RUHR-UNIVERSITÄT BOCHUM

RUB

Run Time Adaptive Processor Architectures

Tensilica Day, Leibniz Universität Hannover

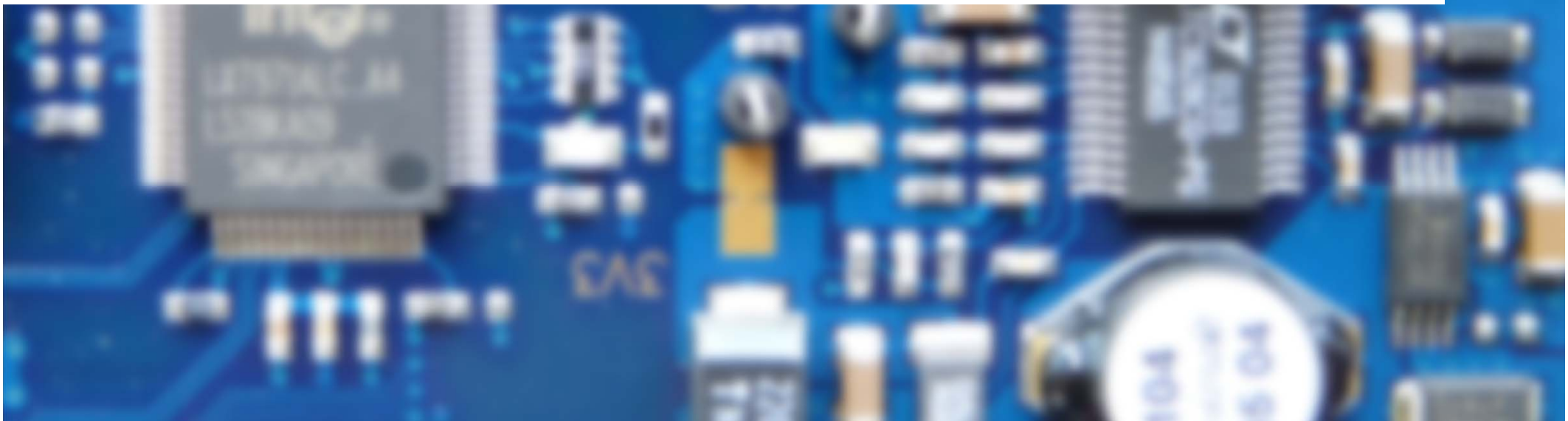
February 9th, 2016

Prof. Dr.-Ing. habil. Michael Hübner

Embedded Systems of Information Technology (ESIT)
Ruhr-University Bochum
Prof. Dr.-Ing. habil. Michael Hübner

www.esit.rub.de

cadence®
ACADEMIC NETWORK



Agenda

- **Short introduction of the institute**
- **Motivation: Current needs and research challenges in adaptive processors**
- **Example of modifying a processor to become adaptive**
- **Summary**

Introduction to the Institute

- Since April 2012: **Chair for Embedded Systems in Information Technology** (ESIT)
  - Short Curriculum: Study (EE), PhD, Habilitation in Karlsruhe (KIT / Uni-KA)
  - Novel and innovative research directions, chair was established new
  - Focus on adaptive hardware, including processors, dynamic and partial reconfiguration, heterogeneous architectures like processors with accelerators
  - Focus low power hardware architecture, tools and applications

