Advantages of using FPGAs in precision inverter modules

Authors:

Benjamin Schuele, ARADEX AG
Johannes Eha, ARADEX AG

WHITE PAPER

ARADEX AG
Ziegelwaldstraße 3
73547 Lorch - Germany

Contact:
Thomas Vetter

Phone:
+49 7172 / 9181 - 0
Fax:
+49 7172 9181- 91
E-Mail:
info@aradex.com

Date:
26. July 2013

Abstract

The use of FPGA technology in high-precision servo drives offers the following important advantages compared to standard solutions based on microcontrollers or digital signal processors (DSPs):

1. FPGA technology can implement many calculations and processes parallel and with deterministic behavior. This enables short cycle and response times, and low jitter during extremely complex calculations.
2. Many sequences with cycle times independent of another can be processed simultaneously.
3. Use of a soft-core processor (Nios II) in the FPGA means that additional calculations and processes can be programmed offering the developer maximum flexibility regarding performance and resource consumption.

Using the example of an integrated servo drive developed for a specific application, in high-end (CNC) machining centers, the following pages will show how the specific advantages of FPGAs enabled maximum demands on regulation accuracy.

Introduction

The given project made extremely high demands on the regulation accuracy of a servo drives. These could not be achieved with standard servo drives available on the market.

In order to achieve these extremely challenging accuracy requirements, a very short cycle time for the position control loop and an ideal combination of the position control loop with subordinate structures such as the current controller had to be achieved.

The customer-specific drive system can control up to three motors simultaneously.

A PC-based real-time controller was integrated together with the power electronics in a compact housing. A very fast communications interface in LVDS technology was developed for the data exchange between the controller and the power electronics.

An Altera Field Programmable Gate Array (FPGA) was used in the control module as well as in the inverter.

The task sharing of the individual components:

- FPGA-based control module:
  - o Current control and commutation
  - o Acquisition of all position encoders
  - o Analog and digital I/Os

- FPGA-supported PC module:
  - o Position control of all three drives
  - o Drive interpolation
  - o Technological functions
  - o Monitoring
  - o External communication

All cyclic processes – from the fastest in microsecond cycle to the slower ones in the Nios II processor were implemented synchronously to the pulse-width modulation (PWM) clock. This is very easy to implement with the combination of FPGA and soft-core processor. The clock synchronous cycles avoid all unwanted beats and interferences.

A decisive factor was the time jitter when sampling the incremental encoders. This requires a hard synchronization of the inverter logic with the controller cycle. Using the configurable clock phased-lock loop (PLL) in the Altera FPGA with switchable clock input we got excellent results!

