FPGA-Technologie im industriellen Umfeld

Embedded Computing Conference, 30.8.2011, ZHAW Winterthur

Marc Oberholzer
Enclustra GmbH
FPGA Design Center

Content

- Enclustra GmbH
  - Company Profile
- The Case for FPGAs
  - Unique Selling Points
  - FPGA vs. ASIC
  - FPGA vs. DSP
  - FPGA vs. uC
- Real-World FPGA Applications
  - Software Defined Radio
  - Linux on FPGA
- Example Project
  - Motion Control
- Conclusions
  - Field Update
  - „Featuritis“
  - Outsourcing
  - How to Stand Out

Content

- **Enclustra GmbH**
  - Company Profile
- The Case for FPGAs
  - Unique Selling Points
  - FPGA vs. ASIC
  - FPGA vs. DSP
  - FPGA vs. uC
- Real-World FPGA Applications
  - Software Defined Radio
  - Linux on FPGA
- Example Project
  - Motion Control
- Conclusions
  - Field Update
  - „Featuritis“
  - Outsourcing
  - How to Stand Out

Enclustra GmbH – Company Profile

- Quick Facts
  - Founded in 2004
  - Located at Technopark Zurich
  - Currently 7 employees
  - Vendor-Independent
- FPGA Design Center
  - FPGA-Related Design Services
  - Firmware (VHDL/Verilog)
  - Hardware (incl. analog and digital interfaces)
  - Embedded Software (for FPGA soft processors)
- FPGA Solution Center
  - FPGA Modules
    - Mars, Mercury and Saturn
  - IP Cores
    - TFT Display Controller
    - Universal Drive Controller
    - Etc.

Content

- Enclustra GmbH
  - Company Profile
- **The Case for FPGAs**
  - Unique Selling Points
  - FPGA vs. ASIC
  - FPGA vs. DSP
  - FPGA vs. uC
- Real-World FPGA Applications
  - Software Defined Radio
  - Linux on FPGA
- Example Project
  - Motion Control
- Conclusions
  - Field Update
  - „Featuritis“
  - Outsourcing
  - How to Stand Out

The Case for FPGAs – Unique Selling Points (1)

- Real parallel processing
  - Vast parallel processing power for DSP applications
  - No conflicts in accessing shared resources (because there aren't any...)
- Hard real-time capabilities
  - No operating system, no scheduler, no IRQ latency, only pure hardware
  - Nanosecond time resolution (e.g. 200 MHz FPGA clock frequency -> 5 ns cycle time)
- High integration and customization potential
  - Single-chip systems with standard and custom parts

The Case for FPGAs – Unique Selling Points (2)

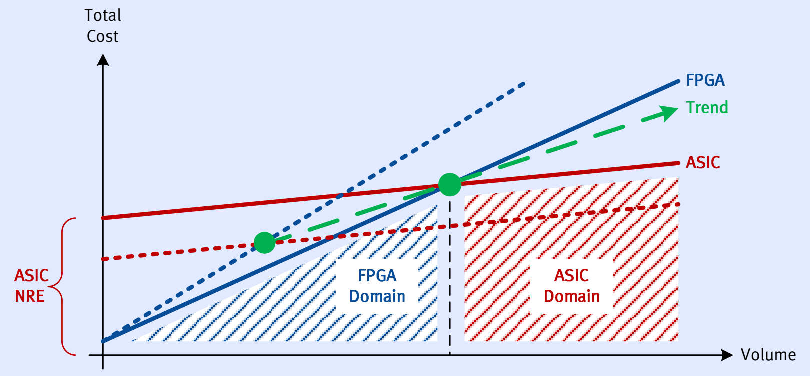
- Reconfiguration / remote update capability
  - Configuration can be changed over and over again
    - Allows early system tests on hardware instead of time-consuming simulations
  - Deployed systems can be updated in the field, e.g. over the internet
  - Therefore often used as configurable external I/O
- Long-term availability
  - Devices are usually available for > 10 years
  - System functionality is defined by HDL code rather than by hardware schematics
  - HDL code is easily ported to a new FPGA generation (no change to embedded processor code)

The Case for FPGAs – FPGA vs. ASIC

- FPGAs can't beat ASICs when it comes to
  - Low power
  - Ultra small form factor
  - Ultra high design security
  - Ultra high volume
- ASICs need volume to overcome the NRE penalty
  - NRE increase with each process shrink
  - FPGA logic gets cheaper with each process shrink
  - The break-even is moving towards higher volumes with each process shrink
- Remote update and faster time to market become more and more important
  - FPGAs gain ground in the ASIC domain
- FPGAs are often used for ASIC prototyping

Parameter	FPGA	ASIC
Clock frequency		✓
Power consumption		✓
Form factor		✓
Design security		✓
Reconfiguration	✓	
Redesign risk (weighted)	✓	
NRE	✓	
Time to market	✓	

