

designideas

READERS SOLVE DESIGN PROBLEMS

Real-world power tests model FPGA's thermal characteristics

Jeremy Willden, Ad Hoc Electronics, Pleasant Grove, UT

Given ever-increasing clock frequencies and higher gate counts, many systems that include high-performance FPGAs (field-programmable gate arrays) routinely require a thoroughly analyzed thermal model. While working on a project that contained an FPGA, I realized that I had insufficient data to determine the FPGA's exact power dissipation, which my mechanical-engineering colleague required to construct a system model for thermal analysis using Flomerics' (www.flomerics.com) Flotherm software.

Although we had created fully functional hardware, we hadn't included a method of measuring the FPGA's exact power consumption, a problem further complicated by the presence of multiple power-supply voltages that fed additional circuits on the board. Although the manufacturer's FPGA-power-cal-

culuation spreadsheet allowed us to approximate the circuit's total wattage, the calculated values related only to its internal power consumption and didn't account for power not dissipated in the chip—that is, power delivered to I/O lines that drive other devices. To further confuse the issue, we lacked information about the FPGA package's thermal properties.

My mechanical-engineering colleague and I decided to create a controlled experiment by placing a functioning PCB (printed-circuit board) inside an improvised temperature chamber—a cardboard box. We would apply a precise amount of power only to the FPGA, measure its package's external temperature, and measure its internal die temperature using the FPGA's on-chip temperature-sensing diode. We would then model the experiment in Flotherm and adjust the package's thermal properties until the simulation's results matched the measurements.

Next, we would measure the FPGA's temperature while it executed an actual VHDL application within the temperature-controlled environment and work backward to determine the true power dissipation. Finally, we would create an accurate Flotherm model that would allow completion of a properly rated heat-sink design for the FPGA.

The only nonobvious part of this process involved how to dissipate a controlled amount of power within the FPGA. Acting on a flash of inspiration, I connected a nonfunctional PCB to a

DIs Inside

102 CPLD autonomously powers battery-powered system

106 Find hex-code values for microcontroller's ADC voltages

106 Cheap and easy inductance tester uses few components

108 Add a manual reset to a standard three-pin-reset supervisor

► What are your design problems and solutions? Publish them here and receive \$150! Send your Design Ideas to edndesignideas@reedbusiness.com.

power supply and reversed the polarity of the FPGA's core voltage. By doing so, I applied forward bias to the FPGA's internal parasitic diodes that connect between power and ground and the device's I/O-voltage rails and protection diodes (Figure 1). Under normal circumstances, these diodes remain reverse-biased and dissipate no power. Reversing the power-supply polarity forward-biased the diodes, dissipating power and heating the FPGA's die.

To obtain an exact voltage measurement, I added Kelvin-connected sense leads to the FPGA's power pins. I configured the power supply to operate in constant-current mode and adjusted its output to deliver exactly 2W of power as determined by multiplying the supply current and the voltage at the FPGA's power pins. My colleague configured the test probes' placement and performed the temperature measurements. Upon completion of our experiments, the temperatures that the Flotherm model predicted agreed with those we measured in our system's final configuration, including its heat sink, within a margin of 3 to 4°C. EDN

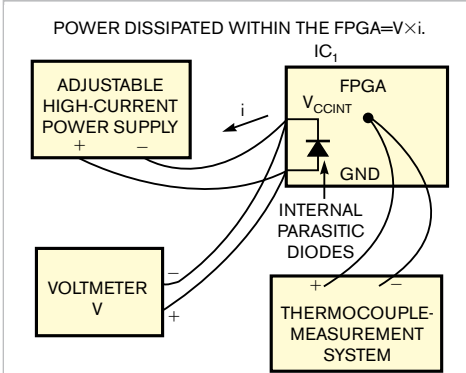


Figure 1 To measure an FPGA's thermal parameters, apply controlled forward bias to its internal parasitic diodes, thereby dissipating a known amount of power within its die.