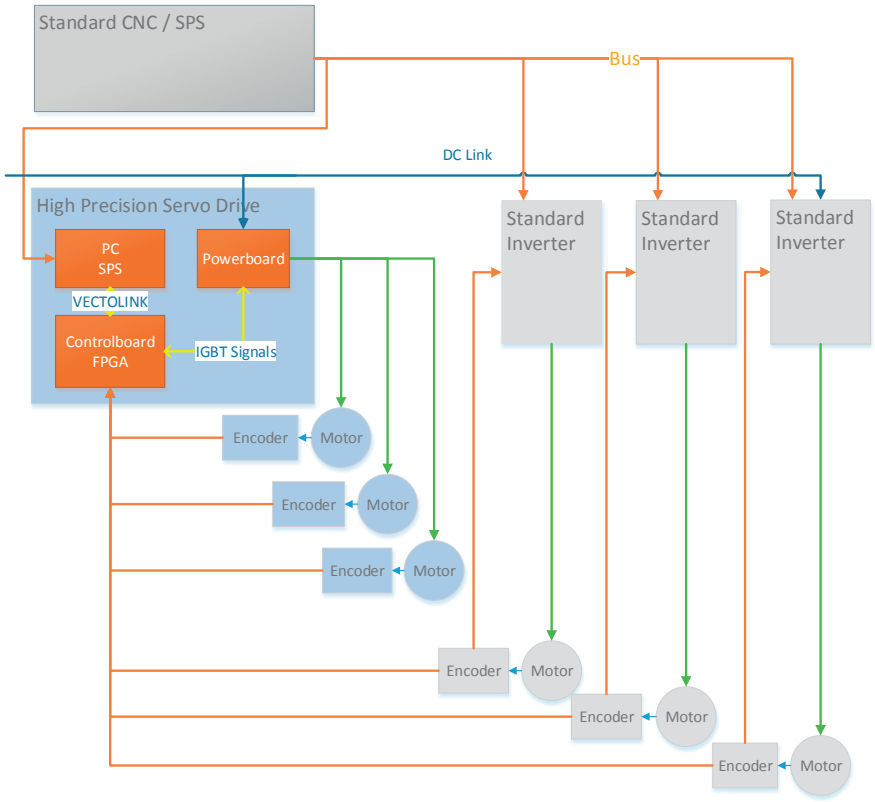

Diagram 1: Drive system structure

System requirements

A common situation in high-end production machines is that a small number of servo axes determine or limit the total machine performance regarding accuracy and dynamics. If this bottleneck is widened by using specialized solutions, then the total production machine can often achieve a considerably higher performance.

We used the following basic conditions for the development of the high precision multi-axis servo drive:

- Up to three servo drives are controlled with high accuracy and dynamics.
- This unit, containing up to three axes, is a subsystem of the total machine that often has considerably more servo drives.
- The subsystem to be developed has to fit into the interpolation cluster of all axes.
- Up to three additional position encoders can be handled. For example monitoring other movements in the machine.

In order to offer maximum flexibility for the different applications, an existing PC-based Linux real-time system was used for the controller.

This carries out position control and receives the encoder positions from the power electronics board. After calculating the closed loop, the nominal values for the motor current are transferred from the controller to the power electronics. The aim is to reduce the dead time to a minimum between reading the encoder position and output the new current nominal values. Only in this way the machine can perform with high accuracy and create the desired advantage compared to standard machines.

Read in position encoders

For speed calculation from the position signal of a position encoder the time-derivation of the position must be calculated within the control cycle. The shorter the cycle time, the more measuring errors in position and possibly jitter in sampling will affect the measuring value of the speed. This method has a negative impact on the accuracy of the control.

On a high resolution position encoder with sine/cosine signals, the analog signals are sampled with A/D converters. On the one hand, interference or component tolerances in the analog input stage can cause measuring errors, and these have a direct influence on the position. This causes a deviation in the actual position within constant limits, independent of the speed itself.

On the other hand, deviations in sampling time (jitter) also affect position. These errors are proportional to motor speed, having an effect especially at high speed.

High accuracy for the sample time of analog signals was a decisive factor in this project along with the specific drive system's extremely short cycle times and high speed range.

Abbreviations:

FPGA	:	Field Programmable Gate Array
DSP	:	Digital signal processor
LVDS	:	Low-voltage differential signaling
PLL	:	Phase-locked loop
PWM	:	Pulse-width modulation

ARADEX

Advantages of using FPGAs

Scalability

During development phase, FPGA technology has a special advantage: the scalability in size and speed with the same pin out of the chip itself.

A functional prototype was created within a very short time to verify the required performance. As the functionality within the FPGAs is determined via firmware, the component selection is very simple as no special hardware functions must be distinguished.

The scalability of the FPGAs means that logic cells and memory size can be adjusted at any time. So it makes sense to use larger FPGAs during the development phase and optimize to smaller and more economical FPGAs for series-production. It is also possible to retain resources to add functions or optimize them by product updates.

In comparison to a microcontroller or DSP, additional functions with an FPGA require more resources (logic cells) but normally not more computing time!

Hard parallel processes without any unwanted interaction

When using DSPs, the timing often changes if further functions are added or if they become more complicated, requiring longer cycle times. These often create unwanted “interactions” that have to be examined in detail at other parts in the closed-loop control. This can be avoided when using FPGAs with parallel processing. This shortens development times and makes developments safer and more reliable.

