


designideas

READERS SOLVE DESIGN PROBLEMS

Small capacitor supports telecom power supply during brownouts

Samuel Kerem, Rockville, MD

 This Design Idea shows how to keep telecom equipment operational during a short brownout. You must first understand a few details regarding the power supply for telecom equipment. The common voltage of the power source that feeds telecom equipment is -48V , although the actual voltage can range from -42.5 to -56V , -40 to -60V , or even beyond. The common power-“brick” dc/dc converter operates over the -36 to -75V range. A brownout occurs when the -48V source drops to 0V and stays there as long as 10 msec.

Using capacitive storage that connects to the brick’s input is an obvious approach to overcoming this problem, but a shortcoming becomes apparent when you understand the reality of the -48V supply. For example the energy in a capacitor charged to voltage is $(C \times V^2)/2$, where C is the capacitance

and V is the voltage. The brick stops its operation when the capacitor discharges to 36V . In general, the energy available to support the brick’s operation is, therefore:

$$U = C \times \frac{(V_1^2 - V_2^2)}{2},$$

where V_1 and V_2 are the beginning and final -36V voltages, respectively, and U is the energy. Also, $U = P \times t$, where P is power and t is time. Using these equations, you can find the time that the equipment will stay operational:

$$t = C \times \frac{(V_1^2 - V_2^2)}{2 \times P},$$

or, to define the capacitor’s value:

$$C = \frac{2 \times P \times t}{V_1^2 - V_2^2}.$$

Assume that the brownout occurs when the voltage at the brick’s input is

DI Inside

66 Tiny microcontroller hosts dual dc/dc-boost converters

68 Cross-coupled gates prevent push-pull-driver overlap

70 Save valuable picoseconds using ECL-wired OR

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

-39V , which is the case when -48V is -40V but the brick loses at least 1V because of protective ORed diodes in a hot-swap configuration. Also, assume that the storage capacitor charges to -39V . The equipment operates until this storage capacitor discharges to -36V . Assume that the equipment consumes 100W . To store enough energy for 5 msec, the capacitor’s value would have to be approximately $4500 \mu\text{F}$. The capacitor must be rated for the maximum possible incoming voltage, which can be more than 75V , so the minimum rating of 100V is a must. The $4500\text{-}\mu\text{F}$, 100V capacitor is a sizable part. If the design requires twice as much operational time at a power consumption of 300W , the capacitor must have a value of $27,000 \mu\text{F}$ and 100V .

This Design Idea still requires a capacitor, but the capacitor has a lower value—that is, $200 \mu\text{F}$ versus $4500 \mu\text{F}$ —and sustains 100W during a 5-msec brownout. This approach increases reliability and reduces cost and size. The hidden feature is the power brick’s ability to stay operational over the input range of -36 to -75V and even to operate under surges greater than

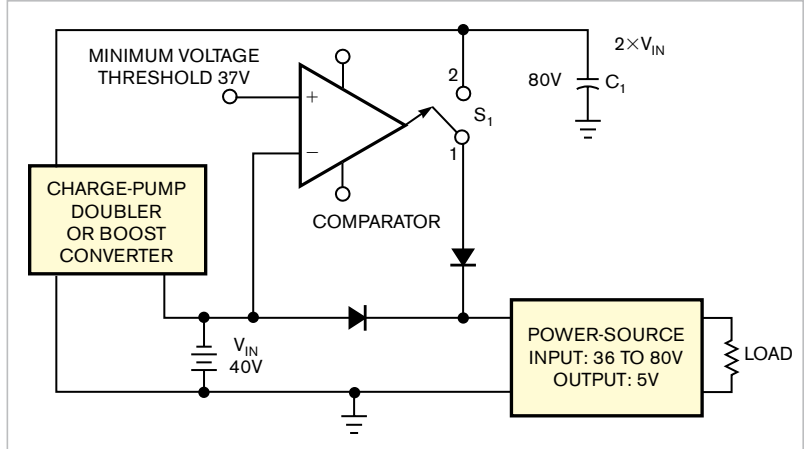


Figure 1 Charged to double the input voltage, the energy stored in capacitor C_1 dumps into the input of the “brick” power supply during brownouts when the input voltage drops to less than -37V .

