fig 2**—These scope photos show the operation of Fig 1's circuit with a 2Ω load ($Z_L$). In a, $Q_4$ and the relay conduct load current alternately as the relay contacts bounce. In b, the lower trace shows that the resulting current through $Z_L$ is free of discontinuities.
Converters yield droop-free S/H circuit

TG Barnett
The London Hospital Medical College,
London, UK

In low-frequency applications, many monolithic sample/hold circuits suffer a droop rate that can cause an unacceptably large output error. The S/H circuit in Fig. 1 eliminates droop error by operating two 8-bit multifunction converters back to back. The circuit requires a 5V supply and accepts analog inputs between 0 and 2.5V (although you can scale and offset any input signal to fall within this range).

The analog input is applied to the in-
vertis the input of an LM324 op amp (IC1), which is wired as a comparator. The op amp and the IC2 multifunction converter form a ramp-and-compare A/D converter. (Because the Ferranti ZN435 multifunction converter includes a voltage-output D/A converter, an 8-bit up/down counter, a 2.5V bandgap reference, and a clock generator, you can configure the device either as an A/D or as a D/A converter.) The converter’s internal counter counts from 0, producing a positive-going ramp at the analog output.

When the ramp voltage exceeds the analog input, the comparator output goes high and sets IC5’s Q1 output high, thus inhibiting IC3’s clock and stopping the counter. IC3’s digital outputs are connected to the digital inputs of IC2, which is wired as a D/A converter. The D/A converter provides the S/H circuit’s analog output.

The output will remain in a hold state until you reset the monostable multivibrator (IC4), whose outputs apply simultaneous reset pulses to IC2 and IC5. The circuit then resamples and holds a new value of analog input. The S/H circuit provides 8-bit hold accuracy for analog input frequencies as high as 1 kHz; you can use a faster op amp for IC1 for higher-frequency operation.

The diode-capacitor network of Fig 1 accepts low current at a high voltage and delivers higher current at a lower voltage, behaving like a step-down transformer. You drive the circuit with a square-wave input signal as shown.

When the input is at its peak voltage, $V_p$, current through $D_{10}$, $D_7$, $D_4$, and $D_1$ charges series capacitors $C_4$, $C_3$, $C_2$, and $C_1$. The voltage on each capacitor reaches approximately $V_f(V_p - 4V_f)$, where $V_f$ is the forward-voltage drop across one diode. Consequently, the circuit is inefficient for low-amplitude drive signals (too much voltage is lost across the diodes).

For 15V and 60V p-p inputs, the circuit’s corresponding outputs are approximately $-1.65V$ and $-12.9V$, depending on the load. An input of 28V p-p produces about $-5V$. Notice that the square-wave generator must sink more current than it sources: It charges the capacitors in series, but discharges them in parallel.

When the input terminal switches to 0V, it connects the capacitors in par-
allel by pulling, the positive side of each capacitor near 0V. The capacitor voltages then produce current flow that creates a negative charge across the load capacitor ($C_L$). The voltages on $C_1$, $C_2$, and $C_3$ each charge $C_L$ through two diodes in series, but the charging path through $C_4$ has only one diode, $D_{11}$. This configuration results in a higher surge current through $D_{11}$ and $C_4$ and a slightly higher negative output voltage, unless you add a diode in series with $D_{11}$.

You can change the output voltage by adding or subtracting sections; $C_1$, $D_1$, $D_3$, and $D_2$ constitute one section, for example. Make the series capacitors equal in value and the total value of these capacitors equal to the load capacitor:

$$C_L = \frac{I}{2V_{rF}f},$$

where $I$ is the load current, $V_{r}$ is the maximum allowed p-p ripple voltage, and $f$ is the input frequency.EDN

---

**Fig 1**—This diode-capacitor network converts an input square wave to a negative dc voltage.
Step-up converter produces 5V from 1.5V

Gerald Grady
Maxim Integrated Products, Sunnyvale, CA

You can produce a regulated 5V output from a 1.5V battery cell by using the step-up dc/dc-converter circuit shown in Fig 1. The circuit can operate with $V_S$ as low as 1V, but it requires at least 1.5V to start. The output can deliver 100 mA when $V_S$ is 1.5V or 1.7A at 70% minimum efficiency when $V_S$ is 3V.

Supply voltage for the switching regulator IC$_1$ appears first at pin 4 (start-up mode) and then at its own V$_{OUT}$ terminal, pin 5. The chip includes an oscillator, a bandgap reference, three voltage comparators, a catch diode, and associated control circuitry. An internal MOSFET lets you implement low-power applications; higher power calls for an external device: Q$_1$ in Fig 1. The on-chip oscillator provides a

---

**Fig 1**—This switching-regulator circuit converts a dc input (as low as 1.5V) to a higher regulated value—to 5V, for example.
55-kHz square-wave drive to both the internal and the external MOSFET.

When Q1 turns off, current through inductor L2 drives the drain-node voltage higher. Similarly, current through L3 drives the voltage at pin 5 higher when the internal MOSFET turns off. This action generates two independent voltages, each higher than V_S. Q1 and L2 generate sufficient overhead voltage to enable the regulator chip to produce a regulated 5V output, and the internally generated voltage ensures adequate gate drive to Q1. The internal voltage, clamped by 10V zener diode D3, ranges from 10V (turn-on) to 15V (normal operation). Q1’s resulting RDS(ON) is only 0.085Ω.

Resistors R1 and R2 determine the regulated output level. For V_REG outputs other than 5V, set

\[ R_1 = R_2 \left( \frac{V_{OUT}}{1.31} - 1 \right). \]

You choose an R2 value in the range from 10 kΩ to 10 MΩ.

For low values of V_S, losses in the internal and external diodes and the Q1 inductor sharply limit the maximum output current. The following design equations let you determine component values while calculating this current. First, Q1 must be able to handle the peak current I_{PK} of inductor L2:

\[ I_{PK} = \frac{V_S t_{ON}}{L_2}, \]

where \( t_{ON} = \frac{0.55}{f_{OSC}} \). For this circuit, then, \( I_{PK} = 1.07A \).

The circuit loss \( V_{LOSS} \) is

\[ V_{LOSS} = I_{PK} (R_{DS(ON)} + 2 R_{L2}) + V_{D2}, \]

where \( R_{L2} \) is the dc resistance of L2 and \( V_{D2} \) is the forward voltage drop of D2. For this circuit, \( V_{LOSS} = 0.56V \). The current output is

\[ I_{OUT} = \frac{0.5 L_2 I_{PK} t_{OSC}}{V_{REG} (V_S - V_{LOSS}) - I_{L_1}}. \]

Therefore, \( I_{OUT} = 103 \) mA for the circuit shown. Table 1 shows the output currents you can obtain for various values of \( V_S \) and \( L_2 \). The corresponding conversion efficiencies range from 70 to 85%. EDN

<table>
<thead>
<tr>
<th>( V_S )</th>
<th>( 14 ) µH</th>
<th>( 27 ) µH</th>
<th>( 50 ) µH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5V</td>
<td>105 mA</td>
<td>48 mA</td>
<td>30 mA</td>
</tr>
<tr>
<td>2.0V</td>
<td>220 mA</td>
<td>95 mA</td>
<td>60 mA</td>
</tr>
<tr>
<td>2.5V</td>
<td>390 mA</td>
<td>160 mA</td>
<td>100 mA</td>
</tr>
<tr>
<td>3.0V</td>
<td>1.75A</td>
<td>1.5A</td>
<td>1.25A</td>
</tr>
</tbody>
</table>

Table 1—Obtainable Values of I_{OUT Max}
V/I converter has zero $I_B$ error

Roberto Burani and Giovanni Stocchino
FATME SpA, Rome, Italy

In a conventional (simplified) voltage-to-current converter (Fig 1), $I_E = V_{IN} / R_P$, and $I_C = I_E - I_B$. Because $I_B = I_E / (1 + \beta)$, the output current is affected by changes in $\beta$, which varies with $I_E, V_{OUT}$, and temperature. The voltage-controlled current source of Fig 2 overcomes this drawback by eliminating the output transistor’s base current as a source of error. (For earlier voltage-to-current circuits, see EDN, September 15, 1983, pg 227, and January 10, 1985, pg 290.)

Notice that the output device in Fig 2 (a composite of Q2 and the optocoupler’s output phototransistor) has only two dc terminals, so $I_C$ and $I_E$ are identical at low frequencies. Output current, then, is proportional to current flowing in the optocoupler’s LED. The output device contributes only a negligible error due to current leakage (about 1 pA/V), which is caused by finite isolation resistance in the package.

To identify potential sources of error, consider the expression for output current:

$$I_C = I_E = \frac{V_{IN} \pm V_{OS}}{R_P} \pm \frac{I_B + 2I_{OS}}{2} = I_{DARK} + I_{OFF},$$

where $V_{OS}$ is the op amp’s input offset voltage, $I_B$ and $I_{OS}$ are the op amp’s respective input bias and offset currents, $I_{DARK}$ is the optocoupler’s dark current, and $I_{OFF}$ is the cut-

**Fig 1**—In this simplified V/I converter, $I_{OUT}$ equals $V_{IN}/R_P$ minus $I_B$, which varies with $I_E, V_{OUT}$, and temperature.
off current for transistor Q₂. Resistors R₅ and R₆ extend the output-current range by reducing I_{DARK} and I_{OFF} to a few nanoamperes.

The maximum deviation \(d\) of output current from the ideal \(\left(\frac{V_{IN}}{R_P}\right)\) is

\[
d = \left| I_C - \frac{V_{IN}}{R_P} \right| = \frac{V_{OS}}{R_P} + \frac{I_B}{2} + I_{OS}.
\]

You can control the major sources of error \(V_{OS}\) and \(dV_{OS}/dT\) by selecting a suitable op amp. (As intended, the quantity d contains no error contributions from the output device.)

