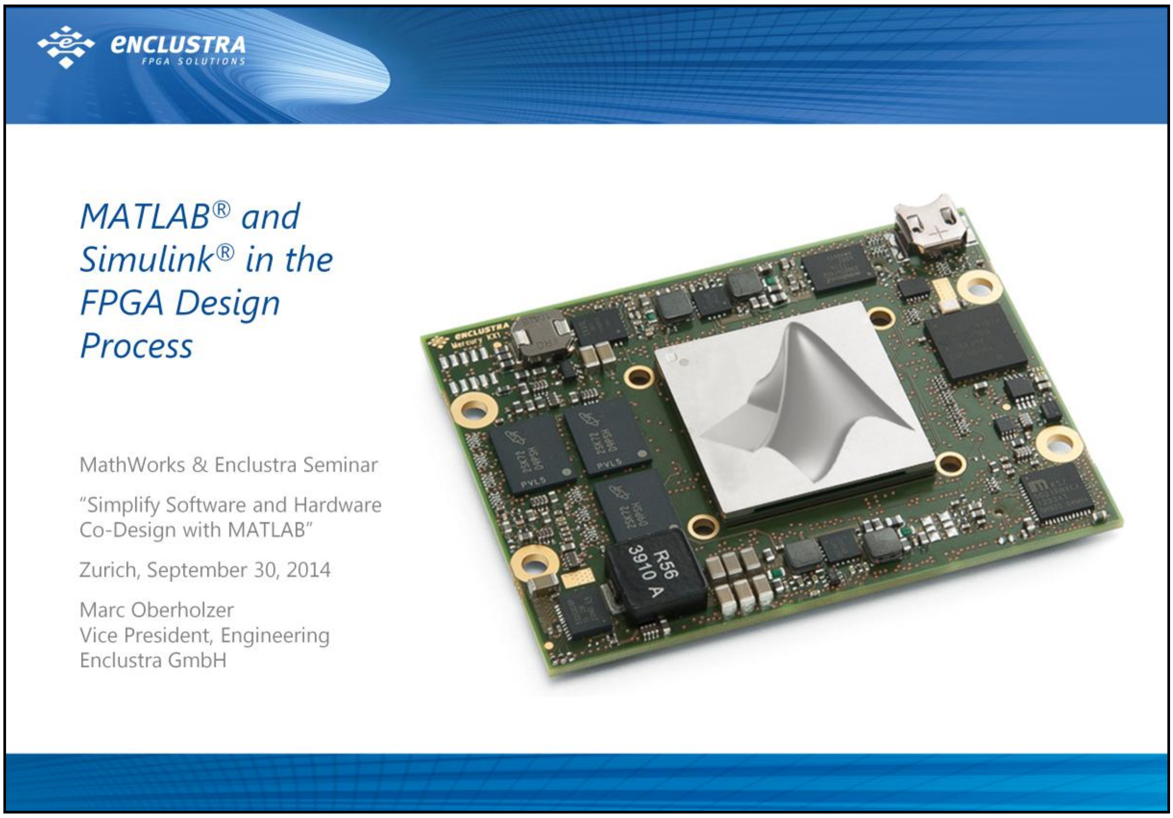


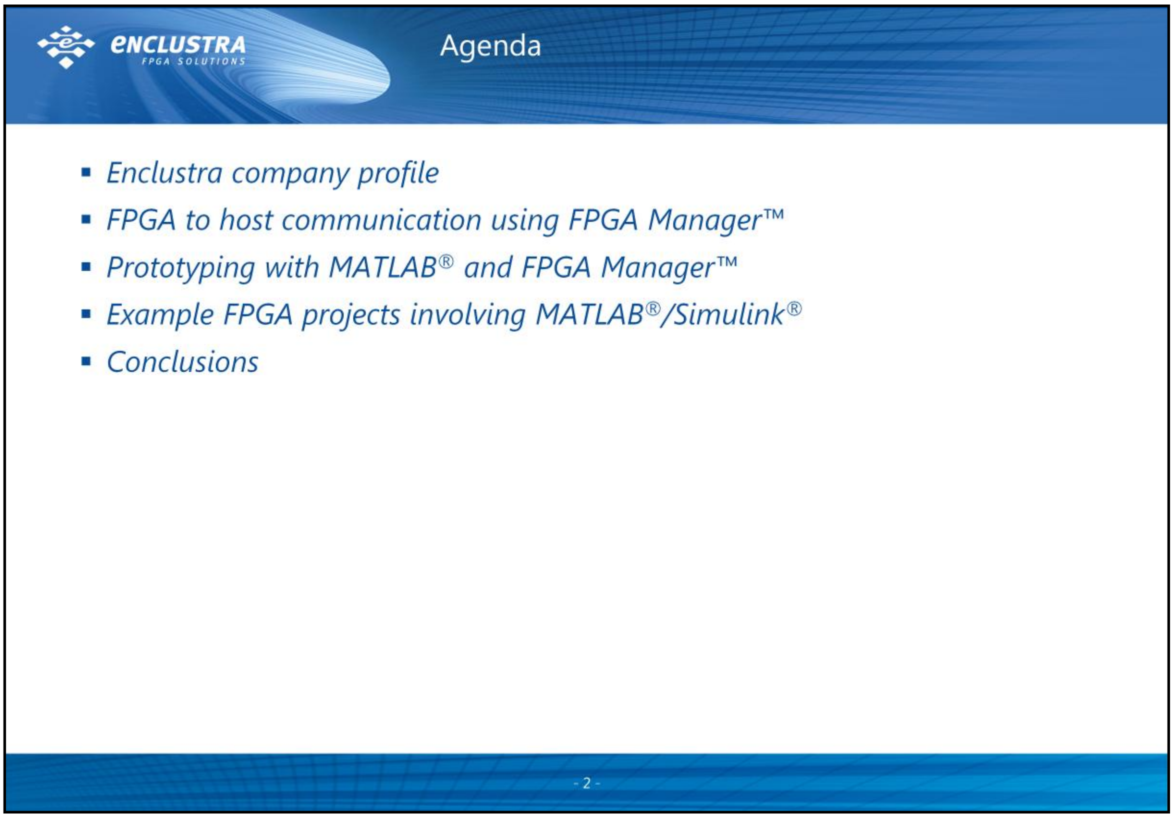
MATLAB® and Simulink® in the FPGA Design Process

MathWorks & Enclustra Seminar

"Simplify Software and Hardware Co-Design with MATLAB"

Zurich, September 30, 2014

Marc Oberholzer
Vice President, Engineering
Enclustra GmbH



Agenda

- *Enclustra company profile*
- *FPGA to host communication using FPGA Manager™*
- *Prototyping with MATLAB® and FPGA Manager™*
- *Example FPGA projects involving MATLAB®/Simulink®*
- *Conclusions*

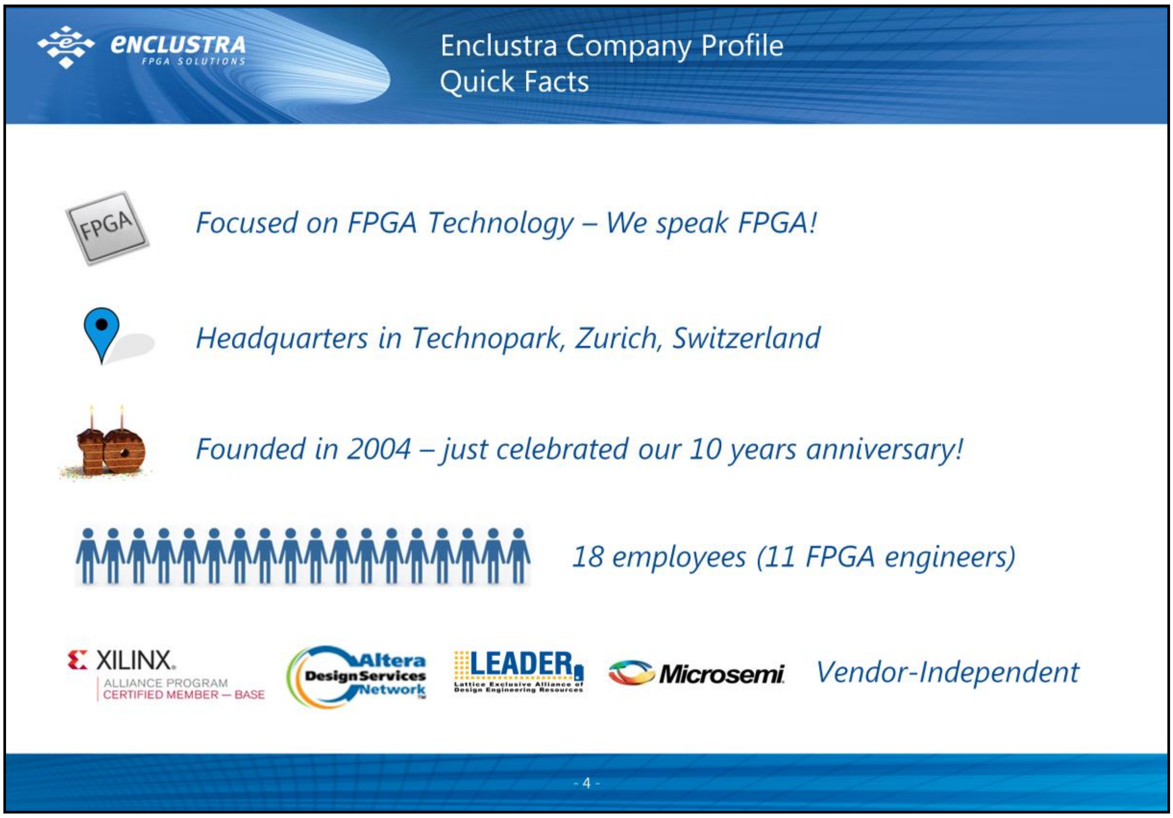
Agenda

- ***Enclustra company profile***
- *FPGA to host communication using FPGA Manager™*
- *Prototyping with MATLAB® and FPGA Manager™*
- *Example FPGA projects involving MATLAB®/Simulink®*
- *Conclusions*

- 3 -

Outlook

The next few slides give an overview of Enclustra as a company, its business segments, design services, hardware and IP products.



Enclustra Company Profile
Quick Facts

Focused on FPGA Technology – We speak FPGA!

Headquarters in Technopark, Zurich, Switzerland

Founded in 2004 – just celebrated our 10 years anniversary!

18 employees (11 FPGA engineers)

Vendor-Independent

XILINX
ALLIANCE PROGRAM
CERTIFIED MEMBER – BASE
Altera
Design Services
Network
LEADER
Lattice Exclusive Alliance of Design Engineering Resources
Microsemi
- 4 -

Short Profile

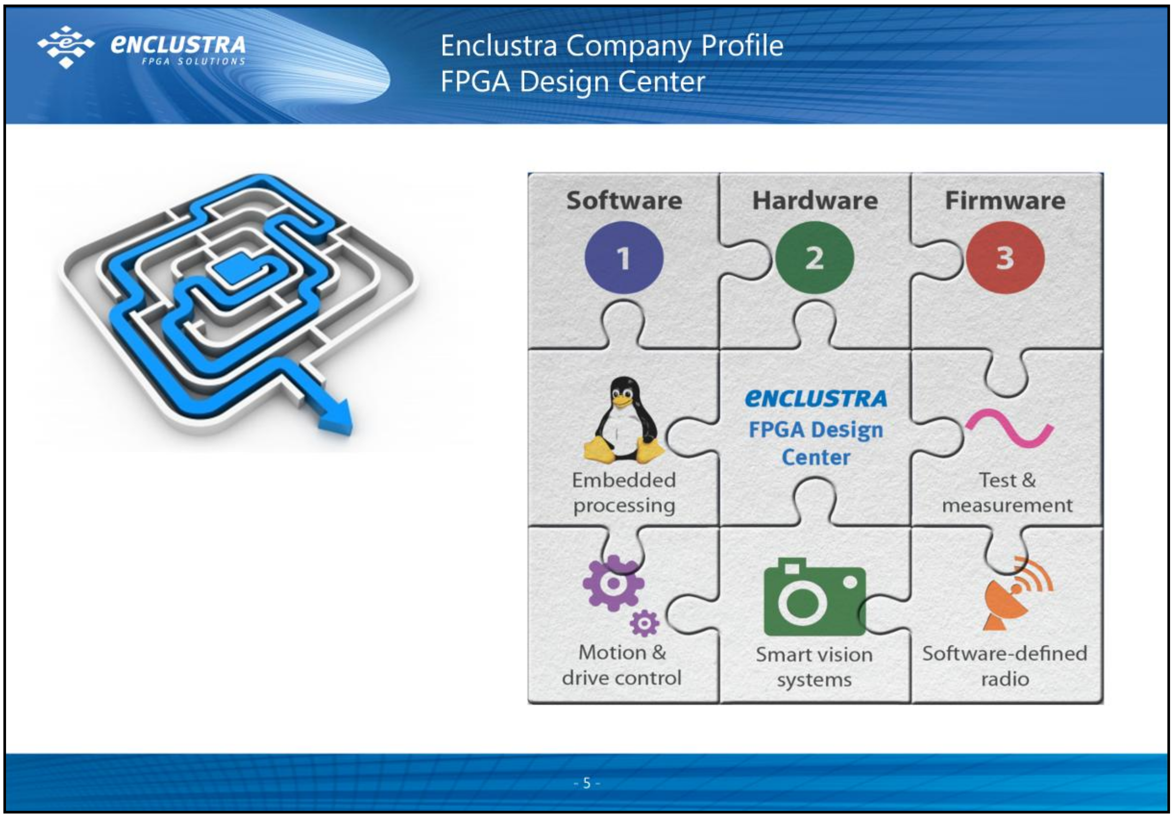
Enclustra is a dynamic, innovative and successful FPGA design service company located in Technopark Zurich, Switzerland.

Our FPGA engineers have in-depth knowledge in various application areas like software defined radio, drive control, digital signal processing and data acquisition systems.

Investing in employee training and keeping critical knowledge up to date on a regular basis enables us to find ideal solutions at a minimal expense for our clients.

We're vendor-independent, and we're design service partners of Xilinx®, Altera® and Lattice Semiconductor – this close communication allows us to be forward-looking in our design process, and remain on the cutting edge of the most advanced FPGA technology.

Enclustra – we speak FPGA!



