EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Strategy processes video in RAM

Yu-Chieh Chen and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

Many video devices, such as the Analog Devices (www.analog.com) ADV7179 DAC, have analog-baseband-TV interfaces for PAL (phase-alternating-line) and NTSC (National Television System Committee) video signals. Unfortunately, these kinds of DACs accept video in interlaced-image format only, but you may need progressive-scan video instead. Furthermore, many of the progressive-scan images vary in size, which makes it more difficult to convert a progressive-scan image to an interlaced image. Therefore, you need a universal and efficient image buffer, such as SDRAM or DRAM, as a strategy for separating the image field.

Figure 1 shows the timing for a typical progressive-image data format. The upper four signals include the progressive-image source, including a frame-synchronization signal, a line-synchronization signal, a signal, and a pixel clock with pixel-image data. The lower two signals are the frame-synchronization signal, which contains many line-synchronization signals when the frame-synchronization signal is high, and the line-synchronization signal.

The pixel clock writes the progressive-image data into FIFO (first-in/first-out) memory. A higher-rate data clock can then write the data into RAM when each line-synchronization signal is low. This procedure ensures that the progressive-image data will correctly write into SDRAM regardless of how the pixel clock changes because of the various progressive-image data sizes. When the RAM write-enable signal or RAM read-enable signal is high, the system writes data into or reads data from SDRAM.

Figure 2 shows the frame-synchronization signal of progressive-image data and the frame-synchronization signal of interlaced-image data. The write-new-data and read-old-data enable signal executes at every line-synchronization signal of the progressive-image data when at a low level and at every frame-synchronization signal when at a high level. You can execute the read-old-data enable

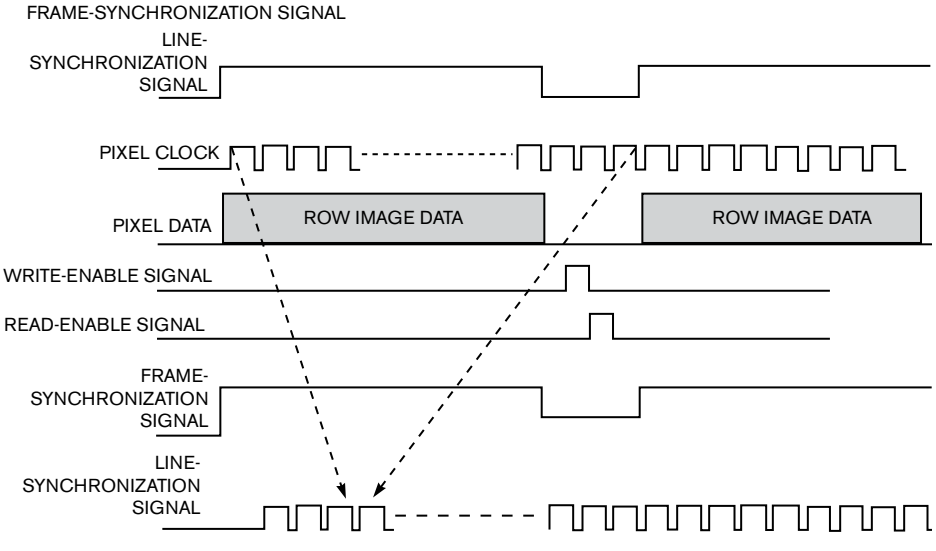


Figure 1 The pixel-clock signal puts image data into FIFO memory, which later synchronizes and goes into system RAM.

DIs Inside

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

signal only when the frame-synchronization signal is low, however. This scenario occurs when there are no valid image data in this period. **Figure 3** shows the data flow of the SDRAM-accessing procedure. A frame may, for example, contain 15 rows, in which you define the row data to count from 00 to 0e. Image data for odd rows are one, three, five, seven, nine, 11, 13, and 15, and image data for even rows are two, four, six, eight, 10, 12, and 14.

