Actel POWER MATTERS

Rate This Document

Motor Control Reference Designs for SmartFusion Using Field Oriented Control and SVPWM

FlashLock

Key Features

- Complete source code in C and Verilog
- Torque, speed, and direction control
- Field oriented control for 3-phased PMSM
- Multi-axis control
- Hall effect, encoder and sensorless (later) feedback
- Critical fault detection using hardware
- Communication via Ethernet
- Quadrature encoder inputs (QEI) in hardware
- Space vector PWM generation with dead band
- Hardware/software partitioned for optimum result
- Programmable FPGA for flexibility and upgrade
- Programmable ADC for flexible triggering

Key Applications

- Industrial drives
- Factory automation
- Robotics
- Medical
- HVAC and appliances
- BLDC or PMSM
- Multi-axis motor control

Additional Information

Contact Actel Sales for more information:

www.actel.com/contact/default.aspx

Description

The SmartFusion™ motor control reference design package includes proven source code and firmware to implement motor control in SmartFusion mixed signal FPGAs.

The designs use a single SmartFusion device controlling up to four permanent magnet synchronous motors (PMSM) simultaneously using field oriented control (FOC). Feedback mechanisms, including incremental encoder, Hall sensors, and sensorless feedback (to be released later) are demonstrated in the reference designs. The designs also use TCP/IP protocol via SmartFusion's built-in 10/100 MAC to communicate to a host PC for status and commands.

The designs implement field oriented control and space vector PWM, providing greater torque control, thus reducing torque ripple, increasing system efficiency and improving precision control.

With the purchase of the Motor Control Reference Designs, the user has the right to modify the reference designs and use them for their own production systems, without additional licensing or royalties. The user does not, however, have the right to distribute the design or any part of it to any other user or their own end customers.

The Motor Control Reference Design package also includes schematics for the Motor Control Development Kit. These schematics are currently going through production and manufacturing test so should be used as a guideline only.