Total Cost FPGA vs. ASIC

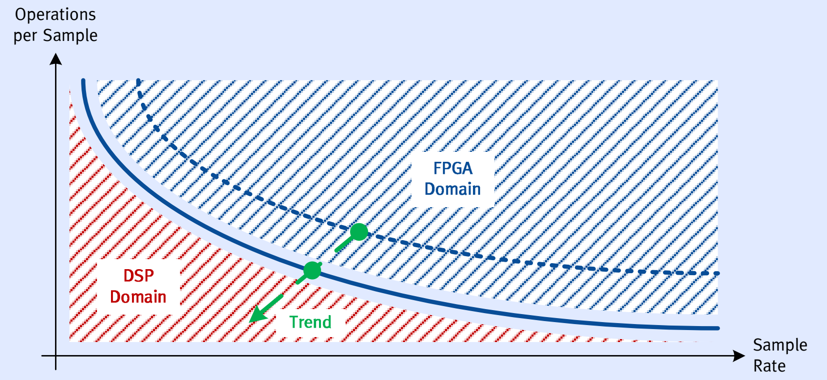


The Case for FPGAs – FPGA vs. DSP

- DSPs are widely used in low-cost, low-power and low- to mid- performance systems
- DSPs suffer from their serial instruction stream when it comes to more complex systems running at high sample rates
- FPGAs can provide a performance boost of 10..1000 compared to DSPs for such applications (e.g. software defined radio).
- FPGAs even excel when compared in MAC/$ and MAC/W.
- Hard-macro CPU cores in the FPGAs take over traditional DSP tasks (e.g. complex protocol stacks), enabling single-chip high-performance signal processing systems

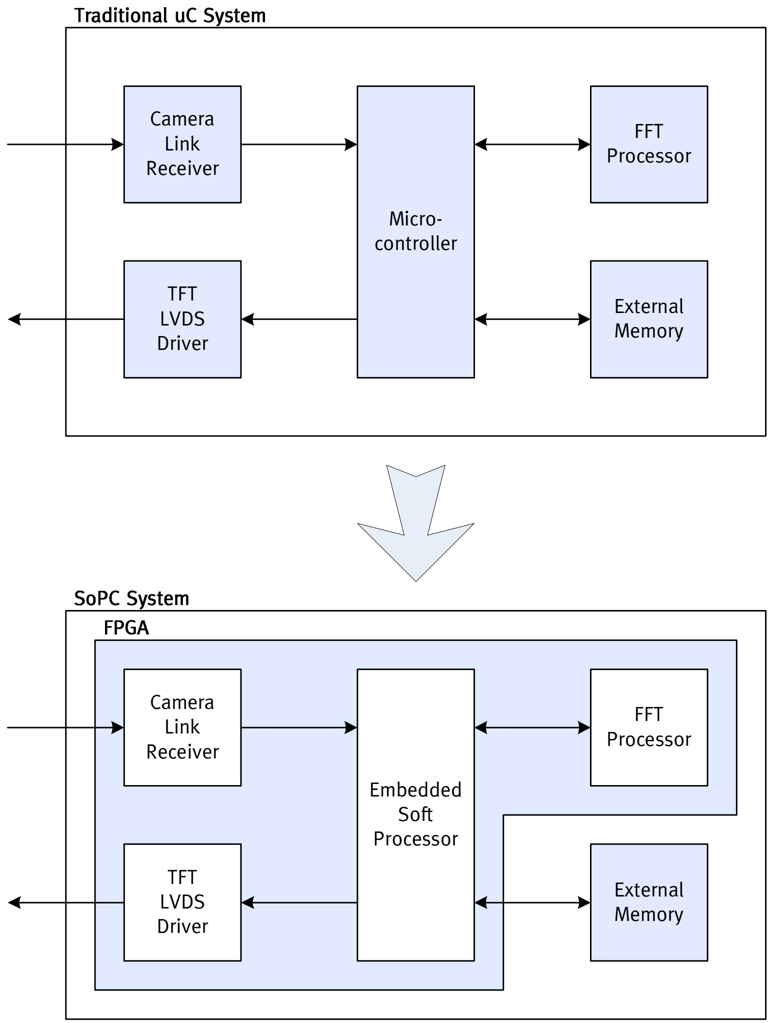
Parameter	FPGA	DSP
System performance	✓	
Multi-channel architecture	✓	
Many operations per sample	✓	
Many conditional operations		✓
Floating point		✓
Absolute power consumption		✓

