Management of Projects Within OpenHW Group

***draft Framework for Review by TWG***

*Change History*

*1 – Duncan Bees, June 30 - initial analysis of current situation*

* 1. *Duncan Bees, July 1 - added clarifications on concept of vertical projects*
  2. *Duncan Bees, July 7 - clarifications on project descriptions, committers added*

*2.0 Duncan Bees, July 14 - rewritten as a draft project management framework*

*3.0 Duncan Bees July 21 - modify tech milestones*

Contents

[Contents 2](#_Toc46236885)

[1 Scope and Introduction 3](#_Toc46236886)

[1.1 Scope of this Document 3](#_Toc46236887)

[1.2 Definition of OpenHW Project 3](#_Toc46236888)

[1.2.1 OpenHW Technical Projects (OTP) 3](#_Toc46236889)

[1.2.2 OpenHW Specification Projects (OSP). 3](#_Toc46236890)

[1.3 OTP Project Gates and Milestones 3](#_Toc46236891)

[1.3.1 Project Gates for OTP 4](#_Toc46236892)

[1.3.2 Technical Milestones Examples 7](#_Toc46236893)

[1.4 Change Controlled-Artifacts 10](#_Toc46236894)

[1.5 Open HW Working Groups role within OpenHW Projects 10](#_Toc46236895)

[1.5.1 Working Groups and Their Roles 10](#_Toc46236896)

[1.6 Project Repository 12](#_Toc46236897)

[1.7 Project Visibility on OpenHW Group Website 12](#_Toc46236898)

[1.7.1 Current OTPs and OSPs 12](#_Toc46236899)

[1.8 Project Gates and Milestone Tracking 13](#_Toc46236900)

[2 OpenHW Projects from the Eclipse Foundation Perspective 14](#_Toc46236901)

[2.1 Eclipse Parent Project: OpenHW Group 14](#_Toc46236902)

[2.2 Eclipse Sub-Project: CORE-V Cores 14](#_Toc46236903)

[2.2.1 Scope 14](#_Toc46236904)

[2.2.2 Committers 15](#_Toc46236905)

[3 Project Management Tools 16](#_Toc46236906)

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

# Scope and Introduction

## Scope of this Document

This document describes the organization of the technical activities of the OpenHW Group members and staff which lead to published output such as open source hardware, open source software, and specifications.

This document does not address internal activities such as board, marketing, or staff activities.

## Definition of OpenHW Project

The OpenHW projects covered by this document are in the following categories:

* OpenHW Technical Projects (OTP)
* OpenHW Specification Projects (OSP).

### OpenHW Technical Projects (OTP)

During OpenHW Technical Projects (OTP), members develop a technical output. The output is made available to the public according to an open source license. The open source development process used by OpenHW is the Eclipse Development Process (EDP). <https://www.eclipse.org/projects/dev_process/>

A typical OTP comprises one or more of the following deliverables in source and/or object format:

* Integrated circuit design IP (DIP). An example of a DIP OTP is the CV32E40P CORE-V core.
* Integrated circuit verification IP (VIP). An example of a VIP OTP is the FORCE\_RISCV instruction stream generator.
* Software
* FPGA IP
* Board level hardware description

### OpenHW Specification Projects (OSP).

During OpenHW Specification Projects (OSP), members develop a technical standard. The process followed is the Eclipse Foundation Specification Process.

<https://www.eclipse.org/projects/efsp/>

No OSP are currently identified.

Further investigation is required for OpenHW OSP project methodology.

## OTP Project Gates and Milestones

OTP checkpoints are tracked to manage and communicate overall project launch progress. There are OTP Gates and OTP Technical Milestones. Gates are high level project control points managed by the Technical Working Group (TWG). Technical Milestones are determined in the project plan and tailored to the project being undertaken. They may be managed at the TWG or Task Group (TG) level.