For the circuit of Fig 2, a single programming resistor (identified as \(R_p\)) provides an output-current range of about six decades. (Note that this resistor’s TC is also a potential source of error; it dissipates 125 mW when \(V_{IN}=5\) V.) The maximum deviation is typically 50 nA—that is, 0.0002% of full-scale.EDN

**Fig 2**—This voltage-controlled current source uses an optocoupler (IC) to eliminate an error found in more conventional circuits and which is caused by the output transistor’s base current.
Direction detector doubles as decoder

Tracy Allen
EME Systems, Berkeley, CA

Fig 1’s circuit, which was developed to monitor the traffic of bumblebees into and out of the hive, differentiates “a-to-b” motion from “b-to-a” motion. When used with an optical decoder, the circuit distinguishes clockwise from counterclockwise rotation and provides a resolution of one output pulse per quadrature cycle. Although an earlier Design Idea achieves a resolution of four output pulses per quadrature cycle (“Improved tachometer eliminates backlash,” EDN, March 31, 1987, pg 210), Fig 1’s circuit is simpler and less expensive.
Q1 and Q2 are mounted so that a moving object first blocks one phototransistor, then both, then the other. Depending on the direction in which the object is moving, either IC1B or IC1D emits a negative pulse when the moving object blocks the second sensor. An object can get as far as condition 3 (see Table 1) and retreat without producing an output pulse; that is, the circuit ignores any probing or jittery motion. (If an object gets as far as condition 4, however, a retreat will produce an opposite-direction pulse.)

The time constants R3C1 and R4C2 set the output-pulse width. A 100-kΩ/100-pF combination, for example, produces 10-µsec pulses. You select a value for pullup resistors R1 and R2 from the, 10- to 100-kΩ range, according to the sensitivity your application requires.

### Table 1—Logic States for a-to-b Motion

<table>
<thead>
<tr>
<th>Condition</th>
<th>Inputs</th>
<th>Intermediate Variables</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) No Passage</td>
<td>a1 0</td>
<td>b1 0</td>
<td>a4 1</td>
</tr>
<tr>
<td>2) Object blocks a</td>
<td>a2 1</td>
<td>b2 1</td>
<td>a4 1</td>
</tr>
<tr>
<td>3) Object blocks a AND b</td>
<td>a3 1</td>
<td>b3 0</td>
<td>a4 1</td>
</tr>
<tr>
<td>4) Object blocks b</td>
<td>a4 1</td>
<td>b4 1</td>
<td>a4 1</td>
</tr>
<tr>
<td>5) No Passage</td>
<td>a5 0</td>
<td>b5 0</td>
<td>a4 1</td>
</tr>
</tbody>
</table>

*Interchange a AND b for b-to-a motion*
Simple circuit suits quadrature detection

SL Black and HL Maddox
AT&T Technologies, Columbus, OH

The circuit in Fig 1 generates an output voltage that you can measure to determine whether two sine waves have a quadrature relationship. If the output voltage is 0V, the inputs (Φ₁ and Φ₂) are exactly in quadrature. If the inputs are other than 90° out of phase, a dc voltage appears at the circuit’s output. The voltage is proportional to the number of degrees that the input signals are out of quadrature. The polarity of the voltage is positive for phase angles of less than 90° and negative for angles of greater than 90°.

The signals A and B in Fig 2 are in

Fig 1—The bilateral switch in this circuit allows you to determine whether two sine waves are in quadrature. If the output voltage is 0V, the inputs (Φ₁ and Φ₂) are exactly in quadrature. If the output voltage is positive or negative, the waves are out of quadrature.
quadrature. When A’s signal is applied to the $\Phi_1$ input, a bilateral CMOS switch turns on during the positive half-cycle and turns off during the negative half-cycle. If B’s signal is applied to $\Phi_2$ simultaneously, an output similar to that of C appears at pin 2. Note that the areas above and below ground are equal. The integrating network, $R_4C_1$ in Fig 1, produces a net voltage of 0V.

If the phase angle is $>$90°, the area above ground is larger than the area below ground, and the output voltage is positive (D). If the phase angle is $<$90°, a negative output voltage results (E). If the 4016 triggers at a value other than 0V, the detector’s accuracy will not change.

$R_3$, $D_1$, and $D_2$ provide input protection for the IC. The performance of the $R_4/R_5/C_1$ integrator depends on the frequency of the input signals and the impedance of the network at pin 1. If you choose 8.2 kΩ for $R_1$ and 2.2 kΩ for $R_2$, the values 8.2Ω, 4.7 kΩ, and 3.2 µF for $R_4$, $R_5$, and $C_1$, respectively, yield good performance at 25 kHz. These values will accommodate a 24V p-p swing at the $\Phi_2$ input. The values of $V_{DD}$ and $V_{SS}$ must be large enough to accommodate the input swings at the 4016. For example, an input swing of ±3V would call for 5V for $V_{DD}$ and −5V for $V_{SS}$.

---

**Fig 2**—When $\Phi_1$ and $\Phi_2$ are in quadrature, the output of pin 2 (C) manifests equal areas above and below ground, resulting in a 0V integrated output from pin 2. If the waves are out of quadrature, a positive (D) or negative (E) voltage appears.
By adding three common ICs to a µP circuit, you can generate 255 software-controlled tones. You can use these tones for audio indicators, to flash annunciator lamps at different rates, or to provide a sweep-generator-type output from your µP.

Unlike conventional designs in which the CPU is virtually dedicated to tone generation, Fig 1’s circuit lets you set the frequency, start the tone generation, and then use the µP for another task. The circuit uses a 555 timer connected in its astable mode. The µP’s active-high (through a latch) output port selects which capacitors determine the circuit’s operating frequency; a separate enable input from the µP turns the circuit on and off.

**Fig 1—Add three common ICs to your µP circuit to generate software-controlled tones.**
are grounded by the inverters. You choose the 555’s frequency of oscillation by grounding different combinations of capacitors. The latch holds the combination until you choose another. Pick the capacitor values to provide the desired frequency range and resolution. By using additional lines from the µP, you can control more capacitors and extend the circuit’s frequency range and number of possible frequencies.

Compute the period of oscillation using

\[
T = 0.699 (R_1 + 2R_2)(D_0C_1 + D_1C_2 + \ldots + D_7C_8),
\]

where \(D_0\) through \(D_7\) are the digital-data outputs from the µP’s output port. By supplying an active-high signal from the µP to the circuit’s enable input, you turn on the generator. In an 8085-µP system, you can use the SOD output to supply this signal.

The circuit in Fig 1 generates a 12-bit pulse that has a duty cycle that is adjustable from 0 to 100%. You can use the circuit as a building block in several applications. For instance, by adding an RC network to it, you can configure a linear 12-bit A/D converter. And if you place the device under computer control, you can use it as a function generator. You might also use it as a precise power controller in feedback applications.

In Fig 1’s circuit, \(IC_4\) divides the circuit’s 250-kHz clock signal by 4096. \(IC_5\) stretches \(IC_4\)’s pulse to generate a 60-Hz strobe pulse. This strobe pulse drives \(IC_6\)’s output high and loads the initial count (\(D_0\) through \(D_{11}\)) into three cascaded, 4-bit, binary down counters (\(IC_1\), \(IC_2\), and \(IC_3\)). The counters count down from the initial count to zero, clearing \(IC_6\) at count zero. The time it takes these counters to count down to zero and drive \(IC_6\)’s output low determines what portion of \(IC_5\)’s stretched pulse will appear at \(IC_6\)’s output, and therefore the circuit’s pulse width.

\(IC_6\)’s output is a 60-Hz pulse whose width equals integer/4096 times the pulse’s period, where integer is equal to the initial count and can have any value from 0 to 4096. This 60-Hz output gates \(IC_7\), a switchable voltage reference that improves the pulse’s voltage accuracy.
To vary the circuit’s precision and pulse width, you can modify the down counter, clock frequency, and frequency divider. For example, to obtain 16-bit precision, you can cascade an additional counter and divide the clock frequency by 65,536. Note, however, that unless you increase the circuit’s clock frequency proportionally, the circuit’s output pulse rate will be 3.8 Hz.

*Fig 1—Featuring a 0 to 100% adjustable duty cycle, this pulse-width modulator provides 12-bit precision.*
TV sync generator acts as clock timebase

Andrew Dart  
Trans-Texas Telegraph Co, Duncanville, TX, and  
Richard Kihn  
KSDM TV, Beaumont, TX

A digital-clock chip that requires a 60-Hz reference may drift three to four seconds every 48 hours when connected to the power line. The clock will maintain ±0.5-sec accuracy, though, if you derive its 60-Hz reference from a TV signal’s vertical sync pulses.

A TV station’s sync generator contains a temperature-stabilized crystal oscillator that governs the station’s horizontal- and vertical-sync pulse rates and color-subcarrier frequency. This oscillator in turn tracks the network signal, which is derived from atomic standards. The result is a stable subcarrier frequency:

$$f_{\text{SUBCARRIER}} = 5 \text{ MHz} \times \frac{63}{88} = 3.5795454545 \ldots \text{ MHz}.$$  

The station’s vertical-sync signal frequency is

$$f_{\text{VSYNC}} = f_{\text{SUBCARRIER}} \left( \frac{2}{455} \right) \left( \frac{2}{525} \right) = 59.9400599400 \ldots \text{ Hz}.$$
So, Fig 1a’s circuit multiplies $f_{\text{VSYNC}}$ by 1.001 to obtain an exact 60-Hz reference frequency $f_{\text{REF}}$.

$$f_{\text{REF}} = \frac{5,000,000(63)(2)(2)(1001)}{1(88)(455)(525)(1000)} = 60 \text{ Hz}.$$ 

The circuit has been in continuous use for 10 years without causing any problems. Using currently available components, however, you can get the same performance from a simpler circuit (Fig 1b).

The circuit in a converts the vertical-sync signal of 59.94 Hz to a clock-reference frequency that averages 60 Hz. The circuit in b performs the same task but uses fewer parts.
Capacitor maintains data during power loss

Stephen Crum
Service Machine Co, Huntington, WV

Capacitive backup can provide a simple, low-cost alternative to battery backup for retaining data during temporary power loss. Fig 1’s circuit uses an electric, double-layer, 1F capacitor to provide backup. The capacitor occupies only 1.1 in.² of board space.