By using this SDRAM-accessing strategy, you can generate the interlace data and synchronize it with the frame-synchronization signal of the original progressive data. Thus, you need not worry about image size. Moreover, it can easily tune the interlaced-image data timing, changing the number of blank rows, without changing the write-into- or read-from-SDRAM sequences. You need to decide only which line-synchronization signal in low-level periods reads the old image data from the SDRAM. EDN

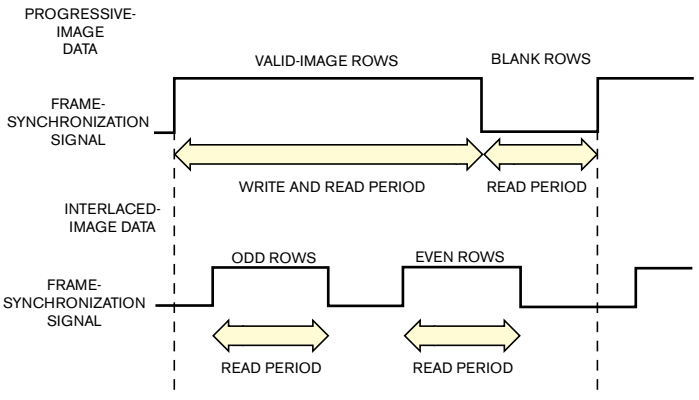


Figure 2 The frame-synchronization signal of progressive-image data alternates between reading odd and even rows of data in memory.

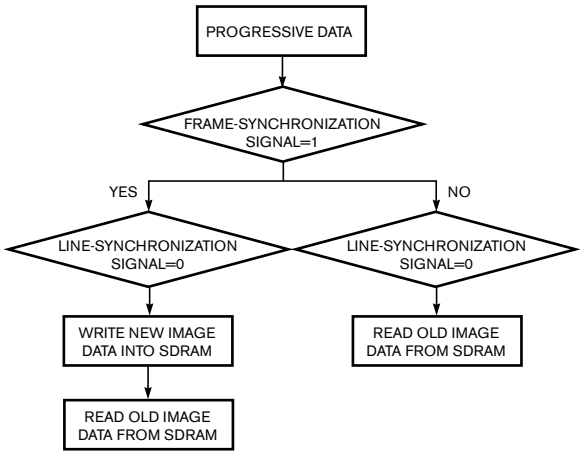


Figure 3 The state of the frame-synchronization signal determines whether the system performs a read or a write operation.

Rectangular-waveform generator produces 25 and 75% duty cycles

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

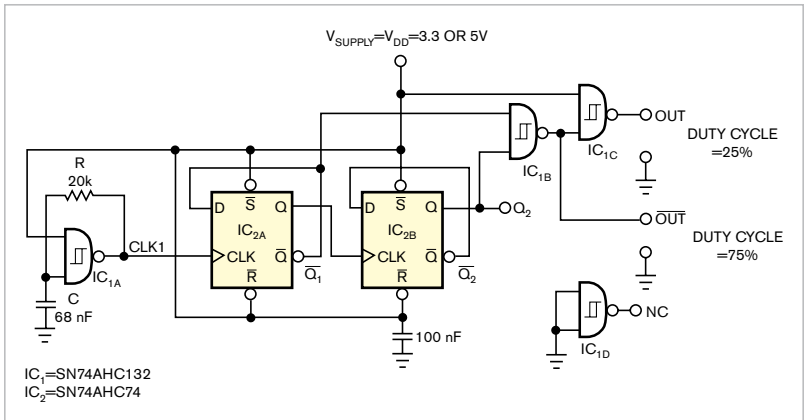


Figure 1 This circuit uses two flip-flops and three NAND gates to generate waveforms with 25 and 75% duty cycles.

Test applications may call for a rectangular waveform having a precision duty cycle higher or lower than 50%. The circuit in **Figure 1** is a free-running generator using just two ICs that produces rectangular-waveform duty cycles of both 25 and 75%. It holds the duty-cycle accuracy regardless of the duty-cycle accuracy from the signal source, an oscillator circuit comprising a Schmitt-trigger input NAND gate, IC_{1A}. Flip-flop IC_{2A} divides the oscillator's frequency by two at its $\overline{Q_1}$ and Q_1 outputs. Flip-flop IC_{2B} functions as a modulo-two divider clocked from the Q_1 output of IC_{2A}. Thus, IC_{2A} and IC_{2B} divide the oscillator's output by four.

