A Time-predictable Microprocessor: the Patmos Approach

Martin Schoeberl
Technical University of Denmark

Outline

- Why time-predictable architectures?
- Definition of time-predictable computer architecture
- The Patmos processor
- Conclusion

Real-Time Systems

- Systems with timing constraints
- In (hard) real-time systems
  - Function has to be correct
  - Function has to deliver result in time
- Timing proof with schedulability analysis
  - Execution time of tasks need to be known
  - WCET analysis gives the input

Worst-Case Execution Time

- Measurement of execution time is not safe
  - Execution time is data dependent
    - Did we trigger the worst-case?
- Static WCET Analysis
- High-level WCET analysis is mature research
  - Considers control flow and flow facts (loop bounds)

Static WCET Analysis Issue

- Low-level analysis is the main issue
  - Modern processors are too complex
    - Lot of (hidden) state information
      - Key for performance
      - Issue for WCET analysis
- WCET analysis about 10 years behind processor technology

New Architectures Needed

- Design a computer architecture for real-time systems
  - WCET is the main design constraint
  - Average-case performance not (so) interesting
- Use and develop features that are
  - WCET analysis driven
  - Have a low WCET

Time-predictable Computer Architecture

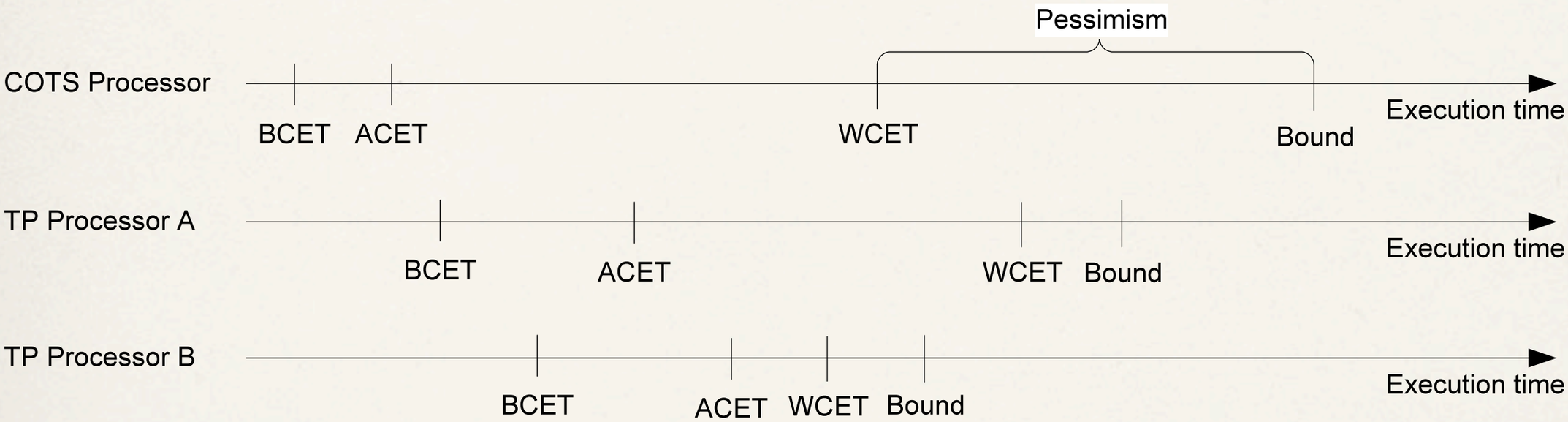
- Common CA wisdom:

Make the common case fast and
the uncommon case just correct

- Time-predictable CA:

Make the worst case fast and
the whole system analyzable

Our WCET Target Architecture



- Trying to catch up with the analysis on the complexity of average case optimized processors is not an option
- We need a sea change and take a constructive approach. Design processors for real-time systems!

