First Major Release of fwk

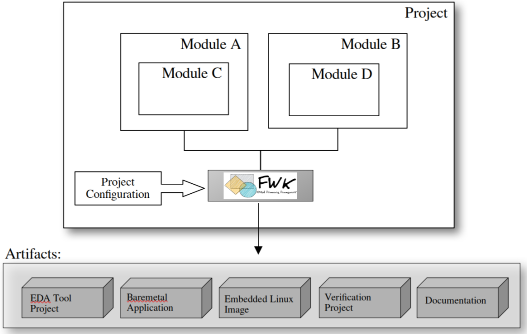
14th MicroTCA Workshop

Çağıl Gümüş

Hamburg, 03.12.2025

What is fwk?

What is fwk?



- It is an **open-source, FPGA Framework.** Written in Tcl-language (>11k lines)
- Automates the creation and build processes of EDA Tools
- Divides the codebase into smaller chunks of IPs **Modules**

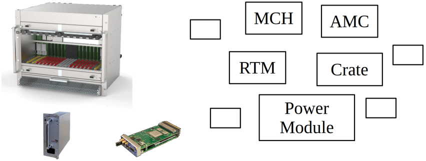
Supported Tools

AMD Vivado
ISE DESIGN SUITE
ModelSim
cocotb
Vitis
PlanAhead 14.5
GHDL
yocto PROJECT
Vivado™ HLS
SDK
Vunit
Libero System-on-Chip
TerosHDL

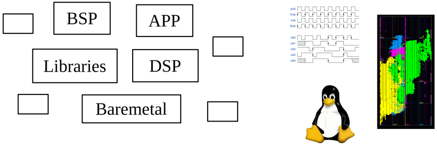
How is fwk related to MicroTCA?

MicroTCA and fwk are abstractions at a different level

µTCA®



FWK
FPGA firmware framework

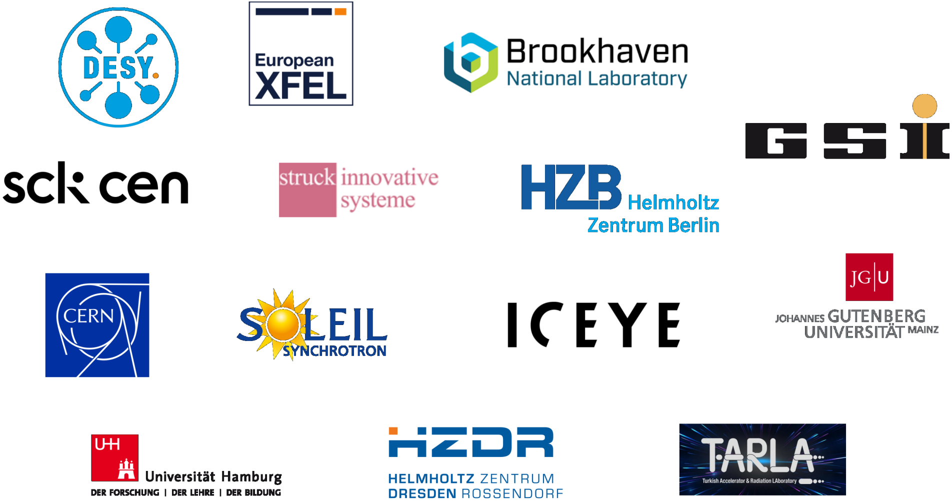


Hardware Abstraction

Firmware Abstraction

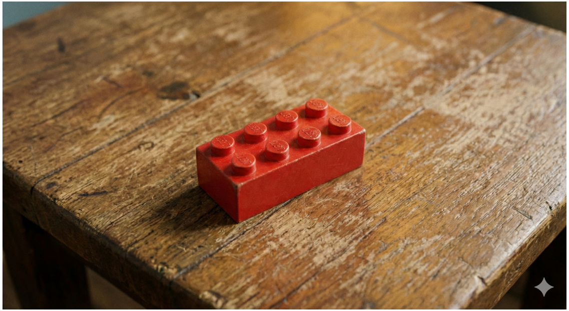
Where is fwk used?

fwk is used in:


DESY.
European XFEL
Brookhaven National Laboratory
GSI
sck cen
struck innovative systeme
HZB Helmholtz Zentrum Berlin
CERN
SOLEIL SYNCHROTRON
ICEYE
JG|U JOHANNES GUTENBERG UNIVERSITÄT MAINZ
UHH Universität Hamburg DER FORSCHUNG | DER LEHRE | DER BILDUNG
HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
TARLA Turkish Accelerator & Radiation LAboratory

How does fwk work?

