Keynote

Towards FPGAs as Platforms for Intelligent Systems and Applications

Peter Rössler, University of Applied Sciences Technikum Wien
roessler@technikum-wien.at

11th IARIA INTELLI 2022, May 22-26, 2022, Venice, Italy

About Myself

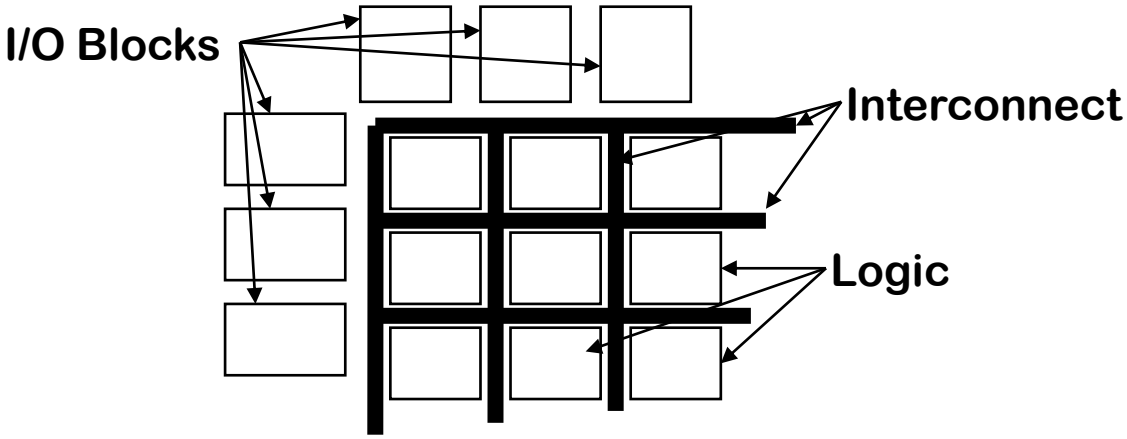
- Born in Vienna, Austria
- Studied Electrical Engineering, received diploma and doctoral degree from the Vienna University of Technology in 1996 and 2002, resp.
- Currently working as a Prof. (FH) at the Dept. of Electronics Engineering, University of Applied Sciences Technikum Wien, Austria
- Research fields: Microelectronics (with a focus on FPGAs and digital ASIC design) and Embedded Systems (in various application areas)
- Author or currently 70+ publications, member of several program committees of national & international scientific conferences, engaged in boards of national organizations like the Austrian Association of Electronic Engineering (OVE), Chair of the IEEE Austrian Section from 2012-2014

Outline of Presentation

- Basic idea & motivation for FPGAs
- History
- FPGA details/technology
- FPGA markets, players and design tools
- Pros and cons in comparison to other technologies (ASIC, CPU, MCU, GPU …)
- **Examples & applications**
- Future trends and challenges in the area of FPGA design

FPGA – What's this?

- The acronym says all …
  - **"Field-Programmable Gate Array"**
  - A number (actually millions) of gates that can be programmed in the field
- **PLD (Programmable Logic Device)** is another frequently (more traditionally) used term (to be precise, FPGAs are a special type of PLDs)
- An integrated circuit containing configurable logic and/or storage elements, which are linked together using a programmable interconnect
- In general, the following resources can be distinguished
  - Logic
  - Interconnect
  - I/O



Idea of FPGAs

- One or more resources are configurable/programmable
  - One-time programmable
  - Reprogrammable
  - In-system programmable
- Today, all resources (logic, I/O and interconnect) are configurable
- Over the years other types of resources have been added to the devices, like local SRAM, PLLs, DSP blocks (hardware multipliers), CPU cores … in order to provide platforms which are suitable to implement huge & really complex hardware/software systems

FPGA – Pros and Cons

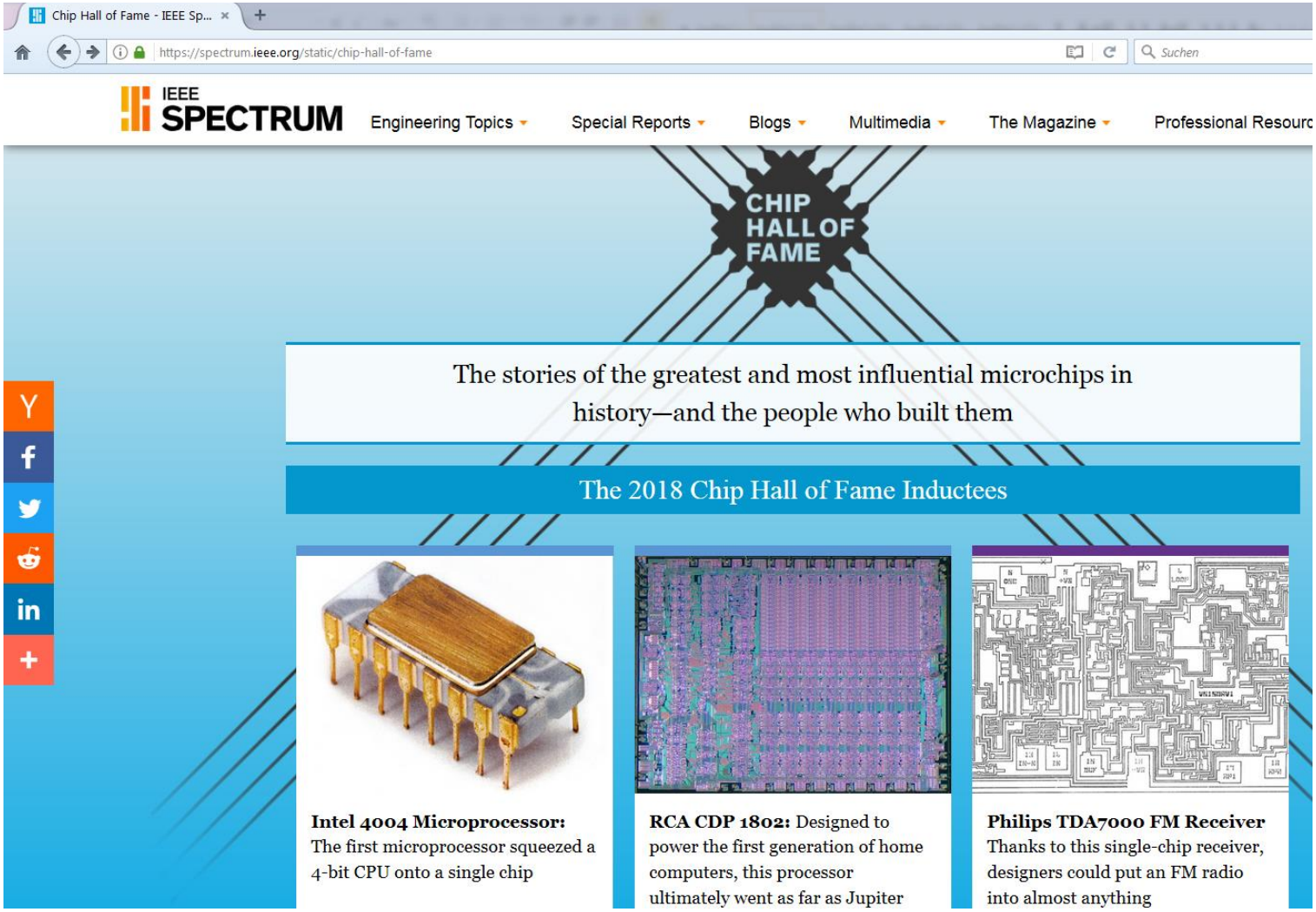
- **The programmability offers**
  - Short development times
  - Short turnaround times
  - Rapid prototyping
  - Flexibility with respect to engineering change orders
  - Option to save printed circuit board (PCB) space
  - Small/Zero one-time (NRE – Non-recurrent Efforts) costs for customers