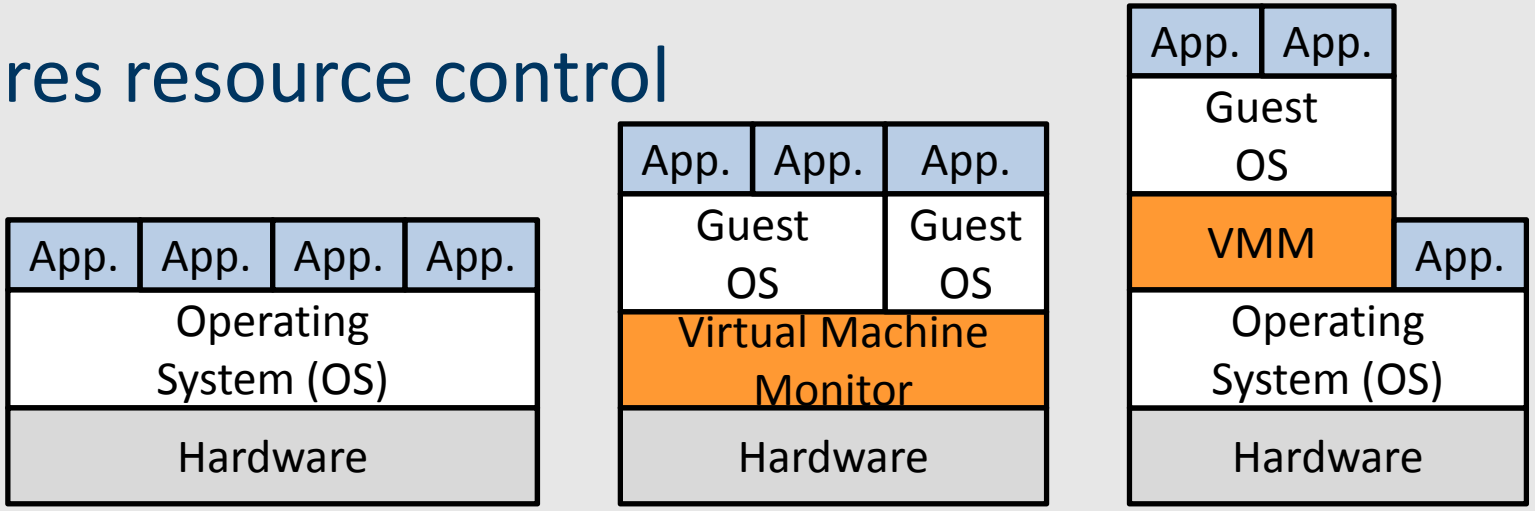
- 10 PhD Students: Muhammed Al Kadi, Tomas Grimm, Fynn Schwiegelshohn, Osvaldo Navarro, Jones Yudi, Uwe Mönks, Florian Fricke, André Werner, Benedikt Janßen, Florian Kästner
- Continuously over 20 Master, Bachelor, Seminar Students and over 15 Student researchers (part time)
- Numerous industry collaborations, several DFG, EU, BMBF and other research projects, Lead institution for System Level Design in the Cadence Academic Network, Certified Tensilica Lab since 2014

Motivation

- Adaptive systems
  - Design space with modern architectures is huge
  - Overcome offline adaptations which can only target expected points of operations
  - Autonomous run-time adaptions enable to target the complete design space
- Future trends:
  - Virtualization techniques of complex hardware
  - Injection of data deep into the processor pipeline
  - Adaptive memory concepts (e.g. cache)

Basic principles: Virtual Machine Monitor (VMM) or Hypervisor

- Software-Layer
- Characteristics:
  - Enable the abstraction from hardware or OS
  - Must be highly efficient
  - Requires resource control



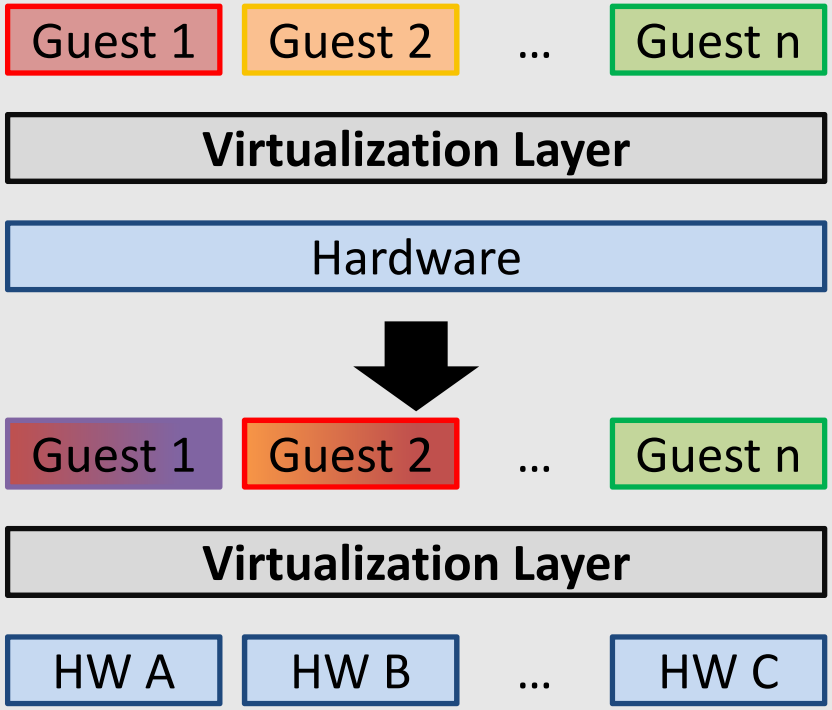
Source: http://de.wikipedia.org/wiki/Virtuelle_Maschine