–80V. **Figure 1** shows how you can use this feature. The **figure** depicts a positive input voltage. The brick is isolated, so polarity is irrelevant, but positive interpretation is easier to illustrate.


Remember that the stored energy

in the capacitor grows exponentially, whereas the capacitor's voltage increases linearly. The doubler charges C_1 to twice the input voltage or at least to 80V. Even if, hypothetically, you expect a 5-msec brownout as often as 10

sec, the current to charge 200 μF is still only approximately 3 mA. The comparator watches the input voltage, and, as soon as it drops below 37V, switch S_1 closes, and the energy from C_1 discharges to the power brick. **EDN**

Tiny microcontroller hosts dual dc/dc-boost converters

Dhananjay V Gadre, Netaji Subhas Institute of Technology, New Delhi, India

 Batteries are the typical power sources for portable-system applications, and it is not unusual these days to find microcontroller-based portable systems. A variety of microcontrollers operates at low power-supply voltages, such as 1.8V. Thus you can employ two AA or AAA cells to power the circuit. However, if the circuit requires higher voltage—for LED backlighting for an LCD, for example, which requires approximately 7.5V dc—you must employ a suitable dc/dc converter to boost the power-supply voltage from, for example, 3V to the required voltage. However, you can also employ a microcontroller to develop a suitable dc/dc-boost-voltage converter (**Reference 1**) with the

help of a few additional discrete components.

This Design Idea shows how to create not just one, but two dc/dc converters with just a tiny eight-pin microcontroller and a few discrete components. The design is scalable, and you can adapt it for a wide range of output-voltage requirements just by changing the control software for the microcontroller. You can even program the microcontroller to generate any required output-voltage start-up rate. **Figure 1** shows the basic topology of a boost switching regulator. The output voltage in such a regulator is more than the input voltage. The boost switching regulator operates in either CCM (continuous-conduction mode) or DCM

(discontinuous-conduction mode). It is easier to set up a circuit for DCM operation (**Reference 2**). The name comes from the fact that the inductor current falls to 0A for some time during each PWM period in DCM; in CCM, the inductor current is never 0A. The maximum current passes through the inductor at the end of high period of the PWM output (when the switch is on) and is:

$$I_{L_{MAX}} = \frac{V_{DC} \times D \times T}{L}, \quad (1)$$

where V_{DC} is the input voltage, D is the duty cycle, T is the total cycle time, and L is the inductance of the inductor. The current through the diode falls to zero in time T_R .

$$T_R = \frac{V_{DC} \times D \times T}{(V_{OUT} - V_{DC})}, \quad (2)$$

The load current is the average diode current,

$$I_{LOAD} = \frac{I_{L_{MAX}} \times T_R}{2 \times T}, \quad (3)$$

from **equations 1** and **2** and simplifies to:

$$I_{LOAD} = \frac{V_{DC}^2 \times D^2 \times T}{2 \times L \times (V_{OUT} - V_{DC})}. \quad (4)$$

The output voltage, V_{OUT} , is:

$$V_{OUT} = V_{DC} \times \left(1 + \frac{V_{DC} \times D^2 \times T}{2 \times L \times I_{LOAD}} \right), \quad (5)$$

The value of the output capacitor, which determines the ripple voltage, is:

$$\frac{dV}{dt} = \frac{I}{C}. \quad (6)$$

where dV/dt represents the drop in the output voltage during the period of the PWM signal, I is the load current, and C is the required output capacitor.