NAND gates IC_{1B} and IC_{1C} generate the output waveform from the $\overline{Q_1}$ and Q_2 signals. **Figure 2** shows the output from NAND gate IC_{1B}. You can generate the 25% duty cycle by simply re-

placing the waveform that IC_{1B} outputs with the one that gate IC_{1C} outputs. If the active level is low instead of high, you can simply interchange the outputs of IC_{1B} and IC_{1C}.

The repetition frequency, $1/T_{REP}$, of the oscillator employing IC_{1A} is almost independent of the supply voltage within the range of 3 to 5V because both the positive and the negative thresholds of the input CMOS Schmitt trigger are roughly proportional to the supply voltage. Rough analysis gives a repetition frequency of approximately $2.7/\tau$, where $\tau=RC$, the time constant of the RC circuit around gate IC_{1A}. Further, the oscillator's waveform duty cycle is approximately 46.3%. EDN

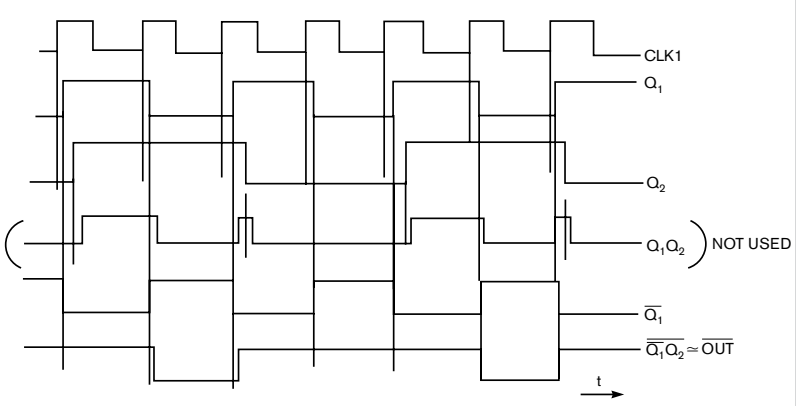


Figure 2 NANDing the $\overline{Q_1}$ and Q_2 logic signals gives a glitch-free output.

Battery simulator has variable ESR response

Barry Galvin, Grae LLC, Simi Valley, CA

You may lack experience and hardware when designing battery-operated products. The battery life of a product can depend more on the ESR (equivalent series resistance) than the terminal voltage. This situation is especially true when you use switching regulators to boost the battery voltage. The switching regulator creates a higher load as the battery voltage decreases. The ESR of a real battery is not constant. When you

