
The PoC-Library Documentation

Release td

Patrick Lehmann, Thomas B. Preusser, Martin Zabel

May 20, 2018

I	Introduction	1
1	What is PoC?	5
1.1	What is the History of PoC?	6
1.2	Which Tool Chains are supported?	6
1.3	Why should I use PoC?	7
1.4	Who uses PoC?	7
2	Quick Start Guide	9
2.1	Requirements and Dependencies	9
2.2	Download	10
2.3	Configuring PoC on a Local System	10
2.4	Integration	10
2.5	Run a Simulation	12
2.6	Run a Synthesis	13
2.7	Updating	13
3	Get Involved	15
3.1	Report a Bug	15
3.2	Feature Request	15
3.3	Talk to us on Gitter	15
3.4	Contributors License Agreement	16
3.5	Contribute to PoC	16
3.6	Give us Feedback	18
3.7	List of Contributors	18
4	Apache License 2.0	19
4.1	1. Definitions.	19
4.2	2. Grant of Copyright License.	20
4.3	3. Grant of Patent License.	20
4.4	4. Redistribution.	20
4.5	5. Submission of Contributions.	21
4.6	6. Trademarks.	21
4.7	7. Disclaimer of Warranty.	21
4.8	8. Limitation of Liability.	21
4.9	9. Accepting Warranty or Additional Liability.	21
II	Main Documentation	23
5	Using PoC	25
5.1	Requirements	26

5.2	Downloading PoC	28
5.3	Integrating PoC into Projects	31
5.4	Configuring PoC's Infrastructure	32
5.5	Creating my_config/my_project.vhdl	36
5.6	Adding IP Cores to a Project	37
5.7	Simulation	38
5.8	Synthesis	45
5.9	Project Management	49
5.10	Pre-Compiling Vendor Libraries	49
5.11	Miscellaneous	56
6	IP Core Interfaces	57
6.1	Command-Status-Error (PoC.CSE) Interface	57
6.2	PoC.FIFO Interface	57
6.3	PoC.Mem Interface	57
6.4	PoC.Stream Interface	59
7	IP Core Documentations	61
7.1	Common Packages	61
7.2	Simulation Packages	62
7.3	PoC.alt	64
7.4	PoC.arith	64
7.5	PoC.bus	74
7.6	PoC.cache	80
7.7	PoC.comm	91
7.8	PoC.dstruct	92
7.9	PoC.fifo	94
7.10	PoC.io	101
7.11	PoC.mem	118
7.12	PoC.misc	137
7.13	PoC.net	152
7.14	PoC.sort	188
7.15	PoC.xil	196
8	Third Party Libraries	203
8.1	Cocotb	203
8.2	OSVVM	203
8.3	UVVM	204
8.4	VUnit	204
8.5	Updating Linked Git Submodules	204
9	Constraint Files	207
9.1	IP Core Constraint Files	207
9.2	Board Constraint Files	208
10	Tool Chain Specifics	213
10.1	Aldec Active-HDL	213
10.2	Mentor QuestaSim	213
10.3	Xilinx ISE	213
10.4	Xilinx Vivado	213
11	Examples	215
III	References	217
12	Command Reference	219
12.1	PoC Wrapper Scripts	219
12.2	Main Program (PoC.py)	220

12.3	Pre-compile Scripts	220
13	IP Core Database	231
13.1	Overview	231
13.2	Database Structure	232
13.3	Supported Options	234
13.4	Files in detail	235
13.5	User Defined Variables	235
14	Python Infrastructure	237
14.1	PoC.py	237
14.2	Base	241
14.3	Compiler	267
14.4	DataBase	283
14.5	Parser	316
14.6	Simulator	332
14.7	ToolChain	347
14.8	lib	497
15	More References	521
15.1	List of Supported FPGA Devices	521
15.2	List of Supported Boards	522
15.3	Wrapper Script Hook Files	522
15.4	File Formats	523
15.5	Naming Conventions	527
15.6	Known Issues	529
15.7	Local License Copies	530
IV	Appendix	545
16	Change Log	547
17	Index	549
	Python Module Index	551

Part I

Introduction

This library is published and maintained by **Chair for VLSI Design, Diagnostics and Architecture** - Faculty of Computer Science, Technische Universität Dresden, Germany <https://tu-dresden.de/ing/informatik/ti/vlsi>



PoC - “Pile of Cores” provides implementations for often required hardware functions such as Arithmetic Units, Caches, Clock-Domain-Crossing Circuits, FIFOs, RAM wrappers, and I/O Controllers. The hardware modules are typically provided as VHDL or Verilog source code, so it can be easily re-used in a variety of hardware designs.

All hardware modules use a common set of VHDL packages to share new VHDL types, sub-programs and constants. Additionally, a set of simulation helper packages eases the writing of testbenches. Because PoC hosts a huge amount of IP cores, all cores are grouped into sub-namespaces to build a better hierarchy.

Various simulation and synthesis tool chains are supported to interoperate with PoC. To generalize all supported free and commercial vendor tool chains, PoC is shipped with a Python based infrastructure to offer a command line based frontend.

News

See *Change Log* for latest updates.

Cite the PoC-Library

The PoC-Library hosted at [GitHub.com](https://github.com). Please use the following [bibtex](#) entry to cite us:

```
# BibLaTeX example entry
@online{poc,
  title={PoC - Pile of Cores},
  author={Chair of VLSI Design, Diagnostics and Architecture},
  organization={Technische Universität Dresden},
  year={2016},
  url={https://github.com/VLSI-EDA/PoC},
  urldate={2016-10-28},
}
```


CHAPTER 1

What is PoC?

PoC - “Pile of Cores” provides implementations for often required hardware functions such as Arithmetic Units, Caches, Clock-Domain-Crossing Circuits, FIFOs, RAM wrappers, and I/O Controllers. The hardware modules are typically provided as VHDL or Verilog source code, so it can be easily re-used in a variety of hardware designs.

All hardware modules use a common set of VHDL packages to share new VHDL types, sub-programs and constants. Additionally, a set of simulation helper packages eases the writing of testbenches. Because PoC hosts a huge amount of IP cores, all cores are grouped into sub-namespaces to build a better hierarchy.

Various simulation and synthesis tool chains are supported to interoperate with PoC. To generalize all supported free and commercial vendor tool chains, PoC is shipped with a Python based Infrastructure to offer a command line based frontend.

The PoC-Library pursues the following five goals:

- independence in the platform, target, vendor and tool chain
- generic, efficient, resource sparing and fast implementations of IP cores
- optimized for several device architectures, if suitable
- supportive scripts to ease the IP core handling with all supported vendor tools on all listed operating systems
- ship all IP cores with testbenches for local and online verification

In detail the PoC-Library is:

- synthesizable for ASIC and FPGA devices, e.g. from Altera, Lattice, Xilinx, ... ,
- supports a wide range of simulation and synthesis tool chains, and is
- executable on several host platforms: Darwin, Linux or Windows.

This is achieved by using generic HDL descriptions, which work with most synthesis and simulation tools mentioned above. If this is not the case, then PoC uses vendor or tool dependent work-arounds. These work-arounds can be different implementations switched by VHDL *generate* statements as well as different source files containing modified implementations.

One special feature of PoC is it, that the user has not to take care of such implementation switchings. PoC's IP cores decide on their own what's the *best* implementation for the chosen target platform. For this feature, PoC

implements a configuration package, which accepts a well-known development board name or a target device string. For example a FPGA device string is decoded into: vendor, device, generation, family, subtype, speed grade, pin count, etc. Out of these information, the PoC component can for example implement a vendor specific carry-chain description to speed up an algorithm or group computation units to effectively use 6-input LUTs.

1.1 What is the History of PoC?

In the past years, a lot of “IP cores” were developed at the chair of VLSI design¹. This loose set of HDL designs was gathered in an old-fashioned CVS repository and grew over the years to a collection of basic HDL implementations like ALUs, FIFOs, UARTs or RAM controllers. For their final projects (bachelor, master, diploma thesis) students got access to PoC, so they could focus more on their main tasks than wasting time in developing and testing basic IP implementations from scratch. But the library was initially for internal and educational use only.

As a university chair for VLSI design, we have a wide range of different FPGA prototyping boards from various vendors and device families as well as generations. So most of the IP cores were developed for both major FPGA vendor platforms and their specific vendor tool chains. The main focus was to describe hardware in a more flexible and generic way, so that an IP core could be reused on multiple target platforms.

As the number of cores increased, the set of common functions and types increased too. In the end PoC is not only a collection of IP cores, it's also shipped with a set of packages containing utility functions, new types and type conversions, which are used by most of the cores. This makes PoC a *library*, not only a *collection* of IPs.

As we started to search for ways to publish IP cores and maybe the whole PoC-Library, we found several platforms on the Internet, but none was very convincing. Some collective websites contained inactive projects, others were controlled by companies without the possibility to contribute and the majority was a long list of private projects with at most a handful of IP cores. Another disagreement were the used license types for these projects. We decided to use the Apache License, because it has no copyleft rule, a patent clause and allows commercial usage.

We transformed the old CVS repository into three Git repositories: An internal repository for the full set of IP cores (incl. classified code), a public one and a repository for examples, called PoC-Examples, both hosted on GitHub. PoC itself can be integrated into other HDL projects as a library directory or a Git submodule. The preferred usage is the submodule integration, which has the advantage of linked repository versions from hosting Git and the submodule Git. This is already exemplified by our PoC-Examples repository.

1.2 Which Tool Chains are supported?

The PoC-Library and its Python-based infrastructure currently supports the following free and commercial vendor tool chains:

- Synthesis Tool Chains:
 - **Altera Quartus** Tested with Quartus-II ≥ 13.0 . Tested with Quartus Prime ≥ 15.1 .
 - **Intel Quartus** Tested with Quartus Prime ≥ 16.1 .
 - **Lattice Diamond** Tested with Diamond ≥ 3.6 .
 - **Xilinx ISE** Only ISE 14.7 inclusive Core Generator 14.7 is supported.
 - **Xilinx PlanAhead** Only PlanAhead 14.7 is supported.
 - **Xilinx Vivado** Tested with Vivado ≥ 2015.4 . Due to a limited VHDL language support compared to ISE 14.7, some PoC IP cores need special work arounds. See the synthesis documentation section for Vivado for more details.
- Simulation Tool Chains:

¹ The PoC-Library is published and maintained by the **Chair for VLSI Design, Diagnostics and Architecture** - Faculty of Computer Science, Technische Universität Dresden, Germany <http://tu-dresden.de/inf/vlsi-edu>

- **Aldec Active-HDL** Tested with Active-HDL (or Student-Edition) ≥ 10.3 Tested with Active-HDL Lattice Edition ≥ 10.2
- **Cocotb with Mentor QuestaSim backend** Tested with Mentor QuestaSim 10.4d
- **Mentor Graphics ModelSim** Tested with ModelSim PE (or Student Edition) $\geq 10.5c$ Tested with ModelSim SE $\geq 10.5c$ Tested with ModelSim Altera Edition 10.3d (or Starter Edition)
- **Mentor Graphics QuestaSim/ModelSim** Tested with Mentor QuestaSim $\geq 10.4d$
- **Xilinx ISE Simulator** Tested with ISE Simulator (iSim) 14.7. The Python infrastructure supports isim, but PoC's simulation helper packages and testbenches rely on VHDL-2008 features, which are not supported by isim.
- **Xilinx Vivado Simulator** Tested with Vivado Simulator (xsim) ≥ 2016.3 . The Python infrastructure supports xsim, but PoC's simulation helper packages and testbenches rely on VHDL-2008 features, which are not fully supported by xsim, yet.
- **GHDL + GTKWave** Tested with [GHDL](#) $\geq 0.34dev$ and [GTKWave](#) $\geq 3.3.70$ Due to ongoing development and bugfixes, we encourage to use the newest GHDL version.

1.3 Why should I use PoC?

Here is a brief list of advantages:

- We explicitly use the wording *PoC-Library* rather than *collection*, because PoC's packages and IP cores build an ecosystem. Complex IP cores are build on-top of basic IP cores - they are no lose set of cores. The cores offer a clean interface and can be configured by many generic parameters.
- PoC is target independent: It's possible to switch the target device or even the device vendor without switching the IP core.

Todo: Use a well tested set of packages to ease the use of VHDL

Use a well tested set of simulation helpers

Run testbenches in various simulators.

Run synthesis tests in various synthesis tools.

Compare hardware usage for different target platforms.

Supports simulation with vendor primitive libraries, ships with script to pre-compile vendor libraries.

Vendor tools have bugs, check you IP cores when a new tool release is available, before changing code base

1.4 Who uses PoC?

PoC has a related Git repository called [PoC-Examples](#) on GitHub. This repository hosts a list of example and reference implementations of the PoC-Library. Additional to reading an IP cores documention and viewing its characteristic stimulus waveform in a simulation, it can helper to investigate an IP core usage example from that repository.

- [The Q27 Project](#) 27-Queens Puzzle: Massively Parellel Enumeration and Solution Counting
- [Reconfigurable Cloud Computing Framework \(RC2F\)](#) An FPGA computing framework for virtualization and cloud integration.
- [PicoBlaze-Library](#) The PicoBlaze-Library offers several PicoBlaze devices and code routines to extend a common PicoBlaze environment to a little System on a Chip (SoC or SoFPGA).

- [PicoBlaze-Examples](#) A SoFPGA reference implementation, based on the PoC-Library and the PicoBlaze-Library.

Quick Start Guide

This **Quick Start Guide** gives a fast and simple introduction into PoC. All topics can be found in the [Using PoC](#) section with much more details and examples.

Contents of this Page

- [Requirements and Dependencies](#)
- [Download](#)
- [Configuring PoC on a Local System](#)
- [Integration](#)
- [Run a Simulation](#)
- [Run a Synthesis](#)
- [Updating](#)

2.1 Requirements and Dependencies

The PoC-Library comes with some scripts to ease most of the common tasks, like running testbenches or generating IP cores. PoC uses Python 3 as a platform independent scripting environment. All Python scripts are wrapped in Bash or PowerShell scripts, to hide some platform specifics of Darwin, Linux or Windows. See [Requirements](#) for further details.

PoC requires:





- A [supported synthesis tool chain](#), if you want to synthesize IP cores.
- A [supported simulator tool chain](#), if you want to simulate IP cores.
- The **Python 3** programming language and runtime, if you want to use PoC's infrastructure.
- A shell to execute shell scripts:
 - **Bash** on Linux and OS X

– PowerShell on Windows

PoC optionally requires:

- **Git** command line tools or
- **Git User Interface**, if you want to check out the latest ‘master’ or ‘release’ branch.

PoC depends on third part libraries:

- **Cocotb**  A coroutine based cosimulation library for writing VHDL and Verilog testbenches in Python.
- **OSVVM**  Open Source VHDL Verification Methodology.
- **UVVM**  Universal VHDL Verification Methodology.
- **VUnit**  An unit testing framework for VHDL.

All dependencies are available as GitHub repositories and are linked to PoC as Git submodules into the **PoC-Rootlib** directory. See *Third Party Libraries* for more details on these libraries.

2.2 Download

The PoC-Library can be downloaded as a [zip-file](#) (latest ‘master’ branch), cloned with `git clone` or embedded with `git submodule add` from GitHub. GitHub offers HTTPS and SSH as transfer protocols. See the [Download](#) page for further details. The installation directory is referred to as **PoCRoot**.

Protocol	Git Clone Command
HTTPS	<code>git clone --recursive https://github.com/VLSI-EDA/PoC.git PoC</code>
SSH	<code>git clone --recursive ssh://git@github.com:VLSI-EDA/PoC.git PoC</code>

2.3 Configuring PoC on a Local System

To explore PoC’s full potential, it’s required to configure some paths and synthesis or simulation tool chains. The following commands start a guided configuration process. Please follow the instructions on screen. It’s possible to relaunch the process at any time, for example to register new tools or to update tool versions. See [Configuration](#) for more details. Run the following command line instructions to configure PoC on your local system:

```
cd PoCRoot
.\poc.ps1 configure
```

Use the keyboard buttons: to accept, to decline, to skip/pass a step and to accept a default value displayed in brackets.

2.4 Integration

The PoC-Library is meant to be integrated into other HDL projects. Therefore it’s recommended to create a library folder and add the PoC-Library as a Git submodule. After the repository linking is done, some short configuration steps are required to setup paths, tool chains and the target platform. The following command line instructions show a short example on how to integrate PoC.

1. Adding the Library as a Git submodule

The following command line instructions will create the folder `lib\PoC\` and clone the PoC-Library as a Git [submodule](#) into that folder. `ProjectRoot` is the directory of the hosting Git. A detailed list of steps can be found at [Integration](#).

```
cd ProjectRoot
mkdir lib | cd
git submodule add https://github.com:VLSI-EDA/PoC.git PoC
cd PoC
git remote rename origin github
cd ../../
git add .gitmodules lib\PoC
git commit -m "Added new git submodule PoC in 'lib\PoC' (PoC-Library)."
```

2. Configuring PoC

The PoC-Library should be configured to explore its full potential. See [Configuration](#) for more details. The following command lines will start the configuration process:

```
cd ProjectRoot
.\lib\PoC\poc.psl configure
```

3. Creating PoC's `my_config.vhdl` and `my_project.vhdl` Files

The PoC-Library needs two VHDL files for its configuration. These files are used to determine the most suitable implementation depending on the provided target information. Copy the following two template files into your project's source folder. Rename these files to `*.vhdl` and configure the VHDL constants in the files:

```
cd ProjectRoot
cp lib\PoC\src\common\my_config.vhdl.template src\common\my_config.vhdl
cp lib\PoC\src\common\my_project.vhdl.template src\common\my_project.vhdl
```

`my_config.vhdl` defines two global constants, which need to be adjusted:

```
constant MY_BOARD           : string := "CHANGE THIS"; -- e.g. Custom, ML505, ↵
↵ KC705, Atlys
constant MY_DEVICE          : string := "CHANGE THIS"; -- e.g. None, XC5VLX50T-
↵ 1FF1136, EP2SGX90FF1508C3
```

`my_project.vhdl` also defines two global constants, which need to be adjusted:

```
constant MY_PROJECT_DIR     : string := "CHANGE THIS"; -- e.g. d:/vhdl/myproject/,
↵ /home/me/projects/myproject/"
constant MY_OPERATING_SYSTEM : string := "CHANGE THIS"; -- e.g. WINDOWS, LINUX
```

Further informations are provided at [Creating my_config/my_project.vhdl](#).

4. Adding PoC's Common Packages to a Synthesis or Simulation Project

PoC is shipped with a set of common packages, which are used by most of its modules. These packages are stored in the `PoCRoot\src\common` directory. PoC also provides a VHDL context in `common.vhdl`, which can be used to reference all packages at once.

5. Adding PoC's Simulation Packages to a Simulation Project

Simulation projects additionally require PoC's simulation helper packages, which are located in the `PoCRoot\src\sim` directory. Because some VHDL version are incompatible among each other, PoC uses version suffixes like `*.v93.vhdl` or `*.v08.vhdl` in the file name to denote the supported VHDL version of a file.

6. Compiling Shipped IP Cores

Some IP Cores are shipped are pre-configured vendor IP Cores. If such IP cores shall be used in a HDL project, it's recommended to use PoC to create, compile and if needed patch these IP cores. See [Synthesis](#) for more details.

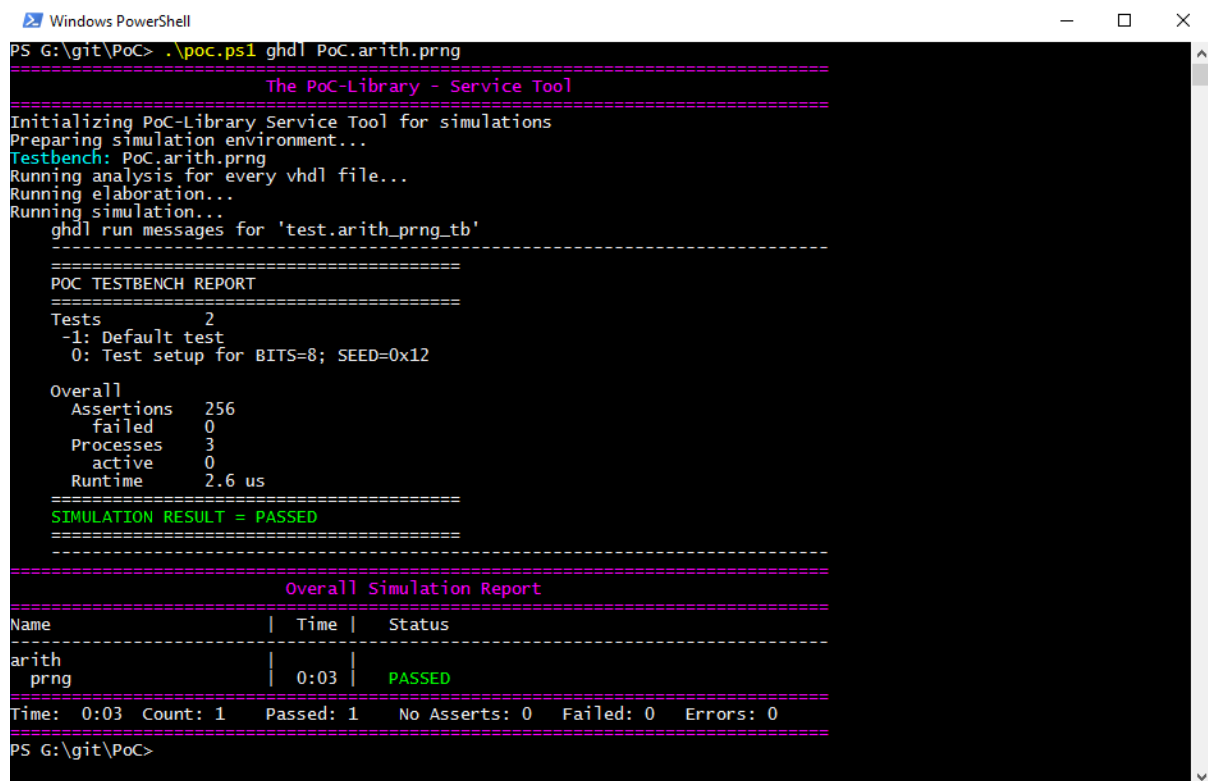
2.5 Run a Simulation

The following quick example uses the GHDL Simulator to analyze, elaborate and simulate a testbench for the module `arith_prng` (Pseudo Random Number Generator - PRNG). The VHDL file `arith_prng.vhdl` is located at `PoCRoot\src\arith` and virtually a member in the *PoC.arith* namespace. So the module can be identified by a unique name: `PoC.arith.prng`, which is passed to the frontend script.

Example:

```
cd PoCRoot
.\poc.ps1 ghdl PoC.arith.prng
```

The CLI command `ghdl` chooses *GHDL Simulator* as the simulator and passes the fully qualified PoC entity name `PoC.arith.prng` as a parameter to the tool. All required source file are gathered and compiled to an executable. Afterwards this executable is launched in CLI mode and its outputs are displayed in console:



```
PS G:\git\PoC> .\poc.ps1 ghdl PoC.arith.prng

=====
The PoC-Library - Service Tool
=====
Initializing PoC-Library Service Tool for simulations
Preparing simulation environment...
Testbench: PoC.arith.prng
Running analysis for every vhd file...
Running elaboration...
Running simulation...
ghdl run messages for 'test.arith_prng_tb'
=====
POC TESTBENCH REPORT
=====
Tests          2
-1: Default test
 0: Test setup for BITS=8; SEED=0x12

Overall
Assertions    256
  failed      0
Processes     3
  active      0
Runtime       2.6 us
=====
SIMULATION RESULT = PASSED
=====

Overall Simulation Report
=====
Name | Time | Status
-----|-----|-----
arith_prng | 0:03 | PASSED
=====
Time: 0:03 Count: 1 Passed: 1 No Asserts: 0 Failed: 0 Errors: 0
=====
PS G:\git\PoC>
```

Each testbench uses PoC's simulation helper packages to count asserts and to track active stimuli and checker processes. After a completed simulation run, an report is written to STDOUT or the simulator's console. Note the

line `SIMULATION RESULT = PASSED`. For each simulated PoC entity, a line in the overall report is created. It lists the runtime per testbench and the simulation status (`... ERROR, FAILED, NO ASSERTS` or `PASSED`). See [Simulation](#) for more details.

2.6 Run a Synthesis

The following quick example uses the Xilinx Systemic Tool (XST) to synthesize a netlist for IP core `arith_prng` (Pseudo Random Number Generator - PRNG). The VHDL file `arith_prng.vhdl` is located at `PoCRoot\src\arith` and virtually a member in the `PoC.arith` namespace. So the module can be identified by an unique name: `PoC.arith.prng`, which is passed to the frontend script.

Example:

```
cd PoCRoot
.\poc.ps1 xst PoC.arith.prng --board=KC705
```

The CLI command `xst` chooses *Xilinx Synthesis Tool* as the synthesizer and passes the fully qualified PoC entity name `PoC.arith.prng` as a parameter to the tool. Additionally, the development board name is required to load the correct `my_config.vhdl` file. All required source file are gathered and synthesized to a netlist.

```
Administrator: posh-git ~ poc [paebells/master]
D:\git\poc [paebells/master] = *5 0 -0 ! *3 ~9 -0 !> .\poc.ps1 xst PoC.arith.prng --board=KC705
Loading Xilinx ISE environment 'C:\Xilinx\14.7\ISE_DS\settings64.bat'
-----
The PoC-Library - Service Tool
-----
Initializing PoC-Library Service Tool for synthesis
IP core: PoC.arith.prng
Preparing synthesis environment...
Executing post-processing tasks...
Running Xilinx Synthesis Tool...
xst messages for 'arith_prng.xst'

* HDL Parsing
*
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/utills.vhdl" Line 1006: Function scale does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 716: Function vendor does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 759: Function device does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 814: Function device family does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 883: Function device number does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 892: Function device subtype does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 1008: Function lut_fanin does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 1035: Function transceiver_type does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/config.vhdl" Line 1121: Function getfsmencoding_gray does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/strings.vhdl" Line 172: Function to_ipstyle does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/strings.vhdl" Line 548: Function to_digit does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/strings.vhdl" Line 632: Function to_natural does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 294: Function to_haud does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 739: Function to_real does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 751: Function to_real does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 762: Function to_real does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 772: Function to_real does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 784: Function to_int does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 795: Function to_int does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 806: Function to_int does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/physical.vhdl" Line 817: Function to_int does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/components.vhdl" Line 117: Function ffdre does not always return a value.
WARNING:HDLCompiler:443 - "D:/git/poc/src/common/components.vhdl" Line 151: Function ffdre does not always return a value.
*
* HDL Elaboration
*
* HDL Synthesis
*
* Advanced HDL Synthesis
*
* Low Level Synthesis
*
* Partition Report
*
* Design Summary
*
Executing post-processing tasks...
Unloading Xilinx ISE environment...
D:\git\poc [paebells/master] = *5 0 -0 ! *3 ~9 -0 !>
```

2.7 Updating

The PoC-Library can be updated by using `git fetch` and `git merge`.

```
cd PoCRoot
# update the local repository
git fetch --prune
# review the commit tree and messages, using the 'treea' alias
git treea
# if all changes are OK, do a fast-forward merge
git merge
```

See also:

Running one or more testbenches The installation can be checked by running one or more of PoC's testbenches.

Running one or more netlist generation flows The installation can also be checked by running one or more of PoC's synthesis flows.

A first step might be to use and explore PoC and its infrastructure in an own project. Moreover, we encourage to read our online help which covers all aspects from quickstart example up to detailed IP core documentation. While using PoC, you might discover issues or missing feature. Please report them as *listed below*. If you have an interesting project, please send us feedback or get listed on our *Who uses PoC?* page.

If you are more familiar with PoC and its components, you might start asking yourself how components internally work. Please read our more advanced topics in the online help, read our inline source code comments or start a discussion on *Gitter* to ask us directly.

Now you should be very familiar with our work and you might be interested in developing own components and contribute them to the main repository. See the *next section* for detailed instructions on the Git fork, commit, push and pull-request flow.

PoC ships some *third-party libraries*. If you are interested in getting your library or components shipped as part of PoC or as a third-party components, please contact us.

3.1 Report a Bug

Please report issues of any kind in our Git provider's issue tracker. This allows us to categorize issues into groups and assign developers to them. You can track the issue's state and see how it's getting solved. All enhancements and feature requests are tracked on GitHub at [GitHub Issues](#).

3.2 Feature Request

Please report missing features of any kind. We are always looking forward to provide a full feature set. Please use our Git provider's issue tracker to report enhancements and feature requests, so you can track the request's status and implementation. All enhancements and feature requests are tracked on GitHub at [GitHub Issues](#).

3.3 Talk to us on Gitter

You can chat with us on [Gitter](#) in our Gitter Room [VLSI-EDA/PoC](#). You can use Gitter for free with your existing GitHub or Twitter account.

3.4 Contributors License Agreement

We require all contributors to sign a Contributor License Agreement (CLA). If you don't know whatfore a CLA is needed and how it prevents legal issues on both sides, read [this short blog post](#). PoC uses the *Apache Contributor License Agreement* to match the *Apache License 2.0*.

So to get started, [sign the Contributor License Agreement \(CLA\)](#) at [CLAHub.com](#). You can authenticate yourself with an existing GitHub account.

3.5 Contribute to PoC

Contributing source code via Git is very easy. We don't provide direct write access to our repositories. Git offers the fork and pull-request philosophy, which means: You clone a repository, provide your changes in your own repository and notify us about outstanding changes via a pull-requests. We will then review your proposed changes and integrate them into our repository.

Steps 1 to 5 are done only once for setting up a forked repository.

3.5.1 1. Fork the PoC Repository

Git repositories can be cloned on a Git provider's server. This procedure is called *forking*. This allows Git providers to track the repository's network, check if repositories are related to each other and notify if pull-requests are available.

Fork our repository `VLSI-EDA/PoC` on GitHub into your or your's Git organisation's account. In the following the forked repository is referenced as `<username>/PoC`.

3.5.2 2. Clone the new Fork

Clone this new fork to your machine. See [Downloading via Git clone](#) for more details on how to clone PoC. If you have already cloned PoC, then you can setup the new fork as an additional *remote*. You should set `VLSI-EDA/PoC` as fetch target and the new fork `<username>/PoC` as push target.

Shell Commands for Cloning:

```
cd GitRoot
git clone --recursive "ssh://git@github.com:<username>/PoC.git" PoC
cd PoC
git remote rename origin github
git remote add upstream "ssh://git@github.com:VLSI-EDA/PoC.git"
git fetch --prune --tags
```

Shell Commands for Editing an existing Clone:

```
cd PoCRoot
git remote rename github upstream
git remote add github "ssh://git@github.com:<username>/PoC.git"
git fetch --prune --tags
```

These commands work for Git submodules too.

3.5.3 3. Checkout a Branch

Checkout the `master` or `release` branch and maybe stash outstanding changes.

```
cd PoCRoot
git checkout release
```

3.5.4 4. Setup PoC for Developers

Run PoC's *configuration routines* and setup the developer tools.

```
cd PoCRoot
.\PoC.ps1 configure git
```

3.5.5 5. Create your own master Branch

Each developer has his own master branch. So create one and check it out.

```
cd PoCRoot
git branch <username>/master
git checkout <username>/master
git push github <username>/master
```

If PoC's branches are moving forward, you can update your own master branch by merging changes into your branch.

3.5.6 6. Create your Feature Branch

Each new feature or bugfix is developed on a feature branch. Examples for branch names:

Branch name	Description
bugfix-utils	Fixes a bug in <code>utils.vhdl</code> .
docs-spelling	Fixes the documentation.
spi-controller	A new SPI controller implementation.

```
cd PoCRoot
git branch <username>/<feature>
git checkout <username>/<feature>
git push github <username>/<feature>
```

3.5.7 7. Commit and Push Changes

Commit your proposed changes onto your feature branch and push all changes to GitHub.

```
cd PoCRoot
# git add ....
git commit -m "Fixed a bug in function bounds() in utils.vhdl."
git push github <username>/<feature>
```

3.5.8 8. Create a Pull-Request

Go to your forked repository and klick on “Compare and Pull-Request” or go to our PoC repository and create a new [pull request](#).

If this is your first Pull-Request, you need to sign our Contributors License Agreement (CLA).

3.5.9 9. Keep your master up-to-date

Todo: undocumented

3.6 Give us Feedback

Please send us feedback about the PoC documentation, our IP cores or your user story on how you use PoC.

3.7 List of Contributors

Contributor ¹	Contact E-Mail
Genßler, Paul	paul.genssler@tu-dresden.de
Köhler, Steffen	steffen.koehler@tu-dresden.de
Lehmann, Patrick ²	patrick.lehmann@tu-dresden.de; paebbels@gmail.com
Preußner, Thomas B. ²	thomas.preusser@tu-dresden.de; thomas.preusser@utexas.edu
Reichel, Peter	peter.reichel@eas.iis.fraunhofer.de; peter@peterreichel.info
Schirok, Jan	janschirok@gmx.net
Voß, Jens	jens.voss@mailbox.tu-dresden.de
Zabel, Martin ²	martin.zabel@tu-dresden.de

Note: This is a local copy of the [Apache License Version 2.0](#).

¹ In alphabetical order.

² Maintainer.

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

4.1 1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **“control”** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or **“Your”**) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **“submitted”** means any form of electronic, verbal,

or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **“Not a Contribution.”**

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

4.2 2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

4.3 3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4.4 4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a **“NOTICE”** text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

4.5 5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

4.6 6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

4.7 7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

4.8 8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

4.9 9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Appendix: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “[]” replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Part II

Main Documentation

PoC can be used in several ways, if all *Requirements* are fulfilled. Chose one of the following integration kinds:

- **Stand-Alone IP Core Library:** Download PoC as archive file (*.zip) from GitHub as latest branch copy or as tagged release file. IP cores can be copied into one or more destination projects or the projects link to the selected IP core source files.

Advantages:

- Simple and fast setup, configuring PoC is optional.
- Needs less disk space than a Git repository.
- After a configuration, PoC's additional features: simulation, synthesis, etc. can be used.

Disadvantages:

- Manual updating via download and file overwrites.
- Updated IP cores need to be copied again into the destination project.
- Using different PoC versions in different projects is not possible.
- No possibility to contribute bugfixes and extensions via Git pull requests.

Next steps: 1. See *Downloads* for how to download a stand-alone version (*.zip-file) of the PoC-Library. 2. See *Configuration* for how to configure PoC on a local system.

- **Stand-Alone IP Core Library cloned from Git:** Download PoC via `git clone` from GitHub as latest branch copy. IP cores can be copied into one or more destination projects or the projects link to the selected IP core source files.

Advantages:

- Simple and fast setup, configuring PoC is optional.
- Access to the newest commits on a branch: New IP cores, new features, bugfixes.
- Fast and simple updates via `git pull`.
- After a configuration, PoC's additional features: simulation, synthesis, etc. can be used.
- Contribute bugfixes and extensions via Git pull requests.

Disadvantages:

- Updated IP cores need to be copied again into the destination project.

- Using different PoC versions in different projects is not possible

Next steps: 1. See [Downloads](#) for how to clone a stand-alone version of the PoC-Library. 2. See [Configuration](#) for how to configure PoC on a local system.

- **Embedded IP Core Library as Git Submodule:** Integrate PoC as a Git submodule into the destination projects Git repository.

Advantages:

- Simple and fast setup, configuring PoC is optional, but recommended.
- Access to the newest commits on a branch: New IP cores, new features, bugfixes.
- Fast and simple updates via `git pull`.
- After a configuration, PoC's additional features: simulation, synthesis, etc. can be used.
- Moreover, some PoC infrastructure features can be used in the hosting repository and project as well.
- Contribute bugfixes and extensions via Git pull requests.
- Version linking between hosting Git and PoC.

Next steps: 1. See [Integration](#) for how to integrate PoC as a Git submodule into an existing Git. 2. See [Configuration](#) for how to configure PoC on a local system.

5.1 Requirements

Contents of this Page

- *Common requirements:*
- *Linux specific requirements:*
 - *Optional Tools on Linux:*
- *Mac OS specific requirements:*
 - *Optional Tools on Mac OS:*
- *Windows specific requirements:*
 - *Optional Tools on Windows:*

The PoC-Library comes with some scripts to ease most of the common tasks, like running testbenches or generating IP cores. We choose to use Python 3 as a platform independent scripting environment. All Python scripts are wrapped in Bash or PowerShell scripts, to hide some platform specifics of Darwin, Linux or Windows.

5.1.1 Common requirements:

Programming Languages and Runtime Environments:

- Python 3 (≥ 3.5):
 - colorama
 - py-flags

All Python requirements are listed in [requirements.txt](#) and can be installed via: `sudo python3.5 -m pip install -r requirements.txt`

Synthesis tool chains:

- Altera Quartus II ≥ 13.0 or

- Altera Quartus Prime ≥ 15.1 or
- Intel Quartus Prime ≥ 16.1 or
- Lattice Diamond ≥ 3.6 or
- Xilinx ISE 14.7¹ or
- Xilinx Vivado ≥ 2016.3 ²

Simulation tool chains

- Aldec Active-HDL (or Student Edition) or
- Aldec Active-HDL Lattice Edition or
- Mentor Graphics ModelSim PE (or Student Edition) or
- Mentor Graphics ModelSim SE or
- Mentor Graphics ModelSim Altera Edition or
- Mentor Graphics QuestaSim or
- Xilinx ISE Simulator 14.7 or
- Xilinx Vivado Simulator ≥ 2016.3 ³ or
- GHDL ≥ 0.34 dev and GTKWave $\geq 3.3.70$

5.1.2 Linux specific requirements:

Debian and Ubuntu specific:

- bash is configured as `/bin/sh` ([read more](#)) `dpkg-reconfigure dash`

Optional Tools on Linux:

Git The command line tools to manage Git repositories. It's possible to extend the shell prompt with Git information.

SmartGit A Git client to handle complex Git flows in a GUI.

Generic Colouriser (grc) ≥ 1.9 Colorizes outputs of foreign scripts and programs. GRC is hosted on [GitHub](#). The latest *.deb installation packages can be downloaded [here](#).

5.1.3 Mac OS specific requirements:

Bash ≥ 4.3 Mac OS is shipped with Bash 3.2. Use Homebrew to install an up-to-date Bash `brew install bash`

coreutils Mac OS' readlink program has a different behavior than the Linux version. The coreutils package installs a GNU readlink clone called greadlink. `brew install coreutils`

¹ Xilinx discontinued ISE since Oct. 2013. The last release was 14.7.

² Due to numerous bugs in the Xilinx Vivado Synthesis (incl. 2016.1), PoC can offer only a restricted Vivado support. See PoC's Vivado branch for a set of workarounds. The list of issues is documented on the [Known Issues](#) page.

³ Due to numerous bugs in the Xilinx Simulator (incl. 2016.1), PoC can offer only a restricted Vivado support. The list of issues is documented on the [Known Issues](#) page.

Optional Tools on Mac OS:

Git The command line tools to manage Git repositories. It's possible to extend the shell prompt with Git information.

SmartGit or SourceTree A Git client to handle complex Git flows in a GUI.

Generic Colouriser (grc) ≥ 1.9 Colorizes outputs of foreign scripts and programs. GRC is hosted on [GitHub](#)

```
brew install Grc
```

5.1.4 Windows specific requirements:

PowerShell

- **Allow local script execution** ([read more](#)) `PS> Set-ExecutionPolicy RemoteSigned`
- **PowerShell ≥ 5.0 (recommended)** PowerShell 5.0 is shipped since Windows 10. It is a part of the [Windows Management Framework 5.0](#) (WMF). Windows 7 and 8/8.1 can be updated to WMF 5.0. The package does not include **PSReadLine**, which is included in the Windows 10 PowerShell environment. Install PSReadLine manually: `PS> Install-Module PSReadline`.
- **PowerShell 4.0** PowerShell is shipped with Windows since Vista. If the required version not already included in Windows, it can be downloaded from Microsoft.com: [WMF 4.0](#)

Optional Tools on Windows:

PowerShell ≥ 4.0

- **PSReadLine** replaces the command line editing experience in PowerShell for versions 3 and up.
- **PowerShell Community Extensions (PSCX) ≥ 3.2** The latest PSCX can be downloaded from [PowerShellGallery](#) `PS> Install-Module Pscx` Note: PSCX $\geq 3.2.1$ is required for PowerShell ≥ 5.0 .

Git (MSys-Git) The command line tools to manage Git repositories.

SmartGit or SourceTree A Git client to handle complex Git flows in a GUI.

posh-git PowerShell integration for Git `PS> Install-Module posh-git`



5.2 Downloading PoC

Contents of this Page

- [Downloading from GitHub](#)
- [Downloading via `git clone`](#)
 - [On Linux](#)
 - [On OS X](#)
 - [On Windows](#)
- [Downloading via `git submodule add`](#)
 - [On Linux](#)
 - [On OS X](#)
 - [On Windows](#)

5.2.1 Downloading from GitHub

The PoC-Library can be downloaded as a zip-file from GitHub. See the following table, to choose your desired git branch.

Branch	Download Link
master	zip-file 
release	zip-file 

5.2.2 Downloading via git clone

The PoC-Library can be downloaded (cloned) with `git clone` from GitHub. GitHub offers the transfer protocols HTTPS and SSH. You should use SSH if you have a GitHub account and have already uploaded an OpenSSH public key to GitHub, otherwise use HTTPS if you have no account or you want to use login credentials.

The created folder `<GitRoot>\PoC` is used as `<PoCRoot>` in later instructions or on other pages in this documentation.

Protocol	GitHub Repository URL
HTTPS	https://github.com/VLSI-EDA/PoC.git
SSH	ssh://git@github.com:VLSI-EDA/PoC.git

On Linux

Command line instructions to clone the PoC-Library onto a Linux machine with HTTPS protocol:

```
cd GitRoot
git clone --recursive "https://github.com/VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
```

Command line instructions to clone the PoC-Library onto a Linux machine machine with SSH protocol:

```
cd GitRoot
git clone --recursive "ssh://git@github.com:VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
```

On OS X

Please see the Linux instructions.

On Windows

Note: All Windows command line instructions are intended for **Windows PowerShell**, if not marked otherwise. So executing the following instructions in Windows Command Prompt (**cmd.exe**) won't function or result in errors! See the [Requirements section](#) on where to download or update PowerShell.

Command line instructions to clone the PoC-Library onto a Windows machine with HTTPS protocol:

```
cd GitRoot
git clone --recursive "https://github.com/VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
```

Command line instructions to clone the PoC-Library onto a Windows machine with SSH protocol:

```
cd GitRoot
git clone --recursive "ssh://git@github.com:VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
```

Note: The option `--recursive` performs a recursive clone operation for all linked `git submodules`. An additional `git submodule init` and `git submodule update` call is not needed anymore.

5.2.3 Downloading via `git submodule add`

The PoC-Library is meant to be integrated into other HDL projects (preferably Git versioned projects). Therefore it's recommended to create a library folder and add the PoC-Library as a `git submodule`.

The following command line instructions will create a library folder `:file:'lib'` and clone PoC as a git submodule into the subfolder `:file:'<ProjectRoot>libPoC'`.

On Linux

Command line instructions to clone the PoC-Library onto a Linux machine with HTTPS protocol:

```
cd ProjectRoot
mkdir lib
git submodule add "https://github.com/VLSI-EDA/PoC.git" lib/PoC
cd lib/PoC
git remote rename origin github
cd ../../
git add .gitmodules lib/PoC
git commit -m "Added new git submodule PoC in 'lib/PoC' (PoC-Library)."
```

Command line instructions to clone the PoC-Library onto a Linux machine machine with SSH protocol:

```
cd ProjectRoot
mkdir lib
git submodule add "ssh://git@github.com:VLSI-EDA/PoC.git" lib/PoC
cd lib/PoC
git remote rename origin github
cd ../../
git add .gitmodules lib/PoC
git commit -m "Added new git submodule PoC in 'lib/PoC' (PoC-Library)."
```

On OS X

Please see the Linux instructions.

On Windows

Note: All Windows command line instructions are intended for **Windows PowerShell**, if not marked otherwise. So executing the following instructions in Windows Command Prompt (**cmd.exe**) won't function or result in errors! See the [Requirements section](#) on where to download or update PowerShell.

Command line instructions to clone the PoC-Library onto a Windows machine with HTTPS protocol:

```
cd <ProjectRoot>
mkdir lib | cd
git submodule add "https://github.com/VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
cd ../../..
git add .gitmodules lib\PoC
git commit -m "Added new git submodule PoC in 'lib\PoC' (PoC-Library)."
```

Command line instructions to clone the PoC-Library onto a Windows machine with SSH protocol:

```
cd <ProjectRoot>
mkdir lib | cd
git submodule add "ssh://git@github.com:VLSI-EDA/PoC.git" PoC
cd PoC
git remote rename origin github
cd ../../..
git add .gitmodules lib\PoC
git commit -m "Added new git submodule PoC in 'lib\PoC' (PoC-Library)."
```

5.3 Integrating PoC into Projects

Contents of this page

- *As a Git submodule*
 - *On Linux*
 - *On OS X*
 - *On Windows*

5.3.1 As a Git submodule

The following command line instructions will integrate PoC into a existing Git repository and register PoC as a Git submodule. Therefore a directory `lib\PoC\` is created and the PoC-Library is cloned as a Git submodule into that directory.

On Linux

```
cd ProjectRoot
mkdir lib
cd lib
git submodule add https://github.com/VLSI-EDA/PoC.git PoC
cd PoC
git remote rename origin github
cd ../../..
git add .gitmodules lib\PoC
git commit -m "Added new git submodule PoC in 'lib/PoC' (PoC-Library)."
```

On OS X

Please see the Linux instructions.

On Windows

Note: All Windows command line instructions are intended for **Windows PowerShell**, if not marked otherwise. So executing the following instructions in Windows Command Prompt (**cmd.exe**) won't function or result in errors! See the *Requirements section* on where to download or update PowerShell.

```
cd ProjectRoot
mkdir lib | cd
git submodule add https://github.com/VLSI-EDA/PoC.git PoC
cd PoC
git remote rename origin github
cd ..\..
git add .gitmodules lib\PoC
git commit -m "Added new git submodule PoC in 'lib\PoC' (PoC-Library)."
```

See also:

Configuring PoC on a Local System

Create PoC's VHDL Configuration Files

5.4 Configuring PoC's Infrastructure

To explore PoC's full potential, it's required to configure some paths and synthesis or simulation tool chains. It's possible to relaunch the process at any time, for example to register new tools or to update tool versions.

Contents of this page

- *Overview*
- *The PoC-Library*
- *Git*
- *Aldec*
 - *Active-HDL*
- *Altera*
 - *Quartus*
 - *ModelSim Altera Edition*
- *Lattice*
 - *Diamond*
 - *Active-HDL Lattice Edition*
- *Mentor Graphics*
 - *QuestaSim*
- *Xilinx*
 - *ISE*

- *Vivado*
- *GHDL*
- *GTKWave*
- *Hook Files*

5.4.1 Overview

The setup process is started by invoking PoC's frontend script with the command `configure`. Please follow the instructions on screen. Use the keyboard buttons: to accept, to decline, to skip/pass a step and to accept a default value displayed in brackets.

Optionally, a vendor or tool chain name can be passed to the configuration process to launch only its configuration routines.

On Linux:

```
cd ProjectRoot
./lib/PoC/poc.sh configure
# with tool chain name
./lib/PoC/poc.sh configure Xilinx.Vivado
```

On OS X

Please see the Linux instructions.

On Windows

Note: All Windows command line instructions are intended for **Windows PowerShell**, if not marked otherwise. So executing the following instructions in Windows Command Prompt (**cmd.exe**) won't function or result in errors! See the [Requirements section](#) on where to download or update PowerShell.

```
cd ProjectRoot
.\lib\PoC\poc.ps1 configure
# with tool chain name
.\lib\PoC\poc.ps1 configure Xilinx.Vivado
```

Introduction screen:

```
PS D:\git\PoC> .\poc.ps1 configure
=====
                        The PoC-Library - Service Tool
=====
Explanation of abbreviations:
  Y - yes          P          - pass (jump to next question)
  N - no          Ctrl + C - abort (no changes are saved)
Upper case or value in '[...]' means default value
-----

Configuring PoC
PoC version: v1.0.1 (found in git)
Installation directory: D:\git\PoC (found in environment variable)
```

5.4.2 The PoC-Library

PoC itself has a fully automated configuration routine. It detects if PoC is under Git control. If so, it extracts the current version number from the latest Git tag. The installation directory is inferred from `$PoCRootDirectory` setup by `PoC.ps1` or `poc.sh`.

```
Configuring PoC
PoC version: v1.0.1 (found in git)
Installation directory: D:\git\PoC (found in environment variable)
```

5.4.3 Git

Note: Setting up Git and Git developer settings, is an advanced feature recommended for all developers interested in providing Git pull requests or patches.

```
Configuring Git
Git installation directory [C:\Program Files\Git]:
Install Git mechanisms for PoC developers? [y/N/p]: y
Install Git filters? [Y/n/p]:
Installing Git filters...
Install Git hooks? [Y/n/p]:
Installing Git hooks...
Setting 'pre-commit' hook for PoC...
```

5.4.4 Aldec

Configure the installation directory for all Aldec tools.

```
Configuring Aldec
Are Aldec products installed on your system? [Y/n/p]: Y
Aldec installation directory [C:\Aldec]:
```

Active-HDL

```
Configuring Aldec Active-HDL
Is Aldec Active-HDL installed on your system? [Y/n/p]: Y
Aldec Active-HDL version [10.3]:
Aldec Active-HDL installation directory [C:\Aldec\Active-HDL]: C:\Aldec\Active-
↪HDL-Student-Edition
```

5.4.5 Altera

Configure the installation directory for all Altera tools.

```
Configuring Altera
Are Altera products installed on your system? [Y/n/p]: Y
Altera installation directory [C:\Altera]:
```

Quartus

```
Configuring Altera Quartus
Is Altera Quartus-II or Quartus Prime installed on your system? [Y/n/p]: Y
Altera Quartus version [15.1]: 16.0
Altera Quartus installation directory [C:\Altera\16.0\quartus]:
```


ModelSim Altera Edition

```
Configuring ModelSim Altera Edition
  Is ModelSim Altera Edition installed on your system? [Y/n/p]: Y
  ModelSim Altera Edition installation directory [C:\Altera\15.0\modelsim_ae]: ↵
↪C:\Altera\16.0\modelsim_ase
```

5.4.6 Lattice

Configure the installation directory for all Lattice Semiconductor tools.

```
Configuring Lattice
  Are Lattice products installed on your system? [Y/n/p]: Y
  Lattice installation directory [D:\Lattice]:
```

Diamond

```
Configuring Lattice Diamond
  Is Lattice Diamond installed on your system? [Y/n/p]: >
  Lattice Diamond version [3.7]:
  Lattice Diamond installation directory [D:\Lattice\Diamond\3.7_x64]:
```

Active-HDL Lattice Edition

```
Configuring Active-HDL Lattice Edition
  Is Aldec Active-HDL installed on your system? [Y/n/p]: Y
  Active-HDL Lattice Edition version [10.2]:
  Active-HDL Lattice Edition installation directory [D:\Lattice\Diamond\3.7_
↪x64\active-hdl]:
```

5.4.7 Mentor Graphics

Configure the installation directory for all mentor Graphics tools.

```
Configuring Mentor
  Are Mentor products installed on your system? [Y/n/p]: Y
  Mentor installation directory [C:\Mentor]:
```

QuestaSim

```
Configuring Mentor QuestaSim
  Is Mentor QuestaSim installed on your system? [Y/n/p]: Y
  Mentor QuestaSim version [10.4d]: 10.4c
  Mentor QuestaSim installation directory [C:\Mentor\QuestaSim\10.4c]: ↵
↪C:\Mentor\QuestaSim64\10.4c
```

5.4.8 Xilinx

Configure the installation directory for all Xilinx tools.

```
Configuring Xilinx
```

```
Are Xilinx products installed on your system? [Y/n/p]: Y
Xilinx installation directory [C:\Xilinx]:
```

ISE

If an Xilinx ISE environment is available and shall be configured in PoC, then answer the following questions:

```
Configuring Xilinx ISE
```

```
Is Xilinx ISE installed on your system? [Y/n/p]: Y
Xilinx ISE installation directory [C:\Xilinx\14.7\ISE_DS]:
```

Vivado

If an Xilinx ISE environment is available and shall be configured in PoC, then answer the following questions:

```
Configuring Xilinx Vivado
```

```
Is Xilinx Vivado installed on your system? [Y/n/p]: Y
Xilinx Vivado version [2016.2]:
Xilinx Vivado installation directory [C:\Xilinx\Vivado\2016.2]:
```

5.4.9 GHDL

```
Configuring GHDL
```

```
Is GHDL installed on your system? [Y/n/p]: Y
GHDL installation directory [C:\Tools\GHDL\0.34dev]:
```

5.4.10 GTKWave

```
Configuring GTKWave
```

```
Is GTKWave installed on your system? [Y/n/p]: Y
GTKWave installation directory [C:\Tools\GTKWave\3.3.71]:
```

5.4.11 Hook Files

PoC's wrapper scripts can be customized through pre- and post-hook file. See *Wrapper Script Hook Files* for more details.

5.5 Creating my_config/my_project.vhdl

The PoC-Library needs two VHDL files for its configuration. These files are used to determine the most suitable implementation depending on the provided platform information. These files are also used to select appropriate work arounds.

5.5.1 Create my_config.vhdl

The **my_config.vhdl** file can easily be created from the template file `my_config.vhdl.template` provided by PoC in `PoCRoot\src\common`. (View source on [GitHub](#).) Copy this file into the project's source directory and rename it to `my_config.vhdl`.

This file should be included in version control systems and shared with other systems. `my_config.vhdl` defines three global constants, which need to be adjusted:

```
constant MY_BOARD      : string := "CHANGE THIS"; -- e.g. Custom, ML505, KC705, Atlys
constant MY_DEVICE     : string := "CHANGE THIS"; -- e.g. None, XC5VLX50T-1FF1136,
↳EP2SGX90FF1508C3
constant MY_VERBOSE    : boolean := FALSE;         -- activate report statements in
↳VHDL subprograms
```

The easiest way is to define a board name and set `MY_DEVICE` to `None`. So the device name is inferred from the board information stored in `PoCRoot\src\common\config.vhdl`. If the requested board is not known to PoC or it's custom made, then set `MY_BOARD` to `Custom` and `MY_DEVICE` to the full FPGA device string.

Example 1: A “Stratix II GX Audio Video Development Kit” board:

```
constant MY_BOARD      : string := "S2GXAV"; -- Stratix II GX Audio Video Development
↳Kit
constant MY_DEVICE     : string := "None";    -- infer from MY_BOARD
```

Example 2: A custom made Spartan-6 LX45 board:

```
constant MY_BOARD      : string := "Custom";
constant MY_DEVICE     : string := "XC6SLX45-3CSG324";
```

5.5.2 Create `my_project.vhdl`

The `my_project.vhdl` file can also be created from a template file `my_project.vhdl.template` provided by PoC in `PoCRoot\src\common`.

The file should to be copied into a projects source directory and renamed into `my_project.vhdl`. This file **must not** be included into version control systems – it's private to a computer. `my_project.vhdl` defines two global constants, which need to be adjusted:

```
constant MY_PROJECT_DIR      : string := "CHANGE THIS"; -- e.g. "d:/vhdl/myproject/"
↳", "/home/me/projects/myproject/"
constant MY_OPERATING_SYSTEM : string := "CHANGE THIS"; -- e.g. "WINDOWS", "LINUX"
```

Example 1: A Windows System:

```
constant MY_PROJECT_DIR      : string := "D:/git/GitHub/PoC/";
constant MY_OPERATING_SYSTEM : string := "WINDOWS";
```

Example 2: A Debian System:

```
constant MY_PROJECT_DIR      : string := "/home/paebbels/git/GitHub/PoC/";
constant MY_OPERATING_SYSTEM : string := "LINUX";
```

See also:

Running one or more testbenches The installation can be checked by running one or more of PoC's testbenches.

Running one or more netlist generation flows The installation can also be checked by running one or more of PoC's synthesis flows.

5.6 Adding IP Cores to a Project

5.6.1 Manually Addind IP Cores

Adding IP Cores to Altera Quartus

Todo: No documentation available.

Adding IP Cores to Lattice Diamond

Todo: No documentation available.

Adding IP Cores to Xilinx ISE

Todo: No documentation available.

Adding IP Cores to Xilinx Vivado

Todo: No documentation available.

5.7 Simulation

Contents of this Page

- *Overview*
- *Quick Example*
- *Vendor Specific Testbenches*
- *Running a Single Testbench*
 - *Aldec Active-HDL*
 - *Cocotb with QuestaSim backend*
 - *GHDL (plus GTKwave)*
 - *Mentor Graphics QuestaSim*
 - *Xilinx ISE Simulator*
 - *Xilinx Vivado Simulator*
- *Running a Group of Testbenches*
- *Continuous Integration (CI)*

5.7.1 Overview

The Python Infrastructure shipped with the PoC-Library can launch manual, half-automated and fully automated testbenches. The testbench can be run in command line or GUI mode. If available, the used simulator is launched with pre-configured waveform files. This can be done by invoking one of PoC's frontend script:

- **poc.sh:** `poc.sh <common options> <simulator> <module> <simulator options>`
Use this frontend script on Darwin, Linux and Unix platforms.

- **poc.psl:** `poc.psl <common options> <simulator> <module> <simulator options>` Use this frontend script Windows platforms.

Attention: All Windows command line instructions are intended for Windows PowerShell, if not marked otherwise. So executing the following instructions in Windows Command Prompt (`cmd.exe`) won't function or result in errors!

See also:

PoC Configuration See the Configuration page on how to configure PoC and your installed simulator tool chains. This is required to invoke the simulators.

Supported Simulators See the Intruction page for a list of supported simulators.

5.7.2 Quick Example

The following quick example uses the GHDL Simulator to analyze, elaborate and simulate a testbench for the module `arith_prng` (Pseudo Random Number Generator - PRNG). The VHDL file `arith_prng.vhdl` is located at `PoCRoot\src\arith` and virtually a member in the `PoC.arith` namespace. So the module can be identified by an unique name: `PoC.arith.prng`, which is passed to the frontend script.

Example 1:

```
cd PoCRoot
.\poc.psl ghdl PoC.arith.prng
```

The CLI command `ghdl` chooses *GHDL Simulator* as the simulator and passes the fully qualified PoC entity name `PoC.arith.prng` as a parameter to the tool. All required source file are gathered and compiled to an executable. Afterwards this executable is launched in CLI mode and it's outputs are displayed in console:

```

PS G:\git\PoC> .\poc.psl ghdl PoC.arith.prng

=====
The PoC-Library - Service Tool
=====
Initializing PoC-Library Service Tool for simulations
Preparing simulation environment...
Testbench: PoC.arith.prng
Running analysis for every vhd file...
Running elaboration...
Running simulation...
ghdl run messages for 'test.arith_prng_tb'
=====
POC TESTBENCH REPORT
=====
Tests      2
-1: Default test
  0: Test setup for BITS=8; SEED=0x12

Overall
Assertions 256
  failed   0
Processes  3
  active   0
Runtime    2.6 us
=====
SIMULATION RESULT = PASSED
=====

Overall Simulation Report
=====
Name      | Time | Status
-----
arith     | 0:03 | PASSED
prng
=====
Time: 0:03 Count: 1 Passed: 1 No Asserts: 0 Failed: 0 Errors: 0
=====
PS G:\git\PoC>

```

Each testbench uses PoC's simulation helper packages to count asserts and to track active stimuli and checker processes. After a completed simulation run, an report is written to STDOUT or the simulator's console. Note the

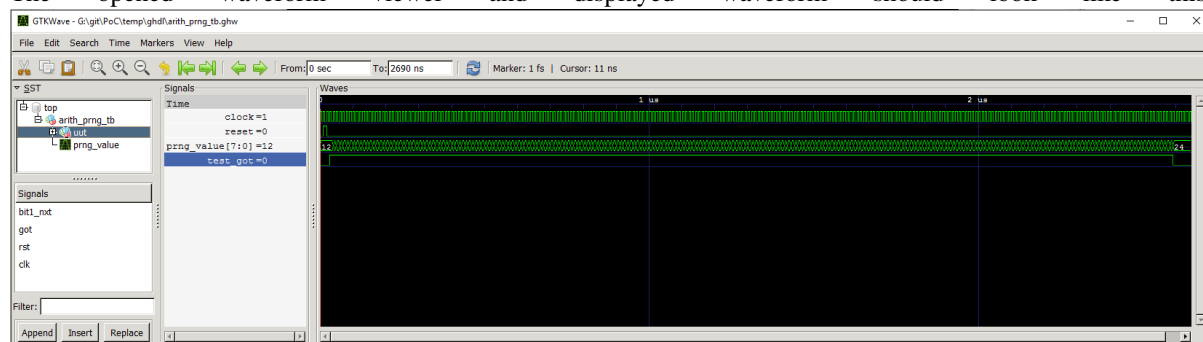
line `SIMULATION RESULT = PASSED`. For each simulated PoC entity, a line in the overall report is created. It lists the runtime per testbench and the simulation status (. . . `ERROR`, `FAILED`, `NO ASSERTS` or `PASSED`).

Example 2:

Passing an additional option `--gui` to the service tool, opens the testbench in GUI-mode. If a waveform configuration file is present (e.g. a `*.gtkw` file for GTKWave), then it is preloaded into the simulator's waveform viewer.

```
cd PoCRoot
.\poc.ps1 ghdl PoC.arith.prng --gui
```

The opened waveform viewer and displayed waveform should look like this:



5.7.3 Vendor Specific Testbenches

PoC is shipped with a set of well known FPGA development boards. This set is extended by a list of generic boards, named after each supported FPGA vendor. These generic boards can be used in simulations to select a representative FPGA of a supported device vendor. If no board or device name is passed to a testbench run, the `GENERIC` board is chosen.

Board Name	Target Board	Target Device
GENERIC	GENERIC	GENERIC
Altera	DE4	Stratix-IV 230
Lattice	ECP5Versa	ECP5-45UM
Xilinx	KC705	Kintex-7 325T

A vendor specific testbench can be launched by passing either `--board=xxx` or `--device=yyy` as an additional parameter to the PoC scripts.

```
# Example 1 - A Lattice board
.\poc.ps1 ghdl PoC.arith.prng --board=Lattice
# Example 2 - A Altera Stratix IV board
.\poc.ps1 ghdl PoC.arith.prng --board=DE4
# Example 3 - A Xilinx Kintex-7 325T device
.\poc.ps1 ghdl PoC.arith.prng --device=XC7K325T-2FFG900
```

Note: Running vendor specific testbenches may require pre-compiled vendor libraries. Some simulators are shipped with diverse pre-compiled libraries, others include scripts or user guides to pre-compile them on the target system.

PoC is shipped with a set of pre-compile scripts to offer a unified interface and common storage for all supported vendor's pre-compile procedures. See [Pre-Compiling Vendor Libraries](#).

5.7.4 Running a Single Testbench

A testbench run is supervised by PoC's `PoCRoot\py\PoC.py` service tool, which offers a consistent interface to all simulators. Unfortunately, every platform has its specialties, so a wrapper script is needed as abstraction from the host's operating system. Depending on the chosen tool chain, the wrapper script will source or invoke the vendor tool's environment scripts to pre-load the needed environment variables, paths or license file settings.

The order of options to the frontend script is as following: `<common options> <simulator> <module> <simulator options>`

The frontend offers several common options:

Common Option		Description
-q	-quiet	Quiet-mode (print nothing)
-v	-verbose	Print more messages
-d	-debug	Debug mode (print everything)
	-dryrun	Run in dry-run mode

One of the following supported simulators can be chosen, if installed and configured in PoC:

Simulator	Description
asim	Active-HDL Simulator
cocotb	Cocotb simulation using QuestaSim Simulator
ghdl	GHDL Simulator
isim	Xilinx ISE Simulator
vsim	QuestaSim Simulator or ModelSim
xsim	Xilinx Vivado Simulator

A testbench run can be interrupted by sending a keyboard interrupt to Python. On most operating systems this is done by pressing `Ctrl + C`. If PoC runs multiple testbenches at once, all finished testbenches are reported with their testbench result. The aborted testbench will be listed as errored.

Aldec Active-HDL

The command to invoke a simulation using Active-HDL is `asim` followed by a list of PoC entities. The following options are supported for Active-HDL:

Simulator Option		Description
	-board=<BOARD>	Specify a target board.
	-device=<DEVICE>	Specify a target device.
	-std=[87/93/02/08]	Select a VHDL standard. Default: 08

Note: GUI mode for Active-HDL is not yet supported.

Example:

```
cd PoCRoot
.\poc.ps1 asim PoC.arith.prng --std=93
```

Cocotb with QuestaSim backend

The command to invoke a Cocotb simulation using QuestaSim is `cocotb` followed by a list of PoC entities. The following options are supported for Cocotb:

Simulator Option		Description
	<code>-board=<BOARD></code>	Specify a target board.
	<code>-device=<DEVICE></code>	Specify a target device.
<code>-g</code>	<code>-gui</code>	Start the simulation in the QuestaSim GUI.

Note: Cocotb is currently only on Linux with QuestaSim supported. We are working to support the Windows platform and the GHDL backend.

Example:

```
cd PoCRoot
.\poc.ps1 cocotb PoC.cache.par
```

GHDL (plus GTKwave)

The command to invoke a simulation using GHDL is `ghdl` followed by a list of PoC entities. The following options are supported for GHDL:

Simulator Option		Description
	<code>-board=<BOARD></code>	Specify a target board.
	<code>-device=<DEVICE></code>	Specify a target device.
<code>-g</code>	<code>-gui</code>	Start GTKwave, if installed. Open *.gtkw, if available.
	<code>-std=[87 93 02 08]</code>	Select a VHDL standard. Default: 08

Example:

```
cd PoCRoot
.\poc.ps1 ghdl PoC.arith.prng --board=Atlys -g
```

Mentor Graphics QuestaSim

The command to invoke a simulation using QuestaSim or ModelSim is `vsim` followed by a list of PoC entities. The following options are supported for QuestaSim:

Simulator Option		Description
	<code>-board=<BOARD></code>	Specify a target board.
	<code>-device=<DEVICE></code>	Specify a target device.
<code>-g</code>	<code>-gui</code>	Start the simulation in the QuestaSim GUI.
	<code>-std=[87 93 02 08]</code>	Select a VHDL standard. Default: 08

Example:

```
cd PoCRoot
.\poc.ps1 vsim PoC.arith.prng --board=DE4 --gui
```

If QuestaSim is started in GUI mode (`--gui`), PoC will provide several Tcl files (*.do) in the simulator's working directory to recompile, restart or rerun the current simulation. The rerun command is based on the saved IP core's run script, which may default to `run -all`.

Tcl Script	Performed Tasks
recompile.do	recompile and restart
relaunch.do	recompile, restart and rerun
saveWaveform.do	save the current waveform viewer settings

Xilinx ISE Simulator

The command to invoke a simulation using ISE Simulator (`isim`) is `isim` followed by a list of PoC entities. The following options are supported for ISE Simulator:

Simulator Option	Description
<code>-board=<BOARD></code>	Specify a target board.
<code>-device=<DEVICE></code>	Specify a target device.
<code>-g</code>	Start the simulation in the ISE Simulator GUI (iSim).

Example:

```
cd PoCRoot
.\poc.ps1 isim PoC.arith.prng --board=Atlys -g
```

Xilinx Vivado Simulator

The command to invoke a simulation using Vivado Simulator (`xsim`) is `xsim` followed by a list of PoC entities. The following options are supported for Vivado Simulator:

Simulator Option	Description
<code>-board=<BOARD></code>	Specify a target board.
<code>-device=<DEVICE></code>	Specify a target device.
<code>-g</code>	Start Vivado in simulation mode.
<code>-std=[93 08]</code>	Select a VHDL standard. Default: 93

Example:

```
cd PoCRoot
.\poc.ps1 xsim PoC.arith.prng --board=Atlys -g
```

5.7.5 Running a Group of Testbenches

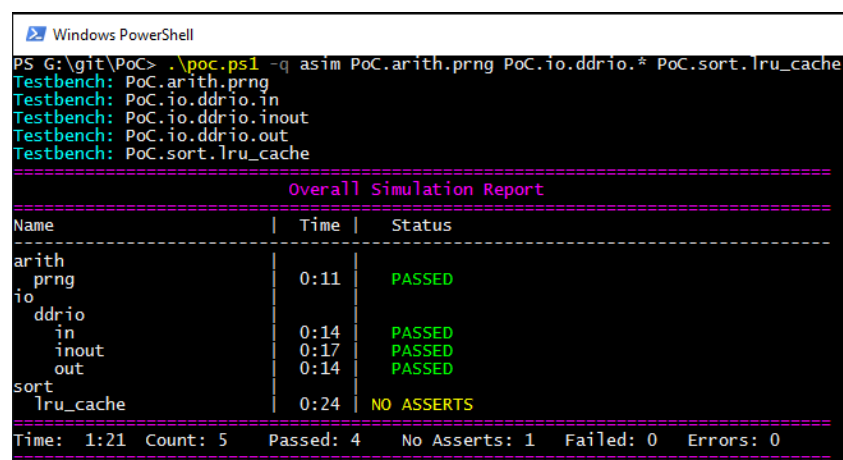
Each simulator can be invoked with a space separated list of PoC entities or a wildcard at the end of the fully qualified entity name.

Supported wildcard patterns are * and ?. Question mark refers to all entities in a PoC (sub-)namespace. Asterisk refers to all PoC entiries in the current namespace and all sub-namespaces.

Examples for testbenches groups:

PoC entity list	Description
PoC.arith.prng	A single PoC entity: arith_prng
PoC.*	All entities in the whole library
PoC.io.ddrio.?	All entities in PoC.io.ddrio: ddrio_in, ddrio_inout, ddrio_out
PoC.fifo.* PoC.cache.* PoC.dstruct.*	All FIFO, cache and data-structure testbenches.

```
cd PoCRoot
.\poc.ps1 -q asim PoC.arith.prng PoC.io.ddrio.* PoC.sort.lru_cache
```



```

PS G:\git\PoC> .\poc.ps1 -q asim PoC.arith.prng PoC.io.ddrio.* PoC.sort.lru_cache
Testbench: PoC.arith.prng
Testbench: PoC.io.ddrio.in
Testbench: PoC.io.ddrio.inout
Testbench: PoC.io.ddrio.out
Testbench: PoC.sort.lru_cache

=====
Overall Simulation Report
=====
Name | Time | Status
-----|-----|-----
arith | 0:11 | PASSED
prng  |      |
io    |      |
ddrio | 0:14 | PASSED
in    | 0:17 | PASSED
inout | 0:14 | PASSED
out   |      |
sort  |      |
lru_cache | 0:24 | NO ASSERTS
=====
Time: 1:21 Count: 5 Passed: 4 No Asserts: 1 Failed: 0 Errors: 0
=====

```

Resulting output: PS G:\git\PoC>

5.7.6 Continuous Integration (CI)

All PoC testbenches are executed on every GitHub upload (push) via Travis-CI. The testsuite runs all testbenches for the virtual board `GENERIC` with an FPGA device called `GENERIC`. We can't run vendor dependent testbenches, because we can't upload the vendor simulation libraries to Travis-CI.

To reproduce the Travis-CI results on a local machine, run the following command. The `-q` option, launches the frontend in quiet mode to reduce the command line messages:

```
cd PoCRoot
.\poc.ps1 -q ghdl PoC.*
```

```

Windows PowerShell
Testbench: PoC.sort.sortnet.OddEvenSort
Testbench: PoC.sort.sortnet.OddEvenMergeSort
Testbench: PoC.sort.sortnet.Stream_Adapter
Testbench: PoC.sort.sortnet.Stream_Adapter2
Testbench: PoC.sort.lru_cache

=====
Overall Simulation Report
=====
Name | Time | Status
-----|-----|-----
arith
  addw          3:04 PASSED
  convert_bin2bcd 0:01 PASSED
  counter_bcd    0:02 PASSED
  div           0:03 PASSED
  firststone     0:02 PASSED
  prefix_and     0:01 PASSED
  prefix_or      0:01 PASSED
  prng           0:01 PASSED
  scaler        0:02 PASSED
dstruct
  deque         0:02 PASSED
  stack         0:02 PASSED
fifo
  cc_got        0:02 PASSED
  cc_got_tempput 0:02 PASSED
  ic_assembly   0:02 PASSED
  ic_got        0:02 PASSED
io
  ddrio
    in          0:01 PASSED
    inout       0:01 PASSED
    out         0:01 PASSED
  uart
    rx          0:02 PASSED
  Debounce     0:02 PASSED
mem
  lut
    Sine        0:01 NO ASSERTS
  ocram
    sdp         0:01 PASSED
misc
  gearbox
    down_cc     0:02 NO ASSERTS
    down_dc     0:02 FAILED
    up_cc       0:02 NO ASSERTS
    up_dc       0:01 FAILED
  stat
    Average     0:02 PASSED
    Histogram   0:04 NO ASSERTS
    Minimum     0:01 PASSED
    Maximum     0:01 PASSED
  sync
    Bits        0:01 PASSED
    Reset       0:01 NO ASSERTS
    Strobe      0:02 NO ASSERTS
    Vector      0:02 NO ASSERTS
    Command     0:02 NO ASSERTS
sort
  sortnet
    BitonicSort 0:56 PASSED
    OddEvenSort 2:54 PASSED
    OddEvenMergeSort 0:46 PASSED
    Stream_Adapter 0:03 PASSED
    Stream_Adapter2 0:04 PASSED
    lru_cache   0:02 NO ASSERTS
=====
Time: 9:07 Count: 41 Passed: 30 No Asserts: 9 Failed: 2 Errors: 0
=====
PS G:\git\PoC>

```

If the vendor libraries are available and pre-compiled, then it's also possible to run a CI flow for a specific vendor. This is an Altera example for the Terrasic DE4 board:

```
cd PoCRoot
.\poc.ps1 -q vsim PoC.* --board=DE4
```

See also:

PoC Configuration See the Configuration page on how to configure PoC and your installed simulator tool chains. This is required to invoke the simulators.

Latest Travis-CI Report Browse the list of branches at Travis-CI.org.

5.8 Synthesis

Contents of this Page

- *Overview*
- *Quick Example*
- *Running a single Synthesis*
 - *Altera / Intel Quartus*
 - *Lattice Diamond*
 - *Xilinx ISE Synthesis Tool (XST)*
 - *Xilinx ISE Core Generator*
 - *Xilinx Vivado Synthesis*

5.8.1 Overview

The Python infrastructure shipped with the PoC-Library can launch manual, half-automated and fully automated synthesis runs. This can be done by invoking one of PoC's frontend script:

- **poc.sh:** `poc.sh <common options> <compiler> <module> <compiler options>` Use this frontend script on Darwin, Linux and Unix platforms.
- **poc.ps1:** `poc.ps1 <common options> <compiler> <module> <compiler options>` Use this frontend script Windows platforms.

Attention: All Windows command line instructions are intended for Windows PowerShell, if not marked otherwise. So executing the following instructions in Windows Command Prompt (`cmd.exe`) won't function or result in errors!

See also:

PoC Configuration See the Configuration page on how to configure PoC and your installed synthesis tool chains. This is required to invoke the compilers.

Supported Compiler See the Intruction page for a list of supported compilers.

See also:

List of Supported FPGA Devices See this list to find a supported and well known target device.

List of Supported Development Boards See this list to find a supported and well known development board.

5.8.2 Quick Example

The following quick example uses the Xilinx Systesis Tool (XST) to synthesize a netlist for IP core `arith_prng` (Pseudo Random Number Generator - PRNG). The VHDL file `arith_prng.vhdl` is located at `PoCRoot\src\arith` and virtually a member in the *PoC.arith* namespace. So the module can be identified by an unique name: `PoC.arith.prng`, which is passed to the frontend script.

Example 1:

```
cd PoCRoot
.\poc.ps1 xst PoC.arith.prng --board=KC705
```

The CLI command `xst` chooses *Xilinx Synthesis Tool* as the synthesizer and passes the fully qualified PoC entity name `PoC.arith.prng` as a parameter to the tool. Additionally, the development board name is required to load the correct `my_config.vhdl` file. All required source files are gathered and synthesized to a netlist.

5.8.3 Running a single Synthesis

A synthesis run is supervised by PoC's *PoCRootPyPoC.py* service tool, which offers a consistent interface to all synthesizers. Unfortunately, every platform has its specialties, so a wrapper script is needed as abstraction from the host's operating system. Depending on the chosen tool chain, the wrapper script will source or invoke the vendor tool's environment scripts to pre-load the needed environment variables, paths or license file settings.

The order of options to the frontend script is as following: `<common options> <synthesizer> <module> [<module>] <synthesizer options>`

The frontend offers several common options:

Common Option		Description
<code>-q</code>	<code>--quiet</code>	Quiet-mode (print nothing)
<code>-v</code>	<code>--verbose</code>	Print more messages
<code>-d</code>	<code>--debug</code>	Debug mode (print everything)
	<code>--dryrun</code>	Run in dry-run mode

One of the following supported synthesizers can be chosen, if installed and configured in PoC:

Synthesizer	Command Reference
<i>Altera Quartus II or Intel Quartus Prime</i>	<code>PoC.py quartus</code>
<i>Lattice (Diamond) Synthesis Engine (LSE)</i>	<code>PoC.py lse</code>
<i>Xilinx ISE Synthesis Tool (XST)</i>	<code>PoC.py xst</code>
<i>Xilinx ISE Core Generator (CoreGen)</i>	<code>PoC.py coregen</code>
<i>Xilinx Vivado Synthesis</i>	<code>PoC.py vivado</code>

Altera / Intel Quartus

The command to invoke a synthesis using Altera Quartus II or Intel Quartus Prime is `quartus` followed by a list of PoC entities. The following options are supported for Quartus:

Simulator Option		Description
	--board=<Board>	Specify a target board.
	--device=<Device>	Specify a target device.

Example:

```
cd PoCRoot
.\poc.ps1 quartus PoC.arith.prng --board=DE4
```

Lattice Diamond

The command to invoke a synthesis using Lattice Diamond is `lse` followed by a list of PoC entities. The following options are supported for the Lattice Synthesis Engine (LSE):

Simulator Option		Description
	--board=<Board>	Specify a target board.
	--device=<Device>	Specify a target device.

Example:

```
cd PoCRoot
.\poc.ps1 lse PoC.arith.prng --board=ECP5Versa
```

Xilinx ISE Synthesis Tool (XST)

The command to invoke a synthesis using Xilinx ISE Synthesis is `xst` followed by a list of PoC entities. The following options are supported for the Xilinx Synthesis Tool (XST):

Simulator Option		Description
	--board=<Board>	Specify a target board.
	--device=<Device>	Specify a target device.

Example:

```
cd PoCRoot
.\poc.ps1 xst PoC.arith.prng --board=KC705
```

Xilinx ISE Core Generator

The command to invoke an IP core generation using Xilinx Core Generator is `coregen` followed by a list of PoC entities. The following options are supported for Core Generator (CG):

Simulator Option		Description
	--board=<Board>	Specify a target board.
	--device=<Device>	Specify a target device.

Example:

```
cd PoCRoot
.\poc.ps1 coregen PoC.xil.mig.Atlys_1x128 --board=Atlys
```

Xilinx Vivado Synthesis

The command to invoke a synthesis using Xilinx Vivado Synthesis is vivado followed by a list of PoC entities. The following options are supported for Vivado Synthesis (Synth):

Simulator Option		Description
	--board=<Board>	Specify a target board.
	--device=<Device>	Specify a target device.

Example:

```
cd PoCRoot
.\poc.ps1 vivado PoC.arith.prng --board=KC705
```

5.9 Project Management

5.9.1 Overview

5.9.2 Solutions

5.9.3 Projects

5.10 Pre-Compiling Vendor Libraries

Contents of this Page

- *Overview*
- *Supported Simulators*
- *FPGA Vendor's Primitive Libraries*
 - *Altera*
 - *Lattice*
 - *Xilinx ISE*
 - *Xilinx Vivado*
- *Third-Party Libraries*
 - *OSVVM*
 - *UVVM*
- *Simulator Adapters*
 - *Cocotb*

5.10.1 Overview

Running vendor specific testbenches may require pre-compiled vendor libraries. Some vendors ship their simulators with diverse pre-compiled libraries, but these don't include primitive libraries from hardware vendors. More over, many auxillary libraries are outdated. Hardware vendors ship their tool chains with pre-compile scripts or user guides to pre-compile the primitive libraries for a list of supported simulators on a target system.

PoC is shipped with a set of pre-compile scripts to offer a unified interface and common storage for all supported vendor's pre-compile procedures. The scripts are located in `\tools\precompile\` and the output is stored in `\temp\precompiled\<Simulator>\<Library>`.

5.10.2 Supported Simulators

The current set of pre-compile scripts support these simulators:

Vendor	Simulator and Edition	Altera	Lattice	Xilinx (ISE)	Xilinx (Vivado)
20. Gingold	GHDL with --std=93c GHDL with --std=08	yes yes	yes yes	yes yes	yes yes
Aldec	Active-HDL (or Student Ed.) Active-HDL Lattice Ed. Reviera-PRO	planned planned planned	planned shipped planned	planned planned planned	planned planned planned
Mentor	ModelSim PE (or Student Ed.) ModelSim SE ModelSim Altera Ed. QuestaSim	yes yes shipped yes	yes yes yes yes	yes yes yes yes	yes yes yes yes
Xilinx	ISE Simulator Vivado Simulator			shipped not supported	not supported shipped

5.10.3 FPGA Vendor's Primitive Libraries

Altera

Note: The Altera Quartus tool chain needs to be configured in PoC. See [Configuring PoC's Infrastructure](#) for further details.

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-altera.sh --all
# Example 2 - Compile only for GHDL and VHDL-2008
./tools/precompile/compile-altera.sh --ghdl --vhdl2008
```


List of command line arguments:

Common Option		Parameter Description
-h	<code>--help</code>	Print embedded help page(s).
-c	<code>--clean</code>	Clean-up directories.
-a	<code>--all</code>	Compile for all simulators.
	<code>--ghdl</code>	Compile for GHDL.
	<code>--questa</code>	Compile for QuestaSim.
	<code>--vhdl93</code>	GHDL only: Compile only for VHDL-93.
	<code>--vhdl2008</code>	GHDL only: Compile only for VHDL-2008.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-altera.ps1 -All
# Example 2 - Compile only for GHDL and VHDL-2008
.\tools\precompile\compile-altera.ps1 -GHDL -VHDL2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<code>-Help</code>	Print embedded help page(s).
-c	<code>-Clean</code>	Clean-up directories.
-a	<code>-All</code>	Compile for all simulators.
	<code>-GHDL</code>	Compile for GHDL.
	<code>-Questa</code>	Compile for QuestaSim.
	<code>-VHDL93</code>	GHDL only: Compile only for VHDL-93.
	<code>-VHDL2008</code>	GHDL only: Compile only for VHDL-2008.

Lattice

Note: The Lattice Diamond tool chain needs to be configured in PoC. See [Configuring PoC's Infrastructure](#) for further details.

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-lattice.sh --all
# Example 2 - Compile only for GHDL and VHDL-2008
./tools/precompile/compile-lattice.sh --ghdl --vhdl2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<code>--help</code>	Print embedded help page(s).
-c	<code>--clean</code>	Clean-up directories.
-a	<code>--all</code>	Compile for all simulators.
	<code>--ghdl</code>	Compile for GHDL.
	<code>--questa</code>	Compile for QuestaSim.
	<code>--vhdl93</code>	GHDL only: Compile only for VHDL-93.
	<code>--vhdl2008</code>	GHDL only: Compile only for VHDL-2008.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-lattice.ps1 -All
# Example 2 - Compile only for GHDL and VHDL-2008
.\tools\precompile\compile-lattice.ps1 -GHDL -VHDL2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>-Help</i>	Print embedded help page(s).
-c	<i>-Clean</i>	Clean-up directories.
-a	<i>-All</i>	Compile for all simulators.
	<i>-GHDL</i>	Compile for GHDL.
	<i>-Questa</i>	Compile for QuestaSim.
	<i>-VHDL93</i>	GHDL only: Compile only for VHDL-93.
	<i>-VHDL2008</i>	GHDL only: Compile only for VHDL-2008.

Xilinx ISE

Note: The Xilinx ISE tool chain needs to be configured in PoC. See *Configuring PoC's Infrastructure* for further details.

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-xilinx-ise.sh --all
# Example 2 - Compile only for GHDL and VHDL-2008
./tools/precompile/compile-xilinx-ise.sh --ghdl --vhd12008
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>--help</i>	Print embedded help page(s).
-c	<i>--clean</i>	Clean-up directories.
-a	<i>--all</i>	Compile for all simulators.
	<i>--ghdl</i>	Compile for GHDL.
	<i>--questa</i>	Compile for QuestaSim.
	<i>--vhd193</i>	GHDL only: Compile only for VHDL-93.
	<i>--vhd12008</i>	GHDL only: Compile only for VHDL-2008.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-xilinx-ise.ps1 -All
# Example 2 - Compile only for GHDL and VHDL-2008
.\tools\precompile\compile-xilinx-ise.ps1 -GHDL -VHDL2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>-Help</i>	Print embedded help page(s).
-c	<i>-Clean</i>	Clean-up directories.
-a	<i>-All</i>	Compile for all simulators.
	<i>-GHDL</i>	Compile for GHDL.
	<i>-Questa</i>	Compile for QuestaSim.
	<i>-VHDL93</i>	GHDL only: Compile only for VHDL-93.
	<i>-VHDL2008</i>	GHDL only: Compile only for VHDL-2008.

Xilinx Vivado

Note: The Xilinx Vivado tool chain needs to be configured in PoC. See [Configuring PoC's Infrastructure](#) for further details.

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-xilinx-vivado.sh --all
# Example 2 - Compile only for GHDL and VHDL-2008
./tools/precompile/compile-xilinx-vivado.sh --ghdl --vhdl2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>--help</i>	Print embedded help page(s).
-c	<i>--clean</i>	Clean-up directories.
-a	<i>--all</i>	Compile for all simulators.
	<i>--ghdl</i>	Compile for GHDL.
	<i>--questa</i>	Compile for QuestaSim.
	<i>--vhdl93</i>	GHDL only: Compile only for VHDL-93.
	<i>--vhdl2008</i>	GHDL only: Compile only for VHDL-2008.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-xilinx-vivado.ps1 -All
# Example 2 - Compile only for GHDL and VHDL-2008
.\tools\precompile\compile-xilinx-vivado.ps1 -GHDL -VHDL2008
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>-Help</i>	Print embedded help page(s).
-c	<i>-Clean</i>	Clean-up directories.
-a	<i>-All</i>	Compile for all simulators.
	<i>-GHDL</i>	Compile for GHDL.
	<i>-Questa</i>	Compile for QuestaSim.
	<i>-VHDL93</i>	GHDL only: Compile only for VHDL-93.
	<i>-VHDL2008</i>	GHDL only: Compile only for VHDL-2008.

5.10.4 Third-Party Libraries

OSVVM

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-osvvm.sh --all
# Example 2 - Compile only for GHDL
./tools/precompile/compile-osvvm.sh --ghdl
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>--help</i>	Print embedded help page(s).
-c	<i>--clean</i>	Clean-up directories.
-a	<i>--all</i>	Compile for all simulators.
	<i>--ghdl</i>	Compile for GHDL.
	<i>--questa</i>	Compile for QuestaSim.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-osvvm.ps1 -All
# Example 2 - Compile only for GHDL
.\tools\precompile\compile-osvvm.ps1 -GHDL
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>-Help</i>	Print embedded help page(s).
-c	<i>-Clean</i>	Clean-up directories.
-a	<i>-All</i>	Compile for all simulators.
	<i>-GHDL</i>	Compile for GHDL.
	<i>-Questa</i>	Compile for QuestaSim.

UVVM

On Linux

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-uvvm.sh --all
# Example 2 - Compile only for GHDL
./tools/precompile/compile-uvvm.sh --ghdl
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>--help</i>	Print embedded help page(s).
-c	<i>--clean</i>	Clean-up directories.
-a	<i>--all</i>	Compile for all simulators.
	<i>--ghdl</i>	Compile for GHDL.
	<i>--questa</i>	Compile for QuestaSim.

On Windows

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-uvvm.ps1 -All
# Example 2 - Compile only for GHDL
.\tools\precompile\compile-uvvm.ps1 -GHDL
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>-Help</i>	Print embedded help page(s).
-c	<i>-Clean</i>	Clean-up directories.
-a	<i>-All</i>	Compile for all simulators.
	<i>-GHDL</i>	Compile for GHDL.
	<i>-Questa</i>	Compile for QuestaSim.

5.10.5 Simulator Adapters

Cocotb

On Linux

Attention: This is an experimental compile script.

```
# Example 1 - Compile for all Simulators
./tools/precompile/compile-cocotb.sh --all
# Example 2 - Compile only for GHDL
./tools/precompile/compile-cocotb.sh --ghdl
```

List of command line arguments:

Common Option		Parameter Description
-h	<i>--help</i>	Print embedded help page(s).
-c	<i>--clean</i>	Clean-up directories.
-a	<i>--all</i>	Compile for all simulators.
	<i>--ghdl</i>	Compile for GHDL.
	<i>--questa</i>	Compile for QuestaSim.

On Windows

Attention: This is an experimental compile script.

```
# Example 1 - Compile for all Simulators
.\tools\precompile\compile-cocotb.ps1 -All
# Example 2 - Compile only for GHDL
.\tools\precompile\compile-cocotb.ps1 -GHDL
```

List of command line arguments:

Common Option		Parameter Description
-h	-Help	Print embedded help page(s).
-c	-Clean	Clean-up directories.
-a	-All	Compile for all simulators.
	-GHDL	Compile for GHDL.
	-Questa	Compile for QuestaSim.

5.11 Miscellaneous

The directory `PoCRoot\tools\` contains several tools and addons to ease the work with the PoC-Library and VHDL.

5.11.1 GNU Emacs

Todo: No documentation available.

5.11.2 Git

- `git-alias.setup.ps1/git-alias.setup.sh` registers new global aliases in Git
 - `git tree` - Prints the colored commit tree into the console
 - `git treea` - Prints the colored commit tree into the console

```
git config --global alias.tree 'log --decorate --pretty=oneline --abbrev-  
→commit --date-order --graph'  
git config --global alias.tree 'log --decorate --pretty=oneline --abbrev-  
→commit --date-order --graph --all'
```

Browse the [Git](#) directory.

5.11.3 Notepad++

The PoC-Library is shipped with syntax highlighting rules for [Notepad++](#). The following additional file types are supported:

- PoC Configuration Files (*.ini)
- PoC *.Files Files* (.files)
- PoC *.Rules Files* (.rules)
- Xilinx User Constraint Files (*.ucf): Syntax Highlighting - Xilinx UCF

Browse the [Notepad++](#) directory.

PoC defines a set of on-chip interfaces described in the next sections.

6.1 Command-Status-Error (PoC.CSE) Interface

Todo: Define the PoC.CSE (Command-Status-Error) interface used in ...

6.2 PoC.FIFO Interface

Todo: Define the PoC.FIFO interface (writer and reader) used in `PoC.fifo.*` ...

6.3 PoC.Mem Interface

PoC.Mem is a single-cycle, pipelined memory interface used by various memory controllers and related components like caches. Memory accesses are always word aligned, and during writes a mask defines which bytes are actually written to the memory (if supported by the memory controller).

6.3.1 Configuration

Each entity may have an individual configuration, especially if it has two PoC.Mem interfaces or if it adapts between PoC.Mem and another interface.

The typical configuration parameters are:

Parameter	Description
ADDR_BITS or A_BITS	Number of address bits. Each address identifies exactly one memory word.
DATA_BITS or D_BITS	Size of a memory word in bits. DATA_BITS must be divisible by 8.

A memory word consists of DATA_BITS/8 bytes.

Individual bytes are only addressed during writes by the write mask. The write mask has one mask-bit for each byte in a memory word.

For example, a 1 KiByte memory with a 32-bit datapath has the following configuration:

- 4 bytes per memory word,
- ADDR_BITS=8 because $\log_2(1 \text{ KiByte}/4 \text{ bytes}) = 8$, and
- DATA_BITS=32 which is the datapath size in bits.

6.3.2 Interface signals

The following signal names are typically prefixed in the port list of a concrete entity to separate the PoC.Mem interface from other interfaces of the entity. Moreover, clock and reset may be shared with other interfaces of the entity.

The PoC.Mem interface consists of the following signals:

Signal	Description
clk	The clock. All other signals are synchronous to the rising edge of this clock.
rst	High-active synchronous reset.
rdy	High-active ready for request.
req	High-active request.
write	'1' if write request, '0' if read request
addr	The (word) address.
wdata	The data to be written to the memory.
wmask (optional)	Write-mask, for each byte: '0' = write byte, '1' = mask byte from write. Signal/port is omitted if write mask is not supported.
rstb	High-active read-strobe.
rdata	The read-data returned from the memory.

The interface is actually splitted into two parts:

- the request part: signals `rdy`, `req`, `write`, `addr`, `wdata` and `wmask`, and
- the read-reply part: signals `rstb` and `rdata`.

6.3.3 Operation

The request and the read-reply part operate indepent of each other to support pipelined reading from memory. The pipeline depth is defined by the actual memory controller. If a user application does support only a specific number of outstanding reads, then the application must limit the number of issued reads on its own.

Requests

If `req` is low, then no request is issued to the memory in the current clock cycle. The state of the signals `write`, `addr`, `wdata` and `wmask` doesn't care.

If `req` is high, then a request is issued to the memory in the current clock cycle as given by `write`, `addr`, `wdata` and `wmask`. The request will be accepted by the memory, if `rdy` is high in the same clock cycle, otherwise the request will be ignored. `wdata` and `wmask` doesn't care if a read request is issued.

`rdy` does not depend on `req` in the current clock cycle. `rdy` may go low in the following clock cycle after a request has been issued or a synchronous reset has been applied.

Read Replies

If `rstb` is high in the current clock cycle, then `rdata` delivers the requested read data (read reply). Otherwise, if `rstb` is low, then `rdata` is unknown. The user application has to immediatly handle the incoming read data, because it cannot signal ready or acknowledge.

After issuing a read request, the memory responds with a read reply in the following clock cycle (i.e. synchronous read) or any later clock cycle depending on the pipeline depth. For each read request, a read reply is generated. Read requests are not reordered.

6.4 PoC.Stream Interface

Todo: Define the `PoC.Stream` interface used in `PoC.net.*` and `PoC.bus.stream.*...`

Namespace for Packages:

7.1 Common Packages

These are common packages. . . .

7.1.1 components

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.2 context

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.3 config

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.4 fileio

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.5 math

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.6 strings

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.7 utils

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.1.8 vectors

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2 Simulation Packages

7.2.1 sim_types

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam

voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.2 sim_global (VHDL-93)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.3 sim_global (VHDL-2008)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.4 sim_unprotected (VHDL-93)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.5 sim_protected (VHDL-2008)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.6 simulation (VHDL-93)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.7 simulation (VHDL-2008)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur

sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.2.8 sim_waveform

Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

Namespaces for Entities:

7.3 PoC.alt

Todo: This namespace is reserved for Altera specific entities.

7.4 PoC.arith

These are arithmetic entities....

Package

PoC.arith Package

Entities

- *PoC.arith.addw*
- *PoC.arith.carrychain_inc*
- *PoC.arith.convert_bin2bcd*
- *PoC.arith.counter_bcd*
- *PoC.arith.counter_free*
- *PoC.arith.counter_gray*
- *PoC.arith.counter_ring*
- *PoC.arith.div*
- *PoC.arith.firstone*
- *PoC.arith.muls_wide*
- *PoC.arith.prefix_and*
- *PoC.arith.prefix_or*
- *PoC.arith.prng*
- *PoC.arith.same*
- *PoC.arith.scaler*
- *PoC.arith.shifter_barrel*
- *PoC.arith.sqrt*

7.4.1 PoC.arith Package

This package holds all component declarations for this namespace.

Exported Enumerations

- `tArch`
- `tBlocking`
- `tSkipping`

Exported Functions

- `arith_div_latency`

Exported Components

- *PoC.arith.addw*
- `PoC.arith.carrychain_inc_xilinx`
- *PoC.arith.counter_bcd*
- *PoC.arith.counter_gray*
- *PoC.arith.div*
- *PoC.arith.firstone*
- `PoC.arith.inc_ovcy_xilinx`
- *PoC.arith.muls_wide*
- `PoC.arith.prefix_and_xilinx`
- `PoC.arith.prefix_or_xilinx`
- *PoC.arith.prng*
- *PoC.arith.same*
- *PoC.arith.sqrt*

Source file: `arith.pkg.vhdl`

7.4.2 PoC.arith.addw

Implements wide addition providing several options all based on an adaptation of a carry-select approach.

References:

- Hong Diep Nguyen and Bogdan Pasca and Thomas B. Preusser: FPGA-Specific Arithmetic Optimizations of Short-Latency Adders, FPL 2011. -> ARCH: AAM, CAI, CCA -> SKIPPING: CCC
- Marcin Rogawski, Kris Gaj and Ekawat Homsirikamol: A Novel Modular Adder for One Thousand Bits and More Using Fast Carry Chains of Modern FPGAs, FPL 2014. -> ARCH: PAI -> SKIPPING: PPN_KS, PPN_BK

Entity Declaration:

```
1 entity arith_addw is
2   generic (
3     N : positive;           -- Operand Width
4     K : positive;           -- Block Count
5
6     ARCH      : tArch      := AAM;      -- Architecture
7     BLOCKING   : tBlocking := DFLT;      -- Blocking Scheme
8     SKIPPING   : tSkipping := CCC;       -- Carry Skip Scheme
9     P_INCLUSIVE : boolean   := false     -- Use Inclusive Propagate, i.e. c^1
10  );
11  port (
12    a, b : in std_logic_vector(N-1 downto 0);
13    cin  : in std_logic;
14
15    s      : out std_logic_vector(N-1 downto 0);
16    cout   : out std_logic
17  );
18 end entity;
```

Source file: arith/arith_addw.vhdl

7.4.3 PoC.arith.carrychain_inc

This is a generic carry-chain abstraction for increment by one operations.

$$Y \leq X + (0 \dots 0) \& Cin$$
Entity Declaration:

```
1 entity arith_carrychain_inc is
2   generic (
3     BITS      : positive
4  );
5  port (
6    X : in std_logic_vector(BITS - 1 downto 0);
7    CIn : in std_logic := '1';
8    Y : out std_logic_vector(BITS - 1 downto 0)
9  );
10 end entity;
```

Source file: arith/arith_carrychain_inc.vhdl

7.4.4 PoC.arith.convert_bin2bcd

Todo: No documentation available.

Entity Declaration:

```
1 entity arith_convert_bin2bcd is
2   generic (
3     BITS      : positive := 8;
```

(continues on next page)

(continued from previous page)

```

4     DIGITS      : positive := 3;
5     RADIX       : positive := 2
6 );
7 port (
8     Clock       : in  std_logic;
9     Reset       : in  std_logic;
10
11     Start       : in  std_logic;
12     Busy        : out std_logic;
13
14     Binary       : in  std_logic_vector (BITS - 1 downto 0);
15     IsSigned     : in  std_logic := '0';
16     BCDDigits    : out T_BCD_VECTOR (DIGITS - 1 downto 0);
17     Sign        : out std_logic
18 );
19 end entity;
```

Source file: `arith/arith_convert_bin2bcd.vhdl`

7.4.5 PoC.arith.counter_bcd

Counter with output in binary coded decimal (BCD). The number of BCD digits is configurable by `DIGITS`.

All control signals (reset `rst`, increment `inc`) are high-active and synchronous to clock `clk`. The output `val` is the current counter state. Groups of 4 bit represent one BCD digit. The lowest significant digit is specified by `val(3 downto 0)`.

Todo:

- implement a `dec` input for decrementing
- implement a `load` input to load a value

Entity Declaration:

```

1 entity arith_counter_bcd is
2     generic (
3         DIGITS : positive -- Number of BCD digits
4     );
5     port (
6         clk : in  std_logic;
7         rst : in  std_logic; -- Reset to 0
8         inc : in  std_logic; -- Increment
9         val : out T_BCD_VECTOR (DIGITS-1 downto 0) -- Value output
10    );
11 end entity;
```

Source file: `arith/arith_counter_bcd.vhdl`

7.4.6 PoC.arith.counter_free

Implements a free-running counter that generates a strobe signal every `DIVIDER`-th cycle the increment input was asserted. There is deliberately no output or specification of the counter value so as to allow an implementation to optimize as much as possible.

The implementation guarantees a strobe output directly from a register. It is asserted exactly for one clock after `DIVIDER` cycles of an asserted increment input have been observed.

Entity Declaration:

```
1 entity arith_counter_free is
2   generic (
3     DIVIDER : positive
4   );
5   port (
6     -- Global Control
7     clk : in std_logic;
8     rst : in std_logic;
9
10    inc : in std_logic;
11    stb : out std_logic           -- End-of-Period Strobe
12  );
13 end entity arith_counter_free;
```

Source file: arith/arith_counter_free.vhdl

7.4.7 PoC.arith.counter_gray

Todo: No documentation available.

Entity Declaration:

```
1 entity arith_counter_gray is
2   generic (
3     BITS : positive;           -- Bit width of the counter
4     INIT : natural             := 0      -- Initial/reset counter value
5   );
6   port (
7     clk : in std_logic;
8     rst : in std_logic;        -- Reset to INIT value
9     inc : in std_logic;        -- Increment
10    dec : in std_logic         := '0';   -- Decrement
11    val : out std_logic_vector (BITS-1 downto 0); -- Value output
12    cry : out std_logic        -- Carry output
13  );
14 end entity arith_counter_gray;
```

Source file: arith/arith_counter_gray.vhdl

7.4.8 PoC.arith.counter_ring

This module implements an up/down ring-counter with loadable initial value (seed) on reset. The counter can be configured to a Johnson counter by enabling `INVERT_FEEDBACK`. The number of counter bits is configurable with `BITS`.

Entity Declaration:

```
1 entity arith_counter_ring is
2   generic (
3     BITS : positive;
4     INVERT_FEEDBACK : boolean := FALSE           -- FALSE_
5   )
6   --> ring counter; TRUE -> johnson counter
```

(continues on next page)

(continued from previous page)

```

5   );
6   port (
7       Clock    : in  std_logic;           -- Clock
8       Reset    : in  std_logic;           -- Reset
9       seed     : in  std_logic_vector(BITS - 1 downto 0) := (others => '0'); --
↳initial counter vector / load value
10      inc      : in  std_logic              := '0';           --
↳increment counter
11      dec      : in  std_logic              := '0';           --
↳decrement counter
12      value    : out std_logic_vector(BITS - 1 downto 0)      --
↳counter value
13   );
14 end entity;

```

Source file: arith/arith_counter_ring.vhdl

7.4.9 PoC.arith.div

Implementation of a Non-Performing restoring divider with a configurable radix. The multi-cycle division is controlled by ‘start’ / ‘rdy’. A new division is started by asserting ‘start’. The result $Q = A/D$ is available when ‘rdy’ returns to ‘1’. A division by zero is identified by output Z. The Q and R outputs are undefined in this case.

Entity Declaration:

```

1  entity arith_div is
2      generic (
3          A_BITS      : positive;           -- Dividend Width
4          D_BITS      : positive;           -- Divisor Width
5          RAPOW       : positive := 1;      -- Power of Compute Radix (2**RAPOW)
6          PIPELINED   : boolean  := false  -- Computation Pipeline
7      );
8      port (
9          -- Global Reset/Clock
10         clk : in  std_logic;
11         rst : in  std_logic;
12
13         -- Ready / Start
14         start : in  std_logic;
15         ready : out std_logic;
16
17         -- Arguments / Result (2's complement)
18         A : in  std_logic_vector(A_BITS-1 downto 0); -- Dividend
19         D : in  std_logic_vector(D_BITS-1 downto 0); -- Divisor
20         Q : out std_logic_vector(A_BITS-1 downto 0); -- Quotient
21         R : out std_logic_vector(D_BITS-1 downto 0); -- Remainder
22         Z : out std_logic  -- Division by Zero
23     );
24 end entity arith_div;

```

Source file: arith/arith_div.vhdl

7.4.10 PoC.arith.firstone

Computes from an input word, a word of the same size that has, at most, one bit set. The output contains a set bit at the position of the rightmost set bit of the input if and only if such a set bit exists in the input.

A typical use case for this computation would be an arbitration over requests with a fixed and strictly ordered priority. The terminology of the interface assumes this use case and provides some useful extras:

- Set `tin` (no input token) to disallow grants altogether.
- Read `tout` (unused token) to see whether or any grant was issued.
- Read `bin` to obtain the binary index of the rightmost detected one bit. The index starts at zero (0) in the rightmost bit position.

This implementation uses carry chains for wider implementations.

Entity Declaration:

```

1 entity arith_firstone is
2   generic (
3     N : positive                                -- Length of Token Chain
4   );
5   port (
6     tin  : in  std_logic := '1';               -- Enable: Fed Token
7     rqst : in  std_logic_vector(N-1 downto 0); -- Request: Token Requests
8     grnt : out std_logic_vector(N-1 downto 0); -- Grant: Token Output
9     tout : out std_logic;                       -- Inactive: Unused Token
10    bin  : out std_logic_vector(log2ceil(N)-1 downto 0) -- Binary Grant Index
11  );
12 end entity arith_firstone;
```

Source file: `arith/arith_firstone.vhdl`

7.4.11 PoC.arith.muls_wide

Signed wide multiplication spanning multiple DSP or MULT blocks. Small partial products are calculated through LUTs. For detailed documentation see below.

Entity Declaration:

Source file: `arith/arith_muls_wide.vhdl`

7.4.12 PoC.arith.prefix_and

Prefix AND computation: `y(i) <= '1' when x(i downto 0) = (i downto 0 => '1') else '0'`; This implementation uses carry chains for wider implementations.

Entity Declaration:

```

1 entity arith_prefix_and is
2   generic (
3     N : positive
4   );
5   port (
6     x : in  std_logic_vector(N-1 downto 0);
7     y : out std_logic_vector(N-1 downto 0)
8   );
9 end entity;
```

Source file: `arith/arith_prefix_and.vhdl`

7.4.13 PoC.arith.prefix_or

Prefix OR computation: $y(i) \leq '0'$ when $x(i \text{ downto } 0) = (i \text{ downto } 0 \Rightarrow '0')$ else $'1'$; This implementation uses carry chains for wider implementations.

Entity Declaration:

```

1 entity arith_prefix_or is
2   generic (
3     N : positive
4   );
5   port (
6     x : in std_logic_vector(N-1 downto 0);
7     y : out std_logic_vector(N-1 downto 0)
8   );
9 end entity;
```

Source file: arith/arith_prefix_or.vhdl

7.4.14 PoC.arith.prng

This module implements a Pseudo-Random Number Generator (PRNG) with configurable bit count (BITS). This module uses an internal list of FPGA optimized polynomials from 3 to 168 bits. The polynomials have at most 5 tap positions, so that long shift registers can be inferred instead of single flip-flops.

The generated number sequence includes the value all-zeros, but not all-ones.

Entity Declaration:

```

1 entity arith_prng is
2   generic (
3     BITS : positive      := 32;
4     SEED : std_logic_vector := "0"
5   );
6   port (
7     clk : in std_logic;
8     rst : in std_logic;      -- reset value to initial seed
9     got : in std_logic;      -- the current value has been_
10    -- got, and a new value should be calculated
11    val : out std_logic_vector(BITS - 1 downto 0) -- the pseudo-random number
12  );
13 end entity;
```

Source file: arith/arith_prng.vhdl

7.4.15 PoC.arith.same

This circuit may, for instance, be used to detect the first sign change and, thus, the range of a two's complement number.

These components may be chained by using the output of the predecessor as guard input. This chaining allows to have intermediate results available while still ensuring the use of a fast carry chain on supporting FPGA architectures. When chaining, make sure to overlap both vector slices by one bit position as to avoid an undetected sign change between the slices.

Entity Declaration:

```
1 entity arith_same is
2   generic (
3     N : positive                                -- Input width
4   );
5   port (
6     g : in std_logic := '1';                  -- Guard Input (!g => !y)
7     x : in std_logic_vector(N-1 downto 0);    -- Input Vector
8     y : out std_logic                          -- All-same Output
9   );
10 end entity;
```

Source file: arith/arith_same.vhdl

7.4.16 PoC.arith.scaler

A flexible scaler for fixed-point values. The scaler is implemented for a set of multiplier and divider values. Each individual scaling operation can arbitrarily select one value from each these sets.

The computation calculates: $\text{unsigned}(\text{arg}) * \text{MULS}(\text{msel}) / \text{DIVS}(\text{dsel})$ rounded to the nearest (tie upwards) fixed-point result of the same precision as *arg*.

The computation is started by asserting *start* to high for one cycle. If a computation is running, it will be restarted. The completion of a calculation is signaled via *done*. *done* is high when no computation is in progress. The result of the last scaling operation is stable and can be read from *res*. The weight of the LSB of *res* is the same as the LSB of *arg*. Make sure to tap a sufficient number of result bits in accordance to the highest scaling ratio to be used in order to avoid a truncation overflow.

Entity Declaration:

```
1 entity arith_scaler is
2   generic (
3     MULS : T_POSVEC := (0 => 1); -- The set of multipliers to choose from in_
4     ↪scaling operations.
5     DIVS : T_POSVEC := (0 => 1)  -- The set of divisors to choose from in scaling_
6     ↪operations.
7   );
8   port (
9     clk : in std_logic;
10    rst : in std_logic;
11
12    start : in std_logic; -- Start of Computation
13    arg : in std_logic_vector; -- Fixed-point value to be scaled
14    msel : in std_logic_vector(log2ceil(MULS'length)-1 downto 0) := (others => '0
15    ↪');
16    dsel : in std_logic_vector(log2ceil(DIVS'length)-1 downto 0) := (others => '0
17    ↪');
18
19    done : out std_logic; -- Completion
20    res : out std_logic_vector -- Result
21  );
22 end entity arith_scaler;
```

Source file: arith/arith_scaler.vhdl

7.4.17 PoC.arith.shifter_barrel

This Barrel-Shifter supports:

- shifting and rotating
- right and left operations
- arithmetic and logic mode (only valid for shift operations)

This is equivalent to the CPU instructions: SLL, SLA, SRL, SRA, RL, RR

Entity Declaration:

```

1 entity arith_shifter_barrel is
2   generic (
3     BITS          : positive      := 32
4   );
5   port (
6     Input          : in  std_logic_vector(BITS - 1 downto 0);
7     ShiftAmount    : in  std_logic_vector(log2ceilnz(BITS) - 1 downto 0);
8     ShiftRotate    : in  std_logic;
9     LeftRight      : in  std_logic;
10    ArithmeticLogic : in  std_logic;
11    Output          : out std_logic_vector(BITS - 1 downto 0)
12  );
13 end entity;
```

Source file: arith/arith_shifter_barrel.vhdl

7.4.18 PoC.arith.sqrt

Iterative Square Root Extractor.

Its computation requires $(N+1)/2$ steps for an argument bit width of N.

Entity Declaration:

```

1 entity arith_sqrt is
2   generic (
3     N : positive -- := 8                                -- Bit Width of Argument
4   );
5   port (
6     -- Global Control
7     rst : in std_logic;                                -- Reset (synchronous)
8     clk : in std_logic;                                -- Clock
9
10    -- Inputs
11    arg : in std_logic_vector(N-1 downto 0); -- Radicand
12    start : in std_logic;                    -- Start Strobe
13
14    -- Outputs
15    sqrt : out std_logic_vector((N-1)/2 downto 0); -- Result
16    rdy : out std_logic;                        -- Ready / Done
17  );
18 end entity arith_sqrt;
```

Source file: arith/arith_sqrt.vhdl

7.5 PoC.bus

These are bus entities....

Sub-namespaces

- *PoC.bus.stream*
- *PoC.bus.wb*

Entities

- *PoC.bus.Arbitrator*

7.5.1 PoC.bus.stream

PoC.Stream modules ...

PoC.bus.stream Package

Source file: `stream.pkg.vhdl`

PoC.bus.stream.Buffer

This module implements a generic buffer (FIFO) for the *PoC.Stream* protocol. It is generic in `DATA_BITS` and in `META_BITS` as well as in FIFO depths for data and meta information.

Entity Declaration:

```
1  entity stream_Buffer is
2      generic (
3          FRAMES           : positive                                := 2;
4          DATA_BITS       : positive                                := 8;
5          DATA_FIFO_DEPTH : positive                                := 8;
6          META_BITS        : T_POSVEC                               := 8;
7          META_FIFO_DEPTH  : T_POSVEC                               := 16;
8      );
9      port (
10         Clock           : in  std_logic;
11         Reset            : in  std_logic;
12         -- IN Port
13         In_Valid         : in  std_logic;
14         In_Data           : in  std_logic_vector (DATA_BITS - 1 downto 0);
15         In_SOF            : in  std_logic;
16         In_EOF            : in  std_logic;
17         In_Ack            : out std_logic;
18         In_Meta_rst       : out std_logic;
19         In_Meta_nxt       : out std_logic_vector (META_BITS'length - 1 downto 0);
20         In_Meta_Data      : in  std_logic_vector (isum(META_BITS) - 1 downto 0);
21         -- OUT Port
22         Out_Valid         : out std_logic;
23         Out_Data          : out std_logic_vector (DATA_BITS - 1 downto 0);
24         Out_SOF           : out std_logic;
```

(continues on next page)

(continued from previous page)

```

25     Out_EOF           : out std_logic;
26     Out_Ack           : in  std_logic;
27     Out_Meta_rst      : in  std_logic;
28     Out_Meta_nxt      : in  std_logic_vector(META_BITS'length - 1 downto 0);
29     Out_Meta_Data     : out std_logic_vector(isum(META_BITS) - 1 downto 0)
30 );
31 end entity;

```

Source file: [bus/stream/stream_Buffer.vhdl](#)

PoC.bus.stream.DeMux

Todo: No documentation available.

Entity Declaration:

```

1  entity stream_DeMux is
2      generic (
3          PORTS           : positive           := 2;
4          DATA_BITS      : positive           := 8;
5          META_BITS       : natural            := 8;
6          META_REV_BITS   : natural            := 2
7      );
8      port (
9          Clock           : in  std_logic;
10         Reset            : in  std_logic;
11         -- Control interface
12         DeMuxControl     : in  std_logic_vector(PORTS - 1 downto 0);
13         -- IN Port
14         In_Valid         : in  std_logic;
15         In_Data           : in  std_logic_vector(DATA_BITS - 1 downto 0);
16         In_Meta          : in  std_logic_vector(META_BITS - 1 downto 0);
17         In_Meta_rev      : out std_logic_vector(META_REV_BITS - 1 downto 0);
18         In_SOF           : in  std_logic;
19         In_EOF           : in  std_logic;
20         In_Ack           : out std_logic;
21         -- OUT Ports
22         Out_Valid        : out std_logic_vector(PORTS - 1 downto 0);
23         Out_Data         : out T_SLM(PORTS - 1 downto 0, DATA_BITS - 1 downto 0);
24         Out_Meta         : out T_SLM(PORTS - 1 downto 0, META_BITS - 1 downto 0);
25         Out_Meta_rev     : in  T_SLM(PORTS - 1 downto 0, META_REV_BITS - 1 downto 0);
26         Out_SOF          : out std_logic_vector(PORTS - 1 downto 0);
27         Out_EOF          : out std_logic_vector(PORTS - 1 downto 0);
28         Out_Ack          : in  std_logic_vector(PORTS - 1 downto 0)
29     );
30 end entity;

```

Source file: [bus/stream/stream_DeMux.vhdl](#)

PoC.bus.stream.Mux

Todo: No documentation available.

Entity Declaration:

```
1 entity stream_Mux is
2   generic (
3     PORTS          : positive          := 2;
4     DATA_BITS     : positive          := 8;
5     META_BITS      : natural           := 8;
6     META_REV_BITS  : natural           := 2--;
7     -- WEIGHTS      : T_INTVEC          := (1, 1)
8   );
9   port (
10    Clock           : in  std_logic;
11    Reset           : in  std_logic;
12    -- IN Ports
13    In_Valid        : in  std_logic_vector(PORTS - 1 downto 0);
14    In_Data         : in  T_SLM(PORTS - 1 downto 0, DATA_BITS - 1 downto 0);
15    In_Meta         : in  T_SLM(PORTS - 1 downto 0, META_BITS - 1 downto 0);
16    In_Meta_rev     : out T_SLM(PORTS - 1 downto 0, META_REV_BITS - 1 downto 0);
17    In_SOF          : in  std_logic_vector(PORTS - 1 downto 0);
18    In_EOF          : in  std_logic_vector(PORTS - 1 downto 0);
19    In_Ack          : out std_logic_vector(PORTS - 1 downto 0);
20    -- OUT Port
21    Out_Valid       : out std_logic;
22    Out_Data        : out std_logic_vector(DATA_BITS - 1 downto 0);
23    Out_Meta        : out std_logic_vector(META_BITS - 1 downto 0);
24    Out_Meta_rev    : in  std_logic_vector(META_REV_BITS - 1 downto 0);
25    Out_SOF         : out std_logic;
26    Out_EOF         : out std_logic;
27    Out_Ack         : in  std_logic
28  );
29 end entity;
```

Source file: [bus/stream/stream_Mux.vhdl](#)

PoC.bus.stream.Mirror

Todo: No documentation available.

Entity Declaration:

```
1 entity stream_Mirror is
2   generic (
3     PORTS          : positive          := 2;
4     DATA_BITS     : positive          := 8;
5     META_BITS      : T_POSVEC          := (0 => 8);
6     META_LENGTH    : T_POSVEC          := (0 => 16)
7   );
8   port (
9     Clock           : in  std_logic;
10    Reset           : in  std_logic;
11    -- IN Port
12    In_Valid        : in  std_logic;
13    In_Data         : in  std_logic_vector(DATA_BITS - 1 downto 0);
14    In_SOF          : in  std_logic;
15    In_EOF          : in  std_logic;
16    In_Ack          : out std_logic;
17    In_Meta_rst     : out std_logic;
```

(continues on next page)

(continued from previous page)

```

18   In_Meta_nxt      : out std_logic_vector(META_BITS'length - 1 downto 0);
19   In_Meta_Data     : in  std_logic_vector(isum(META_BITS) - 1 downto 0);
20   -- OUT Port
21   Out_Valid        : out std_logic_vector(PORTS - 1 downto 0);
22   Out_Data         : out T_SLM(PORTS - 1 downto 0, DATA_BITS - 1 downto 0);
23   Out_SOF          : out std_logic_vector(PORTS - 1 downto 0);
24   Out_EOF          : out std_logic_vector(PORTS - 1 downto 0);
25   Out_Ack          : in  std_logic_vector(PORTS - 1 downto 0);
26   Out_Meta_rst     : in  std_logic_vector(PORTS - 1 downto 0);
27   Out_Meta_nxt     : in  T_SLM(PORTS - 1 downto 0, META_BITS'length - 1 downto
↪0);
28   Out_Meta_Data    : out T_SLM(PORTS - 1 downto 0, isum(META_BITS) - 1 downto 0)
29   );
30 end entity;
```

Source file: [bus/stream/stream_Mirror.vhdl](#)

PoC.bus.stream.Sink

Todo: No documentation available.

Entity Declaration:

Source file: [bus/stream/stream_Sink.vhdl](#)

PoC.bus.stream.Source

Todo: No documentation available.

Entity Declaration:

```

1  entity stream_Source is
2    generic (
3      TESTCASES          : T_SIM_STREAM_FRAMEGROUP_VECTOR_8
4    );
5    port (
6      Clock               : in  std_logic;
7      Reset               : in  std_logic;
8      -- Control interface
9      Enable              : in  std_logic;
10     -- OUT Port
11     Out_Valid            : out std_logic;
12     Out_Data             : out T_SLV_8;
13     Out_SOF              : out std_logic;
14     Out_EOF              : out std_logic;
15     Out_Ack              : in  std_logic
16   );
17 end entity;
```

Source file: [bus/stream/stream_Source.vhdl](#)

PoC.bus.stream.FrameGenerator

Todo: No documentation available.

Entity Declaration:

```
1 entity stream_FrameGenerator is
2   generic (
3     DATA_BITS      : positive           := 8;
4     WORD_BITS       : positive           := 16;
5     APPEND           : T_FRAMEGEN_APPEND := FRAMEGEN_APP_NONE;
6     FRAMEGROUPS      : T_FRAMEGEN_FRAMEGROUP_VECTOR_8 := (0 => C_FRAMEGEN_
    ↪ FRAMEGROUP_EMPTY)
7   );
8   port (
9     Clock           : in  std_logic;
10    Reset            : in  std_logic;
11    -- CSE interface
12    Command          : in  T_FRAMEGEN_COMMAND;
13    Status            : out T_FRAMEGEN_STATUS;
14    -- Control interface
15    Pause            : in  T_SLV_16;
16    FrameGroupIndex  : in  T_SLV_8;
17    FrameIndex       : in  T_SLV_8;
18    Sequences        : in  T_SLV_16;
19    FrameLength      : in  T_SLV_16;
20    -- OUT Port
21    Out_Valid        : out std_logic;
22    Out_Data          : out std_logic_vector (DATA_BITS - 1 downto 0);
23    Out_SOF           : out std_logic;
24    Out_EOF           : out std_logic;
25    Out_Ack           : in  std_logic
26  );
27 end entity;
```

Source file: [bus/stream/stream_FrameGenerator.vhdl](#)

7.5.2 PoC.bus.wb

WishBone modules ...

Entities:

PoC.bus.wb Package

Source file: [wb.pkg.vhdl](#)

PoC.bus.wb.ocram

This slave supports Wishbone Registered Feedback bus cycles (aka. burst transfers / advanced synchronous cycle termination). The mode “Incrementing burst cycle” (CTI = 010) with “Linear burst” (BTE = 00) is supported.

If your master does support Wishbone Classis bus cycles only, then connect `wb_cti_i` = “000” and `wb_bte_i` = “00”.

Connect the ocram of your choice to the `ram_*` port signals. (Every RAM with single cycle read latency is supported.)

Configuration:

PIPE_STAGES = 1 The RAM output is directly connected to the bus. Thus, the read access latency (one cycle) is short. But, the RAM's read timing delay must be respected.

PIPE_STAGES = 2 The RAM output is registered again. Thus, the read access latency is two cycles.

Entity Declaration:

Source file: `bus/wb/wb_ocram.vhdl`

PoC.bus.wb.fifo_adapter

Small FIFOs are included in this module, if larger or asynchronous transmit / receive FIFOs are required, then they must be connected externally.

old comments: UART BAUD rate generator `bclk_r` = bit clock is rising `bclk_x8_r` = bit clock times 8 is rising

Entity Declaration:

Source file: `bus/wb/wb_fifo_adapter.vhdl`

PoC.bus.wb.uart_wrapper

Wrapper module for *PoC.io.uart.rx* and *PoC.io.uart.tx* to support the Wishbone interface. Synchronized reset is used.

Entity Declaration:

Source file: `bus/wb/wb_uart_wrapper.vhdl`

7.5.3 PoC.bus.Arbitrator

This module implements a generic arbitrator. It currently supports the following arbitration strategies:

- Round Robin (RR)

Entity Declaration:

```

1  entity bus_Arbitrator is
2      generic (
3          STRATEGY          : string          := "RR";          -- RR, LOT
4          PORTS              : positive       := 1;
5          WEIGHTS            : T_INTVEC      := (0 => 1);
6          OUTPUT_REG        : boolean        := TRUE
7      );
8      port (
9          Clock              : in  std_logic;
10         Reset              : in  std_logic;
11
12         Arbitrate           : in  std_logic;
13         Request_Vector      : in  std_logic_vector (PORTS - 1 downto 0);
14

```

(continues on next page)

(continued from previous page)

```

15   Arbitrated           : out std_logic;
16   Grant_Vector         : out std_logic_vector (PORTS - 1 downto 0);
17   Grant_Index          : out std_logic_vector (log2ceilnz (PORTS) - 1 downto
↪ 0)
18   );
19 end entity;
```

Source file: `bus/bus_Arbiter.vhdl`

7.6 PoC.cache

The namespace *PoC.cache* offers different cache implementations.

Entities

- *PoC.cache.cpu*: Cache with cache controller to be used within a CPU.
- *PoC.cache.mem*: Cache with *PoC.Mem Interface* interface on the “CPU” side.
- *PoC.cache.par*: Cache with parallel tag-unit and data memory (using inferred memory).
- *PoC.cache.par2*: Cache with parallel tag-unit and data memory (using *PoC.mem.ocram.sp*).
- *PoC.cache.tagunit_par*: Tag-Unit with parallel tag comparison. Configurable as:
 - Full-associative cache,
 - Direct-mapped cache, or
 - Set-associative cache.
- *PoC.cache.tagunit_seq*: Tag-Unit with sequential tag comparison. Configurable as:
 - Full-associative cache,
 - Direct-mapped cache, or
 - Set-associative cache.

7.6.1 PoC.cache.cpu

This unit provides a cache (*PoC.cache.par2*) together with a cache controller which reads / writes cache lines from / to memory. The memory is accessed using a *PoC.Mem Interface* interfaces, the related ports and parameters are prefixed with `mem_`.

The CPU side (prefix `cpu_`) has a modified *PoC.Mem* interface, so that this unit can be easily integrated into processor pipelines. For example, let’s have a pipeline where a load/store instruction is executed in 3 stages (after fetching, decoding, ...):

1. Execute (EX) for address calculation,
2. Load/Store 1 (LS1) for the cache access,
3. Load/Store 2 (LS2) where the cache returns the read data.

The read data is always returned one cycle after the cache access completes, so there is conceptually a pipeline register within this unit. The stage LS2 can be merged with a write-back stage if the clock period allows so.

The stage LS1 and thus EX and LS2 must stall, until the cache access is completed, i.e., the EX/LS1 pipeline register must hold the cache request until it is acknowledged by the cache. This is signaled by `cpu_got` as described in Section Operation below. The pipeline moves forward (is enabled) when:

```
pipeline_enable <= (not cpu_req) or cpu_got;
```

If the pipeline can stall due to other reasons, care must be taken to not unintentionally executing the cache access twice or missing the read data.

Of course, the EX/LS1 pipeline register can be omitted and the CPU side directly fed by the address calculator. But be aware of the high setup time of this unit and high propagate time for `cpu_got`.

This unit supports only one outstanding CPU request. More outstanding requests are provided by [PoC.cache.mem](#).

Configuration

Parameter	Description
REPLACE- MENT_POLICY	Replacement policy of embedded cache. For supported values see <code>PoC.cache_replacement_policy</code> .
CACHE_LINES	Number of cache lines.
ASSOCIATIV- ITY	Associativity of embedded cache.
CPU_ADDR_BITS	Number of address bits on the CPU side. Each address identifies one memory word as seen from the CPU. Calculated from other parameters as described below.
CPU_DATA_BITS	Width of the data bus (in bits) on the CPU side. CPU_DATA_BITS must be divisible by 8.
MEM_ADDR_BITS	Number of address bits on the memory side. Each address identifies one word in the memory.
MEM_DATA_BITS	Width of a memory word and of a cache line in bits. MEM_DATA_BITS must be divisible by CPU_DATA_BITS.

If the CPU data-bus width is smaller than the memory data-bus width, then the CPU needs additional address bits to identify one CPU data word inside a memory word. Thus, the CPU address-bus width is calculated from:

$$\text{CPU_ADDR_BITS} = \log_2 \text{ceil}(\text{MEM_DATA_BITS} / \text{CPU_DATA_BITS}) + \text{MEM_ADDR_BITS}$$

The write policy is: write-through, no-write-allocate.

Operation

Alignment of Cache / Memory Accesses

Memory accesses are always aligned to a word boundary. Each memory word (and each cache line) consists of MEM_DATA_BITS bits. For example if MEM_DATA_BITS=128:

- memory address 0 selects the bits 0..127 in memory,
- memory address 1 selects the bits 128..256 in memory, and so on.

Cache accesses are always aligned to a CPU word boundary. Each CPU word consists of CPU_DATA_BITS bits. For example if CPU_DATA_BITS=32:

- CPU address 0 selects the bits 0.. 31 in memory word 0,
- CPU address 1 selects the bits 32.. 63 in memory word 0,
- CPU address 2 selects the bits 64.. 95 in memory word 0,
- CPU address 3 selects the bits 96..127 in memory word 0,
- CPU address 4 selects the bits 0.. 31 in memory word 1,
- CPU address 5 selects the bits 32.. 63 in memory word 1, and so on.

Shared and Memory Side Interface

A synchronous reset must be applied even on a FPGA.

The memory side interface is documented in detail [here](#).

CPU Side Interface

The CPU (pipeline stage LS1, see above) issues a request by setting `cpu_req`, `cpu_write`, `cpu_addr`, `cpu_wdata` and `cpu_wmask` as in the *PoC.Mem Interface* interface. The cache acknowledges the request by setting `cpu_got` to '1'. If the request is not acknowledged (`cpu_got = '0'`) in the current clock cycle, then the request must be repeated in the following clock cycle(s) until it is acknowledged, i.e., the pipeline must stall.

A cache access is completed when it is acknowledged. A new request can be issued in the following clock cycle.

Of course, `cpu_got` may be asserted in the same clock cycle where the request was issued if a read hit occurs. This allows a throughput of one (read) request per clock cycle, but the drawback is, that `cpu_got` has a high propagation delay. Thus, this output should only control a simple pipeline enable logic.

When `cpu_got` is asserted for a read access, then the read data will be available in the following clock cycle.

Due to the write-through policy, a write will always take several clock cycles and acknowledged when the data has been issued to the memory.

Warning: If the design is synthesized with Xilinx ISE / XST, then the synthesis option “Keep Hierarchy” must be set to SOFT or TRUE.

Entity Declaration:

```

1 entity cache_cpu is
2   generic (
3     REPLACEMENT_POLICY : string := "LRU";
4     CACHE_LINES         : positive;
5     ASSOCIATIVITY       : positive;
6     CPU_DATA_BITS       : positive;
7     MEM_ADDR_BITS       : positive;
8     MEM_DATA_BITS       : positive
9   );
10  port (
11    clk : in std_logic; -- clock
12    rst : in std_logic; -- reset
13
14    -- "CPU" side
15    cpu_req  : in  std_logic;
16    cpu_write : in  std_logic;
17    cpu_addr  : in  unsigned(log2ceil(MEM_DATA_BITS/CPU_DATA_BITS)+MEM_ADDR_BITS-1,
18    ↪downto 0);
19    cpu_wdata : in  std_logic_vector(CPU_DATA_BITS-1 downto 0);
20    cpu_wmask : in  std_logic_vector(CPU_DATA_BITS/8-1 downto 0);
21    cpu_got   : out std_logic;
22    cpu_rdata : out std_logic_vector(CPU_DATA_BITS-1 downto 0);
23
24    -- Memory side
25    mem_req   : out std_logic;
26    mem_write : out std_logic;
27    mem_addr  : out unsigned(MEM_ADDR_BITS-1 downto 0);
28    mem_wdata : out std_logic_vector(MEM_DATA_BITS-1 downto 0);

```

(continues on next page)

(continued from previous page)

```

28     mem_wmask : out std_logic_vector(MEM_DATA_BITS/8-1 downto 0);
29     mem_rdy   : in  std_logic;
30     mem_rstb  : in  std_logic;
31     mem_rdata : in  std_logic_vector(MEM_DATA_BITS-1 downto 0)
32 );
33 end entity;
```

See also:*PoC.cache.mem*Source file: `cache/cache_cpu.vhdl`

7.6.2 PoC.cache.mem

This unit provides a cache (*PoC.cache.par2*) together with a cache controller which reads / writes cache lines from / to memory. It has two *PoC.Mem Interface* interfaces:

- one for the “CPU” side (ports with prefix `cpu_`), and
- one for the memory side (ports with prefix `mem_`).

Thus, this unit can be placed into an already available memory path between the CPU and the memory (controller). If you want to plugin a cache into a CPU pipeline, see *PoC.cache.cpu*.

Configuration

Parameter	Description
REPLACE-MENT_POLICY	Replacement policy of embedded cache. For supported values see <code>PoC.cache_replacement_policy</code> .
CACHE_LINES	Number of cache lines.
ASSOCIATIV-ITY	Associativity of embedded cache.
CPU_ADDR_BITS	Number of address bits on the CPU side. Each address identifies one memory word as seen from the CPU. Calculated from other parameters as described below.
CPU_DATA_BITS	Width of the data bus (in bits) on the CPU side. <code>CPU_DATA_BITS</code> must be divisible by 8.
MEM_ADDR_BITS	Number of address bits on the memory side. Each address identifies one word in the memory.
MEM_DATA_BITS	Width of a memory word and of a cache line in bits. <code>MEM_DATA_BITS</code> must be divisible by <code>CPU_DATA_BITS</code> .
OUTSTAND-ING_REQ	Number of outstanding requests, see notes below.

If the CPU data-bus width is smaller than the memory data-bus width, then the CPU needs additional address bits to identify one CPU data word inside a memory word. Thus, the CPU address-bus width is calculated from:

$$\text{CPU_ADDR_BITS} = \log_2 \text{ceil}(\text{MEM_DATA_BITS} / \text{CPU_DATA_BITS}) + \text{MEM_ADDR_BITS}$$

The write policy is: write-through, no-write-allocate.

The maximum throughput is one request per clock cycle, except for `OUTSTANDING_REQ = 1`.

If `OUTSTANDING_REQ` is:

- 1: then 1 request is buffered by a single register. To give a short critical path (clock-to-output delay) for `cpu_rdy`, the throughput is degraded to one request per 2 clock cycles at maximum.
- 2: then 2 requests are buffered by *PoC.fifo.glue*. This setting has the lowest area requirements without degrading the performance.

- >2: then the requests are buffered by *PoC.fifo.cc_got*. The number of outstanding requests is rounded up to the next suitable value. This setting is useful in applications with out-of-order execution (of other operations). The CPU requests to the cache are always processed in-order.

Operation

Memory accesses are always aligned to a word boundary. Each memory word (and each cache line) consists of MEM_DATA_BITS bits. For example if MEM_DATA_BITS=128:

- memory address 0 selects the bits 0..127 in memory,
- memory address 1 selects the bits 128..256 in memory, and so on.

Cache accesses are always aligned to a CPU word boundary. Each CPU word consists of CPU_DATA_BITS bits. For example if CPU_DATA_BITS=32:

- CPU address 0 selects the bits 0.. 31 in memory word 0,
- CPU address 1 selects the bits 32.. 63 in memory word 0,
- CPU address 2 selects the bits 64.. 95 in memory word 0,
- CPU address 3 selects the bits 96..127 in memory word 0,
- CPU address 4 selects the bits 0.. 31 in memory word 1,
- CPU address 5 selects the bits 32.. 63 in memory word 1, and so on.

A synchronous reset must be applied even on a FPGA.

The interface is documented in detail [here](#).

Warning: If the design is synthesized with Xilinx ISE / XST, then the synthesis option “Keep Hierarchy” must be set to SOFT or TRUE.

Entity Declaration:

```

1  entity cache_mem is
2      generic (
3          REPLACEMENT_POLICY : string := "LRU";
4          CACHE_LINES         : positive;
5          ASSOCIATIVITY        : positive;
6          CPU_DATA_BITS        : positive;
7          MEM_ADDR_BITS        : positive;
8          MEM_DATA_BITS        : positive;
9          OUTSTANDING_REQ      : positive := 2
10     );
11     port (
12         clk : in std_logic; -- clock
13         rst : in std_logic; -- reset
14
15         -- "CPU" side
16         cpu_req  : in std_logic;
17         cpu_write : in std_logic;
18         cpu_addr  : in unsigned(log2ceil(MEM_DATA_BITS/CPU_DATA_BITS)+MEM_ADDR_BITS-1,
19         ↪downto 0);
20         cpu_wdata : in std_logic_vector(CPU_DATA_BITS-1 downto 0);
21         cpu_wmask : in std_logic_vector(CPU_DATA_BITS/8-1 downto 0) := (others => '0
22         ↪');
23         cpu_rdy   : out std_logic;
24         cpu_rstb  : out std_logic;
25         cpu_rdata : out std_logic_vector(CPU_DATA_BITS-1 downto 0);

```

(continues on next page)

(continued from previous page)

