Stepper Motor Controller Using MAX II CPLDs

December 2007, version 1.0 Application Note 488

Introduction

This application note details the implementation of a stepper motor controller using an Altera® MAX® II CPLD. You can control the motor using just a few inputs from a custom-made keyboard.

Stepper Motors

The advantage of precision control, open-loop control of the motor, self contained braking, and the absence of brushes, makes the stepper motor a convenient choice for a variety of specialized applications. Printers and plotters, disk drives, robots, CNC machines, and other precision machines are common examples where the stepper motor is used extensively.

A stepper motor's operation can be explained by considering a series of electromagnets arranged in a circle to encapsulate a rotor made up of a magnetic material. When these solenoids, or electromagnets, are energized in sequence, the magneto motive force (MMF) developed in them interacts with the rotor and causes it to re-align to the magnetic field, thereby causing it to rotate in a clockwise or counterclockwise direction. The motor's angular displacement can be controlled by simply switching these electromagnets on or off in a fixed pattern for the desired motion of the motor.

Stepper Motor Controllers Using MAX II CPLDs

The motor controller implemented in this design uses a MAX II device to govern (as you have predetermined) the performance and operation of a unipolar permanent magnet stepper motor. The design uses a few switches and buttons on the demo board to serve as the user interface.

This motor controller design offers the following advantages:

- Two types of control for starting and stopping the motor and selecting forward or reverse rotation: manual control (through the user interface) or automatic control (through a microcontroller).
- Two modes of operation: continuous mode and step mode.
- Eliminates the need for an external clock signal as all MAX II devices have a unique internal oscillator which is incorporated in this design.

Figure 1 shows the organization of the stepper motor controller. The driver circuitry is supplied by a DC source, due to the requirements of the motor's windings. In accordance with the controller logic, the ports connected to the CPLD drive the inputs of the motor driver, thereby driving the stepper motor. The controller incorporates logic for half stepping the motor. The inputs to the controller are explained in Table 1.

Figure 1. Implementing a Stepper Motor Controller with a MAX II CPLD

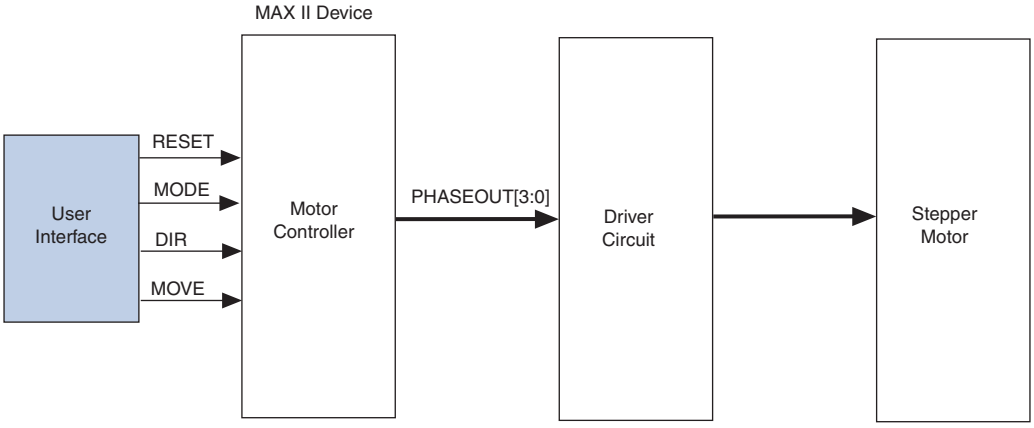

Table 1. Controller Inputs

Signal	Direction	Description
RESET	Input	Resets the position of the motor to the initial reference position.
MODE	Input	Chooses between the continuous mode (normal motor) or the step mode of operation.
DIR	Input	Counterclockwise rotation of the rotor.
MOVE	Input	Required in step mode for the rotor to move in a particular direction by half step (45°).
PHASEOUT[3:0]	Output	4-bit output of the CPLD going to the MOSFET driver circuit.

The rotation steps are controlled by the excitation on each of the coils that follows. A 4-pole stepper motor has four electromagnets uniformly placed over its circumference. The motor controller must provide the following sequence of inputs to these electromagnets through its output ports (PHASEOUT) to rotate the rotor by four times the step size of the motor being used:

1000, 1100, 0100, 0110, 0010, 0011, 0001, 1001, 1000...