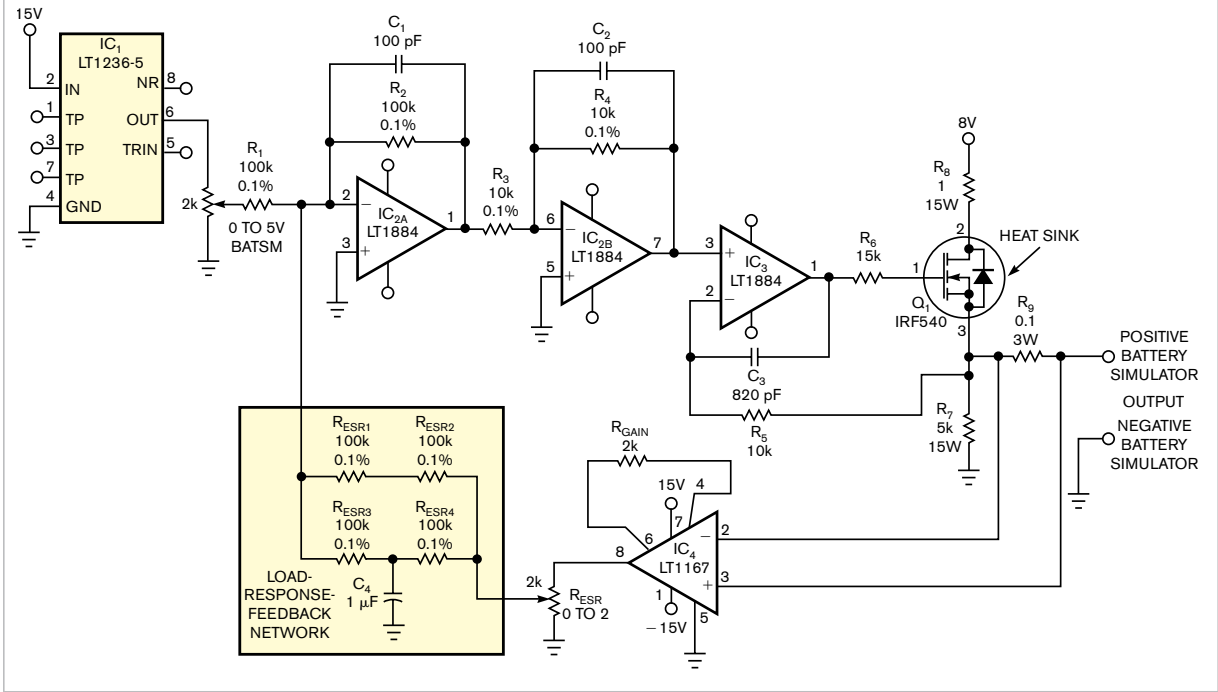


Figure 1 This simulator circuit represents the load response of many battery types.

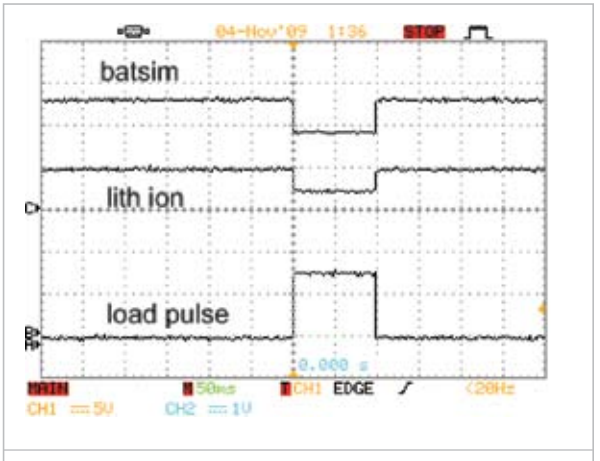


Figure 2 With no feedback capacitor, the simulator closely matches the response of a large lithium-ion battery.

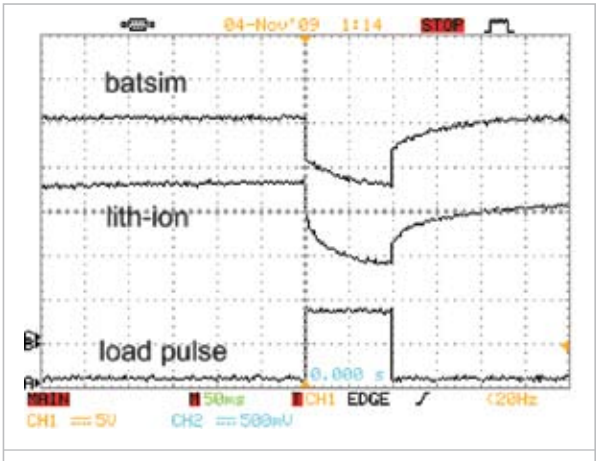


Figure 3 Adding capacitive feedback causes the simulator to act like a much smaller battery.

remove a battery load, it reacts and "heals" as its ions rediffuse. Portable electronics may include a low-power or a sleep mode. The device takes short high-power pulses from the battery.

The battery simulator in this Design Idea duplicates a battery's ESR-response curve. If you place different values in the feedback network, you can obtain various ESR curves. The circuit simulates most battery types, including lithium ion and alkaline. It supplies 0.5 to 4.2V at several amperes to the device under test, and it can simulate the ESR of a variety of battery types. You can change the delay to the final value of ESR by setting the ESR potentiometer. Some battery types exhibit this unique characteristic. It has a large influence on the delivery of pulsed current to a load.

