FPGA-Based Hardware Accelerators for 10/40 GigE TCP/IP and Other Protocols

Dr. Endric Schubert, Univ. Ulm / Missing Link Electronics
- Ulrich Langenbach, Fraunhofer Heinrich-Hertz-Institute

We are
a Silicon Valley based technology company with offices in Germany. We are partner of leading electronic device and solution providers and have been enabling key innovators in the automotive, industrial, test & measurement markets to build better Embedded Systems, faster.

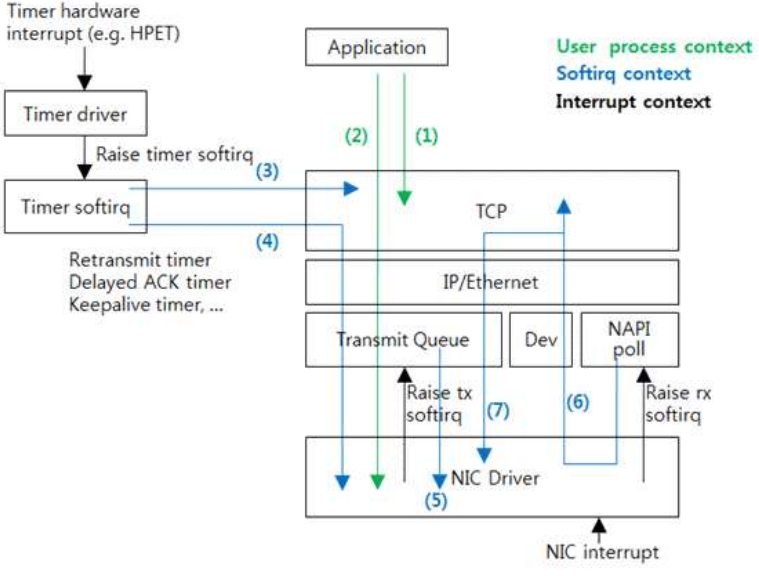
Our Mission is
To develop and market technology solutions for Embedded Systems Realization via pre-validated IP and expert application support, and to combine off-the-shelf FPGA devices with Open-Source Software for dependable, configurable Embedded System platforms

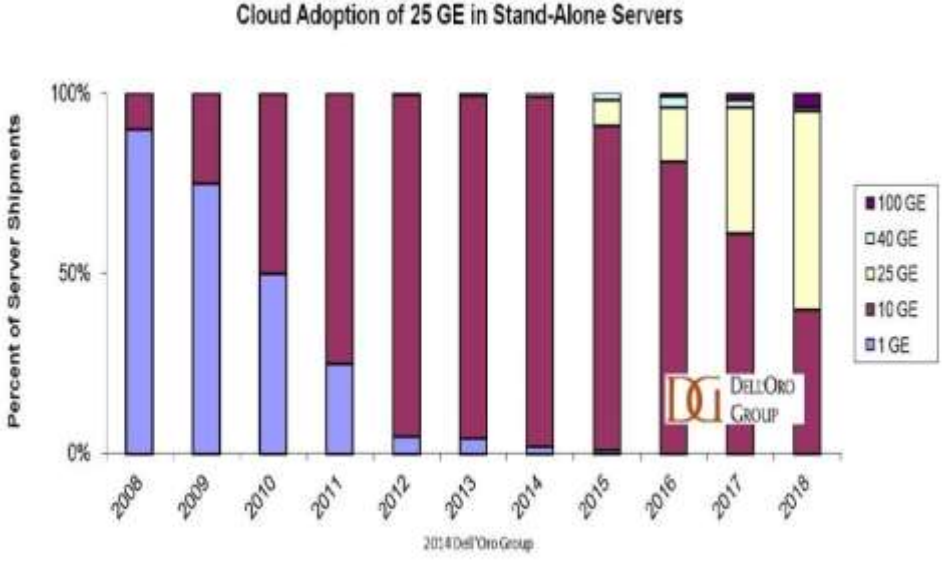
Our Expertise is
I/O connectivity and acceleration of data communication protocols, additionally opening up FPGA technology for analog applications, and the integration and optimization of Open Source Linux and Android software stacks on modern extensible processing architectures.