The total period of the PWM wave

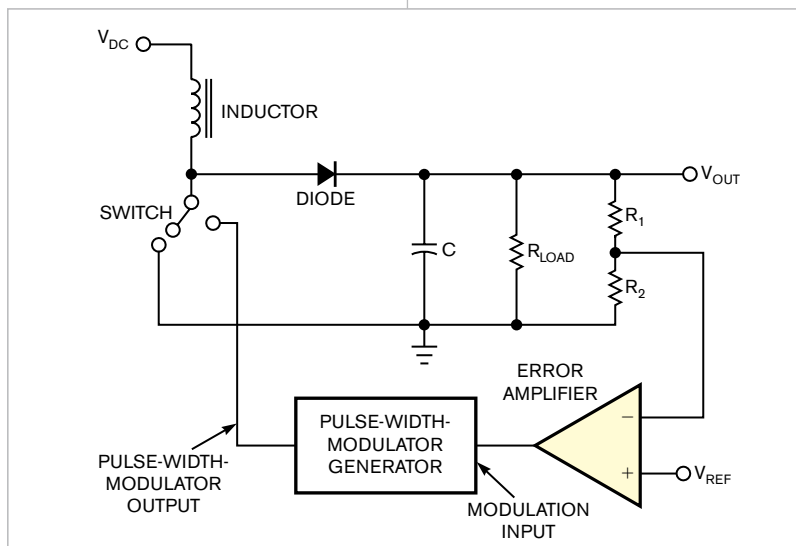


Figure 1 The output voltage in a boost switching regulator is more than the input voltage. The boost switching regulator operates in either CCM (continuous-conduction mode) or DCM (discontinuous-conduction mode)

is T and is a system constant. D is the duty cycle of the PWM wave, and T_R is the time during which the diode conducts. At the end of T_R , the diode current falls to 0A. The period of the wave is $T > D \times T + T_R$ for DCM. The difference of the PWM period, T , and $(D \times T + T_R)$ is the dead time.

The switch that operates the inductor is usually a BJT (bipolar-junction transistor) or a MOSFET. A MOSFET is preferable because of its ability to handle large current, better efficiency, and higher switching speed. However, at low voltages, a suitable MOSFET with low enough gate-to-source threshold voltage is hard to find and can be expensive. So, this design uses a BJT (Figure 2).

Microcontrollers offer PWM frequencies of 10 kHz to more than 200 kHz. A high PWM frequency is desirable because it leads to a lower inductor value, which translates to a small inductor. The Tiny13 AVR microcontroller from Atmel (www.atmel.com) has a “fast” PWM mode with a frequency of approximately 37.5 kHz and a resolution of 8 bits. A higher PWM resolution offers the ability to more closely track the desired output voltage. The maximum inductor current from Equation 1 is 0.81A for a 20- μ H inductor. The transistor that switches the inductor should have a maximum collector current greater than this value. A 2SD789 NPN transistor has a 1A collector-current limit, so it is suitable for this dc/dc converter. The maximum load current achievable with these values, from Equation 4, is 54 mA and thus meets the requirement of maximum required load current for an output voltage of 7.5V.

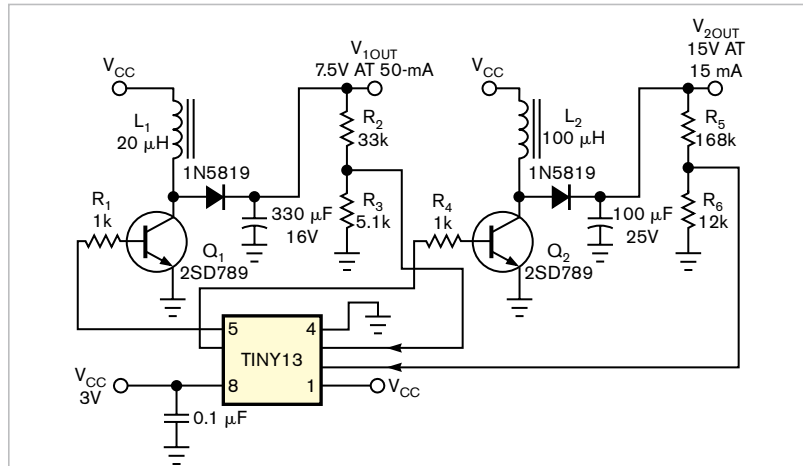


Figure 2 An Atmel Tiny13 AVR microcontroller regulates two boost-dc/dc-converter outputs using its internal ADCs and PWMs.