- **Drawbacks of FPGAs**
  - Higher device costs (when compared to other technologies)
  - Limited clock frequencies ➔ Can be compensated by parallelization of computation tasks
  - (High power consumption) ➔ Still true for today's applications?

FPGA – Applications & Markets

- Telecom industry
- Automotive sector (ADAS, autonomous driving …)
- Automation technology (machine learning, neuronal networks, AI …)
- IoT
- Medical devices
- Consumer electronics
- …
- According to a market study published by the EE Times magazine, FPGAs outgrew the overall semiconductor market in 2016 (6% versus 1.5%)

IEEE “Chip Hall of Fame”



IEEE “Chip Hall of Fame”

Chip Hall of Fame: Xilinx X...

https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-xilinx-xc2064-fpga

IEEE SPECTRUM Engineering Topics Special Reports Blogs Multimedia The Magazine

Chip Hall of Fame: Xilinx XC2064 FPGA

Hardware that can transform itself on command has proven incredibly useful

Photo: Xilinx

CHIP HALL OF FAME

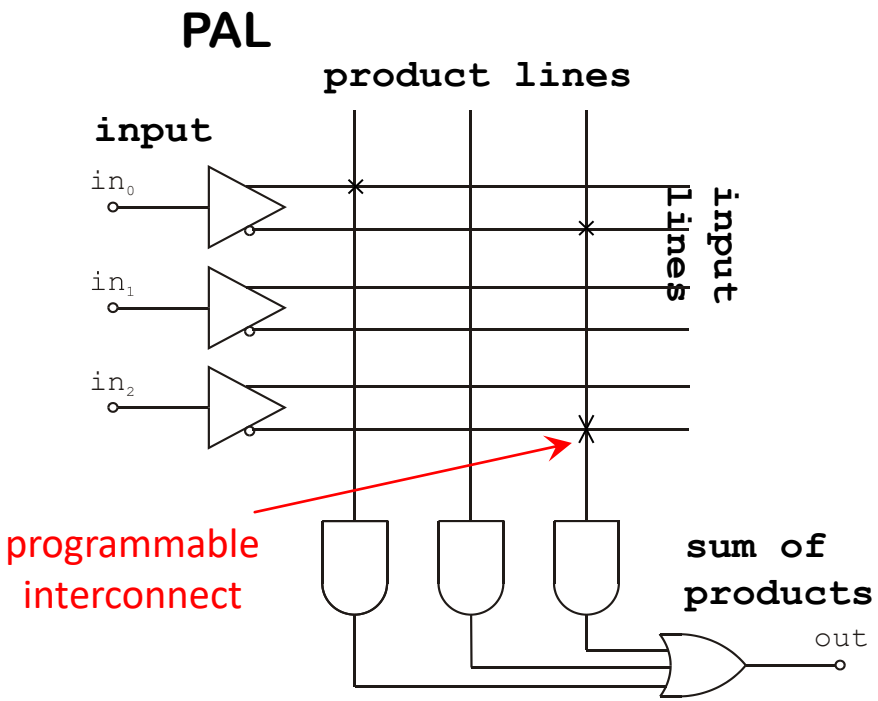
Back in the early 1980s, chip designers tried to get the most out of each and every transistor on their circuits. But then Ross Freeman had a pretty radical idea. He came up with a chip packed with transistors that formed loosely organized logic blocks with connections that could be configured and reconfigured with software. As a result, sometimes a bunch of transistors wouldn’t be used—heresy!—but Freeman was betting that Moore’s Law would eventually make transistors so cheap that no

FPGA – History

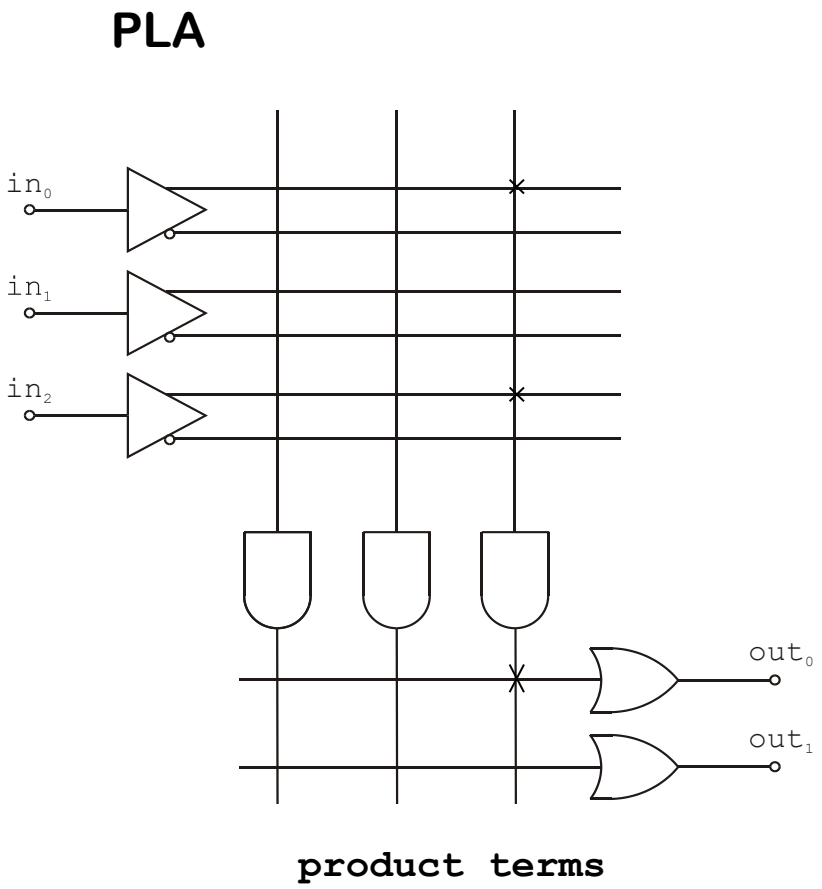
- 1967: Fairchild's "Micromosaic"
- early '70s: First appearance of PLDs
- late '70s: Arrival of Complex PLDs (CPLDs)
- 1985: First SRAM based devices
- late '80s: Arrival of FPGAs
- 1991: In-System Programmability (Lattice)
- 1998: First 1 million gates device
- 2000: First 3 million gates device
- 2005: First 10 million gates device
- …

