Inexpensive logic controls stepper motor

DAVID ELLIS, ELLIS LINDAUER, PULLMAN, WA

A number of sophisticated ICs for stepper-motor control are now available. However, the advanced features of these chips—self-clocking, high-current drive, and full-step, half-step, and direction control—are often unnecessary or remain unused. For a design that needs to control only the number of steps, drive speed, and direction, you can make a very simple and inexpensive driver using two low-level logic chips (**Figure 1**). The cost of this controller is less than $1; the cost of dedicated motor-control ICs starts at around $5. The drawback is a slight increase in board space.

Going back to the basics, you can control a standard stepper-motor drive, whether bipolar or unipolar, using a four-step sequence (**Table 1a**). By replacing the on and off states with ones and zeros, respectively (**Table 1b**), Column B becomes the logical inverse of Column A, and Column D becomes the logical inverse of Column C. Thus, the corresponding state diagram (**Figure 2**) comprises just 2 bits. Clockwise rotation results from using a logical one to move sequentially from state one to state four and back to state one. Likewise, counterclockwise rotation results from using a logical zero to move through the states in the reverse order.

TABLE 1—STEPPER-MOTOR-DRIVE SEQUENCE

(a) (↑ CCW)

State	A	B	C	D
1	ON	OFF	ON	OF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

(b) (↓ CW)

State	A	B	C	D
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

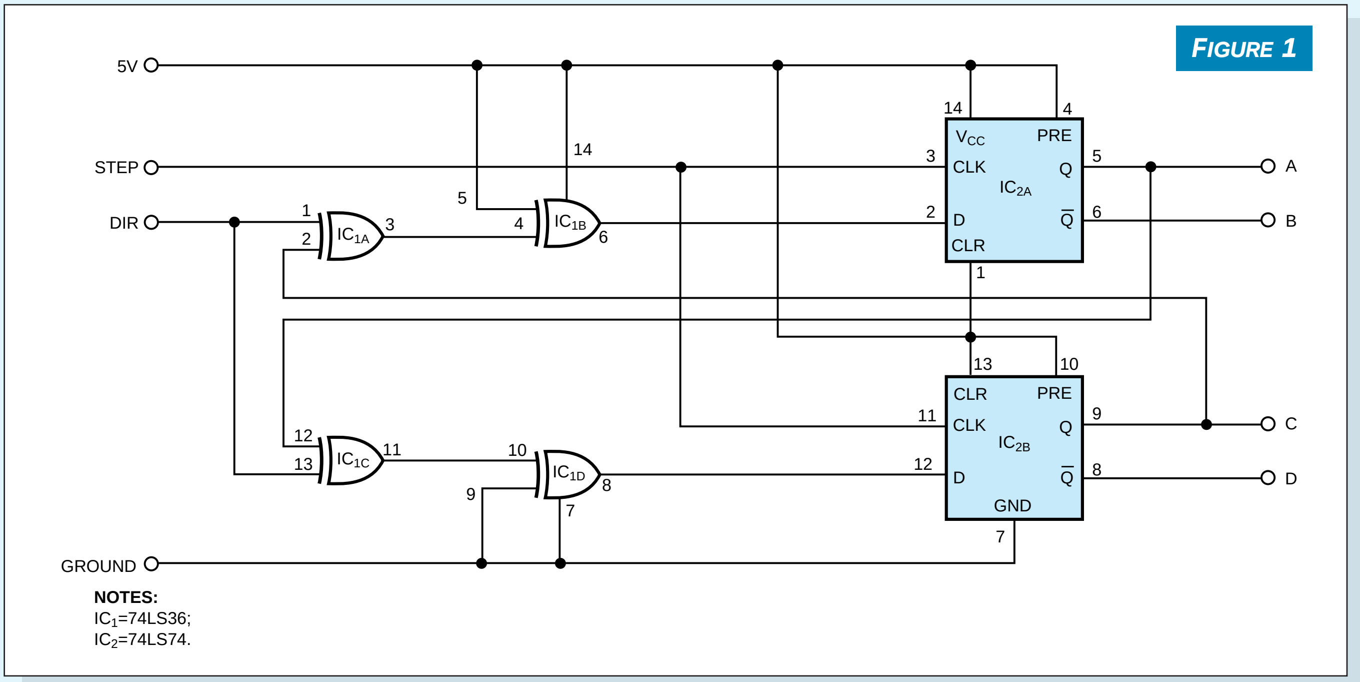
You can then produce the present-state/next-state assignment (**Table 2**) and the next-state maps (**Figure 3**). Then, by inspection, the logical choice is to loop the state maps out for D flip-flops, which produces the following two logic equations:

$$DA=((DIR)(C))+((DIR)\ (C)); \quad (1)$$
$$DB=((DIR)\ (A))+((DIR)\ (A)). \quad (2)$$