```

24
25  -- Memory side
26  mem_req   : out std_logic;
27  mem_write : out std_logic;
28  mem_addr  : out unsigned(MEM_ADDR_BITS-1 downto 0);
29  mem_wdata : out std_logic_vector(MEM_DATA_BITS-1 downto 0);
30  mem_wmask : out std_logic_vector(MEM_DATA_BITS/8-1 downto 0);
31  mem_rdy   : in  std_logic;
32  mem_rstb  : in  std_logic;
33  mem_rdata : in  std_logic_vector(MEM_DATA_BITS-1 downto 0)
34  );
35 end entity;

```

See also:

[PoC.cache.cpu](#)

Source file: [cache/cache_mem.vhdl](#)

7.6.3 PoC.cache.par

Implements a cache with parallel tag-unit and data memory.

Note: This component infers a single-port memory with read-first behavior, that is, upon writes the old-data is returned on the read output. Such memory (e.g. LUT-RAM) is not available on all devices. Thus, synthesis may infer a lot of flip-flops plus multiplexers instead, which is very inefficient. It is recommended to use [PoC.cache.par2](#) instead which has a slightly different interface.

All inputs are synchronous to the rising-edge of the clock *clock*.

Command truth table:

Request	ReadWrite	Invalidate	Replace	Command
0	0	0	0	None
1	0	0	0	Read cache line
1	1	0	0	Update cache line
1	0	1	0	Read cache line and discard it
1	1	1	0	Write cache line and discard it
0		0	1	Replace cache line.

All commands use *Address* to lookup (request) or replace a cache line. *Address* and *OldAddress* do not include the word/byte select part. Each command is completed within one clock cycle, but outputs are delayed as described below.

Upon requests, the outputs *CacheMiss* and *CacheHit* indicate (high-active) whether the *Address* is stored within the cache, or not. Both outputs have a latency of one clock cycle.

Upon writing a cache line, the new content is given by *CacheLineIn*. Upon reading a cache line, the current content is outputted on *CacheLineOut* with a latency of one clock cycle.

Upon replacing a cache line, the new content is given by *CacheLineIn*. The old content is outputted on *CacheLineOut* and the old tag on *OldAddress*, both with a latency of one clock cycle.

Warning: If the design is synthesized with Xilinx ISE / XST, then the synthesis option “Keep Hierarchy” must be set to SOFT or TRUE.

Entity Declaration:

```
1 entity cache_par is
2   generic (
3     REPLACEMENT_POLICY : string := "LRU";
4     CACHE_LINES         : positive := 32; --1024;
5     ASSOCIATIVITY       : positive := 32; --4;
6     ADDRESS_BITS        : positive := 8;  --32-6;
7     DATA_BITS          : positive := 8   --64*8
8   );
9   port (
10    Clock : in std_logic;
11    Reset : in std_logic;
12
13    Request      : in std_logic;
14    ReadWrite    : in std_logic;
15    Invalidate   : in std_logic;
16    Replace      : in std_logic;
17    Address      : in std_logic_vector(ADDRESS_BITS - 1 downto 0);
18
19    CacheLineIn  : in std_logic_vector(DATA_BITS - 1 downto 0);
20    CacheLineOut : out std_logic_vector(DATA_BITS - 1 downto 0);
21    CacheHit     : out std_logic := '0';
22    CacheMiss    : out std_logic := '0';
23    OldAddress   : out std_logic_vector(ADDRESS_BITS - 1 downto 0)
24  );
25 end entity;
```

Source file: [cache/cache_par.vhdl](#)

7.6.4 PoC.cache.par2

Cache with parallel tag-unit and data memory. For the data memory, *PoC.mem.ocram.sp* is used.

Configuration

Parameter	Description
REPLACE-MENT_POLICY	Replacement policy. For supported policies see PoC.cache_replacement_policy.
CACHE_LINES	Number of cache lines.
ASSOCIATIVITY	Associativity of the cache.
ADDR_BITS	Number of address bits. Each address identifies exactly one cache line in memory.
DATA_BITS	Size of a cache line in bits. DATA_BITS must be divisible by 8.

Command truth table

Request	ReadWrite	Invalidate	Replace	Command
0	0	0	0	None
1	0	0	0	Read cache line
1	1	0	0	Update cache line
1	0	1	0	Read cache line and discard it
1	1	1	0	Write cache line and discard it
0	0	0	1	Read cache line before replace.
0	1	0	1	Replace cache line.

Operation

All inputs are synchronous to the rising-edge of the clock *clock*.

All commands use *Address* to lookup (request) or replace a cache line. *Address* and *OldAddress* do not include the word/byte select part. Each command is completed within one clock cycle, but outputs are delayed as described below.

Upon requests, the outputs *CacheMiss* and *CacheHit* indicate (high-active) whether the *Address* is stored within the cache, or not. Both outputs have a latency of one clock cycle (pipelined) if *HIT_MISS_REG* is true, otherwise the result is outputted immediately (combinational).

Upon writing a cache line, the new content is given by *CacheLineIn*. Only the bytes which are not masked, i.e. the corresponding bit in *WriteMask* is '0', are actually written.

Upon reading a cache line, the current content is outputted on *CacheLineOut* with a latency of one clock cycle.

Replacing a cache line requires two steps, both with *Replace* = '1':

1. Read old contents of cache line by setting *ReadWrite* to '0'. The old content is outputted on *CacheLineOut* and the old tag on *OldAddress*, both with a latency of one clock cycle.
2. Write new cache line by setting *ReadWrite* to '1'. The new content is given by *CacheLineIn*. All bytes shall be written, i.e. *WriteMask* = 0. The new cache line content will be outputted again on *CacheLineOut* in the next clock cycle (latency = 1).

Warning: If the design is synthesized with Xilinx ISE / XST, then the synthesis option "Keep Hierarchy" must be set to SOFT or TRUE.

Entity Declaration:

```

1  entity cache_par2 is
2      generic (
3          REPLACEMENT_POLICY : string    := "LRU";
4          CACHE_LINES        : positive  := 32;
5          ASSOCIATIVITY       : positive  := 32;
6          ADDR_BITS           : positive  := 8;
7          DATA_BITS          : positive  := 8;
8          HIT_MISS_REG        : boolean   := true  -- must be true for Cocotb.
9      );
10     port (
11         Clock : in std_logic;
12         Reset  : in std_logic;
13
14         Request      : in std_logic;
15         ReadWrite    : in std_logic;
16         WriteMask     : in std_logic_vector(DATA_BITS/8 - 1 downto 0) := (others => '0');
17         Invalidate    : in std_logic;
18         Replace       : in std_logic;
19         Address       : in std_logic_vector(ADDR_BITS-1 downto 0);
20
21         CacheLineIn  : in std_logic_vector(DATA_BITS - 1 downto 0);
22         CacheLineOut : out std_logic_vector(DATA_BITS - 1 downto 0);
23         CacheHit      : out std_logic := '0';
24         CacheMiss     : out std_logic := '0';
25         OldAddress    : out std_logic_vector(ADDR_BITS-1 downto 0)
26     );
27 end entity;
```

Source file: `cache/cache_par2.vhdl`

7.6.5 PoC.cache.replacement_policy

Supported policies:

Abbr.	Policies	supported
RR	round robin	not yet
RAND	random	not yet
CLOCK	clock algorithm	not yet
LRU	least recently used	YES
LFU	least frequently used	not yet

Command thruth table:

TagAccess	ReadWrite	Invalidate	Replace	Command
0			0	None
1	0	0	0	TagHit and reading a cache line
1	1	0	0	TagHit and writing a cache line
1	0	1	0	TagHit and invalidate a cache line (while reading)
1	1	1	0	TagHit and invalidate a cache line (while writing)
0		0	1	Replace cache line

In a set-associative cache, each cache-set has its own instance of this component.

The input `HitWay` specifies the accessed way in a fully-associative or set-associative cache.

The output `ReplaceWay` identifies the way which will be replaced as next by a replace command. In a set-associative cache, this is the way in a specific cache set (see above).

Entity Declaration:

```
1  entity cache_replacement_policy is
2      generic (
3          REPLACEMENT_POLICY : string      := "LRU";
4          CACHE_WAYS          : positive   := 32
5      );
6      port (
7          Clock : in std_logic;
8          Reset : in std_logic;
9
10         -- replacement interface
11         Replace      : in std_logic;
12         ReplaceWay   : out std_logic_vector(log2ceilnz(CACHE_WAYS) - 1 downto 0);
13
14         -- cacheline usage update interface
15         TagAccess    : in std_logic;
16         ReadWrite    : in std_logic;
17         Invalidate    : in std_logic;
18         HitWay       : in std_logic_vector(log2ceilnz(CACHE_WAYS) - 1 downto 0)
19     );
20 end entity;
```

Source file: `cache/cache_replacement_policy.vhdl`

7.6.6 PoC.cache.tagunit_par

Tag-unit with fully-parallel compare of tag.

Configuration

Parameter	Description
REPLACE- MENT_POLICY	Replacement policy. For supported policies see PoC.cache_replacement_policy.
CACHE_LINES	Number of cache lines.
ASSOCIATIVITY	Associativity of the cache.
ADDRESS_BITS	Number of address bits. Each address identifies exactly one cache line in memory.

Command truth table

Request	ReadWrite	Invalidate	Replace	Command
0	0	0	0	None
1	0	0	0	Read cache line
1	1	0	0	Update cache line
1	0	1	0	Read cache line and discard it
1	1	1	0	Write cache line and discard it
0		0	1	Replace cache line.

Operation

All inputs are synchronous to the rising-edge of the clock *clock*.

All commands use *Address* to lookup (request) or replace a cache line. Each command is completed within one clock cycle.

Upon requests, the outputs *CacheMiss* and *CacheHit* indicate (high-active) immediately (combinational) whether the *Address* is stored within the cache, or not. But, the cache-line usage is updated at the rising-edge of the clock. If hit, *LineIndex* specifies the cache line where to find the content.

The output *ReplaceLineIndex* indicates which cache line will be replaced as next by a replace command. The output *OldAddress* specifies the old tag stored at this index. The replace command will store the *Address* and update the cache-line usage at the rising-edge of the clock.

For a direct-mapped cache, the number of *CACHE_LINES* must be a power of 2. For a set-associative cache, the expression $CACHE_LINES / ASSOCIATIVITY$ must be a power of 2.

Note: The port *NewAddress* has been removed. Use *Address* instead as described above.

If *Address* is fed from a register and an Altera FPGA is used, then Quartus Map converts the tag memory from a memory with asynchronous read to a memory with synchronous read by adding a pass-through logic. Quartus Map reports warning 276020 which is intended.

Warning: If the design is synthesized with Xilinx ISE / XST, then the synthesis option “Keep Hierarchy” must be set to SOFT or TRUE.

Entity Declaration:

```

1 entity cache_tagunit_par is
2   generic (
3     REPLACEMENT_POLICY : string := "LRU";

```

(continues on next page)

(continued from previous page)

```

4  CACHE_LINES      : positive := 32;
5  ASSOCIATIVITY    : positive := 32;
6  ADDRESS_BITS     : positive := 8;
7  );
8  port (
9      Clock : in std_logic;
10     Reset : in std_logic;
11
12     Replace      : in std_logic;
13     ReplaceLineIndex : out std_logic_vector(log2ceilnz(CACHE_LINES) - 1 downto 0);
14     OldAddress    : out std_logic_vector(ADDRESS_BITS - 1 downto 0);
15
16     Request      : in std_logic;
17     ReadWrite    : in std_logic;
18     Invalidate   : in std_logic;
19     Address      : in std_logic_vector(ADDRESS_BITS - 1 downto 0);
20     LineIndex    : out std_logic_vector(log2ceilnz(CACHE_LINES) - 1 downto 0);
21     TagHit       : out std_logic;
22     TagMiss      : out std_logic;
23 );
24 end entity;

```

Source file: `cache/cache_tagunit_par.vhdl`

7.6.7 PoC.cache.tagunit_seq

Todo: No documentation available.

Entity Declaration:

```

1  entity cache_tagunit_seq is
2      generic (
3          REPLACEMENT_POLICY : string      := "LRU";
4          CACHE_LINES        : positive    := 32;
5          ASSOCIATIVITY      : positive    := 32;
6          TAG_BITS           : positive    := 128;
7          CHUNK_BITS         : positive    := 8;
8          TAG_BYTE_ORDER     : T_BYTE_ORDER := LITTLE_ENDIAN;
9          USE_INITIAL_TAGS   : boolean     := false;
10         INITIAL_TAGS       : T_SLM       := (0 downto 0 => (127 downto 0 => '0'))
11     );
12     port (
13         Clock : in std_logic;
14         Reset : in std_logic;
15
16         Replace      : in std_logic;
17         Replaced     : out std_logic;
18         Replace_NewTag_rst : out std_logic;
19         Replace_NewTag_rev : out std_logic;
20         Replace_NewTag_nxt : out std_logic;
21         Replace_NewTag_Data : in std_logic_vector(CHUNK_BITS - 1 downto 0);
22         Replace_NewIndex  : out std_logic_vector(log2ceilnz(CACHE_LINES) - 1 downto _
↪0);
23
24         Request      : in std_logic;
25         Request_ReadWrite : in std_logic;

```

(continues on next page)

(continued from previous page)

```

26     Request_Invalidate : in  std_logic;
27     Request_Tag_rst    : out std_logic;
28     Request_Tag_rev    : out std_logic;
29     Request_Tag_nxt    : out std_logic;
30     Request_Tag_Data   : in  std_logic_vector(CHUNK_BITS - 1 downto 0);
31     Request_Index      : out std_logic_vector(log2ceilnz(CACHE_LINES) - 1 downto
↳ 0);
32     Request_TagHit     : out std_logic;
33     Request_TagMiss    : out std_logic
34 );
35 end entity;

```

Source file: `cache/cache_tagunit_seq.vhdl`

7.7 PoC.comm

These are communication entities. ...

7.7.1 PoC.comm Package

Source file: `comm.pkg.vhdl`

7.7.2 PoC.comm.crc

Computes the Cyclic Redundancy Check (CRC) for a data packet as remainder of the polynomial division of the message by the given generator polynomial (GEN).

The computation is unrolled so as to process an arbitrary number of message bits per step. The generated CRC is independent from the chosen processing width.

Entity Declaration:

```

1  entity comm_crc is
2      generic (
3          GEN      : bit_vector;           -- Generator Polynomial
4          BITS     : positive;             -- Number of Bits to be
↳ processed in parallel
5
6          STARTUP_RMD : std_logic_vector := "0";
7          OUTPUT_REGS : boolean          := true
8      );
9      port (
10         clk : in  std_logic;              -- Clock
11
12         set : in  std_logic;              -- Parallel Preload of
↳ Remainder
13         init : in std_logic_vector(abs(mssb_idx(GEN)-GEN'right)-1 downto 0); --
14         step : in std_logic;              -- Process Input Data (MSB
↳ first)
15         din : in  std_logic_vector(BITS-1 downto 0); --
16
17         rmd : out std_logic_vector(abs(mssb_idx(GEN)-GEN'right)-1 downto 0); --
↳ Remainder
18         zero : out std_logic              -- Remainder is Zero

```

(continues on next page)

(continued from previous page)

```
19 );  
20 end entity comm_crc;
```

Source file: `comm/comm_crc.vhdl`

7.7.3 PoC.comm.scramble

The LFSR computation is unrolled to generate an arbitrary number of mask bits in parallel. The mask are output in little endian. The generated bit sequence is independent from the chosen output width.

Entity Declaration:

```
1  entity comm_scramble is  
2      generic (  
3          GEN    : bit_vector;      -- Generator Polynomial (little endian)  
4          BITS   : positive         -- Width of Mask Bits to be computed in parallel in_  
--each step  
5      );  
6      port (  
7          clk    : in  std_logic;    -- Clock  
8  
9          set    : in  std_logic;     -- Set LFSR to value provided on din  
10         din    : in  std_logic_vector(GEN'length-2 downto 0) := (others => '0');  
11  
12         step   : in  std_logic;     -- Compute a Mask Output  
13         mask   : out std_logic_vector(BITS-1 downto 0)  
14     );  
15 end entity comm_scramble;
```

Source file: `comm/comm_scramble.vhdl`

7.8 PoC.dstruct

The namespace *PoC.dstruct* offers different data structure implementations.

Package

The package *PoC.dstruct* holds all component declarations for this namespace.

Entities

- *PoC.dstruct.deque* implements a deque (two-sided FIFO).
- *PoC.dstruct.stack* implements a regular stack.

7.8.1 PoC.dstruct.deque

Implements a deque (double-ended queue). This data structure allows two acting entities to queue data elements for the consumption by the other while still being able to unqueue untaken ones in LIFO fashion.

Entity Declaration:

```

1 entity dstruct_deque is
2   generic (
3     D_BITS      : positive;           -- Data Width
4     MIN_DEPTH   : positive           -- Minimum Deque Depth
5   );
6   port (
7     -- Shared Ports
8     clk, rst : in std_logic;
9
10    -- Port A
11    dinA      : in  std_logic_vector(D_BITS-1 downto 0); -- DataA Input
12    putA      : in  std_logic;
13    gotA      : in  std_logic;
14    doutA     : out std_logic_vector(D_BITS-1 downto 0); -- DataA Output
15    validA    : out std_logic;
16    fullA     : out std_logic;
17
18    -- Port B
19    dinB      : in  std_logic_vector(D_BITS-1 downto 0); -- DataB Input
20    putB      : in  std_logic;
21    gotB      : in  std_logic;
22    doutB     : out std_logic_vector(D_BITS-1 downto 0);
23    validB    : out std_logic;
24    fullB     : out std_logic
25  );
26 end entity dstruct_deque;

```

Source file: dstruct/dstruct_deque.vhdl

7.8.2 PoC.dstruct.stack

Implements a stack, a LIFO storage abstraction.

Entity Declaration:

```

1 entity dstruct_stack is
2   generic (
3     D_BITS      : positive;           -- Data Width
4     MIN_DEPTH   : positive           -- Minimum Stack Depth
5   );
6   port (
7     -- INPUTS
8     clk, rst : in std_logic;
9
10    -- Write Ports
11    din      : in  std_logic_vector(D_BITS-1 downto 0); -- Data Input
12    put      : in  std_logic; -- 0 -> pop, 1 -> push
13    full     : out std_logic;
14
15    -- Read Ports
16    got      : in  std_logic;
17    dout     : out std_logic_vector(D_BITS-1 downto 0);
18    valid    : out std_logic
19  );
20 end entity dstruct_stack;

```

Source file: dstruct/dstruct_stack.vhdl

7.9 PoC.fifo

The namespace *PoC.fifo* offers different FIFO (first-in, first-out) implementations.

Package

The package *PoC.fifo* holds all component declarations for this namespace.

Entities

PoC offers FIFOs with a *got*-interface. This means, the current read-pointer value is available on the output. Asserting the *got*-input, acknowledge the processing of the current output signals and moves the read-pointer to the next value, if available.

All FIFOs implement a bidirectional flow control (*put/full* and *valid/got*). Each FIFO also offers a *EmptyState* (write-side) and *FullState* (read-side) to indicate the current fill-state.

The prefixes *cc_* (common clock), *dc_* (dependent clock) and *ic_* (independent clock) refer to the write- and read-side clock relationship.

- *PoC.fifo.cc_got* implements a regular FIFO (one common clock, got-interface)
- *PoC.fifo.cc_got_tempgot* implements a regular FIFO (one common clock, got-interface), extended by a transactional *tempgot*-interface (read-side).
- *PoC.fifo.cc_got_tempput* implements a regular FIFO (one common clock, got-interface), extended by a transactional *tempput*-interface (write-side).
- *IP:fifo_dc_got* implements a cross-clock FIFO (two related clocks, got-interface)
- *PoC.fifo.ic_got* implements a cross-clock FIFO (two independent clocks, got-interface)
- *PoC.fifo.glue* implements a two-stage FIFO (one common clock, got-interface)
- *PoC.fifo.shift* implements a regular FIFO (one common clock, got-interface, optimized for FPGAs with shifter primitives)

7.9.1 PoC.fifo Package

This package holds all component declarations for this namespace.

Source file: [fifo.pkg.vhdl](#)

7.9.2 PoC.fifo.cc_got

This module implements a regular FIFO with common clock (*cc*), pipelined interface. Common clock means read and write port use the same clock. The FIFO size can be configured in word width (*D_BITS*) and minimum word count *MIN_DEPTH*. The specified depth is rounded up to the next suitable value.

DATA_REG (=true) is a hint, that distributed memory or registers should be used as data storage. The actual memory type depends on the device architecture. See implementation for details.

STATE_*_BITS* defines the granularity of the fill state indicator **state_. If a fill state is not of interest, set **STATE_*_BITS* = 0. *fstate_rd* is associated with the read clock domain and outputs the guaranteed number of words available in the FIFO. *estate_wr* is associated with the write clock domain and outputs the number of words that is guaranteed to be accepted by the FIFO without a capacity overflow. Note that both these indicators cannot replace the *full* or *valid* outputs as they may be implemented as giving pessimistic bounds that are minimally off the true fill state.

fstate_rd and *estate_wr* are combinatorial outputs and include an address comparator (subtractor) in their path.

Examples:

- FSTATE_RD_BITS = 1:

fstate_rd	filled (at least)
0	0/2 full
1	1/2 full (half full)

- FSTATE_RD_BITS = 2:

fstate_rd	filled (at least)
0	0/4 full
1	1/4 full
2	2/4 full (half full)
3	3/4 full

Entity Declaration:

```

1 entity fifo_cc_got is
2   generic (
3     D_BITS          : positive;           -- Data Width
4     MIN_DEPTH       : positive;           -- Minimum FIFO Depth
5     DATA_REG       : boolean := false;   -- Store Data Content in Registers
6     STATE_REG       : boolean := false;   -- Registered Full/Empty Indicators
7     OUTPUT_REG      : boolean := false;   -- Registered FIFO Output
8     ESTATE_WR_BITS  : natural := 0;       -- Empty State Bits
9     FSTATE_RD_BITS  : natural := 0;       -- Full State Bits
10  );
11  port (
12    -- Global Reset and Clock
13    rst, clk : in std_logic;
14
15    -- Writing Interface
16    put       : in std_logic;               -- Write Request
17    din       : in std_logic_vector(D_BITS-1 downto 0); -- Input Data
18    full      : out std_logic;
19    estate_wr : out std_logic_vector(imax(0, ESTATE_WR_BITS-1) downto 0);
20
21    -- Reading Interface
22    got       : in std_logic;               -- Read Completed
23    dout      : out std_logic_vector(D_BITS-1 downto 0); -- Output Data
24    valid     : out std_logic;
25    fstate_rd : out std_logic_vector(imax(0, FSTATE_RD_BITS-1) downto 0)
26  );
27 end entity fifo_cc_got;
```

See also:

IP:[fifo_dc_got](#) For a FIFO with dependent clocks.

[PoC.fifo.ic_got](#) For a FIFO with independent clocks (cross-clock FIFO).

[PoC.fifo.glue](#) For a minimal FIFO / pipeline decoupling.

Source file: [fifo/fifo_cc_got.vhdl](#)

7.9.3 PoC.fifo.cc_got_tempgot

The specified depth (`MIN_DEPTH`) is rounded up to the next suitable value.

As uncommitted reads occupy FIFO space that is not yet available for writing, an instance of this FIFO can, indeed, report full and not vld at the same time. While a commit would eventually make space available for writing (not ful), a rollback would re-iterate data for reading (vld).

commit and rollback are inclusive and apply to all reads (got) since the previous commit or rollback up to and including a potentially simultaneous read.

The FIFO state upon a simultaneous assertion of commit and rollback is *undefined*!

*STATE_*_BITS defines the granularity of the fill state indicator *state_*. fstate_rd is associated with the read clock domain and outputs the guaranteed number of words available in the FIFO. estate_wr is associated with the write clock domain and outputs the number of words that is guaranteed to be accepted by the FIFO without a capacity overflow. Note that both these indicators cannot replace the full or valid outputs as they may be implemented as giving pessimistic bounds that are minimally off the true fill state.

If a fill state is not of interest, set *STATE_*_BITS = 0.

fstate_rd and estate_wr are combinatorial outputs and include an address comparator (subtractor) in their path.

Examples:

- FSTATE_RD_BITS = 1:
 - fstate_rd == 0 => 0/2 full
 - fstate_rd == 1 => 1/2 full (half full)
- FSTATE_RD_BITS = 2:
 - fstate_rd == 0 => 0/4 full
 - fstate_rd == 1 => 1/4 full
 - fstate_rd == 2 => 2/4 full
 - fstate_rd == 3 => 3/4 full

Entity Declaration:

```

1  entity fifo_cc_got_tempgot is
2      generic (
3          D_BITS          : positive;          -- Data Width
4          MIN_DEPTH       : positive;          -- Minimum FIFO Depth
5          DATA_REG       : boolean := false;   -- Store Data Content in Registers
6          STATE_REG       : boolean := false;   -- Registered Full/Empty Indicators
7          OUTPUT_REG      : boolean := false;   -- Registered FIFO Output
8          ESTATE_WR_BITS  : natural := 0;       -- Empty State Bits
9          FSTATE_RD_BITS  : natural := 0;       -- Full State Bits
10     );
11     port (
12         -- Global Reset and Clock
13         rst, clk : in  std_logic;
14
15         -- Writing Interface
16         put      : in  std_logic;              -- Write Request
17         din      : in  std_logic_vector(D_BITS-1 downto 0); -- Input Data
18         full     : out std_logic;
19         estate_wr : out std_logic_vector(imax(0, ESTATE_WR_BITS-1) downto 0);
20
21         -- Reading Interface
22         got      : in  std_logic;              -- Read Completed
23         dout     : out std_logic_vector(D_BITS-1 downto 0); -- Output Data
24         valid    : out std_logic;
25         fstate_rd : out std_logic_vector(imax(0, FSTATE_RD_BITS-1) downto 0);

```

(continues on next page)

(continued from previous page)

```

26
27     commit      : in std_logic;
28     rollback    : in std_logic
29 );
30 end entity fifo_cc_got_tempgot;

```

Source file: `fifo/fifo_cc_got_tempgot.vhdl`

7.9.4 PoC.fifo.cc_got_tempput

The specified depth (`MIN_DEPTH`) is rounded up to the next suitable value.

As uncommitted writes populate FIFO space that is not yet available for reading, an instance of this FIFO can, indeed, report `full` and `not vld` at the same time. While a `commit` would eventually make data available for reading (`vld`), a `rollback` would free the space for subsequent writing (`not ful`).

`commit` and `rollback` are inclusive and apply to all writes (`put`) since the previous ‘commit’ or ‘rollback’ up to and including a potentially simultaneous write.

The FIFO state upon a simultaneous assertion of `commit` and `rollback` is *undefined*.

`*STATE*_BITS` defines the granularity of the fill state indicator `*state_*`. `fstate_rd` is associated with the read clock domain and outputs the guaranteed number of words available in the FIFO. `estate_wr` is associated with the write clock domain and outputs the number of words that is guaranteed to be accepted by the FIFO without a capacity overflow. Note that both these indicators cannot replace the `full` or `valid` outputs as they may be implemented as giving pessimistic bounds that are minimally off the true fill state.

If a fill state is not of interest, set `*STATE*_BITS = 0`.

`fstate_rd` and `estate_wr` are combinatorial outputs and include an address comparator (subtractor) in their path.

Examples:

- `FSTATE_RD_BITS = 1`:
 - `fstate_rd == 0` => 0/2 full
 - `fstate_rd == 1` => 1/2 full (half full)
- `FSTATE_RD_BITS = 2`:
 - `fstate_rd == 0` => 0/4 full
 - `fstate_rd == 1` => 1/4 full
 - `fstate_rd == 2` => 2/4 full
 - `fstate_rd == 3` => 3/4 full

Entity Declaration:

```

1  entity fifo_cc_got_tempput is
2      generic (
3          D_BITS          : positive;           -- Data Width
4          MIN_DEPTH       : positive;           -- Minimum FIFO Depth
5          DATA_REG       : boolean := false;   -- Store Data Content in Registers
6          STATE_REG       : boolean := false;   -- Registered Full/Empty Indicators
7          OUTPUT_REG      : boolean := false;   -- Registered FIFO Output
8          ESTATE_WR_BITS  : natural := 0;       -- Empty State Bits
9          FSTATE_RD_BITS  : natural := 0;       -- Full State Bits
10 );
11 port (

```

(continues on next page)

(continued from previous page)

```

12  -- Global Reset and Clock
13  rst, clk : in  std_logic;
14
15  -- Writing Interface
16  put      : in  std_logic;           -- Write Request
17  din      : in  std_logic_vector(D_BITS-1 downto 0); -- Input Data
18  full     : out std_logic;
19  estate_wr : out std_logic_vector(imax(0, ESTATE_WR_BITS-1) downto 0);
20
21  commit   : in  std_logic;
22  rollback : in  std_logic;
23
24  -- Reading Interface
25  got      : in  std_logic;           -- Read Completed
26  dout     : out std_logic_vector(D_BITS-1 downto 0); -- Output Data
27  valid    : out std_logic;
28  fstate_rd : out std_logic_vector(imax(0, FSTATE_RD_BITS-1) downto 0)
29  );
30  end entity fifo_cc_got_tempput;

```

Source file: `fifo/fifo_cc_got_tempput.vhdl`

7.9.5 PoC.fifo.glue

Its primary use is the decoupling of enable domains in a processing pipeline. Data storage is limited to two words only so as to allow both the `ful` and the `vld` indicators to be driven by registers.

Entity Declaration:

```

1  entity fifo_glue is
2    generic (
3      D_BITS : positive           -- Data Width
4    );
5    port (
6      -- Control
7      clk : in  std_logic;        -- Clock
8      rst : in  std_logic;        -- Synchronous Reset
9
10     -- Input
11     put : in  std_logic;         -- Put Value
12     di  : in  std_logic_vector(D_BITS-1 downto 0); -- Data Input
13     ful : out std_logic;         -- Full
14
15     -- Output
16     vld : out std_logic;         -- Data Available
17     do  : out std_logic_vector(D_BITS-1 downto 0); -- Data Output
18     got : in  std_logic         -- Data Consumed
19   );
20  end entity fifo_glue;

```

Source file: `fifo/fifo_glue.vhdl`

7.9.6 PoC.fifo.ic_assembly

This module assembles a FIFO stream from data blocks that may arrive slightly out of order. The arriving data is ordered according to their address. The streamed output starts with the data word written to address zero (0) and may proceed all the way to just before the first yet missing data. The association of data with addresses is used on

the input side for the sole purpose of reconstructing the correct order of the data. It is assumed to wrap so as to allow an infinite input sequence. Addresses are not actively exposed to the purely stream-based FIFO output.

The implemented functionality enables the reconstruction of streams that are tunnelled across address-based transports that are allowed to reorder the transmission of data blocks. This applies to many DMA implementations.

Entity Declaration:

```

1  entity fifo_ic_assembly is
2      generic (
3          D_BITS : positive;           -- Data Width
4          A_BITS : positive;           -- Address Bits
5          G_BITS : positive            -- Generation Guard Bits
6      );
7      port (
8          -- Write Interface
9          clk_wr : in std_logic;
10         rst_wr : in std_logic;
11
12         -- Only write addresses in the range [base, base+2**(A_BITS-G_BITS)) are
13         -- acceptable. This is equivalent to the test
14         -- tmp(A_BITS-1 downto A_BITS-G_BITS) = 0 where tmp = addr - base.
15         -- Writes performed outside the allowable range will assert the failure
16         -- indicator, which will stick until the next reset.
17         -- No write is to be performed before base turns zero (0) for the first
18         -- time.
19         base : out std_logic_vector(A_BITS-1 downto 0);
20         failed : out std_logic;
21
22         addr : in std_logic_vector(A_BITS-1 downto 0);
23         din : in std_logic_vector(D_BITS-1 downto 0);
24         put : in std_logic;
25
26         -- Read Interface
27         clk_rd : in std_logic;
28         rst_rd : in std_logic;
29
30         dout : out std_logic_vector(D_BITS-1 downto 0);
31         vld : out std_logic;
32         got : in std_logic
33     );
34 end entity fifo_ic_assembly;

```

Source file: `fifo/fifo_ic_assembly.vhdl`

7.9.7 PoC.fifo.ic_got

Independent clocks means that read and write clock are unrelated.

This implementation uses dedicated block RAM for storing data.

First-word-fall-through (FWFT) mode is implemented, so data can be read out as soon as `valid` goes high. After the data has been captured, then the signal `got` must be asserted.

Synchronous reset is used. Both resets may overlap.

`DATA_REG` (`=true`) is a hint, that distributed memory or registers should be used as data storage. The actual memory type depends on the device architecture. See implementation for details.

`*STATE_*_BITS` defines the granularity of the fill state indicator `*state_*`. `fstate_rd` is associated with the read clock domain and outputs the guaranteed number of words available in the FIFO. `estate_wr` is associated with the write clock domain and outputs the number of words that is guaranteed to be accepted by the FIFO

without a capacity overflow. Note that both these indicators cannot replace the `full` or `valid` outputs as they may be implemented as giving pessimistic bounds that are minimally off the true fill state.

If a fill state is not of interest, set `*STATE_*_BITS = 0`.

`fstate_rd` and `estate_wr` are combinatorial outputs and include an address comparator (subtractor) in their path.

Examples: - `FSTATE_RD_BITS = 1`: `fstate_rd == 0` => 0/2 full

`fstate_rd == 1` => 1/2 full (half full)

- **`FSTATE_RD_BITS = 2`**: `fstate_rd == 0` => 0/4 full `fstate_rd == 1` => 1/4 full `fstate_rd == 2` => 2/4 full `fstate_rd == 3` => 3/4 full

Entity Declaration:

```
1 entity fifo_ic_got is
2   generic (
3     D_BITS          : positive;           -- Data Width
4     MIN_DEPTH       : positive;           -- Minimum FIFO Depth
5     DATA_REG       : boolean := false;   -- Store Data Content in Registers
6     OUTPUT_REG      : boolean := false;   -- Registered FIFO Output
7     ESTATE_WR_BITS  : natural := 0;       -- Empty State Bits
8     FSTATE_RD_BITS  : natural := 0;       -- Full State Bits
9   );
10  port (
11    -- Write Interface
12    clk_wr   : in  std_logic;
13    rst_wr   : in  std_logic;
14    put      : in  std_logic;
15    din      : in  std_logic_vector(D_BITS-1 downto 0);
16    full     : out std_logic;
17    estate_wr : out std_logic_vector(imax(ESTATE_WR_BITS-1, 0) downto 0);
18
19    -- Read Interface
20    clk_rd   : in  std_logic;
21    rst_rd   : in  std_logic;
22    got      : in  std_logic;
23    valid    : out std_logic;
24    dout     : out std_logic_vector(D_BITS-1 downto 0);
25    fstate_rd : out std_logic_vector(imax(FSTATE_RD_BITS-1, 0) downto 0)
26  );
27 end entity fifo_ic_got;
```

Source file: `fifo/fifo_ic_got.vhdl`

7.9.8 PoC.fifo.shift

This FIFO implementation is based on an internal shift register. This is especially useful for smaller FIFO sizes, which can be implemented in LUT storage on some devices (e.g. Xilinx' SRLs). Only a single read pointer is maintained, which determines the number of valid entries within the underlying shift register.

The specified depth (`MIN_DEPTH`) is rounded up to the next suitable value.

Entity Declaration:

```
1 entity fifo_shift is
2   generic (
```

(continues on next page)

(continued from previous page)

```

3      D_BITS      : positive;           -- Data Width
4      MIN_DEPTH   : positive           -- Minimum FIFO Size in Words
5  );
6  port (
7      -- Global Control
8      clk : in std_logic;
9      rst : in std_logic;
10
11     -- Writing Interface
12     put : in std_logic;                 -- Write Request
13     din : in std_logic_vector(D_BITS-1 downto 0); -- Input Data
14     ful : out std_logic;               -- Capacity Exhausted
15
16     -- Reading Interface
17     got : in std_logic;                 -- Read Done Strobe
18     dout : out std_logic_vector(D_BITS-1 downto 0); -- Output Data
19     vld : out std_logic               -- Data Valid
20 );
21 end entity fifo_shift;

```

Source file: fifo/fifo_shift.vhdl

7.10 PoC.io

The namespace `PoC.io` offers different general purpose I/O (GPIO) implementations, as well as low-speed bus protocol controllers.

Sub-namespaces

- *PoC.io.ddrio* - Double-Data-Rate (DDR) input/output abstraction layer.
- *PoC.io.iic* - I²C bus controllers
- *PoC.io.jtag* - JTAG implementations
- *PoC.io.lcd* - LC-Display bus controllers
- *PoC.io.mdio* - Management Data I/O (MDIO) controllers for Ethernet PHYs
- *PoC.io.ow* - OneWire / iButton bus controllers
- *PoC.io.ps2* - Periphery bus of the Personal System/2 (PS/2)
- *PoC.io.uart* - Universal Asynchronous Receiver Transmitter (UART) controllers
- *PoC.io.vga* - VGA, DVI, HDMI controllers

Package

The package *PoC.io* holds all enum, function and component declarations for this namespace.

Entities

- *PoC.io.Debounce*
- *PoC.io.7SegmentMux_BCD*
- *PoC.io.7SegmentMux_HEX*
- *PoC.io.FanControl*
- *PoC.io.FrequencyCounter*
- *PoC.io.GlitchFilter*
- *PoC.io.PulseWidthModulation*

- *PoC.io.TimingCounter*

7.10.1 PoC.io.ddrio

These are DDR-I/O (Double Data Rate - Input/Output) entities. ...

Entities

- *PoC.io.ddrio.in*
- *PoC.io.ddrio.inout*
- *PoC.io.ddrio.out*

PoC.io.ddrio Package

Source file: `ddrio.pkg.vhdl`

PoC.io.ddrio.in

Instantiates chip-specific DDR (Double Data Rate) input registers.

Both data `DataIn_high/low` are synchronously outputted to the on-chip logic with the rising edge of `Clock`. `DataIn_high` is the value at the Pad sampled with the same rising edge. `DataIn_low` is the value sampled with the falling edge directly before this rising edge. Thus sampling starts with the falling edge of the clock as depicted in the following waveform.

After power-up, the output ports `DataIn_high` and `DataIn_low` both equal `INIT_VALUE`.

Pad must be connected to a PAD because FPGAs only have these registers in IOBs.

Entity Declaration:

```
1 entity ddrio_in is
2   generic (
3     BITS          : positive;
4     INIT_VALUE    : bit_vector := x"FFFFFFF"
5   );
6   port (
7     Clock          : in    std_logic;
8     ClockEnable    : in    std_logic;
9     DataIn_high    : out   std_logic_vector (BITS - 1 downto 0);
10    DataIn_low     : out   std_logic_vector (BITS - 1 downto 0);
11    Pad            : in    std_logic_vector (BITS - 1 downto 0)
12  );
13 end entity;
```

Source file: `io/ddrio/ddrio_in.vhdl`

PoC.io.ddrio.inout

Instantiates chip-specific DDR input and output registers.

Both data `DataOut_high/low` as well as `OutputEnable` are sampled with the `rising_edge(Clock)` from the on-chip logic. `DataOut_high` is brought out with this rising edge. `DataOut_low` is brought out with the falling edge.

`OutputEnable` (Tri-State) is high-active. It is automatically inverted if necessary. Output is disabled after power-up.

Both data `DataIn_high/low` are synchronously outputted to the on-chip logic with the rising edge of `Clock`. `DataIn_high` is the value at the Pad sampled with the same rising edge. `DataIn_low` is the value sampled with the falling edge directly before this rising edge. Thus sampling starts with the falling edge of the clock as depicted in the following waveform.

Pad must be connected to a PAD because FPGAs only have these registers in IOBs.

Entity Declaration:

```

1 entity ddrio_inout is
2   generic (
3     BITS          : positive
4   );
5   port (
6     ClockOut       : in    std_logic;
7     ClockOutEnable : in    std_logic;
8     OutputEnable   : in    std_logic;
9     DataOut_high   : in    std_logic_vector(BITS - 1 downto 0);
10    DataOut_low    : in    std_logic_vector(BITS - 1 downto 0);
11
12    ClockIn        : in    std_logic;
13    ClockInEnable  : in    std_logic;
14    DataIn_high    : out   std_logic_vector(BITS - 1 downto 0);
15    DataIn_low     : out   std_logic_vector(BITS - 1 downto 0);
16
17    Pad            : inout std_logic_vector(BITS - 1 downto 0)
18  );
19 end entity;
```

Source file: `io/ddrio/ddrio_inout.vhdl`

PoC.io.ddrio.out

Instantiates chip-specific DDR output registers.

Both data `DataOut_high/low` as well as `OutputEnable` are sampled with the rising edge (`Clock`) from the on-chip logic. `DataOut_high` is brought out with this rising edge. `DataOut_low` is brought out with the falling edge.

`OutputEnable` (Tri-State) is high-active. It is automatically inverted if necessary. If an output enable is not required, you may save some logic by setting `NO_OUTPUT_ENABLE = true`.

If `NO_OUTPUT_ENABLE = false` then output is disabled after power-up. If `NO_OUTPUT_ENABLE = true` then output after power-up equals `INIT_VALUE`.

Pad must be connected to a PAD because FPGAs only have these registers in IOBs.

Entity Declaration:

```

1 entity ddrio_out is
2   generic (
3     NO_OUTPUT_ENABLE : boolean := false;
4     BITS              : positive;
5     INIT_VALUE        : bit_vector := x"FFFFFFFF"
6   );
7   port (
8     Clock       : in    std_logic;
9     ClockEnable : in    std_logic := '1';
10    OutputEnable : in    std_logic := '1';
```

(continues on next page)

(continued from previous page)

```
11     DataOut_high   : in std_logic_vector (BITS - 1 downto 0);
12     DataOut_low    : in std_logic_vector (BITS - 1 downto 0);
13     Pad            : out std_logic_vector (BITS - 1 downto 0)
14 );
15 end entity;
```

Source file: [io/ddrio/ddrio_out.vhdl](#)

7.10.2 PoC.io.iic

These are I2C entities. . . .

PoC.io.iic Package

Source file: [iic.pkg.vhdl](#)

PoC.io.iic.BusController

The I2C BusController transmits bits over the I2C bus (SerialClock - SCL, SerialData - SDA) and also receives them. To send/receive words over the I2C bus, use the I2C Controller, which utilizes this controller. This controller is compatible to the System Management Bus (SMBus).

Entity Declaration:

Source file: [io/iic/iic_BusController.vhdl](#)

PoC.io.iic.Controller

The I2C Controller transmits words over the I2C bus (SerialClock - SCL, SerialData - SDA) and also receives them. This controller utilizes the I2C BusController to send/receive bits over the I2C bus. This controller is compatible to the System Management Bus (SMBus).

Entity Declaration:

Source file: [io/iic/iic_Controller.vhdl](#)

PoC.io.iic.Switch_PCA9548A

Todo: No documentation available. TODO

Entity Declaration:

Source file: [io/iic/iic_Switch_PCA9548A.vhdl](#)

7.10.3 PoC.io.jtag

These are JTAG entities. . . .

7.10.4 PoC.io.lcd

These are LCD entities. . .

PoC.io.lcd.Package

Source file: [lcd.pkg.vhdl](#)

PoC.io.lcd.LCDBuffer

Todo: No documentation available.

Entity Declaration:

Source file: [io/lcd/lcd_LCDBuffer.vhdl](#)

PoC.io.lcd.LCDBusController

Todo: No documentation available.

Entity Declaration:

Source file: [io/lcd/lcd_LCDBusController.vhdl](#)

PoC.io.lcd.LCDController_KS0066U

Todo: No documentation available.

Entity Declaration:

Source file: [io/lcd/lcd_LCDController_KS0066U.vhdl](#)

PoC.io.lcd.LCDSynchronizer

Todo: No documentation available.

Entity Declaration:

Source file: [io/lcd/lcd_LCDSynchronizer.vhdl](#)

7.10.5 PoC.io.mdio

These are MDIO entities. . . .

mdio_BusController

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

PoC.io.mdio.Controller

Todo: No documentation available.

Entity Declaration:

Source file: [io/mdio/mdio_Controller.vhdl](#)

PoC.io.mdio.IIC_Adapter

Todo: No documentation available.

Entity Declaration:

Source file: [io/mdio/mdio_IIC_Adapter.vhdl](#)

7.10.6 PoC.io.ow

These are OneWire entities. . . .

ow_BusController

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

ow_Controller

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.10.7 PoC.io.pio

These are Pmod entities. . . .

PoC.io.pio.in

Entity Declaration:

Source file: [io/pio/pio_in.vhdl](#)

PoC.io.pio.out

Entity Declaration:

Source file: [io/pio/pio_out.vhdl](#)

PoC.io.pio.fifo_in

Entity Declaration:

Source file: [io/pio/pio_fifo_in.vhdl](#)

PoC.io.pio.fifo_out

Entity Declaration:

Source file: [io/pio/pio_fifo_out.vhdl](#)

7.10.8 PoC.io.pmod

These are Pmod entities. . . .

Entities

- *PoC.io.pmod.KYPD*
- *PoC.io.pmod.SSD*
- *PoC.io.pmod.USBUART*

PoC.io.pmod Package

Source file: [pmod.pkg.vhdl](#)

PoC.io.pmod.KYPD

This module drives a 4-bit one-cold encoded column vector to read back a 4-bit rows vector. By scanning column-by-column it's possible to extract the current button state of the whole keypad. This wrapper converts the high-active signals from *PoC.io.KeypadScanner* to low-active signals for the pmod. An additional debounce circuit filters the button signals. The scan frequency and bounce time can be configured.

Entity Declaration:

```
1 entity pmod_KYPD is
2   generic (
3     CLOCK_FREQ      : FREQ      := 100 MHz;
4     SCAN_FREQ       : FREQ      := 1 kHz;
5     BOUNCE_TIME     : time      := 10 ms
6   );
7   port (
8     Clock           : in  std_logic;
9     Reset           : in  std_logic;
10    -- Matrix interface
11    Keys             : out T_PMOD_KYPD_KEYPAD;
12    -- KeyPad interface
13    Columns_n        : out std_logic_vector(3 downto 0);
14    Rows_n           : in  std_logic_vector(3 downto 0)
15  );
16 end entity;
```

Source file: `io/pmod/pmod_KYPD.vhdl`

PoC.io.pmod.SSD

This module drives a dual-digit 7-segment display (Pmod_SSD). The module expects two binary encoded 4-bit Digit<i> signals and drives a 2x6 bit Pmod connector (7 anode bits, 1 cathode bit).

Segment	Pos./	Index
AAA		000
F B		5 1
F B		5 1
GGG		666
E C		4 2
E C		4 2
DDD DOT		333 7

Entity Declaration:

```
1 entity pmod_SSD is
2   generic (
3     CLOCK_FREQ      : FREQ      := 100 MHz;
4     REFRESH_RATE    : FREQ      := 1 kHz
5   );
6   port (
7     Clock           : in  std_logic;
8
9     Digit0          : in  std_logic_vector(3 downto 0);
10    Digit1          : in  std_logic_vector(3 downto 0);
11
12    SSD              : out T_PMOD_SSD_PINS
```

(continues on next page)

(continued from previous page)

```

13   );
14   end entity;

```

Source file: `io/pmod/pmod_SSD.vhdl`

PoC.io.pmod.USBUART

This module abstracts a FTDI FT232R USB-UART bridge by instantiating a *PoC.io.uart.fifo*. The FT232R supports up to 3 MBaud. A synchronous FIFO interface with a 32 words buffer is provided. Hardware flow control (RTS_CTS) is enabled.

Entity Declaration:

```

1  entity pmod_USBUART is
2      generic (
3          CLOCK_FREQ      : FREQ      := 100 MHz;
4          BAUDRATE        : BAUD      := 115200 Bd
5      );
6      port (
7          Clock           : in  std_logic;
8          Reset           : in  std_logic;
9
10         TX_put          : in  std_logic;
11         TX_Data         : in  std_logic_vector(7 downto 0);
12         TX_Full         : out std_logic;
13
14         RX_Valid        : out std_logic;
15         RX_Data         : out std_logic_vector(7 downto 0);
16         RX_got          : in  std_logic;
17
18         UART_TX         : out std_logic;
19         UART_RX         : in  std_logic;
20         UART_RTS        : out std_logic;
21         UART_CTS        : in  std_logic
22     );
23   end entity;

```

Source file: `io/pmod/pmod_USBUART.vhdl`

7.10.9 PoC.io.ps2

These are PS/2 entities....

7.10.10 PoC.io.uart

These are UART (Universal Asynchronous Receiver Transmitter) entities....

Entities

- *PoC.io.uart.bclk*
- *PoC.io.uart.rx*
- *PoC.io.uart.tx*
- *PoC.io.uart.fifo*

PoC.io.uart Package

Source file: `uart.pkg.vhdl`

PoC.io.uart.bclk

Todo: No documentation available.

old comments: UART BAUD rate generator `bclk_r` = bit clock is rising `bclk_x8_r` = bit clock times 8 is rising

Entity Declaration:

```
1 entity uart_bclk is
2   generic (
3     CLOCK_FREQ      : FREQ      := 100 MHz;
4     BAUDRATE        : BAUD      := 115200 Bd
5   );
6   port (
7     clk             : in  std_logic;
8     rst             : in  std_logic;
9     bclk            : out std_logic;
10    bclk_x8          : out std_logic
11  );
12 end entity;
```

Source file: `io/uart/uart_bclk.vhdl`

PoC.io.uart.rx

UART Receiver: 1 Start + 8 Data + 1 Stop

Entity Declaration:

```
1 entity uart_rx is
2   generic (
3     SYNC_DEPTH : natural := 2  -- use zero for already clock-synchronous rx
4   );
5   port (
6     -- Global Control
7     clk : in  std_logic;
8     rst : in  std_logic;
9
10    -- Bit Clock and RX Line
11    bclk_x8 : in  std_logic;  -- bit clock, eight strobes per bit length
12    rx      : in  std_logic;
13
14    -- Byte Stream Output
15    do : out std_logic_vector(7 downto 0);
16    stb : out std_logic
17  );
18 end entity;
```

Source file: `io/uart/uart_rx.vhdl`

PoC.io.uart.tx

UART Transmitter: 1 Start + 8 Data + 1 Stop

Entity Declaration:

```

1 entity uart_tx is
2   port (
3     -- Global Control
4     clk : in std_logic;
5     rst : in std_logic;
6
7     -- Bit Clock and TX Line
8     bclk : in std_logic; -- bit clock, one strobe each bit length
9     tx   : out std_logic;
10
11    -- Byte Stream Input
12    di : in std_logic_vector(7 downto 0);
13    put : in std_logic;
14    ful : out std_logic
15  );
16 end entity;
```

Source file: io/uart/uart_tx.vhdl

PoC.io.uart.fifo

Small FIFO s are included in this module, if larger or asynchronous transmit / receive FIFOs are required, then they must be connected externally.

old comments: UART BAUD rate generator bclk = bit clock is rising bclk_x8 = bit clock times 8 is rising

Entity Declaration:

```

1 entity uart_fifo is
2   generic (
3     -- Communication Parameters
4     CLOCK_FREQ      : FREQ;
5     BAUDRATE        : BAUD;
6     ADD_INPUT_SYNCHRONIZERS : boolean := TRUE;
7
8     -- Buffer Dimensioning
9     TX_MIN_DEPTH    : positive := 16;
10    TX_ESTATE_BITS   : natural  := 0;
11    RX_MIN_DEPTH     : positive := 16;
12    RX_FSTATE_BITS   : natural  := 0;
13
14    -- Flow Control
15    FLOWCONTROL       : T_IO_UART_FLOWCONTROL_KIND := UART_FLOWCONTROL_
16    ↪ NONE;
17    SWFC_XON_CHAR     : std_logic_vector(7 downto 0) := x"11"; -- ^Q
18    SWFC_XON_TRIGGER  : real := 0.0625;
19    SWFC_XOFF_CHAR    : std_logic_vector(7 downto 0) := x"13"; -- ^S
20    SWFC_XOFF_TRIGGER : real := 0.75;
21  );
22  port (
23    Clock      : in std_logic;
24    Reset      : in std_logic;
```

(continues on next page)

(continued from previous page)

```
24
25  -- FIFO interface
26  TX_put      : in  std_logic;
27  TX_Data     : in  std_logic_vector(7 downto 0);
28  TX_Full     : out std_logic;
29  TX_EmptyState : out std_logic_vector(imax(0, TX_ESTATE_BITS-1) downto 0);
30
31  RX_Valid    : out std_logic;
32  RX_Data     : out std_logic_vector(7 downto 0);
33  RX_got      : in  std_logic;
34  RX_FullState : out std_logic_vector(imax(0, RX_FSTATE_BITS-1) downto 0);
35  RX_Overflow  : out std_logic;
36
37  -- External pins
38  UART_TX     : out std_logic;
39  UART_RX     : in  std_logic;
40  UART_RTS    : out std_logic;
41  UART_CTS    : in  std_logic
42  );
43  end entity;
```

Source file: `io/uart/uart_fifo.vhdl`

7.10.11 PoC.io.vga

These are VGA entities. . . .

PoC.io.vga Package

Source file: `vga.pkg.vhdl`

PoC.io.vga.phy

Entity Declaration:

Source file: `io/vga/vga_phy.vhdl`

PoC.io.vga.phy_ch7301c

The clock frequency must be the same as used for the timing module, e.g., 25 MHZ for VGA 640x480. A phase-shifted clock must be provided: - clk0 : 0 degrees - clk90 : 90 degrees

pixel_data(23 downto 16) : red pixel_data(15 downto 8) : green pixel_data(7 downto 0) : blue

The reset_b-pin must be driven by other logic (such as the reset button).

The IIC_interface is not part of this modules, as an IIC-master controls several slaves. The following registers must be set, see tests/ml505/vga_test_ml505.vhdl for an example.

Register	Value	Description
0x49 PM	0xC0 0xD0	Enable DVI, RGB bypass off Enable DVI, RGB bypass on
0x33 TPCP	0x08 if clk_freq <= 65 MHz else 0x06	
0x34 TPD	0x16 if clk_freq <= 65 MHz else 0x26	
0x36 TPF	0x60 if clk_freq <= 65 MHz else 0xA0	
0x1F IDF	0x80 0x90	when using SMT (VS0, HS0) when using CVT (VS1, HS0)
0x21 DC	0x09	Enable DAC if RGB bypass is on

Entity Declaration:

Source file: [io/vga/vga_phy_ch7301c.vhdl](#)

PoC.io.vga.timing**Configuration:**

MODE = 0: VGA mode with 640x480 pixels, 60 Hz, frequency(clk) ~ 25 MHz
 MODE = 1: HD 720p with 1280x720 pixels, 60 Hz, frequency(clk) = 74,5 MHz
 MODE = 2: HD 1080p with 1920x1080 pixels, 60 Hz, frequency(clk) = 138,5 MHz

MODE = 2 uses reduced blanking => only suitable for LCDs.

For MODE = 0, CVT can be configured: - CVT = false: Use Safe Mode Timing (SMT).

The legacy fall-back mode supported by CRTs as well as LCDs. HSync: low-active. VSync: low-active. frequency(clk) = 25.175 MHz. (25 MHz works => 31 kHz / 59 Hz)

- **CVT = true: The “new” Coordinated Video Timing (since 2003).** The CVT supports some new features, such as reduced blanking (for LCDs) or aspect ratio encoding. See the web for more details. Standard CRT-based timing (CVT-GTF) has been implemented for best compatibility: HSync: low-active. VSync: high-active. frequency(clk) = 23.75 MHz. (25 MHz works => 31 kHz / 62 Hz)

Usage:

The frequency of `clk` must be equal to the pixel clock frequency of the selected video mode, see also above.

When using analog output, the VGA color signals must be blanked, during horizontal and vertical beam return. This could be achieved by combinatorial “anding” the color value with “beam_on” (part of “phy_ctrl”) inside the PHY.

When using digital output (DVI), then “beam_on” is equal to “DE” (Data Enable) of the DVI transmitter.

`xvalid` and `yvalid` show if `xpos` respectively `ypos` are in a valid range. `beam_on` is ‘1’ iff both `xvalid` and `yvalid` = ‘1’.

`xpos` and `ypos` also show the pixel location during blanking. This might be useful in some applications. But be careful, that the ranges differ between SMT and CVT.

Entity Declaration:

Source file: [io/vga/vga_timing.vhdl](#)

7.10.12 PoC.io Package

This package holds all component declarations for this namespace.

Source file: [io.pkg.vhdl](#)

7.10.13 PoC.io.7SegmentMux_BCD

This module is a 7 segment display controller that uses time multiplexing to control a common anode for each digit in the display. The shown characters are BCD encoded. A dot per digit is optional. A minus sign for negative numbers is supported.

Entity Declaration:

```
1 entity io_7SegmentMux_BCD is
2   generic (
3     CLOCK_FREQ      : FREQ      := 100 MHz;
4     REFRESH_RATE    : FREQ      := 1 kHz;
5     DIGITS           : positive  := 4
6   );
7   port (
8     Clock            : in  std_logic;
9
10    BCDDigits         : in  T_BCD_VECTOR(DIGITS - 1 downto 0);
11    BCDDots           : in  std_logic_vector(DIGITS - 1 downto 0);
12
13    SegmentControl    : out std_logic_vector(7 downto 0);
14    DigitControl      : out std_logic_vector(DIGITS - 1 downto 0)
15  );
16 end entity;
```

Source file: [io/io_7SegmentMux_BCD.vhdl](#)

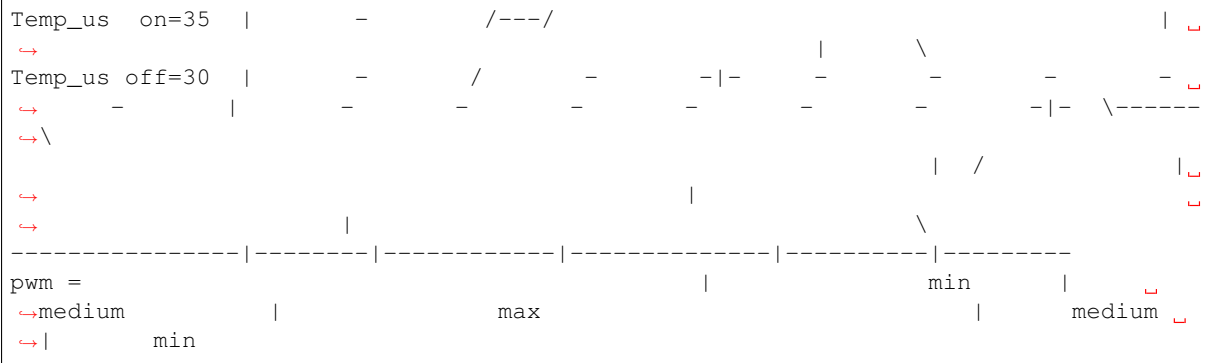
7.10.14 PoC.io.7SegmentMux_HEX

This module is a 7 segment display controller that uses time multiplexing to control a common anode for each digit in the display. The shown characters are HEX encoded. A dot per digit is optional.

Entity Declaration:

```
1 entity io_7SegmentMux_HEX is
2   generic (
3     CLOCK_FREQ      : FREQ      := 100 MHz;
4     REFRESH_RATE    : FREQ      := 1 kHz;
5     DIGITS           : positive  := 4
6   );
7   port (
8     Clock            : in  std_logic;
9
10    HexDigits         : in  T_SLVV_4(DIGITS - 1 downto 0);
11    HexDots           : in  std_logic_vector(DIGITS - 1 downto 0);
12
13    SegmentControl    : out std_logic_vector(7 downto 0);
14    DigitControl      : out std_logic_vector(DIGITS - 1 downto 0)
15  );
16 end entity;
```


(continued from previous page)



Entity Declaration:

```

1 entity io_FanControl is
2   generic (
3     CLOCK_FREQ          : FREQ;
4     ADD_INPUT_SYNCHRONIZERS : boolean := TRUE;
5     ENABLE_TACHO         : boolean := FALSE
6   );
7   port (
8     -- Global Control
9     Clock          : in std_logic;
10    Reset           : in std_logic;
11
12    -- Fan Control derived from internal System Health Monitor
13    Fan_PWM         : out std_logic;
14
15    -- Decoding of Speed Sensor (Requires ENABLE_TACHO)
16    Fan_Tacho       : in std_logic := 'X';
17    TachoFrequency : out std_logic_vector(15 downto 0)
18  );
19 end entity;
```

Source file: io/io_FanControl.vhdl

7.10.17 PoC.io.FrequencyCounter

Todo: No documentation available.

Entity Declaration:

```

1 entity io_FrequencyCounter is
2   generic (
3     CLOCK_FREQ          : FREQ          := 100 MHz;
4     TIMEBASE            : time          := 1 sec;
5     RESOLUTION          : positive     := 8
6   );
7   port (
8     Clock              : in std_logic;
9     Reset              : in std_logic;
10    FreqIn              : in std_logic;
11    FreqOut             : out std_logic_vector(RESOLUTION - 1 downto 0)
```

(continues on next page)

(continued from previous page)

```

12     );
13 end entity;

```

Source file: `io/io_FrequencyCounter.vhdl`

7.10.18 PoC.io.GlitchFilter

This module filters glitches on a wire. The high and low spike suppression cycle counts can be configured.

Entity Declaration:

```

1 entity io_GlitchFilter is
2   generic (
3     HIGH_SPIKE_SUPPRESSION_CYCLES : natural := 5;
4     LOW_SPIKE_SUPPRESSION_CYCLES  : natural := 5;
5   );
6   port (
7     Clock   : in std_logic;
8     Input    : in std_logic;
9     Output   : out std_logic
10  );
11 end entity;

```

Source file: `io/io_GlitchFilter.vhdl`

7.10.19 PoC.io.KeyPadScanner

This module drives a one-hot encoded column vector to read back a rows vector. By scanning column-by-column it's possible to extract the current button state of the whole keypad. The scanner uses high-active logic. The keypad size and scan frequency can be configured. The outputed signal matrix is not debounced.

Entity Declaration:

```

1 entity io_KeyPadScanner is
2   generic (
3     CLOCK_FREQ      : FREQ := 100 MHz;
4     SCAN_FREQ       : FREQ := 1 kHz;
5     ROWS             : positive := 4;
6     COLUMNS         : positive := 4;
7     ADD_INPUT_SYNCHRONIZERS : boolean := TRUE;
8   );
9   port (
10    Clock      : in std_logic;
11    Reset      : in std_logic;
12    -- Matrix interface
13    KeyPadMatrix : out T_SLM(COLUMNS - 1 downto 0, ROWS - 1 downto 0);
14    -- KeyPad interface
15    ColumnVector : out std_logic_vector(COLUMNS - 1 downto 0);
16    RowVector    : in std_logic_vector(ROWS - 1 downto 0)
17  );
18 end entity;

```

Source file: `io/io_KeyPadScanner.vhdl`

7.10.20 PoC.io.PulseWidthModulation

This module generates a pulse width modulated signal, that can be configured in frequency (PWM_FREQ) and modulation granularity (PWM_RESOLUTION).

Entity Declaration:

```
1 entity io_PulseWidthModulation is
2   generic (
3     CLOCK_FREQ          : FREQ          := 100 MHz;
4     PWM_FREQ            : FREQ          := 1 kHz;
5     PWM_RESOLUTION      : positive      := 8
6   );
7   port (
8     Clock               : in  std_logic;
9     Reset               : in  std_logic;
10    PWMIn                : in  std_logic_vector (PWM_RESOLUTION - 1 downto 0);
11    PWMOut               : out std_logic
12  );
13 end entity;
```

Source file: `io/io_PulseWidthModulation.vhdl`

7.10.21 PoC.io.TimingCounter

This down-counter can be configured with a TIMING_TABLE (a ROM), from which the initial counter value is loaded. The table index can be selected by Slot. Timeout is a registered output. Up to 16 values fit into one ROM consisting of $\log_2 \text{ceil}(\text{imax}(\text{TIMING_TABLE})) + 1$ 6-input LUTs.

Entity Declaration:

```
1 entity io_TimingCounter is
2   generic (
3     TIMING_TABLE : T_NATVEC                                -- timing_
4   )
5   port (
6     Clock        : in  std_logic;                          -- clock
7     Enable       : in  std_logic;                          -- enable_
8     Load         : in  std_logic;                          -- load_
9     Slot          : in  natural range 0 to (TIMING_TABLE'length - 1); --
10    Timeout       : out std_logic                          -- timing_
11  )
12  reached
13 end entity;
```

Source file: `io/io_TimingCounter.vhdl`

7.11 PoC.mem

The namespace `PoC.mem` offers different on-chip and off-chip memory and memory-controller implementations.

Sub-Namespaces

- *PoC.mem.ldr2* - DDR2 memory controllers
- *PoC.mem.ldr3* - DDR3 memory controllers
- *PoC.mem.lut* - Lookup-Table (LUT) implementations
- *PoC.mem.ocram* - On-Chip RAM abstraction layer
- *PoC.mem.ocrom* - On-Chip ROM abstraction layer
- *PoC.mem.sdram* - SDRAM controllers

Package

PoC.mem

7.11.1 PoC.mem Package

This package holds all component declarations, types and functions of the *PoC.mem* namespace.

It provides the following enumerations:

- `T_MEM_FILEFORMAT` specifies whether a file is in Intel Hex, Lattice Mem, or Xilinx Mem format.
- `T_MEM_CONTENT` specifies whether data in text file is in binary, decimal or hexadecimal format.

It provides the following functions:

- `mem_FileExtension` returns the file extension of a given filename.
- `mem_ReadMemoryFile` reads initial memory content from a given file.

Source file: [mem.pkg.vhdl](#)

7.11.2 PoC.mem.ldr2

The namespace `PoC.mem.ldr2` is designated for own implementations of DDR2 memory controllers as well as for adapters for vendor-specific implementations. At the top-level, all controllers and adapters provide the same simple memory interface to the user application.

Entities

- *PoC.mem.ldr2.mem2mig_adapter_Spartan6* - Adapter for the Xilinx MIG core for Spartan-6 FPGAs

PoC.mem.ldr2.mem2mig_adapter_Spartan6

Adapter between the *PoC.Mem* interface and the User Interface of the Xilinx MIG IP core for the Spartan-6 FPGA Memory Controller Block (MCB). The MCB can be configured to have multiple ports. One instance of this adapter is required for every port. The control signals for one port of the MIG IP core are prefixed by “cX_pY”, meaning port Y on controller X.

Simplifies the User Interface (“user”) of the Xilinx MIG IP core (UG388). The PoC.Mem interface provides single-cycle fully pipelined read/write access to the memory. All accesses are word-aligned. Always all bytes of a word are written to the memory. More details can be found [here](#).

Generic parameters:

- `D_BITS`: Data bus width of the PoC.Mem and MIG / MCBinterface. Also size of one word in bits.
- `MEM_A_BITS`: Address bus width of the PoC.Mem interface.
- `APP_A_BTIS`: Address bus width of the MIG / MCB interface.

Contains only combinational logic.

Entity Declaration:

```
1 entity ddr2_mem2mig_adapter_Spartan6 is
2
3   generic (
4       D_BITS      : positive;
5       MEM_A_BITS  : positive;
6       APP_A_BITS  : positive
7   );
8
9   port (
10      -- PoC.Mem interface
11      mem_req      : in  std_logic;
12      mem_write    : in  std_logic;
13      mem_addr     : in  unsigned(MEM_A_BITS-1 downto 0);
14      mem_wdata    : in  std_logic_vector(D_BITS-1 downto 0);
15      mem_wmask    : in  std_logic_vector(D_BITS/8-1 downto 0) := (others => '0');
16      mem_rdy      : out std_logic;
17      mem_rstb     : out std_logic;
18      mem_rdata    : out std_logic_vector(D_BITS-1 downto 0);
19
20      -- Xilinx MIG IP Core interface
21      mig_calib_done : in  std_logic;
22      mig_cmd_full   : in  std_logic;
23      mig_wr_full    : in  std_logic;
24      mig_rd_empty   : in  std_logic;
25      mig_rd_data    : in  std_logic_vector((D_BITS)-1 downto 0);
26      mig_cmd_instr  : out std_logic_vector(2 downto 0);
27      mig_cmd_en     : out std_logic;
28      mig_cmd_bl     : out std_logic_vector(5 downto 0);
29      mig_cmd_byte_addr : out std_logic_vector(APP_A_BITS-1 downto 0);
30      mig_wr_data    : out std_logic_vector((D_BITS)-1 downto 0);
31      mig_wr_mask    : out std_logic_vector((D_BITS)/8-1 downto 0);
32      mig_wr_en      : out std_logic;
33      mig_rd_en      : out std_logic
34   );
35
36 end entity ddr2_mem2mig_adapter_Spartan6;
```

Source file: `mem/ddr2/ddr2_mem2mig_adapter_Spartan6.vhdl`

7.11.3 PoC.mem.ddr3

The namespace `PoC.mem.ddr3` is designated for own implementations of DDR3 memory controllers as well as for adapters for vendor-specific implementations. At the top-level, all controllers and adapters provide the same simple memory interface to the user application.

Entities

- `PoC.mem.ddr3.mem2mig_adapter_Series7` - Adapter for the Xilinx MIG core for 7-Series FPGAs

PoC.mem.ddr3.mem2mig_adapter_Series7

Adapter between the *PoC.Mem* interface and the application interface (“app”) of the Xilinx MIG IP core for 7-Series FPGAs.

Simplifies the application interface (“app”) of the Xilinx MIG IP core. The *PoC.Mem* interface provides single-cycle fully pipelined read/write access to the memory. All accesses are word-aligned. Always all bytes of a word are written to the memory. More details can be found [here](#).

Generic parameters:

- D_BITS: Data bus width of the PoC.Mem and “app” interface. Also size of one word in bits.
- DQ_BITS: Size of data bus between memory controller and external memory (DIMM, SoDIMM).
- MEM_A_BITS: Address bus width of the PoC.Mem interface.
- APP_A_BITS: Address bus width of the “app” interface.

Contains only combinational logic.

Entity Declaration:

```

1  entity ddr3_mem2mig_adapter_Series7 is
2
3  generic (
4      D_BITS      : positive;
5      DQ_BITS     : positive;
6      MEM_A_BITS  : positive;
7      APP_A_BITS  : positive
8  );
9
10 port (
11     -- PoC.Mem interface
12     mem_req      : in  std_logic;
13     mem_write    : in  std_logic;
14     mem_addr     : in  unsigned(MEM_A_BITS-1 downto 0);
15     mem_wdata    : in  std_logic_vector(D_BITS-1 downto 0);
16     mem_wmask    : in  std_logic_vector(D_BITS/8-1 downto 0) := (others => '0');
17     mem_rdy      : out std_logic;
18     mem_rstb     : out std_logic;
19     mem_rdata    : out std_logic_vector(D_BITS-1 downto 0);
20
21     -- Xilinx MIG IP Core interface
22     init_calib_complete : in  std_logic;
23     app_rd_data         : in  std_logic_vector((D_BITS)-1 downto 0);
24     app_rd_data_end     : in  std_logic;
25     app_rd_data_valid   : in  std_logic;
26     app_rdy             : in  std_logic;
27     app_wdf_rdy         : in  std_logic;
28     app_addr            : out std_logic_vector(APP_A_BITS-1 downto 0);
29     app_cmd             : out std_logic_vector(2 downto 0);
30     app_en              : out std_logic;
31     app_wdf_data        : out std_logic_vector((D_BITS)-1 downto 0);
32     app_wdf_end         : out std_logic;
33     app_wdf_mask        : out std_logic_vector((D_BITS)/8-1 downto 0);
34     app_wdf_wren        : out std_logic
35 );
36
37 end entity ddr3_mem2mig_adapter_Series7;
```

Source file: mem/ddr3/ddr3_mem2mig_adapter_Series7.vhdl

7.11.4 PoC.mem.lut

The namespace `PoC.mem.lut` offers different lookup-tables (LUTs).

Entities

- *PoC.mem.lut.Sine* - a Sine implementation with 1,2 or 4 quadrants.

PoC.mem.lut.Sine

Todo: No documentation available.

Entity Declaration:

Source file: `mem/lut/lut_Sine.vhdl`

7.11.5 PoC.mem.ocram

The namespace `PoC.mem.ocram` offers different on-chip RAM abstractions.

Package

The package `PoC.mem.ocram` holds all component declarations for this namespace.

```
library PoC;
use     PoC.ocram.all;
```

Entities

- *PoC.mem.ocram.sp* - An on-chip RAM with a single port interface.
- *PoC.mem.ocram.sdp* - An on-chip RAM with a simple dual-port interface.
- *PoC.mem.ocram.sdp_wf* - An on-chip RAM with a simple dual-port interface and write-first behavior.
- *PoC.mem.ocram.tdp* - An on-chip RAM with a true dual-port interface.
- *PoC.mem.ocram.tdp_wf* - An on-chip RAM with a true dual-port interface and write-first behavior.

Simulation Helper

- *PoC.mem.ocram.tdp_sim* - Simulation model of on-chip RAM with a true dual port interface.

Deprecated Entities

- *PoC.mem.ocram.esdp* - An on-chip RAM with an extended simple dual port interface.

PoC.mem.ocram Package

Source file: `ocram.pkg.vhdl`

PoC.mem.ocram.sp

Inferring / instantiating single port memory, with:

- single clock, clock enable,
- 1 read/write port.

Command Truth Table:

ce	we	Command
0	X	No operation
1	0	Read from memory
1	1	Write to memory

Both reading and writing are synchronous to the rising-edge of the clock. Thus, when reading, the memory data will be outputted after the clock edge, i.e, in the following clock cycle.

When writing data, the read output will output the new data (in the following clock cycle) which is aka. “write-first behavior”. This behavior also applies to Altera M20K memory blocks as described in the Altera: “Stratix 5 Device Handbook” (S5-5V1). The documentation in the Altera: “Embedded Memory User Guide” (UG-01068) is wrong.

Entity Declaration:

```

1 entity ocram_sp is
2   generic (
3     A_BITS      : positive;           -- number of address bits
4     D_BITS      : positive;           -- number of data bits
5     FILENAME    : string := ""       -- file-name for RAM_
6   )
7   port (
8     clk : in std_logic;               -- clock
9     ce  : in std_logic;               -- clock enable
10    we  : in std_logic;               -- write enable
11    a   : in unsigned(A_BITS-1 downto 0); -- address
12    d   : in std_logic_vector(D_BITS-1 downto 0); -- write data
13    q   : out std_logic_vector(D_BITS-1 downto 0); -- read output
14  );
15 end entity;
```

Source file: [mem/ocram/ocram_sp.vhdl](#)

PoC.mem.ocram.sdp

Inferring / instantiating simple dual-port memory, with:

- dual clock, clock enable,
- 1 read port plus 1 write port.

Both reading and writing are synchronous to the rising-edge of the clock. Thus, when reading, the memory data will be outputted after the clock edge, i.e, in the following clock cycle.

The generalized behavior across Altera and Xilinx FPGAs since Stratix/Cyclone and Spartan-3/Virtex-5, respectively, is as follows:

Mixed-Port Read-During-Write When reading at the write address, the read value will be unknown which is aka. “don’t care behavior”. This applies to all reads (at the same address) which are issued during the write-cycle time, which starts at the rising-edge of the write clock and (in the worst case) extends until the next rising-edge of the write clock.

For simulation, always our dedicated simulation model *PoC.mem.ocram.tdp_sim* is used.

Entity Declaration:

```

1 entity ocram_sdp is
2   generic (
3     A_BITS      : positive;           -- number of address bits
4     D_BITS      : positive;           -- number of data bits
5     FILENAME    : string := ""       -- file-name for RAM_
6   )
7   port (
8     rclk : in std_logic;               -- read clock
9     rce  : in std_logic;               -- read clock-enable
```

(continues on next page)

(continued from previous page)

```

10     wclk  : in  std_logic;           -- write clock
11     wce   : in  std_logic;           -- write clock-enable
12     we    : in  std_logic;           -- write enable
13     ra    : in  unsigned(A_BITS-1 downto 0); -- read address
14     wa    : in  unsigned(A_BITS-1 downto 0); -- write address
15     d     : in  std_logic_vector(D_BITS-1 downto 0); -- data in
16     q     : out std_logic_vector(D_BITS-1 downto 0) -- data out
17 );
18 end entity;

```

Source file: `mem/ocram/ocram_sdp.vhdl`

PoC.mem.ocram.sdp_wf

Inferring / instantiating simple dual-port memory, with:

- single clock, clock enable,
- 1 read port plus 1 write port.

Command truth table:

ce	we	Command
0	X	No operation
1	0	Read only from memory
1	1	Read from and Write to memory

Both reading and writing are synchronous to the rising-edge of the clock. Thus, when reading, the memory data will be outputted after the clock edge, i.e, in the following clock cycle.

Mixed-Port Read-During-Write When reading at the write address, the read value will be the new data, aka. “write-first behavior”. Of course, the read is still synchronous, i.e, the latency is still one clock cycle.

Entity Declaration:

```

1  entity ocram_sdp_wf is
2      generic (
3          A_BITS    : positive;           -- number of address bits
4          D_BITS    : positive;           -- number of data bits
5          FILENAME  : string := ""        -- file-name for RAM_
6      )
7      port (
8          clk : in  std_logic;           -- clock
9          ce  : in  std_logic;           -- clock-enable
10         we   : in  std_logic;           -- write enable
11         ra   : in  unsigned(A_BITS-1 downto 0); -- read address
12         wa   : in  unsigned(A_BITS-1 downto 0); -- write address
13         d    : in  std_logic_vector(D_BITS-1 downto 0); -- data in
14         q    : out std_logic_vector(D_BITS-1 downto 0) -- data out
15     );
16 end entity;

```

Source file: `mem/ocram/ocram_sdp_wf.vhdl`

PoC.mem.ocram.tdp

Inferring / instantiating true dual-port memory, with:

- dual clock, clock enable,
- 2 read/write ports.

Command truth table for port 1, same applies to port 2:

ce1	we1	Command
0	X	No operation
1	0	Read from memory
1	1	Write to memory

Both reading and writing are synchronous to the rising-edge of the clock. Thus, when reading, the memory data will be outputted after the clock edge, i.e, in the following clock cycle.

The generalized behavior across Altera and Xilinx FPGAs since Stratix/Cyclone and Spartan-3/Virtex-5, respectively, is as follows:

Same-Port Read-During-Write When writing data through port 1, the read output of the same port (q1) will output the new data (d1, in the following clock cycle) which is aka. “write-first behavior”.

Same applies to port 2.

Mixed-Port Read-During-Write When reading at the write address, the read value will be unknown which is aka. “don’t care behavior”. This applies to all reads (at the same address) which are issued during the write-cycle time, which starts at the rising-edge of the write clock and (in the worst case) extends until the next rising-edge of that write clock.

For simulation, always our dedicated simulation model *PoC.mem.ocram.tdp_sim* is used.

Entity Declaration:

```

1 entity ocram_tdp is
2   generic (
3     A_BITS    : positive;           -- number of address bits
4     D_BITS    : positive;           -- number of data bits
5     FILENAME  : string := "";       -- file-name for RAM_
6   )
7   port (
8     clk1 : in std_logic;           -- clock for 1st port
9     clk2 : in std_logic;           -- clock for 2nd port
10    ce1  : in std_logic;           -- clock-enable for 1st port
11    ce2  : in std_logic;           -- clock-enable for 2nd port
12    we1  : in std_logic;           -- write-enable for 1st port
13    we2  : in std_logic;           -- write-enable for 2nd port
14    a1   : in unsigned(A_BITS-1 downto 0); -- address for 1st port
15    a2   : in unsigned(A_BITS-1 downto 0); -- address for 2nd port
16    d1   : in std_logic_vector(D_BITS-1 downto 0); -- write-data for 1st port
17    d2   : in std_logic_vector(D_BITS-1 downto 0); -- write-data for 2nd port
18    q1   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 1st port
19    q2   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 2nd port
20  );
21 end entity;

```

Source file: `mem/ocram/ocram_tdp.vhdl`

PoC.mem.ocram.tdp_wf

Inferring / instantiating true dual-port memory, with:

- single clock, clock enable,

- 2 read/write ports.

Command truth table:

ce	we1	we2	Command
0	X	X	No operation
1	0	0	Read only from memory
1	0	1	Read from memory on port 1, write to memory on port 2
1	1	0	Write to memory on port 1, read from memory on port 2
1	1	1	Write to memory on both ports

Both reads and writes are synchronous to the clock.

The generalized behavior across Altera and Xilinx FPGAs since Stratix/Cyclone and Spartan-3/Virtex-5, respectively, is as follows:

Same-Port Read-During-Write When writing data through port 1, the read output of the same port (q1) will output the new data (d1, in the following clock cycle) which is aka. “write-first behavior”.

Same applies to port 2.

Mixed-Port Read-During-Write When reading at the write address, the read value will be the new data, aka. “write-first behavior”. Of course, the read is still synchronous, i.e, the latency is still one clock cycle.

If a write is issued on both ports to the same address, then the output of this unit and the content of the addressed memory cell are undefined.

For simulation, always our dedicated simulation model *PoC.mem.ocram.tdp_sim* is used.

Entity Declaration:

```
1  entity ocram_tdp_wf is
2      generic (
3          A_BITS      : positive;           -- number of address bits
4          D_BITS      : positive;           -- number of data bits
5          FILENAME    : string := "";       -- file-name for RAM_
6      ) initialization
7  );
8      port (
9          clk : in  std_logic;               -- clock
10         ce  : in  std_logic;               -- clock-enable
11         we1 : in  std_logic;               -- write-enable for 1st port
12         we2 : in  std_logic;               -- write-enable for 2nd port
13         a1  : in  unsigned(A_BITS-1 downto 0); -- address for 1st port
14         a2  : in  unsigned(A_BITS-1 downto 0); -- address for 2nd port
15         d1  : in  std_logic_vector(D_BITS-1 downto 0); -- write-data for 1st port
16         d2  : in  std_logic_vector(D_BITS-1 downto 0); -- write-data for 2nd port
17         q1  : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 1st port
18         q2  : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 2nd port
19     );
20 end entity;
```

Source file: mem/ocram/ocram_tdp_wf.vhdl

PoC.mem.ocram.tdp_sim

Simulation model for true dual-port memory, with:

- dual clock, clock enable,
- 2 read/write ports.

The interface matches that of the IP core PoC.mem.ocram.tdp. But the implementation there is restricted to the description supported by various synthesis compilers. The implementation here also simulates the correct Mixed-Port Read-During-Write Behavior and handles X propagation.

Entity Declaration:

```

1 entity ocram_tdp_sim is
2   generic (
3     A_BITS      : positive;           -- number of address bits
4     D_BITS      : positive;           -- number of data bits
5     FILENAME    : string := ""       -- file-name for RAM_
6   )
7   port (
8     clk1 : in std_logic;              -- clock for 1st port
9     clk2 : in std_logic;              -- clock for 2nd port
10    ce1  : in std_logic;               -- clock-enable for 1st port
11    ce2  : in std_logic;               -- clock-enable for 2nd port
12    we1  : in std_logic;               -- write-enable for 1st port
13    we2  : in std_logic;               -- write-enable for 2nd port
14    a1   : in unsigned(A_BITS-1 downto 0); -- address for 1st port
15    a2   : in unsigned(A_BITS-1 downto 0); -- address for 2nd port
16    d1   : in std_logic_vector(D_BITS-1 downto 0); -- write-data for 1st port
17    d2   : in std_logic_vector(D_BITS-1 downto 0); -- write-data for 2nd port
18    q1   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 1st port
19    q2   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 2nd port
20  );
21 end entity;

```

Source file: [mem/ocram/ocram_tdp_sim.vhdl](#)

PoC.mem.ocram.esdp

Inferring / instantiating enhanced simple dual-port memory, with:

- dual clock, clock enable,
- 1 read/write port (1st port) plus 1 read port (2nd port).

Deprecated since version 1.1: **Please use [PoC.mem.ocram.tdp](#) for new designs.** This component has been provided because older FPGA compilers were not able to infer true dual-port memory from an RTL description.

Command truth table for port 1:

ce1	we1	Command
0	X	No operation
1	0	Read from memory
1	1	Write to memory

Command truth table for port 2:

ce2	Command
0	No operation
1	Read from memory

Both reading and writing are synchronous to the rising-edge of the clock. Thus, when reading, the memory data will be outputted after the clock edge, i.e, in the following clock cycle.

The generalized behavior across Altera and Xilinx FPGAs since Stratix/Cyclone and Spartan-3/Virtex-5, respectively, is as follows:

Same-Port Read-During-Write When writing data through port 1, the read output of the same port (`q1`) will output the new data (`d1`, in the following clock cycle) which is aka. “write-first behavior”.

Mixed-Port Read-During-Write When reading at the write address, the read value will be unknown which is aka. “don’t care behavior”. This applies to all reads (at the same address) which are issued during the write-cycle time, which starts at the rising-edge of the write clock (`clk1`) and (in the worst case) extends until the next rising-edge of the write clock.

For simulation, always our dedicated simulation model *PoC.mem.ocram.tdp_sim* is used.

Entity Declaration:

```
1 entity ocram_esdp is
2   generic (
3     A_BITS      : positive;           -- number of address bits
4     D_BITS      : positive;           -- number of data bits
5     FILENAME    : string := ""       -- file-name for RAM_
6   ) -- initialization
7   port (
8     clk1 : in std_logic;             -- clock for 1st port
9     clk2 : in std_logic;             -- clock for 2nd port
10    ce1  : in std_logic;             -- clock-enable for 1st port
11    ce2  : in std_logic;             -- clock-enable for 2nd port
12    we1  : in std_logic;             -- write-enable for 1st port
13    a1   : in unsigned(A_BITS-1 downto 0); -- address for 1st port
14    a2   : in unsigned(A_BITS-1 downto 0); -- address for 2nd port
15    d1   : in std_logic_vector(D_BITS-1 downto 0); -- write-data for 1st port
16    q1   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 1st port
17    q2   : out std_logic_vector(D_BITS-1 downto 0); -- read-data from 2nd port
18  );
19 end entity;
```

Source file: `mem/ocram/ocram_esdp.vhdl`

7.11.6 PoC.mem.ocrom

The namespace `PoC.mem.ocrom` offers different on-chip ROM abstractions.

Package

The package `PoC.mem.ocrom` holds all component declarations for this namespace.

```
library PoC;
use      PoC.ocrom.all;
```

Entities

- *ocrom_sp* is a on-chip RAM with a single port interface.
- *ocrom_dp* is a on-chip RAM with a dual port interface.

PoC.mem.ocrom Package

Source file: `ocrom.pkg.vhdl`

PoC.mem.ocrom.sp

Inferring / instantiating single-port read-only memory

- single clock, clock enable
- 1 read port

Entity Declaration:

```

1 entity ocrom_sp is
2   generic (
3     A_BITS    : positive;
4     D_BITS    : positive;
5     FILENAME  : string    := ""
6   );
7   port (
8     clk : in  std_logic;
9     ce  : in  std_logic;
10    a   : in  unsigned(A_BITS-1 downto 0);
11    q   : out std_logic_vector(D_BITS-1 downto 0)
12  );
13 end entity;
```

Source file: `mem/ocrom/ocrom_sp.vhdl`

PoC.mem.ocrom.dp

Inferring / instantiating dual-port read-only memory, with:

- dual clock, clock enable,
- 2 read ports.

The generalized behavior across Altera and Xilinx FPGAs since Stratix/Cyclone and Spartan-3/Virtex-5, respectively, is as follows:

WARNING: The simulated behavior on RT-level is not correct.

TODO: add timing diagram TODO: implement correct behavior for RT-level simulation

Entity Declaration:

```

1 entity ocrom_dp is
2   generic (
3     A_BITS    : positive;
4     D_BITS    : positive;
5     FILENAME  : string    := ""
6   );
7   port (
8     clk1 : in  std_logic;
9     clk2 : in  std_logic;
10    ce1  : in  std_logic;
11    ce2  : in  std_logic;
12    a1   : in  unsigned(A_BITS-1 downto 0);
13    a2   : in  unsigned(A_BITS-1 downto 0);
14    q1   : out std_logic_vector(D_BITS-1 downto 0);
15    q2   : out std_logic_vector(D_BITS-1 downto 0)
16  );
17 end entity;
```

Source file: `mem/ocrom/ocrom_dp.vhdl`

7.11.7 PoC.mem.sdram

The namespace `PoC.mem.sdram` offers components for the access of external SDRAMs. A common finite state-machine is used to address the memory via banks, rows and columns. Different physical layers are provide for the single-data-rate (SDR) or double-data-rate (DDR, DDR2, ...) data bus. One has to instantiate the specific module required by the FPGA board.

SDRAM Controller for the Altera DE0 Board

The module `sdram_ctrl_de0` combines the finite state machine `sdram_ctrl_fsm` and the DE0 specific physical layer `sdram_ctrl_phy_de0`. It has been tested with the IS42S16400F SDR memory at a frequency of 133 MHz. A usage example is given in [PoC-Examples](#).

SDRAM Controller for the Xilinx Spartan-3E Starter Kit (S3ESK)

The module `sdram_ctrl_s3esk` combines the finite state machine `sdram_ctrl_fsm` and the S3ESK specific physical layer `sdram_ctrl_phy_s3esk`. It has been tested with the MT46V32M16-6T DDR memory at a frequency of 100 MHz (DDR-200). A usage example is given in [PoC-Examples](#).

Note: See also [PoC.xil.mig](#) for board specific memory controller implementations created by Xilinx's Memory Interface Generator (MIG).

PoC.mem.sdram.ctrl_fsm

This file contains the FSM as well as parts of the datapath. The board specific physical layer is defined in another file.

Configuration

SDRAM_TYPE activates some special cases:

- 0 for SDR-SDRAM
- 1 for DDR-SDRAM
- 2 for DDR2-SDRAM (no special support yet like ODT)

`2**A_BITS` specifies the number of memory cells in the SDRAM. This is the size of th memory in bits divided by the native data-path width of the SDRAM (also in bits).

`D_BITS` is the native data-path width of the SDRAM. The width might be doubled by the physical interface for DDR interfaces.

Furthermore, the memory array is divided into `2**R_BITS` rows, `2**C_BITS` columns and `2**B_BITS` banks.

Note: For example, the MT46V32M16 has 512 Mbit = 8M x 4 banks x 16 bit = 32M cells x 16 bit, with 8K rows and 1K columns. Thus, the configuration is:

- $A_BITS = \log_2(32\text{ M}) = 25$
- $D_BITS = 16$
- data-path width of phy on user side: 32-bit because of DDR
- $R_BITS = \log_2(8\text{ K}) = 13$

- $C_BITS = \log_2(1\text{ K}) = 10$
- $B_BITS = \log_2(4) = 2$

Set CAS latency (CL, MR_CL) and burst length (BL, MR_BL) according to your needs.

If you have a DDR-SDRAM then set INIT_DLL = true, otherwise false.

The definition and values of generics T_* can be calculated from the datasheets of the specific SDRAM (e.g. MT46V). Just divide the minimum/maximum times by clock period. Auto refreshs are applied periodically, the datasheet either specifies the average refresh interval (T_REFI) or the total refresh cycle time (T_REF). In the latter case, divide the total time by the row count to get the average refresh interval. Subtract about 50 clock cycles to account for pending read/writes.

INIT_WAIT specifies the time period to wait after the SDRAM is powered up. It is typically 100–200 us long, see datasheet. The waiting time is specified in number of average refresh periods (specified by T_REFI): $INIT_WAIT = \text{ceil}(\text{wait_time} / \text{clock_period} / T_REFI)$ e.g. $INIT_WAIT = \text{ceil}(200\text{ us} / 10\text{ ns} / 700) = 29$

Operation

After user_cmd_valid is asserted high, the command (user_write) and address (user_addr) must be hold until user_got_cmd is asserted.

The FSM automatically waits for user_wdata_valid on writes. The data should be available soon. Otherwise the auto refresh might fail. The FSM only waits for the first word to write. All successive words of a burst must be valid in the following cycles. (A burst can't be stalled.) ATTENTION: During writes, user_cmd_got is asserted only if user_wdata_valid is set.

The write data must directly connected to the physical layer.

Entity Declaration:

```

1  entity sdram_ctrl_fsm is
2      generic (
3          SDRAM_TYPE : natural;           -- SDRAM type
4
5          A_BITS      : positive;         -- log2ceil(memory cell count)
6          D_BITS      : positive;         -- native data width
7          R_BITS      : positive;         -- log2ceil(rows)
8          C_BITS      : positive;         -- log2ceil(columns)
9          B_BITS      : positive;         -- log2ceil(banks)
10
11         CL : positive;  -- CAS Latency in clock cycles
12         BL : positive;  -- Burst Length
13
14         T_MRD      : integer;           -- in clock cycles
15         T_RAS      : integer;           -- in clock cycles
16         T_RCD      : integer;           -- in clock cycles
17         T_RFC      : integer;           -- (or T_RC) in clock cycles
18         T_RP       : integer;           -- in clock cycles
19         T_WR        : integer;           -- in clock cycles
20         T_WTR       : integer;           -- in clock cycles
21         T_REFI      : integer;           -- in clock cycles
22         INIT_WAIT   : integer);         -- in T_REFI periods
23     port (
24         clk : in std_logic;
25         rst : in std_logic;
26
27         user_cmd_valid : in std_logic;
28         user_wdata_valid : in std_logic;

```

(continues on next page)

(continued from previous page)

```

29  user_write      : in  std_logic;
30  user_addr      : in  std_logic_vector(A_BITS-1 downto 0);
31  user_got_cmd   : out std_logic;
32  user_got_wdata : out std_logic;
33
34  sd_cke_nxt      : out std_logic;
35  sd_cs_nxt       : out std_logic;
36  sd_ras_nxt      : out std_logic;
37  sd_cas_nxt      : out std_logic;
38  sd_we_nxt       : out std_logic;
39  sd_a_nxt        : out std_logic_vector(imax(R_BITS,C_BITS+1)-1 downto 0);
40  sd_ba_nxt       : out std_logic_vector(B_BITS-1 downto 0);
41  rden_nxt        : out std_logic;
42  wren_nxt        : out std_logic);
43
44
45  end sdram_ctrl_fsm;

```

Source file: `mem/sdram/sdram_ctrl_fsm.vhdl`

PoC.mem.sdram.ctrl_de0

Complete controller for ISSI SDR-SDRAM for Altera DE0 Board.

SDRAM Device: IS42S16400F

Configuration

Parameter	Description
CLK_PERIOD	Clock period in nano seconds. All SDRAM timings are calculated for the device stated above.
CL	CAS latency, choose according to clock frequency.
BL	Burst length. Choose BL=1 for single cycle memory transactions as required for the PoC.Mem interface.

Tested with: CLK_PERIOD = 7.5 (133 MHz), CL=2, BL=1.

Operation

Command, address and write data is sampled with `clk`. Read data is also aligned with `clk`.

For description on `clkout` see `sdram_ctrl_phy_de0`.

Synchronous resets are used.

Entity Declaration:

```

1  entity sdram_ctrl_de0 is
2
3      generic (
4          CLK_PERIOD : real;
5          CL          : positive;
6          BL          : positive);
7
8      port (
9          clk          : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

10  clkout      : in    std_logic;
11  rst        : in    std_logic;
12
13  user_cmd_valid : in  std_logic;
14  user_wdata_valid : in std_logic;
15  user_write    : in  std_logic;
16  user_addr     : in  std_logic_vector(21 downto 0);
17  user_wdata    : in  std_logic_vector(15 downto 0);
18  user_got_cmd  : out std_logic;
19  user_got_wdata : out std_logic;
20  user_rdata    : out std_logic_vector(15 downto 0);
21  user_rstb     : out std_logic;
22
23  sd_ck       : out  std_logic;
24  sd_cke      : out  std_logic;
25  sd_cs       : out  std_logic;
26  sd_ras      : out  std_logic;
27  sd_cas      : out  std_logic;
28  sd_we       : out  std_logic;
29  sd_ba       : out  std_logic_vector(1 downto 0);
30  sd_a        : out  std_logic_vector(11 downto 0);
31  sd_dq       : inout std_logic_vector(15 downto 0);
32
33  end sdram_ctrl_de0;

```

Source file: `mem/sdram/sdram_ctrl_de0.vhdl`

PoC.mem.sdram.ctrl_phy_de0

Physical layer used by module `sdram_ctrl_de0`.

Instantiates input and output buffer components and adjusts the timing for the Altera DE0 board.

Clock and Reset Signals

Port	Description
clk	Base clock for command and write data path.
rst	Reset for clk.

Command signals and write data are sampled with `clk`. Read data is also aligned with `clk`.

Write and read enable (`wren_nxt`, `rden_nxt`) must be hold for:

- 1 clock cycle if `BL = 1`,
- 2 clock cycles if `BL = 2`, or
- 4 clock cycles if `BL = 4`, or
- 8 clock cycles if `BL = 8`.

They must be first asserted with the read and write command. Proper delay is included in this unit.

The first word to write must be asserted with the write command. Proper delay is included in this unit.

Synchronous resets are used. Reset must be hold for at least two cycles.

Entity Declaration:

```
1 entity sdram_ctrl_phy_de0 is
2   generic (
3     CL : positive);           -- CAS latency
4   port (
5     clk      : in std_logic;
6     clkout   : in std_logic;
7     rst      : in std_logic;
8
9     sd_cke_nxt : in std_logic;
10    sd_cs_nxt  : in std_logic;
11    sd_ras_nxt : in std_logic;
12    sd_cas_nxt : in std_logic;
13    sd_we_nxt  : in std_logic;
14    sd_ba_nxt  : in std_logic_vector(1 downto 0);
15    sd_a_nxt   : in std_logic_vector(11 downto 0);
16
17    wren_nxt   : in std_logic;
18    wdata_nxt  : in std_logic_vector(15 downto 0);
19
20    rden_nxt   : in std_logic;
21    rdata      : out std_logic_vector(15 downto 0);
22    rstb       : out std_logic;
23
24    sd_ck      : out std_logic;
25    sd_cke     : out std_logic;
26    sd_cs      : out std_logic;
27    sd_ras     : out std_logic;
28    sd_cas     : out std_logic;
29    sd_we      : out std_logic;
30    sd_ba      : out std_logic_vector(1 downto 0);
31    sd_a       : out std_logic_vector(11 downto 0);
32    sd_dq      : inout std_logic_vector(15 downto 0));
33
34 end sdram_ctrl_phy_de0;
```

Source file: mem/sdram/sdram_ctrl_phy_de0.vhdl

PoC.mem.sdram.ctrl_s3esk

Controller for Micron DDR-SDRAM on Spartan-3E Starter Kit Board.

SDRAM Device: MT46V32M16-6T

Configuration

Parameter	Description
CLK_PERIOD	Clock period in nano seconds. All SDRAM timings are calculated for the device stated above.
CL	CAS latency, choose according to clock frequency.
BL	Burst length. Choose BL=2 for single cycle memory transactions as required for the PoC.Mem interface.

Tested with: CLK_PERIOD = 10.0, CL=2, BL=2.

Operation

Command, address and write data are sampled with the rising edge of `clk`.

Read data is aligned with `clk_fb90_n`. Either process data in this clock domain, or connect a FIFO to transfer data into another clock domain of your choice. This FIFO should be capable of storing at least one burst (size $BL/2$) + start of next burst (size 1).

Synchronous resets are used.

Entity Declaration:

```

1  entity sdram_ctrl_s3esk is
2
3      generic (
4          CLK_PERIOD : real;
5          BL          : positive);
6
7      port (
8          clk          : in    std_logic;
9          clk_n        : in    std_logic;
10         clk90         : in    std_logic;
11         clk90_n      : in    std_logic;
12         rst          : in    std_logic;
13         rst90        : in    std_logic;
14         rst180       : in    std_logic;
15         rst270       : in    std_logic;
16         clk_fb90     : in    std_logic;
17         clk_fb90_n   : in    std_logic;
18         rst_fb90     : in    std_logic;
19         rst_fb270    : in    std_logic;
20
21         user_cmd_valid : in    std_logic;
22         user_wdata_valid : in    std_logic;
23         user_write     : in    std_logic;
24         user_addr      : in    std_logic_vector(24 downto 0);
25         user_wdata     : in    std_logic_vector(31 downto 0);
26         user_got_cmd   : out   std_logic;
27         user_got_wdata : out   std_logic;
28         user_rdata     : out   std_logic_vector(31 downto 0);
29         user_rstb      : out   std_logic;
30
31         sd_ck_p       : out   std_logic;
32         sd_ck_n       : out   std_logic;
33         sd_cke        : out   std_logic;
34         sd_cs         : out   std_logic;
35         sd_ras        : out   std_logic;
36         sd_cas        : out   std_logic;
37         sd_we         : out   std_logic;
38         sd_ba         : out   std_logic_vector(1 downto 0);
39         sd_a          : out   std_logic_vector(12 downto 0);
40         sd_ldqs       : out   std_logic;
41         sd_udqs       : out   std_logic;
42         sd_dq         : inout  std_logic_vector(15 downto 0));
43
44  end sdram_ctrl_s3esk;
```

Source file: `mem/sdram/sdram_ctrl_s3esk.vhdl`

PoC.mem.sdram.ctrl_phy_s3esk

Physical layer used by module *sdram_ctrl_s3esk*.

Instantiates input and output buffer components and adjusts the timing for the Spartan-3E Starter Kit Board.

Clock and Reset Signals

Port	Description
clk	Base clock for command and write data path.
clk_n	clk phase shifted by 180 degrees.
clk90	clk phase shifted by 90 degrees.
clk90_n	clk phase shifted by 270 degrees.
clk_fb (on PCB)	Driven by external feedback (sd_ck_fb) of DDR-SDRAM clock (sd_ck_p). Actually unused, just referenced below.
clk_fb90	clk_fb phase shifted by 90 degrees.
clk_fb90_n	clk_fb phase shifted by 270 degrees.
rst	Reset for clk.
rst180	Reset for clk_n
rst90	Reset for clk90.
rst270	Reset for clk270.
rst_fb90	Reset for clk_fb90.
rst_fb90_n	Reset for clk_fb90_n.

Operation

Command signals and write data are sampled with the rising edge of clk.

Read data is aligned with clk_fb90_n. Either process data in this clock domain, or connect a FIFO to transfer data into another clock domain of your choice. This FIFO should capable of storing at least one burst (size BL/2) + start of next burst (size 1).

Write and read enable (wren_nxt, rden_nxt) must be hold for:

- 1 clock cycle if BL = 2,
- 2 clock cycles if BL = 4, or
- 4 clock cycles if BL = 8.

They must be first asserted with the read and write command. Proper delay is included in this unit.

The first word to write must be asserted with the write command. Proper delay is included in this unit.

The SDRAM clock is regenerated in this module. The following timing is chosen for minimum latency (should work up to 100 MHz):

- `rising_edge(clk90)` triggers `rising_edge(sd_ck_p)`,
- `rising_edge(clk90_n)` triggers `falling_edge(sd_ck_p)`.

XST options: Disable equivalent register removal.

Synchronous resets are used. Reset must be hold for at least two cycles.

Entity Declaration:

```

1  entity sdram_ctrl_phy_s3esk is
2      port (
3          clk      : in std_logic;
4          clk_n    : in std_logic;
5          clk90    : in std_logic;
6          clk90_n  : in std_logic;
7          rst      : in std_logic;
8          rst90    : in std_logic;
9          rst180   : in std_logic;
10         rst270   : in std_logic;
11
12         clk_fb90  : in std_logic;
13         clk_fb90_n : in std_logic;
14         rst_fb90  : in std_logic;
15         rst_fb270 : in std_logic;
16
17         sd_cke_nxt : in std_logic;
18         sd_cs_nxt  : in std_logic;
19         sd_ras_nxt : in std_logic;
20         sd_cas_nxt : in std_logic;
21         sd_we_nxt  : in std_logic;
22         sd_ba_nxt  : in std_logic_vector(1 downto 0);
23         sd_a_nxt   : in std_logic_vector(12 downto 0);
24
25         wren_nxt   : in std_logic;
26         wdata_nxt  : in std_logic_vector(31 downto 0);
27
28         rden_nxt   : in std_logic;
29         rdata      : out std_logic_vector(31 downto 0);
30         rstb       : out std_logic;
31
32         sd_ck_p    : out std_logic;
33         sd_ck_n    : out std_logic;
34         sd_cke     : out std_logic;
35         sd_cs      : out std_logic;
36         sd_ras     : out std_logic;
37         sd_cas     : out std_logic;
38         sd_we      : out std_logic;
39         sd_ba      : out std_logic_vector(1 downto 0);
40         sd_a       : out std_logic_vector(12 downto 0);
41         sd_ldqs    : out std_logic;
42         sd_udqs    : out std_logic;
43         sd_dq      : inout std_logic_vector(15 downto 0));
44
45 end sdram_ctrl_phy_s3esk;

```

Source file: mem/sdram/sdram_ctrl_phy_s3esk.vhdl

7.12 PoC.misc

The namespace `PoC.misc` offers different yet uncategorized entities.

Sub-Namespaces

- *PoC.misc.filter* contains 1-bit filter algorithms.
- *PoC.misc.stat* contains statistic modules.
- *PoC.misc.sync* offers clock-domain-crossing (CDC) modules.

Package

The package *PoC.misc* holds all component declarations for this namespace.

Entities

- *PoC.misc.Delay*
- *PoC.misc.FrequencyMeasurement*
- *PoC.misc.PulseTrain*
- *PoC.misc.Sequencer*
- *PoC.misc.StrobeGenerator*
- *PoC.misc.StrobeLimiter*
- IP:misc_WordAligner

7.12.1 PoC.misc.filter

These are filter entities. . . .

Entities

- *PoC.misc.filter.and*
- *PoC.misc.filter.mean*
- *PoC.misc.filter.or*

PoC.misc.filter.and

Todo: No documentation available.

Entity Declaration:

```

1  entity filter_and is
2      generic (
3          TAPS           : positive      := 4;          --
4          INIT           : std_logic     := '0';        --
5          ADD_OUTPUT_REG : boolean       := FALSE      --
6      );
7      port (
8          Clock           : in  std_logic;               -- clock
9          DataIn          : in  std_logic;               -- data to filter
10         DataOut         : out std_logic                -- filtered signal
11     );
12 end entity;
```

Source file: misc/filter/filter_and.vhdl

PoC.misc.filter.mean

Todo: No documentation available.

Entity Declaration:

```

1 entity filter_mean is
2   generic (
3     TAPS          : positive      := 4;      --
4     INIT          : std_logic     := '1';    --
5     ADD_OUTPUT_REG : boolean      := FALSE   --
6   );
7   port (
8     Clock         : in  std_logic;    -- clock
9     DataIn        : in  std_logic;    -- data to filter
10    DataOut       : out std_logic     -- filtered signal
11  );
12 end entity;

```

Source file: `misc/filter/filter_mean.vhdl`

PoC.misc.filter.or

Todo: No documentation available.

Entity Declaration:

```

1 entity filter_or is
2   generic (
3     TAPS          : positive      := 4;      --
4     INIT          : std_logic     := '1';    --
5     ADD_OUTPUT_REG : boolean      := FALSE   --
6   );
7   port (
8     Clock         : in  std_logic;    -- clock
9     DataIn        : in  std_logic;    -- data to filter
10    DataOut       : out std_logic     -- filtered signal
11  );
12 end entity;

```

Source file: `misc/filter/filter_or.vhdl`

7.12.2 PoC.misc.gearbox

These are gearbox entities. . .

Entities

- *PoC.misc.gearbox.down_cc*
- *PoC.misc.gearbox.down_dc*
- *PoC.misc.gearbox.up_cc*
- *PoC.misc.gearbox.up_dc*

PoC.misc.gearbox.down_cc

This module provides a downscaling gearbox with a common clock (cc) interface. It performs a ‘word’ to ‘byte’ splitting. The default order is LITTLE_ENDIAN (starting at byte(0)). Input “In_Data” and output

“Out_Data” are of the same clock domain “Clock”. Optional input and output registers can be added by enabling (ADD_***PUT_REGISTERS = TRUE).

Entity Declaration:

```

1 entity gearbox_down_cc is
2   generic (
3     INPUT_BITS           : positive := 32;
4     OUTPUT_BITS          : positive := 24;
5     META_BITS            : natural  := 0;
6     ADD_INPUT_REGISTERS  : boolean  := FALSE;
7     ADD_OUTPUT_REGISTERS : boolean  := FALSE
8   );
9   port (
10    Clock      : in  std_logic;
11
12    In_Sync     : in  std_logic;
13    In_Valid    : in  std_logic;
14    In_Next     : out std_logic;
15    In_Data     : in  std_logic_vector (INPUT_BITS - 1 downto 0);
16    In_Meta     : in  std_logic_vector (META_BITS - 1 downto 0);
17
18    Out_Sync    : out std_logic;
19    Out_Valid   : out std_logic;
20    Out_Data    : out std_logic_vector (OUTPUT_BITS - 1 downto 0);
21    Out_Meta    : out std_logic_vector (META_BITS - 1 downto 0);
22    Out_First   : out std_logic;
23    Out_Last    : out std_logic
24  );
25 end entity;

```

Source file: misc/gearbox/gearbox_down_cc.vhdl

PoC.misc.gearbox.down_dc

This module provides a downscaling gearbox with a dependent clock (dc) interface. It performs a ‘word’ to ‘byte’ splitting. The default order is LITTLE_ENDIAN (starting at byte(0)). Input “In_Data” is of clock domain “Clock1”; output “Out_Data” is of clock domain “Clock2”. Optional input and output registers can be added by enabling (ADD_***PUT_REGISTERS = TRUE).

Assertions:

- Clock periods of Clock1 and Clock2 MUST be multiples of each other.
- Clock1 and Clock2 MUST be phase aligned (related) to each other.

Entity Declaration:

```

1 entity gearbox_down_dc is
2   generic (
3     INPUT_BITS           : positive := 32;
4     OUTPUT_BITS          : positive := 8;
5     OUTPUT_ORDER         : T_BIT_ORDER := LSB_FIRST;
6     FIRST: start at byte(0), MSB_FIRST: start at byte(n-1)
7   );
8   port (
9     In_Data              : in  std_logic_vector (INPUT_BITS - 1 downto 0);
10    In_Meta               : in  std_logic_vector (META_BITS - 1 downto 0);
11    In_Valid              : in  std_logic;
12    In_Sync               : in  std_logic;
13    Out_Data              : out std_logic_vector (OUTPUT_BITS - 1 downto 0);
14    Out_Meta              : out std_logic_vector (META_BITS - 1 downto 0);
15    Out_Valid             : out std_logic;
16    Out_Sync              : out std_logic;
17    Out_First             : out std_logic;
18    Out_Last              : out std_logic
19  );
20 end entity;

```

(continues on next page)

(continued from previous page)

```

6   ADD_INPUT_REGISTERS    : boolean           := FALSE;           -- add_
   ↪input register @Clock1
7   ADD_OUTPUT_REGISTERS  : boolean           := FALSE           -- add_
   ↪output register @Clock2
8   );
9   port (
10    Clock1                : in  std_logic;           --_
   ↪input clock domain
11    Clock2                : in  std_logic;           --_
   ↪output clock domain
12    In_Data               : in  std_logic_vector(INPUT_BITS - 1 downto 0); --_
   ↪input word
13    Out_Data              : out std_logic_vector(OUTPUT_BITS - 1 downto 0) --_
   ↪output word
14   );
15 end entity;

```

Source file: `misc/gearbox/gearbox_down_dc.vhdl`

PoC.misc.gearbox.up_cc

This module provides a downscaling gearbox with a common clock (cc) interface. It performs a ‘byte’ to ‘word’ collection. The default order is LITTLE_ENDIAN (starting at byte(0)). Input “In_Data” and output “Out_Data” are of the same clock domain “Clock”. Optional input and output registers can be added by enabling (ADD_***PUT_REGISTERS = TRUE).

Entity Declaration:

```

1 entity gearbox_up_cc is
2   generic (
3     INPUT_BITS           : positive := 24;
4     OUTPUT_BITS          : positive := 32;
5     META_BITS            : natural  := 0;
6     ADD_INPUT_REGISTERS  : boolean  := FALSE;
7     ADD_OUTPUT_REGISTERS : boolean  := FALSE
8   );
9   port (
10    Clock                : in  std_logic;
11
12    In_Sync              : in  std_logic;
13    In_Valid             : in  std_logic;
14    In_Data              : in  std_logic_vector(INPUT_BITS - 1 downto 0);
15    In_Meta              : in  std_logic_vector(META_BITS - 1 downto 0);
16
17    Out_Sync             : out std_logic;
18    Out_Valid            : out std_logic;
19    Out_Data             : out std_logic_vector(OUTPUT_BITS - 1 downto 0);
20    Out_Meta             : out std_logic_vector(META_BITS - 1 downto 0);
21    Out_First            : out std_logic;
22    Out_Last             : out std_logic
23   );
24 end entity;

```

Source file: `misc/gearbox/gearbox_up_cc.vhdl`

PoC.misc.gearbox.up_dc

This module provides a upscaling gearbox with a dependent clock (dc) interface. It performs a ‘byte’ to ‘word’ collection. The default order is LITTLE_ENDIAN (starting at byte(0)). Input “In_Data” is of clock domain “Clock1”; output “Out_Data” is of clock domain “Clock2”. The “In_Align” is required to mark the starting byte in the word. An optional input register can be added by enabling (ADD_INPUT_REGISTERS = TRUE).

Assertions:

- Clock periods of Clock1 and Clock2 MUST be multiples of each other.
- Clock1 and Clock2 MUST be phase aligned (related) to each other.

Entity Declaration:

```
1 entity gearbox_up_dc is
2   generic (
3     INPUT_BITS          : positive      := 8;                --
    ↪input bit width
4     INPUT_ORDER         : T_BIT_ORDER   := LSB_FIRST;       -- LSB_
    ↪FIRST: start at byte(0), MSB_FIRST: start at byte(n-1)
5     OUTPUT_BITS         : positive      := 32;              --
    ↪output bit width
6     ADD_INPUT_REGISTERS : boolean        := FALSE            -- add_
    ↪input register @Clock1
7   );
8   port (
9     Clock1              : in  std_logic;                    --
    ↪input clock domain
10    Clock2               : in  std_logic;                    --
    ↪output clock domain
11    In_Align             : in  std_logic;                    --
    ↪align word (one cycle high impulse)
12    In_Data              : in  std_logic_vector(INPUT_BITS - 1 downto 0); --
    ↪input word
13    Out_Data             : out std_logic_vector(OUTPUT_BITS - 1 downto 0); --
    ↪output word
14    Out_Valid            : out std_logic                      --
    ↪output is valid
15  );
16 end entity;
```

Source file: [misc/gearbox/gearbox_up_dc.vhdl](#)

7.12.3 PoC.misc.stat

These are stat entities. . . .

Entities

- *PoC.misc.stat.Average*
- *PoC.misc.stat.Histogram*
- *PoC.misc.stat.Maximum*
- *PoC.misc.stat.Minimum*

PoC.misc.stat.Average

Todo: No documentation available.

Entity Declaration:

```

1 entity stat_Average is
2   generic (
3     DATA_BITS      : positive      := 8;
4     COUNTER_BITS    : positive      := 16
5   );
6   port (
7     Clock           : in  std_logic;
8     Reset           : in  std_logic;
9
10    Enable           : in  std_logic;
11    DataIn           : in  std_logic_vector(DATA_BITS - 1 downto 0);
12
13    Count            : out std_logic_vector(COUNTER_BITS - 1 downto 0);
14    Sum              : out std_logic_vector(COUNTER_BITS - 1 downto 0);
15    Average          : out std_logic_vector(COUNTER_BITS - 1 downto 0);
16    Valid            : out std_logic
17  );
18 end entity;
```

Source file: [misc/stat/stat_Average.vhdl](#)

PoC.misc.stat.Histogram

Todo: No documentation available.

Entity Declaration:

```

1 entity stat_Histogram is
2   generic (
3     DATA_BITS      : positive      := 16;
4     COUNTER_BITS    : positive      := 16
5   );
6   port (
7     Clock           : in  std_logic;
8     Reset           : in  std_logic;
9
10    Enable           : in  std_logic;
11    DataIn           : in  std_logic_vector(DATA_BITS - 1 downto 0);
12
13    Histogram        : out T_SLM(2*DATA_BITS - 1 downto 0, COUNTER_BITS - 1 downto 0)
14  );
15 end entity;
```

Source file: [misc/stat/stat_Histogram.vhdl](#)

PoC.misc.stat.Maximum

Todo: No documentation available.

Entity Declaration:

```
1 entity stat_Maximum is
2   generic (
3     DEPTH          : positive      := 8;
4     DATA_BITS     : positive      := 16;
5     COUNTER_BITS   : positive      := 16
6   );
7   port (
8     Clock          : in  std_logic;
9     Reset          : in  std_logic;
10
11     Enable         : in  std_logic;
12     DataIn         : in  std_logic_vector (DATA_BITS - 1 downto 0);
13
14     Valid          : out std_logic_vector (DEPTH - 1 downto 0);
15     Maximums       : out T_SLM (DEPTH - 1 downto 0, DATA_BITS - 1 downto 0);
16     Counts         : out T_SLM (DEPTH - 1 downto 0, COUNTER_BITS - 1 downto 0)
17   );
18 end entity;
```

Source file: [misc/stat/stat_Maximum.vhdl](#)

PoC.misc.stat.Minimum

Todo: No documentation available.

Entity Declaration:

```
1 entity stat_Minimum is
2   generic (
3     DEPTH          : positive      := 8;
4     DATA_BITS     : positive      := 16;
5     COUNTER_BITS   : positive      := 16
6   );
7   port (
8     Clock          : in  std_logic;
9     Reset          : in  std_logic;
10
11     Enable         : in  std_logic;
12     DataIn         : in  std_logic_vector (DATA_BITS - 1 downto 0);
13
14     Valid          : out std_logic_vector (DEPTH - 1 downto 0);
15     Minimums       : out T_SLM (DEPTH - 1 downto 0, DATA_BITS - 1 downto 0);
16     Counts         : out T_SLM (DEPTH - 1 downto 0, COUNTER_BITS - 1 downto 0)
17   );
18 end entity;
```

Source file: [misc/stat/stat_Minimum.vhdl](#)

7.12.4 PoC.misc.sync

The namespace `PoC.misc.sync` offers different clock-domain-crossing (CDC) synchronizer circuits. All synchronizers are based on the basic 2 flip-flop synchronizer called *sync_Bits*. PoC has two platform specific implementations for Altera and Xilinx, which are choosen, if the appropriate `MY_DEVICE` constant is configured in `my_config.vhdl`.

Decision Table:

Behavior	Flag ¹	Strobe ²	Continuous Data	Reset ⁴	Pulse ³
1 Bit	<i>sync_Bits</i>	<i>sync_Strobe</i>	<i>fifo_ic_got</i> ⁵	<i>sync_Reset</i>	<i>sync_Pulse</i>
n Bit	<i>sync_Vector</i>	<i>sync_Command</i>	<i>fifo_ic_got</i> ⁵		

Basic 2 Flip-Flop Synchronizer

The basic 2 flip-flop synchronizer is called *sync_Bits*. It's possible to configure the bit count of indivital bits. If a vector shall be synchronized, use one of the special synchronizers like *sync_Vector*. The vendor specific implementations are named *sync_Bits_Altera* and *sync_Bits_Xilinx* respectively.

A second variant of the 2-FF synchronizer is called *sync_Reset*. It's for Reset-signals, implementing asynchronous assertion and synchronous deassertion. The vendor specific implementations are named *sync_Reset_Altera* and *sync_Reset_Xilinx* respectively.

A third variant of a 2-FF synchronizer is called *sync_Pulse*. It's for very short Pulsed-signals. It uses an addition asynchronous capture FF to latch the very short pulse. The vendor specific implementations are named *sync_Pulse_Altera* and *sync_Pulse_Xilinx* respectively.

Special Synchronizers

Based on the 2-FF synchronizer, several “high-level” synchronizers are build.

- *sync_Strobe* synchronizer *strobe*-signals across clock-domain-boundaries. A busy signal indicates the synchronization status and can be used as a internal gate-signal to disallow new incoming strobes. A *strobe*-signal is only for one clock period active.
- *sync_Command* like *sync_Strobe*, it synchronizes a one clock period active signal across the clock-domain-boundary, but the input has multiple bits. After the multi bit strobe (Command) was transfered, the output goes to its idle value.
- *sync_Vector* synchronizes a complete vector across the clock-domain-boundary. A changed detection on the input vector causes a register to latch the current state. The changed event is transfered to the new clock-domain and triggers a register to store the latched content, but in the new clock domain.

See also:

PoC.fifo.ic_got For a cross-clock capable FIFO.

PoC.misc.sync Package

Source file: *sync.pkg.vhdl*

¹ A *flag* or *status* signal is a continuous, long time stable signal.

² A *strobe* signal is active for only one cycle.

⁴ To be refumented

³ A *pulse* signal is a very short event.

⁵ See the `PoC.fifo` namespace for cross-clock capable FIFOs.

PoC.misc.sync.Bits

This module synchronizes multiple flag bits into clock-domain `ClOCK`. The clock-domain boundary crossing is done by two synchronizer D-FFs. All bits are independent from each other. If a known vendor like Altera or Xilinx are recognized, a vendor specific implementation is chosen.

Attention: Use this synchronizer only for long time stable signals (flags).

Constraints:

General: Please add constraints for meta stability to all ‘_meta’ signals and timing ignore constraints to all ‘_async’ signals.

Xilinx: In case of a Xilinx device, this module will instantiate the optimized module `PoC.xil.sync.Bits`. Please attend to the notes of `sync_Bits.vhdl`.

Altera sdc file: TODO

Entity Declaration:

```
1 entity sync_Bits is
2   generic (
3     BITS          : positive          := 1;           -- number of_
↳ bit to be synchronized
4     INIT          : std_logic_vector := x"00000000";  --_
↳ initialization bits
5     SYNC_DEPTH    : T_MISC_SYNC_DEPTH := T_MISC_SYNC_DEPTH'low -- generate_
↳ SYNC_DEPTH many stages, at least 2
6   );
7   port (
8     Clock         : in  std_logic;      -- <Clock> _
↳ output clock domain
9     Input         : in  std_logic_vector (BITS - 1 downto 0); -- @async: _
↳ input bits
10    Output        : out std_logic_vector (BITS - 1 downto 0)  -- @Clock: _
↳ output bits
11  );
12 end entity;
```

See also:

PoC.misc.sync.Reset For a special 2 D-FF synchronizer for *reset*-signals.

PoC.misc.sync.Pulse For a special 1+2 D-FF synchronizer for *pulse*-signals.

PoC.misc.sync.Strobe For a synchronizer for *strobe*-signals.

PoC.misc.sync.Vector For a multiple bits capable synchronizer.

Source file: `misc/sync/sync_Bits.vhdl`

PoC.misc.sync.Command

This module synchronizes a vector of bits from clock-domain `ClOCK1` to clock-domain `ClOCK2`. The clock-domain boundary crossing is done by a change comparator, a T-FF, two synchronizer D-FFs and a reconstructive XOR indicating a value change on the input. This changed signal is used to capture the input for the new output. A busy flag is additionally calculated for the input clock-domain. The output has strobe character and is reset to it's `INIT` value after one clock cycle.

Constraints: This module uses sub modules which need to be constrained. Please attend to the notes of the instantiated sub modules.

Entity Declaration:

```

1 entity sync_Command is
2   generic (
3     BITS          : positive          := 8;          -- number of_
    ↳bit to be synchronized
4     INIT          : std_logic_vector  := x"00000000";  --
5     SYNC_DEPTH    : T_MISC_SYNC_DEPTH := T_MISC_SYNC_DEPTH'low -- generate_
    ↳SYNC_DEPTH many stages, at least 2
6   );
7   port (
8     Clock1        : in  std_logic;      -- <Clock> _
    ↳input clock
9     Clock2        : in  std_logic;      -- <Clock> _
    ↳output clock
10    Input          : in  std_logic_vector(BITS - 1 downto 0); -- @Clock1:_
    ↳input vector
11    Output         : out std_logic_vector(BITS - 1 downto 0); -- @Clock2:_
    ↳output vector
12    Busy          : out std_logic;      -- @Clock1:_
    ↳busy bit
13    Changed        : out std_logic      -- @Clock2:_
    ↳changed bit
14  );
15 end entity;
```

Source file: misc/sync/sync_Command.vhdl

PoC.misc.sync.Pulse

This module synchronizes multiple pulsed bits into the clock-domain `Clock`. The clock-domain boundary crossing is done by two synchronizer D-FFs. All bits are independent from each other. If a known vendor like Altera or Xilinx are recognized, a vendor specific implementation is chosen.

Attention: Use this synchronizer for very short signals (pulse).

Constraints:

General: Please add constraints for meta stability to all ‘_meta’ signals and timing ignore constraints to all ‘_async’ signals.

Xilinx: In case of a Xilinx device, this module will instantiate the optimized module `PoC.xil.sync.Pulse`. Please attend to the notes of `sync_Bits.vhdl`.

Altera sdc file: TODO

Entity Declaration:

```

1 entity sync_Pulse is
2   generic (
3     BITS          : positive          := 1;          -- number of_
    ↳bit to be synchronized
4     SYNC_DEPTH    : T_MISC_SYNC_DEPTH := T_MISC_SYNC_DEPTH'low -- generate_
    ↳SYNC_DEPTH many stages, at least 2
5   );
6   port (
7     Clock         : in  std_logic;      -- <Clock> _
    ↳output clock domain
```

(continues on next page)

(continued from previous page)

```

8   Input      : in std_logic_vector (BITS - 1 downto 0);      -- @async: ␣
↪input bits
9   Output     : out std_logic_vector (BITS - 1 downto 0)      -- @Clock: ␣
↪output bits
10  );
11  end entity;
```

See also:*PoC.misc.sync.Bits* For a common 2 D-FF synchronizer for *flag*-signals.*PoC.misc.sync.Reset* For a special 2 D-FF synchronizer for *reset*-signals.*PoC.misc.sync.Strobe* For a synchronizer for *strobe*-signals.*PoC.misc.sync.Vector* For a multiple bits capable synchronizer.Source file: `misc/sync/sync_Pulse.vhdl`**PoC.misc.sync.Reset**

This module synchronizes an asynchronous reset signal to the clock `Clock`. The `Input` can be asserted and de-asserted at any time. The `Output` is asserted asynchronously and de-asserted synchronously to the clock.

Attention: Use this synchronizer only to asynchronously reset your design. The ‘Output’ should be feed by global buffer to the destination FFs, so that, it reaches their reset inputs within one clock cycle.

Constraints:

General: Please add constraints for meta stability to all ‘_meta’ signals and timing ignore constraints to all ‘_async’ signals.

Xilinx: In case of a Xilinx device, this module will instantiate the optimized module `xil_SyncReset`. Please attend to the notes of `xil_SyncReset`.

Altera sdc file: TODO

Entity Declaration:

```

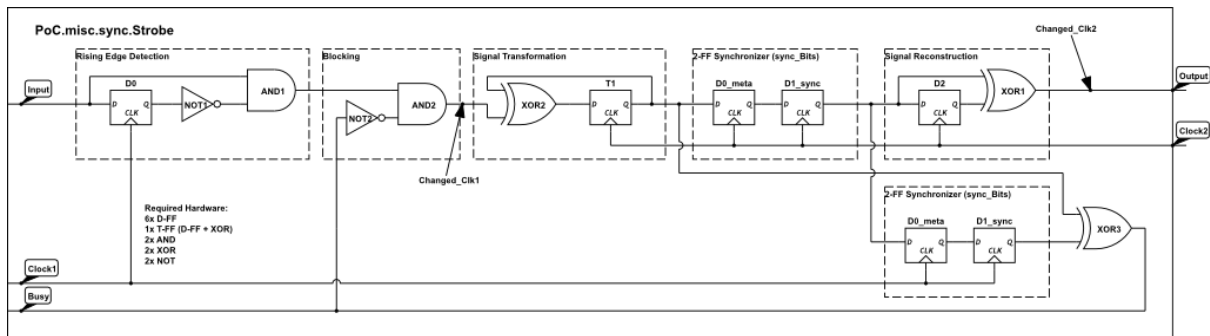
1  entity sync_Reset is
2    generic (
3      SYNC_DEPTH      : T_MISC_SYNC_DEPTH      := T_MISC_SYNC_DEPTH'low      -- generate_
↪SYNC_DEPTH many stages, at least 2
4    );
5    port (
6      Clock           : in std_logic;          -- <Clock> ␣
↪output clock domain
7      Input          : in std_logic;          -- @async: ␣
↪reset input
8      Output         : out std_logic          -- @Clock: ␣
↪reset output
9    );
10  end entity;
```

Source file: `misc/sync/sync_Reset.vhdl`

PoC.misc.sync.Strobe

This module synchronizes multiple high-active bits from clock-domain `Clock1` to clock-domain `Clock2`. The clock-domain boundary crossing is done by a T-FF, two synchronizer D-FFs and a reconstructive XOR. A busy flag is additionally calculated and can be used to block new inputs. All bits are independent from each other. Multiple consecutive strobes are suppressed by a rising edge detection.

Attention: Use this synchronizer only for one-cycle high-active signals (strobes).



Constraints: This module uses sub modules which need to be constrained. Please attend to the notes of the instantiated sub modules.

Entity Declaration:

```

1  entity sync_Strobe is
2      generic (
3          BITS                : positive                := 1;
4          ↪number of bit to be synchronized
5          GATED_INPUT_BY_BUSY : boolean                 := TRUE;
6          ↪gated input (by busy signal)
7          SYNC_DEPTH          : T_MISC_SYNC_DEPTH      := T_MISC_SYNC_DEPTH'low
8          ↪generate SYNC_DEPTH many stages, at least 2
9      );
10     port (
11         Clock1                : in  std_logic;
12         ↪input clock domain
13         Clock2                : in  std_logic;
14         ↪output clock domain
15         Input                 : in  std_logic_vector(BITS - 1 downto 0);
16         ↪input bits
17         Output                : out std_logic_vector(BITS - 1 downto 0);
18         ↪output bits
19         Busy                  : out std_logic_vector(BITS - 1 downto 0);
20         ↪busy bits
21     );
22 end entity;

```

Source file: `misc/sync/sync_Strobe.vhdl`

PoC.misc.sync.Vector

This module synchronizes a vector of bits from clock-domain `Clock1` to clock-domain `Clock2`. The clock-domain boundary crossing is done by a change comparator, a T-FF, two synchronizer D-FFs and a reconstructive XOR indicating a value change on the input. This changed signal is used to capture the input for the new output. A busy flag is additionally calculated for the input clock domain.

Constraints: This module uses sub modules which need to be constrained. Please attend to the notes of the instantiated sub modules.

Entity Declaration:

```

1 entity sync_Vector is
2   generic (
3     MASTER_BITS    : positive           := 8;           -- number of_
↳ bit to be synchronized
4     SLAVE_BITS     : natural            := 0;
5     INIT           : std_logic_vector   := x"00000000";   --
6     SYNC_DEPTH     : T_MISC_SYNC_DEPTH := T_MISC_SYNC_DEPTH'low -- generate_
↳ SYNC_DEPTH many stages, at least 2
7   );
8   port (
9     Clock1         : in  std_logic;
↳ -- <Clock> input clock
10    Clock2         : in  std_logic;
↳ -- <Clock> output clock
11    Input          : in  std_logic_vector (MASTER_BITS + SLAVE_BITS) - 1 downto 0);
↳ -- @Clock1: input vector
12    Output         : out std_logic_vector (MASTER_BITS + SLAVE_BITS) - 1 downto 0);
↳ -- @Clock2: output vector
13    Busy           : out std_logic;
↳ -- @Clock1: busy bit
14    Changed        : out std_logic
↳ -- @Clock2: changed bit
15  );
16 end entity;
```

Source file: misc/sync/sync_Vector.vhdl

7.12.5 PoC.misc Package

This package holds all component declarations for this namespace.

Source file: misc.pkg.vhdl

7.12.6 PoC.misc.Delay

Todo: No documentation available.

Entity Declaration:

```

1 entity misc_Delay is
2   generic (
3     BITS           : positive;
4     TAPS           : T_NATIVEC         -- select one or multiple delay tap points
5   );
6   port (
7     Clock          : in  std_logic;
↳ clock
8     Reset          : in  std_logic := '0';
↳ reset; avoid reset to enable SRL16/SRL32 usage
```

(continues on next page)

(continued from previous page)

```

9      Enable      : in std_logic := '1';
↳enable
10     DataIn      : in std_logic_vector (BITS - 1 downto 0);
↳data to delay
11     DataOut     : out T_SLM(TAPS'length - 1 downto 0, BITS - 1 downto 0)
↳delayed ouputs, tapped at TAPS(i)
12     );
13 end entity;
```

Source file: misc/misc_Delay.vhdl

7.12.7 PoC.misc.FrequencyMeasurement

This module counts 1 second in a reference timer at reference clock. This reference time is used to start and stop a timer at input clock. The counter value is the measured frequency in Hz.

Entity Declaration:

```

1 entity misc_FrequencyMeasurement is
2   generic (
3     REFERENCE_CLOCK_FREQ : FREQ      := 100 MHz
4   );
5   port (
6     Reference_Clock : in std_logic;
7     Input_Clock    : in std_logic;
8
9     Start          : in std_logic;
10    Done            : out std_logic;
11    Result          : out T_SLV_32
12  );
13 end entity;
```

Source file: misc/misc_FrequencyMeasurement.vhdl

7.12.8 PoC.misc.PulseTrain

This module generates pulse trains. This module was written as a answer for a StackOverflow question: <http://stackoverflow.com/questions/25783320>

Entity Declaration:

Source file: misc/misc_PulseTrain.vhdl

7.12.9 PoC.misc.Sequencer

Todo: No documentation available.

Entity Declaration:

Source file: misc/misc_Sequencer.vhdl

7.12.10 PoC.misc.StrobeGenerator

Todo: No documentation available.

Entity Declaration:

Source file: `misc/misc_StrobeGenerator.vhdl`

7.12.11 PoC.misc.StrobeLimiter

Todo: No documentation available.

Entity Declaration:

Source file: `misc/misc_StrobeLimiter.vhdl`

7.12.12 WordAligner

Todo: No documentation available.

Entity Declaration:

Source file: `misc/misc_WordAligner.vhdl`

7.13 PoC.net

These are bus entities...

Sub-Namespaces

- *PoC.net.arp*
- *PoC.net.eth*
- *PoC.net.icmpv4*
- *PoC.net.icmpv6*
- *PoC.net.ipv4*
- *PoC.net.ipv6*
- *PoC.net.mac*
- *PoC.net.ndp*
- *PoC.net.stack*
- *PoC.net.udp*

Entities

- *PoC.net.FrameChecksum*
- IP:net_FrameLoopback

7.13.1 PoC.net.arp

These are ARP entities....

PoC.net.arp.Broadcast_Receiver

Todo: No documentation available.

Entity Declaration:

```

1  entity arp_BroadCast_Receiver is
2      generic (
3          ALLOWED_PROTOCOL_IPV4      : boolean           := TRUE;
4          ALLOWED_PROTOCOL_IPV6      : boolean           := FALSE;
5      );
6      port (
7          Clock                       : in  std_logic;
8          Reset                       : in  std_logic;
9
10         RX_Valid                    : in  std_logic;
11         RX_Data                     : in  T_SLV_8;
12         RX_SOF                      : in  std_logic;
13         RX_EOF                      : in  std_logic;
14         RX_Ack                      : out std_logic;
15         RX_Meta_rst                 : out std_logic;
16         RX_Meta_SrcMACAddress_nxt   : out std_logic;
17         RX_Meta_SrcMACAddress_Data : in  T_SLV_8;
18         RX_Meta_DestMACAddress_nxt  : out std_logic;
19         RX_Meta_DestMACAddress_Data : in  T_SLV_8;
20
21         Clear                       : in  std_logic;
22         Error                       : out std_logic;
23
24         RequestReceived              : out std_logic;
25         Address_rst                  : in  std_logic;
26         SenderMACAddress_nxt         : in  std_logic;
27         SenderMACAddress_Data        : out T_SLV_8;
28         SenderIPAddress_nxt          : in  std_logic;
29         SenderIPAddress_Data         : out T_SLV_8;
30         TargetIPAddress_nxt          : in  std_logic;
31         TargetIPAddress_Data         : out T_SLV_8;
32     );
33 end entity;
```

Source file: net/arp/arp_BroadCast_Receiver.vhdl

PoC.net.arp.Broadcast_Requester

Todo: No documentation available.

Entity Declaration:

```

1  entity arp_BroadCast_Requester is
2      generic (
3          ALLOWED_PROTOCOL_IPV4      : boolean           := TRUE;
4          ALLOWED_PROTOCOL_IPV6      : boolean           := FALSE;
5      );
6      port (
7          Clock                      : in  std_logic;
8          Reset                      : in  std_logic;
9
10         SendRequest                : in  std_logic;
11         Complete                   : out std_logic;
12
13         Address_rst                 : out std_logic;
14         SenderMACAddress_nxt        : out std_logic;
15         SenderMACAddress_Data       : in  T_SLV_8;
16         SenderIPv4Address_nxt       : out std_logic;
17         SenderIPv4Address_Data      : in  T_SLV_8;
18         TargetMACAddress_nxt        : out std_logic;
19         TargetMACAddress_Data       : in  T_SLV_8;
20         TargetIPv4Address_nxt       : out std_logic;
21         TargetIPv4Address_Data      : in  T_SLV_8;
22
23         TX_Valid                    : out std_logic;
24         TX_Data                     : out T_SLV_8;
25         TX_SOF                      : out std_logic;
26         TX_EOF                      : out std_logic;
27         TX_Ack                      : in  std_logic;
28         TX_Meta_DestMACAddress_rst  : in  std_logic;
29         TX_Meta_DestMACAddress_nxt  : in  std_logic;
30         TX_Meta_DestMACAddress_Data : out T_SLV_8;
31     );
32 end entity;

```

Source file: net/arp/arp_BroadCast_Requester.vhdl

PoC.net.arp.Cache

Todo: No documentation available.

Entity Declaration:

```

1  entity arp_Cache is
2      generic (
3          CLOCK_FREQ                : FREQ              := 125 MHz;
4          REPLACEMENT_POLICY         : string            := "LRU";
5          TAG_BYTE_ORDER              : T_BYTE_ORDER     := BIG_
6          ↪ ENDIAN;
7          DATA_BYTE_ORDER            : T_BYTE_ORDER     := BIG_
8          ↪ ENDIAN;

```

(continues on next page)

(continued from previous page)

```

7     INITIAL_CACHE_CONTENT      : T_NET_ARP_ARPCACHE_VECTOR
8 );
9 port (
10     Clock                      : in  std_logic;           --
11     Reset                     : in  std_logic;           --
12
13     Command                   : in  T_NET_ARP_ARPCACHE_COMMAND;
14     Status                   : out  T_NET_ARP_ARPCACHE_STATUS;
15     NewIPv4Address_rst       : out  std_logic;
16     NewIPv4Address_nxt       : out  std_logic;
17     NewIPv4Address_Data      : in   T_SLV_8;
18     NewMACAddress_rst        : out  std_logic;
19     NewMACAddress_nxt        : out  std_logic;
20     NewMACAddress_Data       : in   T_SLV_8;
21
22     Lookup                   : in  std_logic;
23     IPv4Address_rst          : out  std_logic;
24     IPv4Address_nxt          : out  std_logic;
25     IPv4Address_Data         : in   T_SLV_8;
26
27     CacheResult              : out  T_CACHE_RESULT;
28     MACAddress_rst           : in  std_logic;
29     MACAddress_nxt           : in  std_logic;
30     MACAddress_Data          : out  T_SLV_8;
31 );
32 end entity;
```

Source file: [net/arp/arp_Cache.vhdl](#)

PoC.net.arp.IPPool

Todo: No documentation available.

Entity Declaration:

```

1 entity arp_IPPool is
2     generic (
3         IPPool_SIZE           : positive;
4         INITIAL_IPV4_ADDRESSES : T_NET_IPV4_ADDRESS_VECTOR := (0 to 7 => C_
↳ NET_IPV4_ADDRESS_EMPTY)
5     );
6     port (
7         Clock                 : in  std_logic;           ↳
↳ --
8         Reset                 : in  std_logic;           ↳
↳ --
9
10    -- Command                 : in  T_ETHERNET_ARP_IPPOOL_COMMAND;
11    -- IPv4Address              : in  T_NET_IPV4_ADDRESS;
12    -- MACAddress               : in  T_ETHERNET_MAC_ADDRESS;
13
14    Lookup                     : in  std_logic;
15    IPv4Address_rst            : out  std_logic;
16    IPv4Address_nxt            : out  std_logic;
17    IPv4Address_Data           : in   T_SLV_8;
18
```

(continues on next page)

(continued from previous page)

```

19     PoolResult                : out T_CACHE_RESULT
20 );
21 end entity;
```

Source file: [net/arp/arp_IPPool.vhdl](#)

PoC.net.arp.Tester

Todo: No documentation available.

Entity Declaration:

Source file: [net/arp/arp_Tester.vhdl](#)

PoC.net.arp.UniCast_Receiver

Todo: No documentation available.

Entity Declaration:

```

1 entity arp_UniCast_Receiver is
2   generic (
3     ALLOWED_PROTOCOL_IPV4      : boolean      := TRUE;
4     ALLOWED_PROTOCOL_IPV6      : boolean      := FALSE;
5   );
6   port (
7     Clock                      : in  std_logic;
8     Reset                      : in  std_logic;
9
10    RX_Valid                   : in  std_logic;
11    RX_Data                   : in  T_SLV_8;
12    RX_SOF                    : in  std_logic;
13    RX_EOF                    : in  std_logic;
14    RX_Ack                    : out  std_logic;
15    RX_Meta_rst               : out  std_logic;
16    RX_Meta_SrcMACAddress_nxt : out  std_logic;
17    RX_Meta_SrcMACAddress_Data : in  T_SLV_8;
18    RX_Meta_DestMACAddress_nxt : out  std_logic;
19    RX_Meta_DestMACAddress_Data : in  T_SLV_8;
20
21    Clear                      : in  std_logic;
22    Error                      : out  std_logic;
23
24    ResponseReceived           : out  std_logic;
25    Address_rst                : in  std_logic;
26    SenderMACAddress_nxt       : in  std_logic;
27    SenderMACAddress_Data      : out  T_SLV_8;
28    SenderIPAddress_nxt        : in  std_logic;
29    SenderIPAddress_Data       : out  T_SLV_8;
```

(continues on next page)

(continued from previous page)

```

30     TargetIPAddress_nxt      : in std_logic;
31     TargetIPAddress_Data    : out T_SLV_8;
32     TargetMACAddress_nxt    : in std_logic;
33     TargetMACAddress_Data    : out T_SLV_8
34 );
35 end entity;
```

Source file: [net/arp/arp_UniCast_Receiver.vhdl](#)

PoC.net.arp.UniCast_Responder

Todo: No documentation available.

Entity Declaration:

```

1  entity arp_UniCast_Responder is
2      generic (
3          ALLOWED_PROTOCOL_IPV4      : boolean           := TRUE;
4          ALLOWED_PROTOCOL_IPV6      : boolean           := FALSE;
5      );
6      port (
7          Clock                      : in std_logic;
8          Reset                      : in std_logic;
9
10         SendResponse               : in std_logic;
11         Complete                   : out std_logic;
12
13         Address_rst                 : out std_logic;
14         SenderMACAddress_nxt        : out std_logic;
15         SenderMACAddress_Data       : in T_SLV_8;
16         SenderIPv4Address_nxt       : out std_logic;
17         SenderIPv4Address_Data      : in T_SLV_8;
18         TargetMACAddress_nxt        : out std_logic;
19         TargetMACAddress_Data       : in T_SLV_8;
20         TargetIPv4Address_nxt       : out std_logic;
21         TargetIPv4Address_Data      : in T_SLV_8;
22
23         TX_Valid                    : out std_logic;
24         TX_Data                     : out T_SLV_8;
25         TX_SOF                      : out std_logic;
26         TX_EOF                      : out std_logic;
27         TX_Ack                      : in std_logic;
28         TX_Meta_DestMACAddress_rst  : in std_logic;
29         TX_Meta_DestMACAddress_nxt  : in std_logic;
30         TX_Meta_DestMACAddress_Data : out T_SLV_8;
31     );
32 end entity;
```

Source file: [net/arp/arp_UniCast_Responder.vhdl](#)

PoC.net.arp.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1  entity arp_Wrapper is
2      generic (
3          CLOCK_FREQ                : FREQ                        := 125 MHz;
4          INTERFACE_MACADDRESS      : T_NET_MAC_ADDRESS          := C_NET_MAC_ADDRESS_EMPTY;
5          INITIAL_IPV4ADDRESSES     : T_NET_IPV4_ADDRESS_VECTOR  := (0 => C_NET_IPV4_ADDRESS_EMPTY);
6          INITIAL_ARPCACHE_CONTENT  : T_NET_ARP_ARPCACHE_VECTOR  := (0 => (Tag => C_NET_IPV4_ADDRESS_EMPTY, MAC => C_NET_MAC_ADDRESS_EMPTY));
7          APR_REQUEST_TIMEOUT       : time                       := 100 ms
8      );
9      port (
10         Clock                     : in  std_logic;
11         Reset                     : in  std_logic;
12
13         IPPool_Announce           : in  std_logic;
14         IPPool_Announced         : out std_logic;
15
16         IPCache_Lookup            : in  std_logic;
17         IPCache_IPv4Address_rst   : out std_logic;
18         IPCache_IPv4Address_nxt   : out std_logic;
19         IPCache_IPv4Address_Data  : in  T_SLV_8;
20
21         IPCache_Valid             : out std_logic;
22         IPCache_MACAddress_rst    : in  std_logic;
23         IPCache_MACAddress_nxt    : in  std_logic;
24         IPCache_MACAddress_Data   : out T_SLV_8;
25
26         Eth_UC_TX_Valid           : out std_logic;
27         Eth_UC_TX_Data            : out T_SLV_8;
28         Eth_UC_TX_SOF             : out std_logic;
29         Eth_UC_TX_EOF             : out std_logic;
30         Eth_UC_TX_Ack             : in  std_logic;
31         Eth_UC_TX_Meta_rst        : in  std_logic;
32         Eth_UC_TX_Meta_DestMACAddress_nxt : in std_logic;
33         Eth_UC_TX_Meta_DestMACAddress_Data : out T_SLV_8;
34
35         Eth_UC_RX_Valid           : in  std_logic;
36         Eth_UC_RX_Data            : in  T_SLV_8;
37         Eth_UC_RX_SOF             : in  std_logic;
38         Eth_UC_RX_EOF             : in  std_logic;
39         Eth_UC_RX_Ack             : out std_logic;
40         Eth_UC_RX_Meta_rst        : out std_logic;
41         Eth_UC_RX_Meta_SrcMACAddress_nxt : out std_logic;
42         Eth_UC_RX_Meta_SrcMACAddress_Data : in  T_SLV_8;
43         Eth_UC_RX_Meta_DestMACAddress_nxt : out std_logic;
44         Eth_UC_RX_Meta_DestMACAddress_Data : in  T_SLV_8;
45
46         Eth_BC_RX_Valid           : in  std_logic;
47         Eth_BC_RX_Data            : in  T_SLV_8;
48         Eth_BC_RX_SOF             : in  std_logic;
49         Eth_BC_RX_EOF             : in  std_logic;
50         Eth_BC_RX_Ack             : out std_logic;

```

(continues on next page)

(continued from previous page)

```
51     Eth_BC_RX_Meta_rst           : out std_logic;
52     Eth_BC_RX_Meta_SrcMACAddress_nxt : out std_logic;
53     Eth_BC_RX_Meta_SrcMACAddress_Data : in  T_SLV_8;
54     Eth_BC_RX_Meta_DestMACAddress_nxt : out std_logic;
55     Eth_BC_RX_Meta_DestMACAddress_Data : in  T_SLV_8
56 );
57 end entity;
```

Source file: [net/arp/arp_Wrapper.vhdl](#)

7.13.2 PoC.net.eth

These are eth entities. . . .

