SoC Systeme ultra-schnell entwickeln mit Vivado und Visual System Integrator

Embedded Computing Conference 2017

Matthias Frei | zhaW InES | Patrick Müller | Enclustra GmbH | 5 September 2017
Agenda

- Enclustra introduction
- What is Visual System Integrator?
  - Introduction
  - Platform
  - System
Enclustra Company Profile

Focused on FPGA Technology – Everything FPGA!

Founded in 2004 – successfully in business for 13 years!

Headquarters in Zürich, Switzerland

Branch offices in Germany, USA and China

2 Business units: FPGA Design Center, FPGA Solution Center

30 employees: 15 FPGA engineers plus 11 staff in Zurich, 4 employees abroad

Vendor-Independent
Enclustra Company Profile

- FPGA Design Center
  - Hardware (High-Speed, Analog, RF)
  - HDL firmware (VHDL, Verilog)
  - Embedded software (for FPGA processors)
- FPGA Solution Center
  - IP-Cores
  - FPGA & SoC Modules

FPGA? Enclustra!
The Problem

Embedded Systems
Getting Increasingly Complex

System Design Span Multiple Chips
(CPUs, GPUs, FPGAs, DSPs ...)

Difficult to Integrate, No Visibility

No Universal Tool Exists for Integration
The Solution: Visual System Integrator

Rapid, visual application development:

- Describe the hardware platform.
- Develop the application by importing C/C++/RTL blocks and interconnecting them.
- Automatic code generation for the complete system: software/hardware projects, drivers, DMAs.

Get unprecedented transaction-level visibility at runtime through the trace function.

Focus on your application, not the platform & firmware!
Visual System Integrator: Work Flow

- Describe Hardware Platform (Import Existing)
- Compile Platform
- Generate System
- Import Platform
- Platform Meta data

- Develop Application System

- Compile

- Software Projects: Eclipse, Qt, VC++, ...
- Runtime, Drivers & OS Configurations...
- FPGA Projects
Development Context

**Platform**
- Defines what Chips/Devices are used
- Defines how these are linked
- Builds the basis of your system/application
  - Can be reused for different applications

**System**
- The place where you actually develop your application/system
- No need to think about links between devices
Application Development: Example

Import Platform
- TCP SERVER
- User C/C++ Code
- Synthesizable C/C++ Function

Develop Application
- DSP Function
- Synthesizable C/C++ Function

Software Library
- X86
- FPGA
- C/C++ Functions
- FPGA Library
Platform

- Basic structure of your project
- Use existing templates
- Reusable easily
MPSoC + X86 Platform

- Contexts
- Connectivity
  - Associated IP & drivers
- Additional Memory and/or IO
Supported Platforms

- Xilinx FPGA and MPSoC
  - Zynq 7000
  - Ultrascalse+
  - Ultrascalse
  - Artix-7
  - Kintex-7

- CPUs
  - ARM
  - ARM64
  - Linux x86

- Roadmap
  - Windows x86

- Also runs in the Amazon EC2 F1 cloud!
Interconnectivity

- AXI
  - FPGA
- Ethernet (TCP/IP)
- PCIe
- Shared Memory
- The place where your applications is developed.
- Add blocks using drag-and-drop.
- Change execution context of a block easily.
MPSoC + X86 Application

- Application
- TCP Server
- X86 (Linux)
- Compute Block
- UltraScale FPGA Fabric
- timer
- 1 Second
- Timer Driven Command
- converter
- control
- I/O Driver
- Cortex-R5 (FreeRTOS)
- Cortex A53 (Linux)
Supported Languages for CPUs

- C/C++
- Java
- Python
Supported Languages for FPGA/SoC

- Xilinx IPs
- Synthesizable C/C++ Code
- Custom VHDL/Verilog Blocks
  - AXI memory mapped
  - AXI streaming interface
What VSI does for you

- Generating FPGA Bitstream
- Generating Software executables
- Creating the linux driver
  - with device-tree entry if necessary
- Script to load the driver with parameters
Pay attention on how to distribute processing across platforms
  - Performance
  - Resources

Blocks can be moved per drag-and-drop

Measure the performance with Built-in tool
VSI: System Design Life Cycle

**System Design**
- Define “Abstract Platform”
- Create Blocks with interfaces
- Represent external interfaces

**Unit Design**
- Formalize Platform
- Groups are assigned to tasks
- External dependencies formalized

**Code**
- Platform becomes concrete
- Block designs are completed
- External interfaces frozen

**System Testing**
- Functional system integration
- Performance analysis

**Unit / Subsystem Testing**
- Blocks moved across “Contexts”
- Functional Debug, Co-Simulation
- Performance analysis
- Stimulus generation
Current Status:

- Release 1.0 (2017.1)
- User Interface
  - Platform canvas
  - System canvas
- Run Time & Transports
- Road Map
  - Partial Reconfiguration
HAVING A DIFFICULT TIME WITH FPGA/EMBEDDED SYSTEM INTEGRATION?

VISUAL SYSTEM INTEGRATOR MAKES IT SIMPLE TO INTEGRATE FPGA AND EMBEDDED SYSTEMS.
Evaluate Visual System Integrator today!

90 days for free!

https://systemviewinc.com/license.html
Everything FPGA.