Benefits

- More than one OS on one system
- Isolation of the virtual machines
- Increase in system flexibility
- Increase of efficiency
- Hardware independence

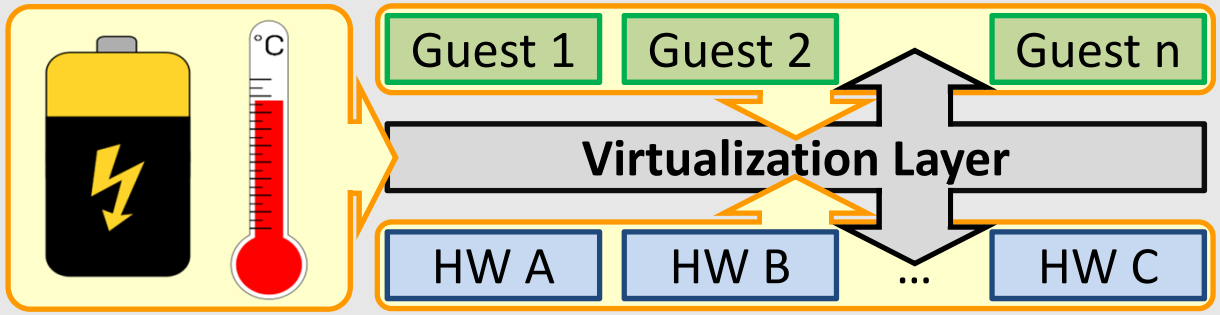
Virtualization Techniques

- State of the art:
  - Sharing resources
  - Guest priority
- Challenges:
  - Dynamic guest priorities
  - Dynamic resource availability



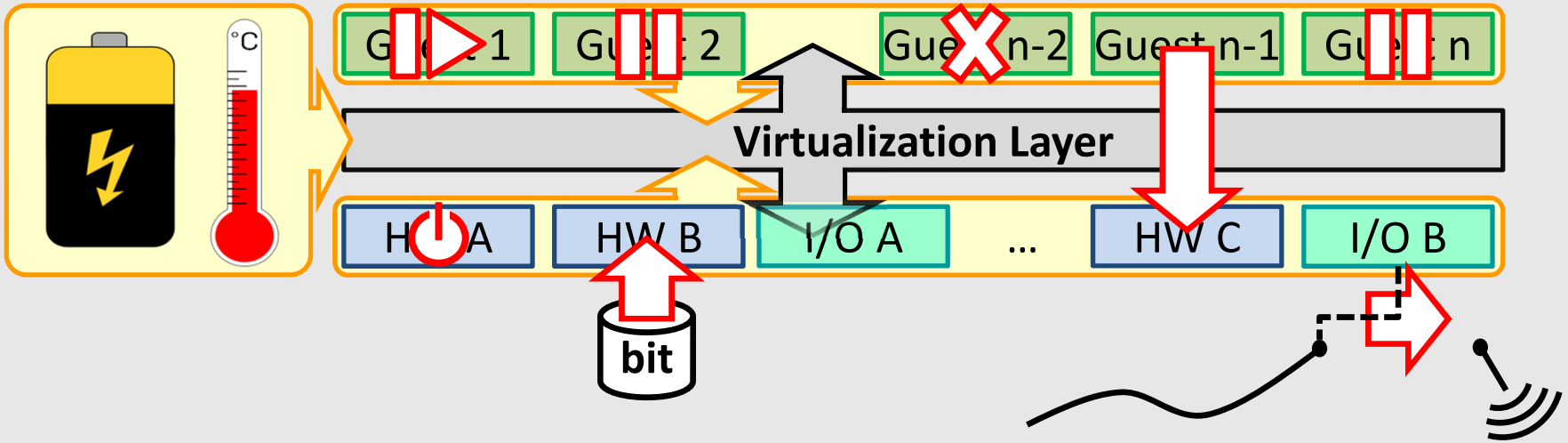
Virtualization Challenges

- Run-time sensibility:
  - Constraints
  - Resources
- Control:
  - Guest
  - Hardware