### Project Gates for OTP

Project gates for OpenHW Technical Projects are used by the TWG to track projects throughout the proposal development, engineering development, and project closure phases. The flow of a project from initial project proposal to project complete is show in Figure 1

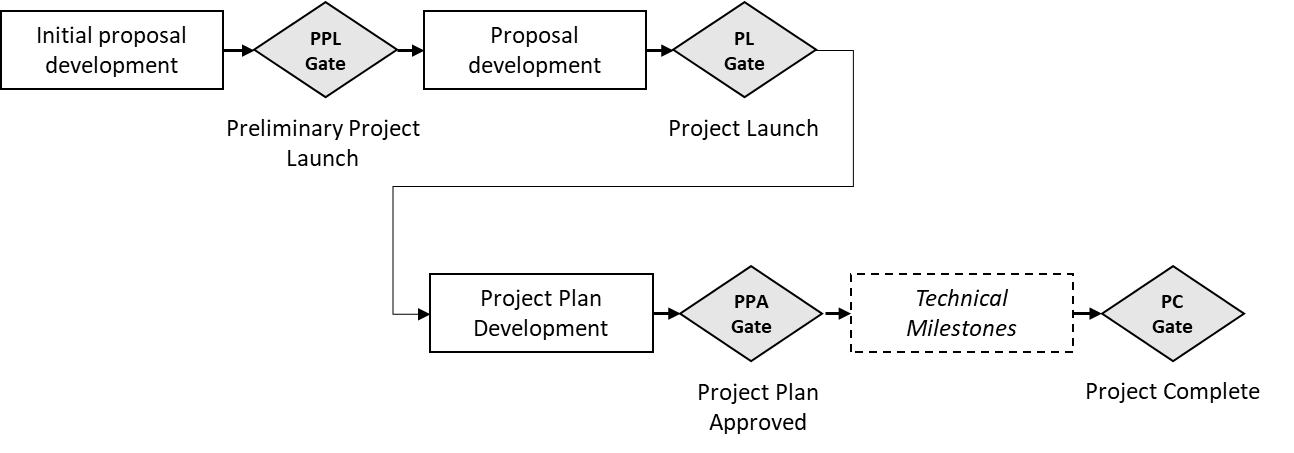


Figure 1 OTP Project Gates

The project gates are summarized in Table 1 below.

|  |  |  |
| --- | --- | --- |
| **Gate** | **Gate Criteria** | **Activities Leading up to Gate** |
| Preliminary Project Launch (PPL) | * Preliminary project proposal meets checklist * TWG majority vote to move forward with project proposal | * OTP pre-proposal development |
| Project Launch (PL) | * PPL gate is already attained. * TWG can hold PPL and PL gate review simultaneously if initial proposal meets both checklists. * PL checklist meets requirements. * TWG majority vote to proceed with project | * OTP Proposal development |
| Project Plan Approved (PPA) | * Project plan including schedule and technical milestones has been accepted by TWG majority vote | * Project plan development |
| Technical milestones as appropriate for the project (see Section 1.3.2) | * Technical milestone and criteria are documented in project plan. * Technical milestone review held in TG | * Technical work |
| Project Complete | * Project deliverables and documentation as set out in project plan are complete * TWG majority vote to complete the project | * Completion of project work |

Table 1 OTP Project Gates

Detailed gate descriptions are provided in the following sections.

#### Project Launch Gates (PPL and PL)

For a project to be launched, the TWG must review and accept project information prepared by the project proponents. These gates are intended to ensure that project selection criteria as determined by the OpenHW community are met prior to project launch. For example, the criteria include number of project participants, a description of the industry landscape, and identification of reasons why the project should be undertaken.

The Preliminary Project Launch (PPL) Gate is intended to provide a assessment by the TWG that OpenHW is potentially interested in the project proposal subject to further information being provided, and also to identify gaps in the proposal which must be filled before launch.

The Project Launch (PL) Gate means that the TWG has voted to accept the project proposal and launch the project.