Performance FPGA vs. DSP



The Case for FPGAs – FPGA vs. uC

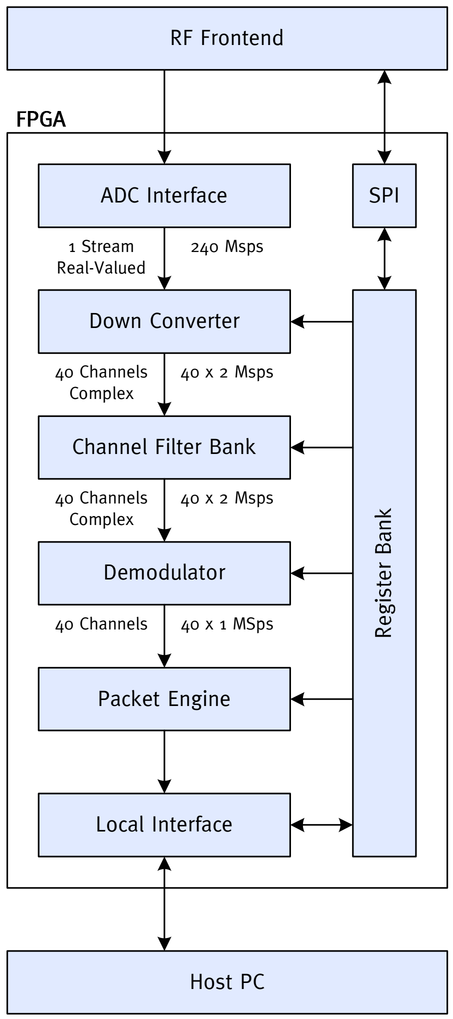
- „Microcontrollers are cheap and energy-efficient, FPGAs are expensive and power-consuming“
  - If a microcontroller can do it, there is usually no need for an FPGA
- SoPC designs with FPGA-internal soft processors are beneficial if
  - The system requires an FPGA anyway
  - Many external ICs would be needed along with a microcontroller
  - PCB space is a major concern
  - High design flexibility is required
  - Long-term availability is a major concern
    - Reduced part count
    - BSP defined through VHDL-code



Content

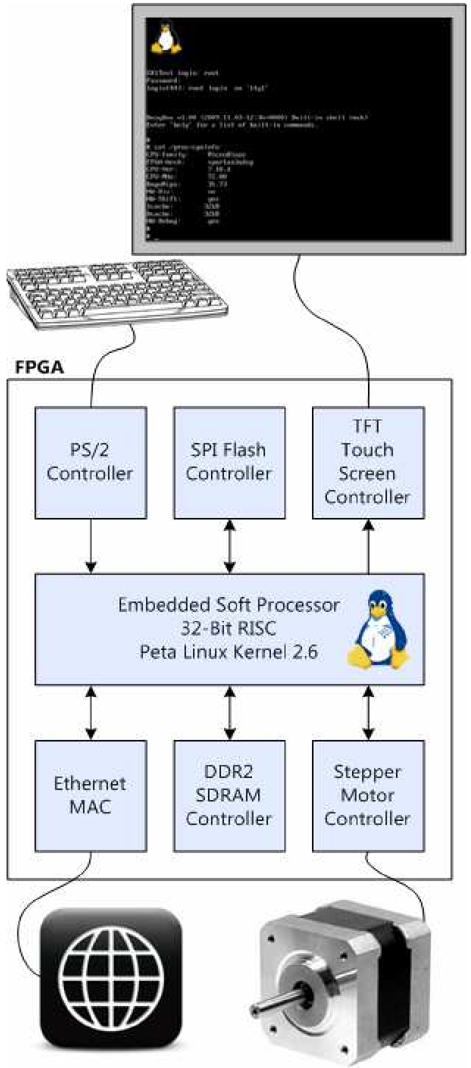
Real-World Applications – Software Defined Radio

- Software defined radio
  - Most of the signal processing of a RF receiver/transmitter is done in „software“
- Real-world application
  - 2.4 GHz RF receiver
  - 240 Msps sampling rate
  - Down conversion to 40 channels at 2 Msps each
  - Parallel baseband-processing of all 40 channels with a time division multiplexed datapath architecture
    - Channel filters
    - Demodulators (FSK, PSK)
  - Spartan-3A DSP low-cost FPGA
    - 126 multipliers running at 240 MHz clock frequency
    - Ported to Spartan-6 LX
    - TDM / Floorplanning



Real-World Applications – Linux on FPGA

- FPGA SoPC demonstrator
  - TFT display with touch function
  - PS/2 keyboard
  - Gigabit Ethernet (TCP/IP)
  - Stepper motor controller
  - 32-bit RISC CPU running Linux 2.6
  - GUI based on Nano-X
  - Everything in a single low-cost FPGA (Xilinx Spartan-3A DSP on a Enclustra Saturn SX1 FPGA module)
- Linux provides user I/O, networking and a well-known application development platform
- FPGA logic provides custom functions



