Formal Verification of the VAMP Microprocessor (Project Status / Poster)

C. Berg, S. Beyer, C. Jacobi, D. Kröning, D. Leinenbach

{cb,sbeyer,cj,kroening,dirkl}@cs.uni-sb.de

Saarland University

Institute of Computer Architecture, Prof. W. J. Paul

Saarbrücken, Germany

VAMP VERIFIED ARCHITECTURE MICROPROCESSOR

Project Description

Project at Saarland University, Institute of Prof. W. J. Paul

- Verification of a complete CPU featuring
  - RISC Tomasulo core [Kröning]
  - IEEE compliant Floating Point Unit [Berg,Jacobi]
  - MMU: cache, virtual memory [Beyer]

- Designs based on textbook [Müller/Paul 00]
- Verification on the gate-level
- Verification uses PVS theorem prover
- Automatic translation to Verilog HDL [Beyer,Leinenbach]
- Implementation on Xilinx FPGA [Leinenbach, Meyer]

FPU

- Fully IEEE compatible design
- $+ - \times \div$, conversion, comparison
- Single/double precision
- Denormals handled in hardware
- All IEEE rounding modes, exceptions
- Modular design

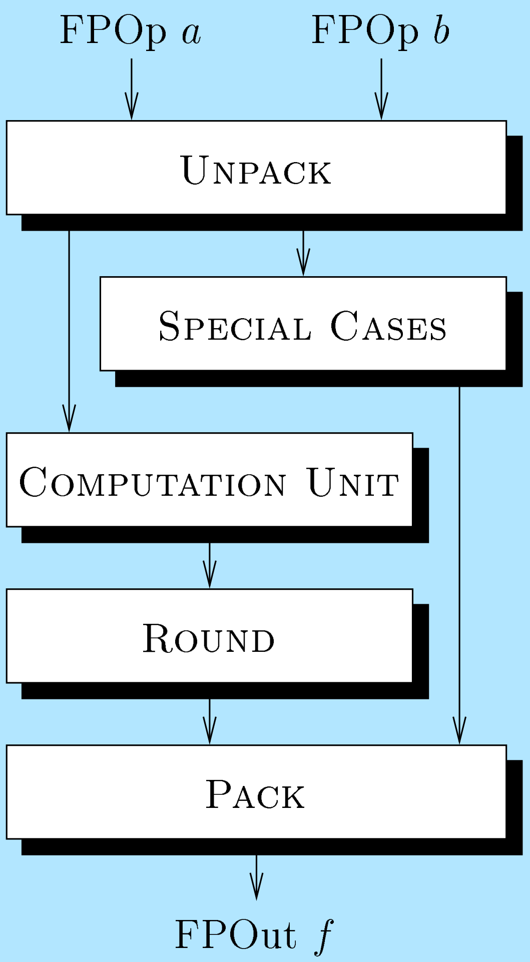
  $\rightarrow$ Generic unpacker

  $\rightarrow$ Computation unit

  $\rightarrow$ Generic rounder
- OOO pipeline to exploit Tomasulo scheduler advantages

  $\rightarrow$ Special cases on fast path

  $\rightarrow$ Fewer stalling for divisions



FPU Verification (1)

- Verification against high level description of IEEE standard 754
- Abstract IEEE numbers to factorings [MP00]

$$(s, e, f) \in \mathbb{B} \times \mathbb{Z} \times \mathbb{R}_0^+$$

  Value $[\![s, e, f]\!] := (-1)^s \cdot 2^e \cdot f$

  $\rightarrow$ normal, denormal, representable factorings

  representable iff $f \cdot 2^{P-1} \in \mathbb{N}$

- Partition $\mathbb{R}$ in equivalence classes $[\cdot]_\alpha \quad (\alpha = e - P)$