Network Processing at 10 GigE has a Huge Compute Burden …

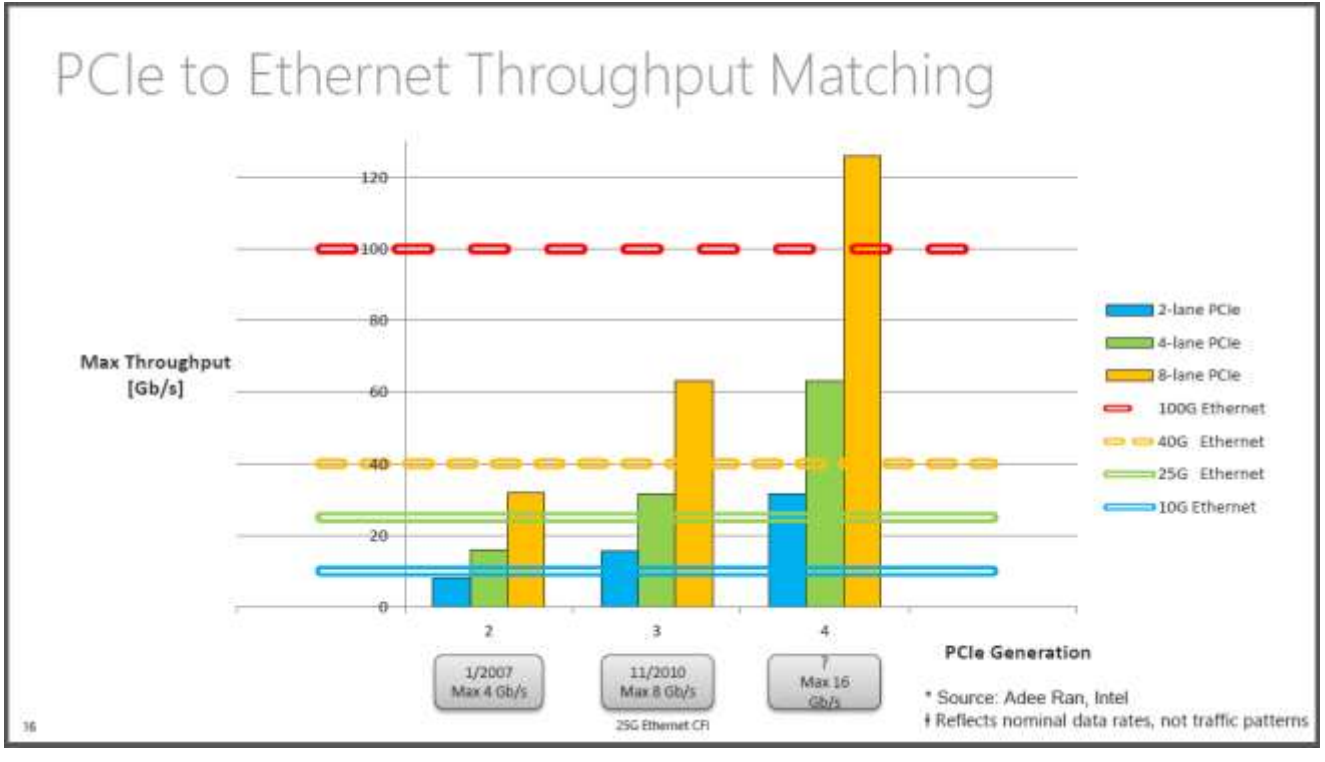
Transporting 1 bit per second needs 1 Hz