Real-time communication on both system boards: VECTOLINK

For communication between the FPGA in the PC module and the FPGA in the power electronics, a communication interface was developed to exchange cyclic data. The following characteristics were required:

- Low latency times for real-time data transmission.
- Synchronization of the cycle time of the power electronics with the real-time core of the PC module.

- Full duplex data transmission.
- In addition to the real-time data, further data must be transmitted for parameterization and diagnostic functions.

In order to achieve a low timing jitter while sampling incremental encoder signals, the cycle times of the position controller in the PC module and the other control circuits in the power electronics must be synchronized with one another. With high-resolution encoders with sine/cosine signals, analog signal frequencies of up to 300 kHz typically occur.

The resolution of the analog-to-digital converter (ADC) is typically 12 bits and, therefore, the theoretic quantization error corresponds to approx. 1/8000 of a sine period. In practice a resolution of 1/1000 up to 1/4000 sine period is typically achieved.

The jitter must be smaller than 1/4000 of a sine period so that any errors occurring due to the jitter in sampling time are not larger than the quantization error by the A/D converter. At a frequency of 300 kHz the jitter must therefore be better than 1 ns!

To get such a high synchronicity the position controller in the PC module and the power electronics must be clocked from the same clock source.

PLLs

The extremely flexible PLLs of the Altera FPGAs are ideal for such uses. The synchronous bus has one clock and two data lines (200 Mbit/s each) in each direction. The differential LVDS I/O ports on the Altera FPGA are used for this. Additional drivers or interface components are not required. Encoding of the bits on the bus is accomplished analog to fast Ethernet via 8B/10B encoding.

All data is transferred in one block. In order to guarantee data security, each data block is finished with a check sum. The overhead on the bus is thus limited to approximately 25%. A net data rate of approximately 300 Mbit/s is realized.

Because the clock signal is transmitted continuously on one line, the clock signal coming from the PCI board of the PC module can be used for the power electronics. The PLLs provide a function for switching the clock source at run-time.

As long as no clock signal is received from the PCI board a local quartz oscillator serves as a clock source. When the clock signal from the PCI board is stable, the PLL is switched to this source and both FPGAs operate absolutely synchronous. Therefore, also interpolation and position control in the PC module is synchronous to the PWM creation in the power electronics. The jitter is much better than 1 ns.

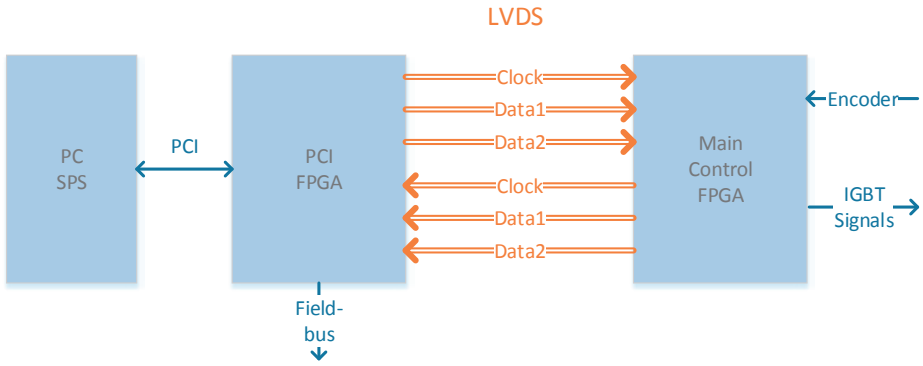

Diagram 2: Serial communications channel

The cycle time for position control and PWM frequency of the power stage is set to 62.5 µs (corresponding to 16 kHz). To read in all actual values, approximately 100 bytes have to be transmitted. To write all set values, approximately 16 bytes have to be transmitted. Assuming a data rate of approximately 300 MBit/s, the transmission for 100 bytes will last about 2.7 µs.