Earliest PLDs

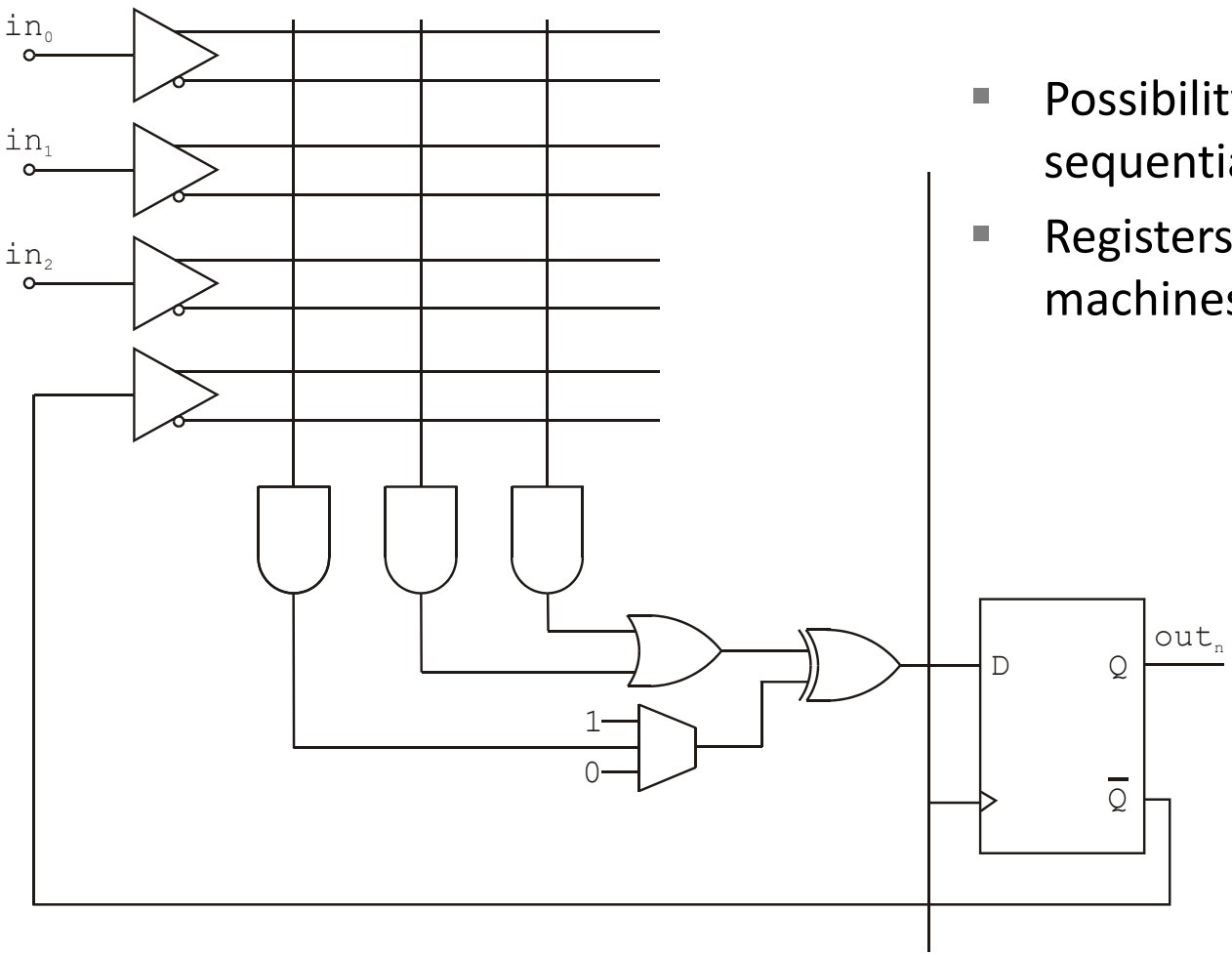
- Pure combinatorial logic only
- **PALs** (Programmable Array Logic) and **PLAs** (Programmable Logic Arrays)



out = in_0 **OR** (/in_0 **AND** /in_2)



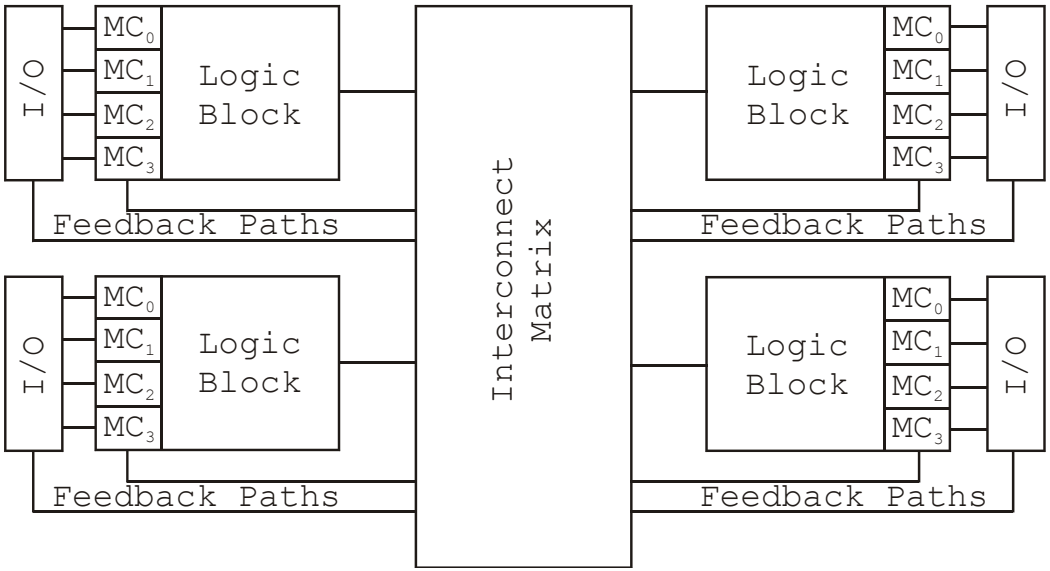
PLDs with Registers



- Possibility to implement sequential logic
- Registers, counters, state machines …