Enclustra Company Profile
FPGA Design Center

Software 1
Hardware 2
Firmware 3

Embedded processing
ENCLUSTRA FPGA Design Center
Test & measurement
Motion & drive control
Smart vision systems
Software-defined radio

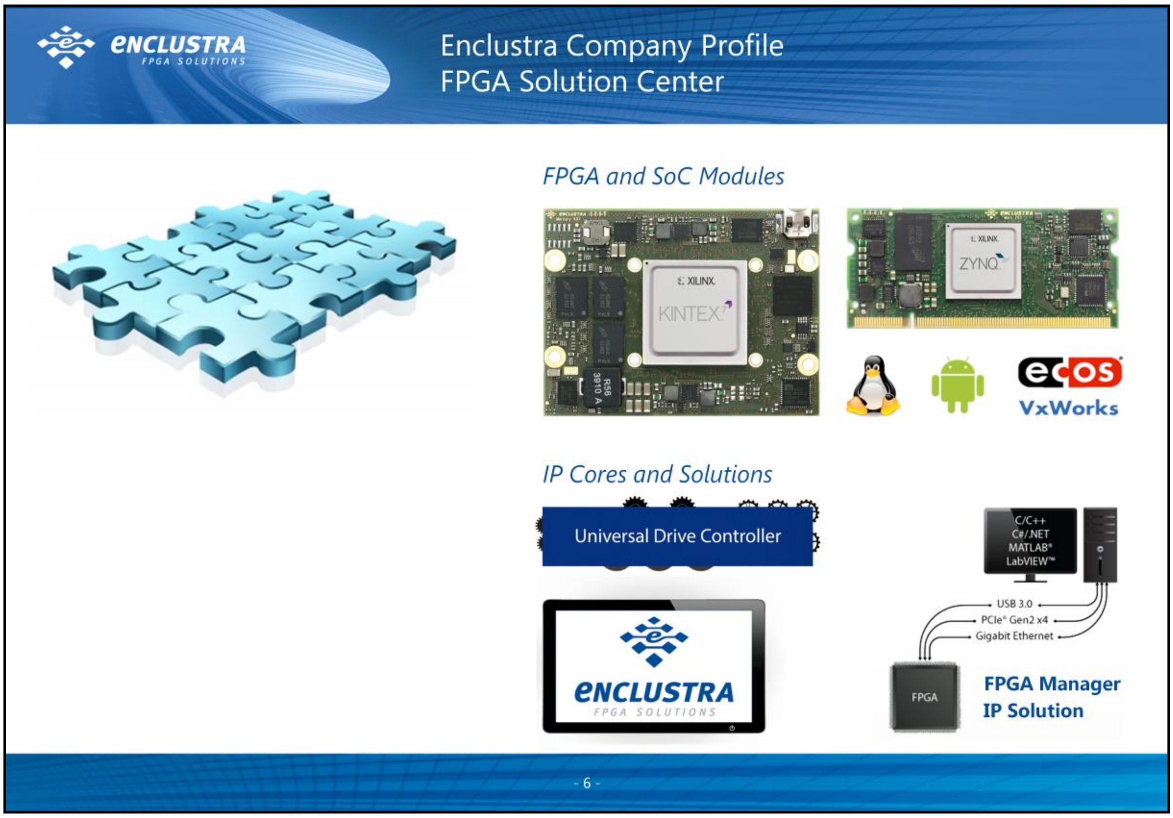
- 5 -

Short Profile

With the FPGA Design Center we accompany our customers on their way from an initial idea to a complete FPGA-based system.

Our design center offers design and support in all areas of FPGA-based system development, in a wide number of applications. High-speed hardware, HDL firmware, embedded software, real-time operating systems – our expertise covers every stage of the design process, from specification up to industrialization and manufacturing.

We have accumulated almost 100 person-years of FPGA hardware, firmware and software design experience and have immediate access to a comprehensive network of 3rd party consultants with specialized expertise for tasks like analog RF hardware design or thermal design. A lot of our past projects, particularly in the software defined radio and test & measurement areas, involved MATLAB and/or Simulink.



Enclustra Company Profile
FPGA Solution Center

FPGA and SoC Modules

IP Cores and Solutions

Universal Drive Controller

FPGA Manager
IP Solution

- 6 -

ENCLUSTRA
FPGA SOLUTIONS
XILINX
KINTEX
XILINX
ZYNQ
ecos
VxWorks
C/C++
C#/.NET
MATLAB®
LabVIEW™
USB 3.0
PCIe® Gen2 x4
Gigabit Ethernet
FPGA
ENCLUSTRA
FPGA SOLUTIONS

FPGA and SoC Modules

We develop and sell our own FPGA and system-on-chip (SoC) modules, based on Xilinx and Altera devices, for our customers to integrate into their own systems.

9 different modules, in 2 different families, compatible with 6 different base boards – the diversity of our products allows the customer to select exactly the features and they need, down to a fine grain.

Mars Module Family

- SO-DIMM 67.6 x 30 mm, 96-108 user I/Os, 0-2 MGTs
- 1-2 Ethernet ports, 0-1 USB port, 3.3V single supply voltage
- Very low-priced SO-DIMM connectors produced for the laptop industry

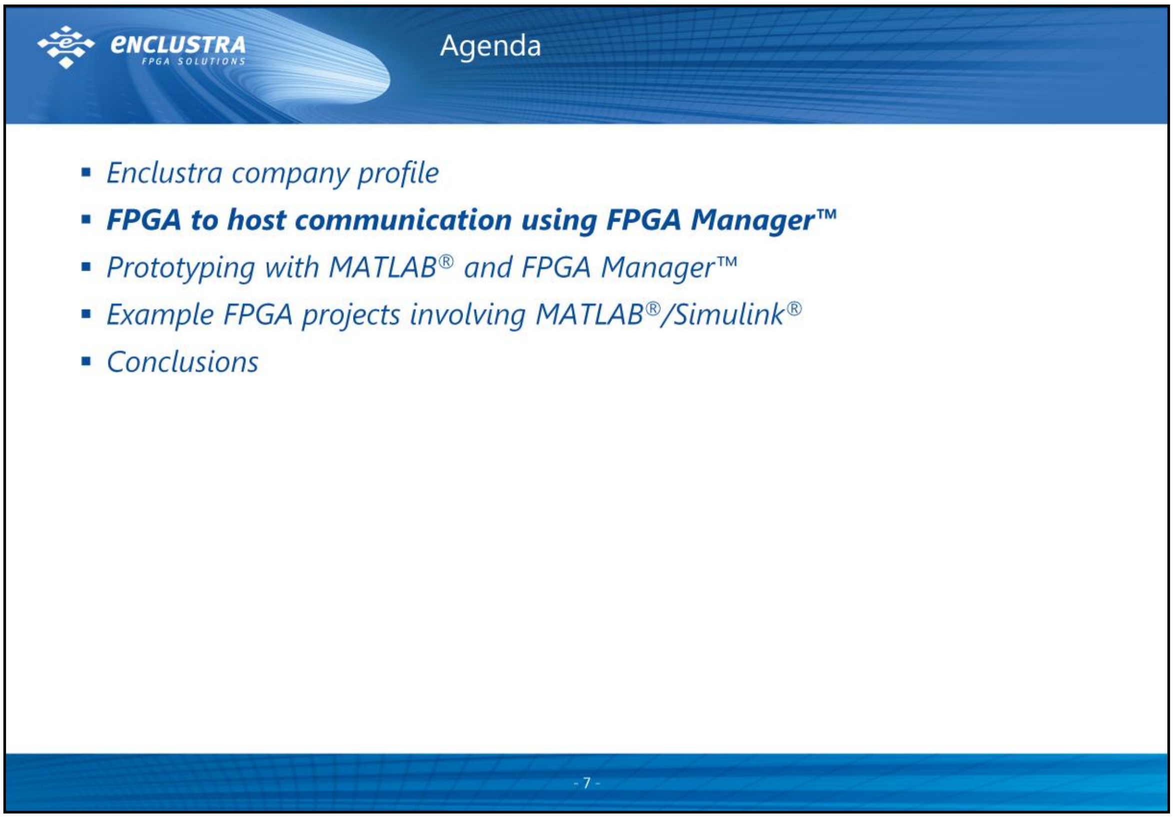
Mercury Module Family

- 56 x 54 to 72 x 54 mm, 146-188 user I/Os, 0-8 MGTs
- 1-2 Ethernet ports, 1 USB port, 5-15V single supply voltage
- High-performance Hirose FX10 connectors

IP Cores and Solutions

Our FPGA-optimized IP cores and solutions enable quick, easy addition of desired functionality to any FPGA design, with minimal resource usage and minimal design cost.

- Universal Drive Controller – Advanced Velocity Estimator
- PROFINET IRT – UDP/IP Ethernet – Display Controller 2D
- FPGA Manager – Stream Buffer Controller



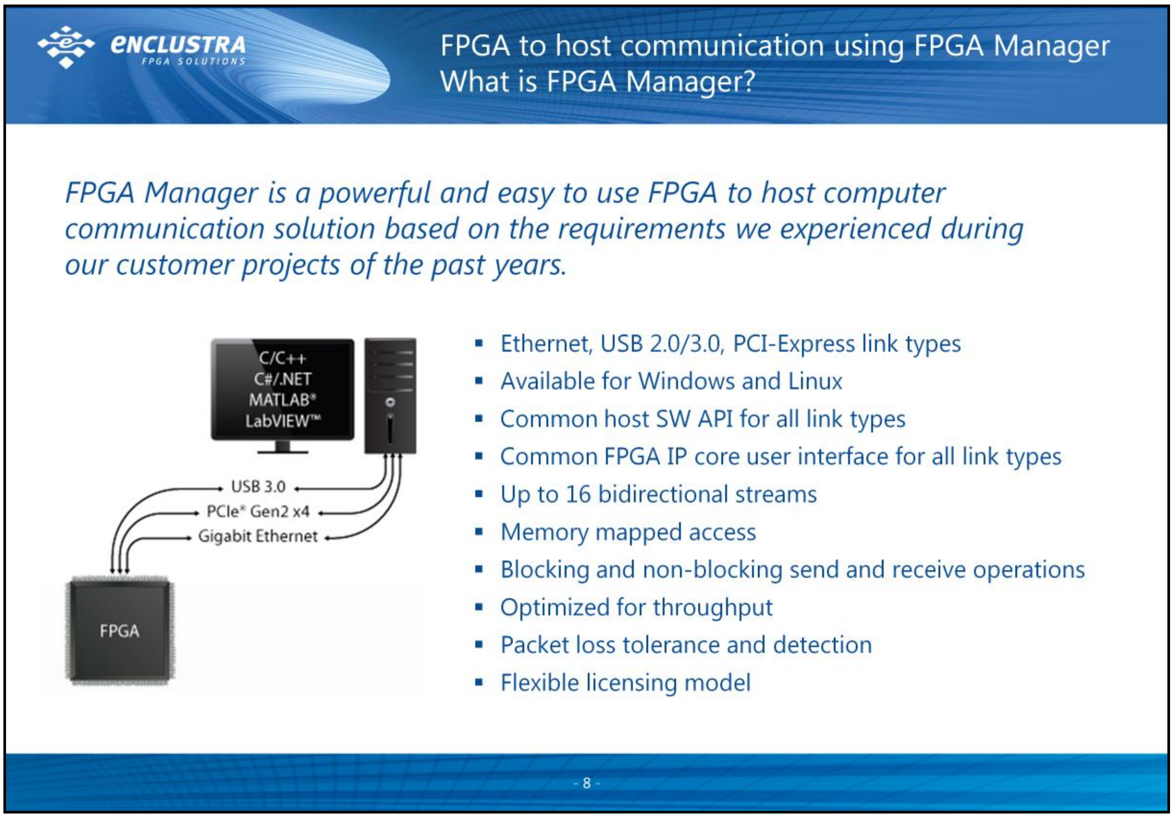
Agenda

- *Enclustra company profile*
- ***FPGA to host communication using FPGA Manager™***
- *Prototyping with MATLAB® and FPGA Manager™*
- *Example FPGA projects involving MATLAB®/Simulink®*
- *Conclusions*

- 7 -

Outlook

The next few slides show how the ever-present FPGA to host communication issue can be solved for good by using Enclustra's FPGA Manager IP solution.



ENCLUSTRA FPGA SOLUTIONS

FPGA to host communication using FPGA Manager
What is FPGA Manager?

FPGA Manager is a powerful and easy to use FPGA to host computer communication solution based on the requirements we experienced during our customer projects of the past years.

C/C++
C#/.NET
MATLAB®
LabVIEW™
USB 3.0
PCIe® Gen2 x4
Gigabit Ethernet
FPGA

- Ethernet, USB 2.0/3.0, PCI-Express link types
- Available for Windows and Linux
- Common host SW API for all link types
- Common FPGA IP core user interface for all link types
- Up to 16 bidirectional streams
- Memory mapped access
- Blocking and non-blocking send and receive operations
- Optimized for throughput
- Packet loss tolerance and detection
- Flexible licensing model

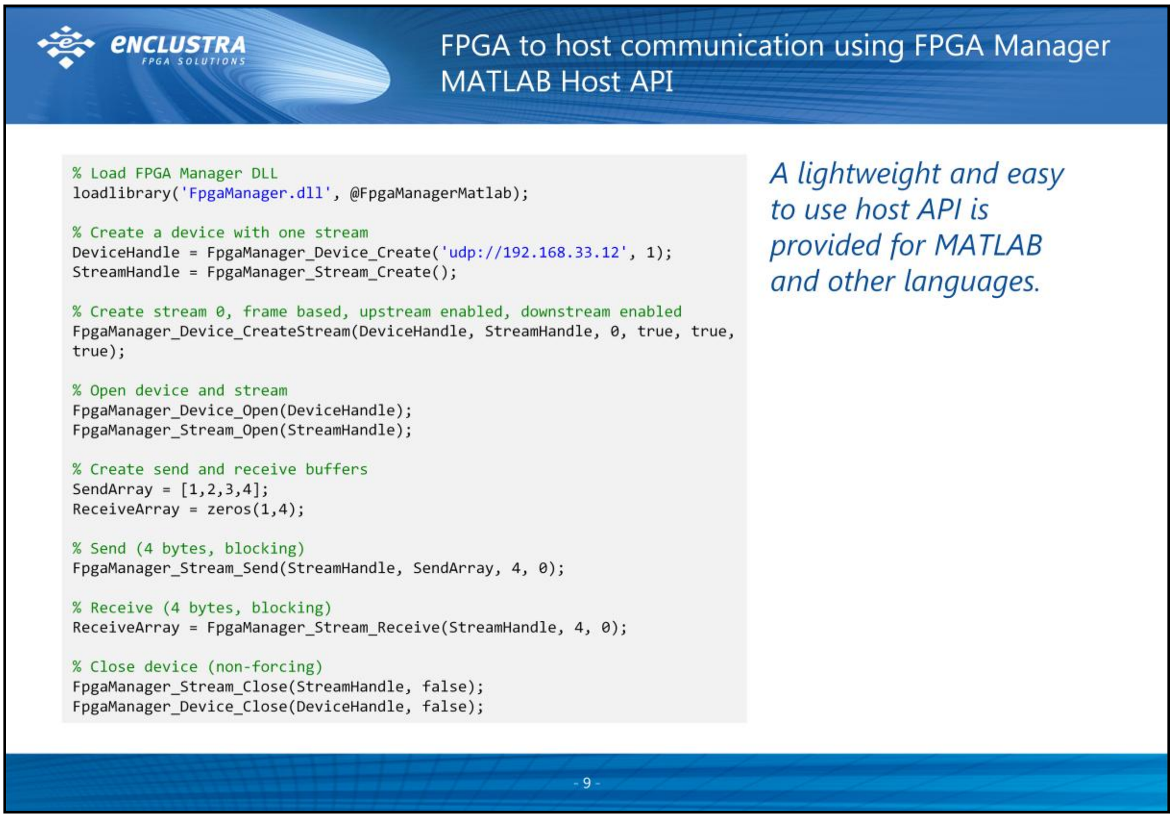
- 8 -

Background

The majority of our past projects required an FPGA to communicate with some kind of host computer in one or another way. Some of the requirements have been common to most of the projects (e.g. streaming and memory mapped data transfers), while others turned out to be diametrically opposed (e.g. operating systems or link types). FPGA Manager is an attempt to accommodate all of these requirements in a single communication solution, commercially available from Enclustra.

