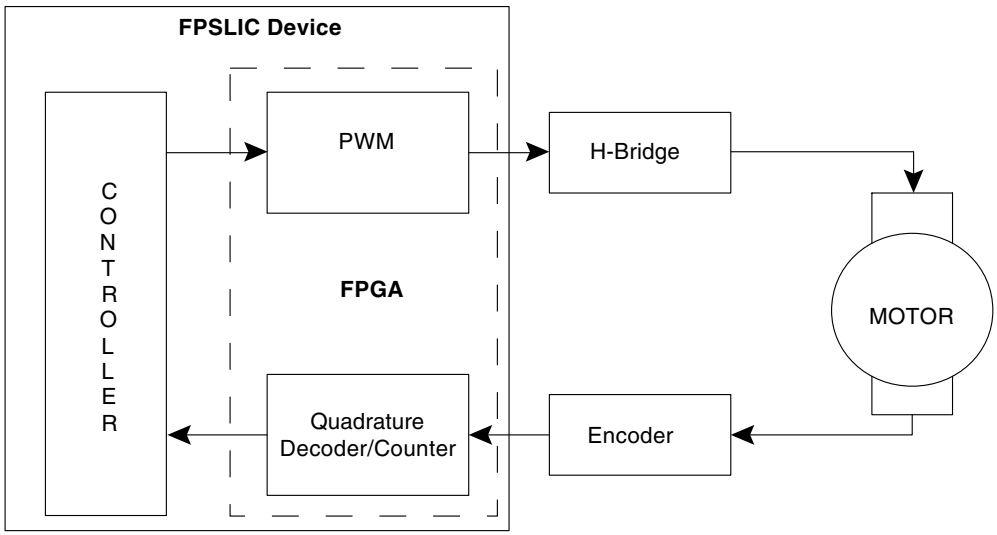
Motor Control using FPSLIC™/FPGA

This application note describes the implementation of Pulse Width Modulation (PWM) and Quadrature Decoder/Counter modules for motor control and motor sensor applications.

The on-chip FPGA (up to 40K gates) can be used to implement multiple programmable PWM and Quadrature Decoder modules, allowing designers to implement multiple channels of programmable PWM and Quadrature Decoders. The high performance 8 - bit microcontroller provides up to 3 additional PWM counters that can be configured as 8-, 9- or 10-bit PWM's. The added advantage of the on-chip microcontroller is that it can be used for reading PWMs and Quadrature decoders, and to perform the required processing in determining the motor speed and the distance traveled, providing the single chip solution for most of the motor related applications, see Figure 1. Typical applications are printers, plotters, Data X-Y Logger, industrial and medical precision equipment, robotics, audio amplifiers, copiers, scanners and many others.

Figure 1. Motor Control Block Diagram



Programmable SLI
AT40K
AT94K

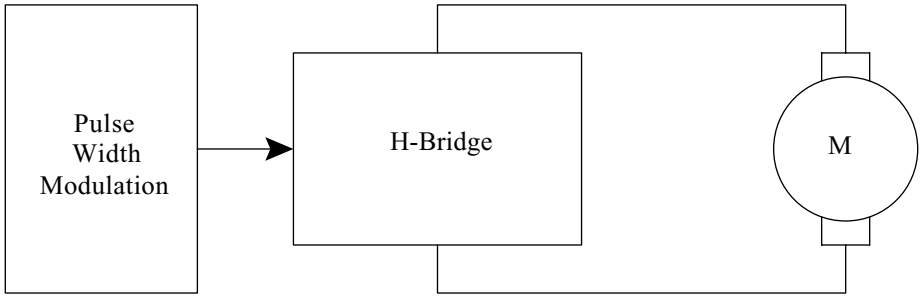
Application Note

Rev. 3023A–FPSLI–06/02

Description

To control the speed of a D.C motor we need a variable voltage D.C power source. Pulse Width Modulation provides a logic “1” and logic “0” for a controlled period of time. The speed of the motor can be adjusted by changing the Pulse Width Ratio. To control the speed of the motor, the switches of the H-bridge are opened and closed at different rates in order to apply different average voltages across the motor, see Figure 2.