Content

Example Project – Motion Control Specification

- Technical requirements:
  - Motion control module
  - Up to 4 DC or 2 stepper motors
  - Up to 2 BLDC motors in a later stage
  - CAN interface
  - Trajectory planner/integrator
  - All calculations in SI units
  - 1..5 KHz position/velocity control
  - 10..100 KHz current control
  - 4 integrated FET H-bridges
  - Credit-card size
- General information:
  - Motion control platform for next-generation products
  - High-volume (> 10'000 units/year)
  - Must comply with various engineering standards
- Commercial requirements:
  - Manufacturing costs < X $
  - Available no later than day Y
  - Engineering costs are secondary

Example Project – Motion Control
Project Setup

- General project setup:
  - The customer is responsible for hardware design, production and embedded software
  - Enclustra is responsible for FPGA firmware and FPGA-related system design issues
- Team setup at Enclustra:
  - 1 project manager
  - 1 FPGA firmware engineer
  - 1 hardware consultant
- Team setup at the customer:
  - 1 project manager
  - 2 embedded software engineers
  - 2 hardware engineers

  and

  - The strategic procurement department

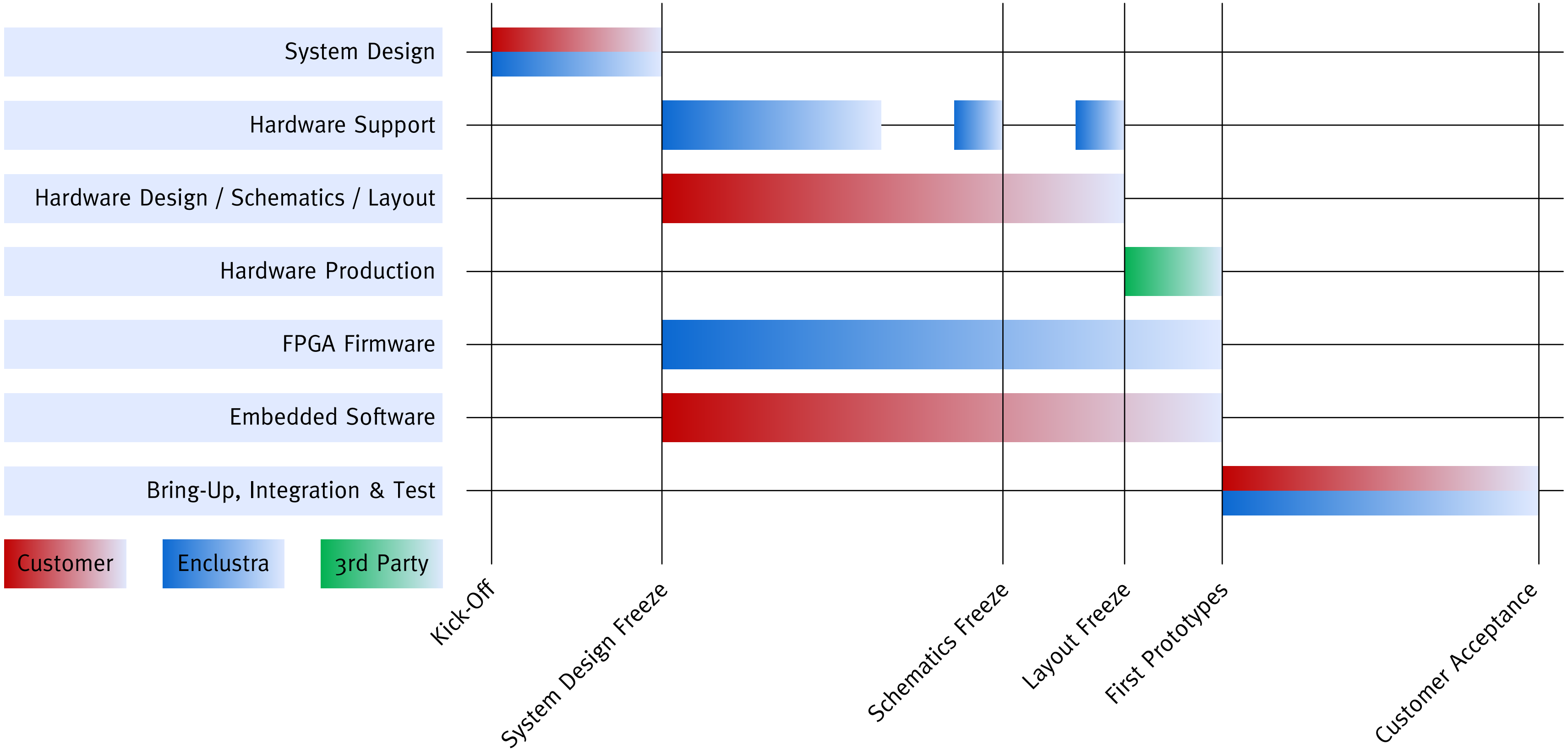
  and

  - The upper management

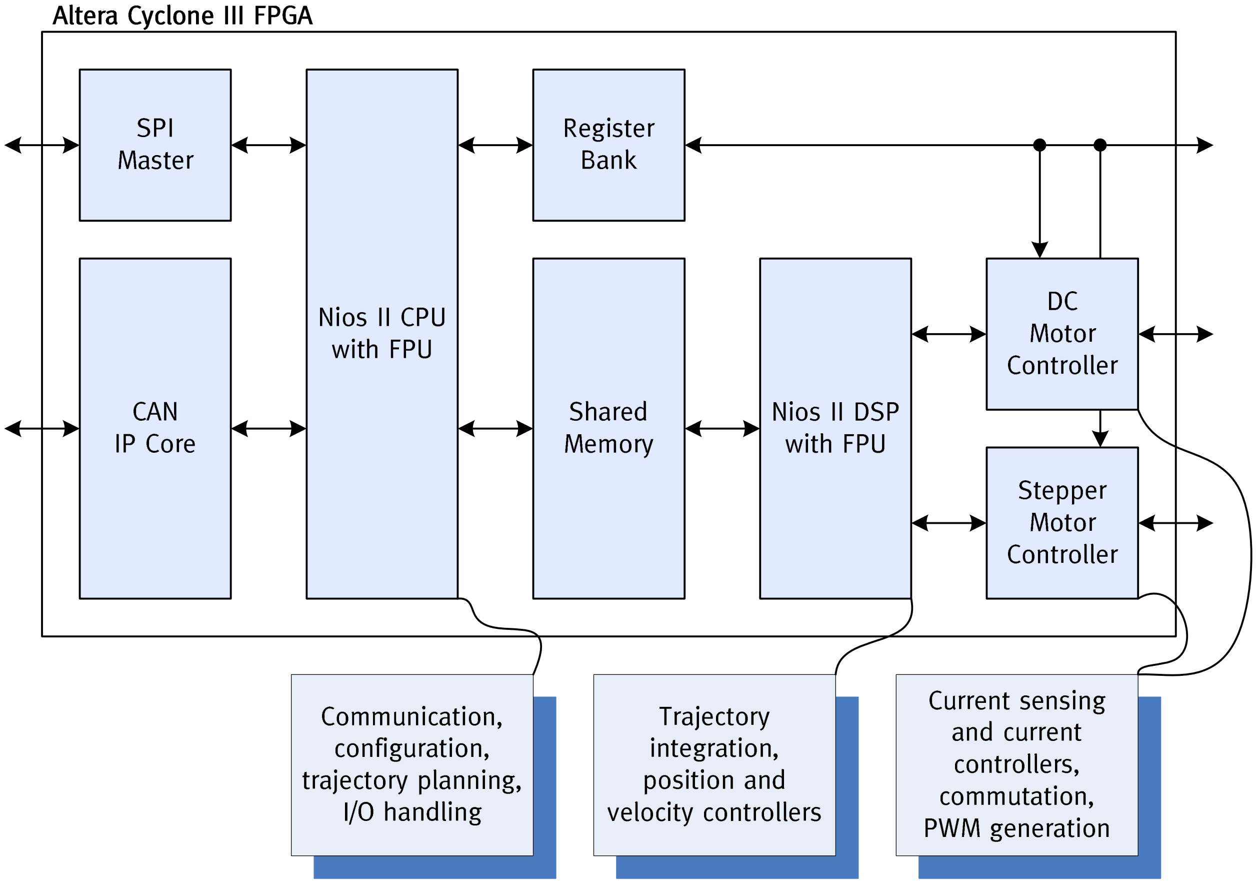
  and

  - Many potential users of the motion control module

Example Project – Motion Control Project Schedule (Basic Functions)