Short Description

Enclustra's FPGA Manager solution allows for easy and efficient data transfer between a host and a FPGA over different interface standards like USB 2.0/3.0, Gigabit Ethernet and PCI-Express. The solution includes a host software library (DLL), a suitable IP core for the FPGA and device controller firmware, if necessary. The user host application can communicate with the FPGA through a simple API consisting of simple read/write data commands hiding the complexity of the underlying protocols. Streaming and memory-mapped accesses are supported.



FPGA to host communication using FPGA Manager MATLAB Host API

```
% Load FPGA Manager DLL
loadlibrary('FpgaManager.dll', @FpgaManagerMatlab);

% Create a device with one stream
DeviceHandle = FpgaManager_Device_Create('udp://192.168.33.12', 1);
StreamHandle = FpgaManager_Stream_Create();

% Create stream 0, frame based, upstream enabled, downstream enabled
FpgaManager_Device_CreateStream(DeviceHandle, StreamHandle, 0, true, true,
true);

% Open device and stream
FpgaManager_Device_Open(DeviceHandle);
FpgaManager_Stream_Open(StreamHandle);

% Create send and receive buffers
SendArray = [1,2,3,4];
ReceiveArray = zeros(1,4);

% Send (4 bytes, blocking)
FpgaManager_Stream_Send(StreamHandle, SendArray, 4, 0);

% Receive (4 bytes, blocking)
ReceiveArray = FpgaManager_Stream_Receive(StreamHandle, 4, 0);

% Close device (non-forcing)
FpgaManager_Stream_Close(StreamHandle, false);
FpgaManager_Device_Close(DeviceHandle, false);
```

A lightweight and easy to use host API is provided for MATLAB and other languages.

- 9 -

Design Paradigm

The FPGA Manager host API has been designed with simplicity, portability and ease of use in mind.

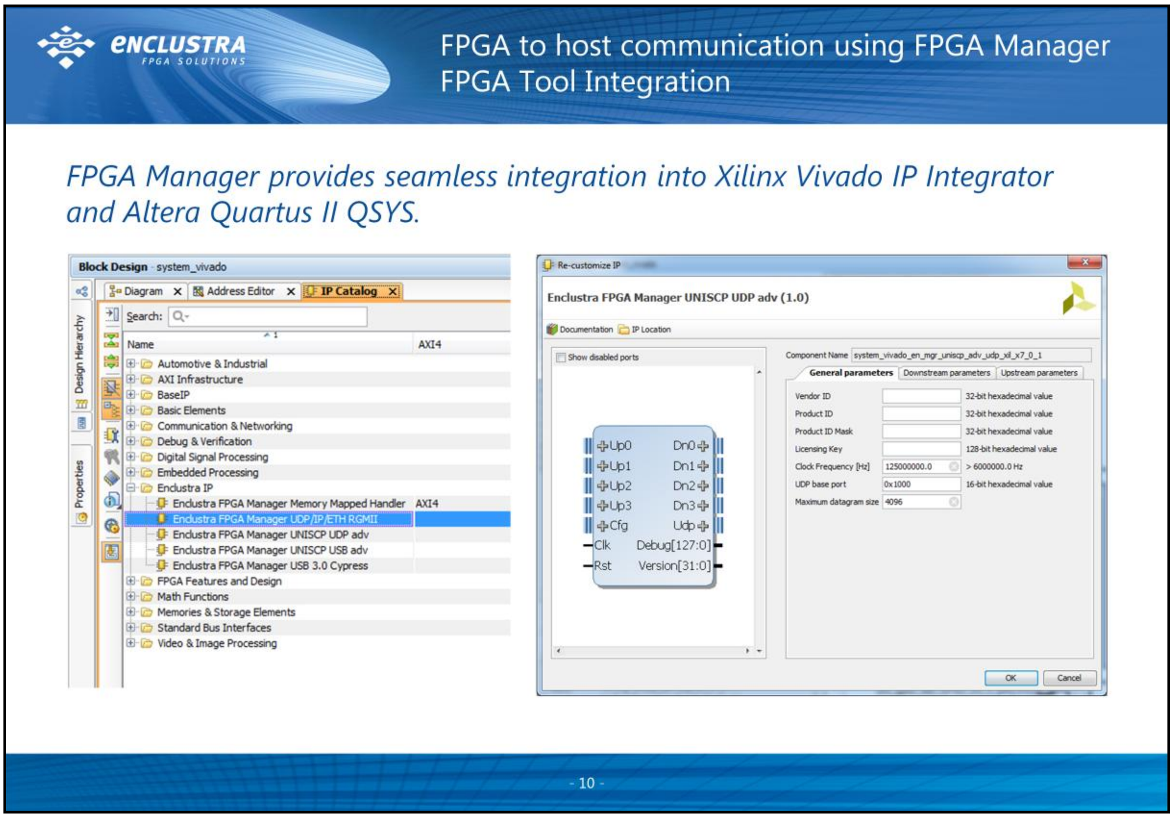
Example Code

The example above shows a very simple MATLAB script for setting up FPGA Manager communication between an FPGA and a host computer running MATLAB.

First, the FPGA Manager DLL is loaded. After that, device and stream handles are created. After this, a stream is configured on the created device. Once the device and the stream are configured, the device and the stream are opened. At this point, FPGA manager is ready for communication. Since the user is responsible for providing send and reveice buffers, these buffers are prepared. After that a single send and a single receive operation are issued. As a last step, the stream and the device are closed again.

Portability

Other languages than MATLAB, such as C, C++ and C#, are supported with a very similar API, which makes the migration between different host programming languages as easy as it gets.



FPGA to host communication using FPGA Manager
FPGA Tool Integration

FPGA Manager provides seamless integration into Xilinx Vivado IP Integrator and Altera Quartus II QSYS.

Component Name: system_vivado_en_mgr_uniscp_adv_udp_xil_x7_0_1

Vendor ID		32-bit hexadecimal value
Product ID		32-bit hexadecimal value
Product ID Mask		32-bit hexadecimal value
Licensing Key		128-bit hexadecimal value
Clock Frequency [Hz]	125000000.0	> 6000000.0 Hz
UDP base port	0x1000	16-bit hexadecimal value
Maximum datagram size	4096	

Graphically assisted Design Environments

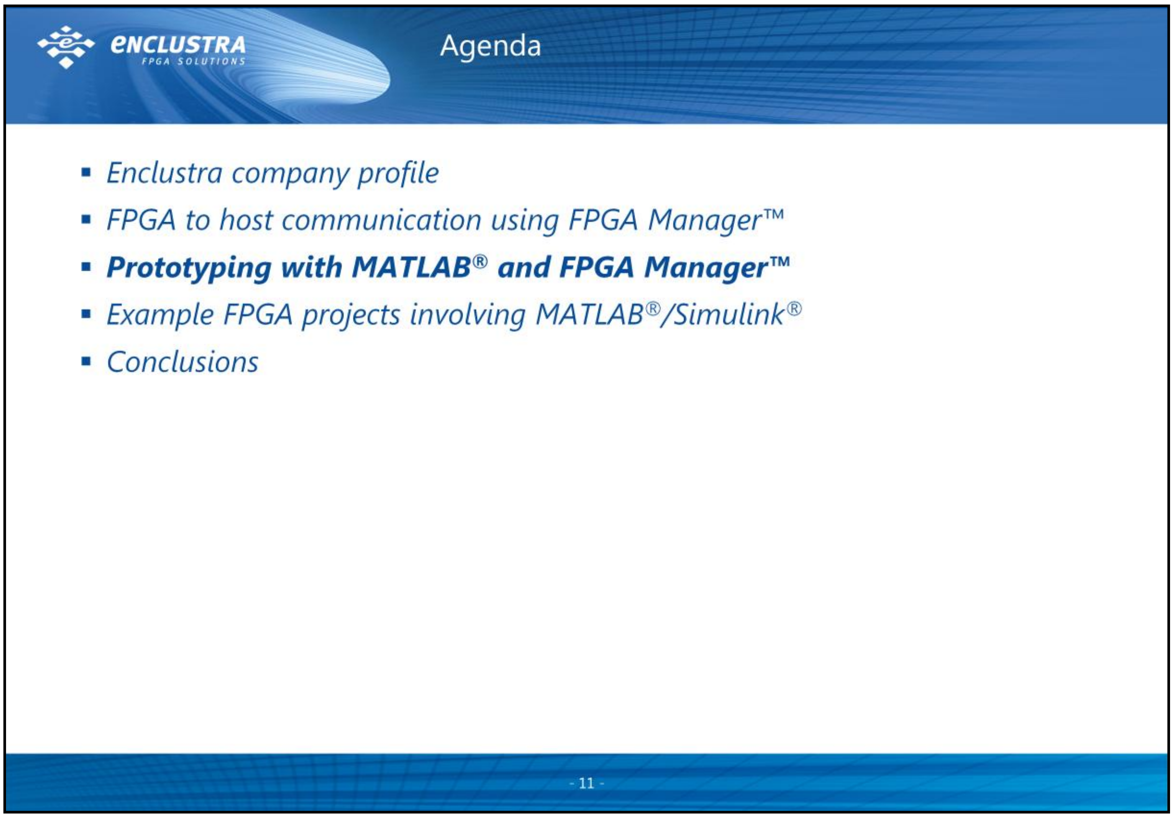
Today's FPGA design tools provide graphically assisted design environments which allows even the novice user to integrate company-developed or third-party IP blocks into new or existing FPGA designs. The successful use of the technology requires a carefully chosen concept when developing IP cores. Instead of specifying dozens of textual generic parameters, a flexible IP core should be divided into a set of blocks that can then be interconnected graphically by an FPGA developer in order to obtain the features required for his target application.

FPGA Manager Block Set

The FPGA portion of Enclustra's FPGA Manager IP solution is basically a set of 6 IP cores, which together provide PCI-Express, USB 3.0 and Gigabit Ethernet streaming capabilities for up to 16 data streams. The use of AXI4 memory-mapped and AXI4-Stream interfaces for interconnecting these IP cores is fully supported by the Xilinx Vivado and Altera QSYS tools.

Integration with generated HDL Code

Since generated HDL code can also be deployed as a suitable IP core for the most common FPGA design tools, the integration of FPGA Manager with generated HDL code can be accomplished without writing a single line of HDL code.



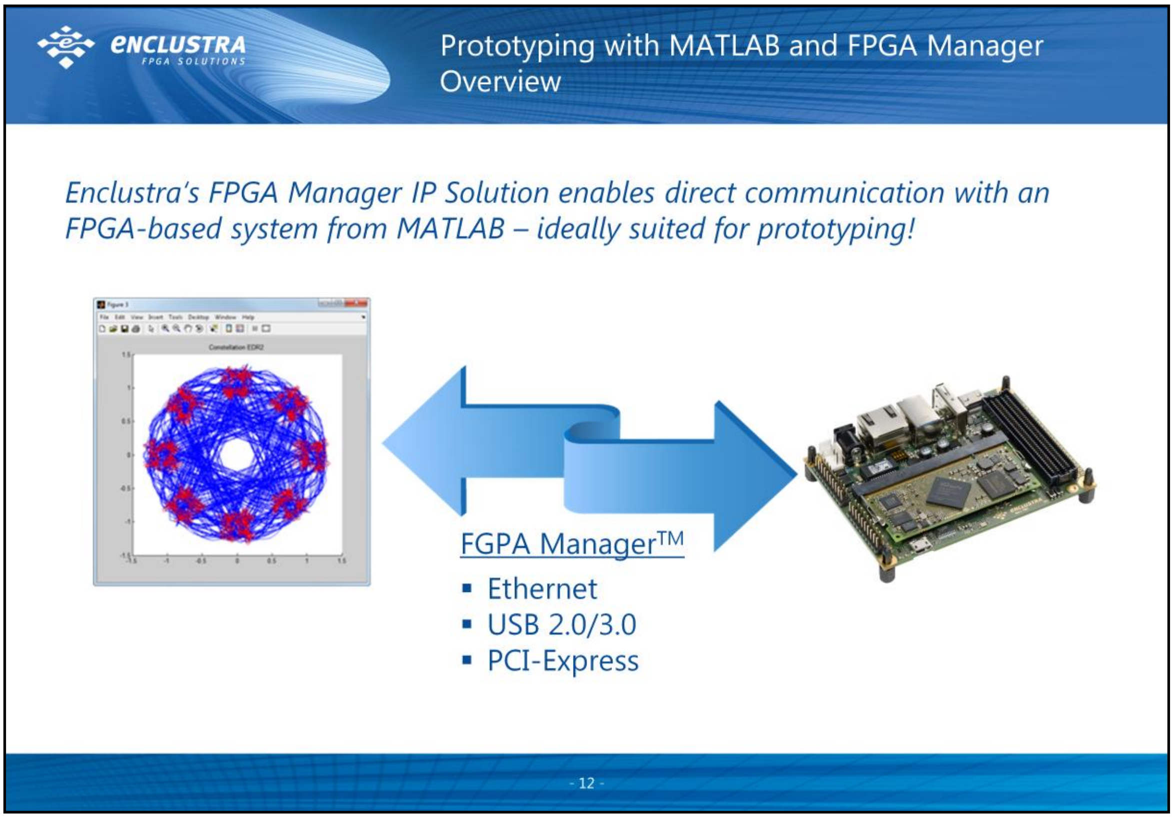
Agenda

- *Enclustra company profile*
- *FPGA to host communication using FPGA Manager™*
- ***Prototyping with MATLAB® and FPGA Manager™***
- *Example FPGA projects involving MATLAB®/Simulink®*
- *Conclusions*

- 11 -

Outlook

The next few slides show how powerful the combination of MATLAB/Simulink with FPGA Manager is for prototyping during FPGA-based system development.



Prototyping with MATLAB and FPGA Manager
Overview

Enclustra's FPGA Manager IP Solution enables direct communication with an FPGA-based system from MATLAB – ideally suited for prototyping!