The Tiny13 microcontroller boasts two high-speed PWM channels and four 10-bit ADC channels. Another PWM channel and an ADC channel create the second dc/dc converter for an output voltage of 15V and a maximum load current of 15 mA. The inductor for this converter has a value of 100 μ H. To calculate the output-capacitor value, use Equation 6. For a 5-mV ripple, the value of the capacitor for 7.5V output voltage is 270 μ F, because the output current is 50 mA and the PWM-time period is 27 μ sec, so this circuit uses the nearest larger value of 330 μ F. Similarly, for the 15V output voltage, the required capacitor value is 81 μ F, so the design uses a 100- μ F capacitor.

The programs for the microcontroller are in C and use the open-source AVR GCC compiler (www.avrfreaks.net). They are available in the Web version of this Design Idea at www.edn.com/

080515di1. The AVR Tiny13 microcontroller operates at an internal clock frequency of 9.6 MHz without an internal-clock-frequency divider, so the PWM frequency is 9.6 MHz/256=37.5 kHz. The internal reference voltage is 1.1V. The main program alternately reads two channels of ADCs that monitor the output voltages in an interrupt subroutine. The main program executes an endless loop, monitoring the output voltage by reading the ADC values and adjusting the PWM values accordingly.EDN

REFERENCES

- 1 "Boost converter," Wikipedia, http://en.wikipedia.org/wiki/Boost_converter.
- 2 Pressman, Abraham I, *Switching Power Supply Design, Second Edition*, McGraw-Hill Professional, Nov 1, 1997, ISBN-10: 0070522367, ISBN-13: 978-0070522367.

Cross-coupled gates prevent push-pull-driver overlap

Richard Rice, Oconomowoc, WI

Overlap—the short period during which a push-pull drive's two transistors are both simultaneously on—is a common problem with

these drives in a center-tapped transformer's primary. Overlap causes a large current spike and increased switching losses. The fact that saturated transis-

tors turn off more slowly than they turn on causes the problem. One method of preventing overlap is to provide a time delay after turning off one transistor and before turning on the other one. This method requires several extra components and must include enough delay for a worst-case scenario. This Design Idea uses cross-coupled gates to prevent one transistor from turning