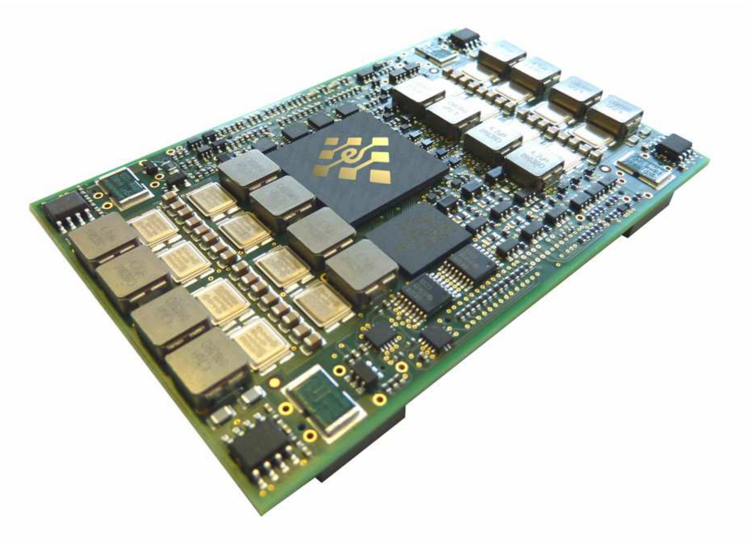


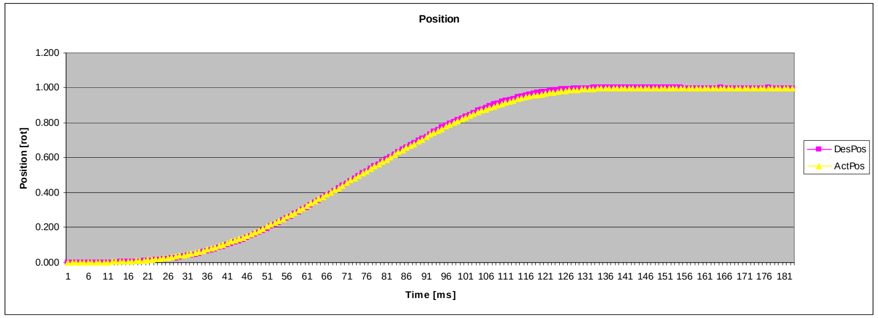
Example Project – Motion Control System Design (1)

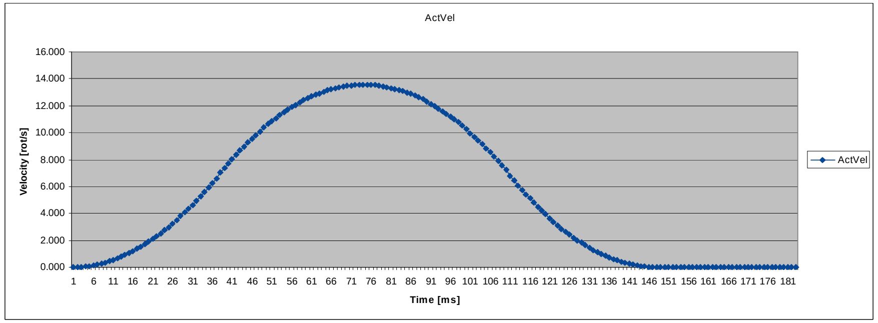


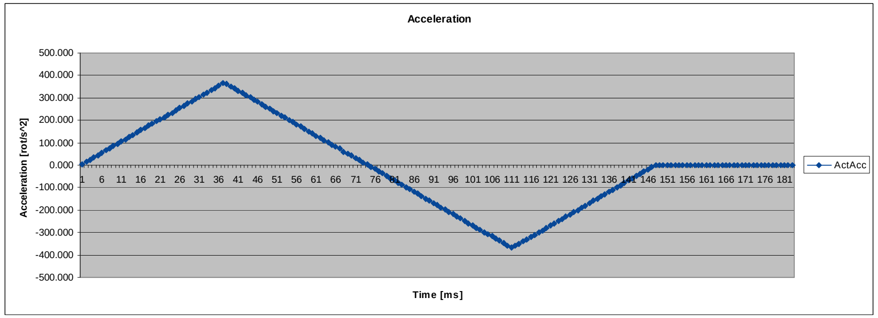
Example Project – Motion Control
First Prototypes!

- Bring-Up
  - Power, clocks, FPGA configuration
  - Nios II booting and JTAG communication
- First tests on hardware
  - The first logged move!







