FPGA Technology and Industry Experience

Guest Lecture in the context of the VLSI III course at D-ITET, ETH Zurich

May 26th 2010

Martin Heimlicher, FPGA Solution Center, Enclustra GmbH
Marc Oberholzer, FPGA Design Center, Enclustra GmbH
Content

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
Quick Facts
- Founded in 2004
- Located at Technopark Zurich
- Currently 6 FPGA Engineers
- 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.
Mars MX1/MX2 FPGA Module

- Low-cost, low-power Spartan-6 FPGA
- SO-DIMM form factor
- Single supply voltage
- 16 MB quad SPI Flash
- 256 MB DDR2 SDRAM
- Dual Fast Ethernet (MX1 only)
- Real time clock (MX1 only)
- Gigabit Ethernet (MX2 only)
- Dual multi-gigabit transceivers (MX2 only)
- PCIe endpoint (MX2 only)
Mercury CA1 FPGA Module

- Low-cost Cyclone IV FPGA
- Single supply voltage
- 16 MB SPI Flash
- 256 MB DDR2 SDRAM
- Gigabit Ethernet PHY
- USB 2.0 High-Speed interface
- High-power 8A core power supply
Content

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
- **Field Programmable Gate Array**
- Regular array of configurable logic blocks
- Embedded RAM blocks
- DSP blocks
- Configurable I/O blocks
- Dedicated clock management blocks
- Connected through configurable routing resources
  - Global routing (long, fast, few lines)
  - Local routing (short, abundant)
  - Switch boxes
- Configuration data stored in distributed SRAM cells
- LUTs are used for implementing logic functions
  - 1-4 input functions in a single LUT
  - LUT trees or carry chain architectures for more complex logic functions
  - LUTs are in fact small SRAMs
- FFs are used for storing data and for pipelining
  - Increasing logic levels result in increased propagation time
  - Routing delay is dominant over LUT delay
  - Pipelining helps to break long paths
  - FFs are expensive in ASICs but are cheap in FPGAs
  - Up to 500 MHz for real-world designs
- ~1500 .. 950’000 (1.5 Mio) per FPGA
DSP blocks are used to implement fixed-point arithmetic operations
- Typically 18 x 18 bit multiplier
- 48 + 48 bit adder/accumulator
- Pre-adder for symmetric FIR filters
- Dynamic configuration via OPMODE
- Highly pipelined (configurable)
- Up to 600 MHz clock frequency
- Support for carry and adder chains
- ~4 .. 2000 (3600) per FPGA
- Up to 1200 (2160) GMAC/s per FPGA!!!
Memory Blocks are used to implement
- Simple data storage
- Shared memory
- Synchronous/asynchronous FIFOs
- Configurable large delays
- ROM (content is part of FPGA bitstream)
- Data tap / coefficient storage for FIR filters
- Program/data memory for embedded soft processors

Widely configurable in
- Width/depth aspect ratio
- Size (cascadeable)
- Read behaviour (registered/combinatorial)

Granularities from 512 bit to 36 Kbit
- 72 kbit .. 38 Mbit per FPGA
- I/O blocks are used to interface to the outside world
- Highly configurable in
  - Direction (input, output, bidirectional)
  - Data rate (SDR, DDR, SERDES)
  - I/O standard (single-ended, differential, referenced, etc.)
  - I/O voltage (1.2 V .. 3.3 V for single-ended standards)
- FFs in the I/O block guarantee a stable I/O timing
- ~100 .. 1200 per FPGA
Hard macro blocks provide complex functions like:

- Embedded processor cores (PowerPC, ARM, etc.)
- External memory controllers (DDR, DDR2, DDR3, QDR, QDR II, etc.)
- Triple-speed Ethernet MACs
- Multi-gigabit serial transceivers
- PCI-Express endpoints
- Etc.

Even low-cost FPGAs provide some hard macro blocks today:

- No hard macro CPU in low-cost FPGAs so far
Embedded processor core implemented with FPGA logic, DSP and memory blocks

- Highly configurable
  - Starting with an ultra-simple 8-bit uC core
  - Ending with 32-bit RISC CPUs including FPU and MMU, capable of running a desktop Linux distribution

- Enables SoPCs with low-cost FPGAs
  - System on a Programmable Chip
  - Single-chip solution, uC and legacy peripheral ICs are replaced by FPGA logic resources
Architecture design is vital for:
- System performance
- Resource usage
- Timing closure
- Power consumption

HDL: **Hardware** Description Language

Only strict synchronous design practices lead to reliable FPGA systems!