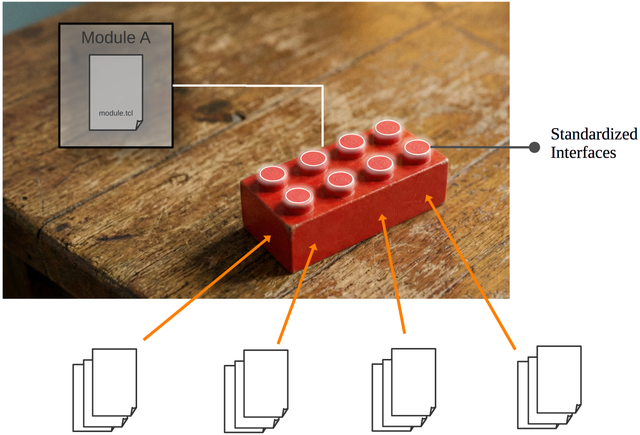
Module



Generated by Gemini 3

Module is a container for an IP. It is the building block of fwk.

Module



Standardized Interfaces

HDL Source Files:
(.vhd, .sv, .v, ...)

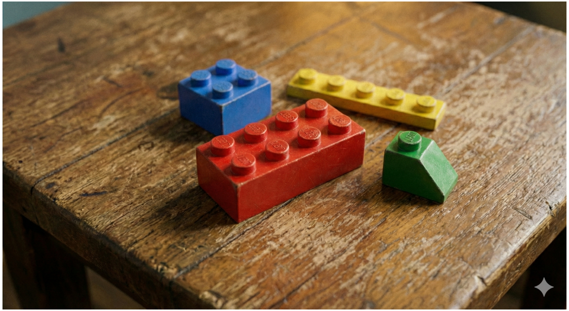
Software
(.py, .c, .c++, ...)

Documentation
(.adoc, .md, ...)

Yocto Recipes
(.bb, .bbappend, ...)

Generated by Gemini 3 edited by a human

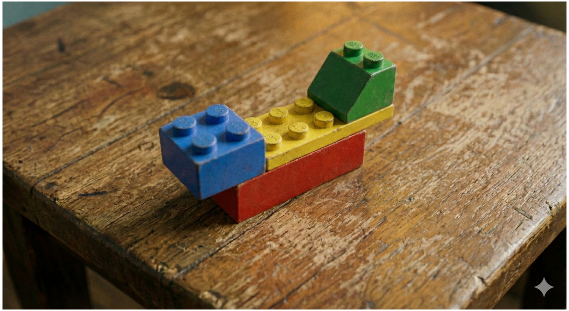
Types of Modules



Generated by Gemini 3

There are many types of modules: BSP, RTM, FMC, APP, etc.

Modules using Modules



Generated by Gemini 3

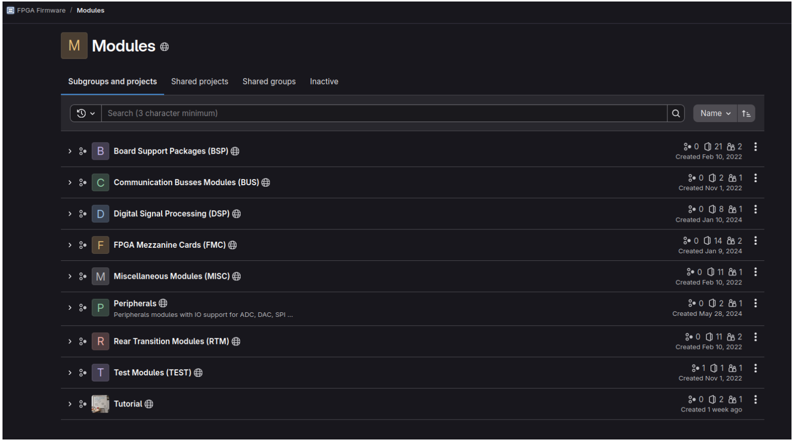
Modules can encapsulate other modules.

