Formale Verifikation des VAMP-Mikroprozessors

Christoph Berg

cb@cs.uni-sb.de

Graduiertenkolleg Leistungsgarantien für Rechnersysteme

Lehrstuhl Paul, Universität des Saarlandes

In Zusammenarbeit mit

Beyer, Jacobi, Kröning, Leinenbach, Meyer

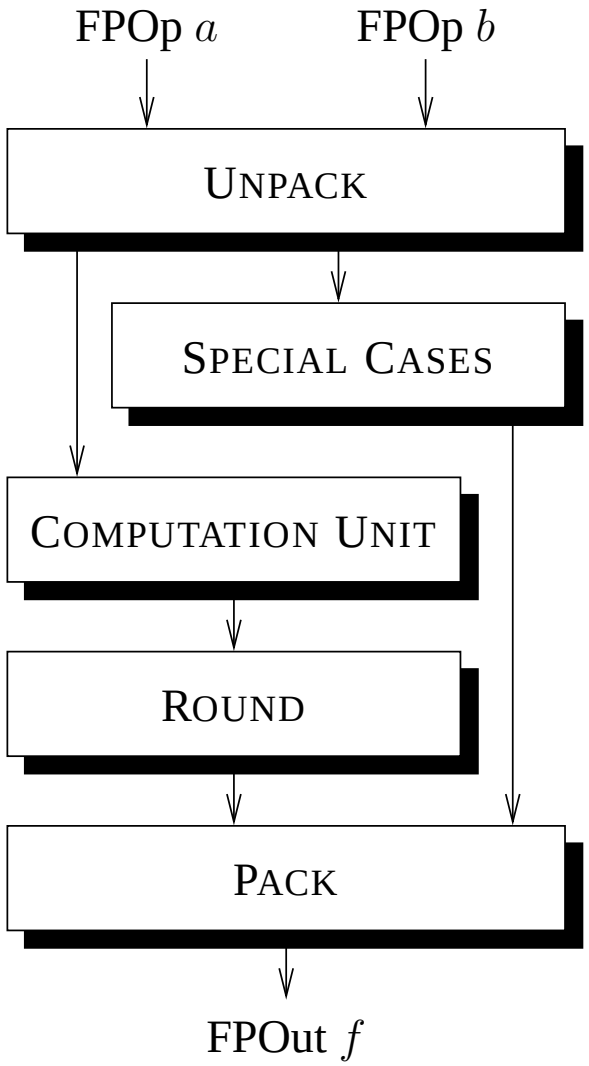
Project Description

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 (I, D) [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: Generic unpacker, Computation unit, Generic rounder
- OOO pipeline to exploit Tomasulo scheduler advantages
  - ▷ Special cases on fast path
  - ▷ Fewer stalling for divisions



FPU Verification (1)

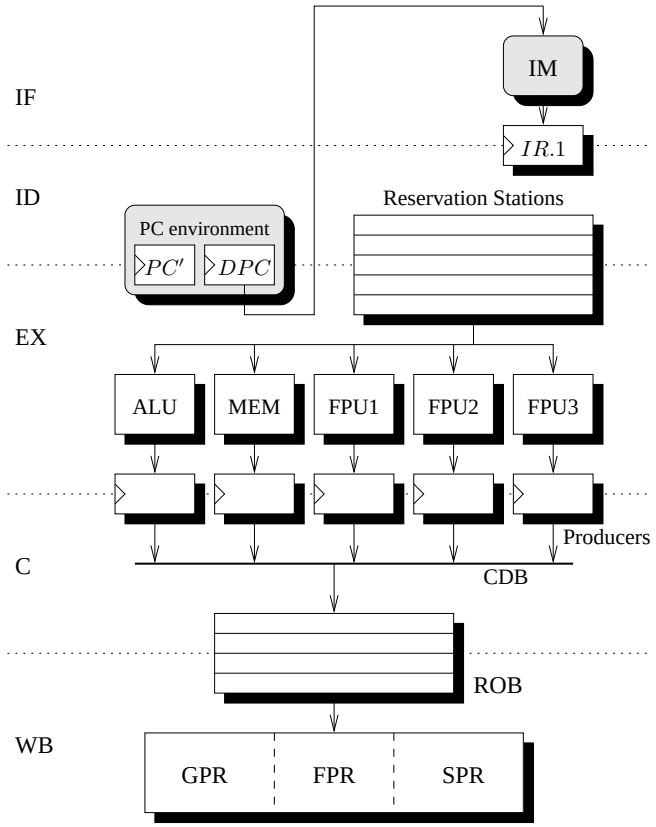
- Verification against formal description of IEEE standard 754
- Abstract IEEE numbers to factorings [MP00]
  $(s, e, f) \in \mathbb{B} \times \mathbb{Z} \times \mathbb{R}_0^+$, Value $[\![s, e, f]\!] := (-1)^s \cdot 2^e \cdot f$
  - ▷ normal, denormal, representable factorings
  - ▷ representable iff $f \cdot 2^{P-1} \in \mathbb{N}$
- Partition $\mathbb{R}$ in equivalence classes $[\cdot]_\alpha$ $\quad (\alpha = e - P)$
  - ▷ Equivalent numbers round to same IEEE number
- Computation unit outputs factoring α-equivalent to correct result
  - ▷ Computation unit verification 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 in pipe
  - OOO pipe
  - Combine theorem proving and model checking
- *First FPU fully verified on gate level published*