CPLD autonomously powers battery-powered system

Rafael Camarota, Altera Corp, San Jose, CA

A common industrial and consumer application is a system that samples an environmental condition, such as GPS (global-positioning-system) location, voltage, temper-

ature, or light, at a wide interval, such as once every minute. This type of system is becoming increasingly wireless and battery-powered; it wakes up every minute, takes a sample, transmits data

to a central data-collection terminal, and then goes back to sleep. This Design Idea uses a small portion of an Altera (www.altera.com) EPM240-T100 CPLD (complex programmable-logic device) with a few discrete capacitors, resistors, diodes, and MOSFETs to autonomously wake a CPLD-based system from a full power-down state to an on state using an RC-timer circuit. This approach results in minimal

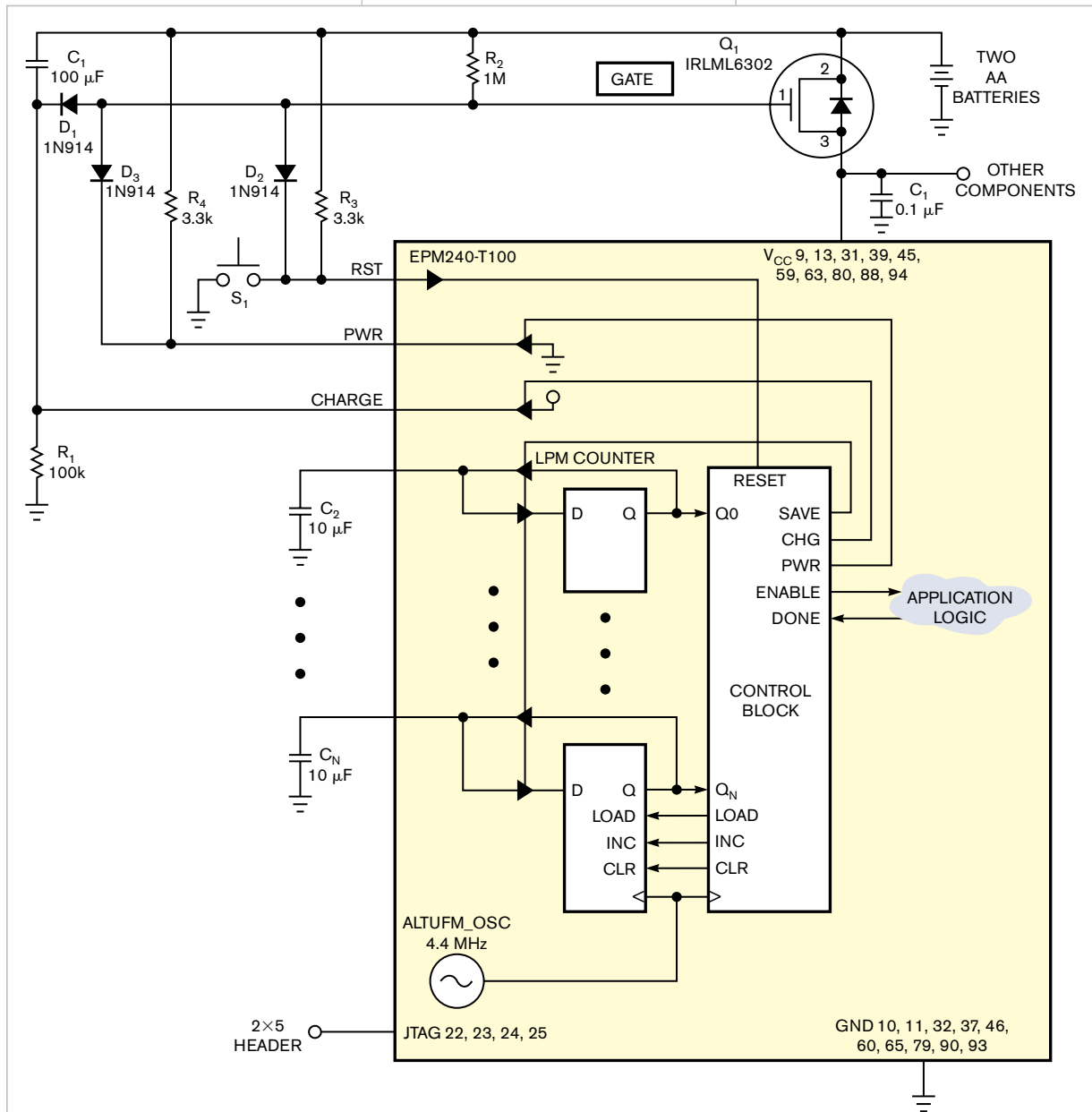


Figure 1 The CPLD comprises a control block, a 4.4-MHz internal oscillator, a 3-bit register, and six I/Os.

power consumption during samples when the power is on and between samples when the system, except for the RC circuit, is effectively off.

