


designideas

READERS SOLVE DESIGN PROBLEMS

Simple method uses PSpice for Thevenin-equivalent circuits

David Karpaty, Analog Devices Inc, Wilmington, MA

 Thevenin- and Norton-equivalent circuits, among the most fundamental circuit-analysis theorems, can be useful for determining a load resistance for maximum power transfer, simplifying circuit models, and a variety of other analysis techniques. Unfortunately, calculating the Thevenin voltage and resistance can become difficult as circuit complexity increases. **Figures 1, 2, and 3** illustrate a simple method for obtaining the Thevenin voltage and resistance—and, subsequently, the Norton equivalence—with the aid of simulation. First, you choose an arbitrary load resistance, R_{LOAD} — $2\text{ k}\Omega$ in this example—and run the simulation to get the current through the load resistance. Next, you remove the load resistance and simulate the open-circuit voltage across nodes A and B to obtain the Thevenin voltage. You obtain the Thevenin resistance from those two values.

The Thevenin-equivalent circuit

must produce the same current through the load. The total resistance in the Thevenin circuit is $R_{TOTAL} = (V_{TH} / I_{LOAD}) = (374.095\text{ mV} / 60.301\text{ }\mu\text{A}) \approx 6.203\text{ k}\Omega$, where R_{TOTAL} is the total resistance. Therefore, the Thevenin resistance is simply $[(V_{TH} / I_{LOAD}) - R_{LOAD}] = (R_{TOTAL} - R_{LOAD}) = 6.203\text{ k}\Omega - 2\text{ k}\Omega \approx 4.203\text{ k}\Omega$, where V_{TH} is the Thevenin voltage and I_{LOAD} is the load current.

Figure 4 shows the Thevenin-equivalent circuit, and **Figure 5** shows the Norton-equivalent circuit. Note that, because the net current through the load flows to the left, the positive Thevenin terminal is grounded.

$$\begin{bmatrix} +(6\text{k})I_1 & -(2\text{k})I_2 & -(0)I_3 & -(2\text{k})I_4 & -(0)I_5 & -(0)I_6 \\ -(2\text{k})I_1 & +(9.5\text{k})I_2 & -(2\text{k})I_3 & -(0)I_4 & -(2\text{k})I_5 & -(0)I_6 \\ -(0)I_1 & -(2\text{k})I_2 & +(15\text{k})I_3 & -(0)I_4 & -(0)I_5 & -(3)I_6 \\ -(2\text{k})I_1 & -(0)I_2 & -(0)I_3 & +(4\text{k})I_4 & -(2\text{k})I_5 & -(0)I_6 \\ -(0)I_1 & -(2\text{k})I_2 & -(0)I_3 & -(2\text{k})I_4 & +(14\text{k})I_5 & -(2\text{k})I_6 \\ -(0)I_1 & -(0)I_2 & -(3\text{k})I_3 & -(0)I_4 & -(2\text{k})I_5 & +(17\text{k})I_6 \end{bmatrix} \begin{bmatrix} 1\text{V} \\ 0\text{V} \\ 0\text{V} \\ 5\text{V} \\ -2\text{V} \\ 2\text{V} \end{bmatrix} \quad (1)$$

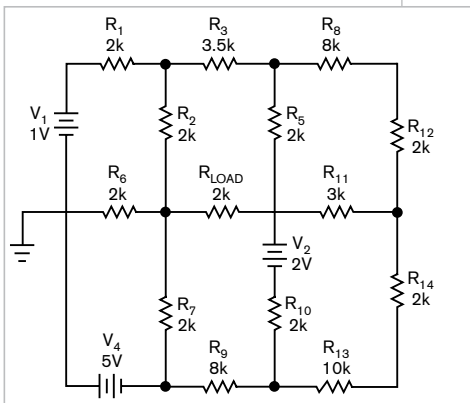


Figure 1 To calculate Thevenin-equivalent circuits, you first choose a load resistance— $2\text{ k}\Omega$ in this circuit.