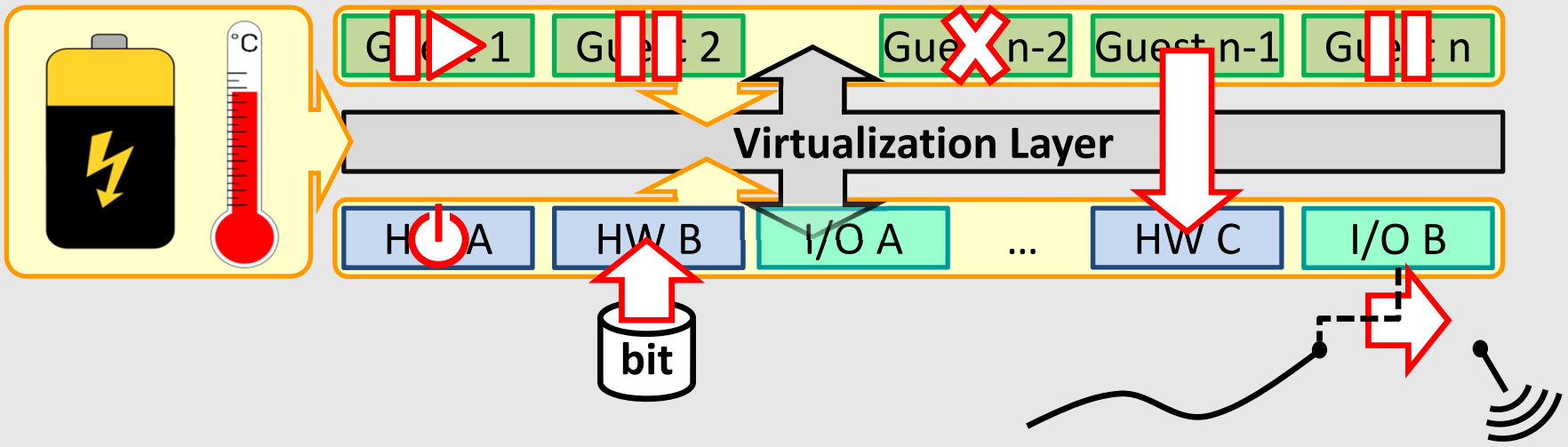
Future Virtualization

- Proposal:
  - Schedule application execution by run-time parameters
  - Handle and perform I/O reconfiguration and hardware adaptations



Future Virtualization Challenges

- Challenges:
  - Application execution control
  - Adaptive hardware abstraction



The huge challenge with the novel processor architecture and complex applications

- Processor architectures become complex
  → Design space (also due to parameterization) with modern architectures is huge
- Modern applications have effects which need more than one specialization of the underlying hardware (keyword mixed criticality)
- Solution: Run-time Adaptive systems
- Extend offline adaptations by run-time adaptivity which can target varying points of operations required by the application and environment
- Autonomous run-time adaptions enable to exploit the complete design space of a parmeterizable processor

Things which could be adapted during run-time

- Dependently to the processor itself, different possibilities for an adaptation is provided

- Superscalar (only some examples):
  → Out of Order vs. In Order execution
  → Branch Prediction and Speculative Execution

- VLIW (also only some examples):
  → Dynamic issue (ex) stages clustering / de-clustering (e.g. for several threads with different priorities / real-time requirements, means HW support for “dynamic real threads”)
  → RISC mode, VLIW mode

(Non) adaptive processor architectures RISC / CISC

- Traditional processors are developed to support **a huge bandwidth of application with a sufficient performance they are not adaptive**

- Example: Intel Atom Processor

  - 45nm technology, 25mm^2 die, power consumption 0.65-2.4Watt
  - Was developed for mobile interface devices, Netbooks (here the power consumption plays a more important role as performance)
  - Superscalar 2-issue **in order execution** architecture, 16 pipeline stages

  → Lower performance, data dependencies more critical **but**

  → Less power consumption through the elimination of the reordering unit

- Other processor architectures have other features…. See next slide!

(Non) adaptive processor architectures DSP

- Traditional processor, developed for **dataflow oriented applications**
- Example: Texas Instruments DSP Processor
  - Developed for data flow oriented applications
    (here the data throughput plays a important role, also power consumption)
  - VLIW architecture (256bit), deep pipeline stages (can be over 20)