The backup circuit uses a 5V supply to charge the 1F storage capacitor (C₁) via R₁ and D₁. Upon receipt of a power-fail interrupt, the circuit’s μC (a CMOS MC146805) enters a low-power wait state. In this state, the μC draws only 200 μA (at 3V), and all dynamic circuitry except for the internal timer is disabled. The circuit also backs up the μC’s 112 bytes of RAM.

To allow you to determine how long power has been lost, the circuit includes a low-power pulse generator (a CMOS 555 timer). The timer clocks the μC’s internal timer via pin 37. Using this configuration, you’ll be able to record down times as long as an hour. The accuracy of this counting technique is limited by the pulse generator’s accuracy over the capacitor’s usable discharge range (4.5 to 2.5V).

EDN
Circuit monitors RS-232C communications

Roy Griffin
M/A-Com, El Paso, TX

You can use the circuit in Fig 1 to monitor an RS-232C link’s transmit and receive pins. When a negative marking voltage appears on P1’s pin 3, optocoupler IC1 connects D3’s cathode to P1’s pin 2. To monitor the data, you can attach a terminal (if the data is in ASCII format) or a microcomputer to P3. P3 is configured as a null modem.

P3’s pin 2 connects to pin 3 via R1 and supplies a negative marking voltage to the monitor. This connection also allows the monitor terminal to echo data to itself. (You will find this feature useful for stop-
Monitor thermocouples electronically

Marc Saberi
Masscomp, Westford, MA

Thermocouples are rugged and inexpensive temperature sensors, but their use in accurate temperature-measurement applications can involve a cumbersome procedure. You can simplify the procedure, however, by electronically eliminating the effect of uncontrolled temperature at the cold end (Fig 1). Further, you can perform a mathematical operation on the output voltage (preferably after conversion to digital form) to remove the thermocouple’s nonlinearity.

Traditionally, you maintain a thermocouple’s cold junction in an ice bath to allow a direct association of output voltage with temperature. Alternatively, you can sense the junction temperature and compensate the output voltage. In Fig 1, the instrumentation amplifier’s VREF input terminal offers a convenient way to perform the sensing and compensation. The amplifier’s transfer function is

\[ V_{OUT} = (V_{IN} \times \text{gain}) + V_{REF} \]

where \( V_{IN} \) is the thermocouple output voltage, \( V_{REF} \) is the compensation voltage.

Fig 1—Make accurate temperature measurements with a thermocouple by electronically sensing the cold-end temperature and using that information to adjust amplifier output.
from the LM35 temperature sensor, and gain is equal to 200,000/R. To eliminate the influence of the cold-end temperature, each term to the right of the equals sign in the equation must have the same slope in mV/°C; choose R to set the instrumentation amp’s gain to the appropriate value.

For example, the variation of \( V_{\text{REF}} \) is 10 mV/°C; that of a J-type thermocouple is 54 µV/°C. Therefore,

\[
\text{gain} = 10 \, \text{mV} + 54 \, \mu\text{V} = 185.2
\]

and

\[
R = 200,000 + \text{gain} = 200,000 + 185.2 = 200,185.2 \, \Omega.
\]

The thermocouple’s temperature-vs-voltage relationship isn’t linear, however, and you can approximate its transfer function using a power-series polynomial:

\[
t = a_0 + a_1 \cdot V + a_2 \cdot V^2 + a_3 \cdot V^3 + \ldots + a_n \cdot V^n,
\]

where \( t \) is thermocouple temperature, \( V \) is thermocouple voltage, \( a_n \) represents the polynomial coefficients, and \( n \) is the order of the polynomial. The National Bureau of Standards publishes lists of these coefficients for each type of thermocouple (Table 1). Omega Engineering’s Temperature Measurement Handbook, pg T-11, is one source of these lists.

To complete the sample calculation, suppose \( V_{\text{OUT}} = 1.2256 \, \text{V} \) in Fig 1 for a J-type thermocouple whose cold end is at 25°C (LM35 output is 0.25V). For the first equation,

\[
V_{\text{IN}} = (V_{\text{OUT}} - V_{\text{REF}}) + \text{gain}
\]

\[
= (1.2256 - 0.25) + 185.2 = 0.005268 \, \text{V}.
\]

Substituting the National Bureau of Standards’ coefficients and \( V = 0.005268 \) into the power-series polynomial yields a temperature difference of 100°C, so the thermocouple temperature is 125°C. EDN
The C program in Table 1 implements the Soundex algorithm (Ref 1), which finds words that sound alike. By assigning similar-sounding words the same code, the routine enables a computer to compare alternative spellings of a word and thereby detect spelling errors. For example, the words Grapple, Gravel, Grappelli, and Grapall would all receive the code G614.

The Soundex algorithm divides the alphabet into seven groups (Table 2). Group 0 contains vowels as well as consonants that are often used as vowels. The other six groups contain letters that sound alike.

Because the sound of a particular vowel depends on its context, the Soundex algorithm ignores group 0. Because the user is
unlikely to begin a word with the wrong letter, the algorithm copies the first letter of the word directly to the Soundex output. When the algorithm encounters double letters, it saves only the first letter of the pair. The algorithm works only with letters; it terminates when it encounters any other type of character.

The program converts an input string to its corresponding Soundex string. Note that C strings are terminated with a null (zero) character. The function TOUPPER, normally provided as a standard component in a C library, converts lowercase letters to uppercase letters. The function TOASCII, also part of the library, maps a character to its corresponding ASCII representation (this function is not neccessary on a computer that uses the ASCII character set). The operator “*” indicates that the program is using a pointer to perform an indirection operation. The prefix “++” indicates that the program should increment the pointer before it uses that pointer.

The Soundex routine doesn’t work in all cases. For example, the word “tough” wouldn’t receive the same code as the word “tuff.” In most cases, however, the routine provides a simple method of comparing alternative spellings of words.

Reference
Simple routines provide calendar date

Robert D Grappel
MIT Lincoln Laboratory, Lexington, MA

Two subroutines allow a computer to represent calendar dates as 16-bit unsigned integers. The computer can easily store and manipulate this integer; the alternative is to handle dates as people do (month-day-year or day-month-year), using cumbersome tables (“Thirty days hath September ... ”). The two subroutines are presented first in Basic (Fig 1) to illustrate how they work, and then in Motorola 68000 assembly language (Fig 2).

Lines 10 through 80 in Fig 1 convert conventional date information to a 16-bit day number; lines 100 through 210 convert the day number back to the day-month-year format. Conversions are
accurate for dates between March 1, 1900, and December 31, 1999. Note that 1461/4 and 153/5 approximate the number of days in an average year and month, respectively. Lines 40 through 80, line 150, and lines 180 through 200 adjust for February and the 30-day months. Lines 140 and 170 ensure proper rounding of D, the day variable; lines 130 and 160 perform a modulo function.

The two Fig 2 routines use only the registers D0 to D2 and require no other storage. In addition, the routines are position-independent and fully re-entrant (i.e., they can be interrupted and will later resume where they left off). The simple integer math in these routines is easily executed by any 16-bit processor.

EDN
Resistors provide nonlinear pot tapers

Mark Rumreich
RCA Corp, Indianapolis, IN

For some applications of nonlinear potentiometers, you can avoid the expense of a custom potentiometer by adding fixed resistors to a conventional linear potentiometer. For the variable-resistor or rheostat mode, an external resistor (R' in Fig 1a) gives you the curves of Fig 1b. Note that varying \( \alpha \) also varies the maximum resistance (R+R'). For convenience, all values of the variable resistor are normalized by R + R'.

For the voltage-divider mode of operation, you can add two resistors (Fig 2a). Fig 2d shows the relationship of the normalized wiper voltage (V_{OUT}/V_{IN}) to the

Fig 1—Adding R' to linear potentiometer R (a) results in a nonlinear relationship (b) between the normalized resistance and the wiper position.
wiper position for the case $\beta=0.5$, in which $R_1=R_2$. Note that the curves’ point of intersection equals $\beta$, which you can shift from 0 to 1 by changing the values of $R_1$ and $R_2$. Note also that such a shift changes the circuit’s loading effects.

Figs 2b and 2c show the voltage divider’s behavior for the cases $\beta=0$ and $\beta=1$. (Interpret $\beta=0$ to mean that $R_1$ is omitted; $\beta=1$ means $R_1$ is omitted.) The combination $\alpha=5$ and $\beta=1$, for example, provides an excellent modified log with 20% taper (20% taper means 20% of maximum resistance at 50% of the wiper travel). The combination $\alpha=10$ and $\beta=1$ provides a reasonable approximation of a semilog (audio) taper.

**EDN**

---

**Fig 2**—You can make a nonlinear voltage divider (a) by adding resistors to a linear potentiometer. The curves in b, c, and d show respective results for the three cases $\beta=0$ (omit $R_1$), $\beta=1$ (omit $R_2$), and $\beta=0.5$ ($R_1=R_2$).
Sync separator flags odd fields

Luen-Yuan Lin
DRS, Precision Echo Inc, Santa Clara, CA

In 1982, *EDN* published three Design Ideas that showed how to separate the horizontal and vertical sync pulses from a video composite-sync waveform (Refs 1, 2, and 3). The circuit in Fig 1 also separates the sync pulses, and identifies odd vs even frames as well, by producing a field index pulse (E) for each odd field. Component values are suited for operation on an RS-170 standard waveform (A), but the circuit will operate with other interlaced-scan waveforms if you modify the time constants (t₁ and t₂) associated with one-shots IC₁A and IC₁B.

The RS-170 standard defines one video frame as two fields of 262.5 lines each. A slight difference in the fields’ vertical-in-
Interval timing allows them to be distinguished electronically.

