Many modern systems operate with multiple power supplies that must meet tight tolerances to accommodate high-end µPs and peripherals. For instance, PCI applications may use as many as four supplies: 5V, 3.3V, and ±12V. The circuit in Figure 1 can monitor all these voltages for an undervoltage condition and can also monitor the three positive voltages for an overvoltage condition. The LTC1536 has a built-in ability to monitor the 3.3 and 5V levels and one adjustable level for undervoltage conditions. The adjustable input of the IC monitors the 12V supply. The LTC1536 also acts as the logic input for the open-drain outputs of the LTC1444 quad comparator.

Comparators A, B, and C of the LTC1444, along with resistors R1 through R6, monitor the 3.3, 5, and 12V positive supplies, respectively, for an overvoltage condition. The nominal overvoltage trip points are 3.61, 5.28, and 12.6V. Comparator D, aided by resistors R7 and R8, monitors the −12V supply for an undervoltage condition. The LTC1536 issues a reset until 200 msec after the fault condition goes away. (DI #2403).

To Vote For This Design, Circle No. 411

Circuit monitors quad supply for PCI systems
Roger Zemke, Linear Technology Corp, Milpitas, CA

Two ICs monitor PCI-system quad supplies for undervoltage and overvoltage fault conditions.
Circuit controls multiple thermoelectric coolers

By Frank Effenberger, Bellcore, Morristown, NJ

Optoelectronic and other components sometimes use a thermoelectric cooler and a thermistor for temperature control. A typical thermoelectric cooler has 1A maximum current and 1Ω impedance. These parameters make simple series-pass drive circuits inefficient, given the typical available supply voltages (5, 12, or 15V). Often, several thermoelectric coolers exist in a single circuit (for example, a multiple-wavelength optical transmitter). In this case, you can obtain improved efficiency by using the circuit in Figure 1. This circuit has three devices to cool, so it contains three coolers and three thermistors. On the sensor side, each thermistor connects to a standard proportional-integral-differential op-amp or µP-based-controller circuit. Each controller produces a command voltage that is proportional to the current its respective cooler requires.

The command voltages drive precision rectifier circuits (op amps IC1 to IC3) that generate the maximum control V_{MAX}. Op amp IC4 limits V_{MAX} making the maximum current delivered to the coolers. This limited output, V_{LIM}, is the input to the main transistor-driver amplifier, IC5. V_{LIM} commands the main transistor to pass a current corresponding to the largest demand, subject to the limit constraint. The thermoelectric coolers connect in series, and this chain receives its current via the main series-pass transistor Q4. IC8 drives this transistor such that the voltage drop across R_{SENSE} equals V_{LIM}. You choose R_{SENSE} to equal the impedance of one cooler. The coolers have shunt transistors (Q1 through Q3) that divert any excess current around the individual coolers. The shunt transistors receive their drive from op amps IC5 through IC7, which are connected as difference amplifiers. The op amps drive the shunt transistors such that the voltage across each thermoelectric cooler equals the command voltage for that cooler.

Because the sense resistance equals that of the coolers, the current passing through each cooler assumes the correct value. The actual values of the difference-amplifier resistors, R, and the transistor-base resistors depend on the type of op amps and transistors you select. In most cases, the desired cooler currents have an average value, I_{MEAN}, and a maximum value, I_{MAX}, that are almost equal. If you cool N coolers using separate circuits, the supply needs to deliver N\times I_{MEAN}. Using the series-connected circuit, the supply current reduces to just I_{MEAN}. Even in the case in which you optimize the supply voltage for either circuit, the power dissipation for the series-connected circuit is lower because it drops a smaller fraction of the supply voltage through the series-pass transistor. (DI #2395).

Figure 1

A series connection of thermoelectric coolers provides more efficient temperature control and fewer supply-current requirements than circuits using individual cooler controllers.

To Vote For This Design, Circle No. 412
Current-input ADC measures voltages
Jim Todsen, Burr-Brown Corp, Tuscon, AZ

A practical realization of a spread-spectrum technique lowers a µP’s clock-related EMI by approximately 4 dB without the drawbacks associated with modulation (Figure 1). The spread-spectrum technique is a popular method to reduce µP-clock related EMI (Reference 1). Using this method, the µP’s clock frequency constantly shifts around and creates a moving target for quasipeak EMI detection. Although this method dramatically reduces measured EMI, it has a few drawbacks.