  $\rightarrow$ Equivalent numbers round to same IEEE number

- Computation unit outputs factoring α-equivalent to correct result

  $\rightarrow$ Verification of computation unit independent of rounder

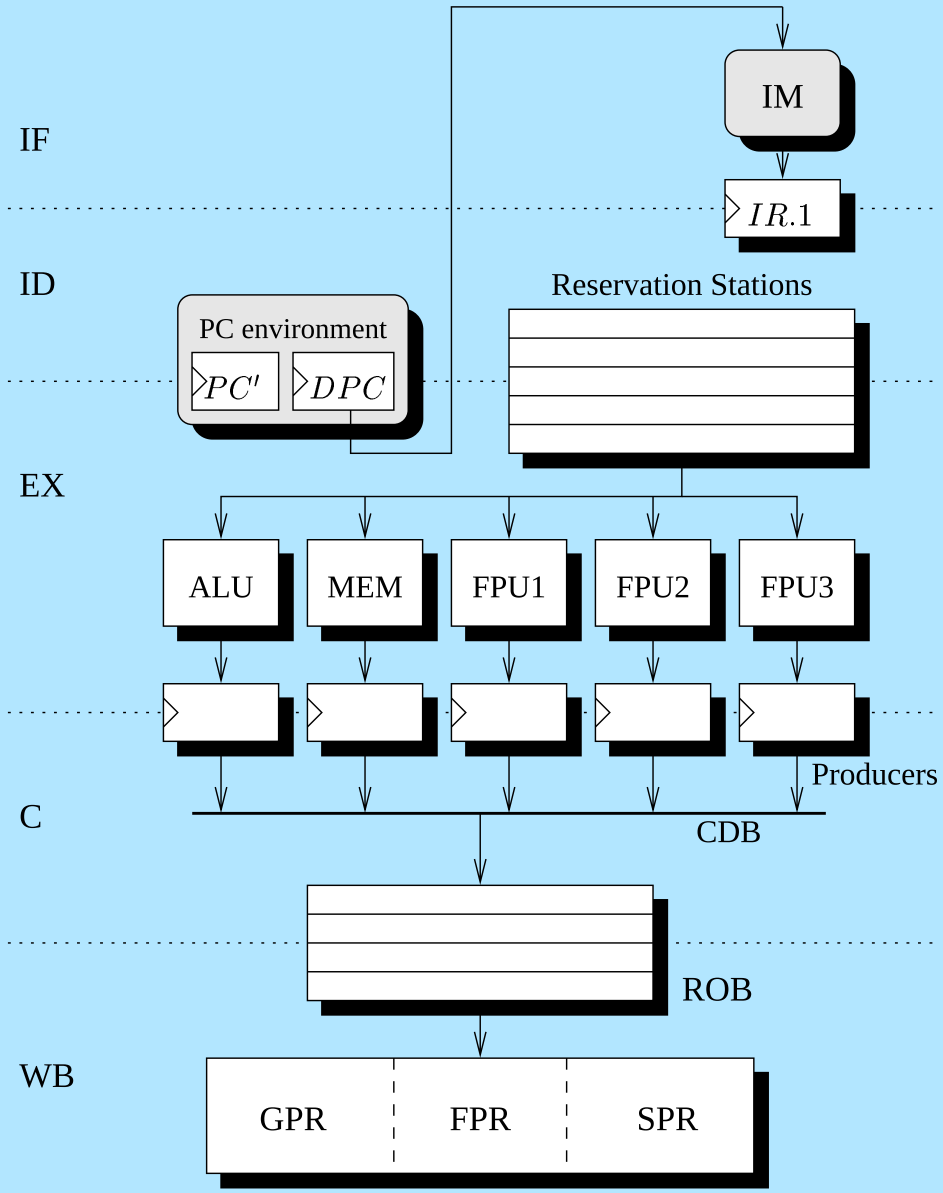
FPU Verification (2)

- New: Denormal Numbers
- α-equivalence $\equiv$ sticky-bit computations
- Decomposition of FPU in modules
- Complex computations units: $>$1 instruction, OOO pipe

  $\rightarrow$ Combine theorem proving and model checking
- *First FPU fully verified on gate level published*

CPU

- RISC instruction set (MIPS/DLX)
- Precise, nested interrupts
- Branch prediction, delayed branch
- Tomasulo scheduler
- Extends DLX from [Müller/Paul 00]



