How to use FPGAs for quadrature encoder-based motor control applications

Glen Young, Actel - September 11, 2007

Precisely tracking speed, acceleration, and position of a motor's rotor is an essential requirement for many motor control applications found in everyday equipment such as fax machines, elevators, and medical equipment. A closed-loop control scheme is able to bring motor feedback information, such as back electromotive force (BEMF) voltage or supply current to the control system. Rotary encoding is a common mechanism for the delivery of accurate speed, acceleration, and position of the motor rotor.

Rotary encoders are commonly deployed in the closed-loop rotor systems used in a wide variety of applications from robotics and high end photographic lenses to opto-mechanical mice and trackballs to rotating radar platforms. A rotary encoder is an electro-mechanical device for converting the angular position of a shaft or axle to a digital code. For many applications and equipment that need to track object location, velocity and accelerations accurately, a rotary encoder offers a cost-effective solution.

Relative and Absolute are two primary types of rotary encoders. A quadrature encoder is in the relative encoder family and is most commonly used in high-speed motor control systems; it also facilitates the ability to determine motor direction.

Relative rotary encoder
The relative rotary encoder, also known as incremental encoder, is an applied optical encoding methodology. Generally, the rotary encoder utilizes a patterned spinning disc attached to the motor shaft marked with a large number of radial lines like the spokes of a wheel (see Fig 1 below). An optical switch, such as a photodiode, generates an electrical pulse whenever one of the lines passes through its field of view. An electronic control circuit counts the pulses to determine the angle through which the shaft has turned.
1. Example of rotary encoder.

This system, in its simplest form, cannot measure the absolute angle of the shaft. It can only measure the change in angle relative to some arbitrary datum, such as the shaft's position at the time when the power was switched on. This uncertainty is not a problem for computer input devices, such as mice and trackballs. However, when an application demands the determination of absolute position, a second sensor can be added that detects when the shaft passes its zero position. Neither of these two implementations can determine the direction of rotation. Rotation direction is determined by adding a second sensor placed at a slightly different angle around the shaft from the initial sensor. The direction of rotation can then be inferred from the order in which the two sensors detect each radial line. This type of encoder is known as a quadrature encoder.

Quadrature encoder

A quadrature encoder is a common type of incremental encoder that supports two output channels (A and B) to sense position, velocity, and direction of rotation. Using two code tracks with sectors positioned 90 degrees out of phase, the two output channels of the quadrature encoder indicate both position and direction of rotation. If A leads B, the disk is rotating in one direction (clockwise in this case). By comparison, if B leads A, then the disk is rotating in the opposite (or counter-clockwise) direction (Fig 2).

2. Quadrature encoder output channels.

Position monitoring

In order to decode the output from quadrature encoders, two sequential samples are required. Sample from Time 'n' and Time 'n+1' (Tn and Tn+1) respectively. These two sets of data points create a 4-bit word that is used for a 16 entry look-up table. The table will contain four each of "go-
The major advantage offered by a quadrature encoder is high resolution. Precision quadrature encoders are available that produce over 10,000 pulses per revolution at speeds exceeding 5000 RPM. Their high resolutions, extreme durability, and ease of installation make quadrature encoders ideal for a wide variety of applications in many divergent markets such as:

- **Office equipment**: printers, fax machines and copy machines use quadrature encoders to synchronize various moving parts for trouble free operation.
- **Industrial Machinery**: elevators use quadrature encoders to maintain velocity and acceleration and for correct door alignment.
- **Medical**: automated blood analyzers use quadrature encoders to ensure the exact position of vials containing blood samples while performing automated tests.

High-speed and precision applications that utilize these quadrature encoders for velocity and acceleration control require hardware that can deterministically process quadrature encoder signals to determine velocity and acceleration and create the corresponding output controls signals at very low latency.

**Velocity and acceleration monitoring**

There are several accepted methods for determining the angular velocity and acceleration of a quadrature encoder. The following is an example that will count the number of quadrature encoder pulses in a fixed time interval to estimate the velocity and acceleration of the encoder - Fig 3 demonstrates this procedure. This method is appropriate for high speed applications.

3. **Speed estimation.**

Once the number of pulses in a fixed time interval is measured, the angular velocity of the quadrature encoder can be calculated using the following formula:

\[
\text{Velocity} = \frac{\text{Encoder Pulses}}{\frac{\text{Pulses Per Revolution}}{\text{Fixed Time Interval (sec)}}} \times \frac{60 \text{ sec}}{1 \text{ min}} \quad \text{(RPM)}
\]

Where "Encoder Pulses" is the number of quadrature encoder pulses received in the Fixed Time Interval.

Acceleration is the rate of change of velocity. The following formula can be used to estimate the acceleration of the quadrature encoder:
Acceleration = \[
\frac{\text{Encoder Pulses}_n - \text{Encoder Pulses}_{n-1}}{\text{Pulses Per Revolution} \div (\text{Fixed Time Interval})^2 (\text{sec}^2)} \times \left(\frac{50\text{ sec}}{1\text{ min}}\right)^2 (\text{RPM}^2)
\]

Where the numerator divided by one Fixed Time Interval represents the change in velocity, and dividing the change in velocity by one Fixed Time Interval gives the acceleration. **Quadrature encoder interface and decoder implementation on an FPGA**

The quadrature encoder interface (QEI) will enable the user to connect the encoder to the FPGA, performing the required operations to effectively use the information coming from the encoder. The two quadrature encoder output signals (channel A - QEA and channel B - QEB) are affected by noise; therefore, each input must be filtered to account for system noise.