Figure 1 shows the basic CPLD on/off timer. Q_1 , an IRLML6302 P-channel MOSFET, is the power-control switch for the system. When the gate node is at V_{CC} , which R_2 pulls up, the power to the CPLD and the entire system is off, leaving only the RC circuit to use a minute amount of power. The CPLD comprises a control block, a 4.4-MHz internal oscillator, a 3-bit register, and six I/Os.

Figure 2 shows the state machine of the control block. The outputs in the state box are high, and all others are low. The dashed line from power-down to power-up represents the time delay, which the RC circuit comprising R_1 and C_1 measures when the system is off. Switch S_1 turns on and initializes the circuit. When S_1 closes, D_2 drives the gate node low, consequently turning on Q_1 when the gate voltage is 0.7V below V_{CC} . The EPM240-T100 is then operating in the power-up state less than 200 μ sec after Q_1 applies power. The power-up state drives the power node low, which holds the gate voltage at 0.7V, keeping Q_1 on after the switch is open. The power-up state also drives the charge node to V_{CC} . This action charges the negative terminal of C_1 to V_{CC} . Because reset=0, the control block goes to the reset state and Register 1 gets reset. Once S_1 opens, the control block goes to the enable state and drives the enable signal to one.

The sample-and-transmit circuit then begins operation and drives the done signal to zero. Once the sample and transmit are complete, the done signal becomes one, and the control block goes to the save state. The save state charges capacitors C_2 to C_N based on the value in Register 1. The

CAPACITORS C_2 , C_3 , AND C_4 ACT AS NON-VOLATILE MEMORY, STORING THE COUNT OF PREVIOUS POWER CYCLES.

save state is active for 100 μ sec, allowing the outputs to fully charge the 10- μ F capacitors. After 100 μ sec, the control block goes to the power-down state, which stops driving the charge and power nodes. R_4 pulls the power node high, leaving R_2 to pull up the gate node.

Once the gate node reaches $V_{CC} - V_{TQ1}$ at about 2.3V, Q_1 shuts off power to the system. All EPM240-T100 I/O is in a high-impedance state and does not affect the gate or charge nodes. The charge node starts at V_{CC} and begins to discharge through R_1 once power is off. Once the charge node drops to 2.3V, D_1 pulls down the gate node. Once the charge node reaches 1.6V, the gate node reaches 2.3V, and Q_1 turns on. The time for Q_1 to turn on is slightly

less than the τ of R_1 and C_1 . Off time equals $R_1 \times C_1 = 100,000 \times 0.0001 = 10$ sec.

The device powers up in the power-up state but moves quickly to the sample state. The sample state reads the value on capacitors C_2 , C_3 , and C_4 . These capacitors act as nonvolatile memory, storing the count of previous power cycles. If the Register 1 value sampled on C_2 through C_4 is less than 7, then the control block goes to increment, and the Register 1 value increments by one. Then, the control block again goes to the save state to charge C_2 through C_4 to a new binary value, 001. The device powers down again. On the eighth power cycle, or about 80 seconds after power-up, the control block moves to the enable state, thus enabling a new sample-and-transmit sequence. This process repeats every 80 seconds. You can change the period by adjusting C_1 and R_1 and by changing the Register 1 size and count between enable cycles. Based on an 80-second period comprising eight smaller power-up samples, test, and power-down cycles, the duty cycle for power is less than 3%; therefore, this approach increases battery life by as much as 33 times. **EDN**