CPU Verification

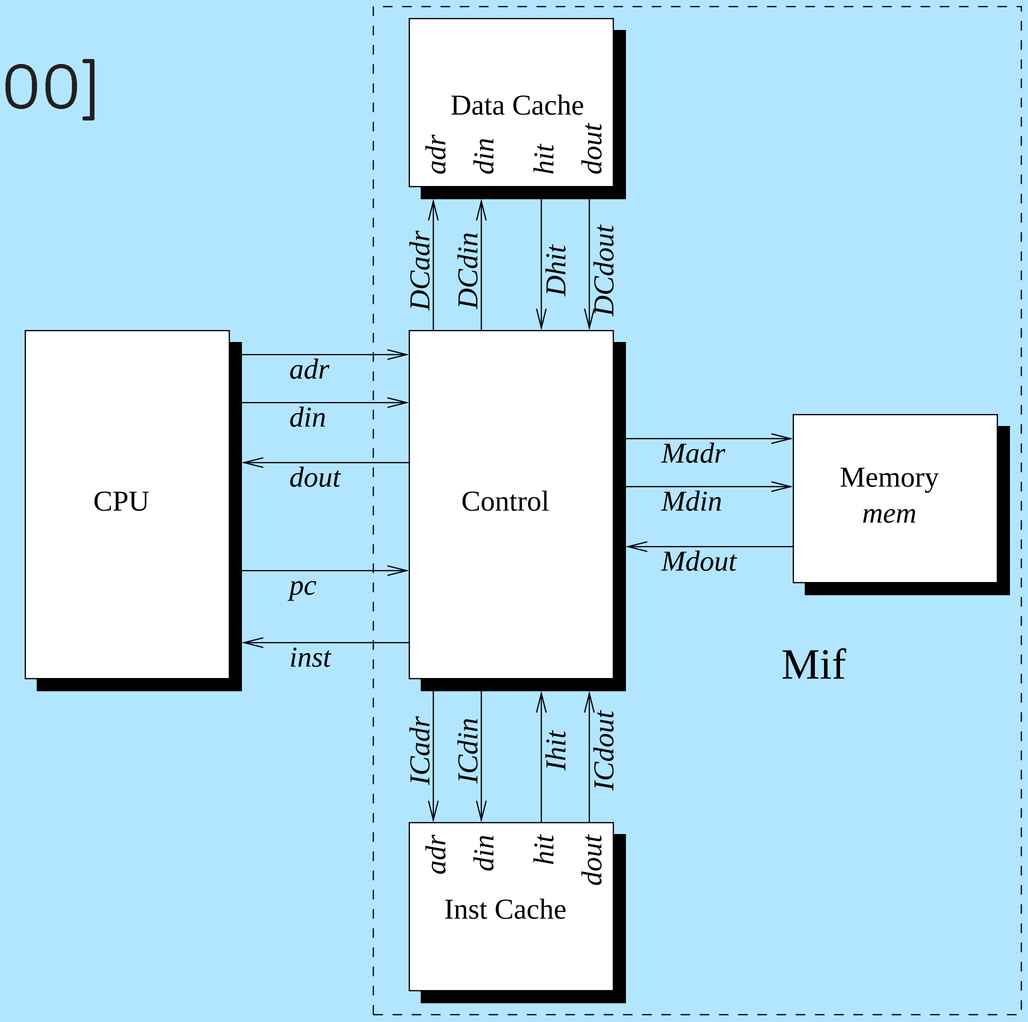
- Correctness criterion:
  simulate in-order DLX, 1 instruction/step
- Scheduling functions to identify stage instruction is in
- Forwarding: read from CDB or ROB
  $\rightarrow$ Use tag to identify instruction
- Liveness: all instructions terminate eventually
- *First CPU verified on gate level with these complex features*

Cache

- Split data/instruction cache

  $\rightarrow$ snooping for instruction cache updates
- Based on design from [Müller/Paul 00]
- Write-back strategy
- Correctness criterion:

  simulate non-cached memory
- Virtual memory later this year



HDL Translation

- Model combinatorial circuits in functional PVS language

  Use next-state function to model registers
- Unroll recursion
- Translate hardware to Verilog HDL
- Xilinx FPGA on PCI card with SD-RAM
- FPU running on FPGA, CPU expected within next weeks
- *No debugging of Verilog code!*

Summary

- A complete microprocessor with FPU and caches has been verified on the gate level
- The textbook [Müller/Paul 00] was used as design guideline
- Bugs in [Müller/Paul 00] were found

  → Confidence in verified design
- Verification effort:

  3 PhD theses: CPU, FPU, cache

  3 master theses: FP adder, HDL translator, software environment
- → Poster in MPI foyer

Publications

- Kröning, Paul: Automated Pipeline Design, DAC'01
- Berg, Jacobi: Formal Verification of the VAMP FPU, CHARME'01
- Jacobi: Formal Verification of Complex OOO Pipelines, submitted
- Beyer: Formal Verification of a Cache Memory Interface, submitted
- Beyer, Leinenbach: Correct HW by Synthesis from PVS, submitted