FGPA Manager™

- Ethernet
- USB 2.0/3.0
- PCI-Express

- 12 -

MATLAB

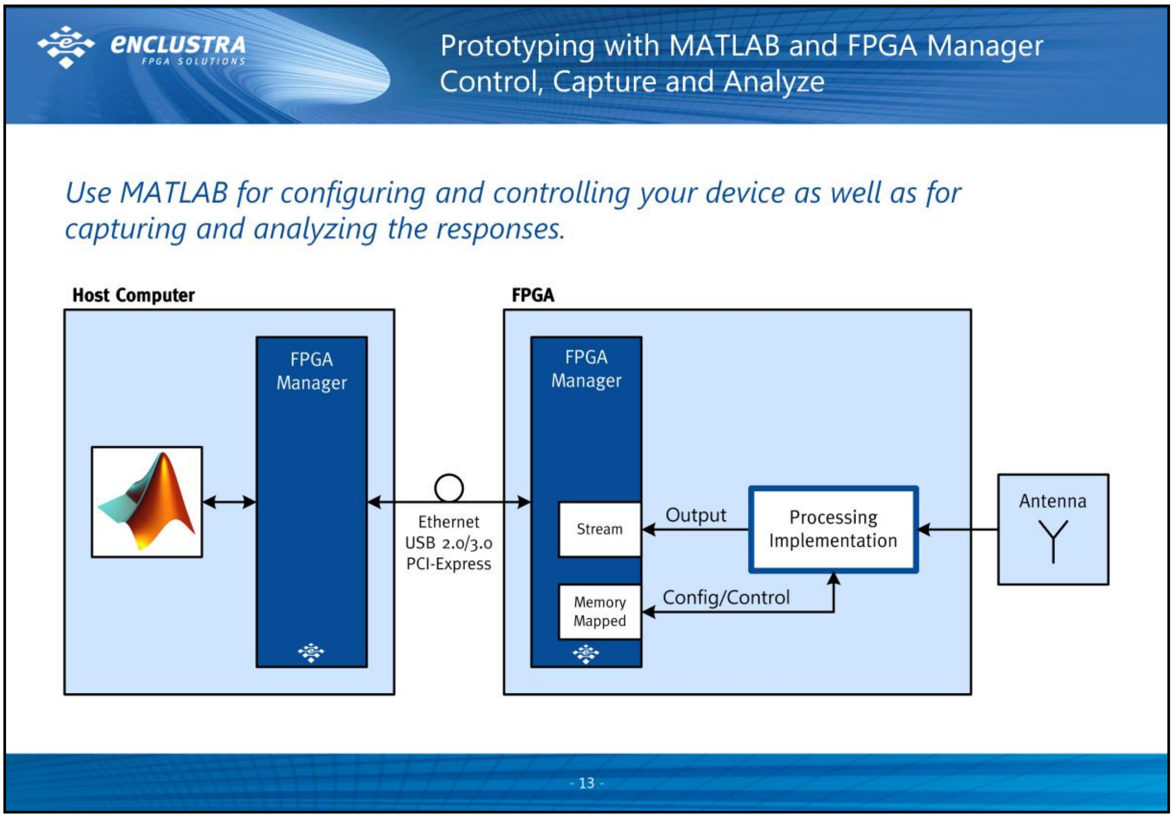
MATLAB is a de facto industry standard tool for signal processing and data analysis, almost every engineer working in these areas is familiar with it. MATLAB not only provides numerous toolboxes and commands for processing and analyzing data, but also contains a very powerful plotting engine.

In a traditional FPGA-based signal processing project, MATLAB is used for a) implementing a behavioral model of the processing block and b) building up a validation and verification framework around the processing block model.

FPGA Manager

With the ability of FPGA Manager to provide direct access to data on the FPGA from MATLAB, the MATLAB infrastructure developed during the modeling and design phase can directly be reused for the development, validation and verification of the associated FPGA implementation.

Using MATLAB together with FPGA Manager results in much more streamlined development, validation and verification flows when compared to file-based data exchange, e.g. via .csv files.



Prototyping with MATLAB and FPGA Manager
Control, Capture and Analyze

Use MATLAB for configuring and controlling your device as well as for capturing and analyzing the responses.

MATLAB as a Development User Interface

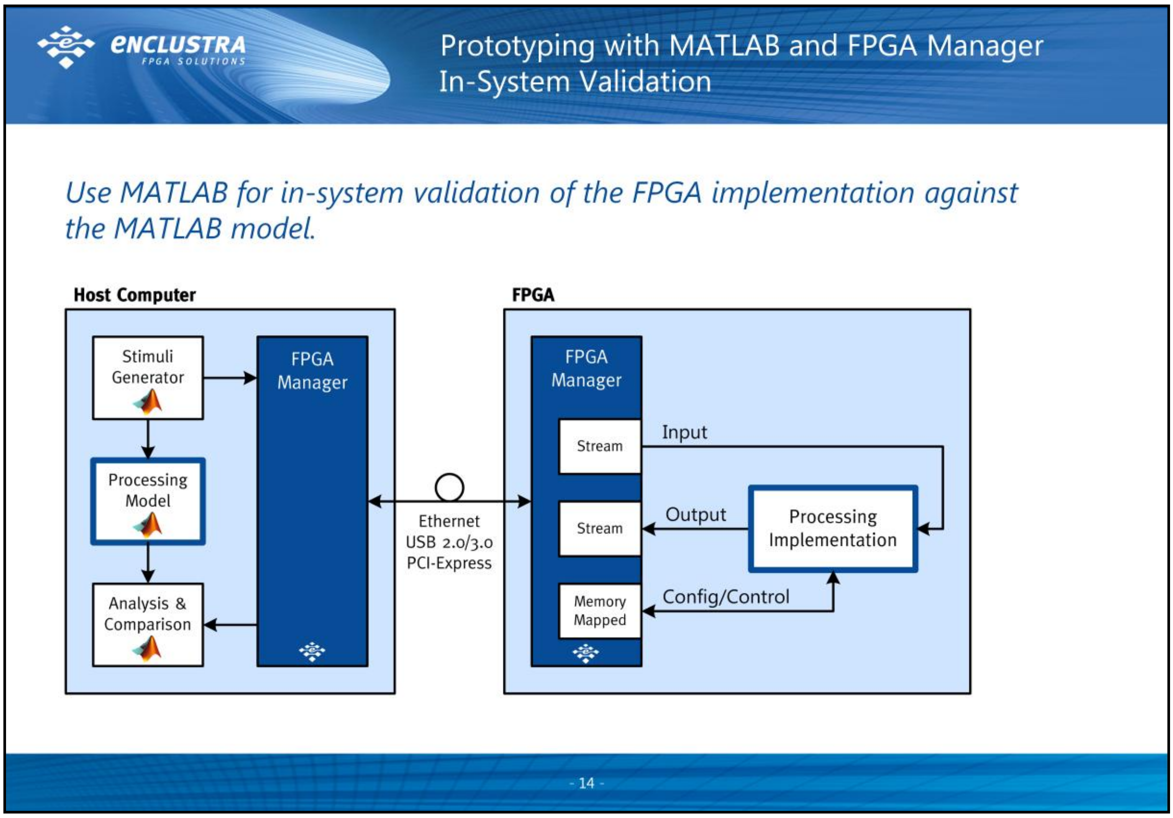
MATLAB scripts and/or GUIs are a convenient and fast way for coming up with a first user interface that provides the flexibility and ease of use required during the prototyping phase. This is not least the case because the vast number of data analysis commands are directly available in the user interface.

Integration with FPGA Manager

Thanks to PGA Manager's capability to directly access FPGA data from MATLAB, the user interface can not only be used for displaying acquired data and analysis results, but also for configuring and controlling the FPGA implementation.

Example

The example above shows a typical software defined radio (SDR) receiver setup using FPGA Manager and MATLAB. MATLAB is used for calculating filter coefficients and writing them to the FPGA filter implementation via FPGA Manager. After that, a configurabke number of samples is acquired and directly transferred to a MATLAB array variable. MATLAB is then used for e.g. calculating an FFT and plotting the frequency spectrum.



Prototyping with MATLAB and FPGA Manager In-System Validation

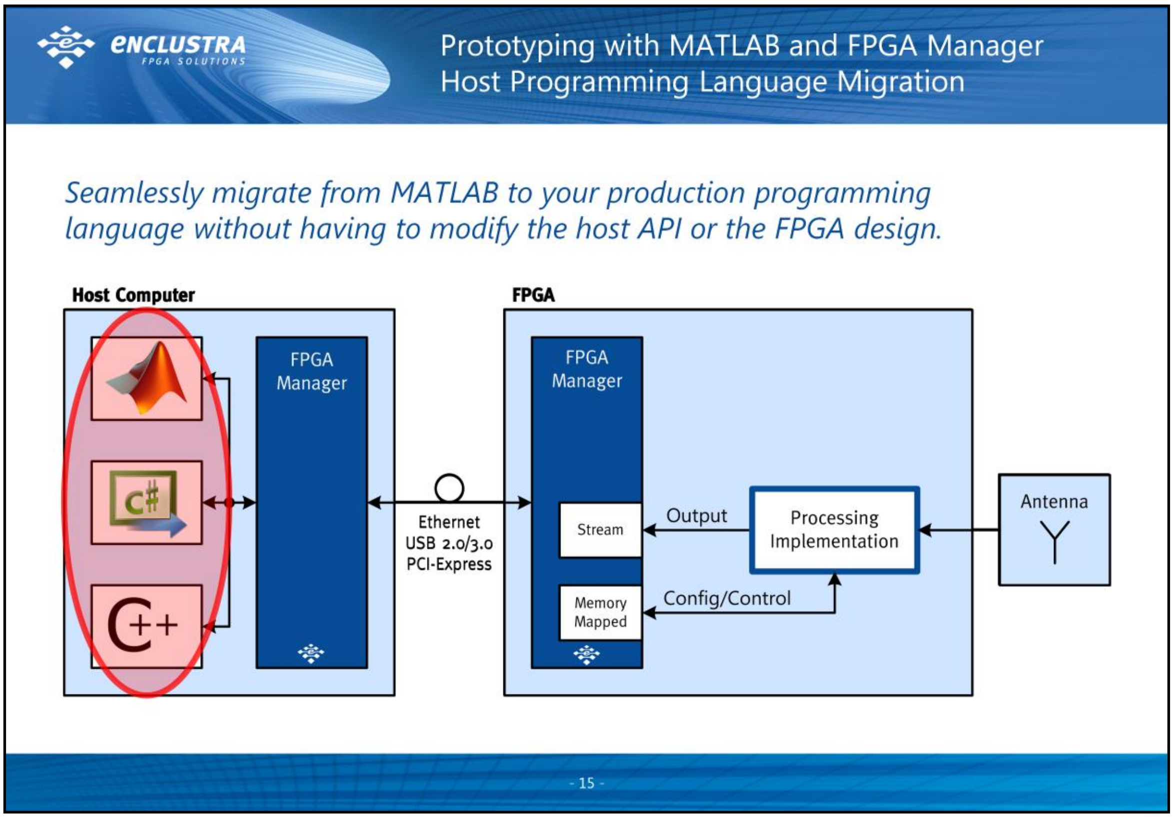
Use MATLAB for in-system validation of the FPGA implementation against the MATLAB model.

In-System Validation Testing

The combination of MATLAB and FPGA Manager not only provides a convenient user interface to the FPGA hardware, it even enables in-system validation testing of the FPGA implementation against the MATLAB model. With the vast data analysis capabilities of MATLAB directly at hand, also complex validation conditions can be efficiently implemented and checked.

FPGA-in-the-Loop also for non-generated Code

FPGA Manager works with all kind of processing block implementations, be it HDL code generated from MATLAB/Simulink, HDL code generated from C/C++/Open CL, 3rd party IP cores or manually written HDL code. This concept thus even enables hardware-in-the-loop (HIL) for all kind of implementations, not only for HDL code generated from MATLAB/Simulink.



Prototyping with MATLAB and FPGA Manager
Host Programming Language Migration

Seamlessly migrate from MATLAB to your production programming language without having to modify the host API or the FPGA design.

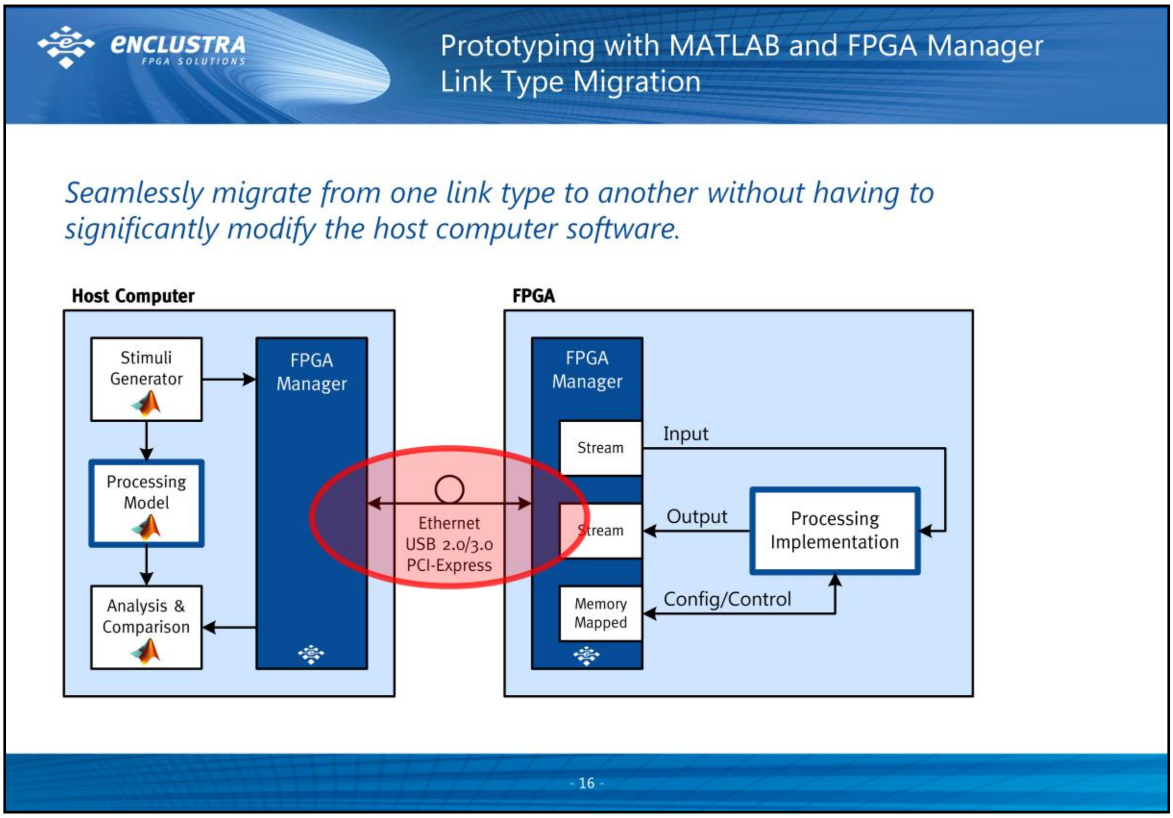
Seamless Migration between Propgramming Languages

MATLAB is often used during the development and prototyping phases where flexible data processing and analysis capabilities are most important. When it comes to user interfaces for production systems however, other programming languages like C/C++ or C#/.NET are more common.

Since FPGA Manager provides the same host API and IP core interfaces for all of these languages, the migration to the production system programming language goes without touching the FPGA implementation and is as simple as it probably gets.

Parallel Implementation

The well-defined common API and IP core interfaces also simplify and encourage the parallel development of the MATLAB user interface, the FPGA implementation and the production system user interface, which significantly reduces development time.