- 1 GigE ➔ 1 CPU at 1 GHz
- 10 GigE ➔ 4 CPUs at 2.5 GHz





… and soon we will see 25 GigE



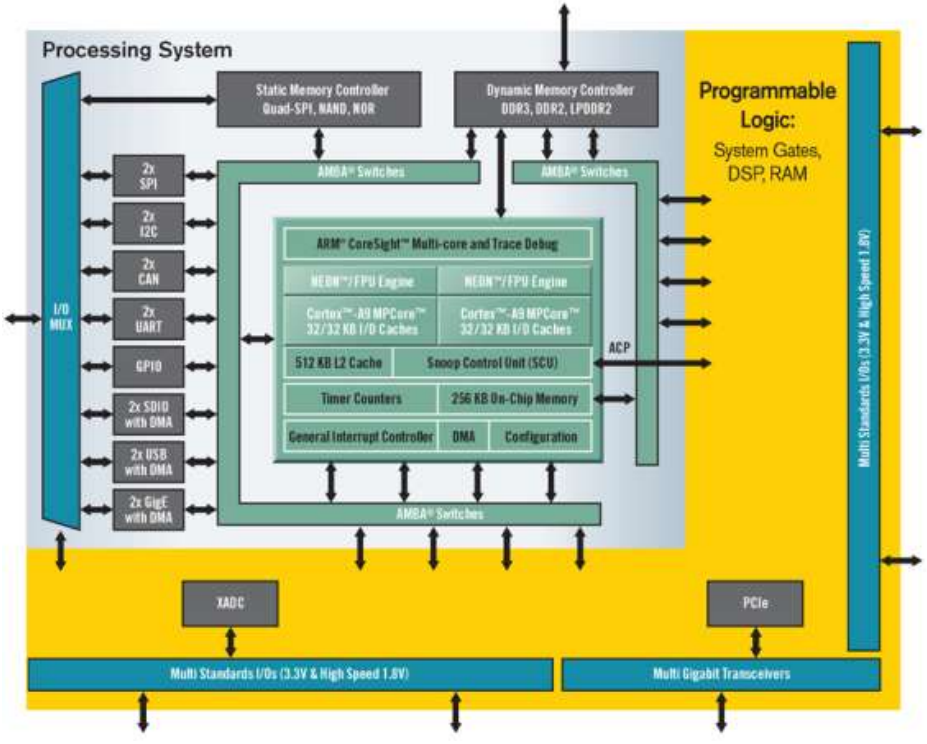
Design Choices for Network Processing in SoC FPGAs

SoC FPGA as (yet) another computer

	Intel i7-4770	Xilinx Zynq 7045
Compute	~100 GFLOPS	5 GFLOPS (PS) 778 GFLOPS (PL)
TDP	84 W	<20 W (typ)

SOC FPGA has 4x more compute

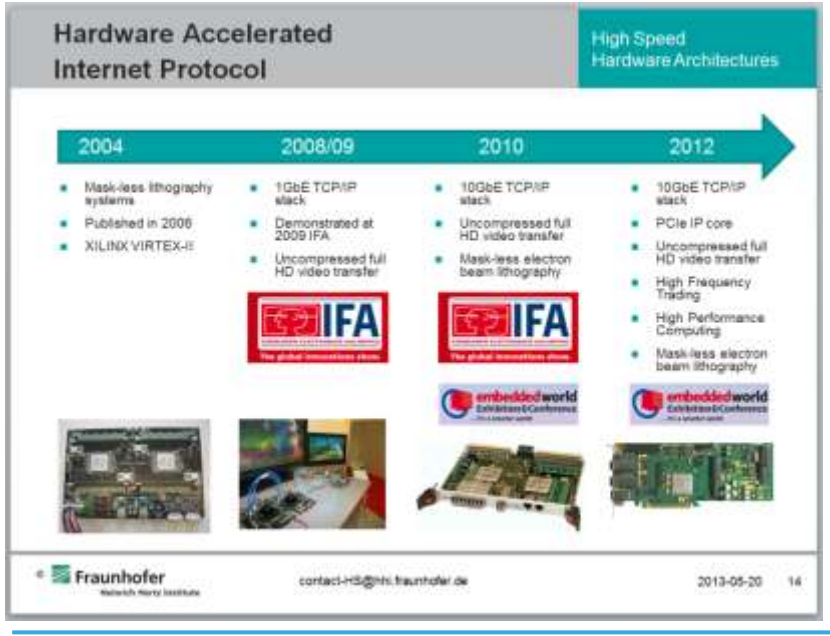
With ¼ the power dissipation!