Falling edges of the composite-sync input (A) trigger the IC1A one-shot, producing output (B) at the horizontal sync frequency. The one-shot sets the low intervals \( t_1 \) in this waveform to \( 0.5H < t_1 < H \), where \( H \) is the 63.5-µsec time interval between horizontal lines. Similarly, one-shot IC1B triggers on the rising edges of waveform (B) to produce a waveform (C) in which the low intervals \( t_2 \) are less than 0.5H.

Flip-flops IC2A and IC2B sample the input (A) on the rising edges of their clock inputs (waveforms (C) and (B), respectively). The resulting field index signal (E) remains high during even fields (Fig 2), but pro-

Fig 2—Waveforms show no change in Fig 1’s output (E) at the start of an even field. The dashed lines indicate the pulse that would occur without a reset signal to IC2A.
duces a negative pulse (Fig 3) for odd fields. Use of the vertical frequency (D) as a reset to flip-flop IC2A makes the circuit work; the dashed lines indicate the pulses you will get if this connection is omitted.

**References**


*Fig 3—The field index output (E) in Fig 1 emits a negative pulse only at the beginning of an odd field.*
Digital AGC is fast, stable, and accurate

Kerry Lacanette
National Semiconductor Corp, Santa Clara, CA

The AGC (automatic-gain-control) circuit in Fig 1 maintains a constant output-voltage level of 10V p-p for repetitive waveforms. The circuit senses the input amplitude and adjusts the output gain once every cycle, returning the output amplitude to its proper level within two cycles. (Conventional AGC circuits respond much more slowly to avoid modulating the gain within one cycle of a low-frequency input signal.)

The digital approach in Fig 1 has other advantages, as well. Using an A/D converter for input-amplitude sensing eliminates the conventional gain-control...
element (a FET, a saturated bipolar transistor, a light-dependent resistor, or a voltage- or current-controlled amplifier), which can introduce distortion, noise, and gain error. Using a CMOS D/A converter provides accurate gain control and introduces less than 0.1% distortion. Further, the open-loop circuit isn’t subject to instability.

AGC action improves with input amplitude, which should be between 1 and 10V. AGC control is better than 0.1 dB for amplitudes above 5V p-p and about 0.4 dB for a 1V p-p input. The input frequency range is 1 Hz to 5 kHz.

Op amps IC$_{2A}$ and IC$_{2B}$ constitute a peak-detector circuit in which the hold capacitor (C$_1$) captures the peak amplitude during a negative half-cycle of the input signal. (Transistor Q$_1$ discharges C$_1$ during the input’s positive half-cycle.) Waveform A drops to 0.7V as the input passes its negative peak value, as you can see in Fig 2.

Comparator IC$_{3A}$ senses the zero crossing that follows the input signal’s negative peak, causing NAND gates IC$_{4A}$ and IC$_{4B}$ to generate a negative WR pulse that triggers a conversion by IC$_{5}$, the 8-bit A/D converter. The diode-clamped amplifier, IC$_{1A}$, protects the comparator from overdrive (input

Fig 2—These analog and digital waveforms depict the operation of Fig 1’s AGC circuit.
diodes protect the other comparator, IC₃B.

IC₃B’s output drives the A/D converter’s RD signal low when the input signal is negative, placing the converter’s output on the data bus. Next, NAND gates IC₄C and IC₄D generate a negative WR₁ pulse that latches the data into the D/A converter IC₆, thereby adjusting the gain of the output amplifier IC₁B to oppose any change in the input signal’s amplitude. Fig 3 illustrates this action.EDN

Fig 3—The top trace shows Fig 1’s AGC action. Following a change in input amplitude, the output amplitude returns to the proper level within two cycles.

By using a plug-in adapter board (Fig 1), you can replace a 16-bit 68000 CPU with the 32-bit 68020 CPU, which offers a larger instruction set and faster program execution. The resulting system requires some modification of operating-system software but no additional hardware changes.

The adapter board plugs into the 68000 socket as an emulator does. Because the 68020 can operate on a 16-bit data bus as well as a 32-bit data bus, wiring modifications consist only of connecting D₀ through D₁₅ of the 68000 to D₁₆ through D₃₁ of the 68020 data bus. All other address (A₁ through A₂₃) and control signals with corresponding names are wired one for one; address lines A₂₄ through A₃₁ are unconnected.

The 68020 requires six additional signals: AVEC, E, VMA, LDS, UDS, and DSACK₁. AVEC, E, VMA, and DSACK₁ are generated with proper timing relationships by logic gates on the adapter board. LDS and UDS are generated by decoding the SIZ₀, SIZ₁, and A₃ signals and qualifying them with DS.

Keep in mind that the two processors handle exception processing differently. The 68020 stacks as many as 46 words during exception processing, but the 68000 stacks only three words. In the worst case, the CPU will halt if software doesn’t compensate for this difference. Also, the 68020 generates the address strobe (AS) on the clock’s falling edge (state 1), but the 68000 generates AS on the rising edge (state 2). If your hardware timing relies on the latter, you must invert the clock signal (CPU-CLK) to the 68020.EDN
Fig 1—This 68020 adapter board plugs into a 68000 socket to provide the benefits of the more powerful 32-bit processor.
A simple solenoid driver (Fig 1) uses incandescent-lamp filaments to limit power consumption. As a welcome side effect, the lamps also serve as on-indicators.

High magnetic reluctance (opposition to flux) in the coil of an armature-driven device (solenoid, relay, etc) calls for a surge of activation current, followed by a lower dc level to remain on (surge to on-current ratio is typically 5:1). You can roughly approximate this current requirement by connecting a lamp filament in series with the coil. The cold filament allows a surge of coil-activation current to pass through; as the filament heats up, it throttles the current to a more reasonable hold value.
The solenoid driver circuit in Fig 1 offers these features:

- 5V logic swings turn the power-MOSFET switch (Q1) fully on and off.
- Two low-cost flashlight lamps in parallel handle the peak current. Because their dc current is only 50% of peak and because they operate at 60% of their rated voltage, the lamps have an operating life of 12,000 hours. Further, the lamp filaments positive temperature coefficients raise each filament’s resistance, which eliminates current-hogging problems and provides short-circuit protection.
- Steady-state on-current is 700 mA (vs 1700 mA without the lamps).
- A 4.6V min supply rating allows battery operation.

Schottky diodes D1 and D2 clamp the DUT’s output within 300 mV of ground, and the JFET buffer transistor (Q1) minimizes capacitive loading. IC1, a fast hybrid op amp, amplifies the buffer’s output with a gain of (1+ Rf/Rg). Two more Schottky diodes, D3 and D4, clamp the circuit’s output voltage to minimize the effect of saturation in the scope’s input amplifier.

The resistor values shown produce an output that is 10× that of the DUT. This gain allows measurement of 1-mV error signals (0.01% of a 10V step), which in turn lets you use a digitizing oscilloscope—the HP54100A, for example, which has a maximum resolution of 10
mV/div.

The remaining components form an auto-
zero circuit that reduces the relatively large
V_{OS} of IC₁. By offsetting the bias voltage for
Q₁, the components force the average output
offset to less than 500 µV (the input offset
voltage of IC₂).

During testing, the DUT’s output pro-
duces a 10V step in response to each 10V
transition of the pulse-generator signal and
immediately slews back and settles at about
the 0V level (Fig 2). The DUT output sup-
plies load current through R₄ (via the pulse
generator) as the waveform begins slewing.
The load current then drops nearly to zero
as the DUT’s output voltage enters the
±300-mV clamp band set by D₁ and D₂. By
observing this voltage on an oscilloscope,
you can measure op-amp settling times of
less than 400 nsec, to within ±0.01%.

Note that R₄ doesn’t load the DUT during

---

**Fig 1**—By driving the op amp’s supply voltages with a pulse generator, this circuit refers the op-amp output’s settling waveform to
ground for 10V as well as –10V output steps.
its settling phase. To simulate a load in a real application, you may want to connect a resistor or an RC network between the DUT’s output and its virtual ground (connected to the noninverting input). Note also that ringing and long settling tails on the pulse generator’s waveform have a critical effect on measurement accuracy. The test setup allows the DUT’s common-mode rejection (CMR) and power-supply rejection (PSR) to reject some of these aberrations, but the pulse generator should produce a clean waveform whose transitions are much faster than the DUT’s slew rate. Your pulser may or may not produce such a signal without a 50Ω termination, for example.

You can automate your settling-time measurements by using a programmable, digitizing oscilloscope. First, observe the circuit output while triggering on either positive or negative edges of the input square wave. To reduce the effects of noise, you must set the scope so that it averages 20 to 30 readings for each data point. A controller will then read the data, starting at a point well beyond the DUT’s expected settling time. The first few points establish the settled, dc output level, and the controller sets an error band around this level—say ±0.01% (Fig 3). (For Fig 1’s circuit, this band is ±10 mV, that is, 1 mV times the gain of IC1.) The controller then tests successive earlier data points until it encounters one outside the error band. The preceding point marks the end of the settling-time interval; the trigger signal marks the beginning.

**References**

Robert Grappel
MIT Lincoln Laboratory, Lexington, MA

The C-language function shown in Fig 1a returns the binary equivalent of its Gray-code argument; >> is the right-shift operator, and ^= is the exclusive-OR operator. Fig 1b gives the equivalent Motorola 68000 assembly-code subroutine; register D0 handles the C version’s Value variable, and register D1 handles the C version’s Stemp variable. When you use this subroutine with 16-bit data, it requires only 160 clock cycles.

The C program that converts binary code to Gray code (Fig 1c) is even simpler than the Gray-to-binary program. You can even substitute a macroinstruction for this function. The assembly-language version is relatively simple, too (Fig 1d).

Fig 1—C- and assembly-language routines (a and b) convert Gray code to binary code. The routines in c and d convert binary code to Gray code.
Control duty cycle with up-down counters

Einar Abell
ADA Instruments, Burlington, VT

The three up-down counters and a dual JK flip-flop in Fig 1 form a divider that produces an output with a digitally controlled duty cycle; you can adjust the duty cycle from 0.1 to 99.9%. The circuit requires two fewer ICs than are needed in more traditional circuits based on magnitude comparators.