The first drawback is an unpredictable clock frequency. Peripheral devices that share the same clock with the µP and rely on a stable clock frequency might suffer. One example is an ADC that relies on direct µP control to define the sampling time. The second drawback is the periodic nature of the frequency shift. The technique essentially modulates the clock frequency with an approximately 50-kHz frequency. This frequency is slightly higher than the audio band to prevent audio “hum.” In some systems, however, this 50-kHz modulation frequency may be in band with data-acquisition or other sensitive analog circuitry. Under these circumstances, separate nonmodulated digital-control and clock signals are necessary to prevent demodulation of 50-kHz frequency and to prevent analog noise.

Consider the product of two square signals with unity amplitude, , where is a square signal with frequency and is a square signal with frequency in radians (Figure 2a).

The Fourier transforms of square waves and are:

\[
x_1(t) = 4 \pi \left[ \frac{\sin(\omega_1)}{\pi} + \frac{\sin(3\omega_1)}{3\pi} + \frac{\sin(5\omega_1)}{5\pi} + \ldots \right]
\]

\[
x_2(t) = 4 \pi \left[ \frac{\sin(\omega_2)}{\pi} + \frac{\sin(3\omega_2)}{3\pi} + \frac{\sin(5\omega_2)}{5\pi} + \ldots \right]
\]

The Fourier transform of the product of and is:

\[
x(t) = x_1(t)x_2(t) = 4 \pi \left[ \frac{\sin(\omega_1)\sin(\omega_2)}{\pi} + \frac{\sin(3\omega_1)\sin(\omega_2)}{3\pi} + \frac{\sin(5\omega_1)\sin(\omega_2)}{5\pi} + \ldots \right]
\]

You can limit the series to the first term.

Simple logic gates implement a spread-spectrum technique that produces predictable clock behavior and introduces no unwanted modulation frequencies.
for simplification:

\[ x(t) = 4 \sum \frac{\sin(\omega_1) \sin(\omega_2)}{\pi} = \]

\[ 4 \left[ \frac{1}{2} \cos(\omega_1 - \omega_2) - \frac{1}{2} \cos(\omega_1 + \omega_2) \right]. \]

If \( \omega_1 \) is the frequency of the crystal oscillator and \( \omega_2 \) is the result of the frequency division of \( \omega_1 \) by 128, for example, then you can rewrite the previous equation as follows:

\[ x(t) = 4 \sum \frac{\sin(\omega_1) \sin(\omega_2)}{\pi} = \]

\[ 4 \left[ \frac{1}{2} \cos(\omega_1 - \omega_2/128) - \frac{1}{2} \cos(\omega_1 + \omega_2/128) \right]. \]

In other words, the frequency peak of \( x_1(t) \) multiplied by \( x_1(t)/128 \) splits into two frequency peaks separated by \( 2 \times \omega_1/128 \). Each peak has half of the energy of the original \( \omega_1 \) peak. Figure 2b shows a Matlab-generated spectrum of \( x_1(t) \) and \( x(t) \).

Multiplying the nth harmonic of the original \( x_1(t) \) signal by \( x_1(t) \) splits the nth harmonic into two major frequency components with frequencies \( n \times \omega_1 + \omega_2/128 \) and \( n \times \omega_1 - \omega_2/128 \). (These product terms are the most significant.) If \( x_1(t) \) is purely sinusoidal and the frequency analyzer has an unlimited narrow frequency bandwidth, the initial \( x_1(t) \) nth harmonic splits into two frequency spikes. Each of these spikes is approximately 6 dBm, or two times, lower than the initial frequency spike. In practice, you can obtain a 4-dBm reduction. In many cases, this reduction is a lifesaver because it helps the circuit pass an EMI test, particularly when bulky ferrites on each cable turn your portable electronic device into a boat anchor.

**Figure 1**'s circuit realizes this technique using a few simple logic gates. You can obtain \( x_1(t) \) from the \( \mu P \) timer or the counter by dividing \( x_1(t) \) by any number—in this example, 128. Flip-flop IC3 locks \( x_1(t) \) to the crystal oscillator’s phase. The XOR gate, IC2, is the key element. Algebraic multiplication of signals \( x_1(t) \) and \( x_2(t) \) in **Figure 2a** is equivalent to the XOR function of \( x_1(t) \) and \( x_2(t) \) when they are “logic” signals. IC3A and IC3B compensate for IC3’s propagation delay. The output of IC3 routes directly to the clock input of the \( \mu P \). The resulting signal \( x_1(t) \) experiences two phase shifts over one period of \( x_2(t) \) (**Figure 2a**). The first shift of 180° occurs during \( x_1(t) \)’s transition from logic 0 to logic 1; the second shift of −180° occurs during the transition from logic 1 to logic 0.

