Our History of Embedding MathWorks Design Flows:
How and Why We Use Them

MATLAB EXPO 2016 Switzerland
Zentrum Paul Klee, Bern
23 June 2016

Martin Heimlicher
CEO and Founder
Enclustra GmbH
Impression
FPGA-Based Motion Control in Medical Testing
Key Takeaways

- Model-Based Design integrates seamlessly with existing FPGA design flows.
- Model-Based Design enables quick iterations.
- Use models to contract work. This lowers risks and reduces integration cost.
Everything FPGA.

Agenda

- Enclustra Company Profile
- Common Challenges in FPGA Design
- FPGA Design Flow and MathWorks Capabilities
- Model-Based FPGA Design
- MATLAB-Based FPGA Prototyping
- Conclusions
Enclustra Company Profile
Common Challenges in FPGA Design
FPGA Design Flow and MathWorks Capabilities
Model-Based FPGA Design
MATLAB-Based FPGA Prototyping
Conclusions
Focused on FPGA Technology – Everything FPGA!

Founded in 2004 – successfully in business for 12 years!

Headquarters in Technopark, Zurich, Switzerland

Branch offices in Germany, USA and China

2 Business units: FPGA Design Center, FPGA Solution Center

26 employees: 15 FPGA engineers plus 7 staff in Zurich, 4 employees abroad

Vendor-Independent
Enclustra Company Profile

**Common Challenges in FPGA Design**

FPGA Design Flow and MathWorks Capabilities

Model-Based FPGA Design

MATLAB-Based FPGA Prototyping

Conclusions
Common Challenges in FPGA Design
Adoption of new Technology

Changes in technology require changes in methodology
FPGAs have less inherent visibility than processors or analog electronics.

- No single stepping
- No breakpoints
- No simple measurement
- Difficult data interpretation
- Long compile times
Application specialists cannot easily access FPGA designs in general.

- Exception: Model-Based Design with e.g. MATLAB and Simulink
- However: Code generation is not always applicable
- FPGA Implementation specialists are often required
Agenda

- Enclustra Company Profile
- Common Challenges in FPGA Design
- **FPGA Design Flow and MathWorks Capabilities**
- Model-Based FPGA Design
- MATLAB-Based FPGA Prototyping
- Conclusions
FPGA Design Flow and MathWorks Capabilities

Requirements
- Specification
- System Design
- Functional Blocks Implementation
- Top-Level Integration
- Physical Implementation

Functional Requirements
- Performance Requirements
- Environmental Requirements
- Etc.
FPGA Design Flow and MathWorks Capabilities

System Design

1. Requirements Specification
2. System Design
3. Functional Blocks Implementation
4. Top-Level Integration
5. Physical Implementation
6. Overall System Design
7. Algorithm Design
8. Detailed Architecture Design
9. Pin Assignments, Clock and Reset Strategy
FPGA Design Flow and MathWorks Capabilities

Functional Blocks

Requirements Specification
System Design
Functional Blocks Implementation
Top-Level Integration
Physical Implementation

Functional Blocks Detailed Design
Functional Blocks Design Entry
Block-Level Functional Verification

Everything FPGA.
FPGA Design Flow and MathWorks Capabilities
Top-Level Integration

- Requirements Specification
- System Design
- Functional Blocks Implementation
- Top-Level Integration
- Physical Implementation

- Top-Level Functional Verification
FPGA Design Flow and MathWorks Capabilities
Physical Implementation

- Requirements Specification
- System Design
- Functional Blocks Implementation
- Top-Level Integration
- Physical Implementation

- Synthesis
- Place and Route
- Static Timing Analysis
- Bitstream Generation and FPGA Configuration
- In-Circuit Verification

Everything FPGA.
Enclustra Company Profile
Common Challenges in FPGA Design
FPGA Design Flow and MathWorks Capabilities
Model-Based FPGA Design
MATLAB-Based FPGA Prototyping
Conclusions
Model-Based FPGA Design
The Case for Model-Based Design

Electronics development is not a solo endeavour
Embedded FPGA/SoC systems consist of many individual components with different requirements.

- Processing algorithms
- Sensor/actor interfaces
- Host communication stacks
- Device drivers
- House keeping
Model-Based FPGA Design
HDL is no Software

**Software**
- Instructions are processed sequentially
- Simulink algorithms runs as software natively
- Floating-point is nearly as quick as fixed-point
- Readable to almost every engineer
- Achieving target speed
  - Less instructions per execution!

**HDL (Logic)**
- Everything happens in parallel
- Simulink algorithms must be converted to a circuit structure
- Fixed-point is way more efficient
- Hard to read and understand
- Achieving target speed
  - Increasing clock speed?
  - More operations in one clock cycle?
  - More parallelism?
  - Latency vs. Throughput
Model-Based FPGA Design
Algorithm Designers don’t need FPGA Know-How

Code generation from Simulink makes FPGAs and SoCs accessible to algorithm designers.

- No FPGA knowledge required
- Function and performance verification in Simulink
- No HDL implementation bugs
- Suitable subset of Simulink needs to be established
- For SoCs, all kinds of code (HDL, C, C++) can be generated from the same model. Partitioning is still manual work.
Model-Based FPGA Design
Design Teams still need FPGA Know-How

*Remember that electronics development is not a solo endeavour?*

- Simulink cannot be used to create a complete FPGA design
  - External interfaces
  - Communication protocols
  - Pin assignments and timing constraints
  - Synthesis, place and route
- FPGA know-how helps get there even quicker
- Framework can be reused for many projects
Enclustra Company Profile
Common Challenges in FPGA Design
FPGA Design Flow and MathWorks Capabilities
Model-Based FPGA Design
MATLAB-Based FPGA Prototyping
Conclusions
MATLAB-Based Prototyping
Basic Concept

- Algorithm development in MATLAB
- FPGA setup/control and data analysis in MATLAB
- Bit-true hardware/MATLAB co-simulation
- **Design for test**
- **Provide different data access points**
MATLAB-Based Prototyping Regression Testing

- Run huge data sets very quickly
- Analysis of output in MATLAB
MATLAB-Based Prototyping
Development Risk

MATLAB based prototypes significantly reduce development risk.

- Thorough testing becomes possible before the complete system is implemented.
- Application specialists are more involved and can detect issues earlier.
- In real-world projects, we detect more than 95% of the issues leading to design changes before the first system-level tests are performed.
MATLAB-Based Prototyping
Development Time

MATLAB based prototypes shorten development time.

- Issues can be located quickly due to signal capture and analysis at various places in the design.
- Less iterations on hardware due to real-world data from the design being available in MATLAB.
- Simulation times can be reduced by orders of magnitude due to hardware/MATLAB co-simulations.
- Regression testing on hardware instead of HDL simulations.
Enclustra Company Profile
Common Challenges in FPGA Design
FPGA Design Flow and MathWorks Capabilities
Model-Based FPGA Design
MATLAB-Based FPGA Prototyping
Conclusions
MATLAB and Simulink can be successfully employed in the FPGA design process, even by SMEs.

- Model-Based Design/HD[/D code generation brings algorithm designers closer to the FPGA design process.
- Model-Based Design greatly reduces development time and risk.
- However, a design team still needs FPGA-specific know-how to succeed.
- Choose the appropriate tool for every task in your FPGA design flow.