GenSim

A Toolset for Binary Translation and Emulation

Harry Wagstaff, Tom Spink, Bruno Bodin, Bjoern Franke

Institute for Computing Systems Architecture
University of Edinburgh

October 2018

Introduction

Elevator Pitch

Hands-On Peek

This tutorial includes a hands-on - participation will require a Linux machine (several common distributions are supported).

To participate, please go to http://gensim.org/download and install the dependencies for your particular distribution.

GenSim History

Ongoing project from University of Edinburgh

- 2011	Arcsim Simulator
2011 - 2014	GenSim PhD Project
2014 - 2018	PAMELA Project
2018 -	Open Source

What are we doing with GenSim?

We use the GenSim toolset in the following research areas:

- Dynamic Binary Translation
- GPU Simulation
- ADL Design and Implementation
- DSP/VLIW Simulation
- Cross-architecture Virtualisation

Structure of the Tutorial

- Talk 1: Introduction (1 Hour)
  - Brief Introduction to Simulation
  - Overview of GenSim and Tools
- Hands-On Session (90 Minutes)
  - Downloading & Installing GenSim
  - Using GenSim/ArchSim to perform experiments
- Talk 2: Building a Model (1 Hour)
  - More detailed look at GenSim
  - Future plans & Conclusion

In This Talk

Introduction

Instruction Set Simulation
- Introduction to Simulation
- Architecture Description Languages

GenSim
- The GenSim ADL
- The ArchSim Simulator
- Captive

Conclusion

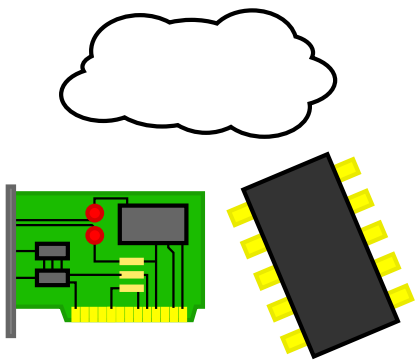
Instruction Set Simulation

What is Instruction Set Simulation?

Instruction Set Simulation

What is Instruction Set Simulation?

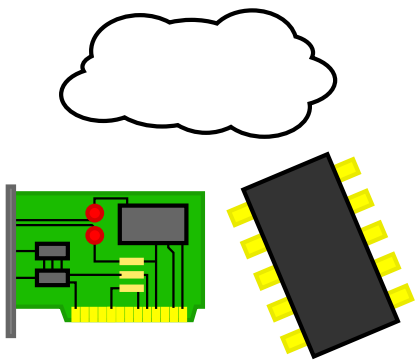
- Model a partial or complete computer system

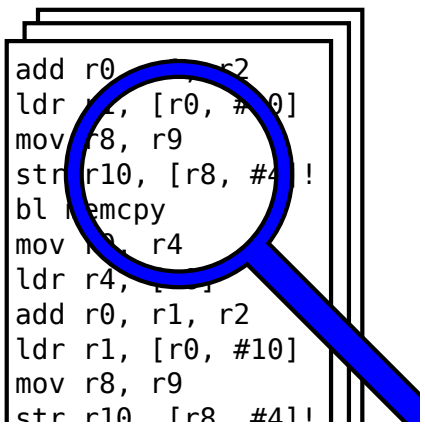


Instruction Set Simulation

What is Instruction Set Simulation?

- Model a partial or complete computer system
- Potentially add instrumentation or perform analysis

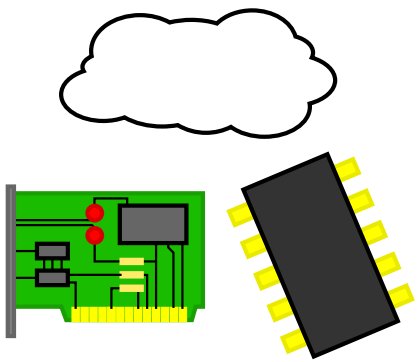


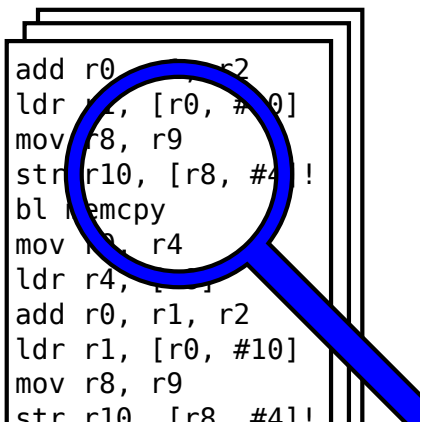


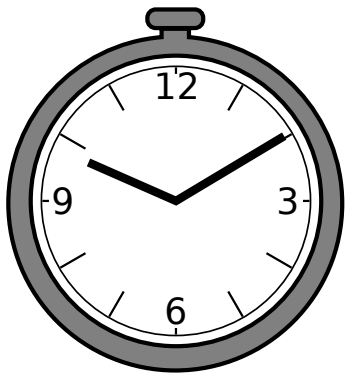
Instruction Set Simulation

What is Instruction Set Simulation?