The preceding half-step sequence of operation causes the rotor of a 200-tooth, 4-pole motor to rotate by 0.9° at each step. This design offers continuous mode and step mode for motor control. When you select continuous mode (by making the MODE input low), the motor rotates normally, without any discontinuity in rotor displacement. When you select step mode (by making MODE high), the motor operates every time you press the MOVE button. The MOVE input line is normally pulled high by a pull-up resistor. Every time a negative edge is detected on this line, the motor advances by half a step.

The RESET input returns the motor to its reference position. The rotor works until it reaches the position where PHASEOUT is 1000, regardless of its initial displacement.

Implementation

You can implement this design with an EPM240 or any other MAX II CPLD. The stepper motor works by supplying inputs through a user interface, which form the controller inputs to the MAX II CPLD. You can input mode, direction, and step control through user buttons and switches that are assigned to various general purpose I/Os (GPIOs) of the MAX II CPLD.

The following details the implementation of this design on the MDN-B2 demo board using this design source code. For illustrative purposes, this design is made using LEDs. Each LED represents a winding of the motor. The winding energizing sequence can thus be "seen" on the demo board. However, note that the LEDs glow on a logic 0 and are turned off when the port sees a logic 1. You can replace these LEDs by a motor driver circuit to run a motor. All the lines are parallel and independent of each other. Each line excites one separate coil winding of the motor and causes the rotor to rotate in a specific direction and manner as set by the user control.

Table 2 shows the EPM240 pin assignments for the MDN-B2 demo board.

Table 2. EPM240 Pin Assignments

Pin Assignments			
Signal	**Pin**	**Signal**	**Pin**
dir:	pin 38	mode:	pin 37
move:	pin 82	rst:	pin 77
phaseout[0]:	pin 71	phaseout[1]:	pin 72
phaseout[2]:	pin 73	phaseout[3]:	pin 74

Unused pins are assigned as i**nput tri-stated** in the Quartus® II software's Device and Pin option settings prior to compilation.

Design Notes

To demonstrate this design on the MDN-B2 demo board, complete the following:

1. Select the EPM240GT100C3 device, compile the source code, and assign the pins (as shown in Table 2).
2. Switch on power to the board (use slide switch SW1) and download the design to the MAX II CPLD using the JTAG header JP5 on the MDN-B2 demo board and a conventional programming cable (ByteBlaster™ II, USB-Blaster™).
3. Press switch SW4 on the board as you begin the download process. Remove the JTAG connector after programming the MAX II CPLD. Switch off power after programming.
4. Switch on power to the demo board by sliding switch SW1 to **ON**.
5. This version of stepper motor control displays the phase excitation on the LEDs on the demo board. The four phaseout outputs are connected to red LEDs (D5, D6, D8, and D10) on the demo board.
6. DIP switch SW3 (Control Switch) has its switch #1 assigned to set the mode (to either continuous mode or step mode). DIP switch SW3 (Control Switch) has its switch #2 assigned to set the direction. Select any particular mode and direction.

7. Use push button SW9 for the "move" function. Note the LEDs progress in the excitation sequence that was set in step 6 as you push this button. When the mode is set to **step** mode and SW9 is pressed, the LEDs progress one step for each push. When the mode is set to **continuous** mode and SW9 is pressed, the LEDs appear to be moving one step after another without any user intervention. The direction switch changes the direction of this sequence. Push button SW6 on the board is used to reset the demo board.

Source Code

This design has been implemented in Verilog and successful operation has been demonstrated using the MDN-B2 demo board, as referenced in the documentation. The source code, test bench, and complete Quartus II project are available at:

www.altera.com/literature/an/an488_design_example.zip

Conclusion

As shown in this design, MAX II CPLDs are an excellent choice to implement motor control logic. Their low power, easy power-on feature, and unique built-in internal oscillator make them ideal programmable logic devices to implement motor controlling systems.

Additional Resources

The following are additional resources for this application note:

- MAX II CPLD home page:
  http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- MAX II Device Literature page:
  http://www.altera.-com/literature/lit-max2.jsp
- MAX II Power-Down Designs:
  http://www.altera.com/support/examples/max/exm-power-down.html
- MAX II Application Notes:
  AN 428: MAX II CPLD Design Guidelines
  AN 422: Power Management in Portable Systems Using MAX II CPLDs

Document Revision History

Table 3 shows the revision history for this application note.

Table 3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
December 2007, v1.0	Initial release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support
Literature Services:
literature@altera.com