



XAPP940 (v1.0.1) March 23, 2009

Using Xilinx CPLDs as Motor Controllers

Summary

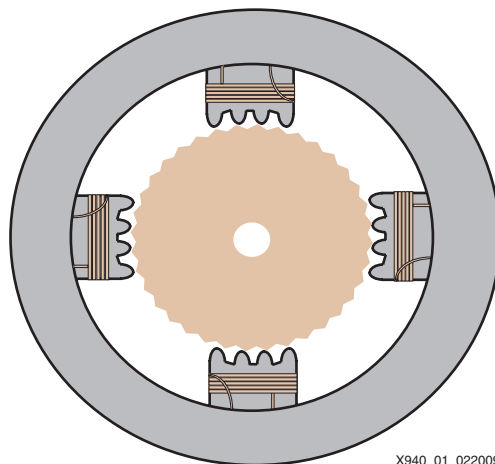
This document discusses the different types of motors commonly found in electronic systems and their different uses. It shows how a Xilinx® CPLD can be used as a simple stepper motor controller.

Introduction

There are many different types of motors used in electronic systems – each with their own advantages. Standard DC motors are often found in automotive applications as well as consumer applications and even toys. Brushless DC motors are commonplace in power tools and, along with standard DC motors can operate in the medium to high speed range. AC induction motors are often found in white goods and transportation applications and these can operate at very high speeds. On the lower end of the operating speed range, where extremely high speed responses are not often required, stepper motors can be found. These have a huge variety of different uses in the consumer and industrial markets including a variety of different positioning systems; printers, scanners, plotters, disk drives, fax machines, and medical equipment.

Stepper Motor

A stepper motor consists of four electromagnets that are charged at different times to move a rotor by a fixed degree of rotation (Figure 1). The degree of rotation is dependent on the number of teeth present on the rotor and the stator. At the low-cost end, the motor's rotor rotates through 7.5 degrees per step, or 48 steps per revolution. More accurate, higher cost versions have a basic resolution of 1.8 degrees per step. Furthermore, it is possible to half-step these motors to achieve a resolution of 0.9 degrees per step. Stepper motors tend to have a much lower torque than other motors, which is advantageous in precise positional control. There are different ways in which motors can be advanced – wave drive, full step drive and half step drive. The example in this document shows a motor operating in wave drive mode – where only one phase is on at any time.



X940_01_022009

Figure 1: Stepper Motor

To enable a CPLD to turn the electromagnets on and off, external transistor drivers must be used. A unipolar stepper motor can be driven by four identical NPN or N-channel drive transistors. The various Xilinx CPLDs can operate with I/O voltages from 1.5V to 5V. The CPLD capable of delivering the appropriate output voltages should be selected for use with transistor drivers that can accept and understand the voltage. For more information on device I/O voltages and V_{IH}/V_{OH} levels, consult the device data sheets available on www.xilinx.com.

CPLD Design

Within the Xilinx CPLD, eight inputs are required to fully control the stepper motor. The CLK input synchronizes the logic and determines the speed of rotation. The motor advances one step per clock period. The angle of rotation of the shaft will depend on the specific motor used. The direction (DIR) control input changes the sequence at the outputs (PH1 to PH4) to reverse the motor direction. The enable input (EN) determines whether the motor is rotating or holding. The active low reset input (RST) initializes the circuit to ensure that the correct starting sequence is provided to the outputs. The feedback signal (POSITION_FEEDBACK) tells the CPLD in which phase the rotor is located. Not until the rotor reaches the next electromagnet will the feedback signal increment. The design is very small, leaving resources in the CPLD for further functions on the board if needed (Table 1). This design fits on an XC2C32A device in a QF32 package.

Table 1: Device Utilization

Macrocells Used/Total	Product Terms Used/Total	Function Block Inputs Used/Total	Registers Used/Total	Pins Used/Total
4/32 (12%)	24/112 (21%)	10/80 (12%)	4/32 (12%)	12/21 (57%)

The hardware description motor control module can be embedded within a macro function and saved as a reusable standard logic block, which can be shared by many designers within the same organization – this is the beauty of design reuse. This “hardware” macro function is independent of any other function or event not related to its operation. Therefore, it cannot be affected by extraneous system interrupts or other unconnected system state changes. Such independence is critical in safety systems. Extraneous system interrupts in a purely software based system could cause indeterminate states that are hard to test or simulate.

Alternatively, the stepper motor controller design can work in an open loop mode, requiring no feedback from the motor. To achieve this, the number of steps per rotation of the motor must be known. Then a counter can be implemented to keep track of the number of cycles for which the motor has been enabled in each direction giving the position of the rotor.

Design Source Code

The ISE design files and VHDL code to compile and simulate these designs are located at:

<https://secure.xilinx.com/webreg/clickthrough.do?cid=55658>

Conclusion

Stepper motors are commonly found in consumer, industrial, medical and scientific applications. The Xilinx CoolRunner-II CPLD is the perfect device with which to control stepper motors with the added benefit that the CPLD can be reprogrammed to accommodate a different motor if the system specifications change.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/22/06	1.0	Initial Xilinx release.
03/23/09	1.0.1	Updated link to reference design.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.