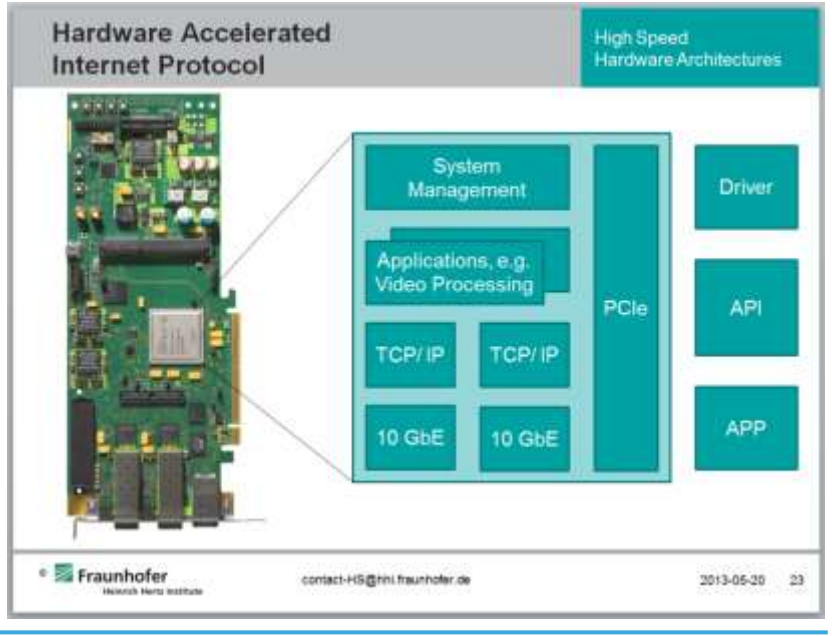


[http://www.xilinx.com/products/technology/dsp.html]

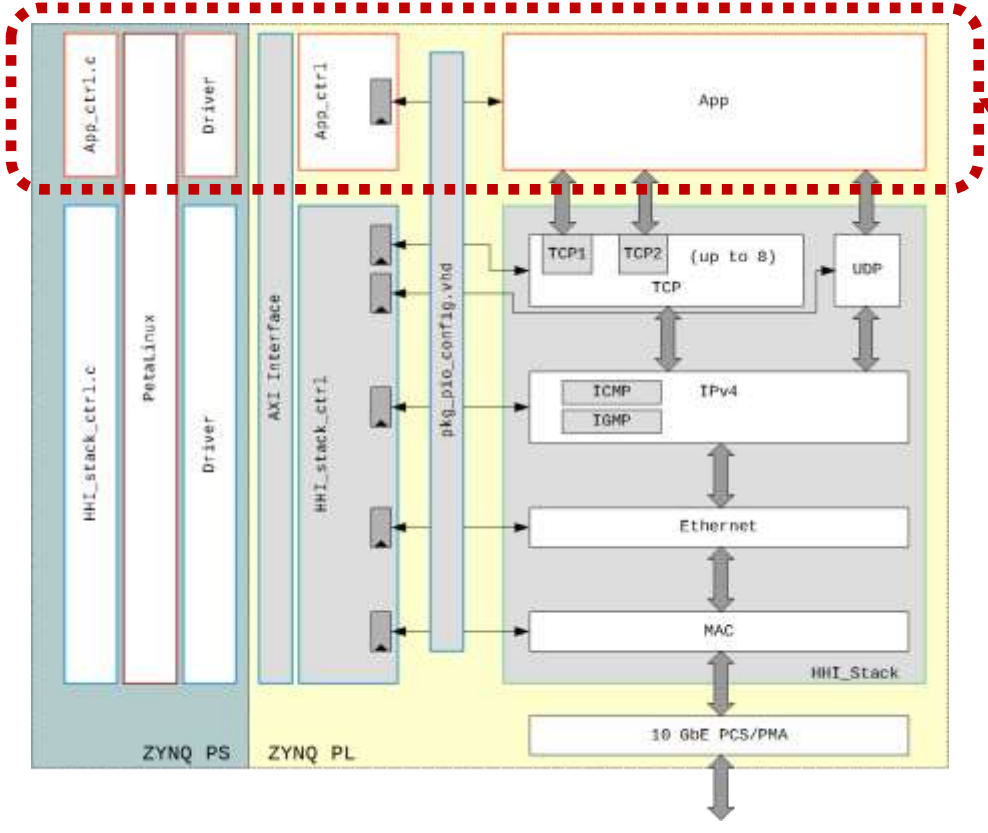
Network Stack in RTL from Fraunhofer Heinrich-Hertz-Institute

- Brings full TCP/UDP/IP connectivity to FPGAs even when there is no CPU available. Accelerate CPUs by offloading TCP/UDP/IP processing into programmable logic.