- Model a partial or complete computer system
- Potentially add instrumentation or perform analysis
- May be timing accurate



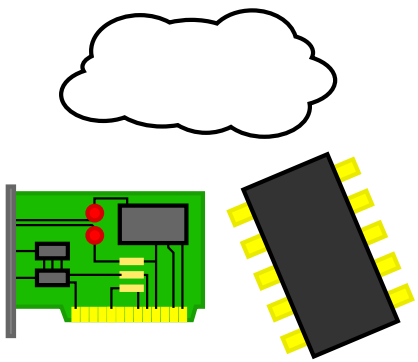


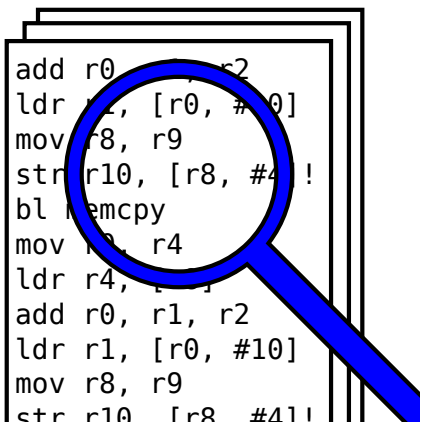


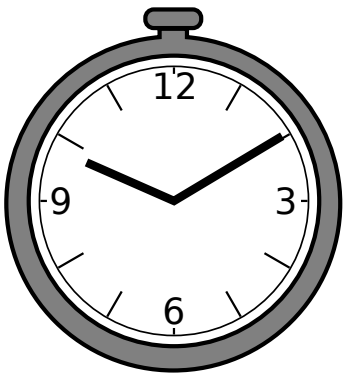
Instruction Set Simulation

What is Instruction Set Simulation?

- Model a partial or complete computer system
- Potentially add instrumentation or perform analysis
- May be timing accurate







Note that this also covers associated technologies!

Instruction Set Simulation

Instruction Set Simulation is used in a wide variety of contexts:

Instruction Set Simulation

Instruction Set Simulation is used in a wide variety of contexts:

- Design Space Exploration

Instruction Set Simulation

Instruction Set Simulation is used in a wide variety of contexts:

- Design Space Exploration
- Software Development

Instruction Set Simulation

Instruction Set Simulation is used in a wide variety of contexts:

- Design Space Exploration
- Software Development
- Backwards Compatibility

Simulation Technologies

Broadly, there are two software-based simulation technologies:

Technology	Slowdown	Complexity
Interpretation	1000x	Low
Binary Translation	10x	High

Simulation Technologies

Broadly, there are two software-based simulation technologies:

Technology	Slowdown	Complexity
Interpretation	1000x	Low
Binary Translation	**10x**	**High**

Simulation Technologies

Broadly, there are two software-based simulation technologies:

Technology	Slowdown	Complexity
Interpretation	1000x	Low
Binary Translation	**10x**	**High**

Can we get the **speed** of BT without the **complexity**?

Example QEMU Instruction

```
target_long imm = sextract64(inst, 20,12);
int rs1 = extract32(inst, 15, 5);
int rd  = extract32(inst, 7, 5);
TCGv source1 = tcg_temp_new();
gen_get_gpr(source1, rs1);
tcg_gen_addi_tl(source1, source1, imm);
gen_set_gpr(rd, source1);
tcg_temp_free(source1);
```

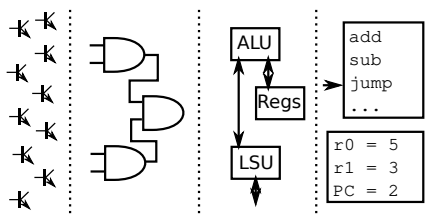
Architecture Description Languages

ADLs are useful tools for systems development. They let us:

Architecture Description Languages

ADLs are useful tools for systems development. They let us:

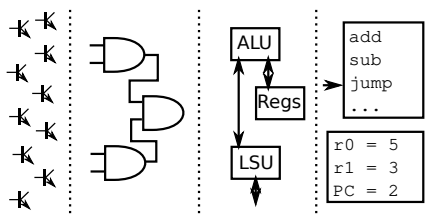
- ...describe systems at different levels of abstraction

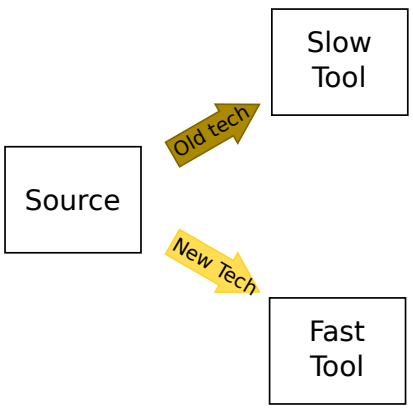


Architecture Description Languages

ADLs are useful tools for systems development. They let us:

- ...describe systems at different levels of abstraction
- ...use new technologies without rewriting all of our tools

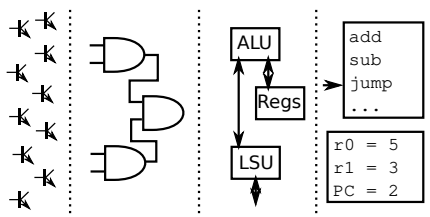


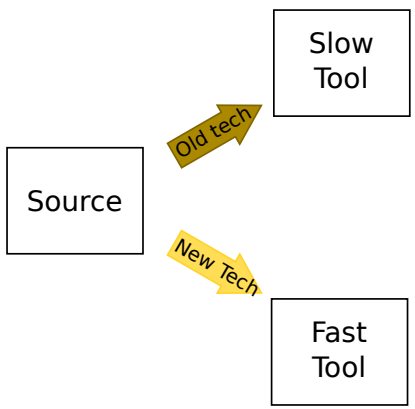


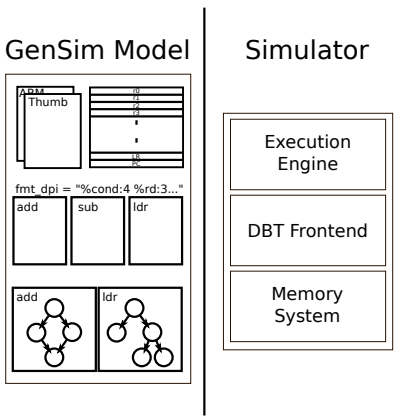
Architecture Description Languages

ADLs are useful tools for systems development. They let us:

- ...describe systems at different levels of abstraction
- ...use new technologies without rewriting all of our tools
- ...separate the behaviour of tools from their implementation



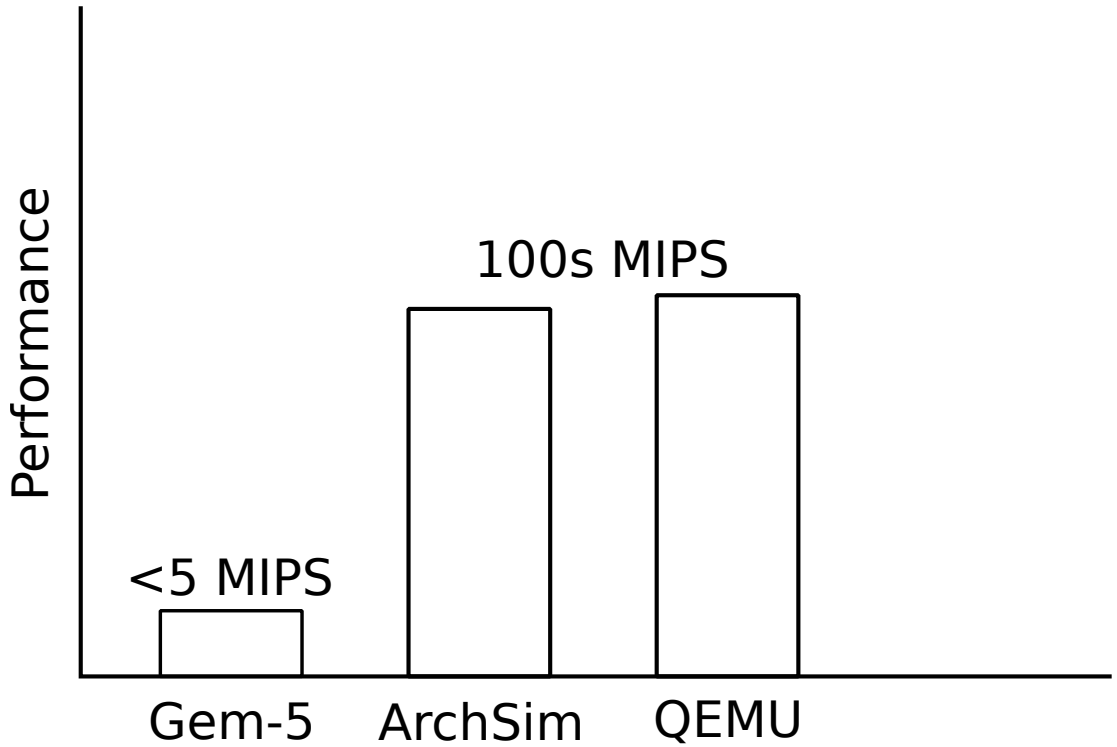




QEMU/ADL Comparison

```
target_long imm = sextract64(inst, 20,12);
int rs1 = extract32(inst, 15, 5);
int rd  = extract32(inst, 7, 5);
TCGv source1 = tcg_temp_new();
gen_get_gpr(source1, rs1);
tcg_gen_addi_tl(source1, source1, imm);
gen_set_gpr(rd, source1);
tcg_temp_free(source1);
```

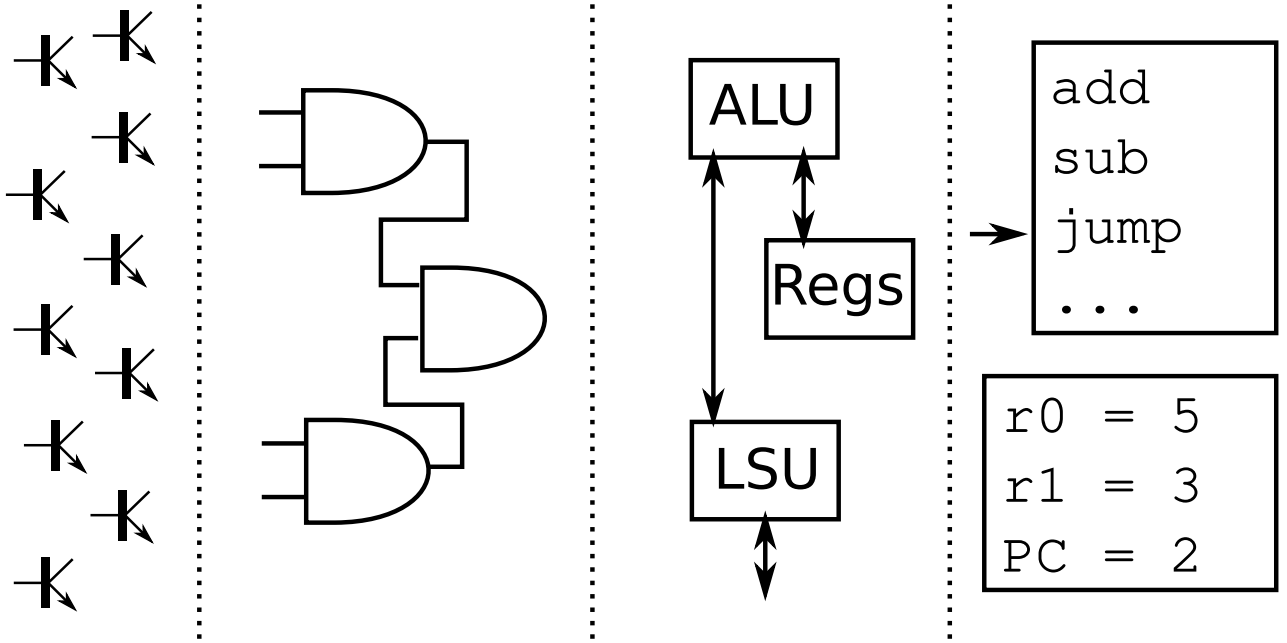
```
sint32 imm = inst.imm;
imm <<= 20;
imm >>= 20;

sint32 rs = read_register_bank(inst.rs1);

rs += imm;

write_register_bank(GPR, inst.rd, rs);
```