Equation 1 is an exclusive NOR, and **Equation 2** is an exclusive OR. To save space, you can use a single quad XOR chip to implement both equations. A dual D flip-flop completes the logic driver, as **Figure 1** shows. Using rising-edge-triggered D flip-flops helps keep the design simple while eliminating mode-change faults.

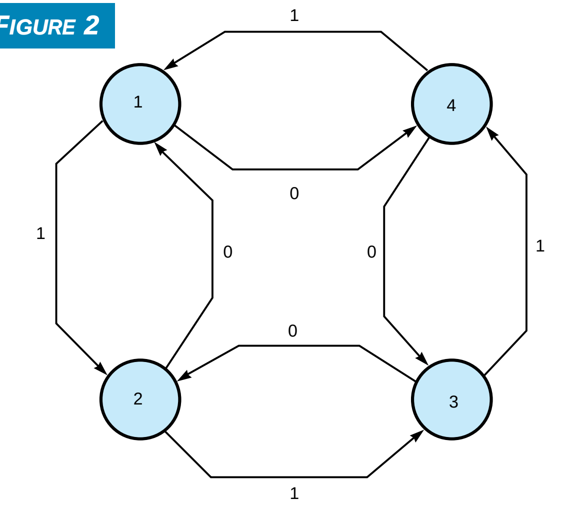
The circuit derives the four outputs from the Q and Q

FIGURE 1



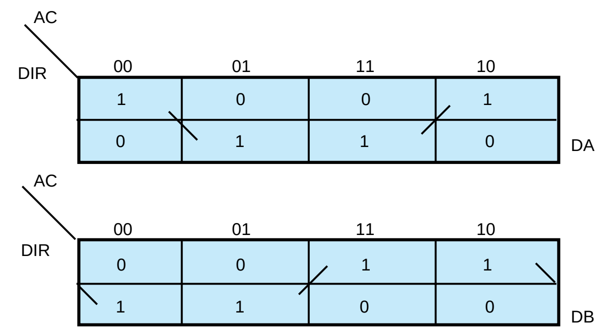
Two logic-level ICs can implement simple and inexpensive control of a stepper-motor driver.

FIGURE 2



The state diagram for the stepper-motor controller comprises just 2 bits.

FIGURE 3



The next-state maps correspond to two simple logic equations.

outputs of D flip-flops IC_{2A} and IC_{2B} in **Figure 1**. IC_{1A} XORs the Q output of flip-flop IC_{2B} with the DIR input, and the circuit transforms the output into an XNOR by using IC_{1B} as a controlled inverter. IC_{1B} then drives the D input of flip-flop IC_{2A}. Similarly, IC_{1C} XORs the Q output of IC_{2A} with the DIR input. The output of IC_{1C} drives XOR IC_{1D}, which acts as a noninverting buffer. The output of IC_{1D} drives the D input of IC_{2B}. Using XOR gate IC_{1D} as a buffer keeps the propagation delays to the D inputs of the flip-flops equal, which helps the circuit avoid any race conditions. The STEP signal is the step-rate input, which drives the clock inputs of both flip-flops.

The last design task is to add the appropriate-sized transistors to drive the stepper motor. In the case of the unipolar motor, output signals A, B, C, and D can directly drive the transistors. To drive a bipolar motor, you can use the A and C outputs to drive one-half of two H-bridges and the B and D outputs to drive the other corresponding half of the H-bridges. This design is possible because the B output is the inverse of A, and D is the inverse of C. (DI #2176)

EDN

TABLE 2—PRESENT-STATE/NEXT-STATE ASSIGNMENT TABLE

Present state			Next state	
A	C	DIR	A	C
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

To Vote For This Design, Circle No. 418