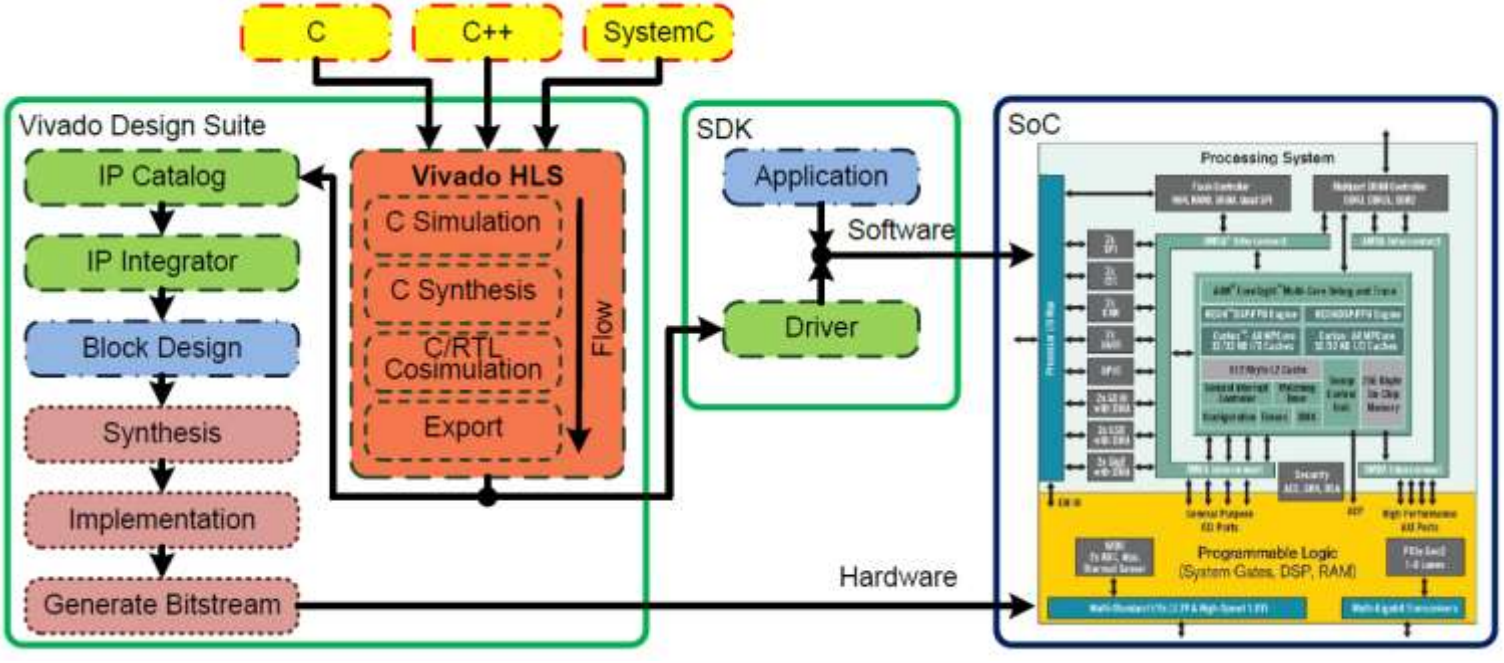
Network Protocol Acceleration Platform Architecture



Network protocol processing at application layer (ISO Layer 7) can more efficiently be implemented via a programming approach (in C or C++) than by digital circuit design (in VHDL or Verilog).

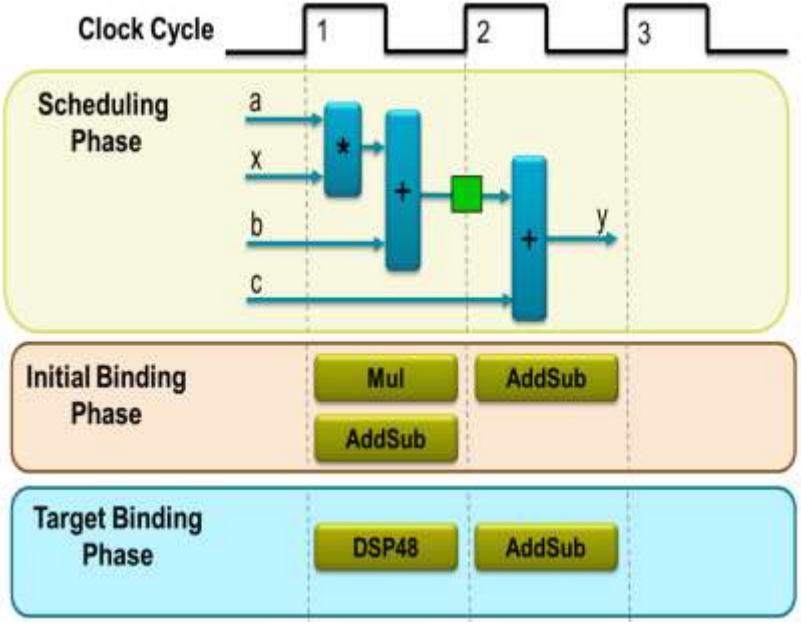
High-Level Synthesis Design Flow for SoC FPGA