If all the criteria for PPL and PL are met, the TWG can decide to grant both milestones simultaneously.

|  |  |  |
| --- | --- | --- |
| **Criterion: information presented** | **Preliminary Project Launch (PPL)** | **Project Launch (PL)** |
| Summary of project | preliminary | detailed |
| OpenHW Members committed to participate in project | at least 1 | at least 3 |
| Project Leader | proposed | confirmed |
| Summary of requirements (e.g. from assessment of market requirements) | preliminary | detailed |
| Explanation of why OpenHW should do this project | preliminary | detailed |
| List of project outputs | preliminary | detailed |
| TGs Impacted/Resource requirements | preliminary | detailed |
| OpenHW engineering staff resource- requirement and availability | preliminary | detailed |
| Engineering resource supplied by members - requirement and availability | preliminary | detailed |
| OpenHW marketing resource - requirement and availability | preliminary | detailed |
| Marketing resource supplied by members - requirement and availability | preliminary | detailed |
| Funding supplied by OpenHW - requirement and availability | preliminary | detailed |
| Funding supplied by members - requirement and availability | preliminary | detailed |
| Industry landscape: description of competing, alternative, or related efforts in the industry | preliminary | detailed |
| Description of initial code contribution if any | preliminary | detailed |
| External dependencies | preliminary | detailed |
| Architecture diagram | preliminary | detailed |
| Who would make use of OpenHW output | preliminary | detailed |
| Project license model | preliminary | detailed |
| Project distribution model | preliminary | detailed |
| Project plan | preliminary | preliminary  (note, a completed and detailed project plan is not necessarily required for PL) |
| **TWG decision on gate by:** | majority vote | majority vote |

Table 2 Project Approval Gate Details

#### Project Plan Approved Gate (PPA)

The Project Plan Approved (PPA) Gate is a checkpoint indicating that a full project plan is available and has been approved by the TWG.

The following table shows the typical elements in the project plan. Note that by approval from the TWG, a Project Lead may modify this list so that an appropriate set of elements is determined for a particular project.

|  |  |
| --- | --- |
| **Project Plan Element** | **Description** |
| Project Leader | The person in charge of preparing the project plan and leading the work through its milestones |
| Scope, Project Outputs | Overall project scope and the set of project deliverables |
| Work Breakdown | A set of tasks to achieve the project outputs |
| Schedule Baseline | A first-cut view of the project schedule. |
| Design methodology (waterfall, agile, other). | How the work will be organized and planned, for example use of Sprints. |
| Phases | If there are different releases or phases of the project, they will be described here. |
| Resource Breakdown including key personnel | Resources and their availability, including from OpenHW staff, member companies or other sources. |
| Tools required | Requirements for tooling, software, etc. |
| Funding resources | Description of how the project and its resources will be funded. |
| Design milestones to be tracked in project | The list of Technical Milestones |
| Project Artifacts | The documents or technical artifacts produced by the project |
| Project Artifact under change control | Of the project artifacts, which are under change control. |
| Risk Breakdown and management | Identify the project risks and management plan for the risks |
| Project Repository | The project repository structure |
| Eclipse sub-project under which this project falls | Where does the project align with the Eclipse OpenHW sub-project |
| Initial code contribution description | Where will the project initial code contribution come from, and outline of the contents |
| Eclipse Foundation Contribution Questionnaire result (for IP due diligence) | Result of IP due diligence on the initial code contribution |

Table 3 Project Plan Typical Elements

### Technical Milestones Examples

The technical milestones which will be tracked and reported to the OpenHW membership are determined as appropriate for each project. , by the TWG.

#### Design Integrated Circuit IP OpenHW Technical Project (DIP OTP)

A Design Integrated Circuit IP (DIP) OTP comprises the design and verification of an RTL “IP block”. For example, a RISC-V core would be designed or imported, and OpenHW would verify the IP. The end-result of a DIP OTP would be fully verified RTL code. A typical flow of technical milestones is shown in Figure 2

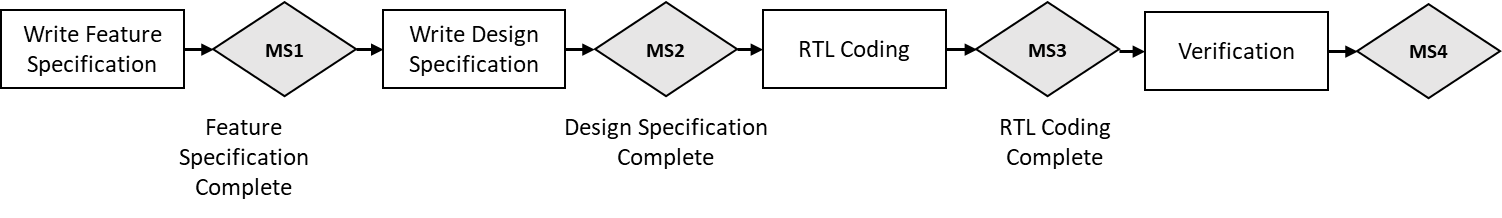


Figure 2 Design Integrated Circuit - Typical Technical Milestones

Typical engineering milestones tracked in OpenHW Integrated Circuit design and verification OTP are shown in Table 4.