The Patmos Design

- A time-predictable RISC/VLIW processor
- Explore VLIW vs. CMP tradeoff
- Explore JOP ideas on a C/C++/RISC target
- WCET aware compiler
  - Optimization for single-path programming

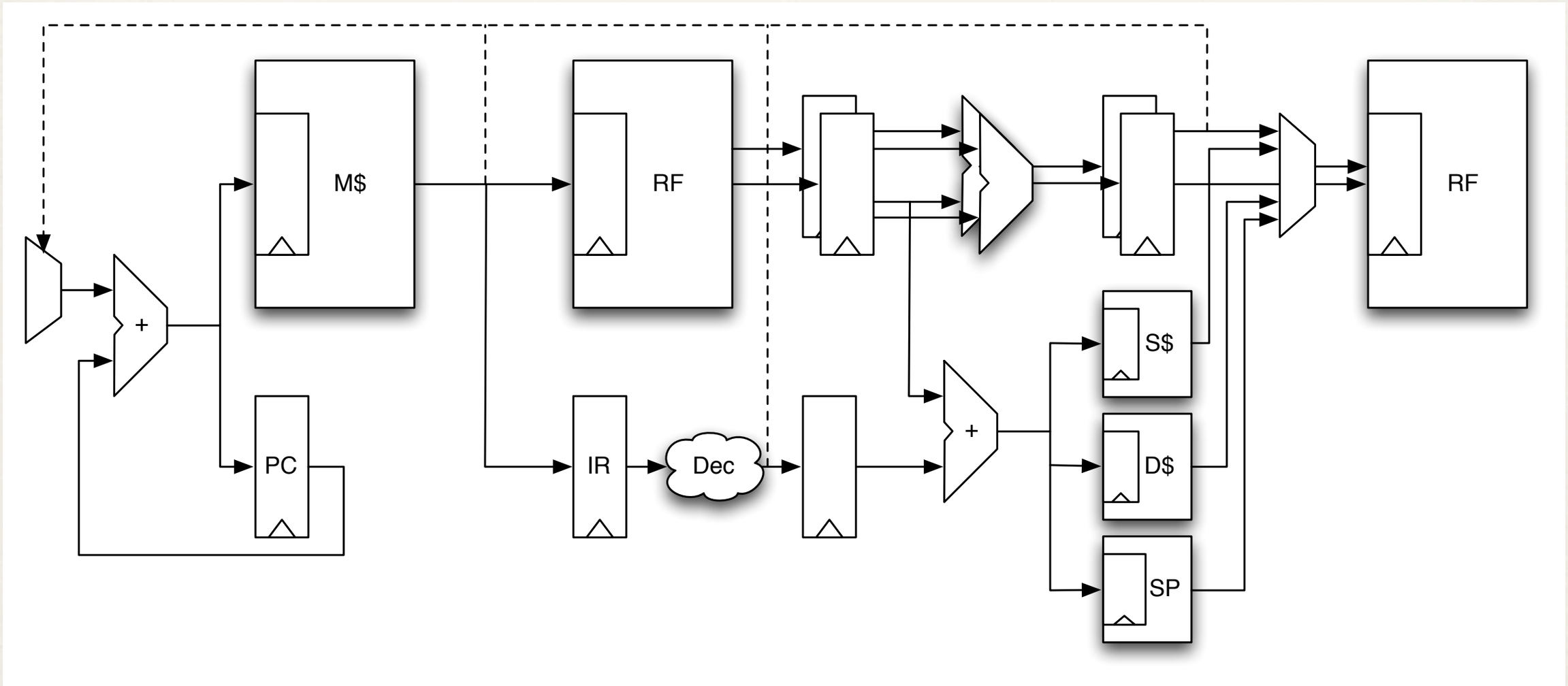
The Architecture

- 32-bit RISC style microprocessor
- Dual issue VLIW
  - Variable length instructions (32 and 64 bit)
- All instruction delays are visible
- No pipeline stalls
  - Except wait instruction for memory (split load)
- Split cache for data cache

Instruction Set

- Standard RISC instruction set
  - Global register file for both execution paths
- All instructions are predicated
  - Dual issue plus predicates support single-path programs
- Typed load/store instructions for split cache

The Pipeline



Software Development

- Adaption of the LLVM compiler framework
  - Support of Patmos ISA
- Explore typed load and store instructions
- Explore cache prefetching versus scratchpads
- WCET driven optimization
  - We will use AbsInt's WCET analysis tool aiT

Patmos as Research Platform

- A dual issue VLIW is nothing special, but a base line for time-predictable architecture research
- EU project T-CREST on time-predictable CMP
- Development of static scheduled network-on-chip
- Time-predictable DRAM controller
- Explore instruction set for single path programming
- Explore memory hierarchy
- WCET analysis for CMP

Conclusion

- Future real-time systems need new processor architectures
  - WCET optimized instead of average case optimization
    - Avoid the average-case performance trap
    - Only WCET analyzable features count
- Patmos processor to explore time-predictable architectures
- Patmos will be available in open source