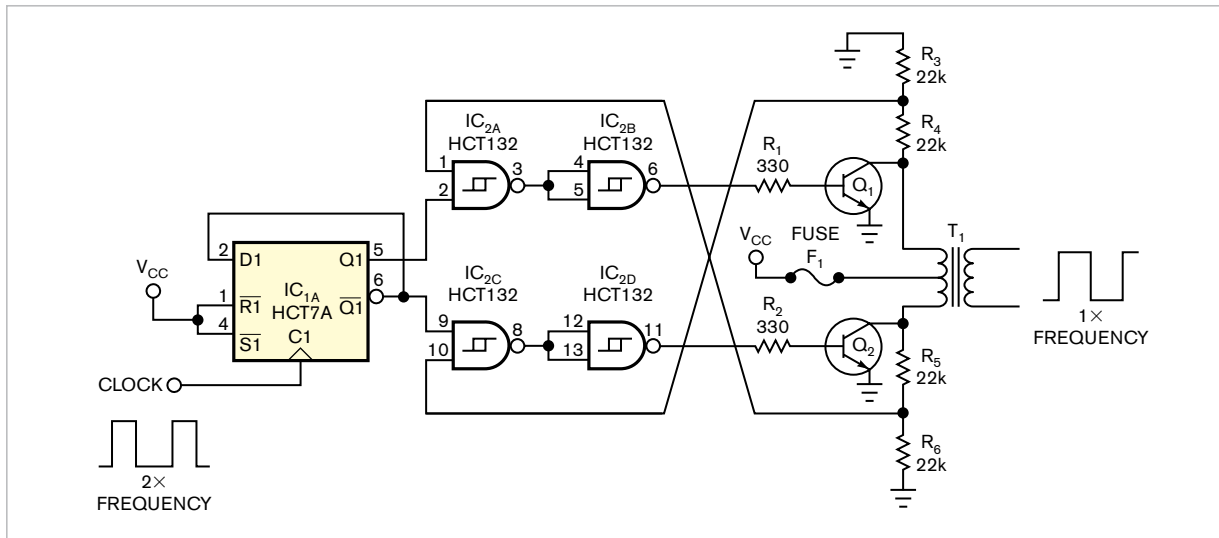


Figure 1 Cross-coupled gates prevent one transistor from turning on before the other turns off. Fuse F_1 protects against catastrophic failures.

on before the other turns off (**Figure 1**). For simplicity, the **figure** omits the depiction of bypass capacitors, snubber networks, and other components unnecessary for illustrating the method.

Gate IC_{2A} prevents Q_1 from turning on until Q_2 turns off. Likewise, gate IC_{2C} prevents Q_2 from turning on until Q_1 turns off. Gates IC_{2B} and IC_{2D} function as inverters to provide the correct

polarity to drive the switching transistors. Monitoring the transistors' collector voltages senses the turn-off of each transistor using the voltage dividers R_3/R_4 and R_5/R_6 . Because the collector voltage swings to twice the supply voltage, the voltage dividers halve the voltage. The impedance of the voltage dividers also limits the gates' input current to a safe level during overshoot.

The switching frequency is one-half the input-clock frequency. D-type flip-flop IC_{1A} divides the input-clock frequency by two and provides complementary outputs with a 50% duty cycle. The complementary outputs drive the switching transistors in an alternating sequence. The secondary of transformer T_1 provides an isolated square-wave output. **EDN**

Save valuable picoseconds using ECL-wired OR

Glen Chenier, TeeterTotterTreeStuff, Allen, TX

Often, when you are designing with high-speed ECL (emitter-coupled logic), you have too little time between clock cycles to implement logic functions using gates between flip-flops. In these cases, you can derive equivalent-logic functions using the wired-OR and flip-flop complementary inverted outputs (**references 1, 2, and 3**). You can parallel the emitter-follower outputs of ECL with a pulldown resistor to implement the OR function with almost no time-delay penalty. Complementary outputs—one inverted—provide delay-free logic inversions.

This Design Idea uses the older Mo-

torola (www.motorola.com) 10H ECL logic family, the fastest available when I was building the design (**Figure 1**). Newer ECL families are much faster, but the same wired-OR principle applies. For clarity, the **figure** omits power and 50Ω pulldown resistors. This design needed an XOR comparison between a PRBS (pseudorandom-binary-sequence) data stream and a local PRBS reference for a BER (bit-error-rate) counter running at 250 Mbps (**Figure 1a**). A problem occurred with the design, however: The clock period at 250 Mbps is 4 nsec, whereas the 10H107 XOR/XNOR gate's maximum

propagation delay is 1.7 nsec. In addition, the 10H131 flip-flop's maximum propagation delay is 1.8 nsec, and the required input-setup time is 0.7 nsec. All these delays total 4.2 nsec, which exceeds the 4-nsec clock period by 200 psec. Adding a fourth flip-flop with wired-OR outputs to replace the 10H107 XOR/XNOR solves the problem (**Figure 1d**).