The counters undergo a cycle of counting to 999 from a preset value and then to zero from the same value. For the counters to cycle in this fashion, you must connect the ripple-clock output of the most significant counter so that it loads the counters and toggles the JK flip-flop (IC1b) that controls the count direction.

Because the above circuitry alone produces a count cycle of only 999 clock pulses, use a second flip-flop (IC1a) to disable the counters for one pulse when the counting direction changes from down to up; this additional pulse yields the desired division ratio of 1000:1. R3 and C2 provide enough delay for the ripple-clock output to load the counters.

Fig 1—Using two fewer ICs than traditional duty-cycle controllers require, this circuit produces an output with a digitally controlled 0.1 to 99.9% duty cycle. You can easily change the count length by adding to or subtracting from the three up-down counters (IC2, IC3, and IC4) or by switching to binary counters.
Circuit monitors duty cycle

Gerald Mallonee
Litton Industries G/GS, Woodland Hills, GA

The circuit shown in Fig 1 monitors and displays a digital signal’s duty cycle and provides accuracy as high as ±1%. Using a switch (S2), you can choose a frequency range of either 250 Hz to 2.5 kHz at ±1% accuracy or 2 kHz to 50 kHz at ±10% accuracy. The common-cathode display gives the signal’s duty-cycle percentage.

Phase-locked loop IC4 and a counter (IC5A and IC5B) multiply the input frequency by a factor of either 10 or 100, depending on switch S2’s setting. IC6A and IC6B count this multiplied frequency during the incoming signal’s mark interval. IC7 and IC8 then latch this count and display it at the clock’s sample rate. For example, if you select a 1% resolution, when the signal’s mark period is 40% of the total period, the circuit will enable the counter comprising IC6A and IC6B for 40 counts.

To obtain space-interval sampling, you can reverse the input polarity using switch S1. IC3A samples the input signal’s period and enables gate IC3C and resets the counter. IC2E and IC2F form the sample-rate clock; IC3B synchronizes the clock’s output with the input so that the circuit can update latches IC7 and IC8.

---

**Fig 1** — This circuit will continuously monitor a digital signal’s duty cycle; its accuracy is ±1% from 250 Hz to 2.5 kHz and ±10% from 2 kHz to 50 kHz.
To combine two video sources in one CRT display (such as a VCR signal and a computer-generated graphics overlay), you must first synchronize their timing waveforms. Fig 1’s circuit accomplishes this task by extracting timing information from one source for use in the other. You provide only a composite-video input to the circuit.

Most of the work is then done by the video-sync separator IC1, which restores dc to the video at pin 2, separates the composite-sync signal and provides this output at pin 1, and provides a vertical-sync signal at pin 3 that’s precisely synchronized with the vertical interval’s first serration. IC1 provides a field index pulse at pin 7 (for interlaced video sources) that’s high during the even field (1) and low during the odd field (2).

To extract the color-subcarrier signal, IC1 issues a burst-gate output (pin 5) that turns on the transistor amplifier, Q1, only during the composite waveform’s color-burst interval. Op amp IC2A amplifies the frequency burst and applies it to the 3.579545-MHz crystal, Y1. Because the input resistance of op amp IC2B sets a high Q for the crystal circuit (approximately 10k), the crystal continues to ring between bursts, providing a constant subcarrier output.

**Fig 1—This circuit strips the timing information from a composite-video signal for use in synchronizing the signal with another video source.**
Perform a Z80 reset without memory loss

Leo M Almazan
Naval Ocean Systems Center, San Diego, CA

The Z80 microprocessor includes circuitry for refresh of its dynamic RAM. If you initiate a manual reset of the Z80 µP during a refresh cycle (they typically occur every 2 msec), you could lose an entire row of data. By using the circuit in Fig 1, however, you can initiate a manual reset at any time.

Fig 1’s circuit provides a reset without interfering with refresh activity; instead, it introduces the reset during an op-code fetch. The circuit synchronizes the reset with a negative-going transition of the Z80’s M/1 (machine cycle one) output signal. Gate IC3 accommodates the normal power-up reset routine, as well.

In Fig 1, IC1A and IC1B debounce the manual switch and present a negative-going transition to IC2A (a monostable multivibrator). IC2A’s output goes high, allowing M/1’s next negative-going transition to produce a negative reset pulse at the output of IC2B.

Fig 1—This circuit lets you perform a manual reset of the Z80 µP at any time, without losing data from the Z80’s dynamic RAM. The circuit delays the manual reset so that it coincides with an op-code fetch, thus avoiding interference with the refresh cycle.
Written in Commodore 64 Basic, the program in Listing 1 gives you the log magnitude (in dB) and the phase (in degrees) for any generalized transfer function of the form

\[ H(s) = \frac{A_0 + A_1s + \ldots + A_ns^n}{B_0 + B_1s + \ldots + B_ms^m} \]

where \( s \) is the complex-frequency variable. You can use this program for evaluating new filter designs.

As an example, the program in Listing 2 provides the inputs required to describe a third-order, elliptic, lowpass filter with a 3-kHz passband, a 50% reflection coefficient, and a 25° modular angle (an elliptic-filter term). After execution, the program sends its output to the computer's printer port. This data (Listing 3) tells you that the filter’s passband ripple is approximately 1.25 dB and that its stopband attenuation is –40.5 dB at 14 kHz.

Continued on pg 57
LISTING 2

TRANSFER FUNCTION EVALUATION
ORDER OF NUMERATOR? 2
A(0) = -7.311E12
A(1) = -7.0
A(2) = -7.120E3
ORDER OF DENOMINATOR? 3
B(0) = -7.311E12
B(1) = 7.415E6
B(2) = 7.169E4
STEP FREQ (Hz)? 0
END FREQ (Hz)? 600
STEP FREQ 0 600

LISTING 3

<table>
<thead>
<tr>
<th>FREQUENCY (Hz)</th>
<th>DECIBELS</th>
<th>DEGREES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>-21.3401827</td>
<td>127.097772</td>
</tr>
<tr>
<td>200</td>
<td>-.0516809483</td>
<td>-9.5667617</td>
</tr>
<tr>
<td>400</td>
<td>-.195662332</td>
<td>-18.4082625</td>
</tr>
<tr>
<td>600</td>
<td>-.409594752</td>
<td>-27.7207364</td>
</tr>
<tr>
<td>800</td>
<td>-.633984143</td>
<td>-36.0802568</td>
</tr>
<tr>
<td>1000</td>
<td>-.929479748</td>
<td>-43.9442207</td>
</tr>
<tr>
<td>1200</td>
<td>-1.09011364</td>
<td>-51.4179619</td>
</tr>
<tr>
<td>1400</td>
<td>-1.21511911</td>
<td>-58.6766154</td>
</tr>
<tr>
<td>1600</td>
<td>-1.36425687</td>
<td>-65.3774441</td>
</tr>
<tr>
<td>1800</td>
<td>-1.5166461</td>
<td>-73.6463229</td>
</tr>
<tr>
<td>2000</td>
<td>-1.6964045</td>
<td>-82.1472173</td>
</tr>
<tr>
<td>2200</td>
<td>-2.09056595</td>
<td>-92.1248526</td>
</tr>
<tr>
<td>2400</td>
<td>-2.4653208</td>
<td>-104.438087</td>
</tr>
<tr>
<td>2600</td>
<td>-6.04649186E-03</td>
<td>-119.95732</td>
</tr>
<tr>
<td>2800</td>
<td>-2.3372916</td>
<td>-138.705771</td>
</tr>
<tr>
<td>3000</td>
<td>-1.25079436</td>
<td>-158.656817</td>
</tr>
<tr>
<td>3200</td>
<td>-3.02777849</td>
<td>-176.586788</td>
</tr>
<tr>
<td>3400</td>
<td>-5.2335481</td>
<td>-196.210057</td>
</tr>
<tr>
<td>3600</td>
<td>-7.52786681</td>
<td>-215.52055</td>
</tr>
<tr>
<td>3800</td>
<td>-9.78432703</td>
<td>-235.483697</td>
</tr>
<tr>
<td>4000</td>
<td>-1.1395166</td>
<td>-255.40593</td>
</tr>
<tr>
<td>4200</td>
<td>-1.3962059</td>
<td>-275.433687</td>
</tr>
<tr>
<td>4400</td>
<td>-1.5934067</td>
<td>-295.478126</td>
</tr>
<tr>
<td>4600</td>
<td>-1.798892</td>
<td>-315.51997</td>
</tr>
<tr>
<td>4800</td>
<td>-1.95957878</td>
<td>-335.462402</td>
</tr>
</tbody>
</table>
Pulse generators offer improved performance

With certain modifications and additions to a circuit (Fig 1) from an earlier Design Idea (Barnett, TG, “Pulse generator has variable duty cycle,” EDN, October 4, 1984, pg 249), you can enhance the performance of a variable-duty-cycle pulse generator.

If only ±15V power supplies are available, it’s better to attenuate the output of the duty-cycle potentiometer (Fig 2b) than to reduce the voltage across the potentiometer (Fig 2a). Either method produces the desired ±1V output range, but Fig 2b offers as much as a $\times 14$ improvement in rejecting asymmetrical changes in the supply voltages.

The voltage follower IC<sub>C</sub> in Fig 1 is not essential for buffering the potentiometer. If desired, this op amp could instead be used to provide an inverted output. Or, using a second potentiometer, it could generate an output with a different duty cycle.
Some hysteresis via positive feedback (obtained by adding 1-kΩ and 1-MΩ resistors, as shown in Fig 3) will improve the output stability and shorten the rise and fall times. The resistors allow a clean response for output frequencies as low as 0.02 Hz.

By replacing Fig 1’s 2-op-amp (ICA and ICB) triangle-wave gen-

Fig 2—The divider shown in b is preferable to that shown in a because it provides more rejection for asymmetrical changes in the supply voltages.

Fig 3—Improve the low-frequency response of the output stage in Fig 1 by adding 1-kΩ and 1-MΩ resistors to provide hysteresis.