FPGAs allow on-chip debugging with fast iteration times (typically less than one hour)
**FPGA Basics – FPGA Design Flow Design Entry**

- **Code Generation vs. HDL Entry**
  - **Code Generation**
    - Allows fast migration of already available code (e.g. written in C or MATLAB) to an FPGA target
    - Not very resource-efficient, lacks performance
    - Tools are very expensive and often complicated to use and maintain
  - **HDL Entry**
    - Allows very resource-efficient and performant implementations
    - Same functionality fits into a smaller, slower and therefore cheaper FPGA
    - Longer development time
    - Simple tools, available for free (text editor...)

- **IP Cores and HDL Libraries**
  - **IP Cores**
    - Device-specific implementation (netlist)
    - Provide complex functions
      - MPEG encoder
      - CAN interface
    - Many vendors
  - **HDL Libraries**
    - Device-independent implementation (HDL)
    - Provide mostly basic functions
      - Asynchronous FIFO
      - SPI Master
    - Company-internal
### Floorplanning, mapping, place and route

- **Floorplanning**
  - Necessary for high-speed designs on large FPGAs

- **Map**
  - Mapping of the synthesized netlist to available resource elements of the target device

- **Place**
  - Placing of the mapped elements to the available physical resources of the device

- **Route**
  - Implementation of the logical connections with physical routing resources
### On-Chip Debugging

Today’s FPGA tools allow to easily implement logic analyzer functionality in FPGA logic:

- **Real-time capturing of FPGA-internal signals**
- **Offline readout of captured signals via JTAG**
- **Logic analyzer GUI on host computer**
- **Almost unlimited trigger-conditions, configurable on the fly**
- **Possibility to control FPGA-internal signals from the host PC**
- **Very limited memory depth**
- **FPGA has to be re-implemented in order to change the logic analyzer structure**
FPGA Basics – Market Overview Revenue

- Roughly 1.3% of total semiconductor market revenue
- PLD market revenue is believed to grow by 35% from 2009 to 2013
- FPGAs make up about 75% of the PLD market revenue
FPGA Basics – Market Overview

Vendors

- Xilinx and Altera share the high-performance FPGA market as well as the major part of the low-cost FPGA market.
- Lattice Semi is a traditional player in the low-cost FPGA and CPLD market.
- Actel provides Flash-based, low-power, mixed-signal and space-qualified FPGAs.

FPGA market by Vendors

- 51% Xilinx
- 36% Altera
- 6% Lattice Semi
- 6% Actel
- 1% Others
- Japan actually belongs to APAC
- Europe is the smallest market
- The Americas are not the biggest market
- Industrial applications are dominant in Switzerland
- There is also some significant activity in the communications sector in Switzerland
Content

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
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 -> 4 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
- 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>FPGA</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>System performance</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Multi-channel architecture</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Many operations per sample</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Many conditional operations</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Floating point</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Absolute power consumption</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

Performance FPGA vs. DSP

Operations per Sample vs. Sample Rate
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 a FPGA
- SoPC designs with FPGA-internal soft processors are beneficial if
  - The system requires a 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

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
    - Skills
    - How to Stand Out
- Outlook & Trends
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
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 an Enclustra Saturn SX1 FPGA module)

- Linux provides user I/O, networking and a well-known application development platform
- FPGA logic provides custom functions
# Content

<table>
<thead>
<tr>
<th>Enclustra GmbH</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Company Profile</td>
</tr>
<tr>
<td>FPGA Basics</td>
</tr>
<tr>
<td>- Programmable Logic History</td>
</tr>
<tr>
<td>- FPGA Architecture</td>
</tr>
<tr>
<td>- FPGA Design Flow</td>
</tr>
<tr>
<td>- Market Overview</td>
</tr>
<tr>
<td>The Case for FPGAs</td>
</tr>
<tr>
<td>- Unique Selling Points</td>
</tr>
<tr>
<td>- FPGA vs. ASIC</td>
</tr>
<tr>
<td>- FPGA vs. DSP</td>
</tr>
<tr>
<td>- FPGA vs. uC</td>
</tr>
<tr>
<td>Real-World FPGA Applications</td>
</tr>
<tr>
<td>- Software Defined Radio</td>
</tr>
<tr>
<td>- Linux on FPGA</td>
</tr>
<tr>
<td><strong>Example Project</strong></td>
</tr>
<tr>
<td>- Motion Control</td>
</tr>
<tr>
<td>Conclusions</td>
</tr>
<tr>
<td>- Field Update</td>
</tr>
<tr>
<td>- „Featuritis“</td>
</tr>
<tr>
<td>- Outsourcing</td>
</tr>
<tr>
<td>- Skills</td>
</tr>
<tr>
<td>- How to Stand Out</td>
</tr>
<tr>
<td>Outlook &amp; Trends</td>
</tr>
</tbody>
</table>