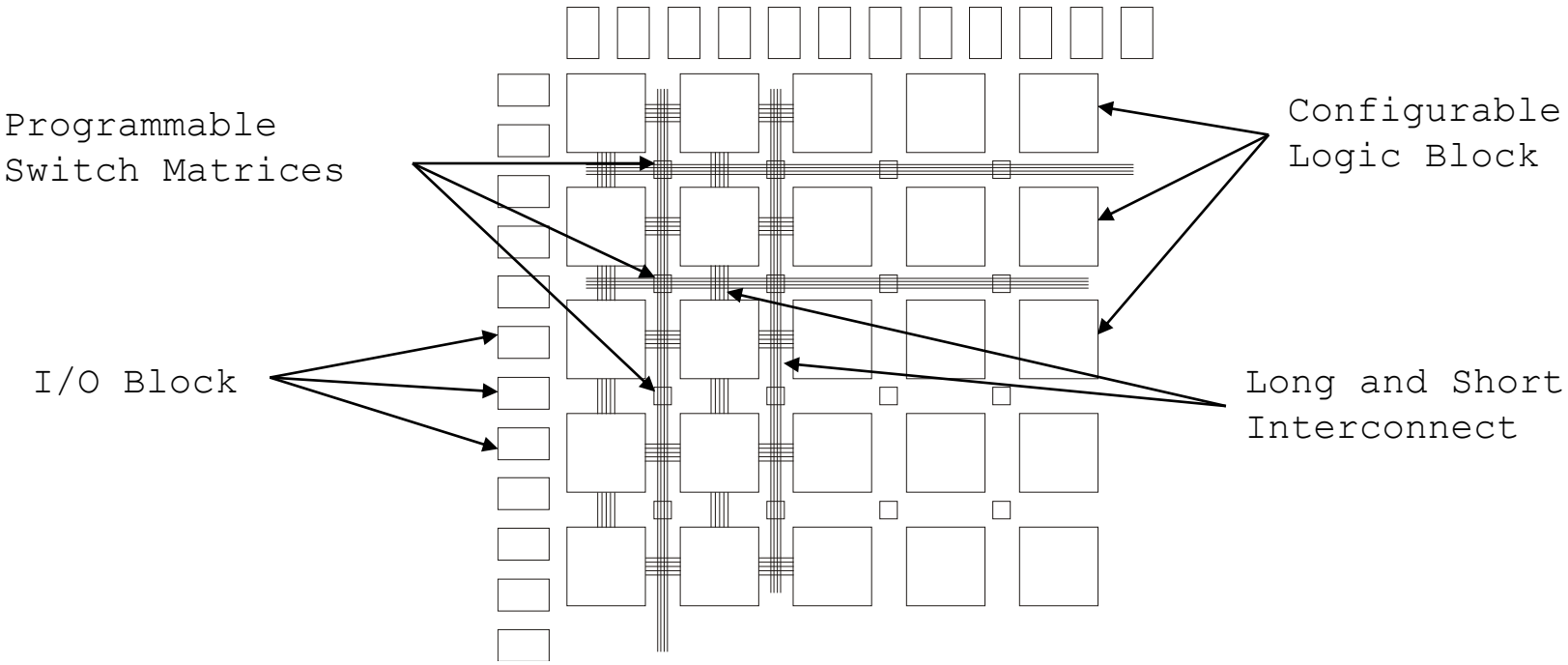
Complex PLDs (CPLDs)

- As the complexity increased, the programmable logic was arranged different to build up smaller portions of logic (“Logic Blocks”)
- New software (“Fitter”) has been developed to utilize the vendor-specific portions of logic



FPGAs

- Demand for increasing design complexities led to FPGAs
- Configurable logic blocks are arranged in a regular array



Today's FPGAs

- Newest FPGA devices are manufactured in 7nm FinFET process technologies using stacked silicon interconnects
- Example: Xilinx/AMD FPGA devices
  - Up to 9 million logic cells
  - Up to six 64/32-Bit ARM CPU cores, clocked above 1 GHz
  - > 10.000 DSP blocks (hardware multipliers)
  - Up to 500 MBit on-chip SRAM + 16 GByte DRAM in-package
  - 32/58 GBit/s IO transceivers providing support for hi-speed interface standards (PCIe Gen2, USB3.0, SATA3.1, Display Port, GBit Ethernet, …)
  - Up to 2.000 IO pins
  - General-purpose ADCs
  - …

FPGA Configurable Logic Blocks & IO Cells

- Example: For a simplified view of ”Configurable Logic Blocks” as implemented in Xilinx/AMD 7-series FPGA family, see

Xilinx 7 Series FPGAs,
Configurable Logic Block,
User Guide UG474 (v1.8), Page 20:

Source: https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf

- For an example of an FPGA I/O cell, see

Altera Stratix III Device Handbook,
Volume I, Document Version 2.2,
Page 7-13

Source: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stx3/stratix3_handbook.pdf

FPGA Configuration Memory

- Holds the design/circuitry
- Access typically over serial interface (e.g., JTAG/Boundary Scan)
- SRAM-based
  - Offers highest densities and most sophisticated devices
  - Need for non-volatile storage device that is used to “boot” the FPGA with a design after power-up
  - Examples: FPGAs from Intel/Altera and Xilinx/AMD
- FLASH-based
  - No need for extra non-volatile memory device
  - Lower densities
  - Example: FPGAs from Microsemi/Microchip
- Fuse/anti-fuse based
  - One-time programmable
  - Traditional technology used for PLAs, PALs, GALs …