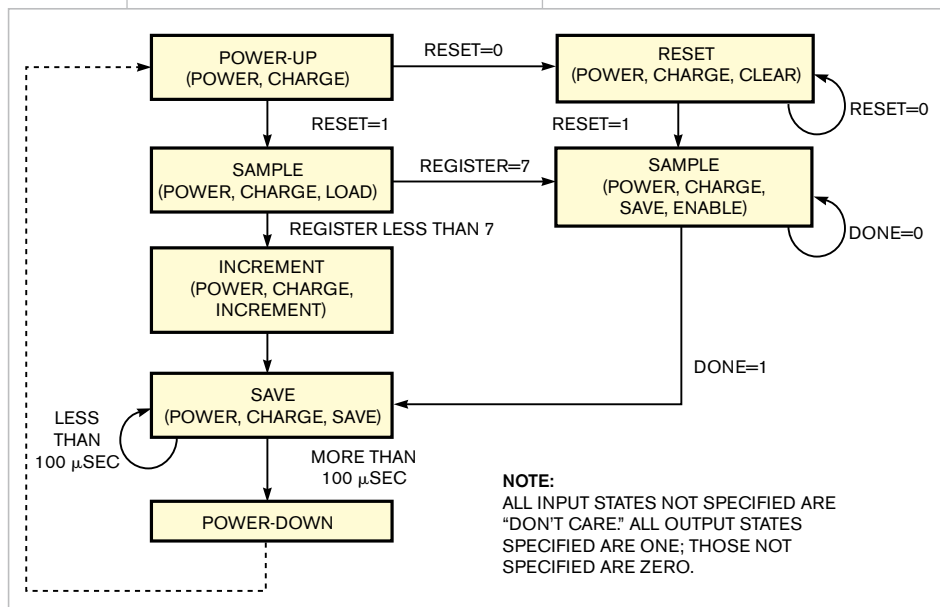



Figure 2 In the state machine of the control block, the outputs in the state box are high, and all others are low.

Find hex-code values for microcontroller's ADC voltages

Harry Gibbens Jr, Deafworks, Provo, UT

 This Design Idea is for low-end, eight-pin, flash-memory, 8-bit microcontrollers, such as the MC68HC908QT4A from Freescale (www.freescale.com), but it would apply to any 8-bit microcontrollers that use the ADC feature. In a nutshell, the ADC converts an input-analog-voltage level to a digital-signal format. The digital-signal format has an 8-bit hex-code value, such as \$00. The microcontroller “sees” the input-analog-voltage level from its ADC ports ranging from \$00 at V_{SS} to \$FF

at V_{DD} . Based on those hex-code values, there are a total of 256 ticks. The input voltages between V_{SS} and V_{DD} represent a straight-line linear conversion. In other words, the higher the input voltage, the higher the hex-code value.


The difficulty is that a programmer who needs to write assembly code for a programming algorithm must know what the hex-code value is for a different input-analog-voltage level—1.6V, for example. Referring to the microcontroller's specs and even contacting

its manufacturers do not yield satisfactory answers.

However, this Design Idea presents a solution to the problem. Given the microcontroller's power operating-voltage source, V_{DD} , use the following simple formula to obtain the hex-code value corresponding to an identified input-analog-voltage level: $V_{IN}/(V_{DD}/255)=\text{result value}=\text{hex code}$. Note that you must round off the result value to a whole number before converting to a hex-code value for better accuracy. The following sample calculation finds the hex-code value for a measured input-analog-voltage level of 1.6V when using a known microcontroller's V_{DD} of 5V: $1.6V/(5V/255)=81.6=82$, or \$52. **EDN**

Cheap and easy inductance tester uses few components

Al Dutcher, Consulting Engineer, Paulsboro, NJ

 In the absence of expensive test equipment, the circuit in **Figure 1** offers a simple and rapid alternative method of measuring inductance. Its applications include verifying that an inductor's value falls close to its design parameters and characterizing magnetic cores of unknown parameters that accumulate in the “junk box.” As designed, the circuit tests most inductors for use in power supplies and many inductors for RF circuits.

The circuit comprises two cascaded common-emitter-amplifier stages that form a nonsaturating, cross-coupled flip-flop. A common-emitter stage performs a phase inversion, and two cascaded stages form a noninverting feedback amplifier with gain that produces regeneration. Without the presence of the inductor that is undergoing test, L , regeneration occurs at dc, and the circuit behaves as a bistable flip-flop that assumes either of two stable states. Connecting the inductor reduces the dc positive feedback to below the regeneration level. Thus, regeneration

can occur only at ac, and the circuit becomes an astable oscillator.

Keeping the transistors out of saturation speeds the circuit's operation by minimizing the transistors' storage time. Although virtually any type of high-speed, small-signal RF transis-

tor provides adequate switching speed, lower frequency devices also work but decrease the low-inductance-measurement range. The circuit's frequency of oscillation is inversely proportional to the inductance that is undergoing test, and you can use either a frequency counter or an oscilloscope to measure the frequency of oscillation.