Prototyping with MATLAB and FPGA Manager
Link Type Migration

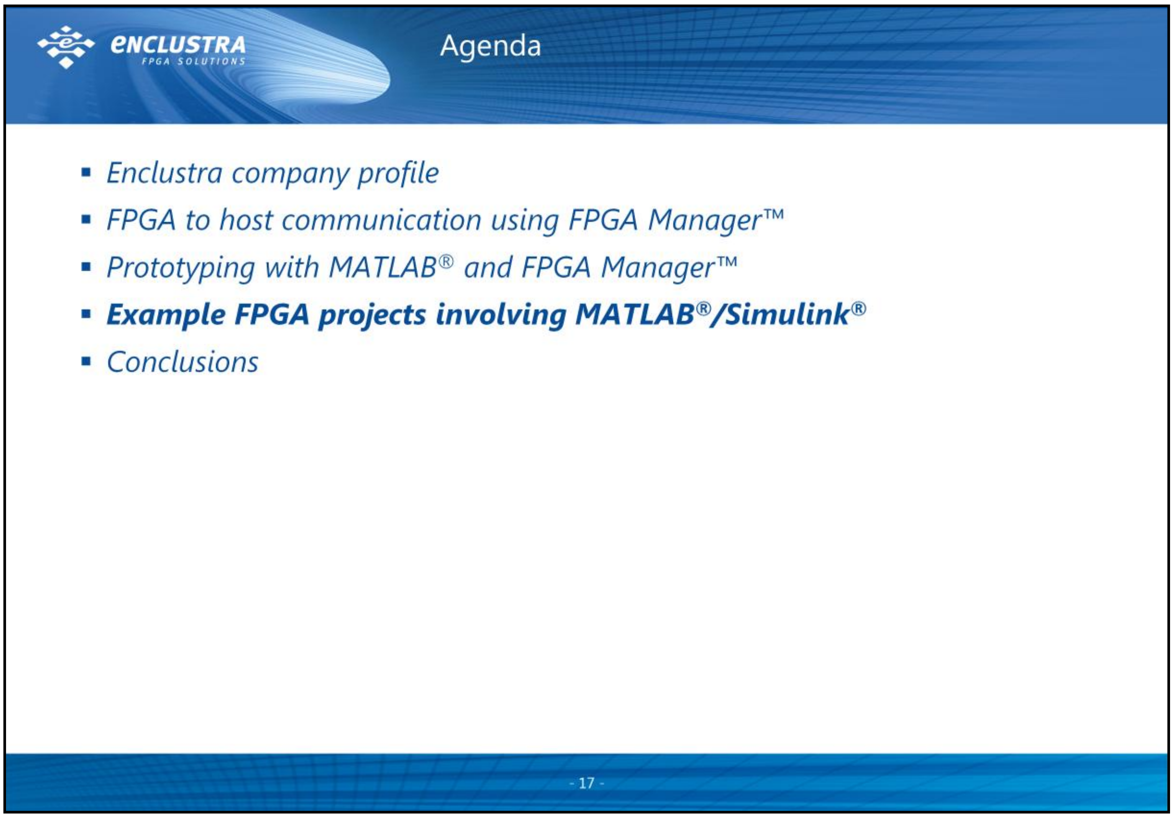
Seamlessly migrate from one link type to another without having to significantly modify the host computer software.

- 16 -

Seamless Migration between Ethernet, USB 2.0/3.0 and PCI-Express

The first steps in FPGA-based system prototyping often takes place with the help of FPGA evaluation boards – most often due to a lack of production system hardware. The FPGA evalzation board might not provide the exact interface foreseen for the production hardware, but it will most likely provide a Gigabit Ethernet port. It is thus common practice to start development with an Ethernet link and migrate to the production system link as soon as the production system hardware is available.

Since FPGA Manager provides the same host API and IP core interfaces for all of the supported link types, the migration to the production system link type goes without touching the host software implementation and FPGA processing implementation and is thus as simple as it probably gets.



ENCLUSTRA FPGA SOLUTIONS

Agenda

- *Enclustra company profile*
- *FPGA to host communication using FPGA Manager™*
- *Prototyping with MATLAB® and FPGA Manager™*
- ***Example FPGA projects involving MATLAB®/Simulink®***
- *Conclusions*

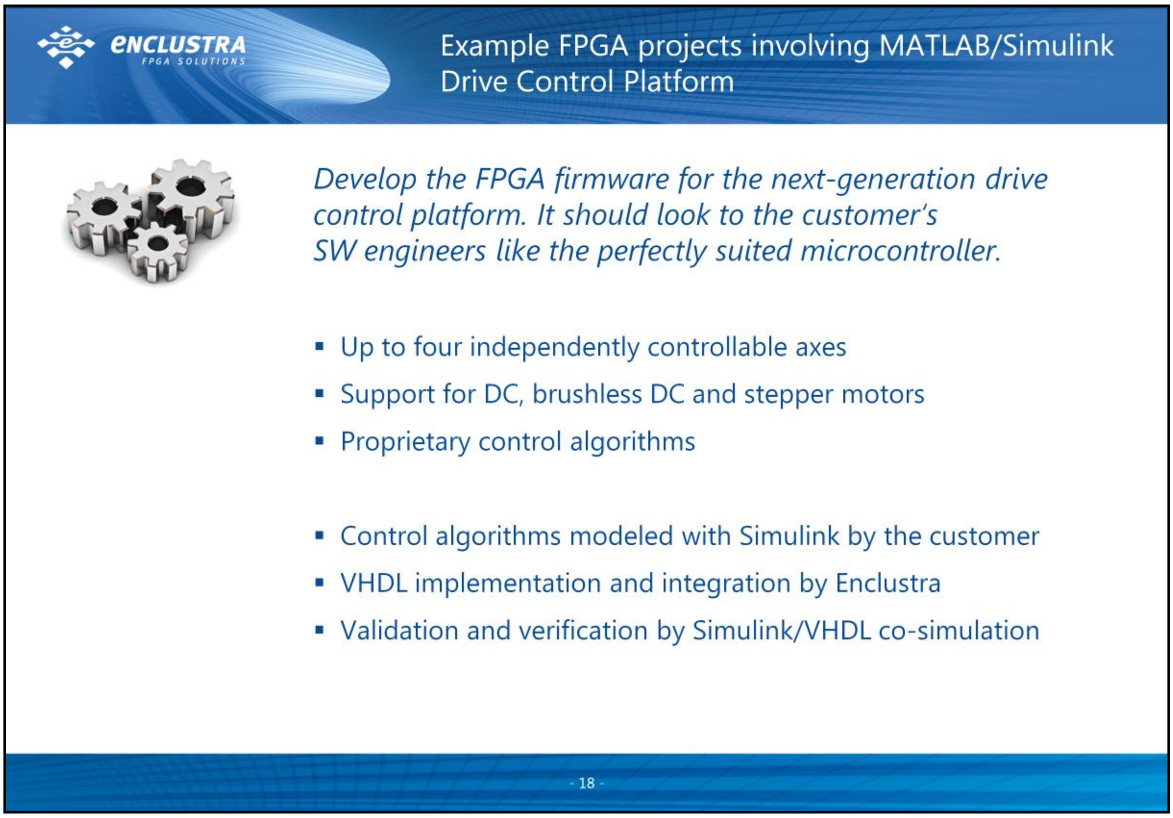
- 17 -

Outlook

The following examples show how MATLAB and/or Simulink have been employed in a selection of past and ongoing customer projects.

While EDA companies currently strongly highlight the ability to generate code from high level languages, MATLAB/Simulink are in most projects still used as a tools to help with modeling and verification of functional blocks in projects that follow a HDL design entry strategy.

The last example project makes use of HDL code generation while the other examples show the more traditional style of FPGA projects.



Example FPGA projects involving MATLAB/Simulink
Drive Control Platform

Develop the FPGA firmware for the next-generation drive control platform. It should look to the customer's SW engineers like the perfectly suited microcontroller.

- Up to four independently controllable axes
- Support for DC, brushless DC and stepper motors
- Proprietary control algorithms

- Control algorithms modeled with Simulink by the customer
- VHDL implementation and integration by Enclustra
- Validation and verification by Simulink/VHDL co-simulation

- 18 -

Starting Point

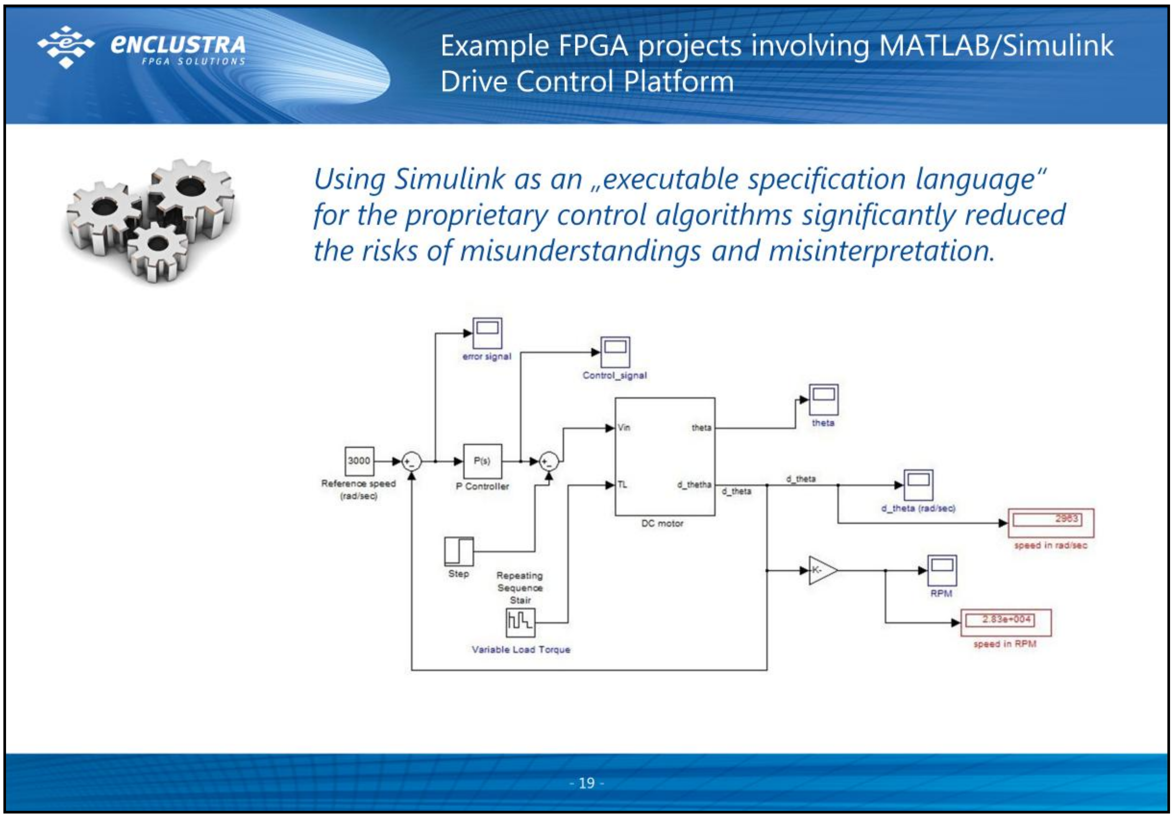
The customer plans to develop a drive control platform for powering their next-generation medical testing products.

- The customer is very experienced and has a very good expertise in designing drive control systems.
- The position and velocity control algorithms have already been designed by the customer using Simulink.
- The customer wishes to get a platform containing an FPGA soft processor, which they are free to program and which provides their algorithms and additional application-specific circuitry as peripherals implemented in FPGA logic.

Employment of MATLAB/Simulink

Simulink was used as an „executable specification language", the FPGA implementation has been done by manual HDL design entry. HDL code generation from Simulink has not been used, mainly because of the following reasons:

- The custom control algorithms were full of special cases and therefore quite logic-heavy and not very well suited for HDL code generation.
- A lot of additional and also logic-heavy circuitry that was only loosely coupled to the control algorithms had to be integrated with the soft processor too.
- Because of the planned production volumes, FPGA resource usage was a major concern in this project – we had to play a lot of tricks to fit the design into the target device.

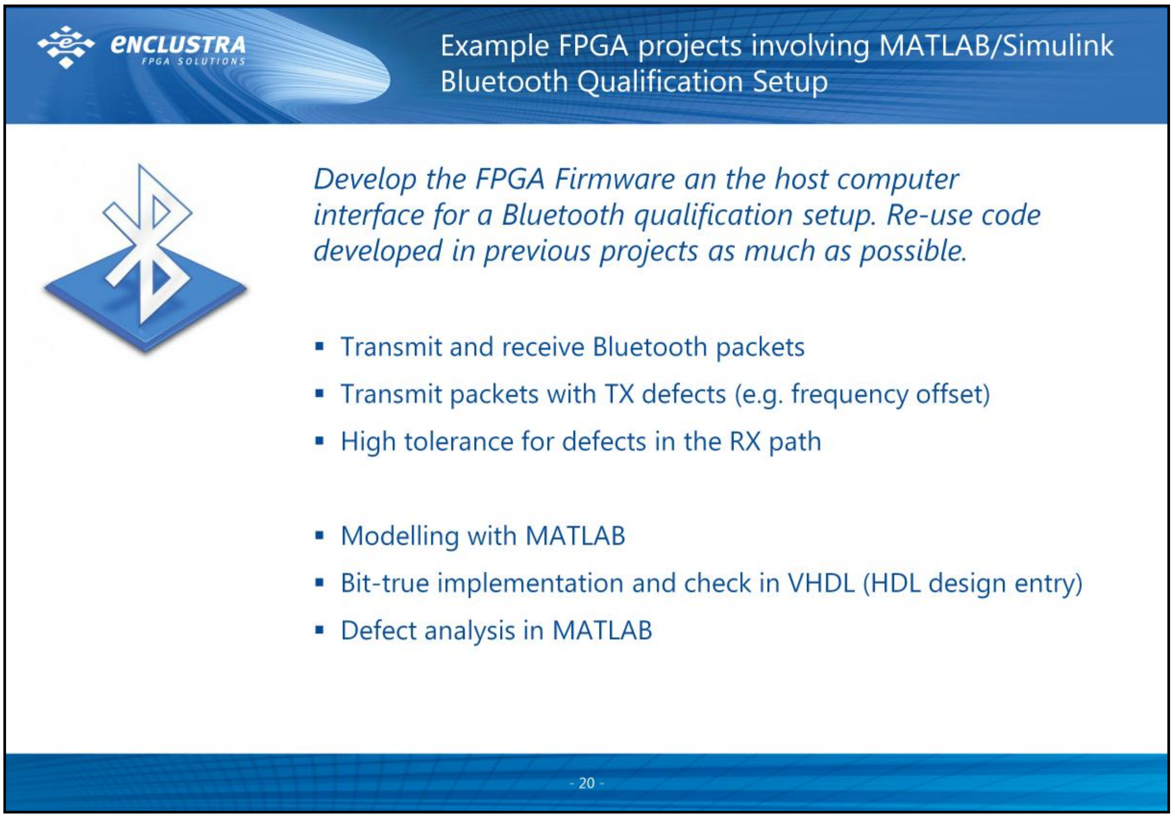


Example FPGA projects involving MATLAB/Simulink
Drive Control Platform

Using Simulink as an „executable specification language" for the proprietary control algorithms significantly reduced the risks of misunderstandings and misinterpretation.