CPU

- RISC instruction set (MIPS/DLX)
- Precise, nested interrupts
- Delayed branch
- Speculative execution
- Tomasulo scheduler
- Extends DLX from [MP00]

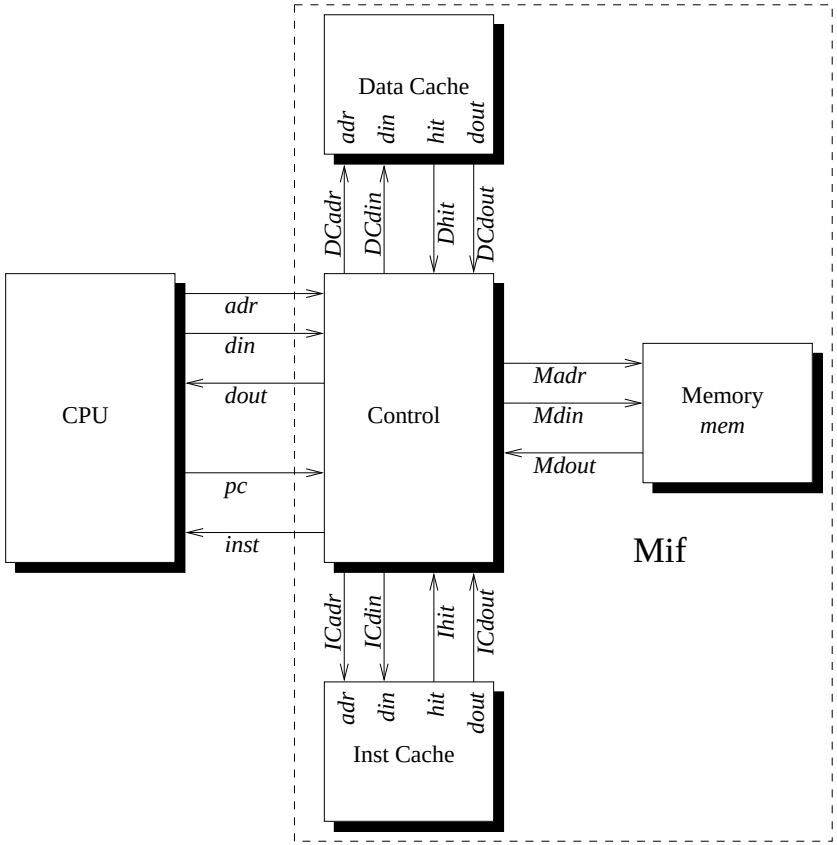


CPU Verification

- Correctness criterion:
  simulate in-order DLX, 1 instruction/step
- Scheduling functions to identify stage instruction is in
- Forwarding: read from CDB or ROB
  - Use tag to identify instruction
- Liveness: all instructions terminate eventually
- Tomasulo core verification completed
- Gate level implementation work in progress

Cache

- Split data/instruction cache
  - Snooping for instruction cache updates
- Fully parameterized
- Write-back strategy
- Based on design from [MP00]
- Correctness criterion:
  simulate non-cached memory
- Verification using
  theorem proving



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
- VAMP (CPU, FPU, Cache) running on FPGA
- *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 [MP00] were found
  - Confidence in verified design
- Verification effort:
  - 3 PhD theses: CPU, FPU, cache
  - 3 master theses: FP adder, HDL translator, software environment

Literature (1)

- [Müller/Paul 00] Computer Architecture: Complexity and Correctness. Springer, 2000.
- John Harrison: A machine checked theory of floating point arithmetic, TPHOL'99.
- Paul Miner: Defining the IEEE-854 floating-point standard in PVS, Tech Rep., 1995.
- J Moore et al.: A mechanically checked proof of the AMD5K86 floating point division program, IEEE Comp. 1998.
- David Russinoff: A mechanically checked proof [. . .] of the AMD-K7 processor, LMS JCM 1998.

Literature (2)

- Aagaard, Seger: The Formal Verification of a Pipelined Double-Precision IEEE Floating-Point Multiplier, ICCAD'95.
- Kaivola, Narasimhan: Formal Verification of the Pentium 4 Floating-Point Multiplier, DATE'02.
- Chen et al.: Verification of All Circuits in a Floating-Point Unit Using Word-Level Model Checking, FMCAD'96.

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, CAV'02
- Beyer: Formal Verification of a Cache Memory Interface, submitted