  → Data dependencies are solved by the compiler during design time

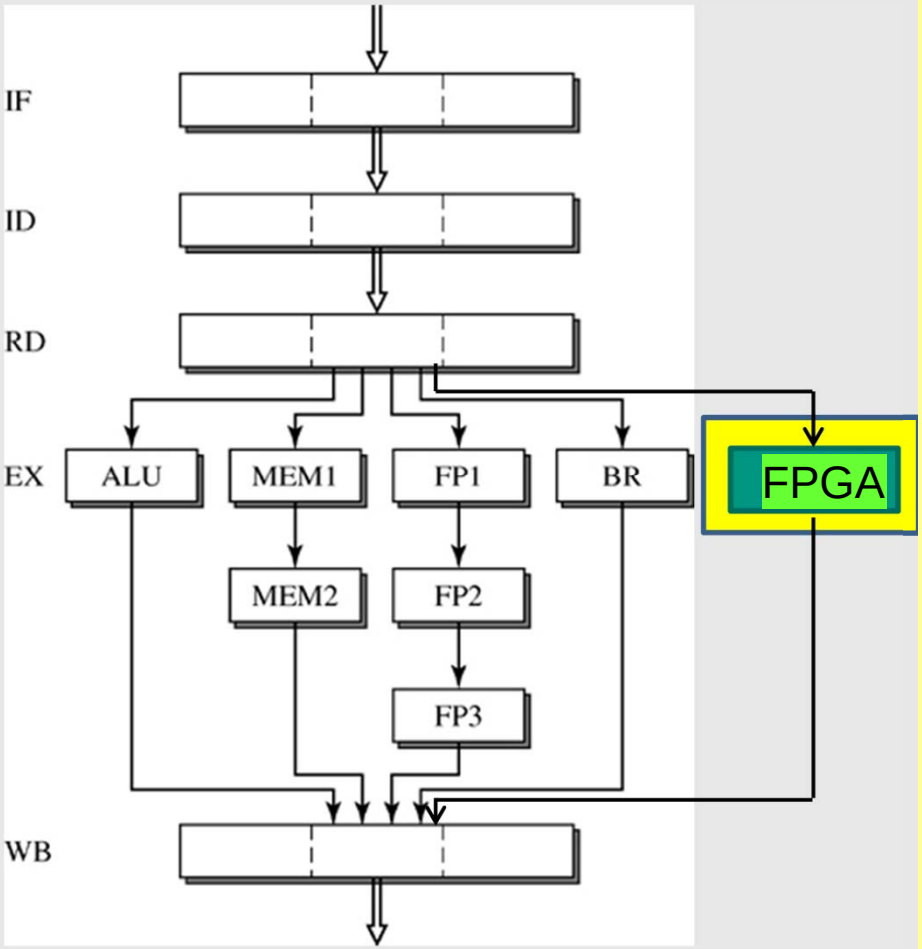
  → Application scenario with low control flow (therefore parallelization during design time by compiler) **but**

  → **every control flow, reduces the performance tremendously**

- So why not an adaptive processor, or at least providing a heterogeneous architecture

Novel exploitation possibilities of the FPGA in adaptive microprocessor

- Lets assume FPGA is integrated into the processor pipeline

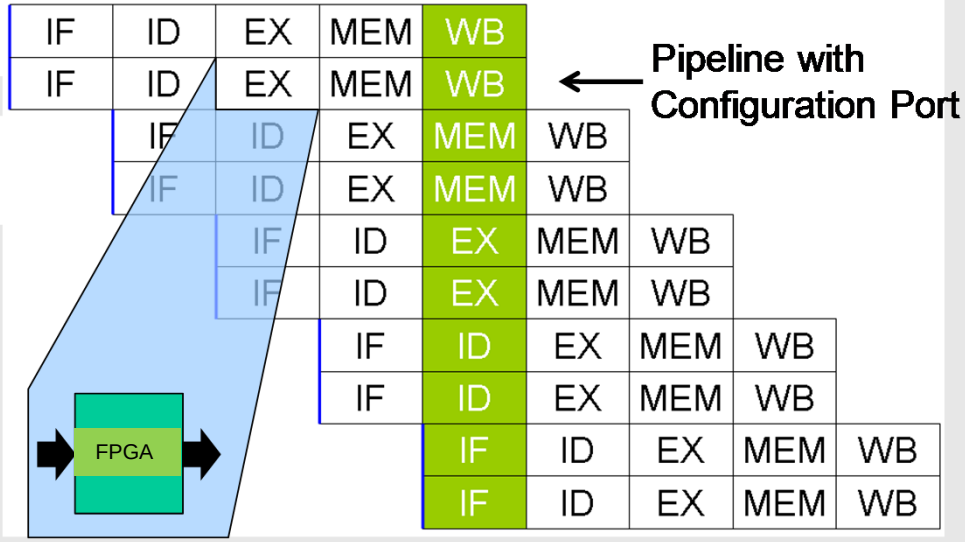


Extended version based on the picture used by Prof. Lizy Kurian John, Univ. Austin, Texas

That would mean:
•Processor commands are reserved for the FPGA:
Reconfiguration mode:
•FPGA write config.
•FPGA read config.
Data transfer mode:
•FPGA write process data
•FPGA read process data

