Chair of Electronic Design Automation
Department of Computer Engineering
School of Computation, Information and Technology
Technical University of Munich

Towards a Flexible, Fast and Accurate Software Performance Simulation Environment for RISC-V

Conrad Foik, Daniel Mueller-Gritschneder*, and Ulf Schlichtmann

Design Space Exploration

Simulation-based design space exploration (DES) to find **optimal** application-specific microarchitecture designs.

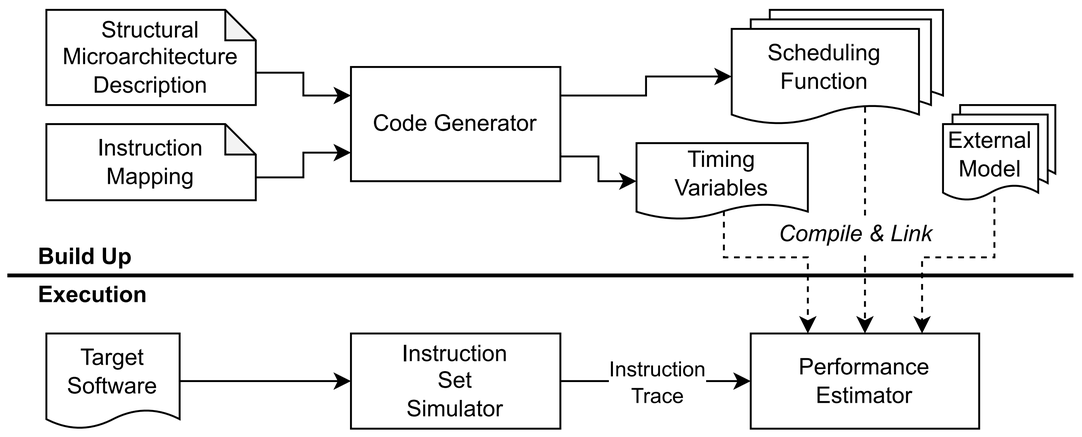
Requirements:

- **Accurate** performance estimates to evaluate variants
- High simulation **speed** to run exhaustive simulations
- **Flexibility** to adapt and evaluate new microarchitecture variants

Current Approaches:

- RTL simulation: Too slow for exhaustive simulations
- Instruction Set Simulators (ISS): Inaccurate, no performance variations visible
- Performance Simulators: Inflexible, focus on single target processor

Methodology



Code Generator:

- *Scheduling function*: Model instruction type specific timing behavior
- *Timing variables*: Represent availability of processor's components

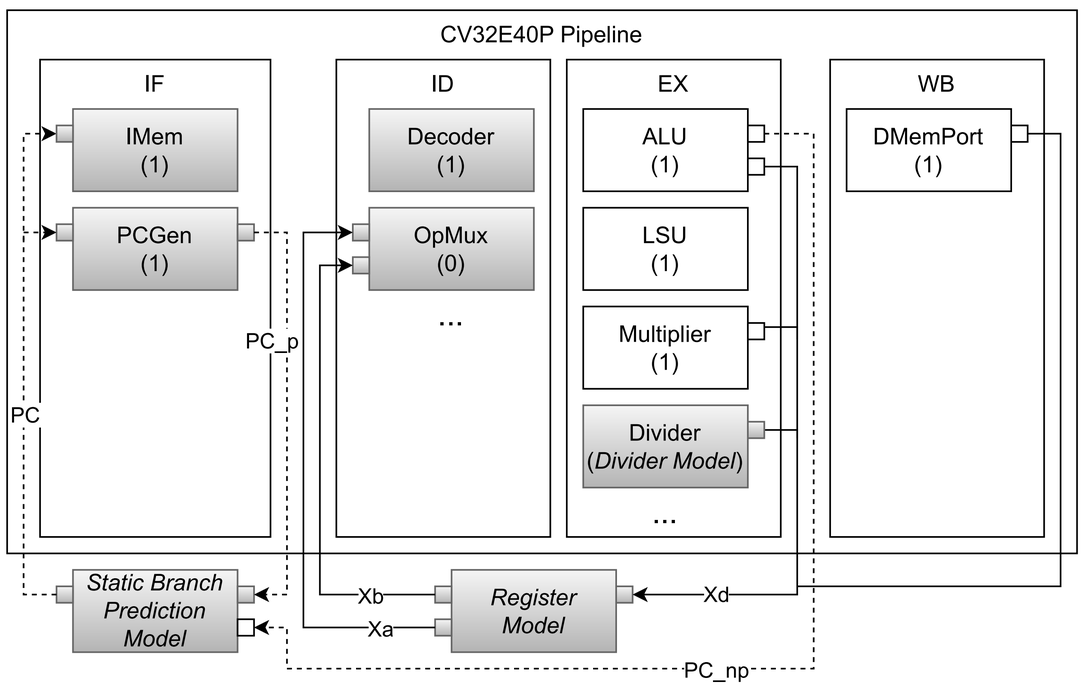
Performance Estimator:

- Instruction-trace-based: Portable to existing ISS.
- Per instruction: Select scheduling function to update timing variable
- Custom *external models* of dynamic behaviors (e.g. branch prediction)

Input Description

CorePerfDSL [1]:

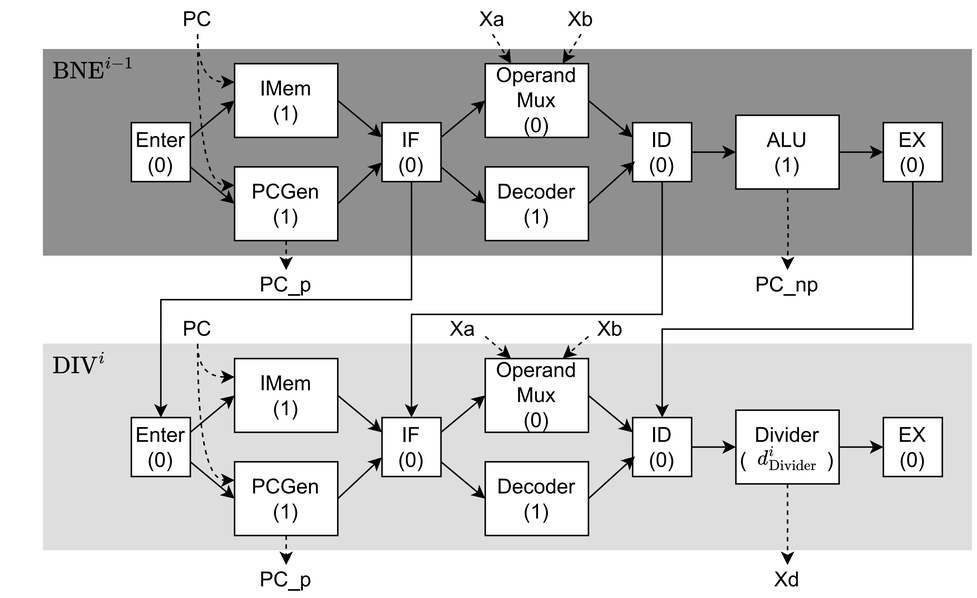
- **Structural** (non-functional) description of microarchitecture
- **Instruction mapping** assigning instructions to resources
- Focus: **Compact** representation and high **flexibility**



Code Generation

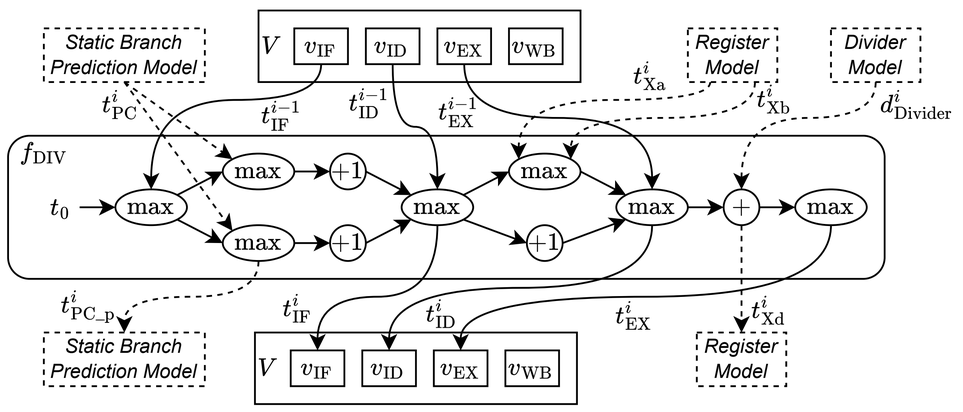
Timing Constraints: Automatically derived from input description

