# Title of Project - "CORE-V CV-X-IF"
# Project Launch Proposal
## Date of the proposal - 2021-10-18
## Author(s) - Trefor Southwell

## Summary of project

This project is the creation of the specification of the CV-X-IF interface between RISC-V cores and co-processors by the **OpenHW Cores TG**.

The input specification as above is written with 32-bit RISC-V cores in mind and this project will validate the specification for 64-bit cores and make modifications if required.
This project will extend the specification to specify how to support the compressed instruction format.

CV-X-IF is intended to provide a generalized framework in which custom co-processors (accelerators) provide ISA extensions for RISC-V CPU cores. It features independent channels for accelerator-agnostic offloading of instructions and writeback of the result. It supports pseudo dual-issue behaviour, and allows sharing of accelerators across multiple cores.

Test vehicles:

- The primary OpenHW target for OpenHW's implementation is the CV32E40X core, this work will gate the specification signoff.
- There are also plans to review the extension for the CVA6 core, but this implementation will not gate the work.
- There are also activities to use this interface for the RV32F extensions are on-going for the CV32E40P core, but this implementation will not gate the work.

### Components of the Project

#### Component 1 Description

Specification document submitted into GIT Hub and released to v1.0:

    ○ Referenced to a design that implements the SPEC (could be cv32e40x)
    ○ With links to timing Diagrams that illustrate the interface in operation

This specification has been tested with an implementation of the interface (as part of the CV32E40X core project). While the specification aims to support 64-bits its implementation can be ratified in another project.

#### Component 2 Description

Formal description of the CV-X-IF specification [?]

- Formal Specification on GitHub written in a formal language (TBC)

## Summary of market or input requirements
### Known market/project requirements at PL gate

a) Provide a unified accelerator interface.
b) Decouple development of accelerators and CPU cores
c) Re-use extension accelerators with different cores
d) Share expensive accelerator units across multiple cores in a cluster.
e) Where possible minimise the overhead (performance, area and clock frequency) of using the extension interface instead of modifying the core internally.
f) Support for multiple outstanding load/store operations
g) Ability to kill instructions in the accelerator

Nice to have: Additional specification information such as timing diagrams, architectural examples, connections, examples, etc.

### Potential future enhancements for future project phases

## Who would make use of OpenHW output

- OpenHW Group members and other CORE-V developers who want to design CPU cores that support a range of custom accelerators (including the CV32E40X and the CVA6 projects inside OpenHW).
- Designers of RISC-V accelerators who want to be able to use them on multiple different cores.
- Any RISC-V user who wants to find optimised solutions for their application can experiment with a selection of RISC-V cores and accelerators without having to undertake major design work.

## Summary of Timeline

June 2021 - V0.1 moved to GIT Hub (https://github.com/openhwgroup/core-v-xif) and archived in a branch
July 2021 - first Feedback from the implementation of the RV32F extension in CV32E40P using the X Interface v0.1 https://github.com/davideschiavone/cv32e40p/tree/x-interface
August 2021: feedbacks from lowRISC and SiLabs about memory operations
August 2021: Thales provided updates on 64-bit implementation.
September 2021 - Major updates from SiLabs and ReadtheDocs version of the SPEC [from v0.1 to v0.8]

October 2021 - ETH starts updating the CV32E40P.FPU with the newest v0.8 SPEC to report RTL implications and results

October 2021 -> June 2022

- Implementation of CV32E40X core.
- Development of formal specification of the X-Interface.
- Development of compiler for custom accelerators.

June 2022

- Release v1.0 - text specification and formal specification.

## Explanation of why OpenHW should do this project

Open Hardware is member-driven, multiple members including SiLabs, Yosys, Embecosm, Thales and Imagination Technologies have expressed an interest in this project and are willing to provide resources to make it successful.

The X-Interface promotes the sharing of RISC-V accelerators across multiple cores, thus improving the overall ecosystem.

The X-Interface specification is not a RISC-V ISA specification but a hardware specification and hence is more suitable for OpenHW than RISC-V International. Any RISC-V processor hardware implementation could utilise this interface.

## Industry landscape: description of competing, alternative, or related efforts in the industry

Many processor extension interfaces exist for specific cores, however this interface aims to be core independent allowing the sharing of accelerators between different RISC-V CPU cores.

## OpenHW Members/Participants committed to participate

## Project Leader(s)
### Technical Project Leader(s)
### Project Manager, if a PM is designated

Trefor Southwell, Imagination Technologies

## Project Documents
### Project Planning Documents
### Project Output Documents

## List of project technical outputs

As above components 1 & 2

### Feature Requirements
Features are more granular than Components.
For SW porting projects, this list serves as the detailed project reference for features
For IP Cores or more complext projects, a user manual with requirements specification is produced at the PA gate, which may supercede this list of features

#### Feature 1
#### Feature 2

## External dependencies
These are external factors on which the project depends, such as external standards ratification, external technology input, etc.

## OpenHW TGs Involved

Cores HW

## Resource Requirements
This is a list of major resources/people required to implement the project and indication of whether the resources are available

### Engineering resource supplied by members - requirement and availability

ETH - RV32F implementation in CV32E40P and subsequent feedback on the specification.
Sillabs - Implementation of the interface into CV32E40X and feedback into the specification based on this work.
Thales - Feedback on the interface when used with CVA6 (this will not gate the project).
Yosys - Formal interface specification.
Embecosm - Compiler support for custom extensions.

### OpenHW engineering staff resource plan: requirement and availability
### Marketing resource - requirement and availability
### Funding for project aspects - requirement and availability

## Architecture and/or context diagrams
Architecture (internal blocks and interconnections), and context (depiction of the the project content within its operational context), are both encouraged where appropriate to depict functionality to both subject matter experts and to non-experts

## Project license model

## Description of initial code contribution, if required

## Repository Requirements

## Project distribution model

## Preliminary Project plan
A full project plan is not required at PL. A preliminary plan, which can be for instance the schedule for completion of component or feature list, together with responsible resource, should be provided. Full details should be provided at PA gate.

## Risk Register
A list of known risks, for example external dependencies, and any mitigation strategy