Fig 4—This pulse generator uses one dual op amp and provides independent control of frequency and duty cycle. In addition, it tolerates an imbalance of 3V or more in the power supplies.
erator with a generator based on one op amp, you can build an economy-version pulse generator using a dual op amp (Fig 4). Although Fig 4’s triangle waveform consists of exponential rather than linear segments, the resulting error is small because the timing capacitor charges to only 10% of maximum on each cycle. This circuit tolerates power-supply variations—a 3V change in either supply causes little effect on the duty cycle.

Finally, you can sweep or otherwise modulate the duty cycle by replacing the potentiometer with a suitable ac signal. The deluxe-version swept-pulse generator in Fig 5 uses a second low-frequency wave generator for the sweeping.

Fig 5—A deluxe version of the pulse generator offers a swept duty cycle as an option.
Pulse-delay circuit has dual-edge trigger

S Murugesan
ISRO Satellite Centre, Bangalore, India

In Fig 1, a single monostable multivibrator delays a pulse train by a variable amount; nonetheless, this amount can be no less than the minimum allowed pulse width $t_w$. This approach simplifies an earlier Design Idea (EDN, June 26, 1986, pg 225) by lowering the parts count.

The exclusive-OR gate IC₁ generates a short pulse following every leading or falling edge of the input waveform. These pulses cause one-shot IC₂ to produce a negative-going pulse with a duration equal to the desired time delay $t_D$, which you set by adjusting the potentiometer (R). Flip-flop IC₃ then creates a delayed replica of the input pulse by latching the $\overline{Q_1}$ output of IC₂ between positive-going transitions.

You can independently control the output-pulse duration by cascading a second one-shot with the first. EDN

Fig 1—This circuit delays the input pulse train by a time interval $t_D$, which is less than or equal to the minimum pulse width $t_w$. 
Subroutines handle speech companding

R Garrick and K Siil
The Logistics Software Group, Sunnyside, NY

You can use a compression subroutine (Listing 1) in tandem with an expansion subroutine (Listing 2) to increase channel bandwidth in a 68000-based speech-processing system. The compression routine takes 14-bit sign-magnitude PCM data from location IN_DAT, logarithmically compresses it, and returns 8-bit sign-exponent-mantissa data to register D0. Transmission of 8-bit data—rather than 14-bit—allows you to reduce the channel bandwidth by 43%. The expansion routine then takes 8-bit data from location COMP_DAT, transforms it back to 14-bit sign-magnitude form, and returns the expanded result to register D2.
The subroutines contain algorithms based on µ-law companding. Fig 1 shows how these algorithms transform an 8-bit digital word to 14 bits and back. The compression routine, for example, must first mask and store the input data’s sign bit. The routine then adds a bias to the input value to ensure a minimum 1 at bit position 5. The next step locates the leading 1 in the sum (which determines the exponent value) and takes the next four bits as the mantissa. Finally, the routine combines sign, exponent, and mantissa bits in an 8-bit word as shown in the flow chart (Fig 2a).

The simpler expansion routine first masks and stores the sign, mantissa, and exponent bits. Next, it pads the mantissa with a leading and trailing 1 and shifts the result left, according to the value of the exponent. After subtracting the original bias from this expanded data, the routine adds the sign bit to form a 14-bit output value. Although the algorithm transmits only a 4-bit mantissa, it achieves 5½-bit accuracy because it adds the leading 1 and because the trailing 1 represents a median value for the truncated bits.

Substituting two look-up tables for a series of

---

**LISTING 2—µ-LAW EXPANSION ROUTINE**

```plaintext
; INPUT: 8 BIT COMPRESSED SIGN-EXP-MANTISSA DATA AT LOCATION COMP_DATA
; OUTPUT: 14 BIT EXPANDED SIGN-MAGNITUDE DATA IN REGISTER D2
; EXPAND:
CLR.W DO ; INITIALIZE OUTPUT WORD
MOVE.W #0,9.D0 ; GET COMPRESSED DATA
MOVE.W #0,9.D1 ; COPY DATA FOR SIGN EXTRACTION
ANDI.L.W #15.D1 ; MASK AND STORE SIGN
AND.W #15.D1 ; POSITION SIGN IN OUTPUT
MOVE.W #0,9.D2 ; COPY INPUT FOR MANTISSA MASKING
AND.W #0,9.D2 ; MASK AND STORE MANTISSA
LSR.W #4,D0 ; PREPARE MANTISSA FOR PACKING W/ 1'S
OR.W #0,9.D0 ; ADD LEADING AND TRAILING 1'S
AND.W #7,DO ; MASK EXPONENT
LSR.W #4,D0 ; RIGHT JUSTIFY EXPONENT
LSR.W #2,D0 ; EXTRACT EXPONENT AND MANTISSA
Sub.W #3,DO ; REMOVE BIAS
OR.W D1,D0 ; ADD SIGN TO EXPANDED DATA
RTS
```

---

**BIASED COMPRESSION (ENCODING) TABLE**

<table>
<thead>
<tr>
<th>F</th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
<td>M₀</td>
<td>M₀</td>
<td>M₀</td>
<td>M₀</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
<td>M₀</td>
<td>M₀</td>
<td>M₀</td>
<td>M₀</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
<td>M₀</td>
<td>M₀</td>
<td>M₀</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
<td>M₀</td>
<td>M₀</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
<td>M₀</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
<td>M₁</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M₂</td>
</tr>
<tr>
<td>X</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**BIASED EXPANSION (DECODING) TABLE**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |
| S | 0 | 0 | 0 | M₂ | M₁ | M₀ | M₀ | M₀ | M₀ | X | X | X | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | M₂ | M₁ | M₀ | M₀ |

Fig 1—These encoding and decoding tables show the result of the subroutine operations in Listing 1 and Listing 2.
lengthy linear or binary bit searches speeds execution of the compression routine by returning values for the leading 1 and the exponent. The compression subroutine requires 156 clock cycles; the expansion subroutine requires an additional 144 cycles. Thus, operating the 68000 with a 12.5-MHz clock and a tight executive program loop provides ample bandwidth for one processor to handle full-duplex voice communications.EDN

Fig 2—One flow chart (a) illustrates the algorithm for Listing 1’s compression subroutine, and the other (b) shows Listing 2’s expansion subroutine.
IC emulates many types of logic gates

Kurt W Christner
Carson City, NV

Normally, you’d use a 4053-type triple-spdt analog switch (also called a triple, 2 channel analog multiplexer/demultiplexer) for analog signal-switching applications. When properly connected, however, the CMOS device can also emulate a variety of simple gates and flip-flops. What’s more, you can save space with a 4053 by packing three types of simple logic gates into one DIP.

Fig 1, for example, shows the 4053’s implementations and truth tables for eight logic functions. The inverter, buffers, and gates use one spdt section each; the flip-
flops use two sections each. Note that the buffers are bidirectional because the 4053’s input-to-output path is a symmetrical transmission gate. Also, unlike conventional buffers, they have an output impedance (on-resistance) of several hundred ohms.

In Fig 2, the 2-input gates require the complement of one input; if the complement isn’t available, though, you can generate it using a \( \frac{1}{4} \) 4053 as an inverter. The circuits of Fig 1 and Fig 2 will work with the \( V_{EE} \) terminal connected either to ground or to a negative supply voltage, provided that the 4053 and the external logic are operating from the same \( V_{DD} \) and \( V_{SS} \) supply levels. (Ed Note: The 4053 requires supply voltages \( V_{DD} \) and \( V_{SS} \) pins 16 and 8), plus a bias supply \( V_{EE} \) pin 7.)

The level-shifting circuits of Fig 3 shift the \( A_{IN} \) signal from \( V_{SS} - V_{DD} \) levels to \( V_{EE} - V_{SS} \) or \( V_{EE} - V_{DD} \) levels. \( V_{EE} \) must connect to a negative supply voltage in these circuits.

In these applications, the 4053’s inhibit input (pin 6) must connect to \( V_{SS} \). Also, the response of the 4053-based circuits is generally slower than that of equivalent CMOS digital gates. Even though the 4053’s propagation delay from switch input to switch output is faster—about one-fifth that of a CMOS gate—the delay from a 4053 control input is approximately twice that from a CMOS-gate control input.
By combining a $10 analog multiplier with a $2 wideband op amp (Fig 1), you can make a high-speed spst switch that has a 50-nsec response and a bandwidth that exceeds 30 MHz. The switch uses both channels of an AD539 multiplier to provide two features: the option of inverting or noninverting inputs and the elimination of the switching pedestal caused by step changes in output current as the multiplier is gated on or off.

Gain is approximately 1 (0 when off). The output can drive ±1V into a 75Ω load or ±2V into a 150Ω load. An output-offset adjustment is optional.

The Fig 2 waveforms were taken...
across a 75Ω termination. In Fig 2a, the output response to a 0 to 1V, 1-MHz sine wave shows the switching completed in about 50 nsec. Feedthrough of the control signal is minimal; feedthrough of the sine-wave signal (off isolation) is approximately –55 dB, determined largely by parasitics in the board layout. In Fig 2b, response to a 0 to 1V pulse in the signal channel shows a rise time of less than 10 nsec (control input held at 0V). Response from the inverting input is similar.

Differential gain (less than 0.5 dB) and differential phase (less than 0.5° at 3.58 MHz) are compatible with video applications. Output noise with a 75Ω load is 200 μV typ in a 0- to 5-MHz bandwidth, or approximately 100 nV√Hz. The noise spectral density is essentially flat to 40 MHz.
Square-root algorithm is fast and simple

Robert D Grappel  
MIT Lincoln Laboratory, Lexington, MA

Most software routines for the calculation of square roots are based on the Newtonian-iteration algorithm or the shift-and-subtract algorithm. Shift-and-subtract routines are usually written in assembly language and are therefore fast, whereas the Newtonian-iteration routines are easy to express in high-level languages but run somewhat slower. A third algorithm, however, is fast and is easy to express in a high-level language, as well.