PoC.net.eth.GEMAC_GMII

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_GEMAC_GMII.vhdl](#)

PoC.net.eth.GEMAC_RX

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_GEMAC_RX.vhdl](#)

PoC.net.eth.GEMAC_TX

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_GEMAC_TX.vhdl](#)

PoC.net.eth.PHYController

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_PHYController.vhdl](#)

PoC.net.eth.PHYController_Marvell_88E1111

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_PHYController_Marvell_88E1111.vhdl](#)

PoC.net.eth.Wrapper

Todo: No documentation available.

Entity Declaration:

Source file: [net/eth/eth_Wrapper.vhdl](#)

7.13.3 PoC.net.icmpv4

These are icmpv4 entities....

PoC.net.icmpv4.RX

Todo: No documentation available.

Entity Declaration:

```
1 entity icmpv4_RX is
2   generic (
3     DEBUG                      : boolean                := FALSE
4   );
5   port (
6     Clock                      : in  std_logic;
7     Reset                      : in  std_logic;
8     -- CSE interface
9     Command                    : in  T_NET_ICMPV4_RX_COMMAND;
10    Status                      : out T_NET_ICMPV4_RX_STATUS;
11    Error                      : out T_NET_ICMPV4_RX_ERROR;
12    -- IN port
13    In_Valid                   : in  std_logic;
14    In_Data                    : in  T_SLV_8;
```

(continues on next page)

(continued from previous page)

```

15     In_SOF                                : in  std_logic;
16     In_EOF                                : in  std_logic;
17     In_Ack                                : out std_logic;
18     In_Meta_rst                           : out std_logic;
19     In_Meta_SrcMACAddress_nxt             : out std_logic;
20     In_Meta_SrcMACAddress_Data            : in   T_SLV_8;
21     In_Meta_DestMACAddress_nxt            : out std_logic;
22     In_Meta_DestMACAddress_Data           : in   T_SLV_8;
23     In_Meta_SrcIPv4Address_nxt            : out std_logic;
24     In_Meta_SrcIPv4Address_Data           : in   T_SLV_8;
25     In_Meta_DestIPv4Address_nxt           : out std_logic;
26     In_Meta_DestIPv4Address_Data          : in   T_SLV_8;
27     In_Meta_Length                        : in   T_SLV_16;
28     -- OUT Port
29     Out_Meta_rst                          : in  std_logic;
30     Out_Meta_SrcMACAddress_nxt            : in  std_logic;
31     Out_Meta_SrcMACAddress_Data           : out T_SLV_8;
32     Out_Meta_DestMACAddress_nxt           : in  std_logic;
33     Out_Meta_DestMACAddress_Data          : out T_SLV_8;
34     Out_Meta_SrcIPv4Address_nxt           : in  std_logic;
35     Out_Meta_SrcIPv4Address_Data          : out T_SLV_8;
36     Out_Meta_DestIPv4Address_nxt          : in  std_logic;
37     Out_Meta_DestIPv4Address_Data         : out T_SLV_8;
38     Out_Meta_Length                      : out T_SLV_16;
39     Out_Meta_Type                         : out T_SLV_8;
40     Out_Meta_Code                         : out T_SLV_8;
41     Out_Meta_Identification               : out T_SLV_16;
42     Out_Meta_SequenceNumber               : out T_SLV_16;
43     Out_Meta_Payload_nxt                  : in  std_logic;
44     Out_Meta_Payload_last                 : out std_logic;
45     Out_Meta_Payload_Data                 : out T_SLV_8
46 );
47 end entity;

```

Source file: `net/icmpv4/icmpv4_RX.vhdl`

PoC.net.icmpv4.TX

Todo: No documentation available.

Entity Declaration:

```

1  entity icmpv4_TX is
2      generic (
3          DEBUG                                : boolean                := FALSE;
4          SOURCE_IPV4ADDRESS                   : T_NET_IPV4_ADDRESS      := C_NET_IPV4_
5      ) ADDRESS_EMPTY
6  );
7  port (
8      Clock                                : in  std_logic;
9      --
10     Reset                                : in  std_logic;
11     --
12     -- CSE interface
13     Command                             : in   T_NET_ICMPV4_TX_COMMAND;
14     Status                              : out  T_NET_ICMPV4_TX_STATUS;

```

(continues on next page)

(continued from previous page)

```

12      Error                                : out T_NET_ICMPV4_TX_ERROR;
13      -- OUT port
14      Out_Valid                            : out std_logic;
15      Out_Data                             : out T_SLV_8;
16      Out_SOF                             : out std_logic;
17      Out_EOF                             : out std_logic;
18      Out_Ack                             : in  std_logic;
19      Out_Meta_rst                         : in  std_logic;
20      Out_Meta_SrcIPv4Address_nxt         : in  std_logic;
21      Out_Meta_SrcIPv4Address_Data        : out T_SLV_8;
22      Out_Meta_DestIPv4Address_nxt        : in  std_logic;
23      Out_Meta_DestIPv4Address_Data       : out T_SLV_8;
24      Out_Meta_Length                     : out T_SLV_16;
25      -- IN port
26      In_Meta_rst                         : out std_logic;
27      In_Meta_IPv4Address_nxt             : out std_logic;
28      In_Meta_IPv4Address_Data            : in  T_SLV_8;
29      In_Meta_Type                        : in  T_SLV_8;
30      In_Meta_Code                        : in  T_SLV_8;
31      In_Meta_Identification               : in  T_SLV_16;
32      In_Meta_SequenceNumber              : in  T_SLV_16;
33      In_Meta_Payload_nxt                 : out std_logic;
34      In_Meta_Payload_last                : in  std_logic;
35      In_Meta_Payload_Data                : in  T_SLV_8
36  );
37  end entity;

```

Source file: [net/icmpv4/icmpv4_TX.vhdl](#)

PoC.net.icmpv4.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1  entity icmpv4_Wrapper is
2      generic (
3          DEBUG                                : boolean           := FALSE;
4          SOURCE_IPV4ADDRESS                   : T_NET_IPV4_ADDRESS := C_NET_IPV4_
5          ↪ ADDRESS_EMPTY
6      );
7      port (
8          Clock                                : in  std_logic;
9          Reset                                : in  std_logic;
10         -- CSE interface
11         Command                              : in  T_NET_ICMPV4_COMMAND;
12         Status                               : out T_NET_ICMPV4_STATUS;
13         Error                                : out T_NET_ICMPV4_ERROR;
14         -- Echo-Request destination address
15         IPv4Address_rst                      : out std_logic;
16         IPv4Address_nxt                      : out std_logic;
17         IPv4Address_Data                     : in  T_SLV_8;
18         -- to IPv4 layer
19         IP_TX_Valid                          : out std_logic;
20         IP_TX_Data                           : out T_SLV_8;
21         IP_TX_SOF                            : out std_logic;

```

(continues on next page)

(continued from previous page)

```

21     IP_TX_EOF                                : out std_logic;
22     IP_TX_Ack                                : in  std_logic;
23     IP_TX_Meta_rst                           : in  std_logic;
24     IP_TX_Meta_SrcIPv4Address_nxt            : in  std_logic;
25     IP_TX_Meta_SrcIPv4Address_Data           : out T_SLV_8;
26     IP_TX_Meta_DestIPv4Address_nxt           : in  std_logic;
27     IP_TX_Meta_DestIPv4Address_Data          : out T_SLV_8;
28     IP_TX_Meta_Length                        : out T_SLV_16;
29     -- from IPv4 layer
30     IP_RX_Valid                              : in  std_logic;
31     IP_RX_Data                                : in  T_SLV_8;
32     IP_RX_SOF                                : in  std_logic;
33     IP_RX_EOF                                : in  std_logic;
34     IP_RX_Ack                                : out std_logic;
35     IP_RX_Meta_rst                           : out std_logic;
36     IP_RX_Meta_SrcMACAddress_nxt             : out std_logic;
37     IP_RX_Meta_SrcMACAddress_Data            : in  T_SLV_8;
38     IP_RX_Meta_DestMACAddress_nxt            : out std_logic;
39     IP_RX_Meta_DestMACAddress_Data           : in  T_SLV_8;
40     -- IP_RX_Meta_EthType                     : in  T_SLV_16;
41     IP_RX_Meta_SrcIPv4Address_nxt            : out std_logic;
42     IP_RX_Meta_SrcIPv4Address_Data           : in  T_SLV_8;
43     IP_RX_Meta_DestIPv4Address_nxt           : out std_logic;
44     IP_RX_Meta_DestIPv4Address_Data          : in  T_SLV_8;
45     -- IP_RX_Meta_TrafficClass                 : in  T_SLV_8;
46     -- IP_RX_Meta_FlowLabel                    : in  T_SLV_24;
47     IP_RX_Meta_Length                        : in  T_SLV_16
48     -- IP_RX_Meta_Protocol                     : in  T_SLV_8
49 );
50 end entity;
```

Source file: net/icmpv4/icmpv4_Wrapper.vhdl

7.13.4 PoC.net.icmpv6

These are icmpv6 entities. . .

PoC.net.icmpv6.RX

Todo: No documentation available.

Entity Declaration:

Source file: net/icmpv6/icmpv6_RX.vhdl

PoC.net.icmpv6.TX

Todo: No documentation available.

Entity Declaration:

Source file: [net/icmpv6/icmpv6_TX.vhdl](#)

PoC.net.icmpv6.Wrapper

Todo: No documentation available.

Entity Declaration:

Source file: [net/icmpv6/icmpv6_Wrapper.vhdl](#)

7.13.5 PoC.net.ipv4

These are ipv4 entities...

PoC.net.ipv4.RX

Todo: No documentation available.

Entity Declaration:

```

1  entity ipv4_RX is
2      generic (
3          DEBUG                      : boolean           := FALSE
4      );
5      port (
6          Clock                      : in  std_logic;      --
7          Reset                      : in  std_logic;      --
8          -- STATUS port
9          Error                      : out std_logic;
10         -- IN port
11         In_Valid                   : in  std_logic;
12         In_Data                    : in  T_SLV_8;
13         In_SOF                     : in  std_logic;
14         In_EOF                     : in  std_logic;
15         In_Ack                     : out std_logic;
16         In_Meta_rst                : out std_logic;
17         In_Meta_SrcMACAddress_nxt  : out std_logic;
18         In_Meta_SrcMACAddress_Data : in  T_SLV_8;
19         In_Meta_DestMACAddress_nxt : out std_logic;
20         In_Meta_DestMACAddress_Data : in  T_SLV_8;
21         In_Meta_EthType            : in  T_SLV_16;
22         -- OUT port
23         Out_Valid                   : out std_logic;
24         Out_Data                    : out T_SLV_8;
25         Out_SOF                     : out std_logic;
26         Out_EOF                     : out std_logic;
27         Out_Ack                     : in  std_logic;
28         Out_Meta_rst                : in  std_logic;
29         Out_Meta_SrcMACAddress_nxt : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

30 Out_Meta_SrcMACAddress_Data      : out T_SLV_8;
31 Out_Meta_DestMACAddress_nxt      : in  std_logic;
32 Out_Meta_DestMACAddress_Data     : out T_SLV_8;
33 Out_Meta_EthType                  : out T_SLV_16;
34 Out_Meta_SrcIPv4Address_nxt      : in  std_logic;
35 Out_Meta_SrcIPv4Address_Data     : out T_SLV_8;
36 Out_Meta_DestIPv4Address_nxt     : in  std_logic;
37 Out_Meta_DestIPv4Address_Data    : out T_SLV_8;
38 Out_Meta_Length                   : out T_SLV_16;
39 Out_Meta_Protocol                 : out T_SLV_8
40 );
41 end entity;

```

Source file: net/ipv4/ipv4_RX.vhdl

PoC.net.ipv4.TX

Todo: No documentation available.

Entity Declaration:

```

1  entity ipv4_TX is
2      generic (
3          DEBUG                      : boolean           := FALSE
4      );
5      port (
6          Clock                      : in  std_logic;      --
7          Reset                      : in  std_logic;      --
8          -- IN port
9          In_Valid                   : in  std_logic;
10         In_Data                     : in  T_SLV_8;
11         In_SOF                     : in  std_logic;
12         In_EOF                     : in  std_logic;
13         In_Ack                     : out std_logic;
14         In_Meta_rst                : out std_logic;
15         In_Meta_SrcIPv4Address_nxt : out std_logic;
16         In_Meta_SrcIPv4Address_Data : in  T_SLV_8;
17         In_Meta_DestIPv4Address_nxt : out std_logic;
18         In_Meta_DestIPv4Address_Data : in  T_SLV_8;
19         In_Meta_Length              : in  T_SLV_16;
20         In_Meta_Protocol            : in  T_SLV_8;
21         -- ARP port
22         ARP_IPCache_Query           : out std_logic;
23         ARP_IPCache_IPv4Address_rst : in  std_logic;
24         ARP_IPCache_IPv4Address_nxt : in  std_logic;
25         ARP_IPCache_IPv4Address_Data : out T_SLV_8;
26         ARP_IPCache_Valid           : in  std_logic;
27         ARP_IPCache_MACAddress_rst  : out std_logic;
28         ARP_IPCache_MACAddress_nxt  : out std_logic;
29         ARP_IPCache_MACAddress_Data : in  T_SLV_8;
30         -- OUT port
31         Out_Valid                   : out std_logic;
32         Out_Data                    : out T_SLV_8;
33         Out_SOF                     : out std_logic;
34         Out_EOF                     : out std_logic;
35         Out_Ack                     : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

36     Out_Meta_rst                : in std_logic;
37     Out_Meta_DestMACAddress_nxt : in std_logic;
38     Out_Meta_DestMACAddress_Data : out T_SLV_8
39 );
40 end entity;
```

Source file: [net/ipv4/ipv4_TX.vhdl](#)

PoC.net.ipv4.FrameLoopback

Todo: No documentation available.

Entity Declaration:

```

1  entity ipv4_FrameLoopback is
2      generic (
3          MAX_FRAMES                : positive           := 4
4      );
5      port (
6          Clock                    : in std_logic;
7          Reset                    : in std_logic;
8          -- IN port
9          In_Valid                 : in std_logic;
10         In_Data                  : in T_SLV_8;
11         In_SOF                   : in std_logic;
12         In_EOF                   : in std_logic;
13         In_Ack                   : out std_logic;
14         In_Meta_rst              : out std_logic;
15         In_Meta_SrcIPv4Address_nxt : out std_logic;
16         In_Meta_SrcIPv4Address_Data : in T_SLV_8;
17         In_Meta_DestIPv4Address_nxt : out std_logic;
18         In_Meta_DestIPv4Address_Data : in T_SLV_8;
19         In_Meta_Length           : in T_SLV_16;
20         -- OUT port
21         Out_Valid                : out std_logic;
22         Out_Data                 : out T_SLV_8;
23         Out_SOF                  : out std_logic;
24         Out_EOF                  : out std_logic;
25         Out_Ack                  : in std_logic;
26         Out_Meta_rst             : in std_logic;
27         Out_Meta_SrcIPv4Address_nxt : in std_logic;
28         Out_Meta_SrcIPv4Address_Data : out T_SLV_8;
29         Out_Meta_DestIPv4Address_nxt : in std_logic;
30         Out_Meta_DestIPv4Address_Data : out T_SLV_8;
31         Out_Meta_Length          : out T_SLV_16
32     );
33 end entity;
```

Source file: [net/ipv4/ipv4_FrameLoopback.vhdl](#)

PoC.net.ipv4.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1 entity ipv4_Wrapper is
2   generic (
3     DEBUG                      : boolean                := FALSE;
4     PACKET_TYPES               : T_NET_IPV4_PROTOCOL_VECTOR := (0 => x"00")
5   );
6   port (
7     Clock                      : in  std_logic;
8     Reset                      : in  std_logic;
9     -- to MAC layer
10    MAC_TX_Valid               : out std_logic;
11    MAC_TX_Data                : out T_SLV_8;
12    MAC_TX_SOF                 : out std_logic;
13    MAC_TX_EOF                 : out std_logic;
14    MAC_TX_Ack                 : in  std_logic;
15    MAC_TX_Meta_rst            : in  std_logic;
16    MAC_TX_Meta_DestMACAddress_nxt : in  std_logic;
17    MAC_TX_Meta_DestMACAddress_Data : out T_SLV_8;
18    -- from MAC layer
19    MAC_RX_Valid               : in  std_logic;
20    MAC_RX_Data                : in  T_SLV_8;
21    MAC_RX_SOF                 : in  std_logic;
22    MAC_RX_EOF                 : in  std_logic;
23    MAC_RX_Ack                 : out std_logic;
24    MAC_RX_Meta_rst            : out std_logic;
25    MAC_RX_Meta_SrcMACAddress_nxt : out std_logic;
26    MAC_RX_Meta_SrcMACAddress_Data : in  T_SLV_8;
27    MAC_RX_Meta_DestMACAddress_nxt : out std_logic;
28    MAC_RX_Meta_DestMACAddress_Data : in  T_SLV_8;
29    MAC_RX_Meta_EthType        : in  T_SLV_16;
30    -- to ARP
31    ARP_IPCache_Query          : out std_logic;
32    ARP_IPCache_IPv4Address_rst : in  std_logic;
33    ARP_IPCache_IPv4Address_nxt : in  std_logic;
34    ARP_IPCache_IPv4Address_Data : out T_SLV_8;
35    -- from ARP
36    ARP_IPCache_Valid          : in  std_logic;
37    ARP_IPCache_MACAddress_rst  : out std_logic;
38    ARP_IPCache_MACAddress_nxt  : out std_logic;
39    ARP_IPCache_MACAddress_Data : in  T_SLV_8;
40    -- from upper layer
41    TX_Valid                   : in  std_logic_vector(PACKET_TYPES'length - 1,
42    ↪downto 0);
43    TX_Data                    : in  T_SLVV_8(PACKET_TYPES'length - 1 downto,
44    ↪0);
45    TX_SOF                     : in  std_logic_vector(PACKET_TYPES'length - 1,
46    ↪downto 0);
47    TX_EOF                     : in  std_logic_vector(PACKET_TYPES'length - 1,
48    ↪downto 0);
49    TX_Ack                     : out std_logic_vector(PACKET_TYPES'length - 1,
50    ↪downto 0);
51    TX_Meta_rst                : out std_logic_vector(PACKET_TYPES'length - 1,
52    ↪downto 0);
53    TX_Meta_SrcIPv4Address_nxt : out std_logic_vector(PACKET_TYPES'length - 1,
54    ↪downto 0);
55    TX_Meta_SrcIPv4Address_Data : in  T_SLVV_8(PACKET_TYPES'length - 1 downto,
56    ↪0);
57    TX_Meta_DestIPv4Address_nxt : out std_logic_vector(PACKET_TYPES'length - 1,
58    ↪downto 0);
59    TX_Meta_DestIPv4Address_Data : in  T_SLVV_8(PACKET_TYPES'length - 1 downto,
60    ↪0);

```

(continues on next page)

(continued from previous page)

```

51     TX_Meta_Length      : in  T_SLVV_16 (PACKET_TYPES'length - 1 downto
↪ 0);
52     -- to upper layer
53     RX_Valid            : out std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
54     RX_Data             : out T_SLVV_8 (PACKET_TYPES'length - 1 downto
↪ 0);
55     RX_SOF              : out std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
56     RX_EOF              : out std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
57     RX_Ack              : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
58     RX_Meta_rst         : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
59     RX_Meta_SrcMACAddress_nxt : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
60     RX_Meta_SrcMACAddress_Data : out T_SLVV_8 (PACKET_TYPES'length - 1
↪ downto 0);
61     RX_Meta_DestMACAddress_nxt : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
62     RX_Meta_DestMACAddress_Data : out T_SLVV_8 (PACKET_TYPES'length - 1
↪ downto 0);
63     RX_Meta_EthType     : out T_SLVV_16 (PACKET_TYPES'length - 1
↪ 0);
64     RX_Meta_SrcIPv4Address_nxt : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
65     RX_Meta_SrcIPv4Address_Data : out T_SLVV_8 (PACKET_TYPES'length - 1
↪ downto 0);
66     RX_Meta_DestIPv4Address_nxt : in  std_logic_vector (PACKET_TYPES'length - 1
↪ downto 0);
67     RX_Meta_DestIPv4Address_Data : out T_SLVV_8 (PACKET_TYPES'length - 1
↪ downto 0);
68     RX_Meta_Length     : out T_SLVV_16 (PACKET_TYPES'length - 1
↪ 0);
69     RX_Meta_Protocol    : out T_SLVV_8 (PACKET_TYPES'length - 1
↪ 0);
70 );
71 end entity;
```

Source file: net/ipv4/ipv4_Wrapper.vhdl

7.13.6 PoC.net.ipv6

These are ipv6 entities....

PoC.net.ipv6.RX

Todo: No documentation available.

Entity Declaration:

```

1 entity ipv6_RX is
2   generic (
3     DEBUG                      : boolean          := FALSE
```

(continues on next page)

(continued from previous page)

```

4   );
5   port (
6       Clock                : in  std_logic;          --
7       Reset                : in  std_logic;          --
8       -- STATUS port
9       Error                : out std_logic;
10      -- IN port
11      In_Valid              : in  std_logic;
12      In_Data               : in  T_SLV_8;
13      In_SOF                : in  std_logic;
14      In_EOF                : in  std_logic;
15      In_Ack                : out std_logic;
16      In_Meta_rst           : out std_logic;
17      In_Meta_SrcMACAddress_nxt : out std_logic;
18      In_Meta_SrcMACAddress_Data : in  T_SLV_8;
19      In_Meta_DestMACAddress_nxt : out std_logic;
20      In_Meta_DestMACAddress_Data : in  T_SLV_8;
21      In_Meta_EthType       : in  T_SLV_16;
22      -- OUT port
23      Out_Valid             : out std_logic;
24      Out_Data              : out T_SLV_8;
25      Out_SOF               : out std_logic;
26      Out_EOF               : out std_logic;
27      Out_Ack               : in  std_logic;
28      Out_Meta_rst          : in  std_logic;
29      Out_Meta_SrcMACAddress_nxt : in  std_logic;
30      Out_Meta_SrcMACAddress_Data : out T_SLV_8;
31      Out_Meta_DestMACAddress_nxt : in  std_logic;
32      Out_Meta_DestMACAddress_Data : out T_SLV_8;
33      Out_Meta_EthType      : out T_SLV_16;
34      Out_Meta_SrcIPv6Address_nxt : in  std_logic;
35      Out_Meta_SrcIPv6Address_Data : out T_SLV_8;
36      Out_Meta_DestIPv6Address_nxt : in  std_logic;
37      Out_Meta_DestIPv6Address_Data : out T_SLV_8;
38      Out_Meta_TrafficClass   : out T_SLV_8;
39      Out_Meta_FlowLabel      : out T_SLV_24; --STD_LOGIC_VECTOR(19 downto_
↳ 0);
40      Out_Meta_Length         : out T_SLV_16;
41      Out_Meta_NextHeader     : out T_SLV_8
42  );
43  end entity;

```

Source file: net/ipv6/ipv6_RX.vhdl

PoC.net.ipv6.TX

Todo: No documentation available.

Entity Declaration:

```

1   entity ipv6_TX is
2       generic (
3           DEBUG                : boolean                := FALSE
4       );
5       port (
6           Clock                : in  std_logic;          --

```

(continues on next page)

(continued from previous page)

```

7      Reset                                : in  std_logic;          --
8      -- IN port
9      In_Valid                             : in  std_logic;
10     In_Data                              : in  T_SLV_8;
11     In_SOF                               : in  std_logic;
12     In_EOF                               : in  std_logic;
13     In_Ack                               : out std_logic;
14     In_Meta_rst                           : out std_logic;
15     In_Meta_SrcIPv6Address_nxt            : out std_logic;
16     In_Meta_SrcIPv6Address_Data           : in  T_SLV_8;
17     In_Meta_DestIPv6Address_nxt           : out std_logic;
18     In_Meta_DestIPv6Address_Data          : in  T_SLV_8;
19     In_Meta_TrafficClass                  : in  T_SLV_8;
20     In_Meta_FlowLabel                     : in  T_SLV_24; --STD_LOGIC_VECTOR(19 downto_
↳ 0);
21     In_Meta_Length                        : in  T_SLV_16;
22     In_Meta_NextHeader                    : in  T_SLV_8;
23     -- to NDP layer
24     NDP_NextHop_Query                     : out std_logic;
25     NDP_NextHop_IPv6Address_rst           : in  std_logic;
26     NDP_NextHop_IPv6Address_nxt           : in  std_logic;
27     NDP_NextHop_IPv6Address_Data          : out T_SLV_8;
28     -- from NDP layer
29     NDP_NextHop_Valid                     : in  std_logic;
30     NDP_NextHop_MACAddress_rst            : out std_logic;
31     NDP_NextHop_MACAddress_nxt            : out std_logic;
32     NDP_NextHop_MACAddress_Data           : in  T_SLV_8;
33     -- OUT port
34     Out_Valid                             : out std_logic;
35     Out_Data                              : out T_SLV_8;
36     Out_SOF                               : out std_logic;
37     Out_EOF                               : out std_logic;
38     Out_Ack                               : in  std_logic;
39     Out_Meta_rst                           : in  std_logic;
40     Out_Meta_DestMACAddress_nxt            : in  std_logic;
41     Out_Meta_DestMACAddress_Data          : out T_SLV_8
42 );
43 end entity;
```

Source file: `net/ipv6/ipv6_TX.vhdl`

PoC.net.ipv6.FrameLoopback

Todo: No documentation available.

Entity Declaration:

```

1  entity ipv6_FrameLoopback is
2      generic (
3          MAX_FRAMES                        : positive          := 4
4      );
5      port (
6          Clock                             : in  std_logic;
7          Reset                             : in  std_logic;
8          -- IN port
9          In_Valid                           : in  std_logic;
```

(continues on next page)

(continued from previous page)

```

10   In_Data                : in  T_SLV_8;
11   In_SOF                 : in  std_logic;
12   In_EOF                 : in  std_logic;
13   In_Ack                 : out std_logic;
14   In_Meta_rst            : out std_logic;
15   In_Meta_SrcIPv6Address_nxt : out std_logic;
16   In_Meta_SrcIPv6Address_Data : in  T_SLV_8;
17   In_Meta_DestIPv6Address_nxt : out std_logic;
18   In_Meta_DestIPv6Address_Data : in  T_SLV_8;
19   In_Meta_Length         : in  T_SLV_16;
20   -- OUT port
21   Out_Valid              : out std_logic;
22   Out_Data               : out T_SLV_8;
23   Out_SOF                : out std_logic;
24   Out_EOF                : out std_logic;
25   Out_Ack                : in  std_logic;
26   Out_Meta_rst           : in  std_logic;
27   Out_Meta_SrcIPv6Address_nxt : in  std_logic;
28   Out_Meta_SrcIPv6Address_Data : out T_SLV_8;
29   Out_Meta_DestIPv6Address_nxt : in  std_logic;
30   Out_Meta_DestIPv6Address_Data : out T_SLV_8;
31   Out_Meta_Length        : out T_SLV_16
32 );
33 end entity;

```

Source file: net/ipv6/ipv6_FrameLoopback.vhdl

PoC.net.ipv6.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1  entity ipv6_Wrapper is
2      generic (
3          DEBUG                : boolean                := FALSE;
4          PACKET_TYPES         : T_NET_IPV6_NEXT_HEADER_VECTOR := (0 => x"00
↳ ")
5      );
6      port (
7          Clock                : in  std_logic;
8          Reset                 : in  std_logic;
9          -- to MAC layer
10         MAC_TX_Valid          : out std_logic;
11         MAC_TX_Data           : out T_SLV_8;
12         MAC_TX_SOF            : out std_logic;
13         MAC_TX_EOF            : out std_logic;
14         MAC_TX_Ack            : in  std_logic;
15         MAC_TX_Meta_rst       : in  std_logic;
16         MAC_TX_Meta_DestMACAddress_nxt : in  std_logic;
17         MAC_TX_Meta_DestMACAddress_Data : out T_SLV_8;
18         -- from MAC layer
19         MAC_RX_Valid          : in  std_logic;
20         MAC_RX_Data           : in  T_SLV_8;
21         MAC_RX_SOF            : in  std_logic;
22         MAC_RX_EOF            : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

23     MAC_RX_Ack                                : out std_logic;
24     MAC_RX_Meta_rst                          : out std_logic;
25     MAC_RX_Meta_SrcMACAddress_nxt           : out std_logic;
26     MAC_RX_Meta_SrcMACAddress_Data          : in  T_SLV_8;
27     MAC_RX_Meta_DestMACAddress_nxt          : out std_logic;
28     MAC_RX_Meta_DestMACAddress_Data         : in  T_SLV_8;
29     MAC_RX_Meta_EthType                     : in  T_SLV_16;
30     -- to NDP layer
31     NDP_NextHop_Query                       : out std_logic;
32     NDP_NextHop_IPv6Address_rst             : in  std_logic;
33     NDP_NextHop_IPv6Address_nxt            : in  std_logic;
34     NDP_NextHop_IPv6Address_Data           : out T_SLV_8;
35     -- from NDP layer
36     NDP_NextHop_Valid                      : in  std_logic;
37     NDP_NextHop_MACAddress_rst             : out std_logic;
38     NDP_NextHop_MACAddress_nxt            : out std_logic;
39     NDP_NextHop_MACAddress_Data            : in  T_SLV_8;
40     -- from upper layer
41     TX_Valid                               : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
42     TX_Data                               : in  T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);
43     TX_SOF                               : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
44     TX_EOF                               : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
45     TX_Ack                               : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
46     TX_Meta_rst                          : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
47     TX_Meta_SrcIPv6Address_nxt           : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
48     TX_Meta_SrcIPv6Address_Data          : in  T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);
49     TX_Meta_DestIPv6Address_nxt          : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
50     TX_Meta_DestIPv6Address_Data         : in  T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);
51     TX_Meta_TrafficClass                  : in  T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);
52     TX_Meta_FlowLabel                     : in  T_SLVV_24(PACKET_TYPES'length - 1 downto
↳0);
53     TX_Meta_Length                       : in  T_SLVV_16(PACKET_TYPES'length - 1 downto
↳0);
54     -- to upper layer
55     RX_Valid                               : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
56     RX_Data                               : out T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);
57     RX_SOF                               : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
58     RX_EOF                               : out std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
59     RX_Ack                               : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
60     RX_Meta_rst                          : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
61     RX_Meta_SrcMACAddress_nxt           : in  std_logic_vector(PACKET_TYPES'length - 1,
↳downto 0);
62     RX_Meta_SrcMACAddress_Data          : out T_SLVV_8(PACKET_TYPES'length - 1 downto
↳0);

```

(continues on next page)

(continued from previous page)

```

63     RX_Meta_DestMACAddress_nxt      : in std_logic_vector(PACKET_TYPES'length - 1_
↪downto 0);
64     RX_Meta_DestMACAddress_Data    : out T_SLVV_8(PACKET_TYPES'length - 1 downto_
↪0);
65     RX_Meta_EthType                : out T_SLVV_16(PACKET_TYPES'length - 1 downto_
↪0);
66     RX_Meta_SrcIPv6Address_nxt     : in std_logic_vector(PACKET_TYPES'length - 1_
↪downto 0);
67     RX_Meta_SrcIPv6Address_Data    : out T_SLVV_8(PACKET_TYPES'length - 1 downto_
↪0);
68     RX_Meta_DestIPv6Address_nxt    : in std_logic_vector(PACKET_TYPES'length - 1_
↪downto 0);
69     RX_Meta_DestIPv6Address_Data    : out T_SLVV_8(PACKET_TYPES'length - 1 downto_
↪0);
70     RX_Meta_TrafficClass           : out T_SLVV_8(PACKET_TYPES'length - 1 downto_
↪0);
71     RX_Meta_FlowLabel              : out T_SLVV_24(PACKET_TYPES'length - 1 downto_
↪0);
72     RX_Meta_Length                 : out T_SLVV_16(PACKET_TYPES'length - 1 downto_
↪0);
73     RX_Meta_NextHeader              : out T_SLVV_8(PACKET_TYPES'length - 1 downto_
↪0);
74 );
75 end entity;
```

Source file: net/ipv6/ipv6_Wrapper.vhdl

7.13.7 PoC.net.mac

These are mac entities...

PoC.net.mac.RX_DestMAC_Switch

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_RX_DestMAC_Switch is
2      generic (
3          DEBUG                      : boolean                      := FALSE;
4          MAC_ADDRESSES              : T_NET_MAC_ADDRESS_VECTOR    := (0 => C_NET_MAC_
↪ADDRESS_EMPTY);
5          MAC_ADDRESSES_MASKS       : T_NET_MAC_ADDRESS_VECTOR    := (0 => C_NET_MAC_
↪MASK_DEFAULT)
6      );
7      port (
8          Clock                      : in std_logic;
9          Reset                      : in std_logic;
10
11         In_Valid                   : in std_logic;
12         In_Data                    : in T_SLV_8;
13         In_SOF                     : in std_logic;
14         In_EOF                     : in std_logic;
15         In_Ack                     : out std_logic;
16     );
```

(continues on next page)

(continued from previous page)

```

17     Out_Valid                                : out std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
18     Out_Data                                : out T_SLVV_8(MAC_ADDRESSES'length - 1 downto_
↪0);
19     Out_SOF                                : out std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
20     Out_EOF                                : out std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
21     Out_Ack                                : in std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
22     Out_Meta_DestMACAddress_rst             : in std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
23     Out_Meta_DestMACAddress_nxt             : in std_logic_vector(MAC_ADDRESSES'length - 1_
↪downto 0);
24     Out_Meta_DestMACAddress_Data            : out T_SLVV_8(MAC_ADDRESSES'length - 1 downto 0)
25 );
26 end entity;
```

Source file: net/mac/mac_RX_DestMAC_Switch.vhdl

PoC.net.mac.RX_SrcMAC_Filter

Todo: No documentation available.

Entity Declaration:

```

1 entity mac_RX_SrcMAC_Filter is
2     generic (
3         DEBUG                                : boolean                := FALSE;
4         MAC_ADDRESSES                        : T_NET_MAC_ADDRESS_VECTOR := (0 => C_NET_
↪MAC_ADDRESS_EMPTY);
5         MAC_ADDRESSE_MASKS                  : T_NET_MAC_ADDRESS_VECTOR := (0 => C_NET_
↪MAC_MASK_DEFAULT)
6     );
7     port (
8         Clock                                : in std_logic;
9         Reset                                : in std_logic;
10
11         In_Valid                             : in std_logic;
12         In_Data                              : in T_SLV_8;
13         In_SOF                              : in std_logic;
14         In_EOF                              : in std_logic;
15         In_Ack                              : out std_logic;
16         In_Meta_rst                         : out std_logic;
17         In_Meta_DestMACAddress_nxt          : out std_logic;
18         In_Meta_DestMACAddress_Data         : in T_SLV_8;
19
20         Out_Valid                           : out std_logic;
21         Out_Data                            : out T_SLV_8;
22         Out_SOF                             : out std_logic;
23         Out_EOF                             : out std_logic;
24         Out_Ack                             : in std_logic;
25         Out_Meta_rst                        : in std_logic;
26         Out_Meta_DestMACAddress_nxt         : in std_logic;
27         Out_Meta_DestMACAddress_Data        : out T_SLV_8;
28         Out_Meta_SrcMACAddress_nxt         : in std_logic;
```

(continues on next page)

(continued from previous page)

```

29     Out_Meta_SrcMACAddress_Data    : out T_SLV_8
30 );
31 end entity;
```

Source file: [net/mac/mac_RX_SrcMAC_Filter.vhdl](#)

PoC.net.mac.RX_Type_Switch

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_RX_Type_Switch is
2      generic (
3          DEBUG                                : boolean                := FALSE;
4          ETHERNET_TYPES                       : T_NET_MAC_ETHERNETTYPE_VECTOR := (0 => C_NET_
↳MAC_ETHERNETTYPE_EMPTY)
5      );
6      port (
7          Clock                                : in  std_logic;
8          Reset                                : in  std_logic;
9
10         In_Valid                             : in  std_logic;
11         In_Data                              : in  T_SLV_8;
12         In_SOF                               : in  std_logic;
13         In_EOF                               : in  std_logic;
14         In_Ack                               : out std_logic;
15         In_Meta_rst                          : out std_logic;
16         In_Meta_SrcMACAddress_nxt            : out std_logic;
17         In_Meta_SrcMACAddress_Data           : in  T_SLV_8;
18         In_Meta_DestMACAddress_nxt          : out std_logic;
19         In_Meta_DestMACAddress_Data         : in  T_SLV_8;
20
21         Out_Valid                             : out std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
22         Out_Data                              : out T_SLVV_8(ETHERNET_TYPES'length - 1 downto_
↳0);
23         Out_SOF                               : out std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
24         Out_EOF                               : out std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
25         Out_Ack                               : in  std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
26         Out_Meta_rst                          : in  std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
27         Out_Meta_SrcMACAddress_nxt            : in  std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
28         Out_Meta_SrcMACAddress_Data           : out T_SLVV_8(ETHERNET_TYPES'length - 1 downto_
↳0);
29         Out_Meta_DestMACAddress_nxt          : in  std_logic_vector(ETHERNET_TYPES'length - 1_
↳downto 0);
30         Out_Meta_DestMACAddress_Data         : out T_SLVV_8(ETHERNET_TYPES'length - 1 downto_
↳0);
31         Out_Meta_EthType                     : out T_NET_MAC_ETHERNETTYPE_VECTOR(ETHERNET_
↳TYPES'length - 1 downto 0)
32     );
```

(continues on next page)

(continued from previous page)

```
end entity;
```

Source file: `net/mac/mac_RX_Type_Switch.vhdl`

PoC.net.mac.TX_SrcMAC_Prepender

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_TX_SrcMAC_Prepender is
2      generic (
3          DEBUG                      : boolean                := FALSE;
4          MAC_ADDRESSES              : T_NET_MAC_ADDRESS_VECTOR := (0 => C_NET_
↳MAC_ADDRESS_EMPTY)
5      );
6      port (
7          Clock                      : in  std_logic;
8          Reset                      : in  std_logic;
9          -- IN Port
10         In_Valid                   : in  std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
11         In_Data                    : in  T_SLVV_8(MAC_ADDRESSES'length - 1 downto_
↳0);
12         In_SOF                     : in  std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
13         In_EOF                     : in  std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
14         In_Ack                     : out std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
15         In_Meta_rst                : out std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
16         In_Meta_DestMACAddress_nxt : out std_logic_vector(MAC_ADDRESSES'length - 1_
↳downto 0);
17         In_Meta_DestMACAddress_Data : in  T_SLVV_8(MAC_ADDRESSES'length - 1 downto_
↳0);
18         -- OUT Port
19         Out_Valid                   : out std_logic;
20         Out_Data                    : out T_SLV_8;
21         Out_SOF                     : out std_logic;
22         Out_EOF                     : out std_logic;
23         Out_Ack                     : in  std_logic;
24         Out_Meta_rst                : in  std_logic;
25         Out_Meta_DestMACAddress_nxt : in  std_logic;
26         Out_Meta_DestMACAddress_Data : out T_SLV_8
27     );
28 end entity;
```

Source file: `net/mac/mac_TX_SrcMAC_Prepender.vhdl`

PoC.net.mac.TX_DestMAC_Prepender

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_TX_DestMAC_Prepender is
2      generic (
3          DEBUG                      : boolean           := FALSE
4      );
5      port (
6          Clock                      : in  std_logic;
7          Reset                      : in  std_logic;
8
9          In_Valid                   : in  std_logic;
10         In_Data                     : in  T_SLV_8;
11         In_SOF                     : in  std_logic;
12         In_EOF                     : in  std_logic;
13         In_Ack                     : out std_logic;
14         In_Meta_rst                : out std_logic;
15         In_Meta_DestMACAddress_nxt : out std_logic;
16         In_Meta_DestMACAddress_Data : in  T_SLV_8;
17
18         Out_Valid                   : out std_logic;
19         Out_Data                     : out T_SLV_8;
20         Out_SOF                     : out std_logic;
21         Out_EOF                     : out std_logic;
22         Out_Ack                     : in  std_logic
23     );
24  end entity;

```

Source file: [net/mac/mac_TX_DestMAC_Prepender.vhdl](#)

PoC.net.mac.TX_Type_Prepender

Todo: No documentation available.

Entity Declaration:

Source file: [net/mac/mac_TX_Type_Prepender.vhdl](#)

PoC.net.mac.FrameLoopback

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_FrameLoopback is
2      generic (
3          MAX_FRAMES                 : positive           := 4
4      );
5      port (
6          Clock                      : in  std_logic;
7          Reset                      : in  std_logic;
8          -- IN Port
9          In_Valid                   : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

10      In_Data                : in  T_SLV_8;
11      In_SOF                : in  std_logic;
12      In_EOF                : in  std_logic;
13      In_Ack                : out std_logic;
14      In_Meta_rst           : out std_logic;
15      In_Meta_SrcMACAddress_nxt : out std_logic;
16      In_Meta_SrcMACAddress_Data : in  T_SLV_8;
17      In_Meta_DestMACAddress_nxt : out std_logic;
18      In_Meta_DestMACAddress_Data : in  T_SLV_8;
19      -- OUT Port
20      Out_Valid             : out std_logic;
21      Out_Data              : out T_SLV_8;
22      Out_SOF               : out std_logic;
23      Out_EOF               : out std_logic;
24      Out_Ack               : in  std_logic;
25      Out_Meta_rst          : in  std_logic;
26      Out_Meta_SrcMACAddress_nxt : in  std_logic;
27      Out_Meta_SrcMACAddress_Data : out T_SLV_8;
28      Out_Meta_DestMACAddress_nxt : in  std_logic;
29      Out_Meta_DestMACAddress_Data : out T_SLV_8
30  );
31  end entity;

```

Source file: [net/mac/mac_FrameLoopback.vhdl](#)

PoC.net.mac.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1  entity mac_Wrapper is
2      generic (
3          DEBUG                : boolean                := FALSE;
4          MAC_CONFIG           : T_NET_MAC_CONFIGURATION_VECTOR
5      );
6      port (
7          Clock                : in  std_logic;
8          Reset                : in  std_logic;
9
10         Eth_TX_Valid          : out std_logic;
11         Eth_TX_Data           : out T_SLV_8;
12         Eth_TX_SOF            : out std_logic;
13         Eth_TX_EOF            : out std_logic;
14         Eth_TX_Ack            : in  std_logic;
15
16         Eth_RX_Valid          : in  std_logic;
17         Eth_RX_Data           : in  T_SLV_8;
18         Eth_RX_SOF            : in  std_logic;
19         Eth_RX_EOF            : in  std_logic;
20         Eth_RX_Ack            : out std_logic;
21
22         TX_Valid              : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
23         TX_Data               : in  T_SLVV_8 (getPortCount (MAC_CONFIG) - 1 downto
↪0);

```

(continues on next page)

(continued from previous page)

```

24     TX_SOF                                     : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
25     TX_EOF                                     : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
26     TX_Ack                                     : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
27     TX_Meta_rst                               : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
28     TX_Meta_DestMACAddress_nxt                : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
29     TX_Meta_DestMACAddress_Data              : in   T_SLVV_8 (getPortCount (MAC_CONFIG) - 1 downto
↪0);
30
31     RX_Valid                                   : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
32     RX_Data                                   : out T_SLVV_8 (getPortCount (MAC_CONFIG) - 1 downto
↪0);
33     RX_SOF                                     : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
34     RX_EOF                                     : out std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
35     RX_Ack                                     : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
36     RX_Meta_rst                               : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
37     RX_Meta_SrcMACAddress_nxt                : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
38     RX_Meta_SrcMACAddress_Data              : out T_SLVV_8 (getPortCount (MAC_CONFIG) - 1 downto
↪0);
39     RX_Meta_DestMACAddress_nxt                : in  std_logic_vector (getPortCount (MAC_CONFIG) -
↪1 downto 0);
40     RX_Meta_DestMACAddress_Data              : out T_SLVV_8 (getPortCount (MAC_CONFIG) - 1 downto
↪0);
41     RX_Meta_EthType                          : out T_NET_MAC_ETHERNETTYPE_
↪VECTOR (getPortCount (MAC_CONFIG) - 1 downto 0)
42 );
43 end entity;
```

Source file: [net/mac/mac_Wrapper.vhdl](#)

7.13.8 PoC.net.ndp

These are ndp entities. ...

PoC.net.ndp.DestinationCache

Todo: No documentation available.

Entity Declaration:

Source file: [net/ndp/ndp_DestinationCache.vhdl](#)

PoC.net.ndp.FSMQuery

Todo: No documentation available.

Entity Declaration:

Source file: [net/ndp/ndp_FSMQuery.vhdl](#)

PoC.net.ndp.NeighborCache

Todo: No documentation available.

Entity Declaration:

Source file: [net/ndp/ndp_NeighborCache.vhdl](#)

PoC.net.ndp.Wrapper

Todo: No documentation available.

Entity Declaration:

Source file: [net/ndp/ndp_Wrapper.vhdl](#)

7.13.9 PoC.net.stack

These are udp entities. . . .

stack_IPv4

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

stack_IPv6

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

PoC.net.stack.UDIPv4

Todo: No documentation available.

Entity Declaration:

Source file: [net/stack/stack_UDIPv4.vhdl](#)

stack_UDIPv6

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

stack_MAC

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

7.13.10 PoC.net.udp

These are udp entities. . .

PoC.net.udp.RX

Todo: No documentation available.

Entity Declaration:

```

1  entity udp_RX is
2      generic (
3          DEBUG                : boolean           := FALSE;
4          IP_VERSION           : positive          := 6
5      );
6      port (
7          Clock                : in  std_logic;      --
8          Reset                : in  std_logic;      --
9          -- STATUS port
10         Error                : out std_logic;
11         -- IN port
12         In_Valid             : in  std_logic;
13         In_Data              : in  T_SLV_8;

```

(continues on next page)

(continued from previous page)

```

14     In_SOF                : in std_logic;
15     In_EOF                : in std_logic;
16     In_Ack                : out std_logic;
17     In_Meta_rst           : out std_logic;
18     In_Meta_SrcMACAddress_nxt : out std_logic;
19     In_Meta_SrcMACAddress_Data : in T_SLV_8;
20     In_Meta_DestMACAddress_nxt : out std_logic;
21     In_Meta_DestMACAddress_Data : in T_SLV_8;
22     In_Meta_EthType        : in T_SLV_16;
23     In_Meta_SrcIPAddress_nxt : out std_logic;
24     In_Meta_SrcIPAddress_Data : in T_SLV_8;
25     In_Meta_DestIPAddress_nxt : out std_logic;
26     In_Meta_DestIPAddress_Data : in T_SLV_8;
27     -- In_Meta_TrafficClass    : in T_SLV_8;
28     -- In_Meta_FlowLabel       : in T_SLV_24;
29     In_Meta_Length          : in T_SLV_16;
30     In_Meta_Protocol        : in T_SLV_8;
31     -- OUT port
32     Out_Valid               : out std_logic;
33     Out_Data                : out T_SLV_8;
34     Out_SOF                : out std_logic;
35     Out_EOF                : out std_logic;
36     Out_Ack                : in std_logic;
37     Out_Meta_rst           : in std_logic;
38     Out_Meta_SrcMACAddress_nxt : in std_logic;
39     Out_Meta_SrcMACAddress_Data : out T_SLV_8;
40     Out_Meta_DestMACAddress_nxt : in std_logic;
41     Out_Meta_DestMACAddress_Data : out T_SLV_8;
42     Out_Meta_EthType        : out T_SLV_16;
43     Out_Meta_SrcIPAddress_nxt : in std_logic;
44     Out_Meta_SrcIPAddress_Data : out T_SLV_8;
45     Out_Meta_DestIPAddress_nxt : in std_logic;
46     Out_Meta_DestIPAddress_Data : out T_SLV_8;
47     -- Out_Meta_TrafficClass    : out T_SLV_8;
48     -- Out_Meta_FlowLabel       : out T_SLV_24;
49     Out_Meta_Length          : out T_SLV_16;
50     Out_Meta_Protocol        : out T_SLV_8;
51     Out_Meta_SrcPort         : out T_SLV_16;
52     Out_Meta_DestPort        : out T_SLV_16;
53 );
54 end entity;

```

Source file: net/udp/udp_RX.vhdl

PoC.net.udp.TX

Todo: No documentation available.

Entity Declaration:

```

1 entity udp_TX is
2   generic (
3     DEBUG                : boolean          := FALSE;
4     IP_VERSION           : positive         := 6
5   );
6   port (

```

(continues on next page)

(continued from previous page)

```

7      Clock                : in  std_logic;          --
8      Reset                : in  std_logic;          --
9      -- IN port
10     In_Valid              : in  std_logic;
11     In_Data               : in  T_SLV_8;
12     In_SOF                : in  std_logic;
13     In_EOF                : in  std_logic;
14     In_Ack                : out  std_logic;
15     In_Meta_rst           : out  std_logic;
16     In_Meta_SrcIPAddress_nxt : out std_logic;
17     In_Meta_SrcIPAddress_Data : in  T_SLV_8;
18     In_Meta_DestIPAddress_nxt : out std_logic;
19     In_Meta_DestIPAddress_Data : in  T_SLV_8;
20     In_Meta_SrcPort        : in  T_SLV_16;
21     In_Meta_DestPort       : in  T_SLV_16;
22     In_Meta_Length         : in  T_SLV_16;
23     In_Meta_Checksum       : in  T_SLV_16;
24     -- OUT port
25     Out_Valid             : out  std_logic;
26     Out_Data              : out  T_SLV_8;
27     Out_SOF               : out  std_logic;
28     Out_EOF               : out  std_logic;
29     Out_Ack               : in  std_logic;
30     Out_Meta_rst          : in  std_logic;
31     Out_Meta_SrcIPAddress_nxt : in  std_logic;
32     Out_Meta_SrcIPAddress_Data : out T_SLV_8;
33     Out_Meta_DestIPAddress_nxt : in  std_logic;
34     Out_Meta_DestIPAddress_Data : out T_SLV_8;
35     Out_Meta_Length       : out  T_SLV_16
36 );
37 end entity;
```

Source file: `net/udp/udp_TX.vhdl`

PoC.net.udp.FrameLoopback

Todo: No documentation available.

Entity Declaration:

```

1  entity udp_FrameLoopback is
2      generic (
3          IP_VERSION        : positive      := 6;
4          MAX_FRAMES        : positive      := 4;
5      );
6      port (
7          Clock              : in  std_logic;
8          Reset              : in  std_logic;
9          -- IN port
10         In_Valid           : in  std_logic;
11         In_Data             : in  T_SLV_8;
12         In_SOF              : in  std_logic;
13         In_EOF              : in  std_logic;
14         In_Ack              : out  std_logic;
15         In_Meta_rst         : out  std_logic;
16         In_Meta_DestIPAddress_nxt : out std_logic;
```

(continues on next page)

(continued from previous page)

```

17   In_Meta_DestIPAddress_Data    : in  T_SLV_8;
18   In_Meta_SrcIPAddress_nxt     : out std_logic;
19   In_Meta_SrcIPAddress_Data    : in  T_SLV_8;
20   In_Meta_DestPort             : in  T_NET_UDP_PORT;
21   In_Meta_SrcPort              : in  T_NET_UDP_PORT;
22   -- OUT port
23   Out_Valid                    : out std_logic;
24   Out_Data                     : out T_SLV_8;
25   Out_SOF                     : out std_logic;
26   Out_EOF                     : out std_logic;
27   Out_Ack                     : in  std_logic;
28   Out_Meta_rst                : in  std_logic;
29   Out_Meta_DestIPAddress_nxt   : in  std_logic;
30   Out_Meta_DestIPAddress_Data : out T_SLV_8;
31   Out_Meta_SrcIPAddress_nxt   : in  std_logic;
32   Out_Meta_SrcIPAddress_Data  : out T_SLV_8;
33   Out_Meta_DestPort           : out T_NET_UDP_PORT;
34   Out_Meta_SrcPort            : out T_NET_UDP_PORT
35 );
36 end entity;

```

Source file: [net/udp/udp_FrameLoopback.vhdl](#)

PoC.net.udp.Wrapper

Todo: No documentation available.

Entity Declaration:

```

1  entity udp_Wrapper is
2      generic (
3          DEBUG                      : boolean           := FALSE;
4          IP_VERSION                 : positive          := 6;
5          PORTPAIRS                  : T_NET_UDP_PORTPAIR_VECTOR := (0 => (x
↪ "0000", x"0000"))
6      );
7      port (
8          Clock                      : in  std_logic;
9          Reset                      : in  std_logic;
10         -- from IP layer
11         IP_TX_Valid                 : out std_logic;
12         IP_TX_Data                  : out T_SLV_8;
13         IP_TX_SOF                   : out std_logic;
14         IP_TX_EOF                   : out std_logic;
15         IP_TX_Ack                   : in  std_logic;
16         IP_TX_Meta_rst              : in  std_logic;
17         IP_TX_Meta_SrcIPAddress_nxt : in  std_logic;
18         IP_TX_Meta_SrcIPAddress_Data : out T_SLV_8;
19         IP_TX_Meta_DestIPAddress_nxt : in  std_logic;
20         IP_TX_Meta_DestIPAddress_Data : out T_SLV_8;
21         IP_TX_Meta_Length           : out T_SLV_16;
22         -- to IP layer
23         IP_RX_Valid                 : in  std_logic;
24         IP_RX_Data                  : in  T_SLV_8;
25         IP_RX_SOF                   : in  std_logic;
26         IP_RX_EOF                   : in  std_logic;

```

(continues on next page)

(continued from previous page)

```

27     IP_RX_Ack                                : out std_logic;
28     IP_RX_Meta_rst                          : out std_logic;
29     IP_RX_Meta_SrcMACAddress_nxt            : out std_logic;
30     IP_RX_Meta_SrcMACAddress_Data           : in  T_SLV_8;
31     IP_RX_Meta_DestMACAddress_nxt           : out std_logic;
32     IP_RX_Meta_DestMACAddress_Data          : in  T_SLV_8;
33     IP_RX_Meta_EthType                      : in  T_SLV_16;
34     IP_RX_Meta_SrcIPAddress_nxt             : out std_logic;
35     IP_RX_Meta_SrcIPAddress_Data            : in  T_SLV_8;
36     IP_RX_Meta_DestIPAddress_nxt            : out std_logic;
37     IP_RX_Meta_DestIPAddress_Data           : in  T_SLV_8;
38     -- IP_RX_Meta_TrafficClass               : in  T_SLV_8;
39     -- IP_RX_Meta_FlowLabel                  : in  T_SLV_24;
40     IP_RX_Meta_Length                       : in  T_SLV_16;
41     IP_RX_Meta_Protocol                     : in  T_SLV_8;
42     -- from upper layer
43     TX_Valid                               : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
44     TX_Data                                : in  T_SLVV_8(PORTPAIRS'length - 1 downto_
↳0);
45     TX_SOF                                 : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
46     TX_EOF                                 : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
47     TX_Ack                                 : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
48     TX_Meta_rst                          : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
49     TX_Meta_SrcIPAddress_nxt              : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
50     TX_Meta_SrcIPAddress_Data             : in  T_SLVV_8(PORTPAIRS'length - 1 downto_
↳0);
51     TX_Meta_DestIPAddress_nxt              : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
52     TX_Meta_DestIPAddress_Data            : in  T_SLVV_8(PORTPAIRS'length - 1 downto_
↳0);
53     TX_Meta_SrcPort                       : in  T_SLVV_16(PORTPAIRS'length - 1 downto_
↳0);
54     TX_Meta_DestPort                      : in  T_SLVV_16(PORTPAIRS'length - 1 downto_
↳0);
55     TX_Meta_Length                        : in  T_SLVV_16(PORTPAIRS'length - 1 downto_
↳0);
56     -- to upper layer
57     RX_Valid                               : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
58     RX_Data                                : out T_SLVV_8(PORTPAIRS'length - 1 downto_
↳0);
59     RX_SOF                                 : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
60     RX_EOF                                 : out std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
61     RX_Ack                                 : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
62     RX_Meta_rst                          : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
63     RX_Meta_SrcMACAddress_nxt              : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);
64     RX_Meta_SrcMACAddress_Data            : out T_SLVV_8(PORTPAIRS'length - 1 downto_
↳0);
65     RX_Meta_DestMACAddress_nxt            : in  std_logic_vector(PORTPAIRS'length - 1_
↳downto 0);

```

(continues on next page)

(continued from previous page)

```

66     RX_Meta_DestMACAddress_Data      : out T_SLVV_8(PORTPAIRS'length - 1 downto
↪0);
67     RX_Meta_EthType                  : out T_SLVV_16(PORTPAIRS'length - 1 downto
↪0);
68     RX_Meta_SrcIPAddress_nxt         : in  std_logic_vector(PORTPAIRS'length - 1
↪downto 0);
69     RX_Meta_SrcIPAddress_Data        : out T_SLVV_8(PORTPAIRS'length - 1 downto
↪0);
70     RX_Meta_DestIPAddress_nxt        : in  std_logic_vector(PORTPAIRS'length - 1
↪downto 0);
71     RX_Meta_DestIPAddress_Data       : out T_SLVV_8(PORTPAIRS'length - 1 downto
↪0);
72     -- RX_Meta_TrafficClass           : out T_SLVV_8(PORTPAIRS'length - 1 downto
↪0);
73     -- RX_Meta_FlowLabel              : out T_SLVV_24(PORTPAIRS'length - 1
↪downto 0);
74     RX_Meta_Length                   : out T_SLVV_16(PORTPAIRS'length - 1 downto
↪0);
75     RX_Meta_Protocol                 : out T_SLVV_8(PORTPAIRS'length - 1 downto
↪0);
76     RX_Meta_SrcPort                  : out T_SLVV_16(PORTPAIRS'length - 1 downto
↪0);
77     RX_Meta_DestPort                 : out T_SLVV_16(PORTPAIRS'length - 1 downto
↪0);
78     );
79 end entity;
```

Source file: net/udp/udp_Wrapper.vhdl

7.13.11 PoC.net Package

Source file: net.pkg.vhdl

7.13.12 PoC.net.FrameChecksum

Todo: No documentation available.

Entity Declaration:

```

1  entity net_FrameChecksum is
2      generic (
3          MAX_FRAMES                  : positive      := 8;
4          MAX_FRAME_LENGTH            : positive      := 2048;
5          META_BITS                   : T_POSVEC      := (0 => 8);
6          META_FIFO_DEPTH             : T_POSVEC      := (0 => 16)
7      );
8      port (
9          Clock                       : in  std_logic;
10         Reset                        : in  std_logic;
11         -- IN port
12         In_Valid                     : in  std_logic;
13         In_Data                      : in  T_SLV_8;
14         In_SOF                       : in  std_logic;
15         In_EOF                       : in  std_logic;
16         In_Ack                      : out std_logic;
```

(continues on next page)

(continued from previous page)

```

17     In_Meta_rst                : out std_logic;
18     In_Meta_nxt                : out std_logic_vector(META_BITS'length - 1_
↳downto 0);
19     In_Meta_Data               : in  std_logic_vector(isum(META_BITS) - 1_
↳downto 0);
20     -- OUT port
21     Out_Valid                  : out std_logic;
22     Out_Data                    : out T_SLV_8;
23     Out_SOF                    : out std_logic;
24     Out_EOF                    : out std_logic;
25     Out_Ack                    : in  std_logic;
26     Out_Meta_rst              : in  std_logic;
27     Out_Meta_nxt              : in  std_logic_vector(META_BITS'length - 1_
↳downto 0);
28     Out_Meta_Data              : out std_logic_vector(isum(META_BITS) - 1_
↳downto 0);
29     Out_Meta_Length            : out T_SLV_16;
30     Out_Meta_Checksum          : out T_SLV_16
31 );
32 end entity;

```

Source file: [net/net_FrameChecksum.vhdl](#)

7.13.13 PoC.net.FrameLoopback

Todo: No documentation available.

Entity Declaration:

```

1  entity FrameLoopback is
2      generic (
3          DATA_BW                : positive      := 8;
4          META_BW                 : natural       := 0;
5      );
6      port (
7          Clock                   : in  std_logic;
8          Reset                   : in  std_logic;
9
10         In_Valid                 : in  std_logic;
11         In_Data                   : in  std_logic_vector(DATA_BW - 1_downto 0);
12         In_Meta                   : in  std_logic_vector(META_BW - 1_downto 0);
13         In_SOF                    : in  std_logic;
14         In_EOF                    : in  std_logic;
15         In_Ack                    : out std_logic;
16
17
18         Out_Valid                 : out std_logic;
19         Out_Data                   : out std_logic_vector(DATA_BW - 1_downto 0);
20         Out_Meta                   : out std_logic_vector(META_BW - 1_downto 0);
21         Out_SOF                    : out std_logic;
22         Out_EOF                    : out std_logic;
23         Out_Ack                    : in  std_logic
24     );
25 end entity;

```

Source file: [net/net_FrameLoopback.vhdl](#)

7.14 PoC.sort

These are sorting entities. . . .

Sub-Namespaces

- *PoC.sort.sortnet*

Entities

- IP:sort_ExpireList
- IP:sort_InsertSort
- *PoC.sort.LeastFrequentlyUsed*
- *PoC.sort.lru_cache*
- *PoC.sort.lru_list*

7.14.1 PoC.sort.sortnet

This sub-namespace contains sorting network implementations.

Entities

- *PoC.sort.sortnet.BitonicSort*
- *PoC.sort.sortnet.MergeSort_Streamed*
- *PoC.sort.sortnet.OddEvenMergeSort*
- *PoC.sort.sortnet.OddEvenSort*
- *PoC.sort.sortnet.Stream_Adapter*
- *PoC.sort.sortnet.Stream_Adapter2*
- *PoC.sort.sortnet.Transform*

PoC.sort.sortnet Package

```
type T_SORTNET_IMPL is (  
    SORT_SORTNET_IMPL_ODDEVEN_SORT,  
    SORT_SORTNET_IMPL_ODDEVEN_MERGESORT,  
    SORT_SORTNET_IMPL_BITONIC_SORT  
);
```

T_SORTNET_IMPL

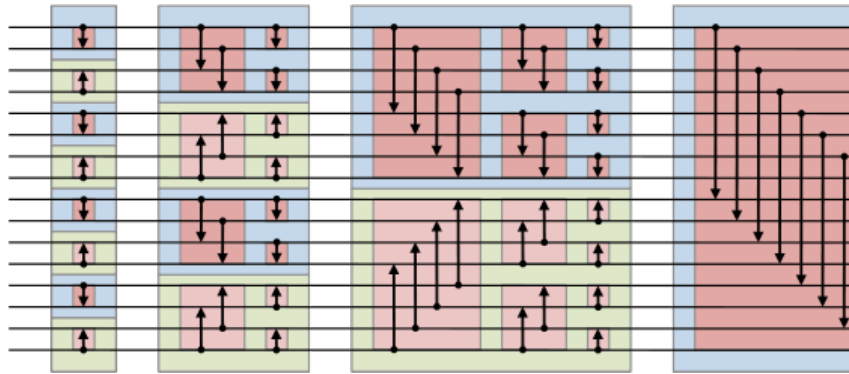
SORT_SORTNET_IMPL_ODDEVEN_SORT Instantiate a *PoC.sort.sortnet.OddEvenSort* sorting network.

SORT_SORTNET_IMPL_ODDEVEN_MERGESORT Instantiate a *PoC.sort.sortnet.OddEvenMergeSort* sorting network.

SORT_SORTNET_IMPL_BITONIC_SORT Instantiate a *PoC.sort.sortnet.BitonicSort* sorting network.

Source file: [sortnet.pkg.vhdl](#)

PoC.sort.sortnet.BitonicSort



This sorting network uses the *bitonic sort* algorithm.

Entity Declaration:

```

1  entity sortnet_BitonicSort is
2      generic (
3          INPUTS          : positive := 32;      -- input count
4          KEY_BITS         : positive := 32;      -- the first KEY_BITS of In_Data_
5          DATA_BITS       : positive := 64;      -- inclusive KEY_BITS
6          META_BITS        : natural  := 2;      -- additional bits, not sorted_
7          PIPELINE_STAGE_AFTER : natural := 2;    -- add a pipeline stage after n_
8          ADD_INPUT_REGISTERS : boolean := FALSE; --
9          ADD_OUTPUT_REGISTERS : boolean := TRUE  --
10     );
11     port (
12         Clock      : in  std_logic;
13         Reset      : in  std_logic;
14
15         Inverse     : in  std_logic := '0';
16
17         In_Valid    : in  std_logic;
18         In_IsKey    : in  std_logic;
19         In_Data     : in  T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
20         In_Meta     : in  std_logic_vector(META_BITS - 1 downto 0);
21
22         Out_Valid   : out std_logic;
23         Out_IsKey   : out std_logic;
24         Out_Data    : out T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
25         Out_Meta    : out std_logic_vector(META_BITS - 1 downto 0);
26     );
27 end entity;

```

Source file: `sort/sortnet/sortnet_BitonicSort.vhdl`

PoC.sort.sortnet.MergeSort_Streamed

Todo: No documentation available.

Entity Declaration:

```
1 entity sortnet_MergeSort_Streamed is
2   generic (
3     FIFO_DEPTH   : positive := 32;
4     KEY_BITS     : positive := 32;
5     DATA_BITS   : positive := 32;
6   );
7   port (
8     Clock      : in  std_logic;
9     Reset      : in  std_logic;
10
11     Inverse    : in  std_logic := '0';
12
13     In_Valid   : in  std_logic;
14     In_Data    : in  std_logic_vector(DATA_BITS - 1 downto 0);
15     In_SOF     : in  std_logic;
16     In_IsKey   : in  std_logic;
17     In_EOF     : in  std_logic;
18     In_Ack     : out std_logic;
19
20     Out_Sync   : out std_logic;
21     Out_Valid  : out std_logic;
22     Out_Data   : out std_logic_vector(DATA_BITS - 1 downto 0);
23     Out_SOF    : out std_logic;
24     Out_IsKey  : out std_logic;
25     Out_EOF    : out std_logic;
26     Out_Ack    : in  std_logic;
27   );
28 end entity;
```

Source file: sort/sortnet/sortnet_MergeSort_Streamed.vhdl

PoC.sortnet.OddEvenMergeSort

Todo: No documentation available.

Entity Declaration:

```
1 entity sortnet_OddEvenMergeSort is
2   generic (
3     INPUTS           : positive := 128;  -- input count
4     KEY_BITS         : positive := 32;    -- the first KEY_BITS of In_Data_
5     DATA_BITS       : positive := 32;    -- inclusive KEY_BITS
6     META_BITS        : natural  := 2;     -- additional bits, not sorted but_
7     PIPELINE_STAGE_AFTER : natural := 2;  -- add a pipeline stage after n_
8     ADD_INPUT_REGISTERS : boolean := FALSE; --
9     ADD_OUTPUT_REGISTERS : boolean := TRUE  --
10  );
11  port (
12    Clock      : in  std_logic;
13    Reset      : in  std_logic;
14
15    Inverse    : in  std_logic := '0';
```

(continues on next page)

(continued from previous page)

```

16      In_Valid      : in std_logic;
17      In_IsKey     : in std_logic;
18      In_Data      : in T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
19      In_Meta      : in std_logic_vector(META_BITS - 1 downto 0);
20
21
22      Out_Valid     : out std_logic;
23      Out_IsKey     : out std_logic;
24      Out_Data      : out T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
25      Out_Meta      : out std_logic_vector(META_BITS - 1 downto 0)
26  );
27  end entity;

```

Source file: sort/sortnet/sortnet_OddEvenMergeSort.vhdl

PoC.sort.sortnet.OddEvenSort

Todo: No documentation available.

Entity Declaration:

```

1  entity sortnet_OddEvenSort is
2      generic (
3          INPUTS          : positive := 8;      -- input count
4          KEY_BITS        : positive := 32;      -- the first KEY_BITS of In_Data_
5          --are used as a sorting criteria (key)
6          DATA_BITS      : positive := 32;      -- inclusive KEY_BITS
7          META_BITS       : natural  := 2;      -- additional bits, not sorted but_
8          --delayed as long as In_Data
9          PIPELINE_STAGE_AFTER : natural := 2;      -- add a pipeline stage after n_
10         --sorting stages
11         ADD_INPUT_REGISTERS  : boolean := FALSE; --
12         ADD_OUTPUT_REGISTERS : boolean := TRUE  --
13     );
14     port (
15         Clock      : in std_logic;
16         Reset      : in std_logic;
17
18         Inverse     : in std_logic := '0';
19
20         In_Valid    : in std_logic;
21         In_IsKey    : in std_logic;
22         In_Data     : in T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
23         In_Meta     : in std_logic_vector(META_BITS - 1 downto 0);
24
25         Out_Valid   : out std_logic;
26         Out_IsKey   : out std_logic;
27         Out_Data    : out T_SLM(INPUTS - 1 downto 0, DATA_BITS - 1 downto 0);
28         Out_Meta    : out std_logic_vector(META_BITS - 1 downto 0)
29     );
30 end entity;

```

Source file: sort/sortnet/sortnet_OddEvenSort.vhdl

PoC.sort.sortnet.Stream_Adapter

Todo: No documentation available.

Entity Declaration:

```

1 entity sortnet_Stream_Adapter is
2   generic (
3     STREAM_DATA_BITS      : positive      := 32;
4     STREAM_META_BITS      : positive      := 2;
5     SORTNET_IMPL          : T_SORTNET_IMPL := SORT_SORTNET_IMPL_ODDEVEN_MERGESORT;
6     SORTNET_SIZE          : positive      := 32;
7     SORTNET_KEY_BITS      : positive      := 32;
8     SORTNET_DATA_BITS     : natural       := 32;
9     INVERSE               : boolean       := FALSE
10  );
11  port (
12    Clock      : in  std_logic;
13    Reset      : in  std_logic;
14
15    In_Valid   : in  std_logic;
16    In_IsKey   : in  std_logic;
17    In_Data    : in  std_logic_vector (STREAM_DATA_BITS - 1 downto 0);
18    In_Meta    : in  std_logic_vector (STREAM_META_BITS - 1 downto 0);
19    In_Ack     : out std_logic;
20
21    Out_Valid  : out std_logic;
22    Out_IsKey  : out std_logic;
23    Out_Data   : out std_logic_vector (STREAM_DATA_BITS - 1 downto 0);
24    Out_Meta   : out std_logic_vector (STREAM_META_BITS - 1 downto 0);
25    Out_Ack    : in  std_logic
26  );
27 end entity;
```

Source file: [sort/sortnet/sortnet_Stream_Adapter.vhdl](#)

PoC.sort.sortnet.Stream_Adapter2

Todo: No documentation available.

Entity Declaration:

```

1 entity sortnet_Stream_Adapter2 is
2   generic (
3     STREAM_DATA_BITS      : positive      := 32;
4     STREAM_META_BITS      : positive      := 2;
5     DATA_COLUMNS         : positive      := 2;
6     SORTNET_IMPL          : T_SORTNET_IMPL := SORT_SORTNET_IMPL_ODDEVEN_MERGESORT;
7     SORTNET_SIZE          : positive      := 32;
8     SORTNET_KEY_BITS      : positive      := 32;
9     SORTNET_DATA_BITS     : natural       := 32;
10    SORTNET_REG_AFTER      : natural       := 2;
11    MERGENET_STAGES        : positive      := 2
12  );
13  port (
```

(continues on next page)

(continued from previous page)

```

14   Clock      : in  std_logic;
15   Reset      : in  std_logic;
16
17   Inverse     : in  std_logic      := '0';
18
19   In_Valid    : in  std_logic;
20   In_Data     : in  std_logic_vector (STREAM_DATA_BITS - 1 downto 0);
21   In_Meta     : in  std_logic_vector (STREAM_META_BITS - 1 downto 0);
22   In_SOF      : in  std_logic;
23   In_IsKey    : in  std_logic;
24   In_EOF      : in  std_logic;
25   In_Ack      : out std_logic;
26
27   Out_Valid   : out std_logic;
28   Out_Data    : out std_logic_vector (STREAM_DATA_BITS - 1 downto 0);
29   Out_Meta    : out std_logic_vector (STREAM_META_BITS - 1 downto 0);
30   Out_SOF     : out std_logic;
31   Out_IsKey   : out std_logic;
32   Out_EOF     : out std_logic;
33   Out_Ack     : in  std_logic
34 );
35 end entity;
```

Source file: `sort/sortnet/sortnet_Stream_Adapter2.vhdl`

PoC.sort.sortnet.Transform

Todo: No documentation available.

Entity Declaration:

```

1  entity sortnet_Transform is
2    generic (
3      ROWS      : positive      := 16;
4      COLUMNS  : positive      := 4;
5      DATA_BITS : positive      := 8
6    );
7    port (
8      Clock      : in  std_logic;
9      Reset      : in  std_logic;
10
11      In_Valid    : in  std_logic;
12      In_Data     : in  T_SLM(ROWS - 1 downto 0, DATA_BITS - 1 downto 0);
13      In_SOF      : in  std_logic;
14      In_EOF      : in  std_logic;
15
16      Out_Valid   : out std_logic;
17      Out_Data    : out T_SLM(COLUMNS - 1 downto 0, DATA_BITS - 1 downto 0);
18      Out_SOF     : out std_logic;
19      Out_EOF     : out std_logic
20    );
21  end entity;
```

Source file: `sort/sortnet/sortnet_Transform.vhdl`

7.14.2 PoC.sort.ExpireList

Todo: No documentation available.

Entity Declaration:

Source file: `sort/sort_ExpireList.vhdl`

7.14.3 PoC.sort.InsertSort

Todo: No documentation available.

Entity Declaration:

Source file: `sort/sort_InsertSort.vhdl`

7.14.4 PoC.sort.LeastFrequentlyUsed

Todo: No documentation available.

Entity Declaration:

Source file: `sort/sort_LeastFrequentlyUsed.vhdl`

7.14.5 PoC.sort.lru_cache

This is an optimized implementation of `sort_lru_list` to be used for caches. Only keys are stored within this list, and these keys are the index of the cache lines. The list initially contains all indices from 0 to `ELEMENTS-1`. The least-recently used index `KeyOut` is always valid.

The first outputted least-recently used index will be `ELEMENTS-1`.

The inputs `Insert`, `Free`, `KeyIn`, and `Reset` are synchronous to the rising-edge of the clock `clock`. All control signals are high-active.

Supported operations:

- **Insert:** Mark index `KeyIn` as recently used, e.g., when a cache-line was accessed.
- **Free:** Mark index `KeyIn` as least-recently used. Apply this operation, when a cache-line gets invalidated.

Entity Declaration:


```

1 entity sort_lru_cache is
2   generic (
3     ELEMENTS      : positive      := 32
4   );
5   port (
6     Clock      : in std_logic;
7     Reset      : in std_logic;
8
9     Insert     : in std_logic;
10    Free        : in std_logic;
11    KeyIn       : in std_logic_vector(log2ceilnz(ELEMENTS) - 1 downto 0);
12
13    KeyOut      : out std_logic_vector(log2ceilnz(ELEMENTS) - 1 downto 0)
14  );
15 end entity;

```

Source file: `sort/sort_lru_cache.vhdl`

7.14.6 PoC.sort.lru_list

List storing (key, value) pairs. The least-recently inserted pair is outputted on DataOut if Valid = '1'. If Valid = '0', then the list empty.

The inputs Insert, Remove, DataIn, and Reset are synchronous to the rising-edge of the clock clock. All control signals are high-active.

Supported operations:

- **Insert:** Insert DataIn as recently used (key, value) pair. If key is already within the list, then the corresponding value is updated and the pair is moved to the recently used position.
- **Remove:** Remove (key, value) pair with the given key. The list is not modified if key is not within the list.

Entity Declaration:

```

1 entity sort_lru_list is
2   generic (
3     ELEMENTS      : positive      := 16;
4     KEY_BITS      : positive      := 4;
5     DATA_BITS    : positive      := 8;
6     INITIAL_ELEMENTS : T_SLM      := (0 to 15 => (0_
7     to 7 => '0'));
8     INITIAL_VALIDS : std_logic_vector := (0 to 15 => '0')
9   );
10  port (
11    Clock      : in std_logic;
12    Reset      : in std_logic;
13
14    Insert     : in std_logic;
15    Remove     : in std_logic;
16    DataIn     : in std_logic_vector(DATA_BITS - 1 downto 0);
17
18    Valid      : out std_logic;
19    DataOut    : out std_logic_vector(DATA_BITS - 1 downto 0)
20  );
21 end entity;

```

Source file: `sort/sort_lru_list.vhdl`

7.15 PoC.xil

This namespace is for Xilinx specific modules.

Sub-Namespaces

- *PoC.xil.mig*
- *PoC.xil.reconfig*

Entities

- *PoC.xil.BSCAN*
- *PoC.xil.ChipScopeICON*
- *PoC.xil.DRP_BusMux*
- *PoC.xil.DRP_BusSync*
- *PoC.xil.ICAP*
- *PoC.xil.Reconfigurator*
- *PoC.xil.SystemMonitor*
- IP:xil_SystemMonitor_Virtex6
- IP:xil_SystemMonitor_Series7

7.15.1 PoC.xil.mig

The namespace `PoC.xil.mig` offers pre-configured memory controllers generated with Xilinx's Memory Interface Generator (MIG).

- **for Spartan-6 boards:**
 - *mig_Atlys_1x128* - A DDR2 memory controller for the Digilent Atlys board.
- **for Kintex-7 boards:**
 - *mig_KC705_MT8JTF12864HZ_1G6* - A DDR3 memory controller for the Xilinx KC705 board.
- **for Virtex-7 boards:**

mig_Atlys_1x128

This DDR2 memory controller is pre-configured for the Digilent Atlys development board. The board is equipped with a single 1 GiBit DDR2 memory chip (128 MiByte) from MIRA (MIRA P3R1GE3EGF G8E DDR2).

Run the following two steps to create the IP core:

1. Generate the source files from the IP core using Xilinx MIG and afterwards patch them

```
PS> .\poc.ps1 coregen PoC.xil.mig.Atlys_1x128 --board=Atlys
```
2. Compile the patched sources into a ready to use netlist (*.ngc) and constraint file (*.ucf)

```
PS> .\poc.ps1 xst PoC.xil.mig.Atlys_1x128 --board=Atlys
```

See also:

Using PoC -> Synthesis For how to run Core Generator and XST from PoC.

mig_KC705_MT8JTF12864HZ_1G6

This DDR2 memory controller is pre-configured for the Xilinx KC705 development board. The board is equipped with a single 1 GiBit DDR3 memory chip (128 MiByte) from Micron Technology (MT8JTF12864HZ-1G6G1).

Run the following two steps to create the IP core:

1. Generate the source files from the IP core using Xilinx MIG and afterwards patch them PS> .\poc.ps1 coregen PoC.xil.mig.KC705_MT8JTF12864HZ_1G6 --board=KC705
2. Compile the patched sources into a ready to use netlist (*.ngc) and constraint file (*.ucf) PS> .\poc.ps1 xst PoC.xil.mig.KC705_MT8JTF12864HZ_1G6 --board=KC705

See also:

Using PoC -> Synthesis For how to run Core Generator and XST from PoC.

7.15.2 PoC.xil.reconfig

These are reconfig entities. ...

Entities

- *PoC.xil.reconfig.icap_fsm*
- *PoC.xil.reconfig.icap_wrapper*

PoC.xil.reconfig.icap_fsm

This module parses the data stream to the Xilinx “Internal Configuration Access Port” (ICAP) primitives to generate control signals. Tested on:

- Virtex-6
- Virtex-7

Entity Declaration:

```

1  entity reconfig_icap_fsm is
2      port (
3          clk          : in  std_logic;
4          reset        : in  std_logic;          -- high-active reset
5          -- interface to connect to the icap
6          icap_in       : out std_logic_vector(31 downto 0); -- data that will go into_
7          --the icap
8          icap_out       : in  std_logic_vector(31 downto 0); -- data from the icap
9          icap_csb       : out std_logic;
10         icap_rw        : out std_logic;
11
12         -- data interface, no internal fifos
13         in_data         : in  std_logic_vector(31 downto 0); -- new configuration data
14         in_data_valid   : in  std_logic;          -- input data is valid
15         in_data_rden    : out std_logic;          -- possible to send data
16         out_data        : out std_logic_vector(31 downto 0); -- data read from the fifo
17         out_data_valid  : out std_logic;          -- data from icap is valid
18         out_data_full   : in  std_logic;          -- receiving buffer is full, halt_
19         --icap
20
21         -- control structures
22         status          : out std_logic_vector(31 downto 0) -- status vector
23     );
24 end reconfig_icap_fsm;
```

Source file: `xil/reconfig/reconfig_icap_fsm.vhdl`

PoC.xil.reconfig.icap_wrapper

This module was designed to connect the Xilinx “Internal Configuration Access Port” (ICAP) to a PCIe endpoint on a Dini board. Tested on:

tbd

Entity Declaration:

```

1  entity reconfig_icap_wrapper is
2      generic (
3          MIN_DEPTH_OUT      : positive := 256;
4          MIN_DEPTH_IN       : positive := 256
5      );
6      port (
7          clk      : in  std_logic;
8          reset    : in  std_logic;
9          clk_icap : in  std_logic;    -- clock signal for ICAP, max 100 MHz (double_
↳check with manual)
10
11         icap_busy : out std_logic;    -- the ICAP is processing the data
12         icap_readback : out std_logic; -- high during a readback
13         icap_partial_res: out std_logic; -- high during reconfiguration
14
15         -- data in
16         write_put      : in  std_logic;
17         write_full     : out std_logic;
18         write_data     : in  std_logic_vector(31 downto 0);
19         write_done     : in  std_logic;    -- high pulse/edge after all data was written
20
21         -- data out
22         read_got       : in  std_logic;
23         read_valid     : out std_logic;
24         read_data      : out std_logic_vector(31 downto 0)
25     );
26 end reconfig_icap_wrapper;

```

Source file: `xil/reconfig/reconfig_icap_wrapper.vhdl`

7.15.3 PoC.xil Package

This package holds all component declarations for this namespace.

Source file: `xil.pkg.vhdl`

7.15.4 PoC.xil.BSCAN

This module wraps Xilinx “Boundary Scan” (JTAG) primitives in a generic module. Supported devices are:

- Spartan-3, Spartan-6
- Virtex-5, Virtex-6
- Series-7 (Artix-7, Kintex-7, Virtex-7, Zynq-7000)

Entity Declaration:

```

1  entity xil_BSCAN is
2      generic (
3          JTAG_CHAIN          : natural;
4          DISABLE_JTAG       : boolean    := FALSE
5      );
6      port (
7          Reset              : out std_logic;
8          RunTest            : out std_logic;
9          Sel                : out std_logic;
10         Capture            : out std_logic;
11         drck               : out std_logic;
12         Shift              : out std_logic;
13         Test_Clock         : out std_logic;
14         Test_DataIn        : out std_logic;
15         Test_DataOut       : in  std_logic;
16         Test_ModeSelect    : out std_logic;
17         Update             : out std_logic
18     );
19 end entity;

```

Source file: `xil/xil_BSCAN.vhdl`

7.15.5 PoC.xil.ChipScopeICON

This module wraps 15 ChipScope ICON IP core netlists generated from ChipScope ICON xco files. The generic parameter PORTS selects the appropriate ICON instance with 1 to 15 ICON ControlBus ports. Each ControlBus port is of type T_XIL_CHIPSOCPE_CONTROL and of mode inout.

Compile required CoreGenerator IP Cores to Netlists with PoC

Please use the provided Xilinx ISE compile command `ise` in PoC to recreate the needed source and netlist files on your local machine.

```

cd PoCRoot
.\poc.ps1 ise PoC.xil.ChipScopeICON --board=KC705

```

Entity Declaration:

```

1  entity xil_ChipScopeICON is
2      generic (
3          PORTS              : positive
4      );
5      port (
6          ControlBus        : inout T_XIL_CHIPSOCPE_CONTROL_VECTOR(PORTS - 1 downto 0)
7      );
8  end entity;

```

See also:

Using PoC -> Synthesis For how to run synthesis with PoC and CoreGenerator.

Source file: `xil/xil_ChipScopeICON.vhdl`

7.15.6 PoC.xil.DRP_BusMux

Todo: No documentation available.

Entity Declaration:

Source file: `xil/xil_DRP_BusMux.vhdl`

7.15.7 PoC.xil.DRP_BusSync

Todo: No documentation available.

Entity Declaration:

Source file: `xil/xil_DRP_BusSync.vhdl`

7.15.8 PoC.xil.ICAP

This module wraps Xilinx “Internal Configuration Access Port” (ICAP) primitives in a generic module. Supported devices are:

- Spartan-6
- Virtex-4, Virtex-5, Virtex-6
- Series-7 (Artix-7, Kintex-7, Virtex-7, Zynq-7000)

Entity Declaration:

```
1 entity xil_ICAP is
2   generic (
3     ICAP_WIDTH  : string := "X32";           -- Specifies the input and output data_
↳width to be used
4
5     -- Spartan 6: fixed to 16 bit
6     -- Virtex 4:  X8 or X32
7     -- Rest:  X8, X16, X32
8     DEVICE_ID   : bit_vector := X"1234567";  -- pre-programmed Device ID value_
↳for simulation
9
10    -- supported by Spartan 6, Virtex 6 and above
11    SIM_CFG_FILE_NAME : string := "NONE"      -- Raw Bitstream (RBT) file to be_
↳parsed by the simulation model
12
13    -- supported by Spartan 6, Virtex 6 and above
14  );
15  port (
16    clk      : in std_logic;                  -- up to 100 MHz (Virtex-6 and above, Virtex-
↳5??)
17    disable   : in std_logic;                  -- low active enable -> high active disable
18    rd_wr     : in std_logic;                  -- 0 - write, 1 - read
19    busy      : out std_logic;                  -- on Series-7 devices always '0'
20    data_in   : in std_logic_vector(31 downto 0); -- on Spartan-6 only 15 downto 0
21    data_out  : out std_logic_vector(31 downto 0); -- on Spartan-6 only 15 downto 0
```

(continues on next page)

(continued from previous page)

```

19     );
20 end entity;

```

Source file: xil/xil_ICAP.vhdl

7.15.9 PoC.xil.Reconfigurator

Many complex primitives in a Xilinx device offer a Dynamic Reconfiguration Port (DRP) to reconfigure a primitive at runtime without reconfiguring the whole FPGA.

This module is a DRP master that can be pre-configured at compile time with different configuration sets. The configuration sets are mapped into a ROM. The user can select a stored configuration with `ConfigSelect`. Sending a strobe to `Reconfig` will start the reconfiguration process. The operation completes with another strobe on `ReconfigDone`.

Entity Declaration:

```

1  entity xil_Reconfigurator is
2      generic (
3          DEBUG                : boolean                := FALSE;
4          CLOCK_FREQ           : FREQ                   := 100 MHz;
5          CONFIG_ROM           : in  T_XIL_DRP_CONFIG_ROM := (0 downto 0 => C_XIL_DRP_CONFIG_
6          SET_EMPTY)          --
7      );
8      port (
9          Clock                : in  std_logic;
10         Reset                 : in  std_logic;
11         Reconfig              : in  std_logic;
12         ReconfigDone          : out std_logic;
13         ConfigSelect          : in  std_logic_vector(log2ceilnz(CONFIG_ROM'length) - 1_
14         downto 0);
15         DRP_en                : out std_logic;
16         DRP_Address           : out T_XIL_DRP_ADDRESS;
17         DRP_we                : out std_logic;
18         DRP_DataIn            : in  T_XIL_DRP_DATA;
19         DRP_DataOut           : out T_XIL_DRP_DATA;
20         DRP_Ack               : in  std_logic
21     );
22 end entity;

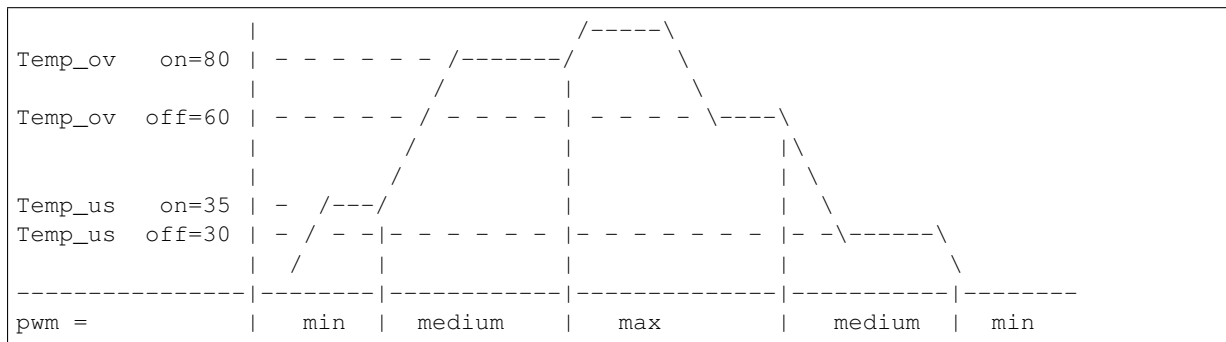
```

Source file: xil/xil_Reconfigurator.vhdl

7.15.10 PoC.xil.SystemMonitor

This module wraps a SYSMON or XADC to report if preconfigured temperature values are overrun. The XADC was formerly known as “System Monitor”.

Temperature Curve



Entity Declaration:

```

1  entity xil_SystemMonitor is
2      port (
3          Reset                : in  std_logic;           -- Reset signal for the System_
4          ↪Monitor control logic
5          Alarm_UserTemp       : out std_logic;           -- Temperature-sensor alarm output
6          Alarm_OverTemp       : out std_logic;           -- Over-Temperature alarm output
7          Alarm                : out std_logic;           -- OR'ed output of all the Alarms
8          VP                   : in  std_logic;           -- Dedicated Analog Input Pair
9          VN                   : in  std_logic
10     );
11 end entity;
```

Source file: xil/xil_SystemMonitor.vhdl

Third Party Libraries

The PoC-Library is shipped with different third party libraries, which are located in the `<PoCRoot>/lib/` folder. This document lists all these libraries, their websites and licenses.

8.1 Cocotb

`Cocotb` is a coroutine based cosimulation library for writing VHDL and Verilog testbenches in Python.

Folder:	<code><PoCRoot>/lib/cocotb/</code>
Copyright:	Copyright © 2013, Potential Ventures Ltd., SolarFlare Communications Inc.
License:	<i>Revised BSD License (local copy)</i>
Documentation:	http://cocotb.readthedocs.org/
Source:	https://github.com/potentialventures/cocotb

8.2 OSVVM

Open Source VHDL Verification Methodology (OS-VVM) is an intelligent testbench methodology that allows mixing of “Intelligent Coverage” (coverage driven randomization) with directed, algorithmic, file based, and constrained random test approaches. The methodology can be adopted in part or in whole as needed. With OSVVM you can add advanced verification methodologies to your current testbench without having to learn a new language or throw out your existing testbench or testbench models.

Folder:	<code><PoCRoot>/lib/osvmm/</code>
Copyright:	Copyright © 2012-2016 by SynthWorks Design Inc.
License:	<i>Artistic License 2.0 (local copy)</i>
Website:	http://osvmm.org/
Source:	https://github.com/JimLewis/OSVVM

8.3 UVVM

The Open Source **UVVM (Universal VHDL Verification Methodology) - VVC (VHDL Verification Component) Framework** for making structured VHDL testbenches for verification of FPGA. UVVM consists currently of: Utility Library, VVC Framework and Verification IPs (VIP) for various protocols.

For what do I need this VVC Framework? The VVC Framework is a VHDL Verification Component system that allows multiple interfaces on a DUT to be stimulated/handled simultaneously in a very structured manner, and controlled by a very simple to understand software like a test sequencer. VVC Framework is unique as an open source VHDL approach to building a structured testbench architecture using Verification components and a simple protocol to access these. As an example a simple command like `uart_expect (UART_VVCT, my_data)`, or `axilite_write (AXILITE_VVCT, my_addr, my_data, my_message)` will automatically tell the respective VVC (for UART or AXI-Lite) to execute the `uart_receive()` or `axilite_write()` BFM respectively.

Folder:	<PoCRoot>\lib\uvvm\
Copyright:	Copyright © 2016 by Bitvis AS
License:	<i>The MIT License (local copy)</i>
Website:	http://bitvis.no/
Source:	https://github.com/UVVM/UVVM_All

8.4 VUnit

VUnit is an open source unit testing framework for VHDL released under the terms of *Mozilla Public License, v. 2.0*. It features the functionality needed to realize continuous and automated testing of your VHDL code. VUnit doesn't replace but rather complements traditional testing methodologies by supporting a "test early and often" approach through automation.

Folder:	<PoCRoot>\lib\vunit\
Copyright:	Copyright © 2014-2016, Lars Asplund lars.anders.asplund@gmail.com
License:	<i>Mozilla Public License, Version 2.0 (local copy)</i>
Website:	https://vunit.github.io/
Source:	https://github.com/VUnit/vunit

8.5 Updating Linked Git Submodules

The third party libraries are embedded as Git submodules. So if the PoC-Library was not cloned with option `--recursive` it's required to run the sub-module initialization manually:

8.5.1 On Linux

```
cd PoCRoot
git submodule init
git submodule update
```

We recommend to rename the default remote repository name from 'origin' to 'github'.

```
cd PoCRoot\lib\
```

Todo: write Bash code for Linux

8.5.2 On OS X

Please see the Linux instructions.

8.5.3 On Windows

```
cd PoCRoot
git submodule init
git submodule update
```

We recommend to rename the default remote repository name from ‘origin’ to ‘github’.

```
cd PoCRoot\lib\
foreach($dir in (dir -Directory)) {
    cd $dir
    git remote rename origin github
    cd ..
}
```


9.1 IP Core Constraint Files

- fifo
- misc
 - sync
- net
 - eth

9.1.1 fifo

- fifo_ic_got

fifo_ic_got

9.1.2 misc

- sync

sync

- sync_Bits
- sync_Reset
- sync_Vector
- sync_Command

`fifo_ic_got`

`fifo_ic_got`

`fifo_ic_got`

`fifo_ic_got`

9.1.3 net

- `eth`

`eth`

- `eth_RSLayer_GMII_GMII_KC705`
- `eth_RSLayer_GMII_GMII_ML505`
- `eth_RSLayer_GMII_GMII_ML605`

`eth_RSLayer_GMII_GMII_KC705`

`eth_RSLayer_GMII_GMII_ML505`

`eth_RSLayer_GMII_GMII_ML605`

9.2 Board Constraint Files

- Altera Boards
 - Cyclone III
 - Stratix IV
 - Stratix V
- Lattice Boards
- Xilinx Boards
 - Artix-7
 - Kintex-7
 - Spartan-3 Boards
 - Spartan-6 Boards
 - Virtex-5
 - Virtex-6
 - Virtex-7
 - Zynq-7000

9.2.1 Altera

- Cyclone III * DE0 * DE0 nano
- Stratix IV * DE4
- Stratix V * DE5

Cyclone III

- DE0
- DE0 nano

ECP5 Versa

ECP5 Versa

Stratix IV

- DE4

DE4

Stratix V

- DE5

DE5

9.2.2 Lattice

- ECP5 * ECP5 Versa

ECP5

- ECP5 Versa

ECP5 Versa

9.2.3 Xilinx

- **Spartan-3 Boards**
 - Spartan-3 Starter Kit (S3SK)
 - Spartan-3E Starter Kit (S3ESK)
- **Spartan-6 Boards**
 - Atlys
- **Artix-7**
 - AC701
- **Kintex-7**

- KC705
- **Virtex-5**
 - ML505
 - ML506
 - XUPV5
- **Virtex-6**
 - ML605
- **Virtex-7**
 - VC707
- **Zynq-7000**
 - ZC706
 - ZedBoard

Spartan-3

- Spartan-3 Starter Kit (S3SK)
- Spartan-3E Starter Kit (S3ESK)

S3SK

S3ESK

Spartan-6

- Atlys

Atlys

Artix-7

- AC701

AC701

Kintex-7

- KC705

KC705

Virtex-5

- ML505
- ML506
- XUPV5

ML505

ML506

XUPV5

Virtex-6

- ML605

ML605

Virtex-7

- VC707

VC707

Zynq-7000

- ZC706
- ZedBoard

ZC706

ZedBoard

CHAPTER 10

Tool Chain Specifics

Attention: This page is under construction.

10.1 Aldec Active-HDL

Todo:

- No GUI mode supported
 - VHDL-2008 parser bug in Active-HDL 10.3
-

10.2 Mentor QuestaSim

Special feature: embedded poc prodecures to recompile relaunch, rerun and save waveforms. . .

10.3 Xilinx ISE

- Describe the `use_new_parser yes` option

10.4 Xilinx Vivado

- Describe the `vivado` branch (Git).

CHAPTER 11

Examples

Note: Under construction.

PoC-Exmaples repository on GitHub.

Part III

References

Command Reference

This is the command line option reference for all provided scripts (Bash, PowerShell, Perl) and programs (Python) shipped with PoC.

12.1 PoC Wrapper Scripts

The PoC main program **PoC.py** requires a prepared environment, which needs to be setup by platform specific wrapper scripts written as shell scripts language (PowerShell/Bash). Moreover, the main program requires a supported Python version, so the wrapper script will search the best matching language environment.

The wrapper script offers the ability to hook in user-defined scripts to prepared (before) and clean up the environment (after) a PoC execution. E.g. it's possible to load the environment variable `LM_LICENSE_FILE` for the FlexLM license manager.

Created Environment Variables

PoCRootDirectory

The path to PoC's root directory.

12.1.1 poc.ps1

`PoC.ps1` is the wrapper for the Windows platform using a PowerShell script. It can be debugged by adding the command line switch `-D`. All parameters are passed to `PoC.py`.

-D

Enabled debug mode in the wrapper script.

Other arguments

All remaining arguments are passed to `PoC.py`.

12.1.2 poc.sh

`PoC.sh` is the wrapper for Linux and Unix platforms using a Bash script. It can be debugged by adding the command line switch `-D`. All parameters are passed to `PoC.py`.

-D

Enabled debug mode in the wrapper script.

Other arguments

All remaining arguments are passed to `PoC.py`.

12.2 Main Program (`PoC.py`)

The main program `PoC.py` expects the environment variable `PoCRootDirectory` to be set.

12.3 Pre-compile Scripts

The following scripts can be used to pre-compile vendor's primitives or third party libraries. Pre-compile vendor primitives are required for vendor specific simulations or if no generic IP core implementation is available. Third party libraries are usually used as simulation helpers and thus needed by many testbenches.

The pre-compiled packages and libraries are stored in the directory `/temp/precompiled/`. Per simulator, one `<simulator>/` sub-directory is created. Each simulator directory in turn contains library directories, which may be grouped by the library vendor's name: `[<vendor>/]<library>/`.

So for example: *OSVVM* pre-compiled with GHDL is stored in `/temp/precompiled/ghdl/osvvm/`. Note OSVVM is a single library and thus no vendor directory is used to group the generated files. GHDL will also create VHDL language revision sub-directories like `v93/` or `v08/`.

Currently the provided scripts support 2 simulator targets and one combined target:

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

The GHDL simulator distinguishes various VHDL language revisions and thus can pre-compile the source for these language revisions into separate output directories. The command line switch `-All/--all` will build the libraries for all major VHDL revisions (93, 2008).

Pre-compile Altera Libraries

12.3.1 compile-altera.sh

This script pre-compiles the Altera primitives. This script will generate all outputs into a `altera` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- help**
Show the embedded help page(s).
- clean**
Clean up directory before analyzing.
- all**
Pre-compile all libraries and packages for all simulators.
- ghdl**
Pre-compile the Altera Quartus libraries for GHDL.
- questa**
Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

- vhd193**
For GHDL only: Set VHDL Standard to '93.
- vhd12008**
For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Altera.

12.3.2 compile-altera.ps1

This script pre-compiles the Altera primitives. This script will generate all outputs into a `altera` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- Help**
Show the embedded help page(s).
- Clean**
Clean up directory before analyzing.

-All

Pre-compile all libraries and packages for all simulators.

-GHDL

Pre-compile the Altera Quartus libraries for GHDL.

-Questa

Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

-VHDL93

For GHDL only: Set VHDL Standard to '93.

-VHDL2008

For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Altera.

Pre-compile Lattice Libraries

12.3.3 compile-lattice.sh

This script pre-compiles the Lattice primitives. This script will generate all outputs into a `lattice` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

--help

Show the embedded help page(s).

--clean

Clean up directory before analyzing.

--all

Pre-compile all libraries and packages for all simulators.

--ghdl

Pre-compile the Altera Quartus libraries for GHDL.

--questa

Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL**--vhd193**

For GHDL only: Set VHDL Standard to '93.

--vhd12008

For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Lattice.

12.3.4 compile-lattice.ps1

This script pre-compiles the Lattice primitives. This script will generate all outputs into a `lattice` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options**-Help**

Show the embedded help page(s).

-Clean

Clean up directory before analyzing.

-All

Pre-compile all libraries and packages for all simulators.

-GHDL

Pre-compile the Altera Quartus libraries for GHDL.

-Questa

Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL**-VHDL93**

For GHDL only: Set VHDL Standard to '93.

-VHDL2008

For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Lattice.

Pre-compile OSVVM Libraries

12.3.5 compile-osvvm.sh

This script pre-compiles the OSVVM packages. This script will generate all outputs into a `osvvm` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

--help

Show the embedded help page(s).

--clean

Clean up directory before analyzing.

--all

Pre-compile all libraries and packages for all simulators.

--ghdl

Pre-compile the Altera Quartus libraries for GHDL.

--questa

Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

--vhd193

For GHDL only: Set VHDL Standard to '93.

--vhd12008

For GHDL only: Set VHDL Standard to '08.

12.3.6 compile-osvvm.ps1

This script pre-compiles the OSVVM packages. This script will generate all outputs into a `osvvm` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

-Help

Show the embedded help page(s).

-Clean

Clean up directory before analyzing.

-All

Pre-compile all libraries and packages for all simulators.

-GHDL

Pre-compile the Altera Quartus libraries for GHDL.

-Questa

Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

-VHDL93

For GHDL only: Set VHDL Standard to '93.

-VHDL2008

For GHDL only: Set VHDL Standard to '08.

Pre-compile UVVM Libraries

12.3.7 compile-uvvm.sh

This script pre-compiles the UVVM framework. This script will generate all outputs into a `uvvm` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

--help

Show the embedded help page(s).

--clean

Clean up directory before analyzing.

- all**
Pre-compile all libraries and packages for all simulators.
- ghdl**
Pre-compile the Altera Quartus libraries for GHDL.
- questa**
Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

- vhd193**
For GHDL only: Set VHDL Standard to '93.
- vhd12008**
For GHDL only: Set VHDL Standard to '08.

12.3.8 compile-uvvm.ps1

This script pre-compiles the UVVM framework. This script will generate all outputs into a `uvvm` directory.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- Help**
Show the embedded help page(s).
- Clean**
Clean up directory before analyzing.
- All**
Pre-compile all libraries and packages for all simulators.
- GHDL**
Pre-compile the Altera Quartus libraries for GHDL.
- Questa**
Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

- VHDL93**
For GHDL only: Set VHDL Standard to '93.
- VHDL2008**
For GHDL only: Set VHDL Standard to '08.

Pre-compile Xilinx ISE Libraries

12.3.9 compile-xilinx-ise.sh

This script pre-compiles the Xilinx primitives. Because Xilinx offers two tool chains (ISE, Vivado), this script will generate all outputs into a `xilinx-ise` directory and a symlink to `xilinx` will be created. This eases the coexistence of pre-compiled primitives from ISE and Vivado.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- help**
Show the embedded help page(s).
- clean**
Clean up directory before analyzing.
- all**
Pre-compile all libraries and packages for all simulators.
- ghdl**
Pre-compile the Altera Quartus libraries for GHDL.
- questa**
Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

- vhd193**
For GHDL only: Set VHDL Standard to '93.
- vhd12008**
For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted SecureIP primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Xilinx.

12.3.10 compile-xilinx-ise.ps1

This script pre-compiles the Xilinx primitives. Because Xilinx offers two tool chains (ISE, Vivado), this script will generate all outputs into a `xilinx-ise` directory and a symlink to `xilinx` will be created. This eases the coexistence of pre-compiled primitives from ISE and Vivado. The symlink can be changed by the user or via `-ReLink`.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- Help**
Show the embedded help page(s).
- Clean**
Clean up directory before analyzing.
- All**
Pre-compile all libraries and packages for all simulators.
- GHDL**
Pre-compile the Altera Quartus libraries for GHDL.
- Questa**
Pre-compile the Altera Quartus libraries for QuestaSim.
- ReLink**
Change the 'xilinx' symlink to 'xilinx-ise'.

Additional Options for GHDL

- VHDL93**
For GHDL only: Set VHDL Standard to '93.
- VHDL2008**
For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted SecureIP primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Xilinx.

Pre-compile Xilinx Vivado Libraries

12.3.11 compile-xilinx-vivado.sh

This script pre-compiles the Xilinx primitives. Because Xilinx offers two tool chains (ISE, Vivado), this script will generate all outputs into a `xilinx-vivado` directory and a symlink to `xilinx` will be created. This eases the coexistence of pre-compiled primitives from ISE and Vivado.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- help**
Show the embedded help page(s).
- clean**
Clean up directory before analyzing.
- all**
Pre-compile all libraries and packages for all simulators.
- ghdl**
Pre-compile the Altera Quartus libraries for GHDL.
- questa**
Pre-compile the Altera Quartus libraries for QuestaSim.

Additional Options for GHDL

- vhd193**
For GHDL only: Set VHDL Standard to '93.
- vhd12008**
For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted SecureIP primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Xilinx.

12.3.12 compile-xilinx-vivado.ps1

This script pre-compiles the Xilinx primitives. Because Xilinx offers two tool chains (ISE, Vivado), this script will generate all outputs into a `xilinx-vivado` directory and a symlink to `xilinx` will be created. This eases the coexistence of pre-compiled primitives from ISE and Vivado. The symlink can be changed by the user or via `-ReLink`.

Supported Simulators

Target	Description
All	pre-compile for all simulators
GHDL	pre-compile for the GHDL simulator
Questa	pre-compile for Metor Graphics QuestaSim

Command Line Options

- Help**
Show the embedded help page(s).
- Clean**
Clean up directory before analyzing.
- All**
Pre-compile all libraries and packages for all simulators.
- GHDL**
Pre-compile the Altera Quartus libraries for GHDL.
- Questa**
Pre-compile the Altera Quartus libraries for QuestaSim.
- ReLink**
Change the 'xilinx' symlink to 'xilinx-vivado'.

Additional Options for GHDL

- VHDL93**
For GHDL only: Set VHDL Standard to '93.
- VHDL2008**
For GHDL only: Set VHDL Standard to '08.

GHDL Notes

Not all primitives and macros are available as plain VHDL source code. Encrypted SecureIP primitives and netlists cannot be pre-compiled by GHDL.

QuestaSim Notes

The pre-compilation for QuestaSim uses a build in program from Xilinx.

Contents of this Page

- *Overview*
- *Database Structure*
 - *Nodes*
 - *References*
 - *Options*
 - *Values*
 - *Value Interpolation*
 - *Node Interpolation*
 - *Root Nodes*
- *Supported Options*
- *Files in detail*
 - *config.structure.ini*
 - *config.entity.ini*
 - *config.boards.ini*
 - *config.private.ini*
- *User Defined Variables*

13.1 Overview

PoC internal IP core database uses INI files and advanced interpolation rules provided by `ExtendedConfigParser`. The database consists of 5 *.ini files:

- **`pyconfig.boards.ini`** This file contains all known *FPGA boards* and *FPGA devices*.

- **pyconfig.defaults.ini** This file contains all default options and values for all supported node types.
- **pyconfig.entity.ini** This file contains all IP cores (entities) and their corresponding testbench or netlist settings.
- **pyconfig.private.ini** This file is created by `.\poc.ps1 configure` and contains settings for the local PoC installation. This file must not be shared with other PoC instances. See [Configuring PoC's Infrastructure](#) on how to configure PoC on a local system.
- **pyconfig.structure.ini** Nodes in this file describe PoC's namespace tree and which IP cores are assigned to which namespace.

Additionally, the database refers to **.files* and **.rules* files. The first file type describes, in an imperative language, which files are needed to compile a simulation or to run a synthesis. The latter file type contains patch instructions per IP core. See [Files Formats](#) for more details.

13.2 Database Structure

The database is stored in multiple *INI files*, which are merged in memory to a single configuration database. Each INI file defines an associative array of *sections* and option lines. The content itself is an associative array of *options* and values. Section names are enclosed in square brackets `[...]` and allow simple case-sensitive strings as names. A section name is followed by its section content, which consists of option lines.

One option is stored per option line and consists of an option name and a value separated by an equal sign `=`. The option name is also a case-sensitive simple string. The value is string, starts after the first non-whitespace character and ends before the newline character at the end of the line. The content can be of any string, except for the newline characters. Support for escape sequences depends on the option usage.

Values containing `${...}` and `%{...}` are raw values, which need to be interpolated by the ExtendedConfigParser. See [Value Interpolation](#) and [Node Interpolation](#) for more details.

Sections can have a default section called `DEFAULT`. Options not found in a normal section are looked up in the default section. If found, the value of the matching option name is the lookup result.

Example

```
[section1]
option1 = value1
opt2 =    val ue $2

[section2]
option1 = ${section1:option1}
opt2 =    ${option1}
```

Option lines can be of three kinds: An option, a reference, or a user defined variable. While the syntax is always the same, the meaning is inferred from the context.

Option Line Kind	Distinguishing Characteristic
Reference	The option name is called a (node) reference, if the value of an option is a predefined keyword for the current node class. Because the option's value is a keyword, it cannot be an interpolated value.
Option	The option uses a defined option name valid for the current node class. The value can be a fixed or interpolated string.
User Defined Variable	Otherwise an option line is a user defined variable. It can have fixed or interpolated string values.

```

[PoC]
Name =
Prefix =
arith =      Namespace
bus =      Namespace

[PoC.arith]
addw =      Entity
prng =      Entity

[PoC.bus]
stream =    Namespace
wb =        Namespace
Arbiter =   Entity

[PoC.bus.stream]
Buffer =    Entity
DeMux =     Entity
Mirror =    Entity
Mux =       Entity

[PoC.bus.wb]
fifo_adapter = Entity
ocram_adapter = Entity
uart_wrapper = Entity

```

13.2.1 Nodes

The database is build of nested associative arrays and generated in-memory from 5 *.ini files. This implies that all section names are required to be unique. (Section merging is not allowed.) A fully qualified section name has a prefix and a section name delimited by a dot character. The section name itself can consist of parts also delimited by dot characters. All nodes with the same prefix shape a node class.

The following table lists all used prefixes:

Prefix	Description
INSTALL	A installed tool (chain) or program.
SOLUTION	Registered external solutions / projects.
CONFIG	Configurable PoC settings.
BOARD	A node to describe a known board.
CONST	A node to describe constraint file set for a known board.
PoC	Nodes to describe PoC's namespace structure.
IP	A node describing an IP core.
TB	A node describing testbenches.
COCOTB	A node describing Cocotb testbenches.
CG	A node storing Core Generator settings.
LSE	A node storing settings for LSE based netlist generation.
QMAP	A node storing settings for Quartus based netlist generation.
XST	A node storing settings for XST based netlist generation.
VIVADO	A node storing settings for Vivado based netlist generation.
XCI	A node storing settings for IP Catalog based netlist generation.

The database has 3 special sections without prefixes:

Section Name	Description
PoC	Root node for PoC's namespace hierarchy.
BOARDS	Lists all known boards.
SPECIAL	Section with dummy values. This is needed by synthesis and overwritten at runtime.

Example section names

```
[PoC]
[PoC.arith]
[PoC.bus]
[PoC.bus.stream]
[PoC.bus.wb]
```

The fully qualified section name `PoC.bus.stream` has the prefix `PoC` and the section name `bus.stream`. The section name has two parts: `bus` and `stream`. The dot delimited section name can be considered a path in a hierarchical database. The parent node is `PoC.bus` and its grandparent is `PoC`. (Note this is a special section. See the special sections table from above.)

13.2.2 References

Whatever this is handy to create new field

13.2.3 Options

13.2.4 Values

13.2.5 Value Interpolation

13.2.6 Node Interpolation

13.2.7 Root Nodes

13.3 Supported Options

Note: See `py\config.defaults.ini` for predefined default values (options) and predefined variables, which can be used as a shortcut.

13.4 Files in detail

13.4.1 config.structure.ini

13.4.2 config.entity.ini

13.4.3 config.boards.ini

13.4.4 config.private.ini

13.5 User Defined Variables

14.1 PoC.py

Classes

- PoCEntityAttribute: Undocumented.
- BoardDeviceAttributeGroup: Undocumented.
- VHDLVersionAttribute: Undocumented.
- SimulationStepsAttributeGroup: Undocumented.
- CompileStepsAttributeGroup: Undocumented.
- PileOfCores: A mixin class to provide local logging methods.

Functions

- `main()`: This is the entry point for PoC.py written as a function.

class PoC.PoCEntityAttribute

Inheritance

Members

classmethod GetAttributes (*method*)

classmethod GetMethods (*cl*)

static _AppendAttribute (*attribute*)

_debug = False

class PoC.BoardDeviceAttributeGroup

Inheritance

Members

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
_debug = False
class PoC.VHDLVersionAttribute
```

Inheritance

Members

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
_debug = False
class PoC.SimulationStepsAttributeGroup
```

Inheritance

Members

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
_debug = False
class PoC.CompileStepsAttributeGroup
```

Inheritance

Members

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
_debug = False
class PoC.PileOfCores (debug, verbose, quiet, dryRun, sphinx=False)
```

Inheritance

Members

```
HeadLine = 'The PoC-Library - Service Tool'
Platform
```

```
DryRun
Directories
ConfigFiles
PoCConfig
Root
Repository
SaveAndReloadPoCConfiguration ()
Run ()
PrintHeadline ()
HandleDefault (__)
HandleHelp (args)
HandleInfo (args)
HandleConfiguration (args)
    Handle 'configure' command.
HandleSelection (args)
    Handle 'select' command.
HandleAddSolution (__)
HandleListSolution (__)
HandleRemoveSolution (args)
HandleListProject (args)
HandleQueryConfiguration (args)
__ExtractBoard (BoardName, DeviceName, force=False)
__ExtractFQNs (fqns, defaultLibrary='PoC', defaultType=<EntityTypes.Testbench: 2>)
__ExtractVHDLVersion (vhdlVersion, defaultVersion=None)
__CheckActiveHDL ()
__CheckRivieraPRO ()
__CheckQuartus ()
__CheckDiamond ()
__CheckModelSim ()
__CheckISE ()
__CheckVivado ()
__CheckGHDL ()
static __ExtractSimulationSteps (analyze, elaborate, optimize, recompile, simulate,
                                showWaveform, showCoverage, resimulate, showRe-
                                port, cleanUp)
static __ExtractCompileSteps (synthesize, showReport, cleanUp)
HandleListTestbenches (args)
HandleActiveHDLSimulation (args)
HandleGHDLSimulation (args)
HandleISESimulation (args)
```

```
HandleModelSimSimulation (args)
HandleAnyMentorSimulation (args)
HandleRivieraPROSimulation (args)
HandleQuestaSimSimulation (args)
HandleVivadoSimulation (args)
HandleCocotbSimulation (args)
HandleListNetlist (args)
HandleISECompilation (args)
HandleCoreGeneratorCompilation (args)
HandleXstCompilation (args)
HandleIpCatalogCompilation (args)
HandleVivadoCompilation (args)
HandleQuartusCompilation (args)
HandleLSECompilation (args)
static GetAttributes ()
GetMethods ()
static HasAttribute ()
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
MainParser
SubParsers
__ArgParseMixin__mainParser = None
__ArgParseMixin__subParser = None
__ArgParseMixin__subParsers = {}
__FileOfCores__BackupPoCConfiguration ()
__FileOfCores__CONFIGFILE_BOARDS = 'config.boards.ini'
__FileOfCores__CONFIGFILE_DEFAULTS = 'config.defaults.ini'
__FileOfCores__CONFIGFILE_DIRECTORY = 'py'
__FileOfCores__CONFIGFILE_IPCORES = 'config.entity.ini'
```

```
_FileOfCores__CONFIGFILE_PRIVATE = 'config.private.ini'
_FileOfCores__CONFIGFILE_STRUCTURE = 'config.structure.ini'
_FileOfCores__CheckEnvironment ()
_FileOfCores__CheckSection (sectionName, toolName)
_FileOfCores__PLATFORM = 'Linux'
_FileOfCores__PrepareForConfiguration ()
_FileOfCores__PrepareForSimulation ()
_FileOfCores__PrepareForSynthesis ()
_FileOfCores__ReadPoCConfiguration ()
_FileOfCores__WritePoCConfiguration ()
_TryLog (*args, condition=True, **kwargs)
```

Functions

`PoC.main ()`

This is the entry point for PoC.py written as a function.

1. It extracts common flags from the script's arguments list, before `ArgumentParser` is fully loaded.
2. It initializes colorama for colored outputs
3. It creates an instance of PoC and hands over to class based execution. All is wrapped in a big `try . . except` block to catch every unhandled exception.
4. Shutdown the script and return its exit code.

14.2 Base

Submodules

14.2.1 Base.Exceptions

Exceptions

- `ExceptionBase`: Base exception derived from `Exception` for all
- `EnvironmentException`: `EnvironmentException` is raised when an expected environment variable is
- `PlatformNotSupportedException`: `PlatformNotSupportedException` is raise if the platform is not supported
- `NotConfiguredException`: `NotConfiguredException` is raise if PoC or the requested tool chain
- `SkipableException`: Base class for all skipable exceptions.
- `CommonException`: Base exception derived from `Exception` for all
- `SkipableCommonException`: `SkipableCommonException` is a `CommonException`, which can be

exception `Base.Exceptions.ExceptionBase (message="")`

Base exception derived from `Exception` for all custom exceptions in PoC.

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

exception `Base.Exceptions.EnvironmentException(message="")`
`EnvironmentException` is raised when an expected environment variable is missing for PoC.

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

exception `Base.Exceptions.PlatformNotSupportedException(message="")`
`PlatformNotSupportedException` is raised if the platform is not supported by PoC, or the selected tool flow is not supported on the host system by PoC.

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

exception `Base.Exceptions.NotConfiguredException(message="")`
`NotConfiguredException` is raised if PoC or the requested tool chain setting is not configured in PoC.

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`

Returns the exception's message text.

args

exception `Base.Exceptions.SkipableException (message="")`

Base class for all skipable exceptions.

Inheritance

Members

`__init__(message="")`

Exception initializer

Parameters `message (str)` – The exception message.

`__str__()`

Returns the exception's message text.

args

exception `Base.Exceptions.CommonException (message="")`

Inheritance

Members

`__init__(message="")`

Exception initializer

Parameters `message (str)` – The exception message.

`__str__()`

Returns the exception's message text.

args

exception `Base.Exceptions.SkipableCommonException (message="")`

`SkipableCommonException` is a [*CommonException*](#), which can be skipped.

Inheritance

Members

`__init__(message="")`

Exception initializer

Parameters `message (str)` – The exception message.

`__str__()`

Returns the exception's message text.

args

14.2.2 Base.Executable

Exceptions

- [*ExecutableException*](#): This exception is raised by all executable abstraction classes.

Classes

- *CommandLineArgument*: Base class (and meta class) for all Arguments classes.
- *ExecutableArgument*: Represents the executable.
- *NamedCommandLineArgument*: Base class for all command line arguments with a name.
- *CommandArgument*: Represents a command name.
- *ShortCommandArgument*: Represents a command name with a single dash.
- *LongCommandArgument*: Represents a command name with a double dash.
- *WindowsCommandArgument*: Represents a command name with a single slash.
- *StringArgument*: Represents a simple string argument.
- *StringListArgument*: Represents a list of string arguments.
- *PathArgument*: Represents a path argument.
- *FlagArgument*: Base class for all FlagArgument classes, which represents a simple flag argument.
- *ShortFlagArgument*: Represents a flag argument with a single dash.
- *LongFlagArgument*: Represents a flag argument with a double dash.
- *WindowsFlagArgument*: Represents a flag argument with a single slash.
- *ValuedFlagArgument*: Class and base class for all ValuedFlagArgument classes, which represents a flag argument with data.
- *ShortValuedFlagArgument*: Represents a *ValuedFlagArgument* with a single dash.
- *LongValuedFlagArgument*: Represents a *ValuedFlagArgument* with a double dash.
- *WindowsValuedFlagArgument*: Represents a *ValuedFlagArgument* with a single slash.
- *ValuedFlagListArgument*: Class and base class for all ValuedFlagListArgument classes, which represents a list of *ValuedFlagArgument* instances.
- *ShortValuedFlagListArgument*: Represents a *ValuedFlagListArgument* with a single dash.
- *LongValuedFlagListArgument*: Represents a *ValuedFlagListArgument* with a double dash.
- *WindowsValuedFlagListArgument*: Represents a *ValuedFlagListArgument* with a single slash.
- *TupleArgument*: Class and base class for all TupleArgument classes, which represents a switch with separate data.
- *ShortTupleArgument*: Represents a *TupleArgument* with a single dash in front of the switch name.
- *LongTupleArgument*: Represents a *TupleArgument* with a double dash in front of the switch name.
- *WindowsTupleArgument*: Represents a *TupleArgument* with a single slash in front of the switch name.
- *CommandLineArgumentList*: Represent a list of all available commands, flags and switch of an executable.
- *Environment*: Undocumented.
- *Executable*: Represent an executable.

exception `Base.Executable.ExecutableException (message=)`
This exception is raised by all executable abstraction classes.

Inheritance

Members

`__str__()`
Returns the exception's message text.

`args`

class `Base.Executable.CommandLineArgument`
Base class (and meta class) for all Arguments classes.

Inheritance

Members

`_value = None`
`mro()` → list
return a type's method resolution order

class `Base.Executable.ExecutableArgument`
Represents the executable.

Inheritance

Members

`Value`
`AsArgument()`
`_value = None`
`mro()` → list
return a type's method resolution order

class `Base.Executable.NamedCommandLineArgument`
Base class for all command line arguments with a name.

Inheritance

Members

`_name = None`
`Name`
`_value = None`
`mro()` → list
return a type's method resolution order

class `Base.Executable.CommandArgument`
Represents a command name.

It is usually used to select a sub parser in a CLI argument parser or to hand over all following parameters to a separate tool. An example for a command is 'checkout' in `git.exe checkout`, which calls `git-checkout.exe`.

Inheritance

Members

`_pattern = '{0}'`

Value

`AsArgument()`

Name

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.ShortCommandArgument`
Represents a command name with a single dash.

Inheritance

Members

`_pattern = '-{0}'`

`AsArgument()`

Name

Value

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.LongCommandArgument`
Represents a command name with a double dash.

Inheritance

Members

`_pattern = '--{0}'`

`AsArgument()`

Name

Value

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.WindowsCommandArgument`
Represents a command name with a single slash.

Inheritance

Members

`_pattern = '/{0}'`

`AsArgument()`

`Name`

`Value`

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.StringArgument`
Represents a simple string argument.

Inheritance

Members

`_pattern = '{0}'`

`Value`

`AsArgument()`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.StringListArgument`
Represents a list of string arguments.

Inheritance

Members

`_pattern = '{0}'`

`Value`

`AsArgument()`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.PathArgument`
Represents a path argument.

The output format can be forced to the POSIX format with `_PosixFormat`.

Inheritance

Members

`_PosixFormat = False`

Value

`AsArgument()`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.FlagArgument`

Base class for all FlagArgument classes, which represents a simple flag argument.

A simple flag is a single boolean value (absent/present or off/on) with no data.

Inheritance

Members

`_pattern = '{0}'`

Value

`AsArgument()`

Name

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class `Base.Executable.ShortFlagArgument`

Represents a flag argument with a single dash.

Example: `-optimize`

Inheritance

Members

`_pattern = '-{0}'`

`AsArgument()`

Name

Value

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class Base.Executable.LongFlagArgument

Represents a flag argument with a double dash.

Example: --optimize

Inheritance

Members

`_pattern = '--{0}'`

`AsArgument()`

`Name`

`Value`

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class Base.Executable.WindowsFlagArgument

Represents a flag argument with a single slash.

Example: /optimize

Inheritance

Members

`_pattern = '/{0}'`

`AsArgument()`

`Name`

`Value`

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class Base.Executable.ValuedFlagArgument

Class and base class for all ValuedFlagArgument classes, which represents a flag argument with data.

A valued flag is a flag name followed by a value. The default delimiter sign is equal (=). Name and value are passed as one arguments to the executable even if the delimiter sign is a whitespace character.

Example: width=100

Inheritance

Members

`_pattern = '{0}={1}'`

`Value`

AsArgument ()

Name

_name = None

_value = None

mro () → list

return a type's method resolution order

class Base.Executable.ShortValuedFlagArgument

Represents a *ValuedFlagArgument* with a single dash.

Example: -optimizer=on

Inheritance

Members

_pattern = '-{0}={1}'

AsArgument ()

Name

Value

_name = None

_value = None

mro () → list

return a type's method resolution order

class Base.Executable.LongValuedFlagArgument

Represents a *ValuedFlagArgument* with a double dash.

Example: --optimizer=on

Inheritance

Members

_pattern = '--{0}={1}'

AsArgument ()

Name

Value

_name = None

_value = None

mro () → list

return a type's method resolution order

class Base.Executable.WindowsValuedFlagArgument

Represents a *ValuedFlagArgument* with a single slash.

Example: /optimizer:on

Inheritance

Members

`_pattern = '/{0}:{1}'`

`AsArgument()`

`Name`

`Value`

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class Base.Executable.ValuedFlagListArgument

Class and base class for all *ValuedFlagListArgument* classes, which represents a list of *ValuedFlagArgument* instances.

Each list item gets translated into a *ValuedFlagArgument*, with the same flag name, but differing values. Each *ValuedFlagArgument* is passed as a single argument to the executable, even if the delimiter sign is a whitespace character.

Example: `file=file1.txt file=file2.txt`

Inheritance

Members

`_pattern = '{0}={1}'`

`Value`

`AsArgument()`

`Name`

`_name = None`

`_value = None`

`mro()` → list

return a type's method resolution order

class Base.Executable.ShortValuedFlagListArgument

Represents a *ValuedFlagListArgument* with a single dash.

Example: `-file=file1.txt -file=file2.txt`

Inheritance

Members

`_pattern = '-{0}={1}'`

`AsArgument()`

`Name`

`Value`

```
_name = None  
_value = None  
mro() → list  
    return a type's method resolution order
```

class `Base.Executable.LongValuedFlagListArgument`
Represents a *ValuedFlagListArgument* with a double dash.
Example: `--file=file1.txt --file=file2.txt`

Inheritance

Members

```
_pattern = '--{0}={1}'  
AsArgument()  
Name  
Value  
_name = None  
_value = None  
mro() → list  
    return a type's method resolution order
```

class `Base.Executable.WindowsValuedFlagListArgument`
Represents a *ValuedFlagListArgument* with a single slash.
Example: `/file:file1.txt /file:file2.txt`

Inheritance

Members

```
_pattern = '/{0}:{1}'  
AsArgument()  
Name  
Value  
_name = None  
_value = None  
mro() → list  
    return a type's method resolution order
```

class `Base.Executable.TupleArgument`
Class and base class for all *TupleArgument* classes, which represents a switch with separate data.

A tuple switch is a command line argument followed by a separate value. Name and value are passed as two arguments to the executable.

Example: `width 100`

Inheritance

Members

```
_switchPattern = '{0}'
```

```
_valuePattern = '{0}'
```

Value

AsArgument ()

Name

```
_name = None
```

```
_value = None
```

mro () → list

return a type's method resolution order

```
class Base.Executable.ShortTupleArgument
```

Represents a *TupleArgument* with a single dash in front of the switch name.

Example: -file file1.txt

Inheritance

Members

```
_switchPattern = '-{0}'
```

AsArgument ()

Name

Value

```
_name = None
```

```
_value = None
```

```
_valuePattern = '{0}'
```

mro () → list

return a type's method resolution order

```
class Base.Executable.LongTupleArgument
```

Represents a *TupleArgument* with a double dash in front of the switch name.

Example: --file file1.txt

Inheritance

Members

```
_switchPattern = '--{0}'
```

AsArgument ()

Name

Value

```
_name = None
```

```
_value = None
_valuePattern = '{0}'
mro() → list
    return a type's method resolution order
```

class Base.Executable.WindowsTupleArgument

Represents a *TupleArgument* with a single slash in front of the switch name.

Example: /file file1.txt

Inheritance

Members

```
_switchPattern = '/{0}'
AsArgument()
Name
Value
_name = None
_value = None
_valuePattern = '{0}'
mro() → list
    return a type's method resolution order
```

class Base.Executable.CommandLineArgumentList (*args)

Represent a list of all available commands, flags and switch of an executable.

Inheritance

Members

```
ToArgumentList()
append(object) → None – append object to end
clear() → None – remove all items from L
copy() → list – a shallow copy of L
count(value) → integer – return number of occurrences of value
extend(iterable) → None – extend list by appending elements from the iterable
index(value[, start[, stop]]) → integer – return first index of value.
    Raises ValueError if the value is not present.
insert()
    L.insert(index, object) – insert object before index
pop([index]) → item – remove and return item at index (default last).
    Raises IndexError if list is empty or index is out of range.
remove(value) → None – remove first occurrence of value.
    Raises ValueError if the value is not present.
reverse()
    L.reverse() – reverse IN PLACE
```

`sort (key=None, reverse=False) → None` – stable sort *IN PLACE*

class `Base.Executable.Environment`

Inheritance

Members

class `Base.Executable.Executable` (*platform: str, dryrun: bool, executablePath: pathlib.Path, environment: Base.Executable.Environment = None, logger: Base.Logging.Logger = None*)

Represent an executable.

Inheritance

Members

`_POC_BOUNDARY = '==== POC BOUNDARY ====='`

`Log (entry, condition=True)`

Write an entry to the local logger.

`LogDebug (*args, condition=True, **kwargs)`

`LogDryRun (*args, condition=True, **kwargs)`

`LogError (*args, condition=True, **kwargs)`

`LogFatal (*args, condition=True, **kwargs)`

`LogInfo (*args, condition=True, **kwargs)`

`LogNormal (*args, condition=True, **kwargs)`

`LogQuiet (*args, condition=True, **kwargs)`

`LogVerbose (*args, condition=True, **kwargs)`

`LogWarning (*args, condition=True, **kwargs)`

`Logger`

Return the local logger instance.

`_TryLog (*args, condition=True, **kwargs)`

`Path`

`StartProcess (parameterList)`

`Send (line, end='\n')`

`SendBoundary ()`

`Terminate ()`

`GetReader ()`

`ReadUntilBoundary (indent=0)`

14.2.3 Base.Logging

Classes

- `Severity`: Logging message severity levels.
- `LogEntry`: Represents a single line log message with a severity and indentation level.

- *Logger*: Undocumented.
- *ILogable*: A mixin class to provide local logging methods.

class Base.Logging.**Severity** (*_)
Logging message severity levels.

Inheritance

Members

Fatal = 30
Error = 25
Quiet = 20
Warning = 15
Info = 10
DryRun = 5
Normal = 4
Verbose = 2
Debug = 1
All = 0

class Base.Logging.**LogEntry** (*message*, *severity*=<Severity.Normal: 4>, *indent*=0, *append-*
Linebreak=True)
Represents a single line log message with a severity and indentation level.

Inheritance

Members

_Log_MESSAGE_FORMAT__ = {<Severity.Debug: 1>: 'DEBUG: {message}', <Severity.Verbose

Severity

Return the log message's severity level.

Indent

Return the log message's indentation level.

Message

Return the indented log message.

IndentBy (*indent*)

Increase a log message's indentation level.

class Base.Logging.**Logger** (*logLevel*, *printToStdOut*=True)

Inheritance

Members

LogLevel

Return the currently logged minimal severity level.

BaseIndent

```

__Log_MESSAGE_FORMAT__ = {<Severity.Debug: 1>: ' {DARK_GRAY} {message} {NOCOLOR} ', <S
Write (entry)
TryWrite (entry)
WriteFatal (message, indent=0, appendLinebreak=True)
WriteError (message, indent=0, appendLinebreak=True)
WriteWarning (message, indent=0, appendLinebreak=True)
WriteInfo (message, indent=0, appendLinebreak=True)
WriteQuiet (message, indent=0, appendLinebreak=True)
WriteNormal (message, indent=0, appendLinebreak=True)
WriteVerbose (message, indent=1, appendLinebreak=True)
WriteDebug (message, indent=2, appendLinebreak=True)
WriteDryRun (message, indent=2, appendLinebreak=True)
class Base.Logging.ILogable (logger=None)
    A mixin class to provide local logging methods.

```

Inheritance

Members

Logger

Return the local logger instance.

Log (entry, condition=True)

Write an entry to the local logger.

_TryLog (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

14.2.4 Base.Project

Classes

- *FileTypes*: Undocumented.
- *Environment*: An enumeration.
- *ToolChain*: An enumeration.
- *Tool*: An enumeration.
- *VHDLVersion*: An enumeration.

- *Project*: Undocumented.
- *FileSet*: Undocumented.
- *VHDLLibrary*: Undocumented.
- *File*: Undocumented.
- *ProjectFile*: Undocumented.
- *ConstraintFile*: Undocumented.
- *SettingsFile*: Undocumented.
- *SourceFile*: Undocumented.
- *VHDLSourceFile*: Undocumented.
- *VerilogSourceFile*: Undocumented.
- *PythonSourceFile*: Undocumented.
- *CocotbSourceFile*: Undocumented.