Figure 2 shows the waveform produced by an inductor with a value of approximately 100 μH . The frequency of oscillation depends on the L/R time

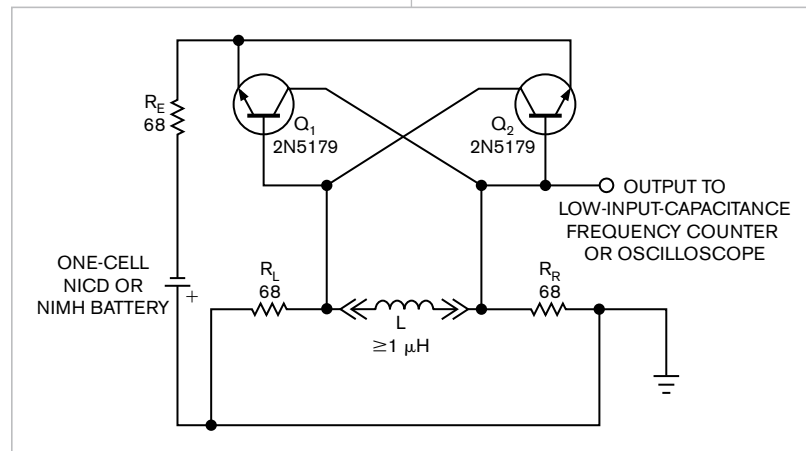


Figure 1 An inductance-test oscillator comprises two transistors and a few passive components. (Editor's note: For best results, minimize the lengths of all components' leads.)

constant comprising the inductance under test and resistors R_L and R_R . The time the waveform takes to change its state is directly proportional to the inductance, and, for one-half cycle, it approaches $T_{HALF}=L/100$. The period of a full oscillation cycle is twice that amount, or $T_{FULL}=L/50$. Solving for the inductance yields $L=50 \times T_{FULL}$. As an alternative, the frequency is inversely proportional to the inductance, or $f_{OSC}=50/L$. Using a frequency counter allows measurement of inductance as $L=50/f_{OSC}$.

The circuit's finite switching speed

of approximately 10 nsec imposes a lower floor of 1 μH on its measurement range. You can measure a small inductance by connecting it in series with a larger inductance, noting the reading, measuring the larger inductance alone, and subtracting the two measurements.

Although the circuit imposes no upper limit on inductance values, when the inductor's ESR (equivalent-series resistance) exceeds approximately 70 Ω , the circuit stops oscillating and reverts to bistable operation. The circuit measures values of all inductors

and transformer windings except for small, low-frequency iron-core devices that present a high ESR. For greatest accuracy, use a low-input-capacitance instrument to measure the frequency of oscillation.

A single NiCd (nickel-cadmium) or NiMH (nickel-metal-hydride) rechargeable cell provides power for the circuit. These cells present a relatively flat voltage-versus-time discharge characteristic that enhances the circuit's measurement accuracy. The circuit consumes approximately 6 mA during operation. **EDN**

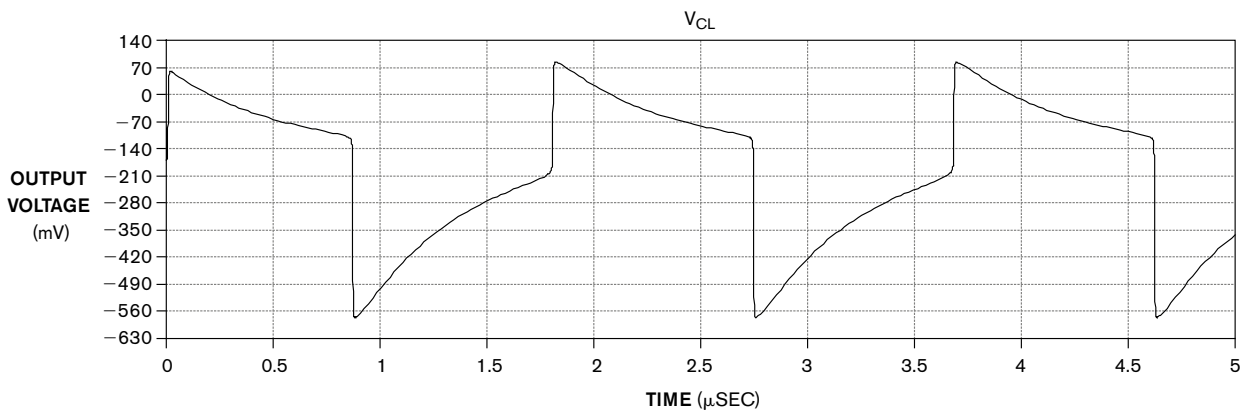


Figure 2 Testing an inductor with a value of approximately 100 μH produces this output waveform.