•The FPGA is included directly into the data path of the processor
→ lowest delay for data transfer
→ see FPGA from „the software point of view“ and write simple programs for accessing it

Exploitation of the novel concept



Legend:

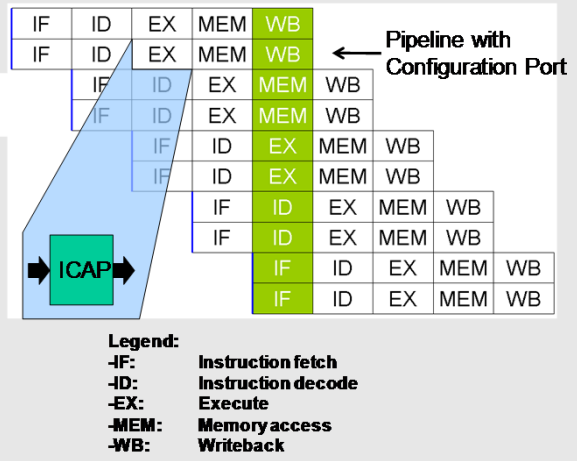
-IF:	**Instruction fetch**
-ID:	**Instruction decode**
-EX:	**Execute**
-MEM:	**Memory access**
-WB:	**Writeback**

- The novel concept increases the flexibility of a FPGA based processor tremendously
  - The FPGA as **data sink and source** can be seen as a **multipurpose ALU**
  - From the user **(programmer)** point of view the hardware complexity is hidden through the provided libraries
  - Accessible with standard C construction
  - Further hardware abstraction which definitely will increase the acceptance of run-time adaptive hardware

Exploitation of the novel concept (cont)

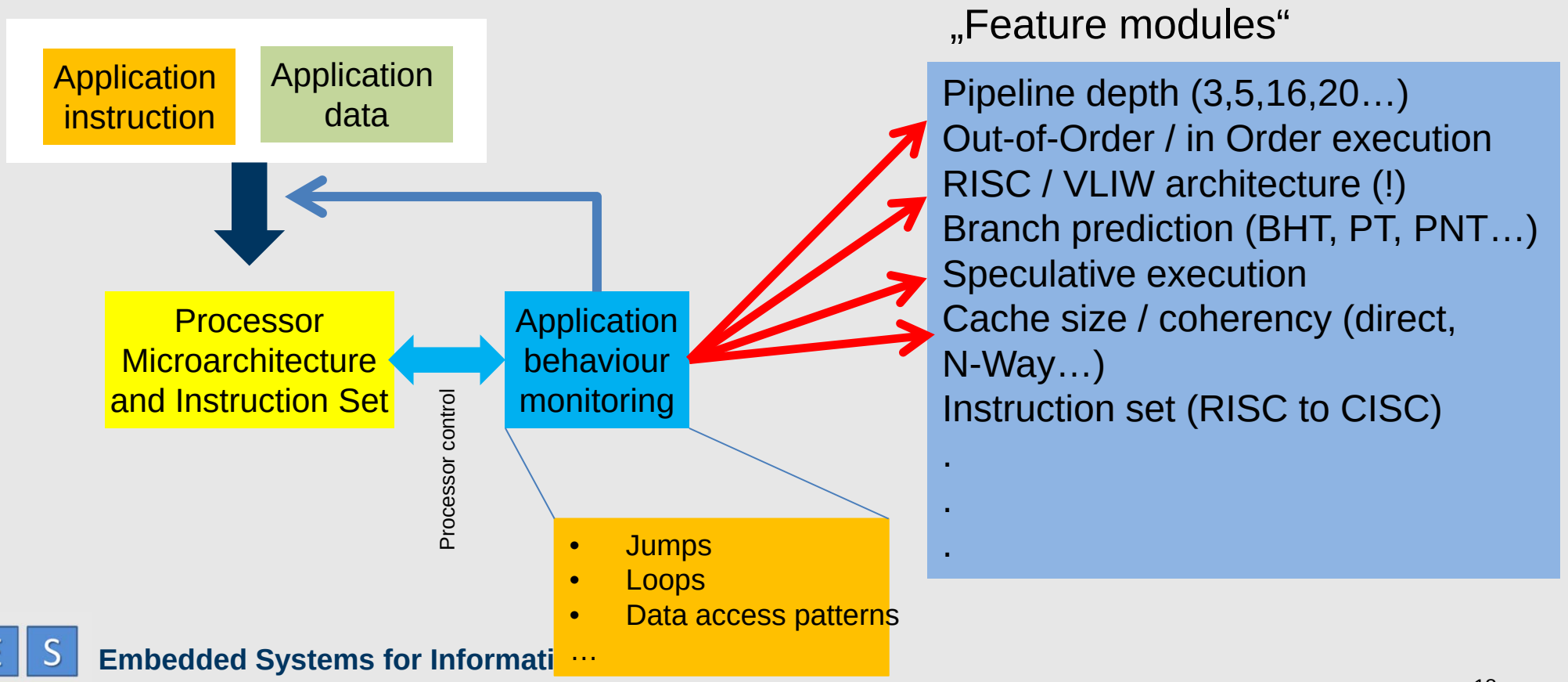
- The novel concept enables the run-time adaptation of the processors microarchitecture:
  - Realized instruction (within the ISA) are reconfigured at runtime and realizes therefore a dynamic reconfigurable instruction set processor
  - In general: An adaptive microarchitecture is possible:
    - **Power and energy reduction via pipeline balancing**
    - **Using ipc (instruction per cycles) variation reduce power consumption**
    - **Dynamic instruction level parallelism pipeline adaptation**
    - **Adaptive issue queue for reduced power at high performance**