In the circuit, IC_1 supplies a stable voltage, setting the unloaded output voltage (**Figure 1**). IC_2 provides the necessary inversions for the ESR function. IC_3 and Q_1 form a power-output stage that receives a voltage of 8V. Resistor R_8 limits the power. IC_4 senses the output current through R_9 and provides a gain of 20. This signal goes to the ESR timing circuit, providing both the ESR effect and the response timing.

You can simulate battery chemistries and sizes by varying the component values. If you omit C_4 and replace R_{ESR1} through R_{ESR4} with one 100-kΩ resistor, only the basic ESR function results. **Figure 1** omits power and bypass capacitors.

Applying a 1A load pulse without the capacitor in the feedback network causes the simulator response to closely follow the response of a 2000-mAhr lithium-ion 18650 battery (**Figure 2**). You can also add the capacitor to the feedback network to make the simulator better represent the response of a small, 200-mAhr lithium-ion battery (**Figure 3**). With proper adjustment of the circuit, you can produce many response curves. You can download National Instruments' (www.ni.com) LabView software and the voltage-ESR curves of selected battery types from Grae LLC (www.graellc.net). EDN

Create LED-lighting patterns without a controller

Jeff Tregre, www.BuildingUltimateModels.com, Dallas, TX

This Design Idea describes a simple LED-lighting-effects circuit comprising only five chips and costing only a few dollars. When you first observe the circuit in action, you will think that it uses a PIC (peripheral-interface-controller) chip requiring you to program hundreds of lines of code to generate the lighting effects. You can view the lighting effects in a video with the Web version of this Design Idea at www.edn.com/100318dia.

The circuit comprises seven functional blocks (**Figure 1**). IC_1 is an LM556, which has two 555 timers in one package. The first timer produces the main clock frequency of approximately 0.105 Hz. It toggles high to low approximately every 10 seconds. The polarity of the clock's signal changes the frequency of the VCO (voltage-controlled oscillator) that makes up the other half of IC_1 from low to high. Resistor R_2 and capacitor C_2 set the clock frequency. Changing either component changes the frequency.

The output from the first 555 timer feeds the control voltage input on the second 555 timer, letting it function as a VCO whose output frequency ranges from approximately 10 Hz when the first 555 timer output is high to approximately 33 Hz when the output is low. Components R_4 and C_3 set the VCO's frequency, and R_6 and C_4 control the smooth transi-

