EXCALIBUR Hardware and Enabling Software RISC-V Testbed

What is RISC-V?

Instruction Set Architectures (ISAs) are typically proprietary, limiting the number of implementations (e.g. x86) and/or requiring licences and restrictions (e.g. ARM). By contrast, RISC-V is an open ISA developed by the community where anybody is able to take the specification and then provide a CPU implementation of this. Not only does this encourage a collaborative effort in developing a solid and mature ISA, it also results in a large number of RISC-V CPU implementations and a rich software ecosystem.

Testbed project timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022</td>
<td>April: Project starts, website and documentation available</td>
</tr>
<tr>
<td></td>
<td>June: Enabling software development starts</td>
</tr>
<tr>
<td></td>
<td>September: First soft-cores made available to users</td>
</tr>
<tr>
<td></td>
<td>April: Training courses around using the testbed and hardware design</td>
</tr>
<tr>
<td></td>
<td>December: Final results of benchmarking and enabling software</td>
</tr>
<tr>
<td>2023</td>
<td>January: Final soft-core accelerators released into catalogue</td>
</tr>
<tr>
<td></td>
<td>March: Coupling soft-cores with accelerators begins</td>
</tr>
<tr>
<td></td>
<td>March: User case studies published and project completes</td>
</tr>
<tr>
<td></td>
<td>September: First soft-cores made available to users</td>
</tr>
<tr>
<td></td>
<td>June: Early access testbed available</td>
</tr>
<tr>
<td></td>
<td>April: Training courses around using the testbed and hardware design</td>
</tr>
<tr>
<td></td>
<td>December: Final results of benchmarking and enabling software</td>
</tr>
</tbody>
</table>

The testbed will be available for use at least until 2025

Physical RISC-V CPUs

We provide access to physical RISC-V CPUs which represent a variety of different technologies and capabilities. These all run Linux and are provided as compute nodes, where compilation is undertaken on the login node. A variety of common HPC libraries are provided including FFTW, PETSc, MPI etc.

The testbed currently contains the following type of nodes:
- HiFive Unmatched (quad core U740)
- StarFive VisionFive V2 (quad core U74)
- Allwinner D1-H (C906 CPU)
- Lichee RV Dock (C906 CPU)
- MangoPi M1q-Pro (C906 CPU)
- And more types of RISC-V node added as they become available!

These cores contain the 0.7 version of the new vectorisation ISA specification, enabling experimentation with SIMD.

Physical boards enable easy access to RISC-V, however RISC-V is moving very quickly and so can be somewhat behind the cutting-edge state of the art

Access to cutting edge soft core RISC-V designs

Soft cores provide a software description of a CPU which can then be used to program an FPGA. This enables us to provide a catalogue of many different types and configurations of RISC-V CPU at larger core counts

For example, the image on the right illustrates a single-core NeorV32 (the central large block) with other blocks providing infrastructure support. This enables the CPU core to access memory, GPIO, UART and interact with the host machine.

- We provide numerous pre-built soft cores in a catalogue which can be loaded by users. Additional configurations of these can be provided as required
- All soft cores run on a state of the art AlphaData P101 which provides the Versal FPGA.
- We have developed Launchpad, which provides seamless interaction with the soft cores

Enabling software development

An important aspect of the project is to also enhance the software ecosystem for RISC-V. To this end we have been porting libraries and developing new tools

An example: Supporting 1.0 vectorisation on 0.7 hardware:
- Problem: Physical RISC-V cores tend to support version 0.7 of vectorisation, whereas version 1.0 has been released and the only version supported by up-to-date/current/upstream compilers
- Solution: We have developed a tool that manipulates the generated assembly code, to backport executables so that they comply with 0.7 vectorisation standard
  
Enabling software development

An important aspect of the project is to also enhance the software ecosystem for RISC-V. To this end we have been porting libraries and developing new tools

An example: Supporting 1.0 vectorisation on 0.7 hardware:
- Problem: Physical RISC-V cores tend to support version 0.7 of vectorisation, whereas version 1.0 has been released and the only version supported by up-to-date/current/upstream compilers
- Solution: We have developed a tool that manipulates the generated assembly code, to backport executables so that they comply with 0.7 vectorisation standard

Exploratory benchmarking

We are also undertaking benchmarking to understand the relative performance of the RISC-V cores and options. when vectorisation is enabled

Coupling soft cores with accelerators

RISC-V is not just for future HPC CPUs, it also provides an extension interface, enabling coupling with accelerators
- RISC-V core undertakes management and executes non-accelerated portions of code
- The accelerator might or might not be driven by the RISC-V ISA (e.g. V extension)
- This specialisation is a benefit for HPC, coupling CPUs with bespoke accelerators

Funded by ExCALIBUR H&ES

The ExCALIBUR H&ES component aims to provide novel hardware in the form of testbeds. Our testbed enables HPC developers to experiment with RISC-V.

Free access

Users from across the world welcome

riscv.epcc.ed.ac.uk

https://excalibur.ac.uk/

The Hardware and Enabling Software (H&ES) component aims to provide novel hardware in the form of testbeds. Our testbed enables HPC developers to experiment with RISC-V.