From the \( \mu P \)'s perspective, the clock signal loses one full period of \( x_1(t) \) over one full period of \( x_1(t) \). In this example, if for example, sampling with every period of \( x_1(t) \) introduces no noise into the ADC’s reading. The frequency content of the digital clock and other digital signals contains no low frequencies, such as 50 kHz, so the digital clock does not cause any noise in the analog sections. (DI #2391)

**Reference**


To Vote For This Design, Circle No. 413

---

**You’ve got mail**

by Gary Kath and Craig Bishop, Scotch Plains, NJ

Many e-mail programs provide a “beep” or a pop-up message box signaling the user that a new e-mail message has arrived. If you are too far from the computer to hear the audible signal or if the monitor is turned off, then you miss the new-mail audible and visual signals. The simple circuit in **Figure 1** (see pg 136) latches on an LED and an audio sounder when an appropriate new audible e-mail signal occurs. The method replaces the normal e-mail sound.wav file with a.wav file of any valid recorded dual-tone multifrequency (DTMF) sound. The circuit listens for the DTMF tone and latches on LED D1 and the piezoelectric buzzer, P1. R1 biases the microphone’s FET, and C1 couples the audio to the M-8870 DTMF-receiver, IC1 (Teltonic Corp, www.teltonic.com). IC1 integrates both bandsplit-filter and decoder functions into an 18-pin DIP.

Resistors R1 and R2 configure the on-chip differential amplifier for a gain of 47. The 3.579545-MHz crystal, X1, provides a precise clock generator for IC1’s digital-counting decoding circuitry. R3 and C2 provide an RC guard time to place accept and reject limits on tone duration. IC1’s STD output switches to logic high for the duration of any valid DTMF tone. The NAND gate, IC1A, inverts this logic signal and then directs it to a latch configured with NAND gates IC1B and IC1C. Pushbutton switch S1 resets the latch. IC1D buffers and inverts the latch’s output and drives a 2N2222 transistor, Q1, thereby turning on the LED and the piezo buzzer. The circuit latches for any valid DTMF tone. You can add additional circuitry to use the M-8870’s 4-bit binary outputs, Q2 to Q4, if you need to discriminate between DTMF tones. (DI #2399)

To Vote For This Design, Circle No. 414
In the reset and watchdog-timer circuit in Figure 1, IC₁ is a 74HC14 Schmitt-trigger inverter that, with R₁ and C₁, acts as an astable oscillator. The circuit provides an active-high reset for an 80C31 μC. The watchdog trigger (WDT) consists of watchdog-trigger pulses from a port line. At power-on, the voltage on C₁ is 0V, and reset = 1. As C₁ charges, reset goes low, and the μC generates watchdog-trigger signals. These ac-coupled pulses periodically turn on Q₁ and charge C₁ to VCC. This action prevents C₁ from discharging through R₁ when reset is low. If the watchdog-trigger pulses stop, Q₁ turns off and C₁ discharges through R₁, reset goes high, resetting the μC. Now, C₁ charges through R₁, and reset goes low after the reset period. D₁ prevents charge-pump action, and D₂ provides a fast discharge path for C₁ when the supply goes down. The Q₁-Q₃ combination acts as a low-voltage reset circuit. When VCC decreases to less than approximately 4.5V, Q₂ turns off and Q₃ turns on, discharging C₁; reset then goes high. The circuit works with voltages as low as 1.5V. During power-up and -down, hysteresis of the inverter provides a clean reset signal. (DI #2400).

To Vote For This Design, Circle No. 415

An 80C31 μC receives a clean reset signal from this circuit, which monitors power-up and brownout conditions on the power supply.

Tired of missing incoming e-mail? This circuit provides a permanent indication of incoming messages.

Circuit provides brownout control of 80C31

By N Kannan, Mediatronix, Pappanamcode, India

Figure 1

An 80C31 μC receives a clean reset signal from this circuit, which monitors power-up and brownout conditions on the power supply.
The circuit in Figure 1 produces timing signals with a sound like that of a mechanical metronome. IC1 is a 555 timer that oscillates at approximately 3200 Hz. The two 3-k\( \Omega \) resistors and the 0.047-\( \mu F \) capacitor set the frequency. IC2 divides the frequency of IC1’s output by 2. IC2 produces a square wave with an exact 50% duty cycle. The frequency of the output of IC2 determines the oscillator frequency.