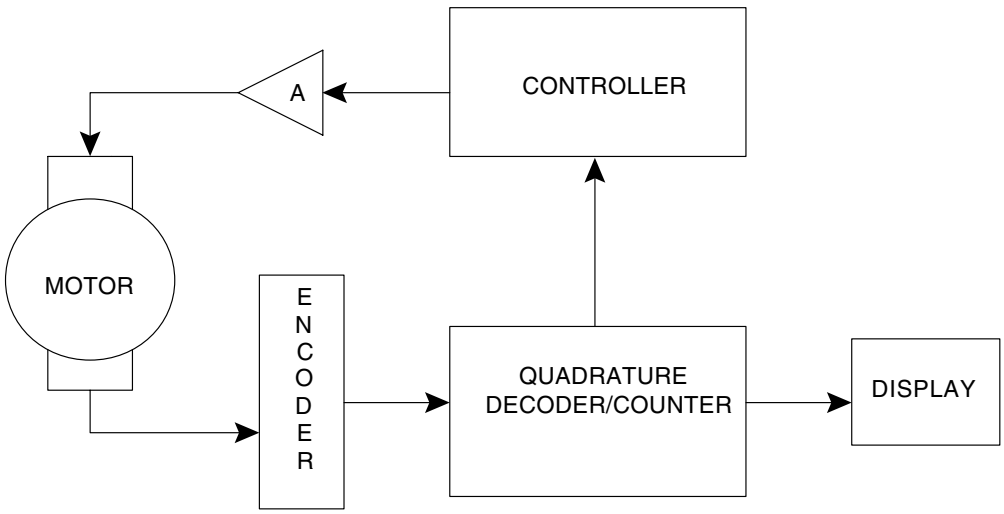
Figure 2. Typical Application of PWM in Motor Speed Control



The quadrature decoder/counter takes some of the real-time computational load off the microprocessor. When the processor reads the output signals of the position encoder, every change in state must be detected. This means that for an encoder of 250 pulse per channel per revolution turning at a modest 6000 rpm, the processor must detect and decode 100,000 state changes per second. This is difficult for most microcontrollers, and many systems use higher speeds and/or even denser encoders.

The solution is a circuit that will read the encoder outputs (90 degree out of phase), and detect each of these state changes and generate the direction signal depending upon the relation between the encoder outputs. Each time a state change is detected in the positive direction, the decoder will increment the counter; a change in the other direction causes the counter to be decremented. Thus the counter will keep a running count of how far the encoder has moved, until it overflows. If you use an 8-bit counter, the counter can count from -128 to +127. Now the microprocessor can read the number in the counter and compare that number to a previous reading in order to measure the distance traveled, see Figure 3.

Figure 3. Typical Application of Quadrature Decoder/Counter in Motor Control



Design Implementation

Pulse Width Modulator

This document details VHDL® implementations of a 12-bit Pulse Width Modulation timer. The PWM timer is often used instead of a Digital to Analog Converter (DAC) and is well suited for FPGA implementations. With a filter outside, it can be used to generate slow analog waveforms.

The first implementation uses a simple comparator. This allows a system frequency of +30 MHz. The second implementation uses a 4-stage pipelined 12-bit comparator to achieve 50 MHz + system frequency.

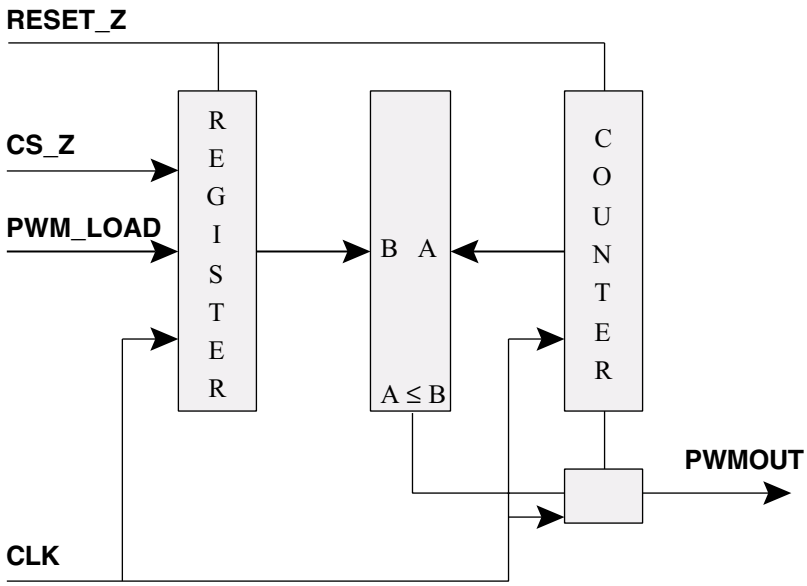
The PWM timer consists of three basic elements: a 12-bit wrap around up counter, a PWM register, which contains the expected duty cycle; and a comparator, which compares the counter with the register, see Figure 4. The PWM output is High as long as the counter is less than or equal to the register. The user can load the register by keeping the chip select (CS_Z) Low at a system clock edge and thus program the duty cycle of the PWM timer.