Conclusion

Using Simulink as an „executable specification language" for the proprietary control algorithms significantly reduced the risks of misunderstandings and misinterpretation.



Example FPGA projects involving MATLAB/Simulink
Bluetooth Qualification Setup

Develop the FPGA Firmware an the host computer interface for a Bluetooth qualification setup. Re-use code developed in previous projects as much as possible.

- Transmit and receive Bluetooth packets
- Transmit packets with TX defects (e.g. frequency offset)
- High tolerance for defects in the RX path

- Modelling with MATLAB
- Bit-true implementation and check in VHDL (HDL design entry)
- Defect analysis in MATLAB

- 20 -

Starting Point

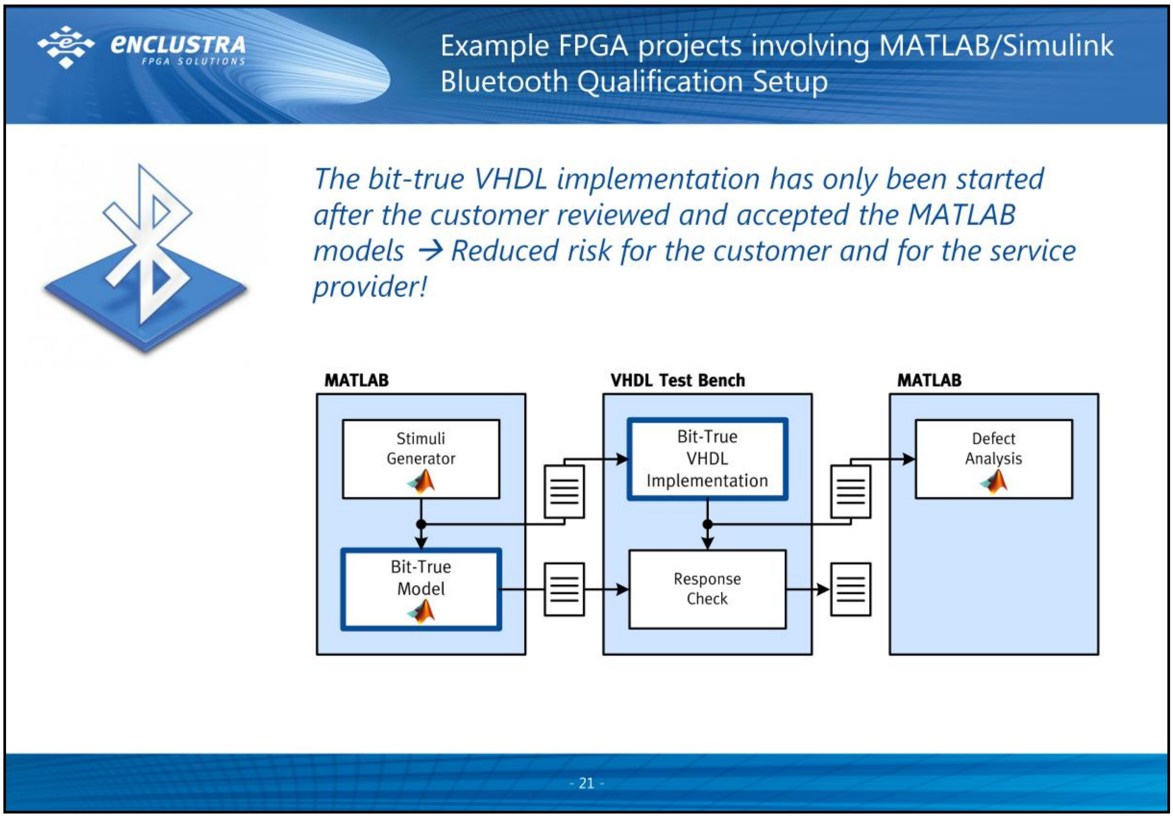
The customer plans to develop a Bluetooth qualification setup that should be developed further into a Bluetooth qualification product in a second step.

- We already completed previous Bluetooth-related projects for the same customer.
- Code developed during these previous projects should be reused as much as possible.
- Some parts of the datapath have to be designed and implemented from scratch.

Employment of MATLAB/Simulink

MATLAB has been used for creating and validating bit-true models, which the customer could review and approve. These models then served as bit-true specification for the FPGA implementation, which was done by manual HDL entry. HDL code generation from MATLAB has not been used, mainly because of the following reasons:

- A lot of already existing HDL code could be reused.
- Some of the required signal processing features were not well supported by the HDL code generator tools (e.g. runtime configurable fractional resampling)
- A majority of building blocks have already been availabke in Enclustra's fixed-point and bit-true libraries (en_cl_fix, en_cl_bittrue)
- Building blocks that were not already available were quite logic-heavy and thus not well suited for HDL code generation (e.g. Bluetooth packet engine).

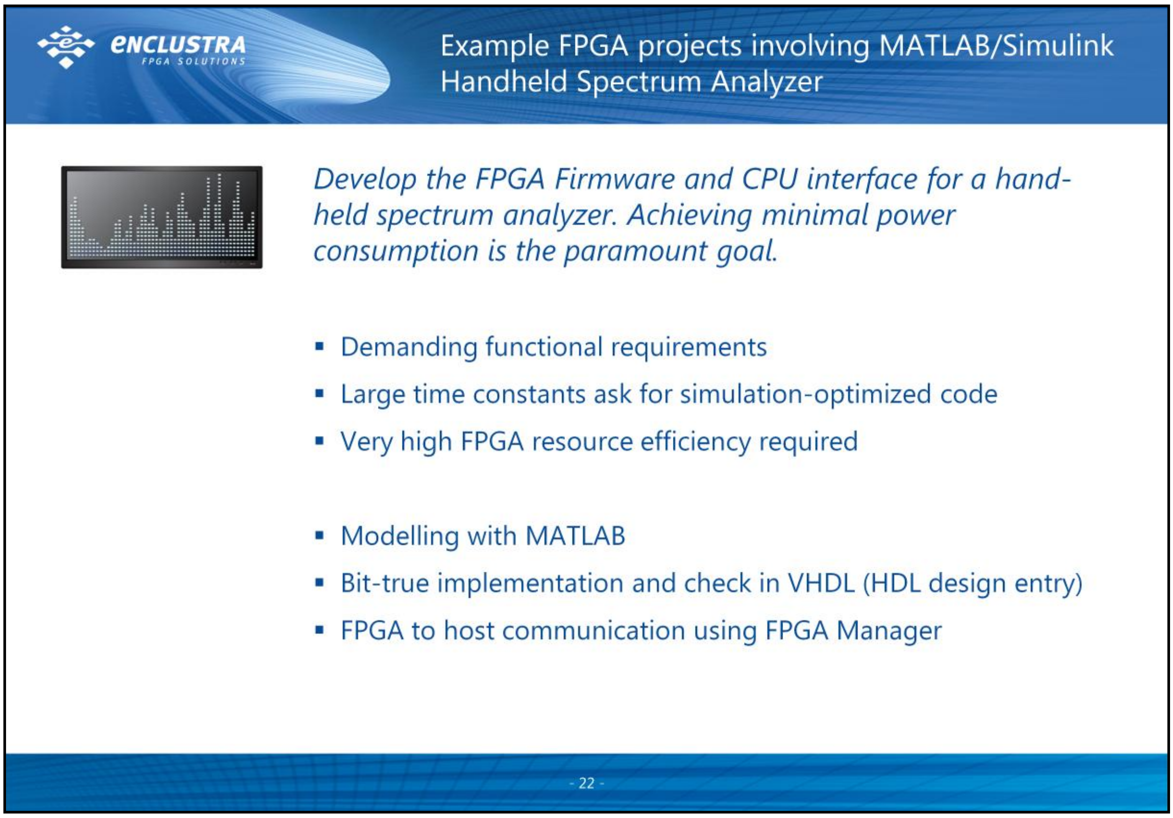


Example FPGA projects involving MATLAB/Simulink
Bluetooth Qualification Setup

The bit-true VHDL implementation has only been started after the customer reviewed and accepted the MATLAB models → Reduced risk for the customer and for the service provider!

Conclusion

The customer acceptance of the MATLAB models together with the bit-true implementation in VHDL greatly reduced the risk of misinterpreting the specification. There were virtually no design modifications during or after VHDL implementation.



Example FPGA projects involving MATLAB/Simulink
Handheld Spectrum Analyzer

Develop the FPGA Firmware and CPU interface for a hand-held spectrum analyzer. Achieving minimal power consumption is the paramount goal.

- Demanding functional requirements
- Large time constants ask for simulation-optimized code
- Very high FPGA resource efficiency required

- Modelling with MATLAB
- Bit-true implementation and check in VHDL (HDL design entry)
- FPGA to host communication using FPGA Manager

- 22 -

Starting Point

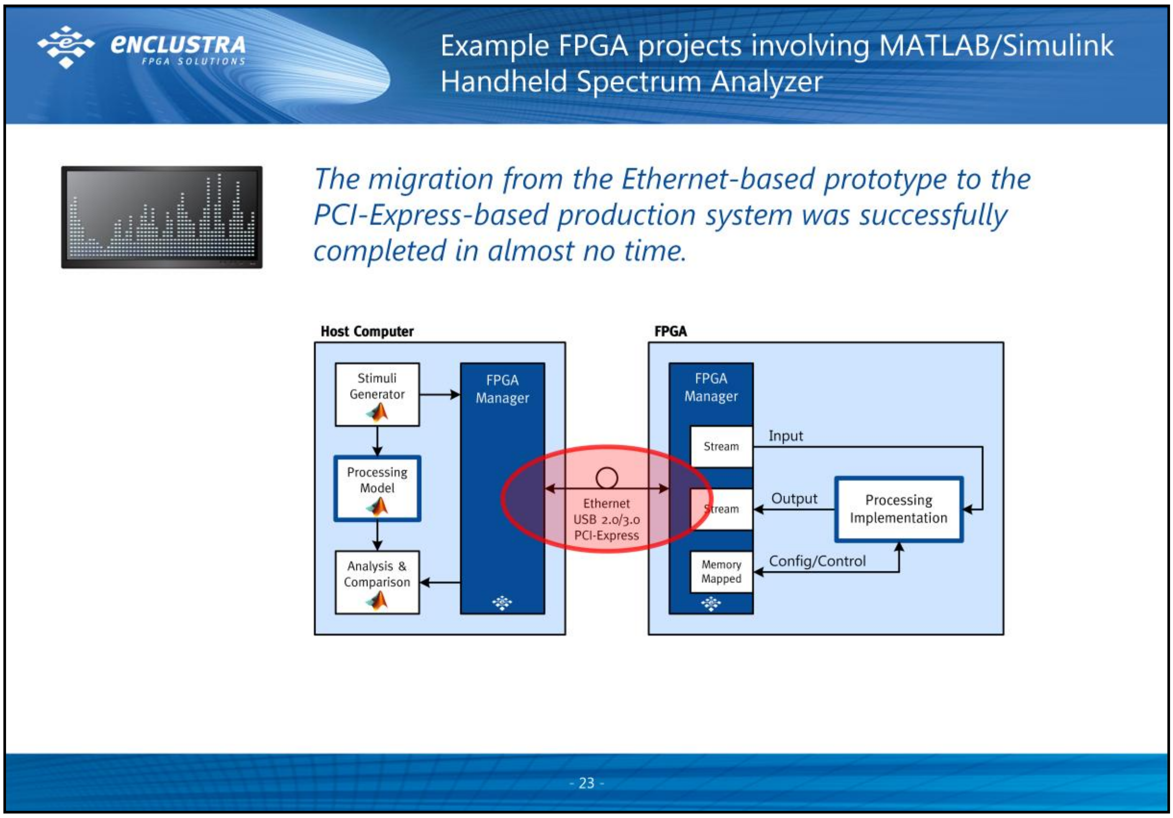
The customer plans to develop a handheld spectrum analyzer with a large set of features. Since the device runs from a battery, power consumption must be minimized.

- The first implementation shall, as a proof of concept, run on an FPGA evaluation board connected to the host computer via Ethernet.
- After proof of concept, a production system hardware shall be built that connects to an (embedded) host computer via PCI-Express.
- The migration from MATLAB to C# and from Ethernet to PCI-Express shall be as simple as possible.

Employment of MATLAB/Simulink

MATLAB has been used for creating and validating bit-true models, which the customer could review and approve. Due to the large time constants in the system, some critical MATLAB functions have been implemented in Java in order to speed up simulation. These models then served as bit-true specification for the FPGA implementation, which was done by manual HDL entry. HDL code generation from MATLAB has not been used, mainly because of the following reasons:

- Some of the required signal processing features were not well supported by the HDL code generator tools (e.g. runtime configurable fractional resampling)
- A majority of building blocks have already been available in Enclustra's fixed-point and bit-true libraries (en_cl_fix, en_cl_bittrue)
- Manually written HDL code can more easily be optimized for power consumption than generated code.

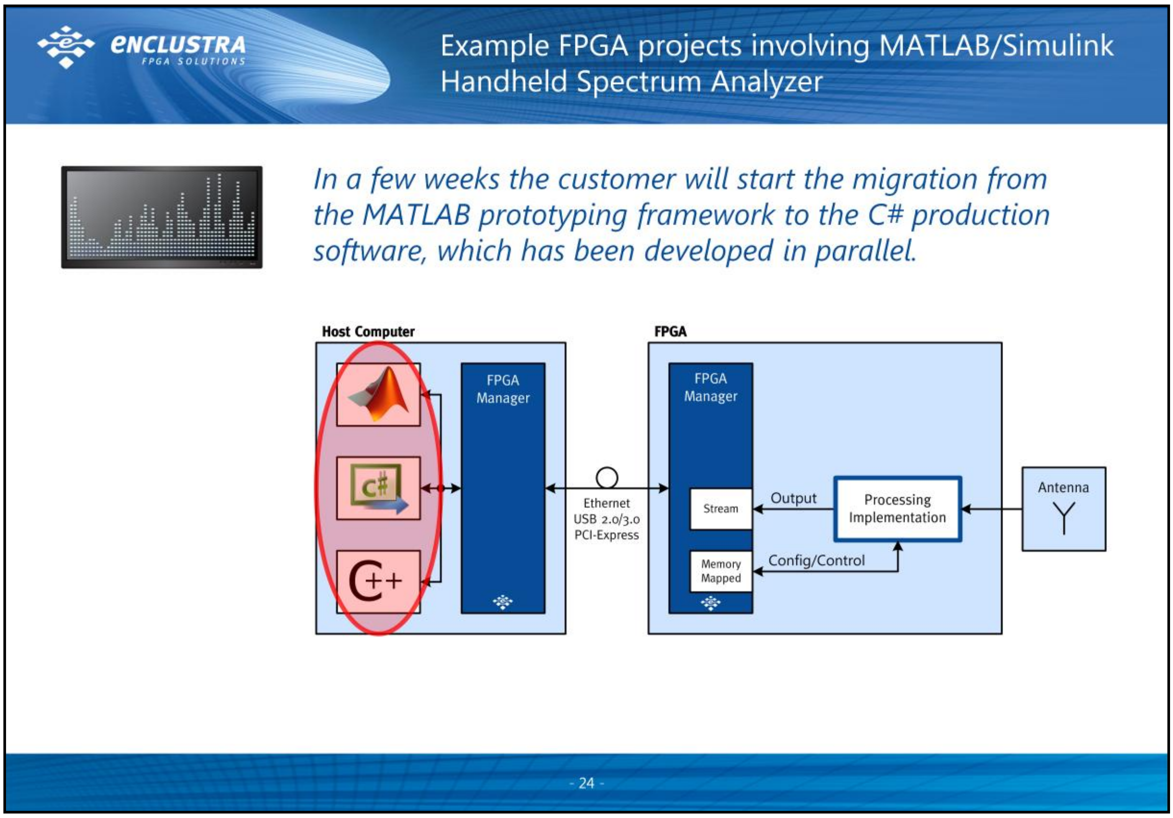


Example FPGA projects involving MATLAB/Simulink
Handheld Spectrum Analyzer

The migration from the Ethernet-based prototype to the PCI-Express-based production system was successfully completed in almost no time.

