An Open-Source Application Core: CVA6 from the OpenHW Group

Jérôme Quévremont
Thales Research & Technology, Palaiseau, France

Spring 2022 RISC-V Week, Paris
Why Thales invests in RISC-V and open-source HW

**Software**
Large ecosystem compatible across implementations

**Security**
A fully auditable processor

**Safety**
No black-box

**SWaP & customization**
Exact fit between features and application needs

**Performance**
State-of-the-art processor

**No vendor-locking**
LTS; business opportunities for support, customization…

**Sovereignty**
Ability to fork if needed

Thales member of OpenHW Group and RISC-V International
CVA6 core

Open-source RISC-V application core

- Supports rich OSes like Linux

One RTL code base, two flavors:

CV64A6
- 64-bit
- ARIANE donated to OpenHW Group by ETH Zürich

CV32A6
- 32-bit
- Compact version designed by Thales
CVA6 in a nutshell

Configurable CPU core:
- 32 or 64 bit RISC-V (CV32A6 / CV64A6)
- L1 cache organization
- SP/DP floating point
- Instruction set extension interface (CV-X-IF)
- MMU
- Memory protection (PMP)
- Hypervisor support (H)
- Safe & secure features

An academic project turning into an industrial-grade CPU core
- 100% coverage verification target
- Permissive Apache/Solderpad license

Software support:
- Linux (32/64 bit)
- Embedded OS (FreeRTOS…)
- RISC-V standard compilers (GCC…)
- Debug: GDB, OpenOCD, Eclipse IDE

Architecture:
6-stage, single-issue, in-order, branch prediction, 2.5 CoreMark/MHz
CVA6 can be assembled into a multi/many-core SMP CPU with OpenPiton
CV32A6: RV32IMA[F][C]_Zicsr_Zifencei M/S/U [Sv32]
CV64A6: RV64IMA[F][D][C]_Zicsr_Zifencei M/S/U/[H] [Sv39]

Same core
- For ASIC targets (32/64 bit)
- As a 32-bit vendor-independent FPGA soft-core

Requirement specification:
CVA6: a highly configurable core

CVA6 RISC-V configuration

- rv64
- A
- F/FD
- C
- MMU sv39
- PMP
- H
- Synchronous
- active low
- no counter
- no cache
- 16kB 4ways
- 4kB 3ways
- no cache
- 16kB 4ways
- 32kB 8ways
- No CV-X-IF
- FPGA
- rv32
- not F
- F not D
- sv32
- sv0
- not H
- no user bit
- WB Cache
- - 16kB 4ways
- - 4kB 3ways
- - no cache
- - 16kB 4ways
- - 32kB 8ways
- - no counter
- - no user bit
- - WB Cache
- - FPG
CVA6: an extendable core

- **CV-X-IF interface to extend the CVA6 instruction set**
  - Current or future RISC-V extensions (B, P…)
  - Custom extensions (cryptography, signal processing…)

- **CV-X-IF specified by OpenHW Group**
  - Open specification, can be used off OpenHW

- **Compiler support**
  - Seamless for supported standard extensions (e.g. B)
  - LLVM should ease the support of custom extensions
  - Inline ASM possible for specific processing

- **Benefits**
  - Add extensions without a full re-validation of the core
  - Reuse coprocessors between CORE-V cores (CVA6, CV32E40X, CVE2 tbc)
FPGA soft-core

- CVA6 initially designed for ASIC targets
- CV32A6 is being optimized to also be an FPGA soft-core
  - Technology-agnostic (Xilinx, Microchip…)
  - Same common RTL code
  - Benefits: ease FPGA technology migration, same architectures in ASIC & FPGA, white box analysis…

FPGA optimizations:
- +50% frequency achieved ★
- -30% resources achieved ★
- More optimizations to come
- Some also improve ASIC PPA and CV64A6
Linux support and toolchain

**MMU**
- I&D TLB, hardware PTW
- Designed Sv32 MMU (CV32A6) to complement Sv39 (CV64A6)

**Linux support**
- Available in 32 & 64 bit
- Currently supported: U-Boot, OpenSBI, BuildRoot
- Yocto to come

**Other OSes**
- FreeRTOS 32 & 64 bit supported
- As an application core, it should support many other OSes

**Compiler: GCC**
- CVA6 features RISC-V standard extensions
- LLVM and custom extension support on the roadmap

**Debug: GDB, OpenOCD, Eclipse IDE**

Have you visited our demo on [OPENHW] booth?
Open-source project

<table>
<thead>
<tr>
<th>Full open-source package</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVA6 core</td>
</tr>
<tr>
<td>Verification: testbench, sequences, ISS…</td>
</tr>
<tr>
<td>Linux support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easier collaboration</td>
</tr>
<tr>
<td>Evaluate CVA6 without paperwork</td>
</tr>
<tr>
<td>Audit and white box analysis (safety, security)</td>
</tr>
<tr>
<td>Apache/Solderpad permissive license eases industrial use</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core: <a href="https://github.com/openhwgroup/cva6">https://github.com/openhwgroup/cva6</a></td>
</tr>
<tr>
<td>Linux: <a href="https://github.com/openhwgroup/cva6-sdk">https://github.com/openhwgroup/cva6-sdk</a></td>
</tr>
<tr>
<td>Verification: <a href="https://github.com/openhwgroup/core-v-verif">https://github.com/openhwgroup/core-v-verif</a></td>
</tr>
</tbody>
</table>
The CVA6 project team @OpenHW

Three Thales teams:
- Thales Research & Technology (TRT), France
  - Technical project leader
- Thales DIS (INVIA), France
  - Verification leader
- Thales India / Engineering Competence Center (ECC)

Academy & Research contributors:
- ETH Zürich
- U. Bologna (past)
- U. Minho (TBC)

More industrial contributors are welcome
Acknowledgement

Some Thales Research & Technology’s CVA6 activities are supported by the FRACTAL project which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Italy, Austria, Germany, Finland and Switzerland.

https://fractal-project.eu/
https://www.linkedin.com/company/fractal-european-research-project/
@project_fractal
Thank you!