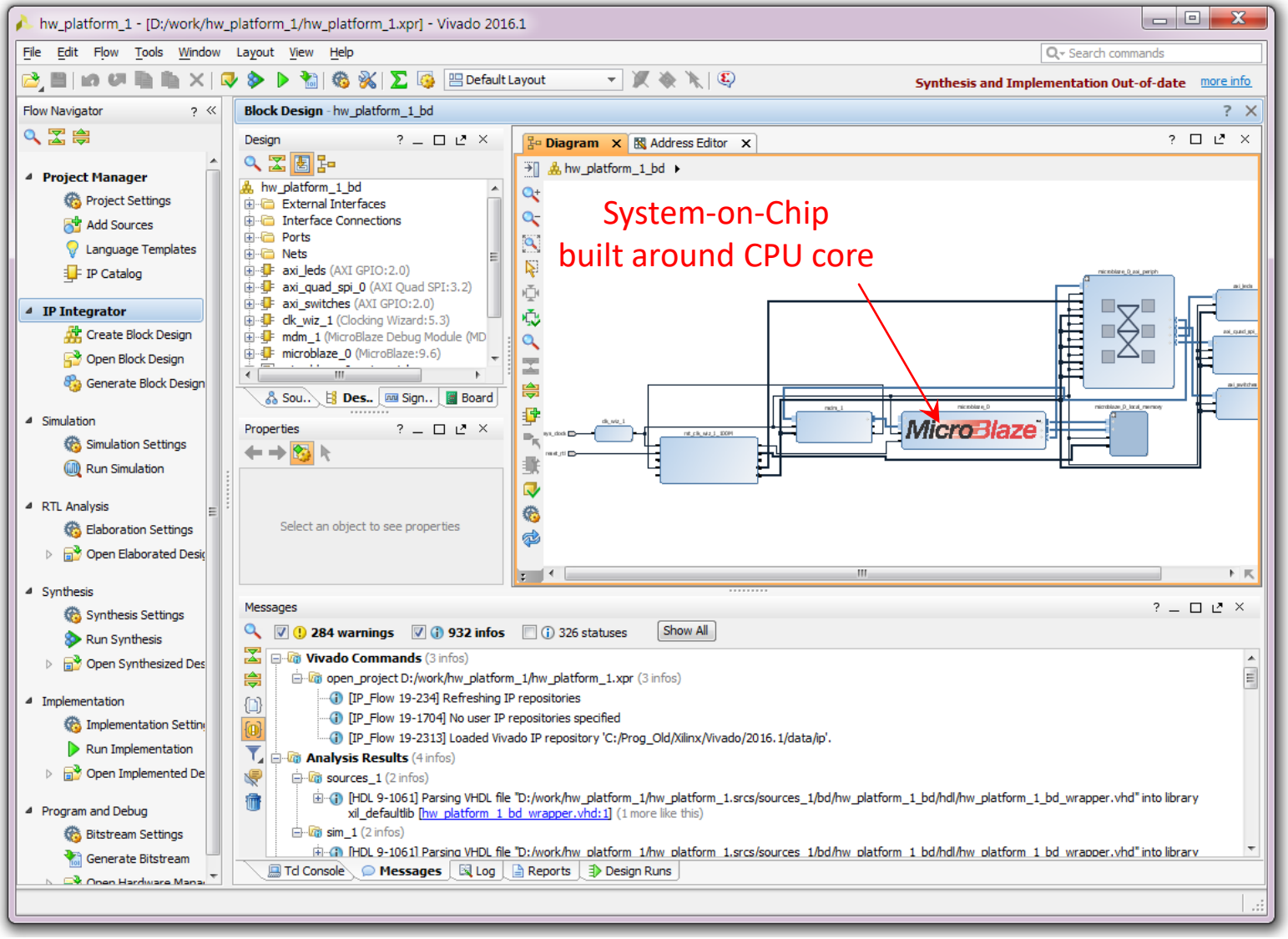
FPGA Vendors

- Xilinx/AMD, Intel/Altera (market leaders, since decades)
  - ➔ Highly sophisticated devices and tool chains
- Lattice Semiconductor
- Microsemi/Microchip (FLASH-based devices)
- Atmel/Microchip
- Achronix (high-speed FPGAs)
- QuickLogic (provide FPGA cores that can be embedded into ASICs)
- Efinix (low-power FPGAs in small packages)
- …

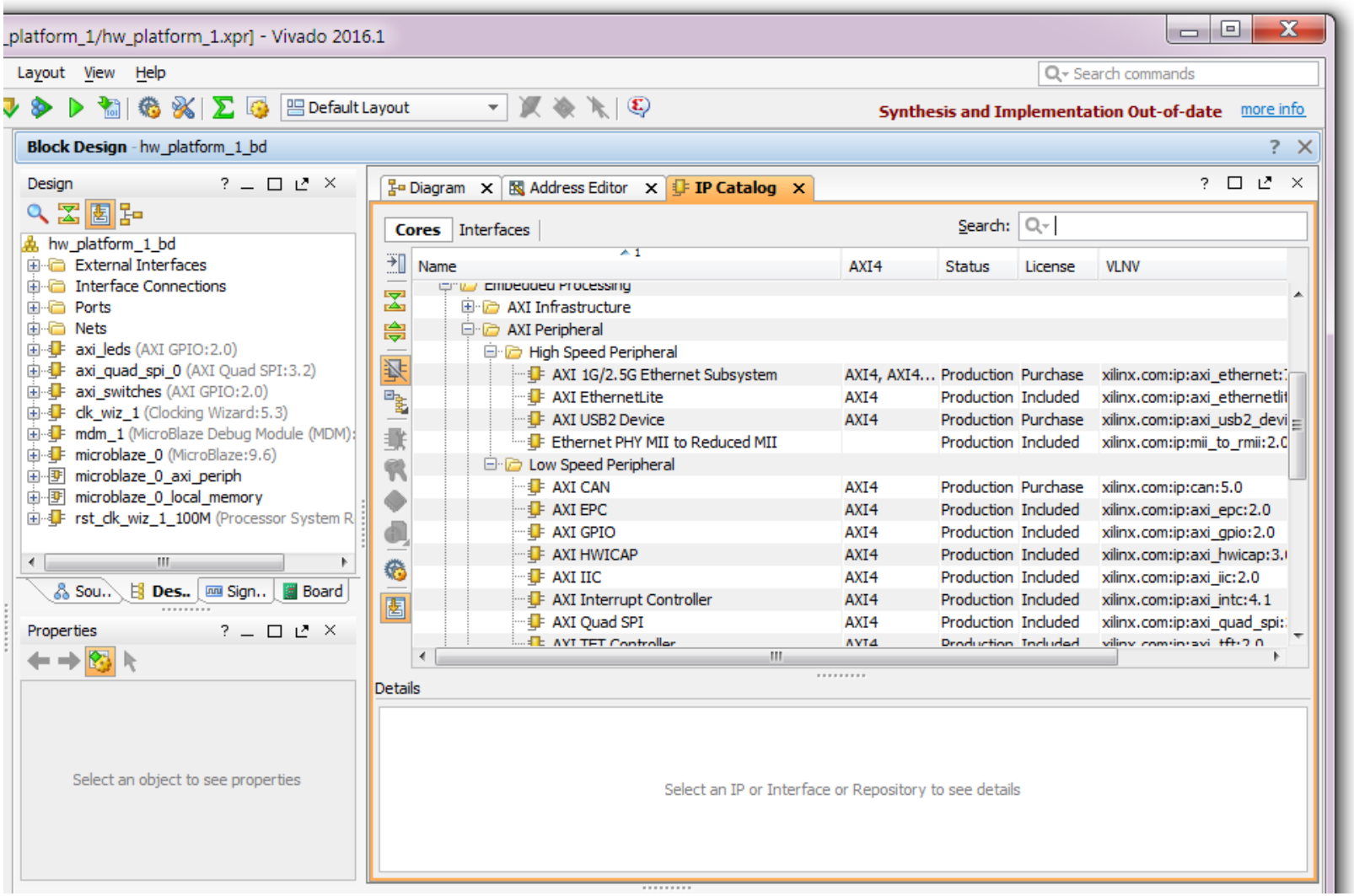
Design Tools

- For smaller FPGA devices, tools come for free
  - Intel/Altera “Quartus Prime”
  - Xilinx “Vivado”, Xilinx “Vitis”
- Tools also provide various IP (Ethernet, TCP/IP stacks, CPU cores, DDRx memory controllers, SPI/I^2C/UART …)
- Today, tools support both hardware **AND** software development
- Design entry is typically based on languages like
  - Hardware: VHDL, Verilog, SystemVerilog … but also C/C++/SystemC
  - Software: C/C++
- Additional third-party tools might be useful for
  - Modeling (e.g., Matlab/Simulink)
  - Simulation (PSL, e, OpenVera …)
  - (High-Level) Logic Synthesis