Assembled Project

Generated by Gemini 3

Final assembled project.

Open Source Modules and Projects



FPGA Firmware / Modules

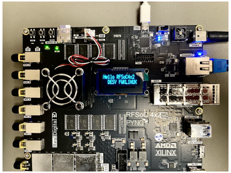
Modules

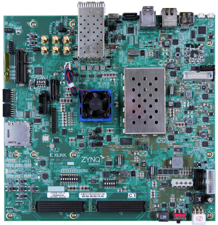
Board Support Packages (BSP)	0	21	2	Created Feb 10, 2022
Communication Busses Modules (BUS)	0	2	1	Created Nov 1, 2022
Digital Signal Processing (DSP)	0	8	1	Created Jan 10, 2024
FPGA Mezzanine Cards (FMC)	0	14	2	Created Jan 9, 2024
Miscellaneous Modules (MISC)	0	11	1	Created Feb 10, 2022
Peripherals Peripherals modules with IO support for ADC, DAC, SPI ...	0	2	1	Created May 28, 2024
Rear Transition Modules (RTM)	0	11	2	Created Feb 10, 2022
Test Modules (TEST)	1	1	1	Created Nov 1, 2022
Tutorial	0	2	1	Created 1 week ago

Visit our Gitlab Page to see all DESY-Modules

fwk on Evaluation Boards

- fwk can support various evaluation boards from AMD. Available:
- ZCU208, ZCU102, RFSoC4x2
- Universidad Politécnica de Madrid has created an fwk example design for Kria260!
- GitHub Link: I2A2 Instrumentation R&D Group





Future

Milestones & Future

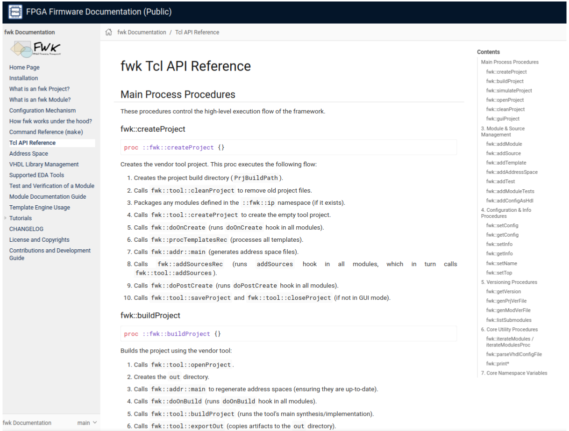
Milestone 1.0.0 Reached:

- Complete overhaul of the Tcl API
- Documentation written from scratch (w/ the help of LLMs)
- Evolved into a community project with >20 contributors.
- External Facilities and Companies started to join the cause!

Outlook:

- MSK Group of DESY will continue maintain and develop for many years to come.
- 2.0.0 Planning has already started...

fwk Documentation and Tutorials



FPGA Firmware Documentation (Public)

fwk Documentation

- Home Page
- Installation
- What is an fwk Project?
- What is an fwk Module?
- Configuration Mechanism
- How fwk works under the hood?
- Command Reference (make)
- Tcl API Reference
- Address Space
- VHDL Library Management
- Supported EDA Tools
- Test and Verification of a Module
- Module Documentation Guide
- Template Engine Usage
- Tutorials
- CHANGELOG
- License and Copyrights
- Contributions and Development Guide

fwk Documentation main

fwk Documentation / Tcl API Reference

fwk Tcl API Reference

Main Process Procedures

These procedures control the high-level execution flow of the framework.

fwk::createProject

```
proc ::fwk::createProject {}
```

Creates the vendor tool project. This proc executes the following flow:

1. Creates the project build directory (`PrjBuildPath`).
2. Calls `fwk::tool::cleanProject` to remove old project files.
3. Packages any modules defined in the `::fwk::ip` namespace (if it exists).
4. Calls `fwk::tool::createProject` to create the empty tool project.
5. Calls `fwk::doOnCreate` (runs `doOnCreate` hook in all modules).
6. Calls `fwk::procTemplatesRec` (processes all templates).
7. Calls `fwk::addr::main` (generates address space files).
8. Calls `fwk::addSourcesRec` (runs `addSources` hook in all modules, which in turn calls `fwk::tool::addSources`).
9. Calls `fwk::doPostCreate` (runs `doPostCreate` hook in all modules).
10. Calls `fwk::tool::saveProject` and `fwk::tool::closeProject` (if not in GUI mode).

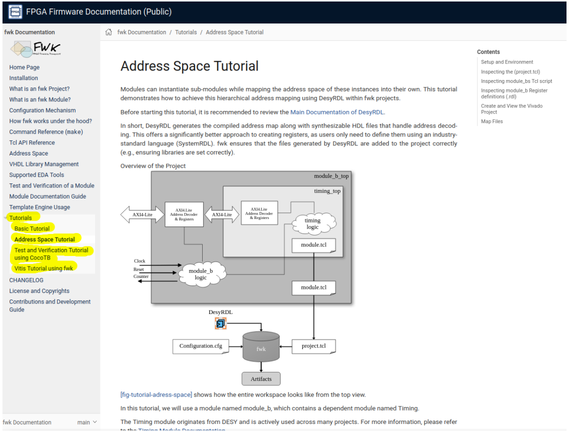
fwk::buildProject

```
proc ::fwk::buildProject {}
```

Builds the project using the vendor tool:

1. Calls `fwk::tool::openProject` .
2. Creates the `out` directory.
3. Calls `fwk::addr::main` to regenerate address spaces (ensuring they are up-to-date).
4. Calls `fwk::doOnBuild` (runs `doOnBuild` hook in all modules).
5. Calls `fwk::tool::buildProject` (runs the tool's main synthesis/implementation).
6. Calls `fwk::tool::exportOut` (copies artifacts to the `out` directory).

Contents



FPGA Firmware Documentation (Public)

fwk Documentation

- Home Page
- Installation
- What is an fwk Project?
- What is an fwk Module?
- Configuration Mechanism
- How fwk works under the hood?
- Command Reference (make)
- Tcl API Reference
- Address Space
- VHDL Library Management
- Supported EDA Tools
- Test and Verification of a Module
- Module Documentation Guide
- Template Engine Usage
- Tutorials
  - Basic Tutorial
  - Address Space Tutorial
  - Test and Verification Tutorial using CocoTB
  - Vitis Tutorial using fwk
- CHANGELOG
- License and Copyrights
- Contributions and Development Guide

fwk Documentation main

fwk Documentation / Tutorials / Address Space Tutorial

Address Space Tutorial

Modules can instantiate sub-modules while mapping the address space of these instances into their own. This tutorial demonstrates how to achieve this hierarchical address mapping using DesyRDL within fwk projects.

Before starting this tutorial, it is recommended to review the Main Documentation of DesyRDL.

In short, DesyRDL generates the compiled address map along with synthesizable HDL files that handle address decoding. This offers a significantly better approach to creating registers, as users only need to define them using an industry-standard language (SystemRDL). fwk ensures that the files generated by DesyRDL are added to the project correctly (e.g., ensuring libraries are set correctly).

Overview of the Project

[fig-tutorial-adress-space] shows how the entire workspace looks like from the top view.

In this tutorial, we will use a module named module_b, which contains a dependent module named Timing.

The Timing module originates from DESY and is actively used across many projects. For more information, please refer to the Timing Module Documentation

Contents

Check out the brand new documentation and tutorials for 1.0 release!

Thank you!

Contact

Deutsches Elektronen-Synchrotron DESY

www.desy.de

Çağıl Gümüş
iD 0000-0003-2176-0212
MSK
cagil.guemues@desy.de
+49–40–8998–3760
`https://cagil-gumus.github.io/`