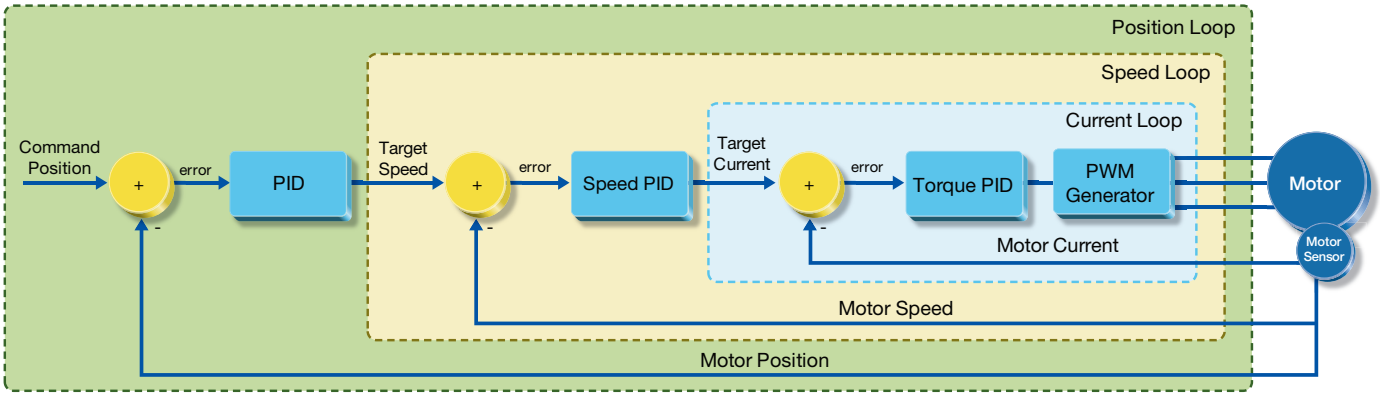

Figure 1 • **Motor Control Example**

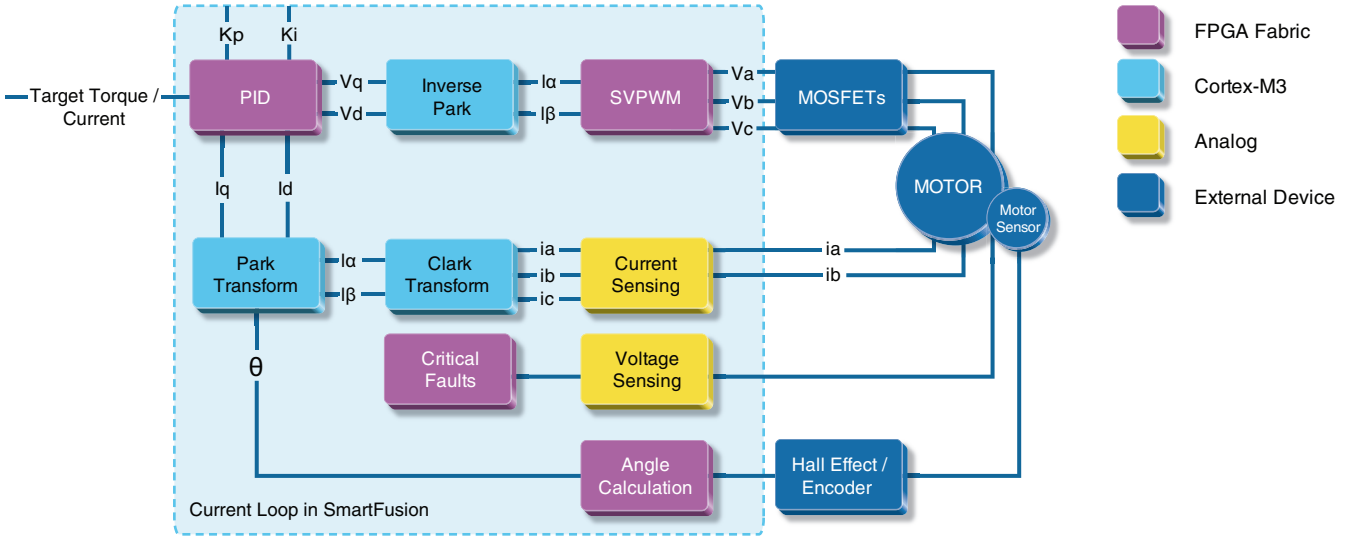

Figure 2 • **Motor Control Implementation in a SmartFusion Device**

The design utilizes analog, software, and hardware features of the SmartFusion device. The on-chip programmable ADC translates current and voltage from analog to digital form. Sampling this digital data can determine motor status and detect critical faults. Motor position data is gathered from sensors via digital interface or interpolated from other ADC samples. Software is executed on the ARM® Cortex™-M3 microcontroller to perform various functions such as the Clark/Parke/Inverse Parke transform, communication to the host, triggering computation in fabric, and reporting status. The FPGA fabric is used to offload processor bandwidth and reduce computation time, acting as hardware acceleration. The most time consuming functions in the FOC algorithm are placed inside FPGA fabric, such as PI controller calculation, space vector generation, PWM waveform using counter, and comparators. The result is an improved control of the inner current loop, thus yielding higher performance motor control.

Table 1 • **SmartFusion Features Used in Motor Control**

SmartFusion Features	Motor Control Function
Cortex-M3	Management of all components and functions
	Processing and simple calculation
	Software execution of less critical functions
10/100, UART, I^2C, SPI	Communication protocols with external device/host/peer
Flash	Code and parameter storage
SRAM	Temporary code storage or boot
ADC and ACE	Converts current into digital (used in Clark Transform)
	Converts voltage into digital (used in Back-EMF)
	Comparators used to catch faults
Analog I/Os	Analog input of motor current and voltage
FPGA	Hardware implementation of system critical tasks
	Customized protocols and communication (CAN)
	Customized algorithm (PID, SVPWM, etc.)
FPGA I/Os	PWM outputs, encoder feedback, Hall sensor feedback