Performance
<5 MIPS
100s MIPS
Gem-5
ArchSim
QEMU

Architecture Description Languages

We consider ADLs mostly in the context of simulation

- Low level ADLs are more suited to detailed simulation
- More abstract ADLs are better suited to fast simulation



The GenSim ADL

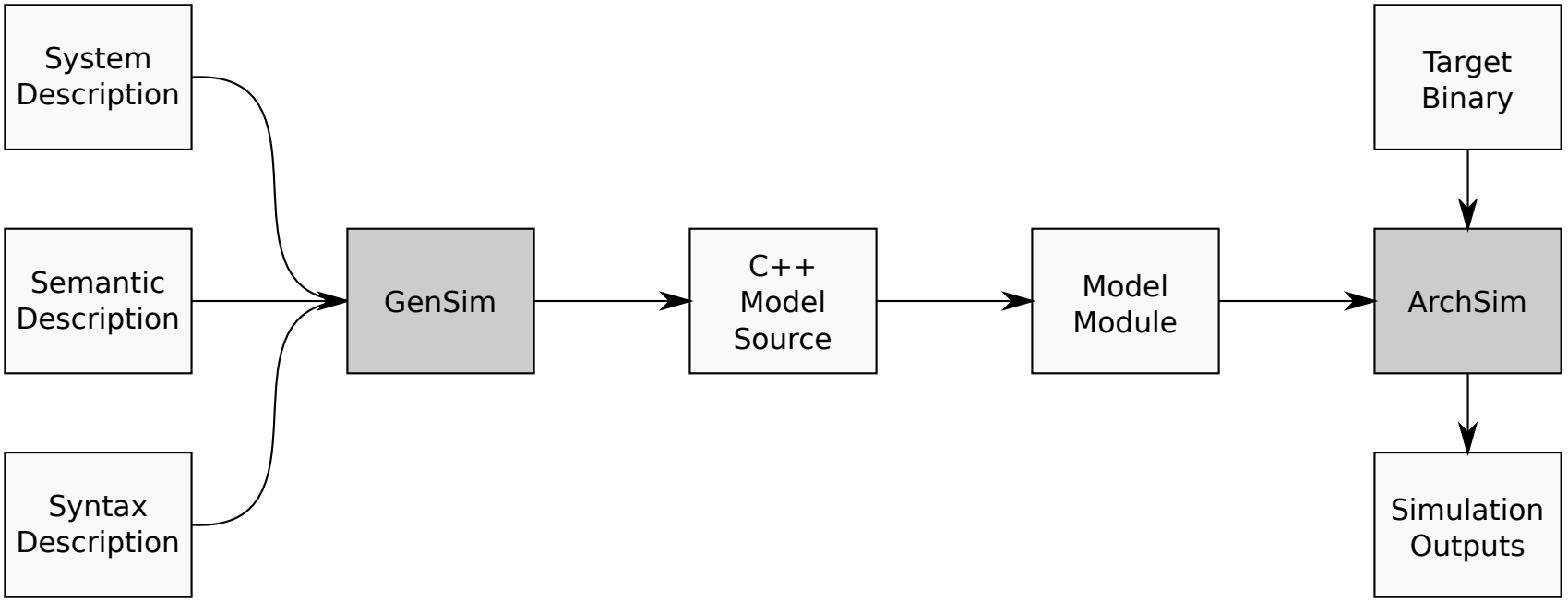
The GenSim ADL

GenSim is the name of our ADL toolset. It is designed to:

- ... be intuitive and easy to learn
- ... create high performance simulation tools[1]
- ... be extensible in terms of analyses

