Stepper-motor motion controller and driver fit into a CPLD/FPGA

Stephan Roche, Santa Rosa, CA; Edited by Charles H Small and Fran Granville - June 21, 2007

This Design Idea further develops a previous one integrating a stepper-motor driver in a CPLD (Reference 1). However, this idea integrates not only the driver, but also a simple one-axis stepper-motor motion controller. Depending on the size of the target CPLD, you can implement multiple motion controllers into a single device. For example, a single-axis motion controller fits into a Xilinx XC95108 using 68 of, or 63% of, the available macrocells. The motion controller rotates the stepper motor clockwise or counterclockwise a given number of steps with a given speed profile versus time. When a motion begins, the controller accelerates until it reaches the cruise speed and then decelerates before stopping (Figure 1).

![Figure 1](image)

**Figure 1** This design can control the motor with 16 speeds. The maximum speed in the cruise phase is such that the motor makes a step or half-step every 16 clock cycles.

The controller can adjust the motor speed to 16 values, \( V = V_{\text{MAX}} \times \frac{\text{speed}}{16} \), where speed is an integer with a value of one to 16. During the acceleration phase, the speed ramps up by increasing from one to 16; during the cruise phase, speed stays at 16; finally, during the deceleration phase, speed ramps down to one before stopping. If there are insufficient steps for the controller to reach the cruise phase, the controller goes directly from the acceleration phase to the deceleration phase. You can adjust the acceleration/deceleration rate in the program by the constant “accel,” which can be one to 255. A high value of accel results in a slow acceleration/deceleration, and a low value results in a fast acceleration/deceleration. The inputs of the CPLD stepper-motor controller are clock, direction, full/half-step, reset, Nstep, start, and stop.
The clock input is active on the positive edge of the clock pulse. The maximum motor speed is one step every 16 clocks. The direction input determines the motor’s rotational direction. The motor runs clockwise or counterclockwise, depending on the level of this input and the motor connections. That value is latched at the first rising clock edge after start goes high. The full/half-step input determines the angular rotation of the motor for each clock pulse. In the low state, the motor makes a full step for each applied clock pulse, and, in the high state, the motor makes a half-step. A high level on the reset input sets the motor in a defined state. The motor ignores any clock pulse when reset is high. The 16-bit Nstep value defines the number of steps the next motion will perform. That value is latched at the first rising clock edge after start goes high. A high level on the start input starts the motion, and a high level on the stop input stops the motion, aborting the current motion.

The outputs of the CPLD stepper-motor driver are A, A_N, B, and B_N (Figure 2). The A and A_N outputs control one of the motor’s coils through power drivers, and the B and B_N outputs control the motor’s second coil through power drivers.

![Figure 2: The FPGA/CPLD requires external drivers.](image)

The CPLD/FPGA cannot directly drive the motor, so it requires external drivers. The driver must arrive at the motor’s nominal voltage. The Schottky diodes at the output of each driver allow current freewheeling in the motor coils. If you use MOSFET drivers, external Schottky diodes should be unnecessary because MOSFETs have built-in diodes; the Microchip TC4424A dual driver can drive motor coils to 18V and 3A.

Reference