Conclusion

The combination of MATLAB and FPGA Manager over Ethernet provided exactly the flexibility and analysis capabilities that were required for the proof of concept phase. Thanks to FPGA Manager, the migration from Ethernet to PCI-Express was successfully completed with only very little efffort.

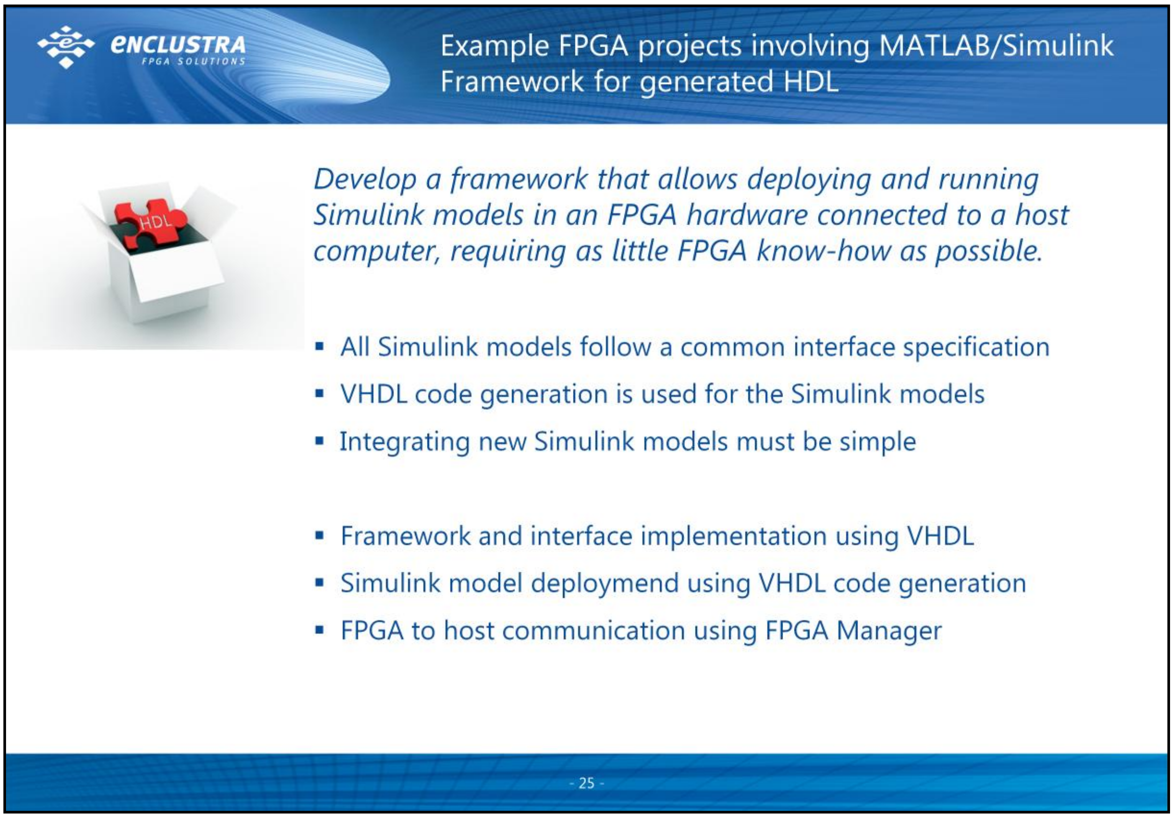


Example FPGA projects involving MATLAB/Simulink
Handheld Spectrum Analyzer

In a few weeks the customer will start the migration from the MATLAB prototyping framework to the C# production software, which has been developed in parallel.

Conclusion

The customer could already start implementation and test of the C# production software based on the FPGA evaluation board and an Ethernet connection, even before the production system hardware was available. This significantly reduces the effort required for bringing up the production system in a few weeks.



Example FPGA projects involving MATLAB/Simulink
Framework for generated HDL

Develop a framework that allows deploying and running Simulink models in an FPGA hardware connected to a host computer, requiring as little FPGA know-how as possible.

- All Simulink models follow a common interface specification
- VHDL code generation is used for the Simulink models
- Integrating new Simulink models must be simple

- Framework and interface implementation using VHDL
- Simulink model deploymend using VHDL code generation
- FPGA to host communication using FPGA Manager

- 25 -

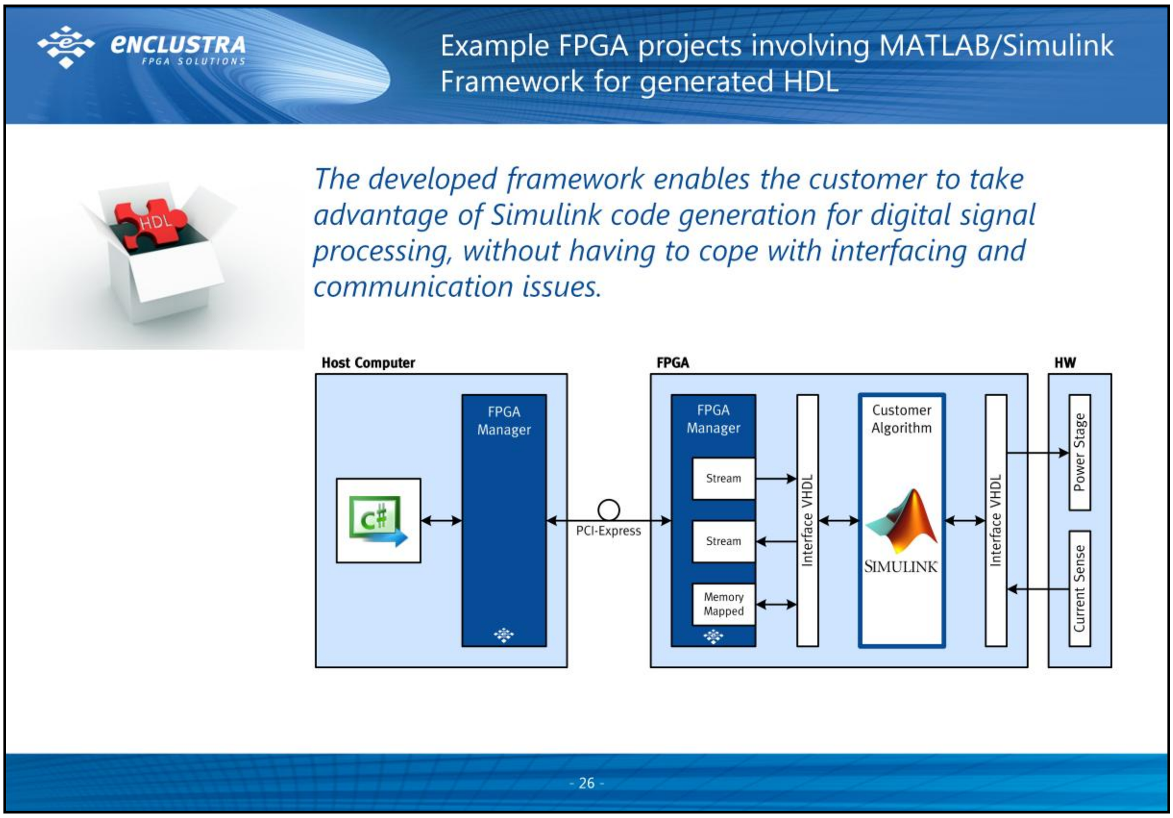
Starting Point

The customer plans to develop a system that can run Simulink models in an FPGA, communicating with a host computer and external power electronics.

- The customer has in-depth application knowledge, but has almost no expertise in FPGA design.
- All Simulink models to run in the FPGA share a common interface specification.
- Integrating new Simulink models must be as simple as possible.

Employment of MATLAB/Simulink

Simulink, together with Xilinx System Generator for DSP, is used for implementing the models and generating HDL. The communication and I/O framework has been implemented using manual HDL entry, FPGA Manager is used for FPGA to host communication. The Simulink hardware-in-the-loop (HIL) concept could not be used due specific project requirements.

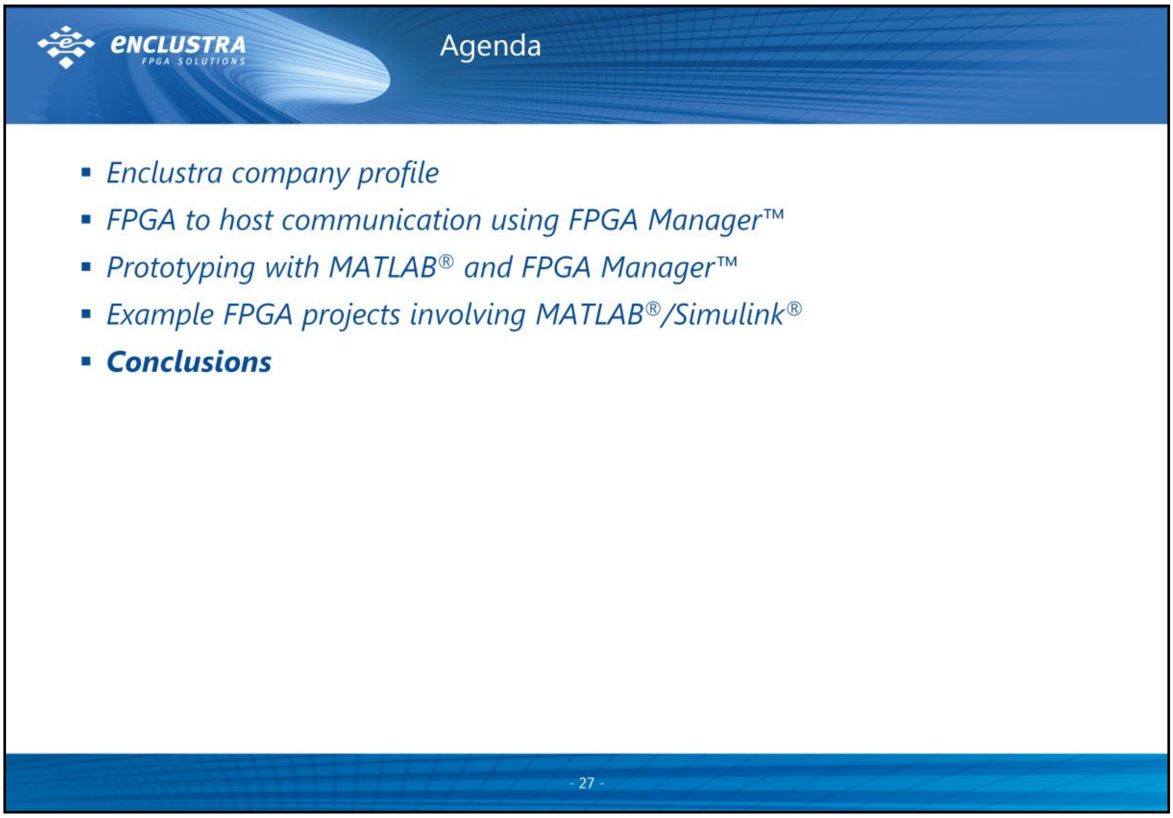


Example FPGA projects involving MATLAB/Simulink
Framework for generated HDL

The developed framework enables the customer to take advantage of Simulink code generation for digital signal processing, without having to cope with interfacing and communication issues.

Conclusion

The developed framework enables the customer to take advantage of Simulink code generation for digital signal processing, without having to cope with interfacing and communication issues.



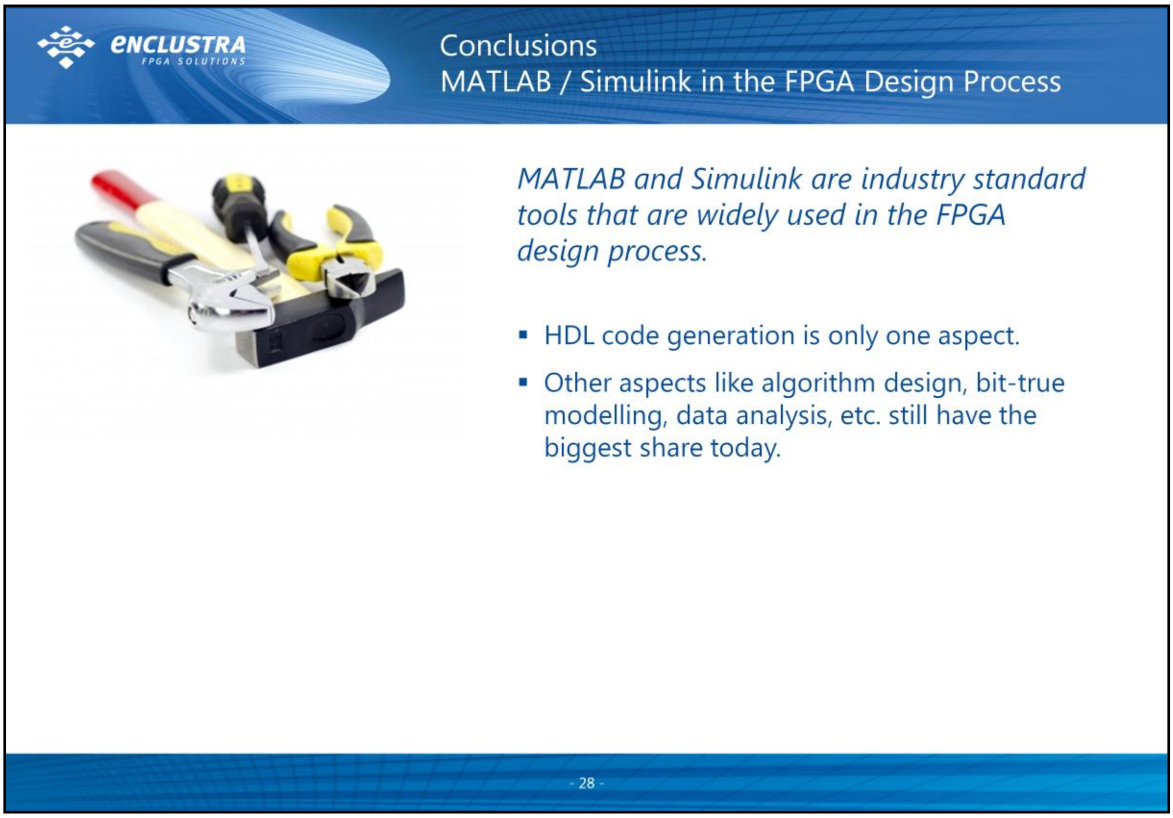
Agenda

- *Enclustra company profile*
- *FPGA to host communication using FPGA Manager™*
- *Prototyping with MATLAB® and FPGA Manager™*
- *Example FPGA projects involving MATLAB®/Simulink®*
- ***Conclusions***

- 27 -

Outlook

The next few slides give our main conclusions on how to successfully develop FPGA-based systems with the help of MATLAB, Simulink and FPGA Manager.