```
class Base.Project.FileTypes
```

Inheritance

Members

```
Extension()
```

```
_FlagsArithmeticMixin__bits
```

```
_FlagsArithmeticMixin__create_flags_instance(bits)
```

```
_Flags__internal_str()
```

```
classmethod bits_from_simple_str(s)
```

```
classmethod bits_from_str(s)
```

Converts the output of `__str__` into an integer.

data

```
classmethod from_simple_str(s)
```

Accepts only the output of `to_simple_str()`. The output of `__str__()` is invalid as input.

```
classmethod from_str(s)
```

Accepts both the output of `to_simple_str()` and `__str__()`.

```
is_disjoint(*flags_instances)
```

```
is_member
```

flags.is_member is a shorthand for *flags.properties is not None*. If this property is False then this Flags instance has either zero bits or holds a combination of flag member bits. If this property is True then the bits of this Flags instance match exactly the bits associated with one of the members. This however doesn't necessarily mean that this flag instance isn't a combination of several flags because the bits of a member can be the subset of another member. For example if `member0_bits=0x1` and `member1_bits=0x3` then the bits of member0 are a subset of the bits of member1. If a flag instance holds the bits of member1 then `Flags.is_member` returns True and `Flags.properties` returns the properties of member1 but `__len__()` returns 2 and `__iter__()` yields both member0 and member1.

name

properties

Returns Returns None if this flag isn't an exact member of a flags class but a combination of flags,

returns an object holding the properties (e.g.: name, data, index, ...) of the flag otherwise. We don't store flag properties directly in Flags instances because this way Flags instances that are the (temporary) result of flags arithmetic don't have to maintain these fields and it also has some benefits regarding memory usage.

```
to_simple_str()
```

```
class Base.Project.Environment
```

An enumeration.

Inheritance

Members

```
Any = 0
```

```
Simulation = 1
```

```
Synthesis = 2
```

```
class Base.Project.ToolChain
```

An enumeration.

Inheritance

Members

```
Any = 0
```

```
Aldec_ActiveHDL = 10
```

```
Aldec_RivieraPRO = 15
```

```
Altera_Quartus = 20
```

```
Altera_ModelSim = 21
```

```
Cocotb = 30
```

```
GHDL_GTKWave = 40
```

```
Intel_Quartus = 50
```

```
Intel_ModelSim = 51
```

```
Lattice_Diamond = 60
```

```
Lattice_Synplify = 65
```

```
Mentor_ModelSim = 70
```

```
Mentor_QuestaSim = 75
```

```
Xilinx_ISE = 80
```

```
Xilinx_PlanAhead = 81
```

```
Xilinx_Vivado = 82
```

```
class Base.Project.Tool(*_)
```

An enumeration.

Inheritance

Members

```
Any = 0
Aldec_aSim = ('ASIM', 'Aldec Active-HDL', 'Aldec Active-HDL')
Aldec_rPro = ('RPRO', 'Aldec Riviera-PRO', 'Aldec Riviera-PRO')
Altera_Quartus_Map = ('QMAP', 'Quartus Map', 'Altera Quartus Map (quartus_map)')
Cocotb_QuestaSim = ('COCO', 'Cocotb', 'Coroutine Cosimulation Testbench (Cocotb)')
GHDL = ('GHDL', 'GHDL', 'GHDL')
GTKwave = ('GTKW', 'GTKWave', 'GTKWave')
Lattice_LSE = ('LSE', 'Lattice LSE', 'Lattice Synthesis Engine (LSE)')
Mentor_vSim = ('VSIM', 'Mentor ModelSim', 'Mentor Graphics ModelSim (vSim)')
Xilinx_iSim = ('XSIM', 'Xilinx iSim', 'Xilinx ISE Simulator (iSim)')
Xilinx_XST = ('XST', 'Xilinx XST', 'Xilinx Synthesis Tool (XST)')
Xilinx_CoreGen = ('CG', 'Xilinx CoreGen', 'Xilinx Core Generator Tool (CoreGen)')
Xilinx_xSim = ('XSIM', 'Xilinx xSim', 'Xilinx Vivado Simulator (xSim)')
Xilinx_Synth = ('VIVADO', 'Xilinx Vivado Synthesis', 'Xilinx Vivado Synthesis (synth)')
Xilinx_IPCatalog = ('XCI', 'Xilinx Vivado IP Catalog', 'Xilinx Vivado IP Catalog')
```

```
class Base.Project.VHDLVersion(*_)
    An enumeration.
```

Inheritance

Members

```
Any = 0
VHDL87 = 87
VHDL93 = 93
VHDL2002 = 2002
VHDL2008 = 2008
```

```
class Base.Project.Project(name)
```

Inheritance

Members

```
Name
RootDirectory
Board
Device
Environment
```

```
ToolChain
Tool
VHDLVersion
CreateFileSet (name, setDefault=True)
AddFileSet (fileSet)
FileSets
DefaultFileSet
AddFile (file, fileSet=None)
AddSourceFile (file, fileSet=None)
Files (fileType=<FileTypes(Text\ProjectFile\FileListFile\RulesFile\SourceFile\VHDLSourceFile\VerilogSourceFile\Python
bits=0xFFFF>, fileSet=None)
ExtractVHDLLibrariesFromVHDLSourceFiles ()
VHDLLibraries
ExternalVHDLLibraries
AddExternalVHDLLibraries (library)
GetVariables ()
pprint (indent=0)
class Base.Project.FileSet (name, project=None)
```

Inheritance

Members

```
Name
Project
Files
AddFile (file)
AddSourceFile (file)
class Base.Project.VHDLLibrary (name, project=None)
```

Inheritance

Members

```
Name
Project
Files
AddFile (file)
class Base.Project.File (file, project=None, fileSet=None)
```

Inheritance

Members

`_FileType = <FileTypes() bits=0x0000>`

`Project`

`FileSet`

`FileType`

`FileName`

`Path`

`Open()`

`ReadFile()`

`_ReadContent()`

`class Base.Project.ProjectFile (file, project=None, fileSet=None)`

Inheritance

Members

`_FileType = <FileTypes.ProjectFile bits=0x0002 data=UNDEFINED>`

`FileName`

`FileSet`

`FileType`

`Open()`

`Path`

`Project`

`ReadFile()`

`_ReadContent()`

`class Base.Project.ConstraintFile (file, project=None, fileSet=None)`

Inheritance

Members

`_FileType = <FileTypes.ConstraintFile bits=0x0200 data=UNDEFINED>`

`FileName`

`FileSet`

`FileType`

`Open()`

`Path`

`Project`

`ReadFile()`

```
_ReadContent ()
```

```
class Base.Project.SettingsFile (file, project=None, fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.SettingsFile bits=0x4000 data=UNDEFINED>
```

```
FileName
```

```
FileSet
```

```
FileType
```

```
Open ()
```

```
Path
```

```
Project
```

```
ReadFile ()
```

```
_ReadContent ()
```

```
class Base.Project.SourceFile (file, project=None, fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.SourceFile bits=0x0010 data=UNDEFINED>
```

```
FileName
```

```
FileSet
```

```
FileType
```

```
Open ()
```

```
Path
```

```
Project
```

```
ReadFile ()
```

```
_ReadContent ()
```

```
class Base.Project.VHDLSourceFile (file, vhdlLibraryName, project=None, fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.VHDLSourceFile bits=0x0020 data=UNDEFINED>
```

```
Parse ()
```

```
File
```

```
FileName
```

```
FileSet
```

FileType

LibraryName

Open()

Path

Project

ReadFile()

_ReadContent()

```
class Base.Project.VerilogSourceFile (file, project=None, fileSet=None)
```

Inheritance

Members

File

FileName

FileSet

FileType

Open()

Path

Project

ReadFile()

_ReadContent()

_FileType = <FileTypes.VerilogSourceFile bits=0x0040 data=UNDEFINED>

```
class Base.Project.PythonSourceFile (file, project=None, fileSet=None)
```

Inheritance

Members

FileName

FileSet

FileType

Open()

Path

Project

ReadFile()

_ReadContent()

_FileType = <FileTypes.PythonSourceFile bits=0x0080 data=UNDEFINED>

```
class Base.Project.CocotbSourceFile (file, project=None, fileSet=None)
```

Inheritance

Members

`File`

`FileName`

`FileSet`

`FileType`

`Open()`

`Path`

`Project`

`ReadFile()`

`_ReadContent()`

`_FileType = <FileTypes.CocotbSourceFile bits=0x0100 data=UNDEFINED>`

14.2.5 Base.Shared

Classes

- *Shared*: Base class for Simulator and Compiler.

Functions

- *to_time()*: Convert *n* seconds to a `str` with this pattern: “{min}:{sec:02}”.

class `Base.Shared.Shared` (*host: Base.IHost, dryRun*)

Base class for Simulator and Compiler.

Parameters

- **host** (*object*) – The hosting instance for this instance.
- **dryRun** (*bool*) – Enable dry-run mode
- **noCleanUp** (*bool*) – Don’t clean up after a run.

Inheritance

Members

`ENVIRONMENT = 0`

`TOOL_CHAIN = 0`

`TOOL = 0`

`VHDL_VERSION = 2008`

`Host`

`DryRun`

`VHDLVersion`

`PoCProject`

`Directories`

`_GetTimeDeltaSinceLastEvent()`

```
_PrepareEnvironment ()  
_PrepareEnvironment_PurgeDirectory ()  
_PrepareEnvironment_CreatingDirectory ()  
_PrepareEnvironment_ChangeDirectory ()  
    Change working directory to temporary path 'temp/<tool>'.  
_Prepare ()  
_CreatePoCProject (projectName, board)  
_AddFileListFile (fileListFilePath)  
_GetHDLParameters (configSectionName)  
    Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.  
Log (entry, condition=True)  
    Write an entry to the local logger.  
LogDebug (*args, condition=True, **kwargs)  
LogDryRun (*args, condition=True, **kwargs)  
LogError (*args, condition=True, **kwargs)  
LogFatal (*args, condition=True, **kwargs)  
LogInfo (*args, condition=True, **kwargs)  
LogNormal (*args, condition=True, **kwargs)  
LogQuiet (*args, condition=True, **kwargs)  
LogVerbose (*args, condition=True, **kwargs)  
LogWarning (*args, condition=True, **kwargs)  
Logger  
    Return the local logger instance.  
_TryLog (*args, condition=True, **kwargs)
```

Functions

Base.Shared.to_time (*seconds*)
 Convert *n* seconds to a `str` with this pattern: "{min}:{sec:02}".

 Parameters **seconds** (*int*) – Number of seconds to be converted.

 Return type `str`

 Returns Returns a string formatted as #:##. E.g. "1:05"

Classes

- `IHost`: This is a type hint class (interface description) for a host instance.

```
class Base.IHost (logger=None)  
    This is a type hint class (interface description) for a host instance.  
    It's needed until PoC requires Python 3.6.
```

Inheritance

Members

```
Platform = 'string'  
PoCConfig = <lib.ExtendedConfigParser.ExtendedConfigParser object>
```



```
SaveAndReloadPoCConfiguration ()
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
_TryLog (*args, condition=True, **kwargs)
```

14.3 Compiler

Submodules

14.3.1 Compiler.ISECompiler

Classes

- `Compiler`: Base class for all Compiler classes.

```
class Compiler.ISECompiler.Compiler (host, dryRun, noCleanUp)
```

Inheritance

Members

```
TOOL_CHAIN = 80
```

```
TOOL = 0
```

```
_PrepareCompiler ()
```

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

```
RunAll (fqnList, *args, **kwargs)
```

```
Run (entity, args, kwargs)
```

Run a testbench.

```
Directories
```

```
DryRun
```

```
ENVIRONMENT = 2
```

```
Host
```

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject*, *indent*, *nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist*, **args*, ***kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName*, *board*)

_ExecuteCopyTasks (*tasks*, *text*)

_ExecuteDeleteTasks (*tasks*, *text*)

_ExecuteReplaceTasks (*tasks*, *text*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList*, *copyTasks*, *text*)

_ParseDeleteRules (*rawList*, *deleteTasks*, *text*)

_ParseReplaceRules (*rawList*, *replaceTasks*, *text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (*netlist*)

```

    _RunPostDelete (netlist)
    _RunPostReplace (netlist)
    _RunPreCopy (netlist)
    _RunPreReplace (netlist)
    _TryLog (*args, condition=True, **kwargs)
    _WriteSpecialSectionIntoConfig (device)

```

14.3.2 Compiler.LSECompiler

Classes

- *Compiler*: Base class for all Compiler classes.

```
class Compiler.LSECompiler.Compiler (host, dryRun, noCleanUp)
```

Inheritance

Members

```
TOOL_CHAIN = 60
```

```
TOOL = ('LSE', 'Lattice LSE', 'Lattice Synthesis Engine (LSE)')
```

```
_PrepareCompiler ()
```

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

```
RunAll (fqList, *args, **kwargs)
```

Run a list of netlist compilations. Expand wildcards to all selected netlists.

```
Run (netlist, board)
```

Run a testbench.

```
_WriteLSEProjectFile (netlist, board)
```

```
_RunCompile (netlist, lseArgumentFile)
```

Directories

DryRun

```
ENVIRONMENT = 2
```

Host

```
Log (entry, condition=True)
```

Write an entry to the local logger.

```
LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
```

```
LogError (*args, condition=True, **kwargs)
```

```
LogFatal (*args, condition=True, **kwargs)
```

```
LogInfo (*args, condition=True, **kwargs)
```

```
LogNormal (*args, condition=True, **kwargs)
```

```
LogQuiet (*args, condition=True, **kwargs)
```

```
LogVerbose (*args, condition=True, **kwargs)
```

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject, indent, nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist, *args, **kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName, board*)

_ExecuteCopyTasks (*tasks, text*)

_ExecuteDeleteTasks (*tasks, text*)

_ExecuteReplaceTasks (*tasks, text*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList, copyTasks, text*)

_ParseDeleteRules (*rawList, deleteTasks, text*)

_ParseReplaceRules (*rawList, replaceTasks, text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (*netlist*)

_RunPostDelete (*netlist*)

_RunPostReplace (*netlist*)

_RunPreCopy (*netlist*)

_RunPreReplace (*netlist*)

_TryLog (**args, condition=True, **kwargs*)

_WriteSpecialSectionIntoConfig (*device*)

14.3.3 Compiler.QuartusCompiler

Classes

- *Compiler*: Base class for all Compiler classes.

class `Compiler.QuartusCompiler.Compiler` (*host, dryRun, noCleanUp*)

Inheritance

Members

TOOL_CHAIN = 20

TOOL = ('QMAP', 'Quartus Map', 'Altera Quartus Map (quartus_map)')

_PrepareCompiler()

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from *Base.Shared.Shared*.

RunAll (*fqnList, *args, **kwargs*)

Run a list of netlist compilations. Expand wildcards to all selected netlists.

Run (*netlist, board*)

Run a testbench.

_WriteSpecialSectionIntoConfig (*device*)

_WriteQuartusProjectFile (*netlist, device*)

_RunCompile (*netlist*)

Directories

DryRun

ENVIRONMENT = 2

Host

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject, indent, nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist*, **args*, ***kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName*, *board*)

_ExecuteCopyTasks (*tasks*, *text*)

_ExecuteDeleteTasks (*tasks*, *text*)

_ExecuteReplaceTasks (*tasks*, *text*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList*, *copyTasks*, *text*)

_ParseDeleteRules (*rawList*, *deleteTasks*, *text*)

_ParseReplaceRules (*rawList*, *replaceTasks*, *text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (*netlist*)

_RunPostDelete (*netlist*)

_RunPostReplace (*netlist*)

_RunPreCopy (*netlist*)

_RunPreReplace (*netlist*)

_TryLog (**args*, *condition=True*, ***kwargs*)

14.3.4 Compiler.VivadoCompiler

Classes

- *Compiler*: Base class for all Compiler classes.

class `Compiler.VivadoCompiler.Compiler` (*host*, *dryRun*, *noCleanUp*)

Inheritance

Members

TOOL_CHAIN = 82

TOOL = ('VIVADO', 'Xilinx Vivado Synthesis', 'Xilinx Vivado Synthesis (synth)')

_PrepareCompiler()
Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

RunAll (*fqnList*, *args, **kwargs)
Run a list of netlist compilations. Expand wildcards to all selected netlists.

Run (*netlist*, *board*)
Run a testbench.

_WriteSpecialSectionIntoConfig (*device*)

_RunCompile (*netlist*)

_WriteTclFile (*netlist*, *device*)

Directories

DryRun

ENVIRONMENT = 2

Host

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (*args, *condition=True*, **kwargs)

LogDryRun (*args, *condition=True*, **kwargs)

LogError (*args, *condition=True*, **kwargs)

LogFatal (*args, *condition=True*, **kwargs)

LogInfo (*args, *condition=True*, **kwargs)

LogNormal (*args, *condition=True*, **kwargs)

LogQuiet (*args, *condition=True*, **kwargs)

LogVerbose (*args, *condition=True*, **kwargs)

LogWarning (*args, *condition=True*, **kwargs)

Logger
Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject*, *indent*, *nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist*, *args, **kwargs)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName*, *board*)

_ExecuteCopyTasks (*tasks*, *text*)

_ExecuteDeleteTasks (*tasks*, *text*)

_ExecuteReplaceTasks (*tasks*, *text*)

```
_GetHDLParameters (configSectionName)
    Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (rawList, copyTasks, text)

_ParseDeleteRules (rawList, deleteTasks, text)

_ParseReplaceRules (rawList, replaceTasks, text)

_Prepare ()

_PrepareCompilerEnvironment (device)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (netlist)

_RunPostDelete (netlist)

_RunPostReplace (netlist)

_RunPreCopy (netlist)

_RunPreReplace (netlist)

_TryLog (*args, condition=True, **kwargs)
```

14.3.5 Compiler.XCICompiler

Classes

- *Compiler*: Base class for all Compiler classes.

```
class Compiler.XCICompiler.Compiler (host, dryRun, noCleanUp)
```

Inheritance

Members

```
TOOL_CHAIN = 82
```

```
TOOL = ('XCI', 'Xilinx Vivado IP Catalog', 'Xilinx Vivado IP Catalog')
```

```
_PrepareCompiler ()
```

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

```
RunAll (fqnList, *args, **kwargs)
```

Run a list of netlist compilations. Expand wildcards to all selected netlists.

```
Run (netlist, board)
```

Run a testbench.

```
_WriteSpecialSectionIntoConfig (device)
```

```
_RunCompile (netlist, device)
```

```
_WriteTclFile (netlist, device)
```

Directories

DryRun

ENVIRONMENT = 2

Host

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject*, *indent*, *nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist*, **args*, ***kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName*, *board*)

_ExecuteCopyTasks (*tasks*, *text*)

_ExecuteDeleteTasks (*tasks*, *text*)

_ExecuteReplaceTasks (*tasks*, *text*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList*, *copyTasks*, *text*)

_ParseDeleteRules (*rawList*, *deleteTasks*, *text*)

_ParseReplaceRules (*rawList*, *replaceTasks*, *text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

```
_PrepareEnvironment_CreatingDirectory ()
_PrepateEnvironment_PurgeDirectory ()
_RunPostCopy (netlist)
_RunPostDelete (netlist)
_RunPostReplace (netlist)
_RunPreCopy (netlist)
_RunPreReplace (netlist)
_TryLog (*args, condition=True, **kwargs)
```

14.3.6 Compiler.XCOCCompiler

Classes

- *Compiler*: Base class for all Compiler classes.

```
class Compiler.XCOCCompiler.Compiler (host, dryRun, noCleanUp)
```

Inheritance

Members

```
TOOL_CHAIN = 80
```

```
TOOL = ('CG', 'Xilinx CoreGen', 'Xilinx Core Generator Tool (CoreGen)')
```

```
_PrepareCompiler ()
```

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

```
RunAll (fqnList, *args, **kwargs)
```

Run a list of netlist compilations. Expand wildcards to all selected netlists.

```
Run (netlist, board)
```

Run a testbench.

```
_WriteSpecialSectionIntoConfig (device)
```

```
_RunCompile (netlist, device)
```

Directories

DryRun

```
ENVIRONMENT = 2
```

Host

```
Log (entry, condition=True)
```

Write an entry to the local logger.

```
LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
```

```
LogError (*args, condition=True, **kwargs)
```

```
LogFatal (*args, condition=True, **kwargs)
```

```
LogInfo (*args, condition=True, **kwargs)
```

```
LogNormal (*args, condition=True, **kwargs)
```

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject, indent, nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist, *args, **kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName, board*)

_ExecuteCopyTasks (*tasks, text*)

_ExecuteDeleteTasks (*tasks, text*)

_ExecuteReplaceTasks (*tasks, text*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList, copyTasks, text*)

_ParseDeleteRules (*rawList, deleteTasks, text*)

_ParseReplaceRules (*rawList, replaceTasks, text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (*netlist*)

_RunPostDelete (*netlist*)

_RunPostReplace (*netlist*)

_RunPreCopy (*netlist*)

_RunPreReplace (*netlist*)

_TryLog (**args, condition=True, **kwargs*)

14.3.7 Compiler.XSTCompiler

Classes

- *Compiler*: Base class for all Compiler classes.

class Compiler.XSTCompiler.**Compiler** (*host, dryRun, noCleanUp*)

Inheritance

Members

TOOL_CHAIN = 80

TOOL = ('XST', 'Xilinx XST', 'Xilinx Synthesis Tool (XST)')

_PrepareCompiler()

Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

RunAll (*fqnList, *args, **kwargs*)

Run a list of netlist compilations. Expand wildcards to all selected netlists.

Run (*netlist, board*)

Run a testbench.

_WriteSpecialSectionIntoConfig (*device*)

_RunCompile (*netlist*)

_WriteXstOptionsFile (*netlist, device*)

Directories

DryRun

ENVIRONMENT = 2

Host

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

NoCleanUp

PoCProject

PrintCompileReportLine (*testObject, indent, nameColumnWidth*)

PrintOverallCompileReport ()

TryRun (*netlist*, *args, **kwargs)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 93

_AddFileListFile (*fileListFilePath*)

_AddRulesFiles (*rulesFilePath*)

_CreatePoCProject (*projectName*, *board*)

_ExecuteCopyTasks (*tasks*, *text*)

_ExecuteDeleteTasks (*tasks*, *text*)

_ExecuteReplaceTasks (*tasks*, *text*)

_GenerateXilinxProjectFileContent (*tool*, *vhdlVersion=93*)

_GetHDLParameters (*configSectionName*)

Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_ParseCopyRules (*rawList*, *copyTasks*, *text*)

_ParseDeleteRules (*rawList*, *deleteTasks*, *text*)

_ParseReplaceRules (*rawList*, *replaceTasks*, *text*)

_Prepare ()

_PrepareCompilerEnvironment (*device*)

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()

Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_RunPostCopy (*netlist*)

_RunPostDelete (*netlist*)

_RunPostReplace (*netlist*)

_RunPreCopy (*netlist*)

_RunPreReplace (*netlist*)

_TryLog (*args, *condition=True*, **kwargs)

_WriteXilinxProjectFile (*projectFilePath*, *tool*, *vhdlVersion=93*)

Exceptions

- *CompilerException*: Base class for all CompilerException classes. It is raised while running
- *SkipableCompilerException*: SkipableCompilerException is a *CompilerException*, which

Classes

- *CopyTask*: This class represents a 'copy task' and inherits the partial class
- *DeleteTask*: This class represents a 'delete task' and inherits the partial class
- *ReplaceTask*: This class represents a 'replace task' and inherits the partial class
- *AppendLineTask*: This class represents a 'append line task' and inherits the partial class

- *CompileState*: Compile state enumeration.
- *CompileResult*: Compilation result enumeration.
- *Compiler*: Base class for all Compiler classes.

exception `Compiler.CompilerException` (*message*=")

Base class for all `CompilerException` classes. It is raised while running compiler (synthesis) tasks in PoC.

Inheritance

Members

`__init__` (*message*=")

Exception initializer

Parameters *message* (*str*) – The exception message.

`__str__` ()

Returns the exception's message text.

args

exception `Compiler.SkipableCompilerException` (*message*=")

`SkipableCompilerException` is a *CompilerException*, which can be skipped.

Inheritance

Members

`__init__` (*message*=")

Exception initializer

Parameters *message* (*str*) – The exception message.

`__str__` ()

Returns the exception's message text.

args

class `Compiler.CopyTask` (*sourcePath*, *destinationPath*)

This class represents a 'copy task' and inherits the partial class *CopyRuleMixin*.

Inheritance

Members

DestinationPath

SourcePath

class `Compiler.DeleteTask` (*filePath*)

This class represents a 'delete task' and inherits the partial class *DeleteRuleMixin*.

Inheritance

Members

FilePath

class `Compiler.ReplaceTask` (*filePath, searchPattern, replacePattern, multiLine, dotAll, caseIn-Sensitive*)

This class represents a ‘replace task’ and inherits the partial class [ReplaceRuleMixin](#).

Inheritance

Members

FilePath

RegExpOption_CaseInsensitive

RegExpOption_DotAll

RegExpOption_MultiLine

ReplacePattern

SearchPattern

class `Compiler.AppendLineTask` (*filePath, appendPattern*)

This class represents a ‘append line task’ and inherits the partial class [AppendLineRuleMixin](#).

Inheritance

Members

AppendPattern

FilePath

class `Compiler.CompileState`

Compile state enumeration.

Inheritance

Members

Prepare = 0

PreCopy = 10

PrePatch = 11

Compile = 50

PostCopy = 90

PostPatch = 91

PostDelete = 92

CleanUp = 99

class `Compiler.CompileResult`

Compilation result enumeration.

Inheritance

Members

NotRun = 0

Error = 1

Failed = 2

Success = 3

class `Compiler.Compiler` (*host: Base.IHost, dryRun, noCleanUp*)
Base class for all Compiler classes.

Inheritance

Members

ENVIRONMENT = 2

VHDL_VERSION = 93

__init__ (*host: Base.IHost, dryRun, noCleanUp*)
Class initializer

Parameters

- **host** (*object*) – The hosting instance for this instance.
- **dryRun** (*bool*) – Enable dry-run mode
- **noCleanUp** (*bool*) – Don't clean up after a run.

NoCleanUp

_PrepareCompiler ()
Prepare for compilation. This method forwards to `Base.Compiler.Compiler._Prepare()`, which is inherited from `Base.Shared.Shared`.

TryRun (*netlist, *args, **kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

Run (*netlist, board*)
Run a testbench.

_PrepareCompilerEnvironment (*device*)

_WriteSpecialSectionIntoConfig (*device*)

_AddRulesFiles (*rulesFilePath*)

_RunPreCopy (*netlist*)

_RunPostCopy (*netlist*)

_ParseCopyRules (*rawList, copyTasks, text*)

_ExecuteCopyTasks (*tasks, text*)

_RunPostDelete (*netlist*)

_ParseDeleteRules (*rawList, deleteTasks, text*)

_ExecuteDeleteTasks (*tasks, text*)

_RunPreReplace (*netlist*)

_RunPostReplace (*netlist*)

_ParseReplaceRules (*rawList, replaceTasks, text*)

_ExecuteReplaceTasks (*tasks, text*)

PrintOverallCompileReport ()

PrintCompileReportLine (*testObject, indent, nameColumnWidth*)

Directories

DryRun

Host

Log (*entry, condition=True*)
Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PoCProject

TOOL = 0

TOOL_CHAIN = 0

VHDLVersion

_AddFileListFile (*fileListFilePath*)

_CreatePoCProject (*projectName, board*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_Prepare ()

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_TryLog (**args, condition=True, **kwargs*)

14.4 DataBase

Submodules

14.4.1 DataBase.Config

Classes

- *BaseEnum*: An enumeration.
- *Vendors*: An enumeration.
- *Families*: Base enum for all Family enums.
- *GenericFamilies*: Enumeration of all generic families.
- *AlteraFamilies*: Enumeration of all Altera families.
- *LatticeFamilies*: Enumeration of all Lattice families.
- *XilinxFamilies*: Enumeration of all Xilinx families.
- *Devices*: Base enum for all Device enums.
- *GenericDevices*: Enumeration of all generic devices.
- *AlteraDevices*: Enumeration of all Altera devices.
- *LatticeDevices*: Enumeration of all Lattice devices.
- *XilinxDevices*: Enumeration of all Xilinx devices.
- *SubTypes*: Base enum for all SubType enums.
- *GenericSubTypes*: Enumeration of all generic device subtype.
- *AlteraSubTypes*: Enumeration of all Altera device subtype.
- *LatticeSubTypes*: Enumeration of all Lattice device subtype.
- *XilinxSubTypes*: Enumeration of all Xilinx device subtype.
- *Packages*: An enumeration.
- *Device*: Undocumented.
- *Board*: Undocumented.

class DataBase.Config.**BaseEnum**
An enumeration.

Inheritance

Members

class DataBase.Config.**Vendors**
An enumeration.

Inheritance

Members

Unknown = 0
Generic = 1
Altera = 2
Lattice = 3
MicroSemi = 4

```
Xilinx = 5
```

```
class DataBase.Config.Families  
    Base enum for all Family enums.
```

Inheritance

Members

```
class DataBase.Config.GenericFamilies  
    Enumeration of all generic families.
```

Inheritance

Members

```
Unknown = None
```

```
Generic = 'g'
```

```
class DataBase.Config.AlteraFamilies  
    Enumeration of all Altera families.
```

Inheritance

Members

```
Max = 'm'
```

```
Cyclone = 'c'
```

```
Arria = 'a'
```

```
Stratix = 's'
```

```
class DataBase.Config.LatticeFamilies  
    Enumeration of all Lattice families.
```

Inheritance

Members

```
ECP = 'lfe'
```

```
class DataBase.Config.XilinxFamilies  
    Enumeration of all Xilinx families.
```

Inheritance

Members

```
Spartan = 's'
```

```
Artix = 'a'
```

```
Kintex = 'k'
```

```
Virtex = 'v'
```

```
Zynq = 'z'
```

```
class DataBase.Config.Devices
    Base enum for all Device enums.
```

Inheritance

Members

```
class DataBase.Config.GenericDevices
    Enumeration of all generic devices.
```

Inheritance

Members

```
Unknown = 0
```

```
Generic = 1
```

```
class DataBase.Config.Alteradevices
    Enumeration of all Altera devices.
```

Inheritance

Members

```
Max2 = 100
```

```
Max4 = 101
```

```
Max5 = 102
```

```
Max10 = 103
```

```
Cyclone3 = 110
```

```
Cyclone4 = 111
```

```
Cyclone5 = 112
```

```
Arria2 = 120
```

```
Arria5 = 121
```

```
Stratix2 = 130
```

```
Stratix4 = 131
```

```
Stratix5 = 132
```

```
Stratix10 = 133
```

```
class DataBase.Config.LatticeDevices
    Enumeration of all Lattice devices.
```

Inheritance

Members

```
iCE40 = 200
MachXO = 210
MachXO2 = 211
MachXO3 = 212
ECP2 = 220
ECP3 = 221
ECP5 = 222
```

```
class DataBase.Config.XilinxDevices
    Enumeration of all Xilinx devices.
```

Inheritance

Members

```
Spartan3 = 310
Spartan6 = 311
Artix7 = 320
Kintex7 = 330
KintexUltraScale = 331
KintexUltraScalePlus = 332
Virtex2 = 340
Virtex4 = 341
Virtex5 = 342
Virtex6 = 343
Virtex7 = 344
VirtexUltraScale = 345
VirtexUltraScalePlus = 346
Zynq7000 = 350
```

```
class DataBase.Config.SubTypes
    Base enum for all SubType enums.
```

Inheritance

Members

```
class DataBase.Config.GenericSubTypes
    Enumeration of all generic device subtype.
```

Inheritance

Members

Unknown = None

Generic = 1

NoSubType = ('', '')

class DataBase.Config.**AlteraSubTypes**
Enumeration of all Altera device subtype.

Inheritance

Members

NoSubType = ('', '')

LS = ('ls', '')

E = ('e', '')

GS = ('gs', '')

GX = ('gx', '')

GT = ('gt', '')

GZ = ('gz', '')

SX = ('sx', '')

ST = ('st', '')

class DataBase.Config.**LatticeSubTypes**
Enumeration of all Lattice device subtype.

Inheritance

Members

NoSubType = ('', '')

U = ('u', '')

UM = ('um', '')

class DataBase.Config.**XilinxSubTypes**
Enumeration of all Xilinx device subtype.

Inheritance

Members

NoSubType = ('', '')

DA = ('d', 'a')

E = ('', 'e')

AN = ('', 'an')

```
X = ('x', '')
T = ('', 't')
XT = ('x', 't')
HT = ('h', 't')
LX = ('lx', '')
SXT = ('sx', 't')
LXT = ('lx', 't')
TXT = ('tx', 't')
FXT = ('fx', 't')
CXT = ('cx', 't')
HXT = ('hx', 't')
```

```
class DataBase.Config.Packages
    An enumeration.
```

Inheritance

Members

```
Unknown = 0
Generic = 1
TQG = 10
CLG = 20
CPG = 21
CSG = 22
CABGA = 25
FBG = 30
FF = 31
FFG = 32
FG = 33
FGG = 34
FLG = 35
FT = 36
FTG = 37
RB = 40
RBG = 41
RF = 42
RS = 43
E = 50
Q = 51
F = 52
```

U = 53

M = 54

class DataBase.Config.**Device** (*deviceString*)

Inheritance

Members

_DecodeGeneric ()

_DecodeAltera (*deviceString*)

_DecodeLatticeICE (*deviceString*)

_DecodeLatticeLCM (*deviceString*)

_DecodeLatticeLFE (*deviceString*)

_DecodeLatticeECP3 (*deviceString*)

_DecodeLatticeECP5 (*deviceString*)

_DecodeXilinx (*deviceString*)

Vendor

Family

Device

Generation

Number

SpeedGrade

PinCount

Package

Name

ShortName

FullName

FullName2

FamilyName

Series

GetVariables ()

class DataBase.Config.**Board** (*host, boardName=None, device=None*)

Inheritance

Members

Name

Device

GetVariables ()

14.4.2 DataBase.Entity

Classes

- *EntityTypes*: An enumeration.
- *BaseFlags*: Undocumented.
- *TestbenchKind*: Undocumented.
- *NetlistKind*: Undocumented.
- *NamespaceRoot*: Undocumented.
- *Visibility*: An enumeration.
- *PathElement*: Undocumented.
- *Namespace*: Undocumented.
- *Library*: Undocumented.
- *Wildcard*: Undocumented.
- *StarWildcard*: Undocumented.
- *AskWildcard*: Undocumented.
- *IPCore*: Undocumented.
- *LazyPathElement*: Undocumented.
- *Testbench*: Undocumented.
- *VHDLTestbench*: Undocumented.
- *CocoTestbench*: Undocumented.
- *Netlist*: Undocumented.
- *XstNetlist*: Undocumented.
- *QuartusNetlist*: Undocumented.
- *LatticeNetlist*: Undocumented.
- *CoreGeneratorNetlist*: Undocumented.
- *VivadoNetlist*: Undocumented.
- *FQN*: Undocumented.

Functions

- *_PoCEntityTypes_parser()*: Undocumented.

class DataBase.Entity.**EntityTypes**
An enumeration.

Inheritance

Members

Unknown = 0
Source = 1
Testbench = 2
NetList = 3

class DataBase.Entity.**BaseFlags**

Inheritance

Members

`__FlagsArithmeticMixin__bits`

`__FlagsArithmeticMixin__create_flags_instance` (*bits*)

`__Flags__internal_str` ()

`classmethod bits_from_simple_str` (*s*)

`classmethod bits_from_str` (*s*)

Converts the output of `__str__` into an integer.

data

`classmethod from_simple_str` (*s*)

Accepts only the output of `to_simple_str()`. The output of `__str__()` is invalid as input.

`classmethod from_str` (*s*)

Accepts both the output of `to_simple_str()` and `__str__()`.

`is_disjoint` (**flags_instances*)

`is_member`

flags.is_member is a shorthand for *flags.properties is not None*. If this property is False then this Flags instance has either zero bits or holds a combination of flag member bits. If this property is True then the bits of this Flags instance match exactly the bits associated with one of the members. This however doesn't necessarily mean that this flag instance isn't a combination of several flags because the bits of a member can be the subset of another member. For example if `member0_bits=0x1` and `member1_bits=0x3` then the bits of member0 are a subset of the bits of member1. If a flag instance holds the bits of member1 then `Flags.is_member` returns True and `Flags.properties` returns the properties of member1 but `__len__()` returns 2 and `__iter__()` yields both member0 and member1.

name

properties

Returns Returns None if this flag isn't an exact member of a flags class but a combination of flags,

returns an object holding the properties (e.g.: `name`, `data`, `index`, ...) of the flag otherwise. We don't store flag properties directly in Flags instances because this way Flags instances that are the (temporary) result of flags arithmetic don't have to maintain these fields and it also has some benefits regarding memory usage.

`to_simple_str` ()

`class DataBase.Entity.TestbenchKind`

Inheritance

Members

`__FlagsArithmeticMixin__bits`

`__FlagsArithmeticMixin__create_flags_instance` (*bits*)

`__Flags__internal_str` ()

`classmethod bits_from_simple_str` (*s*)

`classmethod bits_from_str` (*s*)

Converts the output of `__str__` into an integer.

data

classmethod from_simple_str (*s*)

Accepts only the output of `to_simple_str()`. The output of `__str__()` is invalid as input.

classmethod from_str (*s*)

Accepts both the output of `to_simple_str()` and `__str__()`.

is_disjoint (**flags_instances*)

is_member

flags.is_member is a shorthand for *flags.properties is not None*. If this property is False then this Flags instance has either zero bits or holds a combination of flag member bits. If this property is True then the bits of this Flags instance match exactly the bits associated with one of the members. This however doesn't necessarily mean that this flag instance isn't a combination of several flags because the bits of a member can be the subset of another member. For example if `member0_bits=0x1` and `member1_bits=0x3` then the bits of `member0` are a subset of the bits of `member1`. If a flag instance holds the bits of `member1` then `Flags.is_member` returns True and `Flags.properties` returns the properties of `member1` but `__len__()` returns 2 and `__iter__()` yields both `member0` and `member1`.

name

properties

Returns Returns None if this flag isn't an exact member of a flags class but a combination of flags,

returns an object holding the properties (e.g.: `name`, `data`, `index`, ...) of the flag otherwise. We don't store flag properties directly in Flags instances because this way Flags instances that are the (temporary) result of flags arithmetic don't have to maintain these fields and it also has some benefits regarding memory usage.

to_simple_str ()

class DataBase.Entity.NetlistKind

Inheritance

Members

_FlagsArithmeticMixin__bits

_FlagsArithmeticMixin__create_flags_instance (*bits*)

_Flags__internal_str ()

classmethod bits_from_simple_str (*s*)

classmethod bits_from_str (*s*)

Converts the output of `__str__` into an integer.

data

classmethod from_simple_str (*s*)

Accepts only the output of `to_simple_str()`. The output of `__str__()` is invalid as input.

classmethod from_str (*s*)

Accepts both the output of `to_simple_str()` and `__str__()`.

is_disjoint (**flags_instances*)

is_member

flags.is_member is a shorthand for *flags.properties is not None*. If this property is False then this Flags instance has either zero bits or holds a combination of flag member bits. If this property is True then the bits of this Flags instance match exactly the bits associated with one of the members. This however doesn't necessarily mean that this flag instance isn't a combination of several flags because the

bits of a member can be the subset of another member. For example if `member0_bits=0x1` and `member1_bits=0x3` then the bits of `member0` are a subset of the bits of `member1`. If a flag instance holds the bits of `member1` then `Flags.is_member` returns `True` and `Flags.properties` returns the properties of `member1` but `__len__()` returns 2 and `__iter__()` yields both `member0` and `member1`.

name

properties

Returns Returns `None` if this flag isn't an exact member of a flags class but a combination of flags,

returns an object holding the properties (e.g.: `name`, `data`, `index`, ...) of the flag otherwise. We don't store flag properties directly in `Flags` instances because this way `Flags` instances that are the (temporary) result of flags arithmetic don't have to maintain these fields and it also has some benefits regarding memory usage.

to_simple_str()

class `DataBase.Entity.NamespaceRoot` (*host*)

Inheritance

Members

Libraries

LibraryNames

GetLibraries()

GetLibraryNames()

AddLibrary (*libraryName*, *libraryPrefix*)

__NamespaceRoot__POCRoot_Name = 'PoC'

__NamespaceRoot__POCRoot_SectionName = 'PoC'

class `DataBase.Entity.Visibility`

An enumeration.

Inheritance

Members

Unknown = 0

Private = 1

Public = 2

class `DataBase.Entity.PathElement` (*host*, *name*, *configSectionName*, *parent*)

Inheritance

Members

Name

Parent

ConfigSectionName

ConfigSection
Level
Visibility
IsVisible
Path
_Load()

class DataBase.Entity.**Namespace** (*host, name, configSectionName, parent*)

Inheritance

Members

_Load()
Namespaces
NamespaceNames
Entities
EntityNames
GetNamespaces()
GetNamespaceNames()
GetEntities()
GetEntityNames()
GetAllEntities()
pprint (*indent=0*)
ConfigSection
ConfigSectionName
IsVisible
Level
Name
Parent
Path
Visibility

class DataBase.Entity.**Library** (*host, name, configSectionName, parent*)

Inheritance

Members

Level
ConfigSection
ConfigSectionName
Entities

```
EntityNames
GetAllEntities ()
GetEntities ()
GetEntityNames ()
GetNamespaceNames ()
GetNamespaces ()
IsVisible
Name
NamespaceNames
Namespaces
Parent
Path
Visibility
_Load ()
pprint (indent=0)
```

```
class DataBase.Entity.WildCard (host, name, configSectionName, parent)
```

Inheritance

Members

```
GetEntities ()
GetTestbenches (kind=<TestbenchKind(VHDLTestbench\CocoTestbench) bits=0x0003>)
GetVHDLTestbenches ()
GetCocoTestbenches ()
GetNetlists (kind=<NetlistKind(LatticeNetlist\QuartusNetlist\XstNetlist\CoreGeneratorNetlist\VivadoNetlist)
               bits=0x001F>)
GetLatticeNetlists ()
GetQuartusNetlists ()
GetXSTNetlists ()
GetCoreGenNetlists ()
GetVivadoNetlists ()
Testbenches
VHDLTestbenches
CocoTestbenches
Netlists
LatticeNetlists
QuartusNetlists
XSTNetlists
CoreGenNetlists
```

VivadoNetlists
ConfigSection
ConfigSectionName
IsVisible
Level
Name
Parent
Path
Visibility
_Load()

class DataBase.Entity.**StarWildCard** (*host, name, configSectionName, parent*)

Inheritance

Members

_Load()
GetEntities()
CocoTestbenches
ConfigSection
ConfigSectionName
CoreGenNetlists
GetCocoTestbenches()
GetCoreGenNetlists()
GetLatticeNetlists()
GetNetlists (*kind=<NetlistKind(LatticeNetlist\QuartusNetlist\XstNetlist\CoreGeneratorNetlist\VivadoNetlist)*
bits=0x001F>)
GetQuartusNetlists()
GetTestbenches (*kind=<TestbenchKind(VHDLTestbench\CocoTestbench) bits=0x0003>*)
GetVHDLTestbenches()
GetVivadoNetlists()
GetXSTNetlists()
IsVisible
LatticeNetlists
Level
Name
Netlists
Parent
Path
QuartusNetlists

Testbenches
VHDLTestbenches
Visibility
VivadoNetlists
XSTNetlists

```
class DataBase.Entity.AskWildCard (host, name, configSectionName, parent)
```

Inheritance

Members

_Load()
GetEntities()
CocoTestbenches
ConfigSection
ConfigSectionName
CoreGenNetlists
GetCocoTestbenches()
GetCoreGenNetlists()
GetLatticeNetlists()
GetNetlists (*kind=<NetlistKind(LatticeNetlist\QuartusNetlist\XstNetlist\CoreGeneratorNetlist\VivadoNetlist)*
bits=0x001F>)
GetQuartusNetlists()
GetTestbenches (*kind=<TestbenchKind(VHDLTestbench\CocoTestbench) bits=0x0003>*)
GetVHDLTestbenches()
GetVivadoNetlists()
GetXSTNetlists()
IsVisible
LatticeNetlists
Level
Name
Netlists
Parent
Path
QuartusNetlists
Testbenches
VHDLTestbenches
Visibility
VivadoNetlists
XSTNetlists

```
class DataBase.Entity.IPCore (host, name, configSectionName, parent)
```


Inheritance

Members

Dependencies

VHDLTestbench

CocoTestbench

GetTestbenches (*kind=<TestbenchKind(VHDLTestbench\CocoTestbench) bits=0x0003>*)

LatticeNetlist

QuartusNetlist

XSTNetlist

CGNetlist

VivadoNetlist

GetNetlists (*kind=<NetlistKind(LatticeNetlist\QuartusNetlist\XstNetlist\CoreGeneratorNetlist\VivadoNetlist) bits=0x001F>*)

_Load()

pprint (*indent=0*)

ConfigSection

ConfigSectionName

IsVisible

Level

Name

Parent

Path

Visibility

class DataBase.Entity.LazyPathElement (*host, name, configSectionName, parent*)

Inheritance

Members

Kind

ConfigSection

ConfigSectionName

IsVisible

LazyLoadable_IsLoaded

Level

Name

Parent

Path

Visibility

```
_LazyLoadable_Load()
```

```
_Load()
```

```
class DataBase.Entity.Testbench (host, name, configSectionName, parent)
```

Inheritance

Members

ModuleName

FilesFile

Result

```
_LazyLoadable_Load()
```

```
pprint (indent)
```

ConfigSection

ConfigSectionName

IsVisible

Kind

LazyLoadable_IsLoaded

Level

Name

Parent

Path

Visibility

```
_Load()
```

```
class DataBase.Entity.VHDLTestbench (host, name, configSectionName, parent)
```

Inheritance

Members

```
_LazyLoadable_Load()
```

```
pprint (indent)
```

ConfigSection

ConfigSectionName

FilesFile

IsVisible

Kind

LazyLoadable_IsLoaded

Level

ModuleName

Name

Parent
Path
Result
Visibility
_Load()

```
class DataBase.Entity.CocoTestbench (host, name, configSectionName, parent)
```

Inheritance

Members

TopLevel
_LazyLoadable_Load()
pprint (*indent*)
ConfigSection
ConfigSectionName
FilesFile
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path
Result
Visibility
_Load()

```
class DataBase.Entity.Netlist (host, name, configSectionName, parent)
```

Inheritance

Members

ModuleName
RulesFile
_LazyLoadable_Load()
ConfigSection
ConfigSectionName
IsVisible
Kind

LazyLoadable_IsLoaded
Level
Name
Parent
Path
Visibility
_Load()

```
class DataBase.Entity.XstNetlist (host, name, configSectionName, parent)
```

Inheritance

Members

FilesFile
XcfFile
FilterFile
XstTemplateFile
PrjFile
XstFile
_LazyLoadable_Load()
pprint (*indent*)
ConfigSection
ConfigSectionName
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path
RulesFile
Visibility
_Load()

```
class DataBase.Entity.QuartusNetlist (host, name, configSectionName, parent)
```

Inheritance

Members

FilesFile

```
QsfFile
_LazyLoadable_Load()
pprint (indent)
ConfigSection
ConfigSectionName
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path
RulesFile
Visibility
_Load()
```

```
class DataBase.Entity.LatticeNetlist (host, name, configSectionName, parent)
```

Inheritance

Members

```
FilesFile
PrjFile
_LazyLoadable_Load()
pprint (indent)
ConfigSection
ConfigSectionName
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path
RulesFile
Visibility
_Load()
```

```
class DataBase.Entity.CoreGeneratorNetlist (host, name, configSectionName, parent)
```

Inheritance

Members

FilesFile
XcoFile
_LazyLoadable_Load()
pprint (*indent*)
ConfigSection
ConfigSectionName
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path
RulesFile
Visibility
_Load()

class DataBase.Entity.**VivadoNetlist** (*host, name, configSectionName, parent*)

Inheritance

Members

FilesFile
TclFile
_LazyLoadable_Load()
pprint (*indent*)
ConfigSection
ConfigSectionName
IsVisible
Kind
LazyLoadable_IsLoaded
Level
ModuleName
Name
Parent
Path

RulesFile

Visibility

_Load()

```
class DataBase.Entity.FQN(host, fqn, defaultLibrary='PoC', defaultType=<EntityTypes.Source: 1>)
```

Inheritance

Members

Root()

Entity

Functions

```
DataBase.Entity._PoCEntityTypes_parser(cls, value)
```

14.4.3 DataBase.Solution

Classes

- *Base*: Base class for Repository, Solution and Project.
- *Repository*: Base class for Repository, Solution and Project.
- *Solution*: Base class for Repository, Solution and Project.
- *Project*: Base class for Repository, Solution and Project.
- *ISEProject*: Base class for Repository, Solution and Project.
- *VivadoProject*: Base class for Repository, Solution and Project.
- *QuartusProject*: Base class for Repository, Solution and Project.
- *LatticeProject*: Base class for Repository, Solution and Project.
- *VirtualProject*: Undocumented.
- *FileListFile*: Undocumented.
- *RulesFile*: Undocumented.

```
class DataBase.Solution.Base(host, sectionPrefix, sectionID, parent)  
    Base class for Repository, Solution and Project. It implements ILazyLoadable.
```

Inheritance

Members

ID

Parent

ConfigSectionName

_Load()

Implement this method for early loading.

LazyLoadable_IsLoaded

_LazyLoadable_Load()

```
class DataBase.Solution.Repository (host)
```

Inheritance

Members

Kind

`_Load()`

Implement this method for early loading.

`_LazyLoadable_Load()`

`AddSolution (solutionID, solutionName, solutionRootPath)`

`RemoveSolution (solution)`

`Solutions`

`SolutionNames`

`ConfigSectionName`

`ID`

`LazyLoadable_IsLoaded`

`Parent`

```
class DataBase.Solution.Solution (host, slnID, parent)
```

Inheritance

Members

`Register()`

`Unregister()`

`CreateFiles()`

`_LazyLoadable_Load()`

`Name`

`Path`

`Projects`

`ProjectNames`

`ConfigSectionName`

`ID`

`LazyLoadable_IsLoaded`

`Parent`

`_Load()`

Implement this method for early loading.

```
class DataBase.Solution.Project (host, prjID, parent)
```


Inheritance

Members

Name

ConfigSectionName

ID

LazyLoadable_IsLoaded

Parent

_LazyLoadable_Load()

_Load()

Implement this method for early loading.

class DataBase.Solution.**ISEProject** (*host, prjID, parent*)

Inheritance

Members

ConfigSectionName

ID

LazyLoadable_IsLoaded

Name

Parent

_LazyLoadable_Load()

_Load()

Implement this method for early loading.

class DataBase.Solution.**VivadoProject** (*host, prjID, parent*)

Inheritance

Members

ConfigSectionName

ID

LazyLoadable_IsLoaded

Name

Parent

_LazyLoadable_Load()

_Load()

Implement this method for early loading.

class DataBase.Solution.**QuartusProject** (*host, prjID, parent*)

Inheritance

Members

ConfigSectionName

ID

LazyLoadable_IsLoaded

Name

Parent

_LazyLoadable_Load()

_Load()

Implement this method for early loading.

class DataBase.Solution.LatticeProject (*host, prjID, parent*)

Inheritance

Members

ConfigSectionName

ID

LazyLoadable_IsLoaded

Name

Parent

_LazyLoadable_Load()

_Load()

Implement this method for early loading.

class DataBase.Solution.VirtualProject (*name*)

Inheritance

Members

AddExternalVHDLLibraries (*library*)

AddFile (*file, fileSet=None*)

AddFileSet (*fileSet*)

AddSourceFile (*file, fileSet=None*)

Board

CreateFileSet (*name, setDefault=True*)

DefaultFileSet

Device

Environment

ExternalVHDLLibraries

ExtractVHDLLibrariesFromVHDLSourceFiles ()

FileSets

Files (*fileType=<FileTypes(Text\ProjectFile\FileListFile\RulesFile\SourceFile\VHDLSourceFile\VerilogSourceFile\Python\bits=0xFFFF>,fileSet=None)*)

GetVariables ()

Name

RootDirectory

Tool

ToolChain

VHDLLibraries

VHDLVersion

pprint (*indent=0*)

class DataBase.Solution.**FileListFile** (*file, project=None, fileSet=None*)

Inheritance**Members**

_FileType = <FileTypes.FileListFile bits=0x0004 data=UNDEFINED>

_classVHDLSourceFile

alias of *Parser.FilesParser.VHDLSourceFileMixIn*

_classVerilogSourceFile

alias of *Parser.FilesParser.VerilogSourceFileMixIn*

_classCocotbSourceFile

alias of *Parser.FilesParser.CocotbSourceFileMixIn*

Parse (*host*)

CopyFilesToFileSet ()

CopyExternalLibraries ()

FileName

FileSet

FileType

Files

Includes

Libraries

Open ()

Path

Project

ReadFile ()

Warnings

_Evaluate (*host, expr*)

_EvaluatePath (*host, expr*)

_Parse ()

```
_ReadContent ()  
_Resolve (host, statements=None)  
_classIncludeFile  
    alias of Parser.FilesParser.IncludeFileMixin  
_classLDCSourceFile  
    alias of Parser.FilesParser.LDCSourceFileMixin  
_classSDCSourceFile  
    alias of Parser.FilesParser.SDCSourceFileMixin  
_classUCFSourceFile  
    alias of Parser.FilesParser.UCFSourceFileMixin  
_classXDCSourceFile  
    alias of Parser.FilesParser.XDCSourceFileMixin  
class DataBase.Solution.RulesFile (file, project=None, fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.RulesFile bits=0x0008 data=UNDEFINED>  
FileName  
FileSet  
FileType  
Open ()  
Path  
PostProcessRules  
PreProcessRules  
Project  
ReadFile ()  
_Parse ()  
_ReadContent ()  
_Resolve ()  
_ResolveRule (ruleStatement, lst)  
_classAppendLineRule  
    alias of Parser.RulesParser.AppendLineRuleMixin  
_classCopyRule  
    alias of Parser.RulesParser.CopyRuleMixin  
_classDeleteRule  
    alias of Parser.RulesParser.DeleteRuleMixin  
_classReplaceRule  
    alias of Parser.RulesParser.ReplaceRuleMixin  
Parse ()
```

14.4.4 DataBase.TestCase

Classes

- *SimulationStatus*: An enumeration.
- *CompileStatus*: An enumeration.
- *ElementBase*: Undocumented.
- *GroupBase*: Undocumented.
- *TestGroup*: Undocumented.
- *SynthesisGroup*: Undocumented.
- *SuiteMixin*: Undocumented.
- *TestSuite*: Undocumented.
- *SynthesisSuite*: Undocumented.
- *TestBase*: Undocumented.
- *TestCase*: Undocumented.
- *Synthesis*: Undocumented.

```
class DataBase.TestCase.SimulationStatus
    An enumeration.
```

Inheritance

Members

```
Unknown = 0
DryRun = 1
SystemError = 5
InternalError = 6
AnalyzeError = 7
ElaborationError = 8
SimulationError = 9
SimulationFailed = 10
SimulationNoAsserts = 15
SimulationSuccess = 20
SimulationGUIRun = 30
```

```
class DataBase.TestCase.CompileStatus
    An enumeration.
```

Inheritance

Members

```
Unknown = 0
DryRun = 1
```

```
SystemError = 5
InternalError = 6
CompileError = 7
CompileFailed = 10
CompileSuccess = 20
```

```
class DataBase.TestCase.ElementBase (name, parent)
```

Inheritance

Members

Name

Parent

```
class DataBase.TestCase.GroupBase (name, parent)
```

Inheritance

Members

Groups

Count

Name

Parent

```
class DataBase.TestCase.TestGroup (name, parent)
```

Inheritance

Members

TestCases

PassedCount

NoAssertsCount

DryRunCount

FailedCount

ErrorCount

Count

Groups

Name

Parent

```
class DataBase.TestCase.SynthesisGroup (name, parent)
```

Inheritance

Members

Synthesises
SuccessCount
DryRunCount
FailedCount
ErrorCount
Count
Groups
Name
Parent

```
class DataBase.TestCase.SuiteMixIn
```

Inheritance

Members

StartTimer()
StopTimer()
StartTime
EndTime
InitializationTime
OverallRunTime

```
class DataBase.TestCase.TestSuite
```

Inheritance

Members

IsAllPassed
AddTestCase(*testCase*)
Count
DryRunCount
EndTime
ErrorCount
FailedCount
Groups
InitializationTime
Name
NoAssertsCount

OverallRunTime
Parent
PassedCount
StartTime
StartTimer()
StopTimer()
TestCases

class DataBase.TestCase.**SynthesisSuite**

Inheritance

Members

IsAllSuccess
AddSynthesis (*synthesis*)
Count
DryRunCount
EndTime
ErrorCount
FailedCount
Groups
InitializationTime
Name
OverallRunTime
Parent
StartTime
StartTimer()
StopTimer()
SuccessCount
Synthesises

class DataBase.TestCase.**TestBase** (*test*)

Inheritance

Members

Parent
TestGroup
Status
StartTimer()
StopTimer()

OverallRunTime

Name

```
class DataBase.TestCase.TestCase (testbench)
```

Inheritance

Members

Testbench

UpdateStatus (*testResult*)

Name

OverallRunTime

Parent

StartTimer ()

Status

StopTimer ()

TestGroup

```
class DataBase.TestCase.Synthesis (synthesis)
```

Inheritance

Members

Name

OverallRunTime

Parent

StartTimer ()

Status

StopTimer ()

TestGroup

Netlist

UpdateStatus (*synthResult*)

Variables

- `__POC_SOLUTION_KEYWORD__`
- `__POC_PROJECT_KEYWORD__`

Classes

- `Query`: Undocumented.

`DataBase.__POC_SOLUTION_KEYWORD__`

`str(object='') -> str(str(bytes_or_buffer[, encoding[, errors]]) -> str`

Create a new string object from the given object. If encoding or errors is specified, then the object must expose a data buffer that will be decoded using the given encoding and error handler. Otherwise, returns the

result of object.__str__() (if defined) or repr(object). encoding defaults to sys.getdefaultencoding(). errors defaults to 'strict'.

'Solution'

DataBase.__POC_PROJECT_KEYWORD__

str(object='') -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

Create a new string object from the given object. If encoding or errors is specified, then the object must expose a data buffer that will be decoded using the given encoding and error handler. Otherwise, returns the result of object.__str__() (if defined) or repr(object). encoding defaults to sys.getdefaultencoding(). errors defaults to 'strict'.

'Project'

class DataBase.**Query** (*host*)

Inheritance

Members

Host

Platform

PoCConfig

QueryConfiguration (*query*)

_GetModelSimInstallationDirectory ()

_GetModelSimBinaryDirectory ()

_GetXilinxISESettingsFile ()

_GetXilinxVivadoSettingsFile ()

14.5 Parser

Submodules

14.5.1 Parser.FilesCodeDOM

Classes

- *BlockedStatement*: Undocumented.
- *IfThenElseExpressions*: Undocumented.
- *ListElementExpressions*: Undocumented.
- *PathExpressions*: Undocumented.
- *ListConstructorExpression*: Undocumented.
- *SubDirectoryExpression*: Undocumented.
- *ConcatenateExpression*: Undocumented.
- *ExistsFunction*: Undocumented.
- *VHDLStatement*: Undocumented.
- *VerilogStatement*: Undocumented.

- *CocotbStatement*: Undocumented.
- *ConstraintStatement*: Undocumented.
- *LDCStatement*: Undocumented.
- *SDCStatement*: Undocumented.
- *UCFStatement*: Undocumented.
- *XDCStatement*: Undocumented.
- *InterpolateLiteral*: Undocumented.
- *PathStatement*: Undocumented.
- *ReportStatement*: Undocumented.
- *LibraryStatement*: Undocumented.
- *IncludeStatement*: Undocumented.
- *IfStatement*: Undocumented.
- *ElseIfStatement*: Undocumented.
- *ElseStatement*: Undocumented.
- *IfElseIfElseStatement*: Undocumented.
- *Document*: Undocumented.

```
class Parser.FilesCodeDOM.BlockedStatement
```

Inheritance

Members

```
_allowedStatements = [<class 'Parser.FilesCodeDOM.IncludeStatement'>, <class 'Parser.FilesCodeDOM.IfStatement'>]
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
```

```
class Parser.FilesCodeDOM.IfThenElseExpressions
```

Inheritance

Members

```
_allowedExpressions = [<class 'lib.CodeDOM.Identifier'>, <class 'lib.CodeDOM.StringLiteral'>]
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
```

```
class Parser.FilesCodeDOM.ListElementExpressions
```

Inheritance

Members

```
_allowedExpressions = [<class 'lib.CodeDOM.Identifier'>, <class 'lib.CodeDOM.StringI
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
class Parser.FilesCodeDOM.PathExpressions
```

Inheritance

Members

```
_allowedExpressions = [<class 'lib.CodeDOM.Identifier'>, <class 'lib.CodeDOM.StringI
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
class Parser.FilesCodeDOM.ListConstructorExpression
```

Inheritance

Members

```
List
AddElement (element)
classmethod GetParser ()
classmethod Parse (string, printChar)
class Parser.FilesCodeDOM.SubDirectoryExpression (leftChild, rightChild)
```

Inheritance

Members

```
classmethod GetParser ()
LeftChild
classmethod Parse (string, printChar)
RightChild
class Parser.FilesCodeDOM.ConcatenateExpression (leftChild, rightChild)
```

Inheritance

Members

`classmethod GetParser()`

`LeftChild`

`classmethod Parse(string, printChar)`

`RightChild`

`class Parser.FilesCodeDOM.ExistsFunction(expression)`

Inheritance

Members

`Expression`

`classmethod GetParser()`

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.VHDLStatement(libraryName, pathExpression, commentText)`

Inheritance

Members

`LibraryName`

`PathExpression`

`classmethod GetParser()`

`CommentText`

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.VerilogStatement(pathExpression, commentText)`

Inheritance

Members

`PathExpression`

`classmethod GetParser()`

`CommentText`

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.CocotbStatement(pathExpression, commentText)`

Inheritance

Members

PathExpression

classmethod **GetParser**()

CommentText

classmethod **Parse**(*string*, *printChar*)

class **Parser.FilesCodeDOM.ConstraintStatement** (*pathExpression*, *commentText*)

Inheritance

Members

PathExpression

classmethod **GetParser**()

CommentText

classmethod **Parse**(*string*, *printChar*)

class **Parser.FilesCodeDOM.LDCStatement** (*pathExpression*, *commentText*)

Inheritance

Members

CommentText

classmethod **GetParser**()

classmethod **Parse**(*string*, *printChar*)

PathExpression

class **Parser.FilesCodeDOM.SDCStatement** (*pathExpression*, *commentText*)

Inheritance

Members

CommentText

classmethod **GetParser**()

classmethod **Parse**(*string*, *printChar*)

PathExpression

class **Parser.FilesCodeDOM.UCFStatement** (*pathExpression*, *commentText*)

Inheritance

Members

CommentText

classmethod **GetParser**()

classmethod **Parse**(*string*, *printChar*)

PathExpression

class **Parser.FilesCodeDOM.XDCStatement** (*pathExpression*, *commentText*)

Inheritance

Members

CommentText

classmethod **GetParser**()

classmethod **Parse**(*string*, *printChar*)

PathExpression

class **Parser.FilesCodeDOM.InterpolateLiteral** (*sectionName*, *optionName*)

Inheritance

Members

SectionName

OptionName

classmethod **GetParser**()

classmethod **Parse**(*string*, *printChar*)

class **Parser.FilesCodeDOM.PathStatement** (*variable*, *pathExpression*, *commentText*)

Inheritance

Members

Variable

PathExpression

classmethod **GetParser**()

CommentText

classmethod **Parse**(*string*, *printChar*)

class **Parser.FilesCodeDOM.ReportStatement** (*message*, *commentText*)

Inheritance

Members

Message

`classmethod GetParser()`

CommentText

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.LibraryStatement (library, pathExpression, commentText)`

Inheritance

Members

Library

PathExpression

`classmethod GetParser()`

CommentText

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.IncludeStatement (pathExpression, commentText)`

Inheritance

Members

PathExpression

`classmethod GetParser()`

CommentText

`classmethod Parse(string, printChar)`

`class Parser.FilesCodeDOM.IfStatement (expression, commentText)`

Inheritance

Members

`classmethod GetParser()`

`AddStatement(stmt)`

CommentText

Expression

`classmethod Parse(string, printChar)`

Statements

`class Parser.FilesCodeDOM.ElseIfStatement (expression, commentText)`

Inheritance

Members

`classmethod` `GetParser()`

`AddStatement` (*stmt*)

`CommentText`

`Expression`

`classmethod` `Parse` (*string*, *printChar*)

`Statements`

`class` `Parser.FilesCodeDOM.ElseStatement` (*commentText*)

Inheritance

Members

`classmethod` `GetParser()`

`AddStatement` (*stmt*)

`CommentText`

`classmethod` `Parse` (*string*, *printChar*)

`Statements`

`class` `Parser.FilesCodeDOM.IfElseIfElseStatement`

Inheritance

Members

`IfClause`

`ElseIfClauses`

`ElseClause`

`classmethod` `GetParser()`

`CommentText`

`classmethod` `Parse` (*string*, *printChar*)

`class` `Parser.FilesCodeDOM.Document` (*commentText=""*)

Inheritance

Members

`AddStatement` (*stmt*)

`CommentText`

`classmethod` `Parse` (*string*, *printChar*)

`Statements`

```
classmethod GetParser()
```

14.5.2 Parser.FilesParser

Classes

- *FileReference*: Undocumented.
- *IncludeFileMixIn*: Undocumented.
- *VHDLSourceFileMixIn*: Undocumented.
- *VerilogSourceFileMixIn*: Undocumented.
- *CocotbSourceFileMixIn*: Undocumented.
- *LDCSourceFileMixIn*: Undocumented.
- *SDCSourceFileMixIn*: Undocumented.
- *UCFSourceFileMixIn*: Undocumented.
- *XDCSourceFileMixIn*: Undocumented.
- *VHDLLibraryReference*: Undocumented.
- *FilesParserMixIn*: Undocumented.

```
class Parser.FilesParser.FileReference (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.IncludeFileMixIn (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.VHDLSourceFileMixIn (file, library)
```

Inheritance

Members

LibraryName

File

```
class Parser.FilesParser.VerilogSourceFileMixIn (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.CocotbSourceFileMixin (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.LDCSourceFileMixin (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.SDCSourceFileMixin (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.UCFSourceFileMixin (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.XDCSourceFileMixin (file)
```

Inheritance

Members

File

```
class Parser.FilesParser.VHDDLlibraryReference (name, path)
```

Inheritance

Members

Name

Path

class Parser.FilesParser.FilesParserMixin

Inheritance

Members

```
_classIncludeFile
    alias of IncludeFileMixin

_classVHDLSourceFile
    alias of VHDLSourceFileMixin

_classVerilogSourceFile
    alias of VerilogSourceFileMixin

_classCocotbSourceFile
    alias of CocotbSourceFileMixin

_classLDCSourceFile
    alias of LDCSourceFileMixin

_classSDCSourceFile
    alias of SDCSourceFileMixin

_classUCFSourceFile
    alias of UCFSourceFileMixin

_classXDCSourceFile
    alias of XDCSourceFileMixin

_Parse ()

_Resolve (host, statements=None)

_Evaluate (host, expr)

_EvaluatePath (host, expr)

Files

Includes

Libraries

Warnings
```

14.5.3 Parser.RulesCodeDOM

Classes

- *InFileStatements*: Undocumented.
- *PreProcessStatements*: Undocumented.
- *PostProcessStatements*: Undocumented.
- *DocumentStatements*: Undocumented.

- *CopyStatement*: Undocumented.
- *DeleteStatement*: Undocumented.
- *ReplaceStatement*: Undocumented.
- *AppendLineStatement*: Undocumented.
- *FileStatement*: Undocumented.
- *ProcessRulesBlockStatement*: Undocumented.
- *PreProcessRulesStatement*: Undocumented.
- *PostProcessRulesStatement*: Undocumented.
- *Document*: Undocumented.

```
class Parser.RulesCodeDOM.InFileStatements
```

Inheritance

Members

```
_allowedStatements = [<class 'Parser.RulesCodeDOM.ReplaceStatement'>, <class 'Parser.R
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
```

```
class Parser.RulesCodeDOM.PreProcessStatements
```

Inheritance

Members

```
_allowedStatements = [<class 'Parser.RulesCodeDOM.CopyStatement'>, <class 'Parser.R
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
```

```
class Parser.RulesCodeDOM.PostProcessStatements
```

Inheritance

Members

```
_allowedStatements = [<class 'Parser.RulesCodeDOM.CopyStatement'>, <class 'Parser.R
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
```

```
class Parser.RulesCodeDOM.DocumentStatements
```

Inheritance

Members

```
_allowedStatements = [<class 'Parser.RulesCodeDOM.PreProcessRulesStatement'>, <class 'Parser.RulesCodeDOM.CopyStatement'>]
classmethod AddChoice (value)
classmethod GetParser ()
classmethod Parse (string, printChar)
class Parser.RulesCodeDOM.CopyStatement (source, destination, commentText)
```

Inheritance

Members

```
SourcePath
DestinationPath
classmethod GetParser ()
CommentText
classmethod Parse (string, printChar)
class Parser.RulesCodeDOM.DeleteStatement (file, commentText)
```

Inheritance

Members

```
FilePath
classmethod GetParser ()
CommentText
classmethod Parse (string, printChar)
class Parser.RulesCodeDOM.ReplaceStatement (searchPattern, replacePattern, caseInsensitive, multiLine, dotAll, commentText)
```

Inheritance

Members

```
SearchPattern
ReplacePattern
CaseInsensitive
MultiLine
DotAll
classmethod GetParser ()
CommentText
classmethod Parse (string, printChar)
```

```
class Parser.RulesCodeDOM.AppendLineStatement (appendPattern, commentText)
```

Inheritance

Members

AppendPattern

```
classmethod GetParser ()
```

CommentText

```
classmethod Parse (string, printChar)
```

```
class Parser.RulesCodeDOM.FileStatement (file, commentText)
```

Inheritance

Members

FilePath

```
classmethod GetParser ()
```

```
AddStatement (stmt)
```

CommentText

```
classmethod Parse (string, printChar)
```

Statements

```
class Parser.RulesCodeDOM.ProcessRulesBlockStatement (commentText)
```

Inheritance

Members

```
classmethod GetParser ()
```

```
AddStatement (stmt)
```

CommentText

```
classmethod Parse (string, printChar)
```

Statements

```
class Parser.RulesCodeDOM.PreProcessRulesStatement (commentText)
```

Inheritance

Members

```
AddStatement (stmt)
```

CommentText

```
classmethod GetParser ()
```

```
classmethod Parse (string, printChar)
```

Statements

```
class Parser.RulesCodeDOM.PostProcessRulesStatement (commentText)
```

Inheritance

Members

AddStatement (*stmt*)

CommentText

classmethod **GetParser** ()

classmethod **Parse** (*string*, *printChar*)

Statements

```
class Parser.RulesCodeDOM.Document (commentText="")
```

Inheritance

Members

classmethod **GetParser** ()

AddStatement (*stmt*)

CommentText

classmethod **Parse** (*string*, *printChar*)

Statements

14.5.4 Parser.RulesParser

Classes

- *Rule*: Base class for all Rule and RuleMixin classes.
- *CopyRuleMixin*: A partial class (Mixin) to represent a ‘copy rule’.
- *DeleteRuleMixin*: A partial class (Mixin) to represent a ‘delete rule’.
- *ReplaceRuleMixin*: A partial class (Mixin) to represent a ‘replace rule’.
- *AppendLineRuleMixin*: A partial class (Mixin) to represent a ‘append line rule’.
- *RulesParserMixin*: Undocumented.

```
class Parser.RulesParser.Rule
    Base class for all Rule and RuleMixin classes.
```

Inheritance

Members

```
class Parser.RulesParser.CopyRuleMixin (sourcePath, destinationPath)
    A partial class (Mixin) to represent a ‘copy rule’.
```


Inheritance

Members

SourcePath

DestinationPath

class Parser.RulesParser.**DeleteRuleMixin** (*filePath*)
A partial class (Mixin) to represent a ‘delete rule’.

Inheritance

Members

FilePath

class Parser.RulesParser.**ReplaceRuleMixin** (*filePath*, *searchPattern*, *replacePattern*,
multiLine, *dotAll*, *caseInsensitive*)
A partial class (Mixin) to represent a ‘replace rule’.

Inheritance

Members

FilePath

SearchPattern

ReplacePattern

RegExpOption_MultiLine

RegExpOption_DotAll

RegExpOption_CaseInsensitive

class Parser.RulesParser.**AppendLineRuleMixin** (*filePath*, *appendPattern*)
A partial class (Mixin) to represent a ‘append line rule’.

Inheritance

Members

FilePath

AppendPattern

class Parser.RulesParser.**RulesParserMixin**

Inheritance

Members

_classCopyRule
alias of *CopyRuleMixin*

```
_classDeleteRule
    alias of DeleteRuleMixin

_classReplaceRule
    alias of ReplaceRuleMixin

_classAppendLineRule
    alias of AppendLineRuleMixin

_Parse()

_Resolve()

_ResolveRule (ruleStatement, lst)

PreProcessRules

PostProcessRules
```

14.6 Simulator

Submodules

14.6.1 Simulator.ActiveHDL Simulator

Classes

- *Simulator*: Base class for all Simulator classes.

```
class Simulator.ActiveHDL Simulator.Simulator (host, dryRun, simulationSteps)
```

Inheritance

Members

```
TOOL_CHAIN = 10

TOOL = ('ASIM', 'Aldec Active-HDL', 'Aldec Active-HDL')

_PrepareSimulator()
    Create the Active-HDL executable factory.

_RunAnalysis()

_RunSimulation (testbench)

_RunSimulationWithGUI (testbench)

Directories

DryRun

ENVIRONMENT = 1

Host

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)
```

```

LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
PoCProject
PrintOverallSimulationReport ()
PrintSimulationReportLine (testObject, indent, nameColumnWidth)
Run (testbench, board, vhdlVersion, vhdlGenerics=None)
    Write the Testbench message line, create a PoCProject and add the first *.files file to it.
RunAll (fqnList, *args, **kwargs)
    Run a list of testbenches. Expand wildcards to all selected testbenches.
TestSuite
TryRun (testbench, *args, **kwargs)
    Try to run a testbench. Skip skipable exceptions by printing the error and its cause.
VHDLVersion
VHDL_VERSION = 2008
_AddFileListFile (fileListFilePath)
_CreatePoCProject (projectName, board)
_GetHDLParameters (configSectionName)
    Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.
_GetTimeDeltaSinceLastEvent ()
_Prepare ()
_PrepareEnvironment ()
_PrepareEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.
_PrepareEnvironment_CreatingDirectory ()
_PrepareEnvironment_PurgeDirectory ()
_PrepareSimulationEnvironment ()
_RunCoverage (testbench)
_RunElaboration (testbench)
_RunView (testbench)
_TryLog (*args, condition=True, **kwargs)

```

14.6.2 Simulator.CocotbSimulator

Classes

- *Simulator*: Base class for all Simulator classes.

```
class Simulator.CocotbSimulator.Simulator (host, dryRun, simulationSteps)
```

Inheritance

Members

TOOL_CHAIN = 30

TOOL = ('COCO', 'Cocotb', 'Coroutine Cosimulation Testbench (Cocotb)')

COCOTB_SIMBUILD_DIRECTORY = 'sim_build'

_PrepareSimulator()

RunAll(*fqnList*, **args*, ***kwargs*)

Run a list of testbenches. Expand wildcards to all selected testbenches.

_RunSimulation(*testbench*)

Directories

DryRun

ENVIRONMENT = 1

Host

Log(*entry*, *condition*=True)

Write an entry to the local logger.

LogDebug(**args*, *condition*=True, ***kwargs*)

LogDryRun(**args*, *condition*=True, ***kwargs*)

LogError(**args*, *condition*=True, ***kwargs*)

LogFatal(**args*, *condition*=True, ***kwargs*)

LogInfo(**args*, *condition*=True, ***kwargs*)

LogNormal(**args*, *condition*=True, ***kwargs*)

LogQuiet(**args*, *condition*=True, ***kwargs*)

LogVerbose(**args*, *condition*=True, ***kwargs*)

LogWarning(**args*, *condition*=True, ***kwargs*)

Logger

Return the local logger instance.

PoCProject

PrintOverallSimulationReport()

PrintSimulationReportLine(*testObject*, *indent*, *nameColumnWidth*)

Run(*testbench*, *board*, *vhdlVersion*, *vhdlGenerics*=None)

Write the Testbench message line, create a PoCProject and add the first *.files file to it.

TestSuite

TryRun(*testbench*, **args*, ***kwargs*)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 2008

_AddFileListFile(*fileListFilePath*)

_CreatePoCProject(*projectName*, *board*)

_GetHDLParameters(*configSectionName*)

Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

```
_GetTimeDeltaSinceLastEvent ()
_Prepere ()
_PrepereEnvironment ()
_PrepereEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.
_PrepereEnvironment_CreatingDirectory ()
_PrepereEnvironment_PurgeDirectory ()
_PrepereSimulationEnvironment ()
_RunAnalysis (testbench)
_RunCoverage (testbench)
_RunElaboration (testbench)
_RunView (testbench)
_TryLog (*args, condition=True, **kwargs)
```

14.6.3 Simulator.GHDL Simulator

Classes

- *Simulator*: This class encapsulates the GHDL simulator.

class Simulator.GHDL Simulator.Simulator (*host, dryRun, simulationSteps*)
This class encapsulates the GHDL simulator.

Inheritance

Members

```
TOOL_CHAIN = 40
TOOL = ('GHDL', 'GHDL', 'GHDL')
_PrepereSimulator ()
    Create the GHDL executable factory instance.
Run (testbench, board, vhdlVersion, vhdlGenerics=None, withCoverage=False)
    Write the Testbench message line, create a PoCProject and add the first *.files file to it.
_RunAnalysis (testbench)
_SetVHDLVersionAndIEEEFlavor (ghdl)
_SetExternalLibraryReferences (ghdl)
_RunElaboration (testbench)
_RunSimulation (testbench)
_RunView (testbench)
    foo
_RunCoverage (testbench)
Directories
DryRun
ENVIRONMENT = 1
```

Host**Log** (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)**LogDryRun** (**args*, *condition=True*, ***kwargs*)**LogError** (**args*, *condition=True*, ***kwargs*)**LogFatal** (**args*, *condition=True*, ***kwargs*)**LogInfo** (**args*, *condition=True*, ***kwargs*)**LogNormal** (**args*, *condition=True*, ***kwargs*)**LogQuiet** (**args*, *condition=True*, ***kwargs*)**LogVerbose** (**args*, *condition=True*, ***kwargs*)**LogWarning** (**args*, *condition=True*, ***kwargs*)**Logger**

Return the local logger instance.

PoCProject**PrintOverallSimulationReport** ()**PrintSimulationReportLine** (*testObject*, *indent*, *nameColumnWidth*)**RunAll** (*fqnList*, **args*, ***kwargs*)

Run a list of testbenches. Expand wildcards to all selected testbenches.

TestSuite**TryRun** (*testbench*, **args*, ***kwargs*)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion**VHDL_VERSION** = 2008**_AddFileListFile** (*fileListFilePath*)**_CreatePoCProject** (*projectName*, *board*)**_GetHDLParameters** (*configSectionName*)

Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()**_Prepare** ()**_PrepareEnvironment** ()**_PrepareEnvironment_ChangeDirectory** ()

Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()**_PrepareEnvironment_PurgeDirectory** ()**_PrepareSimulationEnvironment** ()**_TryLog** (**args*, *condition=True*, ***kwargs*)

14.6.4 Simulator.ISESimulator

Classes

- *Simulator*: Base class for all Simulator classes.

class Simulator.ISESimulator.**Simulator** (*host, dryRun, simulationSteps*)

Inheritance

Members

TOOL_CHAIN = 80

TOOL = ('XSIM', 'Xilinx iSim', 'Xilinx ISE Simulator (iSim)')

_PrepareSimulator ()

_RunElaboration (*testbench*)

_RunSimulation (*testbench*)

Directories

DryRun

ENVIRONMENT = 1

Host

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PoCProject

PrintOverallSimulationReport ()

PrintSimulationReportLine (*testObject, indent, nameColumnWidth*)

Run (*testbench, board, vhdlVersion, vhdlGenerics=None*)

Write the Testbench message line, create a PoCProject and add the first *.files file to it.

RunAll (*fqnList, *args, **kwargs*)

Run a list of testbenches. Expand wildcards to all selected testbenches.

TestSuite

TryRun (*testbench, *args, **kwargs*)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

```
VHDLVersion
VHDL_VERSION = 2008
_AddFileListFile (fileListFilePath)
_CreatePoCProject (projectName, board)
_GenerateXilinxProjectFileContent (tool, vhdlVersion=93)
_GetHDLParameters (configSectionName)
    Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.
_GetTimeDeltaSinceLastEvent ()
_Prepare ()
_PrepareEnvironment ()
_PrepareEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.
_PrepareEnvironment_CreatingDirectory ()
_PrepareEnvironment_PurgeDirectory ()
_PrepareSimulationEnvironment ()
_RunAnalysis (testbench)
_RunCoverage (testbench)
_RunView (testbench)
_TryLog (*args, condition=True, **kwargs)
_WriteXilinxProjectFile (projectFilePath, tool, vhdlVersion=93)
```

14.6.5 Simulator.ModelSimSimulator

Classes

- *Simulator*: Base class for all Simulator classes.

```
class Simulator.ModelSimSimulator.Simulator (host, dryRun, simulationSteps)
```

Inheritance

Members

```
TOOL_CHAIN = 70
TOOL = ('VSIM', 'Mentor ModelSim', 'Mentor Graphics ModelSim (vSim)')
_PrepareSimulator ()
Run (testbench, board, vhdlVersion, vhdlGenerics=None, withCoverage=False)
    Write the Testbench message line, create a PoCProject and add the first *.files file to it.
_RunAnalysis (_)
_RunSimulation (testbench)
_RunSimulationWithGUI (testbench)
Directories
DryRun
ENVIRONMENT = 1
```


Host**Log** (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)**LogDryRun** (**args*, *condition=True*, ***kwargs*)**LogError** (**args*, *condition=True*, ***kwargs*)**LogFatal** (**args*, *condition=True*, ***kwargs*)**LogInfo** (**args*, *condition=True*, ***kwargs*)**LogNormal** (**args*, *condition=True*, ***kwargs*)**LogQuiet** (**args*, *condition=True*, ***kwargs*)**LogVerbose** (**args*, *condition=True*, ***kwargs*)**LogWarning** (**args*, *condition=True*, ***kwargs*)**Logger**

Return the local logger instance.

PoCProject**PrintOverallSimulationReport** ()**PrintSimulationReportLine** (*testObject*, *indent*, *nameColumnWidth*)**RunAll** (*fqnList*, **args*, ***kwargs*)

Run a list of testbenches. Expand wildcards to all selected testbenches.

TestSuite**TryRun** (*testbench*, **args*, ***kwargs*)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion**VHDL_VERSION** = 2008**_AddFileListFile** (*fileListFilePath*)**_CreatePoCProject** (*projectName*, *board*)**_GetHDLParameters** (*configSectionName*)

Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()**_Prepare** ()**_PrepareEnvironment** ()**_PrepareEnvironment_ChangeDirectory** ()

Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()**_PrepareEnvironment_PurgeDirectory** ()**_PrepareSimulationEnvironment** ()**_RunCoverage** (*testbench*)**_RunElaboration** (*testbench*)**_RunView** (*testbench*)**_TryLog** (**args*, *condition=True*, ***kwargs*)

14.6.6 Simulator.QuestaSimulator

Classes

- *Simulator*: Base class for all Simulator classes.

class Simulator.QuestaSimulator.Simulator (*host, dryRun, simulationSteps*)

Inheritance

Members

TOOL_CHAIN = 75

TOOL = ('VSIM', 'Mentor ModelSim', 'Mentor Graphics ModelSim (vSim)')

_PrepareSimulator ()

Run (*testbench, board, vhdlVersion, vhdlGenerics=None*)

Write the Testbench message line, create a PoCProject and add the first *.files file to it.

RunAnalysis ()

_RunSimulation (*testbench*)

_RunSimulationWithGUI (*testbench*)

Directories

DryRun

ENVIRONMENT = 1

Host

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PoCProject

PrintOverallSimulationReport ()

PrintSimulationReportLine (*testObject, indent, nameColumnWidth*)

RunAll (*fqnList, *args, **kwargs*)

Run a list of testbenches. Expand wildcards to all selected testbenches.

TestSuite

```

TryRun (testbench, *args, **kwargs)
    Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 2008

_AddFileListFile (fileListFilePath)

_CreatePoCProject (projectName, board)

_GetHDLParameters (configSectionName)
    Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_Prepare ()

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_PrepareSimulationEnvironment ()

_RunCoverage (testbench)

_RunElaboration (testbench)

_RunView (testbench)

_TryLog (*args, condition=True, **kwargs)

```

14.6.7 Simulator.RivieraPROSimulator

14.6.8 Simulator.VivadoSimulator

Classes

- *Simulator*: Base class for all Simulator classes.

```
class Simulator.VivadoSimulator.Simulator (host, dryRun, simulationSteps)
```

Inheritance

Members

```

TOOL_CHAIN = 82

TOOL = ('XSIM', 'Xilinx xSim', 'Xilinx Vivado Simulator (xSim)')

_PrepareSimulator ()

_RunElaboration (testbench)

_RunSimulation (testbench)

Directories

DryRun

ENVIRONMENT = 1

Host

```

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

PoCProject

PrintOverallSimulationReport ()

PrintSimulationReportLine (*testObject*, *indent*, *nameColumnWidth*)

Run (*testbench*, *board*, *vhdlVersion*, *vhdlGenerics=None*)
Write the Testbench message line, create a PoCProject and add the first *.files file to it.

RunAll (*fqnList*, **args*, ***kwargs*)
Run a list of testbenches. Expand wildcards to all selected testbenches.

TestSuite

TryRun (*testbench*, **args*, ***kwargs*)
Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

VHDLVersion

VHDL_VERSION = 2008

_AddFileListFile (*fileListFilePath*)

_CreatePoCProject (*projectName*, *board*)

_GenerateXilinxProjectFileContent (*tool*, *vhdlVersion=93*)

_GetHDLParameters (*configSectionName*)
Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.

_GetTimeDeltaSinceLastEvent ()

_Prepare ()

_PrepareEnvironment ()

_PrepareEnvironment_ChangeDirectory ()
Change working directory to temporary path 'temp/<tool>'.

_PrepareEnvironment_CreatingDirectory ()

_PrepareEnvironment_PurgeDirectory ()

_PrepareSimulationEnvironment ()

_RunAnalysis (*testbench*)

_RunCoverage (*testbench*)

_RunView (*testbench*)

```
_TryLog (*args, condition=True, **kwargs)
```

```
_WriteXilinxProjectFile (projectFilePath, tool, vhdlVersion=93)
```

Exceptions

- *SimulatorException*: Base class for all SimulatorException classes. It is raised while running
- *SkipableSimulatorException*: *SkipableSimulatorException* is a *SimulatorException*, which
- *PoCSimulationResultNotFoundException*: This exception is raised if the expected PoC simulation result string was

Classes

- *SimulationSteps*: Simulation step enumeration.
- *SimulationState*: Simulation state enumeration.
- *SimulationResult*: Simulation result enumeration.
- *Simulator*: Base class for all Simulator classes.

Functions

- *PoCSimulationResultFilter()*: Undocumented.

exception `Simulator.SimulatorException (message=)`

Base class for all SimulatorException classes. It is raised while running simulation tasks in PoC.

Inheritance

Members

```
__init__ (message=)
```

Exception initializer

Parameters `message (str)` – The exception message.

```
__str__ ()
```

Returns the exception's message text.

args

exception `Simulator.SkipableSimulatorException (message=)`

SkipableSimulatorException is a *SimulatorException*, which can be skipped.

Inheritance

Members

```
__init__ (message=)
```

Exception initializer

Parameters `message (str)` – The exception message.

```
__str__ ()
```

Returns the exception's message text.

args

exception `Simulator.PoCSimulationResultNotFoundException (message=)`

This exception is raised if the expected PoC simulation result string was not found in the simulator's output.

Inheritance

Members

`__init__(message="")`

Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`

Returns the exception's message text.

args

class `Simulator.SimulationSteps`

Simulation step enumeration.

Inheritance

Members

`_FlagsArithmeticMixin__bits`

`_FlagsArithmeticMixin__create_flags_instance(bits)`

`_Flags__internal_str()`

classmethod `bits_from_simple_str(s)`

classmethod `bits_from_str(s)`

Converts the output of `__str__` into an integer.

data

classmethod `from_simple_str(s)`

Accepts only the output of `to_simple_str()`. The output of `__str__()` is invalid as input.

classmethod `from_str(s)`

Accepts both the output of `to_simple_str()` and `__str__()`.

is_disjoint (**flags_instances*)

is_member

flags.is_member is a shorthand for *flags.properties is not None*. If this property is False then this Flags instance has either zero bits or holds a combination of flag member bits. If this property is True then the bits of this Flags instance match exactly the bits associated with one of the members. This however doesn't necessarily mean that this flag instance isn't a combination of several flags because the bits of a member can be the subset of another member. For example if `member0_bits=0x1` and `member1_bits=0x3` then the bits of member0 are a subset of the bits of member1. If a flag instance holds the bits of member1 then `Flags.is_member` returns True and `Flags.properties` returns the properties of member1 but `__len__()` returns 2 and `__iter__()` yields both member0 and member1.

name

properties

Returns Returns None if this flag isn't an exact member of a flags class but a combination of flags,

returns an object holding the properties (e.g.: name, data, index, ...) of the flag otherwise. We don't store flag properties directly in Flags instances because this way Flags instances that are the (temporary) result of flags arithmetic don't have to maintain these fields and it also has some benefits regarding memory usage.

to_simple_str()

```
class Simulator.SimulationState
    Simulation state enumeration.
```

Inheritance

Members

```
Prepare = 0
Analyze = 1
Elaborate = 2
Optimize = 3
Simulate = 4
View = 5
Coverage = 6
```

```
class Simulator.SimulationResult
    Simulation result enumeration.
```

Inheritance

Members

```
NotRun = 0
DryRun = 1
Error = 2
Failed = 3
NoAsserts = 4
Passed = 5
GUIRun = 6
```

```
class Simulator.Simulator(host: Base.IHost, dryRun, simulationSteps: Simula-
                           tor.SimulationSteps)
    Base class for all Simulator classes.
```

Inheritance

Members

```
ENVIRONMENT = 1
VHDL_VERSION = 2008
__init__(host: Base.IHost, dryRun, simulationSteps: Simulator.SimulationSteps)
    Class initializer
```

Parameters

- **host** (*object*) – The hosting instance for this instance.
- **dryRun** (*bool*) – Enable dry-run mode
- **simulationSteps** (*SimulationSteps*) – A set of simulation step to precess.

TestSuite**_PrepareSimulationEnvironment** ()**_PrepareEnvironment_PurgeDirectory** ()**_PrepareSimulator** ()**RunAll** (*fqnList*, *args, **kwargs)

Run a list of testbenches. Expand wildcards to all selected testbenches.

TryRun (*testbench*, *args, **kwargs)

Try to run a testbench. Skip skipable exceptions by printing the error and its cause.

Run (*testbench*, *board*, *vhdlVersion*, *vhdlGenerics=None*)

Write the Testbench message line, create a PoCProject and add the first *.files file to it.

_RunAnalysis (*testbench*)**_RunElaboration** (*testbench*)**_RunSimulation** (*testbench*)**_RunView** (*testbench*)**_RunCoverage** (*testbench*)**PrintOverallSimulationReport** ()**PrintSimulationReportLine** (*testObject*, *indent*, *nameColumnWidth*)**Directories****DryRun****Host****Log** (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (*args, *condition=True*, **kwargs)**LogDryRun** (*args, *condition=True*, **kwargs)**LogError** (*args, *condition=True*, **kwargs)**LogFatal** (*args, *condition=True*, **kwargs)**LogInfo** (*args, *condition=True*, **kwargs)**LogNormal** (*args, *condition=True*, **kwargs)**LogQuiet** (*args, *condition=True*, **kwargs)**LogVerbose** (*args, *condition=True*, **kwargs)**LogWarning** (*args, *condition=True*, **kwargs)**Logger**

Return the local logger instance.

PoCProject**TOOL** = 0**TOOL_CHAIN** = 0**VHDLVersion****_AddFileListFile** (*fileListFilePath*)**_CreatePoCProject** (*projectName*, *board*)**_GetHDLParameters** (*configSectionName*)

Parse option 'HDLParameters' for Verilog Parameters / VHDL Generics.


```
_GetTimeDeltaSinceLastEvent ()
_Prepere ()
_PrepereEnvironment ()
_PrepereEnvironment_ChangeDirectory ()
    Change working directory to temporary path 'temp/<tool>'.
_PrepereEnvironment_CreatingDirectory ()
_TryLog (*args, condition=True, **kwargs)
```

Functions

`Simulator.PoCSimulationResultFilter` (*gen*, *simulationResult*)

14.7 ToolChain

Submodules

14.7.1 ToolChain.Aldec

Submodules

ToolChain.Aldec.ActiveHDL

Exceptions

- *ActiveHDLException*: An ActiveHDLException is raised if Active-HDL catches a system exception.

Classes

- *AldecActiveHDLEditions*: Enumeration of all Active-HDL editions provided by Aldec itself.
- *ActiveHDLEditions*: Enumeration of all Active-HDL editions provided by Aldec inclusive editions
- *Configuration*: Base class for all tool Configuration classes.
- *ActiveHDL*: Factory for executable abstractions in Active-HDL.
- *VHDLLibraryTool*: Abstraction layer of Active-HDL's VHDL library management tool 'vlib'.
- *VHDLCompiler*: Abstraction layer of Active-HDL's VHDL compiler 'vcom'.
- *VHDLStandaloneSimulator*: Abstraction layer of Active-HDL's VHDL standalone simulator 'vsimsa'.

Functions

- *VLibFilter*(): A line based output stream filter for Active-HDL's VHDL library management
- *VComFilter*(): A line based output stream filter for Active-HDL's VHDL compiler.
- *VSimFilter*(): A line based output stream filter for Active-HDL's VHDL simulator.

exception `ToolChain.Aldec.ActiveHDL.ActiveHDLException` (*message=""*)
An ActiveHDLException is raised if Active-HDL catches a system exception.

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

class `ToolChain.Aldec.ActiveHDL.AldecActiveHDLEditions` (*name, section*)
Enumeration of all Active-HDL editions provided by Aldec itself.

Inheritance

Members

`StandardEdition = 1`

`StudentEdition = 2`

class `ToolChain.Aldec.ActiveHDL.ActiveHDLEditions` (*name, section*)
Enumeration of all Active-HDL editions provided by Aldec inclusive editions shipped by other vendors.

Inheritance

Members

`StandardEdition = 1`

`LatticeEdition = 2`

class `ToolChain.Aldec.ActiveHDL.Configuration` (*host: Base.IHost*)

Inheritance

Members

`_vendor = 'Aldec'`
The name of the tools vendor.

`_toolName = 'Aldec Active-HDL'`
The name of the tool.

`_section = 'INSTALL.Aldec.ActiveHDL'`
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`
Aldec Active-HDL supports multiple versions installed on the same system.

`_template = {'Windows': {'INSTALL.Aldec.ActiveHDL': {'Version': '10.3', 'SectionName': 'Aldec Active-HDL'}}`
The template for the configuration sections represented as nested dictionaries.

`CheckDependency()`
Check if general Aldec support is configured in PoC.

ConfigureForAll ()

Configuration routine for Aldec Active-HDL on all supported platforms.

1. Ask if Active-HDL is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for Active-HDL.
 - No → clear the Active-HDL configuration section.
1. Ask for Active-HDL's version.
2. Ask for Active-HDL's edition (normal, student).
3. Ask for Active-HDL's installation directory.

_ConfigureEdition ()

Configure Active-HDL for Aldec.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)**

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_Configuration__CheckActiveHDLVersion (*binPath, version*)
Compare the given Active-HDL version with the tool's version string.

_ConfigureBinaryDirectory ()
Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()
If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()
Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)
Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

```
class ToolChain.Aldec.ActiveHDL.ActiveHDL (platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Factory for executable abstractions in Active-HDL.

Inheritance

Members

```
GetVHDLLibraryTool ()
```

Return an instance of Active-HDL's VHDL library management tool 'vlib'.

```
GetVHDLCompiler ()
```

Return an instance of Active-HDL's VHDL compiler 'vcom'.

```
GetSimulator ()
```

Return an instance of Active-HDL's VHDL simulator 'vsim'.

```
class ToolChain.Aldec.ActiveHDL.VHDLLibraryTool (toolchain: ToolChain.ToolMixIn)
```

Abstraction layer of Active-HDL's VHDL library management tool 'vlib'.

Inheritance

Members

```
class Executable
```

```
    _value = None
```

```
class SwitchLibraryName
```

```
    _value = None
```

```
Parameters = [<class 'ToolChain.Aldec.ActiveHDL.VHDLLibraryTool.Executable'>, <class
```

```
CreateLibrary ()
```

```
GetReader ()
```

```
HasErrors
```

True if errors or fatals errors were found while processing the output stream.

```
HasWarnings
```

True if errors or fatals errors were found while processing the output stream.

```
Log (entry, condition=True)
```

Write an entry to the local logger.

```
LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
```

```
LogError (*args, condition=True, **kwargs)
```

```
LogFatal (*args, condition=True, **kwargs)
```

```
LogInfo (*args, condition=True, **kwargs)
```

```
LogNormal (*args, condition=True, **kwargs)
```

```
LogQuiet (*args, condition=True, **kwargs)
```

```
LogVerbose (*args, condition=True, **kwargs)
```

```
LogWarning (*args, condition=True, **kwargs)
```

Logger

Return the local logger instance.

Path

ReadUntilBoundary (*indent=0*)

Send (*line*, *end*='\\n')

SendBoundary ()

StartProcess (*parameterList*)

Terminate ()

_POC_BOUNDARY = '===== **POC BOUNDARY** ====='

_TryLog (**args*, *condition=True*, ***kwargs*)

class ToolChain.Aldec.ActiveHDL.VHDLCompiler (*toolchain: ToolChain.ToolMixIn*)
Abstraction layer of Active-HDL's VHDL compiler 'vcom'.

Inheritance**Members**

class Executable

_value = None

class FlagNoRangeCheck

_name = 'norangecheck'

_value = None

class SwitchVHDLVersion

_pattern = '-{1}'

_name = ''

_value = None

class SwitchVHDLLibrary

_name = 'work'

_value = None

class ArgSourceFile

_value = None

Parameters = [**<class 'ToolChain.Aldec.ActiveHDL.VHDLCompiler.Executable'>**, **<class 'ToolChain.Aldec.ActiveHDL.VHDLCompiler.FlagNoRangeCheck'>**, **<class 'ToolChain.Aldec.ActiveHDL.VHDLCompiler.SwitchVHDLVersion'>**, **<class 'ToolChain.Aldec.ActiveHDL.VHDLCompiler.SwitchVHDLLibrary'>**, **<class 'ToolChain.Aldec.ActiveHDL.VHDLCompiler.ArgSourceFile'>**]

Compile ()

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

```

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POC_BOUNDARY = '==== POC BOUNDARY ====='

_TryLog (*args, condition=True, **kwargs)

```

```

class ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator (toolchain:
    ToolChain.ToolMixIn)
    Abstraction layer of Active-HDL's VHDL standalone simulator 'vsimsa'.

```

Inheritance

Members

```

class Executable

```

```

    _value = None

```

```

class SwitchBatchCommand

```

```

    _name = 'do'

```

```

    _value = None

```

```

Parameters = [<class 'ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator.Executable'>]

```

```

Simulate ()

```

```

GetReader ()

```

```

HasErrors

```

True if errors or fatals errors were found while processing the output stream.

```

HasWarnings

```

True if errors or fatals errors were found while processing the output stream.

```
Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POC_BOUNDARY = '===== POC BOUNDARY ====='

_TryLog (*args, condition=True, **kwargs)
```

Functions

```
ToolChain.Aldec.ActiveHDL.VLibFilter (gen)
    A line based output stream filter for Active-HDL's VHDL library management tool.

ToolChain.Aldec.ActiveHDL.VComFilter (gen)
    A line based output stream filter for Active-HDL's VHDL compiler.

ToolChain.Aldec.ActiveHDL.VSimFilter (gen)
    A line based output stream filter for Active-HDL's VHDL simulator.
```

ToolChain.Aldec.RivieraPRO

Exceptions

- *RivieraPROException*: An RivieraPROException is raised if Riviera-PRO catches a system exception.

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *RivieraPRO*: Factory for executable abstractions in Riviera-PRO.
- *VHDLLibraryTool*: Abstraction layer of Riviera-PRO's VHDL library management tool 'vlib'.
- *VHDLCompiler*: Abstraction layer of Riviera-PRO's VHDL compiler 'vcom'.
- *VHDL Simulator*: Represent an executable.

Functions

- *VLibFilter*(): A line based output stream filter for Riviera-PRO's VHDL library management tool.

- `VComFilter()`: A line based output stream filter for Riviera-PRO's VHDL compiler.
- `VSimFilter()`: A line based output stream filter for Riviera-PRO's VHDL simulator.

exception `ToolChain.Aldec.RivieraPRO.RivieraPROException (message=)`
 An RivieraPROException is raised if Riviera-PRO catches a system exception.

Inheritance

Members

`__init__ (message=)`
 Exception initializer

Parameters `message (str)` – The exception message.

`__str__ ()`
 Returns the exception's message text.

args

class `ToolChain.Aldec.RivieraPRO.Configuration (host: Base.IHost)`

Inheritance

Members

`_vendor = 'Aldec'`
 The name of the tools vendor.

`_toolName = 'Aldec Riviera-PRO'`
 The name of the tool.

`_section = 'INSTALL.Aldec.RivieraPRO'`
 The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`
 Aldec Riviera-PRO supports multiple versions installed on the same system.

`_template = {'Linux': {'INSTALL.Aldec.RivieraPRO': {'Version': '2017.02', 'Section': ...}}`
 The template for the configuration sections represented as nested dictionaries.

CheckDependency ()
 Check if general Aldec support is configured in PoC.

ConfigureForAll ()
 Configuration routine for Aldec Riviera-PRO on all supported platforms.

1. Ask if Riviera-PRO is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for Riviera-PRO.
 - No → clear the Riviera-PRO configuration section.

1. Ask for Riviera-PRO's version.
2. Ask for Riviera-PRO's edition (normal, student).
3. Ask for Riviera-PRO's installation directory.

ClearSection (writeWarnings=False)
 Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions (writeWarnings=True)

PrepareSections (warningWasWritten, writeWarnings=True)

PrepareVersionedSections (writeWarnings=False)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)

`_AskInstalled` (*question*)

Ask a Yes/No/Pass question.

`_AskYes_NoPass` (*question, indent=1*)

Ask a yes/NO/pass question.

`_Ask_YesNoPass` (*question, indent=1*)

Ask a YES/no/pass question.

`_Configuration__CheckRivieraPROVersion` (*binPath, version*)

Compare the given Riviera-PRO version with the tool's version string.

`_ConfigureBinaryDirectory` ()

Updates section with value from `_template` and returns directory as `Path` object.

`_ConfigureEdition` (*editions, defaultEdition*)

`_ConfigureInstallationDirectory` ()

Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

`_ConfigureVersion` ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

`_GetDefaultEdition` ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

`_GetDefaultInstallationDirectory` ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

`_GetDefaultOptionValue` (*optionName*)

`_GetDefaultVersion` ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

`_PrintAvailableEditions` (*editions, selectedEdition*)

Print all available editions and return the selected index.

`_TestDefaultInstallPath` (*defaults*)

Helper function for automatic search of installation directory.

`_TryLog` (**args, condition=True, **kwargs*)

`class ToolChain.Aldec.RivieraPRO.RivieraPRO` (*platform, dryrun, binaryDirectoryPath,*
version, logger=None)

Factory for executable abstractions in Riviera-PRO.

Inheritance

Members

`GetVHDLLibraryTool` ()

Return an instance of Riviera-PRO's VHDL library management tool 'vlib'.

`GetVHDLCompiler` ()

Return an instance of Riviera-PRO's VHDL compiler 'vcom'.

`GetSimulator` ()

Return an instance of Riviera-PRO's VHDL simulator 'vsim'.

```
class ToolChain.Aldec.RivieraPRO.VHDLLibraryTool (toolchain:  
                                                    ToolChain.ToolMixIn)  
    Abstraction layer of Riviera-PRO's VHDL library management tool 'vlib'.
```

Inheritance

Members

```
class Executable
```

```
    _value = None
```

```
class SwitchLibraryName
```

```
    _value = None
```

```
Parameters = [<class 'ToolChain.Aldec.RivieraPRO.VHDLLibraryTool.Executable'>, <class 'ToolChain.Aldec.RivieraPRO.VHDLLibraryTool.SwitchLibraryName'>]
```

```
CreateLibrary ()
```

```
GetReader ()
```

```
HasErrors
```

```
    True if errors or fatals errors were found while processing the output stream.
```

```
HasWarnings
```

```
    True if errors or fatals errors were found while processing the output stream.
```

```
Log (entry, condition=True)
```

```
    Write an entry to the local logger.
```

```
LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
```

```
LogError (*args, condition=True, **kwargs)
```

```
LogFatal (*args, condition=True, **kwargs)
```

```
LogInfo (*args, condition=True, **kwargs)
```

```
LogNormal (*args, condition=True, **kwargs)
```

```
LogQuiet (*args, condition=True, **kwargs)
```

```
LogVerbose (*args, condition=True, **kwargs)
```

```
LogWarning (*args, condition=True, **kwargs)
```

```
Logger
```

```
    Return the local logger instance.
```

```
Path
```

```
ReadUntilBoundary (indent=0)
```

```
Send (line, end='\n')
```

```
SendBoundary ()
```

```
StartProcess (parameterList)
```

```
Terminate ()
```

```
_POC_BOUNDARY = '===== POC BOUNDARY ====='
```

```
_TryLog (*args, condition=True, **kwargs)
```

```
class ToolChain.Aldec.RivieraPRO.VHDLCompiler (toolchain: ToolChain.ToolMixIn)
    Abstraction layer of Riviera-PRO's VHDL compiler 'vcom'.
```

Inheritance

Members

```
class Executable
```

```
    _value = None
```

```
class SwitchVHDLVersion
```

```
    _pattern = '-{1}'
```

```
    _name = ''
```

```
    _value = None
```

```
class SwitchVHDLLibrary
```

```
    _name = 'work'
```

```
    _value = None
```

```
class ArgSourceFile
```

```
    _value = None
```

```
Parameters = [<class 'ToolChain.Aldec.RivieraPRO.VHDLCompiler.Executable'>, <class
Compile()
```

```
GetReader()
```

```
HasErrors
```

True if errors or fatals errors were found while processing the output stream.

```
HasWarnings
```

True if errors or fatals errors were found while processing the output stream.

```
Log (entry, condition=True)
```

Write an entry to the local logger.

```
LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
```

```
LogError (*args, condition=True, **kwargs)
```

```
LogFatal (*args, condition=True, **kwargs)
```

```
LogInfo (*args, condition=True, **kwargs)
```

```
LogNormal (*args, condition=True, **kwargs)
```

```
LogQuiet (*args, condition=True, **kwargs)
```

```
LogVerbose (*args, condition=True, **kwargs)
```

```
LogWarning (*args, condition=True, **kwargs)
```

```
Logger
```

Return the local logger instance.

```
Path
```

```
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POCKET_BOUNDARY = '==== POC BOUNDARY ===='
_TryLog (*args, condition=True, **kwargs)

class ToolChain.Aldec.RivieraPRO.VHDL Simulator (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
    The executable to launch.
    _value = None

class SwitchBatchCommand
    Specify a Tcl batch script for the batch mode.
    _name = 'do'
    _value = None

class FlagCommandLineMode
    Run simulation in command line mode.
    _name = 'c'
    _value = None

class SwitchTimeResolution
    Set simulation time resolution.
    _name = 't'
    _value = None

class SwitchTopLevel
    The top-level for simulation.
    _value = None

Parameters = [<class 'ToolChain.Aldec.RivieraPRO.VHDL Simulator.Executable'>, <class
    Specify all accepted command line arguments

Simulate ()
    Start a simulation.

GetReader ()

HasErrors
    True if errors or fatals errors were found while processing the output stream.

HasWarnings
    True if errors or fatals errors were found while processing the output stream.

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)
```

```
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POC_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)
```

Functions

```
ToolChain.Aldec.RivieraPRO.VLibFilter (gen)
    A line based output stream filter for Riviera-PRO's VHDL library management tool.
ToolChain.Aldec.RivieraPRO.VComFilter (gen)
    A line based output stream filter for Riviera-PRO's VHDL compiler.
ToolChain.Aldec.RivieraPRO.VSimFilter (gen)
    A line based output stream filter for Riviera-PRO's VHDL simulator.
```

Exceptions

- *AldecException*: Base class for all Aldec tool's exceptions.

Classes

- *Configuration*: Configuration routines for Aldec as a vendor.

```
exception ToolChain.Aldec.AldecException (message="")
    Base class for all Aldec tool's exceptions.
```

Inheritance

Members

```
__init__ (message="")
    Exception initializer

    Parameters message (str) – The exception message.
__str__ ()
    Returns the exception's message text.
args
```

```
class ToolChain.Aldec.Configuration (host: Base.IHost)
```

Configuration routines for Aldec as a vendor.

This configuration provides a common installation directory setup for all Aldec tools installed on a system.

Inheritance

Members

```
_vendor = 'Aldec'
```

The name of the tools vendor.

```
_section = 'INSTALL.Aldec'
```

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

```
_template = {'ALL': {'INSTALL.ActiveHDL': {'SectionName': '', 'Version': '${Se
```

The template for the configuration sections represented as nested dictionaries.

```
ConfigureForAll ()
```

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

```
_GetDefaultInstallationDirectory ()
```

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

```
CheckDependency ()
```

Check if all vendor or tool dependencies are fulfilled to configure this tool.

```
ClearSection (writeWarnings=False)
```

Clear the configuration section associated to this Configuration class.

```
ConfigureForDarwin ()
```

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

```
ConfigureForLinux ()
```

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

```
ConfigureForWindows ()
```

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

```
classmethod GetSections (platform)
```

Return all section names for this configuration.

```
Host
```

Return the hosting object.

```
IsConfigured ()
```

Return true if the configurations section is configured

```
IsSupportedPlatform ()
```

Return true if the given platform is supported by this configuration routine.

```
Log (entry, condition=True)
```

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_GetDefaultOptionValue (*optionName*)

_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)
Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

_multiVersionSupport = False

14.7.2 ToolChain.Alter

Submodules

ToolChain.Altera.ModelSim

Exceptions

- *ModelSimException*: Base class for all tool specific exceptions

Classes

- *AlteraModelSimEditions*: Enumeration of all ModelSim editions provided by Altera.
- *Configuration*: Base class for all tool Configuration classes.
- *AlteraEditionConfiguration*: Base class for all tool Configuration classes.
- *AlteraStarterEditionConfiguration*: Base class for all tool Configuration classes.

exception ToolChain.Altera.ModelSim.**ModelSimException** (*message*=")

Inheritance

Members

__init__ (*message*=")
Exception initializer

Parameters *message* (*str*) – The exception message.

__str__ ()
Returns the exception's message text.

args

class ToolChain.Altera.ModelSim.**AlteraModelSimEditions** (*name*, *section*)
Enumeration of all ModelSim editions provided by Altera.

Inheritance

Members

ModelSimAlteraEdition = 1

ModelSimAlteraStarterEdition = 2

class ToolChain.Altera.ModelSim.**Configuration** (*host*: *Base.IHost*)

Inheritance

Members

_vendor = 'Altera'
The name of the tools vendor.

_multiVersionSupport = **False**
ModelSim Altera (Starter) Edition doesn't support multiple versions.

CheckDependency ()
Check if general Altera support is configured in PoC.

ConfigureForAll ()
Configuration routine for Mentor Graphics ModelSim on all supported platforms.
1. Ask if ModelSim is installed.

- Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
 2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
 3. Ask for ModelSim's installation directory.

_ConfigureEdition ()

Configure ModelSim for Altera.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question, default, beforeDefault= "", afterDefault= "", indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)

Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath, version*)

_Configuration__GetModelSimVersion (*binPath*)

_ConfigureBinaryDirectory ()

Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (*binPath*)

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

_section = `'INSTALL.Vendor.Tool'`

_template = `{ 'ALL': { 'INSTALL.Vendor.Tool': { 'Version': '1.0' } }, 'Darwin': { 'INS'`

```
_toolName = 'Mentor ModelSim'  
class ToolChain.Altera.ModelSim.AlteraEditionConfiguration (host: Base.IHost)
```

Inheritance

Members

```
_toolName = 'Altera ModelSim'  
    The name of the tool.
```

```
__editionName = None  
    The name of the tool.
```

```
_section = 'INSTALL.Altera.ModelSimAE'  
    The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.
```

```
_template = {'Linux': {'INSTALL.Altera.ModelSimAE': {'Version': '10.5b', 'Edition
```

```
CheckDependency ()  
    Check if general Altera support is configured in PoC.
```

```
ClearSection (writeWarnings=False)  
    Clear the configuration section associated to this Configuration class.
```

```
ConfigureForAll ()  
    Configuration routine for Mentor Graphics ModelSim on all supported platforms.
```

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

```
ConfigureForDarwin ()  
    Start the configuration procedure for Darwin.
```

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

```
ConfigureForLinux ()  
    Start the configuration procedure for Linux.
```

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

```
ConfigureForWindows ()  
    Start the configuration procedure for Windows.
```

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

```
classmethod GetSections (platform)  
    Return all section names for this configuration.
```

```
Host  
    Return the hosting object.
```

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten*, *writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_AlteraEditionConfiguration__editionName = 'ModelSim Altera Edition'

_Ask (*question*, *default*, *beforeDefault=""*, *afterDefault=""*, *indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)
Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath*, *version*)

_Configuration__GetModelSimVersion (*binPath*)

_ConfigureBinaryDirectory ()
Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition ()
Configure ModelSim for Altera.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)**_GetDefaultVersion ()**

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (binPath)**_PrintAvailableEditions (editions, selectedEdition)**

Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)

Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)**_multiVersionSupport = False****_vendor = 'Altera'**

```
class ToolChain.Alttera.ModelSim.AltteraStarterEditionConfiguration (host:
                                                                    Base.IHost)
```

Inheritance

Members

_toolName = 'Altera ModelSim (Starter Edition)'

The name of the tool.

__editionName = None

The name of the tool.

_section = 'INSTALL.Alttera.ModelSimASE'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'Linux': {'INSTALL.Alttera.ModelSimASE': {'Version': '10.5b', 'Edition': 'Starter Edition'}}**CheckDependency ()**

Check if general Altera support is configured in PoC.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForAll ()

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.

- Pass → skip this configuration. Don't change existing settings.
- Yes → collect installation information for ModelSim.

- No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
 2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
 3. Ask for ModelSim's installation directory.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions (writeWarnings=True)

PrepareSections (warningWasWritten, writeWarnings=True)

PrepareVersionedSections (writeWarnings=False)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_AlteraStarterEditionConfiguration__editionName = 'ModelSim Altera Starter Edition'

_Ask (*question*, *default*, *beforeDefault*=", *afterDefault*=", *indent*=1)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent*=1)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent*=1)

Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath*, *version*)

_Configuration__GetModelSimVersion (*binPath*)

_ConfigureBinaryDirectory ()

Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureEdition ()

Configure ModelSim for Altera.

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()

If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (*binPath*)

_PrintAvailableEditions (*editions*, *selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args*, *condition*=True, ***kwargs*)

_multiVersionSupport = False

_vendor = 'Altera'

ToolChain.Altera.Quartus

Exceptions

- *QuartusException*: Base class for all tool specific exceptions

Classes

- *QuartusEditions*: Enumeration of all Quartus editions provided by Altera itself.
- *Configuration*: Base class for all tool Configuration classes.
- *Quartus*: Undocumented.
- *Map*: Represent an executable.
- *TclShell*: Represent an executable.
- *QuartusSession*: Undocumented.
- *QuartusProject*: Undocumented.
- *QuartusSettings*: Undocumented.
- *QuartusProjectFile*: Undocumented.

Functions

- *MapFilter()*: Undocumented.

exception ToolChain.Altera.Quartus.**QuartusException** (*message=""*)

Inheritance

Members

__init__ (*message=""*)
Exception initializer

Parameters **message** (*str*) – The exception message.

__str__ ()
Returns the exception's message text.

args

class ToolChain.Altera.Quartus.**QuartusEditions** (*name, section*)
Enumeration of all Quartus editions provided by Altera itself.

Inheritance

Members

AlteraQuartus = 1

IntelQuartus = 2

class ToolChain.Altera.Quartus.**Configuration** (*host: Base.IHost*)

Inheritance

Members

_vendor = 'Altera'

The name of the tools vendor.

_toolName = 'Altera Quartus'

The name of the tool.

_section = 'INSTALL.Altera.Quartus'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_multiVersionSupport = True

Altera Quartus supports multiple versions installed on the same system.

_template = {'Linux': {'INSTALL.Altera.Quartus': {'Version': '16.0', 'SectionName': 'INSTALL.Altera.Quartus'}}

The template for the configuration sections represented as nested dictionaries.

CheckDependency ()

Check if general Altera support is configured in PoC.

ConfigureForAll ()

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)

Ask a YES/no/pass question.

_Configuration__CheckQuartusVersion (*binPath, version*)

_ConfigureBinaryDirectory ()

Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureEdition (*editions, defaultEdition*)

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()

If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

class ToolChain.Altera.Quartus.**Quartus** (*platform, dryrun, binaryDirectoryPath, version, logger=None*)

Inheritance

Members

GetMap ()

GetTclShell ()

class ToolChain.Altera.Quartus.**Map** (*toolchain: ToolChain.ToolMixIn*)

Inheritance

Members

class Executable

class ArgProjectName

class SwitchArgumentFile

_name = 'f'

class SwitchDeviceFamily

_name = 'family'

class SwitchDevicePart

_name = 'part'

Parameters = [**<class 'ToolChain.Altera.Quartus.Map.Executable'>**, **<class 'ToolChain.A**

Compile ()

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

```
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POC_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Altera.Quartus.TclShell (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
class SwitchShell

    _name = 's'
Parameters = [<class 'ToolChain.Altera.Quartus.TclShell.Executable'>, <class 'ToolChain.Altera.Quartus.TclShell.SwitchShell'>]
GetReader ()
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
```

```
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POC_BOUNDARY = '==== POC BOUNDARY ===='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Altera.Quartus.QuartusSession (host)
```

Inheritance

Members

```
exit ()
class ToolChain.Altera.Quartus.QuartusProject (host, name, projectFile=None)
```

Inheritance

Members

```
Create (session=None)
Save (session)
Read ()
Open (session)
Close (session)
AddExternalVHDLLibraries (library)
AddFile (file, fileSet=None)
AddFileSet (fileSet)
AddSourceFile (file, fileSet=None)
Board
CreateFileSet (name, setDefault=True)
DefaultFileSet
Device
Environment
ExternalVHDLLibraries
ExtractVHDLLibrariesFromVHDLSourceFiles ()
FileSets
Files (fileType=<FileTypes(Text\ProjectFile\FileListFile\RulesFile\SourceFile\VHDLSourceFile\VerilogSourceFile\Python\
bits=0xFFFF>, fileSet=None)
GetVariables ()
Name
RootDirectory
```

```

Tool
ToolChain
VHDLLibraries
VHDLVersion
pprint (indent=0)

```

```
class ToolChain.Altera.Quartus.QuartusSettings (name, settingsFile=None)
```

Inheritance

Members

```

File
GlobalAssignments
Parameters
CopySourceFilesFromProject (project)
Write ()
FileName
FileSet
FileType
Open ()
Path
Project
ReadFile ()
_FileType = <FileTypes.SettingsFile bits=0x4000 data=UNDEFINED>
_ReadContent ()

```

```
class ToolChain.Altera.Quartus.QuartusProjectFile (file)
```

Inheritance

Members

```

FileName
FileSet
FileType
Open ()
Path
Project
ReadFile ()
_FileType = <FileTypes.ProjectFile bits=0x0002 data=UNDEFINED>
_ReadContent ()

```

Functions

ToolChain.Alterta.Quartus.**MapFilter** (*gen*)

Exceptions

- *AltertaException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Configuration routines for Alterta as a vendor.

exception ToolChain.Alterta.**AltertaException** (*message=""*)

Inheritance

Members

__init__ (*message=""*)
Exception initializer

Parameters *message* (*str*) – The exception message.

__str__ ()
Returns the exception's message text.

args

class ToolChain.Alterta.**Configuration** (*host: Base.IHost*)
Configuration routines for Alterta as a vendor.

This configuration provides a common installation directory setup for all Alterta tools installed on a system.

Inheritance

Members

_vendor = 'Alterta'
The name of the tools vendor.

_section = 'INSTALL.Alterta'
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'ALL': {'INSTALL.Quartus': {'SectionName': '', 'Version': '\${\${Sect

The template for the configuration sections represented as nested dictionaries.

_GetDefaultInstallationDirectory ()
Return default installation directory (*str*).

CheckDependency ()
Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (*writeWarnings=False*)
Clear the configuration section associated to this Configuration class.

ConfigureForAll ()
Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

ConfigureForDarwin ()
Start the configuration procedure for Darwin.

This method is a wrapper for *ConfigureForAll* (). Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions (writeWarnings=True)

PrepareSections (warningWasWritten, writeWarnings=True)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)

_AskInstalled (question)

Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)

Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)

Ask a YES/no/pass question.

```
_ConfigureInstallationDirectory ()  
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path  
    object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.  
  
_GetDefaultOptionValue (optionName)  
  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
  
_TestDefaultInstallPath (defaults)  
    Helper function for automatic search of installation directory.  
  
_TryLog (*args, condition=True, **kwargs)  
  
_multiVersionSupport = False
```

14.7.3 ToolChain.GHDL

Exceptions

- *GHDLException*: Base class for all tool specific exceptions
- *GHDLReanalyzeException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *GHDL*: Represent an executable.
- *GHDLAnalyze*: Represent an executable.
- *GHDLElaborate*: Represent an executable.
- *GHDLRun*: Represent an executable.

Functions

- *GHDLAnalyzeFilter*(): Undocumented.
- *GHDLElaborateFilter*(): Undocumented.
- *GHDLRunFilter*(): Undocumented.

```
exception ToolChain.GHDL.GHDLException (message="")
```

Inheritance

Members

```
__init__ (message="")  
    Exception initializer  
  
    Parameters message (str) – The exception message.  
  
__str__ ()  
    Returns the exception's message text.  
  
args
```

```
exception ToolChain.GHDL.GHDLReanalyzeException (message="")
```

Inheritance

Members

`__init__(message=)`
Exception initializer

Parameters **message** (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

class `ToolChain.GHDL.Configuration` (*host: Base.IHost*)

Inheritance

Members

`_vendor = 'tgingold'`
The name of the tools vendor.

`_toolName = 'GHDL'`
The name of the tool.

`_section = 'INSTALL.GHDL'`
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`
Git supports multiple versions installed on the same system.

`_template = {'Darwin': {'INSTALL.GHDL': {'Version': '0.34-dev', 'Backend': 'llvm'}}`
The template for the configuration sections represented as nested dictionaries.

`ConfigureForAll()`
Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

`_GetDefaultInstallationDirectory()`
Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

`_ConfigureBinaryDirectory()`
Updates section with value from `_template` and returns directory as Path object.

`_ConfigureScriptDirectory()`
Updates section with value from `_template` and returns directory as Path object.

`CheckDependency()`
Check if all vendor or tool dependencies are fulfilled to configure this tool.

`ClearSection(writeWarnings=False)`
Clear the configuration section associated to this Configuration class.

