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

A Flexible Simulation Environment for RISC-V

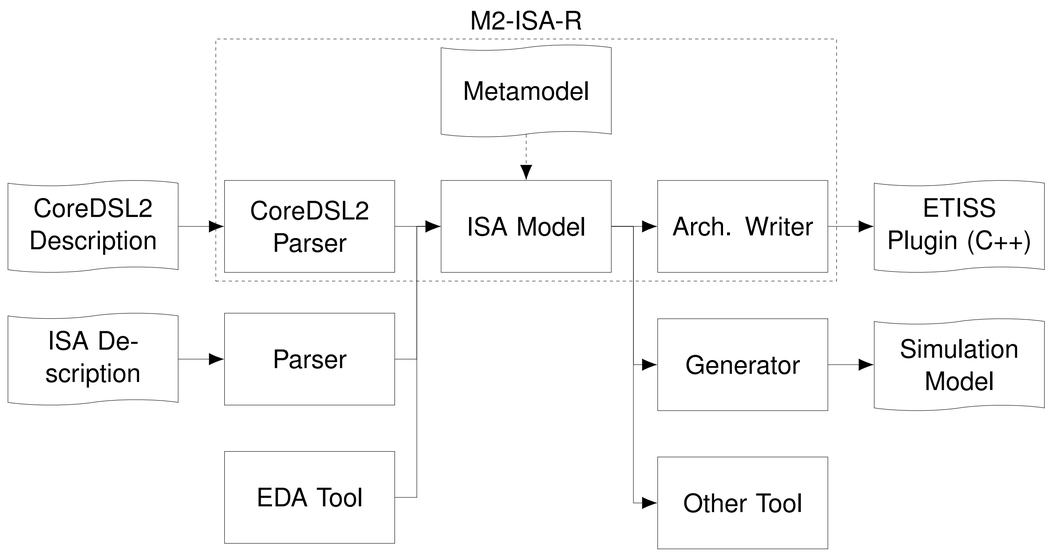
Karsten Emrich[1], Conrad Foik[1], Johannes Kappes[2], Sebastian Prebeck[2], Daniel Mueller-Gritschneder[1], Wolfgang Ecker[2], Ulf Schlichtmann[1]
[1] Chair of Electronic Design Automation, Technical University of Munich
[2] Infineon Technologies AG

ETISS: Extendable Translating Instruction Set Simulator

Key Features [1]:

- ISA-Independent instruction set simulator core
- Fast dynamic binary translation execution engine, based on a C-language intermediate format
- Easy integration into SystemC / TLM VPs
- Open Source with permissive license model (BSD 3-clause)
- Plugin mechanism with full access to various parts of the execution loop and memory interfaces -> easy integration of tracing, debugging, timing and memory models without needing to touch the actual simulation loop
- Dhrystone MIPS: ~110 for bare RV32GC model

M2-ISA-R: ISA Meta-Model and Code Generator



M2-ISA-R:

- M2-ISA-R provides a Python-based metamodel for describing arbitrary ISAs
- A two-step framework is provided for fast creation and generation of ISA extensions or new ISAs:
  - A front-end or producer produces a M2-ISA-R model hierarchy
  - Back-end or consumer tools to generate e.g. simulation models
- A reference front-end for the ISA description DSL CoreDSL 2 as well as a model generator backend for ETISS are provided

CoreDSL [2]:

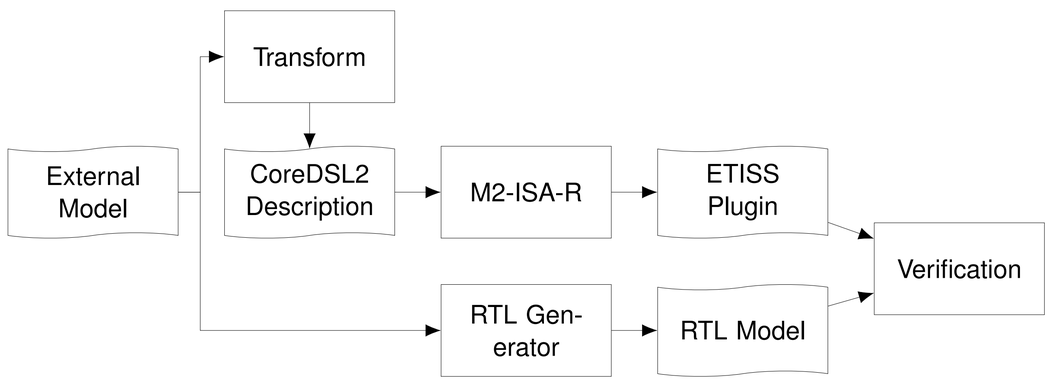
- DSL for modeling complex ISAs
- Features / Support:
  - Human-readable, simple instruction description
  - Parameterization of Cores (accounting for e.g. bit-width)
  - Consumer-independent descriptions

Sample CoreDSL instruction description

```
CMV {
    encoding: b100 :: b0 :: rd[4:0] ::
  rs2[4:0] :: b10;
    assembly: "{name(rd)}, {name(rs2)}";
    behavior: if (rd != 0) X[rd] = X[rs2];
}
```

Cross Verification:

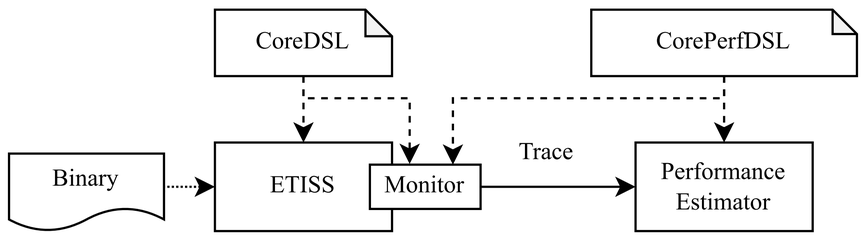
- Existing EDA tools can output CoreDSL files directly if Python-based output is infeasible
- CoreDSL output allows broader possibilities of code reuse, also in other tools
- Coherent descriptions for verification during HW-/SW-Codesign



Future Work:

- Goal: Generate datasheet-accurate simulation models from a single CoreDSL 2 description
- Missing pieces:
  - Support of exception and interrupt behavior
  - Core-close peripherals (MMU, MPU, interrupt controller, SysTick)

Performance Estimation



Performance Estimation:

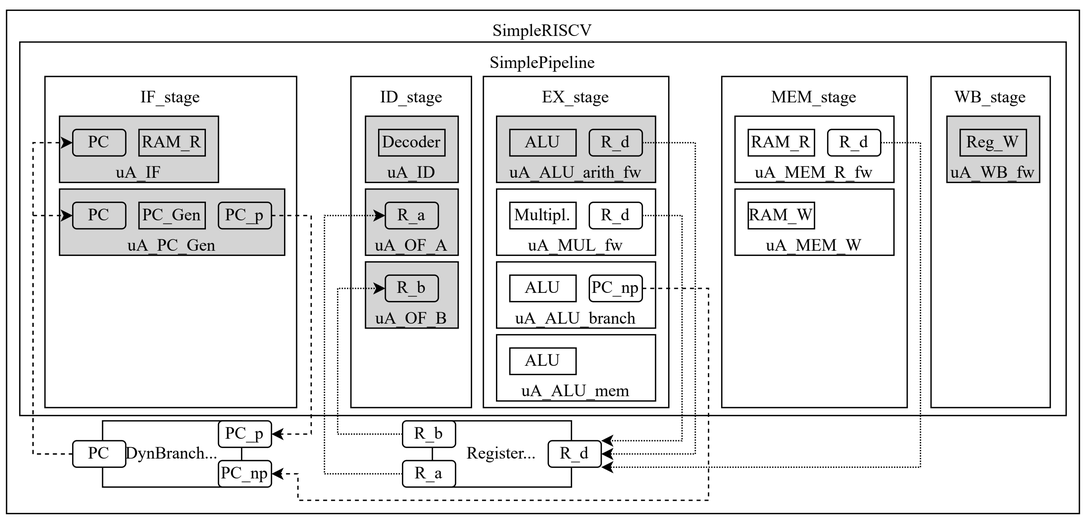
- Detailed timing information lost on ISS-level
- Estimation of software performance based on instruction trace
- CorePerfDSL to automatically set up estimator

CorePerfDSL [3]:

- Describes Performance Model
- Features / Support:
  - Single-issue, in-order pipelines
  - Structural, data and control hazards
  - External models for complex behavior
  - Generic trace definition
  - High flexibility for architectural exploration

Extract of a CorePerfDSL description

```
Connector {PC, PC_p,...}
Resource {RAM_R, PC_Gen,...}
Microaction{
    uA_PC_Gen(PC->PC_Gen->PC_p),
    uA_IF    (PC->RAM_R),
 ...
}
Stage{
    IF_stage (uA_PC_Gen, uA_IF),
    ID_stage (...),
 ...
}
Pipeline SimplePipeline
    (IF_stage -> ID_stage -> ...)
```



Experimental Results:

Architecture	Forwarding	Branch prediction	Est. target perf. (CPI)	Simulation speed (MIPS)
Harvard	No	No	1.59	15.07
		Static	1.52	13.97
		Dynamic	1.48	12.27
	Yes	No	1.23	15.09
		Static	1.15	13.92
		Dynamic	1.11	12.72
Von Neumann	No	No	1.68	6.67
		Static	1.60	6.44
		Dynamic	1.57	6.13
	Yes	No	1.36	6.75
		Static	1.30	6.42
		Dynamic	1.27	6.08

Future Work:

- Support of advance pipeline features (multi-issue, buffer, compressed instructions, etc.)
- Increased simulation speed
- Integration / adaptation to other simulators

References

[1] Daniel Mueller-Gritschneder, Martin Dittrich, Marc Greim, Keerthikumara Devarajegowda, Wolfgang Ecker, and Ulf 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

[2] MINRES Technologies. "CoreDSL2.0". https://www.minres/work/coredsl/. Accessed: 15.03.2023

[3] Conrad Foik, Daniel Mueller-Gritschneder, and Ulf Schlichtmann. "CorePerfDSL: A Flexible Processor Description Language for Software Performance Simulation". In: *Forum on Specification & Design Languages (FDL)*. 2022

This work has been developed in the ZuSE project Scale4Edge. Scale4Edge is funded by the German ministry of education and research (BMBF) (reference numbers: 16ME0122K-16ME0140+16ME0465). The authors are responsible for the content of this publication.

SPONSORED BY THE Federal Ministry of Education and Research

Contact:
daniel.mueller@tum.de
wolfgang.ecker@infineon.com

Open Source:
https://github.com/tum-ei-eda/etiss
https://github.com/tum-ei-eda/m2-isa-r
https://github.com/tum-ei-eda/CorePerfDSL