Because actual values are read in at the beginning of a control cycle and the calculated current nominal values are sent to the power electronics directly after calculations are finished, two transmissions within a control cycle are required. The total latency period caused by the data transmission is therefore approximately 4 µs.

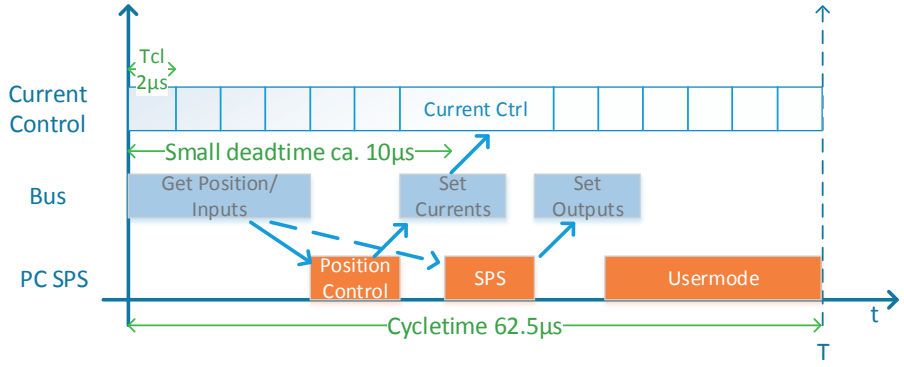

Diagram 3: Data transmission in real-time cycle

Current control

The current control was implemented completely in the FPGA logic cells. An internal soft-core processor (Nios II) implements the normalization and scaling of the control parameters. The controller attains signal transit times of approximately 200 ns. Therefore, it makes sense to use this calculation unit for several power stages together. Even if the data are processed one after the other, the total computing time is still less than 1 µs. This saves logic cells and at the same time achieves a much higher performance compared to a microcontroller or DSP.

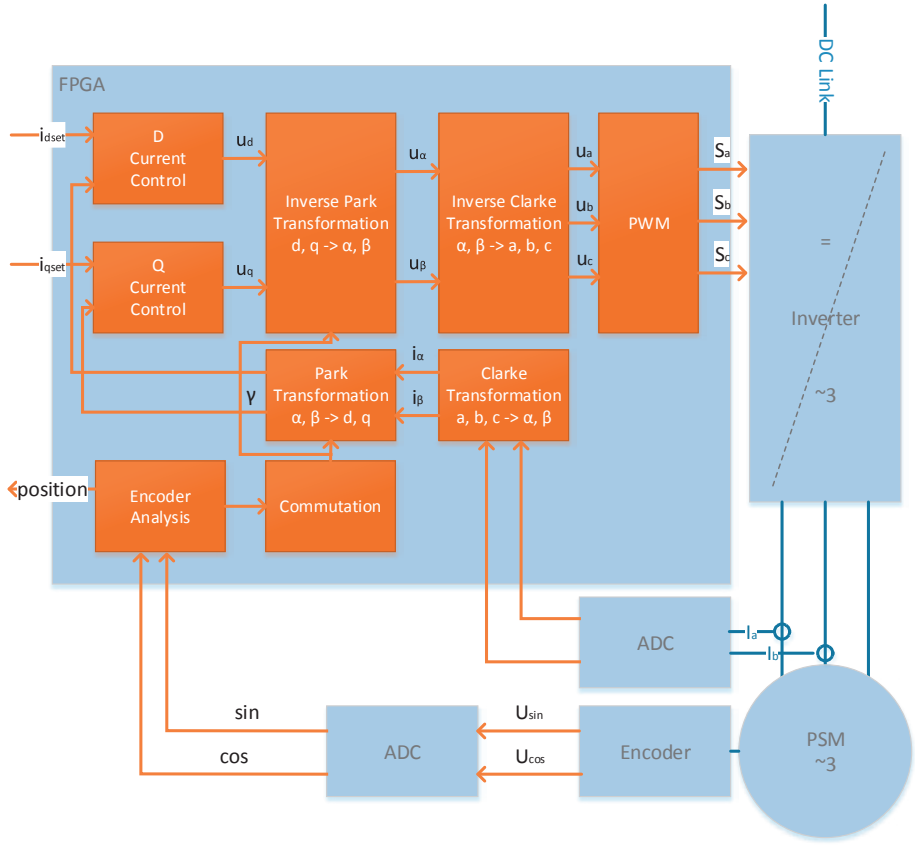

Diagram 4: Current controller block diagram

A combined Clarke/Park-Transformation for the transformation of the current actual values into the D/Q system is used which can be easily implemented in an FPGA.