The PWM frequency is calculated as:

$$f_{PWM} = f_{system} / 2^{BITWIDTH}$$

However, a minimum PWM frequency around 10 kHz would keep the cost of the digital filter down.

Figure 4. Simple PWM Timer



While the design above is simple, the PWM frequency is a little bit lower than desired. Synthesizing the design for an AT40K05AL (see VHDL file) with Leonardo v1999.1i and Place and Route with timing constraints using Atmel Figaro (IDS) provided a maximum system clock cycle of 31 MHz +. A system frequency of 30 MHz is selected to provide some margin and the PWM frequency can be calculated as:

$$f_{PWM} = 30 \times 10^6 / 2^{12} = 7{,}324 \text{ kHz}$$

This is still lower than the desired 10-20 kHz PWM frequency.

It is always possible to increase PWM frequency by reducing the resolution. If the PWM timer is reduced to 10 bits, then the PWM frequency becomes:

$$f_{PWM} = 30 \times 10^6 / 2^{10} = 29{,}297 \text{ kHz}$$

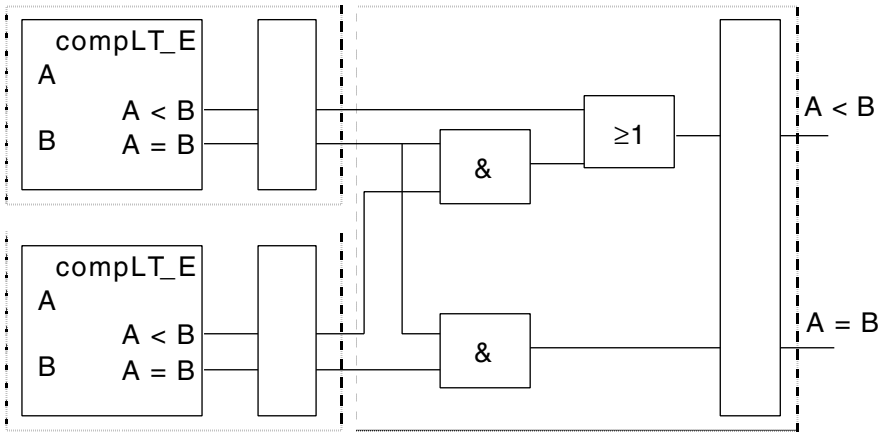
It is likely that, when the bit width goes down from 12 to 10 bits, the system frequency can be increased as well, providing a further increase in the PWM frequency. If the 12-bit resolution is important, then other means are needed. The desired system frequency to achieve a 12-bit resolution and a 10-kHz PWM frequency is:

$$f_{SYSTEM} = f_{PWM} \times 2^{BITWIDTH} = 10{,}000 \text{ X } 4096 = 40{,}96 \text{ kHz}$$

The static timing analysis provided by Figaro is used to find the critical path. This turned out to be the 12-bit compare function and it was decided to move to a pipelined architecture of the comparator. The pipelined comparator introduces a latency of one or more clock cycles, which sometimes is a problem. Specifically in a PWM timer, the problem does not exist as long as the phase is not important. Regardless of the phase, the duty cycle remains the same.

Two different pipelined comparator designs were tested. The first pipeline uses a 4-stage pipeline and the second uses a 1-stage pipeline followed by combinatorial logic. Care was taken to implement in a fashion where a single CLB would contain all the functions in a pipeline stage. The AT40K CLB has up to 4 inputs so not more than 4 inputs were used, see Figure 5.