For every rising edge from IC3, the IC4 one-shot produces one low-to-high-to-low output pulse. The 30-k\( \Omega \) resistor and the 0.15-\( \mu F \) capacitor determine the length of IC4’s output pulse. This pulse is approximately 2 msec long with the values shown. The output from IC4 allows three to four pulses from IC2 to pass through to the output of IC5. When IC5’s Pin 2 is high, the pulses from IC2 appear at IC5’s Pin 3; when Pin 2 is low, no pulses appear at Pin 3. From IC5’s Pin 3, the series of pulses routes to the volume control and then to the power audio amplifier. All the circuitry except IC6 uses a 5V supply. IC6 needs a 12V supply to drive the speaker loud enough to hear over the sound of an instrument. If you want a visual indication of the beat, you can connect an npn transistor, with a 470\( \Omega \) series base resistor, to IC4’s Pin 6. An LED in series with a 470\( \Omega \) resistor from the collector to 5V produces the visual indication. (DI #2404).

To Vote For This Design, Circle No. 416
PC controls light dimmer
Afshin Mellati, Burr-Brown Corp, Tucson, AZ

Using the simple circuit in Figure 1, you can control the light intensity in your room or work area from your PC. The heart of the circuit is a low-power D/A converter that converts digital words from a computer’s parallel port to analog-voltage signals. To isolate the dc low-voltage part of the circuit from the high-voltage part, the circuit uses an optoisolator, which prevents any direct electrical connection between the two sections. The optoisolator triggers triac T1, which behaves like a switch. In each power cycle, T1 switches on, the ac supply voltage connects to the load (lamps), and current starts flowing in the triac. At the end of a half-period, when the current drops to zero, T1 turns off and awaits another trigger in the opposite direction. This additional trigger occurs in the second half-period of the power cycle. A lower triggering voltage makes T1 conduct at an earlier point in and stay on for a larger fraction of the cycle. The larger fraction corresponds with transferring more power to the lamp, resulting in a higher intensity.

The output voltage of the D/A converter sets the triggering point. The DAC, after one stage of buffering, provides enough current to drive the optoisolator. IC3 generates a 2.5V reference; the crystal oscillator and capacitors C1 through C4 set the DAC’s timing characteristics. The DAC1220 (Burr-Brown Corp, www.burr-brown.com) connects to the parallel port with three wires for serial transfer of the digital codes. The Pascal program of Listing 1 (pg 142) reads the PC’s keyboard; when you press Q or W, the routine increments or decrements a digital code and sends it to the DAC. The DAC then controls the lamp’s intensity. Upon power-up, the DAC receives a digital code of zero, which corresponds to a 2.5V output (the reference voltage). You then adjust potentiometer R3, such that the lamp is half on. Using the keyboard, you can change the light intensity to the desired level. The dc part of the circuit consumes only approximately 5 mA.

Listing 1 is available for downloading from EDN’s Web site, www.ednmag.com. Click on “Search Databases” and then enter the Software Center to download the file for Design Idea 2401. (DI #2401).

To Vote For This Design, Circle No. 417

Set your computer area’s lighting intensity from the comfort of your swivel chair, using keyboard commands. A simple Pascal routine and some low-cost components do the trick.
LISTING 1—PC-CONTROLLED LIGHT DIMMER

Program proj1;
uses CRT, Strings;

procedure writedata(dword : longint);
var
wraddr, readdr, strbaddr : word;
pPrintPort : ^word;
rdta, ibte, bit, n : byte;
to20, to28, num, data : longint;

begin
  pPrintPort := Ptr($40, $08);
  wraddr := pPrintPort^;
  to20 := $100000;
  to28 := $10000000;
  num := to20 + $40 + dword;
  writeln(data = , num);
  PORT[wraddr] := $97;  (* Initialize to cs=0, sclk=0, sdin=1 *)
  for n := 1 to 32 do
    begin
      ibte := PORT[wraddr];
      PORT[wraddr] := ibte OR 64;  (* set scik=1 *)
      num := num * 2;
      bit := 0;
      if num > to28 then
        begin
          bit := 1;
          num := num - to28;
        end;
      ibte := PORT[wraddr];
      if bit = 0 then PORT[wraddr] := ibte AND 239 [set sdin=0] else PORT[wraddr] := ibte OR 16;  (* set sdin=1 *)
    end;
end;

end;

{******************************************************************************

 var
 inpkey : char;
 num, num1 : longint;
 label start;
 begin
   num := 0;
   start: inpkey := readkey;
   if inpkey = 'q' then
     begin
       if (num+$1000) > $7fff0 then num := $7fff0
       else num := num + $1000;
     end;
   if inpkey = 'w' then
     begin
       if (num-$1000) < $7fff0 then num := $7fff0
       else num := num-$1000;
     end;
   if inpkey = 'x' then exit;
   if num < 0 then num1 := $ffff + num
   else num1 := num;
   writedata(num1);
   goto start;
 end.

Circle 10 or visit www.ednmag.com/infoaccess.asp