Simulation of RISC-V based Systems in gem5

by Robert Scheffel
Agenda

• Introduction
• Full-System Simulation in gem5
• The RISC-V Extension Parser
• Conclusion
Introduction
Goodix

Connected Smart Things

Innovative Biometric Solutions

Complete Human Interface Solutions
Commsolid at a Glance

• **Company mission**
  • Provide ultra-low-power IoT solutions for the fast emerging multi billion market

• **Engineering team**
  • Experienced team of 30+ engineers
  • Former Intel’s LTE team in Dresden/Germany
  • Successful track record in LTE product development
  • 20+ years of competence in wireless, 10+ years in LTE

• **NB-IoT product SoC in IoDT testing**
Architectural Evaluations

- deficiency of cycle-accurate full system simulators
- poor support for custom extensions
- extend gem5’s basic RISC-V support to close this gap

Full system simulation of RISC-V based Systems with custom extensions
gem5

- free, open source architecture simulator
- two different simulation modes
- multiple ISA support
- **RISC-V:**
  - syscall emulation mode
  - RV62GC

<table>
<thead>
<tr>
<th>ISA</th>
<th>Level of ISA support</th>
<th>Full-system support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>high</td>
<td>Linux</td>
</tr>
<tr>
<td>ARM</td>
<td>high</td>
<td>Linux, BSD, Android</td>
</tr>
<tr>
<td>MIPS</td>
<td>low</td>
<td>None</td>
</tr>
<tr>
<td>RISC-V</td>
<td>medium</td>
<td>None</td>
</tr>
<tr>
<td>x86</td>
<td>medium</td>
<td>Linux, BSD</td>
</tr>
</tbody>
</table>
# gem5 System definition

- system definition in python files
  - configure parameters of object in python
  - functionality is implemented in c++
- choose from many implemented objects

<table>
<thead>
<tr>
<th>Processor</th>
<th>Memory System</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Model</td>
<td>System Mode</td>
</tr>
<tr>
<td>Atomic Simple</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>FS</td>
</tr>
<tr>
<td>Timing Simple</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>FS</td>
</tr>
<tr>
<td>In-Order</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>FS</td>
</tr>
<tr>
<td>O3</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>FS</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
</tbody>
</table>
Full-System Simulation in gem5
Requirements

- simulation of unmodified RISC-V binaries in gem5
- simulation including peripherals
  - UART
  - timer
- 32 Bit architecture
Implementations in gem5

- Full-System mode in gem5 for RISC-V
  - for bare-metal systems
- interrupt handling
  - needed by peripherals
  - according to privileged architecture
- Rv32IMC
The RISC-V Extension Parser
Proof of Concept

• Is it possible to conveniently define custom extensions and simulate them in gem5?

• Requirements:
  • Define custom instructions
  • Define opcodes and cycle counts
  • Define custom registers
  • Extend toolchain
  • Extend gem5
Interface for Custom Extension

Extensions → Extension Parser → Toolchain Patches → Toolchain → Binary → Simulation with Custom Extensions

Plug-In → gem5
The RISC-V Extension Parser as gem5 plug-in

- custom decoder
  - accessed by the RISC-V decoder
- custom timing information
  - used by functional units of the CPU model
gem5 ISA decoder

- domain specific language to describe instructions

```c
decode OPCODE {
    0: add({{ Rc = Ra + Rb; }});
    1: sub({{ Rc = Ra - Rb; }});
}
```

- parsed with gem5 ISA parser:
  - one instruction in the ISA is represented by one c++ class
  - generation of decoding function
Simulations of Custom Instructions

- generation of custom decoder in DSL
- gem5 ISA parser is used to translate custom decoder into c++
Conclusion
Conclusion

- simulation of unmodified RISC-V binaries
- support for RV32IMC instructions
- proof of concept for simulation of custom extensions

Links:
- gem5 GitHub repo: https://github.com/gem5/gem5
- RISC-V Extension Parser: https://gitlab.com/_rob_/riscv-extension
- gem5 with integrated Extension Parser: https://gitlab.com/_rob_/gem5/tree/riscv
  - be careful: the extension parser is located in ext/ folder
  - the integration happened on branch riscv, did not merge the master for a longer time
Thank you