- **Instruction-internal** dependencies (e.g. $t^i_{IF} \geq t^i_{IMem}$)
- **Cross-instruction** dependencies (e.g. $t^i_{IF} \geq t^{i-1}_{ID}$)
- **Dynamic** dependencies (e.g. $t^i_{PCGen} \geq t^i_{PC} + 1$)



ASAP Scheduling:

- Simple algebraic scheduling functions (max, +)
- Single value timing variables (over-written by each scheduling function)



Experimental Results

Setup:

- Target processor: **CV32E40P**
- Instruction set: RV32IM
- Target SW: Embench suite
- ISS: ETISS [2]

Results:

- Accuracy: **>99%**
- Sim. Speed: up to **24 MIPS**
- Error per instr.: $\sim$ **0.0001 CCs**

Benchmark	Num. Instr.	RTL Cycles	RTL CPI	ISS[a] Error	ISS + Performance Estimator Cycles	ISS + Performance Estimator CPI	ISS + Performance Estimator Error
aha-mont64	4,534,817	5,313,414	1.17	14.7%	5,313,364	1.17	0.000941%
crc32	4,183,376	4,881,112	1.17	14.3%	4,881,062	1.17	0.00102%
cubic	6,854,333	8,189,299	1.19	16.3%	8,189,249	1.19	**0.000611%**
edn	3,447,366	4,132,445	1.20	16.6%	4,132,395	1.20	0.00121%
huffbench	2,302,258	2,963,665	1.29	22.3%	2,963,615	1.29	0.00169%
matmult-int	3,602,152	4,377,284	1.22	17.7%	4,377,234	1.22	0.00114%
minver	2,509,246	3,581,566	1.43	29.9%	3,581,516	1.43	0.00140%
nbody	3,177,393	3,783,510	1.19	16.0%	3,783,460	1.19	0.00132%
nettle-aes	4,406,553	4,628,360	1.05	4.79%	4,628,310	1.05	0.00108%
nettle-sha256	3,965,640	3,991,463	**1.01**	**0.647%**	3,991,413	1.01	0.00125%
nsichneu	2,241,955	3,619,320	1.61	38.1%	3,619,270	1.61	0.00138%
picojpeg	3,595,305	4,215,381	1.17	14.7%	4,215,331	1.17	0.00119%
qrduino	2,822,182	3,487,235	1.24	19.1%	3,487,185	1.24	0.00143%
sglib-combined	2,344,372	3,290,199	1.40	28.7%	3,290,149	1.40	0.00152%
slre	2,376,079	2,920,434	1.23	18.6%	2,920,384	1.23	0.00171%
st	3,969,153	4,905,662	1.24	19.1%	4,905,612	1.24	0.00102%
statemate	2,098,097	2,305,566	1.10	9.00%	2,305,516	1.10	0.00217%
ud	923,462	2,034,580	**2.20**	**54.6%**	2,034,530	2.20	0.00246%
wikisort	1,056,343	1,413,159	1.34	25.2%	1,413,109	1.34	**0.00354%**
Average			**1.29**	**20.0%**		**1.29**	**0.00148%**

[a] Assuming CPI=1

Publications

[1] C. Foik, D. Müller-Gritschneder, and U. Schlichtmann. "CorePerfDSL: A Flexible Processor Description Language for Software Performance Simulation". In *Forum on Specification & Design Languages (FDL)*. 2022

[2] D. Müller-Gritschneder, M. Dittrich, M. Greim, K. Devarajegowda, W. Ecker, and U. Schlichtmann. "The Extendable Translating Instruction Set Simulator (ETISS) Interlinked with an MDA Framework for Fast RISC Prototyping". In *International Symposium on Rapid System Prototyping (RSP)*. 2017

* Prof. Müller-Gritschneder is now working with the:
Institute of Computer Engineering
TU Wien

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) and ITEA within the project GenerIoT under contract no. 01IS22084G

SPONSORED BY THE
Federal Ministry of Education and Research

Contact:
conrad.foik@tum.de
daniel.mueller-gritschneder@tuwien.ac.at

Open Source:
https://github.com/tum-ei-eda