Add a manual reset to a standard three-pin-reset supervisor

Derek Vanditmars, Delta Controls, Surrey, BC, Canada

Adding a manual reset to a design usually involves designing in a new part with a manual-reset input. But, by adding a couple of low-value resistors, a standard three-pin-reset supervisor can work in most applications. The circuit in **Figure 1** ensures a clean $\overline{\text{RESET}}$ signal during and after you have pressed the manual-reset button. When you activate the manual-

reset button, the supply voltage drops below the reset supervisor's minimum reset threshold because of the R_1/R_2 voltage divider formed when S_1 is active. This action causes the reset supervisor to activate its $\overline{\text{RESET}}$ output. When you release S_1 , the supply voltage returns to above the reset-supervisor maximum-reset threshold, and $\overline{\text{RESET}}$ remains active for the time-

out period of the reset supervisor.

When you do not press S_1 , R_2 has a voltage drop arising from the reset supervisor's supply current and $\overline{\text{RESET}}$ output loading. For most reset supervisors, the maximum supply current is 50 μA . For most designs, the $\overline{\text{RESET}}$ output goes to one or more CMOS inputs that require about 10 μA each. With two CMOS devices connected to $\overline{\text{RESET}}$, the total current through R_2 would be $(2 \times 10 \mu\text{A}) + 50 \mu\text{A} = 70 \mu\text{A}$. The voltage drop across R_2 due to the current flow effectively adds $70 \mu\text{A} \times 100\Omega = 7 \text{ mV}$ to the reset su-

supervisor's reset-threshold voltage.

You should consider several trade-offs for the selection of values for R_1 , R_2 , and C_1 . The value of the local bypass capacitor, C_1 , for the reset supervisor should be low enough to allow the reset supervisor to detect transient supply-voltage drops. The time constant of R_2 and C_1 determines this factor; in this example, the time constant is $100\Omega \times 0.01\ \mu\text{F} = 1\ \mu\text{sec}$. This figure is typically much higher than the decay rate of a regulated power supply that has lost power.

When you activate S_1 , current flows through R_1 and R_2 . In the circuit in **Figure 1**, the current flow when you activate S_1 is $3.3\text{V}/(100\Omega + 100\Omega) = 16.5\ \text{mA}$. This amount of current would be OK for a line-powered system but may not be OK for a battery-powered system. You can reduce the current by increasing the value of R_1 and ensuring that the reset supervisor's supply voltage drops below the minimum reset threshold. You can also increase

the value of R_2 , along with that of R_1 , but doing so will cause increased voltage drop and slower response to transients. Note that the increased current

of the manual reset occurs only while the manual reset is active, and typical system current drops while $\overline{\text{RESET}}$ is active. **EDN**

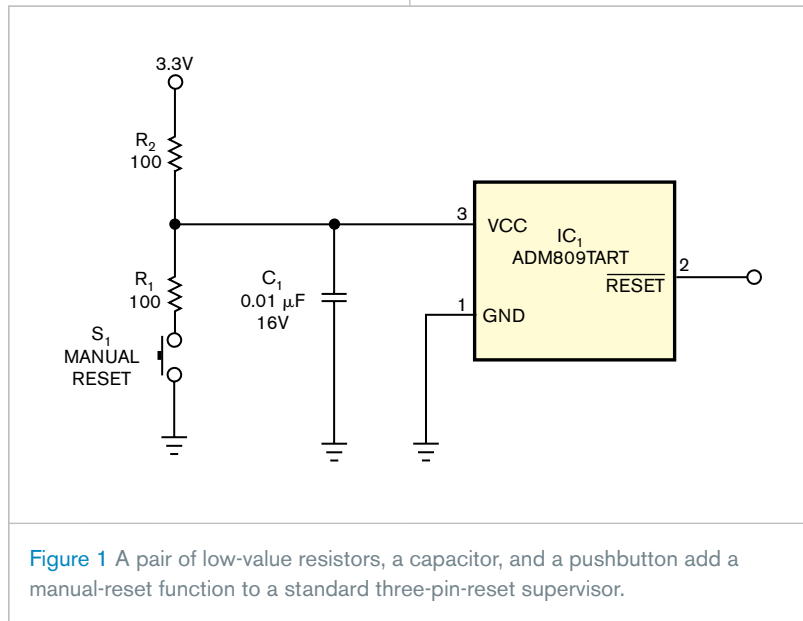


Figure 1 A pair of low-value resistors, a capacitor, and a pushbutton add a manual-reset function to a standard three-pin-reset supervisor.