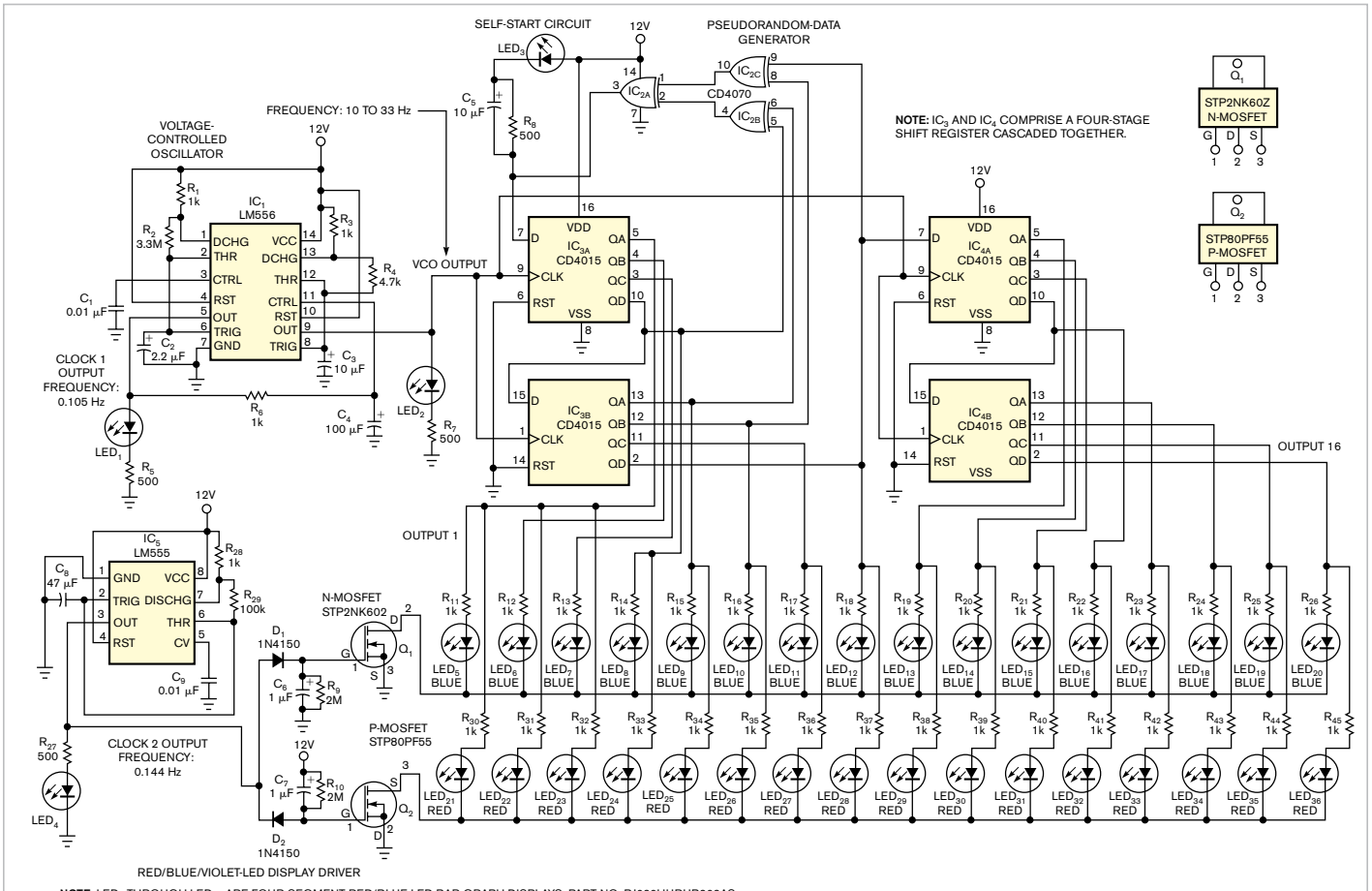


Figure 1 Two 555 timers create the clock pulses that drive blue and red LEDs.

tion of the VCO from 10 to 33 Hz.

LED_3, C_5, and R_8 act as a self-start circuit. Without it, you would need to add a pushbutton switch to toggle the data input of IC_{3A} from low to high during start-up. IC_2, a CD4070 quad exclusive-OR gate, acts as a pseudorandom-data generator. This circuit gives the illusion that bits of data span the bar graph.

IC_3 and IC_4 are CD4015 four-stage shift registers cascaded together. The data bits span the bar-graph displays in sequence from Output 1 to Output 16. IC_5, an LM555 timer, produces Clock 2's frequency of approximately 0.144 Hz. The inverse of this frequency toggles high to low approximately every 7 seconds, feeding the gates of N-MOSFET Q_1 and P-MOSFET Q_2, which act as the red/blue/violet LED-display driver. Clock 2 toggles high, enabling Q_1 and giving the blue LEDs a source to ground.

When Clock 2 toggles low, Q_2 turns on, giving the red LEDs a path to ground. C_6 and C_7, together with R_9 and R_{10}, respectively, act as a slow discharge circuit on the gates of the MOSFETs, keeping them on for approximately 2 seconds longer than Clock 2's pulse. The delay lets both the blue and the red LEDs be on at the same time for approximately 2 seconds and produces the color violet. This circuit uses N- and P-channel MOSFETs from ST-Microelectronics (www.st.com), but any general-purpose MOSFET should work. Just make sure that each one can handle at least 0.5A.

Figure 2 The bar graphs have both red and blue LEDs; turning on both yields violet.