|  |  |
| --- | --- |
| **Engineering Milestone** | **Description** |
| Feature Specification Complete | An agreed Feature Specification has been reviewed and agreed by the working group. |
| Design Specification Complete | An agreed Design Specification has been reviewed and agreed by the working group |
| RTL Coding Complete | Final post-verification RTL coding is complete |
| Verification Complete | The Testbench has been reviewed and agreed by the working group, test cases prepared and test cases run |

Table 4 Design IP OTP – Typical Technical Milestones

#### Verification Integrated Circuit IP Project (VIP OTP)

A variety of Verification Integrated Circuit IP (VIP) projects could be envisaged, for example, a project to develop a RISC-V instruction generator.

Although engineering milestones tracked in a VIP OTP would be highly dependent on the nature of the project, some typical milestones are show in Table 5 . The project team sets out the milestones in the project plan which is agreed at the PPA Gate.

|  |  |
| --- | --- |
| **Engineering Milestone** | **Description** |
| Verification Feature Specification Complete | An agreed Feature Specification has been reviewed and agreed by the working group. |
| Verification Design Specification Complete | An agreed Design Specification has been reviewed and agreed by the working group |
| Verification Design Complete | The Verification tool or capability is completed |

Table 5 Verification IP project - typical milestones

#### ***Software Project (SW OTP)***

Software projects may be run according to either waterfall or agile design methodology

**SW Waterfall Approach**

|  |  |
| --- | --- |
| Engineering Milestone | Description |
| Feature Specification Complete | An agreed Feature Specification has been reviewed and agreed by the working group. |
| Design Specification Complete | An agreed Design Specification has been reviewed and agreed by the working group |
| Initial Coding Complete | Initial Coding is complete |
| Testing Complete | Final testing is complete |
| Distribution Complete | Packaging for distribution is complete |

Table 6 - SW OTP - Waterfall Typical Milestones

**SW Agile Approach**

OpenHW Software projects which are run in an agile methodology will need a model which allows initial agreement on a feature specification, then iterative milestones allowing for incremental software releases.

Further investigation is needed to develop the technical milestone tracking strategy for agile projects.

#### FPGA Project

Further investigation is needed to develop the technical milestone tracking strategy for FPGA OTP.

#### Board Level Hardware Project

Further investigation is needed to develop the technical milestone tracking strategy for Board Level Hardware OTP.

## Change Controlled-Artifacts

The OTP Project Plan includes a list of project artifacts and determines which of those artifacts are under change control. For example, an OTP project’s Feature Description would normally be under change control.

The project plan also specifies related milestones to those change controlled artifacts, such as Feature Description Approved.

A process for modifying a previously approved, change-controlled artifact is required. From a preliminary point of view this process will include:

* The change request can be made by any OpenHW member but would normally be made by an active participant in the OTP.
* A notification to the relevant TG and to the TWG of the request to modify the change-controlled artifact
* A justification for the requested change
* An assessment of schedule and resource impact
* A recommendation by the project lead
* An approval or denial by the TWG for the change request

## Open HW Working Groups role within OpenHW Projects

OpenHW has a number of technical working groups and task groups organized by domain. They are shown at <https://www.openhwgroup.org/working-groups/>

### Working Groups and Their Roles

Working groups undertake the technical work of OTPs and OSPs.

The Technical Working Group (TWG) plays an overall supervisory role, including approving new project launches and monitoring progress of technical work carried out by Task Groups (TG).

A particular OTP or OSP may pull in effort from one or more task groups.

