Architecture Exploration of RISC-V Processor and Comparison with ARM Cortex A53

Karthikeyan S
Architecture Modeling Intern
Mirabilis Design
karthi.sugumar@gmail.com

Tom Jose
Application Specialist
Mirabilis Design
tjose@mirabilisdesign.com
Problem

- Limited devices for understanding RISC-V
- No software platform for observation

Solution

- Ease of understanding using System-Level Modeling Technique
- VisualSim Architect provides a platform for observation and analysis
Features of VisualSim

- Develop and Test Processor designs using Processor Generator Package
- Establish and Observe current and upcoming real-world applications
- Compare and benchmark a variety of hardware and software implementations
RISC-V Processor Modeling

Let’s start with how we used VisualSim Architect to model the Processor
RISC-V ISA

- Built using VisualSim’s Processor Generator Technology
- Specifications of SiFive’s E31 Core was referred
- RV32I is the ISA used

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>1</td>
</tr>
<tr>
<td>MUL</td>
<td>2</td>
</tr>
<tr>
<td>DIV</td>
<td>Min: 2 Max: 33</td>
</tr>
<tr>
<td>LW, SW</td>
<td>2</td>
</tr>
</tbody>
</table>
RISC-V Processor Specs

- Processor Bits – 32
- ISA – RV32I
- Clock Speed – 500 MHz
- Pipeline type – In-Order
- Pipeline Stages - 5
- Cache – 32 KBytes of I-Cache and D-Cache
  - 64 KBytes of L2 Cache
Representation of System-Level RISC-V Processor
## Task Profile

### NpBench: Benchmarking Suite for Network Processors was used to generate the task profile

<table>
<thead>
<tr>
<th>Task</th>
<th>Number of Instructions</th>
<th>Integer/Floating (%)</th>
<th>Load/Store (%)</th>
<th>Shift (%)</th>
<th>Logic (%)</th>
<th>Branch (%)</th>
<th>Others (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
<td>276</td>
<td>15</td>
<td>30</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>FRAG</td>
<td>573</td>
<td>38</td>
<td>14</td>
<td>4</td>
<td>13</td>
<td>27</td>
<td>4</td>
</tr>
<tr>
<td>RED</td>
<td>802</td>
<td>38</td>
<td>24</td>
<td>8</td>
<td>0</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>MD5</td>
<td>3147</td>
<td>44</td>
<td>9</td>
<td>13</td>
<td>18</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>SSLD</td>
<td>828</td>
<td>51</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>3</td>
</tr>
<tr>
<td>WFQ</td>
<td>905</td>
<td>24</td>
<td>19</td>
<td>17</td>
<td>0</td>
<td>34</td>
<td>5</td>
</tr>
<tr>
<td>MTC</td>
<td>2427</td>
<td>33</td>
<td>27</td>
<td>3</td>
<td>10</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>MPLS-UP</td>
<td>1603</td>
<td>31</td>
<td>14</td>
<td>9</td>
<td>8</td>
<td>34</td>
<td>4</td>
</tr>
<tr>
<td>MPLS-DOWN</td>
<td>1574</td>
<td>32</td>
<td>15</td>
<td>9</td>
<td>8</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>AES</td>
<td>785</td>
<td>14</td>
<td>23</td>
<td>17</td>
<td>22</td>
<td>20</td>
<td>4</td>
</tr>
</tbody>
</table>

**Instruction Mix Table for Network Task Simulation**
Results – Task Latency

Time taken for completion of every task
Results – Task Set Latency

Time taken for completion of every task set (10 Tasks)
Results – Average Power Consumption

Average Power Consumption

RISC-V
26.28 uW/MHz
~13mW @ 500 Mhz
Comparison with ARM Cortex A53

Let’s see how a RISC-V Core compares with A53
A53 Specs

- Processor Bits – 64
- ISA – ARM v8
- Clock Speed – 500 MHz
- Pipeline type – In-Order
- Pipeline Stages - 8
- Cache - 64-KBytes of I-Cache and D-Cache
  - 512 KBytes of L2 Cache

Modeling technique was the same as RISC-V’s
Same task profile was used for simulation
RISC-V has a slightly higher task completion rate.
RISC-V completes 1 Extra Task Set
A53

RISC-V shines when it comes to Power Consumption

Samsung Exynos 5433
38mW @ 500 MHz

RISC-V
26.28uW/MHz → 13mW @ 500 Mhz

RISC-V shines when it comes to Power Consumption
RISC-V consumes 3 TIMES LESSER power than Cortex A53
RISC-V based Solid-State Drive!

Another application of RISC-V: Solid-State Drives
Read/Write Latencies and Average Power

Results obtained from system-level model of a RISC-V based Solid-State Drive

Traffic Rate: 30us

Traffic Rate: 10us
VisualSim Explorer

Try these links to get the feel of VisualSim

Requirements ➔ Just a browser and a Java Runtime Environment

Links

- RISC-V Processor System
  [Link](http://www.mirabilisdesign.com/launchdemo/demo/HAL/RISC_V/RISCV_InOrder/)

- ARM Cortex A53 Processor System
  [Link](http://www.mirabilisdesign.com/launchdemo/demo/HAL/A_Cortex/ARM_Cortex_A53/)

- RISC-V based Solid-State Drive
  [Link](http://www.mirabilisdesign.com/launchdemo/demo/system_architecture/SSD/SSD_RISC_V/)
Future Development

- A 64 bit RISC-V System-Level Model
- Machine Learning Applications
- Multi-Core SoC Design
Conclusion

- Successfully simulated RISC-V ISA as a Processor Core
- Compared RISC-V Core with ARM Cortex A53 using a network application
- Showcased a Solid-State Drive Design using the RISC-V Processor Core
Thank You

Get back to us at  www.mirabilisdesign.com

karthi.sugumar@gmail.com  |  tjose@mirabilisdesign.com