→ Novel quality of processors: The **adaptive core** provides the run-time adaptation of the microarchitecture



The adaptive processor

- Application depend on the “position” of the processor
- Same basic architecture but adaptive to the requirement
- Different “requirements” of the application with different control / data flow overhead or even both in separate phases of the application



The adaptive processor: Advantages

- The adaptive processor is able to “react” to application requirements
- It can be deployed without modification in many application
  it starts as “general purpose processor and ends as application specific processor”
- The monitoring can be adapted to many signatures, even a “history” can be stored and reused (keyword case based reasoning from AI)

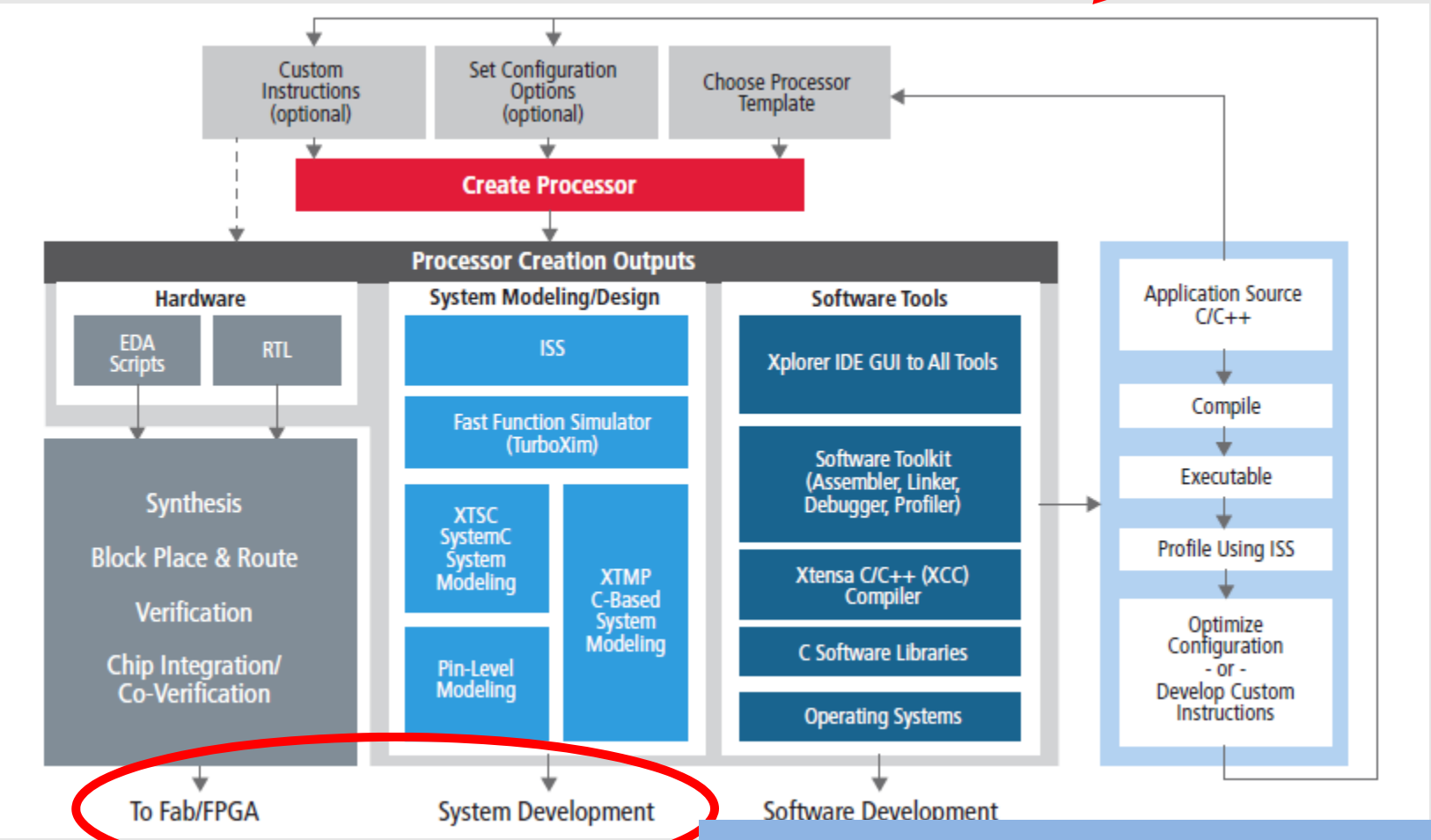
- **And:** it combines the methods of embedded computing with the ones from supercomputing (keyword multicore, power saving modes etc.)