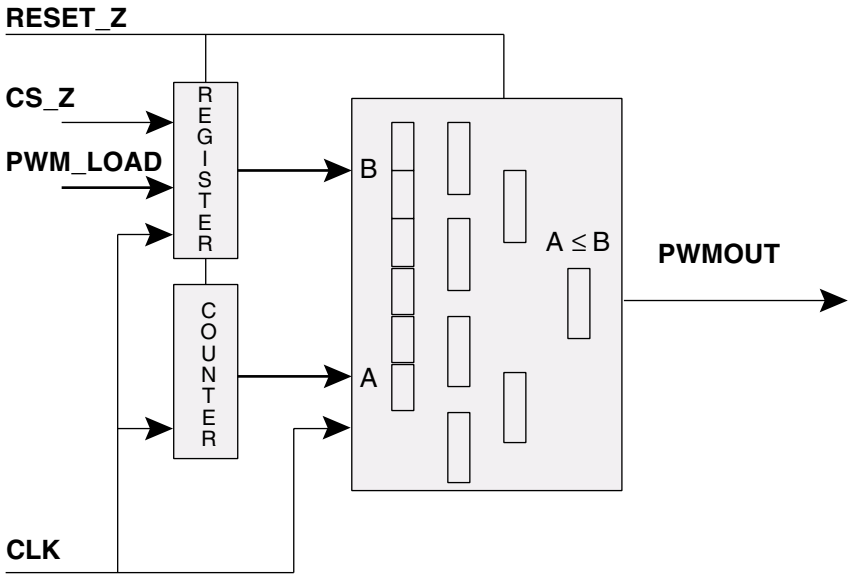
Figure 5. Pipelined Comparator Decision Logic



The 4-input limit means that a 2-bit comparator is the largest possible, which can fit into a single CLB. The first pipeline stage consists of 6 x 2-bit comparators, which each provide an “Equal” and “Less Than” output.

The second pipeline stage will look at the output from two 2-bit comparators again generating “Equal” and “Less Than” outputs, forming a 4-bit comparator, see Figure 6.

Figure 6. PWM using a Pipelined Comparator



The “Less Than” signal is asserted if the most significant 2-bit comparator activates its “Less Than” signal or, if the most significant comparator activates the “Equal” signal, the “Less Than” signal from the least significant comparator is forwarded as the result.

Further pipeline stages will look at the output of two 4 or 8-bit comparators and generate the final result.

The PWM OUTPUT signal is asserted if either the “Equal” or “Less Than” signal is asserted.

The comparator block was synthesized using Leonardo and a Timing driven Place and Route was done. The achieved frequency was 76 MHz, showing promise for the complete PWM timer design.

When the complete PWM timer block based on the new pipelined timer block was analyzed, the resulting frequency became 51 MHz+ providing a PWM frequency of:

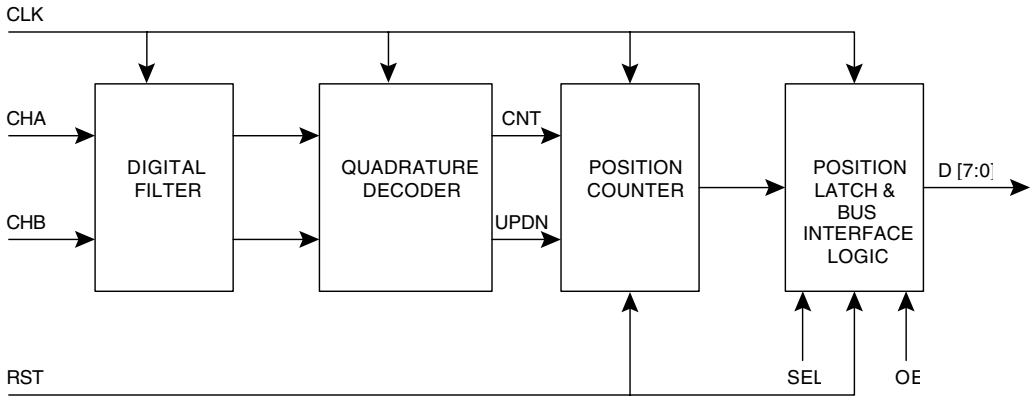
$f_{PWM} = 50 \times 10^6 / 2^{12} = 12{,}207$ kHz

The design uses 68 Logic cells, of which 49 are registered, and 19 are pure combinatorial.