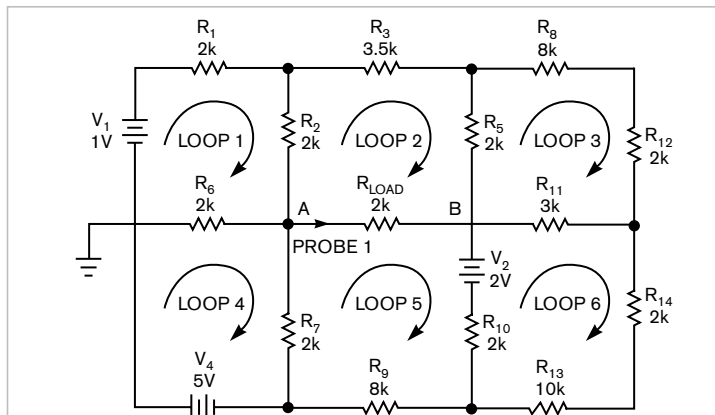


Figure 2 The simulation for current through the load resistance yields $-60.3\text{ }\mu\text{A}$.

DI's Inside

48 DAC and flip-flops form constant-current source

52 Convert negative inputs to positive outputs

▶ To see all of EDN's Design Ideas, visit www.edn.com/designideas.

Without the aid of simulation, you can calculate $V_{THEVENIN}$ and $R_{THEVENIN}$ as follows. The array for the loop currents in **Figure 2**, assuming a clockwise current flow in each loop, gives the current through the load resistance (**Equation 1**).

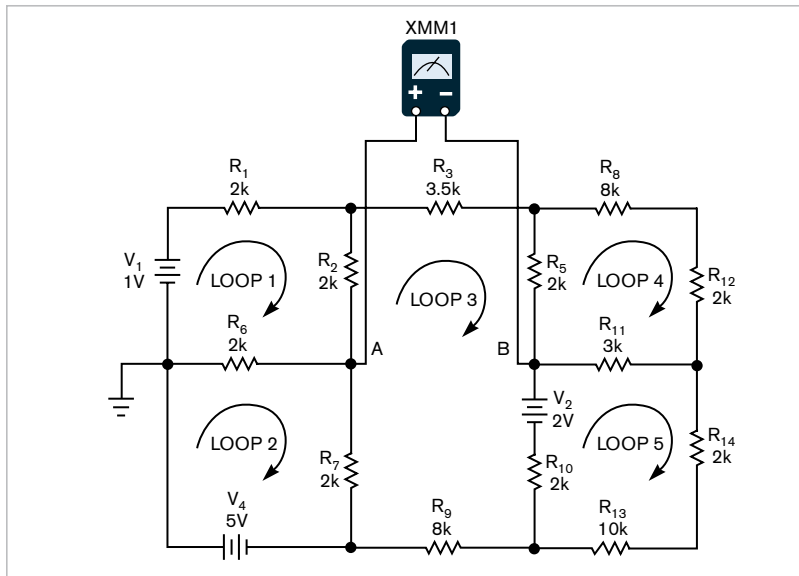


Figure 3 The simulation for the open-circuit voltage yields approximately -374 mV.

From Equation 1, you can calculate I_2 and I_5 : $I_2 \approx 217.77 \mu\text{A}$, and $I_5 \approx 157.47 \mu\text{A}$. Thus, $I_2 - I_5 \approx 60.3 \mu\text{A}$, assuming a leftward flow through the load resistor.

You calculate the array for the loop

$$\begin{bmatrix} +(6k)I_1 & -(2k)I_2 & -(2k)I_3 & -(0)I_4 & -(0)I_5 \\ -(2k)I_1 & +(4k)I_2 & -(2k)I_3 & -(0)I_4 & -(0)I_5 \\ -(2k)I_1 & -(2k)I_2 & +(19.5k)I_3 & -(2k)I_4 & -(2k)I_5 \\ -(0)I_1 & -(0)I_2 & -(2k)I_3 & +(15k)I_4 & -(3k)I_5 \\ -(0)I_1 & -(0)I_2 & -(2k)I_3 & -(3k)I_4 & +(14k)I_5 \end{bmatrix} \begin{bmatrix} 1\text{V} \\ 5\text{V} \\ -2\text{V} \\ 0\text{V} \\ 2\text{V} \end{bmatrix} \quad (2)$$

currents in Figure 3 without the load resistance, as Equation 2 shows. From Equation 2, you can calculate the following currents: $I_1 \approx 807.92 \mu\text{A}$, $I_2 \approx 1.744 \text{ mA}$, $I_3 \approx 179.87 \mu\text{A}$, $I_4 \approx 53.64 \mu\text{A}$, and $I_5 \approx 148.27 \mu\text{A}$.