`ConfigureForDarwin()`
Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for *ConfigureForAll ()*. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for *ConfigureForAll ()*. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****PrepareVersionedSections (writeWarnings=False)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)**_AskInstalled (question)**

Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)

Ask a yes/NO/pass question.

```
_Ask_YesNoPass (question, indent=1)  
    Ask a YES/no/pass question.  
  
_Configuration__WriteGHDLSection (binPath)  
  
_ConfigureEdition (editions, defaultEdition)  
  
_ConfigureInstallationDirectory ()  
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path  
    object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.  
  
_ConfigureVersion ()  
    If no version was configured before, then _GetDefaultVersion is called. Asks for version and updates  
    section. Returns version as string.  
  
_GetDefaultEdition ()  
    Returns unresolved default edition (str) from template.  
  
    Overwrite this method in a sub-class for automatic search of editions.  
  
_GetDefaultOptionValue (optionName)  
  
_GetDefaultVersion ()  
    Returns unresolved default version (str) from template.  
  
    Overwrite this method in a sub-class for automatic search of version.  
  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
  
_TestDefaultInstallPath (defaults)  
    Helper function for automatic search of installation directory.  
  
_TryLog (*args, condition=True, **kwargs)  
  
class ToolChain.GHDL.GHDL (platform, dryrun, binaryDirectoryPath, version, backend, log-  
                           ger=None)
```

Inheritance

Members

BinaryDirectoryPath

Backend

Version

deco ()

Executable

class CmdAnalyze

_name = 'a'

class CmdElaborate

_name = 'e'

class CmdRun

_name = 'r'

class FlagVerbose

```
    _name = 'v'
class FlagDebug

    _name = 'g'
class FlagExplicit

    _name = 'fexplicit'
class FlagRelaxedRules

    _name = 'frelaxed-rules'
class FlagWarnBinding

    _name = 'warn-binding'
class FlagNoVitalChecks

    _name = 'no-vital-checks'
class FlagMultiByteComments

    _name = 'mb-comments'
class FlagSynBinding

    _name = 'syn-binding'
class FlagPSL

    _name = 'fpsl'
class FlagProfileArcs

    _name = 'fprofile-arcs'
class FlagTestCoverage

    _name = 'ftest-coverage'
class SwitchCompilerOption

    _pattern = '-{0},{1}'
    _name = 'Wc'
class SwitchAssemblerOption

    _pattern = '-{0},{1}'
    _name = 'Wa'
class SwitchLinkerOption

    _pattern = '-{0},{1}'
    _name = 'Wl'
```

```
class SwitchIEEEFlavor

    _name = 'ieee'
class SwitchVHDLVersion

    _name = 'std'
class SwitchVHDLLibrary

    _name = 'work'
class ArgListLibraryReferences

    _pattern = '-{0}{1}'
    _name = 'P'
class ArgSourceFile
class ArgTopLevel
Parameters = [<class 'ToolChain.GHDL.GHDL.CmdAnalyze'>, <class 'ToolChain.GHDL.GHDL.CmdAnalyze'>]
class SwitchIEEEAsserts

    _name = 'ieee-asserts'
class SwitchStopDelta

    _name = 'stop-delta'
class SwitchVCDWaveform

    _name = 'vcd'
class SwitchVCDGZWaveform

    _name = 'vcdgz'
class SwitchFastWaveform

    _name = 'fst'
class SwitchGHDLWaveform

    _name = 'wave'
class SwitchWaveformOptionFile

    _name = 'read-wave-opt'
RunOptions = [<class 'ToolChain.GHDL.GHDL.SwitchIEEEAsserts'>, <class 'ToolChain.GHDL.GHDL.CmdAnalyze'>]
GetGHDLAnalyze()
GetGHDLElaborate()
GetGHDLRun()
GetReader()
```


HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

Path

ReadUntilBoundary (*indent=0*)

Send (*line*, *end='\n'*)

SendBoundary ()

StartProcess (*parameterList*)

Terminate ()

_POC_BOUNDARY = '===== **POC BOUNDARY** ====='

_TryLog (**args*, *condition=True*, ***kwargs*)

```
class ToolChain.GHDL.GHDLAnalyze (platform, dryrun, binaryDirectoryPath, version, backend,  
                                logger=None)
```

Inheritance**Members**

Analyze ()

class ArgListLibraryReferences

_name = 'P'

_pattern = '-{0}{1}'

class ArgSourceFile

class ArgTopLevel

Backend

BinaryDirectoryPath

```
class CmdAnalyze

    _name = 'a'
class CmdElaborate

    _name = 'e'
class CmdRun

    _name = 'r'
Executable
class FlagDebug

    _name = 'g'
class FlagExplicit

    _name = 'fexplicit'
class FlagMultiByteComments

    _name = 'mb-comments'
class FlagNoVitalChecks

    _name = 'no-vital-checks'
class FlagPSL

    _name = 'fpsl'
class FlagProfileArcs

    _name = 'fprofile-arcs'
class FlagRelaxedRules

    _name = 'frelaxed-rules'
class FlagSynBinding

    _name = 'syn-binding'
class FlagTestCoverage

    _name = 'ftest-coverage'
class FlagVerbose

    _name = 'v'
class FlagWarnBinding

    _name = 'warn-binding'
GetGHDLAnalyze()
```

GetGHDLElaborate ()

GetGHDLRun ()

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

Parameters = [<class 'ToolChain.GHDL.GHDL.CmdAnalyze'>, <class 'ToolChain.GHDL.GHDL

Path

ReadUntilBoundary (*indent=0*)

RunOptions = [<class 'ToolChain.GHDL.GHDL.SwitchIEEEAsserts'>, <class 'ToolChain.GHDL

Send (*line*, *end='\n'*)

SendBoundary ()

StartProcess (*parameterList*)

class SwitchAssemblerOption

_name = 'Wa'

_pattern = '-{0},{1}'

class SwitchCompilerOption

_name = 'Wc'

_pattern = '-{0},{1}'

class SwitchFastWaveform

_name = 'fst'

class SwitchGHDLWaveform

_name = 'wave'

```
class SwitchIEEEAsserts

    _name = 'ieee-asserts'
class SwitchIEEEFlavor

    _name = 'ieee'
class SwitchLinkerOption

    _name = 'Wl'
    _pattern = '-{0},{1}'
class SwitchStopDelta

    _name = 'stop-delta'
class SwitchVCDGZWaveform

    _name = 'vcdgz'
class SwitchVCDWaveform

    _name = 'vcd'
class SwitchVHDLLibrary

    _name = 'work'
class SwitchVHDLVersion

    _name = 'std'
class SwitchWaveformOptionFile

    _name = 'read-wave-opt'
Terminate()
Version
_POCKET_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog(*args, condition=True, **kwargs)
deco()
class ToolChain.GHDL.GHDLElaborate(platform, dryrun, binaryDirectoryPath, version, back-
                                     end, logger=None)
```

Inheritance

Members

```
Elaborate()
class ArgListLibraryReferences

    _name = 'P'
```

```
    _pattern = '-{0}{1}'
class ArgSourceFile
class ArgTopLevel
Backend
BinaryDirectoryPath
class CmdAnalyze

    _name = 'a'
class CmdElaborate

    _name = 'e'
class CmdRun

    _name = 'r'
Executable
class FlagDebug

    _name = 'g'
class FlagExplicit

    _name = 'fexplicit'
class FlagMultiByteComments

    _name = 'mb-comments'
class FlagNoVitalChecks

    _name = 'no-vital-checks'
class FlagPSL

    _name = 'fpsl'
class FlagProfileArcs

    _name = 'fprofile-arcs'
class FlagRelaxedRules

    _name = 'frelaxed-rules'
class FlagSynBinding

    _name = 'syn-binding'
class FlagTestCoverage

    _name = 'fctest-coverage'
```

```
class FlagVerbose

    __name__ = 'v'

class FlagWarnBinding

    __name__ = 'warn-binding'

GetGHDLAnalyze ()
GetGHDLLElaborate ()
GetGHDLRun ()
GetReader ()

HasErrors
    True if errors or fatals errors were found while processing the output stream.

HasWarnings
    True if errors or fatals errors were found while processing the output stream.

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Parameters = [<class 'ToolChain.GHDL.GHDL.CmdAnalyze'>, <class 'ToolChain.GHDL.GHDL.
Path

ReadUntilBoundary (indent=0)

RunOptions = [<class 'ToolChain.GHDL.GHDL.SwitchIEEEAsserts'>, <class 'ToolChain.GHDL.GHDL.
Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

class SwitchAssemblerOption

    __name__ = 'Wa'
    __pattern__ = '-{0},{1}'

class SwitchCompilerOption

    __name__ = 'Wc'
    __pattern__ = '-{0},{1}'
```

```
class SwitchFastWaveform

    _name = 'fst'
class SwitchGHDLWaveform

    _name = 'wave'
class SwitchIEEEAsserts

    _name = 'ieee-asserts'
class SwitchIEEEFlavor

    _name = 'ieee'
class SwitchLinkerOption

    _name = 'Wl'
    _pattern = '-{0},{1}'
class SwitchStopDelta

    _name = 'stop-delta'
class SwitchVCDGZWaveform

    _name = 'vcdgz'
class SwitchVCDWaveform

    _name = 'vcd'
class SwitchVHDLLibrary

    _name = 'work'
class SwitchVHDLVersion

    _name = 'std'
class SwitchWaveformOptionFile

    _name = 'read-wave-opt'
Terminate()
Version
_POCLIBRARY_BOUNDARY = '===== POC BOUNDARY ====='
_TryLog(*args, condition=True, **kwargs)
deco()
class ToolChain.GHDL.GHDLRun(platform, dryrun, binaryDirectoryPath, version, backend, logger=None)
```

Inheritance

Members

Run()

class ArgListLibraryReferences

 _name = 'P'

 _pattern = '-{0}{1}'

class ArgSourceFile

class ArgTopLevel

Backend

BinaryDirectoryPath

class CmdAnalyze

 _name = 'a'

class CmdElaborate

 _name = 'e'

class CmdRun

 _name = 'r'

Executable

class FlagDebug

 _name = 'g'

class FlagExplicit

 _name = 'fexplicit'

class FlagMultiByteComments

 _name = 'mb-comments'

class FlagNoVitalChecks

 _name = 'no-vital-checks'

class FlagPSL

 _name = 'fpsl'

class FlagProfileArcs

 _name = 'fprofile-arcs'

class FlagRelaxedRules


```

    _name = 'frelaxed-rules'
class FlagSynBinding

    _name = 'syn-binding'
class FlagTestCoverage

    _name = 'fctest-coverage'
class FlagVerbose

    _name = 'v'
class FlagWarnBinding

    _name = 'warn-binding'
GetGHDLAnalyze ()
GetGHDLElaborate ()
GetGHDLRun ()
GetReader ()
HasErrors
    True if errors or fatals errors were found while processing the output stream.
HasWarnings
    True if errors or fatals errors were found while processing the output stream.
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Parameters = [<class 'ToolChain.GHDL.GHDL.CmdAnalyze'>, <class 'ToolChain.GHDL.GHDL
Path
ReadUntilBoundary (indent=0)
RunOptions = [<class 'ToolChain.GHDL.GHDL.SwitchIEEEAsserts'>, <class 'ToolChain.GHDL
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)

```

```
class SwitchAssemblerOption

    _name = 'Wa'
    _pattern = '-{0},{1}'
class SwitchCompilerOption

    _name = 'Wc'
    _pattern = '-{0},{1}'
class SwitchFastWaveform

    _name = 'fst'
class SwitchGHDLWaveform

    _name = 'wave'
class SwitchIEEEAsserts

    _name = 'ieee-asserts'
class SwitchIEEEFlavor

    _name = 'ieee'
class SwitchLinkerOption

    _name = 'Wl'
    _pattern = '-{0},{1}'
class SwitchStopDelta

    _name = 'stop-delta'
class SwitchVCDGZWaveform

    _name = 'vcdgz'
class SwitchVCDWaveform

    _name = 'vcd'
class SwitchVHDLLibrary

    _name = 'work'
class SwitchVHDLVersion

    _name = 'std'
class SwitchWaveformOptionFile

    _name = 'read-wave-opt'
Terminate()
```

Version

```
_POC_BOUNDARY = '==== POC BOUNDARY ===='
_TryLog(*args, condition=True, **kwargs)
deco()
```

Functions

```
ToolChain.GHDL.GHDLAnalyzeFilter(gen)
ToolChain.GHDL.GHDLElaborateFilter(gen)
ToolChain.GHDL.GHDLRunFilter(gen)
```

14.7.4 ToolChain.GNU

Exceptions

- *GNUException*: Base class for all tool specific exceptions

Classes

- *Make*: Represent an executable.

Functions

- *GNUMakeQuestaSimFilter()*: Undocumented.
- *CocotbSimulationResultFilter()*: Undocumented.

```
exception ToolChain.GNU.GNUException(message="")
```

Inheritance**Members**

```
__init__(message="")
Exception initializer
```

Parameters *message* (*str*) – The exception message.

```
__str__()
Returns the exception's message text.
```

args

```
class ToolChain.GNU.Make(platform, dryrun, logger=None)
```

Inheritance**Members**

```
class Executable
class SwitchGui
```

```
__name__ = 'GUI'
```

```
Parameters = [<class 'ToolChain.GNU.Make.Executable'>, <class 'ToolChain.GNU.Make.S'
```

```
RunCocotb()
```

```
GetReader()
```

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)**LogDryRun** (**args*, *condition=True*, ***kwargs*)**LogError** (**args*, *condition=True*, ***kwargs*)**LogFatal** (**args*, *condition=True*, ***kwargs*)**LogInfo** (**args*, *condition=True*, ***kwargs*)**LogNormal** (**args*, *condition=True*, ***kwargs*)**LogQuiet** (**args*, *condition=True*, ***kwargs*)**LogVerbose** (**args*, *condition=True*, ***kwargs*)**LogWarning** (**args*, *condition=True*, ***kwargs*)**Logger**

Return the local logger instance.

Path**ReadUntilBoundary** (*indent=0*)**Send** (*line*, *end='\n'*)**SendBoundary** ()**StartProcess** (*parameterList*)**Terminate** ()

_POC_BOUNDARY = '===== **POC BOUNDARY** ====='

_TryLog (**args*, *condition=True*, ***kwargs*)**Functions**

ToolChain.GNU.GNUMakeQuestaSimFilter (*gen*)

ToolChain.GNU.CocotbSimulationResultFilter (*gen*, *simulationResult*)

14.7.5 ToolChain.GTKWave

Exceptions

- *GTKWaveException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *GTKWave*: Represent an executable.

Functions

- *GTKWaveFilter*(): Undocumented.

exception **ToolChain.GTKWave.GTKWaveException** (*message=""*)

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters **message** (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

class ToolChain.GTKWave.Configuration (*host: Base.IHost*)

Inheritance

Members

`_vendor = 'TonyBybell'`
The name of the tools vendor.

`_toolName = 'GTKWave'`
The name of the tool.

`_section = 'INSTALL.GTKWave'`
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`
GTKWave supports multiple versions installed on the same system.

`_template = {'Darwin': {'INSTALL.GTKWave': {'Version': '3.3.80', 'SectionName':`
The template for the configuration sections represented as nested dictionaries.

`CheckDependency()`
Check if all vendor or tool dependencies are fulfilled to configure this tool.

`ConfigureForAll()`
Start a generic (platform independent) configuration procedure.
Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

`_GetDefaultInstallationDirectory()`
Return unresolved default installation directory (str) from template.
Overwrite function in sub-class for automatic search of installation directory.

`ClearSection(writeWarnings=False)`
Clear the configuration section associated to this Configuration class.

`ConfigureForDarwin()`
Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux()`
Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****PrepareVersionedSections (writeWarnings=False)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)**_AskInstalled (question)**

Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)

Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)

Ask a YES/no/pass question.

_Configuration__WriteGtkWaveSection (binPath)**_ConfigureBinaryDirectory ()**

Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition (*editions, defaultEdition*)

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

```
class ToolChain.GTKWave.GTKWave (platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Inheritance

Members

BinaryDirectoryPath

Version

class Executable

class SwitchDumpFile

`__name__ = 'dump'`

class SwitchSaveFile

`__name__ = 'save'`

Parameters = [`<class 'ToolChain.GTKWave.GTKWave.Executable'>`, `<class 'ToolChain.GTKWave.GTKWave.SwitchDumpFile'>`, `<class 'ToolChain.GTKWave.GTKWave.SwitchSaveFile'>`]

View ()

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

```

LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POC_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)

```

Functions

```
ToolChain.GTKWave.GTKWaveFilter (gen)
```

14.7.6 ToolChain.Git

Exceptions

- *GitException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *Git*: Undocumented.
- *GitSCM*: Represent an executable.
- *GitRevParse*: Represent an executable.
- *GitRevList*: Represent an executable.
- *GitDescribe*: Represent an executable.
- *GitConfig*: Represent an executable.

```
exception ToolChain.Git.GitException (message="")
```

Inheritance

Members

```
__init__ (message="")
    Exception initializer
```


Parameters **message** (*str*) – The exception message.

__str__ ()

Returns the exception's message text.

args

class ToolChain.Git.Configuration (*host*)

Inheritance

Members

_vendor = 'Git SCM'

The name of the tools vendor.

_toolName = 'Git'

The name of the tool.

_section = 'INSTALL.Git'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'Linux': {'INSTALL.Git': {'Version': '2.8.1', 'InstallationDirectory': ''}}}

The template for the configuration sections represented as nested dictionaries.

ConfigureForAll ()

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

CheckDependency ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod **GetSections** (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten*, *writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question*, *default*, *beforeDefault="*, *afterDefault="*, *indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)
Ask a YES/no/pass question.

_Configuration__GetGitDirectory ()

_Configuration__InstallGitFilters ()

_Configuration__InstallGitHooks ()

_Configuration__IsUnderGitControl ()

_Configuration__UninstallGitFilters ()

_Configuration__UninstallGitHooks ()

_Configuration__WriteGitSection (*binPath*)

_ConfigureBinaryDirectory ()
Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureEdition (*editions*, *defaultEdition*)

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultOptionValue (optionName)**_GetDefaultVersion ()**

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (editions, selectedEdition)

Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)

Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)**_multiVersionSupport = False**

```
class ToolChain.Git.Git (platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Inheritance

Members

```
GetGitRevParse ()
```

```
GetGitRevList ()
```

```
GetGitDescribe ()
```

```
GetGitConfig ()
```

```
class ToolChain.Git.GitSCM (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
Clear ()
```

```
class Executable
```

```
class Switch_Version
```

```
    _name = 'version'
```

```
Parameters = [<class 'ToolChain.Git.GitSCM.Executable'>, <class 'ToolChain.Git.GitSCM.Switch_Version'>]
```

```
GetReader ()
```

```
Log (entry, condition=True)
```

Write an entry to the local logger.

```
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POCKET_BOUNDARY = '==== POC BOUNDARY ====='

_TryLog (*args, condition=True, **kwargs)

class ToolChain.Git.GitRevParse (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
Clear ()

class Command

    _name = 'rev-parse'

class SwitchInsideWorkingTree

    _name = 'is-inside-work-tree'

class SwitchShowTopLevel

    _name = 'show-toplevel'

class SwitchGitDir

    _name = 'git-dir'

RevParseParameters = [<class 'ToolChain.Git.GitRevParse.Command'>, <class 'ToolChain.Git.GitRevParse.Command'>]

Execute ()

class Executable

    GetReader ()
```

```
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Parameters = [<class 'ToolChain.Git.GitSCM.Executable'>, <class 'ToolChain.Git.GitSCM.Executable'>]
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
class Switch_Version
    _name = 'version'
Terminate ()
_POCC_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Git.GitRevList (toolchain: ToolChain.ToolMixin)
```

Inheritance

Members

```
Clear ()
class Command
    _name = 'rev-list'
class SwitchTags
    _name = 'tags'
class SwitchMaxCount
    _name = 'max-count'
RevListParameters = [<class 'ToolChain.Git.GitRevList.Command'>, <class 'ToolChain.Git.GitRevList.Command'>]
```

```
Execute ()

class Executable

GetReader ()

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Parameters = [<class 'ToolChain.Git.GitSCM.Executable'>, <class 'ToolChain.Git.GitSCM.Executable'>]

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

class Switch_Version

    _name = 'version'

Terminate ()

_POCKET_BOUNDARY = '==== POC BOUNDARY ====='

_TryLog (*args, condition=True, **kwargs)

class ToolChain.Git.GitDescribe (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
Clear ()

class Command

    _name = 'describe'

class SwitchAbbrev

    _name = 'abbrev'
```

```

class SwitchTags

    __name__ = 'tags'

    DescribeParameters = [<class 'ToolChain.Git.GitDescribe.Command'>, <class 'ToolChain.Git.GitDescribe.Command'>]

    Execute ()

    class Executable

    GetReader ()

    Log (entry, condition=True)
        Write an entry to the local logger.

    LogDebug (*args, condition=True, **kwargs)

    LogDryRun (*args, condition=True, **kwargs)

    LogError (*args, condition=True, **kwargs)

    LogFatal (*args, condition=True, **kwargs)

    LogInfo (*args, condition=True, **kwargs)

    LogNormal (*args, condition=True, **kwargs)

    LogQuiet (*args, condition=True, **kwargs)

    LogVerbose (*args, condition=True, **kwargs)

    LogWarning (*args, condition=True, **kwargs)

    Logger
        Return the local logger instance.

    Parameters = [<class 'ToolChain.Git.GitSCM.Executable'>, <class 'ToolChain.Git.GitSCM.Executable'>]

    Path

    ReadUntilBoundary (indent=0)

    Send (line, end='\n')

    SendBoundary ()

    StartProcess (parameterList)

    class Switch_Version

        __name__ = 'version'

    Terminate ()

    _POC_BOUNDARY = '==== POC BOUNDARY ===='

    _TryLog (*args, condition=True, **kwargs)

class ToolChain.Git.GitConfig (toolchain: ToolChain.ToolMixin)

```

Inheritance

Members

```

Clear ()

class Command

    __name__ = 'config'

```

```
class SwitchUnset

    _name = 'unset'
class SwitchRemoveSection

    _name = 'remove-section'
class ValueFilterClean

    _name = 'clean'
    _pattern = 'filter.{1}.{0}'
class ValueFilterSmudge

    _name = 'smudge'
    _pattern = 'filter.{1}.{0}'
class ValueFilterParameters
ConfigParameters = [<class 'ToolChain.Git.GitConfig.Command'>, <class 'ToolChain.GitS
Execute()
class Executable
GetReader()
Log(entry, condition=True)
    Write an entry to the local logger.
LogDebug(*args, condition=True, **kwargs)
LogDryRun(*args, condition=True, **kwargs)
LogError(*args, condition=True, **kwargs)
LogFatal(*args, condition=True, **kwargs)
LogInfo(*args, condition=True, **kwargs)
LogNormal(*args, condition=True, **kwargs)
LogQuiet(*args, condition=True, **kwargs)
LogVerbose(*args, condition=True, **kwargs)
LogWarning(*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Parameters = [<class 'ToolChain.Git.GitSCM.Executable'>, <class 'ToolChain.Git.GitS
Path
ReadUntilBoundary(indent=0)
Send(line, end='\n')
SendBoundary()
StartProcess(parameterList)
class Switch_Version

    _name = 'version'
```



```
Terminate()  
_POC_BOUNDARY = '==== POC BOUNDARY ====  
_TryLog(*args, condition=True, **kwargs)
```

14.7.7 ToolChain.Intel

Submodules

ToolChain.Intel.ModelSim

Exceptions

- *ModelSimException*: Base class for all tool specific exceptions

Classes

- *IntelModelSimEditions*: An enumeration.
- *Configuration*: Base class for all tool Configuration classes.
- *IntelEditionConfiguration*: Base class for all tool Configuration classes.
- *IntelStarterEditionConfiguration*: Base class for all tool Configuration classes.

exception ToolChain.Intel.ModelSim.**ModelSimException** (*message*=")

Inheritance

Members

__init__ (*message*=")
Exception initializer

Parameters *message* (*str*) – The exception message.

__str__ ()
Returns the exception's message text.

args

class ToolChain.Intel.ModelSim.**IntelModelSimEditions** (*name*, *section*)
An enumeration.

Inheritance

Members

ModelSimIntelEdition = 1

ModelSimIntelStarterEdition = 2

class ToolChain.Intel.ModelSim.**Configuration** (*host*: *Base.IHost*)

Inheritance

Members

_vendor = 'Intel'

The name of the tools vendor.

_multiVersionSupport = False

Intel ModelSim Edition doesn't support multiple versions.

CheckDependency ()

Check if general Intel support is configured in PoC.

ConfigureForAll ()

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

_ConfigureEdition ()

Configure ModelSim for Intel.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten*, *writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question*, *default*, *beforeDefault="*, *afterDefault="*, *indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)

Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath*, *version*)

_Configuration__GetModelSimVersion (*binPath*)

_ConfigureBinaryDirectory ()

Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()

If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

```
_GetDefaultInstallationDirectory ()  
    Return unresolved default installation directory (str) from template.  
  
    Overwrite function in sub-class for automatic search of installation directory.  
  
_GetDefaultOptionValue (optionName)  
  
_GetDefaultVersion ()  
    Returns unresolved default version (str) from template.  
  
    Overwrite this method in a sub-class for automatic search of version.  
  
_GetModelSimVersion (binPath)  
  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
  
_TestDefaultInstallPath (defaults)  
    Helper function for automatic search of installation directory.  
  
_TryLog (*args, condition=True, **kwargs)  
  
_section = 'INSTALL.Vendor.Tool'  
  
_template = {'ALL': {'INSTALL.Vendor.Tool': {'Version': '1.0'}}, 'Darwin': {'INS'  
  
_toolName = 'Mentor ModelSim'
```

```
class ToolChain.Intel.ModelSim.IntelEditionConfiguration (host: Base.IHost)
```

Inheritance

Members

```
_toolName = 'Intel ModelSim'  
    The name of the tool.  
  
__editionName = None  
    The name of the tool.  
  
_section = 'INSTALL.Intel.ModelSimAE'  
    The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.  
  
_template = {'Linux': {'INSTALL.Intel.ModelSimAE': {'Version': '10.5b', 'Edition'  
  
CheckDependency ()  
    Check if general Intel support is configured in PoC.  
  
ClearSection (writeWarnings=False)  
    Clear the configuration section associated to this Configuration class.  
  
ConfigureForAll ()  
    Configuration routine for Mentor Graphics ModelSim on all supported platforms.  
    1. Ask if ModelSim is installed.  
  
        • Pass → skip this configuration. Don't change existing settings.  
        • Yes → collect installation information for ModelSim.  
        • No → clear the ModelSim configuration section.  
  
    1. Ask for ModelSim's version.  
    2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).  
    3. Ask for ModelSim's installation directory.
```

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****PrepareVersionedSections (writeWarnings=False)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)

```
_AskInstalled (question)
    Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)
    Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)
    Ask a YES/no/pass question.

_CheckModelSimVersion (binPath, version)

_Configuration__GetModelSimVersion (binPath)

_ConfigureBinaryDirectory ()
    Updates section with value from _template and returns directory as Path object.

_ConfigureEdition ()
    Configure ModelSim for Intel.

_ConfigureInstallationDirectory ()
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path
    object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.

_ConfigureVersion ()
    If no version was configured before, then _GetDefaultVersion is called. Asks for version and updates
    section. Returns version as string.

_GetDefaultEdition ()
    Returns unresolved default edition (str) from template.

    Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)

_GetDefaultVersion ()
    Returns unresolved default version (str) from template.

    Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (binPath)

_IntelEditionConfiguration__editionName = 'ModelSim Intel Edition'

_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

_multiVersionSupport = False

_vendor = 'Intel'

class ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration (host:
                                                                Base.IHost)
```

Inheritance

Members

_toolName = 'Intel ModelSim (Starter Edition)'

The name of the tool.

__editionName = None

_section = 'INSTALL.Intel.ModelSimASE'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'Linux': {'INSTALL.Intel.ModelSimASE': {'Version': '10.5b', 'Edition

CheckDependency ()

Check if general Intel support is configured in PoC.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForAll ()

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath, version*)

_Configuration__GetModelSimVersion (*binPath*)

_ConfigureBinaryDirectory ()
Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureEdition ()
Configure ModelSim for Intel.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()
If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.


```
_GetDefaultOptionValue (optionName)
_GetDefaultVersion ()
    Returns unresolved default version (str) from template.
    Overwrite this method in a sub-class for automatic search of version.
_GetModelSimVersion (binPath)
_IntelStarterEditionConfiguration__editionName = 'ModelSim Intel Starter Edition'
_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.
_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.
_TryLog (*args, condition=True, **kwargs)
_multiVersionSupport = False
_vendor = 'Intel'
```

ToolChain.Intel.Quartus

Exceptions

- *QuartusException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *Quartus*: Undocumented.
- *Map*: Represent an executable.

```
exception ToolChain.Intel.Quartus.QuartusException (message="")
```

Inheritance

Members

```
__init__ (message="")
    Exception initializer

    Parameters message (str) – The exception message.

__str__ ()
    Returns the exception's message text.

args
```

```
class ToolChain.Intel.Quartus.Configuration (host: Base.IHost)
```

Inheritance

Members

```
_vendor = 'Intel'
    The name of the tools vendor.

_toolName = 'Intel Quartus Prime'
    The name of the tool.
```

`_section = 'INSTALL.Intel.Quartus'`

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`

Intel Quartus supports multiple versions installed on the same system.

`CheckDependency ()`

Check if general Intel support is configured in PoC.

`ConfigureForAll ()`

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

`ClearSection (writeWarnings=False)`

Clear the configuration section associated to this Configuration class.

`ConfigureForDarwin ()`

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux ()`

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

`ConfigureForWindows ()`

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

`classmethod GetSections (platform)`

Return all section names for this configuration.

`Host`

Return the hosting object.

`IsConfigured ()`

Return true if the configurations section is configured

`IsSupportedPlatform ()`

Return true if the given platform is supported by this configuration routine.

`Log (entry, condition=True)`

Write an entry to the local logger.

`LogDebug (*args, condition=True, **kwargs)`

`LogDryRun (*args, condition=True, **kwargs)`

`LogError (*args, condition=True, **kwargs)`

`LogFatal (*args, condition=True, **kwargs)`

`LogInfo (*args, condition=True, **kwargs)`

`LogNormal (*args, condition=True, **kwargs)`

`LogQuiet (*args, condition=True, **kwargs)`

`LogVerbose (*args, condition=True, **kwargs)`

`LogWarning (*args, condition=True, **kwargs)`

`Logger`

Return the local logger instance.

`PrepareOptions (writeWarnings=True)`

```

PrepareSections (warningWasWritten, writeWarnings=True)
PrepareVersionedSections (writeWarnings=False)
RunPostConfigurationTasks ()
    Virtual method. Overwrite to execute post-configuration tasks.
SectionName
    Return the configuration's section name.
State
    Return the configuration state.
_Ask (question, default, beforeDefault= ", afterDefault= ", indent=1)
_AskInstalled (question)
    Ask a Yes/No/Pass question.
_AskYes_NoPass (question, indent=1)
    Ask a yes/NO/pass question.
_Ask_YesNoPass (question, indent=1)
    Ask a YES/no/pass question.
_Configuration__CheckQuartusVersion (binPath, version)
_ConfigureBinaryDirectory ()
    Updates section with value from _template and returns directory as Path object.
_ConfigureEdition (editions, defaultEdition)
_ConfigureInstallationDirectory ()
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.
_ConfigureVersion ()
    If no version was configured before, then _GetDefaultVersion is called. Asks for version and updates section. Returns version as string.
_GetDefaultEdition ()
    Returns unresolved default edition (str) from template.

    Overwrite this method in a sub-class for automatic search of editions.
_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.
_GetDefaultOptionValue (optionName)
_GetDefaultVersion ()
    Returns unresolved default version (str) from template.

    Overwrite this method in a sub-class for automatic search of version.
_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.
_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.
_TryLog (*args, condition=True, **kwargs)
_template = {'Linux': {'INSTALL.Alterta.Quartus': {'Version': '16.0', 'SectionName'
class ToolChain.Intel.Quartus.Quartus (platform, dryrun, binaryDirectoryPath, version,
                                         logger=None)

```

Inheritance

Members

GetMap ()

GetTclShell ()

```
class ToolChain.Intel.Quartus.Map (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class ArgProjectName
```

Compile ()

```
class Executable
```

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

```
Parameters = [<class 'ToolChain.Altera.Quartus.Map.Executable'>, <class 'ToolChain.2
```

Path

ReadUntilBoundary (*indent=0*)

Send (*line, end='\n'*)

SendBoundary ()

StartProcess (*parameterList*)

```
class SwitchArgumentFile
```

```
    _name = 'f'
```

```

class SwitchDeviceFamily

    _name = 'family'

class SwitchDevicePart

    _name = 'part'

    Terminate()

    _POC_BOUNDARY = '==== POC BOUNDARY ===='

    _TryLog(*args, condition=True, **kwargs)

```

Exceptions

- *IntelException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Configuration routines for Intel as a vendor.

```
exception ToolChain.Intel.IntelException (message="")
```

Inheritance

Members

```

__init__(message="")
    Exception initializer

    Parameters message (str) – The exception message.

__str__()
    Returns the exception's message text.

args

```

```
class ToolChain.Intel.Configuration (host: Base.IHost)
    Configuration routines for Intel as a vendor.
```

This configuration provides a common installation directory setup for all Intel tools installed on a system.

Inheritance

Members

```

_vendor = 'Intel'
    The name of the tools vendor.

_section = 'INSTALL.Intel'
    The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.

_template = {'Linux': {'INSTALL.Intel': {'InstallationDirectory': '/opt/IntelFPGA'}}
    The template for the configuration sections represented as nested dictionaries.

_GetDefaultInstallationDirectory()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.

CheckDependency()
    Check if all vendor or tool dependencies are fulfilled to configure this tool.

```

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForAll ()

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

`_Ask` (*question*, *default*, *beforeDefault*="", *afterDefault*="", *indent*=1)

`_AskInstalled` (*question*)

Ask a Yes/No/Pass question.

`_AskYes_NoPass` (*question*, *indent*=1)

Ask a yes/NO/pass question.

`_Ask_YesNoPass` (*question*, *indent*=1)

Ask a YES/no/pass question.

`_ConfigureInstallationDirectory` ()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

`_GetDefaultOptionValue` (*optionName*)

`_PrintAvailableEditions` (*editions*, *selectedEdition*)

Print all available editions and return the selected index.

`_TestDefaultInstallPath` (*defaults*)

Helper function for automatic search of installation directory.

`_TryLog` (*args, *condition*=True, **kwargs)

`_multiVersionSupport` = False

14.7.8 ToolChain.Lattice

Submodules

ToolChain.Lattice.ActiveHDL

ToolChain.Lattice.Diamond

Exceptions

- *DiamondException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *Diamond*: Undocumented.
- *Synth*: Represent an executable.
- *SynthesisArgumentFile*: Undocumented.

Functions

- *MapFilter*(): Undocumented.
- *CompilerFilter*(): Undocumented.

exception ToolChain.Lattice.Diamond.**DiamondException** (*message*="")

Inheritance

Members

__init__ (*message*=")
Exception initializer

Parameters *message* (*str*) – The exception message.

__str__ ()
Returns the exception's message text.

args

class ToolChain.Lattice.Diamond.Configuration (*host*: Base.IHost)

Inheritance

Members

_vendor = 'Lattice'
The name of the tools vendor.

_toolName = 'Lattice Diamond'
The name of the tool.

_section = 'INSTALL.Lattice.Diamond'
The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.

_multiVersionSupport = True
Lattice Diamond supports multiple versions installed on the same system.

_template = {'Linux': {'INSTALL.Lattice.Diamond': {'Version': '3.8', 'SectionName': ...}}
The template for the configuration sections represented as nested dictionaries.

CheckDependency ()
Check if general Lattice support is configured in PoC.

ConfigureForAll ()
Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

_ConfigureBinaryDirectory ()
Updates section with value from _template and returns directory as Path object.

ClearSection (*writeWarnings=False*)
Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()
Start the configuration procedure for Darwin.

This method is a wrapper for [ConfigureForAll\(\)](#). Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()
Start the configuration procedure for Linux.

This method is a wrapper for [ConfigureForAll\(\)](#). Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()
Start the configuration procedure for Windows.

This method is a wrapper for [ConfigureForAll\(\)](#). Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)
Return all section names for this configuration.

Host
Return the hosting object.

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)
Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault=", afterDefault=", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_Configuration__CheckDiamondVersion (*binPath, version*)

_ConfigureEdition (*editions, defaultEdition*)

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)**_GetDefaultVersion ()**

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (editions, selectedEdition)

Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)

Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

```
class ToolChain.Lattice.Diamond.Diamond(platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Inheritance

Members

PreparseEnvironment (installationDirectory)

GetSynthesizer ()

```
class ToolChain.Lattice.Diamond.Synth(toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
```

```
class SwitchProjectFile
```

```
    _name = 'f'
```

```
    _value = None
```

```
Parameters = [<class 'ToolChain.Lattice.Diamond.Synth.Executable'>, <class 'ToolChain.Lattice.Diamond.Synth.SwitchProjectFile'>]
```

```
static GetLogFileReader()
```

```
Compile(logFile)
```

```
GetReader()
```

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

Path

ReadUntilBoundary (*indent=0*)

Send (*line*, *end='\n'*)

SendBoundary ()

StartProcess (*parameterList*)

Terminate ()

_POC_BOUNDARY = '===== **POC BOUNDARY** ====='

_TryLog (**args*, *condition=True*, ***kwargs*)

class ToolChain.Lattice.Diamond.SynthesisArgumentFile (*file*)

Inheritance**Members**

Architecture

Device

SpeedGrade

Package

TopLevel

FileName

FileSet

FileType

LogFile

Open ()

Path

Project

```

ReadFile ()
_FileType = <FileTypes() bits=0x0000>
_ReadContent ()
VHDLVersion
HDLParams
Write (project)

```

Functions

```

ToolChain.Lattice.Diamond.MapFilter (gen)
ToolChain.Lattice.Diamond.CompilerFilter (gen)

```

ToolChain.Lattice.Synplify

Exceptions

- *SynplifyException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.

```
exception ToolChain.Lattice.Synplify.SynplifyException (message="")
```

Inheritance

Members

```

__init__ (message="")
    Exception initializer

    Parameters message (str) – The exception message.

```

```

__str__ ()
    Returns the exception's message text.

```

args

```
class ToolChain.Lattice.Synplify.Configuration (host)
```

Inheritance

Members

```
CheckDependency ()
    Check if all vendor or tool dependencies are fulfilled to configure this tool.
```

```
ClearSection (writeWarnings=False)
    Clear the configuration section associated to this Configuration class.
```

```
ConfigureForAll ()
    Start a generic (platform independent) configuration procedure.

    Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.
```

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****PrepareVersionedSections (writeWarnings=False)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)

```

_AskInstalled (question)
    Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)
    Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)
    Ask a YES/no/pass question.

_ConfigureBinaryDirectory ()
    Updates section with value from _template and returns directory as Path object.

_ConfigureEdition (editions, defaultEdition)

_ConfigureInstallationDirectory ()
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.

_ConfigureVersion ()
    If no version was configured before, then _GetDefaultVersion is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
    Returns unresolved default edition (str) from template.

    Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)

_GetDefaultVersion ()
    Returns unresolved default version (str) from template.

    Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

_multiVersionSupport = False

_section = 'INSTALL.Vendor.Tool'

_template = {'ALL': {'INSTALL.Vendor.Tool': {'Version': '1.0'}}, 'Darwin': {'INS

_toolName = 'Tool'

_vendor = 'Unknown'

```

Exceptions

- *LatticeException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Configuration routines for Lattice as a vendor.
- *LatticeDesignConstraintFile*: Undocumented.

exception *ToolChain.Lattice.LatticeException* (*message=""*)

Inheritance

Members

`__init__(message=)`
Exception initializer

Parameters **message** (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

class `ToolChain.Lattice.Configuration` (*host: Base.IHost*)

Configuration routines for Lattice as a vendor.

This configuration provides a common installation directory setup for all Lattice tools installed on a system.

Inheritance

Members

`_vendor = 'Lattice'`
The name of the tools vendor.

`_section = 'INSTALL.Lattice'`
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_template = {'Linux': {'INSTALL.Lattice': {'InstallationDirectory': '/usr/local'}}`
The template for the configuration sections represented as nested dictionaries.

`_GetDefaultInstallationDirectory()`
Return unresolved default installation directory (*str*) from template.

Overwrite function in sub-class for automatic search of installation directory.

`CheckDependency()`
Check if all vendor or tool dependencies are fulfilled to configure this tool.

`ClearSection(writeWarnings=False)`
Clear the configuration section associated to this Configuration class.

`ConfigureForAll()`
Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

`ConfigureForDarwin()`
Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux()`
Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

`ConfigureForWindows()`
Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)
Return all section names for this configuration.

Host
Return the hosting object.

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)
Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault=", afterDefault=", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_GetDefaultOptionValue (*optionName*)

_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)
Helper function for automatic search of installation directory.


```
_TryLog (*args, condition=True, **kwargs)

_multiVersionSupport = False

class ToolChain.Lattice.LatticeDesignConstraintFile (file, project=None, file-
Set=None)
```

Inheritance

Members

```
_FileType = <FileTypes.LdcConstraintFile bits=0x2000 data=UNDEFINED>

FileName

FileSet

FileType

Open ()

Path

Project

ReadFile ()

_ReadContent ()
```

14.7.9 ToolChain.Mentor

Submodules

ToolChain.Mentor.ModelSim

Exceptions

- *ModelSimException*: Base class for all tool specific exceptions

Classes

- *MentorModelSimPEEditions*: Enumeration of all ModelSim editions provided by Mentor Graphics itself.
- *ModelSimEditions*: Enumeration of all ModelSim editions provided by Mentor Graphics inclusive
- *Configuration*: Base class for all tool Configuration classes.
- *ModelSimPEConfiguration*: Base class for all tool Configuration classes.
- *ModelSimSE32Configuration*: Base class for all tool Configuration classes.
- *ModelSimSE64Configuration*: Base class for all tool Configuration classes.
- *Selector*: Base class for all Selector classes.
- *ModelSim*: Undocumented.
- *VHDDLlibraryTool*: Represent an executable.
- *VHDLCompiler*: Represent an executable.
- *VHDL Simulator*: Represent an executable.

Functions

- *VLibFilter*(): Undocumented.

- `VComFilter()`: Undocumented.
- `VSIMFilter()`: Undocumented.

exception `ToolChain.Mentor.ModelSim.ModelSimException` (*message=""*)

Inheritance

Members

`__init__` (*message=""*)
Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__` ()
Returns the exception's message text.

args

class `ToolChain.Mentor.ModelSim.MentorModelSimPEEditions` (*name, section*)
Enumeration of all ModelSim editions provided by Mentor Graphics itself.

Inheritance

Members

`ModelSimPE` = 1

`ModelSimPEEducation` = 2

class `ToolChain.Mentor.ModelSim.ModelSimEditions` (*name, section*)
Enumeration of all ModelSim editions provided by Mentor Graphics inclusive editions shipped by other vendors.

Inheritance

Members

`ModelSimPE` = 1

`ModelSimDE` = 2

`ModelSimSE32` = 3

`ModelSimSE64` = 4

`ModelSimAlteraEdition` = 5

`ModelSimAlteraStarterEdition` = 6

`ModelSimIntelEdition` = 7

`ModelSimIntelStarterEdition` = 8

`QuestaSim` = 9

class `ToolChain.Mentor.ModelSim.Configuration` (*host: Base.IHost*)

Inheritance

Members

_vendor = 'Mentor'

The name of the tools vendor.

_toolName = 'Mentor ModelSim'

The name of the tool.

_multiVersionSupport = True

Mentor ModelSim supports multiple versions installed on the same system.

CheckDependency ()

Check if general Mentor Graphics support is configured in PoC.

ConfigureForAll ()

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

_GetModelSimVersion (binPath)

_CheckModelSimVersion (binPath, version)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for [ConfigureForAll \(\)](#). Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for [ConfigureForAll \(\)](#). Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for [ConfigureForAll \(\)](#). Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry*, *condition=True*)
Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten*, *writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question*, *default*, *beforeDefault="*, *afterDefault="*, *indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)
Ask a YES/no/pass question.

_ConfigureBinaryDirectory ()
Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition (*editions*, *defaultEdition*)

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()
If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

```

_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)

_GetDefaultVersion ()
    Returns unresolved default version (str) from template.

    Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

_section = 'INSTALL.Vendor.Tool'

_template = {'ALL': {'INSTALL.Vendor.Tool': {'Version': '1.0'}}, 'Darwin': {'INS

class ToolChain.Mentor.ModelSim.ModelSimPEConfiguration (host: Base.IHost)

```

Inheritance

Members

```

_toolName = 'Mentor ModelSim PE'
    The name of the tool.

_section = 'INSTALL.Mentor.ModelSimPE'
    The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.

_template = {'Windows': {'INSTALL.Mentor.ModelSimPE': {'Version': '10.5c', 'Secti
    The template for the configuration sections represented as nested dictionaries.

_ConfigureEdition ()
    Configure ModelSim PE for Mentor Graphics.

CheckDependency ()
    Check if general Mentor Graphics support is configured in PoC.

ClearSection (writeWarnings=False)
    Clear the configuration section associated to this Configuration class.

ConfigureForAll ()
    Configuration routine for Mentor Graphics ModelSim on all supported platforms.
    1. Ask if ModelSim is installed.

    • Pass → skip this configuration. Don't change existing settings.
    • Yes → collect installation information for ModelSim.
    • No → clear the ModelSim configuration section.

    1. Ask for ModelSim's version.
    2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
    3. Ask for ModelSim's installation directory.

```

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions (writeWarnings=True)

PrepareSections (warningWasWritten, writeWarnings=True)

PrepareVersionedSections (writeWarnings=False)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)

```

_AskInstalled (question)
    Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)
    Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)
    Ask a YES/no/pass question.

_CheckModelSimVersion (binPath, version)

_ConfigureBinaryDirectory ()
    Updates section with value from _template and returns directory as Path object.

_ConfigureInstallationDirectory ()
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.

_ConfigureVersion ()
    If no version was configured before, then _GetDefaultVersion is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
    Returns unresolved default edition (str) from template.

    Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.

    Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)

_GetDefaultVersion ()
    Returns unresolved default version (str) from template.

    Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (binPath)

_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

_multiVersionSupport = True

_vendor = 'Mentor'

```

```

class ToolChain.Mentor.ModelSim.ModelSimSE32Configuration (host: Base.IHost)

```

Inheritance

Members

```

_toolName = 'Mentor ModelSim SE 32-bit'
    The name of the tool.

_section = 'INSTALL.Mentor.ModelSimSE32'
    The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.

_template = { 'Windows': { 'INSTALL.Mentor.ModelSimSE32': { 'Version': '10.5c', 'Se
    The template for the configuration sections represented as nested dictionaries.

```

`_ConfigureEdition ()`

`CheckDependency ()`

Check if general Mentor Graphics support is configured in PoC.

`ClearSection (writeWarnings=False)`

Clear the configuration section associated to this Configuration class.

`ConfigureForAll ()`

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

`ConfigureForDarwin ()`

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux ()`

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

`ConfigureForWindows ()`

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

`classmethod GetSections (platform)`

Return all section names for this configuration.

`Host`

Return the hosting object.

`IsConfigured ()`

Return true if the configurations section is configured

`IsSupportedPlatform ()`

Return true if the given platform is supported by this configuration routine.

`Log (entry, condition=True)`

Write an entry to the local logger.

`LogDebug (*args, condition=True, **kwargs)`

`LogDryRun (*args, condition=True, **kwargs)`

`LogError (*args, condition=True, **kwargs)`

`LogFatal (*args, condition=True, **kwargs)`

`LogInfo (*args, condition=True, **kwargs)`

`LogNormal (*args, condition=True, **kwargs)`

`LogQuiet (*args, condition=True, **kwargs)`

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question, default, beforeDefault= "", afterDefault= "", indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)

Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath, version*)

_ConfigureBinaryDirectory ()

Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()

If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (*binPath*)

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

```
_multiVersionSupport = True
_vendor = 'Mentor'
class ToolChain.Mentor.ModelSim.ModelSimSE64Configuration (host: Base.IHost)
```

Inheritance

Members

```
_toolName = 'Mentor ModelSim SE 64-bit'
```

The name of the tool.

```
_section = 'INSTALL.Mentor.ModelSimSE64'
```

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

```
_template = {'Linux': {'INSTALL.Mentor.ModelSimSE64': {'Version': '10.5c', 'Sect:
```

The template for the configuration sections represented as nested dictionaries.

```
_ConfigureEdition ()
```

```
CheckDependency ()
```

Check if general Mentor Graphics support is configured in PoC.

```
ClearSection (writeWarnings=False)
```

Clear the configuration section associated to this Configuration class.

```
ConfigureForAll ()
```

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

```
ConfigureForDarwin ()
```

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

```
ConfigureForLinux ()
```

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

```
ConfigureForWindows ()
```

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

```
classmethod GetSections (platform)
```

Return all section names for this configuration.

```
Host
```

Return the hosting object.

IsConfigured ()
Return true if the configurations section is configured

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)
Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault=" ", afterDefault=" ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_CheckModelSimVersion (*binPath, version*)

_ConfigureBinaryDirectory ()
Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()
If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

```
_GetDefaultEdition ()  
    Returns unresolved default edition (str) from template.  
  
    Overwrite this method in a sub-class for automatic search of editions.  
  
_GetDefaultInstallationDirectory ()  
    Return unresolved default installation directory (str) from template.  
  
    Overwrite function in sub-class for automatic search of installation directory.  
  
_GetDefaultOptionValue (optionName)  
  
_GetDefaultVersion ()  
    Returns unresolved default version (str) from template.  
  
    Overwrite this method in a sub-class for automatic search of version.  
  
_GetModelSimVersion (binPath)  
  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
  
_TestDefaultInstallPath (defaults)  
    Helper function for automatic search of installation directory.  
  
_TryLog (*args, condition=True, **kwargs)  
  
_multiVersionSupport = True  
  
_vendor = 'Mentor'
```

```
class ToolChain.Mentor.ModelSim.Selector (host: Base.IHost)
```

Inheritance

Members

```
_toolName = 'ModelSim'  
  
Select ()  
  
Log (entry, condition=True)  
    Write an entry to the local logger.  
  
LogDebug (*args, condition=True, **kwargs)  
  
LogDryRun (*args, condition=True, **kwargs)  
  
LogError (*args, condition=True, **kwargs)  
  
LogFatal (*args, condition=True, **kwargs)  
  
LogInfo (*args, condition=True, **kwargs)  
  
LogNormal (*args, condition=True, **kwargs)  
  
LogQuiet (*args, condition=True, **kwargs)  
  
LogVerbose (*args, condition=True, **kwargs)  
  
LogWarning (*args, condition=True, **kwargs)  
  
Logger  
    Return the local logger instance.  
  
ToolName  
  
_Ask (question, default, beforeDefault=", afterDefault=", indent=1)  
  
_AskSelection (editions, defaultEdition)
```

```
_AskYes_NoPass (question, indent=1)  
    Ask a yes/NO/pass question.  
_Ask_YesNoPass (question, indent=1)  
    Ask a YES/no/pass question.  
_GetConfiguredEditions (editions)  
    Return all configured editions.  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
_TryLog (*args, condition=True, **kwargs)  
class ToolChain.Mentor.ModelSim.ModelSim (platform, dryrun, binaryDirectoryPath, ver-  
                                           sion, logger=None)
```

Inheritance

Members

```
GetVHDLLibraryTool ()  
GetVHDLCompiler ()  
GetSimulator ()  
class ToolChain.Mentor.ModelSim.VHDLLibraryTool (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable  
class SwitchLibraryName  
Parameters = [<class 'ToolChain.Mentor.ModelSim.VHDLLibraryTool.Executable'>, <class ...  
CreateLibrary ()  
GetReader ()  
HasErrors  
    True if errors or fatals errors were found while processing the output stream.  
HasWarnings  
    True if errors or fatals errors were found while processing the output stream.  
Log (entry, condition=True)  
    Write an entry to the local logger.  
LogDebug (*args, condition=True, **kwargs)  
LogDryRun (*args, condition=True, **kwargs)  
LogError (*args, condition=True, **kwargs)  
LogFatal (*args, condition=True, **kwargs)  
LogInfo (*args, condition=True, **kwargs)  
LogNormal (*args, condition=True, **kwargs)  
LogQuiet (*args, condition=True, **kwargs)  
LogVerbose (*args, condition=True, **kwargs)
```

```
LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POCKET_BOUNDARY = '==== POC BOUNDARY ====='

_TryLog (*args, condition=True, **kwargs)
```

```
class ToolChain.Mentor.ModelSim.VHDLCompiler (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable

    _value = None

class FlagTime

    _name = 'time'
    _value = None

class FlagExplicit

    _name = 'explicit'
    _value = None

class FlagQuietMode

    _name = 'quiet'
    _value = None

class SwitchModelSimIniFile

    _name = 'modelsimini'
    _value = None

class FlagRangeCheck

    _name = 'rangecheck'
    _value = None

class SwitchCoverage

    _name = 'cover'
```

```
class FlagEnableFocusedExpressionCoverage

    _name = 'coverfec'
class FlagDisableFocusedExpressionCoverage

    _name = 'nocoverfec'
class FlagEnableRapidExpressionCoverage

    _name = 'coverrec'
class FlagDisableRapidExpressionCoverage

    _name = 'nocoverrec'
class FlagEnableRecognitionOfImplicitFSMResetTransitions

    _name = 'fsmresettrans'
class FlagDisableRecognitionOfImplicitFSMResetTransitions

    _name = 'nofsmresettrans'
class FlagEnableRecognitionOfSingleBitFSMState

    _name = 'fsmsingle'
class FlagDisableRecognitionOfSingleBitFSMState

    _name = 'nofsmsingle'
class FlagEnableRecognitionOfImplicitFSMTransitions

    _name = 'fsmimplicittrans'
class FlagDisableRecognitionOfImplicitFSMTransitions

    _name = 'nofsmimplicittrans'
class SwitchFSMVerbosityLevel

    _name = 'fsmverbose'
class FlagReportAsNote

    _name = 'note'
    _value = None
class FlagReportAsError

    _name = 'error'
    _value = None
class FlagReportAsWarning

    _name = 'warning'
```

```
    _value = None
class FlagReportAsFatal

    _name = 'fatal'
    _value = None
class FlagRelaxLanguageChecks

    _name = 'permissive'
class FlagForceLanguageChecks

    _name = 'pedanticerrors'
class SwitchVHDLVersion

    _pattern = '-{0}'
    _value = None
class ArgLogFile

    _name = 'l'
    _value = None
class SwitchVHDLLibrary

    _name = 'work'
    _value = None
class ArgSourceFile

    _value = None
Parameters = [<class 'ToolChain.Mentor.ModelSim.VHDLCompiler.Executable'>, <class 'ToolChain.Mentor.ModelSim.VHDLCompiler.Executable'>]
Compile()
GetTclCommand()
GetReader()
HasErrors
    True if errors or fatals errors were found while processing the output stream.
HasWarnings
    True if errors or fatals errors were found while processing the output stream.
Log(entry, condition=True)
    Write an entry to the local logger.
LogDebug(*args, condition=True, **kwargs)
LogDryRun(*args, condition=True, **kwargs)
LogError(*args, condition=True, **kwargs)
LogFatal(*args, condition=True, **kwargs)
LogInfo(*args, condition=True, **kwargs)
LogNormal(*args, condition=True, **kwargs)
```



```
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POCKET_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Mentor.ModelSim.VHDL Simulator (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
    The executable to launch.
    _value = None
class FlagQuietMode
    Run simulation in quiet mode. (Don't show 'Loading...' messages.
    _name = 'quiet'
    _value = None
class FlagBatchMode
    Run simulation in batch mode.
    _name = 'batch'
    _value = None
class FlagGuiMode
    Run simulation in GUI mode.
    _name = 'gui'
    _value = None
class SwitchBatchCommand
    Specify a Tcl batch script for the batch mode.
    _name = 'do'
    _value = None
class FlagCommandLineMode
    Run simulation in command line mode.
    _name = 'c'
    _value = None
```

```
class SwitchModelSimIniFile
    Specify the used 'modelsim.ini' file.
    _name = 'modelsimini'
    _value = None

class FlagEnableOptimization
    Enabled optimization while elaborating the design.
    _name = 'vopt'

class FlagDisableOptimization
    Disabled optimization while elaborating the design.
    _name = 'novopt'

class FlagEnableOptimizationVerbosity
    Enabled optimization while elaborating the design.
    _name = 'vopt_verbose'

class FlagEnableKeepAssertionCountsForCoverage

    _name = 'assertcover'

class FlagDisableKeepAssertionCountsForCoverage

    _name = 'noassertcover'

class FlagEnableCoverage

    _name = 'coverage'

class FlagDisableCoverage

    _name = 'nocoverage'

class FlagEnablePSL

    _name = 'psl'

class FlagDisablePSL

    _name = 'nops1'

class FlagEnableFSMDebugging

    _name = 'fsmdebug'

class FlagReportAsNote

    _name = 'note'
    _value = None

class FlagReportAsError

    _name = 'error'
    _value = None

class FlagReportAsWarning
```

```
    _name = 'warning'
    _value = None
class FlagReportAsFatal

    _name = 'fatal'
    _value = None
class FlagRelaxLanguageChecks

    _name = 'permissive'
class FlagForceLanguageChecks

    _name = 'pedanticerrors'
class SwitchTimeResolution
    Set simulation time resolution.

    _name = 't'
    _value = None
class ArgLogFile

    _name = 'l'
    _value = None
class ArgKeepStdOut

    _name = 'keepstdout'
class ArgVHDLLibraryName

    _name = 'lib'
    _value = None
class ArgOnFinishMode

    _name = 'onfinish'
    _value = None
class SwitchTopLevel
    The top-level for simulation.

    _value = None
Parameters = [<class 'ToolChain.Mentor.ModelSim.VHDL Simulator.Executable'>, <class
    Specify all accepted command line arguments
Simulate()
    Start a simulation.
GetReader()
HasErrors
    True if errors or fatals errors were found while processing the output stream.
HasWarnings
    True if errors or fatals errors were found while processing the output stream.
```

```
Log (entry, condition=True)  
    Write an entry to the local logger.  
LogDebug (*args, condition=True, **kwargs)  
LogDryRun (*args, condition=True, **kwargs)  
LogError (*args, condition=True, **kwargs)  
LogFatal (*args, condition=True, **kwargs)  
LogInfo (*args, condition=True, **kwargs)  
LogNormal (*args, condition=True, **kwargs)  
LogQuiet (*args, condition=True, **kwargs)  
LogVerbose (*args, condition=True, **kwargs)  
LogWarning (*args, condition=True, **kwargs)  
Logger  
    Return the local logger instance.  
Path  
ReadUntilBoundary (indent=0)  
Send (line, end='\n')  
SendBoundary ()  
StartProcess (parameterList)  
Terminate ()  
_POC_BOUNDARY = '===== POC BOUNDARY ====='  
_TryLog (*args, condition=True, **kwargs)
```

Functions

```
ToolChain.Mentor.ModelSim.VLibFilter (gen)  
ToolChain.Mentor.ModelSim.VComFilter (gen)  
ToolChain.Mentor.ModelSim.VSIMFilter (gen)
```

ToolChain.Mentor.QuestaSim

Exceptions

- *QuestaSimException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.

```
exception ToolChain.Mentor.QuestaSim.QuestaSimException (message="")
```

Inheritance

Members

```
__init__ (message="")  
    Exception initializer  
  
    Parameters message (str) – The exception message.
```

__str__()

Returns the exception's message text.

args

class ToolChain.Mentor.QuestaSim.Configuration (*host: Base.IHost*)

Inheritance

Members

_vendor = 'Mentor'

The name of the tools vendor.

_toolName = 'Mentor QuestaSim'

The name of the tool.

_section = 'INSTALL.Mentor.QuestaSim'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_multiVersionSupport = True

Mentor QuestaSim supports multiple versions installed on the same system.

_template = {'Linux': {'INSTALL.Mentor.QuestaSim': {'Version': '10.5c', 'SectionName': 'QuestaSim'}}

The template for the configuration sections represented as nested dictionaries.

ConfigureForAll()

Configuration routine for Mentor Graphics ModelSim on all supported platforms.

1. Ask if ModelSim is installed.
 - Pass → skip this configuration. Don't change existing settings.
 - Yes → collect installation information for ModelSim.
 - No → clear the ModelSim configuration section.
1. Ask for ModelSim's version.
2. Ask for ModelSim's edition (PE, PE student, SE 32-bit, SE 64-bit).
3. Ask for ModelSim's installation directory.

_CheckQuestaSimVersion(binPath, version)

CheckDependency()

Check if general Mentor Graphics support is configured in PoC.

ClearSection(writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod `GetSections(platform)`

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured()

Return true if the configurations section is configured

IsSupportedPlatform()

Return true if the given platform is supported by this configuration routine.

Log(entry, condition=True)

Write an entry to the local logger.

LogDebug(*args, condition=True, **kwargs)

LogDryRun(*args, condition=True, **kwargs)

LogError(*args, condition=True, **kwargs)

LogFatal(*args, condition=True, **kwargs)

LogInfo(*args, condition=True, **kwargs)

LogNormal(*args, condition=True, **kwargs)

LogQuiet(*args, condition=True, **kwargs)

LogVerbose(*args, condition=True, **kwargs)

LogWarning(*args, condition=True, **kwargs)

Logger

Return the local logger instance.

PrepareOptions(writeWarnings=True)

PrepareSections(warningWasWritten, writeWarnings=True)

PrepareVersionedSections(writeWarnings=False)

RunPostConfigurationTasks()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask(question, default, beforeDefault="", afterDefault="", indent=1)

_AskInstalled(question)

Ask a Yes/No/Pass question.

_AskYes_NoPass(question, indent=1)

Ask a yes/NO/pass question.

_Ask_YesNoPass(question, indent=1)

Ask a YES/no/pass question.

_CheckModelSimVersion(binPath, version)

_ConfigureBinaryDirectory()

Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition(editions, defaultEdition)

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (optionName)**_GetDefaultVersion ()**

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_GetModelSimVersion (binPath)**_PrintAvailableEditions (editions, selectedEdition)**

Print all available editions and return the selected index.

_TestDefaultInstallPath (defaults)

Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)**Exceptions**

- `MentorException`: Base class for all tool specific exceptions

Classes

- `Configuration`: Configuration routines for Mentor Graphics as a vendor.

exception `ToolChain.Mentor.MentorException (message=)`

Inheritance**Members****__init__ (message=)**

Exception initializer

Parameters `message (str)` – The exception message.

__str__ ()

Returns the exception's message text.

args

class `ToolChain.Mentor.Configuration (host: Base.IHost)`

Configuration routines for Mentor Graphics as a vendor.

This configuration provides a common installation directory setup for all Mentor Graphics tools installed on a system.

Inheritance

Members

_vendor = 'Mentor'

The name of the tools vendor.

_section = 'INSTALL.Mentor'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'ALL': {'INSTALL.ModelSim': {'SectionName': '', 'Version': '\${Se

The template for the configuration sections represented as nested dictionaries.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

CheckDependency ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForAll ()

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)
LogInfo (**args, condition=True, **kwargs*)
LogNormal (**args, condition=True, **kwargs*)
LogQuiet (**args, condition=True, **kwargs*)
LogVerbose (**args, condition=True, **kwargs*)
LogWarning (**args, condition=True, **kwargs*)
Logger
Return the local logger instance.
PrepareOptions (*writeWarnings=True*)
PrepareSections (*warningWasWritten, writeWarnings=True*)
RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.
SectionName
Return the configuration's section name.
State
Return the configuration state.
_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)
_AskInstalled (*question*)
Ask a Yes/No/Pass question.
_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.
_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.
_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.
_GetDefaultOptionValue (*optionName*)
_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.
_TestDefaultInstallPath (*defaults*)
Helper function for automatic search of installation directory.
_TryLog (**args, condition=True, **kwargs*)
_multiVersionSupport = False

14.7.10 ToolChain.PoC

Classes

- *Configuration*: Base class for all tool Configuration classes.

class ToolChain.PoC.Configuration (*host: Base.IHost*)

Inheritance

Members

_vendor = 'VLSI-EDA'

The name of the tools vendor.

_toolName = 'PoC'

The name of the tool.

_section = 'INSTALL.PoC'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'ALL': {'INSTALL.PoC': {'Version': '1.1.1', 'InstallationDirectory':

The template for the configuration sections represented as nested dictionaries.

ConfigureForAll ()

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

CheckDependency ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_ConfigureBinaryDirectory ()
Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition (*editions, defaultEdition*)

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_ConfigureVersion ()
If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()
Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()
Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()
Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)
Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

```
_multiVersionSupport = False
```

14.7.11 ToolChain.Synopsys

Exceptions

- *SynopsysException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Configuration routines for Synopsys as a vendor.
- *SynopsysDesignConstraintFile*: Undocumented.

```
exception ToolChain.Synopsys.SynopsysException (message="")
```

Inheritance

Members

```
__init__(message="")  
Exception initializer
```

Parameters *message* (*str*) – The exception message.

```
__str__()   
Returns the exception's message text.
```

args

```
class ToolChain.Synopsys.Configuration (host: Base.IHost)
```

Configuration routines for Synopsys as a vendor.

This configuration provides a common installation directory setup for all Synopsys tools installed on a system.

Inheritance

Members

```
_vendor = 'Synopsys'  
The name of the tools vendor.
```

```
_section = 'INSTALL.Synopsys'  
The name of the configuration section. Pattern: INSTALL.Vendor.ToolName.
```

```
_template = {'Linux': {'INSTALL.Synopsys': {'InstallationDirectory': '/opt/Synopsys'}}  
The template for the configuration sections represented as nested dictionaries.
```

```
_GetDefaultInstallationDirectory()  
Return unresolved default installation directory (str) from template.
```

Overwrite function in sub-class for automatic search of installation directory.

```
CheckDependency()  
Check if all vendor or tool dependencies are fulfilled to configure this tool.
```

```
ClearSection (writeWarnings=False)  
Clear the configuration section associated to this Configuration class.
```

ConfigureForAll ()

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

```
_Ask (question, default, beforeDefault= "", afterDefault= "", indent=1)  
_AskInstalled (question)  
    Ask a Yes/No/Pass question.  
_AskYes_NoPass (question, indent=1)  
    Ask a yes/NO/pass question.  
_Ask_YesNoPass (question, indent=1)  
    Ask a YES/no/pass question.  
_ConfigureInstallationDirectory ()  
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path  
    object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.  
_GetDefaultOptionValue (optionName)  
_PrintAvailableEditions (editions, selectedEdition)  
    Print all available editions and return the selected index.  
_TestDefaultInstallPath (defaults)  
    Helper function for automatic search of installation directory.  
_TryLog (*args, condition=True, **kwargs)  
_multiVersionSupport = False  
class ToolChain.Synopsys.SynopsysDesignConstraintFile (file, project=None, file-  
                                                    Set=None)
```

Inheritance

Members

```
_FileType = <FileTypes.SdcConstraintFile bits=0x1000 data=UNDEFINED>  
FileName  
FileSet  
FileType  
Open ()  
Path  
Project  
ReadFile ()  
_ReadContent ()
```

14.7.12 ToolChain.Windows

Exceptions

- *WindowsException*: Base class for all tool specific exceptions

Classes

- *Cmd*: Represent an executable.

```
exception ToolChain.Windows.WindowsException (message="")
```

Inheritance

Members

`__init__(message="")`

Exception initializer

Parameters `message` (*str*) – The exception message.

`__str__()`

Returns the exception's message text.

args

`class ToolChain.Windows.Cmd(platform, dryrun, logger=None)`

Inheritance

Members

`class Executable`

`class SwitchCommand`

`_name = 'C'`

Parameters = [`<class 'ToolChain.Windows.Cmd.Executable'>`, `<class 'ToolChain.Windows`

`GetEnvironment` (*settingsFile=None*)

`GetReader()`

`Log` (*entry, condition=True*)

Write an entry to the local logger.

`LogDebug` (**args, condition=True, **kwargs*)

`LogDryRun` (**args, condition=True, **kwargs*)

`LogError` (**args, condition=True, **kwargs*)

`LogFatal` (**args, condition=True, **kwargs*)

`LogInfo` (**args, condition=True, **kwargs*)

`LogNormal` (**args, condition=True, **kwargs*)

`LogQuiet` (**args, condition=True, **kwargs*)

`LogVerbose` (**args, condition=True, **kwargs*)

`LogWarning` (**args, condition=True, **kwargs*)

`Logger`

Return the local logger instance.

`Path`

`ReadUntilBoundary` (*indent=0*)

`Send` (*line, end='\n'*)

`SendBoundary()`

`StartProcess` (*parameterList*)

`Terminate()`

`_POC_BOUNDARY = '==== POC BOUNDARY ====='`

`_TryLog (*args, condition=True, **kwargs)`

14.7.13 ToolChain.Xilinx

Submodules

ToolChain.Xilinx.ISE

Exceptions

- `ISEException`: Base class for all tool specific exceptions

Classes

- `Configuration`: Base class for all tool Configuration classes.
- `ISE`: Undocumented.
- `Fuse`: Represent an executable.
- `ISESimulator`: Represent an executable.
- `Xst`: Represent an executable.
- `CoreGenerator`: Represent an executable.
- `ISEProject`: Undocumented.
- `ISEProjectFile`: Undocumented.
- `UserConstraintFile`: Undocumented.

Functions

- `VhCompFilter()`: Undocumented.
- `FuseFilter()`: Undocumented.
- `SimulatorFilter()`: Undocumented.
- `XstFilter()`: Undocumented.
- `CoreGeneratorFilter()`: Undocumented.

exception `ToolChain.Xilinx.ISE.ISEException (message=)`

Inheritance

Members

`__init__ (message=)`
Exception initializer

Parameters `message (str)` – The exception message.

`__str__ ()`
Returns the exception's message text.

args

class `ToolChain.Xilinx.ISE.Configuration (host: Base.IHost)`

Inheritance

Members

_vendor = 'Xilinx'

The name of the tools vendor.

_toolName = 'Xilinx ISE'

The name of the tool.

_section = 'INSTALL.Xilinx.ISE'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_multiVersionSupport = True

Xilinx ISE supports multiple versions installed on the same system.

_template = {'Linux': {'INSTALL.Xilinx.ISE': {'Version': '14.7', 'SectionName':

The template for the configuration sections represented as nested dictionaries.

CheckDependency ()

Check if general Xilinx support is configured in PoC.

ConfigureForAll ()

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

PrepareVersionedSections (*writeWarnings=False*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question, default, beforeDefault= "", afterDefault= "", indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)

Ask a YES/no/pass question.

_Configuration__CheckISEVersion (*binPath*)

_ConfigureBinaryDirectory ()

Updates section with value from *_template* and returns directory as *Path* object.

_ConfigureEdition (*editions, defaultEdition*)

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns *Path* object. If no installation directory was configured before, then *_GetDefaultInstallationDir* is called.

_ConfigureVersion ()

If no version was configured before, then *_GetDefaultVersion* is called. Asks for version and updates section. Returns version as string.

_GetDefaultEdition ()

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_GetDefaultVersion ()

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args, condition=True, **kwargs*)

```
class ToolChain.Xilinx.ISE.ISE(platform, dryrun, binaryDirectoryPath, version, log-
                               ger=None)
```

Inheritance

Members

PreparseEnvironment (*installationDirectory*)

GetVHDLCompiler ()

GetFuse ()

GetXst ()

GetCoreGenerator ()

```
class ToolChain.Xilinx.ISE.Fuse(toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
```

```
class FlagIncremental
```

```
    _name = 'incremental'
```

```
class FlagRangeCheck
```

```
    _name = 'rangecheck'
```

```
class SwitchMultiThreading
```

```
    _name = 'mt'
```

```
class SwitchTimeResolution
```

```
    _name = 'timeprecision_vhdl'
```

```
class SwitchProjectFile
```

```
    _name = 'prj'
```

```
class SwitchOutputFile
```

```
    _name = 'o'
```

```
class ArgTopLevel
```

```
Parameters = [<class 'ToolChain.Xilinx.ISE.Fuse.Executable'>, <class 'ToolChain.Xil
```

Link ()

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger

Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POC_BOUNDARY = '==== POC BOUNDARY ==='

_TryLog (*args, condition=True, **kwargs)

```
class ToolChain.Xilinx.ISE.ISESimulator (platform, dryrun, executablePath, environment,  
                                         logger=None)
```

Inheritance

Members

```
class Executable
```

```
class SwitchLogFile
```

```
    _name = 'log'
```

```
class FlagGuiMode
```

```
    _name = 'gui'
```

```
class SwitchTclBatchFile
```

```

    _name = 'tclbatch'

class SwitchWaveformFile

    _name = 'view'

Parameters = [<class 'ToolChain.Xilinx.ISE.ISESimulator.Executable'>, <class 'ToolChain.Xilinx.ISE.ISESimulator.Simulator'>]

Simulate()

GetReader()

HasErrors
    True if errors or fatals errors were found while processing the output stream.

HasWarnings
    True if errors or fatals errors were found while processing the output stream.

Log(entry, condition=True)
    Write an entry to the local logger.

LogDebug(*args, condition=True, **kwargs)

LogDryRun(*args, condition=True, **kwargs)

LogError(*args, condition=True, **kwargs)

LogFatal(*args, condition=True, **kwargs)

LogInfo(*args, condition=True, **kwargs)

LogNormal(*args, condition=True, **kwargs)

LogQuiet(*args, condition=True, **kwargs)

LogVerbose(*args, condition=True, **kwargs)

LogWarning(*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary(indent=0)

Send(line, end='\n')

SendBoundary()

StartProcess(parameterList)

Terminate()

_POCKET_BOUNDARY = '==== POC BOUNDARY ====='

_TryLog(*args, condition=True, **kwargs)

class ToolChain.Xilinx.ISE.Xst(toolchain: ToolChain.ToolMixIn)

```

Inheritance

Members

```
class Executable
```

```
class SwitchIntStyle
```

```
    _name = 'intstyle'
```

```
class SwitchXstFile

    _name = 'ifn'

class SwitchReportFile

    _name = 'ofn'

Parameters = [<class 'ToolChain.Xilinx.ISE.Xst.Executable'>, <class 'ToolChain.Xilin

Compile ()

GetReader ()

HasErrors
    True if errors or fatals errors were found while processing the output stream.

HasWarnings
    True if errors or fatals errors were found while processing the output stream.

Log (entry, condition=True)
    Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)

LogDryRun (*args, condition=True, **kwargs)

LogError (*args, condition=True, **kwargs)

LogFatal (*args, condition=True, **kwargs)

LogInfo (*args, condition=True, **kwargs)

LogNormal (*args, condition=True, **kwargs)

LogQuiet (*args, condition=True, **kwargs)

LogVerbose (*args, condition=True, **kwargs)

LogWarning (*args, condition=True, **kwargs)

Logger
    Return the local logger instance.

Path

ReadUntilBoundary (indent=0)

Send (line, end='\n')

SendBoundary ()

StartProcess (parameterList)

Terminate ()

_POX_BOUNDARY = '==== POC BOUNDARY ===='

_TryLog (*args, condition=True, **kwargs)

class ToolChain.Xilinx.ISE.CoreGenerator (toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

```
class Executable
```

```
class FlagRegenerate

    _name = 'r'
class SwitchProjectFile

    _name = 'p'
class SwitchBatchFile

    _name = 'b'
Parameters = [<class 'ToolChain.Xilinx.ISE.CoreGenerator.Executable'>, <class 'ToolChain.Xilinx.ISE.CoreGenerator.Executable'>]
Generate ()
GetReader ()
HasErrors
    True if errors or fatals errors were found while processing the output stream.
HasWarnings
    True if errors or fatals errors were found while processing the output stream.
Log (entry, condition=True)
    Write an entry to the local logger.
LogDebug (*args, condition=True, **kwargs)
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary (indent=0)
Send (line, end='\n')
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POCS_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Xilinx.ISE.ISEProject (name)
```

Inheritance

Members

```

AddExternalVHDLLibraries (library)
AddFile (file, fileSet=None)
AddFileSet (fileSet)
AddSourceFile (file, fileSet=None)
Board
CreateFileSet (name, setDefault=True)
DefaultFileSet
Device
Environment
ExternalVHDLLibraries
ExtractVHDLLibrariesFromVHDLSourceFiles ()
FileSets
Files (fileType=<FileTypes(Text|ProjectFile|FileListFile|RulesFile|SourceFile|VHDLSourceFile|VerilogSourceFile|Python|
        bits=0xFFFF>, fileSet=None)
GetVariables ()
Name
RootDirectory
Tool
ToolChain
VHDLLibraries
VHDLVersion
pprint (indent=0)
class ToolChain.Xilinx.ISE.ISEProjectFile (file)

```

Inheritance

Members

```

FileName
FileSet
FileType
Open ()
Path
Project
ReadFile ()
_FileType = <FileTypes.ProjectFile bits=0x0002 data=UNDEFINED>
_ReadContent ()

```



```
class ToolChain.Xilinx.ISE.UserConstraintFile (file, project=None, fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.UcfConstraintFile bits=0x0400 data=UNDEFINED>  
FileName  
FileSet  
FileType  
Open()  
Path  
Project  
ReadFile()  
_ReadContent()
```

Functions

```
ToolChain.Xilinx.ISE.VhCompFilter (gen)  
ToolChain.Xilinx.ISE.FuseFilter (gen)  
ToolChain.Xilinx.ISE.SimulatorFilter (gen)  
ToolChain.Xilinx.ISE.XstFilter (gen)  
ToolChain.Xilinx.ISE.CoreGeneratorFilter (gen)
```

ToolChain.Xilinx.Vivado

Exceptions

- *VivadoException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Base class for all tool Configuration classes.
- *ToolMixin*: Undocumented.
- *Vivado*: Undocumented.
- *XElab*: Represent an executable.
- *XSim*: Represent an executable.
- *Synth*: Represent an executable.
- *VivadoProject*: Undocumented.
- *VivadoProjectFile*: Undocumented.
- *XilinxDesignConstraintFile*: Undocumented.

Functions

- *ElaborationFilter()*: Undocumented.
- *SimulatorFilter()*: Undocumented.
- *CompilerFilter()*: Undocumented.

```
exception ToolChain.Xilinx.Vivado.VivadoException (message="")
```

Inheritance

Members

`__init__(message="")`
Exception initializer

Parameters **message** (*str*) – The exception message.

`__str__()`
Returns the exception's message text.

args

class `ToolChain.Xilinx.Vivado.Configuration` (*host: Base.IHost*)

Inheritance

Members

`_vendor = 'Xilinx'`
The name of the tools vendor.

`_toolName = 'Xilinx Vivado'`
The name of the tool.

`_section = 'INSTALL.Xilinx.Vivado'`
The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_multiVersionSupport = True`
Xilinx Vivado supports multiple versions installed on the same system.

`_template = {'Linux': {'INSTALL.Xilinx.Vivado': {'Version': '2016.3', 'SectionName': 'Xilinx Vivado'}}`
The template for the configuration sections represented as nested dictionaries.

`CheckDependency()`
Check if general Xilinx support is configured in PoC.

`ConfigureForAll()`
Start a generic (platform independent) configuration procedure.
Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

`ClearSection(writeWarnings=False)`
Clear the configuration section associated to this Configuration class.

`ConfigureForDarwin()`
Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux()`
Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

`ConfigureForWindows()`
Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod `GetSections(platform)`
Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (entry, condition=True)

Write an entry to the local logger.

LogDebug (*args, condition=True, **kwargs)**LogDryRun (*args, condition=True, **kwargs)****LogError (*args, condition=True, **kwargs)****LogFatal (*args, condition=True, **kwargs)****LogInfo (*args, condition=True, **kwargs)****LogNormal (*args, condition=True, **kwargs)****LogQuiet (*args, condition=True, **kwargs)****LogVerbose (*args, condition=True, **kwargs)****LogWarning (*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

PrepareOptions (writeWarnings=True)**PrepareSections (warningWasWritten, writeWarnings=True)****PrepareVersionedSections (writeWarnings=False)****RunPostConfigurationTasks ()**

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (question, default, beforeDefault="", afterDefault="", indent=1)**_AskInstalled (question)**

Ask a Yes/No/Pass question.

_AskYes_NoPass (question, indent=1)

Ask a yes/NO/pass question.

_Ask_YesNoPass (question, indent=1)

Ask a YES/no/pass question.

_Configuration__CheckVivadoVersion (binPath, version)**_ConfigureBinaryDirectory ()**

Updates section with value from `_template` and returns directory as `Path` object.

_ConfigureEdition (editions, defaultEdition)**_ConfigureInstallationDirectory ()**

Asks for installation directory and updates section. Checks if entered directory exists and returns `Path` object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

`_ConfigureVersion()`

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

`_GetDefaultEdition()`

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

`_GetDefaultInstallationDirectory()`

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

`_GetDefaultOptionValue(optionName)`

`_GetDefaultVersion()`

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

`_PrintAvailableEditions(editions, selectedEdition)`

Print all available editions and return the selected index.

`_TestDefaultInstallPath(defaults)`

Helper function for automatic search of installation directory.

`_TryLog(*args, condition=True, **kwargs)`

```
class ToolChain.Xilinx.Vivado.ToolMixIn(platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Inheritance

Members

```
class ToolChain.Xilinx.Vivado.Vivado(platform, dryrun, binaryDirectoryPath, version, logger=None)
```

Inheritance

Members

PreparseEnvironment (*installationDirectory*)

GetElaborator ()

GetSimulator ()

GetSynthesizer ()

```
class ToolChain.Xilinx.Vivado.XElab(toolchain: ToolChain.ToolMixIn)
```

Inheritance

Members

class Executable

_value = None

```
class FlagRangeCheck
```

```
    _name = 'rangecheck'
```

```
    _value = None
```

```
class SwitchMultiThreading
```

```
    _name = 'mt'
```

```
    _value = None
```

```
class SwitchVerbose
```

```
    _name = 'verbose'
```

```
    _value = None
```

```
class SwitchDebug
```

```
    _name = 'debug'
```

```
    _value = None
```

```
class SwitchOptimization
```

```
    _pattern = '--{0}{1}'
```

```
    _name = 'O'
```

```
    _value = None
```

```
class SwitchTimeResolution
```

```
    _name = 'timeprecision_vhdl'
```

```
    _value = None
```

```
class SwitchProjectFile
```

```
    _name = 'prj'
```

```
    _value = None
```

```
class SwitchLogFile
```

```
    _name = 'log'
```

```
    _value = None
```

```
class SwitchSnapshot
```

```
    _name = 's'
```

```
    _value = None
```

```
class ArgTopLevel
```

```
    _value = None
```

```
Parameters = [<class 'ToolChain.Xilinx.Vivado.XElab.Executable'>, <class 'ToolChain
```

```
Link()
```

GetReader ()

HasErrors

True if errors or fatals errors were found while processing the output stream.

HasWarnings

True if errors or fatals errors were found while processing the output stream.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

Path

ReadUntilBoundary (*indent=0*)

Send (*line*, *end='\n'*)

SendBoundary ()

StartProcess (*parameterList*)

Terminate ()

_POC_BOUNDARY = '===== **POC BOUNDARY** ====='

_TryLog (**args*, *condition=True*, ***kwargs*)

class ToolChain.Xilinx.Vivado.XSim (*toolchain: ToolChain.ToolMixIn*)

Inheritance

Members

class Executable

_value = None

class SwitchLogFile

_name = '-log'

_value = None

class FlagGuiMode

_name = '-gui'

```

    _value = None
class SwitchTclBatchFile

    _name = '-tclbatch'
    _value = None
class SwitchWaveformFile

    _name = '-view'
    _value = None
class SwitchSnapshot

    _value = None
Parameters = [<class 'ToolChain.Xilinx.Vivado.XSim.Executable'>, <class 'ToolChain.Xilinx.Vivado.XSim.Executable'>]
Simulate()
GetReader()
HasErrors
    True if errors or fatals errors were found while processing the output stream.
HasWarnings
    True if errors or fatals errors were found while processing the output stream.
Log(entry, condition=True)
    Write an entry to the local logger.
LogDebug(*args, condition=True, **kwargs)
LogDryRun(*args, condition=True, **kwargs)
LogError(*args, condition=True, **kwargs)
LogFatal(*args, condition=True, **kwargs)
LogInfo(*args, condition=True, **kwargs)
LogNormal(*args, condition=True, **kwargs)
LogQuiet(*args, condition=True, **kwargs)
LogVerbose(*args, condition=True, **kwargs)
LogWarning(*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
Path
ReadUntilBoundary(indent=0)
Send(line, end='\n')
SendBoundary()
StartProcess(parameterList)
Terminate()
_POO_BOUNDARY = '==== POC BOUNDARY ====='
_TryLog(*args, condition=True, **kwargs)
class ToolChain.Xilinx.Vivado.Synth(toolchain: ToolChain.ToolMixIn)

```

Inheritance

Members

class Executable

 _value = None

class SwitchLogFile

 _name = 'log'

 _value = None

class SwitchSourceFile

 _name = 'source'

 _value = None

class SwitchMode

 _name = 'mode'

 _value = 'batch'

Parameters = [<class 'ToolChain.Xilinx.Vivado.Synth.Executable'>, <class 'ToolChain

Compile()

GetReader()

HasErrors

 True if errors or fatals errors were found while processing the output stream.

HasWarnings

 True if errors or fatals errors were found while processing the output stream.

Log(entry, condition=True)

 Write an entry to the local logger.

LogDebug(*args, condition=True, **kwargs)

LogDryRun(*args, condition=True, **kwargs)

LogError(*args, condition=True, **kwargs)

LogFatal(*args, condition=True, **kwargs)

LogInfo(*args, condition=True, **kwargs)

LogNormal(*args, condition=True, **kwargs)

LogQuiet(*args, condition=True, **kwargs)

LogVerbose(*args, condition=True, **kwargs)

LogWarning(*args, condition=True, **kwargs)

Logger

 Return the local logger instance.

Path

ReadUntilBoundary(indent=0)

Send(line, end='\n')


```
SendBoundary ()
StartProcess (parameterList)
Terminate ()
_POO_BOUNDARY = '==== POC BOUNDARY ===='
_TryLog (*args, condition=True, **kwargs)
class ToolChain.Xilinx.Vivado.VivadoProject (name)
```

Inheritance

Members

```
AddExternalVHDLLibraries (library)
AddFile (file, fileSet=None)
AddFileSet (fileSet)
AddSourceFile (file, fileSet=None)
Board
CreateFileSet (name, setDefault=True)
DefaultFileSet
Device
Environment
ExternalVHDLLibraries
ExtractVHDLLibrariesFromVHDLSourceFiles ()
FileSets
Files (fileType=<FileTypes(Text\ProjectFile\FileListFile\RulesFile\SourceFile\VHDLSourceFile\VerilogSourceFile\Python
bits=0xFFFF>, fileSet=None)
GetVariables ()
Name
RootDirectory
Tool
ToolChain
VHDLLibraries
VHDLVersion
pprint (indent=0)
class ToolChain.Xilinx.Vivado.VivadoProjectFile (file)
```

Inheritance

Members

```
FileName
FileSet
FileType
```

```
Open ()
Path
Project
ReadFile ()
_FileType = <FileTypes.ProjectFile bits=0x0002 data=UNDEFINED>
_ReadContent ()
class ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile (file, project=None,
                                                         fileSet=None)
```

Inheritance

Members

```
_FileType = <FileTypes.XdcConstraintFile bits=0x0800 data=UNDEFINED>
FileName
FileSet
FileType
Open ()
Path
Project
ReadFile ()
_ReadContent ()
```

Functions

```
ToolChain.Xilinx.Vivado.ElaborationFilter (gen)
ToolChain.Xilinx.Vivado.SimulatorFilter (gen)
ToolChain.Xilinx.Vivado.CompilerFilter (gen)
```

Exceptions

- *XilinxException*: Base class for all tool specific exceptions

Classes

- *Configuration*: Configuration routines for Xilinx as a vendor.
- *XilinxProjectExportMixin*: Undocumented.

```
exception ToolChain.Xilinx.XilinxException (message=)
```

Inheritance

Members

```
__init__ (message=)
    Exception initializer
    Parameters message (str) – The exception message.
__str__ ()
    Returns the exception's message text.
```

args

class ToolChain.Xilinx.Configuration (*host: Base.IHost*)

Configuration routines for Xilinx as a vendor.

This configuration provides a common installation directory setup for all Xilinx tools installed on a system.

Inheritance

Members

_vendor = 'Xilinx'

The name of the tools vendor.

_section = 'INSTALL.Xilinx'

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

_template = {'Linux': {'INSTALL.Xilinx': {'InstallationDirectory': '/opt/Xilinx'}}

The template for the configuration sections represented as nested dictionaries.

_GetDefaultInstallationDirectory ()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

CheckDependency ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (writeWarnings=False)

Clear the configuration section associated to this Configuration class.

ConfigureForAll ()

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll ()`. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (platform)

Return all section names for this configuration.

Host

Return the hosting object.

IsConfigured ()

Return true if the configurations section is configured

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten*, *writeWarnings=True*)

RunPostConfigurationTasks ()

Virtual method. Overwrite to execute post-configuration tasks.

SectionName

Return the configuration's section name.

State

Return the configuration state.

_Ask (*question*, *default*, *beforeDefault="*, *afterDefault="*, *indent=1*)

_AskInstalled (*question*)

Ask a Yes/No/Pass question.

_AskYes_NoPass (*question*, *indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)

Ask a YES/no/pass question.

_ConfigureInstallationDirectory ()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_GetDefaultOptionValue (*optionName*)

_PrintAvailableEditions (*editions*, *selectedEdition*)

Print all available editions and return the selected index.

_TestDefaultInstallPath (*defaults*)

Helper function for automatic search of installation directory.

_TryLog (**args*, *condition=True*, ***kwargs*)

_multiVersionSupport = False

class ToolChain.Xilinx.XilinxProjectExportMixin

Inheritance

Members

`_GenerateXilinxProjectFileContent` (*tool*, *vhdlVersion=93*)

`_WriteXilinxProjectFile` (*projectFilePath*, *tool*, *vhdlVersion=93*)

Exceptions

- *ToolChainException*: Base class for all tool specific exceptions
- *ConfigurationException*: ConfigurationException is raise while running configuration or database
- *SkipConfigurationException*: SkipConfigurationException is a ConfigurationException,

Classes

- *ConfigurationState*: Describes the configuration state of a tool or vendor.
- *ChangeState*: Describes if a configuration was changed.
- *ToolMixIn*: Undocumented.
- *AskMixIn*: Undocumented.
- *Configuration*: Base class for all Configuration classes.
- *VendorConfiguration*: Base class for all vendor Configuration classes.
- *ToolConfiguration*: Base class for all tool Configuration classes.
- *EditionDescription*: EditionDescription(Name, Section)
- *Edition*: An enumeration.
- *ToolSelector*: Base class for all Selector classes.
- *Configurator*: A instance of this class controls the interactive configuration process.

exception `ToolChain.ToolChainException` (*message=""*)
Base class for all tool specific exceptions

Inheritance

Members

`__init__` (*message=""*)
Exception initializer

Parameters *message* (*str*) – The exception message.

`__str__` ()
Returns the exception's message text.

args

exception `ToolChain.ConfigurationException` (*message=""*)
ConfigurationException is raise while running configuration or database tasks in PoC

Inheritance

Members

`__init__` (*message=""*)
Exception initializer

Parameters **message** (*str*) – The exception message.

__str__()

Returns the exception's message text.

args

exception `ToolChain.SkipConfigurationException` (*message=""*)

`SkipConfigurationException` is a *ConfigurationException*, which can be skipped.

Inheritance

Members

__init__ (*message=""*)

Exception initializer

Parameters **message** (*str*) – The exception message.

__str__()

Returns the exception's message text.

args

class `ToolChain.ConfigurationState`

Describes the configuration state of a tool or vendor.

Inheritance

Members

Unconfigured = 0

Configured = 1

class `ToolChain.ChangeState`

Describes if a configuration was changed.

Inheritance

Members

Unchanged = 0

Changed = 1

class `ToolChain.ToolMixIn` (*platform, dryrun, binaryDirectoryPath, version, logger=None*)

Inheritance

Members

class `ToolChain.AskMixIn`

Inheritance

Members

`_Ask` (*question, default, beforeDefault="", afterDefault="", indent=1*)

`_Ask_YesNoPass` (*question, indent=1*)

Ask a YES/no/pass question.

`_AskYes_NoPass` (*question, indent=1*)

Ask a yes/NO/pass question.

`_PrintAvailableEditions` (*editions, selectedEdition*)

Print all available editions and return the selected index.

`class ToolChain.Configuration` (*host: Base.IHost*)

Base class for all Configuration classes.

Inheritance

Members

`_vendor` = **`'Unknown'`**

The name of the tools vendor.

`_section` = **`'INSTALL.Name'`**

The name of the configuration section. Pattern: `INSTALL.Tool`.

`_multiVersionSupport` = **`False`**

True if a tool supports multiple versions installed on the same system.

`_template` = **`{ 'ALL': { 'INSTALL.Name': {} }, 'Darwin': { 'INSTALL.Name': {} }, 'Linux`**

`'Linux` The template for the configuration sections represented as nested dictionaries.

`Host`

Return the hosting object.

`State`

Return the configuration state.

`SectionName`

Return the configuration's section name.

`IsSupportedPlatform` ()

Return true if the given platform is supported by this configuration routine.

`IsConfigured` ()

Return true if the configurations section is configured

`CheckDependency` ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

`classmethod GetSections` (*platform*)

Return all section names for this configuration.

`PrepareSections` (*warningWasWritten, writeWarnings=True*)

`ClearSection` (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

`PrepareOptions` (*writeWarnings=True*)

`ConfigureForDarwin` ()

Start the configuration procedure for Darwin.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux()

Start the configuration procedure for Linux.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows()

Start the configuration procedure for Windows.

This method is a wrapper for `ConfigureForAll()`. Overwrite this method to implement a Windows specific configuration routine.

ConfigureForAll()

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

_AskInstalled(question)

Ask a Yes/No/Pass question.

_ConfigureInstallationDirectory()

Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_GetDefaultInstallationDirectory()

Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue(optionName)**_TestDefaultInstallPath(defaults)**

Helper function for automatic search of installation directory.

RunPostConfigurationTasks()

Virtual method. Overwrite to execute post-configuration tasks.

Log(entry, condition=True)

Write an entry to the local logger.

LogDebug(*args, condition=True, **kwargs)**LogDryRun(*args, condition=True, **kwargs)****LogError(*args, condition=True, **kwargs)****LogFatal(*args, condition=True, **kwargs)****LogInfo(*args, condition=True, **kwargs)****LogNormal(*args, condition=True, **kwargs)****LogQuiet(*args, condition=True, **kwargs)****LogVerbose(*args, condition=True, **kwargs)****LogWarning(*args, condition=True, **kwargs)****Logger**

Return the local logger instance.

_Ask(question, default, beforeDefault="", afterDefault="", indent=1)**_AskYes_NoPass(question, indent=1)**

Ask a yes/NO/pass question.

_Ask_YesNoPass(question, indent=1)

Ask a YES/no/pass question.

_PrintAvailableEditions (*editions, selectedEdition*)

Print all available editions and return the selected index.

_TryLog (**args, condition=True, **kwargs*)

class ToolChain.**VendorConfiguration** (*host: Base.IHost*)

Base class for all vendor Configuration classes.

Inheritance

Members

_section = 'INSTALL.Vendor.Tool'

The name of the configuration section. Pattern: `INSTALL.Vendor`.

_template = {'Darwin': {'INSTALL.Vendor.Tool': {'InstallationDirectory': '/opt/V

The template for the configuration section represented as nested dictionaries.

IsConfigured ()

Return true if the vendor represented by this Configuration class is configured in PoC.

Inherited method *IsConfigured()* from class *Configuration*.

ConfigureForAll ()

Start a generic (platform independent) vendor configuration procedure.

This method configures a vendor path. Overwrite this method to implement a vendor specific configuration routine for a vendor Configuration class.

CheckDependency ()

Check if all vendor or tool dependencies are fulfilled to configure this tool.

ClearSection (*writeWarnings=False*)

Clear the configuration section associated to this Configuration class.

ConfigureForDarwin ()

Start the configuration procedure for Darwin.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Darwin specific configuration routine.

ConfigureForLinux ()

Start the configuration procedure for Linux.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Linux specific configuration routine.

ConfigureForWindows ()

Start the configuration procedure for Windows.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Windows specific configuration routine.

classmethod **GetSections** (*platform*)

Return all section names for this configuration.

Host

Return the hosting object.

IsSupportedPlatform ()

Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)

Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

```
LogDryRun (*args, condition=True, **kwargs)
LogError (*args, condition=True, **kwargs)
LogFatal (*args, condition=True, **kwargs)
LogInfo (*args, condition=True, **kwargs)
LogNormal (*args, condition=True, **kwargs)
LogQuiet (*args, condition=True, **kwargs)
LogVerbose (*args, condition=True, **kwargs)
LogWarning (*args, condition=True, **kwargs)
Logger
    Return the local logger instance.
PrepareOptions (writeWarnings=True)
PrepareSections (warningWasWritten, writeWarnings=True)
RunPostConfigurationTasks ()
    Virtual method. Overwrite to execute post-configuration tasks.
SectionName
    Return the configuration's section name.
State
    Return the configuration state.
_Ask (question, default, beforeDefault= ", afterDefault= ", indent=1)
_AskInstalled (question)
    Ask a Yes/No/Pass question.
_AskYes_NoPass (question, indent=1)
    Ask a yes/NO/pass question.
_Ask_YesNoPass (question, indent=1)
    Ask a YES/no/pass question.
_ConfigureInstallationDirectory ()
    Asks for installation directory and updates section. Checks if entered directory exists and returns Path
    object. If no installation directory was configured before, then _GetDefaultInstallationDir is called.
_GetDefaultInstallationDirectory ()
    Return unresolved default installation directory (str) from template.
    Overwrite function in sub-class for automatic search of installation directory.
_GetDefaultOptionValue (optionName)
_PrintAvailableEditions (editions, selectedEdition)
    Print all available editions and return the selected index.
_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.
_TryLog (*args, condition=True, **kwargs)
_multiVersionSupport = False
_vendor = 'Unknown'
class ToolChain.ToolConfiguration (host: Base.IHost)
    Base class for all tool Configuration classes.
```

Inheritance

Members

`_section = 'INSTALL.Vendor.Tool'`

The name of the configuration section. Pattern: `INSTALL.Vendor.ToolName`.

`_toolName = 'Tool'`

The name of the tool.

`_template = {'ALL': {'INSTALL.Vendor.Tool': {'Version': '1.0'}}, 'Darwin': {'INS'`

The template for the configuration section represented as nested dictionaries.

`IsConfigured()`

Return true if the tool represented by this Configuration class is configured in PoC.

Inherited method *IsConfigured()* from class *Configuration*.

`_ConfigureVersion()`

If no version was configured before, then `_GetDefaultVersion` is called. Asks for version and updates section. Returns version as string.

`_GetDefaultVersion()`

Returns unresolved default version (str) from template.

Overwrite this method in a sub-class for automatic search of version.

`_ConfigureEdition(editions, defaultEdition)`

`_GetDefaultEdition()`

Returns unresolved default edition (str) from template.

Overwrite this method in a sub-class for automatic search of editions.

`_ConfigureBinaryDirectory()`

Updates section with value from `_template` and returns directory as *Path* object.

`PrepareVersionedSections(writeWarnings=False)`

`CheckDependency()`

Check if all vendor or tool dependencies are fulfilled to configure this tool.

`ClearSection(writeWarnings=False)`

Clear the configuration section associated to this Configuration class.

`ConfigureForAll()`

Start a generic (platform independent) configuration procedure.

Overwrite this method to implement a generic configuration routine for a (tool) Configuration class.

`ConfigureForDarwin()`

Start the configuration procedure for Darwin.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Darwin specific configuration routine.

`ConfigureForLinux()`

Start the configuration procedure for Linux.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Linux specific configuration routine.

`ConfigureForWindows()`

Start the configuration procedure for Windows.

This method is a wrapper for *ConfigureForAll()*. Overwrite this method to implement a Windows specific configuration routine.

classmethod GetSections (*platform*)
Return all section names for this configuration.

Host
Return the hosting object.

IsSupportedPlatform ()
Return true if the given platform is supported by this configuration routine.

Log (*entry, condition=True*)
Write an entry to the local logger.

LogDebug (**args, condition=True, **kwargs*)

LogDryRun (**args, condition=True, **kwargs*)

LogError (**args, condition=True, **kwargs*)

LogFatal (**args, condition=True, **kwargs*)

LogInfo (**args, condition=True, **kwargs*)

LogNormal (**args, condition=True, **kwargs*)

LogQuiet (**args, condition=True, **kwargs*)

LogVerbose (**args, condition=True, **kwargs*)

LogWarning (**args, condition=True, **kwargs*)

Logger
Return the local logger instance.

PrepareOptions (*writeWarnings=True*)

PrepareSections (*warningWasWritten, writeWarnings=True*)

RunPostConfigurationTasks ()
Virtual method. Overwrite to execute post-configuration tasks.

SectionName
Return the configuration's section name.

State
Return the configuration state.

_Ask (*question, default, beforeDefault= ", afterDefault= ", indent=1*)

_AskInstalled (*question*)
Ask a Yes/No/Pass question.

_AskYes_NoPass (*question, indent=1*)
Ask a yes/NO/pass question.

_Ask_YesNoPass (*question, indent=1*)
Ask a YES/no/pass question.

_ConfigureInstallationDirectory ()
Asks for installation directory and updates section. Checks if entered directory exists and returns Path object. If no installation directory was configured before, then `_GetDefaultInstallationDir` is called.

_GetDefaultInstallationDirectory ()
Return unresolved default installation directory (str) from template.

Overwrite function in sub-class for automatic search of installation directory.

_GetDefaultOptionValue (*optionName*)

_PrintAvailableEditions (*editions, selectedEdition*)
Print all available editions and return the selected index.

```

_TestDefaultInstallPath (defaults)
    Helper function for automatic search of installation directory.

_TryLog (*args, condition=True, **kwargs)

_multiVersionSupport = False

_vendor = 'Unknown'

```

```
class ToolChain.EditionDescription (Name, Section)
```

Inheritance

Members

Name

Alias for field number 0

Section

Alias for field number 1

_asdict ()

Return a new OrderedDict which maps field names to their values.

_fields = ('Name', 'Section')

classmethod **_make** (*iterable, new=<built-in method __new__ of type object>, len=<built-in function len>*)

Make a new EditionDescription object from a sequence or iterable

_replace (***kwargs*)

Return a new EditionDescription object replacing specified fields with new values

_source = "from builtins import property as _property, tuple as _tuple\nfrom operato

count (*value*) → integer – return number of occurrences of value

index (*value* [, *start* [, *stop*]]) → integer – return first index of value.

Raises ValueError if the value is not present.

```
class ToolChain.Edition (name, section)
```

An enumeration.

Inheritance

Members

```
class ToolChain.ToolSelector (host: Base.IHost)
```

Base class for all Selector classes.

Inheritance

Members

```
_toolName = ''
```

ToolName

_GetConfiguredEditions (*editions*)

Return all configured editions.

_AskSelection (*editions, defaultEdition*)

Log (*entry*, *condition=True*)

Write an entry to the local logger.

LogDebug (**args*, *condition=True*, ***kwargs*)

LogDryRun (**args*, *condition=True*, ***kwargs*)

LogError (**args*, *condition=True*, ***kwargs*)

LogFatal (**args*, *condition=True*, ***kwargs*)

LogInfo (**args*, *condition=True*, ***kwargs*)

LogNormal (**args*, *condition=True*, ***kwargs*)

LogQuiet (**args*, *condition=True*, ***kwargs*)

LogVerbose (**args*, *condition=True*, ***kwargs*)

LogWarning (**args*, *condition=True*, ***kwargs*)

Logger

Return the local logger instance.

_Ask (*question*, *default*, *beforeDefault=*”, *afterDefault=*”, *indent=1*)

_AskYes_NoPass (*question*, *indent=1*)

Ask a yes/NO/pass question.

_Ask_YesNoPass (*question*, *indent=1*)

Ask a YES/no/pass question.

_PrintAvailableEditions (*editions*, *selectedEdition*)

Print all available editions and return the selected index.

_TryLog (**args*, *condition=True*, ***kwargs*)

class ToolChain.**Configurator** (*host: Base.IHost*)

A instance of this class controls the interactive configuration process.

Inheritance

Members

ConfigureAll ()

Select all tool chains for configuration

ConfigureTool (*toolChain*)

Select tool chains for configuration.

InitializeConfiguration ()

Initialize PoC’s configuration with empty sections.

The list of sections is gathered from all enabled configurators’ `_template` fields.

UpdateConfiguration ()

Update an existing configuration e.g. after a PoC update.

_ConfigureTools (*configurators*)

Run the configuration routines for a list of configurators

_ConfigurationLoop (*configurator*)

Retry to configure a vendor or tool until it succeeds or the user presses P to pass a configuration step.

A **:py:exec:‘KeyboardInterrupt’** should be handled in a calling method.

ConfigureDefaultTools ()

_ConfigureDefaultTools ()

`_WriteConfigurationHeader()`

Write a header containing general information about the configuration and list allowed input values for yes/no/pass questions.

`_AskConfigureDefaultTools()`

Ask if default tools should be configured now.

`Relocated()`

`Log(entry, condition=True)`

Write an entry to the local logger.

`LogDebug(*args, condition=True, **kwargs)`

`LogDryRun(*args, condition=True, **kwargs)`

`LogError(*args, condition=True, **kwargs)`

`LogFatal(*args, condition=True, **kwargs)`

`LogInfo(*args, condition=True, **kwargs)`

`LogNormal(*args, condition=True, **kwargs)`

`LogQuiet(*args, condition=True, **kwargs)`

`LogVerbose(*args, condition=True, **kwargs)`

`LogWarning(*args, condition=True, **kwargs)`

`Logger`

Return the local logger instance.

`_Ask(question, default, beforeDefault="", afterDefault="", indent=1)`

`_AskYes_NoPass(question, indent=1)`

Ask a yes/NO/pass question.

`_Ask_YesNoPass(question, indent=1)`

Ask a YES/no/pass question.

`_PrintAvailableEditions(editions, selectedEdition)`

Print all available editions and return the selected index.

`_TryLog(*args, condition=True, **kwargs)`

14.8 lib

Submodules

14.8.1 lib.CodeDOM

Classes

- *CodeDOMMeta*: type(object_or_name, bases, dict)
- *CodeDOMObject*: Undocumented.
- *Expression*: Undocumented.
- *UnaryExpression*: Undocumented.
- *NotExpression*: Undocumented.
- *BinaryExpression*: Undocumented.
- *LogicalExpression*: Undocumented.
- *CompareExpression*: Undocumented.

- *EqualExpression*: Undocumented.
- *UnequalExpression*: Undocumented.
- *LessThanExpression*: Undocumented.
- *LessThanEqualExpression*: Undocumented.
- *GreaterThanExpression*: Undocumented.
- *GreaterThanEqualExpression*: Undocumented.
- *AndExpression*: Undocumented.
- *OrExpression*: Undocumented.
- *XorExpression*: Undocumented.
- *InExpression*: Undocumented.
- *NotInExpression*: Undocumented.
- *Function*: Undocumented.
- *ListElement*: Undocumented.
- *Literal*: Undocumented.
- *StringLiteral*: Undocumented.
- *IntegerLiteral*: Undocumented.
- *Identifier*: Undocumented.
- *Statement*: Undocumented.
- *BlockStatement*: Undocumented.
- *ConditionalBlockStatement*: Undocumented.
- *EmptyLine*: Undocumented.
- *CommentLine*: Undocumented.
- *BlockedStatement*: Undocumented.
- *ExpressionChoice*: Undocumented.

class lib.CodeDOM.**CodeDOMMeta**

Inheritance

Members

parse()

static **GetChoiceParser**()

static **GetRepeatParser**(*generator*)

mro() → list

return a type's method resolution order

class lib.CodeDOM.**CodeDOMObject**

Inheritance

Members

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.Expression**

Inheritance

Members

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.UnaryExpression** (*child*)

Inheritance

Members

Child

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.NotExpression** (*child*)

Inheritance

Members

classmethod **GetParser** ()

Child

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.BinaryExpression** (*leftChild*, *rightChild*)

Inheritance

Members

LeftChild

RightChild

classmethod **GetParser** ()

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.LogicalExpression** (*leftChild*, *rightChild*)

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.CompareExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.EqualExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.UnequalExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.LessThanExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.LessThanEqualExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.GreaterThanExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.GreaterThanEqualExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.AndExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.OrExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.XorExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.InExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser ()`

`LeftChild`

`classmethod Parse (string, printChar)`

`RightChild`

`class lib.CodeDOM.NotInExpression (leftChild, rightChild)`

Inheritance

Members

`classmethod GetParser()`

`LeftChild`

`classmethod Parse(string, printChar)`

`RightChild`

`class lib.CodeDOM.Function`

Inheritance

Members

`classmethod Parse(string, printChar)`

`class lib.CodeDOM.ListElement`

Inheritance

Members

`classmethod GetParser()`

`classmethod Parse(string, printChar)`

`class lib.CodeDOM.Literal`

Inheritance

Members

`classmethod Parse(string, printChar)`

`class lib.CodeDOM.StringLiteral(value)`

Inheritance

Members

`Value`

`classmethod GetParser()`

`classmethod Parse(string, printChar)`

`class lib.CodeDOM.IntegerLiteral(value)`

Inheritance

Members

Value

classmethod **GetParser** ()

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.Identifier** (*name*)

Inheritance

Members

Name

classmethod **GetParser** ()

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.Statement** (*commentText*=")

Inheritance

Members

CommentText

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.BlockStatement** (*commentText*=")

Inheritance

Members

AddStatement (*stmt*)

Statements

CommentText

classmethod **Parse** (*string*, *printChar*)

class **lib.CodeDOM.ConditionalBlockStatement** (*expression*, *commentText*=")

Inheritance

Members

Expression

AddStatement (*stmt*)

CommentText

classmethod **Parse** (*string*, *printChar*)

Statements

```
class lib.CodeDOM.EmptyLine
```

Inheritance

Members

```
classmethod GetParser ()
```

```
classmethod Parse (string, printChar)
```

```
class lib.CodeDOM.CommentLine (commentText)
```

Inheritance

Members

Text

```
classmethod GetParser ()
```

```
classmethod Parse (string, printChar)
```

```
class lib.CodeDOM.BlockedStatement
```

Inheritance

Members

```
_allowedStatements = []
```

```
classmethod AddChoice (value)
```

```
classmethod GetParser ()
```

```
classmethod Parse (string, printChar)
```

```
class lib.CodeDOM.ExpressionChoice
```

Inheritance

Members

```
_allowedExpressions = []
```

```
classmethod AddChoice (value)
```

```
classmethod GetParser ()
```

```
classmethod Parse (string, printChar)
```

14.8.2 lib.Decorators

Classes

- *MethodAlias*: `MethodAlias` creates a local method, which is an alias to another method
- *ILazyLoadable*: Undocumented.

- *LazyLoadTrigger*: Undocumented.
- *CachedReadOnlyProperty*: Undocumented.

class lib.Decorators.**MethodAlias** (*method*)

MethodAlias creates a local method, which is an alias to another method local or inherited method.

Inheritance

Members

__init__ (*method*)

Initialize self. See help(type(self)) for accurate signature.

__call__ (*func*)

Call self as a function.

class lib.Decorators.**ILazyLoadable**

Inheritance

Members

_LazyLoadable_Load ()

LazyLoadable_IsLoaded

class lib.Decorators.**LazyLoadTrigger** (*func*)

Inheritance

Members

class lib.Decorators.**CachedReadOnlyProperty** (*func*)

Inheritance

Members

14.8.3 lib.ExtendedConfigParser

Classes

- *ExtendedSectionProxy*: A proxy for a single section from a parser.
- *ExtendedInterpolation*: Dummy interpolation that passes the value through with no changes.
- *ExtendedConfigParser*: ConfigParser implementing interpolation.

class lib.ExtendedConfigParser.**ExtendedSectionProxy** (*parser, name*)

Inheritance

Members

_MutableMapping__marker = <object object>


```

_abc_cache = <_weakrefset.WeakSet object>
_abc_negative_cache = <_weakrefset.WeakSet object>
_abc_negative_cache_version = 42
_abc_registry = <_weakrefset.WeakSet object>
_options ()
clear () → None. Remove all items from D.
get (option, fallback=None, *, raw=False, vars=None, _impl=None, **kwargs)
    Get an option value.

    Unless fallback is provided, None will be returned if the option is not found.
items () → a set-like object providing a view on D's items
keys () → a set-like object providing a view on D's keys
name
parser
pop (k[, d]) → v, remove specified key and return the corresponding value.
    If key is not found, d is returned if given, otherwise KeyError is raised.
popitem () → (k, v), remove and return some (key, value) pair
    as a 2-tuple; but raise KeyError if D is empty.
setdefault (k[, d]) → D.get(k,d), also set D[k]=d if k not in D
update ([E], **F) → None. Update D from mapping/iterable E and F.
    If E present and has a .keys() method, does: for k in E: D[k] = E[k] If E present and lacks .keys()
    method, does: for (k, v) in E: D[k] = v In either case, this is followed by: for k, v in F.items(): D[k] =
    v
values () → an object providing a view on D's values

```

class lib.ExtendedConfigParser.**ExtendedInterpolation**

Inheritance

Members

```

_KEYCRE = re.compile('\\$\\{ (?P<ref>[^\}]+) \\}')
_KEYCRE2 = re.compile('\\$\\[ (?P<ref>[^\]]+ ) \\}')
clear_cache ()
before_get (parser, section, option, value, defaults)
before_set (parser, section, option, value)
interpolate (parser, section, option, value, _, depth=0)
static GetSpecial (option, path)
GetValue (parser, section, option, path)
GetCached (section, option)
UpdateCache (section, option, value)
before_read (parser, section, option, value)
before_write (parser, section, option, value)

```

```
class lib.ExtendedConfigParser.ExtendedConfigParser (defaults=None,
                                                    dict_type=<class 'col-
                                                    lections.OrderedDict'>,
                                                    allow_no_value=False, *,
                                                    delimiters=('=', ':'), com-
                                                    ment_prefixes=(';', ';'), in-
                                                    line_comment_prefixes=None,
                                                    strict=True,
                                                    empty_lines_in_values=True,
                                                    de-
                                                    fault_section='DEFAULT',
                                                    interpolation=<object ob-
                                                    ject>, converters=<object
                                                    object>)
```

Inheritance

Members

```
_DEFAULT_INTERPOLATION = <lib.ExtendedConfigParser.ExtendedInterpolation object>
BOOLEAN_STATES = {'0': False, '1': True, 'false': False, 'no': False, 'off': Fa
NONSPACECRE = re.compile('\\S')
OPTCRE = re.compile('\n (?P<option>.*?) # very permissive!\n \\s*(?P<vi>=|:)\n \\s* # a
OPTCRE_NV = re.compile('\n (?P<option>.*?) # very permissive!\n \\s*(?: # any numbe
SECTCRE = re.compile('\n \\[ # [\n (?P<header>[^]]+) # very permissive!\n \\] # ]\n
_MutableMapping__marker = <object object>
_OPT_NV_TMPL = '\n (?P<option>.*?) # very permissive!\n \\s*(?: # any number of sp
_OPT_TMPL = '\n (?P<option>.*?) # very permissive!\n \\s*(?P<vi>{delim})\n \\s* # any n
_SECT_TMPL = '\n \\[ # [\n (?P<header>[^]]+) # very permissive!\n \\] # ]\n '
_abc_cache = <_weakrefset.WeakSet object>
_abc_negative_cache = <_weakrefset.WeakSet object>
_abc_negative_cache_version = 42
_abc_registry = <_weakrefset.WeakSet object>
_convert_to_boolean (value)
    Return a boolean value translating from other types if necessary.
_get (section, conv, option, **kwargs)
_get_conv (section, option, conv, *, raw=False, vars=None, fallback=<object object>, **kwargs)
_handle_error (exc, fpname, lineno, line)
_join_multiline_values ()
_read (fp, fpname)
    Parse a sectioned configuration file.

    Each section in a configuration file contains a header, indicated by a name in square brackets ('[]'),
    plus key/value options, indicated by 'name' and 'value' delimited with a specific substring ('=' or ':'
    by default).

    Values can span multiple lines, as long as they are indented deeper than the first line of the value.
    Depending on the parser's mode, blank lines may be treated as parts of multiline values or ignored.
```

Configuration files may include comments, prefixed by specific characters ('#' and ';' by default). Comments may appear on their own in an otherwise empty line or may be entered in lines holding values or section names.

_validate_value_types (*, section="", option="", value="")

Raises a `TypeError` for non-string values.

The only legal non-string value if we allow valueless options is `None`, so we need to check if the value is a string if: - we do not allow valueless options, or - we allow valueless options but the value is not `None`

For compatibility reasons this method is not used in classic `set()` for `RawConfigParsers`. It is invoked in every case for mapping protocol access and in `ConfigParser.set()`.

_write_section (fp, section_name, section_items, delimiter)

Write a single section to the specified 'fp'.

add_section (section)

Create a new section in the configuration. Extends `RawConfigParser.add_section` by validating if the section name is a string.

clear () → `None`. Remove all items from D.

converters

defaults ()

get (section, option, *, raw=False, vars=None, fallback=<object object>)

Get an option value for a given section.

If 'vars' is provided, it must be a dictionary. The option is looked up in 'vars' (if provided), 'section', and in 'DEFAULTSECT' in that order. If the key is not found and 'fallback' is provided, it is used as a fallback value. 'None' can be provided as a 'fallback' value.

If interpolation is enabled and the optional argument 'raw' is `False`, all interpolations are expanded in the return values.

Arguments 'raw', 'vars', and 'fallback' are keyword only.

The section `DEFAULT` is special.

getboolean (section, option, *, raw=False, vars=None, fallback=<object object>, **kwargs)

getfloat (section, option, *, raw=False, vars=None, fallback=<object object>, **kwargs)

getint (section, option, *, raw=False, vars=None, fallback=<object object>, **kwargs)

has_section (section)

Indicate whether the named section is present in the configuration.

The `DEFAULT` section is not acknowledged.

items (section=<object object>, raw=False, vars=None)

Return a list of (name, value) tuples for each option in a section.

All % interpolations are expanded in the return values, based on the defaults passed into the constructor, unless the optional argument 'raw' is `true`. Additional substitutions may be provided using the 'vars' argument, which must be a dictionary whose contents overrides any pre-existing defaults.

The section `DEFAULT` is special.

keys () → a set-like object providing a view on D's keys

options (section)

Return a list of option names for the given section name.

optionxform (optionstr)

pop (k[, d]) → v, remove specified key and return the corresponding value.

If key is not found, d is returned if given, otherwise `KeyError` is raised.

popitem()

Remove a section from the parser and return it as a (section_name, section_proxy) tuple. If no section is present, raise KeyError.

The section DEFAULT is never returned because it cannot be removed.

read(filenames, encoding=None)

Read and parse a filename or a list of filenames.

Files that cannot be opened are silently ignored; this is designed so that you can specify a list of potential configuration file locations (e.g. current directory, user's home directory, systemwide directory), and all existing configuration files in the list will be read. A single filename may also be given.

Return list of successfully read files.

read_dict(dictionary, source='<dict>')

Read configuration from a dictionary.

Keys are section names, values are dictionaries with keys and values that should be present in the section. If the used dictionary type preserves order, sections and their keys will be added in order.

All types held in the dictionary are converted to strings during reading, including section names, option names and keys.

Optional second argument is the 'source' specifying the name of the dictionary being read.

read_file(f, source=None)

Like read() but the argument must be a file-like object.

The 'f' argument must be iterable, returning one line at a time. Optional second argument is the 'source' specifying the name of the file being read. If not given, it is taken from f.name. If 'f' has no 'name' attribute, '<??>' is used.

read_string(string, source='<string>')

Read configuration from a given string.

readfp(fp, filename=None)

Deprecated, use read_file instead.

remove_option(section, option)

Remove an option.

remove_section(section)

Remove a file section.

sections()

Return a list of section names, excluding [DEFAULT]

set(section, option, value=None)

Set an option. Extends RawConfigParser.set by validating type and interpolation syntax on the value.

setdefault(k[, d]) → D.get(k,d), also set D[k]=d if k not in D

update([E], **F) → None. Update D from mapping/iterable E and F.

If E present and has a .keys() method, does: for k in E: D[k] = E[k] If E present and lacks .keys() method, does: for (k, v) in E: D[k] = v In either case, this is followed by: for k, v in F.items(): D[k] = v

values() → an object providing a view on D's values

write(fp, space_around_delimiters=True)

Write an .ini-format representation of the configuration state.

If 'space_around_delimiters' is True (the default), delimiters between keys and values are surrounded by spaces.

Interpolation

`_unify_values` (*section, variables*)

Create a sequence of lookups with ‘variables’ taking priority over the ‘section’ which takes priority over the DEFAULTSECT.

`has_option` (*section, option*)

Check for the existence of a given option in a given section. If the specified *section* is None or an empty string, DEFAULT is assumed. If the specified *section* does not exist, returns False.

14.8.4 lib.Functions

Classes

- `CallByRefParam`: Implements a “call by reference” parameter.
- `Init`: Undocumented.
- `Exit`: Undocumented.

Functions

- `merge()`: Merge 2 or more dictionaries.
- `merge_with()`: Merge 2 or more dictionaries. Apply function f to each element during merge.

`class lib.Functions.CallByRefParam` (*value=None*)

Implements a “call by reference” parameter.

See also:

`CallByRefBoolParam` A special “call by reference” implementation for boolean reference types.

`CallByRefIntParam` A special “call by reference” implementation for integer reference types.

Inheritance

Members

`class lib.Functions.Init`

Inheritance

Members

`classmethod init()`

`Foreground` = {'BLUE': '\x1b[94m', 'CYAN': '\x1b[96m', 'DARK_CYAN': '\x1b[36m', 'DARK_GRAY': '\x1b[90m', 'DARK_RED': '\x1b[91m', 'DARK_GREEN': '\x1b[92m', 'DARK_BLUE': '\x1b[93m', 'DARK_MAGENTA': '\x1b[95m', 'DARK_YELLOW': '\x1b[97m', 'DARK_WHITE': '\x1b[98m', 'GRAY': '\x1b[37m', 'RED': '\x1b[31m', 'GREEN': '\x1b[32m', 'BLUE': '\x1b[34m', 'MAGENTA': '\x1b[35m', 'YELLOW': '\x1b[33m', 'WHITE': '\x1b[30m', 'BLACK': '\x1b[30m'}

`class lib.Functions.Exit`

Inheritance

Members

`classmethod exit` (*returnCode=0*)

`classmethod versionCheck` (*version*)

`classmethod printException` (*ex*)

`classmethod printNotImplementedError` (*ex*)

```
classmethod printExceptionBase (ex)
classmethod printPlatformNotSupportedException (ex)
classmethod printEnvironmentException (ex)
classmethod printNotConfiguredException (ex)
```

Functions

```
lib.Functions.merge (*dicts)
    Merge 2 or more dictionaries.

lib.Functions.merge_with (f, *dicts)
    Merge 2 or more dictionaries. Apply function f to each element during merge.
```

14.8.5 lib.Parser

Exceptions

- *ParserException*: Common base class for all non-exit exceptions.
- *MismatchingParserResult*: Signal the end from iterator.__next__().
- *EmptyChoiseParserResult*: Signal the end from iterator.__next__().
- *MatchingParserResult*: Signal the end from iterator.__next__().
- *GreedyMatchingParserResult*: Signal the end from iterator.__next__().

Classes

- *SourceCodePosition*: Undocumented.
- *Token*: Undocumented.
- *SuperToken*: Undocumented.
- *ValuedToken*: Undocumented.
- *StartOfDocumentToken*: Undocumented.
- *CharacterToken*: Undocumented.
- *SpaceToken*: Undocumented.
- *DelimiterToken*: Undocumented.
- *NumberToken*: Undocumented.
- *StringToken*: Undocumented.
- *Tokenizer*: Undocumented.

```
exception lib.Parser.ParserException
```

Inheritance

Members

args

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

```
exception lib.Parser.MismatchingParserResult
```

Inheritance

Members

args

value

generator return value

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `lib.Parser.EmptyChoiseParserResult`

Inheritance

Members

args

value

generator return value

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `lib.Parser.MatchingParserResult`

Inheritance

Members

args

value

generator return value

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `lib.Parser.GreedyMatchingParserResult`

Inheritance

Members

args

value

generator return value

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

class `lib.Parser.SourceCodePosition` (*row, column, absolute*)

Inheritance

Members

```
class lib.Parser.Token (previousToken, start, end=None)
```

Inheritance

Members

PreviousToken

Length

```
class lib.Parser.SuperToken (startToken, endToken=None)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.ValuedToken (previousToken, value, start, end=None)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.StartOfDocumentToken
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.CharacterToken (previousToken, value, start)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.SpaceToken (previousToken, value, start, end=None)
```


Inheritance

Members

Length

PreviousToken

```
class lib.Parser.DelimiterToken (previousToken, value, start, end=None)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.NumberToken (previousToken, value, start, end=None)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.StringToken (previousToken, value, start, end=None)
```

Inheritance

Members

Length

PreviousToken

```
class lib.Parser.Tokenizer
```

Inheritance

Members

```
class TokenKind
```

An enumeration.

SpaceChars = 0

AlphaChars = 1

NumberChars = 2

DelimiterChars = 3

OtherChars = 4

```
static GetCharacterTokenizer()
```

```
classmethod GetWordTokenizer (iterable, alphaCharacters='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ',
                             numberCharacters='0123456789', whiteSpaceCharacters='\t')
```

14.8.6 lib.SphinxExtensions

Classes

- *DocumentMemberAttribute*: Undocumented.

```
class lib.SphinxExtensions.DocumentMemberAttribute (value=True)
```

Inheritance

Members

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
_debug = False
```

14.8.7 lib.pyAttribute

Submodules

lib.pyAttribute.ArgParseAttributes

Classes

- *CommandGroupAttribute*: Undocumented.
- *DefaultAttribute*: Undocumented.
- *CommandAttribute*: Undocumented.
- *ArgumentAttribute*: Undocumented.
- *SwitchArgumentAttribute*: Undocumented.
- *CommonArgumentAttribute*: Undocumented.
- *CommonSwitchArgumentAttribute*: Undocumented.
- *ArgParseMixin*: Undocumented.

```
class lib.pyAttribute.ArgParseAttributes.CommandGroupAttribute (groupName)
```

Inheritance

Members

GroupName

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
```

```
__CommandGroupAttribute__groupName = ''
__debug = False
class lib.pyAttribute.ArgParseAttributes.DefaultAttribute
```

Inheritance

Members

Handler

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
__DefaultAttribute__handler = None
__debug = False
class lib.pyAttribute.ArgParseAttributes.CommandAttribute (command,
                                                         **kwargs)
```

Inheritance

Members

Command

Handler

KWArgs

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
__CommandAttribute__command = ''
__CommandAttribute__handler = None
__CommandAttribute__kwargs = None
__debug = False
class lib.pyAttribute.ArgParseAttributes.ArgumentAttribute (*args, **kwargs)
```

Inheritance

Members

Args

KWArgs

```
classmethod GetAttributes (method)
classmethod GetMethods (cl)
static _AppendAttribute (attribute)
__ArgumentAttribute__args = None
```

```
    _ArgumentAttribute__kwargs = None
    _debug = False
class lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute(*args,
                                                                **kwargs)
```

Inheritance

Members

Args

`classmethod GetAttributes (method)`

`classmethod GetMethods (cl)`

KWArgs

`static _AppendAttribute (attribute)`

`_ArgumentAttribute__args = None`

`_ArgumentAttribute__kwargs = None`

`_debug = False`

```
class lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute(*args,
                                                                **kwargs)
```

Inheritance

Members

Args

`classmethod GetAttributes (method)`

`classmethod GetMethods (cl)`

KWArgs

`static _AppendAttribute (attribute)`

`_ArgumentAttribute__args = None`

`_ArgumentAttribute__kwargs = None`

`_debug = False`

```
class lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute(*args,
                                                                **kwargs)
```

Inheritance

Members

Args

`classmethod GetAttributes (method)`

`classmethod GetMethods (cl)`

KWArgs

```
    static _AppendAttribute (attribute)
    _ArgumentAttribute__args = None
    _ArgumentAttribute__kwargs = None
    _debug = False
class lib.pyAttribute.ArgParseAttributes.ArgParseMixin (**kwargs)
```

Inheritance

Members

```
static GetAttributes ()
GetMethods ()
static HasAttribute ()
_ArgParseMixin__mainParser = None
_ArgParseMixin__subParser = None
_ArgParseMixin__subParsers = {}
Run ()
MainParser
SubParsers
```

Classes

- *Attribute*: Undocumented.
- *AttributeHelperMixin*: Undocumented.