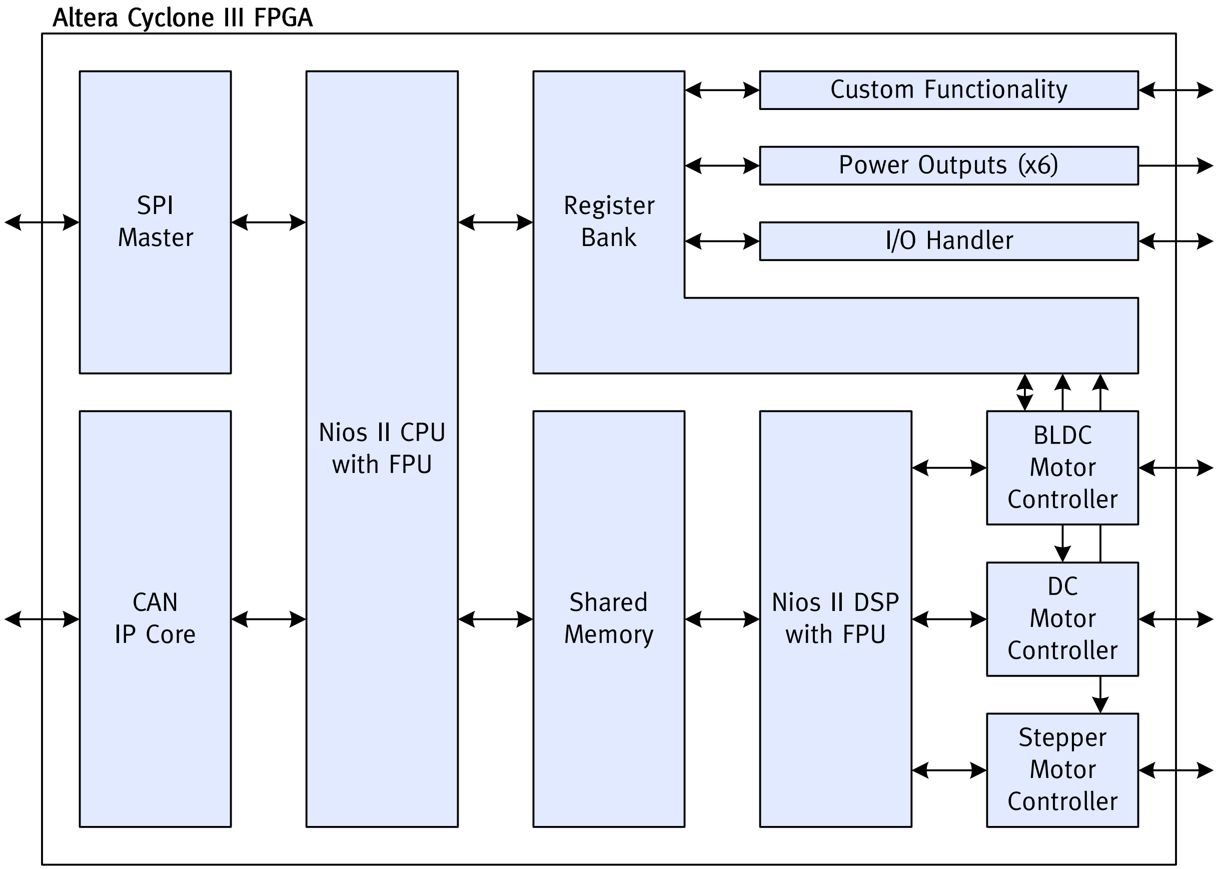


Example Project – Motion Control Next Steps

- In-System testing at the customer's site
  - Bugfixes
  - Small improvements
  - First ideas for new features
- First release to internal users
  - More bugfixes
  - More small improvements
  - More ideas for new features
- Customer acceptance for basic functionality on schedule

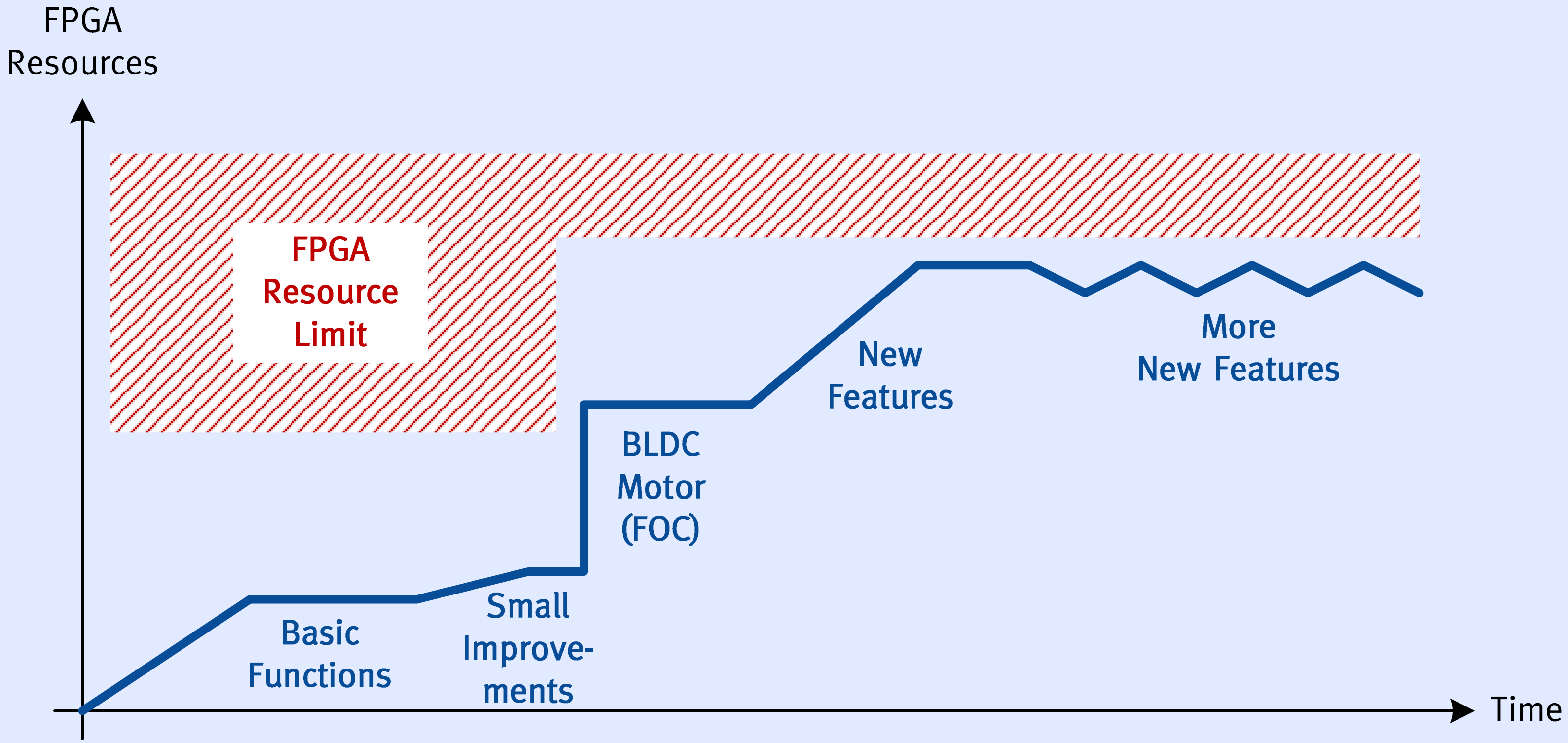
- New feature wishlist
  - BLDC motor: Field oriented control (FOC) instead of block commutation
    - BLDC motor behaves like a DC motor
    - Resource-consuming
  - Versatile I/O handler with interrupt support
    - Big muxes -> resource consuming
  - Power outputs with custom waveforms generated in FPGA logic
    - 6 times -> resource-consuming
  - Additional custom functionality
  - Much more configurable parameters
    - Growing register bank