Quadrature Decoder/Counter

A Quadrature Decoder/Counter is used to decode the signals from a quadrature encoder and to maintain an internal counter, which represents the encoder position. The quadrature decoder/counter module has a 16-bit position counter, 1-bit position latch and an 8-bit microprocessor interface. The module reads the 90-degree out of phase singles (CHA, CHB) from the encoder and decodes the signals to determine the direction using internal state machine and decoding logic. The position counter counts up or down based on the up/dn signal from the decoder and the counter value is stored in the position latch. The bus interface logic will allow placing the 8-bit data on the data bus. SEL controls which byte is being accessed, OE allows the chip to place data on the microprocessor data bus, and the falling edge of CLK is used to initiate data transfers, see Figure 7.

Figure 7. Quadrature Decoder/Counter



Digital Filter

The Digital filter is responsible for rejecting noise on the incoming quadrature signals. This is achieved by using Schmitt triggered inputs and a three-clock cycle delay filter, which reject low-level noise and large, short duration noise spikes that typically occur in motor system applications.

The signals on each channel are sampled on rising clock edges. A time history of the signals is stored in the four-bit shift register. Any change on the input is tested for a stable level being present for three consecutive rising clock edges. Therefore, the filtered output waveforms can change only after an input level has the same value for three consecutive rising clock edges. By this method the short noise spikes between rising clock edges are ignored and pulses shorter than two-clock period are rejected.

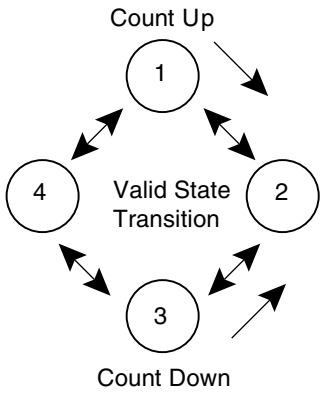
Table 1. Pin Description

CLK	System Clock
CHA CHB	CHA and CHB are inputs that accept the outputs from a quadrature-encoded source, such as incremental optical shaft encoder. Two channels, A & B, nominally 90 degrees out of phase.
RST	Asynchronous active Low reset signal, which clears the internal position counter and position latch.
OE	Active Low output enables input.
SEL	The input controls the data byte from the position latch is enabled into the 8-bit tri-state output buffer. “0” selects High byte and “1” selects Low byte
CNT	Pulse generated from the decoder when the quadrature decoder has detected a state transition
UP_DOWN	Output from the quadrature decoder. This signal will determine whether the counter is counting up or down.
D [7:0]	8-bit output data bus

Quadrature Decider

The decoder logic decodes the incoming filtered signals into count information. The decoder samples the outputs of the Channel A & B samples. Based on the past binary state of the two signals and the present state, it outputs a count signal (CNT) and direction signal (UP_DOWN) to the position counter. Channel A leading Channel B results in counting up. Channel B leading channel A results in counting down, see Figure 8.

Figure 8. Decoder Logic



For better system control and improved resolutions designer can multiply the resolution of the incoming signals by a factor of four (4 x decoding)

Position Counter

Binary up/down counter that counts on rising clock edge. Channel A leading channel B results in counting up. Channel B leading Channel A results in count down.

Position Data Latch

The position data latch captures the position counter output data on each rising clock edge, except when its inputs are disabled by the inhibit logic during 2-byte operation. The inhibit logic samples the OE and SEL control signals and does the conditional inhibit of the position data register, see Table 2.

Table 2. Inhibit Logic Samples

SEL	OE	CLK	INH	Action
L	L	H-L	H	Read High Byte
H	L	H-L	H	Read Low Byte
X	H	X	0	Output Tri-state

Bus I/F Logic

Consists of multiplexers and tri-state output buffer. This allows the independent access to the low and high bytes of the position data latch.

A simple two-channel quadrature decoder/counter is implemented using VHDL. The design involves two-channel digital filter, Quadrature decoder, 16-bit position counter, 16-bit position latch and 8-bit bus interface logic.

The design is successfully fitted into Atmel AT40K05AL (87 core cells and 17 I/OS).

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

Atmel Programmable SLI Hotline
(408) 436-4119

Atmel Programmable SLI e-mail
fpslic@atmel.com

FAQ
Available on web site

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

3023A–FPSLI–06/02 xM