[1] DAC'13: https://doi.org/10.1145/2463209.2488760

GenSim Toolflow



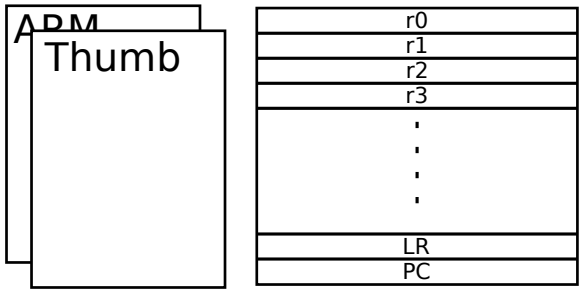
GenSim Model Components

A GenSim description consists of three components

GenSim Model Components

A GenSim description consists of three components

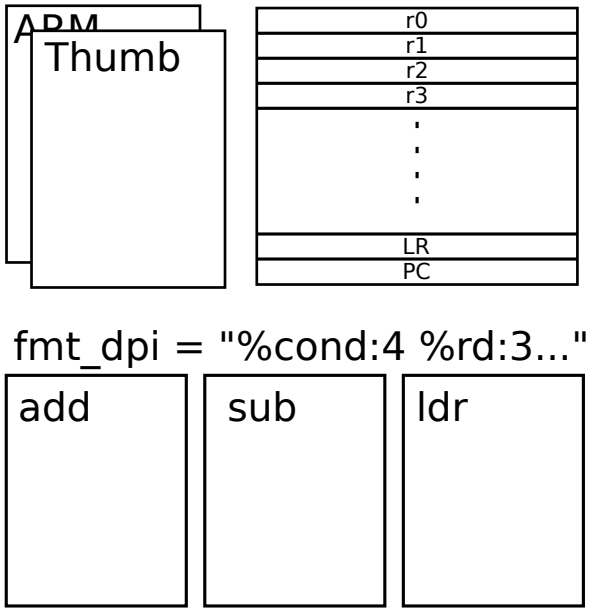
- A ‘System’ component
  - Available instruction sets
  - Register file layout
  - Configurable Features



GenSim Model Components

A GenSim description consists of three components

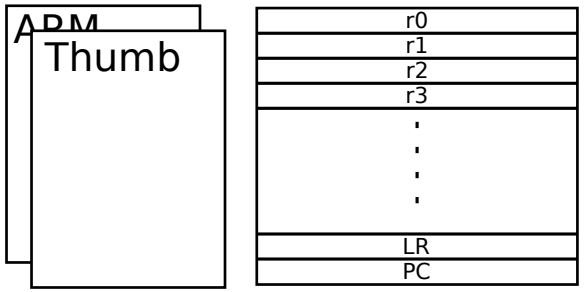
- A ‘System’ component
  - Available instruction sets
  - Register file layout
  - Configurable Features
- A ‘Syntax’ component
  - Instruction formats
  - Instruction encoding

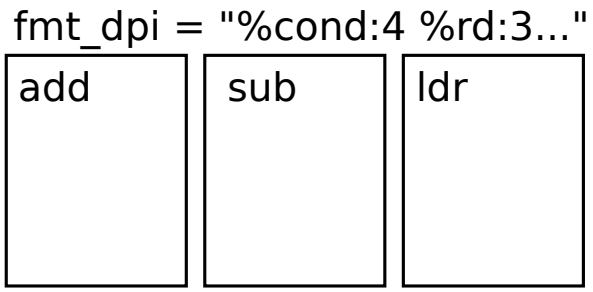


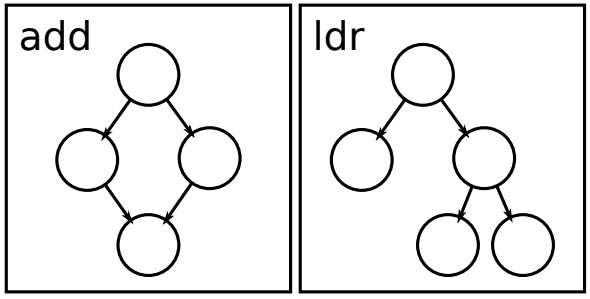
GenSim Model Components

A GenSim description consists of three components

- A ‘System’ component
  - Available instruction sets
  - Register file layout
  - Configurable Features
- A ‘Syntax’ component
  - Instruction formats
  - Instruction encoding
- A ‘Semantics’ component
  - Instruction behaviours
  - Exception behaviour







Available Models

- ARMv7-A
- ARMv8
- RISC-V
- x86-64

Available Models

- ARMv7-A
  - ARM + Thumb-2
  - Some VFP and NEON
  - User Mode + Full System
- ARMv8
- RISC-V
- x86-64

Raspberry Pi Model B

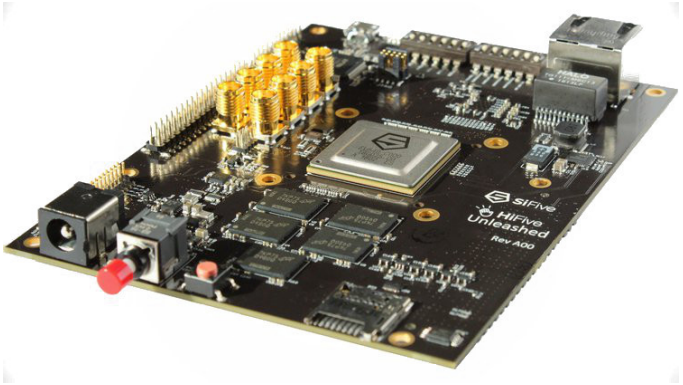
Available Models

- ARMv7-A
- ARMv8
  - AArch64 Only
  - Some FP Support
  - User Mode + Full System
- RISC-V
- x86-64

Raspberry Pi 3 Model B+

Available Models

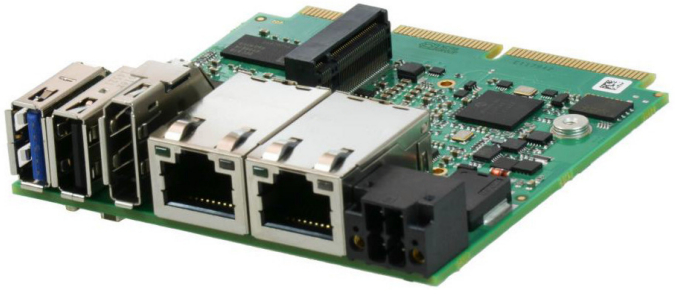
- ARMv7-A
- ARMv8
- RISC-V
  - Core and FP
  - User Mode only
- x86-64