The basis for this algorithm is a mathematical theorem stating that a perfect square is equal to the sum of a sequence of odd integers, and the number of terms in the sequence gives the integer square root. (For example, the sum of the first five odd integers is 25.) The following C function shows how easily you can express this algorithm in high-level form:

```c
int square_root(int arg)
{
    int sqi;
    if (arg < 0) return 0; /* special cases */
    for (sq = 1; sq < arg; sq += 2)
    
        arg -= sq;
    return (sq + 1)/2; /* round count */
}
```

This routine quickly returns a square-root value for small arguments but slows as you increase the size of the argument.

You can also efficiently code this algorithm in assembly language, as illustrated in the following subroutine for the 68000 µP:

The calling routine passes the unsigned 32-bit argument to the subroutine in register D0, and the subroutine returns the square root in that same register; all other registers are preserved. This routine requires about 100 clock periods for an argument of 4290 periods for an argument of 100 and 1480 periods for an argument of 4096. For comparison, a 68000 shift-and-subtract routine (Dr Dobb’s Journal, May 1985, pg 122) requires between 1480 and 1832 periods, depending on the size of the argument, and it doesn’t preserve the working registers.
Reset circuit solves brownout problems

Damian G Bonicatto
Hibbing Electronics Co, Hibbing, MN

Most microprocessor systems include some form of power-on reset circuit, such as the simple RC network shown to the right of the dashed line in Fig 1. The circuit is satisfactory provided that $V_{cc}$ drops low enough for the capacitor to discharge below the Reset line’s minimum input-low level (0.8V, for example). The Reset will never trigger, though, if $V_{cc}$ drops only to, say, 2V and then returns to its normal level. Yet the µP may act erratically while $V_{cc}$ is low.

You can cure this brownout problem by adding $Q_1$ and $R_2$ (shown to the left of the dashed line in Fig 1). $Q_1$ is a programmable unijunction transistor (PUJT) whose operation is similar to that of an SCR. With these additions, $C_1$ supports the anode voltage as $V_{cc}$ drops, and the PUJT will trigger when its gate drops about 0.7V below the anode (unless the $V_{cc}$ change is very gradual). $C_1$ discharges when the PUJT triggers, which ensures a reset by pulling the Reset line low.

![Fig 1—This brownout circuit ensures a reset for $V_{cc}$ excursions that are only 0.7V below nominal.](image-url)
Convert square waves to sine waves

Arnold L. Newman and Henry Riblet, Johns Hopkins University, Laurel, MD

Using a few gates, four MOSFETs, and a few discrete components, you can build Fig 1’s square-wave-to-sine-wave converter. Two pairs of MOSFETs form a bridge that alternately switches current in opposite directions. Two parallel-resonant LC circuits complete the converter.

The L₁/C₁ combination is resonant at the fundamental frequency; the L₂/C₂ combination is resonant at the clock frequency’s third harmonic and acts as a trap. R₁ and C₃ ensure that both halves of the MOSFET bridge are never on at the same time by providing a common delay to the gate drive of each half. You select the values of R₁ and C₃ to yield a time constant that’s less than 5% of the clock’s period. You can add an output amplifier for additional buffering and conditioning of the circuit’s sine-wave output. EDN
A CMOS version of the familiar 555 timer (Fig 1) can monitor battery voltages in the 2 to 15V range while drawing little current (less than 80 µA at 12V, for example). The alarm trigger is based on the nonlinear variation of the CMOS version’s pin 5 voltage with respect to changes in the V+ supply. The output normally equals V+ but drops to 0V (RET potential) when V+ falls below a threshold value. Moreover, an on-chip comparator controls the output, and separate resistive dividers bias each of the comparator’s inputs.

An external divider (R1, R2, and R3), whose variation with regard to V+ is
small, negative, and linear, biases the noninverting input (Fig 2). An internal divider whose variation with respect to $V^+$ is large, negative, and nonlinear, on the other hand, biases the comparator’s inverting input (pin 5). (The bipolar 555’s pin 5 variation, by comparison, is small, positive, and linear.) Consequently, you can set the potentiometer $R_2$ to provide a small bias between the comparator’s inputs—say, –100 mV. Then, different rates of change at pins 5 and 6 will reduce the bias as $V^+$ falls and will cause the comparator to trip near 0V bias.

Fig 1’s waveforms illustrate the voltage monitor’s operation. Pin 7 (DS, or discharge) is an open-drain output that acts simultaneously with pin 3 and that is suitable for driving an LED or another alarm indicator as shown. The alarm will remain on until you reset it manually via a pushbutton switch, or you can connect the autoreset resistor $R_T$, which resets the circuit following an interval, $T=1.1R_TC_T$. The alarm will promptly reactivate, of course, unless $V^+$ has risen above its trip level.

To calibrate the monitor, set $V^+$ at the desired minimum level and adjust $R_2$ until the alarm just trips. Then, return $V^+$ to its nominal level for normal operation. Fig 3 illustrates the performance of the voltage monitor. If, for example, the nominal $V^+$ is 9V and you adjust $R_2$ for a comparator bias of –0.08V (ie, pin 6 voltage with respect to pin 5 voltage), then the alarm will trip at approximately 6V. $R_3$ should be half the size of $R_1$ to obtain the maximum adjustment range at the wiper of $R_2$. $C_1$ and $C_2$ suppress the alarm’s response to noise in the $V^+$ supply; values should be less than 0.005 µF.

The voltage-monitor circuit depends on the nonguaranteed, nonlinear characteristic of the CMOS timer; in most timer applica-
tions, however, the CMOS version’s nonlinear characteristic is of no consequence. Further, the variation of comparator bias with respect to temperature will depend only on the internal divider if you use metal-film resistors in the external divider. Published specs indicate the bias will change less than 1% over the IC’s operating temperature range.

**Fig 3**—For three values of normal operating supply voltage, you can predict the approximate threshold at which Fig 1’s circuit will trip as V increases. Resistor values should be high to minimize power dissipation.

It’s well-known that, using a single op amp, you can build an all-pass phase shifter that provides unity gain and variable phase over the approximate range of 40 to 176° (Fig 1). To provide digital control, though, you can’t easily replace resistor R1 with a D/A converter because the resistor is not referenced to ground.

**Fig 2**’s circuit offers the same phase-shifter function plus digital control of phase with 10-bit resolution. The op amp (IC1) is connected as an adder-subtractor that forces V3 to equal V4 + V IN. This action in turn impresses V IN between the input and the output of the integrator formed by op amp IC2, capacitor C, and the resistance (R DAC) of the D/A converter (IC4). The integrator thus resolves V IN into two orthogonal vectors because the phase of V 3 and V 5 must differ by 90°. V OUT is the sum of V 3 and V 5, yielding the same transfer function as the circuit shown in Fig 1.

**Fig 1**—This conventional all-pass phase shifter offers unity gain. By adjusting the potentiometer, you can vary the output phase from about 4 to about 176°.

Robert A Pease
National Semiconductor, Santa Clara, CA
where p is the Laplacian operator \( \frac{d}{dt} \).

Because the D/A converter offers a high resistance when most bits are off, the input bias currents of IC2 should be low to minimize the output offset; the op amp’s offset voltage should be low for the same reason. You should add the offset-adjust potentiometer if IC2’s offset exceeds about 0.5 mV. For IC1 and IC3 you can use FET-input or bipolar op amps, although the higher slew rate of many FET-input types will let you handle faster signals.

With the components shown, Fig 2’s circuit covers a range of 4 to 176° or 0.1 to 89.9° at a fixed frequency, and it can maintain a fixed phase shift over a frequency range of about 1000:1. When covering the 0 to 90° range, for example, the circuit delivers a resolution of 0.11° per LSB or better, using a value of 0.01061 µF for C. For the range 0 to 180°, and using a value of 331.6 pF for C, the resolution is 3.58° per LSB. Alternatively, you could use a 12-bit D/A converter and a 166-pF capacitor to obtain 1.79° per LSB.

Applications for this circuit include the cancellation of unwanted phase shifts in control systems and the generation of odd sound effects in audio systems. By cascading two or three phase shifters, you can “wobble” the phase controls to create special effects.
The circuit described here expands the Z80 μP’s 64k-byte memory space to 512k bytes by switching among 16 32k-byte banks. The approach allows you to run programs larger than 64k bytes for applications that can tolerate an I/O operation each time the program crosses one of the 32k-byte boundaries.

The Z80’s lower 15 address lines (Fig 1) provide access to the 32k-byte common bank of memory. The 16th line (A15) signals any attempt to access a memory location above 7FFF hex (the 32k-byte limit) by activating four additional bits (A15 through A18). Consequently, the system can access $2^{19}$, or 512k, unique memory locations.

Your program must perform a 4-bit I/O

**Fig 1**—You can add an external bank-select register (IC4) to increase Z80 memory space from 64k bytes to 512k bytes.
write operation each time you access program steps or data in another memory bank. During this operation, IC₂ decodes A₂ through A₄ and strobes IC₃, which latches the data bits D₀ through D₃. Meanwhile, a logic one on A₁₅ switches the quad 2-1 data selector (IC₄), causing these data bits to appear at A₁₅ through A₁₈. **Fig 2** shows the physical addresses produced by combining the Z80 address with different values of IC₄, which functions as a bank register. You can return from any memory bank to the common bank (0000 HEX through FFFF HEX) without an I/O operation by simply issuing a Z80 address below 8000 HEX.

A reset signal latches 0000 into IC₂ and sets the Z80 to address 0000 HEX, yielding a starting address of 00000 HEX. You should also note that the Z80’s MI line must enable IC₂ as shown to prevent a spurious signal to the IC₂ latch circuit during an interrupt-acknowledge cycle. **EDN**
Fig 1 illustrates a handy device for detecting the invisible IR radiation found in fiber-optics systems, position sensors, and TV remote-control units. The device is built on top of a 9V battery and held in place with RTV, and its power dissipation is virtually zero unless IR radiation or high ambient light is present.

Normal fluorescent lighting is not a problem, but if necessary you can add an IR filter to the Q2 detector to exclude ambient light. Exposing the detector to a strong light or an IR source gives a quick check of the battery and the red LED.