The XNOR-equivalent function uses NOR, AND, and OR functions (**Figure 1b**). The circuit in **Figure 1c** separates the NOR into the equivalent OR with an output inverter and converts the AND into the equivalent OR with inverted inputs and output. Now, the circuit uses only ORs and inverters. This form is necessary for implementing the wired-OR equivalent (**Figure 1d**). In this case, the inverted-complementary outputs of the flip-flops replace the

inverters, and a parallel electrical connection between the flip-flops' outputs replaces the OR gates. **EDN**

REFERENCES

1 "Using Wire-OR Ties In ECLinPS

Designs," Application Note AN1650/D, On Semiconductor, www.onsemi.com/pub/Collateral/AN1650-D.PDF.
 2 "Dual D Type Master Slave Flip-Flop," MC10H131 Data Sheet, On Semiconductor, www.onsemi.com/pub/Collateral/MC10H131-D.PDF.

pub/Collateral/MC10H131-D.PDF.
 3 "Triple 2-Input Exclusive OR/ Exclusive NOR Gate," MC10H107 Data Sheet, On Semiconductor, www.onsemi.com/pub/Collateral/MC10H107-D.PDF.

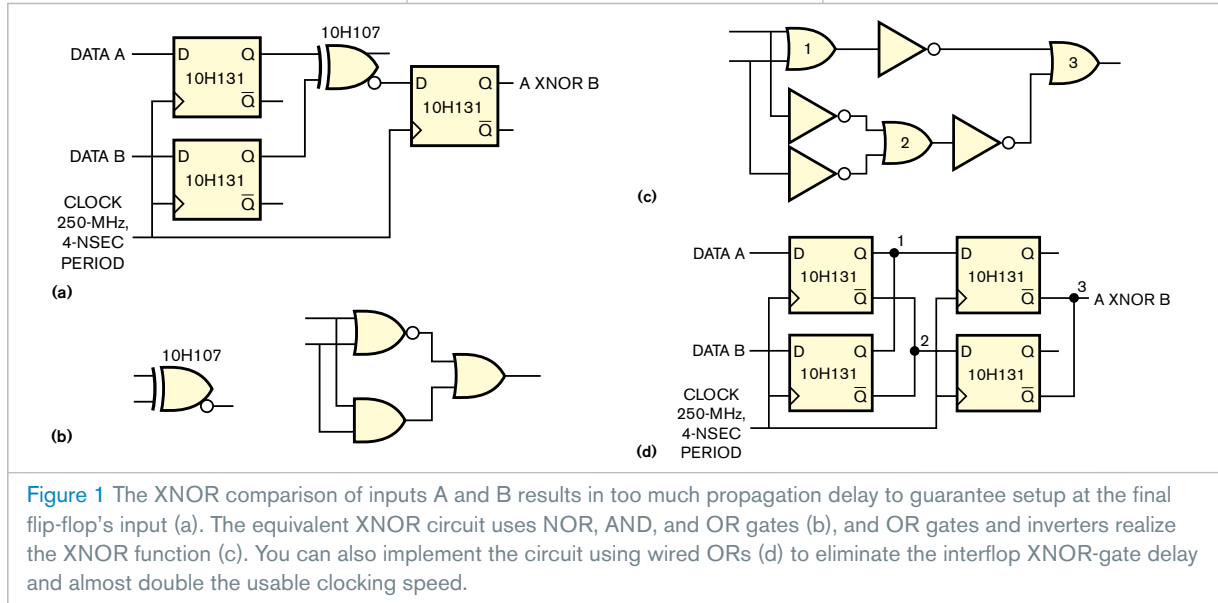


Figure 1 The XNOR comparison of inputs A and B results in too much propagation delay to guarantee setup at the final flip-flop's input (a). The equivalent XNOR circuit uses NOR, AND, and OR gates (b), and OR gates and inverters realize the XNOR function (c). You can also implement the circuit using wired ORs (d) to eliminate the interflip XNOR-gate delay and almost double the usable clocking speed.