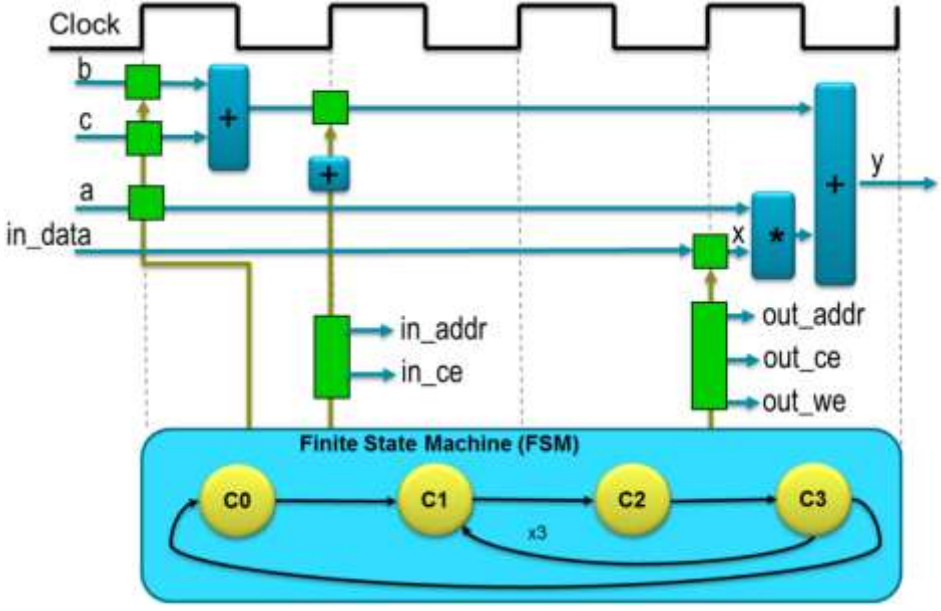
- Input C/C++/SystemC into High-Level Synthesis to generate VHDL/Verilog code



Working Principles of High-Level Synthesis

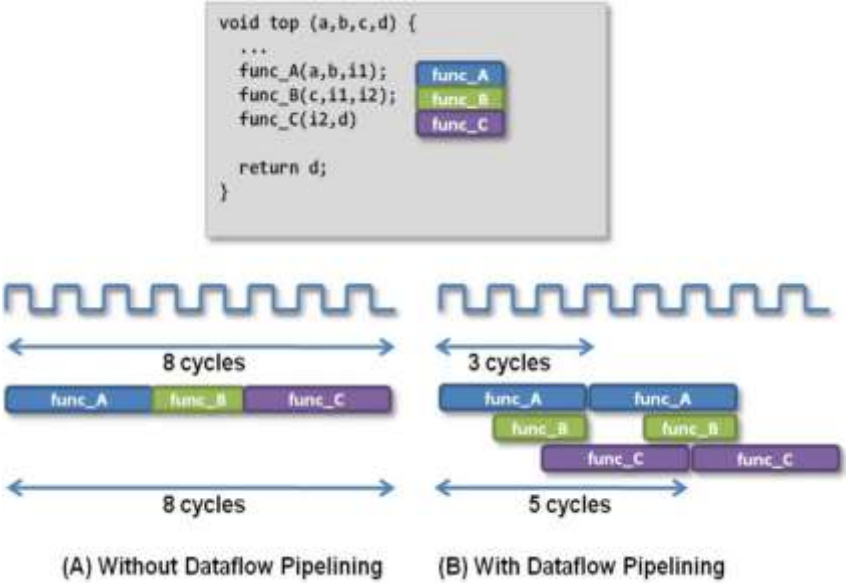
- Design automation runs scheduling and resource allocation to generate RTL code comprising data path plus state machines for control.