---

**Enclustra GmbH**

- Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
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
  - The strategic procurement department
  - The upper management
  - Many potential users of the motion control module
Example Project – Motion Control Project Schedule (Basic Functions)

- System Design
- Hardware Support
- Hardware Design / Schematics / Layout
- Hardware Production
- FPGA Firmware
- Embedded Software
- Bring-Up, Integration & Test

Customer  Enclustra  3rd Party

Kick-Off  System Design Freeze  Schematics Freeze  Layout Freeze  First Prototypes  Customer Acceptance
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)

![Diagram of Altera Cyclone III FPGA](image)

- SPI Master
- Nios II CPU with FPU
- Register Bank
- Shared Memory
- Nios II DSP with FPU
- CAN IP Core
- BLDC Motor Controller
- DC Motor Controller
- Stepper Motor Controller
- Custom Functionality
- Power Outputs (x6)
- I/O Handler

Enclustra GmbH

- 40 -

15.12.2010
Example Project – Motion Control
FPGA Resources over Time

FPGA Resources over Time

- FPGA Resources
- Time
- Basic Functions
- Small Improvements
- BLDC Motor (FOC)
- New Features
- More New Features
- FPGA Resource Limit
Content

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
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 be partly 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
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.)
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
FPGA jobs in engineering services - what skills do we expect?

- Technical skills
  - Basic digital and analog circuit design, chip design, VLSI (very important)
  - HDL (VHDL, Verilog, etc.)
  - Basic understanding of FPGA architecture and tools
  - Basic understanding of DSP and SoPC

- Soft skills
  - Keen perception also for complex structures
  - Good communication with customers and colleagues
  - Ability to work in a team
  - Flexibility for changing tasks quickly
Conclusions – How to Stand Out

- How can a 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
Content

- Enclustra GmbH
  - Company Profile
- FPGA Basics
  - Programmable Logic History
  - FPGA Architecture
  - FPGA Design Flow
  - Market Overview
- 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
  - Skills
  - How to Stand Out
- Outlook & Trends
Enable designers to focus on innovation rather than on infrastructure by the help of:

- Advanced FPGA silicon
- Socketable IP cores
- Development boards
- Reference designs

Base IP and tools provided by silicon vendor for free (e.g. SPI)

Domain-specific IP and tools provided by silicon vendor and design partners (e.g. FFT)

- Market-specific IP and tools provided by design partners and 3rd parties (e.g. video deinterlacer)
- Application-specific boards and FPGA designs done by the end customer
Outlook and Trends – CPU goes FPGA

- CPU goes FPGA (not FPGA goes CPU)

- Xilinx Extensible Processing Platform
  - ARM Dual Cortex-A9 @ 800 MHz
  - Memory interfaces and common peripherals as hard macros
  - Programmable logic custom peripherals and accelerators
  - High-bandwidth interconnect between CPU, peripherals and accelerators
  - Partial reconfiguration of programmable logic via CPU

- Fixed processing system
- Scalable programmable logic
- Broad range of available IP
- Software-centric development flow
Outlook and Trends – FPGA Mezzanine Card (FMC)

- PMC, XMC & Co. allow flexible I/O interfaces for traditional single-board computers
- Providing flexible I/O interfaces for FPGA-based processing boards is a different story
- The FPGA Mezzanine Card (FMC) standard is addressing this issue
  - 160 (LPC) or 400 (HPC) pin connector, up to 10 Gb/s
  - Standardized connector pin assignments optimized for FPGAs

Possible applications
- Different RF frontends for FPGA-based software defined radio systems
- Various I/O engines for a single FPGA-based processing board
Questions?

Martin Heimlicher  
Enclustra GmbH  
heimlicher@enclustra.com  
Fon +41 43 343 39 40

Marc Oberholzer  
Enclustra GmbH  
oberholzer@enclustra.com  
Fon +41 43 343 39 47

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