The roles of the working groups are summarized in the following table:

|  |  |
| --- | --- |
| **OpenHW Working Group** | **Description** |
| Technical Working Group (TWG) | * Main technical body of the OpenHW Group * Considers and may approve new project proposals. * Appoints Project Lead for OTPs and OSPs * Reviews project plans for OTPs and OSPs * Monitors project gates and progress * Creates or closes Task Groups (TG) as required |
| Cores Task Group | * Coordinates engineering design and RTL coding effort for OTPs requiring cores related design * Monitors technical milestones of cores design aspects of OTPs, coordinates with other groups, and reports same into the TWG * Investigates and decides on engineering tools and methodology related to cores design * Participates in OSPs as required from the cores point of view |
| Verification Task Group | * Coordinates engineering verification efforts for OTPs requiring verification * Monitors technical milestones of verification aspects of OTPs, coordinates with other groups, and reports same into the TWG * Investigates and decides on engineering tools and methodology related to verification * Participates in OSPs as required from the verification point of view |
| SW Task Group | * Coordinates software engineering efforts for OTPs requiring software effort * Monitors technical milestones of software aspects of OTPs, coordinates with other groups, and reports same into the TWG * Investigates and decides on software engineering tools and methodology * Drives industry outreach on software tools and issues, in conjunction with TWG and staff |
| HW Task Group | * Coordinates software engineering efforts for OTPs requiring FPGA and/or board level design * Monitors technical milestones of hardware aspects of OTPs, coordinates with other groups, and reports same into the TWG * Investigates and decides on hardware engineering tools and methodology |

#### Matrix Structure of Working Groups and Projects

As mentioned above, OpenHW projects are “vertical” activities which may require effort from one or more working group. The table below shows an example of three projects. The TWG is the central point for project launch and gate review. The individual TGs are responsible to carry out the work and to report technical milestone progress into the TWG.

|  |  |  |  |
| --- | --- | --- | --- |
| Working Group | OpenHW project 1 | OpenHW project 2 | OpenHW project 3 |
| TWG | **** | **** | **** |
| Cores | **** |  |  |
| Verification | **** |  | **** |
| SW |  | **** |  |
| HW |  | **** |  |

#### Working Group Chairs

The Working Group chairs, vice-chairs, or co-chairs play a key role in leading and monitoring OpenHW projects (OTP and OSP).

The working group chairs are also Committers on the Eclipse Sub-Project (Core-V-Cores).

It is expected that initial Committers for any new OpenHW Eclipse Sub-Project will also be drawn from the working group chairs.

Note that OTP or OSP leaders may, or may not, be working group chairs.

## Project Repository

*A description of OTP project repositories and their relationship to Eclipse sub-project repositories will go here.*

## Project Visibility on OpenHW Group Website

The current list of OpenHW OTPs and OSPs is available on the OpenHW Group Website at <https://www.openhwgroup.org/projects/>

Note that the current content on this page will be updated with the specific list of OTPs and OSPs,

### Current OTPs and OSPs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OpenHW Technical Projects (OTPs) | CV32E40P | CVA64(?) CVA32(?) | GCC SW tools | FORCE-RISCV |
| Description | DIP for a fully verified RTL code for 32-bit embedded processor | DIP for a fully verified RTL code for 32/64-bit application processor | An agreed set of GCC tools (needs full description) | VIP for a RISC-V compliant instruction stream generator |
| Type of Project | OTP | OTP | OTP or OSP, to be decided | OTP |
| TG involved | Cores, Verification, SW | Cores, Verification | SW | VTG |
| Project Leader(s) | Mike Thompson, Davide Schiavone | TBD | Jeremy Bennett | TBD |
| Project Gate passed | **PL** | none | none | none |
| Technical Milestones passed | **list to be created from existing project plans and status updated** |  | none | none |
| Eclipse Sub-Project | CORE-V Cores | CORE-V Cores | potentially CORE-V SW | CORE-V VERIF |