SiFive RISC-V SBC

Available Models

- ARMv7-A
- ARMv8
- RISC-V
- x86-64
  - External decoder
  - User Mode only



ADL ADLE3800SEC

Execution Model

GenSim generally has a simplistic execution model

- Instructions have a straightforward encoding
- Instructions are executed in PC order
- Effects occur immediately
- Instructions can be predicated

Execution Model

GenSim generally has a simplistic execution model

- Instructions have a straightforward encoding
- Instructions are executed in PC order
- Effects occur immediately
- Instructions can be predicated

Which architectures do not fit this model?

- x86 (stateful encoding)
- MIPS (delay slots)
- DSP-like architectures (delayed effects)

The ArchSim Simulator

The ArchSim Simulator

ArchSim is our research simulation platform. It is highly modular and configurable.

- User Mode & Full System Simulation
- Modular in terms of processors, devices, and platforms
- High speed trace system

Modularity

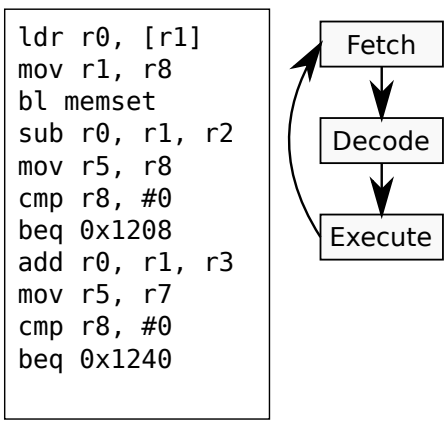
ArchSim can accept a variety of modules

- GenSim processor modules
- External devices
- Execution Engines

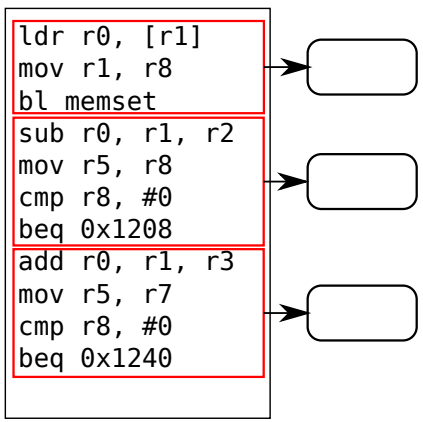
This allows processor models, devices, and new simulation technologies to be developed as closed-source and still be used by ArchSim.

Execution Framework

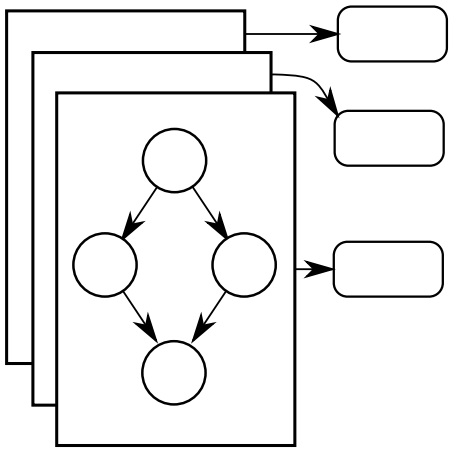
ArchSim supports a variety of execution methods