The programmable fabric of the FPGA lends itself to implementing filters quickly and efficiently. The filtered phase edges are counted by a dedicated Position Counter, usually a 16-bit up/down counter. To establish a reference point for position and speed measurements, the counter can be reset either by using an index signal or by a counter period match. The hardware can also perform error checking on the accumulated count.

**4. Quadrature encoder interface.**

A quadrature decoder is part of the quadrature encoder interface circuitry. Signals will be read and converted into a numeric count of the position pulses. The count will increment in one direction and decrement in reversed direction when the shaft is rotating. The quadrature decoder determines the direction of rotation looking at the two incoming phase signals, and generates corresponding clock that is used by the position counter. In the ×2 mode, impulse at the rising and falling edges of Phase A signal; in the ×4 mode, the clock pulses are generate at each edge of phase A and Phase B. The position counter can be reset externally or by programmed maximum count register.

**5. Quadrature decoder.**

*Fig 5* provides an example as to how the decoder works where the ×4 mode is applied. The clock pulse is at each edge of both phases. In the first part of the timing diagram Phase A leads Phase B, so that the counter is counting up. Then, in the second half, the rotation of the rotor is reversed,
Phase B now leads Phase A and the counter is counting down. This is why an up/down counter is required in this application.

The count direction can be determined by reading the **UPDN** bit in the QEI control register, but the **UPDN** pin can also be used to indicate the count direction status.

**Input Filter:** Filters are for screening out noise from multiple inputs. The filter clock is used for sampling the input signal: if - and only if - three consecutive samples have the same value, the input is considered stable and the value is output from the filter. One of the effects of this sampling is an added latency, because of the propagation delay from input signal through the filter.

**Clocking:** With the high speed mode we can get a very high angular resolution, but we also get a relatively high output clock frequency. With the low speed mode, the resolution is fine, but the frequency will be lower. (The maximum allowed quadrature frequency is one-third of the instruction cycle frequency.)

**Position Counter:** The position counter can be used either for position or speed measurement. To measure motor position, we must know the relationship between the displacement and the number of phase pulses we get from the encoder. This relation can be known in advance, or can be measured during initialization by accumulating the total count for the maximum allowed displacement. We can set a constant value in the Maximum Count register, which is typically the number of pulse edges generated by one encoder revolution. As soon as we have a match between the Position Count and the Maximum Count, an interrupt is generated. In the Interrupt Service Routine, the user software can increment or decrement a software counter containing the most significant bits of the position count. In the case of a speed measurement application, the time interval between two index pulses or count match events gives a measure of the angular velocity.

**Benefits of Fusion implementation**
Fully compliant in QEI logic deployment, the flash-based, mixed-signal Actel Fusion Programmable System Chip offers on-chip Clock Conditioning Circuit (CCC) with phase-locked loop (PLL) and analog front-end circuits with ADC, bringing extra values to the sophisticated motor control scheme.

**Dynamic speed monitoring:** The PLL and CCC of Fusion devices provide designers with very flexible clock conditioning capabilities. Enabling very flexible clock conditioning capabilities, each mixed-signal FPGA contains six CCCs. In the two larger family members, two of these CCCs also include a PLL; the smaller devices each support one PLL. The wide range of output frequency range, 1.5 MHz to 350 MHz, can be controlled dynamically through internal FPGA and external input. This brings the benefit of providing proper filter clock for the various quadrature encoder channel output speeds. Please see following diagram.

![Dynamic clocking for input signal filter](image)

6. **Dynamic clocking for input signal filter.**

**Multiple quadrature encoder support:** Mixed-signal FPGAs also offer another benefit for motor
control - the capability of driving multiple motors (axles) simultaneously with single mixed-signal device. Additionally, with multiple CCC/PLL circuitries, multiple quadrature encoder interface logics can be easily implemented without any overhead.

**Single-chip solution:** On-chip analog inputs and gate drive outputs, like those offered by today's mixed-signal FPGAs, are the perfect solution that can offer temperature and current monitoring in the multiple motors control system. On-chip MCU processes information from the quadrature decoder and responds with the seamless control algorithm to drive motors or multiple motors efficiently.

**Conclusion**

Quadrature encoders are commonly used to provide rotor position and speed monitoring mechanisms. Their accuracy and performance are superior to traditional Hall Effect sensor and BEMF in high-speed motor control schemes.

Actel Fusion, a mixed-signal FPGA device, is able to implement multiple quadrature encoder interfaces to precisely monitor rotor position, speed, and acceleration. The on-chip CCC/PLL circuitries offer a dynamic speed tracking mechanism for various speed alternations efficiently. Moreover, with on-chip processor and analog I/Os, Fusion is a true single device solution that can support multiple sophisticated motor control systems simultaneously without extraordinary overhead.

_Glen Young_ applies more than 20 years experience in the semiconductor industry to his position as product marketing manager for Actel. His background includes product management roles with many high-profile companies including Infineon, Hifn, IDT and NEC. Prior to that, he was an applications engineer at NEC and Intel, and held design engineering positions with Intel and Acer. Glen holds a master's degree in electronic engineering from California Polytechnic State University in San Luis Obispo, and a bachelor's degree in electronic engineering from Tamkang University in Taiwan.