```
class lib.pyAttribute.Attribute
```

Inheritance

Members

```
_debug = False
static _AppendAttribute (attribute)
classmethod GetMethods (cl)
classmethod GetAttributes (method)
class lib.pyAttribute.AttributeHelperMixin
```

Inheritance

Members

```
GetMethods ()
static HasAttribute ()
static GetAttributes ()
```


15.1 List of Supported FPGA Devices

Vendor	Family	Device Name
Altera	Max	Max-II, Max 10
	Cyclone	Cyclone III, Cyclone V
	Stratix	Stratix II, Stratix IV, Stratix V, Stratix 10
	Arria	Arria II, Arria V
Lattice	Mach	MachXO
	ECP	ECP3, ECP5
Xilinx	Coolrunner	Coolrunner-II
	Spartan	Spartan-3, Spartan-6
	Artix	Artix-7
	Kintex	Kintex-7, Kintex UltraScale, Kintex UltraScale+
	Virtex	Virtex-II, Virtex-4, Virtex-5, Virtex-7, Virtex UltraScale, Virtex UltraScale+
	Zynq	Zynq-7000

15.2 List of Supported Boards

Board Name	Device String	Device Name
GENERIC	GENERIC	Generic board and device
Altera	⇒ DE4	
DE0	EP3C16F484	Altera Cyclone III
S2GXAV	EP2SGX90FF1508C3	Altera Stratix II
DE4	EP4SGX230KF40C2	Altera Stratix IV
DE5	EP5SGXEA7N2F45C2	Altera Stratix V
Lattice	⇒ ECP5Versa	
ECP5Versa	LFE5UM-45F-6BG381C	Lattice ECP5
Xilinx	⇒ KC705	
S3SK200	XC3S200FT256	Xilinx Spartan-3
S3ESK500	XC3S500EFT256	Xilinx Spartan-3
S3SK1000	XC3S1000FT256	Xilinx Spartan-3
S3ESK1600	XC3S1600EFT256	Xilinx Spartan-3
ATLYS	XC6SLX45-3CSG324	Xilinx Spartan-6
ZC706	XC7Z045-2FFG900	Xilinx Zynq-7000
ZedBoard	XC7Z020-1CLG484	Xilinx Zynq-7000
AC701	XC7A200T-2FBG676C	Xilinx Artix-7
KC705	XC7K325T-2FFG900C	Xilinx Kintex-7
ML505	XC5VLX50T-1FF1136	Xilinx Virtex-5
ML506	XC5VSX50T-1FFG1136	Xilinx Virtex-5
ML507	XC5VFX70T-1FFG1136	Xilinx Virtex-5
XUPV5	XC5VLX110T-1FF1136	Xilinx Virtex-5
ML605	XC6VLX240T-1FF1156	Xilinx Virtex-6
VC707	XC7VX485T-2FFG1761C	Xilinx Virtex-7
VC709	XC7VX690T-2FFG1761C	Xilinx Virtex-7
Custom	<any device>	

15.3 Wrapper Script Hook Files

The shell scripts `poc.ps1` and `poc.sh` can be customized though hook files, which are executed before and after a PoC command is executed. The wrapper scripts support 4 kinds of hook files:

- VendorPreHookFile
- ToolPreHookFile
- VendorPostHookFile
- ToolPostHookFile

The wrapper scans the arguments given to the front-end script and searches for known commands. If one is found, the hook files are scheduled before and after the execution of the wrapped executable. The hook files are sourced into the current execution and need to be located in the `./py/Wrapper/Hooks` directory.

A common use case is the preparation of special vendor or tool chain environments. For example many EDA tools are using FlexLM as a license manager, which needs the environments variable `LM_LICENSE_FILE` to be set. A `PreHookFile` can be used to load/export such an environment variable.

15.3.1 Examples

Mentor QuestaSim on Linux:

The PoC infrastructure is called with this command line:


```
./poc.sh -v vsim PoC.arith.prng
```

The `vsim` command is recognized and the following events are scheduled:

1. `source ./py/Wrapper/Hooks/Mentor.pre.sh`
2. `source ./py/Wrapper/Hooks/Mentor.QuestaSim.pre.sh`
3. `Execute ./py/PoC.py -v vsim PoC.arith.prng`
4. `source ./py/Wrapper/Hooks/Mentor.QuestaSim.post.sh`
5. `source ./py/Wrapper/Hooks/Mentor.post.sh`

If a hook files doesn't exist, it's skipped.

Mentor QuestaSim on Windows:

The PoC infrastructure is called with this command line:

```
.\poc.ps1 -v vsim PoC.arith.prng
```

The `vsim` command is recognized and the following events are scheduled:

1. `.\py\Wrapper\Hooks\Mentor.pre.ps1`
2. `.\py\Wrapper\Hooks\Mentor.QuestaSim.pre.ps1`
3. `Execute .\py\PoC.py -v vsim PoC.arith.prng`
4. `.\py\Wrapper\Hooks\Mentor.QuestaSim.post.ps1`
5. `.\py\Wrapper\Hooks\Mentor.post.ps1`

If a hook files doesn't exist, it's skipped.

15.3.2 FlexLM

Many EDA tools require an environment variable called `LM_LICENSE_FILE`. If no other tool settings are required, a common `FlexLM.sh` can be generated. This file is used as a symlink target for each tool specific hook file.

Content of the 'FlexLM.sh' script:

```
export LM_LICENSE_FILE=1234@flexlm.company.com
```

Create symlinks:

```
ln -s FlexLM.sh Altera.Quartus.pre.sh
ln -s FlexLM.sh Mentor.QuestaSim.pre.sh
```

15.4 File Formats

15.4.1 *.ini Format

Document rule:

DocumentLine rule:

Section rule:

OptionLine rule:

FQSectionName rule:

15.4.2 *.files Format

Contents of this Page

- *Document*
- *Source File Statements*
- *Conditional Statements*
- *Boolean Expressions*
 - *Unary operators*
 - *Binary operators*
 - *Literals*
 - *Pre-defined constants*
- *Path Expressions*

Files files are used to ...

Line comments start with #.

Document

Source File Statements

Bla VHDLStatement blub

- `vhdl Library "<VHDLFile>"` This statement references a VHDL source file.
- `verilog "<VerilogFile>"` This statement references a Verilog source file.
- `cocotb "<PythonFile>"` This statement references a Cocotb testbench file (Python file).
- `ucf "<UCFFile>"` This statement references a Xilinx User Constraint File (UCF).
- `sdc "<SDCFile>"` This statement references a Synopsys Design Constraint file (SDC).
- `xdc "<XDCFile>"` This statement references a Xilinx Design Constraint file (XDC).
- `ldc "<LDCFile>"` This statement references a Lattice Design Constraint file (LDC).

Conditional Statements

- `If (<Expression>) Then ... [ElseIf (<Expression>) Then ...][Else ...]`
`End IF` This allows the user to define conditions, when to load a source file into the file list. The `ElseIf` and `Else` clause of an `If` statement are optional.

Boolean Expressions

Unary operators

- `!` - not
- `[...]` - list construction
- `?` - file exists

Binary operators

- `and` - and
- `or` - or
- `xor` - exclusive or
- `in` - in list
- `=` - equal
- `!=` - unequal
- `<` - less than
- `<=` - less than or equal
- `>` - greater than
- `>=` - greater than or equal

Literals

- `<constant>` - a pre-defined constant
- `"<String>"` - Strings are enclosed in quote signs
- `<Integer>` - Integers as decimal values

Pre-defined constants

- Environment Variables:
 - **Environment** Values:
 - * `"Simulation"`
 - * `"Synthesis"`
 - `ToolChain` - The used tool chain. E.g. `"Xilinx_ISE"`
 - `Tool` - The used tool. E.g. `"Mentor_QuestaSim"` or `"Xilinx_XST"`
 - `VHDL` - The used VHDL version. 1987, 1993, 2002, 2008
- Board Variables:
 - `BoardName` - A string. E.g. `"KC705"`
- Device Variables:
 - `DeviceVendor` - The vendor of the device. E.g. `"Altera"`
 - `DeviceDevice` -
 - `DeviceFamily` -
 - `DeviceGeneration` -
 - `DeviceSeries` -

Path Expressions

- / - sub-directory
- & - string concat

Other Statements

- `include "<FilesFile>"` Include another *.files file.
- `library <VHDLLibrary> "<LibraryPath>"` Reference an existing (pre-compiled) VHDL library, which is passed to the simulator, if external libraries are supported.
- `report "<Message>"` Print a critical warning in the log window. This critical warning is treated as an error.

15.4.3 *.rules Format

Contents of this Page

- **.rules Format*
 - *Headline 1*
 - *Headline 2*
 - *Headline 3*

Headline 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

Headline 2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

Headline 3

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet

15.5 Naming Conventions

Todo: Write an introduction paragraph for this page.

15.5.1 Root Directory Overview (PoCRoot)

The PoC-Library is structured into several sub-directories, naming the purpose of the directory like `src` for sources files or `tb` for testbench files. The structure within these directories is most likely the same and based on PoC's *sub-namespace tree*. PoC's installation directory is also referred to as `PoCRoot`.

- **lib** Third party libraries like Coctb, OSVVM or VUnit are shipped in this folder. The external library is stored in a sub directory named like the library. If a library is available as a Git submodule, then it is linked as a submodule for better version tracking.
- **netlist** This is the output directory for pre-configured netlists, synthesized by PoC. Netlists and related constraint files are the result of IP core synthesis flows, either from PoC's source files or from vendor specific IP core files like *.xco files from Xilinx Core Generator. Generated IP cores are stored in device sub-directories, because most netlists formats are device specific. For example the IP core `PoC.arith.prng` created from source file `src\arith\arith_prng.vhdl` generated for a Kintex-7 325T mounted on a KC705 board will be copied to `netlist\XC7K325T-2FFG900\arith\arith_prng.ngc` if Xilinx ISE XST is used for synthesis.
- **py** The supporting Python infrastructure, the configuration files and the IP core 'database' is stored in this directory.
- **sim** Some of PoC's testbenches are shipped with pre-configured waveform views/ waveform configuration files for selected simulators or waveform viewers. If a testbench is launched in GUI mode (`--gui`) and a waveform view for the chosen simulator is found, it's loaded as the default view.
- **src** The source files of PoC's IP cores are stored in this directory. The IP cores are grouped by their sub-namespace into sub-directories according to the *sub-namespace tree*. See the paragraph below, for how IP cores are named and how PoC core names map to the sub-namespace hierarchy and the resulting sub-namespace directory structure.
- **tb** PoC is shipped with testbenches. All testbenches are categorized and stored in sub-directories like the IP core, which is tested.
- **tbl** Supporting Tcl files.
- **temp** A pre-created temporary directors for various tool's intermediate outputs. In case of errors in a used vendor tool or in PoC's infrastructure, this directory contains intermediate files, log files and report files, which can be used to analyze the error.
- **tools** This directory contains miscellaneous files or scripts for external tools like emacs, git or text editor syntax highlighting files.
- **ucf** Pre-configured constraint files (*.ucf, *.xdc, *.sdc) for many FPGA boards, containing physical (pin, placement) and timing constraints.
- **xst** Configuration files to synthesize PoC modules with Xilinx XST into a netlist.

15.5.2 Namespaces and Modules

Namespaces

PoC uses namespaces and sub-namespaces to categorize all VHDL and Verilog modules. Despite VHDL doesn't support sub-namespaces yet, PoC already uses sub-namespaces enforced by a strict naming schema.

Rules: 1. Namespace names are lower-case, underscore free, valid VHDL identifiers. 2. A namespace name is unique, but can be part of a entity name.

Module Names

Module names are prefixed with its parents namespace name. A module name can contain underscores to denote implementation variants of a module.

Rules: 3. Modul names are valid VHDL identifiers prefixed with its parent namespace's name. 4. The first part of module name must not contain the parents namespace name.

Example 1 - PoC.fifo.cc_got

For example a FIFO module with a common clock interface and a *got* semantic is named `PoC.fifo.cc_got` (fully qualified name). This name can be split at every dot and underscore sign, resulting in the following table of name parts:

PoC	fifo	cc	got
Root Namespace	Sub-Namespace	Common Clock Interface	Got Semantic

Because `PoC.fifo.cc_got` refers to an IP core, the source file is located in the `<PoCRoot>\src` directory. The (sub-)namespace of the PoC entity is `fifo`, so it's stored in the sub-directory `fifo`. The file name `cc_got` FIFO is prefixed with the last sub-namespace: In this case `fifo_`. This is summarized in the following table:

Property	Value
Fully Qualified Name	PoC.fifo.cc_got
VHDL entity name	fifo_cc_got
File name	fifo_cc_got.vhdl
IP Core Description File	\src\fifo\fifo_cc_got.files
Source File Location	\src\fifo\fifo_cc_got.vhdl
Testbench Location	\tb\fifo\fifo_cc_got_tb.vhdl
Testbench Description File	\tb\fifo\fifo_cc_got_tb.files
Waveform Description Files	\sim\fifo\fifo_cc_got_tb.*

Other implementation variants are:

- `_dc` – dependent clock / related clock
- `_ic` – independent clock / cross clock
- `_got_tempgot` – got interface extended by a temporary got interface
- `_got_tempput` – got interface extended by a temporary put interface

Example 2 - PoC.mem.ocram.tdp

PoC	mem	ocram	tdp
Root Namespace	Sub-Namespace	Sub-Namespace	True-Dual-Port

Property	Value
Fully Qualified Name	PoC.mem.ocram.tdp
VHDL entity name	ocram_tdp
File name	ocram_tdp.vhdl
IP Core Description File	\src\mem\ocram\ocram_tdp.files
Source File Location	\src\mem\ocram\ocram_tdp.vhdl
Testbench Location	\tb\mem\ocram\ocram_tdp_tb.vhdl
Testbench Description File	\tb\mem\ocram\ocram_tdp_tb.files
Waveform Description Files	\sim\mem\ocram\ocram_tdp_tb.*

Note: Not all sub-namespace parts are include as a prefix in the name, only the last one.

15.5.3 Signal Names

Todo: No documentation available.

15.6 Known Issues

15.6.1 General

Synthesis of tri-state signals

Tri-state signals should be only used when they are connected (through the hierarchy) to top-level bidirectional or output pins.

Descriptions which infer a tri-state driver like:

```
pin <= data when tri = '0' else 'Z';
```

should not be included in any IP core description because these hinder or even inhibit block-based design flows. If a netlist is generated from such an IP core, the netlist may contain only a simple internal (on-chip) tri-state buffer instead of the correct tri-state I/O block primitive because I/O buffers are not automatically added for netlist generation. If the netlist is then used in another design, the mapper, e.g. Xilinx ISE Map, may fail to merge the internal tri-state buffer of the IP core netlist with the I/O buffer automatically created for the top-level netlist. This failing behavior is not considered as a tool bug.

Thus, if tri-state drivers should be included in an IP core, then the IP core description must instantiate the appropriate I/O block primitive of the target architecture like it is done by the Xilinx MIG.

Synthesis of bidirectional records

Records are useful to group several signals of an IP core interface. But the corresponding port of this record type should not be of mode `inout` to pass data in both direction. This restriction holds even if a record member will be driven only by one source in the real hardware and even if all the drivers (one for each record member) are visible to the current synthesis run. The following observations have been made:

- An IP core (entity or procedure) must drive all record members with value 'Z' which are only used as an input in the IP core. If this is missed, then the respective record member will be driven by 'U' and the effective value after resolution will be 'U' as well, see IEEE Std. 1076-2008 para. 12.6.1. Thus simulation will fail.

But these 'Z' drivers will flood the RTL / Netlist view of Altera Quartus-II, Intel Quartus Prime and Lattice Diamond with always tri-stated drivers and make this view unusable.

Note: Simulation with ModelSim shows correct output even when the ‘Z’ driver is missing, but a warning is reported that the behavior is not VHDL Standard compliant.

- Altera Quartus-II and Intel Quartus Prime report warnings about this meaningless ‘Z’ drivers. Synthesis result is as expected if each record member is only driven by one source in real hardware.
- The synthesis result of the Lattice Synthesis Engine (3.7.0 / 3.8.0) is not optimal. It seems that the synthesizer tries to implement the internal (on-chip) tristate bus using AND-OR logic but failed to optimize it away because there was only one real source. Test case was a simple SRAM controller which used the record type `T_IO_TRISTATE` to bring-out the data-bus so that the tri-state driver could be instantiated on the top-level.

Use separate records for the input and output data flow instead.

15.6.2 Aldec Active-HDL

- Aliases to functions and protected type methods

15.6.3 Altera Quartus-II / Intel Quartus Prime

- Generic types of type strings filled with NUL

15.6.4 GHDL

- Aliases to protected type methods

15.6.5 Xilinx ISE

- Shared Variables in Simulation (VHDL-93)

15.6.6 Xilinx Vivado

- Physical types in synthesis
- VHDL-2008 mode in simulation
- Shared variables in simulation (VHDL-93 and VHDL-2008))

15.7 Local License Copies

This documentation contains local copies of all used licenses by either PoC itself or embedded libraries or IP cores. Each formatted¹ license text can be downloaded from the original source. Therefor see the top-most INFO box, which contains a link to the original license file source.

Note: This is a local copy of the [Apache License Version 2.0](#).

¹ Formatted means, that reStructured Text (reST) markup is used to highlight headlines, to display lists as unordered or ordered lists and to emphasis special wording in bold. There is no change in words, spelling or paragraph numbering. The formatting is needed to allow a proper rendering with Sphinx and to create a correct toctree (Table of Content Tree).

15.7.1 Apache License 2.0

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **“control”** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or **“Your”**) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **“submitted”** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **“Not a Contribution.”**

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those

patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Appendix: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “[]” replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```



15.7.2 Modified Apache Contributor License Agreement v2.0

Thank you for your interest in the **Chair for VLSI Design, Diagnostics and Architecture** - Faculty of Computer Science, Technische Universität Dresden, Germany (the “Chair”). In order to clarify the intellectual property license granted with Contributions from any person or entity, the Chair must have a Contributor License Agreement (“CLA”) on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Chair and its users; it does not change your rights to use your own Contributions for any other purpose.

The following CLA is an adaption of the [Apache Contributor License Agreement v2.0](#)

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Chair. In return, the Chair shall not use Your Contributions in a way that is contrary to the public benefit or

inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Chair and recipients of software distributed by the Chair, You reserve all right, title, and interest in and to Your Contributions.

1. **Definitions.** “*You*” (or “*Your*”) shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Chair. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, “*control*” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“*Contribution*” shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Chair for inclusion in, or documentation of, any of the products owned or managed by the Chair (the “*Work*”). For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Chair or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Chair for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “**Not a Contribution.**”

2. **Grant of Copyright License.** Subject to the terms and conditions of this Agreement, You hereby grant to the Chair and to recipients of software distributed by the Chair a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.
3. **Grant of Patent License.** Subject to the terms and conditions of this Agreement, You hereby grant to the Chair and to recipients of software distributed by the Chair a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.
4. You represent that you are legally entitled to grant the above license. If your employer(s) has rights to intellectual property that you create that includes your Contributions, you represent that you have received permission to make Contributions on behalf of that employer, that your employer has waived such rights for your Contributions to the Chair, or that your employer has executed a separate *Corporate CLA* with the Chair.
5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others). You represent that Your Contribution submissions include complete details of any third-party license or other restriction (including, but not limited to, related patents and trademarks) of which you are personally aware and which are associated with any part of Your Contributions.
6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON- INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.
7. Should You wish to submit work that is not Your original creation, You may submit it to the Chair separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as “**Submitted on behalf of a third-party: [named here]**”.
8. You agree to notify the Chair of any facts or circumstances of which you become aware that would make these representations inaccurate in any respect.

15.7.3 Modified Apache Corporate Contributor License Agreement v2.0

Thank you for your interest in the **Chair for VLSI Design, Diagnostics and Architecture** - Faculty of Computer Science, Technische Universität Dresden, Germany (the “Chair”). In order to clarify the intellectual property license granted with Contributions from any person or entity, the Chair must have a Contributor License Agreement (CLA) on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Chair and its users; it does not change your rights to use your own Contributions for any other purpose.

This version of the Agreement allows an entity (the “Corporation”) to submit Contributions to the Chair, to authorize Contributions submitted by its designated employees to the Chair, and to grant copyright and patent licenses thereto.

The following CLA is an adaption of the [Apache Corporate Contributor License Agreement v2.0](#)

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Chair. In return, the Chair shall not use Your Contributions in a way that is contrary to the public benefit or inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Chair and recipients of software distributed by the Chair, You reserve all right, title, and interest in and to Your Contributions.

1. **Definitions.** “*You*” (or “*Your*”) shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Chair. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“*Contribution*” shall mean the code, documentation or other original works of authorship expressly identified in Schedule B, as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Chair for inclusion in, or documentation of, any of the products owned or managed by the Chair (the “*Work*”). For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Chair or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Chair for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “**Not a Contribution.**”

2. **Grant of Copyright License.** Subject to the terms and conditions of this Agreement, You hereby grant to the Chair and to recipients of software distributed by the Chair a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.
3. **Grant of Patent License.** Subject to the terms and conditions of this Agreement, You hereby grant to the Chair and to recipients of software distributed by the Chair a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) were submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that You are legally entitled to grant the above license. You represent further that each employee of the Corporation designated on Schedule A below (or in a subsequent written modification to that Schedule) is authorized to submit Contributions on behalf of the Corporation.
5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others).
6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.
7. Should You wish to submit work that is not Your original creation, You may submit it to the Chair separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as “**Submitted on behalf of a third-party: [named here]**”.
8. It is your responsibility to notify the Chair when any change is required to the list of designated employees authorized to submit Contributions on behalf of the Corporation, or to the Corporation’s Point of Contact with the Chair.

Note: This is a local copy of the [Artistic License 2.0](#).

15.7.4 Artistic License 2.0

Copyright © 2000-2006, The Perl Foundation.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

This license establishes the terms under which a given free software Package may be copied, modified, distributed, and/or redistributed. The intent is that the Copyright Holder maintains some artistic control over the development of that Package while still keeping the Package available as open source and free software. You are always permitted to make arrangements wholly outside of this license directly with the Copyright Holder of a given Package. If the terms of this license do not permit the full use that you propose to make of the Package, you should contact the Copyright Holder and seek a different licensing arrangement.

Definitions

“**Copyright Holder**” means the individual(s) or organization(s) named in the copyright notice for the entire Package.

“**Contributor**” means any party that has contributed code or other material to the Package, in accordance with the Copyright Holder’s procedures.

“**You**” and “**your**” means any person who would like to copy, distribute, or modify the Package.

“**Package**” means the collection of files distributed by the Copyright Holder, and derivatives of that collection and/or of those files. A given Package may consist of either the Standard Version, or a Modified Version.

“**Distribute**” means providing a copy of the Package or making it accessible to anyone else, or in the case of a company or organization, to others outside of your company or organization.

“Distributor Fee” means any fee that you charge for Distributing this Package or providing support for this Package to another party. It does not mean licensing fees.

“Standard Version” refers to the Package if it has not been modified, or has been modified only in ways explicitly requested by the Copyright Holder.

“Modified Version” means the Package, if it has been changed, and such changes were not explicitly requested by the Copyright Holder.

“Original License” means this Artistic License as Distributed with the Standard Version of the Package, in its current version or as it may be modified by The Perl Foundation in the future.

“Source” form means the source code, documentation source, and configuration files for the Package.

“Compiled” form means the compiled bytecode, object code, binary, or any other form resulting from mechanical transformation or translation of the Source form.

Permission for Use and Modification Without Distribution

(1) You are permitted to use the Standard Version and create and use Modified Versions for any purpose without restriction, provided that you do not Distribute the Modified Version.

Permissions for Redistribution of the Standard Version

(2) You may Distribute verbatim copies of the Source form of the Standard Version of this Package in any medium without restriction, either gratis or for a Distributor Fee, provided that you duplicate all of the original copyright notices and associated disclaimers. At your discretion, such verbatim copies may or may not include a Compiled form of the Package.

(3) You may apply any bug fixes, portability changes, and other modifications made available from the Copyright Holder. The resulting Package will still be considered the Standard Version, and as such will be subject to the Original License.

Distribution of Modified Versions of the Package as Source

(4) You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) provided that you clearly document how it differs from the Standard Version, including, but not limited to, documenting any non-standard features, executables, or modules, and provided that you do at least ONE of the following:

- (a) make the Modified Version available to the Copyright Holder of the Standard Version, under the Original License, so that the Copyright Holder may include your modifications in the Standard Version.
- (b) ensure that installation of your Modified Version does not prevent the user installing or running the Standard Version. In addition, the Modified Version must bear a name that is different from the name of the Standard Version.
- (c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under (i) the Original License or (ii) a license that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

Distribution of Compiled Forms of the Standard Version or Modified Versions without the Source

(5) You may Distribute Compiled forms of the Standard Version without the Source, provided that you include complete instructions on how to get the Source of the Standard Version. Such instructions must be valid at the time of your distribution. If these instructions, at any time while you are carrying out such distribution, become invalid, you must provide new instructions on demand or cease further distribution. If you provide valid instructions or

cease distribution within thirty days after you become aware that the instructions are invalid, then you do not forfeit any of your rights under this license.

(6) You may Distribute a Modified Version in Compiled form without the Source, provided that you comply with Section 4 with respect to the Source of the Modified Version.

Aggregating or Linking the Package

(7) You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.

(8) You are permitted to link Modified and Standard Versions with other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

Items That are Not Considered Part of a Modified Version

(9) Works (including, but not limited to, modules and scripts) that merely extend or make use of the Package, do not, by themselves, cause the Package to be a Modified Version. In addition, such works are not considered parts of the Package itself, and are not subject to the terms of this license.

General Provisions

(10) Any use, modification, and distribution of the Standard or Modified Versions is governed by this Artistic License. By using, modifying or distributing the Package, you accept this license. Do not use, modify, or distribute the Package, if you do not accept this license.

(11) If your Modified Version has been derived from a Modified Version made by someone other than you, you are nevertheless required to ensure that your Modified Version complies with the requirements of this license.

(12) This license does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.

(13) This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.

14. Disclaimer of Warranty:

THE PACKAGE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT ARE DISCLAIMED TO THE EXTENT PERMITTED BY YOUR LOCAL LAW. UNLESS REQUIRED BY LAW, NO COPYRIGHT HOLDER OR CONTRIBUTOR WILL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Note: This is a local copy of the (Revised) BSD License used in the Cocotb project. The original can be found in file LICENSE in the Cocotb source tree.

Todo: Check link to the lib/cocotb/LICENSE file.

15.7.5 BSD License for Cocotb

Cocotb is licensed under the Revised BSD License. Full license text below.

Copyright © 2013 Potential Ventures Ltd Copyright © 2013 SolarFlare Communications Inc All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Potential Ventures Ltd, SolarFlare Communications Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL POTENTIAL VENTURES LTD BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Note: This is a local copy of [The MIT License \(MIT\)](#) used in the UVVM library. The original can be found in file LICENSE in the UVVM_All source tree.

Todo: Check link to the lib/uvvm/LICENSE file.

15.7.6 The MIT License (MIT)

Copyright © 2016 by Bitvis AS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note: This is a local copy of the [Mozilla Public License, Version 2.0](#).

15.7.7 Mozilla Public License, v. 2.0

1. Definitions

- 1.1. “Contributor”** means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.
- 1.2. “Contributor Version”** means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.
- 1.3. “Contribution”** means Covered Software of a particular Contributor.
- 1.4. “Covered Software”** means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.
- 1.5. “Incompatible With Secondary Licenses”** means
- that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
 - that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.
- 1.6. “Executable Form”** means any form of the work other than Source Code Form.
- 1.7. “Larger Work”** means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.
- 1.8. “License”** means this document.
- 1.9. “Licensable”** means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.
- 1.10. “Modifications”** means any of the following:
- any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
 - any new file in Source Code Form that contains any Covered Software.
- 1.11. “Patent Claims” of a Contributor** means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.
- 1.12. “Secondary License”** means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.
- 1.13. “Source Code Form”** means the form of the work preferred for making modifications.
- 1.14. “You” (or “Your”)** means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

- under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- for any code that a Contributor has removed from Covered Software; or
- for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable

means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Part IV

Appendix

CHAPTER 16

Change Log

CHAPTER 17

Index

b

Base, 241
Base.Exceptions, 241
Base.Executable, 243
Base.Logging, 255
Base.Project, 257
Base.Shared, 265

c

Compiler, 267
Compiler.ISECompiler, 267
Compiler.LSECompiler, 269
Compiler.QuartusCompiler, 271
Compiler.VivadoCompiler, 272
Compiler.XCICompiler, 274
Compiler.XCOCompiler, 276
Compiler.XSTCompiler, 278

d

DataBase, 283
DataBase.Config, 284
DataBase.Entity, 291
DataBase.Solution, 305
DataBase.TestCase, 311

l

lib, 497
lib.CodeDOM, 497
lib.Decorators, 505
lib.ExtendedConfigParser, 506
lib.Functions, 511
lib.Parser, 512
lib.pyAttribute, 516
lib.pyAttribute.ArgParseAttributes, 516
lib.SphinxExtensions, 516

p

Parser, 316
Parser.FilesCodeDOM, 316
Parser.FilesParser, 324
Parser.RulesCodeDOM, 326
Parser.RulesParser, 330
PoC, 237

s

Simulator, 332
Simulator.ActiveHDL Simulator, 332
Simulator.Cocotb Simulator, 333
Simulator.GHDL Simulator, 335
Simulator.ISE Simulator, 337
Simulator.ModelSim Simulator, 338
Simulator.Questa Simulator, 340
Simulator.RivieraPRO Simulator, 341
Simulator.Vivado Simulator, 341

t

ToolChain, 347
ToolChain.Aldec, 347
ToolChain.Aldec.ActiveHDL, 347
ToolChain.Aldec.RivieraPRO, 354
ToolChain.Altera, 363
ToolChain.Altera.ModelSim, 364
ToolChain.Altera.Quartus, 372
ToolChain.GHDL, 381
ToolChain.Git, 402
ToolChain.GNU, 397
ToolChain.GTKWave, 398
ToolChain.Intel, 411
ToolChain.Intel.ModelSim, 411
ToolChain.Intel.Quartus, 419
ToolChain.Lattice, 425
ToolChain.Lattice.ActiveHDL, 425
ToolChain.Lattice.Diamond, 425
ToolChain.Lattice.Synplify, 430
ToolChain.Mentor, 435
ToolChain.Mentor.ModelSim, 435
ToolChain.Mentor.QuestaSim, 454
ToolChain.PoC, 459
ToolChain.Synopsys, 462
ToolChain.Windows, 464
ToolChain.Xilinx, 466
ToolChain.Xilinx.ISE, 466
ToolChain.Xilinx.Vivado, 475

Symbols

–all

compile-altera.sh command line option, 221
 compile-lattice.sh command line option, 222
 compile-osvvm.sh command line option, 224
 compile-uvvm.sh command line option, 225
 compile-xilinx-ise.sh command line option, 227
 compile-xilinx-vivado.sh command line option, 229

–clean

compile-altera.sh command line option, 221
 compile-lattice.sh command line option, 222
 compile-osvvm.sh command line option, 224
 compile-uvvm.sh command line option, 225
 compile-xilinx-ise.sh command line option, 227
 compile-xilinx-vivado.sh command line option, 229

–ghdl

compile-altera.sh command line option, 221
 compile-lattice.sh command line option, 222
 compile-osvvm.sh command line option, 224
 compile-uvvm.sh command line option, 226
 compile-xilinx-ise.sh command line option, 227
 compile-xilinx-vivado.sh command line option, 229

–help

compile-altera.sh command line option, 221
 compile-lattice.sh command line option, 222
 compile-osvvm.sh command line option, 224
 compile-uvvm.sh command line option, 225
 compile-xilinx-ise.sh command line option, 227
 compile-xilinx-vivado.sh command line option, 229

–questa

compile-altera.sh command line option, 221
 compile-lattice.sh command line option, 222
 compile-osvvm.sh command line option, 224
 compile-uvvm.sh command line option, 226
 compile-xilinx-ise.sh command line option, 227
 compile-xilinx-vivado.sh command line option, 229

–vhdl2008

compile-altera.sh command line option, 221

compile-lattice.sh command line option, 223

compile-osvvm.sh command line option, 224

compile-uvvm.sh command line option, 226

compile-xilinx-ise.sh command line option, 227

compile-xilinx-vivado.sh command line option, 229

–vhdl93

compile-altera.sh command line option, 221

compile-lattice.sh command line option, 223

compile-osvvm.sh command line option, 224

compile-uvvm.sh command line option, 226

compile-xilinx-ise.sh command line option, 227

compile-xilinx-vivado.sh command line option, 229

–All

compile-altera.ps1 command line option, 221

compile-lattice.ps1 command line option, 223

compile-osvvm.ps1 command line option, 225

compile-uvvm.ps1 command line option, 226

compile-xilinx-ise.ps1 command line option, 228

compile-xilinx-vivado.ps1 command line option, 230

–Clean

compile-altera.ps1 command line option, 221

compile-lattice.ps1 command line option, 223

compile-osvvm.ps1 command line option, 225

compile-uvvm.ps1 command line option, 226

compile-xilinx-ise.ps1 command line option, 228

compile-xilinx-vivado.ps1 command line option, 230

–D

poc.ps1 command line option, 219

poc.sh command line option, 220

–GHDL

compile-altera.ps1 command line option, 222

compile-lattice.ps1 command line option, 223

compile-osvvm.ps1 command line option, 225

compile-uvvm.ps1 command line option, 226

compile-xilinx-ise.ps1 command line option, 228

compile-xilinx-vivado.ps1 command line option, 230

–Help

compile-altera.ps1 command line option, 221

compile-lattice.ps1 command line option, 223

- compile-osvvm.ps1 command line option, [225](#)
- compile-uvvm.ps1 command line option, [226](#)
- compile-xilinx-ise.ps1 command line option, [228](#)
- compile-xilinx-vivado.ps1 command line option, [230](#)
- Questa
 - compile-altera.ps1 command line option, [222](#)
 - compile-lattice.ps1 command line option, [223](#)
 - compile-osvvm.ps1 command line option, [225](#)
 - compile-uvvm.ps1 command line option, [226](#)
 - compile-xilinx-ise.ps1 command line option, [228](#)
 - compile-xilinx-vivado.ps1 command line option, [230](#)
- ReLink
 - compile-xilinx-ise.ps1 command line option, [228](#)
 - compile-xilinx-vivado.ps1 command line option, [230](#)
- VHDL2008
 - compile-altera.ps1 command line option, [222](#)
 - compile-lattice.ps1 command line option, [223](#)
 - compile-osvvm.ps1 command line option, [225](#)
 - compile-uvvm.ps1 command line option, [226](#)
 - compile-xilinx-ise.ps1 command line option, [228](#)
 - compile-xilinx-vivado.ps1 command line option, [230](#)
- VHDL93
 - compile-altera.ps1 command line option, [222](#)
 - compile-lattice.ps1 command line option, [223](#)
 - compile-osvvm.ps1 command line option, [225](#)
 - compile-uvvm.ps1 command line option, [226](#)
 - compile-xilinx-ise.ps1 command line option, [228](#)
 - compile-xilinx-vivado.ps1 command line option, [230](#)
- _AddFileListFile() (Base.Shared.Shared method), [266](#)
- _AddFileListFile() (Compiler.Compiler method), [283](#)
- _AddFileListFile() (Compiler.ISECompiler.Compiler method), [268](#)
- _AddFileListFile() (Compiler.LSECompiler.Compiler method), [270](#)
- _AddFileListFile() (Compiler.QuartusCompiler.Compiler method), [272](#)
- _AddFileListFile() (Compiler.VivadoCompiler.Compiler method), [273](#)
- _AddFileListFile() (Compiler.XCICompiler.Compiler method), [275](#)
- _AddFileListFile() (Compiler.XCOCCompiler.Compiler method), [277](#)
- _AddFileListFile() (Compiler.XSTCompiler.Compiler method), [279](#)
- _AddFileListFile() (Simulator.ActiveHDL Simulator.Simulator method), [333](#)
- _AddFileListFile() (Simulator.Cocotb Simulator.Simulator method), [334](#)
- _AddFileListFile() (Simulator.GHDL Simulator.Simulator method), [336](#)
- _AddFileListFile() (Simulator.ISE Simulator.Simulator method), [338](#)
- _AddFileListFile() (Simulator.ModelSim Simulator.Simulator method), [339](#)
- _AddFileListFile() (Simulator.Questa Simulator.Simulator method), [341](#)
- _AddFileListFile() (Simulator.Simulator method), [346](#)
- _AddFileListFile() (Simulator.Vivado Simulator.Simulator method), [342](#)
- _AddRulesFiles() (Compiler.Compiler method), [282](#)
- _AddRulesFiles() (Compiler.ISECompiler.Compiler method), [268](#)
- _AddRulesFiles() (Compiler.LSECompiler.Compiler method), [270](#)
- _AddRulesFiles() (Compiler.QuartusCompiler.Compiler method), [272](#)
- _AddRulesFiles() (Compiler.VivadoCompiler.Compiler method), [273](#)
- _AddRulesFiles() (Compiler.XCICompiler.Compiler method), [275](#)
- _AddRulesFiles() (Compiler.XCOCCompiler.Compiler method), [277](#)
- _AddRulesFiles() (Compiler.XSTCompiler.Compiler method), [279](#)
- _AlteraEditionConfiguration__editionName (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), [368](#)
- _AlteraStarterEditionConfiguration__editionName (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), [371](#)
- _AppendAttribute() (lib.SphinxExtensions.DocumentMemberAttribute static method), [516](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute static method), [517](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.CommandAttribute static method), [517](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.CommandGroup static method), [516](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.CommonArgument static method), [518](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.CommonSwitch static method), [518](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.DefaultAttribute static method), [517](#)
- _AppendAttribute() (lib.pyAttribute.ArgParseAttributes.SwitchArgument static method), [518](#)
- _AppendAttribute() (lib.pyAttribute.Attribute static method), [519](#)
- _ArgParseMixin__mainParser (lib.pyAttribute.ArgParseAttributes.ArgParseMixin attribute), [519](#)

[_ArgParseMixin__subParser](#)
 (lib.pyAttribute.ArgParseAttributes.ArgParseMixin
 attribute), 519

[_ArgParseMixin__subParsers](#)
 (lib.pyAttribute.ArgParseAttributes.ArgParseMixin
 attribute), 519

[_ArgumentAttribute__args](#)
 (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute
 attribute), 517

[_ArgumentAttribute__args](#)
 (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute
 attribute), 518

[_ArgumentAttribute__args](#)
 (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute
 attribute), 519

[_ArgumentAttribute__args](#)
 (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute
 attribute), 518

[_ArgumentAttribute__kwargs](#)
 (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute
 attribute), 517

[_ArgumentAttribute__kwargs](#)
 (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute
 attribute), 518

[_ArgumentAttribute__kwargs](#)
 (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute
 attribute), 519

[_ArgumentAttribute__kwargs](#)
 (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute
 attribute), 518

[_Ask\(\)](#) (ToolChain.Aldec.ActiveHDL.Configuration
 method), 350

[_Ask\(\)](#) (ToolChain.Aldec.Configuration method), 363

[_Ask\(\)](#) (ToolChain.Aldec.RivieraPRO.Configuration
 method), 356

[_Ask\(\)](#) (ToolChain.Altera.Configuration method), 380

[_Ask\(\)](#) (ToolChain.Altera.ModelSim.AlteraEditionConfiguration
 method), 368

[_Ask\(\)](#) (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration
 method), 371

[_Ask\(\)](#) (ToolChain.Altera.ModelSim.Configuration
 method), 366

[_Ask\(\)](#) (ToolChain.Altera.Quartus.Configuration
 method), 374

[_Ask\(\)](#) (ToolChain.AskMixIn method), 489

[_Ask\(\)](#) (ToolChain.Configuration method), 490

[_Ask\(\)](#) (ToolChain.Configurator method), 497

[_Ask\(\)](#) (ToolChain.GHDL.Configuration method), 383

[_Ask\(\)](#) (ToolChain.GTKWave.Configuration method),
 400

[_Ask\(\)](#) (ToolChain.Git.Configuration method), 404

[_Ask\(\)](#) (ToolChain.Intel.Configuration method), 425

[_Ask\(\)](#) (ToolChain.Intel.ModelSim.Configuration
 method), 413

[_Ask\(\)](#) (ToolChain.Intel.ModelSim.IntelEditionConfiguration
 method), 415

[_Ask\(\)](#) (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration
 method), 418

[_Ask\(\)](#) (ToolChain.Intel.Quartus.Configuration
 method), 421

[_Ask\(\)](#) (ToolChain.Lattice.Configuration method), 434

[_Ask\(\)](#) (ToolChain.Lattice.Diamond.Configuration
 method), 427

[_Ask\(\)](#) (ToolChain.Lattice.Synplify.Configuration
 method), 431

[_Ask\(\)](#) (ToolChain.Mentor.Configuration method), 459

[_Ask\(\)](#) (ToolChain.Mentor.ModelSim.Configuration
 method), 438

[_Ask\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration
 method), 440

[_Ask\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration
 method), 443

[_Ask\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration
 method), 445

[_Ask\(\)](#) (ToolChain.Mentor.ModelSim.Selector
 method), 446

[_Ask\(\)](#) (ToolChain.Mentor.QuestaSim.Configuration
 method), 456

[_Ask\(\)](#) (ToolChain.PoC.Configuration method), 461

[_Ask\(\)](#) (ToolChain.Synopsys.Configuration method),
 466

[_Ask\(\)](#) (ToolChain.ToolConfiguration method), 494

[_Ask\(\)](#) (ToolChain.ToolSelector method), 496

[_Ask\(\)](#) (ToolChain.Vivado.Configuration method), 492

[_Ask\(\)](#) (ToolChain.Xilinx.Configuration method), 486

[_Ask\(\)](#) (ToolChain.Xilinx.ISE.Configuration method),
 486

[_Ask\(\)](#) (ToolChain.Xilinx.Vivado.Configuration
 method), 477

[_AskConfigureDefaultTools\(\)](#) (ToolChain.Configurator
 method), 497

[_AskInstalled\(\)](#) (ToolChain.Aldec.ActiveHDL.Configuration
 method), 350

[_AskInstalled\(\)](#) (ToolChain.Aldec.Configuration
 method), 363

[_AskInstalled\(\)](#) (ToolChain.Aldec.RivieraPRO.Configuration
 method), 356

[_AskInstalled\(\)](#) (ToolChain.Altera.Configuration
 method), 380

[_AskInstalled\(\)](#) (ToolChain.Altera.ModelSim.AlteraEditionConfiguration
 method), 368

[_AskInstalled\(\)](#) (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration
 method), 371

[_AskInstalled\(\)](#) (ToolChain.Altera.ModelSim.Configuration
 method), 366

[_AskInstalled\(\)](#) (ToolChain.Altera.Quartus.Configuration
 method), 374

[_AskInstalled\(\)](#) (ToolChain.Configuration method),
 490

[_AskInstalled\(\)](#) (ToolChain.GHDL.Configuration
 method), 383

[_AskInstalled\(\)](#) (ToolChain.GTKWave.Configuration
 method), 400

[_AskInstalled\(\)](#) (ToolChain.Git.Configuration method),
 404

[_AskInstalled\(\)](#) (ToolChain.Intel.Configuration
 method), 425

method), 425

_AskInstalled() (ToolChain.Intel.ModelSim.Configuration method), 413

_AskInstalled() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415

_AskInstalled() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418

_AskInstalled() (ToolChain.Intel.Quartus.Configuration method), 421

_AskInstalled() (ToolChain.Lattice.Configuration method), 434

_AskInstalled() (ToolChain.Lattice.Diamond.Configuration method), 427

_AskInstalled() (ToolChain.Lattice.Synplify.Configuration method), 431

_AskInstalled() (ToolChain.Mentor.Configuration method), 459

_AskInstalled() (ToolChain.Mentor.ModelSim.Configuration method), 438

_AskInstalled() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440

_AskInstalled() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443

_AskInstalled() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445

_AskInstalled() (ToolChain.Mentor.QuestaSim.Configuration method), 456

_AskInstalled() (ToolChain.PoC.Configuration method), 461

_AskInstalled() (ToolChain.Synopsys.Configuration method), 464

_AskInstalled() (ToolChain.ToolConfiguration method), 494

_AskInstalled() (ToolChain.VendorConfiguration method), 492

_AskInstalled() (ToolChain.Xilinx.Configuration method), 486

_AskInstalled() (ToolChain.Xilinx.ISE.Configuration method), 468

_AskInstalled() (ToolChain.Xilinx.Vivado.Configuration method), 477

_AskSelection() (ToolChain.Mentor.ModelSim.Selector method), 446

_AskSelection() (ToolChain.ToolSelector method), 495

_AskYes_NoPass() (ToolChain.Aldec.ActiveHDL.Configuration method), 350

_AskYes_NoPass() (ToolChain.Aldec.Configuration method), 363

_AskYes_NoPass() (ToolChain.Aldec.RivieraPRO.Configuration method), 357

_AskYes_NoPass() (ToolChain.Altera.Configuration method), 380

_AskYes_NoPass() (ToolChain.Altera.ModelSim.AlderiaEditionConfiguration method), 368

_AskYes_NoPass() (ToolChain.Altera.ModelSim.AlderiaStarterEditionConfiguration method), 371

_AskYes_NoPass() (ToolChain.Altera.ModelSim.Configuration method), 366

_AskYes_NoPass() (ToolChain.Altera.Quartus.Configuration method), 374

_AskYes_NoPass() (ToolChain.AskMixIn method), 489

_AskYes_NoPass() (ToolChain.Configuration method), 490

_AskYes_NoPass() (ToolChain.Configurator method), 497

_AskYes_NoPass() (ToolChain.GHDL.Configuration method), 383

_AskYes_NoPass() (ToolChain.GTKWave.Configuration method), 400

_AskYes_NoPass() (ToolChain.Git.Configuration method), 404

_AskYes_NoPass() (ToolChain.Intel.Configuration method), 425

_AskYes_NoPass() (ToolChain.Intel.ModelSim.Configuration method), 413

_AskYes_NoPass() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416

_AskYes_NoPass() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418

_AskYes_NoPass() (ToolChain.Intel.Quartus.Configuration method), 421

_AskYes_NoPass() (ToolChain.Lattice.Configuration method), 434

_AskYes_NoPass() (ToolChain.Lattice.Diamond.Configuration method), 427

_AskYes_NoPass() (ToolChain.Lattice.Synplify.Configuration method), 432

_AskYes_NoPass() (ToolChain.Mentor.Configuration method), 459

_AskYes_NoPass() (ToolChain.Mentor.ModelSim.Configuration method), 438

_AskYes_NoPass() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 441

_AskYes_NoPass() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443

_AskYes_NoPass() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445

_AskYes_NoPass() (ToolChain.Mentor.QuestaSim.Configuration method), 456

_AskYes_NoPass() (ToolChain.PoC.Configuration method), 461

_AskYes_NoPass() (ToolChain.Synopsys.Configuration method), 464

_AskYes_NoPass() (ToolChain.ToolConfiguration method), 494

_AskYes_NoPass() (ToolChain.ToolSelector method), 496

_AskYes_NoPass() (ToolChain.VendorConfiguration method), 492

_AskYes_NoPass() (ToolChain.Xilinx.Configuration method), 486

_AskYes_NoPass() (ToolChain.Xilinx.ISE.Configuration method), 468

[_AskYes_NoPass\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)
[_AskYes_NoPass\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 350](#)
[_AskYes_NoPass\(\) \(ToolChain.Aldec.Configuration method\), 363](#)
[_AskYes_NoPass\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 357](#)
[_AskYes_NoPass\(\) \(ToolChain.Altera.Configuration method\), 380](#)
[_AskYes_NoPass\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method\), 368](#)
[_AskYes_NoPass\(\) \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method\), 371](#)
[_AskYes_NoPass\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 366](#)
[_AskYes_NoPass\(\) \(ToolChain.Altera.Quartus.Configuration method\), 374](#)
[_AskYes_NoPass\(\) \(ToolChain.AskMixIn method\), 489](#)
[_AskYes_NoPass\(\) \(ToolChain.Configuration method\), 490](#)
[_AskYes_NoPass\(\) \(ToolChain.Configurator method\), 497](#)
[_AskYes_NoPass\(\) \(ToolChain.GHDL.Configuration method\), 383](#)
[_AskYes_NoPass\(\) \(ToolChain.GTKWave.Configuration method\), 400](#)
[_AskYes_NoPass\(\) \(ToolChain.Git.Configuration method\), 404](#)
[_AskYes_NoPass\(\) \(ToolChain.Intel.Configuration method\), 425](#)
[_AskYes_NoPass\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[_AskYes_NoPass\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 416](#)
[_AskYes_NoPass\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 418](#)
[_AskYes_NoPass\(\) \(ToolChain.Intel.Quartus.Configuration method\), 421](#)
[_AskYes_NoPass\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[_AskYes_NoPass\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 427](#)
[_AskYes_NoPass\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 432](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.Configuration method\), 459](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 438](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 441](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 443](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 445](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.ModelSim.Selector method\), 447](#)
[_AskYes_NoPass\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[_AskYes_NoPass\(\) \(ToolChain.PoC.Configuration method\), 461](#)
[_AskYes_NoPass\(\) \(ToolChain.Synopsys.Configuration method\), 464](#)
[_AskYes_NoPass\(\) \(ToolChain.ToolConfiguration method\), 494](#)
[_AskYes_NoPass\(\) \(ToolChain.ToolSelector method\), 496](#)
[_AskYes_NoPass\(\) \(ToolChain.VendorConfiguration method\), 492](#)
[_AskYes_NoPass\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[_AskYes_NoPass\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 468](#)
[_AskYes_NoPass\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)
[_CheckModelSimVersion\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method\), 368](#)
[_CheckModelSimVersion\(\) \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method\), 371](#)
[_CheckModelSimVersion\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 366](#)
[_CheckModelSimVersion\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[_CheckModelSimVersion\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 416](#)
[_CheckModelSimVersion\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 418](#)
[_CheckModelSimVersion\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 437](#)
[_CheckModelSimVersion\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 441](#)
[_CheckModelSimVersion\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 443](#)
[_CheckModelSimVersion\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 445](#)
[_CheckModelSimVersion\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[_CheckQuestaSimVersion\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 455](#)
[_ShowModelSimConfiguration command \(lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute\), 517](#)
[_CommandAttribute__handler](#)

(lib.pyAttribute.ArgParseAttributes.CommandAttribute), 517	__Configuration__IsUnderGitControl() (ToolChain.Git.Configuration method), 404
__CommandAttribute__kwargs (lib.pyAttribute.ArgParseAttributes.CommandAttribute), 517	__Configuration__UninstallGitFilters() (ToolChain.Git.Configuration method), 404
__CommandGroupAttribute__groupName (lib.pyAttribute.ArgParseAttributes.CommandGroupAttribute), 516	__Configuration__UninstallGitHooks() (ToolChain.Git.Configuration method), 404
__ConfigurationLoop() (ToolChain.Configurator method), 496	__Configuration__WriteGHDLSection() (ToolChain.GHDL.Configuration method), 384
__Configuration__CheckActiveHDLVersion() (ToolChain.Aldec.ActiveHDL.Configuration method), 350	__Configuration__WriteGitSection() (ToolChain.Git.Configuration method), 404
__Configuration__CheckDiamondVersion() (ToolChain.Lattice.Diamond.Configuration method), 427	__Configuration__WriteGtkWaveSection() (ToolChain.GTKWave.Configuration method), 400
__Configuration__CheckISEVersion() (ToolChain.Xilinx.ISE.Configuration method), 468	__ConfigureBinaryDirectory() (ToolChain.Aldec.ActiveHDL.Configuration method), 350
__Configuration__CheckQuartusVersion() (ToolChain.Altera.Quartus.Configuration method), 374	__ConfigureBinaryDirectory() (ToolChain.Aldec.RivieraPRO.Configuration method), 357
__Configuration__CheckQuartusVersion() (ToolChain.Intel.Quartus.Configuration method), 421	__ConfigureBinaryDirectory() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
__Configuration__CheckRivieraPROVersion() (ToolChain.Aldec.RivieraPRO.Configuration method), 357	__ConfigureBinaryDirectory() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371
__Configuration__CheckVivadoVersion() (ToolChain.Xilinx.Vivado.Configuration method), 477	__ConfigureBinaryDirectory() (ToolChain.Altera.ModelSim.Configuration method), 366
__Configuration__GetGitDirectory() (ToolChain.Git.Configuration method), 404	__ConfigureBinaryDirectory() (ToolChain.Altera.Quartus.Configuration method), 374
__Configuration__GetModelSimVersion() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368	__ConfigureBinaryDirectory() (ToolChain.GHDL.Configuration method), 384
__Configuration__GetModelSimVersion() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371	__ConfigureBinaryDirectory() (ToolChain.GTKWave.Configuration method), 400
__Configuration__GetModelSimVersion() (ToolChain.Altera.ModelSim.Configuration method), 366	__ConfigureBinaryDirectory() (ToolChain.Git.Configuration method), 404
__Configuration__GetModelSimVersion() (ToolChain.Intel.ModelSim.Configuration method), 413	__ConfigureBinaryDirectory() (ToolChain.Intel.ModelSim.Configuration method), 413
__Configuration__GetModelSimVersion() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416	__ConfigureBinaryDirectory() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416
__Configuration__GetModelSimVersion() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418	__ConfigureBinaryDirectory() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
__Configuration__InstallGitFilters() (ToolChain.Git.Configuration method), 404	__ConfigureBinaryDirectory() (ToolChain.Intel.Quartus.Configuration method), 421
__Configuration__InstallGitHooks() (ToolChain.Git.Configuration method), 404	__ConfigureBinaryDirectory()

(ToolChain.Lattice.Diamond.Configuration method), 426
 _ConfigureBinaryDirectory() (ToolChain.Lattice.Synplify.Configuration method), 432
 _ConfigureBinaryDirectory() (ToolChain.Mentor.ModelSim.Configuration method), 438
 _ConfigureBinaryDirectory() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 439
 _ConfigureBinaryDirectory() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 441
 _ConfigureBinaryDirectory() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443
 _ConfigureBinaryDirectory() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
 _ConfigureBinaryDirectory() (ToolChain.Mentor.QuestaSim.Configuration method), 456
 _ConfigureBinaryDirectory() (ToolChain.PoC.Configuration method), 461
 _ConfigureBinaryDirectory() (ToolChain.ToolConfiguration method), 493
 _ConfigureBinaryDirectory() (ToolChain.Xilinx.ISE.Configuration method), 468
 _ConfigureBinaryDirectory() (ToolChain.Xilinx.Vivado.Configuration method), 477
 _ConfigureDefaultTools() (ToolChain.Configurator method), 496
 _ConfigureEdition() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
 _ConfigureEdition() (ToolChain.Aldec.RivieraPRO.Configuration method), 357
 _ConfigureEdition() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
 _ConfigureEdition() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371
 _ConfigureEdition() (ToolChain.Altera.ModelSim.Configuration method), 365
 _ConfigureEdition() (ToolChain.Altera.Quartus.Configuration method), 374
 _ConfigureEdition() (ToolChain.GHDL.Configuration method), 384
 _ConfigureEdition() (ToolChain.GTKWave.Configuration method), 400
 _ConfigureEdition() (ToolChain.Git.Configuration method), 404
 _ConfigureEdition() (ToolChain.Intel.ModelSim.Configuration method), 412
 _ConfigureEdition() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416
 _ConfigureEdition() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
 _ConfigureEdition() (ToolChain.Intel.Quartus.Configuration method), 421
 _ConfigureEdition() (ToolChain.Lattice.Diamond.Configuration method), 427
 _ConfigureEdition() (ToolChain.Lattice.Synplify.Configuration method), 432
 _ConfigureEdition() (ToolChain.Mentor.ModelSim.Configuration method), 438
 _ConfigureEdition() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 439
 _ConfigureEdition() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 441
 _ConfigureEdition() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444
 _ConfigureEdition() (ToolChain.Mentor.QuestaSim.Configuration method), 456
 _ConfigureEdition() (ToolChain.PoC.Configuration method), 461
 _ConfigureEdition() (ToolChain.ToolConfiguration method), 493
 _ConfigureEdition() (ToolChain.Xilinx.ISE.Configuration method), 468
 _ConfigureEdition() (ToolChain.Xilinx.Vivado.Configuration method), 477
 _ConfigureInstallationDirectory() (ToolChain.Aldec.ActiveHDL.Configuration method), 350
 _ConfigureInstallationDirectory() (ToolChain.Aldec.Configuration method), 363
 _ConfigureInstallationDirectory() (ToolChain.Aldec.RivieraPRO.Configuration method), 357
 _ConfigureInstallationDirectory() (ToolChain.Altera.Configuration method), 380
 _ConfigureInstallationDirectory() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
 _ConfigureInstallationDirectory() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371
 _ConfigureInstallationDirectory() (ToolChain.Altera.ModelSim.Configuration method), 366
 _ConfigureInstallationDirectory() (ToolChain.Altera.Quartus.Configuration method), 374
 _ConfigureInstallationDirectory() (ToolChain.Configuration method), 490
 _ConfigureInstallationDirectory() (ToolChain.GHDL.Configuration method), 384
 _ConfigureInstallationDirectory() (ToolChain.GTKWave.Configuration method), 401
 _ConfigureInstallationDirectory() (ToolChain.Git.Configuration method),

404
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Intel.Configuration method),
 425
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Intel.ModelSim.Configuration
 method), 413
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Intel.ModelSim.IntelEditionConfiguration
 method), 416
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration
 method), 418
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Intel.Quartus.Configuration
 method), 421
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Lattice.Configuration method),
 434
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Lattice.Diamond.Configuration
 method), 427
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Lattice.Synplify.Configuration
 method), 432
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.Configuration method),
 459
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.ModelSim.Configuration
 method), 438
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration
 method), 441
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration
 method), 443
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration
 method), 445
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Mentor.QuestaSim.Configuration
 method), 456
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.PoC.Configuration method),
 461
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Synopsys.Configuration
 method), 464
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.ToolConfiguration method),
 494
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.VendorConfiguration method),
 492
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.Configuration method),
 486
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.ISE.Configuration
 method), 468
[_ConfigureInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.Vivado.Configuration
 method), 477
[_ConfigureScriptDirectory\(\)](#)
 (ToolChain.GHDL.Configuration method),
 382
[_ConfigureTools\(\)](#) (ToolChain.Configurator method),
 496
[_ConfigureVersion\(\)](#) (ToolChain.Aldec.ActiveHDL.Configuration
 method), 350
[_ConfigureVersion\(\)](#) (ToolChain.Aldec.RivieraPRO.Configuration
 method), 357
[_ConfigureVersion\(\)](#) (ToolChain.Altera.ModelSim.AlteraEditionConfigur
 method), 368
[_ConfigureVersion\(\)](#) (ToolChain.Altera.ModelSim.AlteraStarterEditionC
 method), 371
[_ConfigureVersion\(\)](#) (ToolChain.Altera.ModelSim.Configuration
 method), 366
[_ConfigureVersion\(\)](#) (ToolChain.Altera.Quartus.Configuration
 method), 374
[_ConfigureVersion\(\)](#) (ToolChain.GHDL.Configuration
 method), 384
[_ConfigureVersion\(\)](#) (ToolChain.GTKWave.Configuration
 method), 401
[_ConfigureVersion\(\)](#) (ToolChain.Git.Configuration
 method), 405
[_ConfigureVersion\(\)](#) (ToolChain.Intel.ModelSim.Configuration
 method), 413
[_ConfigureVersion\(\)](#) (ToolChain.Intel.ModelSim.IntelEditionConfiguratio
 method), 416
[_ConfigureVersion\(\)](#) (ToolChain.Intel.ModelSim.IntelStarterEditionConf
 method), 418
[_ConfigureVersion\(\)](#) (ToolChain.Intel.Quartus.Configuration
 method), 421
[_ConfigureVersion\(\)](#) (ToolChain.Lattice.Diamond.Configuration
 method), 427
[_ConfigureVersion\(\)](#) (ToolChain.Lattice.Synplify.Configuration
 method), 432
[_ConfigureVersion\(\)](#) (ToolChain.Mentor.ModelSim.Configuration
 method), 438
[_ConfigureVersion\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimPEConfigur
 method), 441
[_ConfigureVersion\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE32Conf
 method), 443
[_ConfigureVersion\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE64Conf
 method), 445
[_ConfigureVersion\(\)](#) (ToolChain.Mentor.QuestaSim.Configuration
 method), 457
[_ConfigureVersion\(\)](#) (ToolChain.PoC.Configuration
 method), 461
[_ConfigureVersion\(\)](#) (ToolChain.ToolConfiguration
 method), 493
[_ConfigureVersion\(\)](#) (ToolChain.Xilinx.ISE.Configuration
 method), 468
[_ConfigureVersion\(\)](#) (ToolChain.Xilinx.Vivado.Configuration

method), 477		method), 290	
<code>_CreatePoCProject()</code> (Base.Shared.Shared method), 266		<code>_DecodeLatticeLCM()</code> (DataBase.Config.Device method), 290	
<code>_CreatePoCProject()</code> (Compiler.Compiler method), 283		<code>_DecodeLatticeLFE()</code> (DataBase.Config.Device method), 290	
<code>_CreatePoCProject()</code> (Compiler.ISECompiler.Compiler method), 268		<code>_DecodeXilinx()</code> (DataBase.Config.Device method), 290	
<code>_CreatePoCProject()</code> (Compiler.LSECompiler.Compiler method), 270		<code>_DefaultAttribute__handler</code> (lib.pyAttribute.ArgParseAttributes.DefaultAttribute attribute), 517	
<code>_CreatePoCProject()</code> (Compiler.QuartusCompiler.Compiler method), 272		<code>_Evaluate()</code> (DataBase.Solution.FileListFile method), 309	
<code>_CreatePoCProject()</code> (Compiler.VivadoCompiler.Compiler method), 273		<code>_Evaluate()</code> (Parser.FilesParser.FilesParserMixIn method), 326	
<code>_CreatePoCProject()</code> (Compiler.XCICompiler.Compiler method), 275		<code>_EvaluatePath()</code> (DataBase.Solution.FileListFile method), 309	
<code>_CreatePoCProject()</code> (Compiler.XCOCompiler.Compiler method), 277		<code>_EvaluatePath()</code> (Parser.FilesParser.FilesParserMixIn method), 326	
<code>_CreatePoCProject()</code> (Compiler.XSTCompiler.Compiler method), 279		<code>_ExecuteCopyTasks()</code> (Compiler.Compiler method), 282	
<code>_CreatePoCProject()</code> (Simulator.ActiveHDL Simulator.Simulator method), 333		<code>_ExecuteCopyTasks()</code> (Compiler.ISECompiler.Compiler method), 268	
<code>_CreatePoCProject()</code> (Simulator.Cocotb Simulator.Simulator method), 334		<code>_ExecuteCopyTasks()</code> (Compiler.LSECompiler.Compiler method), 270	
<code>_CreatePoCProject()</code> (Simulator.GHDL Simulator.Simulator method), 336		<code>_ExecuteCopyTasks()</code> (Compiler.QuartusCompiler.Compiler method), 272	
<code>_CreatePoCProject()</code> (Simulator.ISE Simulator.Simulator method), 338		<code>_ExecuteCopyTasks()</code> (Compiler.VivadoCompiler.Compiler method), 273	
<code>_CreatePoCProject()</code> (Simulator.ModelSim Simulator.Simulator method), 339		<code>_ExecuteCopyTasks()</code> (Compiler.XCICompiler.Compiler method), 275	
<code>_CreatePoCProject()</code> (Simulator.Questa Simulator.Simulator method), 341		<code>_ExecuteCopyTasks()</code> (Compiler.XCOCompiler.Compiler method), 277	
<code>_CreatePoCProject()</code> (Simulator.Simulator method), 346		<code>_ExecuteCopyTasks()</code> (Compiler.XSTCompiler.Compiler method), 279	
<code>_CreatePoCProject()</code> (Simulator.Vivado Simulator.Simulator method), 342		<code>_ExecuteDeleteTasks()</code> (Compiler.Compiler method), 282	
<code>_DEFAULT_INTERPOLATION</code> (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508		<code>_ExecuteDeleteTasks()</code> (Compiler.ISECompiler.Compiler method), 268	
<code>_DecodeAltera()</code> (DataBase.Config.Device method), 290		<code>_ExecuteDeleteTasks()</code> (Compiler.LSECompiler.Compiler method), 270	
<code>_DecodeGeneric()</code> (DataBase.Config.Device method), 290		<code>_ExecuteDeleteTasks()</code> (Compiler.QuartusCompiler.Compiler method), 272	
<code>_DecodeLatticeECP3()</code> (DataBase.Config.Device method), 290		<code>_ExecuteDeleteTasks()</code> (Compiler.VivadoCompiler.Compiler method), 273	
<code>_DecodeLatticeECP5()</code> (DataBase.Config.Device method), 290		<code>_ExecuteDeleteTasks()</code> (Compiler.XCICompiler.Compiler method), 275	
<code>_DecodeLatticeICE()</code> (DataBase.Config.Device			

<code>_ExecuteDeleteTasks()</code> (Compiler.XCOCCompiler.Compiler method), 277	<code>_ExecuteDeleteTasks()</code> (Compiler.XSTCompiler.Compiler method), 279	<code>_ExecuteReplaceTasks()</code> (Compiler.Compiler method), 283	<code>_ExecuteReplaceTasks()</code> (Compiler.ISECompiler.Compiler method), 268	<code>_ExecuteReplaceTasks()</code> (Compiler.LSECompiler.Compiler method), 270	<code>_ExecuteReplaceTasks()</code> (Compiler.QuartusCompiler.Compiler method), 272	<code>_ExecuteReplaceTasks()</code> (Compiler.VivadoCompiler.Compiler method), 273	<code>_ExecuteReplaceTasks()</code> (Compiler.XCICCompiler.Compiler method), 275	<code>_ExecuteReplaceTasks()</code> (Compiler.XCOCCompiler.Compiler method), 277	<code>_ExecuteReplaceTasks()</code> (Compiler.XSTCompiler.Compiler method), 279	<code>_FileType</code> (Base.Project.CocotbSourceFile attribute), 265	<code>_FileType</code> (Base.Project.ConstraintFile attribute), 262	<code>_FileType</code> (Base.Project.File attribute), 262	<code>_FileType</code> (Base.Project.ProjectFile attribute), 262	<code>_FileType</code> (Base.Project.PythonSourceFile attribute), 264	<code>_FileType</code> (Base.Project.SettingsFile attribute), 263	<code>_FileType</code> (Base.Project.SourceFile attribute), 263	<code>_FileType</code> (Base.Project.VHDLSourceFile attribute), 263	<code>_FileType</code> (Base.Project.VerilogSourceFile attribute), 264	<code>_FileType</code> (DataBase.Solution.FileListFile attribute), 309	<code>_FileType</code> (DataBase.Solution.RulesFile attribute), 310	<code>_FileType</code> (ToolChain.Altera.Quartus.QuartusProjectFile attribute), 378	<code>_FileType</code> (ToolChain.Altera.Quartus.QuartusSettings attribute), 378	<code>_FileType</code> (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 430	<code>_FileType</code> (ToolChain.Lattice.LatticeDesignConstraintFile attribute), 435	<code>_FileType</code> (ToolChain.Synopsys.SynopsysDesignConstraintFile attribute), 464	<code>_FileType</code> (ToolChain.Xilinx.ISE.ISEProjectFile attribute), 474	<code>_FileType</code> (ToolChain.Xilinx.ISE.UserConstraintFile attribute), 475	<code>_FileType</code> (ToolChain.Xilinx.Vivado.VivadoProjectFile attribute), 484	<code>_FileType</code> (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile attribute), 484	<code>_FlagsArithmeticMixin__bits</code> (Base.Project.FileTypes attribute), 258	<code>_FlagsArithmeticMixin__bits</code> (DataBase.Entity.BaseFlags attribute), 292	<code>_FlagsArithmeticMixin__bits</code> (DataBase.Entity.NetlistKind attribute), 293	<code>_FlagsArithmeticMixin__bits</code> (DataBase.Entity.TestbenchKind attribute), 292	<code>_FlagsArithmeticMixin__bits</code> (Simulator.SimulationSteps attribute), 344	<code>_FlagsArithmeticMixin__create_flags_instance()</code> (Base.Project.FileTypes method), 258	<code>_FlagsArithmeticMixin__create_flags_instance()</code> (DataBase.Entity.BaseFlags method), 292	<code>_FlagsArithmeticMixin__create_flags_instance()</code> (DataBase.Entity.NetlistKind method), 293	<code>_FlagsArithmeticMixin__create_flags_instance()</code> (DataBase.Entity.TestbenchKind method), 292	<code>_FlagsArithmeticMixin__create_flags_instance()</code> (Simulator.SimulationSteps method), 344	<code>_Flags__internal_str()</code> (Base.Project.FileTypes method), 258	<code>_Flags__internal_str()</code> (DataBase.Entity.BaseFlags method), 292	<code>_Flags__internal_str()</code> (DataBase.Entity.NetlistKind method), 293	<code>_Flags__internal_str()</code> (DataBase.Entity.TestbenchKind method), 292	<code>_Flags__internal_str()</code> (Simulator.SimulationSteps method), 344	<code>_GenerateXilinxProjectFileContent()</code> (Compiler.XSTCompiler.Compiler method), 279	<code>_GenerateXilinxProjectFileContent()</code> (Simulator.ISESimulator.Simulator method), 338	<code>_GenerateXilinxProjectFileContent()</code> (Simulator.VivadoSimulator.Simulator method), 342	<code>_GenerateXilinxProjectFileContent()</code> (ToolChain.Xilinx.XilinxProjectExportMixIn method), 487	<code>_GetConfiguredEditions()</code> (ToolChain.Mentor.ModelSim.Selector method), 447	<code>_GetConfiguredEditions()</code> (ToolChain.ToolSelector method), 495	<code>_GetDefaultEdition()</code> (ToolChain.Aldec.ActiveHDL.Configuration method), 350	<code>_GetDefaultEdition()</code> (ToolChain.Aldec.RivieraPRO.Configuration method), 350
---	--	--	---	---	---	--	--	--	---	--	---	---	--	--	---	---	--	---	---	---	---	--	---	---	---	---	---	---	--	--	---	---	---	---	---	--	--	---	---	--	---	---	---	---	---	---	---	---	---	--	---	--

method), 357 (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369

_GetDefaultEdition() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369 _GetDefaultInstallationDirectory() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 371

_GetDefaultEdition() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371 _GetDefaultInstallationDirectory() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 366

_GetDefaultEdition() (ToolChain.Altera.ModelSim.Configuration method), 366 (ToolChain.Altera.ModelSim.Configuration method), 374

_GetDefaultEdition() (ToolChain.Altera.Quartus.Configuration method), 374 _GetDefaultInstallationDirectory() (ToolChain.Altera.Quartus.Configuration method), 384

_GetDefaultEdition() (ToolChain.GHDL.Configuration method), 384 (ToolChain.Altera.Quartus.Configuration method), 374

_GetDefaultEdition() (ToolChain.GTKWave.Configuration method), 401 _GetDefaultInstallationDirectory() (ToolChain.Configuration method), 490

_GetDefaultEdition() (ToolChain.Git.Configuration method), 405 _GetDefaultInstallationDirectory() (ToolChain.GHDL.Configuration method), 382

_GetDefaultEdition() (ToolChain.Intel.ModelSim.Configuration method), 413 _GetDefaultInstallationDirectory() (ToolChain.GTKWave.Configuration method), 399

_GetDefaultEdition() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416 _GetDefaultInstallationDirectory() (ToolChain.Intel.Configuration method), 423

_GetDefaultEdition() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418 (ToolChain.Git.Configuration method), 403

_GetDefaultEdition() (ToolChain.Intel.Quartus.Configuration method), 421 _GetDefaultInstallationDirectory() (ToolChain.Intel.Configuration method), 423

_GetDefaultEdition() (ToolChain.Lattice.Diamond.Configuration method), 428 (ToolChain.Intel.Configuration method), 423

_GetDefaultEdition() (ToolChain.Lattice.Synplify.Configuration method), 432 _GetDefaultInstallationDirectory() (ToolChain.Intel.ModelSim.Configuration method), 413

_GetDefaultEdition() (ToolChain.Mentor.ModelSim.Configuration method), 438 _GetDefaultInstallationDirectory() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416

_GetDefaultEdition() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 441 (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416

_GetDefaultEdition() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443 _GetDefaultInstallationDirectory() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 445

_GetDefaultEdition() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445 _GetDefaultInstallationDirectory() (ToolChain.Intel.Quartus.Configuration method), 421

_GetDefaultEdition() (ToolChain.Mentor.QuestaSim.Configuration method), 457 _GetDefaultInstallationDirectory() (ToolChain.Lattice.Configuration method), 433

_GetDefaultEdition() (ToolChain.PoC.Configuration method), 461 _GetDefaultInstallationDirectory() (ToolChain.Lattice.Diamond.Configuration method), 428

_GetDefaultEdition() (ToolChain.ToolConfiguration method), 493 _GetDefaultInstallationDirectory() (ToolChain.Lattice.Synplify.Configuration method), 432

_GetDefaultEdition() (ToolChain.Xilinx.ISE.Configuration method), 468 (ToolChain.Lattice.Synplify.Configuration method), 432

_GetDefaultEdition() (ToolChain.Xilinx.Vivado.Configuration method), 478 _GetDefaultInstallationDirectory() (ToolChain.Lattice.Synplify.Configuration method), 432

_GetDefaultInstallationDirectory() (ToolChain.Aldec.ActiveHDL.Configuration method), 350 _GetDefaultInstallationDirectory() (ToolChain.Mentor.Configuration method), 458

_GetDefaultInstallationDirectory() (ToolChain.Aldec.Configuration method), 362 _GetDefaultInstallationDirectory() (ToolChain.Mentor.ModelSim.Configuration method), 438

_GetDefaultInstallationDirectory() (ToolChain.Aldec.RivieraPRO.Configuration method), 357 _GetDefaultInstallationDirectory() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 441

_GetDefaultInstallationDirectory() (ToolChain.Altera.Configuration method), 379 _GetDefaultInstallationDirectory() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443

_GetDefaultInstallationDirectory() (ToolChain.Altera.Configuration method), 379

[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 401
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Git.Configuration method), 405
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Mentor.QuestaSim.Configuration method), 457
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Intel.Configuration method), 425
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.PoC.Configuration method), 461
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Intel.ModelSim.Configuration method), 414
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Synopsys.Configuration method), 462
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.ToolConfiguration method), 494
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.VendorConfiguration method), 492
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Intel.Quartus.Configuration method), 421
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.Configuration method), 485
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Lattice.Configuration method), 434
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.ISE.Configuration method), 468
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Lattice.Diamond.Configuration method), 428
[_GetDefaultInstallationDirectory\(\)](#)
 (ToolChain.Xilinx.Vivado.Configuration method), 478
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Lattice.Synplify.Configuration method), 432
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Aldec.ActiveHDL.Configuration method), 350
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Aldec.Configuration method), 459
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Aldec.RivieraPRO.Configuration method), 363
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.Configuration method), 439
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.ModelSim.Configuration method), 439
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 441
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Altera.Configuration method), 381
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 446
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Mentor.QuestaSim.Configuration method), 457
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Altera.ModelSim.Configuration method), 366
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.PoC.Configuration method), 461
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Altera.Quartus.Configuration method), 374
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Synopsys.Configuration method), 464
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.Configuration method), 490
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.ToolConfiguration method), 494
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.GHDL.Configuration method), 384
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.VendorConfiguration method), 492
[_GetDefaultOptionValue\(\)](#)
 (ToolChain.GTKWave.Configuration method), 492

_GetDefaultOptionValue() (ToolChain.Xilinx.Configuration method), 486	_GetHDLParameters() (Compiler.Compiler method), 283
_GetDefaultOptionValue() (ToolChain.Xilinx.ISE.Configuration method), 468	_GetHDLParameters() (Compiler.ISECompiler.Compiler method), 268
_GetDefaultOptionValue() (ToolChain.Xilinx.Vivado.Configuration method), 478	_GetHDLParameters() (Compiler.LSECompiler.Compiler method), 270
_GetDefaultVersion() (ToolChain.Aldec.ActiveHDL.Configuration method), 350	_GetHDLParameters() (Compiler.QuartusCompiler.Compiler method), 272
_GetDefaultVersion() (ToolChain.Aldec.RivieraPRO.Configuration method), 357	_GetHDLParameters() (Compiler.VivadoCompiler.Compiler method), 273
_GetDefaultVersion() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369	_GetHDLParameters() (Compiler.XCICompiler.Compiler method), 275
_GetDefaultVersion() (ToolChain.Altera.ModelSim.Configuration method), 371	_GetHDLParameters() (Compiler.XCOCompiler.Compiler method), 277
_GetDefaultVersion() (ToolChain.Altera.Quartus.Configuration method), 374	_GetHDLParameters() (Compiler.XSTCompiler.Compiler method), 279
_GetDefaultVersion() (ToolChain.GHDL.Configuration method), 384	_GetHDLParameters() (Simulator.ActiveHDL Simulator.Simulator method), 333
_GetDefaultVersion() (ToolChain.GTKWave.Configuration method), 401	_GetHDLParameters() (Simulator.Cocotb Simulator.Simulator method), 334
_GetDefaultVersion() (ToolChain.Git.Configuration method), 405	_GetHDLParameters() (Simulator.GHDL Simulator.Simulator method), 336
_GetDefaultVersion() (ToolChain.Intel.ModelSim.Configuration method), 414	_GetHDLParameters() (Simulator.ISE Simulator.Simulator method), 338
_GetDefaultVersion() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416	_GetHDLParameters() (Simulator.ModelSim Simulator.Simulator method), 339
_GetDefaultVersion() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 419	_GetHDLParameters() (Simulator.Questa Simulator.Simulator method), 341
_GetDefaultVersion() (ToolChain.Intel.Quartus.Configuration method), 421	_GetHDLParameters() (Simulator.SimHDL Compiler.Simulator method), 346
_GetDefaultVersion() (ToolChain.Lattice.Diamond.Configuration method), 428	_GetHDLParameters() (Simulator.SimHDL Compiler.Simulator method), 347
_GetDefaultVersion() (ToolChain.Lattice.Synplify.Configuration method), 432	_GetHDLParameters() (Simulator.Vivado Simulator.Simulator method), 348
_GetDefaultVersion() (ToolChain.Mentor.ModelSim.Configuration method), 439	_GetModelSimBinaryDirectory() (DataBase.Query method), 316
_GetDefaultVersion() (ToolChain.Mentor.ModelSim.ModelSimSE64 Configuration method), 441	_GetModelSimInstallationDirectory() (DataBase.Query method), 316
_GetDefaultVersion() (ToolChain.Mentor.ModelSim.ModelSimSE64 Configuration method), 443	_GetModelSimVersion() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369
_GetDefaultVersion() (ToolChain.Mentor.ModelSim.ModelSimSE64 Configuration method), 446	_GetModelSimVersion() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371
_GetDefaultVersion() (ToolChain.Mentor.QuestaSim.Configuration method), 457	_GetModelSimVersion() (ToolChain.Xilinx.ISE.Configuration method), 468
_GetDefaultVersion() (ToolChain.PoC.Configuration method), 461	_GetModelSimVersion() (ToolChain.Xilinx.Vivado.Configuration method), 478
_GetDefaultVersion() (ToolChain.ToolConfiguration method), 493	_GetHDLParameters() (Base.Shared.Shared method),
_GetDefaultVersion() (ToolChain.Xilinx.ISE.Configuration method), 468	
_GetDefaultVersion() (ToolChain.Xilinx.Vivado.Configuration method), 478	
_GetHDLParameters() (Base.Shared.Shared method),	

- method), 366
- `_GetModelSimVersion()` (ToolChain.Intel.ModelSim.Configuration method), 414
- `_GetModelSimVersion()` (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416
- `_GetModelSimVersion()` (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 419
- `_GetModelSimVersion()` (ToolChain.Mentor.ModelSim.Configuration method), 437
- `_GetModelSimVersion()` (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 441
- `_GetModelSimVersion()` (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443
- `_GetModelSimVersion()` (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 446
- `_GetModelSimVersion()` (ToolChain.Mentor.QuestaSim.Configuration method), 457
- `_GetTimeDeltaSinceLastEvent()` (Base.Shared.Shared method), 265
- `_GetTimeDeltaSinceLastEvent()` (Compiler.Compiler method), 283
- `_GetTimeDeltaSinceLastEvent()` (Compiler.ISECompiler.Compiler method), 268
- `_GetTimeDeltaSinceLastEvent()` (Compiler.LSECompiler.Compiler method), 270
- `_GetTimeDeltaSinceLastEvent()` (Compiler.QuartusCompiler.Compiler method), 272
- `_GetTimeDeltaSinceLastEvent()` (Compiler.VivadoCompiler.Compiler method), 274
- `_GetTimeDeltaSinceLastEvent()` (Compiler.XCICompiler.Compiler method), 275
- `_GetTimeDeltaSinceLastEvent()` (Compiler.XCOCCompiler.Compiler method), 277
- `_GetTimeDeltaSinceLastEvent()` (Compiler.XSTCompiler.Compiler method), 279
- `_GetTimeDeltaSinceLastEvent()` (Simulator.ActiveHDL Simulator.Simulator method), 333
- `_GetTimeDeltaSinceLastEvent()` (Simulator.Cocotb Simulator.Simulator method), 334
- `_GetTimeDeltaSinceLastEvent()` (Simulator.GHDL Simulator.Simulator method), 336
- `_GetTimeDeltaSinceLastEvent()` (Simulator.ISE Simulator.Simulator method), 338
- `_GetTimeDeltaSinceLastEvent()` (Simulator.ModelSim Simulator.Simulator method), 339
- `_GetTimeDeltaSinceLastEvent()` (Simulator.Questa Simulator.Simulator method), 341
- `_GetTimeDeltaSinceLastEvent()` (Simulator.Simulator method), 346
- `_GetTimeDeltaSinceLastEvent()` (Simulator.Vivado Simulator.Simulator method), 342
- `_GetXilinxISESettingsFile()` (DataBase.Query method), 316
- `_GetXilinxVivadoSettingsFile()` (DataBase.Query method), 316
- `_IntelEditionConfiguration__editionName` (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 416
- `_IntelStarterEditionConfiguration__editionName` (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 419
- `_KEYCRE` (lib.ExtendedConfigParser.ExtendedInterpolation attribute), 507
- `_KEYCRE2` (lib.ExtendedConfigParser.ExtendedInterpolation attribute), 507
- `_LazyLoadable_Load()` (DataBase.Entity.CocoTestbench method), 301
- `_LazyLoadable_Load()` (DataBase.Entity.CoreGeneratorNetlist method), 304
- `_LazyLoadable_Load()` (DataBase.Entity.LatticeNetlist method), 303
- `_LazyLoadable_Load()` (DataBase.Entity.LazyPathElement method), 299
- `_LazyLoadable_Load()` (DataBase.Entity.Netlist method), 301
- `_LazyLoadable_Load()` (DataBase.Entity.QuartusNetlist method), 303
- `_LazyLoadable_Load()` (DataBase.Entity.Testbench method), 300
- `_LazyLoadable_Load()` (DataBase.Entity.VHDLTestbench method), 300
- `_LazyLoadable_Load()` (DataBase.Entity.VivadoNetlist method), 304
- `_LazyLoadable_Load()` (DataBase.Entity.XstNetlist method), 302
- `_LazyLoadable_Load()` (DataBase.Solution.Base method), 305
- `_LazyLoadable_Load()` (DataBase.Solution.ISEProject method), 307

- `_LazyLoadable_Load()` (DataBase.Solution.LatticeProject method), 308
- `_LazyLoadable_Load()` (DataBase.Solution.Project method), 307
- `_LazyLoadable_Load()` (DataBase.Solution.QuartusProject method), 308
- `_LazyLoadable_Load()` (DataBase.Solution.Repository method), 306
- `_LazyLoadable_Load()` (DataBase.Solution.Solution method), 306
- `_LazyLoadable_Load()` (DataBase.Solution.VivadoProject method), 307
- `_LazyLoadable_Load()` (lib.Decorators.ILazyLoadable method), 506
- `_Load()` (DataBase.Entity.AskWildcard method), 298
- `_Load()` (DataBase.Entity.CocoTestbench method), 301
- `_Load()` (DataBase.Entity.CoreGeneratorNetlist method), 304
- `_Load()` (DataBase.Entity.IPCore method), 299
- `_Load()` (DataBase.Entity.LatticeNetlist method), 303
- `_Load()` (DataBase.Entity.LazyPathElement method), 300
- `_Load()` (DataBase.Entity.Library method), 296
- `_Load()` (DataBase.Entity.Namespace method), 295
- `_Load()` (DataBase.Entity.Netlist method), 302
- `_Load()` (DataBase.Entity.PathElement method), 295
- `_Load()` (DataBase.Entity.QuartusNetlist method), 303
- `_Load()` (DataBase.Entity.StarWildcard method), 297
- `_Load()` (DataBase.Entity.Testbench method), 300
- `_Load()` (DataBase.Entity.VHDLTestbench method), 301
- `_Load()` (DataBase.Entity.VivadoNetlist method), 305
- `_Load()` (DataBase.Entity.WildCard method), 297
- `_Load()` (DataBase.Entity.XstNetlist method), 302
- `_Load()` (DataBase.Solution.Base method), 305
- `_Load()` (DataBase.Solution.ISEProject method), 307
- `_Load()` (DataBase.Solution.LatticeProject method), 308
- `_Load()` (DataBase.Solution.Project method), 307
- `_Load()` (DataBase.Solution.QuartusProject method), 308
- `_Load()` (DataBase.Solution.Repository method), 306
- `_Load()` (DataBase.Solution.Solution method), 306
- `_Load()` (DataBase.Solution.VivadoProject method), 307
- `_Log_MESSAGE_FORMAT__` (Base.Logging.LogEntry attribute), 256
- `_Log_MESSAGE_FORMAT__` (Base.Logging.Logger attribute), 256
- `_MutableMapping__marker` (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
- `_MutableMapping__marker` (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 506
- `_NamespaceRoot__POCRoot_Name` (DataBase.Entity.NamespaceRoot attribute), 294
- `_NamespaceRoot__POCRoot_SectionName` (DataBase.Entity.NamespaceRoot attribute), 294
- `_OPT_NV_TMPL` (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
- `_OPT_TMPL` (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
- `_POC_BOUNDARY` (Base.Executable.Executable attribute), 255
- `_POC_BOUNDARY` (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 353
- `_POC_BOUNDARY` (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 352
- `_POC_BOUNDARY` (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSim attribute), 354
- `_POC_BOUNDARY` (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 360
- `_POC_BOUNDARY` (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358
- `_POC_BOUNDARY` (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 361
- `_POC_BOUNDARY` (ToolChain.Altera.Quartus.Map attribute), 376
- `_POC_BOUNDARY` (ToolChain.Altera.Quartus.TclShell attribute), 377
- `_POC_BOUNDARY` (ToolChain.GHDL.GHDL attribute), 387
- `_POC_BOUNDARY` (ToolChain.GHDL.GHDLAnalyze attribute), 390
- `_POC_BOUNDARY` (ToolChain.GHDL.GHDL Elaborate attribute), 393
- `_POC_BOUNDARY` (ToolChain.GHDL.GHDLRun attribute), 397
- `_POC_BOUNDARY` (ToolChain.GNU.Make attribute), 398
- `_POC_BOUNDARY` (ToolChain.GTKWave.GTKWave attribute), 402
- `_POC_BOUNDARY` (ToolChain.Git.GitConfig attribute), 411
- `_POC_BOUNDARY` (ToolChain.Git.GitDescribe attribute), 409
- `_POC_BOUNDARY` (ToolChain.Git.GitRevList attribute), 408
- `_POC_BOUNDARY` (ToolChain.Git.GitRevParse attribute), 407
- `_POC_BOUNDARY` (ToolChain.Git.GitSCM attribute), 406
- `_POC_BOUNDARY` (ToolChain.Intel.Quartus.Map attribute), 423
- `_POC_BOUNDARY` (ToolChain.Lattice.Diamond.Synth attribute), 429
- `_POC_BOUNDARY` (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 451
- `_POC_BOUNDARY` (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 448

<code>_POC_BOUNDARY</code> (ToolChain.Mentor.ModelSim.VHDLSimulator attribute), 454	<code>_ParseDeleteRules()</code> (Compiler.Compiler method), 279
<code>_POC_BOUNDARY</code> (ToolChain.Windows.Cmd attribute), 465	<code>_ParseReplaceRules()</code> (Compiler.Compiler method), 282
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473	<code>_ParseReplaceRules()</code> (Compiler.Compiler method), 268
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.ISE.Fuse attribute), 470	<code>_ParseReplaceRules()</code> (Compiler.LSECompiler.Compiler method), 270
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.ISE.ISESimulator attribute), 471	<code>_ParseReplaceRules()</code> (Compiler.QuartusCompiler.Compiler method), 272
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.ISE.Xst attribute), 472	<code>_ParseReplaceRules()</code> (Compiler.VivadoCompiler.Compiler method), 274
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.Vivado.Synth attribute), 483	<code>_ParseReplaceRules()</code> (Compiler.XCICompiler.Compiler method), 275
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.Vivado.XElab attribute), 480	<code>_ParseReplaceRules()</code> (Compiler.XCOCompiler.Compiler method), 277
<code>_POC_BOUNDARY</code> (ToolChain.Xilinx.Vivado.XSim attribute), 481	<code>_ParseReplaceRules()</code> (Compiler.XSTCompiler.Compiler method), 279
<code>_Parse()</code> (DataBase.Solution.FileListFile method), 309	<code>_PoCEntityTypes_parser()</code> (in module DataBase.Entity), 305
<code>_Parse()</code> (DataBase.Solution.RulesFile method), 310	<code>_PosixFormat</code> (Base.Executable.PathArgument attribute), 248
<code>_Parse()</code> (Parser.FilesParser.FilesParserMixIn method), 326	<code>_Prepare()</code> (Base.Shared.Shared method), 266
<code>_Parse()</code> (Parser.RulesParser.RulesParserMixIn method), 332	<code>_Prepare()</code> (Compiler.Compiler method), 283
<code>_ParseCopyRules()</code> (Compiler.Compiler method), 282	<code>_Prepare()</code> (Compiler.ISECompiler.Compiler method), 268
<code>_ParseCopyRules()</code> (Compiler.ISECompiler.Compiler method), 268	<code>_Prepare()</code> (Compiler.LSECompiler.Compiler method), 270
<code>_ParseCopyRules()</code> (Compiler.LSECompiler.Compiler method), 270	<code>_Prepare()</code> (Compiler.QuartusCompiler.Compiler method), 272
<code>_ParseCopyRules()</code> (Compiler.QuartusCompiler.Compiler method), 272	<code>_Prepare()</code> (Compiler.VivadoCompiler.Compiler method), 274
<code>_ParseCopyRules()</code> (Compiler.VivadoCompiler.Compiler method), 274	<code>_Prepare()</code> (Compiler.XCICompiler.Compiler method), 275
<code>_ParseCopyRules()</code> (Compiler.XCICompiler.Compiler method), 275	<code>_Prepare()</code> (Compiler.XCOCompiler.Compiler method), 277
<code>_ParseCopyRules()</code> (Compiler.XCOCompiler.Compiler method), 277	<code>_Prepare()</code> (Compiler.XSTCompiler.Compiler method), 279
<code>_ParseCopyRules()</code> (Compiler.XSTCompiler.Compiler method), 279	<code>_Prepare()</code> (Simulator.ActiveHDLSimulator.Simulator method), 333
<code>_ParseDeleteRules()</code> (Compiler.Compiler method), 282	<code>_Prepare()</code> (Simulator.CocotbSimulator.Simulator method), 335
<code>_ParseDeleteRules()</code> (Compiler.ISECompiler.Compiler method), 268	<code>_Prepare()</code> (Simulator.GHDLSimulator.Simulator method), 336
<code>_ParseDeleteRules()</code> (Compiler.LSECompiler.Compiler method), 270	<code>_Prepare()</code> (Simulator.ISESimulator.Simulator method), 338
<code>_ParseDeleteRules()</code> (Compiler.QuartusCompiler.Compiler method), 272	<code>_Prepare()</code> (Simulator.ModelSimSimulator.Simulator method), 339
<code>_ParseDeleteRules()</code> (Compiler.VivadoCompiler.Compiler method), 274	<code>_Prepare()</code> (Simulator.QuestaSimulator.Simulator method), 339
<code>_ParseDeleteRules()</code> (Compiler.XCICompiler.Compiler method), 275	
<code>_ParseDeleteRules()</code> (Compiler.XCOCompiler.Compiler method), 277	

- method), 341
- `_Prepare()` (Simulator.Simulator method), 347
- `_Prepare()` (Simulator.VivadoSimulator.Simulator method), 342
- `_PrepareCompiler()` (Compiler.Compiler method), 282
- `_PrepareCompiler()` (Compiler.ISECompiler.Compiler method), 267
- `_PrepareCompiler()` (Compiler.LSECompiler.Compiler method), 269
- `_PrepareCompiler()` (Compiler.QuartusCompiler.Compiler method), 271
- `_PrepareCompiler()` (Compiler.VivadoCompiler.Compiler method), 272
- `_PrepareCompiler()` (Compiler.XCICompiler.Compiler method), 274
- `_PrepareCompiler()` (Compiler.XCOCompiler.Compiler method), 276
- `_PrepareCompiler()` (Compiler.XSTCompiler.Compiler method), 278
- `_PrepareCompilerEnvironment()` (Compiler.Compiler method), 282
- `_PrepareCompilerEnvironment()` (Compiler.ISECompiler.Compiler method), 268
- `_PrepareCompilerEnvironment()` (Compiler.LSECompiler.Compiler method), 270
- `_PrepareCompilerEnvironment()` (Compiler.QuartusCompiler.Compiler method), 272
- `_PrepareCompilerEnvironment()` (Compiler.VivadoCompiler.Compiler method), 274
- `_PrepareCompilerEnvironment()` (Compiler.XCICompiler.Compiler method), 275
- `_PrepareCompilerEnvironment()` (Compiler.XCOCompiler.Compiler method), 277
- `_PrepareCompilerEnvironment()` (Compiler.XSTCompiler.Compiler method), 279
- `_PrepareEnvironment()` (Base.Shared.Shared method), 265
- `_PrepareEnvironment()` (Compiler.Compiler method), 283
- `_PrepareEnvironment()` (Compiler.ISECompiler.Compiler method), 268
- `_PrepareEnvironment()` (Compiler.LSECompiler.Compiler method), 270
- `_PrepareEnvironment()` (Compiler.QuartusCompiler.Compiler method), 272
- `_PrepareEnvironment()` (Compiler.VivadoCompiler.Compiler method), 274
- `_PrepareEnvironment()` (Compiler.XCICompiler.Compiler method), 275
- `_PrepareEnvironment()` (Compiler.XCOCompiler.Compiler method), 277
- `_PrepareEnvironment()` (Compiler.XSTCompiler.Compiler method), 279
- `_PrepareEnvironment()` (Simulator.ActiveHDL Simulator.Simulator method), 333
- `_PrepareEnvironment()` (Simulator.Cocotb Simulator.Simulator method), 335
- `_PrepareEnvironment()` (Simulator.GHDL Simulator.Simulator method), 336
- `_PrepareEnvironment()` (Simulator.ISE Simulator.Simulator method), 338
- `_PrepareEnvironment()` (Simulator.ModelSim Simulator.Simulator method), 339
- `_PrepareEnvironment()` (Simulator.Questa Simulator.Simulator method), 341
- `_PrepareEnvironment()` (Simulator.Simulator method), 347
- `_PrepareEnvironment()` (Simulator.Vivado Simulator.Simulator method), 342
- `_PrepareEnvironment_ChangeDirectory()` (Base.Shared.Shared method), 266
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.Compiler method), 283
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.ISECompiler.Compiler method), 268
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.LSECompiler.Compiler method), 270
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.QuartusCompiler.Compiler method), 272
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.VivadoCompiler.Compiler method), 274
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.XCICompiler.Compiler method), 275
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.XCOCompiler.Compiler method), 277
- `_PrepareEnvironment_ChangeDirectory()` (Compiler.XSTCompiler.Compiler method), 279
- `_PrepareEnvironment_ChangeDirectory()` (Simulator.ActiveHDL Simulator.Simulator method),

333						
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.CocotbSimulator.Simulator method),	<code>lato.QuestaSimulator.Simulator</code>	<code>method)</code> ,	341		
335				<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.Simulator method), 347	
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.GHDLSimulator.Simulator method),	<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.VivadoSimulator.Simulator	342	<code>method)</code> ,	
336				<code>_PrepareEnvironment_PurgeDirectory()</code>	(Base.Shared.Shared method), 266	
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.ISESimulator.Simulator method), 338			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.Compiler method), 283	
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.ModelSimSimulator.Simulator method),	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.ISECompiler.Compiler	268	<code>method)</code> ,	
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.QuestaSimulator.Simulator method),	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.LSECompiler.Compiler	270	<code>method)</code> ,	
341				<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.QuartusCompiler.Compiler	272
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.Simulator method), 347			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.VivadoCompiler.Compiler	274
<code>_PrepareEnvironment_ChangeDirectory()</code>	(Simulator.VivadoSimulator.Simulator method),	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.XCICompiler.Compiler	276	<code>method)</code> ,	
342				<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.XCOCompiler.Compiler	277
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Base.Shared.Shared method), 266			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Compiler.XSTCompiler.Compiler	279
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.Compiler method), 283			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.ActiveHDLSimulator.Simulator	333
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.ISECompiler.Compiler method), 268			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.CocotbSimulator.Simulator	335
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.LSECompiler.Compiler method), 270			<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.GHDLSimulator.Simulator	336
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.QuartusCompiler.Compiler method),	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.ISESimulator.Simulator method), 338	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.ModelSimSimulator.Simulator	339
272				<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.QuestaSimulator.Simulator	341
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.VivadoCompiler.Compiler method),	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.Simulator method), 346	<code>_PrepareEnvironment_PurgeDirectory()</code>	(Simulator.VivadoSimulator.Simulator	342
274				<code>_PrepareSimulationEnvironment()</code>	(Simulator.ActiveHDLSimulator.Simulator	333
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.XCICompiler.Compiler method), 276			<code>_PrepareSimulationEnvironment()</code>	(Simulator.Simulator	
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.XCOCompiler.Compiler method),	<code>_PrepareSimulationEnvironment()</code>	(Simulator.Simulator			
277						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Compiler.XSTCompiler.Compiler method),					
279						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.ActiveHDLSimulator.Simulator method),					
333						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.CocotbSimulator.Simulator method),					
335						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.GHDLSimulator.Simulator method),					
336						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.ISESimulator.Simulator method),					
338						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.ModelSimSimulator.Simulator method),					
339						
<code>_PrepareEnvironment_CreatingDirectory()</code>	(Simulator.Simulator					

<code>tor.CocotbSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
335		<code>(ToolChain.Altera.ModelSim.Configuration</code>	
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>method),</code>	366
<code>tor.GHDLSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
336		<code>(ToolChain.Altera.Quartus.Configuration</code>	
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>method),</code>	374
<code>tor.ISESimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	<code>(ToolChain.AskMixIn</code>
338		<code>method),</code>	489
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>_PrintAvailableEditions()</code>	<code>(ToolChain.Configuration</code>
<code>tor.ModelSimSimulator.Simulator</code>	method),	<code>method),</code>	490
339		<code>_PrintAvailableEditions()</code>	<code>(ToolChain.Configurator</code>
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>method),</code>	497
<code>tor.QuestaSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
341		<code>(ToolChain.GHDL.Configuration</code>	
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>method),</code>	384
<code>tor.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
346		<code>(ToolChain.GTKWave.Configuration</code>	
<code>_PrepareSimulationEnvironment()</code>	<code>(Simula-</code>	<code>method),</code>	401
<code>tor.VivadoSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
342		<code>(ToolChain.Git.Configuration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	405
<code>tor.ActiveHDLSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
332		<code>(ToolChain.Intel.Configuration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	425
<code>tor.CocotbSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
334		<code>(ToolChain.Intel.ModelSim.Configuration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	414
<code>tor.GHDLSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
335		<code>(ToolChain.Intel.ModelSim.IntelEditionConfiguration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	416
<code>tor.ISESimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
337		<code>(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	419
<code>tor.ModelSimSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
338		<code>(ToolChain.Intel.Quartus.Configuration</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>method),</code>	421
<code>tor.QuestaSimulator.Simulator</code>	method),	<code>_PrintAvailableEditions()</code>	
340		<code>(ToolChain.Lattice.Configuration</code>	
<code>_PrepareSimulator()</code>	<code>(Simulator.Simulator</code>	<code>method),</code>	434
346		<code>_PrintAvailableEditions()</code>	
<code>_PrepareSimulator()</code>	<code>(Simula-</code>	<code>(ToolChain.Lattice.Diamond.Configuration</code>	
<code>tor.VivadoSimulator.Simulator</code>	method),	<code>method),</code>	428
341		<code>_PrintAvailableEditions()</code>	
<code>_PrintAvailableEditions()</code>		<code>(ToolChain.Lattice.Synplify.Configuration</code>	
<code>(ToolChain.Aldec.ActiveHDL.Configuration</code>		<code>method),</code>	432
<code>method),</code>		<code>_PrintAvailableEditions()</code>	
350		<code>(ToolChain.Mentor.Configuration</code>	
<code>_PrintAvailableEditions()</code>		<code>method),</code>	459
<code>(ToolChain.Aldec.Configuration</code>		<code>_PrintAvailableEditions()</code>	
363		<code>(ToolChain.Mentor.ModelSim.Configuration</code>	
<code>_PrintAvailableEditions()</code>		<code>method),</code>	439
<code>(ToolChain.Aldec.RivieraPRO.Configuration</code>		<code>_PrintAvailableEditions()</code>	
<code>method),</code>		<code>(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration</code>	
357		<code>method),</code>	441
<code>_PrintAvailableEditions()</code>		<code>_PrintAvailableEditions()</code>	
<code>(ToolChain.Altera.Configuration</code>		<code>(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration</code>	
381		<code>method),</code>	443
<code>_PrintAvailableEditions()</code>		<code>_PrintAvailableEditions()</code>	
<code>(ToolChain.Altera.ModelSim.AlteraEditionConfiguration</code>		<code>method),</code>	443
<code>method),</code>		<code>_PrintAvailableEditions()</code>	
369		<code>(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration</code>	
<code>_PrintAvailableEditions()</code>		<code>method),</code>	443
<code>(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration</code>		<code>_PrintAvailableEditions()</code>	
<code>method),</code>		<code>(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration</code>	
371		<code>method),</code>	443

(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 446	_ReadContent() (ToolChain.Synopsys.SynopsysDesignConstraintFile method), 464
_PrintAvailableEditions() (ToolChain.Mentor.ModelSim.Selector method), 447	_ReadContent() (ToolChain.Xilinx.ISE.ISEProjectFile method), 474
_PrintAvailableEditions() (ToolChain.Mentor.QuestaSim.Configuration method), 457	_ReadContent() (ToolChain.Xilinx.ISE.UserConstraintFile method), 475
_PrintAvailableEditions() (ToolChain.PoC.Configuration method), 461	_ReadContent() (ToolChain.Xilinx.Vivado.VivadoProjectFile method), 484
_PrintAvailableEditions() (ToolChain.Synopsys.Configuration method), 464	_ReadContent() (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile method), 484
_PrintAvailableEditions() (ToolChain.ToolConfiguration method), 494	_Resolve() (DataBase.Solution.FileListFile method), 310
_PrintAvailableEditions() (ToolChain.ToolSelector method), 496	_Resolve() (DataBase.Solution.RulesFile method), 310
_PrintAvailableEditions() (ToolChain.VendorConfiguration method), 492	_Resolve() (Parser.FilesParser.FilesParserMixIn method), 326
_PrintAvailableEditions() (ToolChain.Xilinx.Configuration method), 486	_Resolve() (Parser.RulesParser.RulesParserMixIn method), 332
_PrintAvailableEditions() (ToolChain.Xilinx.ISE.Configuration method), 468	_ResolveRule() (DataBase.Solution.RulesFile method), 310
_PrintAvailableEditions() (ToolChain.Xilinx.Vivado.Configuration method), 478	_ResolveRule() (Parser.RulesParser.RulesParserMixIn method), 332
_ReadContent() (Base.Project.CocotbSourceFile method), 265	_RunAnalysis() (Simulator.ActiveHDL Simulator.Simulator method), 332
_ReadContent() (Base.Project.ConstraintFile method), 262	_RunAnalysis() (Simulator.Cocotb Simulator.Simulator method), 335
_ReadContent() (Base.Project.File method), 262	_RunAnalysis() (Simulator.GHDL Simulator.Simulator method), 335
_ReadContent() (Base.Project.ProjectFile method), 262	_RunAnalysis() (Simulator.ISE Simulator.Simulator method), 338
_ReadContent() (Base.Project.PythonSourceFile method), 264	_RunAnalysis() (Simulator.ModelSim Simulator.Simulator method), 338
_ReadContent() (Base.Project.SettingsFile method), 263	_RunAnalysis() (Simulator.Questa Simulator.Simulator method), 340
_ReadContent() (Base.Project.SourceFile method), 263	_RunAnalysis() (Simulator.Simulator method), 346
_ReadContent() (Base.Project.VHDLSourceFile method), 264	_RunAnalysis() (Simulator.Vivado Simulator.Simulator method), 342
_ReadContent() (Base.Project.VerilogSourceFile method), 264	_RunCompile() (Compiler.LSE Compiler.Compiler method), 269
_ReadContent() (DataBase.Solution.FileListFile method), 309	_RunCompile() (Compiler.Quartus Compiler.Compiler method), 271
_ReadContent() (DataBase.Solution.RulesFile method), 310	_RunCompile() (Compiler.Vivado Compiler.Compiler method), 273
_ReadContent() (ToolChain.Altera.Quartus.QuartusProjectFile method), 378	_RunCompile() (Compiler.XCI Compiler.Compiler method), 274
_ReadContent() (ToolChain.Altera.Quartus.QuartusSettings method), 378	_RunCompile() (Compiler.XCO Compiler.Compiler method), 276
_ReadContent() (ToolChain.Lattice.Diamond.SynthesisArgumentFile method), 430	_RunCompile() (Compiler.XST Compiler.Compiler method), 278
_ReadContent() (ToolChain.Lattice.LatticeDesignConstraintFile method), 435	_RunCoverage() (Simulator.ActiveHDL Simulator.Simulator method), 333
	_RunCoverage() (Simulator.Cocotb Simulator.Simulator method), 335
	_RunCoverage() (Simulator.GHDL Simulator.Simulator method), 335

tor.ActiveHDL Simulator.Simulator method),
332

_RunSimulation() (Simulator.Cocotb Simulator.Simulator method),
334

_RunSimulation() (Simulator.GHDL Simulator.Simulator method),
335

_RunSimulation() (Simulator.ISE Simulator.Simulator method), 337

_RunSimulation() (Simulator.ModelSim Simulator.Simulator method),
338

_RunSimulation() (Simulator.Questa Simulator.Simulator method),
340

_RunSimulation() (Simulator.Simulator method), 346

_RunSimulation() (Simulator.Vivado Simulator.Simulator method),
341

_RunSimulationWithGUI() (Simulator.ActiveHDL Simulator.Simulator method),
332

_RunSimulationWithGUI() (Simulator.ModelSim Simulator.Simulator method),
338

_RunSimulationWithGUI() (Simulator.Questa Simulator.Simulator method),
340

_RunView() (Simulator.ActiveHDL Simulator.Simulator method),
333

_RunView() (Simulator.Cocotb Simulator.Simulator method), 335

_RunView() (Simulator.GHDL Simulator.Simulator method), 335

_RunView() (Simulator.ISE Simulator.Simulator method), 338

_RunView() (Simulator.ModelSim Simulator.Simulator method), 339

_RunView() (Simulator.Questa Simulator.Simulator method), 341

_RunView() (Simulator.Simulator method), 346

_RunView() (Simulator.Vivado Simulator.Simulator method), 342

_SECT_TMPL (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508

_SetExternalLibraryReferences() (Simulator.GHDL Simulator.Simulator method),
335

_SetVHDLVersionAndIEEEFlavor() (Simulator.GHDL Simulator.Simulator method),
335

_TestDefaultInstallPath()
(ToolChain.Aldec.ActiveHDL.Configuration method), 350

_TestDefaultInstallPath()
(ToolChain.Aldec.Configuration method),

363

_TestDefaultInstallPath()
(ToolChain.Aldec.RivieraPRO.Configuration method), 357

_TestDefaultInstallPath()
(ToolChain.Altera.Configuration method),
381

_TestDefaultInstallPath()
(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 369

_TestDefaultInstallPath()
(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 371

_TestDefaultInstallPath()
(ToolChain.Altera.ModelSim.Configuration method), 366

_TestDefaultInstallPath()
(ToolChain.Altera.Quartus.Configuration method), 375

_TestDefaultInstallPath() (ToolChain.Configuration method), 490

_TestDefaultInstallPath()
(ToolChain.GHDL.Configuration method),
384

_TestDefaultInstallPath()
(ToolChain.GTKWave.Configuration method), 401

_TestDefaultInstallPath() (ToolChain.Git.Configuration method), 405

_TestDefaultInstallPath()
(ToolChain.Intel.Configuration method),
425

_TestDefaultInstallPath()
(ToolChain.Intel.ModelSim.Configuration method), 414

_TestDefaultInstallPath()
(ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 416

_TestDefaultInstallPath()
(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 419

_TestDefaultInstallPath()
(ToolChain.Intel.Quartus.Configuration method), 421

_TestDefaultInstallPath()
(ToolChain.Lattice.Configuration method),
434

_TestDefaultInstallPath()
(ToolChain.Lattice.Diamond.Configuration method), 428

_TestDefaultInstallPath()
(ToolChain.Lattice.Synplify.Configuration method), 432

_TestDefaultInstallPath()
(ToolChain.Mentor.Configuration method),
459

_TestDefaultInstallPath()
(ToolChain.Mentor.ModelSim.Configuration

- method), 439
- `_TestDefaultInstallPath()`
(`ToolChain.Mentor.ModelSim.ModelSimPE.Configuration` method), 441
- `_TestDefaultInstallPath()`
(`ToolChain.Mentor.ModelSim.ModelSimSE32.Configuration` method), 443
- `_TestDefaultInstallPath()`
(`ToolChain.Mentor.ModelSim.ModelSimSE64.Configuration` method), 446
- `_TestDefaultInstallPath()`
(`ToolChain.Mentor.QuestaSim.Configuration` method), 457
- `_TestDefaultInstallPath()`
(`ToolChain.PoC.Configuration` method), 461
- `_TestDefaultInstallPath()`
(`ToolChain.Synopsys.Configuration` method), 464
- `_TestDefaultInstallPath()`
(`ToolChain.ToolConfiguration` method), 494
- `_TestDefaultInstallPath()`
(`ToolChain.VendorConfiguration` method), 492
- `_TestDefaultInstallPath()`
(`ToolChain.Xilinx.Configuration` method), 486
- `_TestDefaultInstallPath()`
(`ToolChain.Xilinx.ISE.Configuration` method), 469
- `_TestDefaultInstallPath()`
(`ToolChain.Xilinx.Vivado.Configuration` method), 478
- `_TryLog()` (`Base.Executable.Executable` method), 255
- `_TryLog()` (`Base.IHost` method), 267
- `_TryLog()` (`Base.Logging.ILogable` method), 257
- `_TryLog()` (`Base.Shared.Shared` method), 266
- `_TryLog()` (`Compiler.Compiler` method), 283
- `_TryLog()` (`Compiler.ISECompiler.Compiler` method), 269
- `_TryLog()` (`Compiler.LSECompiler.Compiler` method), 270
- `_TryLog()` (`Compiler.QuartusCompiler.Compiler` method), 272
- `_TryLog()` (`Compiler.VivadoCompiler.Compiler` method), 274
- `_TryLog()` (`Compiler.XCICompiler.Compiler` method), 276
- `_TryLog()` (`Compiler.XCOCCompiler.Compiler` method), 277
- `_TryLog()` (`Compiler.XSTCompiler.Compiler` method), 279
- `_TryLog()` (`Simulator.ActiveHDL Simulator.Simulator` method), 333
- `_TryLog()` (`Simulator.Cocotb Simulator.Simulator` method), 335
- `_TryLog()` (`Simulator.GHDL Simulator.Simulator` method), 336
- `_TryLog()` (`Simulator.ISE Simulator.Simulator` method), 338
- `_TryLog()` (`Simulator.ModelSim Simulator.Simulator` method), 339
- `_TryLog()` (`Simulator.Questa Simulator.Simulator` method), 341
- `_TryLog()` (`Simulator.Simulator` method), 347
- `_TryLog()` (`Simulator.Vivado Simulator.Simulator` method), 342
- `_TryLog()` (`ToolChain.Aldec.ActiveHDL.Configuration` method), 350
- `_TryLog()` (`ToolChain.Aldec.ActiveHDL.VHDLCompiler` method), 353
- `_TryLog()` (`ToolChain.Aldec.ActiveHDL.VHDLLibraryTool` method), 352
- `_TryLog()` (`ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator` method), 354
- `_TryLog()` (`ToolChain.Aldec.Configuration` method), 363
- `_TryLog()` (`ToolChain.Aldec.RivieraPRO.Configuration` method), 357
- `_TryLog()` (`ToolChain.Aldec.RivieraPRO.VHDLCompiler` method), 360
- `_TryLog()` (`ToolChain.Aldec.RivieraPRO.VHDLLibraryTool` method), 358
- `_TryLog()` (`ToolChain.Aldec.RivieraPRO.VHDL Simulator` method), 361
- `_TryLog()` (`ToolChain.Altera.Configuration` method), 381
- `_TryLog()` (`ToolChain.Altera.ModelSim.AlteraEditionConfiguration` method), 369
- `_TryLog()` (`ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration` method), 371
- `_TryLog()` (`ToolChain.Altera.ModelSim.Configuration` method), 366
- `_TryLog()` (`ToolChain.Altera.Quartus.Configuration` method), 375
- `_TryLog()` (`ToolChain.Altera.Quartus.Map` method), 376
- `_TryLog()` (`ToolChain.Altera.Quartus.TclShell` method), 377
- `_TryLog()` (`ToolChain.Configuration` method), 491
- `_TryLog()` (`ToolChain.Configurator` method), 497
- `_TryLog()` (`ToolChain.GHDL.Configuration` method), 384
- `_TryLog()` (`ToolChain.GHDL.GHDL` method), 387
- `_TryLog()` (`ToolChain.GHDL.GHDLAnalyze` method), 390
- `_TryLog()` (`ToolChain.GHDL.GHDL Elaborate` method), 393
- `_TryLog()` (`ToolChain.GHDL.GHDLRun` method), 397
- `_TryLog()` (`ToolChain.GNU.Make` method), 398
- `_TryLog()` (`ToolChain.GTKWave.Configuration` method), 401
- `_TryLog()` (`ToolChain.GTKWave.GTKWave` method), 402
- `_TryLog()` (`ToolChain.Git.Configuration` method), 405

[_TryLog\(\) \(ToolChain.Git.GitConfig method\), 411](#)
[_TryLog\(\) \(ToolChain.Git.GitDescribe method\), 409](#)
[_TryLog\(\) \(ToolChain.Git.GitRevList method\), 408](#)
[_TryLog\(\) \(ToolChain.Git.GitRevParse method\), 407](#)
[_TryLog\(\) \(ToolChain.Git.GitSCM method\), 406](#)
[_TryLog\(\) \(ToolChain.Intel.Configuration method\), 425](#)
[_TryLog\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 414](#)
[_TryLog\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 416](#)
[_TryLog\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 419](#)
[_TryLog\(\) \(ToolChain.Intel.Quartus.Configuration method\), 421](#)
[_TryLog\(\) \(ToolChain.Intel.Quartus.Map method\), 423](#)
[_TryLog\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[_TryLog\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 428](#)
[_TryLog\(\) \(ToolChain.Lattice.Diamond.Synth method\), 429](#)
[_TryLog\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 432](#)
[_TryLog\(\) \(ToolChain.Mentor.Configuration method\), 459](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 439](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 441](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 443](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 446](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.Selector method\), 447](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler method\), 451](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.VHDLLibraryTool method\), 448](#)
[_TryLog\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 454](#)
[_TryLog\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 457](#)
[_TryLog\(\) \(ToolChain.PoC.Configuration method\), 461](#)
[_TryLog\(\) \(ToolChain.Synopsys.Configuration method\), 464](#)
[_TryLog\(\) \(ToolChain.ToolConfiguration method\), 495](#)
[_TryLog\(\) \(ToolChain.ToolSelector method\), 496](#)
[_TryLog\(\) \(ToolChain.VendorConfiguration method\), 492](#)
[_TryLog\(\) \(ToolChain.Windows.Cmd method\), 466](#)
[_TryLog\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[_TryLog\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 469](#)
[_TryLog\(\) \(ToolChain.Xilinx.ISE.CoreGenerator method\), 473](#)
[_TryLog\(\) \(ToolChain.Xilinx.ISE.Fuse method\), 470](#)
[_TryLog\(\) \(ToolChain.Xilinx.ISE.ISESimulator method\), 471](#)
[_TryLog\(\) \(ToolChain.Xilinx.ISE.Xst method\), 472](#)
[_TryLog\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 478](#)
[_TryLog\(\) \(ToolChain.Xilinx.Vivado.Synth method\), 483](#)
[_TryLog\(\) \(ToolChain.Xilinx.Vivado.XElab method\), 480](#)
[_TryLog\(\) \(ToolChain.Xilinx.Vivado.XSim method\), 481](#)
[_WriteConfigurationHeader\(\) \(ToolChain.Configurator method\), 496](#)
[_WriteLSEProjectFile\(\) \(Compiler.LSECompiler.Compiler method\), 269](#)
[_WriteQuartusProjectFile\(\) \(Compiler.QuartusCompiler.Compiler method\), 271](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.Compiler method\), 282](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.ISECompiler.Compiler method\), 269](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.LSECompiler.Compiler method\), 270](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.QuartusCompiler.Compiler method\), 271](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.VivadoCompiler.Compiler method\), 273](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.XCICompiler.Compiler method\), 274](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.XCOCompiler.Compiler method\), 276](#)
[_WriteSpecialSectionIntoConfig\(\) \(Compiler.XSTCompiler.Compiler method\), 278](#)
[_WriteTclFile\(\) \(Compiler.VivadoCompiler.Compiler method\), 273](#)
[_WriteTclFile\(\) \(Compiler.XCICompiler.Compiler method\), 274](#)
[_WriteXilinxProjectFile\(\) \(Compiler.XSTCompiler.Compiler method\), 279](#)
[_WriteXilinxProjectFile\(\) \(Simulator.ISESimulator.Simulator method\), 338](#)
[_WriteXilinxProjectFile\(\) \(Simulator.VivadoSimulator.Simulator method\), 343](#)
[_WriteXilinxProjectFile\(\) \(ToolChain.Xilinx.XilinxProjectExportMixIn](#)

method), 487

`_WriteXstOptionsFile()` (Compiler.XSTCompiler.Compiler method), 278

`__POC_PROJECT_KEYWORD__` (in module DataBase), 316

`__POC_SOLUTION_KEYWORD__` (in module DataBase), 315

`__call__()` (lib.Decorators.MethodAlias method), 506

`__editionName` (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 367

`__editionName` (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 369

`__editionName` (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 414

`__editionName` (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 417

`__init__()` (Base.Exceptions.CommonException method), 243

`__init__()` (Base.Exceptions.EnvironmentException method), 242

`__init__()` (Base.Exceptions.ExceptionBase method), 242

`__init__()` (Base.Exceptions.NotConfiguredException method), 242

`__init__()` (Base.Exceptions.PlatformNotSupportedException method), 242

`__init__()` (Base.Exceptions.SkipableCommonException method), 243

`__init__()` (Base.Exceptions.SkipableException method), 243

`__init__()` (Compiler.Compiler method), 282

`__init__()` (Compiler.CompilerException method), 280

`__init__()` (Compiler.SkipableCompilerException method), 280

`__init__()` (Simulator.PoCSimulationResultNotFoundException method), 344

`__init__()` (Simulator.Simulator method), 345

`__init__()` (Simulator.SimulatorException method), 343

`__init__()` (Simulator.SkipableSimulatorException method), 343

`__init__()` (ToolChain.Aldec.ActiveHDL.ActiveHDLException method), 348

`__init__()` (ToolChain.Aldec.AldecException method), 361

`__init__()` (ToolChain.Aldec.RivieraPRO.RivieraPROException method), 355

`__init__()` (ToolChain.Altera.AlteraException method), 379

`__init__()` (ToolChain.Altera.ModelSim.ModelSimException method), 364

`__init__()` (ToolChain.Altera.Quartus.QuartusException method), 372

`__init__()` (ToolChain.ConfigurationException method), 487

`__init__()` (ToolChain.GHDL.GHDLException method), 381

`__init__()` (ToolChain.GHDL.GHDLReanalyzeException method), 382

`__init__()` (ToolChain.GNU.GNUException method), 397

`__init__()` (ToolChain.GTKWave.GTKWaveException method), 399

`__init__()` (ToolChain.Git.GitException method), 402

`__init__()` (ToolChain.Intel.IntelException method), 423

`__init__()` (ToolChain.Intel.ModelSim.ModelSimException method), 411

`__init__()` (ToolChain.Intel.Quartus.QuartusException method), 419

`__init__()` (ToolChain.Lattice.Diamond.DiamondException method), 426

`__init__()` (ToolChain.Lattice.LatticeException method), 433

`__init__()` (ToolChain.Lattice.Synplify.SynplifyException method), 430

`__init__()` (ToolChain.Mentor.MentorException method), 457

`__init__()` (ToolChain.Mentor.ModelSim.ModelSimException method), 436

`__init__()` (ToolChain.Mentor.QuestaSim.QuestaSimException method), 454

`__init__()` (ToolChain.SkipConfigurationException method), 488

`__init__()` (ToolChain.Synopsys.SynopsysException method), 462

`__init__()` (ToolChain.ToolChainException method), 487

`__init__()` (ToolChain.Windows.WindowsException method), 465

`__init__()` (ToolChain.Xilinx.ISE.ISEException method), 466

`__init__()` (ToolChain.Xilinx.Vivado.VivadoException method), 476

`__init__()` (ToolChain.Xilinx.XilinxException method), 484

`__init__()` (lib.Decorators.MethodAlias method), 506

`__str__()` (Base.Exceptions.CommonException method), 243

`__str__()` (Base.Exceptions.EnvironmentException method), 242

`__str__()` (Base.Exceptions.ExceptionBase method), 242

`__str__()` (Base.Exceptions.NotConfiguredException method), 242

`__str__()` (Base.Exceptions.PlatformNotSupportedException method), 242

`__str__()` (Base.Exceptions.SkipableCommonException method), 243

`__str__()` (Base.Exceptions.SkipableException method), 243

`__str__()` (Base.Executable.ExecutableException method), 245

`__str__()` (Compiler.CompilerException method), 280

`__str__()` (Compiler.SkipableCompilerException method), 280

<code>__str__()</code> (Simulator.PoCSimulationResultNotFoundException method), 344	<code>_abc_cache</code> (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
<code>__str__()</code> (Simulator.SimulatorException method), 343	<code>_abc_cache</code> (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 506
<code>__str__()</code> (Simulator.SkipableSimulatorException method), 343	<code>_abc_negative_cache</code> (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
<code>__str__()</code> (ToolChain.Aldec.ActiveHDL.ActiveHDLException method), 348	<code>_abc_negative_cache</code> (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 507
<code>__str__()</code> (ToolChain.Aldec.AldecException method), 361	<code>_abc_negative_cache_version</code> (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
<code>__str__()</code> (ToolChain.Aldec.RivieraPRO.RivieraPROException method), 355	<code>_abc_negative_cache_version</code> (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 507
<code>__str__()</code> (ToolChain.Altera.AlteraException method), 379	<code>_abc_registry</code> (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
<code>__str__()</code> (ToolChain.Altera.ModelSim.ModelSimException method), 364	<code>_abc_registry</code> (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 507
<code>__str__()</code> (ToolChain.Altera.Quartus.QuartusException method), 372	<code>_allowedExpressions</code> (Parser.FilesCodeDOM.IfThenElseExpressions attribute), 317
<code>__str__()</code> (ToolChain.ConfigurationException method), 488	<code>_allowedExpressions</code> (Parser.FilesCodeDOM.ListElementExpressions attribute), 318
<code>__str__()</code> (ToolChain.GHDL.GHDLException method), 381	<code>_allowedExpressions</code> (Parser.FilesCodeDOM.PathExpressions attribute), 318
<code>__str__()</code> (ToolChain.GHDL.GHDLReanalyzeException method), 382	<code>_allowedExpressions</code> (lib.CodeDOM.ExpressionChoice attribute), 505
<code>__str__()</code> (ToolChain.GNU.GNUEXception method), 397	<code>_allowedStatements</code> (Parser.FilesCodeDOM.BlockedStatement attribute), 317
<code>__str__()</code> (ToolChain.GTKWave.GTKWaveException method), 399	<code>_allowedStatements</code> (Parser.RulesCodeDOM.DocumentStatements attribute), 328
<code>__str__()</code> (ToolChain.Git.GitException method), 403	<code>_allowedStatements</code> (Parser.RulesCodeDOM.InFileStatements attribute), 327
<code>__str__()</code> (ToolChain.Intel.IntelException method), 423	<code>_allowedStatements</code> (Parser.RulesCodeDOM.PostProcessStatements attribute), 327
<code>__str__()</code> (ToolChain.Intel.ModelSim.ModelSimException method), 411	<code>_allowedStatements</code> (Parser.RulesCodeDOM.PreProcessStatements attribute), 327
<code>__str__()</code> (ToolChain.Intel.Quartus.QuartusException method), 419	<code>_allowedStatements</code> (lib.CodeDOM.BlockedStatement attribute), 505
<code>__str__()</code> (ToolChain.Lattice.Diamond.DiamondException method), 426	<code>_asdict()</code> (ToolChain.EditionDescription method), 495
<code>__str__()</code> (ToolChain.Lattice.LatticeException method), 433	<code>_classAppendLineRule</code> (DataBase.Solution.RulesFile attribute), 310
<code>__str__()</code> (ToolChain.Lattice.Synplify.SynplifyException method), 430	<code>_classAppendLineRule</code> (Parser.RulesParser.RulesParserMixIn attribute), 332
<code>__str__()</code> (ToolChain.Mentor.MentorException method), 457	<code>_classCocotbSourceFile</code> (DataBase.Solution.FileListFile attribute), 309
<code>__str__()</code> (ToolChain.Mentor.ModelSim.ModelSimException method), 436	<code>_classCocotbSourceFile</code> (Parser.FilesParser.FilesParserMixIn attribute), 326
<code>__str__()</code> (ToolChain.Mentor.QuestaSim.QuestaSimException method), 454	<code>_classCopyRule</code> (DataBase.Solution.RulesFile attribute), 310
<code>__str__()</code> (ToolChain.SkipConfigurationException method), 488	<code>_classCopyRule</code> (Parser.RulesParser.RulesParserMixIn attribute), 331
<code>__str__()</code> (ToolChain.Synopsys.SynopsysException method), 462	<code>_classDeleteRule</code> (DataBase.Solution.RulesFile attribute), 310
<code>__str__()</code> (ToolChain.ToolChainException method), 487	<code>_classDeleteRule</code> (Parser.RulesParser.RulesParserMixIn attribute), 310
<code>__str__()</code> (ToolChain.Windows.WindowsException method), 465	
<code>__str__()</code> (ToolChain.Xilinx.ISE.ISEException method), 466	
<code>__str__()</code> (ToolChain.Xilinx.Vivado.VivadoException method), 476	
<code>__str__()</code> (ToolChain.Xilinx.XilinxException method),	

- attribute), 331
- _classIncludeFile (DataBase.Solution.FileListFile attribute), 310
- _classIncludeFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classLDCSourceFile (DataBase.Solution.FileListFile attribute), 310
- _classLDCSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classReplaceRule (DataBase.Solution.RulesFile attribute), 310
- _classReplaceRule (Parser.RulesParser.RulesParserMixIn attribute), 332
- _classSDCSourceFile (DataBase.Solution.FileListFile attribute), 310
- _classSDCSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classUCFSourceFile (DataBase.Solution.FileListFile attribute), 310
- _classUCFSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classVHDLSourceFile (DataBase.Solution.FileListFile attribute), 309
- _classVHDLSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classVerilogSourceFile (DataBase.Solution.FileListFile attribute), 309
- _classVerilogSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _classXDCSourceFile (DataBase.Solution.FileListFile attribute), 310
- _classXDCSourceFile (Parser.FilesParser.FilesParserMixIn attribute), 326
- _convert_to_boolean() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _debug (lib.SphinxExtensions.DocumentMemberAttribute attribute), 516
- _debug (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute attribute), 518
- _debug (lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute), 517
- _debug (lib.pyAttribute.ArgParseAttributes.CommandGroupAttribute attribute), 517
- _debug (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute attribute), 518
- _debug (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute attribute), 519
- _debug (lib.pyAttribute.ArgParseAttributes.DefaultAttribute attribute), 517
- _debug (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute attribute), 518
- _debug (lib.pyAttribute.Attribute attribute), 519
- _fields (ToolChain.EditionDescription attribute), 495
- _get() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _get_conv() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _handle_error() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _join_multiline_values() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _make() (ToolChain.EditionDescription class method), 495
- _multiVersionSupport (ToolChain.Aldec.ActiveHDL.Configuration attribute), 348
- _multiVersionSupport (ToolChain.Aldec.Configuration attribute), 363
- _multiVersionSupport (ToolChain.Aldec.RivieraPRO.Configuration attribute), 355
- _multiVersionSupport (ToolChain.Altera.Configuration attribute), 381
- _multiVersionSupport (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 369
- _multiVersionSupport (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 371
- _multiVersionSupport (ToolChain.Altera.ModelSim.Configuration attribute), 364
- _multiVersionSupport (ToolChain.Altera.Quartus.Configuration attribute), 373
- _multiVersionSupport (ToolChain.Configuration attribute), 489
- _multiVersionSupport (ToolChain.GHDL.Configuration attribute), 382
- _multiVersionSupport (ToolChain.GTKWave.Configuration attribute), 399
- _multiVersionSupport (ToolChain.Git.Configuration attribute), 405
- _multiVersionSupport (ToolChain.Intel.Configuration attribute), 425
- _multiVersionSupport (ToolChain.Intel.ModelSim.Configuration attribute), 412
- _multiVersionSupport (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 416
- _multiVersionSupport (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 419
- _multiVersionSupport (ToolChain.Intel.Quartus.Configuration attribute), 420
- _multiVersionSupport (ToolChain.Lattice.Configuration attribute), 435
- _multiVersionSupport (ToolChain.Lattice.Diamond.Configuration attribute), 426
- _multiVersionSupport (ToolChain.Lattice.Synplify.Configuration attribute), 432
- _multiVersionSupport (ToolChain.Mentor.Configuration attribute), 459
- _multiVersionSupport (ToolChain.Mentor.ModelSim.Configuration attribute), 437
- _multiVersionSupport (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 441
- _multiVersionSupport (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 444

[_multiVersionSupport \(ToolChain.Mentor.ModelSim.ModelSimSE64.Configuration attribute\), 446](#)
[_multiVersionSupport \(ToolChain.Mentor.QuestaSim.Configuration attribute\), 352](#)
[_multiVersionSupport \(ToolChain.PoC.Configuration attribute\), 461](#)
[_multiVersionSupport \(ToolChain.Synopsys.Configuration attribute\), 464](#)
[_multiVersionSupport \(ToolChain.ToolConfiguration attribute\), 495](#)
[_multiVersionSupport \(ToolChain.VendorConfiguration attribute\), 492](#)
[_multiVersionSupport \(ToolChain.Xilinx.Configuration attribute\), 486](#)
[_multiVersionSupport \(ToolChain.Xilinx.ISE.Configuration attribute\), 467](#)
[_multiVersionSupport \(ToolChain.Xilinx.Vivado.Configuration attribute\), 476](#)
[_name \(Base.Executable.CommandArgument attribute\), 246](#)
[_name \(Base.Executable.FlagArgument attribute\), 248](#)
[_name \(Base.Executable.LongCommandArgument attribute\), 246](#)
[_name \(Base.Executable.LongFlagArgument attribute\), 249](#)
[_name \(Base.Executable.LongTupleArgument attribute\), 253](#)
[_name \(Base.Executable.LongValuedFlagArgument attribute\), 250](#)
[_name \(Base.Executable.LongValuedFlagListArgument attribute\), 252](#)
[_name \(Base.Executable.NamedCommandLineArgument attribute\), 245](#)
[_name \(Base.Executable.ShortCommandArgument attribute\), 246](#)
[_name \(Base.Executable.ShortFlagArgument attribute\), 248](#)
[_name \(Base.Executable.ShortTupleArgument attribute\), 253](#)
[_name \(Base.Executable.ShortValuedFlagArgument attribute\), 250](#)
[_name \(Base.Executable.ShortValuedFlagListArgument attribute\), 251](#)
[_name \(Base.Executable.TupleArgument attribute\), 253](#)
[_name \(Base.Executable.ValuedFlagArgument attribute\), 250](#)
[_name \(Base.Executable.ValuedFlagListArgument attribute\), 251](#)
[_name \(Base.Executable.WindowsCommandArgument attribute\), 247](#)
[_name \(Base.Executable.WindowsFlagArgument attribute\), 249](#)
[_name \(Base.Executable.WindowsTupleArgument attribute\), 254](#)
[_name \(Base.Executable.WindowsValuedFlagArgument attribute\), 251](#)
[_name \(Base.Executable.WindowsValuedFlagListArgument attribute\), 251](#)
[_name \(ToolChain.Aldec.ActiveHDL.VHDLCompiler.FlagNoRangeCheck attribute\), 352](#)
[_name \(ToolChain.Aldec.ActiveHDL.VHDLCompiler.SwitchVHDLLibrary attribute\), 352](#)
[_name \(ToolChain.Aldec.ActiveHDL.VHDLCompiler.SwitchVHDLVersion attribute\), 352](#)
[_name \(ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator.Switch attribute\), 353](#)
[_name \(ToolChain.Aldec.RivieraPRO.VHDLCompiler.SwitchVHDLLibrary attribute\), 359](#)
[_name \(ToolChain.Aldec.RivieraPRO.VHDLCompiler.SwitchVHDLVersion attribute\), 359](#)
[_name \(ToolChain.Aldec.RivieraPRO.VHDLSimulator.FlagCommandLine attribute\), 360](#)
[_name \(ToolChain.Aldec.RivieraPRO.VHDLSimulator.SwitchBatchCommand attribute\), 360](#)
[_name \(ToolChain.Aldec.RivieraPRO.VHDLSimulator.SwitchTimeResolution attribute\), 360](#)
[_name \(ToolChain.Altera.Quartus.Map.SwitchArgumentFile attribute\), 375](#)
[_name \(ToolChain.Altera.Quartus.Map.SwitchDeviceFamily attribute\), 375](#)
[_name \(ToolChain.Altera.Quartus.Map.SwitchDevicePart attribute\), 375](#)
[_name \(ToolChain.Altera.Quartus.TclShell.SwitchShell attribute\), 376](#)
[_name \(ToolChain.GHDL.GHDL.ArgListLibraryReferences attribute\), 386](#)
[_name \(ToolChain.GHDL.GHDL.CmdAnalyze attribute\), 384](#)
[_name \(ToolChain.GHDL.GHDL.CmdElaborate attribute\), 384](#)
[_name \(ToolChain.GHDL.GHDL.CmdRun attribute\), 384](#)
[_name \(ToolChain.GHDL.GHDL.FlagDebug attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagExplicit attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagMultiByteComments attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagNoVitalChecks attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagPSL attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagProfileArcs attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagRelaxedRules attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagSynBinding attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagTestCoverage attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.FlagVerbose attribute\), 384](#)
[_name \(ToolChain.GHDL.GHDL.FlagWarnBinding attribute\), 385](#)
[_name \(ToolChain.GHDL.GHDL.SwitchAssemblerOption attribute\), 385](#)

attribute), 385
 _name (ToolChain.GHDL.GHDL.SwitchCompilerOption attribute), 385
 _name (ToolChain.GHDL.GHDL.SwitchFastWaveform attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchGHDLWaveform attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchIEEEAsserts attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchIEEEFlavor attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchLinkerOption attribute), 385
 _name (ToolChain.GHDL.GHDL.SwitchStopDelta attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchVCDGZWaveform attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchVCDWaveform attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchVHDLLibrary attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchVHDLVersion attribute), 386
 _name (ToolChain.GHDL.GHDL.SwitchWaveformOptionFile attribute), 386
 _name (ToolChain.GHDL.GHDLAnalyze.ArgListLibraryReferences attribute), 387
 _name (ToolChain.GHDL.GHDLAnalyze.CmdAnalyze attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.CmdElaborate attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.CmdRun attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagDebug attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagExplicit attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagMultiByteComments attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagNoVitalChecks attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagPSL attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagProfileArcs attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagRelaxedRules attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagSynBinding attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagTestCoverage attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagVerbose attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.FlagWarnBinding attribute), 388
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchAssemblerOption attribute), 389
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchCompilerOption attribute), 389
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchFastWaveform attribute), 389
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchGHDLWaveform attribute), 389
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchIEEEAsserts attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchIEEEFlavor attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchLinkerOption attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchStopDelta attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchVCDGZWaveform attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchVCDWaveform attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchVHDLLibrary attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchVHDLVersion attribute), 390
 _name (ToolChain.GHDL.GHDLAnalyze.SwitchWaveformOptionFile attribute), 390
 _name (ToolChain.GHDL.GHDLElaborate.ArgListLibraryReferences attribute), 390
 _name (ToolChain.GHDL.GHDLElaborate.CmdAnalyze attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.CmdElaborate attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.CmdRun attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagDebug attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagExplicit attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagMultiByteComments attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagNoVitalChecks attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagPSL attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagProfileArcs attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagRelaxedRules attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagSynBinding attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagTestCoverage attribute), 391
 _name (ToolChain.GHDL.GHDLElaborate.FlagVerbose attribute), 392
 _name (ToolChain.GHDL.GHDLElaborate.FlagWarnBinding attribute), 392
 _name (ToolChain.GHDL.GHDLElaborate.SwitchAssemblerOption attribute), 392
 _name (ToolChain.GHDL.GHDLElaborate.SwitchCompilerOption attribute), 392
 _name (ToolChain.GHDL.GHDLElaborate.SwitchFastWaveform attribute), 392

<code>_name (ToolChain.GHDL.GHDLRun.SwitchGHDLElaborate attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchIEEEAsserts attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchIEEEFlavor attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchIEEEFlavor attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchLinkerOptions attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchLinkerOptions attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchStopDelta attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchStopDelta attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchVCDGZWaveform attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchVCDGZWaveform attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchVCDWaveform attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchVCDWaveform attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchVHDLLibrary attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchVHDLLibrary attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchVHDLVersion attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchVHDLVersion attribute), 396</code>
<code>_name (ToolChain.GHDL.GHDLRun.SwitchWaveformOptionFile attribute), 393</code>	<code>_name (ToolChain.GHDL.GHDLRun.SwitchWaveformOptionFile attribute), 396</code>
<code>_name (ToolChain.GNU.Make.SwitchGui attribute), 397</code>	
<code>_name (ToolChain.GTKWave.GTKWave.SwitchDumpFile attribute), 401</code>	
<code>_name (ToolChain.GTKWave.GTKWave.SwitchSaveFile attribute), 401</code>	
<code>_name (ToolChain.Git.GitConfig.Command attribute), 409</code>	
<code>_name (ToolChain.Git.GitConfig.SwitchRemoveSection attribute), 410</code>	
<code>_name (ToolChain.Git.GitConfig.SwitchUnset attribute), 410</code>	
<code>_name (ToolChain.Git.GitConfig.Switch_Version attribute), 410</code>	
<code>_name (ToolChain.Git.GitConfig.ValueFilterClean attribute), 410</code>	
<code>_name (ToolChain.Git.GitConfig.ValueFilterSmudge attribute), 410</code>	
<code>_name (ToolChain.Git.GitDescribe.Command attribute), 408</code>	
<code>_name (ToolChain.Git.GitDescribe.SwitchAbbrev attribute), 408</code>	
<code>_name (ToolChain.Git.GitDescribe.SwitchTags attribute), 409</code>	
<code>_name (ToolChain.Git.GitDescribe.Switch_Version attribute), 409</code>	
<code>_name (ToolChain.Git.GitRevList.Command attribute), 407</code>	
<code>_name (ToolChain.Git.GitRevList.SwitchMaxCount attribute), 407</code>	
<code>_name (ToolChain.Git.GitRevList.SwitchTags attribute), 407</code>	
<code>_name (ToolChain.Git.GitRevList.Switch_Version attribute), 408</code>	
<code>_name (ToolChain.Git.GitRevParse.Command attribute), 406</code>	
<code>_name (ToolChain.Git.GitRevParse.SwitchGitDir attribute), 406</code>	
<code>_name (ToolChain.Git.GitRevParse.SwitchInsideWorkingTree attribute), 406</code>	

_name (ToolChain.Git.GitRevParse.SwitchShowTopLevel_attribute), 406	_name (ToolChain.Mentor.ModelSim.VHDLCompiler.SwitchFSMVerbose_attribute), 448
_name (ToolChain.Git.GitRevParse.Switch_Version_attribute), 407	_name (ToolChain.Mentor.ModelSim.VHDLCompiler.SwitchModelSim_attribute), 448
_name (ToolChain.Git.GitSCM.Switch_Version_attribute), 405	_name (ToolChain.Mentor.ModelSim.VHDLCompiler.SwitchVHDLLibrary_attribute), 450
_name (ToolChain.Intel.Quartus.Map.SwitchArgumentFilename_attribute), 422	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.ArgKeepStdOut_attribute), 453
_name (ToolChain.Intel.Quartus.Map.SwitchDeviceFamily_attribute), 423	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.ArgLogFile_attribute), 453
_name (ToolChain.Intel.Quartus.Map.SwitchDevicePart_attribute), 423	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.ArgOnFinishMode_attribute), 453
_name (ToolChain.Lattice.Diamond.Synth.SwitchProjectFile_attribute), 428	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.ArgVHDLLibrary_attribute), 453
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.ArgLogLevel_attribute), 450	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagBatchMode_attribute), 451
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagDisableExpressionCoverage_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagCommandLine_attribute), 451
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagDisableFPGAExceptionMonitor_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagDisableCover_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagDisableFPGASynthesis_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagDisableKeepA_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagDisableFPGATranslator_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagDisableOptim_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagDisableFPGAVhdlSyn_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagDisablePSL_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagEnableExpressionCoverage_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnableCover_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagEnableFSMD_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnableFSMD_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagEnableKeepA_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnableKeepA_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagEnableOptim_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnableOptim_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagEnableOptim_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnableOptim_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagExplicit_attribute), 448	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagEnablePSL_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagForceLanguage_attribute), 450	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagForceLanguage_attribute), 453
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagQuiet_attribute), 448	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagGuiMode_attribute), 451
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagRange_attribute), 448	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagQuietMode_attribute), 451
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagRelaxLanguage_attribute), 450	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagRelaxLanguage_attribute), 453
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagReportAsError_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagReportAsError_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagReportAsFatal_attribute), 450	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagReportAsFatal_attribute), 453
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagReportAsNote_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagReportAsNote_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagReportAsWarning_attribute), 449	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.FlagReportAsWarning_attribute), 452
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.FlagSwitch_batch_attribute), 448	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.SwitchBatchCom_attribute), 451
_name (ToolChain.Mentor.ModelSim.VHDLCompiler.SwitchGo_attribute), 448	_name (ToolChain.Mentor.ModelSim.VHDLSimulator.SwitchModelSim_attribute), 448

attribute), 452

_name (ToolChain.Mentor.ModelSim.VHDL.Simulator.SwitchIntStyle attribute), 453

_name (ToolChain.Windows.Cmd.SwitchCommand attribute), 465

_name (ToolChain.Xilinx.ISE.CoreGenerator.FlagRegenerate attribute), 473

_name (ToolChain.Xilinx.ISE.CoreGenerator.SwitchBatchFile attribute), 473

_name (ToolChain.Xilinx.ISE.CoreGenerator.SwitchProjectFile attribute), 473

_name (ToolChain.Xilinx.ISE.Fuse.FlagIncremental attribute), 469

_name (ToolChain.Xilinx.ISE.Fuse.FlagRangeCheck attribute), 469

_name (ToolChain.Xilinx.ISE.Fuse.SwitchMultiThreading attribute), 469

_name (ToolChain.Xilinx.ISE.Fuse.SwitchOutputFile attribute), 469

_name (ToolChain.Xilinx.ISE.Fuse.SwitchProjectFile attribute), 469

_name (ToolChain.Xilinx.ISE.Fuse.SwitchTimeResolution attribute), 469

_name (ToolChain.Xilinx.ISE.ISESimulator.FlagGuiMode attribute), 470

_name (ToolChain.Xilinx.ISE.ISESimulator.SwitchLogFile attribute), 470

_name (ToolChain.Xilinx.ISE.ISESimulator.SwitchTclBatchFile attribute), 470

_name (ToolChain.Xilinx.ISE.ISESimulator.SwitchWaveformFile attribute), 471

_name (ToolChain.Xilinx.ISE.Xst.SwitchIntStyle attribute), 471

_name (ToolChain.Xilinx.ISE.Xst.SwitchReportFile attribute), 472

_name (ToolChain.Xilinx.ISE.Xst.SwitchXstFile attribute), 472

_name (ToolChain.Xilinx.Vivado.Synth.SwitchLogFile attribute), 482

_name (ToolChain.Xilinx.Vivado.Synth.SwitchMode attribute), 482

_name (ToolChain.Xilinx.Vivado.Synth.SwitchSourceFile attribute), 482

_name (ToolChain.Xilinx.Vivado.XElab.FlagRangeCheck attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchDebug attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchLogFile attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchMultiThreading attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchOptimization attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchProjectFile attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchSnapshot attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchTimeResolution attribute), 479

attribute), 479

_name (ToolChain.Xilinx.Vivado.XElab.SwitchVerbose attribute), 479

_name (ToolChain.Xilinx.Vivado.XSim.FlagGuiMode attribute), 480

_name (ToolChain.Xilinx.Vivado.XSim.SwitchLogFile attribute), 480

_name (ToolChain.Xilinx.Vivado.XSim.SwitchTclBatchFile attribute), 481

_name (ToolChain.Xilinx.Vivado.XSim.SwitchWaveformFile attribute), 481

_options() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507

_pattern (Base.Executable.CommandArgument attribute), 246

_pattern (Base.Executable.FlagArgument attribute), 248

_pattern (Base.Executable.LongCommandArgument attribute), 246

_pattern (Base.Executable.LongFlagArgument attribute), 249

_pattern (Base.Executable.LongValuedFlagArgument attribute), 250

_pattern (Base.Executable.LongValuedFlagListArgument attribute), 252

_pattern (Base.Executable.ShortCommandArgument attribute), 246

_pattern (Base.Executable.ShortFlagArgument attribute), 248

_pattern (Base.Executable.ShortValuedFlagArgument attribute), 250

_pattern (Base.Executable.ShortValuedFlagListArgument attribute), 251

_pattern (Base.Executable.StringArgument attribute), 247

_pattern (Base.Executable.StringListArgument attribute), 247

_pattern (Base.Executable.ValuedFlagArgument attribute), 249

_pattern (Base.Executable.ValuedFlagListArgument attribute), 251

_pattern (Base.Executable.WindowsCommandArgument attribute), 247

_pattern (Base.Executable.WindowsFlagArgument attribute), 249

_pattern (Base.Executable.WindowsValuedFlagArgument attribute), 251

_pattern (Base.Executable.WindowsValuedFlagListArgument attribute), 252

_pattern (ToolChain.Aldec.ActiveHDL.VHDLCompiler.SwitchVHDLVer attribute), 352

_pattern (ToolChain.Aldec.RivieraPRO.VHDLCompiler.SwitchVHDLVer attribute), 359

_pattern (ToolChain.GHDL.GHDL.ArgListLibraryReferences attribute), 386

_pattern (ToolChain.GHDL.GHDL.SwitchAssemblerOption attribute), 385

_pattern (ToolChain.GHDL.GHDL.SwitchCompilerOption attribute), 385

- attribute), 385
- _pattern (ToolChain.GHDL.GHDL.SwitchLinkerOption attribute), 385
- _pattern (ToolChain.GHDL.GHDLAnalyze.ArgListLibraryReference attribute), 387
- _pattern (ToolChain.GHDL.GHDLAnalyze.SwitchAssemblerOption attribute), 389
- _pattern (ToolChain.GHDL.GHDLAnalyze.SwitchCompilerOption attribute), 389
- _pattern (ToolChain.GHDL.GHDLAnalyze.SwitchLinkerOption attribute), 390
- _pattern (ToolChain.GHDL.GHDLElaborate.ArgListLibraryReference attribute), 390
- _pattern (ToolChain.GHDL.GHDLElaborate.SwitchAssemblerOption attribute), 392
- _pattern (ToolChain.GHDL.GHDLElaborate.SwitchCompilerOption attribute), 392
- _pattern (ToolChain.GHDL.GHDLElaborate.SwitchLinkerOption attribute), 393
- _pattern (ToolChain.GHDL.GHDLRun.ArgListLibraryReference attribute), 394
- _pattern (ToolChain.GHDL.GHDLRun.SwitchAssemblerOption attribute), 396
- _pattern (ToolChain.GHDL.GHDLRun.SwitchCompilerOption attribute), 396
- _pattern (ToolChain.GHDL.GHDLRun.SwitchLinkerOption attribute), 396
- _pattern (ToolChain.Git.GitConfig.ValueFilterClean attribute), 410
- _pattern (ToolChain.Git.GitConfig.ValueFilterSmudge attribute), 410
- _pattern (ToolChain.Mentor.ModelSim.VHDLCompiler.SwitchVHDLVersion attribute), 450
- _pattern (ToolChain.Xilinx.Vivado.XELab.SwitchOptimization attribute), 479
- _read() (lib.ExtendedConfigParser.ExtendedConfigParser method), 508
- _replace() (ToolChain.EditionDescription method), 495
- _section (ToolChain.Aldec.ActiveHDL.Configuration attribute), 348
- _section (ToolChain.Aldec.Configuration attribute), 362
- _section (ToolChain.Aldec.RivieraPRO.Configuration attribute), 355
- _section (ToolChain.Altera.Configuration attribute), 379
- _section (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 367
- _section (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 369
- _section (ToolChain.Altera.ModelSim.Configuration attribute), 366
- _section (ToolChain.Altera.Quartus.Configuration attribute), 373
- _section (ToolChain.Configuration attribute), 489
- _section (ToolChain.GHDL.Configuration attribute), 382
- _section (ToolChain.GTKWave.Configuration attribute), 399
- _section (ToolChain.Git.Configuration attribute), 403
- _section (ToolChain.Intel.Configuration attribute), 423
- _section (ToolChain.Intel.ModelSim.Configuration attribute), 414
- _section (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 414
- _section (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 417
- _section (ToolChain.Intel.Quartus.Configuration attribute), 419
- _section (ToolChain.Lattice.Configuration attribute), 433
- _section (ToolChain.Lattice.Diamond.Configuration attribute), 426
- _section (ToolChain.Lattice.Synplify.Configuration attribute), 432
- _section (ToolChain.Mentor.Configuration attribute), 458
- _section (ToolChain.Mentor.ModelSim.Configuration attribute), 439
- _section (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 439
- _section (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 441
- _section (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute), 444
- _section (ToolChain.Mentor.QuestaSim.Configuration attribute), 455
- _section (ToolChain.PoC.Configuration attribute), 460
- _section (ToolChain.Synopsys.Configuration attribute), 491
- _section (ToolChain.ToolConfiguration attribute), 493
- _section (ToolChain.VendorConfiguration attribute), 491
- _section (ToolChain.Xilinx.Configuration attribute), 485
- _section (ToolChain.Xilinx.ISE.Configuration attribute), 467
- _section (ToolChain.Xilinx.Vivado.Configuration attribute), 476
- _source (ToolChain.EditionDescription attribute), 495
- _switchPattern (Base.Executable.LongTupleArgument attribute), 253
- _switchPattern (Base.Executable.ShortTupleArgument attribute), 253
- _switchPattern (Base.Executable.TupleArgument attribute), 253
- _switchPattern (Base.Executable.WindowsTupleArgument attribute), 254
- _template (ToolChain.Aldec.ActiveHDL.Configuration attribute), 348
- _template (ToolChain.Aldec.Configuration attribute), 362
- _template (ToolChain.Aldec.RivieraPRO.Configuration attribute), 355
- _template (ToolChain.Altera.Configuration attribute), 379

[_template \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute\), 355](#)
[_template \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute\), 367](#)
[_template \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute\), 367](#)
[_template \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute\), 369](#)
[_template \(ToolChain.Altera.ModelSim.Configuration attribute\), 366](#)
[_template \(ToolChain.Altera.Quartus.Configuration attribute\), 373](#)
[_template \(ToolChain.Configuration attribute\), 489](#)
[_template \(ToolChain.GHDL.Configuration attribute\), 382](#)
[_template \(ToolChain.GTKWave.Configuration attribute\), 399](#)
[_template \(ToolChain.Git.Configuration attribute\), 403](#)
[_template \(ToolChain.Intel.Configuration attribute\), 423](#)
[_template \(ToolChain.Intel.ModelSim.Configuration attribute\), 414](#)
[_template \(ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute\), 414](#)
[_template \(ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute\), 414](#)
[_template \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute\), 417](#)
[_template \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute\), 417](#)
[_template \(ToolChain.Intel.Quartus.Configuration attribute\), 421](#)
[_template \(ToolChain.Lattice.Configuration attribute\), 433](#)
[_template \(ToolChain.Lattice.Diamond.Configuration attribute\), 426](#)
[_template \(ToolChain.Lattice.Synplify.Configuration attribute\), 432](#)
[_template \(ToolChain.Mentor.Configuration attribute\), 458](#)
[_template \(ToolChain.Mentor.ModelSim.Configuration attribute\), 439](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute\), 444](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute\), 439](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute\), 446](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute\), 441](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute\), 444](#)
[_template \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute\), 444](#)
[_template \(ToolChain.Mentor.QuestaSim.Configuration attribute\), 455](#)
[_template \(ToolChain.Mentor.QuestaSim.Configuration attribute\), 455](#)
[_template \(ToolChain.PoC.Configuration attribute\), 460](#)
[_template \(ToolChain.PoC.Configuration attribute\), 460](#)
[_template \(ToolChain.Synopsys.Configuration attribute\), 462](#)
[_template \(ToolChain.ToolConfiguration attribute\), 493](#)
[_template \(ToolChain.VendorConfiguration attribute\), 491](#)
[_template \(ToolChain.Xilinx.Configuration attribute\), 485](#)
[_template \(ToolChain.Xilinx.ISE.Configuration attribute\), 467](#)
[_template \(ToolChain.Xilinx.Vivado.Configuration attribute\), 476](#)
[_toolName \(ToolChain.Aldec.ActiveHDL.Configuration attribute\), 348](#)
[_toolName \(ToolChain.Aldec.RivieraPRO.Configuration attribute\), 355](#)
[_toolName \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute\), 367](#)
[_toolName \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute\), 367](#)
[_toolName \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute\), 369](#)
[_toolName \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute\), 369](#)
[_toolName \(ToolChain.Altera.ModelSim.Configuration attribute\), 366](#)
[_toolName \(ToolChain.Altera.Quartus.Configuration attribute\), 373](#)
[_toolName \(ToolChain.Configuration attribute\), 489](#)
[_toolName \(ToolChain.GHDL.Configuration attribute\), 382](#)
[_toolName \(ToolChain.GTKWave.Configuration attribute\), 399](#)
[_toolName \(ToolChain.Git.Configuration attribute\), 403](#)
[_toolName \(ToolChain.Intel.ModelSim.Configuration attribute\), 414](#)
[_toolName \(ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute\), 414](#)
[_toolName \(ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute\), 414](#)
[_toolName \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute\), 417](#)
[_toolName \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute\), 417](#)
[_toolName \(ToolChain.Intel.Quartus.Configuration attribute\), 419](#)
[_toolName \(ToolChain.Lattice.Diamond.Configuration attribute\), 426](#)
[_toolName \(ToolChain.Lattice.Synplify.Configuration attribute\), 432](#)
[_toolName \(ToolChain.Mentor.ModelSim.Configuration attribute\), 437](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute\), 439](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute\), 439](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute\), 441](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute\), 441](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute\), 444](#)
[_toolName \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute\), 444](#)
[_toolName \(ToolChain.Mentor.ModelSim.Selector attribute\), 446](#)
[_toolName \(ToolChain.Mentor.QuestaSim.Configuration attribute\), 455](#)
[_toolName \(ToolChain.Mentor.QuestaSim.Configuration attribute\), 455](#)
[_toolName \(ToolChain.PoC.Configuration attribute\), 460](#)
[_toolName \(ToolChain.PoC.Configuration attribute\), 460](#)
[_toolName \(ToolChain.ToolConfiguration attribute\), 493](#)
[_toolName \(ToolChain.ToolConfiguration attribute\), 493](#)
[_toolName \(ToolChain.ToolSelector attribute\), 495](#)
[_toolName \(ToolChain.Xilinx.ISE.Configuration attribute\), 467](#)
[_toolName \(ToolChain.Xilinx.ISE.Configuration attribute\), 467](#)
[_toolName \(ToolChain.Xilinx.Vivado.Configuration attribute\), 476](#)
[_toolName \(ToolChain.Xilinx.Vivado.Configuration attribute\), 476](#)
[_unify_values\(\) \(lib.ExtendedConfigParser.ExtendedConfigParser method\), 510](#)
[_unify_values\(\) \(lib.ExtendedConfigParser.ExtendedConfigParser method\), 510](#)
[_validate_value_types\(\) \(lib.ExtendedConfigParser.ExtendedConfigParser method\), 509](#)
[_validate_value_types\(\) \(lib.ExtendedConfigParser.ExtendedConfigParser method\), 509](#)
[_value \(Base.Executable.CommandArgument attribute\), 246](#)
[_value \(Base.Executable.CommandArgument attribute\), 246](#)
[_value \(Base.Executable.CommandLineArgument attribute\), 245](#)
[_value \(Base.Executable.CommandLineArgument attribute\), 245](#)
[_value \(Base.Executable.ExecutableArgument attribute\), 245](#)
[_value \(Base.Executable.ExecutableArgument attribute\), 245](#)

attribute), 453	attribute), 479
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.ArgOnFile (ToolChain.Xilinx.Vivado.XELab.SwitchVerbose attribute), 453	attribute), 479
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.ArgVhdl (ToolChain.Xilinx.Vivado.XSim.Executable attribute), 453	attribute), 480
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.Executable (ToolChain.Xilinx.Vivado.XSim.FlagGuiMode attribute), 451	attribute), 480
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagBatchMode (ToolChain.Xilinx.Vivado.XSim.SwitchLogFile attribute), 451	attribute), 480
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagGuiMode (ToolChain.Xilinx.Vivado.XSim.SwitchSnapshot attribute), 451	attribute), 481
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagTclBatchFile (ToolChain.Xilinx.Vivado.XSim.SwitchTclBatchFile attribute), 451	attribute), 481
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagQuietMode (ToolChain.Xilinx.Vivado.XSim.SwitchWaveformFile attribute), 451	attribute), 481
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagReportError (Base.Executable.LongTupleArgument attribute), 452	attribute), 254
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagReportFatal (Base.Executable.ShortTupleArgument attribute), 453	attribute), 253
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagReportNote (Base.Executable.TupleArgument attribute), 452	attribute), 253
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.FlagReportWarning (Base.Executable.WindowsTupleArgument attribute), 453	attribute), 254
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.SwitchBatchMode (ToolChain.Aldec.ActiveHDL.Configuration attribute), 451	attribute), 348
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.SwitchModelSim (ToolChain.Aldec.Configuration attribute), 452	attribute), 362
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.SwitchRTL (ToolChain.Aldec.RivieraPRO.Configuration attribute), 453	attribute), 355
_value (ToolChain.Mentor.ModelSim.VHDL.Simulator.SwitchTopLevel (ToolChain.Altera.Configuration attribute), 453	attribute), 379
_value (ToolChain.Xilinx.Vivado.Synth.Executable attribute), 482	_vendor (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 369
_value (ToolChain.Xilinx.Vivado.Synth.SwitchLogFile attribute), 482	_vendor (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 371
_value (ToolChain.Xilinx.Vivado.Synth.SwitchMode attribute), 482	_vendor (ToolChain.Altera.ModelSim.Configuration attribute), 364
_value (ToolChain.Xilinx.Vivado.Synth.SwitchSourceFile attribute), 482	_vendor (ToolChain.Altera.Quartus.Configuration attribute), 373
_value (ToolChain.Xilinx.Vivado.XELab.ArgTopLevel attribute), 479	_vendor (ToolChain.Configuration attribute), 489
_value (ToolChain.Xilinx.Vivado.XELab.Executable attribute), 478	_vendor (ToolChain.GHDL.Configuration attribute), 382
_value (ToolChain.Xilinx.Vivado.XELab.FlagRangeCheck attribute), 479	_vendor (ToolChain.GTKWave.Configuration attribute), 399
_value (ToolChain.Xilinx.Vivado.XELab.SwitchDebug attribute), 479	_vendor (ToolChain.Git.Configuration attribute), 403
_value (ToolChain.Xilinx.Vivado.XELab.SwitchLogFile attribute), 479	_vendor (ToolChain.Intel.Configuration attribute), 423
_value (ToolChain.Xilinx.Vivado.XELab.SwitchMultiThreading attribute), 479	_vendor (ToolChain.Intel.ModelSim.Configuration attribute), 412
_value (ToolChain.Xilinx.Vivado.XELab.SwitchOptimization attribute), 479	_vendor (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 416
_value (ToolChain.Xilinx.Vivado.XELab.SwitchProjectFile attribute), 479	_vendor (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 419
_value (ToolChain.Xilinx.Vivado.XELab.SwitchSnapshot attribute), 479	_vendor (ToolChain.Intel.Quartus.Configuration attribute), 419
_value (ToolChain.Xilinx.Vivado.XELab.SwitchTimeResolution attribute), 479	_vendor (ToolChain.Lattice.Configuration attribute), 433
	_vendor (ToolChain.Lattice.Diamond.Configuration attribute), 426

<code>_vendor</code> (ToolChain.Lattice.Synplify.Configuration attribute), 432	<code>AddElement()</code> (Parser.FilesCodeDOM.ListConstructorExpression method), 318
<code>_vendor</code> (ToolChain.Mentor.Configuration attribute), 458	<code>AddExternalVHDLLibraries()</code> (Base.Project.Project method), 261
<code>_vendor</code> (ToolChain.Mentor.ModelSim.Configuration attribute), 437	<code>AddExternalVHDLLibraries()</code> (DataBase.Solution.VirtualProject method), 308
<code>_vendor</code> (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 441	<code>AddExternalVHDLLibraries()</code> (ToolChain.Altera.Quartus.QuartusProject method), 377
<code>_vendor</code> (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 444	<code>AddExternalVHDLLibraries()</code> (ToolChain.Xilinx.ISE.ISEProject method), 474
<code>_vendor</code> (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute), 446	<code>AddExternalVHDLLibraries()</code> (ToolChain.Xilinx.Vivado.VivadoProject method), 483
<code>_vendor</code> (ToolChain.Mentor.QuestaSim.Configuration attribute), 455	<code>AddFile()</code> (Base.Project.FileSet method), 261
<code>_vendor</code> (ToolChain.PoC.Configuration attribute), 460	<code>AddFile()</code> (Base.Project.Project method), 261
<code>_vendor</code> (ToolChain.Synopsys.Configuration attribute), 462	<code>AddFile()</code> (Base.Project.VHDLLibrary method), 261
<code>_vendor</code> (ToolChain.ToolConfiguration attribute), 495	<code>AddFile()</code> (DataBase.Solution.VirtualProject method), 308
<code>_vendor</code> (ToolChain.VendorConfiguration attribute), 492	<code>AddFile()</code> (ToolChain.Altera.Quartus.QuartusProject method), 377
<code>_vendor</code> (ToolChain.Xilinx.Configuration attribute), 485	<code>AddFile()</code> (ToolChain.Xilinx.ISE.ISEProject method), 474
<code>_vendor</code> (ToolChain.Xilinx.ISE.Configuration attribute), 467	<code>AddFile()</code> (ToolChain.Xilinx.Vivado.VivadoProject method), 483
<code>_vendor</code> (ToolChain.Xilinx.Vivado.Configuration attribute), 476	<code>AddFileSet()</code> (Base.Project.Project method), 261
<code>_write_section()</code> (lib.ExtendedConfigParser.ExtendedConfigParser method), 509	<code>AddFileSet()</code> (DataBase.Solution.VirtualProject method), 308
A	
<code>ActiveHDL</code> (class in ToolChain.Aldec.ActiveHDL), 350	<code>AddFileSet()</code> (ToolChain.Altera.Quartus.QuartusProject method), 377
<code>ActiveHDLEditions</code> (class in ToolChain.Aldec.ActiveHDL), 348	<code>AddFileSet()</code> (ToolChain.Xilinx.ISE.ISEProject method), 474
<code>ActiveHDLException</code> , 347	<code>AddFileSet()</code> (ToolChain.Xilinx.Vivado.VivadoProject method), 483
<code>add_section()</code> (lib.ExtendedConfigParser.ExtendedConfigParser method), 509	<code>AddLibrary()</code> (DataBase.Entity.NamespaceRoot method), 294
<code>AddChoice()</code> (lib.CodeDOM.BlockedStatement class method), 505	<code>AddSolution()</code> (DataBase.Solution.Repository method), 306
<code>AddChoice()</code> (lib.CodeDOM.ExpressionChoice class method), 505	<code>AddSourceFile()</code> (Base.Project.FileSet method), 261
<code>AddChoice()</code> (Parser.FilesCodeDOM.BlockedStatement class method), 317	<code>AddSourceFile()</code> (Base.Project.Project method), 261
<code>AddChoice()</code> (Parser.FilesCodeDOM.IfThenElseExpressions class method), 317	<code>AddSourceFile()</code> (DataBase.Solution.VirtualProject method), 308
<code>AddChoice()</code> (Parser.FilesCodeDOM.ListElementExpressions class method), 318	<code>AddSourceFile()</code> (ToolChain.Altera.Quartus.QuartusProject method), 377
<code>AddChoice()</code> (Parser.FilesCodeDOM.PathExpressions class method), 318	<code>AddSourceFile()</code> (ToolChain.Xilinx.ISE.ISEProject method), 474
<code>AddChoice()</code> (Parser.RulesCodeDOM.DocumentStatements class method), 328	<code>AddSourceFile()</code> (ToolChain.Xilinx.Vivado.VivadoProject method), 483
<code>AddChoice()</code> (Parser.RulesCodeDOM.InFileStatements class method), 327	<code>AddStatement()</code> (lib.CodeDOM.BlockStatement method), 504
<code>AddChoice()</code> (Parser.RulesCodeDOM.PostProcessStatements class method), 327	<code>AddStatement()</code> (lib.CodeDOM.ConditionalBlockStatement method), 504
<code>AddChoice()</code> (Parser.RulesCodeDOM.PreProcessStatements class method), 327	<code>AddStatement()</code> (Parser.FilesCodeDOM.Document method), 323
	<code>AddStatement()</code> (Parser.FilesCodeDOM.ElseIfStatement method), 323

AddStatement() (Parser.FilesCodeDOM.ElseStatement method), 323
AddStatement() (Parser.FilesCodeDOM.IfStatement method), 322
AddStatement() (Parser.RulesCodeDOM.Document method), 330
AddStatement() (Parser.RulesCodeDOM.FileStatement method), 329
AddStatement() (Parser.RulesCodeDOM.PostProcessRulesAppendLineTask method), 330
AddStatement() (Parser.RulesCodeDOM.PreProcessRulesStatement method), 329
AddStatement() (Parser.RulesCodeDOM.ProcessRulesBlockStatement method), 329
AddSynthesis() (DataBase.TestCase.SynthesisSuite method), 314
AddTestCase() (DataBase.TestCase.TestSuite method), 313
Aldec_ActiveHDL (Base.Project.ToolChain attribute), 259
Aldec_aSim (Base.Project.Tool attribute), 260
Aldec_RivieraPRO (Base.Project.ToolChain attribute), 259
Aldec_rPro (Base.Project.Tool attribute), 260
AldecActiveHDLEditions (class in ToolChain.Aldec.ActiveHDL), 348
AldecException, 361
All (Base.Logging.Severity attribute), 256
AlphaChars (lib.Parser.Tokenizer.TokenKind attribute), 515
Altera
 Pre-compilation, 50
Altera (DataBase.Config.Vendors attribute), 284
Altera_ModelSim (Base.Project.ToolChain attribute), 259
Altera_Quartus (Base.Project.ToolChain attribute), 259
Altera_Quartus_Map (Base.Project.Tool attribute), 260
AlteraDevices (class in DataBase.Config), 286
AlteraEditionConfiguration (class in ToolChain.Altera.ModelSim), 367
AlteraException, 379
AlteraFamilies (class in DataBase.Config), 285
AlteraModelSimEditions (class in ToolChain.Altera.ModelSim), 364
AlteraQuartus (ToolChain.Altera.Quartus.QuartusEditions attribute), 372
AlteraStarterEditionConfiguration (class in ToolChain.Altera.ModelSim), 369
AlteraSubTypes (class in DataBase.Config), 288
AN (DataBase.Config.XilinxSubTypes attribute), 288
Analyze (Simulator.SimulationState attribute), 345
Analyze() (ToolChain.GHDL.GHDLAnalyze method), 387
AnalyzeError (DataBase.TestCase.SimulationStatus attribute), 311
AndExpression (class in lib.CodeDOM), 501
Any (Base.Project.Environment attribute), 259
Any (Base.Project.Tool attribute), 260
Any (Base.Project.ToolChain attribute), 259
Any (Base.Project.VHDLVersion attribute), 260
append() (Base.Executable.CommandLineArgumentList method), 254
AppendLineRuleMixIn (class in Parser.RulesParser), 331
AppendLineStatement (class in Parser.RulesCodeDOM), 329
AppendLineTask (class in Compiler), 281
AppendPattern (Compiler.AppendLineTask attribute), 281
AppendPattern (Parser.RulesCodeDOM.AppendLineStatement attribute), 329
AppendPattern (Parser.RulesParser.AppendLineRuleMixIn attribute), 331
Architecture (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
ArgParseMixin (class in lib.pyAttribute.ArgParseAttributes), 519
args (Base.Exceptions.CommonException attribute), 243
args (Base.Exceptions.EnvironmentException attribute), 242
args (Base.Exceptions.ExceptionBase attribute), 242
args (Base.Exceptions.NotConfiguredException attribute), 243
args (Base.Exceptions.PlatformNotSupportedException attribute), 242
args (Base.Exceptions.SkipableCommonException attribute), 243
args (Base.Exceptions.SkipableException attribute), 243
args (Base.Executable.ExecutableException attribute), 245
args (Compiler.CompilerException attribute), 280
args (Compiler.SkipableCompilerException attribute), 280
args (lib.Parser.EmptyChoseParserResult attribute), 513
args (lib.Parser.GreedyMatchingParserResult attribute), 513
args (lib.Parser.MatchingParserResult attribute), 513
args (lib.Parser.MismatchingParserResult attribute), 513
args (lib.Parser.ParserException attribute), 512
Args (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute attribute), 517
Args (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute attribute), 518
Args (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute attribute), 518
Args (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute attribute), 518
args (Simulator.PoCSimulationResultNotFoundException attribute), 344
args (Simulator.SimulatorException attribute), 343
args (Simulator.SkipableSimulatorException attribute), 343

args (ToolChain.Aldec.ActiveHDL.ActiveHDLException attribute), 348
 args (ToolChain.Aldec.AldecException attribute), 361
 args (ToolChain.Aldec.RivieraPRO.RivieraPROException attribute), 355
 args (ToolChain.Altera.AlteraException attribute), 379
 args (ToolChain.Altera.ModelSim.ModelSimException attribute), 364
 args (ToolChain.Altera.Quartus.QuartusException attribute), 372
 args (ToolChain.ConfigurationException attribute), 488
 args (ToolChain.GHDL.GHDLException attribute), 381
 args (ToolChain.GHDL.GHDLReanalyzeException attribute), 382
 args (ToolChain.Git.GitException attribute), 403
 args (ToolChain.GNU.GNUException attribute), 397
 args (ToolChain.GTKWave.GTKWaveException attribute), 399
 args (ToolChain.Intel.IntelException attribute), 423
 args (ToolChain.Intel.ModelSim.ModelSimException attribute), 411
 args (ToolChain.Intel.Quartus.QuartusException attribute), 419
 args (ToolChain.Lattice.Diamond.DiamondException attribute), 426
 args (ToolChain.Lattice.LatticeException attribute), 433
 args (ToolChain.Lattice.Synplify.SynplifyException attribute), 430
 args (ToolChain.Mentor.MentorException attribute), 457
 args (ToolChain.Mentor.ModelSim.ModelSimException attribute), 436
 args (ToolChain.Mentor.QuestaSim.QuestaSimException attribute), 455
 args (ToolChain.SkipConfigurationException attribute), 488
 args (ToolChain.Synopsys.SynopsysException attribute), 462
 args (ToolChain.ToolChainException attribute), 487
 args (ToolChain.Windows.WindowsException attribute), 465
 args (ToolChain.Xilinx.ISE.ISEException attribute), 466
 args (ToolChain.Xilinx.Vivado.VivadoException attribute), 476
 args (ToolChain.Xilinx.XilinxException attribute), 484
 ArgumentAttribute (class in lib.pyAttribute.ArgParseAttributes), 517
 Arria (DataBase.Config.AlteraFamilies attribute), 285
 Arria2 (DataBase.Config.AlteraDevices attribute), 286
 Arria5 (DataBase.Config.AlteraDevices attribute), 286
 Artix (DataBase.Config.XilinxFamilies attribute), 285
 Artix7 (DataBase.Config.XilinxDevices attribute), 287
 AsArgument() (Base.Executable.CommandArgument method), 246
 AsArgument() (Base.Executable.ExecutableArgument method), 245
 AsArgument() (Base.Executable.FlagArgument method), 248
 AsArgument() (Base.Executable.LongCommandArgument method), 246
 AsArgument() (Base.Executable.LongFlagArgument method), 249
 AsArgument() (Base.Executable.LongTupleArgument method), 253
 AsArgument() (Base.Executable.LongValuedFlagArgument method), 250
 AsArgument() (Base.Executable.LongValuedFlagListArgument method), 252
 AsArgument() (Base.Executable.PathArgument method), 248
 AsArgument() (Base.Executable.ShortCommandArgument method), 246
 AsArgument() (Base.Executable.ShortFlagArgument method), 248
 AsArgument() (Base.Executable.ShortTupleArgument method), 253
 AsArgument() (Base.Executable.ShortValuedFlagArgument method), 250
 AsArgument() (Base.Executable.ShortValuedFlagListArgument method), 251
 AsArgument() (Base.Executable.StringArgument method), 247
 AsArgument() (Base.Executable.StringListArgument method), 247
 AsArgument() (Base.Executable.TupleArgument method), 253
 AsArgument() (Base.Executable.ValuedFlagArgument method), 249
 AsArgument() (Base.Executable.ValuedFlagListArgument method), 251
 AsArgument() (Base.Executable.WindowsCommandArgument method), 247
 AsArgument() (Base.Executable.WindowsFlagArgument method), 249
 AsArgument() (Base.Executable.WindowsTupleArgument method), 254
 AsArgument() (Base.Executable.WindowsValuedFlagArgument method), 251
 AsArgument() (Base.Executable.WindowsValuedFlagListArgument method), 252
 AskMixIn (class in ToolChain), 488
 AskWildCard (class in DataBase.Entity), 298
 Attribute (class in lib.pyAttribute), 519
 AttributeHelperMixin (class in lib.pyAttribute), 519
B
 Backend (ToolChain.GHDL.GHDL attribute), 384
 Backend (ToolChain.GHDL.GHDLAnalyze attribute), 387
 Backend (ToolChain.GHDL.GHDLElaborate attribute), 391
 Backend (ToolChain.GHDL.GHDLRun attribute), 394
 Base (class in DataBase.Solution), 305

- Base (module), 241
 - Base.Exceptions (module), 241
 - Base.Executable (module), 243
 - Base.Logging (module), 255
 - Base.Project (module), 257
 - Base.Shared (module), 265
 - BaseEnum (class in DataBase.Config), 284
 - BaseFlags (class in DataBase.Entity), 291
 - BaseIndent (Base.Logging.Logger attribute), 256
 - before_get() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
 - before_read() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
 - before_set() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
 - before_write() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
 - BinaryDirectoryPath (ToolChain.GHDL.GHDL attribute), 384
 - BinaryDirectoryPath (ToolChain.GHDL.GHDLAnalyze attribute), 387
 - BinaryDirectoryPath (ToolChain.GHDL.GHDLElaborate attribute), 391
 - BinaryDirectoryPath (ToolChain.GHDL.GHDLRun attribute), 394
 - BinaryDirectoryPath (ToolChain.GTKWave.GTKWave attribute), 401
 - BinaryExpression (class in lib.CodeDOM), 499
 - bits_from_simple_str() (Base.Project.FileTypes class method), 258
 - bits_from_simple_str() (DataBase.Entity.BaseFlags class method), 292
 - bits_from_simple_str() (DataBase.Entity.NetlistKind class method), 293
 - bits_from_simple_str() (DataBase.Entity.TestbenchKind class method), 292
 - bits_from_simple_str() (Simulator.SimulationSteps class method), 344
 - bits_from_str() (Base.Project.FileTypes class method), 258
 - bits_from_str() (DataBase.Entity.BaseFlags class method), 292
 - bits_from_str() (DataBase.Entity.NetlistKind class method), 293
 - bits_from_str() (DataBase.Entity.TestbenchKind class method), 292
 - bits_from_str() (Simulator.SimulationSteps class method), 344
 - BlockedStatement (class in lib.CodeDOM), 505
 - BlockedStatement (class in Parser.FilesCodeDOM), 317
 - BlockStatement (class in lib.CodeDOM), 504
 - Board (Base.Project.Project attribute), 260
 - Board (class in DataBase.Config), 290
 - Board (DataBase.Solution.VirtualProject attribute), 308
 - Board (ToolChain.Altera.Quartus.QuartusProject attribute), 377
 - Board (ToolChain.Xilinx.ISE.ISEProject attribute), 474
 - Board (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
 - BOOLEAN_STATES (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
- ## C
- CABGA (DataBase.Config.Packages attribute), 289
 - CachedReadOnlyProperty (class in lib.Decorators), 506
 - CanByRefParam (class in lib.Functions), 511
 - CaseInsensitive (Parser.RulesCodeDOM.ReplaceStatement attribute), 328
 - CGNetlist (DataBase.Entity.IPCore attribute), 299
 - Changed (ToolChain.ChangeState attribute), 488
 - ChangeState (class in ToolChain), 488
 - CharacterToken (class in lib.Parser), 514
 - CheckDependency() (ToolChain.Aldec.ActiveHDL.Configuration method), 348
 - CheckDependency() (ToolChain.Aldec.Configuration method), 362
 - CheckDependency() (ToolChain.Aldec.RivieraPRO.Configuration method), 355
 - CheckDependency() (ToolChain.Altera.Configuration method), 379
 - CheckDependency() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367
 - CheckDependency() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 369
 - CheckDependency() (ToolChain.Altera.ModelSim.Configuration method), 364
 - CheckDependency() (ToolChain.Altera.Quartus.Configuration method), 373
 - CheckDependency() (ToolChain.Configuration method), 489
 - CheckDependency() (ToolChain.GHDL.Configuration method), 382
 - CheckDependency() (ToolChain.Git.Configuration method), 403
 - CheckDependency() (ToolChain.GTKWave.Configuration method), 399
 - CheckDependency() (ToolChain.Intel.Configuration method), 423
 - CheckDependency() (ToolChain.Intel.ModelSim.Configuration method), 412
 - CheckDependency() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 414
 - CheckDependency() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
 - CheckDependency() (ToolChain.Intel.Quartus.Configuration method), 420
 - CheckDependency() (ToolChain.Lattice.Configuration method), 433
 - CheckDependency() (ToolChain.Lattice.Diamond.Configuration method), 426
 - CheckDependency() (ToolChain.Lattice.Synplify.Configuration method), 430
 - CheckDependency() (ToolChain.Mentor.Configuration method), 458

CheckDependency() (ToolChain.Mentor.ModelSim.Configuration method), 382
 method), 437
 CheckDependency() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration
 method), 439
 CheckDependency() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration
 method), 442
 CheckDependency() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration
 method), 444
 CheckDependency() (ToolChain.Mentor.QuestaSim.Configuration method), 412
 method), 455
 CheckDependency() (ToolChain.PoC.Configuration method), 414
 method), 460
 CheckDependency() (ToolChain.Synopsys.Configuration method), 417
 method), 462
 CheckDependency() (ToolChain.ToolConfiguration method), 420
 method), 493
 CheckDependency() (ToolChain.VendorConfiguration method), 433
 method), 491
 CheckDependency() (ToolChain.Xilinx.Configuration method), 426
 method), 485
 CheckDependency() (ToolChain.Xilinx.ISE.Configuration method), 430
 method), 467
 CheckDependency() (ToolChain.Xilinx.Vivado.Configuration method), 458
 method), 476
 Child (lib.CodeDOM.NotExpression attribute), 499
 Child (lib.CodeDOM.UnaryExpression attribute), 499
 CleanUp (Compiler.CompileState attribute), 281
 clear() (Base.Executable.CommandLineArgumentList method), 254
 clear() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
 clear() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
 Clear() (ToolChain.Git.GitConfig method), 409
 Clear() (ToolChain.Git.GitDescribe method), 408
 Clear() (ToolChain.Git.GitRevList method), 407
 Clear() (ToolChain.Git.GitRevParse method), 406
 Clear() (ToolChain.Git.GitSCM method), 405
 clear_cache() (lib.ExtendedConfigParser.ExtendedInterpolation
 method), 507
 ClearSection() (ToolChain.Aldec.ActiveHDL.Configuration method), 491
 method), 349
 ClearSection() (ToolChain.Aldec.Configuration method), 485
 method), 362
 ClearSection() (ToolChain.Aldec.RivieraPRO.Configuration method), 467
 method), 355
 ClearSection() (ToolChain.Altera.Configuration method), 476
 method), 379
 ClearSection() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration
 method), 367
 ClearSection() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration
 method), 369
 ClearSection() (ToolChain.Altera.ModelSim.Configuration
 method), 365
 ClearSection() (ToolChain.Altera.Quartus.Configuration
 method), 373
 ClearSection() (ToolChain.Configuration method), 489
 ClearSection() (ToolChain.GHDL.Configuration
 method), 489
 ClearSection() (ToolChain.Git.Configuration method), 414
 ClearSection() (ToolChain.GTKWave.Configuration method), 417
 ClearSection() (ToolChain.Intel.Configuration method), 420
 ClearSection() (ToolChain.Intel.ModelSim.Configuration method), 426
 ClearSection() (ToolChain.Intel.ModelSim.IntelEditionConfiguration
 method), 414
 ClearSection() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration
 method), 417
 ClearSection() (ToolChain.Intel.Quartus.Configuration method), 420
 ClearSection() (ToolChain.Lattice.Configuration method), 433
 ClearSection() (ToolChain.Lattice.Diamond.Configuration method), 426
 ClearSection() (ToolChain.Lattice.Synplify.Configuration method), 430
 ClearSection() (ToolChain.Mentor.Configuration method), 437
 ClearSection() (ToolChain.Mentor.ModelSim.Configuration method), 437
 ClearSection() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration
 method), 439
 ClearSection() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration
 method), 442
 ClearSection() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration
 method), 444
 ClearSection() (ToolChain.Mentor.QuestaSim.Configuration method), 455
 ClearSection() (ToolChain.PoC.Configuration method), 460
 ClearSection() (ToolChain.Synopsys.Configuration method), 462
 ClearSection() (ToolChain.ToolConfiguration method), 493
 ClearSection() (ToolChain.VendorConfiguration method), 491
 ClearSection() (ToolChain.Xilinx.Configuration method), 485
 ClearSection() (ToolChain.Xilinx.ISE.Configuration method), 467
 ClearSection() (ToolChain.Xilinx.Vivado.Configuration method), 476
 CLG (DataBase.Config.Packages attribute), 289
 Close() (ToolChain.Altera.Quartus.QuartusProject method), 377
 Close() (ToolChain.Windows), 465
 Cmd.Executable (class in ToolChain.Windows), 465
 Cmd.SwitchCommand (class in ToolChain.Windows), 465
 Cocotb
 Pre-compilation, 55
 Third-Party Libraries, 203
 Cocotb (Base.Project.ToolChain attribute), 259

- Cocotb_QuestaSim (Base.Project.Tool attribute), 260
- COCOTB_SIMBUILD_DIRECTORY (Simulator.CocotbSimulator.Simulator attribute), 334
- CocotbSimulationResultFilter() (in module ToolChain.GNU), 398
- CocotbSourceFile (class in Base.Project), 264
- CocotbSourceFileMixIn (class in Parser.FilesParser), 325
- CocotbStatement (class in Parser.FilesCodeDOM), 319
- CocoTestbench (class in DataBase.Entity), 301
- CocoTestbench (DataBase.Entity.IPCore attribute), 299
- CocoTestbenches (DataBase.Entity.AskWildCard attribute), 298
- CocoTestbenches (DataBase.Entity.StarWildCard attribute), 297
- CocoTestbenches (DataBase.Entity.WildCard attribute), 296
- CodeDOMMeta (class in lib.CodeDOM), 498
- CodeDOMObject (class in lib.CodeDOM), 498
- Command (lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute), 517
- CommandArgument (class in Base.Executable), 245
- CommandAttribute (class in lib.pyAttribute.ArgParseAttributes), 517
- CommandGroupAttribute (class in lib.pyAttribute.ArgParseAttributes), 516
- CommandLineArgument (class in Base.Executable), 245
- CommandLineArgumentList (class in Base.Executable), 254
- CommentLine (class in lib.CodeDOM), 505
- CommentText (lib.CodeDOM.BlockStatement attribute), 504
- CommentText (lib.CodeDOM.ConditionalBlockStatement attribute), 504
- CommentText (lib.CodeDOM.Statement attribute), 504
- CommentText (Parser.FilesCodeDOM.CocotbStatement attribute), 320
- CommentText (Parser.FilesCodeDOM.ConstraintStatement attribute), 320
- CommentText (Parser.FilesCodeDOM.Document attribute), 323
- CommentText (Parser.FilesCodeDOM.ElseIfStatement attribute), 323
- CommentText (Parser.FilesCodeDOM.ElseStatement attribute), 323
- CommentText (Parser.FilesCodeDOM.IfElseIfElseStatement attribute), 323
- CommentText (Parser.FilesCodeDOM.IfStatement attribute), 322
- CommentText (Parser.FilesCodeDOM.IncludeStatement attribute), 322
- CommentText (Parser.FilesCodeDOM.LDCStatement attribute), 320
- CommentText (Parser.FilesCodeDOM.LibraryStatement attribute), 322
- CommentText (Parser.FilesCodeDOM.PathStatement attribute), 321
- CommentText (Parser.FilesCodeDOM.ReportStatement attribute), 322
- CommentText (Parser.FilesCodeDOM.SDCStatement attribute), 320
- CommentText (Parser.FilesCodeDOM.UCFStatement attribute), 321
- CommentText (Parser.FilesCodeDOM.VerilogStatement attribute), 319
- CommentText (Parser.FilesCodeDOM.VHDLStatement attribute), 319
- CommentText (Parser.FilesCodeDOM.XDCStatement attribute), 321
- CommentText (Parser.RulesCodeDOM.AppendLineStatement attribute), 329
- CommentText (Parser.RulesCodeDOM.CopyStatement attribute), 328
- CommentText (Parser.RulesCodeDOM.DeleteStatement attribute), 328
- CommentText (Parser.RulesCodeDOM.Document attribute), 330
- CommentText (Parser.RulesCodeDOM.FileStatement attribute), 329
- CommentText (Parser.RulesCodeDOM.PostProcessRulesStatement attribute), 330
- CommentText (Parser.RulesCodeDOM.PreProcessRulesStatement attribute), 329
- CommentText (Parser.RulesCodeDOM.ProcessRulesBlockStatement attribute), 329
- CommentText (Parser.RulesCodeDOM.ReplaceStatement attribute), 328
- CommonArgumentAttribute (class in lib.pyAttribute.ArgParseAttributes), 518
- CommonException, 243
- CommonSwitchArgumentAttribute (class in lib.pyAttribute.ArgParseAttributes), 518
- CompareExpression (class in lib.CodeDOM), 500
- Compile (Compiler.CompileState attribute), 281
- Compile() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 352
- Compile() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- Compile() (ToolChain.Altera.Quartus.Map method), 375
- Compile() (ToolChain.Intel.Quartus.Map method), 422
- Compile() (ToolChain.Lattice.Diamond.Synth method), 428
- Compile() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450
- Compile() (ToolChain.Xilinx.ISE.Xst method), 472
- Compile() (ToolChain.Xilinx.Vivado.Synth method), 482
- compile-altera.ps1 command line option
- All, 221
 - Clean, 221
 - GHDL, 222
 - Help, 221
 - Questa, 222

- VHDL2008, [222](#)
- VHDL93, [222](#)
- compile-altera.sh command line option
 - all, [221](#)
 - clean, [221](#)
 - ghdl, [221](#)
 - help, [221](#)
 - questa, [221](#)
 - vhdl2008, [221](#)
 - vhdl93, [221](#)
- compile-lattice.ps1 command line option
 - All, [223](#)
 - Clean, [223](#)
 - GHDL, [223](#)
 - Help, [223](#)
 - Questa, [223](#)
 - VHDL2008, [223](#)
 - VHDL93, [223](#)
- compile-lattice.sh command line option
 - all, [222](#)
 - clean, [222](#)
 - ghdl, [222](#)
 - help, [222](#)
 - questa, [222](#)
 - vhdl2008, [223](#)
 - vhdl93, [223](#)
- compile-osvvm.ps1 command line option
 - All, [225](#)
 - Clean, [225](#)
 - GHDL, [225](#)
 - Help, [225](#)
 - Questa, [225](#)
 - VHDL2008, [225](#)
 - VHDL93, [225](#)
- compile-osvvm.sh command line option
 - all, [224](#)
 - clean, [224](#)
 - ghdl, [224](#)
 - help, [224](#)
 - questa, [224](#)
 - vhdl2008, [224](#)
 - vhdl93, [224](#)
- compile-uvvm.ps1 command line option
 - All, [226](#)
 - Clean, [226](#)
 - GHDL, [226](#)
 - Help, [226](#)
 - Questa, [226](#)
 - VHDL2008, [226](#)
 - VHDL93, [226](#)
- compile-uvvm.sh command line option
 - all, [225](#)
 - clean, [225](#)
 - ghdl, [226](#)
 - help, [225](#)
 - questa, [226](#)
 - vhdl2008, [226](#)
 - vhdl93, [226](#)
- compile-xilinx-ise.ps1 command line option
 - All, [228](#)
 - Clean, [228](#)
 - GHDL, [228](#)
 - Help, [228](#)
 - Questa, [228](#)
 - ReLink, [228](#)
 - VHDL2008, [228](#)
 - VHDL93, [228](#)
- compile-xilinx-ise.sh command line option
 - all, [227](#)
 - clean, [227](#)
 - ghdl, [227](#)
 - help, [227](#)
 - questa, [227](#)
 - vhdl2008, [227](#)
 - vhdl93, [227](#)
- compile-xilinx-vivado.ps1 command line option
 - All, [230](#)
 - Clean, [230](#)
 - GHDL, [230](#)
 - Help, [230](#)
 - Questa, [230](#)
 - ReLink, [230](#)
 - VHDL2008, [230](#)
 - VHDL93, [230](#)
- compile-xilinx-vivado.sh command line option
 - all, [229](#)
 - clean, [229](#)
 - ghdl, [229](#)
 - help, [229](#)
 - questa, [229](#)
 - vhdl2008, [229](#)
 - vhdl93, [229](#)
- CompileError (DataBase.TestCase.CompileStatus attribute), [312](#)
- CompileFailed (DataBase.TestCase.CompileStatus attribute), [312](#)
- Compiler (class in Compiler), [282](#)
- Compiler (class in Compiler.ISECompiler), [267](#)
- Compiler (class in Compiler.LSECompiler), [269](#)
- Compiler (class in Compiler.QuartusCompiler), [271](#)
- Compiler (class in Compiler.VivadoCompiler), [272](#)
- Compiler (class in Compiler.XCICompiler), [274](#)
- Compiler (class in Compiler.XCOCCompiler), [276](#)
- Compiler (class in Compiler.XSTCompiler), [278](#)
- Compiler (module), [267](#)
- Compiler.ISECompiler (module), [267](#)
- Compiler.LSECompiler (module), [269](#)
- Compiler.QuartusCompiler (module), [271](#)
- Compiler.VivadoCompiler (module), [272](#)
- Compiler.XCICompiler (module), [274](#)
- Compiler.XCOCCompiler (module), [276](#)
- Compiler.XSTCompiler (module), [278](#)
- CompileResult (class in Compiler), [281](#)
- CompilerException, [280](#)
- CompilerFilter() (in module ToolChain.Lattice.Diamond), [430](#)

- CompilerFilter() (in module ToolChain.Xilinx.Vivado), 484
- CompileState (class in Compiler), 281
- CompileStatus (class in DataBase.TestCase), 311
- CompileSuccess (DataBase.TestCase.CompileStatus attribute), 312
- ConcatenateExpression (class in Parser.FilesCodeDOM), 318
- ConditionalBlockStatement (class in lib.CodeDOM), 504
- ConfigParameters (ToolChain.Git.GitConfig attribute), 410
- ConfigSection (DataBase.Entity.AskWildCard attribute), 298
- ConfigSection (DataBase.Entity.CocoTestbench attribute), 301
- ConfigSection (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- ConfigSection (DataBase.Entity.IPCore attribute), 299
- ConfigSection (DataBase.Entity.LatticeNetlist attribute), 303
- ConfigSection (DataBase.Entity.LazyPathElement attribute), 299
- ConfigSection (DataBase.Entity.Library attribute), 295
- ConfigSection (DataBase.Entity.Namespace attribute), 295
- ConfigSection (DataBase.Entity.Netlist attribute), 301
- ConfigSection (DataBase.Entity.PathElement attribute), 294
- ConfigSection (DataBase.Entity.QuartusNetlist attribute), 303
- ConfigSection (DataBase.Entity.StarWildCard attribute), 297
- ConfigSection (DataBase.Entity.Testbench attribute), 300
- ConfigSection (DataBase.Entity.VHDLTestbench attribute), 300
- ConfigSection (DataBase.Entity.VivadoNetlist attribute), 304
- ConfigSection (DataBase.Entity.WildCard attribute), 297
- ConfigSection (DataBase.Entity.XstNetlist attribute), 302
- ConfigSectionName (DataBase.Entity.AskWildCard attribute), 298
- ConfigSectionName (DataBase.Entity.CocoTestbench attribute), 301
- ConfigSectionName (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- ConfigSectionName (DataBase.Entity.IPCore attribute), 299
- ConfigSectionName (DataBase.Entity.LatticeNetlist attribute), 303
- ConfigSectionName (DataBase.Entity.LazyPathElement attribute), 299
- ConfigSectionName (DataBase.Entity.Library attribute), 295
- ConfigSectionName (DataBase.Entity.Namespace attribute), 295
- ConfigSectionName (DataBase.Entity.Netlist attribute), 301
- ConfigSectionName (DataBase.Entity.PathElement attribute), 294
- ConfigSectionName (DataBase.Entity.QuartusNetlist attribute), 303
- ConfigSectionName (DataBase.Entity.StarWildCard attribute), 297
- ConfigSectionName (DataBase.Entity.Testbench attribute), 300
- ConfigSectionName (DataBase.Entity.VHDLTestbench attribute), 300
- ConfigSectionName (DataBase.Entity.VivadoNetlist attribute), 304
- ConfigSectionName (DataBase.Entity.WildCard attribute), 297
- ConfigSectionName (DataBase.Entity.XstNetlist attribute), 302
- Configuration (class in ToolChain), 489
- Configuration (class in ToolChain.Aldec), 361
- Configuration (class in ToolChain.Aldec.ActiveHDL), 348
- Configuration (class in ToolChain.Aldec.RivieraPRO), 355
- Configuration (class in ToolChain.Altera), 379
- Configuration (class in ToolChain.Altera.ModelSim), 364
- Configuration (class in ToolChain.Altera.Quartus), 372
- Configuration (class in ToolChain.GHDL), 382
- Configuration (class in ToolChain.Git), 403
- Configuration (class in ToolChain.GTKWave), 399
- Configuration (class in ToolChain.Intel), 423
- Configuration (class in ToolChain.Intel.ModelSim), 411
- Configuration (class in ToolChain.Intel.Quartus), 419
- Configuration (class in ToolChain.Lattice), 433
- Configuration (class in ToolChain.Lattice.Diamond), 426
- Configuration (class in ToolChain.Lattice.Synplify), 430
- Configuration (class in ToolChain.Mentor), 457

Configuration (class in ToolChain.Mentor.ModelSim), 436
Configuration (class in ToolChain.Mentor.QuestaSim), 455
Configuration (class in ToolChain.PoC), 459
Configuration (class in ToolChain.Synopsys), 462
Configuration (class in ToolChain.Xilinx), 485
Configuration (class in ToolChain.Xilinx.ISE), 466
Configuration (class in ToolChain.Xilinx.Vivado), 476
ConfigurationException, 487
ConfigurationState (class in ToolChain), 488
Configurator (class in ToolChain), 496
ConfigureAll() (ToolChain.Configurator method), 496
Configured (ToolChain.ConfigurationState attribute), 488
ConfigureDefaultTools() (ToolChain.Configurator method), 496
ConfigureForAll() (ToolChain.Aldec.ActiveHDL.Configuration method), 348
ConfigureForAll() (ToolChain.Aldec.Configuration method), 362
ConfigureForAll() (ToolChain.Aldec.RivieraPRO.Configuration method), 355
ConfigureForAll() (ToolChain.Altera.Configuration method), 379
ConfigureForAll() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367
ConfigureForAll() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 369
ConfigureForAll() (ToolChain.Altera.ModelSim.Configuration method), 364
ConfigureForAll() (ToolChain.Altera.Quartus.Configuration method), 373
ConfigureForAll() (ToolChain.Configuration method), 490
ConfigureForAll() (ToolChain.GHDL.Configuration method), 382
ConfigureForAll() (ToolChain.Git.Configuration method), 403
ConfigureForAll() (ToolChain.GTKWave.Configuration method), 399
ConfigureForAll() (ToolChain.Intel.Configuration method), 424
ConfigureForAll() (ToolChain.Intel.ModelSim.Configuration method), 412
ConfigureForAll() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 414
ConfigureForAll() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
ConfigureForAll() (ToolChain.Intel.Quartus.Configuration method), 420
ConfigureForAll() (ToolChain.Lattice.Configuration method), 433
ConfigureForAll() (ToolChain.Lattice.Diamond.Configuration method), 426
ConfigureForAll() (ToolChain.Lattice.Synplify.Configuration method), 430
ConfigureForAll() (ToolChain.Mentor.Configuration method), 458
ConfigureForAll() (ToolChain.Mentor.ModelSim.Configuration method), 437
ConfigureForAll() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 439
ConfigureForAll() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
ConfigureForAll() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444
ConfigureForAll() (ToolChain.Mentor.QuestaSim.Configuration method), 455
ConfigureForAll() (ToolChain.PoC.Configuration method), 460
ConfigureForAll() (ToolChain.Synopsys.Configuration method), 462
ConfigureForAll() (ToolChain.ToolConfiguration method), 493
ConfigureForAll() (ToolChain.VendorConfiguration method), 491
ConfigureForAll() (ToolChain.Xilinx.Configuration method), 485
ConfigureForAll() (ToolChain.Xilinx.ISE.Configuration method), 467
ConfigureForAll() (ToolChain.Xilinx.Vivado.Configuration method), 476
ConfigureForDarwin() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
ConfigureForDarwin() (ToolChain.Aldec.Configuration method), 362
ConfigureForDarwin() (ToolChain.Aldec.RivieraPRO.Configuration method), 355
ConfigureForDarwin() (ToolChain.Altera.Configuration method), 379
ConfigureForDarwin() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367
ConfigureForDarwin() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
ConfigureForDarwin() (ToolChain.Altera.ModelSim.Configuration method), 365
ConfigureForDarwin() (ToolChain.Altera.Quartus.Configuration method), 373
ConfigureForDarwin() (ToolChain.Configuration method), 489
ConfigureForDarwin() (ToolChain.GHDL.Configuration method), 382
ConfigureForDarwin() (ToolChain.Git.Configuration method), 403
ConfigureForDarwin() (ToolChain.GTKWave.Configuration method), 399
ConfigureForDarwin() (ToolChain.Intel.Configuration method), 424
ConfigureForDarwin() (ToolChain.Intel.ModelSim.Configuration method), 412
ConfigureForDarwin() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 414
ConfigureForDarwin() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
ConfigureForDarwin() (ToolChain.Intel.Quartus.Configuration method), 420
ConfigureForDarwin() (ToolChain.Lattice.Configuration method), 433
ConfigureForDarwin() (ToolChain.Lattice.Diamond.Configuration method), 426
ConfigureForDarwin() (ToolChain.Lattice.Synplify.Configuration method), 430
ConfigureForDarwin() (ToolChain.Mentor.Configuration method), 458
ConfigureForDarwin() (ToolChain.Mentor.ModelSim.Configuration method), 437
ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 439
ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444
ConfigureForDarwin() (ToolChain.Mentor.QuestaSim.Configuration method), 455
ConfigureForDarwin() (ToolChain.PoC.Configuration method), 460
ConfigureForDarwin() (ToolChain.Synopsys.Configuration method), 462
ConfigureForDarwin() (ToolChain.ToolConfiguration method), 493
ConfigureForDarwin() (ToolChain.VendorConfiguration method), 491
ConfigureForDarwin() (ToolChain.Xilinx.Configuration method), 485
ConfigureForDarwin() (ToolChain.Xilinx.ISE.Configuration method), 467
ConfigureForDarwin() (ToolChain.Xilinx.Vivado.Configuration method), 476

method), 420

ConfigureForDarwin() (ToolChain.Lattice.Configuration method), 433

ConfigureForDarwin() (ToolChain.Lattice.Diamond.Configuration method), 426

ConfigureForDarwin() (ToolChain.Lattice.Synplify.Configuration method), 430

ConfigureForDarwin() (ToolChain.Mentor.Configuration method), 458

ConfigureForDarwin() (ToolChain.Mentor.ModelSim.Configuration method), 437

ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 439

ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442

ConfigureForDarwin() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444

ConfigureForDarwin() (ToolChain.Mentor.QuestaSim.Configuration method), 455

ConfigureForDarwin() (ToolChain.PoC.Configuration method), 460

ConfigureForDarwin() (ToolChain.Synopsys.Configuration method), 463

ConfigureForDarwin() (ToolChain.ToolConfiguration method), 493

ConfigureForDarwin() (ToolChain.VendorConfiguration method), 491

ConfigureForDarwin() (ToolChain.Xilinx.Configuration method), 485

ConfigureForDarwin() (ToolChain.Xilinx.ISE.Configuration method), 467

ConfigureForDarwin() (ToolChain.Xilinx.Vivado.Configuration method), 476

ConfigureForLinux() (ToolChain.Aldec.ActiveHDL.Configuration method), 349

ConfigureForLinux() (ToolChain.Aldec.Configuration method), 362

ConfigureForLinux() (ToolChain.Aldec.RivieraPRO.Configuration method), 356

ConfigureForLinux() (ToolChain.Altera.Configuration method), 379

ConfigureForLinux() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367

ConfigureForLinux() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370

ConfigureForLinux() (ToolChain.Altera.ModelSim.Configuration method), 365

ConfigureForLinux() (ToolChain.Altera.Quartus.Configuration method), 373

ConfigureForLinux() (ToolChain.Configuration method), 490

ConfigureForLinux() (ToolChain.GHDL.Configuration method), 382

ConfigureForLinux() (ToolChain.Git.Configuration method), 403

ConfigureForLinux() (ToolChain.GTKWave.Configuration method), 399

ConfigureForLinux() (ToolChain.Intel.Configuration method), 424

ConfigureForLinux() (ToolChain.Intel.ModelSim.Configuration method), 412