→ In order to enable such an adaptation, a “low level” monitoring has to be realized
→ Low level monitoring has to be non-intrusive, means, minimal or much better no influence to performance and minimal impact to power consumption

→ BUT: many information comes almost for free: e.g.
IPC, pipeline stalls, activation of forwarding, memory access, branches and many more…

Back to the Tensilica IP

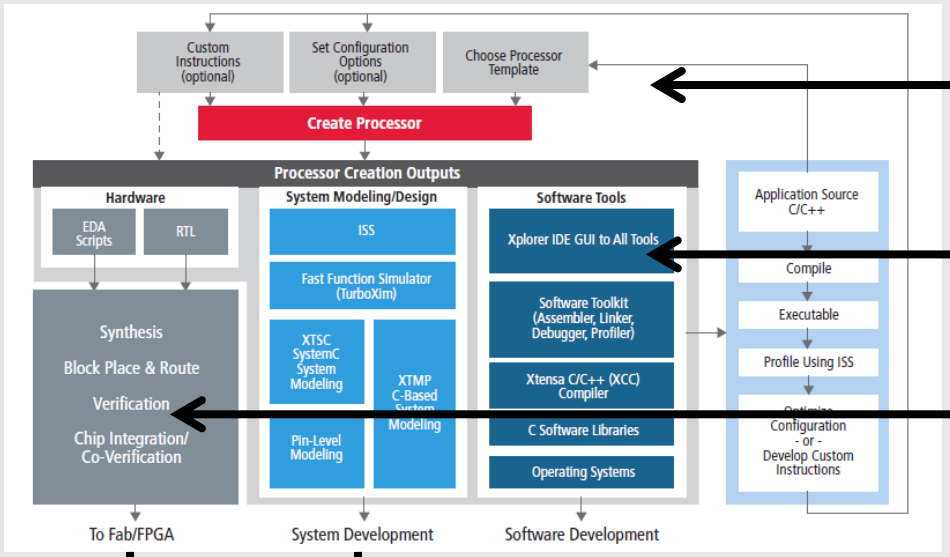
Feedback loop



Source: Cadence

If „low level monitor“ could deliver data and the processor can be adapted an additional feedback could be enabled

Activation of adaptation by insertion of special instructions by the compiler: smaller “adventure” (decision at design-time)



Source: Cadence

Generate

Feedback

Adaptive processor

Profiling / monitoring of the application „needs“

VSP or RPP

- **The feedback allows to insert „adaptation instructions“ according to „phases“ of the application.**
- **It enables a operation of the core in an performance and energy optimal parameterization → this is the VERY conservative approach**
- **Future: more run-time decisions required**

There are many opportunities to adapt more…

- Cache has a lot of influence on power and performance
  → new ideas in making cache run-time adaptive and more intelligent lead to significant impact

- Combining the single core adaptivity with adaptation in multi- and manycore architectures, opens a huge field of further adaptation mechanisms

- Ongoing research topic: hardware architecture and tools

→ **Discussion and research together with the Tensilica IP**

Many Thanks for Your Attention!

Thanks to Anton Klotz and the Cadence Academic Network for making such research possible!

E S I T
www.esit.rub.de

www.esit.rub.de

Contact:

michael.huebner@rub.de