Conclusions
MATLAB / Simulink in the FPGA Design Process

MATLAB and Simulink are industry standard tools that are widely used in the FPGA design process.

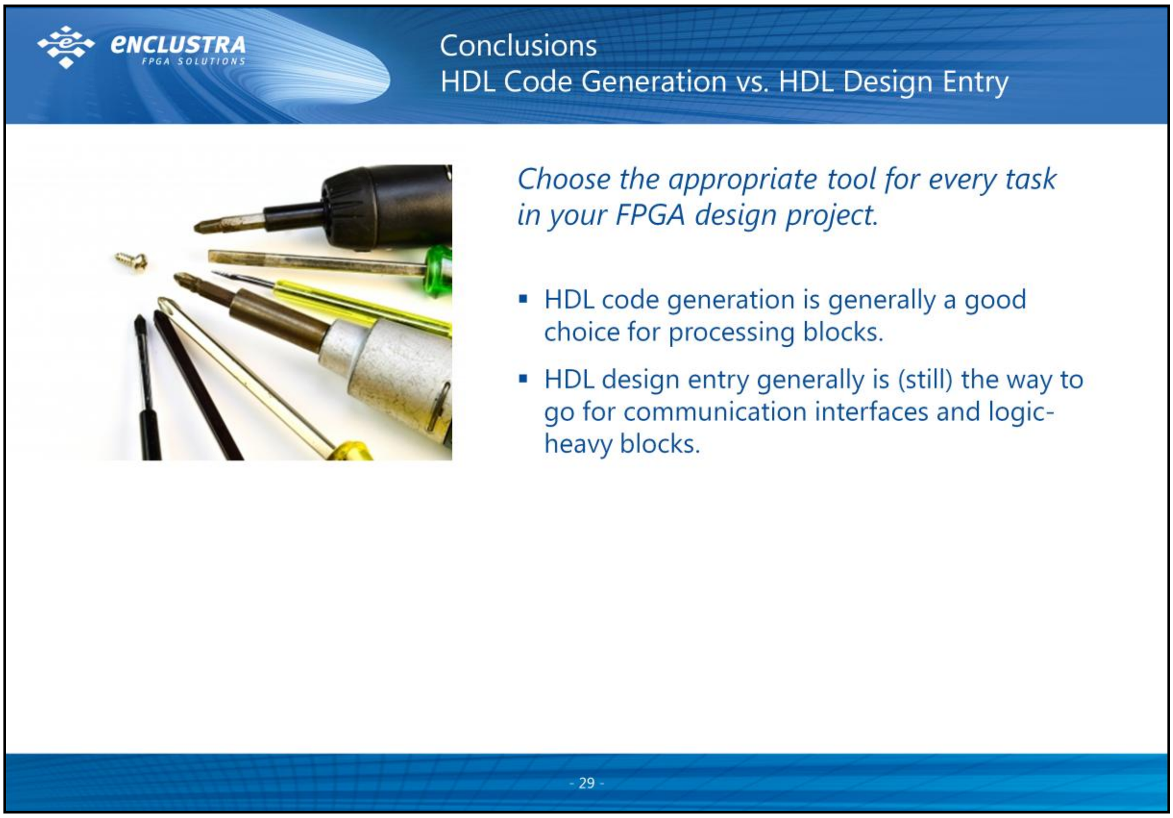
- HDL code generation is only one aspect.
- Other aspects like algorithm design, bit-true modelling, data analysis, etc. still have the biggest share today.

- 28 -

MATLAB/Simulink in the FPGA Design Process

Simulink and especially MATLAB are industry standard tools for signal processing and are widely used in the FPGA design process, mostly for tasks like algorithm design, bit-true modelling and data analysis.

HDL code generation is an interesting concept and the tools are evolving quickly. The practical use of these tools is however still dependent on the characteristics and properties of the individual project at hand.



Conclusions
HDL Code Generation vs. HDL Design Entry

Choose the appropriate tool for every task in your FPGA design project.

- HDL code generation is generally a good choice for processing blocks.
- HDL design entry generally is (still) the way to go for communication interfaces and logic-heavy blocks.

- 29 -

General

HDL code generation is very well suited for implementing signal processing data paths, while the most efficient way to describe interfaces and control logic still is direct HDL entry.

Properties favoring HDL Code Generation

- Streaming signal processing with fixed sample rates
- Time from MATLAB/Simulink to FPGA is a major concern
- HDL code traceability is a major concern

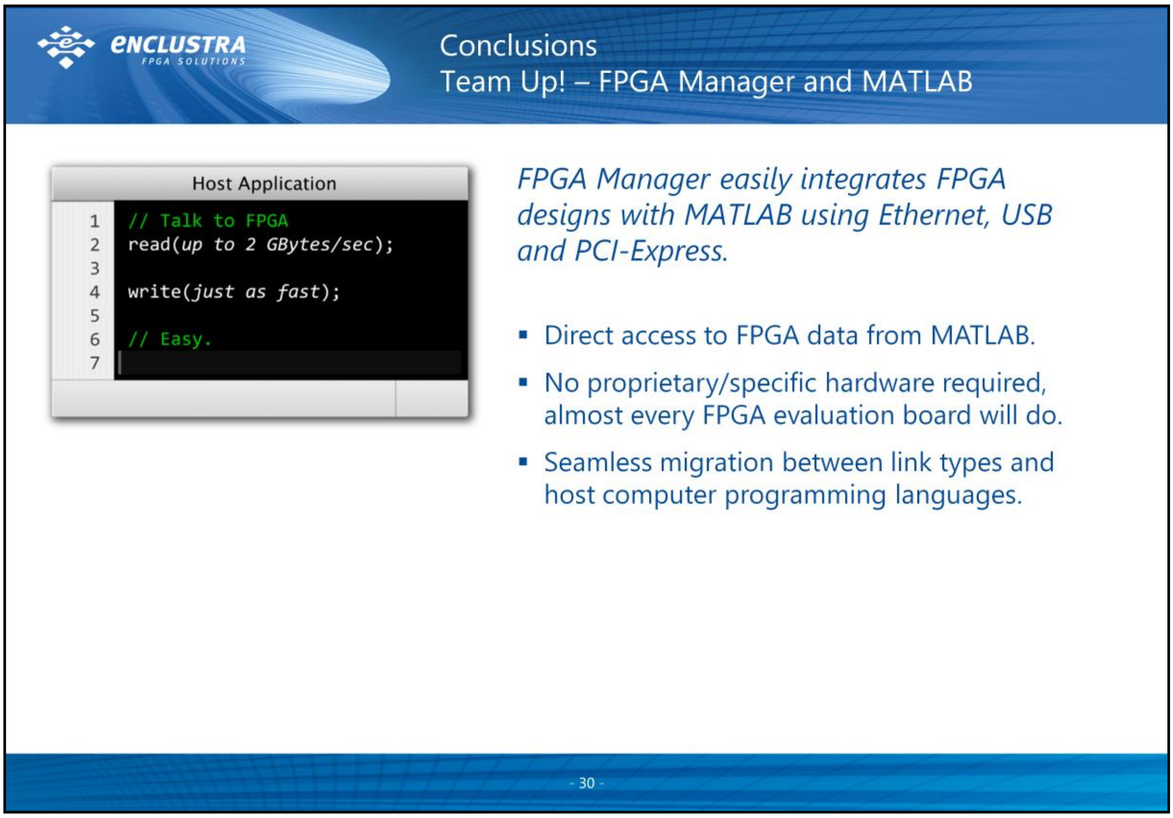
Properties favoring direct HDL entry

- Run-time configurable and/or fractional resampling
- Logic-heavy control or interface blocks
- Resource and power efficiency are major concerns

Bit-true Modeling as the Happy Medium

The use of bit-true models together with appropriate HDL test benches offers many benefits of HDL code generation while avoiding its disadvantages.

- Exact simulation already early in the development process
- Automated validation against a golden model



Conclusions
Team Up! – FPGA Manager and MATLAB

Host Application

```
// Talk to FPGA
read(up to 2 GBytes/sec);

write(just as fast);

// Easy.
```

FPGA Manager easily integrates FPGA designs with MATLAB using Ethernet, USB and PCI-Express.

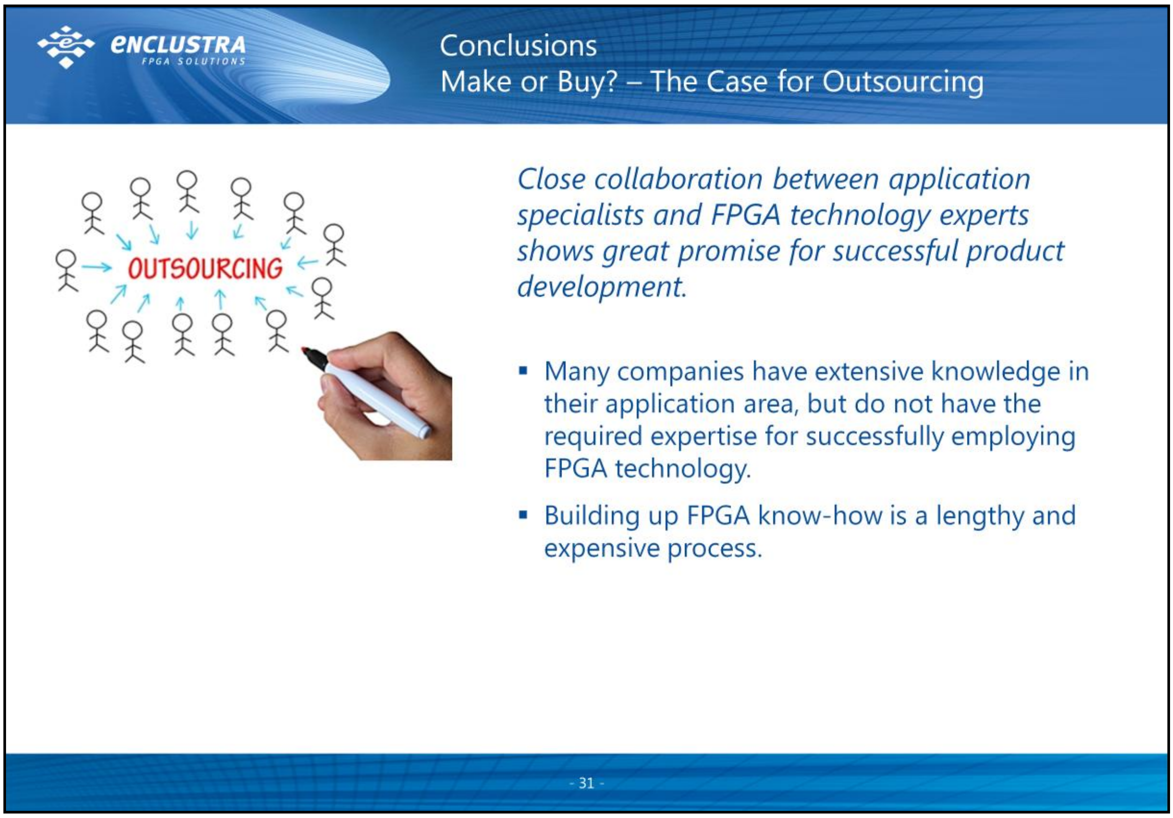
- Direct access to FPGA data from MATLAB.
- No proprietary/specific hardware required, almost every FPGA evaluation board will do.
- Seamless migration between link types and host computer programming languages.

- 30 -

Team Up! – FPGA Manager and MATLAB

FPGA Manager brings direct access to the FPGA to MATLAB, resulting in a very powerful and flexible setup for developing and prototyping FPGA-based systems, without having to buy any specific or proprietary hardware.

When advancing your prototyping platform to a production system, the smooth migration between individual link types and host software programming languages significantly reduces time to market.


OUTSOURCING

Conclusions
Make or Buy? – The Case for Outsourcing

Close collaboration between application specialists and FPGA technology experts shows great promise for successful product development.

- 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.

- 31 -

Make or Buy? – The Case for Outsourcing FPGA Development

Successful and efficient FPGA development 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. Furthermore, 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.

Questions?

Marc Oberholzer
Vice President, Engineering
marc.oberholzer@enclustra.com
+41 43 343 39 47

Quarterly newsletter:
subscribe@enclustra.com

Upcoming Events:

- Vision
  November 4-6, 2014
  Messe Stuttgart
  Germany
- SPS IPC Drives
  November 25-27, 2014
  Messezentrum Nürnberg
  Germany
- Embedded World
  February 24-26, 2015
  Messezentrum Nürnberg
  Germany

Image References

- Slide 4 FPGA image and Xilinx Alliance Program logo courtesy of xilinx.com
- Slide 4 Altera DSN logo courtesy of www.altera.com
- Slide 4 Lattice LEADER logo courtesy of latticesemi.com
- Slide 4 Microsemi logo courtesy of microsemi.com
- Slide 4 map pin image courtesy of C at clker.com
- Slide 5 left image courtesy of ddpavumba at FreeDigitalPhotos.net
- Slide 6 left image courtesy of digitalart at FreeDigitalPhotos.net
- Slides 18 and 19 left image courtesy of Master isolated images at FreeDigitalPhotos.net
- Slides 20 and 21 left image courtesy of Stefan Holliland at all-free-download.com
- Slides 22, 23 and 24 left image courtesy of Ohmega1982 at FreeDigitalPhotos.net
- Slides 25 and 26 left image courtesy of jscreationzs at FreeDigitalPhotos.net
- Slide 28 left image courtesy of dan at FreeDigitalPhotos.net
- Slide 29 left image courtesy of cjansuebsri at FreeDigitalPhotos.net
- Slide 31 left image courtesy of Stuart Miles at FreeDigitalPhotos.net
- Slide 32 Vision logo courtesy of www.messe-stuttgart.de
- Slide 32 sps ipc drives logo courtesy of www.mesago.de
- Slide 32 embedded world logo courtesy of www.embedded-world.de
- MATLAB logo courtesy of The MathWorks, Inc.
- All other images courtesy of Enclustra GmbH

- 33 -