The four-segment red/blue-LED bar-graph displays are unique. Each bar-graph display comprises one red and one blue LED in the same bar (**Figure 2**). Each LED has its own anode and cathode connections, thereby keeping this circuit simple without the need to add extra transistor drivers for each LED. You'd have to add them if their anodes, cathodes, or both were connected. This circuit requires four bar-graph displays. If you install any of the LED bar graphs backward, you will see the second color displayed, so that, if you were expecting red, you would get blue, and vice versa. EDN

Control stepper motors in both directions

Vladimir Rentyuk, Modul-98 Ltd, Zaporozhye, Ukraine

Stepper motors need bidirectional control in automatic machines or robotic applications. The circuit in **Figure 1** lets you control bipolar stepper motors and run them in both rotations. You can use the circuit in automatic devices and as an evaluation circuit for testing stepper motors. The circuit comprises clock oscillators IC_{3A} and IC_{3B}; a bidirectional, two-phase translator using an SN74HC74D dual flip-flop, IC_2, with a directional selector, IC_{3C} and IC_{3D}; and a push-pull L293DD channel-driver, IC_1. The circuit needs one power source, which depends on the stepper-motor specification. You can use a step-down voltage regulator to provide 5V dc. In many applications, an L7805A voltage regulator is suitable. Switch S_2 turns the motor on, and switch S_1 controls the motor's direction. Both signals can come from a sensor or a circuit with an open-collector output.

A circuit surrounding transistor Q_1 starts the motor. A forced starting is necessary because generators that employ two CMOS or TTL inverters are sometimes unstable after powering and can oscillate at a frequency of approximately 18 MHz. Thus, you need a delay after applying power to