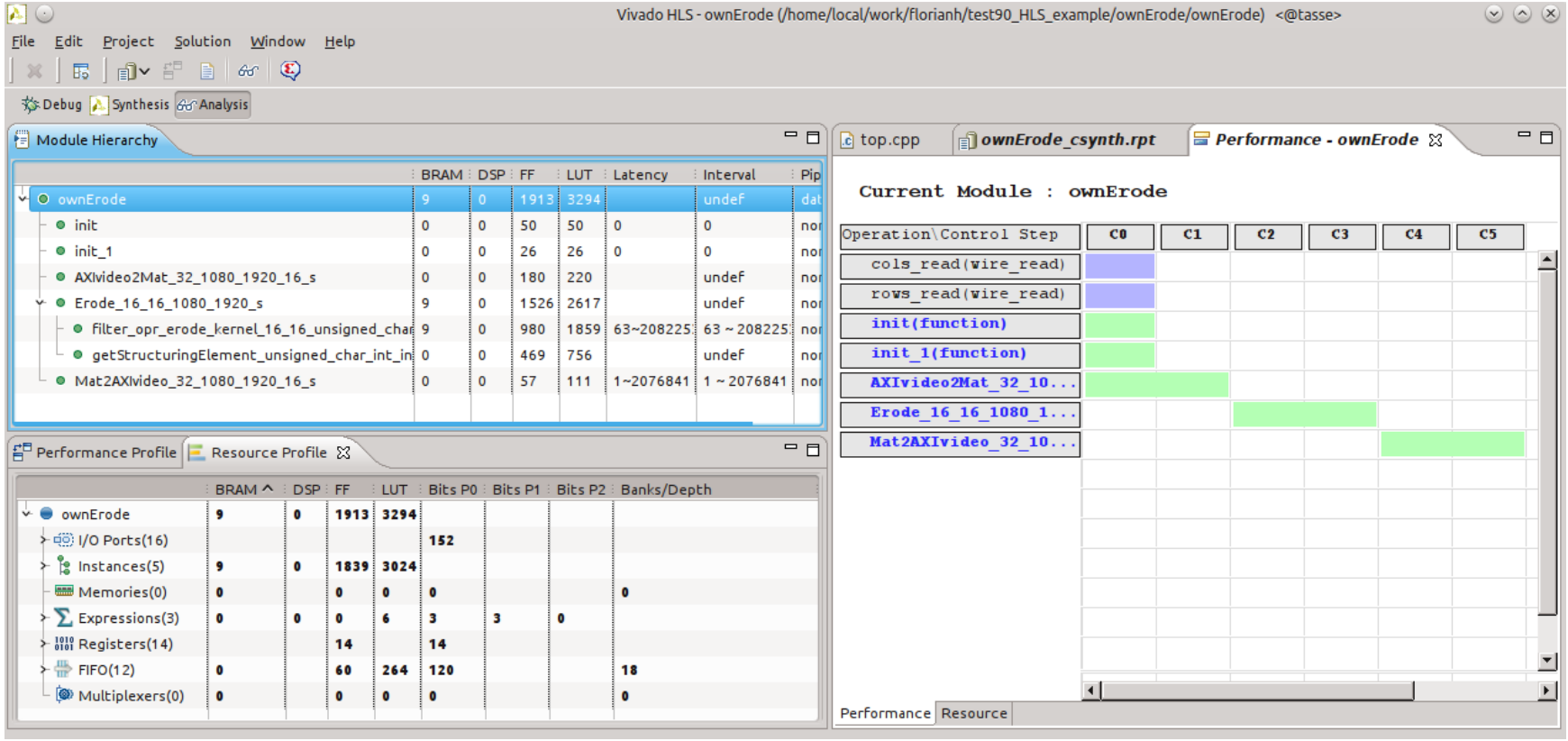
Benefits of High-Level Synthesis

- Automatic performance optimization via parallelization at dataflow level



- Automatic interface synthesis and code generation for variety of real-life HW/SW connectivity

Visualization and User Interaction in High-Level Synthesis Tool



Conclusion and References

- Significant productivity increase for protocol oriented or dataflow based design blocks.
- Easy to adopt: Known languages C/C++ combined with known tool chain.
- → Add this to your bag of tricks!

- UG998 - Introduction to FPGA Design Using High-Level Synthesis
- UG871 - Vivado Design Suite Tutorial: High-Level Synthesis
- XAPP1209 - Designing Protocol Processing Systems with Vivado High-Level Synthesis
- UG949 - UltraFast Design Methodology Guide for the Vivado Design Suite

Contact Information

Dr. Endric Schubert
endric.schubert@uni-ulm.de
endric@MLEcorp.com
Phone US: +1 (408) 320-6139
Phone DE: +49 (731) 141149-66