Interpretation

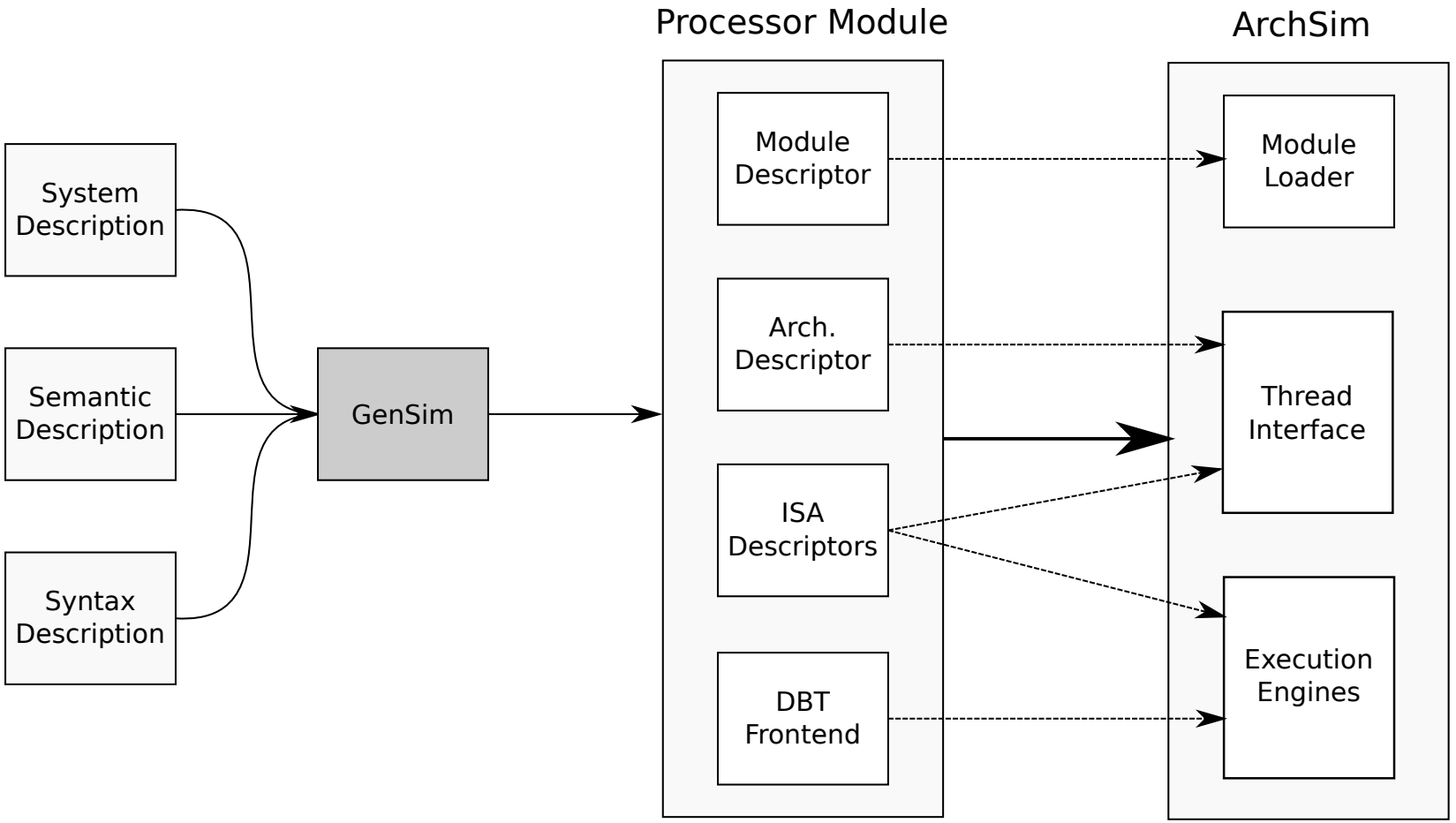


Block-based DBT



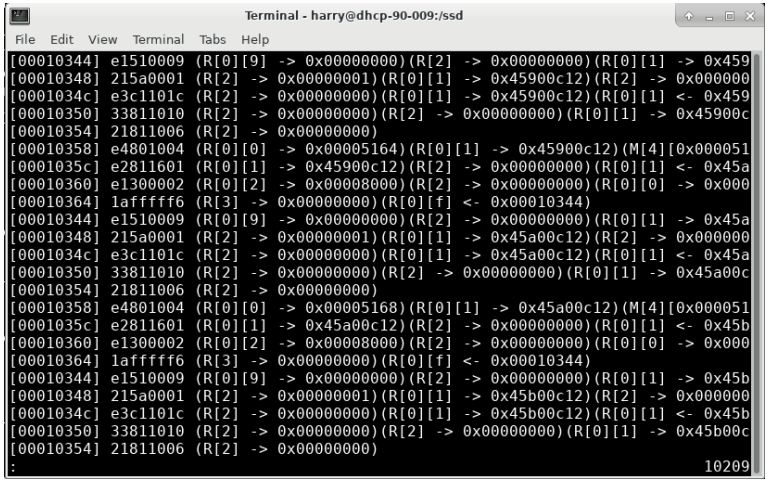
Region-based DBT

Execution Framework



High Speed Tracing

- High performance (>1MIPS)
- Full architectural trace
- Easily extensible
- Tools for manipulation & visualisation