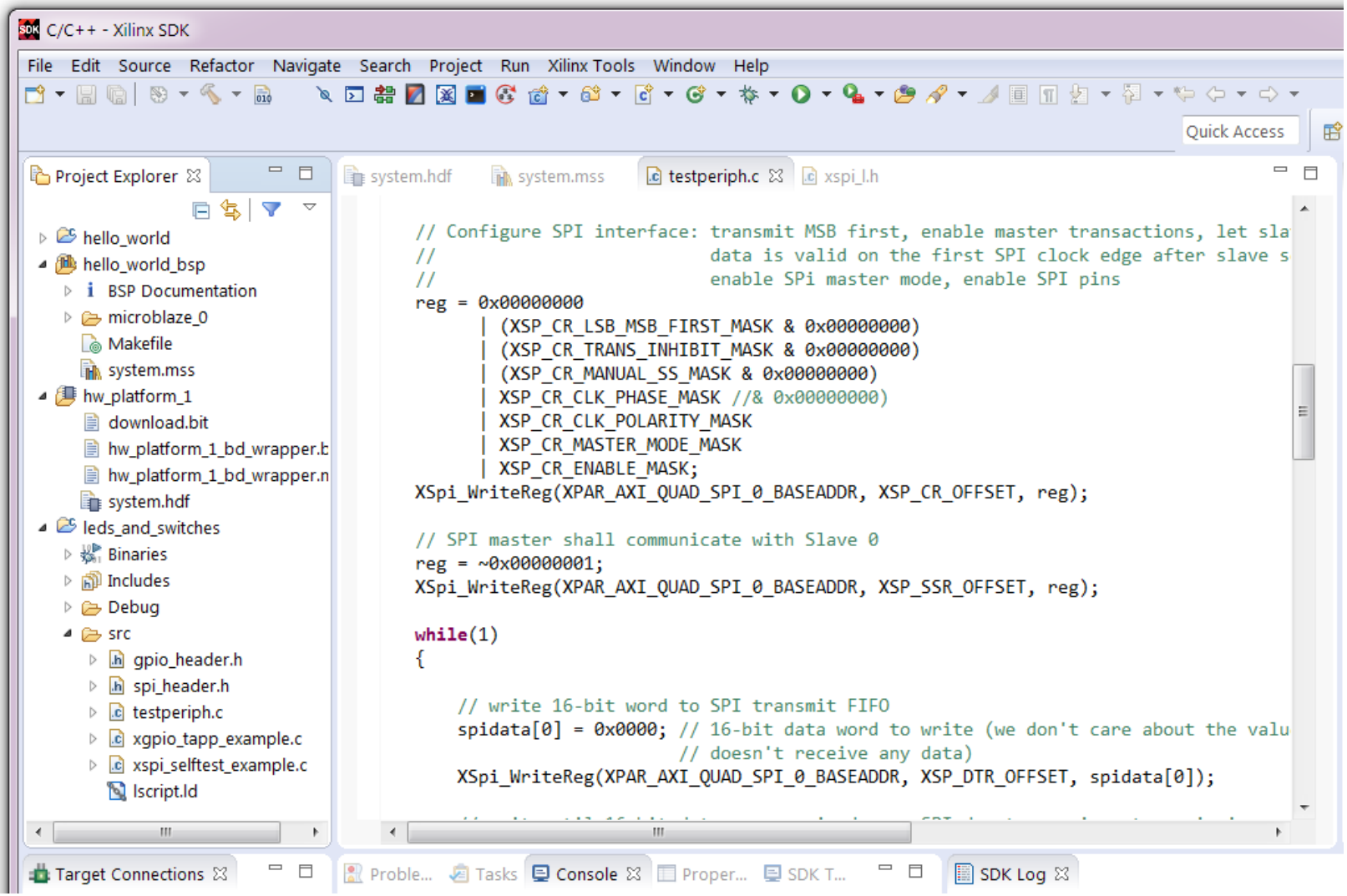
Xilinx Vivado – Hardware Design



Xilinx Vivado – Integrating IP



Xilinx Vivado – Software Development



Xilinx Vitis – Application Acceleration Development Flow

- Recently, Xilinx released an even more sophisticated design tool for FPGAs called “Vitis”
- With Vitis, performance-critical portions of an application can be accelerated by using dedicated libraries or by developing own C/C++, OpenCL or RTL code
  - The acceleration flow is heavily supported by the tool chain
- For details, see Xilinx website

FPGA Trends

- Hardware/Software Co-Design (➔ SoC FPGAs)
  - Support for design space exploration desired
- High-Level Design/Synthesis
  - e.g., C-based hardware development
- (Embedded) Hi-Performance Computing
  - Computer vision, machine learning, AI, neuronal networks …
- Safety & Security requirements in an increasing number of applications
  - Networked/distributed systems, IoT, …
- Dynamic reconfiguration
  - Application can dynamically respond to events
  - Reconfigurable computing
- …

FPGA-related R&D@Technikum Wien

- R&D related to FPGAs exists at our university since >20 years
- Currently, future trends and challenges in the area of FPGA design are investigated at our department in the 5-year public funded project **INES** (**IN**novative Platforms for **E**lectronic-Based **S**ystems) together with several partners from the industry …

- The following slides present some results from INES & other projects of our department as well as projects from others in order to show how applications can benefit from using FPGA technology …

Computer Vision using FPGAs

- Example: “Hi-Speed Smart Camera” developed from Vienna University of Technology, Austrian Institute of Technology (AIT) and the Austrian company Oregano Systems GmbH, see
  - *C. Eckel, E. Bodenstorfer, H. Nachtnebel, P. Rössler et al., Hochgeschwindigkeitskamera mit intelligenter Datenvorverarbeitung (in German), Proceedings of Austrochip 2006, Oct. 11, 2006, Vienna, Austria, pg. 103-108, ISBN 3-200-00770-2*
  - Here, the FPGA is responsible for image pre-processing in order to reduce the huge amount of image data coming from the image sensor
- Another example: Banknote inspection system from Vienna University of Technology, Austrian Institute of Technology (AIT) and the Austrian company Oregano Systems, see
  - *J. Fürtler J., P. Rössler et al., Design Considerations for Scalable High Performance Vision Systems Embedded in Industrial Print Inspection Machines, EURASIP International Journal on Embedded Systems, Special Issue on Embedded Vision Systems, Vol. 2007, Article ID 71794, 10 pages, Hindawi Academic Publishing, doi:10.1155/2007/71794*
  - Image processing system consisting of FPGA + DSP
  - Can process up to 40 banknotes per second