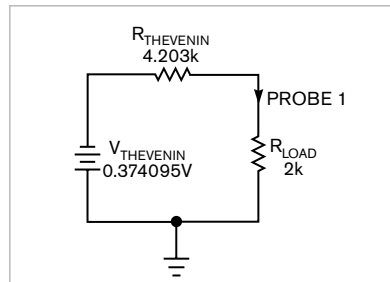


Figure 4 In the Thevenin-equivalent circuit, current flows to the left, so the V_{THEVENIN} terminal is grounded.

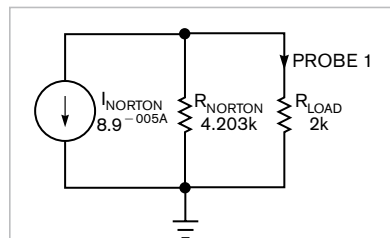


Figure 5 In the Norton-equivalent circuit, R_{NORTON} is 4.203 k Ω .

Thus, $V_A = -V_4 + [(I_2 - I_3) \times R_7] \approx -1.8719\text{V}$, where the net current flows downward. Further, $V_B = [(-V_4 + (I_3 \times R_9)) + ((I_3 - I_5) \times R_{10}) + V_2] \approx -1.498\text{V}$, where the net current in R_{10} flows downward. Thus, $V_{\text{THEVENIN}} = V_A - V_B \approx -374 \text{ mV}$, and you can calculate R_{THEVENIN} according to the previous description.**EDN**

DAC and flip-flops form constant-current source

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The Analog Devices (www.analog.com) AD5422 16-bit serial-input DAC lets you program for a voltage output or a current output. To communicate with the DAC and produce a variable output, you need a data SERDES (serializer/deserializer). If your design needs a constant 4-mA output, however, you can program the device with two flip-flops and test it with S_1 , a mechani-

cal pushbutton switch (Figure 1).

The AD5422's programming uses a 24-bit word in which the upper eight bits form an address for a control register and the lower 16 bits set the DAC's output range, slew-rate step, and slew-rate clock (Table 1, pg 52). Programming a 24-bit 0101 ... 01 pattern into the AD5422 sets it to the bottom of the simultaneously selected current range, 4 to 20 mA at the output-cur-

rent pin (Pin 19). The AD5422's internal shift-register data moves into the data register at every low-to-high transition of the latch signal (Pin 7). The device interprets this alternating bit sequence as a control command during the 23rd time you press and release the switch after IC_1 's power-up. After that sequence, the SCLK signal can remain idle (Figure 2).

Flip-flop FF_1 , configured as a familiar divide-by-two counter, produces the desired alternating sequence. Manually pressing and releasing the pushbutton switch, you cause the generation of an SCLK signal. You must use a debounc-

source, which is no more than a few 10s of microamperes, is harmless to the precision of the reference source.

By connecting a high-precision, 100Ω resistor between the I_{OUT} pin and ground and generating 23 clock pulses,

you can measure a voltage of 0.400xV on this resistor, where x≤4, confirming the high-precision, constantly flowing current of 4 mA. The actual full-scale-range error of IC₁ is far below its guaranteed worst-case value of ±0.3%

full-scale-range error (**Reference 1**). Hence, you must divide the observed relative error of the 4-mA current, with a value not exceeding 0.1%, by four because the current scale is 20 mA–4 mA=16 mA. The total full-scale-range error of the DAC in this case is thus less than 0.1%/4, or 0.025%. By using the constant-current source employing a monolithic DAC, you get high resolution, negligible sensi-

tivity to temperature, immunity to supply-voltage variations, and high initial accuracy. Current-output DACs also exhibit output resistance in the 10s of megohms.