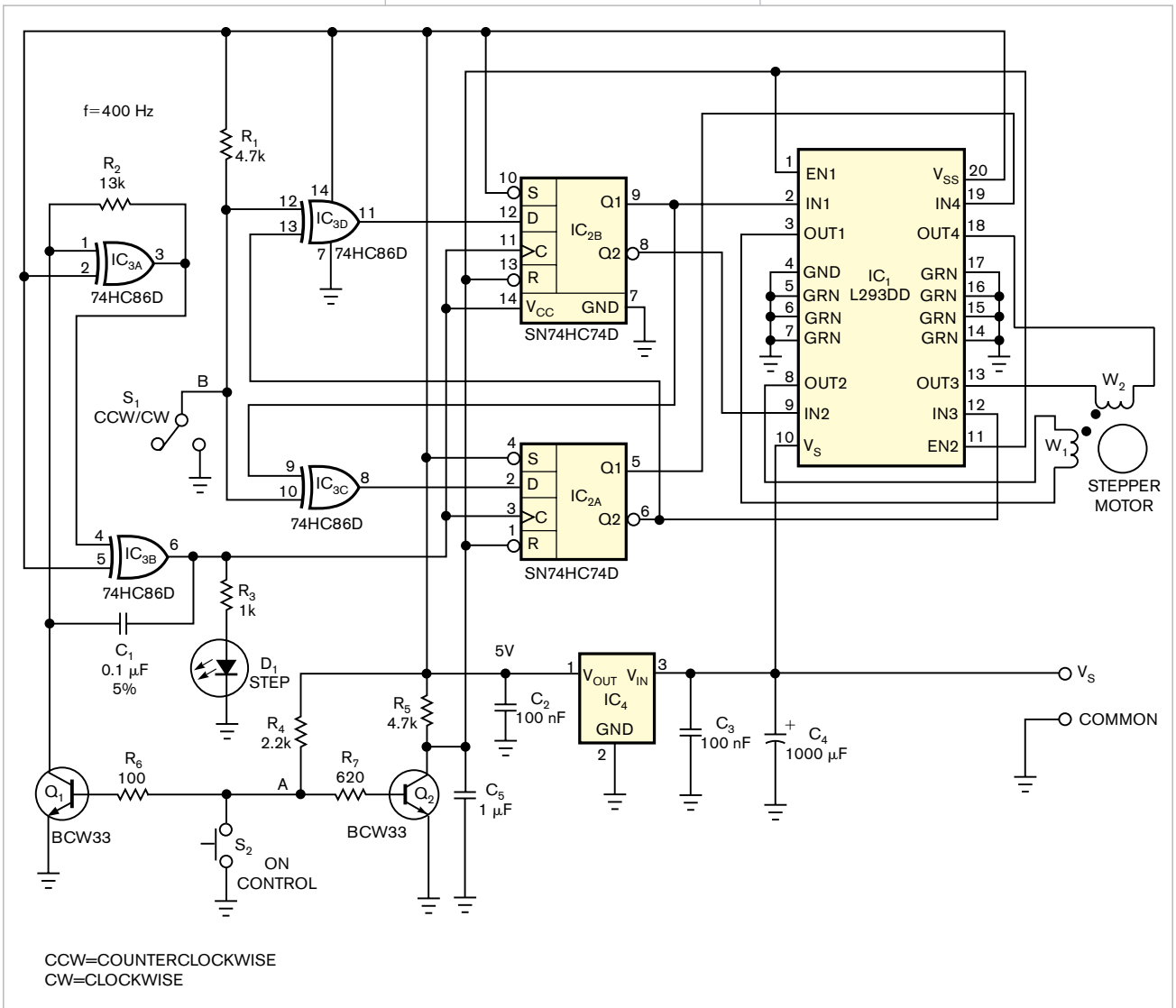
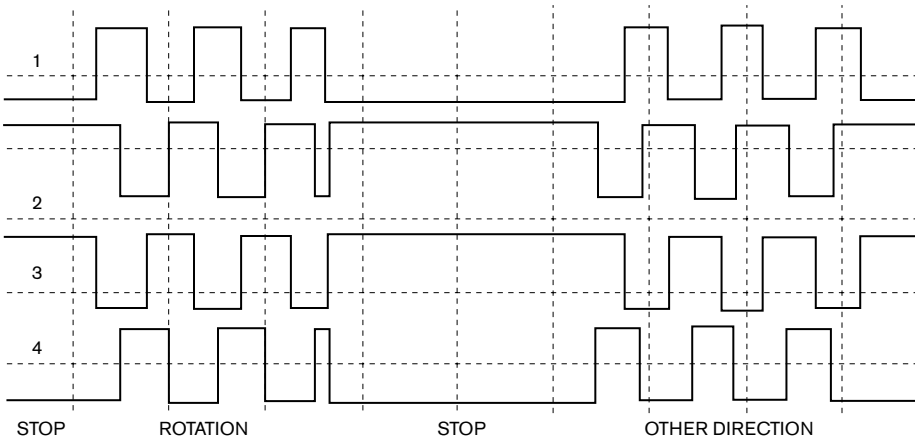


Figure 1 This circuit provides the necessary signals to drive a stepper motor. Switches turn the motor on and off and change its direction.

the circuit before sending the "on" command. The delay must be at least 100 µsec, but a delay of a few milliseconds is best. Capacitor C_5 eliminates the negative influence of bounce from S_2's contacts. The rotation of a rotor of the stepper motor begins when S_2 presents a low level to Point A. C_5 is unnecessary if a low-level signal from a circuit with an open collector comes to Point A—but not mechanical switches or buttons. Switch S_1 can be any suitable signal, such as that from a safety stop switch with a timer, trigger, or any open-collector output that connects to Point B. LED D_1 is a step indicator in "on" mode.



NOTES:
TRACE 1: OUTPUT Q1 OF IC_{2B} (IN1 OF IC_1).
TRACE 2: OUTPUT Q2 OF IC_{2A} (IN3 OF IC_1).
TRACE 3: OUTPUT Q2 OF IC_{2B} (IN2 OF IC_1).
TRACE 4: OUTPUT Q1 OF IC_{2A} (IN4 OF IC_1).

Figure 2 The motor stops when the outputs of IC_{2A}'s Q1 and IC_{2B}'s Q1 stay low. EN1 and EN2 are low, too.

The speed of rotation of a stepper motor depends on its specification from a step angle of the stepper motor and the frequency of the clock oscillator. **Figure 2** shows a timing diagram of the reversal mode of the circuit. EDN