COPACOBANA FPGA Cluster

- COPACOBANA (Cost-Optimized Parallel Code Breaker) from the Ruhr University of Bochum and Christian-Albrechts University Kiel, Germany
- A high-performance computing cluster consisting of 120 FPGAs used for cryptoanalysis
- See **https://www.copacobana.org**

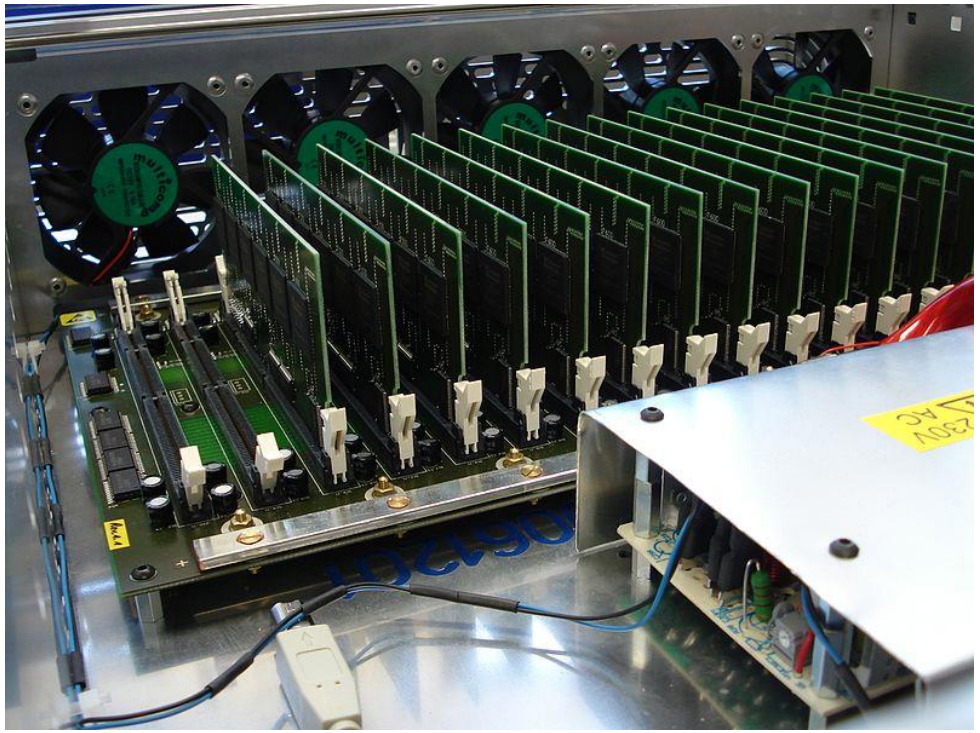


Photo: Gerd Pfeiffer, licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License, see https://creativecommons.org/licenses/by-sa/3.0/deed.en

Data Collection Platform (DCP)

- A DCP is used to transmit environmental data (physical, chemical or biological properties of oceans, lakes, solid earth, atmosphere …) to a satellite
- In the DCP developed by our department an FPGA maintains the accurate timing needed for the PSK modulation which would not be possible using a microcontroller
- For details, see
  - *P. Rössler et al., Development of a Data Collection Platform, Proceedings of IEEE ISIE (International Symposium on Industrial Electronics) 2008, June 30 - July 2, 2008, Cambridge, UK, pg. 1953-1958*

- RISC-V is an open instruction set architecture (ISA) for CPUs that originates from the University of California, Berkeley
- In the last years, RISC-V attracts great interest from the community
- The “RISC-V International” association (formerly “RISC-V Foundation”) already shows >2.000 members, with many of them well known companies
  - see **https://riscv.org/**
- In a survey paper we tried to provide an overview of existing projects, see
  - *R. Höller et al., Open-Source RISC-V Processor IP Cores for FPGAs – Overview and Evaluation, 8th Mediterranean Conference on Embedded Computing (MECO), June 10-14, 2019, Budva, Montenegro, 6 pg*
- Our survey shows that a lot of open-source RISC-V implementations are available, also for FPGA technology
  - However, an open-source tool chain would be desired similarly to commercial tools from Intel/Altera or Xilinx/AMD in order to quickly build up a RISC-V based system-on-chip from scratch