Example Project – Motion Control System Design (2)



Example Project – Motion Control
FPGA Resources over Time

FPGA Resources over Time



Content

- Enclustra GmbH
  - Company Profile
- The Case for FPGAs
  - Unique Selling Points
  - FPGA vs. ASIC
  - FPGA vs. DSP
  - FPGA vs. uC
- Real-World FPGA Applications
  - Software Defined Radio
  - Linux on FPGA
- Example Project
  - Motion Control
- **Conclusions**
  - Field Update
  - „Featuritis“
  - Outsourcing
  - How to Stand Out

Conclusions – Field Update

- **FPGAs allow fast market entry thanks to their field update capability**
- This often leads to the fallacy that FPGA development does not require thorough verification („we can fix an error after it occurs“)
  - This might partly be true for non security relevant applications running in static ambient conditions
  - FPGA development actually IS done like this a lot more than one might think
  - This often wrongs FPGA technology in the user's minds, because there WILL be errors in this case
- The development of a reliable FPGA system does not get by without thorough verification!
  - Verification may include behavioral simulation and tests running on the hardware

Conclusions – „Featuritis"

- **FPGA-based systems allow step-by-step introduction of new features**
- FPGA projects require a thorough change management
  - Request, classification, design, approval, implementation, verification, release
- FPGA projects require a strict release management
  - Define specific feature sets for planned releases and stick to it
  - Build number, build date and time, release number, accurate release history
- Resource usage and power consumption must always be monitored
  - Device migration over different densities (assembly option) is possible, but complicates the initial hardware design
  - Power consumption is highly dependent on the FPGA design (resource usage, clock frequencies, etc.) and the system operating conditions (data toggle rates, etc.)

Conclusions – Outsourcing

- **Make or buy – the case for outsourcing FPGA development**
- Successful and efficient FPGA design requires in-depth knowledge of
  - Basic digital and analog circuit design, chip design, VLSI
  - HDL (VHDL/Verilog/etc.), FPGA architecture and tools
  - High-speed hardware design
  - Deployed algorithms, I/O standards, protocols, etc.
- Many companies have extensive knowledge in their application area, but do not have the required expertise for successfully employing FPGA technology
- Building up FPGA know-how is a lengthy and expensive process
- Collaboration between application specialists and FPGA technology experts shows great promise for successful product development

Conclusions – How to Stand Out

- **How can an FPGA engineering company stand out from the crowd?**
- Focus on
  - FPGA technology (don‘t be a „general merchandise store“)
  - Key application domains (e.g. DSP, SoPC, etc.)
- Provide solutions, not only engineering resources
  - FPGA modules as HW platform
  - IP cores for complex building blocks
  - Custom design for custom functionality
  - System integration
- Not only make the customer happy, but also make him successful
  - What the customer initially wants is most often not what he really needs

Questions?

Marc Oberholzer

Enclustra GmbH

oberholzer@enclustra.com

Fon +41 43 343 39 47

Slides in PDF format:
http://www.enclustra.com/de/company/publications/