Trigonometric calculations can be realized in extremely short times using lookup tables. Modern FPGAs (such as Altera Cyclone IV FPGAs, for example) have sufficiently large SRAM blocks.

For multiplying coefficients there are special hardware multipliers in the FPGA that can be used in parallel if required so that the complete transformation can be calculated within a few clock cycles.

sin(x) / cos(x)	:	25 ns
Clarke/Park-Transformation	:	70 ns
PI controller	:	50 ns
Total current controller	:	200 ns

Table 1: Calculation times for certain computing operations at an internal clock rate of 100 MHz

Commutation

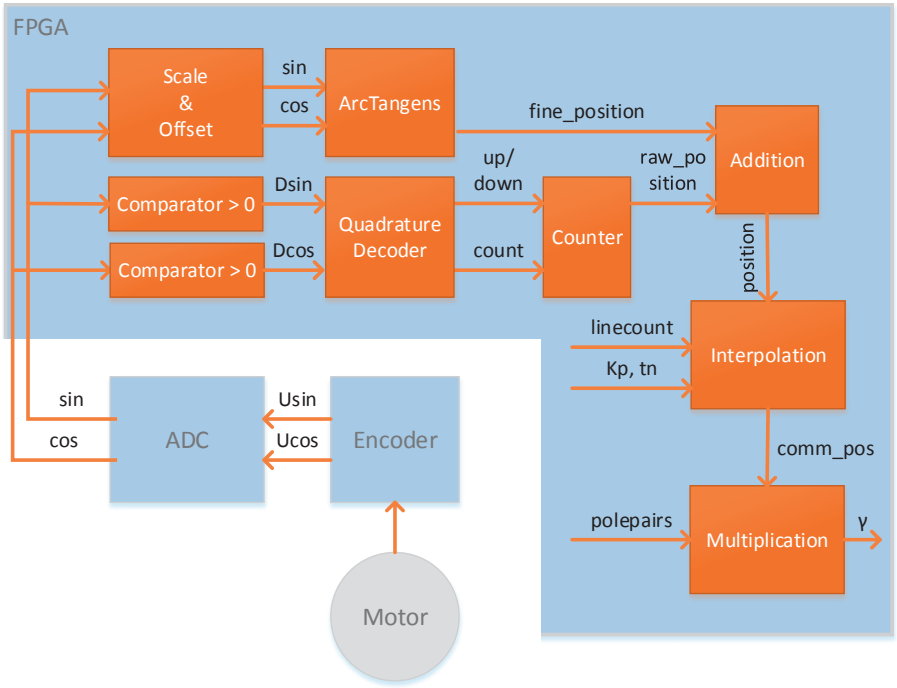

Diagram 5: Commutation with interpolation integrated in the FPGA

The creation of the commutation angle (γ) is realized by evaluating the sine signals of the encoder. This requires a huge flexibility as the encoder systems, depending on the technology used, has resolutions in the range from 1 to over 10,000 sine periods per revolution. The sine and cosine signals were read into the FPGA via an A/D converter. Both signals are digitized via a comparator (sign recognition) and evaluated with a quadrature decoder and counter. In this manner, every single zero crossing of the encoder signal can be recorded.

The position between two zero crossings can be determined using arc tangent calculation from the sine and cosine signal. Combining these values with the counter for the zero crossings yields the rotor position angle with high resolution. For further position evaluation the rotor position angle must be scaled to a motor revolution.