Fig 1—This simple IR detector turns on a red LED when Q2 is exposed to IR radiation.
Biquad filter offers adjustable $f_0$ and $Q$

Prabhas Gangopadhyay  
Hagersten, Sweden

A switched-capacitor bandpass filter (Fig 1) gives you independent control of output center frequency ($f_0$) and selectivity ($Q$). The LF356 op amps used in the circuit are well-suited to this application because they remain stable while driving capacitive loads as high as 0.01 µF. Each switch uses half of a CD4016 quad bilateral switch IC, driven by nonoverlapping clock signals CLK₁ and CLK₂ from the clock generator (Fig 2).

Use the following relationships to calculate the values for $C_1$, $C_5$, $C_6$, and $C_7$, based on a value of 1 for capacitors $C_2$, $C_3$, and $C_4$:

\[
C_1 = 2Q
\]
\[
C_5 = 2\tan(\pi f_0/f_s)
\]
\[
C_6 = C_7 = \sin(2\pi f_0/f_s),
\]

Fig 1—Four op amps and four quad-switch ICs make up a switched-capacitor bandpass filter with independent control of center frequency and $Q$. 
where \( f_s \) is the frequency of the nonoverlapping clock signals and \( f_o < f_s/2 \). The gain of this filter at \( f_o \) is 0 dB.

You can set \( Q \) by adjusting the \( C_1 \) value. In addition, values for \( C_5 \) and \( C_6 \) determine the \( f_o : f_s \) ratio, allowing you to control \( f_o \) by adjusting the clock frequency.

For example, assume \( f_s = 64 \) kHz, and you require \( Q = 2 \) and \( f_o = 10 \) kHz. The equations yield \( C_1 = 4 \) nF, \( C_5 = C_6 = 2.8 \) nF, \( C_7 = 2.9932 \) nF, and \( C_6 \) and \( C_7 = 2.3282 \) nF. You might scale these values by a convenient constant, such as 2.8, and obtain \( C_1 = 4 \) nF, \( C_5 = C_6 = 2.8 \) nF, \( C_7 = 2.9932 \) nF, and \( C_6 \) and \( C_7 = 2.3282 \) nF.

**Fig 2—Using three ICs, you can generate nonoverlapping clock signals to drive the switches of the circuit in Fig 1.**

**Storage-time tester uses two ICs**

Al Pshaenich and Ron Backhaus
Motorola Power Products Div

Using two CMOS, quad 2-input NOR gates, you can build a circuit that tests a transistor’s storage time (Fig 1). If the transistor’s measured storage time (\( T_s \)) is less than the maximum allowed storage time, the pass light (a green LED) goes on. If the storage time is greater than the allowed time, the fail light (a red LED) goes on.

Gates IC1A and IC1B form a one-shot multivibrator (one-shot A). This one-shot supplies a 50 µsec test pulse that turns on the transistor under test (TUT) for 50 µsec. The fail light goes on as soon as the circuit receives a start pulse.

Gates IC2A and IC2B form a one-shot multivibrator (one-shot B) that generates a 4-µsec reference pulse after the first one-shot shuts off. This reference pulse is equal to the TUT’s maximum allowed storage time. One-shot B sends this reference pulse to a phase detector (gate IC2D), which compares the allowed storage time with the actual storage time.

While \( Q_1 \) (the TUT) is on, its collector is high and gate IC2C is on. When one-shot A no longer supplies a turn-on signal to \( Q_1 \) (one-shot A has timed out), \( Q_1 \) shuts off, its collector goes high, and gate IC2C goes low. The time that it takes IC2C to go low once \( Q_1 \) has lost its base drive is equal to \( Q_1 \)’s actual storage time. If this time is less than 4 µsec, both inputs to gate IC2C will be low and
The flip-flop comprising gates IC2C and IC2D will be set, which will turn on the pass light. If Q1’s storage time is greater than 4 µsec, however, gate IC2C will not supply a zero input to gate IC2D until after one-shot B has timed out (one-shot A will now be supplying a one to gate IC2D), and the flip-flop will be reset, which will keep the fail light on. Fig 2 gives a timing diagram for the circuit. Resistors R1 and R2 attenuate the collector-to-emitter voltage required for resistor turn-off to a level that’s compatible with gate IC2C. Clipping diode D1 limits the inductive flyback voltage that the resistive divider generates during collector transitions. You can apply a start pulse to this system either manually, with a pushbutton switch tied to the 5V supply rail, or with a pulse generator. Although the circuit shown uses a 5V supply, it will work with supplies as high as 15V (CMOS VDD max=18V). You can use one of several configurations for the power amp. You should, however, use an amplifier with a fast turn-off time; a slow amp could erroneously extend the TUT’s storage time.

EDN
MOSFET provides glitchless power backup

The power-backup circuit of Fig. 1 switches from $V_{CC}$ to the battery voltage without a glitch when $V_{CC}$ power is lost. $Q_1$ acts as a current source during normal operation to provide the battery with a constant charge current. The loss of $V_{CC}$ drops $Q_1$’s gate voltage to 0V, which allows battery current to flow through the MOSFET’s channel and integral body diode to the load.

For most p-channel MOSFETs, the gate-threshold voltage ranges from −2 to −4V, and thus $Q_1$ may not turn on fully for the 5V backup shown. As a result, $V_{CCM}$ is nearly a diode drop less than the battery voltage. To back up higher values of $V_{CC}$ (6 to 12V), use a higher battery voltage, so that $Q_1$ is fully on when its gate voltage drops to 0V. ($V_{CCM}$ equals the battery voltage minus the product of channel resistance and load current.)

To calibrate this circuit, connect an ammeter in series with the battery. Then, with $V_{CC}$ and the output load present, adjust $R_1$ for a battery-charging current slightly higher than that recommended by the manufacturer.

---

**Fig. 1**—MOSFET $Q_1$ provides a constant charging current to a battery during normal operation; the loss of $V_{CC}$ turns $Q_1$ more fully on, delivering battery current to the load.
Talking meter gives dc-voltage readings

Ricardo Jimenez-G  
San Diego State University, Calexico, CA  
and Francisco Meza and Jose J Lara  
Mexicali Technological Institute, Mexicali, Baja California, Mexico

Fig 1  

The circuit is a low-cost ($30) dc voltmeter that measures a positive 0 to 12.7V input and then voices the result in English. The meter can automatically monitor a dc voltage, thereby freeing a user for other tasks. Its resolution is ±0.1V.

Resistors R1 and R2 attenuate the input voltage, and an 8-bit A/D converter (IC4) converts the result to a decimal equivalent at the outputs DB0-DB6. This 7-bit word drives the EPROM’s upper address lines A6-A12, selecting a block of memory within the EPROM. Counter IC3 then scans those memory locations in sequence by driving the lower address bits A0-A5. As a result, the EPROM delivers a preprogrammed sequence of instructions to the speech-processor chip (IC6).

Timer IC2 is configured as a monostable monostable. When you depress the test switch, S1, the monostable generates a 1.1-msec pulse that sets the Q output of flip-flop IC7 high. The resulting negative transition at the speech processor chip’s ALD input (pin 20) loads the current EPROM output and causes the processor to assert a low logic level at the SBY output (pin 8). This action changes the IC8A output to a logic one, causing the processor to hold SBY low for an interval appropriate to that particular allophone. Note that you must connect an audio amplifier and speaker to the output, as indicated.

The processor initiates the next allophone cycle by driving SBY high. Each audible report requires three to 20 allophones, which you can get from the dictionary that accompanies the speech-processor package (available at Radio Shack). In essence, you must program the EPROM in 200 blocks of 3 to 20 bytes each.
An input of $A_6-A_{12}=1000000$ (corresponding to a 0.1V input), for example, produces the words “zero point one” from the audio amplifier. You store the allophones representing these words in the EPROM as shown in Table 1. After each report, the hex-data instructions 4 and 44 internally reset the speech processor, and via the processor’s 06 output they reset the counter and the flip-flop as well.EDN
Improved tachometer eliminates backlash

Andrew Dart  
Trans-Texas Telegraph Co, Duncanville, TX

The tachometer circuit of Fig 1 interprets the quadrature square-wave signals from an optical encoder. This circuit performs the same function as that of an earlier Design Idea (“Bidirectional tachometer offers low error,” EDN, October 30, 1986, pg 204), but it reduces the IC count from 11 to three and produces twice as many pulses per encoder revolution. It also eliminates backlash: The FWD and REV outputs indicate the direction of encoder rotation at every transition of the tachometer signals A or B, regardless of how little the encoder rotates before changing direction.

The encoder shown is a Hewlett-Packard HEDS-7500; HP recommends Schmitt-trigger buffers IC1A and IC1D to...
improve noise immunity of the tachometer signals A and B. The remaining two pairs of inverters each form a 10-µsec delay line; therefore, a logic transition on either the A or the B signal applies a momentary address at the input of the four- to 16-line decoder, IC2.

In turn, the decoder asserts a momentary low on one of its output lines (Table 1). Each of A’s and B’s eight possible states involving a logic transition produces a unique address, and the corresponding outputs drive the appropriate NAND gate (IC3A or IC3B). During the delay interval following a transition, the state of the other tachometer signal indicates the direction of encoder rotation.

**Fig 2—These timing diagrams illustrate normal operation (a) and a special case in which the encoder shaft wobbles before coming to a complete stop (b).**
The timing diagrams of Fig 2 show normal operation (a), and they show a special case (b) in which the encoder shaft wobbles after it stops, generating a signal on B but not on A. In most cases, the FWD and REV outputs will register even these brief changes in direction.

You can track the encoder’s motion using a counter with separate up and down inputs (Fig 3a) or a counter with one clock input and a separate up/down control line (3b). For slightly more money and a slight increase in power dissipation, you can replace IC₂ and IC₃ with a small PROM.

Fig 3—To monitor the optical encoder’s shaft rotation, one circuit uses a counter with separate up/down inputs (a), and the other uses a counter with a clock input and direction inputs (b).