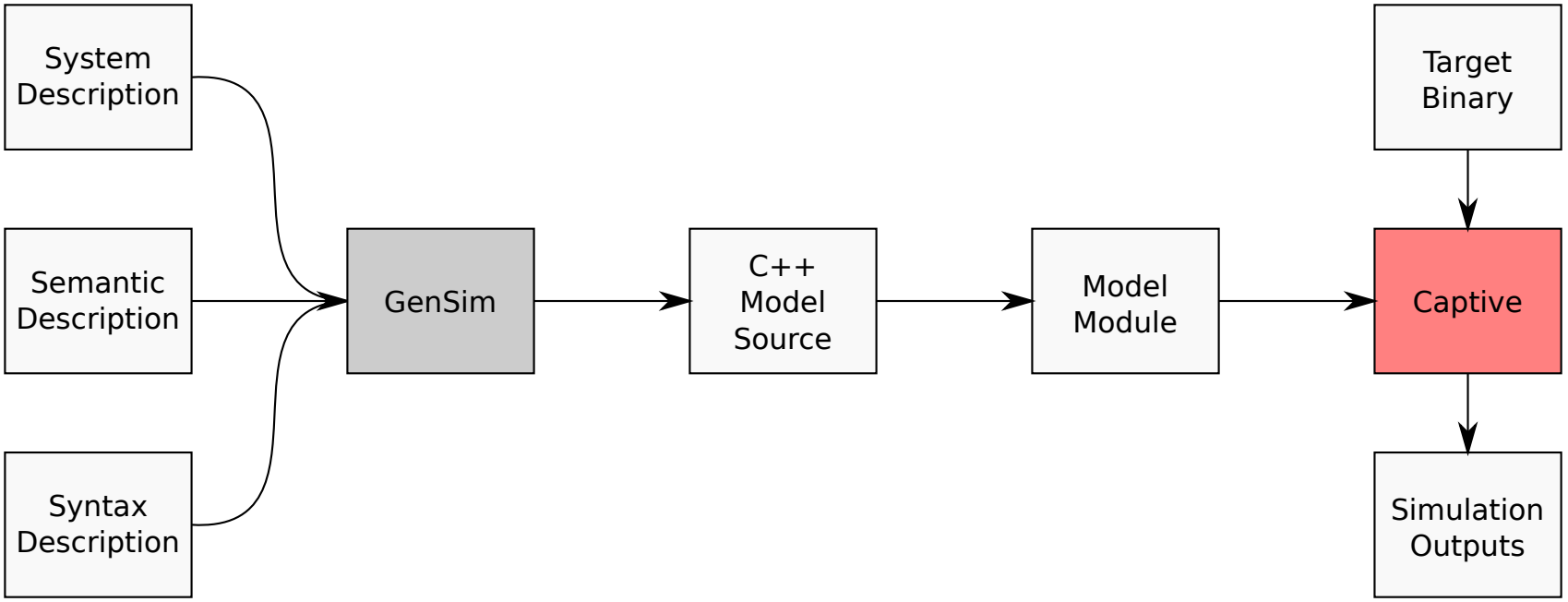


Short Demo

(https://www.youtube.com/watch?v=aZXx17oYumc)

The Captive Cross-Architectural Hypervisor

The Captive Cross-Architectural Hypervisor



The Captive Cross-Architectural Hypervisor

Alternative full-system simulation tool which uses GenSim ADL models[2]

- Uses x86 virtualisation features
- Supports ARMv7 and ARMv8 (AArch64)
- Some host machine devices can also be used

[2] TACO: http://doi.acm.org/10.1145/2996798

MMU Virtualisation

In full-system simulation, one of the largest performance costs is virtual-physical address translations.

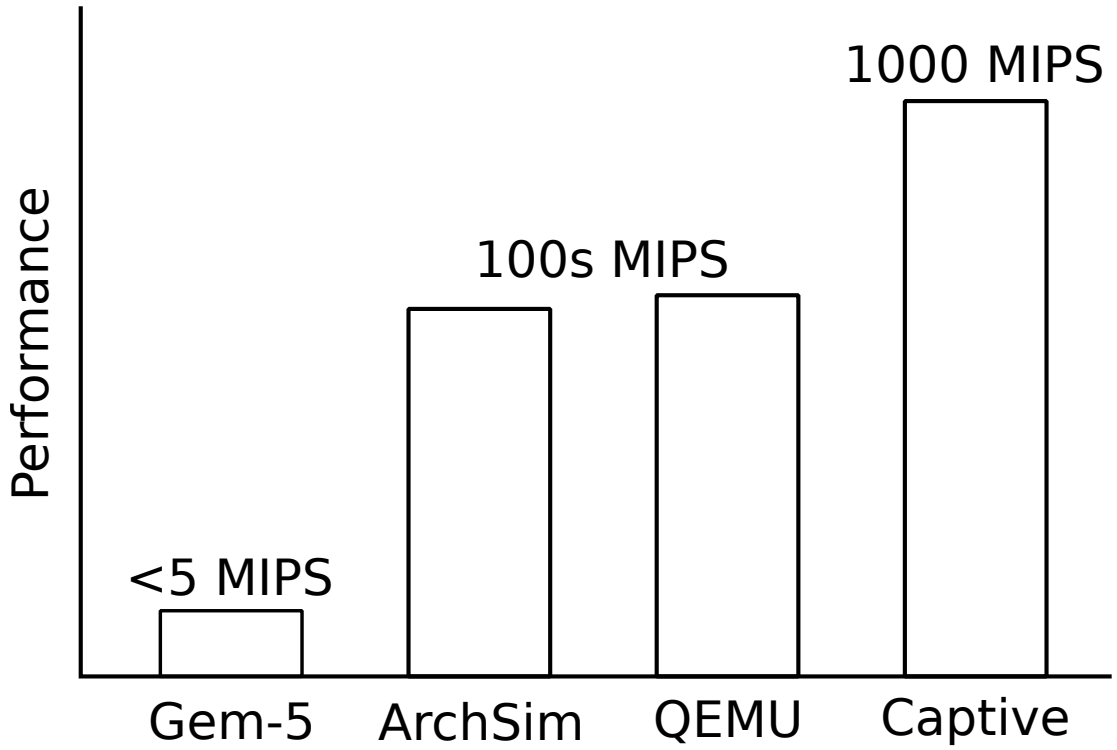
- Need to check privileges
- Perform actual address translation
- Trigger exception if necessary

Most simulators speed up this process using a software cache

MMU Virtualisation in Captive

Captive avoids this cost by using the host MMU to perform translations:

- The host page table is set up to mimic the guest page table
- Host privilege levels used to track guest privilege levels
- Host memory faults can be used to model guest faults


Performance
<5 MIPS
100s MIPS
1000 MIPS
Gem-5
ArchSim
QEMU
Captive

Captive - Host Virtualisation Features

Many possible x86 features which could be used:

- Intel-VT (or AMD-V)
- Memory Protection Keys
- Process Context IDs
- Privileged Execution
- Interrupts/Exceptions
- ...

Future Plans

GenSim is still a work in progress!

- Better support for decoding unusual/CISC architectures
- Dependent Typing
- Better documentation
- New models and platforms

Recap

Recap

- Instruction Set Simulation is useful

Recap

- Instruction Set Simulation is useful
- GenSim can be used to build binary translators

Recap

- Instruction Set Simulation is useful
- GenSim can be used to build binary translators
- Multiple supported GenSim models

Recap

- Instruction Set Simulation is useful
- GenSim can be used to build binary translators
- Multiple supported GenSim models
- Multiple supported simulation tools

Hands-On Session