## Project Gates and Milestone Tracking

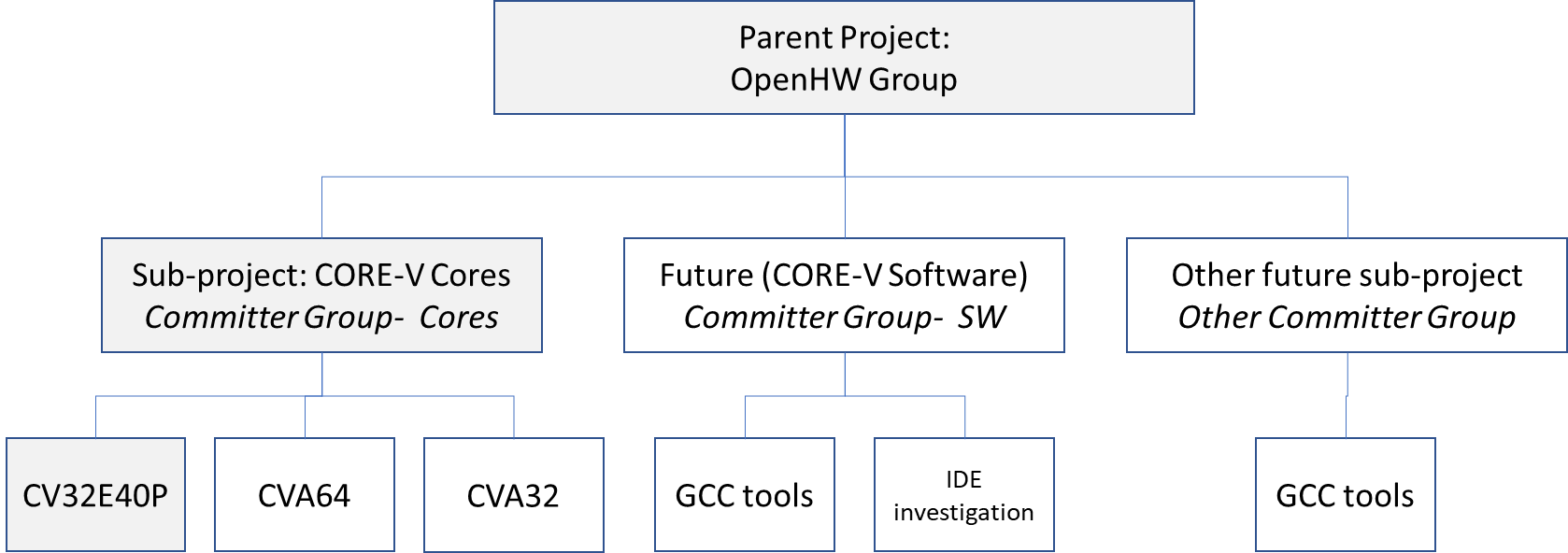
The status of all OpenHW project gates and milestones is available for OpenHW Group members at TBD.

It is to be decided which gate and milestone information is made public.

# OpenHW Projects from the Eclipse Foundation Perspective

OpenHW collaborates with the Eclipse Foundation and makes use of the Eclipse Development Process (EDP) for open source projects.

The Eclipse Foundation maintains a hierarchical project structure corresponding to the OpenHW Group projects. The following Figure shows the current structure (active Eclipse sub-projects and active OpenHW projects are shown in grey shade). In future, other sub-projects may be instantiated.



**Figure 1.** Organization of OpenHW as Eclipse Foundation Projects

Note that each Eclipse sub-project has an associated set of Committers.

*Committer Election is*

The two Eclipse existing projects are described below:

## Eclipse Parent Project: OpenHW Group

The Eclipse Foundation’s parent-level project for the OpenHW Group is described at <https://projects.eclipse.org/projects/>. The parent level project is an organizational notion and doesn’t itself comprise deliverables.

## Eclipse Sub-Project: CORE-V Cores

The Eclipse Foundation’s OpenHW Group CORE-V Cores sub-project is described at <https://projects.eclipse.org/projects/openhw.corev>

### Scope

The scope of this sub-project includes the set of deliverables related to CORE-V cores that OpenHW undertakes as follows:

* Complete documentation: micro-architecture and a user manual.
* Implementation: RTL model and synthesis scripts for both ASIC and FPGA implementations.
* Verification: both dynamic (simulation) and static (formal) verification environments.

Note that this scope comprises several actual OpenHW Technical Projects.

### Committers

The Committers recognized by the Eclipse Foundation for this sub-project are shown at <https://projects.eclipse.org/projects/openhw.corev/who> .The initial committers on the CORE-V Cores sub-project are the OpenHW technical chairs.

# Project Management Tools

TBD - Gantt chart and Kanban/Sprint planning description