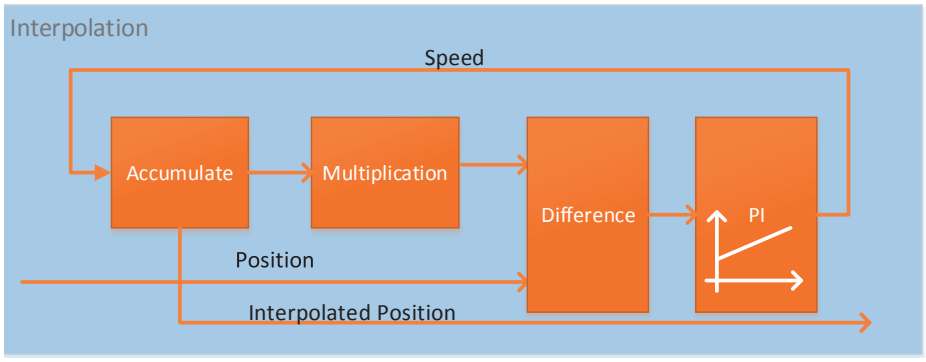

Diagram 6: Interpolation: Accumulator and PI Control Loop

For the Clarke/Park-Transformation, the commutation angle must be provided with a determined scale. A value of 2n, for example 1024 steps, is used for one revolution (360°).

Since the line count of an incremental encoder can have any value, an interpolation is used to adjust the encoder line count to the scale of the commutation angle.

This consists mostly of a counter and a PI controller. The counter output corresponds to the commutation angle. This value is compared to the encoder position, and the difference is fed to a PI controller. The PI controller output determines the counting rate for the counter.

This means that the encoder line count can be adjusted to the scaling of the commutation angle. At the same time the encoder resolution can be increased so that measuring systems, that only deliver one sine/cosine period per revolution (e.g. resolvers), can be used.

Furthermore, the interpolation can be used to eliminate EMC interference, making this an additional positive characteristic for this process.

Extremely short latency times

The result of using an FPGA for interpolation is a very low latency time. This is especially important at high speeds of the motors. DSP or microcontroller solutions often have latency times in the range of several 10 µs, which causes a noticeable phase displacement with high-speed motors. A motor with 1000 Hz sine frequency has a sine period of 1000 µs, a delay of 10 µs therefore causes a phase displacement of 3.6°.

If the phase displacement is too large the efficiency of the drive system goes down and the control performance is negatively influenced. The interpolation in the FPGA is done within a cycle time of 100ns, the latency time is determined mainly by the conversion time in the A/D converter. This is approximately 500 ns with the ADC used here. The total latency time is therefore considerably less than 1 µs.

Using FPGAs can considerably increase the accuracy of the commutation angle when compared to calculating with a DSP or microcontroller.

FPGA resources

One device of the Cyclone IV E FPGA series with approximately 30,000 Logic Elements (LEs) contains approximately 64 Kbyte SRAM and 66 multipliers with a width of 18x18 bit.

Table 2 lists the resources that are required for some selected functions:

	LEs	SRAM	Multiplier
Nios II processor	6000	150KB	2
Clarke/Park Transformation	390	2KB	1
Current controller	2100	12KB	9
VECTOLINK-interface	1000	512B	2
Incremental encoder evaluation	1500	10KB	14
Interpolation of commutation angle	600	0KB	12

Table 2: Required logic cells for selected function blocks in the FPGA (for 3 axes)

Conclusion

The high requirements for the drive system were managed with the help of FPGAs.

One FPGA (Altera Cyclone IV FPGA) was used in the PC module and another one in the power electronics. So performance capability of these components was fully utilized.

The capability to execute many processes in parallel has enormous advantages especially for complex systems. For example, several axes can be controlled in parallel without increasing the cycle time.

Additionally, the use of FPGAs enabled a much higher synchronicity of both components (controller and power-electronics). This was a highly important requirement of this project in order to achieve a precise position measurement and control at high speeds.

A further advantage is that additional functions can be implemented even in the prototype phase if the hardware is more or less at the series production status without influencing the performance of other functions.