This circuit uses S₁ to generate the SCLK signal for testing purposes only. For power-on-the-go applications, you can use a free-running clock with a frequency as high as 200 kHz. You can supply the pull-up resistor at the FAULT output and IC₂ from the AD5422's DV_{CC} pin.**EDN**

TABLE 1 EFFECTS OF THE SINGLE BITS OF THE CONTROL COMMAND

D ₂ D ₁ D ₀ =101	Selects 4- to 20-mA current range
D ₃ =0	Disables daisy-chain operation
D ₄ =1	Enables slew-rate control
D ₇ to D ₅ =101	Selects slew-rate size of 4 LSB
D ₁₁ to D ₈ =0101	Selects slew-rate update-clock frequency of 69.444 kHz
D ₁₂ =1	Enables outputs
D ₁₃ =0	Deactivates external-resistor pin
D ₁₄ =1	Increases output voltage by 10%
D ₁₅ =0	Concerns only the voltage output

REFERENCE

1 "Single Channel, 12/16-Bit, Serial Input, Current Source and Voltage Output DACs, AD5412/AD5422," Analog Devices, 2008, www.analog.com/static/imported_files/data_sheets/AD5412_AD5422.pdf.

Convert negative inputs to positive outputs

Shane Chang and Budge Ing, Maxim Integrated Products, Sunnyvale, CA

You can obtain a precise, positive-output voltage from a negative-voltage supply with a boost converter and a linear regulator. The input and output capabilities of the circuit in **Figure 1** depend on the allowable I/O voltages of IC₁ and IC₂. In this case, IC₁ and IC₂ convert a –5V input voltage to a 3.3V output voltage.

IC₁ is a boost converter that accepts

–5V when its V_{CC} pin connects to common ground—that is, the ground of the negative-power-supply input. Voltage divider R₁/R₂ at IC₁'s output provides feedback that sets the output voltage 10.5V above IC₁'s ground pin. With the feedback-threshold voltage factory-set to 1.226V, you can choose values for R₁ and R₂ using this **equation**: $(1.226V/R_2) \times (R_1 + R_2) = 10.5V$.

Current through R₁ and R₂ should be at least 2 μA. The IC₁ output, which is IC₂'s input, is 10.5V higher than –5V, which is 5.5V with respect to common ground.

IC₂, a linear regulator whose ground pin connects to the common ground, accepts input voltages as high as 6.5V. Its output is factory-set at 3.3V. **Figure 2** shows the output voltage versus the output current for the circuit in **Figure 1** with input voltages of –4.5, –5, and –5.5V.**EDN**

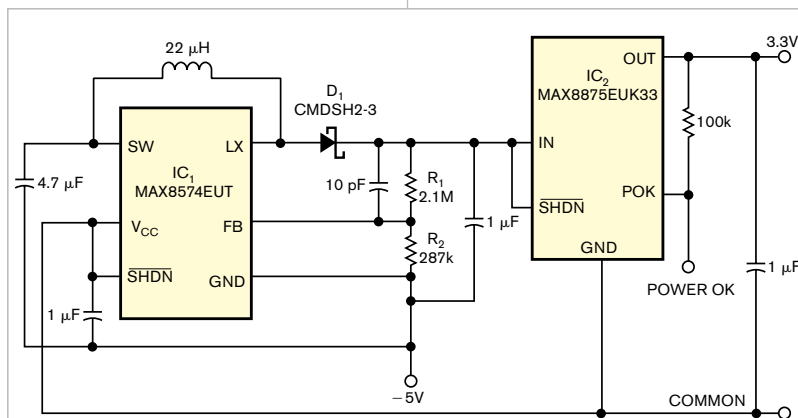


Figure 1 A two-IC circuit converts a –5V input to a 3.3V output.

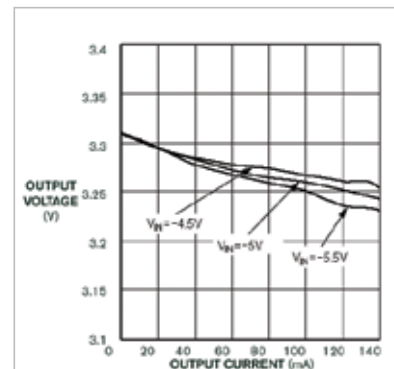


Figure 2 The circuit's output voltage drops as current increases. Plots indicate source voltages of –4.5, –5, and –5.5V.