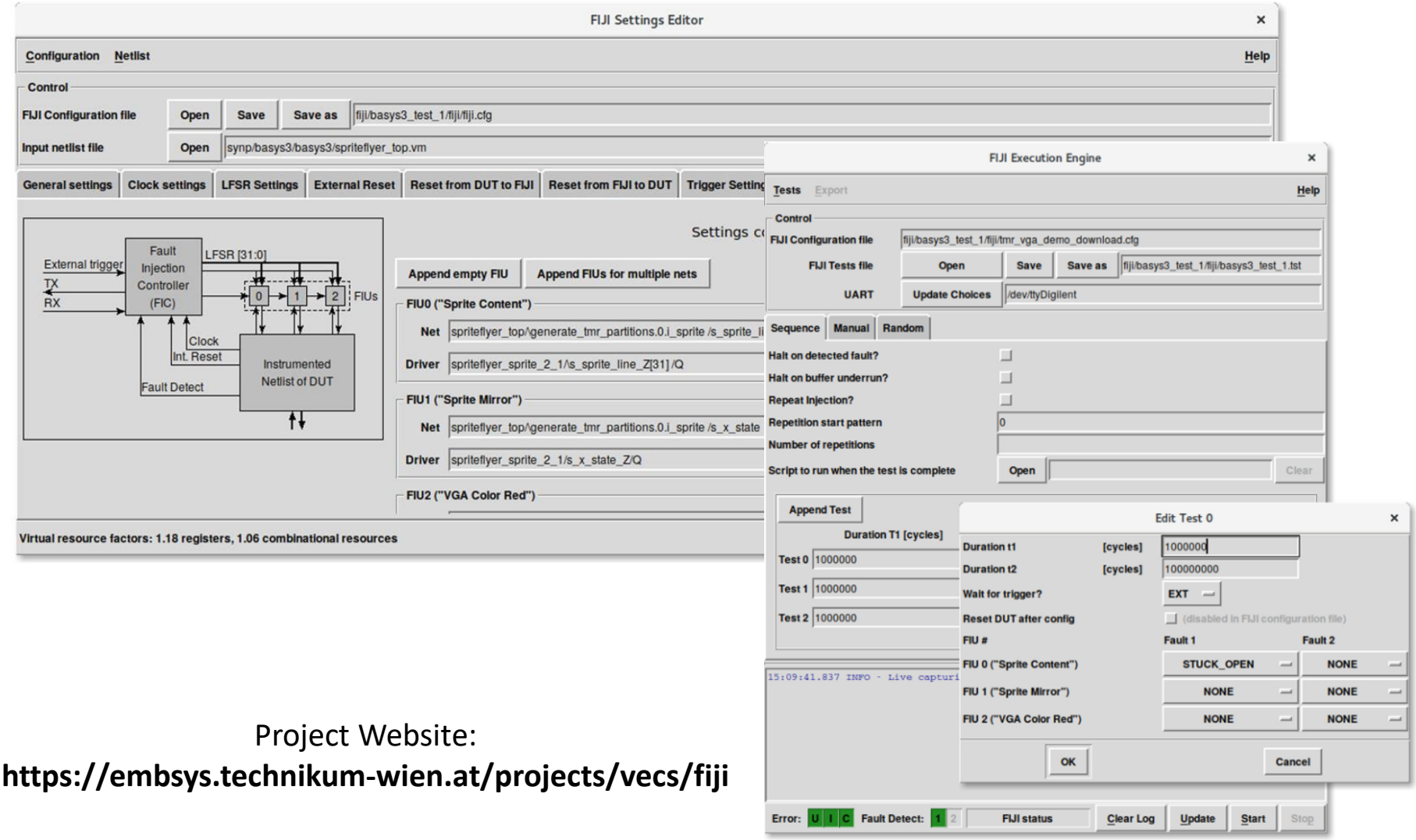
FPGAs & Security

- With the increasing number of electronic devices being connected to the Internet the importance of security aspects is well recognized
- Here, the reconfigurable nature of FPGA logic might be interesting to update security features in the field or to fix bugs
- With this regard, an open-source implementation of security hardware features is highly beneficial since it would allow the worldwide community to inspect the code – a principle that has already proven very valuable in the open-source software community
- In the following paper we have presented an overview of available open-source hardware security building blocks for basic cryptographic functions (AES, SHA, TRNG, PUF …) and show evaluation results of selected cores in FPGA technology:
  - *M. Billmann M., S. Werner S., R. Höller et al., Open-Source Crypto IP Cores for FPGAs - Overview and Evaluation, Proceedings of the 27th Austrochip Conference on Microelectronics, Oct. 24, 2019, Vienna, Austria, 8 pages*

FPGAs & Functional Safety

- FPGAs are increasingly used in safety-critical application domains like aerospace or the automotive industry
  - ➔ Safety aspects have to be taken into account
- In context to safety-critical applications, a fault-injection tool can be very helpful
  - to test fault-detection and avoidance mechanisms or
  - to stress an application and analyze its behavior when faults occur
- Recently, our team has developed a fault-injection tool called **FIJI (Fault InJection Instrumenter)**
- The proposed approach
  - smoothly integrates into an industrial FPGA tool flow
  - supports devices from multiple FPGA vendors
  - and is completely open-source …

FIJI (Fault InJection Instrumenter) Tool



Project Website:

https://embsys.technikum-wien.at/projects/vecs/fiji

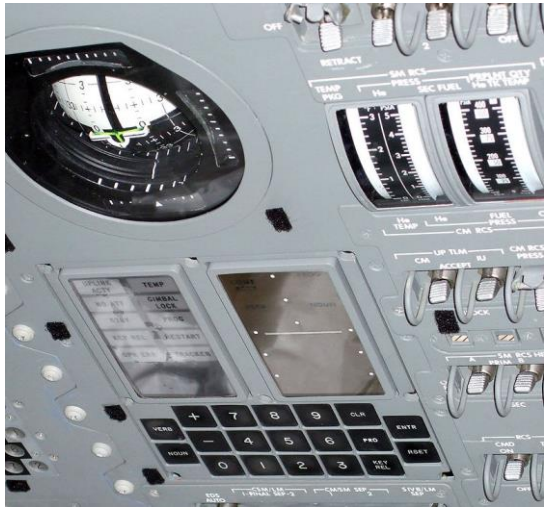
High-Level Synthesis

- High-level synthesis (HLS) promises a boost in productivity by enabling synthesis of low-level electronic circuit descriptions out of high-level descriptions (e.g., based on C/C++ or SystemC)
- Idea is not new – HLS for FPGAs definitely gained momentum when the FPGA vendor Xilinx integrated an HLS design flow into its tool chain (Vivado HLS) some years ago
- HLS approaches have the potential to realize early prototypes of an electronic system very quickly
- However, in comparison to traditional (none-HLS) design flows, room is left for HLS tools concerning implementation results which could also be seen in our evaluations, see:
  - *C. Fibich , S. Tauner, P. Rössler et al., Preliminary Evaluation of High-level Synthesis Tools – Xilinx Vivado and PandA Bambu, IEEE 13th International Symposium on Industrial Embedded Systems (SIES), June 6-8, 2018, Graz, Austria, 4 pages*

Retrocomputing

- FPGAs are perfectly suited to recreate mature hardware and computers
  - Examples: “Commodore 64” (most prominent home computer of the 1980s) or the Apollo Guidance Computer (AGC) that was developed for the NASA Apollo spaceflight program (1961 to 1972)
- A more serious aspect is the replacement of mature/discontinued semiconductor devices by FPGAs

Apollo Guidance Computer (AGC):



Source (Public Domain):
https://commons.wikimedia.org/wiki/File:Dsky.jpg

Final Example: FPGA Prototyping of Novel Test & Debug Approach

- Novel patent-registered approach for test and debug actions of networked (distributed) embedded systems developed from our department
- Idea: A single network is used to transfer both application and debug data
  - Test & debug actions (trace, monitoring, replay, fault injection, …) for all nodes can be synchronized using a system-wide time
  - Our approach has been prototyped by using FPGA technology
- For details, see
  - *Rössler P., Höller R., A Novel Debug Solution for Distributed Embedded Applications and Implementation Options, Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society (IECON), Nov. 7-10, 2011, Melbourne, Australia, pg. 2711-2716*

Conclusion

- FPGAs evolved from implementing small glue-logic designs to large, configurable multi-processor System-on-Chips (SoC)
- Increasing demand for FPGAs in a number of application domains (telecom, automotive, automation, Industry 4.0, IoT, AI, medical devices …)
- FPGA advantages: Performance, speed, (low-power), single-chip solutions, re-configurability, reduced time to market (e.g., compared to ASICs)
  - Some examples have been given on the previous slides
- Main disadvantage of FPGAs: Parts costs
- FPGA vendors increasingly try to attract software engineers
  - FPGAs with hard-wired CPU cores (e.g., ARM Cortex based)
  - C-based hardware development / High-level synthesis
- FPGAs provide an interesting technology option to implement huge & really complex hardware/software applications

Thank You for your Attention!

- Acknowledgements:
  - *Parts of this work received financial support from the Austrian Federal Ministry for Digital and Economic Affairs (BM:DW) and the National Foundation for Research, Technology and Development as related to the Josef Ressel Center "Innovative Platforms for Electronic-Based Systems" (INES), managed by the Christian Doppler Research Association*
  - Many thanks to the INES core team: C. Fibich, R. Höller, A. Puhm and P. Schmitt