ConfigureForLinux() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415

ConfigureForLinux() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417

ConfigureForLinux() (ToolChain.Intel.Quartus.Configuration method), 420

ConfigureForLinux() (ToolChain.Lattice.Configuration method), 433

ConfigureForLinux() (ToolChain.Lattice.Diamond.Configuration method), 426

ConfigureForLinux() (ToolChain.Lattice.Synplify.Configuration method), 431

ConfigureForLinux() (ToolChain.Mentor.Configuration method), 458

ConfigureForLinux() (ToolChain.Mentor.ModelSim.Configuration method), 437

ConfigureForLinux() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440

ConfigureForLinux() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442

ConfigureForLinux() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444

ConfigureForLinux() (ToolChain.Mentor.QuestaSim.Configuration method), 455

ConfigureForLinux() (ToolChain.PoC.Configuration method), 460

ConfigureForLinux() (ToolChain.Synopsys.Configuration method), 463

ConfigureForLinux() (ToolChain.ToolConfiguration method), 493

ConfigureForLinux() (ToolChain.VendorConfiguration method), 491

ConfigureForLinux() (ToolChain.Xilinx.Configuration method), 485

ConfigureForLinux() (ToolChain.Xilinx.ISE.Configuration method), 467

ConfigureForLinux() (ToolChain.Xilinx.Vivado.Configuration method), 476

ConfigureForWindows() (ToolChain.Aldec.ActiveHDL.Configuration method), 349

ConfigureForWindows() (ToolChain.Aldec.Configuration method), 362

ConfigureForWindows() (ToolChain.Aldec.RivieraPRO.Configuration method), 356

ConfigureForWindows() (ToolChain.Altera.Configuration method), 379

ConfigureForWindows() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367

ConfigureForWindows() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370

ConfigureForWindows() (ToolChain.Altera.ModelSim.Configuration method), 365

ConfigureForWindows() (ToolChain.Altera.Quartus.Configuration method), 373

ConfigureForWindows() (ToolChain.Configuration method), 490

ConfigureForWindows() (ToolChain.GHDL.Configuration method), 382

ConfigureForWindows() (ToolChain.Git.Configuration method), 403

ConfigureForWindows() (ToolChain.GTKWave.Configuration method), 399

ConfigureForWindows() (ToolChain.Intel.Configuration method), 424

ConfigureForWindows() (ToolChain.Intel.ModelSim.Configuration method), 412

ConfigureForWindows() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415

ConfigureForWindows() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417

ConfigureForWindows() (ToolChain.Intel.Quartus.Configuration method), 420

ConfigureForWindows() (ToolChain.Lattice.Configuration method), 433

ConfigureForWindows() (ToolChain.Lattice.Diamond.Configuration method), 426

ConfigureForWindows() (ToolChain.Lattice.Synplify.Configuration method), 431

ConfigureForWindows() (ToolChain.Mentor.Configuration method), 458

ConfigureForWindows() (ToolChain.Mentor.ModelSim.Configuration method), 437

ConfigureForWindows() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440

ConfigureForWindows() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442

ConfigureForWindows() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444

ConfigureForWindows() (ToolChain.Mentor.QuestaSim.Configuration method), 455

ConfigureForWindows() (ToolChain.PoC.Configuration method), 460

ConfigureForWindows() (ToolChain.Synopsys.Configuration method), 463

ConfigureForWindows() (ToolChain.ToolConfiguration method), 493

ConfigureForWindows() (ToolChain.VendorConfiguration method), 491

ConfigureForWindows() (ToolChain.Xilinx.Configuration method), 485

ConfigureForWindows() (ToolChain.Xilinx.ISE.Configuration method), 467

ConfigureForWindows() (ToolChain.Xilinx.Vivado.Configuration method), 476

- method), 370
- ConfigureForWindows()
(ToolChain.Altera.ModelSim.Configuration method), 365
- ConfigureForWindows()
(ToolChain.Altera.Quartus.Configuration method), 373
- ConfigureForWindows() (ToolChain.Configuration method), 490
- ConfigureForWindows()
(ToolChain.GHDL.Configuration method), 383
- ConfigureForWindows() (ToolChain.Git.Configuration method), 403
- ConfigureForWindows()
(ToolChain.GTKWave.Configuration method), 399
- ConfigureForWindows()
(ToolChain.Intel.Configuration method), 424
- ConfigureForWindows()
(ToolChain.Intel.ModelSim.Configuration method), 412
- ConfigureForWindows()
(ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- ConfigureForWindows()
(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
- ConfigureForWindows()
(ToolChain.Intel.Quartus.Configuration method), 420
- ConfigureForWindows()
(ToolChain.Lattice.Configuration method), 433
- ConfigureForWindows()
(ToolChain.Lattice.Diamond.Configuration method), 426
- ConfigureForWindows()
(ToolChain.Lattice.Synplify.Configuration method), 431
- ConfigureForWindows()
(ToolChain.Mentor.Configuration method), 458
- ConfigureForWindows()
(ToolChain.Mentor.ModelSim.Configuration method), 437
- ConfigureForWindows()
(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
- ConfigureForWindows()
(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
- ConfigureForWindows()
(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444
- ConfigureForWindows()
(ToolChain.Mentor.QuestaSim.Configuration method), 455
- ConfigureForWindows()
(ToolChain.PoC.Configuration method), 460
- ConfigureForWindows()
(ToolChain.Synopsys.Configuration method), 463
- ConfigureForWindows() (ToolChain.ToolConfiguration method), 493
- ConfigureForWindows()
(ToolChain.VendorConfiguration method), 491
- ConfigureForWindows()
(ToolChain.Xilinx.Configuration method), 485
- ConfigureForWindows()
(ToolChain.Xilinx.ISE.Configuration method), 467
- ConfigureForWindows()
(ToolChain.Xilinx.Vivado.Configuration method), 476
- ConfigureTool() (ToolChain.Configurator method), 496
- ConstraintFile (class in Base.Project), 262
- ConstraintStatement (class in Parser.FilesCodeDOM), 320
- converters (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 509
- CopyBase.Executable.CommandLineArgumentList (class in Base.Executable.CommandLineArgumentList), 254
- CopyExternalLibraries()
(DataBase.Solution.FileListFile method), 309
- CopyFilesToFileSet() (DataBase.Solution.FileListFile method), 309
- CopyRuleMixIn (class in Parser.RulesParser), 330
- CopySourceFilesFromProject()
(ToolChain.Altera.Quartus.QuartusSettings method), 378
- CopyStatement (class in Parser.RulesCodeDOM), 328
- CopyTask (class in Compiler), 280
- CoreGenerator (class in ToolChain.Xilinx.ISE), 472
- CoreGenerator.Executable (class in ToolChain.Xilinx.ISE), 472
- CoreGenerator.FlagRegenerate (class in ToolChain.Xilinx.ISE), 472
- CoreGenerator.SwitchBatchFile (class in ToolChain.Xilinx.ISE), 473
- CoreGenerator.SwitchProjectFile (class in ToolChain.Xilinx.ISE), 473
- CoreGeneratorFilter() (in module ToolChain.Xilinx.ISE), 475
- CoreGeneratorNetlist (class in DataBase.Entity), 303
- CoreGenNetlists (DataBase.Entity.AskWildcard attribute), 298
- CoreGenNetlists (DataBase.Entity.StarWildcard attribute), 297
- CoreGenNetlists (DataBase.Entity.WildCard attribute), 296

- Count (DataBase.TestCase.GroupBase attribute), 312
- Count (DataBase.TestCase.SynthesisGroup attribute), 313
- Count (DataBase.TestCase.SynthesisSuite attribute), 314
- Count (DataBase.TestCase.TestGroup attribute), 312
- Count (DataBase.TestCase.TestSuite attribute), 313
- count() (Base.Executable.CommandLineArgumentList method), 254
- count() (ToolChain.EditionDescription method), 495
- Coverage (Simulator.SimulationState attribute), 345
- CPG (DataBase.Config.Packages attribute), 289
- Create() (ToolChain.Altera.Quartus.QuartusProject method), 377
- CreateFiles() (DataBase.Solution.Solution method), 306
- CreateFileSet() (Base.Project.Project method), 261
- CreateFileSet() (DataBase.Solution.VirtualProject method), 308
- CreateFileSet() (ToolChain.Altera.Quartus.QuartusProject method), 377
- CreateFileSet() (ToolChain.Xilinx.ISE.ISEProject method), 474
- CreateFileSet() (ToolChain.Xilinx.Vivado.VivadoProject method), 483
- CreateLibrary() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
- CreateLibrary() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 328
- CreateLibrary() (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 331
- CSG (DataBase.Config.Packages attribute), 289
- CXT (DataBase.Config.XilinxSubTypes attribute), 289
- Cyclone (DataBase.Config.AlteraFamilies attribute), 285
- Cyclone3 (DataBase.Config.AlteraDevices attribute), 286
- Cyclone4 (DataBase.Config.AlteraDevices attribute), 286
- Cyclone5 (DataBase.Config.AlteraDevices attribute), 286
- D**
- DA (DataBase.Config.XilinxSubTypes attribute), 288
- data (Base.Project.FileTypes attribute), 258
- data (DataBase.Entity.BaseFlags attribute), 292
- data (DataBase.Entity.NetlistKind attribute), 293
- data (DataBase.Entity.TestbenchKind attribute), 292
- data (Simulator.SimulationSteps attribute), 344
- DataBase (module), 283
- DataBase.Config (module), 284
- DataBase.Entity (module), 291
- DataBase.Solution (module), 305
- DataBase.TestCase (module), 311
- Debug (Base.Logging.Severity attribute), 256
- deco() (ToolChain.GHDL.GHDL method), 384
- deco() (ToolChain.GHDL.GHDLAnalyze method), 390
- deco() (ToolChain.GHDL.GHDLElaborate method), 393
- deco() (ToolChain.GHDL.GHDLRun method), 397
- DefaultAttribute (class in lib.pyAttribute.ArgParseAttributes), 517
- DefaultFileSet (Base.Project.Project attribute), 261
- DefaultFileSet (DataBase.Solution.VirtualProject attribute), 308
- DefaultFileSet (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- DefaultFileSet (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- DefaultFileSet (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- defaults() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
- DeleteRuleMixIn (class in Parser.RulesParser), 331
- DeleteStatement (class in Parser.RulesCodeDOM), 328
- DeleteTask (class in Compiler), 280
- DelimiterChars (lib.Parser.Tokenizer.TokenKind attribute), 515
- DelimiterToken (class in lib.Parser), 515
- Dependencies (DataBase.Entity.IPCore attribute), 299
- DescribeParameters (ToolChain.Git.GitDescribe attribute), 409
- DestinationPath (Compiler.CopyTask attribute), 280
- DestinationPath (Parser.RulesCodeDOM.CopyStatement attribute), 328
- DestinationPath (Parser.RulesParser.CopyRuleMixIn attribute), 331
- Device (Base.Project.Project attribute), 260
- Device (class in DataBase.Config), 290
- Device (DataBase.Config.Board attribute), 290
- Device (DataBase.Config.Device attribute), 290
- Device (DataBase.Solution.VirtualProject attribute), 308
- Device (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- Device (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
- Device (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- Device (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- Devices (class in DataBase.Config), 286
- Diamond (class in ToolChain.Lattice.Diamond), 428
- DiamondException, 425
- Directories (Base.Shared.Shared attribute), 265
- Directories (Compiler.Compiler attribute), 283
- Directories (Compiler.ISECompiler.Compiler attribute), 267
- Directories (Compiler.LSECompiler.Compiler attribute), 269
- Directories (Compiler.QuartusCompiler.Compiler attribute), 271
- Directories (Compiler.VivadoCompiler.Compiler attribute), 273
- Directories (Compiler.XCICompiler.Compiler attribute), 273

- tribute), 274
 - Directories (Compiler.XCOCCompiler.Compiler attribute), 276
 - Directories (Compiler.XSTCompiler.Compiler attribute), 278
 - Directories (Simulator.ActiveHDL Simulator.Simulator attribute), 332
 - Directories (Simulator.Cocotb Simulator.Simulator attribute), 334
 - Directories (Simulator.GHDL Simulator.Simulator attribute), 335
 - Directories (Simulator.ISE Simulator.Simulator attribute), 337
 - Directories (Simulator.ModelSim Simulator.Simulator attribute), 338
 - Directories (Simulator.Questa Simulator.Simulator attribute), 340
 - Directories (Simulator.Simulator attribute), 346
 - Directories (Simulator.Vivado Simulator.Simulator attribute), 341
 - Document (class in Parser.FilesCodeDOM), 323
 - Document (class in Parser.RulesCodeDOM), 330
 - DocumentMemberAttribute (class in lib.SphinxExtensions), 516
 - DocumentStatements (class in Parser.RulesCodeDOM), 327
 - DotAll (Parser.RulesCodeDOM.ReplaceStatement attribute), 328
 - DryRun (Base.Logging.Severity attribute), 256
 - DryRun (Base.Shared.Shared attribute), 265
 - DryRun (Compiler.Compiler attribute), 283
 - DryRun (Compiler.ISE Compiler.Compiler attribute), 267
 - DryRun (Compiler.LSE Compiler.Compiler attribute), 269
 - DryRun (Compiler.Quartus Compiler.Compiler attribute), 271
 - DryRun (Compiler.Vivado Compiler.Compiler attribute), 273
 - DryRun (Compiler.XCI Compiler.Compiler attribute), 274
 - DryRun (Compiler.XCOC Compiler.Compiler attribute), 276
 - DryRun (Compiler.XST Compiler.Compiler attribute), 278
 - DryRun (DataBase.TestCase.CompileStatus attribute), 311
 - DryRun (DataBase.TestCase.SimulationStatus attribute), 311
 - DryRun (Simulator.ActiveHDL Simulator.Simulator attribute), 332
 - DryRun (Simulator.Cocotb Simulator.Simulator attribute), 334
 - DryRun (Simulator.GHDL Simulator.Simulator attribute), 335
 - DryRun (Simulator.ISE Simulator.Simulator attribute), 337
 - DryRun (Simulator.ModelSim Simulator.Simulator attribute), 338
 - DryRun (Simulator.Questa Simulator.Simulator attribute), 340
 - DryRun (Simulator.SimulationResult attribute), 345
 - DryRun (Simulator.Simulator attribute), 346
 - DryRun (Simulator.Vivado Simulator.Simulator attribute), 341
 - DryRunCount (DataBase.TestCase.SynthesisGroup attribute), 313
 - DryRunCount (DataBase.TestCase.SynthesisSuite attribute), 314
 - DryRunCount (DataBase.TestCase.TestGroup attribute), 312
 - DryRunCount (DataBase.TestCase.TestSuite attribute), 313
- ## E
- E (DataBase.Config.AlteraSubTypes attribute), 288
 - E (DataBase.Config.Packages attribute), 289
 - E (DataBase.Config.XilinxSubTypes attribute), 288
 - ECP (DataBase.Config.LatticeFamilies attribute), 285
 - ECP2 (DataBase.Config.LatticeDevices attribute), 287
 - ECP3 (DataBase.Config.LatticeDevices attribute), 287
 - ECP5 (DataBase.Config.LatticeDevices attribute), 287
 - Edition (class in ToolChain), 495
 - EditionDescription (class in ToolChain), 495
 - Elaborate (Simulator.SimulationState attribute), 345
 - Elaborate() (ToolChain.GHDL.GHDL Elaborate method), 390
 - ElaborationError (DataBase.TestCase.SimulationStatus attribute), 311
 - ElaborationFilter() (in module ToolChain.Xilinx.Vivado), 484
 - ElementBase (class in DataBase.TestCase), 312
 - ElseClause (Parser.FilesCodeDOM.IfElseIfElseStatement attribute), 323
 - ElseIfClauses (Parser.FilesCodeDOM.IfElseIfElseStatement attribute), 323
 - ElseIfStatement (class in Parser.FilesCodeDOM), 322
 - ElseStatement (class in Parser.FilesCodeDOM), 323
 - EmptyChooseParserResult, 513
 - EmptyLine (class in lib.CodeDOM), 505
 - EndTime (DataBase.TestCase.SuiteMixIn attribute), 313
 - EndTime (DataBase.TestCase.SynthesisSuite attribute), 314
 - EndTime (DataBase.TestCase.TestSuite attribute), 313
 - Entities (DataBase.Entity.Library attribute), 295
 - Entities (DataBase.Entity.Namespace attribute), 295
 - Entity (DataBase.Entity.FQN attribute), 305
 - EntityNames (DataBase.Entity.Library attribute), 295
 - EntityNames (DataBase.Entity.Namespace attribute), 295
 - EntityTypes (class in DataBase.Entity), 291
 - Environment (Base.Project.Project attribute), 260
 - ENVIRONMENT (Base.Shared.Shared attribute), 265
 - Environment (class in Base.Executable), 255
 - Environment (class in Base.Project), 259

- ENVIRONMENT (Compiler.Compiler attribute), 282
- ENVIRONMENT (Compiler.ISECompiler.Compiler attribute), 267
- ENVIRONMENT (Compiler.LSECompiler.Compiler attribute), 269
- ENVIRONMENT (Compiler.QuartusCompiler.Compiler attribute), 271
- ENVIRONMENT (Compiler.VivadoCompiler.Compiler attribute), 273
- ENVIRONMENT (Compiler.XCICompiler.Compiler attribute), 275
- ENVIRONMENT (Compiler.XCOCCompiler.Compiler attribute), 276
- ENVIRONMENT (Compiler.XSTCompiler.Compiler attribute), 278
- Environment (DataBase.Solution.VirtualProject attribute), 308
- ENVIRONMENT (Simulator.ActiveHDL Simulator.Simulator attribute), 332
- ENVIRONMENT (Simulator.Cocotb Simulator.Simulator attribute), 334
- ENVIRONMENT (Simulator.GHDL Simulator.Simulator attribute), 335
- ENVIRONMENT (Simulator.ISE Simulator.Simulator attribute), 337
- ENVIRONMENT (Simulator.ModelSim Simulator.Simulator attribute), 338
- ENVIRONMENT (Simulator.Questa Simulator.Simulator attribute), 340
- ENVIRONMENT (Simulator.Simulator attribute), 345
- ENVIRONMENT (Simulator.Vivado Simulator.Simulator attribute), 341
- Environment (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- Environment (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- Environment (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- environment variable
 - LM_LICENSE_FILE, 219
 - PoCRootDirectory, 219, 220
- EnvironmentException, 242
- EqualExpression (class in lib.CodeDOM), 500
- Error (Base.Logging.Severity attribute), 256
- Error (Compiler.CompileResult attribute), 282
- Error (Simulator.SimulationResult attribute), 345
- ErrorCount (DataBase.TestCase.SynthesisGroup attribute), 313
- ErrorCount (DataBase.TestCase.SynthesisSuite attribute), 314
- ErrorCount (DataBase.TestCase.TestGroup attribute), 312
- ErrorCount (DataBase.TestCase.TestSuite attribute), 313
- ExceptionBase, 241
- Executable (class in Base.Executable), 255
- Executable (ToolChain.GHDL.GHDL attribute), 384
- Executable (ToolChain.GHDL.GHDLAnalyze attribute), 388
- Executable (ToolChain.GHDL.GHDLElaborate attribute), 391
- Executable (ToolChain.GHDL.GHDLRun attribute), 394
- ExecutableArgument (class in Base.Executable), 245
- ExecutableException, 244
- Execute() (ToolChain.Git.GitConfig method), 410
- Execute() (ToolChain.Git.GitDescribe method), 409
- Execute() (ToolChain.Git.GitRevList method), 407
- Execute() (ToolChain.Git.GitRevParse method), 406
- ExistsFunction (class in Parser.FilesCodeDOM), 319
- Exit (class in lib.Functions), 511
- exit() (lib.Functions.Exit class method), 511
- exit() (ToolChain.Altera.Quartus.QuartusSession method), 377
- Expression (class in lib.CodeDOM), 499
- Expression (lib.CodeDOM.ConditionalBlockStatement attribute), 504
- Expression (Parser.FilesCodeDOM.ElseIfStatement attribute), 323
- Expression (Parser.FilesCodeDOM.ExistsFunction attribute), 319
- Expression (Parser.FilesCodeDOM.IfStatement attribute), 322
- ExpressionChoice (class in lib.CodeDOM), 505
- extend() (Base.Executable.CommandLineArgumentList method), 254
- ExtendedConfigParser (class in lib.ExtendedConfigParser), 507
- ExtendedInterpolation (class in lib.ExtendedConfigParser), 507
- ExtendedSectionProxy (class in lib.ExtendedConfigParser), 506
- Extension() (Base.Project.FileTypes method), 258
- ExternalVHDL Libraries (Base.Project.Project attribute), 261
- ExternalVHDL Libraries (DataBase.Solution.VirtualProject attribute), 308
- ExternalVHDL Libraries (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- ExternalVHDL Libraries (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- ExternalVHDL Libraries (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- ExtractVHDL LibrariesFromVHDL SourceFiles()

(Base.Project.Project method), 261
 ExtractVHDLLibrariesFromVHDLSourceFiles()
 (DataBase.Solution.VirtualProject method),
 308
 ExtractVHDLLibrariesFromVHDLSourceFiles()
 (ToolChain.Altera.Quartus.QuartusProject
 method), 377
 ExtractVHDLLibrariesFromVHDLSourceFiles()
 (ToolChain.Xilinx.ISE.ISEProject method),
 474
 ExtractVHDLLibrariesFromVHDLSourceFiles()
 (ToolChain.Xilinx.Vivado.VivadoProject
 method), 483

F

F (DataBase.Config.Packages attribute), 289
 Failed (Compiler.CompileResult attribute), 282
 Failed (Simulator.SimulationResult attribute), 345
 FailedCount (DataBase.TestCase.SynthesisGroup at-
 tribute), 313
 FailedCount (DataBase.TestCase.SynthesisSuite
 attribute), 314
 FailedCount (DataBase.TestCase.TestGroup attribute),
 312
 FailedCount (DataBase.TestCase.TestSuite attribute),
 313
 Families (class in DataBase.Config), 285
 Family (DataBase.Config.Device attribute), 290
 FamilyName (DataBase.Config.Device attribute), 290
 Fatal (Base.Logging.Severity attribute), 256
 FBG (DataBase.Config.Packages attribute), 289
 FF (DataBase.Config.Packages attribute), 289
 FFG (DataBase.Config.Packages attribute), 289
 FG (DataBase.Config.Packages attribute), 289
 FGG (DataBase.Config.Packages attribute), 289
 File (Base.Project.CocotbSourceFile attribute), 265
 File (Base.Project.VerilogSourceFile attribute), 264
 File (Base.Project.VHDLSourceFile attribute), 263
 File (class in Base.Project), 261
 File (Parser.FilesParser.CocotbSourceFileMixIn at-
 tribute), 325
 File (Parser.FilesParser.FileReference attribute), 324
 File (Parser.FilesParser.IncludeFileMixIn attribute),
 324
 File (Parser.FilesParser.LDCSourceFileMixIn at-
 tribute), 325
 File (Parser.FilesParser.SDCSourceFileMixIn at-
 tribute), 325
 File (Parser.FilesParser.UCFSrcSourceFileMixIn at-
 tribute), 325
 File (Parser.FilesParser.VerilogSourceFileMixIn at-
 tribute), 325
 File (Parser.FilesParser.VHDLSourceFileMixIn at-
 tribute), 324
 File (Parser.FilesParser.XDCSourceFileMixIn at-
 tribute), 325
 File (ToolChain.Altera.Quartus.QuartusSettings at-
 tribute), 378

FileListFile (class in DataBase.Solution), 309
 FileName (Base.Project.CocotbSourceFile attribute),
 265
 FileName (Base.Project.ConstraintFile attribute), 262
 FileName (Base.Project.File attribute), 262
 FileName (Base.Project.ProjectFile attribute), 262
 FileName (Base.Project.PythonSourceFile attribute),
 264
 FileName (Base.Project.SettingsFile attribute), 263
 FileName (Base.Project.SourceFile attribute), 263
 FileName (Base.Project.VerilogSourceFile attribute),
 264
 FileName (Base.Project.VHDLSourceFile attribute),
 263
 FileName (DataBase.Solution.FileListFile attribute),
 309
 FileName (DataBase.Solution.RulesFile attribute), 310
 FileName (ToolChain.Altera.Quartus.QuartusProjectFile
 attribute), 378
 FileName (ToolChain.Altera.Quartus.QuartusSettings
 attribute), 378
 FileName (ToolChain.Lattice.Diamond.SynthesisArgumentFile
 attribute), 429
 FileName (ToolChain.Lattice.LatticeDesignConstraintFile
 attribute), 435
 FileName (ToolChain.Synopsys.SynopsysDesignConstraintFile
 attribute), 464
 FileName (ToolChain.Xilinx.ISE.ISEProjectFile
 attribute), 474
 FileName (ToolChain.Xilinx.ISE.UserConstraintFile
 attribute), 475
 FileName (ToolChain.Xilinx.Vivado.VivadoProjectFile
 attribute), 483
 FileName (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile
 attribute), 484
 FilePath (Compiler.AppendLineTask attribute), 281
 FilePath (Compiler.DeleteTask attribute), 280
 FilePath (Compiler.ReplaceTask attribute), 281
 FilePath (Parser.RulesCodeDOM.DeleteStatement at-
 tribute), 328
 FilePath (Parser.RulesCodeDOM.FileStatement at-
 tribute), 329
 FilePath (Parser.RulesParser.AppendLineRuleMixIn at-
 tribute), 331
 FilePath (Parser.RulesParser.DeleteRuleMixIn at-
 tribute), 331
 FilePath (Parser.RulesParser.ReplaceRuleMixIn at-
 tribute), 331
 FileReference (class in Parser.FilesParser), 324
 Files (Base.Project.FileSet attribute), 261
 Files (Base.Project.VHDLLibrary attribute), 261
 Files (DataBase.Solution.FileListFile attribute), 309
 Files (Parser.FilesParser.FilesParserMixIn attribute),
 326
 Files() (Base.Project.Project method), 261
 Files() (DataBase.Solution.VirtualProject method), 309
 Files() (ToolChain.Altera.Quartus.QuartusProject
 method), 377

- Files() (ToolChain.Xilinx.ISE.ISEProject method), 474
- Files() (ToolChain.Xilinx.Vivado.VivadoProject method), 483
- FileSet (Base.Project.CocotbSourceFile attribute), 265
- FileSet (Base.Project.ConstraintFile attribute), 262
- FileSet (Base.Project.File attribute), 262
- FileSet (Base.Project.ProjectFile attribute), 262
- FileSet (Base.Project.PythonSourceFile attribute), 264
- FileSet (Base.Project.SettingsFile attribute), 263
- FileSet (Base.Project.SourceFile attribute), 263
- FileSet (Base.Project.VerilogSourceFile attribute), 264
- FileSet (Base.Project.VHDLSourceFile attribute), 263
- FileSet (class in Base.Project), 261
- FileSet (DataBase.Solution.FileListFile attribute), 309
- FileSet (DataBase.Solution.RulesFile attribute), 310
- FileSet (ToolChain.Altera.Quartus.QuartusProjectFile attribute), 378
- FileSet (ToolChain.Altera.Quartus.QuartusSettings attribute), 378
- FileSet (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
- FileSet (ToolChain.Lattice.LatticeDesignConstraintFile attribute), 435
- FileSet (ToolChain.Synopsys.SynopsysDesignConstraintFile attribute), 464
- FileSet (ToolChain.Xilinx.ISE.ISEProjectFile attribute), 474
- FileSet (ToolChain.Xilinx.ISE.UserConstraintFile attribute), 475
- FileSet (ToolChain.Xilinx.Vivado.VivadoProjectFile attribute), 483
- FileSet (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile attribute), 484
- FileSets (Base.Project.Project attribute), 261
- FileSets (DataBase.Solution.VirtualProject attribute), 308
- FileSets (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- FileSets (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- FileSets (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- FilesFile (DataBase.Entity.CocoTestbench attribute), 301
- FilesFile (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- FilesFile (DataBase.Entity.LatticeNetlist attribute), 303
- FilesFile (DataBase.Entity.QuartusNetlist attribute), 302
- FilesFile (DataBase.Entity.Testbench attribute), 300
- FilesFile (DataBase.Entity.VHDLTestbench attribute), 300
- FilesFile (DataBase.Entity.VivadoNetlist attribute), 304
- FilesFile (DataBase.Entity.XstNetlist attribute), 302
- FilesParserMixIn (class in Parser.FilesParser), 326
- FileStatement (class in Parser.RulesCodeDOM), 329
- FileType (Base.Project.CocotbSourceFile attribute), 265
- FileType (Base.Project.ConstraintFile attribute), 262
- FileType (Base.Project.File attribute), 262
- FileType (Base.Project.ProjectFile attribute), 262
- FileType (Base.Project.PythonSourceFile attribute), 264
- FileType (Base.Project.SettingsFile attribute), 263
- FileType (Base.Project.SourceFile attribute), 263
- FileType (Base.Project.VerilogSourceFile attribute), 264
- FileType (Base.Project.VHDLSourceFile attribute), 263
- FileType (DataBase.Solution.FileListFile attribute), 309
- FileType (DataBase.Solution.RulesFile attribute), 310
- FileType (ToolChain.Altera.Quartus.QuartusProjectFile attribute), 378
- FileType (ToolChain.Altera.Quartus.QuartusSettings attribute), 378
- FileType (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
- FileType (ToolChain.Lattice.LatticeDesignConstraintFile attribute), 435
- FileType (ToolChain.Synopsys.SynopsysDesignConstraintFile attribute), 464
- FileType (ToolChain.Xilinx.ISE.ISEProjectFile attribute), 474
- FileType (ToolChain.Xilinx.ISE.UserConstraintFile attribute), 475
- FileType (ToolChain.Xilinx.Vivado.VivadoProjectFile attribute), 483
- FileType (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile attribute), 484
- FileTypes (class in Base.Project), 258
- FilterFile (DataBase.Entity.XstNetlist attribute), 302
- FlagArgument (class in Base.Executable), 248
- FLG (DataBase.Config.Packages attribute), 289
- Foreground (lib.Functions.Init attribute), 511
- FQN (class in DataBase.Entity), 305
- from_simple_str() (Base.Project.FileTypes class method), 258
- from_simple_str() (DataBase.Entity.BaseFlags class method), 292
- from_simple_str() (DataBase.Entity.NetlistKind class method), 293
- from_simple_str() (DataBase.Entity.TestbenchKind class method), 293
- from_simple_str() (Simulator.SimulationSteps class method), 344
- from_str() (Base.Project.FileTypes class method), 258
- from_str() (DataBase.Entity.BaseFlags class method), 292
- from_str() (DataBase.Entity.NetlistKind class method), 293
- from_str() (DataBase.Entity.TestbenchKind class method), 293
- from_str() (Simulator.SimulationSteps class method), 344
- FT (DataBase.Config.Packages attribute), 289

FTG (DataBase.Config.Packages attribute), 289
 FullName (DataBase.Config.Device attribute), 290
 FullName2 (DataBase.Config.Device attribute), 290
 Function (class in lib.CodeDOM), 503
 Fuse (class in ToolChain.Xilinx.ISE), 469
 Fuse.ArgTopLevel (class in ToolChain.Xilinx.ISE), 469
 Fuse.Executable (class in ToolChain.Xilinx.ISE), 469
 Fuse.FlagIncremental (class in ToolChain.Xilinx.ISE), 469
 Fuse.FlagRangeCheck (class in ToolChain.Xilinx.ISE), 469
 Fuse.SwitchMultiThreading (class in ToolChain.Xilinx.ISE), 469
 Fuse.SwitchOutputFile (class in ToolChain.Xilinx.ISE), 469
 Fuse.SwitchProjectFile (class in ToolChain.Xilinx.ISE), 469
 Fuse.SwitchTimeResolution (class in ToolChain.Xilinx.ISE), 469
 FuseFilter() (in module ToolChain.Xilinx.ISE), 475
 FXT (DataBase.Config.XilinxSubTypes attribute), 289

G

Generate() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
 Generation (DataBase.Config.Device attribute), 290
 Generic (DataBase.Config.GenericDevices attribute), 286
 Generic (DataBase.Config.GenericFamilies attribute), 285
 Generic (DataBase.Config.GenericSubTypes attribute), 288
 Generic (DataBase.Config.Packages attribute), 289
 Generic (DataBase.Config.Vendors attribute), 284
 GenericDevices (class in DataBase.Config), 286
 GenericFamilies (class in DataBase.Config), 285
 GenericSubTypes (class in DataBase.Config), 287
 get() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
 get() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
 GetAllEntities() (DataBase.Entity.Library method), 296
 GetAllEntities() (DataBase.Entity.Namespace method), 295
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.ArgParseMixin static method), 519
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute class method), 517
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.CommandAttribute class method), 517
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.CommandGroupAttribute class method), 516
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute class method), 518
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute class method), 518
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.DefaultAttribute class method), 517
 GetAttributes() (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute class method), 518
 GetAttributes() (lib.pyAttribute.Attribute class method), 519
 GetAttributes() (lib.pyAttribute.AttributeHelperMixin static method), 519
 GetAttributes() (lib.SphinxExtensions.DocumentMemberAttribute class method), 516
 getboolean() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
 GetCached() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
 GetCharacterTokenizer() (lib.Parser.Tokenizer static method), 515
 GetChoiceParser() (lib.CodeDOM.CodeDOMMeta static method), 498
 GetCocoTestbenches() (DataBase.Entity.AskWildCard method), 298
 GetCocoTestbenches() (DataBase.Entity.StarWildCard method), 297
 GetCocoTestbenches() (DataBase.Entity.WildCard method), 296
 GetCoreGenerator() (ToolChain.Xilinx.ISE.ISE method), 469
 GetCoreGenNetlists() (DataBase.Entity.AskWildCard method), 298
 GetCoreGenNetlists() (DataBase.Entity.StarWildCard method), 297
 GetCoreGenNetlists() (DataBase.Entity.WildCard method), 296
 GetElaborator() (ToolChain.Xilinx.Vivado.Vivado method), 478
 GetEntities() (DataBase.Entity.AskWildCard method), 298
 GetEntities() (DataBase.Entity.Library method), 296
 GetEntities() (DataBase.Entity.Namespace method), 295
 GetEntities() (DataBase.Entity.StarWildCard method), 297
 GetEntities() (DataBase.Entity.WildCard method), 296
 GetEntityNames() (DataBase.Entity.Library method), 296
 GetEntityNames() (DataBase.Entity.Namespace method), 295
 GetEnvironment() (ToolChain.Windows.Cmd method), 465
 getfloat() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
 GetFuse() (ToolChain.Xilinx.ISE.ISE method), 469
 GetGHDLAnalyze() (ToolChain.GHDL.GHDL method), 386
 GetGHDLAnalyze() (ToolChain.GHDL.GHDLAnalyze method), 388
 GetGHDLAnalyze() (ToolChain.GHDL.GHDLAnalyze method), 392
 GetGHDLAnalyze() (ToolChain.GHDL.GHDLRun

method), 395		method), 519	
GetGHDLElaborate() (ToolChain.GHDL.GHDL method), 386	(ToolChain.GHDL.GHDL	GetMethods() (lib.SphinxExtensions.DocumentMemberAttribute class method), 516	
GetGHDLElaborate() (ToolChain.GHDL.GHDLAnalyze method), 389		GetNamespaceNames() (DataBase.Entity.Library method), 296	
GetGHDLElaborate() (ToolChain.GHDL.GHDLElaborate method), 392		GetNamespaceNames() (DataBase.Entity.Namespace method), 295	
GetGHDLElaborate() (ToolChain.GHDL.GHDLRun method), 395	(ToolChain.GHDL.GHDLRun	GetNamespaces() (DataBase.Entity.Library method), 296	
GetGHDLRun() (ToolChain.GHDL.GHDL method), 386	method), 386	GetNamespaces() (DataBase.Entity.Namespace method), 295	
GetGHDLRun() (ToolChain.GHDL.GHDLAnalyze method), 389		GetNetlists() (DataBase.Entity.AskWildCard method), 298	
GetGHDLRun() (ToolChain.GHDL.GHDLElaborate method), 392		GetNetlists() (DataBase.Entity.IPCore method), 299	
GetGHDLRun() (ToolChain.GHDL.GHDLRun method), 395	(ToolChain.GHDL.GHDLRun	GetNetlists() (DataBase.Entity.StarWildCard method), 297	
GetGitConfig() (ToolChain.Git.Git method), 405	method), 405	GetNetlists() (DataBase.Entity.WildCard method), 296	
GetGitDescribe() (ToolChain.Git.Git method), 405		GetParser() (lib.CodeDOM.AndExpression class method), 502	
GetGitRevList() (ToolChain.Git.Git method), 405		GetParser() (lib.CodeDOM.BinaryExpression class method), 499	
GetGitRevParse() (ToolChain.Git.Git method), 405		GetParser() (lib.CodeDOM.BlockedStatement class method), 505	
getInt() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509		GetParser() (lib.CodeDOM.CommentLine class method), 505	
GetLatticeNetlists() (DataBase.Entity.AskWildCard method), 298	(DataBase.Entity.AskWildCard	GetParser() (lib.CodeDOM.CompareExpression class method), 500	
GetLatticeNetlists() (DataBase.Entity.StarWildCard method), 297	method), 297	GetParser() (lib.CodeDOM.EmptyLine class method), 505	
GetLatticeNetlists() (DataBase.Entity.WildCard method), 296	(DataBase.Entity.WildCard	GetParser() (lib.CodeDOM.EqualExpression class method), 500	
GetLibraries() (DataBase.Entity.NamespaceRoot method), 294	method), 294	GetParser() (lib.CodeDOM.ExpressionChoice class method), 505	
GetLibraryNames() (DataBase.Entity.NamespaceRoot method), 294		GetParser() (lib.CodeDOM.GreaterThanEqualExpression class method), 501	
GetLogFileReader() (ToolChain.Lattice.Diamond.Synth static method), 428	(ToolChain.Lattice.Diamond.Synth	GetParser() (lib.CodeDOM.GreaterThanExpression class method), 501	
GetMap() (ToolChain.Altera.Quartus.Quartus method), 375	method), 375	GetParser() (lib.CodeDOM.Identifier class method), 504	
GetMap() (ToolChain.Intel.Quartus.Quartus method), 422	method), 422	GetParser() (lib.CodeDOM.InExpression class method), 502	
GetMethods() (lib.pyAttribute.ArgParseAttributes.ArgParse class method), 519	(lib.pyAttribute.ArgParseAttributes.ArgParse	GetParser() (lib.CodeDOM.IntegerLiteral class method), 504	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Argument class method), 517	class method), 517	GetParser() (lib.CodeDOM.LessThanEqualExpression class method), 501	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Command class method), 517	class method), 517	GetParser() (lib.CodeDOM.LessThanExpression class method), 501	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Command class method), 516	class method), 516	GetParser() (lib.CodeDOM.ListElement class method), 503	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Command class method), 518	class method), 518	GetParser() (lib.CodeDOM.LogicalExpression class method), 500	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Command class method), 518	class method), 518	GetParser() (lib.CodeDOM.NotExpression class method), 499	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Default class method), 517	class method), 517	GetParser() (lib.CodeDOM.NotInExpression class method), 503	
GetMethods() (lib.pyAttribute.ArgParseAttributes.Switch class method), 518	class method), 518	GetParser() (lib.CodeDOM.OrExpression class method), 502	
GetMethods() (lib.pyAttribute.Attribute class method), 519	class method), 519	GetParser() (lib.CodeDOM.StringLiteral class method),	
GetMethods() (lib.pyAttribute.AttributeHelperMixin	(lib.pyAttribute.AttributeHelperMixin		

503	
GetParser() (lib.CodeDOM.UnequalExpression class method), 500	GetParser() (Parser.RulesCodeDOM.CopyStatement class method), 328
GetParser() (lib.CodeDOM.XorExpression class method), 502	GetParser() (Parser.RulesCodeDOM.DeleteStatement class method), 328
GetParser() (Parser.FilesCodeDOM.BlockedStatement class method), 317	GetParser() (Parser.RulesCodeDOM.Document class method), 330
GetParser() (Parser.FilesCodeDOM.CocotbStatement class method), 320	GetParser() (Parser.RulesCodeDOM.DocumentStatements class method), 328
GetParser() (Parser.FilesCodeDOM.ConcatenateExpression class method), 319	GetParser() (Parser.RulesCodeDOM.FileStatement class method), 329
GetParser() (Parser.FilesCodeDOM.ConstraintStatement class method), 320	GetParser() (Parser.RulesCodeDOM.InFileStatements class method), 327
GetParser() (Parser.FilesCodeDOM.Document class method), 323	GetParser() (Parser.RulesCodeDOM.PostProcessRulesStatement class method), 330
GetParser() (Parser.FilesCodeDOM.ElseIfStatement class method), 323	GetParser() (Parser.RulesCodeDOM.PostProcessStatements class method), 327
GetParser() (Parser.FilesCodeDOM.ElseStatement class method), 323	GetParser() (Parser.RulesCodeDOM.PreProcessRulesStatement class method), 329
GetParser() (Parser.FilesCodeDOM.ExistsFunction class method), 319	GetParser() (Parser.RulesCodeDOM.PreProcessStatements class method), 327
GetParser() (Parser.FilesCodeDOM.IfElseIfElseStatement class method), 323	GetParser() (Parser.RulesCodeDOM.ProcessRulesBlockStatement class method), 329
GetParser() (Parser.FilesCodeDOM.IfStatement class method), 322	GetParser() (Parser.RulesCodeDOM.ReplaceStatement class method), 328
GetParser() (Parser.FilesCodeDOM.IfThenElseExpression class method), 317	GetQuartusNetlists() (DataBase.Entity.AskWildCard method), 298
GetParser() (Parser.FilesCodeDOM.IncludeStatement class method), 322	GetQuartusNetlists() (DataBase.Entity.StarWildCard method), 297
GetParser() (Parser.FilesCodeDOM.InterpolateLiteral class method), 321	GetQuartusNetlists() (DataBase.Entity.WildCard method), 296
GetParser() (Parser.FilesCodeDOM.LDCStatement class method), 320	GetReader() (Base.Executable.Executable method), 255
GetParser() (Parser.FilesCodeDOM.LibraryStatement class method), 322	GetReader() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 352
GetParser() (Parser.FilesCodeDOM.ListConstructorExpression class method), 318	GetReader() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
GetParser() (Parser.FilesCodeDOM.ListElementExpression class method), 318	GetReader() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 353
GetParser() (Parser.FilesCodeDOM.PathExpressions class method), 318	GetReader() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
GetParser() (Parser.FilesCodeDOM.PathStatement class method), 321	GetReader() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
GetParser() (Parser.FilesCodeDOM.ReportStatement class method), 322	GetReader() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 360
GetParser() (Parser.FilesCodeDOM.SDCStatement class method), 320	GetReader() (ToolChain.Altera.Quartus.Map method), 375
GetParser() (Parser.FilesCodeDOM.SubDirectoryExpression class method), 318	GetReader() (ToolChain.Altera.Quartus.TclShell method), 376
GetParser() (Parser.FilesCodeDOM.UCFStatement class method), 321	GetReader() (ToolChain.GHDL.GHDL method), 386
GetParser() (Parser.FilesCodeDOM.VerilogStatement class method), 319	GetReader() (ToolChain.GHDL.GHDLAnalyze method), 389
GetParser() (Parser.FilesCodeDOM.VHDLStatement class method), 319	GetReader() (ToolChain.GHDL.GHDL Elaborate method), 392
GetParser() (Parser.FilesCodeDOM.XDCStatement class method), 321	GetReader() (ToolChain.GHDL.GHDLRun method), 395
GetParser() (Parser.RulesCodeDOM.AppendLineStatement class method), 329	GetReader() (ToolChain.Git.GitConfig method), 410
	GetReader() (ToolChain.Git.GitDescribe method), 409

[GetReader\(\) \(ToolChain.Git.GitRevList method\), 408](#)
[GetReader\(\) \(ToolChain.Git.GitRevParse method\), 406](#)
[GetReader\(\) \(ToolChain.Git.GitSCM method\), 405](#)
[GetReader\(\) \(ToolChain.GNU.Make method\), 397](#)
[GetReader\(\) \(ToolChain.GTKWave.GTKWave method\), 401](#)
[GetReader\(\) \(ToolChain.Intel.Quartus.Map method\), 422](#)
[GetReader\(\) \(ToolChain.Lattice.Diamond.Synth method\), 428](#)
[GetReader\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler method\), 450](#)
[GetReader\(\) \(ToolChain.Mentor.ModelSim.VHDLLibraryTool method\), 447](#)
[GetReader\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 453](#)
[GetReader\(\) \(ToolChain.Windows.Cmd method\), 465](#)
[GetReader\(\) \(ToolChain.Xilinx.ISE.CoreGenerator method\), 473](#)
[GetReader\(\) \(ToolChain.Xilinx.ISE.Fuse method\), 470](#)
[GetReader\(\) \(ToolChain.Xilinx.ISE.ISESimulator method\), 471](#)
[GetReader\(\) \(ToolChain.Xilinx.ISE.Xst method\), 472](#)
[GetReader\(\) \(ToolChain.Xilinx.Vivado.Synth method\), 482](#)
[GetReader\(\) \(ToolChain.Xilinx.Vivado.XELab method\), 479](#)
[GetReader\(\) \(ToolChain.Xilinx.Vivado.XSim method\), 481](#)
[GetRepeatParser\(\) \(lib.CodeDOM.CodeDOMMeta static method\), 498](#)
[GetSections\(\) \(ToolChain.Aldec.ActiveHDL.Configuration class method\), 349](#)
[GetSections\(\) \(ToolChain.Aldec.Configuration class method\), 362](#)
[GetSections\(\) \(ToolChain.Aldec.RivieraPRO.Configuration class method\), 356](#)
[GetSections\(\) \(ToolChain.Altera.Configuration class method\), 380](#)
[GetSections\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration class method\), 367](#)
[GetSections\(\) \(ToolChain.Altera.ModelSim.Altera Starter Edition Configuration class method\), 370](#)
[GetSections\(\) \(ToolChain.Altera.ModelSim.Configuration class method\), 365](#)
[GetSections\(\) \(ToolChain.Altera.Quartus.Configuration class method\), 373](#)
[GetSections\(\) \(ToolChain.Configuration class method\), 489](#)
[GetSections\(\) \(ToolChain.GHDL.Configuration class method\), 383](#)
[GetSections\(\) \(ToolChain.Git.Configuration class method\), 403](#)
[GetSections\(\) \(ToolChain.GTKWave.Configuration class method\), 400](#)
[GetSections\(\) \(ToolChain.Intel.Configuration class method\), 424](#)
[GetSections\(\) \(ToolChain.Intel.ModelSim.Configuration class method\), 412](#)
[GetSections\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration class method\), 415](#)
[GetSections\(\) \(ToolChain.Intel.ModelSim.Intel Starter Edition Configuration class method\), 417](#)
[GetSections\(\) \(ToolChain.Intel.Quartus.Configuration class method\), 420](#)
[GetSections\(\) \(ToolChain.Lattice.Configuration class method\), 434](#)
[GetSections\(\) \(ToolChain.Lattice.Diamond.Configuration class method\), 426](#)
[GetSections\(\) \(ToolChain.Lattice.Synplify.Configuration class method\), 431](#)
[GetSections\(\) \(ToolChain.Mentor.Configuration class method\), 458](#)
[GetSections\(\) \(ToolChain.Mentor.ModelSim.Configuration class method\), 437](#)
[GetSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration class method\), 440](#)
[GetSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration class method\), 442](#)
[GetSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration class method\), 444](#)
[GetSections\(\) \(ToolChain.Mentor.QuestaSim.Configuration class method\), 456](#)
[GetSections\(\) \(ToolChain.PoC.Configuration class method\), 460](#)
[GetSections\(\) \(ToolChain.Synopsys.Configuration class method\), 463](#)
[GetSections\(\) \(ToolChain.ToolConfiguration class method\), 493](#)
[GetSections\(\) \(ToolChain.VendorConfiguration class method\), 491](#)
[GetSections\(\) \(ToolChain.Xilinx.Configuration class method\), 485](#)
[GetSections\(\) \(ToolChain.Xilinx.ISE.Configuration class method\), 467](#)
[GetSections\(\) \(ToolChain.Xilinx.Vivado.Configuration class method\), 476](#)
[GetSimulation\(\) \(ToolChain.Aldec.ActiveHDL.ActiveHDL Configuration class method\), 351](#)
[GetSimConfig\(\) \(ToolChain.Aldec.RivieraPRO.RivieraPRO Configuration class method\), 357](#)
[GetSimulator\(\) \(ToolChain.Mentor.ModelSim.ModelSim Configuration class method\), 447](#)
[GetSimulator\(\) \(ToolChain.Xilinx.Vivado.Vivado Configuration class method\), 478](#)
[GetSpecial\(\) \(lib.ExtendedConfigParser.ExtendedInterpolation static method\), 507](#)
[GetSynthesizer\(\) \(ToolChain.Lattice.Diamond.Diamond Configuration class method\), 428](#)
[GetSynthesizer\(\) \(ToolChain.Xilinx.Vivado.Vivado Configuration class method\), 478](#)
[GetTclCommand\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler Configuration class method\), 450](#)
[GetTclShell\(\) \(ToolChain.Altera.Quartus.Quartus Configuration class method\), 375](#)
[GetTclShell\(\) \(ToolChain.Intel.Quartus.Quartus Configuration class method\), 424](#)

- method), 422
- GetTestbenches() (DataBase.Entity.AskWildCard method), 298
- GetTestbenches() (DataBase.Entity.IPCore method), 299
- GetTestbenches() (DataBase.Entity.StarWildCard method), 297
- GetTestbenches() (DataBase.Entity.WildCard method), 296
- GetValue() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
- GetVariables() (Base.Project.Project method), 261
- GetVariables() (DataBase.Config.Board method), 290
- GetVariables() (DataBase.Config.Device method), 290
- GetVariables() (DataBase.Solution.VirtualProject method), 309
- GetVariables() (ToolChain.Altera.Quartus.QuartusProject method), 377
- GetVariables() (ToolChain.Xilinx.ISE.ISEProject method), 474
- GetVariables() (ToolChain.Xilinx.Vivado.VivadoProject method), 483
- GetVHDLCompiler() (ToolChain.Aldec.ActiveHDL.ActiveHDL method), 351
- GetVHDLCompiler() (ToolChain.Aldec.RivieraPRO.RivieraPRO method), 357
- GetVHDLCompiler() (ToolChain.Mentor.ModelSim.ModelSim method), 447
- GetVHDLCompiler() (ToolChain.Xilinx.ISE.ISE method), 469
- GetVHDLLibraryTool() (ToolChain.Aldec.ActiveHDL.ActiveHDL method), 351
- GetVHDLLibraryTool() (ToolChain.Aldec.RivieraPRO.RivieraPRO method), 357
- GetVHDLLibraryTool() (ToolChain.Mentor.ModelSim.ModelSim method), 447
- GetVHDLTestbenches() (DataBase.Entity.AskWildCard method), 298
- GetVHDLTestbenches() (DataBase.Entity.StarWildCard method), 297
- GetVHDLTestbenches() (DataBase.Entity.WildCard method), 296
- GetVivadoNetlists() (DataBase.Entity.AskWildCard method), 298
- GetVivadoNetlists() (DataBase.Entity.StarWildCard method), 297
- GetVivadoNetlists() (DataBase.Entity.WildCard method), 296
- GetWordTokenizer() (lib.Parser.Tokenizer class method), 515
- GetXst() (ToolChain.Xilinx.ISE.ISE method), 469
- GetXSTNetlists() (DataBase.Entity.AskWildCard method), 298
- GetXSTNetlists() (DataBase.Entity.StarWildCard method), 297
- GetXSTNetlists() (DataBase.Entity.WildCard method), 296
- GHDL (Base.Project.Tool attribute), 260
- GHDL (class in ToolChain.GHDL), 384
- GHDL.ArgListLibraryReferences (class in ToolChain.GHDL), 386
- GHDL.ArgSourceFile (class in ToolChain.GHDL), 386
- GHDL.ArgTopLevel (class in ToolChain.GHDL), 386
- GHDL.CmdAnalyze (class in ToolChain.GHDL), 384
- GHDL.CmdElaborate (class in ToolChain.GHDL), 384
- GHDL.CmdRun (class in ToolChain.GHDL), 384
- GHDL.FlagDebug (class in ToolChain.GHDL), 385
- GHDL.FlagExplicit (class in ToolChain.GHDL), 385
- GHDL.FlagMultiByteComments (class in ToolChain.GHDL), 385
- GHDL.FlagNoVitalChecks (class in ToolChain.GHDL), 385
- GHDL.FlagProfileArcs (class in ToolChain.GHDL), 385
- GHDL.FlagPSL (class in ToolChain.GHDL), 385
- GHDL.FlagRelaxedRules (class in ToolChain.GHDL), 385
- GHDL.FlagSynBinding (class in ToolChain.GHDL), 385
- GHDL.FlagTestCoverage (class in ToolChain.GHDL), 385
- GHDL.FlagVerbose (class in ToolChain.GHDL), 384
- GHDL.FlagWarnBinding (class in ToolChain.GHDL), 385
- GHDL.SwitchAssemblerOption (class in ToolChain.GHDL), 385
- GHDL.SwitchCompilerOption (class in ToolChain.GHDL), 385
- GHDL.SwitchFastWaveform (class in ToolChain.GHDL), 386
- GHDL.SwitchGHDLWaveform (class in ToolChain.GHDL), 386
- GHDL.SwitchIEEEAsserts (class in ToolChain.GHDL), 386
- GHDL.SwitchIEEEFlavor (class in ToolChain.GHDL), 385
- GHDL.SwitchLinkerOption (class in ToolChain.GHDL), 385
- GHDL.SwitchStopDelta (class in ToolChain.GHDL), 386
- GHDL.SwitchVCDGZWaveform (class in ToolChain.GHDL), 386
- GHDL.SwitchVCDWaveform (class in ToolChain.GHDL), 386
- GHDL.SwitchVHDLLibrary (class in ToolChain.GHDL), 386
- GHDL.SwitchVHDLVersion (class in ToolChain.GHDL), 386
- GHDL.SwitchWaveformOptionFile (class in ToolChain.GHDL), 386
- GHDL_GTKWave (Base.Project.ToolChain attribute),

259					
GHDLANalyze	(class in ToolChain.GHDL), 387				
GHDLANalyze.ArgListLibraryReferences	(class in ToolChain.GHDL), 387				
GHDLANalyze.ArgSourceFile	(class in ToolChain.GHDL), 387				
GHDLANalyze.ArgTopLevel	(class in ToolChain.GHDL), 387				
GHDLANalyze.CmdAnalyze	(class in ToolChain.GHDL), 387				
GHDLANalyze.CmdElaborate	(class in ToolChain.GHDL), 388				
GHDLANalyze.CmdRun	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagDebug	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagExplicit	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagMultiByteComments	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagNoVitalChecks	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagProfileArcs	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagPSL	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagRelaxedRules	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagSynBinding	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagTestCoverage	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagVerbose	(class in ToolChain.GHDL), 388				
GHDLANalyze.FlagWarnBinding	(class in ToolChain.GHDL), 388				
GHDLANalyze.SwitchAssemblerOption	(class in ToolChain.GHDL), 389				
GHDLANalyze.SwitchCompilerOption	(class in ToolChain.GHDL), 389				
GHDLANalyze.SwitchFastWaveform	(class in ToolChain.GHDL), 389				
GHDLANalyze.SwitchGHDLWaveform	(class in ToolChain.GHDL), 389				
GHDLANalyze.SwitchIEEEAsserts	(class in ToolChain.GHDL), 389				
GHDLANalyze.SwitchIEEEFlavor	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchLinkerOption	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchStopDelta	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchVCDGZWaveform	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchVCDWaveform	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchVHDLLibrary	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchVHDLVersion	(class in ToolChain.GHDL), 390				
GHDLANalyze.SwitchWaveformOptionFile	(class in ToolChain.GHDL), 390				
GHDLANalyzeFilter()	(in module ToolChain.GHDL), 397				
GHDLElaborate	(class in ToolChain.GHDL), 390				
GHDLElaborate.ArgListLibraryReferences	(class in ToolChain.GHDL), 390				
GHDLElaborate.ArgSourceFile	(class in ToolChain.GHDL), 391				
GHDLElaborate.ArgTopLevel	(class in ToolChain.GHDL), 391				
GHDLElaborate.CmdAnalyze	(class in ToolChain.GHDL), 391				
GHDLElaborate.CmdElaborate	(class in ToolChain.GHDL), 391				
GHDLElaborate.CmdRun	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagDebug	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagExplicit	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagMultiByteComments	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagNoVitalChecks	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagProfileArcs	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagPSL	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagRelaxedRules	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagSynBinding	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagTestCoverage	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagVerbose	(class in ToolChain.GHDL), 391				
GHDLElaborate.FlagWarnBinding	(class in ToolChain.GHDL), 392				
GHDLElaborate.SwitchAssemblerOption	(class in ToolChain.GHDL), 392				
GHDLElaborate.SwitchCompilerOption	(class in ToolChain.GHDL), 392				
GHDLElaborate.SwitchFastWaveform	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchGHDLWaveform	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchIEEEAsserts	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchIEEEFlavor	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchLinkerOption	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchStopDelta	(class in ToolChain.GHDL), 393				
GHDLElaborate.SwitchVCDGZWaveform	(class in				

ToolChain.GHDL), 393					
GHDLElaborate.SwitchVCDWaveform	(class in			GHDLRun.SwitchLinkerOption	(class in
ToolChain.GHDL), 393				ToolChain.GHDL), 396	
GHDLElaborate.SwitchVHDLLibrary	(class in			GHDLRun.SwitchStopDelta	(class in
ToolChain.GHDL), 393				ToolChain.GHDL), 396	
GHDLElaborate.SwitchVHDLVersion	(class in			GHDLRun.SwitchVCDGZWaveform	(class in
ToolChain.GHDL), 393				ToolChain.GHDL), 396	
GHDLElaborate.SwitchWaveformOptionFile	(class in			GHDLRun.SwitchVCDWaveform	(class in
ToolChain.GHDL), 393				ToolChain.GHDL), 396	
GHDLElaborateFilter() (in module ToolChain.GHDL),				GHDLRun.SwitchVHDLLibrary	(class in
397				ToolChain.GHDL), 396	
GHDLException, 381				GHDLRun.SwitchVHDLVersion	(class in
GHDLReanalyzeException, 381				ToolChain.GHDL), 396	
GHDLRun (class in ToolChain.GHDL), 393				GHDLRun.SwitchWaveformOptionFile	(class in
GHDLRun.ArgListLibraryReferences	(class in			ToolChain.GHDL), 396	
ToolChain.GHDL), 394				GHDLRunFilter() (in module ToolChain.GHDL),	397
GHDLRun.ArgSourceFile (class in ToolChain.GHDL),				Git (class in ToolChain.Git), 405	
394				GitConfig (class in ToolChain.Git), 409	
GHDLRun.ArgTopLevel (class in ToolChain.GHDL),				GitConfig.Command (class in ToolChain.Git), 409	
394				GitConfig.Executable (class in ToolChain.Git), 410	
GHDLRun.CmdAnalyze (class in ToolChain.GHDL),				GitConfig.Switch_Version (class in ToolChain.Git),	410
394				GitConfig.SwitchRemoveSection	(class in
GHDLRun.CmdElaborate (class in ToolChain.GHDL),				ToolChain.Git), 410	
394				GitConfig.SwitchUnset (class in ToolChain.Git), 409	
GHDLRun.CmdRun (class in ToolChain.GHDL), 394				GitConfig.ValueFilterClean (class in ToolChain.Git),	410
GHDLRun.FlagDebug (class in ToolChain.GHDL),				410	
394				GitConfig.ValueFilterParameters	(class in
GHDLRun.FlagExplicit (class in ToolChain.GHDL),				ToolChain.Git), 410	
394				GitConfig.ValueFilterSmudge (class in ToolChain.Git),	410
GHDLRun.FlagMultiByteComments	(class in			410	
ToolChain.GHDL), 394				GitDescribe (class in ToolChain.Git), 408	
GHDLRun.FlagNoVitalChecks	(class in			GitDescribe.Command (class in ToolChain.Git), 408	
ToolChain.GHDL), 394				GitDescribe.Executable (class in ToolChain.Git), 409	
GHDLRun.FlagProfileArcs	(class in			GitDescribe.Switch_Version (class in ToolChain.Git),	409
ToolChain.GHDL), 394				409	
GHDLRun.FlagPSL (class in ToolChain.GHDL), 394				GitDescribe.SwitchAbbrev (class in ToolChain.Git),	408
GHDLRun.FlagRelaxedRules	(class in			408	
ToolChain.GHDL), 394				GitDescribe.SwitchTags (class in ToolChain.Git), 408	
GHDLRun.FlagSynBinding	(class in			GitException, 402	
ToolChain.GHDL), 395				GitRevList (class in ToolChain.Git), 407	
GHDLRun.FlagTestCoverage	(class in			GitRevList.Command (class in ToolChain.Git), 407	
ToolChain.GHDL), 395				GitRevList.Executable (class in ToolChain.Git), 408	
GHDLRun.FlagVerbose (class in ToolChain.GHDL),				GitRevList.Switch_Version (class in ToolChain.Git),	408
395				408	
GHDLRun.FlagWarnBinding	(class in			GitRevList.SwitchMaxCount (class in ToolChain.Git),	407
ToolChain.GHDL), 395				407	
GHDLRun.SwitchAssemblerOption	(class in			GitRevList.SwitchTags (class in ToolChain.Git), 407	
ToolChain.GHDL), 395				GitRevParse (class in ToolChain.Git), 406	
GHDLRun.SwitchCompilerOption	(class in			GitRevParse.Command (class in ToolChain.Git), 406	
ToolChain.GHDL), 396				GitRevParse.Executable (class in ToolChain.Git), 406	
GHDLRun.SwitchFastWaveform	(class in			GitRevParse.Switch_Version (class in ToolChain.Git),	407
ToolChain.GHDL), 396				407	
GHDLRun.SwitchGHDLWaveform	(class in			GitRevParse.SwitchGitDir (class in ToolChain.Git),	406
ToolChain.GHDL), 396				406	
GHDLRun.SwitchIEEEAsserts	(class in			GitRevParse.SwitchInsideWorkingTree	(class in
ToolChain.GHDL), 396				ToolChain.Git), 406	
GHDLRun.SwitchIEEEFlavor	(class in			GitRevParse.SwitchShowTopLevel	(class in
ToolChain.GHDL), 396				ToolChain.Git), 406	

GitSCM (class in ToolChain.Git), 405
 GitSCM.Executable (class in ToolChain.Git), 405
 GitSCM.Switch_Version (class in ToolChain.Git), 405
 GlobalAssignments (ToolChain.Altera.Quartus.QuartusSettings attribute), 378
 GNUException, 397
 GNUMakeQuestaSimFilter() (in module ToolChain.GNU), 398
 GreaterThanEqualExpression (class in lib.CodeDOM), 501
 GreaterThanExpression (class in lib.CodeDOM), 501
 GreedyMatchingParserResult, 513
 GroupBase (class in DataBase.TestCase), 312
 groupName (lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute), 516
 Groups (DataBase.TestCase.GroupBase attribute), 312
 Groups (DataBase.TestCase.SynthesisGroup attribute), 313
 Groups (DataBase.TestCase.SynthesisSuite attribute), 314
 Groups (DataBase.TestCase.TestGroup attribute), 312
 Groups (DataBase.TestCase.TestSuite attribute), 313
 GS (DataBase.Config.AltteraSubTypes attribute), 288
 GT (DataBase.Config.AltteraSubTypes attribute), 288
 GTKwave (Base.Project.Tool attribute), 260
 GTKWave (class in ToolChain(GTKWave)), 401
 GTKWave.Executable (class in ToolChain(GTKWave)), 401
 GTKWave.SwitchDumpFile (class in ToolChain(GTKWave)), 401
 GTKWave.SwitchSaveFile (class in ToolChain(GTKWave)), 401
 GTKWaveException, 398
 GTKWaveFilter() (in module ToolChain(GTKWave)), 402
 GUIRun (Simulator.SimulationResult attribute), 345
 GX (DataBase.Config.AltteraSubTypes attribute), 288
 GZ (DataBase.Config.AltteraSubTypes attribute), 288

H

Handler (lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute), 517
 Handler (lib.pyAttribute.ArgParseAttributes.DefaultAttribute attribute), 517
 has_option() (lib.ExtendedConfigParser.ExtendedConfigParser method), 511
 has_section() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
 HasAttribute() (lib.pyAttribute.ArgParseAttributes.ArgParseMixin static method), 519
 HasAttribute() (lib.pyAttribute.AttributeHelperMixin static method), 519
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 352
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 351
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 353
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 352
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 351
 HasErrors (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 353
 HasErrors (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 359
 HasErrors (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358
 HasErrors (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 360
 HasErrors (ToolChain.Altera.Quartus.Map attribute), 375
 HasErrors (ToolChain.GHDL.GHDL attribute), 386
 HasErrors (ToolChain.GHDL.GHDLAnalyze attribute), 389
 HasErrors (ToolChain.GHDL.GHDElaborate attribute), 392
 HasErrors (ToolChain.GHDL.GHDLRun attribute), 395
 HasErrors (ToolChain.GNU.Make attribute), 397
 HasErrors (ToolChain.GTKWave.GTKWave attribute), 401
 HasErrors (ToolChain.Intel.Quartus.Map attribute), 422
 HasErrors (ToolChain.Lattice.Diamond.Synth attribute), 428
 HasErrors (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 450
 HasErrors (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 447
 HasErrors (ToolChain.Mentor.ModelSim.VHDL Simulator attribute), 453
 HasErrors (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473
 HasErrors (ToolChain.Xilinx.ISE.Fuse attribute), 470
 HasErrors (ToolChain.Xilinx.ISE.ISESimulator attribute), 471
 HasErrors (ToolChain.Xilinx.ISE.Xst attribute), 472
 HasErrors (ToolChain.Xilinx.Vivado.Synth attribute), 482
 HasErrors (ToolChain.Xilinx.Vivado.XElab attribute), 480
 HasErrors (ToolChain.Xilinx.Vivado.XSim attribute), 481
 HasWarnings (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 352
 HasWarnings (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 351
 HasWarnings (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 353
 HasWarnings (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 359
 HasWarnings (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358
 HasWarnings (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 360
 HasWarnings (ToolChain.Altera.Quartus.Map attribute), 375
 HasWarnings (ToolChain.GHDL.GHDL attribute), 387
 HasWarnings (ToolChain.GHDL.GHDLAnalyze attribute), 389
 HasWarnings (ToolChain.GHDL.GHDElaborate attribute), 392

- HasWarnings (ToolChain.GHDL.GHDLRun attribute), 395
- HasWarnings (ToolChain.GNU.Make attribute), 398
- HasWarnings (ToolChain.GTKWave.GTKWave attribute), 401
- HasWarnings (ToolChain.Intel.Quartus.Map attribute), 422
- HasWarnings (ToolChain.Lattice.Diamond.Synth attribute), 428
- HasWarnings (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 450
- HasWarnings (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 447
- HasWarnings (ToolChain.Mentor.ModelSim.VHDL Simulator attribute), 453
- HasWarnings (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473
- HasWarnings (ToolChain.Xilinx.ISE.Fuse attribute), 470
- HasWarnings (ToolChain.Xilinx.ISE.ISESimulator attribute), 471
- HasWarnings (ToolChain.Xilinx.ISE.Xst attribute), 472
- HasWarnings (ToolChain.Xilinx.Vivado.Synth attribute), 482
- HasWarnings (ToolChain.Xilinx.Vivado.XElab attribute), 480
- HasWarnings (ToolChain.Xilinx.Vivado.XSim attribute), 481
- HDLParams (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 430
- Host (Base.Shared.Shared attribute), 265
- Host (Compiler.Compiler attribute), 283
- Host (Compiler.ISECompiler.Compiler attribute), 267
- Host (Compiler.LSECompiler.Compiler attribute), 269
- Host (Compiler.QuartusCompiler.Compiler attribute), 271
- Host (Compiler.VivadoCompiler.Compiler attribute), 273
- Host (Compiler.XCICompiler.Compiler attribute), 275
- Host (Compiler.XCOCCompiler.Compiler attribute), 276
- Host (Compiler.XSTCompiler.Compiler attribute), 278
- Host (DataBase.Query attribute), 316
- Host (Simulator.ActiveHDL Simulator.Simulator attribute), 332
- Host (Simulator.Cocotb Simulator.Simulator attribute), 334
- Host (Simulator.GHDL Simulator.Simulator attribute), 335
- Host (Simulator.ISE Simulator.Simulator attribute), 337
- Host (Simulator.ModelSim Simulator.Simulator attribute), 338
- Host (Simulator.Questa Simulator.Simulator attribute), 340
- Host (Simulator.Simulator attribute), 346
- Host (Simulator.Vivado Simulator.Simulator attribute), 341
- Host (ToolChain.Aldec.ActiveHDL.Configuration attribute), 349
- Host (ToolChain.Aldec.Configuration attribute), 362
- Host (ToolChain.Aldec.RivieraPRO.Configuration attribute), 356
- Host (ToolChain.Altera.Configuration attribute), 380
- Host (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 367
- Host (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 370
- Host (ToolChain.Altera.ModelSim.Configuration attribute), 365
- Host (ToolChain.Altera.Quartus.Configuration attribute), 373
- Host (ToolChain.Configuration attribute), 489
- Host (ToolChain.GHDL.Configuration attribute), 383
- Host (ToolChain.Git.Configuration attribute), 403
- Host (ToolChain.GTKWave.Configuration attribute), 400
- Host (ToolChain.Intel.Configuration attribute), 424
- Host (ToolChain.Intel.ModelSim.Configuration attribute), 412
- Host (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 415
- Host (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 417
- Host (ToolChain.Intel.Quartus.Configuration attribute), 420
- Host (ToolChain.Lattice.Configuration attribute), 434
- Host (ToolChain.Lattice.Diamond.Configuration attribute), 427
- Host (ToolChain.Lattice.Synplify.Configuration attribute), 431
- Host (ToolChain.Mentor.Configuration attribute), 458
- Host (ToolChain.Mentor.ModelSim.Configuration attribute), 437
- Host (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 440
- Host (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 442
- Host (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute), 444
- Host (ToolChain.Mentor.QuestaSim.Configuration attribute), 456
- Host (ToolChain.PoC.Configuration attribute), 460
- Host (ToolChain.Synopsys.Configuration attribute), 463
- Host (ToolChain.ToolConfiguration attribute), 494
- Host (ToolChain.VendorConfiguration attribute), 491
- Host (ToolChain.Xilinx.Configuration attribute), 485
- Host (ToolChain.Xilinx.ISE.Configuration attribute), 467
- Host (ToolChain.Xilinx.Vivado.Configuration attribute), 477
- HT (DataBase.Config.XilinxSubTypes attribute), 289
- HXT (DataBase.Config.XilinxSubTypes attribute), 289
- |
- iCE40 (DataBase.Config.LatticeDevices attribute), 287
- ID (DataBase.Solution.Base attribute), 305

- ID (DataBase.Solution.ISEProject attribute), 307
- ID (DataBase.Solution.LatticeProject attribute), 308
- ID (DataBase.Solution.Project attribute), 307
- ID (DataBase.Solution.QuartusProject attribute), 308
- ID (DataBase.Solution.Repository attribute), 306
- ID (DataBase.Solution.Solution attribute), 306
- ID (DataBase.Solution.VivadoProject attribute), 307
- Identifier (class in lib.CodeDOM), 504
- IfClause (Parser.FilesCodeDOM.IfElseIfElseStatement attribute), 323
- IfElseIfElseStatement (class in Parser.FilesCodeDOM), 323
- IfStatement (class in Parser.FilesCodeDOM), 322
- IfThenElseExpressions (class in Parser.FilesCodeDOM), 317
- IHost (class in Base), 266
- ILazyLoadable (class in lib.Decorators), 506
- ILogable (class in Base.Logging), 257
- IncludeFileMixIn (class in Parser.FilesParser), 324
- Includes (DataBase.Solution.FileListFile attribute), 309
- Includes (Parser.FilesParser.FilesParserMixIn attribute), 326
- IncludeStatement (class in Parser.FilesCodeDOM), 322
- Indent (Base.Logging.LogEntry attribute), 256
- IndentBy() (Base.Logging.LogEntry method), 256
- index() (Base.Executable.CommandLineArgumentList method), 254
- index() (ToolChain.EditionDescription method), 495
- InExpression (class in lib.CodeDOM), 502
- InFileStatements (class in Parser.RulesCodeDOM), 327
- Info (Base.Logging.Severity attribute), 256
- Init (class in lib.Functions), 511
- init() (lib.Functions.Init class method), 511
- InitializationTime (DataBase.TestCase.SuiteMixIn attribute), 313
- InitializationTime (DataBase.TestCase.SynthesisSuite attribute), 314
- InitializationTime (DataBase.TestCase.TestSuite attribute), 313
- InitializeConfiguration() (ToolChain.Configurator method), 496
- insert() (Base.Executable.CommandLineArgumentList method), 254
- IntegerLiteral (class in lib.CodeDOM), 503
- Intel_ModelSim (Base.Project.ToolChain attribute), 259
- Intel_Quartus (Base.Project.ToolChain attribute), 259
- IntelEditionConfiguration (class in ToolChain.Intel.ModelSim), 414
- IntelException, 423
- IntelModelSimEditions (class in ToolChain.Intel.ModelSim), 411
- IntelQuartus (ToolChain.Altera.Quartus.QuartusEditions attribute), 372
- IntelStarterEditionConfiguration (class in ToolChain.Intel.ModelSim), 416
- InternalError (DataBase.TestCase.CompileStatus attribute), 312
- InternalError (DataBase.TestCase.SimulationStatus attribute), 311
- interpolate() (lib.ExtendedConfigParser.ExtendedInterpolation method), 507
- InterpolateLiteral (class in Parser.FilesCodeDOM), 321
- Interpolation (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 510
- IPCore (class in DataBase.Entity), 298
- is_disjoint() (Base.Project.FileTypes method), 258
- is_disjoint() (DataBase.Entity.BaseFlags method), 292
- is_disjoint() (DataBase.Entity.NetlistKind method), 293
- is_disjoint() (DataBase.Entity.TestbenchKind method), 293
- is_disjoint() (Simulator.SimulationSteps method), 344
- is_member (Base.Project.FileTypes attribute), 258
- is_member (DataBase.Entity.BaseFlags attribute), 292
- is_member (DataBase.Entity.NetlistKind attribute), 293
- is_member (DataBase.Entity.TestbenchKind attribute), 293
- is_member (Simulator.SimulationSteps attribute), 344
- IsAllPassed (DataBase.TestCase.TestSuite attribute), 313
- IsAllSuccess (DataBase.TestCase.SynthesisSuite attribute), 314
- IsConfigured() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- IsConfigured() (ToolChain.Aldec.Configuration method), 362
- IsConfigured() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- IsConfigured() (ToolChain.Altera.Configuration method), 380
- IsConfigured() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 367
- IsConfigured() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
- IsConfigured() (ToolChain.Altera.ModelSim.Configuration method), 365
- IsConfigured() (ToolChain.Altera.Quartus.Configuration method), 373
- IsConfigured() (ToolChain.Configuration method), 489
- IsConfigured() (ToolChain.GHDL.Configuration method), 383
- IsConfigured() (ToolChain.Git.Configuration method), 403
- IsConfigured() (ToolChain.GTKWave.Configuration method), 400
- IsConfigured() (ToolChain.Intel.Configuration method), 424
- IsConfigured() (ToolChain.Intel.ModelSim.Configuration method), 412
- IsConfigured() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- IsConfigured() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
- IsConfigured() (ToolChain.Intel.Quartus.Configuration method), 420

[IsConfigured\(\)](#) (ToolChain.Lattice.Configuration method), 434
[IsConfigured\(\)](#) (ToolChain.Lattice.Diamond.Configuration method), 427
[IsConfigured\(\)](#) (ToolChain.Lattice.Synplify.Configuration method), 431
[IsConfigured\(\)](#) (ToolChain.Mentor.Configuration method), 458
[IsConfigured\(\)](#) (ToolChain.Mentor.ModelSim.Configuration method), 437
[IsConfigured\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
[IsConfigured\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
[IsConfigured\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 444
[IsConfigured\(\)](#) (ToolChain.Mentor.QuestaSim.Configuration method), 456
[IsConfigured\(\)](#) (ToolChain.PoC.Configuration method), 460
[IsConfigured\(\)](#) (ToolChain.Synopsys.Configuration method), 463
[IsConfigured\(\)](#) (ToolChain.ToolConfiguration method), 493
[IsConfigured\(\)](#) (ToolChain.VendorConfiguration method), 491
[IsConfigured\(\)](#) (ToolChain.Xilinx.Configuration method), 485
[IsConfigured\(\)](#) (ToolChain.Xilinx.ISE.Configuration method), 467
[IsConfigured\(\)](#) (ToolChain.Xilinx.Vivado.Configuration method), 477
[ISE](#) (class in ToolChain.Xilinx.ISE), 469
[ISEException](#), 466
[ISEProject](#) (class in DataBase.Solution), 307
[ISEProject](#) (class in ToolChain.Xilinx.ISE), 473
[ISEProjectFile](#) (class in ToolChain.Xilinx.ISE), 474
[ISESimulator](#) (class in ToolChain.Xilinx.ISE), 470
[ISESimulator.Executable](#) (class in ToolChain.Xilinx.ISE), 470
[ISESimulator.FlagGuiMode](#) (class in ToolChain.Xilinx.ISE), 470
[ISESimulator.SwitchLogFile](#) (class in ToolChain.Xilinx.ISE), 470
[ISESimulator.SwitchTclBatchFile](#) (class in ToolChain.Xilinx.ISE), 470
[ISESimulator.SwitchWaveformFile](#) (class in ToolChain.Xilinx.ISE), 471
[IsSupportedPlatform\(\)](#) (ToolChain.Aldec.ActiveHDL.Configuration method), 349
[IsSupportedPlatform\(\)](#) (ToolChain.Aldec.Configuration method), 362
[IsSupportedPlatform\(\)](#) (ToolChain.Aldec.RivieraPRO.Configuration method), 356
[IsSupportedPlatform\(\)](#) (ToolChain.Altera.Configuration method), 380
[IsSupportedPlatform\(\)](#) (ToolChain.Altera.ModelSim.AlderiaStarterEditionConfiguration method), 370
[IsSupportedPlatform\(\)](#) (ToolChain.Altera.ModelSim.Configuration method), 365
[IsSupportedPlatform\(\)](#) (ToolChain.Altera.Quartus.Configuration method), 373
[IsSupportedPlatform\(\)](#) (ToolChain.Configuration method), 489
[IsSupportedPlatform\(\)](#) (ToolChain.GHDL.Configuration method), 383
[IsSupportedPlatform\(\)](#) (ToolChain.Git.Configuration method), 404
[IsSupportedPlatform\(\)](#) (ToolChain.GTKWave.Configuration method), 400
[IsSupportedPlatform\(\)](#) (ToolChain.Intel.Configuration method), 424
[IsSupportedPlatform\(\)](#) (ToolChain.Intel.ModelSim.Configuration method), 412
[IsSupportedPlatform\(\)](#) (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
[IsSupportedPlatform\(\)](#) (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 417
[IsSupportedPlatform\(\)](#) (ToolChain.Intel.Quartus.Configuration method), 420
[IsSupportedPlatform\(\)](#) (ToolChain.Lattice.Configuration method), 434
[IsSupportedPlatform\(\)](#) (ToolChain.Lattice.Diamond.Configuration method), 427
[IsSupportedPlatform\(\)](#) (ToolChain.Lattice.Synplify.Configuration method), 431
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.Configuration method), 458
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.ModelSim.Configuration method), 438
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
[IsSupportedPlatform\(\)](#) (ToolChain.Mentor.QuestaSim.Configuration method), 456
[IsSupportedPlatform\(\)](#) (ToolChain.PoC.Configuration method), 460
[IsSupportedPlatform\(\)](#) (ToolChain.Synopsys.Configuration method), 463
[IsSupportedPlatform\(\)](#) (ToolChain.ToolConfiguration method), 494
[IsSupportedPlatform\(\)](#) (ToolChain.VendorConfiguration method), 491
[IsSupportedPlatform\(\)](#) (ToolChain.Xilinx.Configuration method), 485
[IsSupportedPlatform\(\)](#) (ToolChain.Xilinx.ISE.Configuration method), 467
[IsSupportedPlatform\(\)](#) (ToolChain.Xilinx.Vivado.Configuration method), 477
[IsVisible\(\)](#) (DataBase.Entity.AskWildCard attribute), 298
[IsVisible](#) (DataBase.Entity.CocoTestbench attribute),

- 301
- IsVisible (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- IsVisible (DataBase.Entity.IPCore attribute), 299
- IsVisible (DataBase.Entity.LatticeNetlist attribute), 303
- IsVisible (DataBase.Entity.LazyPathElement attribute), 299
- IsVisible (DataBase.Entity.Library attribute), 296
- IsVisible (DataBase.Entity.Namespace attribute), 295
- IsVisible (DataBase.Entity.Netlist attribute), 301
- IsVisible (DataBase.Entity.PathElement attribute), 295
- IsVisible (DataBase.Entity.QuartusNetlist attribute), 303
- IsVisible (DataBase.Entity.StarWildCard attribute), 297
- IsVisible (DataBase.Entity.Testbench attribute), 300
- IsVisible (DataBase.Entity.VHDLTestbench attribute), 300
- IsVisible (DataBase.Entity.VivadoNetlist attribute), 304
- IsVisible (DataBase.Entity.WildCard attribute), 297
- IsVisible (DataBase.Entity.XstNetlist attribute), 302
- items() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
- items() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
- ## K
- keys() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
- keys() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
- Kind (DataBase.Entity.CocoTestbench attribute), 301
- Kind (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- Kind (DataBase.Entity.LatticeNetlist attribute), 303
- Kind (DataBase.Entity.LazyPathElement attribute), 299
- Kind (DataBase.Entity.Netlist attribute), 301
- Kind (DataBase.Entity.QuartusNetlist attribute), 303
- Kind (DataBase.Entity.Testbench attribute), 300
- Kind (DataBase.Entity.VHDLTestbench attribute), 300
- Kind (DataBase.Entity.VivadoNetlist attribute), 304
- Kind (DataBase.Entity.XstNetlist attribute), 302
- Kind (DataBase.Solution.Repository attribute), 306
- Kintex (DataBase.Config.XilinxFamilies attribute), 285
- Kintex7 (DataBase.Config.XilinxDevices attribute), 287
- KintexUltraScale (DataBase.Config.XilinxDevices attribute), 287
- KintexUltraScalePlus (DataBase.Config.XilinxDevices attribute), 287
- KWArgs (lib.pyAttribute.ArgParseAttributes.ArgumentAttribute attribute), 517
- KWArgs (lib.pyAttribute.ArgParseAttributes.CommandAttribute attribute), 517
- KWArgs (lib.pyAttribute.ArgParseAttributes.CommonArgumentAttribute attribute), 518
- KWArgs (lib.pyAttribute.ArgParseAttributes.CommonSwitchArgumentAttribute attribute), 518
- KWArgs (lib.pyAttribute.ArgParseAttributes.SwitchArgumentAttribute attribute), 518
- ## L
- Lattice
- Pre-compilation, 51
- Lattice (DataBase.Config.Vendors attribute), 284
- Lattice_Diamond (Base.Project.ToolChain attribute), 259
- Lattice_LSE (Base.Project.Tool attribute), 260
- Lattice_Synplify (Base.Project.ToolChain attribute), 259
- LatticeDesignConstraintFile (class in ToolChain.Lattice), 435
- LatticeDevices (class in DataBase.Config), 286
- LatticeEdition (ToolChain.Aldec.ActiveHDL.ActiveHDLEditions attribute), 348
- LatticeException, 432
- LatticeFamilies (class in DataBase.Config), 285
- LatticeNetlist (class in DataBase.Entity), 303
- LatticeNetlist (DataBase.Entity.IPCore attribute), 299
- LatticeNetlists (DataBase.Entity.AskWildCard attribute), 298
- LatticeNetlists (DataBase.Entity.StarWildCard attribute), 297
- LatticeNetlists (DataBase.Entity.WildCard attribute), 296
- LatticeProject (class in DataBase.Solution), 308
- LatticeSubTypes (class in DataBase.Config), 288
- LazyLoadable_IsLoaded (DataBase.Entity.CocoTestbench attribute), 301
- LazyLoadable_IsLoaded (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- LazyLoadable_IsLoaded (DataBase.Entity.LatticeNetlist attribute), 303
- LazyLoadable_IsLoaded (DataBase.Entity.LazyPathElement attribute), 299
- LazyLoadable_IsLoaded (DataBase.Entity.Netlist attribute), 301
- LazyLoadable_IsLoaded (DataBase.Entity.QuartusNetlist attribute), 303
- LazyLoadable_IsLoaded (DataBase.Entity.Testbench attribute), 300
- LazyLoadable_IsLoaded (DataBase.Entity.VHDLTestbench attribute), 300
- LazyLoadable_IsLoaded (DataBase.Entity.VivadoNetlist attribute), 304
- LazyLoadable_IsLoaded (DataBase.Entity.XstNetlist attribute), 302
- LazyLoadable_IsLoaded (DataBase.Solution.Base attribute), 305

- LazyLoadable_IsLoaded (DataBase.Solution.ISEProject attribute), 307
- LazyLoadable_IsLoaded (DataBase.Solution.LatticeProject attribute), 308
- LazyLoadable_IsLoaded (DataBase.Solution.Project attribute), 307
- LazyLoadable_IsLoaded (DataBase.Solution.QuartusProject attribute), 308
- LazyLoadable_IsLoaded (DataBase.Solution.Repository attribute), 306
- LazyLoadable_IsLoaded (DataBase.Solution.Solution attribute), 306
- LazyLoadable_IsLoaded (DataBase.Solution.VivadoProject attribute), 307
- LazyLoadable_IsLoaded (lib.Decorators.ILazyLoadable attribute), 506
- LazyLoadTrigger (class in lib.Decorators), 506
- LazyPathElement (class in DataBase.Entity), 299
- LDCSourceFileMixIn (class in Parser.FilesParser), 325
- LDCStatement (class in Parser.FilesCodeDOM), 320
- LeftChild (lib.CodeDOM.AndExpression attribute), 502
- LeftChild (lib.CodeDOM.BinaryExpression attribute), 499
- LeftChild (lib.CodeDOM.CompareExpression attribute), 500
- LeftChild (lib.CodeDOM.EqualExpression attribute), 500
- LeftChild (lib.CodeDOM.GreaterThanEqualExpression attribute), 501
- LeftChild (lib.CodeDOM.GreaterThanExpression attribute), 501
- LeftChild (lib.CodeDOM.InExpression attribute), 502
- LeftChild (lib.CodeDOM.LessThanEqualExpression attribute), 501
- LeftChild (lib.CodeDOM.LessThanExpression attribute), 501
- LeftChild (lib.CodeDOM.LogicalExpression attribute), 500
- LeftChild (lib.CodeDOM.NotInExpression attribute), 503
- LeftChild (lib.CodeDOM.OrExpression attribute), 502
- LeftChild (lib.CodeDOM.UnequalExpression attribute), 500
- LeftChild (lib.CodeDOM.XorExpression attribute), 502
- LeftChild (Parser.FilesCodeDOM.ConcatenateExpressionLibraryName (Parser.FilesParser.VHDLSourceFileMixIn attribute), 319
- LeftChild (Parser.FilesCodeDOM.SubDirectoryExpressionLibraryName (Parser.FilesCodeDOM.SubDirectoryExpressionLibraryName attribute), 318
- Length (lib.Parser.CharacterToken attribute), 514
- Length (lib.Parser.DelimiterToken attribute), 515
- Length (lib.Parser.NumberToken attribute), 515
- Length (lib.Parser.SpaceToken attribute), 515
- Length (lib.Parser.StartOfDocumentToken attribute), 514
- Length (lib.Parser.StringToken attribute), 515
- Length (lib.Parser.SuperToken attribute), 514
- Length (lib.Parser.Token attribute), 514
- Length (lib.Parser.ValuedToken attribute), 514
- LessThanEqualExpression (class in lib.CodeDOM), 501
- LessThanExpression (class in lib.CodeDOM), 500
- Level (DataBase.Entity.AskWildCard attribute), 298
- Level (DataBase.Entity.CocoTestbench attribute), 301
- Level (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- Level (DataBase.Entity.IPCore attribute), 299
- Level (DataBase.Entity.LatticeNetlist attribute), 303
- Level (DataBase.Entity.LazyPathElement attribute), 299
- Level (DataBase.Entity.Library attribute), 295
- Level (DataBase.Entity.Namespace attribute), 295
- Level (DataBase.Entity.Netlist attribute), 302
- Level (DataBase.Entity.PathElement attribute), 295
- Level (DataBase.Entity.QuartusNetlist attribute), 303
- Level (DataBase.Entity.StarWildCard attribute), 297
- Level (DataBase.Entity.Testbench attribute), 300
- Level (DataBase.Entity.VHDLTestbench attribute), 300
- Level (DataBase.Entity.VivadoNetlist attribute), 304
- Level (DataBase.Entity.WildCard attribute), 297
- Level (DataBase.Entity.XstNetlist attribute), 302
- lib (module), 497
- lib.CodeDOM (module), 497
- lib.Decorators (module), 505
- lib.ExtendedConfigParser (module), 506
- lib.Functions (module), 511
- lib.Parser (module), 512
- lib.pyAttribute (module), 516
- lib.pyAttribute.ArgParseAttributes (module), 516
- lib.SphinxExtensions (module), 516
- Libraries (DataBase.Entity.NamespaceRoot attribute), 294
- Libraries (DataBase.Solution.FileListFile attribute), 309
- Libraries (Parser.FilesParser.FilesParserMixIn attribute), 326
- Library (class in DataBase.Entity), 295
- Library (Parser.FilesCodeDOM.LibraryStatement attribute), 322
- LibraryName (Base.Project.VHDLSourceFile attribute), 264
- LibraryName (Parser.FilesCodeDOM.VHDLStatement attribute), 319
- LibraryName (Parser.FilesCodeDOM.VHDLStatementLibraryName (Parser.FilesCodeDOM.VHDLStatementLibraryName attribute), 324
- LibraryNames (DataBase.Entity.NamespaceRoot attribute), 294
- LibraryStatement (class in Parser.FilesCodeDOM), 322
- Link() (ToolChain.Xilinx.ISE.Fuse method), 469

- Link() (ToolChain.Xilinx.Vivado.XElab method), 479
- List (Parser.FilesCodeDOM.ListConstructorExpression attribute), 318
- ListConstructorExpression (class in Parser.FilesCodeDOM), 318
- ListElement (class in lib.CodeDOM), 503
- ListElementExpressions (class in Parser.FilesCodeDOM), 317
- Literal (class in lib.CodeDOM), 503
- LM_LICENSE_FILE, 219
- Log() (Base.Executable.Executable method), 255
- Log() (Base.IHost method), 267
- Log() (Base.Logging.ILogable method), 257
- Log() (Base.Shared.Shared method), 266
- Log() (Compiler.Compiler method), 283
- Log() (Compiler.ISECompiler.Compiler method), 267
- Log() (Compiler.LSECompiler.Compiler method), 269
- Log() (Compiler.QuartusCompiler.Compiler method), 271
- Log() (Compiler.VivadoCompiler.Compiler method), 273
- Log() (Compiler.XCICompiler.Compiler method), 275
- Log() (Compiler.XCOCCompiler.Compiler method), 276
- Log() (Compiler.XSTCompiler.Compiler method), 278
- Log() (Simulator.ActiveHDL Simulator.Simulator method), 332
- Log() (Simulator.Cocotb Simulator.Simulator method), 334
- Log() (Simulator.GHDL Simulator.Simulator method), 336
- Log() (Simulator.ISE Simulator.Simulator method), 337
- Log() (Simulator.ModelSim Simulator.Simulator method), 339
- Log() (Simulator.Questa Simulator.Simulator method), 340
- Log() (Simulator.Simulator method), 346
- Log() (Simulator.Vivado Simulator.Simulator method), 341
- Log() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- Log() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 352
- Log() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
- Log() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 353
- Log() (ToolChain.Aldec.Configuration method), 362
- Log() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- Log() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- Log() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- Log() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 360
- Log() (ToolChain.Altera.Configuration method), 380
- Log() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
- Log() (ToolChain.Altera.ModelSim.Altera Starter Edition Configuration method), 370
- Log() (ToolChain.Altera.ModelSim.Configuration method), 365
- Log() (ToolChain.Altera.Quartus.Configuration method), 373
- Log() (ToolChain.Altera.Quartus.Map method), 375
- Log() (ToolChain.Altera.Quartus.TclShell method), 376
- Log() (ToolChain.Configuration method), 490
- Log() (ToolChain.Configurator method), 497
- Log() (ToolChain.GHDL.Configuration method), 383
- Log() (ToolChain.GHDL.GHDL method), 387
- Log() (ToolChain.GHDL.GHDLAnalyze method), 389
- Log() (ToolChain.GHDL.GHDL Elaborate method), 392
- Log() (ToolChain.GHDL.GHDLRun method), 395
- Log() (ToolChain.Git.Configuration method), 404
- Log() (ToolChain.Git.GitConfig method), 410
- Log() (ToolChain.Git.GitDescribe method), 409
- Log() (ToolChain.Git.GitRevList method), 408
- Log() (ToolChain.Git.GitRevParse method), 406
- Log() (ToolChain.Git.GitSCM method), 405
- Log() (ToolChain.GNU.Make method), 398
- Log() (ToolChain.GTKWave.Configuration method), 400
- Log() (ToolChain.GTKWave.GTKWave method), 401
- Log() (ToolChain.Intel.Configuration method), 424
- Log() (ToolChain.Intel.ModelSim.Configuration method), 412
- Log() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- Log() (ToolChain.Intel.ModelSim.Intel Starter Edition Configuration method), 417
- Log() (ToolChain.Intel.Quartus.Configuration method), 420
- Log() (ToolChain.Intel.Quartus.Map method), 422
- Log() (ToolChain.Lattice.Configuration method), 434
- Log() (ToolChain.Lattice.Diamond.Configuration method), 427
- Log() (ToolChain.Lattice.Diamond.Synth method), 429
- Log() (ToolChain.Lattice.Synplify.Configuration method), 431
- Log() (ToolChain.Mentor.Configuration method), 458
- Log() (ToolChain.Mentor.ModelSim.Configuration method), 438
- Log() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
- Log() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
- Log() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
- Log() (ToolChain.Mentor.ModelSim.Selector method), 446
- Log() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450
- Log() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 447

- Log() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 453
- Log() (ToolChain.Mentor.QuestaSim.Configuration method), 456
- Log() (ToolChain.PoC.Configuration method), 460
- Log() (ToolChain.Synopsys.Configuration method), 463
- Log() (ToolChain.ToolConfiguration method), 494
- Log() (ToolChain.ToolSelector method), 495
- Log() (ToolChain.VendorConfiguration method), 491
- Log() (ToolChain.Windows.Cmd method), 465
- Log() (ToolChain.Xilinx.Configuration method), 485
- Log() (ToolChain.Xilinx.ISE.Configuration method), 467
- Log() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- Log() (ToolChain.Xilinx.ISE.Fuse method), 470
- Log() (ToolChain.Xilinx.ISE.ISE Simulator method), 471
- Log() (ToolChain.Xilinx.ISE.Xst method), 472
- Log() (ToolChain.Xilinx.Vivado.Configuration method), 477
- Log() (ToolChain.Xilinx.Vivado.Synth method), 482
- Log() (ToolChain.Xilinx.Vivado.XElab method), 480
- Log() (ToolChain.Xilinx.Vivado.XSim method), 481
- LogDebug() (Base.Executable.Executable method), 255
- LogDebug() (Base.IHost method), 267
- LogDebug() (Base.Logging.ILogable method), 257
- LogDebug() (Base.Shared.Shared method), 266
- LogDebug() (Compiler.Compiler method), 283
- LogDebug() (Compiler.ISECompiler.Compiler method), 268
- LogDebug() (Compiler.LSECompiler.Compiler method), 269
- LogDebug() (Compiler.QuartusCompiler.Compiler method), 271
- LogDebug() (Compiler.VivadoCompiler.Compiler method), 273
- LogDebug() (Compiler.XCICompiler.Compiler method), 275
- LogDebug() (Compiler.XCOCCompiler.Compiler method), 276
- LogDebug() (Compiler.XSTCompiler.Compiler method), 278
- LogDebug() (Simulator.ActiveHDL Simulator.Simulator method), 332
- LogDebug() (Simulator.Cocotb Simulator.Simulator method), 334
- LogDebug() (Simulator.GHDL Simulator.Simulator method), 336
- LogDebug() (Simulator.ISE Simulator.Simulator method), 337
- LogDebug() (Simulator.ModelSim Simulator.Simulator method), 339
- LogDebug() (Simulator.Questa Simulator.Simulator method), 340
- LogDebug() (Simulator.Simulator method), 346
- LogDebug() (Simulator.Vivado Simulator.Simulator method), 342
- LogDebug() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- LogDebug() (ToolChain.Aldec.ActiveHDL.VHDL Compiler method), 353
- LogDebug() (ToolChain.Aldec.ActiveHDL.VHDL Library Tool method), 351
- LogDebug() (ToolChain.Aldec.ActiveHDL.VHDL Standalone Simulator method), 354
- LogDebug() (ToolChain.Aldec.Configuration method), 362
- LogDebug() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- LogDebug() (ToolChain.Aldec.RivieraPRO.VHDL Compiler method), 359
- LogDebug() (ToolChain.Aldec.RivieraPRO.VHDL Library Tool method), 358
- LogDebug() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 360
- LogDebug() (ToolChain.Altera.Configuration method), 380
- LogDebug() (ToolChain.Altera.ModelSim.Altera Edition Configuration method), 368
- LogDebug() (ToolChain.Altera.ModelSim.Altera Starter Edition Configuration method), 370
- LogDebug() (ToolChain.Altera.ModelSim.Configuration method), 365
- LogDebug() (ToolChain.Altera.Quartus.Configuration method), 373
- LogDebug() (ToolChain.Altera.Quartus.Map method), 375
- LogDebug() (ToolChain.Altera.Quartus.TclShell method), 376
- LogDebug() (ToolChain.Configuration method), 490
- LogDebug() (ToolChain.Configurator method), 497
- LogDebug() (ToolChain.GHDL.Configuration method), 383
- LogDebug() (ToolChain.GHDL.GHDL method), 387
- LogDebug() (ToolChain.GHDL.GHDL Analyze method), 389
- LogDebug() (ToolChain.GHDL.GHDL Elaborate method), 392
- LogDebug() (ToolChain.GHDL.GHDL Run method), 395
- LogDebug() (ToolChain.Git.Configuration method), 404
- LogDebug() (ToolChain.Git.Git Config method), 410
- LogDebug() (ToolChain.Git.Git Describe method), 409
- LogDebug() (ToolChain.Git.Git RevList method), 408
- LogDebug() (ToolChain.Git.Git RevParse method), 407
- LogDebug() (ToolChain.Git.Git SCM method), 405
- LogDebug() (ToolChain.GNU.Make method), 398
- LogDebug() (ToolChain.GTKWave.Configuration method), 400
- LogDebug() (ToolChain.GTKWave.GTKWave method), 401

[LogDebug\(\) \(ToolChain.Intel.Configuration method\), 424](#)
[LogDebug\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[LogDebug\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 415](#)
[LogDebug\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 417](#)
[LogDebug\(\) \(ToolChain.Intel.Quartus.Configuration method\), 420](#)
[LogDebug\(\) \(ToolChain.Intel.Quartus.Map method\), 422](#)
[LogDebug\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[LogDebug\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 427](#)
[LogDebug\(\) \(ToolChain.Lattice.Diamond.Synth method\), 429](#)
[LogDebug\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 431](#)
[LogDebug\(\) \(ToolChain.Mentor.Configuration method\), 458](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 438](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 440](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.ModelSimSEConfiguration method\), 442](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.ModelSimSEConfiguration method\), 445](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.Selector method\), 446](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler method\), 450](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.VHDLLibraryTool method\), 447](#)
[LogDebug\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 454](#)
[LogDebug\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[LogDebug\(\) \(ToolChain.PoC.Configuration method\), 460](#)
[LogDebug\(\) \(ToolChain.Synopsys.Configuration method\), 463](#)
[LogDebug\(\) \(ToolChain.ToolConfiguration method\), 494](#)
[LogDebug\(\) \(ToolChain.ToolSelector method\), 496](#)
[LogDebug\(\) \(ToolChain.VendorConfiguration method\), 491](#)
[LogDebug\(\) \(ToolChain.Windows.Cmd method\), 465](#)
[LogDebug\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[LogDebug\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 467](#)
[LogDebug\(\) \(ToolChain.Xilinx.ISE.CoreGenerator method\), 473](#)
[LogDebug\(\) \(ToolChain.Xilinx.ISE.Fuse method\), 470](#)
[LogDebug\(\) \(ToolChain.Xilinx.ISE.ISE Simulator method\), 471](#)
[LogDebug\(\) \(ToolChain.Xilinx.ISE.Xst method\), 472](#)
[LogDebug\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)
[LogDebug\(\) \(ToolChain.Xilinx.Vivado.Synth method\), 482](#)
[LogDebug\(\) \(ToolChain.Xilinx.Vivado.XElab method\), 480](#)
[LogDebug\(\) \(ToolChain.Xilinx.Vivado.XSim method\), 481](#)
[LogDryRun\(\) \(Base.Executable.Executable method\), 255](#)
[LogDryRun\(\) \(Base.IHost method\), 267](#)
[LogDryRun\(\) \(Base.Logging.ILogable method\), 257](#)
[LogDryRun\(\) \(Base.Shared.Shared method\), 266](#)
[LogDryRun\(\) \(Compiler.Compiler method\), 283](#)
[LogDryRun\(\) \(Compiler.ISECompiler.Compiler method\), 268](#)
[LogDryRun\(\) \(Compiler.LSECompiler.Compiler method\), 269](#)
[LogDryRun\(\) \(Compiler.QuartusCompiler.Compiler method\), 271](#)
[LogDryRun\(\) \(Compiler.VivadoCompiler.Compiler method\), 273](#)
[LogDryRun\(\) \(Compiler.XCICCompiler.Compiler method\), 275](#)
[LogDryRun\(\) \(Compiler.XCOCCompiler.Compiler method\), 276](#)
[LogDryRun\(\) \(Compiler.XSTCompiler.Compiler method\), 278](#)
[LogDryRun\(\) \(Simulator.ActiveHDL Simulator.Simulator method\), 332](#)
[LogDryRun\(\) \(Simulator.Cocotb Simulator.Simulator method\), 334](#)
[LogDryRun\(\) \(Simulator.GHDL Simulator.Simulator method\), 336](#)
[LogDryRun\(\) \(Simulator.ISE Simulator.Simulator method\), 337](#)
[LogDryRun\(\) \(Simulator.ModelSim Simulator.Simulator method\), 339](#)
[LogDryRun\(\) \(Simulator.Questa Simulator.Simulator method\), 340](#)
[LogDryRun\(\) \(Simulator.Simulator method\), 346](#)
[LogDryRun\(\) \(Simulator.Vivado Simulator.Simulator method\), 342](#)
[LogDryRun\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 349](#)
[LogDryRun\(\) \(ToolChain.Aldec.ActiveHDL.VHDLCompiler method\), 353](#)
[LogDryRun\(\) \(ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method\), 351](#)
[LogDryRun\(\) \(ToolChain.Aldec.ActiveHDL.VHDLStandalone Simulator method\), 354](#)
[LogDryRun\(\) \(ToolChain.Aldec.Configuration method\), 363](#)
[LogDryRun\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 363](#)

method), 356

LogDryRun() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359

LogDryRun() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358

LogDryRun() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 360

LogDryRun() (ToolChain.Altera.Configuration method), 380

LogDryRun() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368

LogDryRun() (ToolChain.Altera.ModelSim.Altera StarterEditionConfiguration method), 370

LogDryRun() (ToolChain.Altera.ModelSim.Configuration method), 365

LogDryRun() (ToolChain.Altera.Quartus.Configuration method), 373

LogDryRun() (ToolChain.Altera.Quartus.Map method), 375

LogDryRun() (ToolChain.Altera.Quartus.TclShell method), 376

LogDryRun() (ToolChain.Configuration method), 490

LogDryRun() (ToolChain.Configurator method), 497

LogDryRun() (ToolChain.GHDL.Configuration method), 383

LogDryRun() (ToolChain.GHDL.GHDL method), 387

LogDryRun() (ToolChain.GHDL.GHDLAnalyze method), 389

LogDryRun() (ToolChain.GHDL.GHDL Elaborate method), 392

LogDryRun() (ToolChain.GHDL.GHDLRun method), 395

LogDryRun() (ToolChain.Git.Configuration method), 404

LogDryRun() (ToolChain.Git.GitConfig method), 410

LogDryRun() (ToolChain.Git.GitDescribe method), 409

LogDryRun() (ToolChain.Git.GitRevList method), 408

LogDryRun() (ToolChain.Git.GitRevParse method), 407

LogDryRun() (ToolChain.Git.GitSCM method), 406

LogDryRun() (ToolChain.GNU.Make method), 398

LogDryRun() (ToolChain.GTKWave.Configuration method), 400

LogDryRun() (ToolChain.GTKWave.GTKWave method), 401

LogDryRun() (ToolChain.Intel.Configuration method), 424

LogDryRun() (ToolChain.Intel.ModelSim.Configuration method), 413

LogDryRun() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415

LogDryRun() (ToolChain.Intel.ModelSim.Intel StarterEditionConfiguration method), 417

LogDryRun() (ToolChain.Intel.Quartus.Configuration method), 420

LogDryRun() (ToolChain.Intel.Quartus.Map method), 422

LogDryRun() (ToolChain.Lattice.Configuration method), 434

LogDryRun() (ToolChain.Lattice.Diamond.Configuration method), 427

LogDryRun() (ToolChain.Lattice.Diamond.Synth method), 429

LogDryRun() (ToolChain.Lattice.Synplify.Configuration method), 431

LogDryRun() (ToolChain.Mentor.Configuration method), 458

LogDryRun() (ToolChain.Mentor.ModelSim.Configuration method), 438

LogDryRun() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440

LogDryRun() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442

LogDryRun() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445

LogDryRun() (ToolChain.Mentor.ModelSim.Selector method), 446

LogDryRun() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450

LogDryRun() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 447

LogDryRun() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454

LogDryRun() (ToolChain.Mentor.QuestaSim.Configuration method), 456

LogDryRun() (ToolChain.PoC.Configuration method), 460

LogDryRun() (ToolChain.Synopsys.Configuration method), 463

LogDryRun() (ToolChain.ToolConfiguration method), 494

LogDryRun() (ToolChain.ToolSelector method), 496

LogDryRun() (ToolChain.VendorConfiguration method), 491

LogDryRun() (ToolChain.Windows.Cmd method), 465

LogDryRun() (ToolChain.Xilinx.Configuration method), 486

LogDryRun() (ToolChain.Xilinx.ISE.Configuration method), 467

LogDryRun() (ToolChain.Xilinx.ISE.CoreGenerator method), 473

LogDryRun() (ToolChain.Xilinx.ISE.Fuse method), 470

LogDryRun() (ToolChain.Xilinx.ISE.ISE Simulator method), 471

LogDryRun() (ToolChain.Xilinx.ISE.Xst method), 472

LogDryRun() (ToolChain.Xilinx.Vivado.Configuration method), 477

LogDryRun() (ToolChain.Xilinx.Vivado.Synth method), 482

LogDryRun() (ToolChain.Xilinx.Vivado.XElab method), 480

LogDryRun() (ToolChain.Xilinx.Vivado.XSim method), 481

LogEntry (class in Base.Logging), 256

- LogError() (Base.Executable.Executable method), [255](#)
- LogError() (Base.IHost method), [267](#)
- LogError() (Base.Logging.ILogable method), [257](#)
- LogError() (Base.Shared.Shared method), [266](#)
- LogError() (Compiler.Compiler method), [283](#)
- LogError() (Compiler.ISECompiler.Compiler method), [268](#)
- LogError() (Compiler.LSECompiler.Compiler method), [269](#)
- LogError() (Compiler.QuartusCompiler.Compiler method), [271](#)
- LogError() (Compiler.VivadoCompiler.Compiler method), [273](#)
- LogError() (Compiler.XCICompiler.Compiler method), [275](#)
- LogError() (Compiler.XCOCCompiler.Compiler method), [276](#)
- LogError() (Compiler.XSTCompiler.Compiler method), [278](#)
- LogError() (Simulator.ActiveHDL Simulator.Simulator method), [332](#)
- LogError() (Simulator.Cocotb Simulator.Simulator method), [334](#)
- LogError() (Simulator.GHDL Simulator.Simulator method), [336](#)
- LogError() (Simulator.ISE Simulator.Simulator method), [337](#)
- LogError() (Simulator.ModelSim Simulator.Simulator method), [339](#)
- LogError() (Simulator.Questa Simulator.Simulator method), [340](#)
- LogError() (Simulator.Simulator method), [346](#)
- LogError() (Simulator.Vivado Simulator.Simulator method), [342](#)
- LogError() (ToolChain.Aldec.ActiveHDL.Configuration method), [349](#)
- LogError() (ToolChain.Aldec.ActiveHDL.VHDLCompile method), [353](#)
- LogError() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), [351](#)
- LogError() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), [354](#)
- LogError() (ToolChain.Aldec.Configuration method), [363](#)
- LogError() (ToolChain.Aldec.RivieraPRO.Configuration method), [356](#)
- LogError() (ToolChain.Aldec.RivieraPRO.VHDLCompile method), [359](#)
- LogError() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), [358](#)
- LogError() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), [361](#)
- LogError() (ToolChain.Altera.Configuration method), [380](#)
- LogError() (ToolChain.Altera.ModelSim.AltaraEditionConfiguration method), [368](#)
- LogError() (ToolChain.Altera.ModelSim.Altara Starter Edition Configuration method), [370](#)
- LogError() (ToolChain.Altera.ModelSim.Configuration method), [365](#)
- LogError() (ToolChain.Altera.Quartus.Configuration method), [373](#)
- LogError() (ToolChain.Altera.Quartus.Map method), [375](#)
- LogError() (ToolChain.Altera.Quartus.TclShell method), [376](#)
- LogError() (ToolChain.Configuration method), [490](#)
- LogError() (ToolChain.Configurator method), [497](#)
- LogError() (ToolChain.GHDL.Configuration method), [383](#)
- LogError() (ToolChain.GHDL.GHDL method), [387](#)
- LogError() (ToolChain.GHDL.GHDLAnalyze method), [389](#)
- LogError() (ToolChain.GHDL.GHDL Elaborate method), [392](#)
- LogError() (ToolChain.GHDL.GHDLRun method), [395](#)
- LogError() (ToolChain.Git.Configuration method), [404](#)
- LogError() (ToolChain.Git.GitConfig method), [410](#)
- LogError() (ToolChain.Git.GitDescribe method), [409](#)
- LogError() (ToolChain.Git.GitRevList method), [408](#)
- LogError() (ToolChain.Git.GitRevParse method), [407](#)
- LogError() (ToolChain.Git.GitSCM method), [406](#)
- LogError() (ToolChain.GNU.Make method), [398](#)
- LogError() (ToolChain.GTKWave.Configuration method), [400](#)
- LogError() (ToolChain.GTKWave.GTKWave method), [402](#)
- LogError() (ToolChain.Intel.Configuration method), [424](#)
- LogError() (ToolChain.Intel.ModelSim.Configuration method), [413](#)
- LogError() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), [415](#)
- LogError() (ToolChain.Intel.ModelSim.Intel Starter Edition Configuration method), [418](#)
- LogError() (ToolChain.Intel.Quartus.Configuration method), [420](#)
- LogError() (ToolChain.Intel.Quartus.Map method), [422](#)
- LogError() (ToolChain.Lattice.Configuration method), [434](#)
- LogError() (ToolChain.Lattice.Diamond.Configuration method), [427](#)
- LogError() (ToolChain.Lattice.Diamond.Synth method), [429](#)
- LogError() (ToolChain.Lattice.Synplify.Configuration method), [431](#)
- LogError() (ToolChain.Mentor.Configuration method), [458](#)
- LogError() (ToolChain.Mentor.ModelSim.Configuration method), [438](#)
- LogError() (ToolChain.Mentor.ModelSim.ModelSimPE Configuration method), [440](#)
- LogError() (ToolChain.Mentor.ModelSim.ModelSimSE32 Configuration method), [442](#)

[LogError\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 278](#)
[LogError\(\) \(ToolChain.Mentor.ModelSim.Selector method\), 445](#)
[LogError\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler method\), 446](#)
[LogError\(\) \(ToolChain.Mentor.ModelSim.VHDLLibraryTool method\), 450](#)
[LogError\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 447](#)
[LogError\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 454](#)
[LogError\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[LogError\(\) \(ToolChain.PoC.Configuration method\), 460](#)
[LogError\(\) \(ToolChain.Synopsys.Configuration method\), 463](#)
[LogError\(\) \(ToolChain.ToolConfiguration method\), 494](#)
[LogError\(\) \(ToolChain.ToolSelector method\), 496](#)
[LogError\(\) \(ToolChain.VendorConfiguration method\), 492](#)
[LogError\(\) \(ToolChain.Windows.Cmd method\), 465](#)
[LogError\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[LogError\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 467](#)
[LogError\(\) \(ToolChain.Xilinx.ISE.CoreGenerator method\), 473](#)
[LogError\(\) \(ToolChain.Xilinx.ISE.Fuse method\), 470](#)
[LogError\(\) \(ToolChain.Xilinx.ISE.ISE Simulator method\), 471](#)
[LogError\(\) \(ToolChain.Xilinx.ISE.Xst method\), 472](#)
[LogError\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)
[LogError\(\) \(ToolChain.Xilinx.Vivado.Synth method\), 482](#)
[LogError\(\) \(ToolChain.Xilinx.Vivado.XElab method\), 480](#)
[LogError\(\) \(ToolChain.Xilinx.Vivado.XSim method\), 481](#)
[LogFatal\(\) \(Base.Executable.Executable method\), 255](#)
[LogFatal\(\) \(Base.IHost method\), 267](#)
[LogFatal\(\) \(Base.Logging.ILogable method\), 257](#)
[LogFatal\(\) \(Base.Shared.Shared method\), 266](#)
[LogFatal\(\) \(Compiler.Compiler method\), 283](#)
[LogFatal\(\) \(Compiler.ISECompiler.Compiler method\), 268](#)
[LogFatal\(\) \(Compiler.LSECompiler.Compiler method\), 269](#)
[LogFatal\(\) \(Compiler.QuartusCompiler.Compiler method\), 271](#)
[LogFatal\(\) \(Compiler.VivadoCompiler.Compiler method\), 273](#)
[LogFatal\(\) \(Compiler.XCICompiler.Compiler method\), 275](#)
[LogFatal\(\) \(Compiler.XCOCCompiler.Compiler method\), 276](#)
[LogFatal\(\) \(Compiler.XSTCompiler.Compiler method\), 278](#)
[LogFatal\(\) \(Simulator.ActiveHDL.Simulator.Simulator method\), 332](#)
[LogFatal\(\) \(Simulator.Cocotb.Simulator.Simulator method\), 334](#)
[LogFatal\(\) \(Simulator.GHDL.Simulator.Simulator method\), 336](#)
[LogFatal\(\) \(Simulator.ISE.Simulator.Simulator method\), 337](#)
[LogFatal\(\) \(Simulator.ModelSim.Simulator.Simulator method\), 339](#)
[LogFatal\(\) \(Simulator.Questa.Simulator.Simulator method\), 340](#)
[LogFatal\(\) \(Simulator.Simulator method\), 346](#)
[LogFatal\(\) \(Simulator.Vivado.Simulator.Simulator method\), 342](#)
[LogFatal\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 349](#)
[LogFatal\(\) \(ToolChain.Aldec.ActiveHDL.VHDLCompiler method\), 353](#)
[LogFatal\(\) \(ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method\), 351](#)
[LogFatal\(\) \(ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method\), 354](#)
[LogFatal\(\) \(ToolChain.Aldec.Configuration method\), 363](#)
[LogFatal\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 356](#)
[LogFatal\(\) \(ToolChain.Aldec.RivieraPRO.VHDLCompiler method\), 359](#)
[LogFatal\(\) \(ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method\), 358](#)
[LogFatal\(\) \(ToolChain.Aldec.RivieraPRO.VHDL Simulator method\), 361](#)
[LogFatal\(\) \(ToolChain.Altera.Configuration method\), 380](#)
[LogFatal\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method\), 368](#)
[LogFatal\(\) \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method\), 370](#)
[LogFatal\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 365](#)
[LogFatal\(\) \(ToolChain.Altera.Quartus.Configuration method\), 373](#)
[LogFatal\(\) \(ToolChain.Altera.Quartus.Map method\), 375](#)
[LogFatal\(\) \(ToolChain.Altera.Quartus.TclShell method\), 376](#)
[LogFatal\(\) \(ToolChain.Configuration method\), 490](#)
[LogFatal\(\) \(ToolChain.Configurator method\), 497](#)
[LogFatal\(\) \(ToolChain.GHDL.Configuration method\), 383](#)
[LogFatal\(\) \(ToolChain.GHDL.GHDL method\), 387](#)
[LogFatal\(\) \(ToolChain.GHDL.GHDLAnalyze method\), 389](#)
[LogFatal\(\) \(ToolChain.GHDL.GHDL Elaborate method\), 392](#)
[LogFatal\(\) \(ToolChain.GHDL.GHDLRun method\), 392](#)

- 395
- LogFatal() (ToolChain.Git.Configuration method), 404
 - LogFatal() (ToolChain.Git.GitConfig method), 410
 - LogFatal() (ToolChain.Git.GitDescribe method), 409
 - LogFatal() (ToolChain.Git.GitRevList method), 408
 - LogFatal() (ToolChain.Git.GitRevParse method), 407
 - LogFatal() (ToolChain.Git.GitSCM method), 406
 - LogFatal() (ToolChain.GNU.Make method), 398
 - LogFatal() (ToolChain.GTKWave.Configuration method), 400
 - LogFatal() (ToolChain.GTKWave.GTKWave method), 402
 - LogFatal() (ToolChain.Intel.Configuration method), 424
 - LogFatal() (ToolChain.Intel.ModelSim.Configuration method), 413
 - LogFatal() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
 - LogFatal() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
 - LogFatal() (ToolChain.Intel.Quartus.Configuration method), 420
 - LogFatal() (ToolChain.Intel.Quartus.Map method), 422
 - LogFatal() (ToolChain.Lattice.Configuration method), 434
 - LogFatal() (ToolChain.Lattice.Diamond.Configuration method), 427
 - LogFatal() (ToolChain.Lattice.Diamond.Synth method), 429
 - LogFatal() (ToolChain.Lattice.Synplify.Configuration method), 431
 - LogFatal() (ToolChain.Mentor.Configuration method), 458
 - LogFatal() (ToolChain.Mentor.ModelSim.Configuration method), 438
 - LogFatal() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
 - LogFatal() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
 - LogFatal() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
 - LogFatal() (ToolChain.Mentor.ModelSim.Selector method), 446
 - LogFatal() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450
 - LogFatal() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 447
 - LogFatal() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454
 - LogFatal() (ToolChain.Mentor.QuestaSim.Configuration method), 456
 - LogFatal() (ToolChain.PoC.Configuration method), 460
 - LogFatal() (ToolChain.Synopsys.Configuration method), 463
 - LogFatal() (ToolChain.ToolConfiguration method), 494
 - LogFatal() (ToolChain.ToolSelector method), 496
 - LogFatal() (ToolChain.VendorConfiguration method), 492
 - LogFatal() (ToolChain.Windows.Cmd method), 465
 - LogFatal() (ToolChain.Xilinx.Configuration method), 486
 - LogFatal() (ToolChain.Xilinx.ISE.Configuration method), 467
 - LogFatal() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
 - LogFatal() (ToolChain.Xilinx.ISE.Fuse method), 470
 - LogFatal() (ToolChain.Xilinx.ISE.ISE Simulator method), 471
 - LogFatal() (ToolChain.Xilinx.ISE.Xst method), 472
 - LogFatal() (ToolChain.Xilinx.Vivado.Configuration method), 477
 - LogFatal() (ToolChain.Xilinx.Vivado.Synth method), 482
 - LogFatal() (ToolChain.Xilinx.Vivado.XElab method), 480
 - LogFatal() (ToolChain.Xilinx.Vivado.XSim method), 481
 - LogFile (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
 - Logger (Base.Executable.Executable attribute), 255
 - Logger (Base.IHost attribute), 267
 - Logger (Base.Logging.ILogable attribute), 257
 - Logger (Base.Shared.Shared attribute), 266
 - Logger (class in Base.Logging), 256
 - Logger (Compiler.Compiler attribute), 283
 - Logger (Compiler.ISECompiler.Compiler attribute), 268
 - Logger (Compiler.LSECompiler.Compiler attribute), 270
 - Logger (Compiler.QuartusCompiler.Compiler attribute), 271
 - Logger (Compiler.VivadoCompiler.Compiler attribute), 273
 - Logger (Compiler.XCICompiler.Compiler attribute), 274
 - Logger (Compiler.XCOCCompiler.Compiler attribute), 275
 - Logger (Compiler.XSTCompiler.Compiler attribute), 278
 - Logger (Simulator.ActiveHDL Simulator.Simulator attribute), 333
 - Logger (Simulator.Cocotb Simulator.Simulator attribute), 334
 - Logger (Simulator.GHDL Simulator.Simulator attribute), 336
 - Logger (Simulator.ISE Simulator.Simulator attribute), 337
 - Logger (Simulator.ModelSim Simulator.Simulator attribute), 339
 - Logger (Simulator.Questa Simulator.Simulator attribute), 340
 - Logger (Simulator.Simulator attribute), 346
 - Logger (Simulator.Vivado Simulator.Simulator attribute), 342
 - Logger (ToolChain.Aldec.ActiveHDL.Configuration

- attribute), 350
- Logger (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 353
- Logger (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 351
- Logger (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 354
- Logger (ToolChain.Aldec.Configuration attribute), 363
- Logger (ToolChain.Aldec.RivieraPRO.Configuration attribute), 356
- Logger (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 359
- Logger (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358
- Logger (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 361
- Logger (ToolChain.Altera.Configuration attribute), 380
- Logger (ToolChain.Altera.ModelSim.AlterEditionConfiguration attribute), 368
- Logger (ToolChain.Altera.ModelSim.AlterStarterEditionConfiguration attribute), 370
- Logger (ToolChain.Altera.ModelSim.Configuration attribute), 365
- Logger (ToolChain.Altera.Quartus.Configuration attribute), 374
- Logger (ToolChain.Altera.Quartus.Map attribute), 376
- Logger (ToolChain.Altera.Quartus.TclShell attribute), 376
- Logger (ToolChain.Configuration attribute), 490
- Logger (ToolChain.Configurator attribute), 497
- Logger (ToolChain.GHDL.Configuration attribute), 383
- Logger (ToolChain.GHDL.GHDL attribute), 387
- Logger (ToolChain.GHDL.GHDLAnalyze attribute), 389
- Logger (ToolChain.GHDL.GHDL Elaborate attribute), 392
- Logger (ToolChain.GHDL.GHDLRun attribute), 395
- Logger (ToolChain.Git.Configuration attribute), 404
- Logger (ToolChain.Git.GitConfig attribute), 410
- Logger (ToolChain.Git.GitDescribe attribute), 409
- Logger (ToolChain.Git.GitRevList attribute), 408
- Logger (ToolChain.Git.GitRevParse attribute), 407
- Logger (ToolChain.Git.GitSCM attribute), 406
- Logger (ToolChain.GNU.Make attribute), 398
- Logger (ToolChain.GTKWave.Configuration attribute), 400
- Logger (ToolChain.GTKWave.GTKWave attribute), 402
- Logger (ToolChain.Intel.Configuration attribute), 424
- Logger (ToolChain.Intel.ModelSim.Configuration attribute), 413
- Logger (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 415
- Logger (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 418
- Logger (ToolChain.Intel.Quartus.Configuration attribute), 420
- Logger (ToolChain.Intel.Quartus.Map attribute), 422
- Logger (ToolChain.Lattice.Configuration attribute), 434
- Logger (ToolChain.Lattice.Diamond.Configuration attribute), 427
- Logger (ToolChain.Lattice.Diamond.Synth attribute), 429
- Logger (ToolChain.Lattice.Synplify.Configuration attribute), 431
- Logger (ToolChain.Mentor.Configuration attribute), 459
- Logger (ToolChain.Mentor.ModelSim.Configuration attribute), 438
- Logger (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 440
- Logger (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 443
- Logger (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute), 445
- Logger (ToolChain.Mentor.ModelSim.Selector attribute), 446
- Logger (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 451
- Logger (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 448
- Logger (ToolChain.Mentor.ModelSim.VHDL Simulator attribute), 454
- Logger (ToolChain.Mentor.QuestaSim.Configuration attribute), 456
- Logger (ToolChain.PoC.Configuration attribute), 461
- Logger (ToolChain.Synopsys.Configuration attribute), 463
- Logger (ToolChain.ToolConfiguration attribute), 494
- Logger (ToolChain.ToolSelector attribute), 496
- Logger (ToolChain.VendorConfiguration attribute), 492
- Logger (ToolChain.Windows.Cmd attribute), 465
- Logger (ToolChain.Xilinx.Configuration attribute), 486
- Logger (ToolChain.Xilinx.ISE.Configuration attribute), 468
- Logger (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473
- Logger (ToolChain.Xilinx.ISE.Fuse attribute), 470
- Logger (ToolChain.Xilinx.ISE.ISE Simulator attribute), 471
- Logger (ToolChain.Xilinx.ISE.Xst attribute), 472
- Logger (ToolChain.Xilinx.Vivado.Configuration attribute), 477
- Logger (ToolChain.Xilinx.Vivado.Synth attribute), 482
- Logger (ToolChain.Xilinx.Vivado.XElab attribute), 480
- Logger (ToolChain.Xilinx.Vivado.XSim attribute), 481
- LogicalExpression (class in lib.CodeDOM), 499
- LogInfo() (Base.Executable.Executable method), 255
- LogInfo() (Base.IHost method), 267
- LogInfo() (Base.Logging.ILogable method), 257
- LogInfo() (Base.Shared.Shared method), 266
- LogInfo() (Compiler.Compiler method), 283
- LogInfo() (Compiler.ISECompiler.Compiler method), 268

[LogInfo\(\) \(Compiler.LSECompiler.Compiler method\), 269](#)
[LogInfo\(\) \(Compiler.QuartusCompiler.Compiler method\), 271](#)
[LogInfo\(\) \(Compiler.VivadoCompiler.Compiler method\), 273](#)
[LogInfo\(\) \(Compiler.XCICompiler.Compiler method\), 275](#)
[LogInfo\(\) \(Compiler.XCOCompiler.Compiler method\), 276](#)
[LogInfo\(\) \(Compiler.XSTCompiler.Compiler method\), 278](#)
[LogInfo\(\) \(Simulator.ActiveHDL Simulator.Simulator method\), 332](#)
[LogInfo\(\) \(Simulator.Cocotb Simulator.Simulator method\), 334](#)
[LogInfo\(\) \(Simulator.GHDL Simulator.Simulator method\), 336](#)
[LogInfo\(\) \(Simulator.ISE Simulator.Simulator method\), 337](#)
[LogInfo\(\) \(Simulator.ModelSim Simulator.Simulator method\), 339](#)
[LogInfo\(\) \(Simulator.Questa Simulator.Simulator method\), 340](#)
[LogInfo\(\) \(Simulator.Simulator method\), 346](#)
[LogInfo\(\) \(Simulator.Vivado Simulator.Simulator method\), 342](#)
[LogInfo\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 349](#)
[LogInfo\(\) \(ToolChain.Aldec.ActiveHDL.VHDLCompiler method\), 353](#)
[LogInfo\(\) \(ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method\), 351](#)
[LogInfo\(\) \(ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method\), 354](#)
[LogInfo\(\) \(ToolChain.Aldec.Configuration method\), 363](#)
[LogInfo\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 356](#)
[LogInfo\(\) \(ToolChain.Aldec.RivieraPRO.VHDLCompiler method\), 359](#)
[LogInfo\(\) \(ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method\), 358](#)
[LogInfo\(\) \(ToolChain.Aldec.RivieraPRO.VHDL Simulator method\), 361](#)
[LogInfo\(\) \(ToolChain.Altera.Configuration method\), 380](#)
[LogInfo\(\) \(ToolChain.Altera.ModelSim.AltaraEditionConfiguration method\), 368](#)
[LogInfo\(\) \(ToolChain.Altera.ModelSim.AltaraStarterEditionConfiguration method\), 370](#)
[LogInfo\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 365](#)
[LogInfo\(\) \(ToolChain.Altera.Quartus.Configuration method\), 374](#)
[LogInfo\(\) \(ToolChain.Altera.Quartus.Map method\), 375](#)
[LogInfo\(\) \(ToolChain.Altera.Quartus.TclShell method\), 376](#)
[LogInfo\(\) \(ToolChain.Configuration method\), 490](#)
[LogInfo\(\) \(ToolChain.Configurator method\), 497](#)
[LogInfo\(\) \(ToolChain.GHDL.Configuration method\), 383](#)
[LogInfo\(\) \(ToolChain.GHDL.GHDL method\), 387](#)
[LogInfo\(\) \(ToolChain.GHDL.GHDLAnalyze method\), 389](#)
[LogInfo\(\) \(ToolChain.GHDL.GHDL Elaborate method\), 392](#)
[LogInfo\(\) \(ToolChain.GHDL.GHDLRun method\), 395](#)
[LogInfo\(\) \(ToolChain.Git.Configuration method\), 404](#)
[LogInfo\(\) \(ToolChain.Git.GitConfig method\), 410](#)
[LogInfo\(\) \(ToolChain.Git.GitDescribe method\), 409](#)
[LogInfo\(\) \(ToolChain.Git.GitRevList method\), 408](#)
[LogInfo\(\) \(ToolChain.Git.GitRevParse method\), 407](#)
[LogInfo\(\) \(ToolChain.Git.GitSCM method\), 406](#)
[LogInfo\(\) \(ToolChain.GNU.Make method\), 398](#)
[LogInfo\(\) \(ToolChain.GTKWave.Configuration method\), 400](#)
[LogInfo\(\) \(ToolChain.GTKWave.GTKWave method\), 402](#)
[LogInfo\(\) \(ToolChain.Intel.Configuration method\), 424](#)
[LogInfo\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[LogInfo\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 415](#)
[LogInfo\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 418](#)
[LogInfo\(\) \(ToolChain.Intel.Quartus.Configuration method\), 420](#)
[LogInfo\(\) \(ToolChain.Intel.Quartus.Map method\), 422](#)
[LogInfo\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[LogInfo\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 427](#)
[LogInfo\(\) \(ToolChain.Lattice.Diamond.Synth method\), 429](#)
[LogInfo\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 431](#)
[LogInfo\(\) \(ToolChain.Mentor.Configuration method\), 459](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 438](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 440](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 442](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 445](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.Selector method\), 446](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.VHDLCompiler method\), 450](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.VHDLLibraryTool method\), 447](#)
[LogInfo\(\) \(ToolChain.Mentor.ModelSim.VHDL Simulator method\), 454](#)

- LogInfo() (ToolChain.Mentor.QuestaSim.Configuration method), 456
- LogInfo() (ToolChain.PoC.Configuration method), 460
- LogInfo() (ToolChain.Synopsys.Configuration method), 463
- LogInfo() (ToolChain.ToolConfiguration method), 494
- LogInfo() (ToolChain.ToolSelector method), 496
- LogInfo() (ToolChain.VendorConfiguration method), 492
- LogInfo() (ToolChain.Windows.Cmd method), 465
- LogInfo() (ToolChain.Xilinx.Configuration method), 486
- LogInfo() (ToolChain.Xilinx.ISE.Configuration method), 468
- LogInfo() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- LogInfo() (ToolChain.Xilinx.ISE.Fuse method), 470
- LogInfo() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- LogInfo() (ToolChain.Xilinx.ISE.Xst method), 472
- LogInfo() (ToolChain.Xilinx.Vivado.Configuration method), 477
- LogInfo() (ToolChain.Xilinx.Vivado.Synth method), 482
- LogInfo() (ToolChain.Xilinx.Vivado.XElab method), 480
- LogInfo() (ToolChain.Xilinx.Vivado.XSim method), 481
- LogLevel (Base.Logging.Logger attribute), 256
- LogNormal() (Base.Executable.Executable method), 255
- LogNormal() (Base.IHost method), 267
- LogNormal() (Base.Logging.ILogable method), 257
- LogNormal() (Base.Shared.Shared method), 266
- LogNormal() (Compiler.Compiler method), 283
- LogNormal() (Compiler.ISECompiler.Compiler method), 268
- LogNormal() (Compiler.LSECompiler.Compiler method), 269
- LogNormal() (Compiler.QuartusCompiler.Compiler method), 271
- LogNormal() (Compiler.VivadoCompiler.Compiler method), 273
- LogNormal() (Compiler.XCICompiler.Compiler method), 275
- LogNormal() (Compiler.XCOCompiler.Compiler method), 276
- LogNormal() (Compiler.XSTCompiler.Compiler method), 278
- LogNormal() (Simulator.ActiveHDL Simulator.Simulator method), 333
- LogNormal() (Simulator.Cocotb Simulator.Simulator method), 334
- LogNormal() (Simulator.GHDL Simulator.Simulator method), 336
- LogNormal() (Simulator.ISE Simulator.Simulator method), 337
- LogNormal() (Simulator.ModelSim Simulator.Simulator method), 339
- LogNormal() (Simulator.Questa Simulator.Simulator method), 340
- LogNormal() (Simulator.Simulator method), 346
- LogNormal() (Simulator.Vivado Simulator.Simulator method), 342
- LogNormal() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- LogNormal() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- LogNormal() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
- LogNormal() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 354
- LogNormal() (ToolChain.Aldec.Configuration method), 363
- LogNormal() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- LogNormal() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- LogNormal() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- LogNormal() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361
- LogNormal() (ToolChain.Altera.Configuration method), 380
- LogNormal() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
- LogNormal() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
- LogNormal() (ToolChain.Altera.ModelSim.Configuration method), 365
- LogNormal() (ToolChain.Altera.Quartus.Configuration method), 374
- LogNormal() (ToolChain.Altera.Quartus.Map method), 376
- LogNormal() (ToolChain.Altera.Quartus.TclShell method), 376
- LogNormal() (ToolChain.Configuration method), 490
- LogNormal() (ToolChain.Configurator method), 497
- LogNormal() (ToolChain.GHDL.Configuration method), 383
- LogNormal() (ToolChain.GHDL.GHDL method), 387
- LogNormal() (ToolChain.GHDL.GHDLAnalyze method), 389
- LogNormal() (ToolChain.GHDL.GHDL Elaborate method), 392
- LogNormal() (ToolChain.GHDL.GHDLRun method), 395
- LogNormal() (ToolChain.Git.Configuration method), 404
- LogNormal() (ToolChain.Git.GitConfig method), 410
- LogNormal() (ToolChain.Git.GitDescribe method), 409
- LogNormal() (ToolChain.Git.GitRevList method), 408
- LogNormal() (ToolChain.Git.GitRevParse method), 407

- LogNormal() (ToolChain.Git.GitSCM method), 406
- LogNormal() (ToolChain.GNU.Make method), 398
- LogNormal() (ToolChain.GTKWave.Configuration method), 400
- LogNormal() (ToolChain.GTKWave.GTKWave method), 402
- LogNormal() (ToolChain.Intel.Configuration method), 424
- LogNormal() (ToolChain.Intel.ModelSim.Configuration method), 413
- LogNormal() (ToolChain.Intel.ModelSim.IntelEditionConfigurationmethod), 477
- LogNormal() (ToolChain.Intel.ModelSim.IntelEditionConfigurationmethod), 415
- LogNormal() (ToolChain.Intel.ModelSim.IntelStarterEditionConfigurationmethod), 482
- LogNormal() (ToolChain.Intel.Quartus.Configuration method), 420
- LogNormal() (ToolChain.Intel.Quartus.Map method), 422
- LogNormal() (ToolChain.Lattice.Configuration method), 434
- LogNormal() (ToolChain.Lattice.Diamond.Configuration method), 427
- LogNormal() (ToolChain.Lattice.Diamond.Synth method), 429
- LogNormal() (ToolChain.Lattice.Synplify.Configuration method), 431
- LogNormal() (ToolChain.Mentor.Configuration method), 459
- LogNormal() (ToolChain.Mentor.ModelSim.Configuration method), 438
- LogNormal() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
- LogNormal() (ToolChain.Mentor.ModelSim.ModelSimSE3Configuration method), 442
- LogNormal() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
- LogNormal() (ToolChain.Mentor.ModelSim.Selector method), 446
- LogNormal() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450
- LogNormal() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 447
- LogNormal() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454
- LogNormal() (ToolChain.Mentor.QuestaSim.Configuration method), 456
- LogNormal() (ToolChain.PoC.Configuration method), 460
- LogNormal() (ToolChain.Synopsys.Configuration method), 463
- LogNormal() (ToolChain.ToolConfiguration method), 494
- LogNormal() (ToolChain.ToolSelector method), 496
- LogNormal() (ToolChain.VendorConfiguration method), 492
- LogNormal() (ToolChain.Windows.Cmd method), 465
- LogNormal() (ToolChain.Xilinx.Configuration method), 486
- LogNormal() (ToolChain.Xilinx.ISE.Configuration method), 468
- LogNormal() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- LogNormal() (ToolChain.Xilinx.ISE.Fuse method), 470
- LogNormal() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- LogNormal() (ToolChain.Xilinx.ISE.Xst method), 472
- LogNormal() (ToolChain.Xilinx.Vivado.Configuration method), 477
- LogNormal() (ToolChain.Xilinx.Vivado.Synth method), 480
- LogNormal() (ToolChain.Xilinx.Vivado.XElab method), 481
- LogNormal() (ToolChain.Xilinx.Vivado.XSim method), 481
- LogQuiet() (Base.Executable.Executable method), 255
- LogQuiet() (Base.IHost method), 267
- LogQuiet() (Base.Logging.ILogable method), 257
- LogQuiet() (Base.Shared.Shared method), 266
- LogQuiet() (Compiler.Compiler method), 283
- LogQuiet() (Compiler.ISECompiler.Compiler method), 268
- LogQuiet() (Compiler.LSECompiler.Compiler method), 269
- LogQuiet() (Compiler.QuartusCompiler.Compiler method), 271
- LogQuiet() (Compiler.VivadoCompiler.Compiler method), 273
- LogQuiet() (Compiler.XCICompiler.Compiler method), 275
- LogQuiet() (Compiler.XCOCCompiler.Compiler method), 276
- LogQuiet() (Compiler.XSTCompiler.Compiler method), 278
- LogQuiet() (Simulator.ActiveHDL Simulator.Simulator method), 333
- LogQuiet() (Simulator.Cocotb Simulator.Simulator method), 334
- LogQuiet() (Simulator.GHDL Simulator.Simulator method), 336
- LogQuiet() (Simulator.ISE Simulator.Simulator method), 337
- LogQuiet() (Simulator.ModelSim Simulator.Simulator method), 339
- LogQuiet() (Simulator.Questa Simulator.Simulator method), 340
- LogQuiet() (Simulator.Simulator method), 346
- LogQuiet() (Simulator.Vivado Simulator.Simulator method), 342
- LogQuiet() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- LogQuiet() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- LogQuiet() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
- LogQuiet() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator

- method), 354
- LogQuiet() (ToolChain.Aldec.Configuration method), 363
- LogQuiet() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- LogQuiet() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- LogQuiet() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- LogQuiet() (ToolChain.Aldec.RivieraPRO.VHDLSimulator method), 361
- LogQuiet() (ToolChain.Altera.Configuration method), 380
- LogQuiet() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
- LogQuiet() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
- LogQuiet() (ToolChain.Altera.ModelSim.Configuration method), 365
- LogQuiet() (ToolChain.Altera.Quartus.Configuration method), 374
- LogQuiet() (ToolChain.Altera.Quartus.Map method), 376
- LogQuiet() (ToolChain.Altera.Quartus.TclShell method), 376
- LogQuiet() (ToolChain.Configuration method), 490
- LogQuiet() (ToolChain.Configurator method), 497
- LogQuiet() (ToolChain.GHDL.Configuration method), 383
- LogQuiet() (ToolChain.GHDL.GHDL method), 387
- LogQuiet() (ToolChain.GHDL.GHDLAnalyze method), 389
- LogQuiet() (ToolChain.GHDL.GHDLElaborate method), 392
- LogQuiet() (ToolChain.GHDL.GHDLRun method), 395
- LogQuiet() (ToolChain.Git.Configuration method), 404
- LogQuiet() (ToolChain.Git.GitConfig method), 410
- LogQuiet() (ToolChain.Git.GitDescribe method), 409
- LogQuiet() (ToolChain.Git.GitRevList method), 408
- LogQuiet() (ToolChain.Git.GitRevParse method), 407
- LogQuiet() (ToolChain.Git.GitSCM method), 406
- LogQuiet() (ToolChain.GNU.Make method), 398
- LogQuiet() (ToolChain.GTKWave.Configuration method), 400
- LogQuiet() (ToolChain.GTKWave.GTKWave method), 402
- LogQuiet() (ToolChain.Intel.Configuration method), 424
- LogQuiet() (ToolChain.Intel.ModelSim.Configuration method), 413
- LogQuiet() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- LogQuiet() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
- LogQuiet() (ToolChain.Intel.Quartus.Configuration method), 420
- LogQuiet() (ToolChain.Intel.Quartus.Map method), 422
- LogQuiet() (ToolChain.Lattice.Configuration method), 434
- LogQuiet() (ToolChain.Lattice.Diamond.Configuration method), 427
- LogQuiet() (ToolChain.Lattice.Diamond.Synth method), 429
- LogQuiet() (ToolChain.Lattice.Synplify.Configuration method), 431
- LogQuiet() (ToolChain.Mentor.Configuration method), 459
- LogQuiet() (ToolChain.Mentor.ModelSim.Configuration method), 438
- LogQuiet() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 440
- LogQuiet() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 442
- LogQuiet() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445
- LogQuiet() (ToolChain.Mentor.ModelSim.Selector method), 446
- LogQuiet() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 450
- LogQuiet() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 447
- LogQuiet() (ToolChain.Mentor.ModelSim.VHDLSimulator method), 454
- LogQuiet() (ToolChain.Mentor.QuestaSim.Configuration method), 456
- LogQuiet() (ToolChain.PoC.Configuration method), 461
- LogQuiet() (ToolChain.Synopsys.Configuration method), 463
- LogQuiet() (ToolChain.ToolConfiguration method), 494
- LogQuiet() (ToolChain.ToolSelector method), 496
- LogQuiet() (ToolChain.VendorConfiguration method), 492
- LogQuiet() (ToolChain.Windows.Cmd method), 465
- LogQuiet() (ToolChain.Xilinx.Configuration method), 486
- LogQuiet() (ToolChain.Xilinx.ISE.Configuration method), 468
- LogQuiet() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- LogQuiet() (ToolChain.Xilinx.ISE.Fuse method), 470
- LogQuiet() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- LogQuiet() (ToolChain.Xilinx.ISE.Xst method), 472
- LogQuiet() (ToolChain.Xilinx.Vivado.Configuration method), 477
- LogQuiet() (ToolChain.Xilinx.Vivado.Synth method), 482
- LogQuiet() (ToolChain.Xilinx.Vivado.XElab method), 480
- LogQuiet() (ToolChain.Xilinx.Vivado.XSim method), 481
- LogVerbose() (Base.Executable.Executable method), 481

- 255
- LogVerbose() (Base.IHost method), 267
- LogVerbose() (Base.Logging.ILogable method), 257
- LogVerbose() (Base.Shared.Shared method), 266
- LogVerbose() (Compiler.Compiler method), 283
- LogVerbose() (Compiler.ISECompiler.Compiler method), 268
- LogVerbose() (Compiler.LSECompiler.Compiler method), 269
- LogVerbose() (Compiler.QuartusCompiler.Compiler method), 271
- LogVerbose() (Compiler.VivadoCompiler.Compiler method), 273
- LogVerbose() (Compiler.XCICompiler.Compiler method), 275
- LogVerbose() (Compiler.XCOCompiler.Compiler method), 277
- LogVerbose() (Compiler.XSTCompiler.Compiler method), 278
- LogVerbose() (Simulator.ActiveHDL Simulator.Simulator method), 333
- LogVerbose() (Simulator.Cocotb Simulator.Simulator method), 334
- LogVerbose() (Simulator.GHDL Simulator.Simulator method), 336
- LogVerbose() (Simulator.ISE Simulator.Simulator method), 337
- LogVerbose() (Simulator.ModelSim Simulator.Simulator method), 339
- LogVerbose() (Simulator.Questa Simulator.Simulator method), 340
- LogVerbose() (Simulator.Simulator method), 346
- LogVerbose() (Simulator.Vivado Simulator.Simulator method), 342
- LogVerbose() (ToolChain.Aldec.ActiveHDL.Configuration method), 349
- LogVerbose() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- LogVerbose() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 351
- LogVerbose() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 354
- LogVerbose() (ToolChain.Aldec.Configuration method), 363
- LogVerbose() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- LogVerbose() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- LogVerbose() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- LogVerbose() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361
- LogVerbose() (ToolChain.Altera.Configuration method), 380
- LogVerbose() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
- LogVerbose() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
- LogVerbose() (ToolChain.Altera.ModelSim.Configuration method), 365
- LogVerbose() (ToolChain.Altera.Quartus.Configuration method), 374
- LogVerbose() (ToolChain.Altera.Quartus.Map method), 376
- LogVerbose() (ToolChain.Altera.Quartus.TclShell method), 376
- LogVerbose() (ToolChain.Configuration method), 490
- LogVerbose() (ToolChain.Configurator method), 497
- LogVerbose() (ToolChain.GHDL.Configuration method), 383
- LogVerbose() (ToolChain.GHDL.GHDL method), 387
- LogVerbose() (ToolChain.GHDL.GHDLAnalyze method), 389
- LogVerbose() (ToolChain.GHDL.GHDL Elaborate method), 392
- LogVerbose() (ToolChain.GHDL.GHDLRun method), 395
- LogVerbose() (ToolChain.Git.Configuration method), 404
- LogVerbose() (ToolChain.Git.GitConfig method), 410
- LogVerbose() (ToolChain.Git.GitDescribe method), 409
- LogVerbose() (ToolChain.Git.GitRevList method), 408
- LogVerbose() (ToolChain.Git.GitRevParse method), 407
- LogVerbose() (ToolChain.Git.GitSCM method), 406
- LogVerbose() (ToolChain.GNU.Make method), 398
- LogVerbose() (ToolChain.GTKWave.Configuration method), 400
- LogVerbose() (ToolChain.GTKWave.GTKWave method), 402
- LogVerbose() (ToolChain.Intel.Configuration method), 424
- LogVerbose() (ToolChain.Intel.ModelSim.Configuration method), 413
- LogVerbose() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- LogVerbose() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
- LogVerbose() (ToolChain.Intel.Quartus.Configuration method), 420
- LogVerbose() (ToolChain.Intel.Quartus.Map method), 422
- LogVerbose() (ToolChain.Lattice.Configuration method), 434
- LogVerbose() (ToolChain.Lattice.Diamond.Configuration method), 427
- LogVerbose() (ToolChain.Lattice.Diamond.Synth method), 429
- LogVerbose() (ToolChain.Lattice.Synplify.Configuration method), 431
- LogVerbose() (ToolChain.Mentor.Configuration method), 459
- LogVerbose() (ToolChain.Mentor.ModelSim.Configuration method), 459

- method), 438
- LogVerbose() (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method), 273
- method), 440
- LogVerbose() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 275
- method), 443
- LogVerbose() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 277
- method), 445
- LogVerbose() (ToolChain.Mentor.ModelSim.Selector method), 278
- method), 446
- LogVerbose() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 333
- method), 451
- LogVerbose() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 334
- method), 447
- LogVerbose() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 336
- method), 454
- LogVerbose() (ToolChain.Mentor.QuestaSim.Configuration method), 337
- method), 456
- LogVerbose() (ToolChain.PoC.Configuration method), 339
- 461
- LogVerbose() (ToolChain.Synopsys.Configuration method), 340
- method), 463
- LogVerbose() (ToolChain.ToolConfiguration method), 346
- 494
- LogVerbose() (ToolChain.ToolSelector method), 342
- method), 496
- LogVerbose() (ToolChain.VendorConfiguration method), 342
- method), 492
- LogVerbose() (ToolChain.Windows.Cmd method), 350
- 465
- LogVerbose() (ToolChain.Xilinx.Configuration method), 353
- method), 486
- LogVerbose() (ToolChain.Xilinx.ISE.Configuration method), 351
- method), 468
- LogVerbose() (ToolChain.Xilinx.ISE.CoreGenerator method), 354
- method), 473
- LogVerbose() (ToolChain.Xilinx.ISE.Fuse method), 363
- 470
- LogVerbose() (ToolChain.Xilinx.ISE.ISE Simulator method), 356
- method), 471
- LogVerbose() (ToolChain.Xilinx.ISE.Xst method), 359
- 472
- LogVerbose() (ToolChain.Xilinx.Vivado.Configuration method), 359
- method), 477
- LogVerbose() (ToolChain.Xilinx.Vivado.Synth method), 358
- method), 482
- LogVerbose() (ToolChain.Xilinx.Vivado.XElab method), 361
- method), 480
- LogVerbose() (ToolChain.Xilinx.Vivado.XSim method), 380
- method), 481
- LogWarning() (Base.Executable.Executable method), 368
- 255
- LogWarning() (Base.IHost method), 370
- 267
- LogWarning() (Base.Logging.ILogable method), 370
- 257
- LogWarning() (Base.Shared.Shared method), 365
- 266
- LogWarning() (Compiler.Compiler method), 374
- 283
- LogWarning() (Compiler.ISECompiler.Compiler method), 376
- method), 268
- LogWarning() (Compiler.LSECompiler.Compiler method), 376
- method), 269
- LogWarning() (Compiler.QuartusCompiler.Compiler method), 376
- method), 271
- LogWarning() (Compiler.VivadoCompiler.Compiler method), 273
- LogWarning() (Compiler.XCICompiler.Compiler method), 275
- LogWarning() (Compiler.XCOCompiler.Compiler method), 277
- LogWarning() (Compiler.XSTCompiler.Compiler method), 278
- LogWarning() (Simulator.ActiveHDL Simulator.Simulator method), 333
- LogWarning() (Simulator.Cocotb Simulator.Simulator method), 334
- LogWarning() (Simulator.GHDL Simulator.Simulator method), 336
- LogWarning() (Simulator.ISE Simulator.Simulator method), 337
- LogWarning() (Simulator.ModelSim Simulator.Simulator method), 339
- LogWarning() (Simulator.Questa Simulator.Simulator method), 340
- LogWarning() (Simulator.Simulator method), 346
- LogWarning() (Simulator.Vivado Simulator.Simulator method), 342
- LogWarning() (ToolChain.Aldec.ActiveHDL.Configuration method), 350
- LogWarning() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- LogWarning() (ToolChain.Aldec.ActiveHDL.VHDL Library Tool method), 351
- LogWarning() (ToolChain.Aldec.ActiveHDL.VHDL Standalone Simulator method), 354
- LogWarning() (ToolChain.Aldec.Configuration method), 363
- LogWarning() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- LogWarning() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359
- LogWarning() (ToolChain.Aldec.RivieraPRO.VHDL Library Tool method), 358
- LogWarning() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361
- LogWarning() (ToolChain.Altera.Configuration method), 380
- LogWarning() (ToolChain.Altera.ModelSim.AltaraEdition Configuration method), 368
- LogWarning() (ToolChain.Altera.ModelSim.Altara Starter Edition Configuration method), 370
- LogWarning() (ToolChain.Altera.ModelSim.Configuration method), 365
- LogWarning() (ToolChain.Altera.Quartus.Configuration method), 374
- LogWarning() (ToolChain.Altera.Quartus.Map method), 376
- LogWarning() (ToolChain.Altera.Quartus.TclShell method), 376
- LogWarning() (ToolChain.Configuration method), 490

- LogWarning() (ToolChain.Configurator method), [497](#)
- LogWarning() (ToolChain.GHDL.Configuration method), [383](#)
- LogWarning() (ToolChain.GHDL.GHDL method), [387](#)
- LogWarning() (ToolChain.GHDL.GHDLAnalyze method), [389](#)
- LogWarning() (ToolChain.GHDL.GHDLElaborate method), [392](#)
- LogWarning() (ToolChain.GHDL.GHDLRun method), [395](#)
- LogWarning() (ToolChain.Git.Configuration method), [404](#)
- LogWarning() (ToolChain.Git.GitConfig method), [410](#)
- LogWarning() (ToolChain.Git.GitDescribe method), [409](#)
- LogWarning() (ToolChain.Git.GitRevList method), [408](#)
- LogWarning() (ToolChain.Git.GitRevParse method), [407](#)
- LogWarning() (ToolChain.Git.GitSCM method), [406](#)
- LogWarning() (ToolChain.GNU.Make method), [398](#)
- LogWarning() (ToolChain.GTKWave.Configuration method), [400](#)
- LogWarning() (ToolChain.GTKWave.GTKWave method), [402](#)
- LogWarning() (ToolChain.Intel.Configuration method), [424](#)
- LogWarning() (ToolChain.Intel.ModelSim.Configuration method), [413](#)
- LogWarning() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), [415](#)
- LogWarning() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), [418](#)
- LogWarning() (ToolChain.Intel.Quartus.Configuration method), [420](#)
- LogWarning() (ToolChain.Intel.Quartus.Map method), [422](#)
- LogWarning() (ToolChain.Lattice.Configuration method), [434](#)
- LogWarning() (ToolChain.Lattice.Diamond.Configuration method), [427](#)
- LogWarning() (ToolChain.Lattice.Diamond.Synth method), [429](#)
- LogWarning() (ToolChain.Lattice.Synplify.Configuration method), [431](#)
- LogWarning() (ToolChain.Mentor.Configuration method), [459](#)
- LogWarning() (ToolChain.Mentor.ModelSim.Configuration method), [438](#)
- LogWarning() (ToolChain.Mentor.ModelSim.ModelSimPEMConfiguration method), [440](#)
- LogWarning() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), [443](#)
- LogWarning() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), [445](#)
- LogWarning() (ToolChain.Mentor.ModelSim.Selector method), [446](#)
- LogWarning() (ToolChain.Mentor.ModelSim.VHDLCompiler method), [451](#)
- LogWarning() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), [447](#)
- LogWarning() (ToolChain.Mentor.ModelSim.VHDL Simulator method), [454](#)
- LogWarning() (ToolChain.Mentor.QuestaSim.Configuration method), [456](#)
- LogWarning() (ToolChain.PoC.Configuration method), [461](#)
- LogWarning() (ToolChain.Synopsys.Configuration method), [463](#)
- LogWarning() (ToolChain.ToolConfiguration method), [494](#)
- LogWarning() (ToolChain.ToolSelector method), [496](#)
- LogWarning() (ToolChain.VendorConfiguration method), [492](#)
- LogWarning() (ToolChain.Windows.Cmd method), [465](#)
- LogWarning() (ToolChain.Xilinx.Configuration method), [486](#)
- LogWarning() (ToolChain.Xilinx.ISE.Configuration method), [468](#)
- LogWarning() (ToolChain.Xilinx.ISE.CoreGenerator method), [473](#)
- LogWarning() (ToolChain.Xilinx.ISE.Fuse method), [470](#)
- LogWarning() (ToolChain.Xilinx.ISE.ISE Simulator method), [471](#)
- LogWarning() (ToolChain.Xilinx.ISE.Xst method), [472](#)
- LogWarning() (ToolChain.Xilinx.Vivado.Configuration method), [477](#)
- LogWarning() (ToolChain.Xilinx.Vivado.Synth method), [482](#)
- LogWarning() (ToolChain.Xilinx.Vivado.XELab method), [480](#)
- LogWarning() (ToolChain.Xilinx.Vivado.XSim method), [481](#)
- LongCommandArgument (class in Base.Executable), [246](#)
- LongFlagArgument (class in Base.Executable), [248](#)
- LongTupleArgument (class in Base.Executable), [253](#)
- LongValuedFlagArgument (class in Base.Executable), [250](#)
- LongValuedFlagListArgument (class in Base.Executable), [252](#)
- LS (DataBase.Config.AlteraSubTypes attribute), [288](#)
- LX (DataBase.Config.XilinxSubTypes attribute), [289](#)
- LXT (DataBase.Config.XilinxSubTypes attribute), [289](#)

M

- M (DataBase.Config.Packages attribute), [290](#)
- MachXO (DataBase.Config.LatticeDevices attribute), [287](#)
- MachXO2 (DataBase.Config.LatticeDevices attribute), [287](#)
- MachXO3 (DataBase.Config.LatticeDevices attribute), [287](#)
- MainParser (lib.pyAttribute.ArgParseAttributes.ArgParseMixin attribute), [519](#)
- Make (class in ToolChain.GNU), [397](#)

- Make.Executable (class in ToolChain.GNU), 397
- Make.SwitchGui (class in ToolChain.GNU), 397
- Map (class in ToolChain.Altera.Quartus), 375
- Map (class in ToolChain.Intel.Quartus), 422
- Map.ArgProjectName (class in ToolChain.Altera.Quartus), 375
- Map.ArgProjectName (class in ToolChain.Intel.Quartus), 422
- Map.Executable (class in ToolChain.Altera.Quartus), 375
- Map.Executable (class in ToolChain.Intel.Quartus), 422
- Map.SwitchArgumentFile (class in ToolChain.Altera.Quartus), 375
- Map.SwitchArgumentFile (class in ToolChain.Intel.Quartus), 422
- Map.SwitchDeviceFamily (class in ToolChain.Altera.Quartus), 375
- Map.SwitchDeviceFamily (class in ToolChain.Intel.Quartus), 422
- Map.SwitchDevicePart (class in ToolChain.Altera.Quartus), 375
- Map.SwitchDevicePart (class in ToolChain.Intel.Quartus), 423
- MapFilter() (in module ToolChain.Altera.Quartus), 378
- MapFilter() (in module ToolChain.Lattice.Diamond), 430
- MatchingParserResult, 513
- Max (DataBase.Config.AlteraFamilies attribute), 285
- Max10 (DataBase.Config.AlteraDevices attribute), 286
- Max2 (DataBase.Config.AlteraDevices attribute), 286
- Max4 (DataBase.Config.AlteraDevices attribute), 286
- Max5 (DataBase.Config.AlteraDevices attribute), 286
- Mentor_ModelSim (Base.Project.ToolChain attribute), 259
- Mentor_QuestaSim (Base.Project.ToolChain attribute), 259
- Mentor_vSim (Base.Project.Tool attribute), 260
- MentorException, 457
- MentorModelSimPEEditions (class in ToolChain.Mentor.ModelSim), 436
- merge() (in module lib.Functions), 512
- merge_with() (in module lib.Functions), 512
- Message (Base.Logging.LogEntry attribute), 256
- Message (Parser.FilesCodeDOM.ReportStatement attribute), 322
- MethodAlias (class in lib.Decorators), 506
- MicroSemi (DataBase.Config.Vendors attribute), 284
- MismatchingParserResult, 512
- ModelSim (class in ToolChain.Mentor.ModelSim), 447
- ModelSimAlteraEdition (ToolChain.Altera.ModelSim.AlteraModelSimEditions attribute), 364
- ModelSimAlteraEdition (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimAlteraStarterEdition (ToolChain.Altera.ModelSim.AlteraModelSimEditions attribute), 364
- ModelSimAlteraStarterEdition (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimDE (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimEditions (class in ToolChain.Mentor.ModelSim), 436
- ModelSimException, 364, 411, 436
- ModelSimIntelEdition (ToolChain.Intel.ModelSim.IntelModelSimEditions attribute), 411
- ModelSimIntelEdition (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimIntelStarterEdition (ToolChain.Intel.ModelSim.IntelModelSimEditions attribute), 411
- ModelSimIntelStarterEdition (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimPE (ToolChain.Mentor.ModelSim.MentorModelSimPEEditions attribute), 436
- ModelSimPE (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimPEConfiguration (class in ToolChain.Mentor.ModelSim), 439
- ModelSimPEEducation (ToolChain.Mentor.ModelSim.MentorModelSimPEEditions attribute), 436
- ModelSimSE32 (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimSE32Configuration (class in ToolChain.Mentor.ModelSim), 441
- ModelSimSE64 (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436
- ModelSimSE64Configuration (class in ToolChain.Mentor.ModelSim), 444
- ModuleName (DataBase.Entity.CocoTestbench attribute), 301
- ModuleName (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- ModuleName (DataBase.Entity.LatticeNetlist attribute), 303
- ModuleName (DataBase.Entity.Netlist attribute), 301
- ModuleName (DataBase.Entity.QuartusNetlist attribute), 303
- ModuleName (DataBase.Entity.Testbench attribute), 300
- ModuleName (DataBase.Entity.VHDLTestbench attribute), 300
- ModuleName (DataBase.Entity.VivadoNetlist attribute), 304
- ModuleName (DataBase.Entity.XstNetlist attribute), 302
- mro() (Base.Executable.CommandArgument method), 246
- mro() (Base.Executable.CommandLineArgument method), 245
- mro() (Base.Executable.ExecutableArgument method), 245

mro() (Base.Executable.FlagArgument method), 248
mro() (Base.Executable.LongCommandArgument method), 246
mro() (Base.Executable.LongFlagArgument method), 249
mro() (Base.Executable.LongTupleArgument method), 254
mro() (Base.Executable.LongValuedFlagArgument method), 250
mro() (Base.Executable.LongValuedFlagListArgument method), 252
mro() (Base.Executable.NamedCommandLineArgument method), 245
mro() (Base.Executable.PathArgument method), 248
mro() (Base.Executable.ShortCommandArgument method), 246
mro() (Base.Executable.ShortFlagArgument method), 248
mro() (Base.Executable.ShortTupleArgument method), 253
mro() (Base.Executable.ShortValuedFlagArgument method), 250
mro() (Base.Executable.ShortValuedFlagListArgument method), 252
mro() (Base.Executable.StringArgument method), 247
mro() (Base.Executable.StringListArgument method), 247
mro() (Base.Executable.TupleArgument method), 253
mro() (Base.Executable.ValuedFlagArgument method), 250
mro() (Base.Executable.ValuedFlagListArgument method), 251
mro() (Base.Executable.WindowsCommandArgument method), 247
mro() (Base.Executable.WindowsFlagArgument method), 249
mro() (Base.Executable.WindowsTupleArgument method), 254
mro() (Base.Executable.WindowsValuedFlagArgument method), 251
mro() (Base.Executable.WindowsValuedFlagListArgument method), 252
mro() (lib.CodeDOM.CodeDOMMeta method), 498
MultiLine (Parser.RulesCodeDOM.ReplaceStatement attribute), 328

N

Name (Base.Executable.CommandArgument attribute), 246
Name (Base.Executable.FlagArgument attribute), 248
Name (Base.Executable.LongCommandArgument attribute), 246
Name (Base.Executable.LongFlagArgument attribute), 249
Name (Base.Executable.LongTupleArgument attribute), 253
Name (Base.Executable.LongValuedFlagArgument attribute), 250
Name (Base.Executable.LongValuedFlagListArgument attribute), 252
Name (Base.Executable.NamedCommandLineArgument attribute), 245
Name (Base.Executable.ShortCommandArgument attribute), 246
Name (Base.Executable.ShortFlagArgument attribute), 248
Name (Base.Executable.ShortTupleArgument attribute), 253
Name (Base.Executable.ShortValuedFlagArgument attribute), 250
Name (Base.Executable.ShortValuedFlagListArgument attribute), 251
Name (Base.Executable.TupleArgument attribute), 253
Name (Base.Executable.ValuedFlagArgument attribute), 250
Name (Base.Executable.ValuedFlagListArgument attribute), 251
Name (Base.Executable.WindowsCommandArgument attribute), 247
Name (Base.Executable.WindowsFlagArgument attribute), 249
Name (Base.Executable.WindowsTupleArgument attribute), 254
Name (Base.Executable.WindowsValuedFlagArgument attribute), 251
Name (Base.Executable.WindowsValuedFlagListArgument attribute), 252
Name (Base.Project.FileSet attribute), 261
name (Base.Project.FileTypes attribute), 258
Name (Base.Project.Project attribute), 260
Name (Base.Project.VHDLLibrary attribute), 261
Name (DataBase.Config.Board attribute), 290
Name (DataBase.Config.Device attribute), 290
Name (DataBase.Entity.AskWildCard attribute), 298
name (DataBase.Entity.BaseFlags attribute), 292
Name (DataBase.Entity.CocoTestbench attribute), 301
Name (DataBase.Entity.CoreGeneratorNetlist attribute), 304
Name (DataBase.Entity.IPCore attribute), 299
Name (DataBase.Entity.LatticeNetlist attribute), 303
Name (DataBase.Entity.LazyPathElement attribute), 299
Name (DataBase.Entity.Library attribute), 296
Name (DataBase.Entity.Namespace attribute), 295
Name (DataBase.Entity.Netlist attribute), 302
name (DataBase.Entity.NetlistKind attribute), 294
Name (DataBase.Entity.PathElement attribute), 294
Name (DataBase.Entity.QuartusNetlist attribute), 303
Name (DataBase.Entity.StarWildCard attribute), 297
Name (DataBase.Entity.Testbench attribute), 300
name (DataBase.Entity.TestbenchKind attribute), 293
Name (DataBase.Entity.VHDLTestbench attribute), 300
Name (DataBase.Entity.VivadoNetlist attribute), 304
Name (DataBase.Entity.WildCard attribute), 297
Name (DataBase.Entity.XstNetlist attribute), 302
Name (DataBase.Solution.ISEProject attribute), 307

- Name (DataBase.Solution.LatticeProject attribute), 308
 Name (DataBase.Solution.Project attribute), 307
 Name (DataBase.Solution.QuartusProject attribute), 308
 Name (DataBase.Solution.Solution attribute), 306
 Name (DataBase.Solution.VirtualProject attribute), 309
 Name (DataBase.Solution.VivadoProject attribute), 307
 Name (DataBase.TestCase.ElementBase attribute), 312
 Name (DataBase.TestCase.GroupBase attribute), 312
 Name (DataBase.TestCase.Synthesis attribute), 315
 Name (DataBase.TestCase.SynthesisGroup attribute), 313
 Name (DataBase.TestCase.SynthesisSuite attribute), 314
 Name (DataBase.TestCase.TestBase attribute), 315
 Name (DataBase.TestCase.TestCase attribute), 315
 Name (DataBase.TestCase.TestGroup attribute), 312
 Name (DataBase.TestCase.TestSuite attribute), 313
 Name (lib.CodeDOM.Identifier attribute), 504
 name (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 507
 Name (Parser.FilesParser.VHDLLibraryReference attribute), 326
 name (Simulator.SimulationSteps attribute), 344
 Name (ToolChain.Altera.Quartus.QuartusProject attribute), 377
 Name (ToolChain.EditionDescription attribute), 495
 Name (ToolChain.Xilinx.ISE.ISEProject attribute), 474
 Name (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
 NamedCommandLineArgument (class in Base.Executable), 245
 Namespace (class in DataBase.Entity), 295
 NamespaceNames (DataBase.Entity.Library attribute), 296
 NamespaceNames (DataBase.Entity.Namespace attribute), 295
 NamespaceRoot (class in DataBase.Entity), 294
 Namespaces (DataBase.Entity.Library attribute), 296
 Namespaces (DataBase.Entity.Namespace attribute), 295
 Netlist (class in DataBase.Entity), 301
 NetList (DataBase.Entity.EntityTypes attribute), 291
 Netlist (DataBase.TestCase.Synthesis attribute), 315
 NetlistKind (class in DataBase.Entity), 293
 Netlists (DataBase.Entity.AskWildCard attribute), 298
 Netlists (DataBase.Entity.StarWildCard attribute), 297
 Netlists (DataBase.Entity.WildCard attribute), 296
 NoAsserts (Simulator.SimulationResult attribute), 345
 NoAssertsCount (DataBase.TestCase.TestGroup attribute), 312
 NoAssertsCount (DataBase.TestCase.TestSuite attribute), 313
 NoCleanUp (Compiler.Compiler attribute), 282
 NoCleanUp (Compiler.ISECompiler.Compiler attribute), 268
 NoCleanUp (Compiler.LSECompiler.Compiler attribute), 270
 NoCleanUp (Compiler.QuartusCompiler.Compiler attribute), 271
 NoCleanUp (Compiler.VivadoCompiler.Compiler attribute), 273
 NoCleanUp (Compiler.XCICompiler.Compiler attribute), 275
 NoCleanUp (Compiler.XCOCompiler.Compiler attribute), 277
 NoCleanUp (Compiler.XSTCompiler.Compiler attribute), 278
 NONSPACECRE (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508
 Normal (Base.Logging.Severity attribute), 256
 NoSubType (DataBase.Config.AlteraSubTypes attribute), 288
 NoSubType (DataBase.Config.GenericSubTypes attribute), 288
 NoSubType (DataBase.Config.LatticeSubTypes attribute), 288
 NoSubType (DataBase.Config.XilinxSubTypes attribute), 288
 NotConfiguredException, 242
 NotExpression (class in lib.CodeDOM), 499
 NotInExpression (class in lib.CodeDOM), 502
 NotRun (Compiler.CompileResult attribute), 282
 NotRun (Simulator.SimulationResult attribute), 345
 Number (DataBase.Config.Device attribute), 290
 NumberChars (lib.Parser.Tokenizer.TokenKind attribute), 515
 NumberToken (class in lib.Parser), 515
- ## O
- Open() (Base.Project.CocotbSourceFile method), 265
 Open() (Base.Project.ConstraintFile method), 262
 Open() (Base.Project.File method), 262
 Open() (Base.Project.ProjectFile method), 262
 Open() (Base.Project.PythonSourceFile method), 264
 Open() (Base.Project.SettingsFile method), 263
 Open() (Base.Project.SourceFile method), 263
 Open() (Base.Project.VerilogSourceFile method), 264
 Open() (Base.Project.VHDLSourceFile method), 264
 Open() (DataBase.Solution.FileListFile method), 309
 Open() (DataBase.Solution.RulesFile method), 310
 Open() (ToolChain.Altera.Quartus.QuartusProject method), 377
 Open() (ToolChain.Altera.Quartus.QuartusProjectFile method), 378
 Open() (ToolChain.Altera.Quartus.QuartusSettings method), 378
 Open() (ToolChain.Lattice.Diamond.SynthesisArgumentFile method), 429
 Open() (ToolChain.Lattice.LatticeDesignConstraintFile method), 435
 Open() (ToolChain.Synopsys.SynopsysDesignConstraintFile method), 464
 Open() (ToolChain.Xilinx.ISE.ISEProjectFile method), 474

Open() (ToolChain.Xilinx.ISE.UserConstraintFile method), 475	Parameters (ToolChain.Altera.Quartus.TclShell attribute), 376
Open() (ToolChain.Xilinx.Vivado.VivadoProjectFile method), 484	Parameters (ToolChain.GHDL.GHDL attribute), 386
Open() (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile method), 484	Parameters (ToolChain.GHDL.GHDLAnalyze attribute), 389
OPTCRE (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508	Parameters (ToolChain.GHDL.GHDLElaborate attribute), 392
OPTCRE_NV (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 508	Parameters (ToolChain.GHDL.GHDLRun attribute), 395
Optimize (Simulator.SimulationState attribute), 345	Parameters (ToolChain.Git.GitConfig attribute), 410
OptionName (Parser.FilesCodeDOM.InterpolateLiteral attribute), 321	Parameters (ToolChain.Git.GitDescribe attribute), 409
options() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509	Parameters (ToolChain.Git.GitRevList attribute), 408
optionxform() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509	Parameters (ToolChain.Git.GitRevParse attribute), 407
OrExpression (class in lib.CodeDOM), 502	Parameters (ToolChain.Git.GitSCM attribute), 405
OSVVM	Parameters (ToolChain.GNU.Make attribute), 397
Pre-compilation, 54	Parameters (ToolChain.GTKWave.GTKWave attribute), 401
Third-Party Libraries, 203	Parameters (ToolChain.Intel.Quartus.Map attribute), 422
OtherChars (lib.Parser.Tokenizer.TokenKind attribute), 515	Parameters (ToolChain.Lattice.Diamond.Synth attribute), 428
OverallRunTime (DataBase.TestCase.SuiteMixIn attribute), 313	Parameters (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 450
OverallRunTime (DataBase.TestCase.Synthesis attribute), 315	Parameters (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 447
OverallRunTime (DataBase.TestCase.SynthesisSuite attribute), 314	Parameters (ToolChain.Mentor.ModelSim.VHDL Simulator attribute), 453
OverallRunTime (DataBase.TestCase.TestBase attribute), 314	Parameters (ToolChain.Windows.Cmd attribute), 465
OverallRunTime (DataBase.TestCase.TestCase attribute), 315	Parameters (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473
OverallRunTime (DataBase.TestCase.TestSuite attribute), 313	Parameters (ToolChain.Xilinx.ISE.Fuse attribute), 469
	Parameters (ToolChain.Xilinx.ISE.ISESimulator attribute), 471
P	Parameters (ToolChain.Xilinx.ISE.Xst attribute), 472
Package (DataBase.Config.Device attribute), 290	Parameters (ToolChain.Xilinx.Vivado.Synth attribute), 482
Package (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429	Parameters (ToolChain.Xilinx.Vivado.XElab attribute), 479
Packages (class in DataBase.Config), 289	Parameters (ToolChain.Xilinx.Vivado.XSim attribute), 481
Parameters (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 352	Parent (DataBase.Entity.AskWildCard attribute), 298
Parameters (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 351	Parent (DataBase.Entity.CocoTestbench attribute), 301
Parameters (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 353	Parent (DataBase.Entity.CoreGeneratorNetlist attribute), 304
Parameters (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 359	Parent (DataBase.Entity.IPCore attribute), 299
Parameters (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358	Parent (DataBase.Entity.LatticeNetlist attribute), 303
Parameters (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 360	Parent (DataBase.Entity.LazyPathElement attribute), 299
Parameters (ToolChain.Altera.Quartus.Map attribute), 375	Parent (DataBase.Entity.Library attribute), 296
Parameters (ToolChain.Altera.Quartus.QuartusSettings attribute), 378	Parent (DataBase.Entity.Namespace attribute), 295
	Parent (DataBase.Entity.Netlist attribute), 302
	Parent (DataBase.Entity.PathElement attribute), 294
	Parent (DataBase.Entity.QuartusNetlist attribute), 303
	Parent (DataBase.Entity.StarWildCard attribute), 297
	Parent (DataBase.Entity.Testbench attribute), 300
	Parent (DataBase.Entity.VHDLTestbench attribute), 300
	Parent (DataBase.Entity.VivadoNetlist attribute), 304

- Parent (DataBase.Entity.WildCard attribute), [297](#)
- Parent (DataBase.Entity.XstNetlist attribute), [302](#)
- Parent (DataBase.Solution.Base attribute), [305](#)
- Parent (DataBase.Solution.ISEProject attribute), [307](#)
- Parent (DataBase.Solution.LatticeProject attribute), [308](#)
- Parent (DataBase.Solution.Project attribute), [307](#)
- Parent (DataBase.Solution.QuartusProject attribute), [308](#)
- Parent (DataBase.Solution.Repository attribute), [306](#)
- Parent (DataBase.Solution.Solution attribute), [306](#)
- Parent (DataBase.Solution.VivadoProject attribute), [307](#)
- Parent (DataBase.TestCase.ElementBase attribute), [312](#)
- Parent (DataBase.TestCase.GroupBase attribute), [312](#)
- Parent (DataBase.TestCase.Synthesis attribute), [315](#)
- Parent (DataBase.TestCase.SynthesisGroup attribute), [313](#)
- Parent (DataBase.TestCase.SynthesisSuite attribute), [314](#)
- Parent (DataBase.TestCase.TestBase attribute), [314](#)
- Parent (DataBase.TestCase.TestCase attribute), [315](#)
- Parent (DataBase.TestCase.TestGroup attribute), [312](#)
- Parent (DataBase.TestCase.TestSuite attribute), [314](#)
- Parse() (Base.Project.VHDLSourceFile method), [263](#)
- Parse() (DataBase.Solution.FileListFile method), [309](#)
- Parse() (DataBase.Solution.RulesFile method), [310](#)
- Parse() (lib.CodeDOM.AndExpression class method), [502](#)
- Parse() (lib.CodeDOM.BinaryExpression class method), [499](#)
- Parse() (lib.CodeDOM.BlockedStatement class method), [505](#)
- Parse() (lib.CodeDOM.BlockStatement class method), [504](#)
- parse() (lib.CodeDOM.CodeDOMMeta method), [498](#)
- Parse() (lib.CodeDOM.CodeDOMObject class method), [499](#)
- Parse() (lib.CodeDOM.CommentLine class method), [505](#)
- Parse() (lib.CodeDOM.CompareExpression class method), [500](#)
- Parse() (lib.CodeDOM.ConditionalBlockStatement class method), [504](#)
- Parse() (lib.CodeDOM.EmptyLine class method), [505](#)
- Parse() (lib.CodeDOM.EqualExpression class method), [500](#)
- Parse() (lib.CodeDOM.Expression class method), [499](#)
- Parse() (lib.CodeDOM.ExpressionChoice class method), [505](#)
- Parse() (lib.CodeDOM.Function class method), [503](#)
- Parse() (lib.CodeDOM.GreaterThanEqualExpression class method), [501](#)
- Parse() (lib.CodeDOM.GreaterThanExpression class method), [501](#)
- Parse() (lib.CodeDOM.Identifier class method), [504](#)
- Parse() (lib.CodeDOM.InExpression class method), [502](#)
- Parse() (lib.CodeDOM.IntegerLiteral class method), [504](#)
- Parse() (lib.CodeDOM.LessThanEqualExpression class method), [501](#)
- Parse() (lib.CodeDOM.LessThanExpression class method), [501](#)
- Parse() (lib.CodeDOM.ListElement class method), [503](#)
- Parse() (lib.CodeDOM.Literal class method), [503](#)
- Parse() (lib.CodeDOM.LogicalExpression class method), [500](#)
- Parse() (lib.CodeDOM.NotExpression class method), [499](#)
- Parse() (lib.CodeDOM.NotInExpression class method), [503](#)
- Parse() (lib.CodeDOM.OrExpression class method), [502](#)
- Parse() (lib.CodeDOM.Statement class method), [504](#)
- Parse() (lib.CodeDOM.StringLiteral class method), [503](#)
- Parse() (lib.CodeDOM.UnaryExpression class method), [499](#)
- Parse() (lib.CodeDOM.UnequalExpression class method), [500](#)
- Parse() (lib.CodeDOM.XorExpression class method), [502](#)
- Parse() (Parser.FilesCodeDOM.BlockedStatement class method), [317](#)
- Parse() (Parser.FilesCodeDOM.CocotbStatement class method), [320](#)
- Parse() (Parser.FilesCodeDOM.ConcatenateExpression class method), [319](#)
- Parse() (Parser.FilesCodeDOM.ConstraintStatement class method), [320](#)
- Parse() (Parser.FilesCodeDOM.Document class method), [323](#)
- Parse() (Parser.FilesCodeDOM.ElseIfStatement class method), [323](#)
- Parse() (Parser.FilesCodeDOM.ElseStatement class method), [323](#)
- Parse() (Parser.FilesCodeDOM.ExistsFunction class method), [319](#)
- Parse() (Parser.FilesCodeDOM.IfElseIfElseStatement class method), [323](#)
- Parse() (Parser.FilesCodeDOM.IfStatement class method), [322](#)
- Parse() (Parser.FilesCodeDOM.IfThenElseExpressions class method), [317](#)
- Parse() (Parser.FilesCodeDOM.IncludeStatement class method), [322](#)
- Parse() (Parser.FilesCodeDOM.InterpolateLiteral class method), [321](#)
- Parse() (Parser.FilesCodeDOM.LDCStatement class method), [320](#)
- Parse() (Parser.FilesCodeDOM.LibraryStatement class method), [322](#)
- Parse() (Parser.FilesCodeDOM.ListConstructorExpression class method), [318](#)
- Parse() (Parser.FilesCodeDOM.ListElementExpressions class method), [318](#)

Parse() (Parser.FilesCodeDOM.PathExpressions class method), 318	Path (Base.Project.CocothSourceFile attribute), 265
Parse() (Parser.FilesCodeDOM.PathStatement class method), 321	Path (Base.Project.ConstraintFile attribute), 262
Parse() (Parser.FilesCodeDOM.ReportStatement class method), 322	Path (Base.Project.File attribute), 262
Parse() (Parser.FilesCodeDOM.SDCStatement class method), 320	Path (Base.Project.ProjectFile attribute), 262
Parse() (Parser.FilesCodeDOM.SubDirectoryExpression class method), 318	Path (Base.Project.PythonSourceFile attribute), 264
Parse() (Parser.FilesCodeDOM.UCFStatement class method), 321	Path (Base.Project.SettingsFile attribute), 263
Parse() (Parser.FilesCodeDOM.VerilogStatement class method), 319	Path (Base.Project.SourceFile attribute), 263
Parse() (Parser.FilesCodeDOM.VHDLStatement class method), 319	Path (Base.Project.VerilogSourceFile attribute), 264
Parse() (Parser.FilesCodeDOM.XDCStatement class method), 321	Path (Base.Project.VHDLSourceFile attribute), 264
Parse() (Parser.RulesCodeDOM.AppendLineStatement class method), 329	Path (DataBase.Entity.AskWildCard attribute), 298
Parse() (Parser.RulesCodeDOM.CopyStatement class method), 328	Path (DataBase.Entity.CocoTestbench attribute), 301
Parse() (Parser.RulesCodeDOM.DeleteStatement class method), 328	Path (DataBase.Entity.CoreGeneratorNetlist attribute), 304
Parse() (Parser.RulesCodeDOM.Document class method), 330	Path (DataBase.Entity.IPCore attribute), 299
Parse() (Parser.RulesCodeDOM.DocumentStatements class method), 328	Path (DataBase.Entity.LatticeNetlist attribute), 303
Parse() (Parser.RulesCodeDOM.FileStatement class method), 329	Path (DataBase.Entity.LazyPathElement attribute), 299
Parse() (Parser.RulesCodeDOM.InFileStatements class method), 327	Path (DataBase.Entity.Library attribute), 296
Parse() (Parser.RulesCodeDOM.PostProcessRulesStatement class method), 330	Path (DataBase.Entity.Namespace attribute), 295
Parse() (Parser.RulesCodeDOM.PostProcessStatements class method), 327	Path (DataBase.Entity.Netlist attribute), 302
Parse() (Parser.RulesCodeDOM.PreProcessRulesStatement class method), 329	Path (DataBase.Entity.PathElement attribute), 295
Parse() (Parser.RulesCodeDOM.PreProcessStatements class method), 327	Path (DataBase.Entity.QuartusNetlist attribute), 303
Parse() (Parser.RulesCodeDOM.ProcessRulesBlockStatement class method), 329	Path (DataBase.Entity.StarWildCard attribute), 297
Parse() (Parser.RulesCodeDOM.ReplaceStatement class method), 328	Path (DataBase.Entity.Testbench attribute), 300
parser (lib.ExtendedConfigParser.ExtendedSectionProxy attribute), 507	Path (DataBase.Entity.VHDLTestbench attribute), 301
Parser (module), 316	Path (DataBase.Entity.VivadoNetlist attribute), 304
Parser.FilesCodeDOM (module), 316	Path (DataBase.Entity.WildCard attribute), 297
Parser.FilesParser (module), 324	Path (DataBase.Entity.XstNetlist attribute), 302
Parser.RulesCodeDOM (module), 326	Path (DataBase.Solution.FileListFile attribute), 309
Parser.RulesParser (module), 330	Path (DataBase.Solution.RulesFile attribute), 310
ParserException, 512	Path (DataBase.Solution.Solution attribute), 306
Passed (Simulator.SimulationResult attribute), 345	Path (Parser.FilesParser.VHDLLibraryReference attribute), 326
PassedCount (DataBase.TestCase.TestGroup attribute), 312	Path (ToolChain.Aldec.ActiveHDL.VHDLCompiler attribute), 353
PassedCount (DataBase.TestCase.TestSuite attribute), 314	Path (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool attribute), 352
Path (Base.Executable.Executable attribute), 255	Path (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator attribute), 354
	Path (ToolChain.Aldec.RivieraPRO.VHDLCompiler attribute), 359
	Path (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool attribute), 358
	Path (ToolChain.Aldec.RivieraPRO.VHDL Simulator attribute), 361
	Path (ToolChain.Altera.Quartus.Map attribute), 376
	Path (ToolChain.Altera.Quartus.QuartusProjectFile attribute), 378
	Path (ToolChain.Altera.Quartus.QuartusSettings attribute), 378
	Path (ToolChain.Altera.Quartus.TclShell attribute), 376
	Path (ToolChain.GHDL.GHDL attribute), 387
	Path (ToolChain.GHDL.GHDLAnalyze attribute), 389
	Path (ToolChain.GHDL.GHDL Elaborate attribute), 392
	Path (ToolChain.GHDL.GHDLRun attribute), 395
	Path (ToolChain.Git.GitConfig attribute), 410
	Path (ToolChain.Git.GitDescribe attribute), 409
	Path (ToolChain.Git.GitRevList attribute), 408

- Path (ToolChain.Git.GitRevParse attribute), 407
- Path (ToolChain.Git.GitSCM attribute), 406
- Path (ToolChain.GNU.Make attribute), 398
- Path (ToolChain.GTKWave.GTKWave attribute), 402
- Path (ToolChain.Intel.Quartus.Map attribute), 422
- Path (ToolChain.Lattice.Diamond.Synth attribute), 429
- Path (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
- Path (ToolChain.Lattice.LatticeDesignConstraintFile attribute), 435
- Path (ToolChain.Mentor.ModelSim.VHDLCompiler attribute), 451
- Path (ToolChain.Mentor.ModelSim.VHDLLibraryTool attribute), 448
- Path (ToolChain.Mentor.ModelSim.VHDL Simulator attribute), 454
- Path (ToolChain.Synopsys.SynopsysDesignConstraintFile attribute), 464
- Path (ToolChain.Windows.Cmd attribute), 465
- Path (ToolChain.Xilinx.ISE.CoreGenerator attribute), 473
- Path (ToolChain.Xilinx.ISE.Fuse attribute), 470
- Path (ToolChain.Xilinx.ISE.ISEProjectFile attribute), 474
- Path (ToolChain.Xilinx.ISE.ISESimulator attribute), 471
- Path (ToolChain.Xilinx.ISE.UserConstraintFile attribute), 475
- Path (ToolChain.Xilinx.ISE.Xst attribute), 472
- Path (ToolChain.Xilinx.Vivado.Synth attribute), 482
- Path (ToolChain.Xilinx.Vivado.VivadoProjectFile attribute), 484
- Path (ToolChain.Xilinx.Vivado.XElab attribute), 480
- Path (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile attribute), 484
- Path (ToolChain.Xilinx.Vivado.XSim attribute), 481
- PathArgument (class in Base.Executable), 247
- PathElement (class in DataBase.Entity), 294
- PathExpression (Parser.FilesCodeDOM.CocotbStatement attribute), 320
- PathExpression (Parser.FilesCodeDOM.ConstraintStatement attribute), 320
- PathExpression (Parser.FilesCodeDOM.IncludeStatement attribute), 322
- PathExpression (Parser.FilesCodeDOM.LDCStatement attribute), 320
- PathExpression (Parser.FilesCodeDOM.LibraryStatement attribute), 322
- PathExpression (Parser.FilesCodeDOM.PathStatement attribute), 321
- PathExpression (Parser.FilesCodeDOM.SDCStatement attribute), 320
- PathExpression (Parser.FilesCodeDOM.UCFStatement attribute), 321
- PathExpression (Parser.FilesCodeDOM.VerilogStatement attribute), 319
- PathExpression (Parser.FilesCodeDOM.VHDLStatement attribute), 319
- PathExpression (Parser.FilesCodeDOM.XDCStatement attribute), 321
- PathExpressions (class in Parser.FilesCodeDOM), 318
- PathStatement (class in Parser.FilesCodeDOM), 321
- PinCount (DataBase.Config.Device attribute), 290
- Platform (Base.IHost attribute), 266
- Platform (DataBase.Query attribute), 316
- PlatformNotSupportedException, 242
- PoC (module), 237
- poc.ps1 command line option
 - D, 219
- poc.sh command line option
 - D, 220
- PoCConfig (Base.IHost attribute), 266
- PoCConfig (DataBase.Query attribute), 316
- PoCProject (Base.Shared.Shared attribute), 265
- PoCProject (Compiler.Compiler attribute), 283
- PoCProject (Compiler.ISECompiler.Compiler attribute), 268
- PoCProject (Compiler.LSECompiler.Compiler attribute), 270
- PoCProject (Compiler.QuartusCompiler.Compiler attribute), 271
- PoCProject (Compiler.VivadoCompiler.Compiler attribute), 273
- PoCProject (Compiler.XCICompiler.Compiler attribute), 275
- PoCProject (Compiler.XCOCCompiler.Compiler attribute), 277
- PoCProject (Compiler.XSTCompiler.Compiler attribute), 278
- PoCProject (Simulator.ActiveHDL Simulator.Simulator attribute), 333
- PoCProject (Simulator.Cocotb Simulator.Simulator attribute), 334
- PoCProject (Simulator.GHDL Simulator.Simulator attribute), 336
- PoCProject (Simulator.ISESimulator.Simulator attribute), 337
- PoCProject (Simulator.ModelSim Simulator.Simulator attribute), 339
- PoCProject (Simulator.Questa Simulator.Simulator attribute), 340
- PoCProject (Simulator.Simulator attribute), 346
- PoCProject (Simulator.Vivado Simulator.Simulator attribute), 342
- PoCRootDirectory, 220
- PoCSimulationResultFilter() (in module Simulator), 347
- PoCSimulationResultNotFoundException, 343
- pop() (Base.Executable.CommandLineArgumentList method), 254
- pop() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509
- pop() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
- popitem() (lib.ExtendedConfigParser.ExtendedConfigParser method), 509

popitem() (lib.ExtendedConfigParser.ExtendedSectionPro
method), 507

PostCopy (Compiler.CompileState attribute), 281

PostDelete (Compiler.CompileState attribute), 281

PostPatch (Compiler.CompileState attribute), 281

PostProcessRules (DataBase.Solution.RulesFile at-
tribute), 310

PostProcessRules (Parser.RulesParser.RulesParserMixIn
attribute), 332

PostProcessRulesStatement (class in
Parser.RulesCodeDOM), 330

PostProcessStatements (class in
Parser.RulesCodeDOM), 327

pprint() (Base.Project.Project method), 261

pprint() (DataBase.Entity.CocoTestbench method), 301

pprint() (DataBase.Entity.CoreGeneratorNetlist
method), 304

pprint() (DataBase.Entity.IPCore method), 299

pprint() (DataBase.Entity.LatticeNetlist method), 303

pprint() (DataBase.Entity.Library method), 296

pprint() (DataBase.Entity.Namespace method), 295

pprint() (DataBase.Entity.QuartusNetlist method), 303

pprint() (DataBase.Entity.Testbench method), 300

pprint() (DataBase.Entity.VHDLTestbench method),
300

pprint() (DataBase.Entity.VivadoNetlist method), 304

pprint() (DataBase.Entity.XstNetlist method), 302

pprint() (DataBase.Solution.VirtualProject method),
309

pprint() (ToolChain.Altera.Quartus.QuartusProject
method), 378

pprint() (ToolChain.Xilinx.ISE.ISEProject method),
474

pprint() (ToolChain.Xilinx.Vivado.VivadoProject
method), 483

Pre-compilation, 49

- Altera, 50
- Cocotb, 55
- Lattice, 51
- OSVVM, 54
- Simulator Adapters, 55
- Supported Simulators, 50
- Third-Party Libraries, 53
- UVVM, 54
- Vendor Primitives, 50
- Xilinx ISE, 52
- Xilinx Vivado, 53

PreCopy (Compiler.CompileState attribute), 281

Prepare (Compiler.CompileState attribute), 281

Prepare (Simulator.SimulationState attribute), 345

PrepareOptions() (ToolChain.Aldec.ActiveHDL.Configuration
method), 350

PrepareOptions() (ToolChain.Aldec.Configuration
method), 363

PrepareOptions() (ToolChain.Aldec.RivieraPRO.Configuration
method), 356

PrepareOptions() (ToolChain.Altera.Configuration
method), 380

PrepareOptions() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration
method), 368

PrepareOptions() (ToolChain.Altera.ModelSim.AlteraStarterEditionConf
method), 370

PrepareOptions() (ToolChain.Altera.ModelSim.Configuration
method), 365

PrepareOptions() (ToolChain.Altera.Quartus.Configuration
method), 374

PrepareOptions() (ToolChain.Configuration method),
489

PrepareOptions() (ToolChain.GHDL.Configuration
method), 383

PrepareOptions() (ToolChain.Git.Configuration
method), 404

PrepareOptions() (ToolChain.GTKWave.Configuration
method), 400

PrepareOptions() (ToolChain.Intel.Configuration
method), 424

PrepareOptions() (ToolChain.Intel.ModelSim.Configuration
method), 413

PrepareOptions() (ToolChain.Intel.ModelSim.IntelEditionConfiguration
method), 415

PrepareOptions() (ToolChain.Intel.ModelSim.IntelStarterEditionConfigur
method), 418

PrepareOptions() (ToolChain.Intel.Quartus.Configuration
method), 420

PrepareOptions() (ToolChain.Lattice.Configuration
method), 434

PrepareOptions() (ToolChain.Lattice.Diamond.Configuration
method), 427

PrepareOptions() (ToolChain.Lattice.Synplify.Configuration
method), 431

PrepareOptions() (ToolChain.Mentor.Configuration
method), 459

PrepareOptions() (ToolChain.Mentor.ModelSim.Configuration
method), 438

PrepareOptions() (ToolChain.Mentor.ModelSim.ModelSimPEConfigurati
method), 440

PrepareOptions() (ToolChain.Mentor.ModelSim.ModelSimSE32Configur
method), 443

PrepareOptions() (ToolChain.Mentor.ModelSim.ModelSimSE64Configur
method), 445

PrepareOptions() (ToolChain.Mentor.QuestaSim.Configuration
method), 456

PrepareOptions() (ToolChain.PoC.Configuration
method), 461

PrepareOptions() (ToolChain.Synopsys.Configuration
method), 463

PrepareOptions() (ToolChain.ToolConfiguration
method), 494

PrepareOptions() (ToolChain.VendorConfiguration
method), 492

PrepareOptions() (ToolChain.Xilinx.Configuration
method), 486

PrepareOptions() (ToolChain.Xilinx.ISE.Configuration
method), 468

PrepareOptions() (ToolChain.Xilinx.Vivado.Configuration
method), 477

[PrepareSections\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 350](#)
[PrepareSections\(\) \(ToolChain.Aldec.Configuration method\), 363](#)
[PrepareSections\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 356](#)
[PrepareSections\(\) \(ToolChain.Altera.Configuration method\), 380](#)
[PrepareSections\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method\), 368](#)
[PrepareSections\(\) \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method\), 370](#)
[PrepareSections\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 366](#)
[PrepareSections\(\) \(ToolChain.Altera.Quartus.Configuration method\), 374](#)
[PrepareSections\(\) \(ToolChain.Configuration method\), 489](#)
[PrepareSections\(\) \(ToolChain.GHDL.Configuration method\), 383](#)
[PrepareSections\(\) \(ToolChain.Git.Configuration method\), 404](#)
[PrepareSections\(\) \(ToolChain.GTKWave.Configuration method\), 400](#)
[PrepareSections\(\) \(ToolChain.Intel.Configuration method\), 424](#)
[PrepareSections\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[PrepareSections\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 415](#)
[PrepareSections\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 418](#)
[PrepareSections\(\) \(ToolChain.Intel.Quartus.Configuration method\), 421](#)
[PrepareSections\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[PrepareSections\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 427](#)
[PrepareSections\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 431](#)
[PrepareSections\(\) \(ToolChain.Mentor.Configuration method\), 459](#)
[PrepareSections\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 438](#)
[PrepareSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 440](#)
[PrepareSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 443](#)
[PrepareSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 445](#)
[PrepareSections\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[PrepareSections\(\) \(ToolChain.PoC.Configuration method\), 461](#)
[PrepareSections\(\) \(ToolChain.Synopsys.Configuration method\), 463](#)
[PrepareSections\(\) \(ToolChain.ToolConfiguration method\), 494](#)
[PrepareSections\(\) \(ToolChain.VendorConfiguration method\), 492](#)
[PrepareSections\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[PrepareSections\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 468](#)
[PrepareSections\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)
[PrepareVersionedSections\(\) \(ToolChain.Aldec.ActiveHDL.Configuration method\), 350](#)
[PrepareVersionedSections\(\) \(ToolChain.Aldec.RivieraPRO.Configuration method\), 356](#)
[PrepareVersionedSections\(\) \(ToolChain.Altera.ModelSim.AlteraEditionConfiguration method\), 368](#)
[PrepareVersionedSections\(\) \(ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method\), 370](#)
[PrepareVersionedSections\(\) \(ToolChain.Altera.ModelSim.Configuration method\), 366](#)
[PrepareVersionedSections\(\) \(ToolChain.Altera.Quartus.Configuration method\), 374](#)
[PrepareVersionedSections\(\) \(ToolChain.GHDL.Configuration method\), 383](#)
[PrepareVersionedSections\(\) \(ToolChain.Git.Configuration method\), 404](#)
[PrepareVersionedSections\(\) \(ToolChain.GTKWave.Configuration method\), 400](#)
[PrepareVersionedSections\(\) \(ToolChain.Intel.Configuration method\), 424](#)
[PrepareVersionedSections\(\) \(ToolChain.Intel.ModelSim.Configuration method\), 413](#)
[PrepareVersionedSections\(\) \(ToolChain.Intel.ModelSim.IntelEditionConfiguration method\), 415](#)
[PrepareVersionedSections\(\) \(ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method\), 418](#)
[PrepareVersionedSections\(\) \(ToolChain.Lattice.Configuration method\), 434](#)
[PrepareVersionedSections\(\) \(ToolChain.Lattice.Diamond.Configuration method\), 427](#)
[PrepareVersionedSections\(\) \(ToolChain.Lattice.Synplify.Configuration method\), 431](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.Configuration method\), 459](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.ModelSim.Configuration method\), 438](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimPEConfiguration method\), 440](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method\), 443](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method\), 445](#)
[PrepareVersionedSections\(\) \(ToolChain.Mentor.QuestaSim.Configuration method\), 456](#)
[PrepareVersionedSections\(\) \(ToolChain.PoC.Configuration method\), 461](#)
[PrepareVersionedSections\(\) \(ToolChain.Synopsys.Configuration method\), 463](#)
[PrepareVersionedSections\(\) \(ToolChain.ToolConfiguration method\), 494](#)
[PrepareVersionedSections\(\) \(ToolChain.Xilinx.Configuration method\), 486](#)
[PrepareVersionedSections\(\) \(ToolChain.Xilinx.ISE.Configuration method\), 468](#)
[PrepareVersionedSections\(\) \(ToolChain.Xilinx.Vivado.Configuration method\), 477](#)

method), 440	270	
PrepareVersionedSections() (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration method), 443	PrintCompileReportLine() (Compiler.QuartusCompiler.Compiler method), 271	(Com- method),
PrepareVersionedSections() (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration method), 445	PrintCompileReportLine() (Compiler.VivadoCompiler.Compiler method), 273	(Com- method),
PrepareVersionedSections() (ToolChain.Mentor.QuestaSim.Configuration method), 456	PrintCompileReportLine() (Compiler.XCICompiler.Compiler method), 275	(Com- method),
PrepareVersionedSections() (ToolChain.PoC.Configuration method), 461	PrintCompileReportLine() (Compiler.XCOCCompiler.Compiler method), 277	(Com- method),
PrepareVersionedSections() (ToolChain.ToolConfiguration method), 493	PrintCompileReportLine() (Compiler.XSTCompiler.Compiler method), 278	(Com- method),
PrepareVersionedSections() (ToolChain.Xilinx.ISE.Configuration method), 468	printEnvironmentException() (lib.Functions.Exit class method), 512	
PrepareVersionedSections() (ToolChain.Xilinx.Vivado.Configuration method), 477	printException() (lib.Functions.Exit class method), 511	
PreparseEnvironment() (ToolChain.Lattice.Diamond.Diamond method), 428	printExceptionBase() (lib.Functions.Exit class method), 511	
PreparseEnvironment() (ToolChain.Xilinx.ISE.ISE method), 469	printNotConfiguredException() (lib.Functions.Exit class method), 512	
PreparseEnvironment() (ToolChain.Xilinx.Vivado.Vivado method), 478	printNotImplementedError() (lib.Functions.Exit class method), 511	
PrePatch (Compiler.CompileState attribute), 281	PrintOverallCompileReport() (Compiler.Compiler method), 283	
PreProcessRules (DataBase.Solution.RulesFile at- tribute), 310	PrintOverallCompileReport() (Com- piler.ISECompiler.Compiler method), 268	
PreProcessRules (Parser.RulesParser.RulesParserMixIn attribute), 332	PrintOverallCompileReport() (Com- piler.LSECompiler.Compiler method), 270	
PreProcessRulesStatement (class in Parser.RulesCodeDOM), 329	PrintOverallCompileReport() (Com- piler.QuartusCompiler.Compiler method), 271	
PreProcessStatements (class in Parser.RulesCodeDOM), 327	PrintOverallCompileReport() (Com- piler.VivadoCompiler.Compiler method), 273	
PreviousToken (lib.Parser.CharacterToken attribute), 514	PrintOverallCompileReport() (Com- piler.XCICompiler.Compiler method), 275	
PreviousToken (lib.Parser.DelimiterToken attribute), 515	PrintOverallCompileReport() (Com- piler.XCOCCompiler.Compiler method), 277	
PreviousToken (lib.Parser.NumberToken attribute), 515	PrintOverallCompileReport() (Com- piler.XSTCompiler.Compiler method), 278	
PreviousToken (lib.Parser.SpaceToken attribute), 515	PrintOverallSimulationReport() (Simula- tor.ActiveHDL Simulator.Simulator method), 333	
PreviousToken (lib.Parser.StartOfDocumentToken at- tribute), 514	PrintOverallSimulationReport() (Simula- tor.CocotbSimulator.Simulator method), 334	
PreviousToken (lib.Parser.StringToken attribute), 515	PrintOverallSimulationReport() (Simula- tor.GHDL Simulator.Simulator method), 336	
PreviousToken (lib.Parser.SuperToken attribute), 514	PrintOverallSimulationReport() (Simula- tor.ActiveHDL Simulator.Simulator method), 333	
PreviousToken (lib.Parser.Token attribute), 514	PrintOverallSimulationReport() (Simula- tor.CocotbSimulator.Simulator method), 334	
PreviousToken (lib.Parser.ValuedToken attribute), 514	PrintOverallSimulationReport() (Simula- tor.GHDL Simulator.Simulator method), 336	
PrintCompileReportLine() (Compiler.Compiler method), 283	PrintOverallSimulationReport() (Simula- tor.ActiveHDL Simulator.Simulator method), 333	
PrintCompileReportLine() (Com- piler.ISECompiler.Compiler method), 268	PrintOverallSimulationReport() (Simula- tor.CocotbSimulator.Simulator method), 334	
PrintCompileReportLine() (Com- piler.LSECompiler.Compiler method),	PrintOverallSimulationReport() (Simula- tor.GHDL Simulator.Simulator method), 336	

- tor.ISESimulator.Simulator method), 337
 - PrintOverallSimulationReport() (Simulator.ModelSimSimulator.Simulator method), 339
 - PrintOverallSimulationReport() (Simulator.QuestaSimulator.Simulator method), 340
 - PrintOverallSimulationReport() (Simulator.Simulator method), 346
 - PrintOverallSimulationReport() (Simulator.VivadoSimulator.Simulator method), 342
 - printPlatformNotSupportedException() (lib.Functions.Exit class method), 512
 - PrintSimulationReportLine() (Simulator.ActiveHDLSimulator.Simulator method), 333
 - PrintSimulationReportLine() (Simulator.CocotbSimulator.Simulator method), 334
 - PrintSimulationReportLine() (Simulator.GHDLSimulator.Simulator method), 336
 - PrintSimulationReportLine() (Simulator.ISESimulator.Simulator method), 337
 - PrintSimulationReportLine() (Simulator.ModelSimSimulator.Simulator method), 339
 - PrintSimulationReportLine() (Simulator.QuestaSimulator.Simulator method), 340
 - PrintSimulationReportLine() (Simulator.Simulator method), 346
 - PrintSimulationReportLine() (Simulator.VivadoSimulator.Simulator method), 342
 - Private (DataBase.Entity.Visibility attribute), 294
 - PrjFile (DataBase.Entity.LatticeNetlist attribute), 303
 - PrjFile (DataBase.Entity.XstNetlist attribute), 302
 - ProcessRulesBlockStatement (class in Parser.RulesCodeDOM), 329
 - Project (Base.Project.CocotbSourceFile attribute), 265
 - Project (Base.Project.ConstraintFile attribute), 262
 - Project (Base.Project.File attribute), 262
 - Project (Base.Project.FileSet attribute), 261
 - Project (Base.Project.ProjectFile attribute), 262
 - Project (Base.Project.PythonSourceFile attribute), 264
 - Project (Base.Project.SettingsFile attribute), 263
 - Project (Base.Project.SourceFile attribute), 263
 - Project (Base.Project.VerilogSourceFile attribute), 264
 - Project (Base.Project.VHDLLibrary attribute), 261
 - Project (Base.Project.VHDLSourceFile attribute), 264
 - Project (class in Base.Project), 260
 - Project (class in DataBase.Solution), 306
 - Project (DataBase.Solution.FileListFile attribute), 309
 - Project (DataBase.Solution.RulesFile attribute), 310
 - Project (ToolChain.Altera.Quartus.QuartusProjectFile attribute), 378
 - Project (ToolChain.Altera.Quartus.QuartusSettings attribute), 378
 - Project (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
 - Project (ToolChain.Lattice.LatticeDesignConstraintFile attribute), 435
 - Project (ToolChain.Synopsys.SynopsysDesignConstraintFile attribute), 464
 - Project (ToolChain.Xilinx.ISE.ISEProjectFile attribute), 474
 - Project (ToolChain.Xilinx.ISE.UserConstraintFile attribute), 475
 - Project (ToolChain.Xilinx.Vivado.VivadoProjectFile attribute), 484
 - Project (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile attribute), 484
 - ProjectFile (class in Base.Project), 262
 - ProjectNames (DataBase.Solution.Solution attribute), 306
 - Projects (DataBase.Solution.Solution attribute), 306
 - properties (Base.Project.FileTypes attribute), 258
 - properties (DataBase.Entity.BaseFlags attribute), 292
 - properties (DataBase.Entity.NetlistKind attribute), 294
 - properties (DataBase.Entity.TestbenchKind attribute), 293
 - properties (Simulator.SimulationSteps attribute), 344
 - Public (DataBase.Entity.Visibility attribute), 294
 - PythonSourceFile (class in Base.Project), 264
- ## Q
- Q (DataBase.Config.Packages attribute), 289
 - QsfFile (DataBase.Entity.QuartusNetlist attribute), 302
 - Quartus (class in ToolChain.Altera.Quartus), 375
 - Quartus (class in ToolChain.Intel.Quartus), 421
 - QuartusEditions (class in ToolChain.Altera.Quartus), 372
 - QuartusException, 372, 419
 - QuartusNetlist (class in DataBase.Entity), 302
 - QuartusNetlist (DataBase.Entity.IPCore attribute), 299
 - QuartusNetlists (DataBase.Entity.AskWildCard attribute), 298
 - QuartusNetlists (DataBase.Entity.StarWildCard attribute), 297
 - QuartusNetlists (DataBase.Entity.WildCard attribute), 296
 - QuartusProject (class in DataBase.Solution), 307
 - QuartusProject (class in ToolChain.Altera.Quartus), 377
 - QuartusProjectFile (class in ToolChain.Altera.Quartus), 378
 - QuartusSession (class in ToolChain.Altera.Quartus), 377
 - QuartusSettings (class in ToolChain.Altera.Quartus), 378
 - Query (class in DataBase), 316
 - QueryConfiguration() (DataBase.Query method), 316
 - QuestaSim (ToolChain.Mentor.ModelSim.ModelSimEditions attribute), 436

QuestaSimException, 454

Quiet (Base.Logging.Severity attribute), 256

R

RB (DataBase.Config.Packages attribute), 289

RBG (DataBase.Config.Packages attribute), 289

read() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510

Read() (ToolChain.Altera.Quartus.QuartusProject method), 377

read_dict() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510

read_file() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510

read_string() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510

ReadFile() (Base.Project.CocotbSourceFile method), 265

ReadFile() (Base.Project.ConstraintFile method), 262

ReadFile() (Base.Project.File method), 262

ReadFile() (Base.Project.ProjectFile method), 262

ReadFile() (Base.Project.PythonSourceFile method), 264

ReadFile() (Base.Project.SettingsFile method), 263

ReadFile() (Base.Project.SourceFile method), 263

ReadFile() (Base.Project.VerilogSourceFile method), 264

ReadFile() (Base.Project.VHDLSourceFile method), 264

ReadFile() (DataBase.Solution.FileListFile method), 309

ReadFile() (DataBase.Solution.RulesFile method), 310

ReadFile() (ToolChain.Altera.Quartus.QuartusProjectFile method), 378

ReadFile() (ToolChain.Altera.Quartus.QuartusSettings method), 378

ReadFile() (ToolChain.Lattice.Diamond.SynthesisArgumentFile method), 429

ReadFile() (ToolChain.Lattice.LatticeDesignConstraintFile method), 435

ReadFile() (ToolChain.Synopsys.SynopsysDesignConstraintFile method), 464

ReadFile() (ToolChain.Xilinx.ISE.ISEProjectFile method), 474

ReadFile() (ToolChain.Xilinx.ISE.UserConstraintFile method), 475

ReadFile() (ToolChain.Xilinx.Vivado.VivadoProjectFile method), 484

ReadFile() (ToolChain.Xilinx.Vivado.XilinxDesignConstraintFile method), 484

readfp() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510

ReadUntilBoundary() (Base.Executable.Executable method), 255

ReadUntilBoundary() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353

ReadUntilBoundary() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 352

ReadUntilBoundary() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 354

ReadUntilBoundary() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 359

ReadUntilBoundary() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358

ReadUntilBoundary() (ToolChain.Aldec.RivieraPRO.VHDLSimulator method), 361

ReadUntilBoundary() (ToolChain.Altera.Quartus.Map method), 376

ReadUntilBoundary() (ToolChain.Altera.Quartus.TclShell method), 376

ReadUntilBoundary() (ToolChain.GHDL.GHDL method), 387

ReadUntilBoundary() (ToolChain.GHDL.GHDLAnalyze method), 389

ReadUntilBoundary() (ToolChain.GHDL.GHDLElaborate method), 392

ReadUntilBoundary() (ToolChain.GHDL.GHDLRun method), 395

ReadUntilBoundary() (ToolChain.Git.GitConfig method), 410

ReadUntilBoundary() (ToolChain.Git.GitDescribe method), 409

ReadUntilBoundary() (ToolChain.Git.GitRevList method), 408

ReadUntilBoundary() (ToolChain.Git.GitRevParse method), 407

ReadUntilBoundary() (ToolChain.Git.GitSCM method), 406

ReadUntilBoundary() (ToolChain.GNU.Make method), 398

ReadUntilBoundary() (ToolChain.GTKWave.GTKWave method), 402

ReadUntilBoundary() (ToolChain.Intel.Quartus.Map method), 422

ReadUntilBoundary() (ToolChain.Lattice.Diamond.Synth method), 429

ReadUntilBoundary() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 451

ReadUntilBoundary() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 448

ReadUntilBoundary() (ToolChain.Mentor.ModelSim.VHDLSimulator method), 454

ReadUntilBoundary() (ToolChain.Windows.Cmd method), 465

ReadUntilBoundary() (ToolChain.Xilinx.ISE.CoreGenerator method), 473

ReadUntilBoundary() (ToolChain.Xilinx.ISE.Fuse method), 470

ReadUntilBoundary() (ToolChain.Xilinx.ISE.ISESimulator method), 471

ReadUntilBoundary() (ToolChain.Xilinx.ISE.Xst method), 472

ReadUntilBoundary() (ToolChain.Xilinx.Vivado.Synth method), 482

ReadUntilBoundary() (ToolChain.Xilinx.Vivado.XELab method), 480

- ReadUntilBoundary() (ToolChain.Xilinx.Vivado.XSim method), 481
- RegExpOption_CaseInsensitive (Compiler.ReplaceTask attribute), 281
- RegExpOption_CaseInsensitive (Parser.RulesParser.ReplaceRuleMixIn attribute), 331
- RegExpOption_DotAll (Compiler.ReplaceTask attribute), 281
- RegExpOption_DotAll (Parser.RulesParser.ReplaceRuleMixIn attribute), 331
- RegExpOption_MultiLine (Compiler.ReplaceTask attribute), 281
- RegExpOption_MultiLine (Parser.RulesParser.ReplaceRuleMixIn attribute), 331
- Register() (DataBase.Solution.Solution method), 306
- Relocated() (ToolChain.Configurator method), 497
- remove() (Base.Executable.CommandLineArgumentList method), 254
- remove_option() (lib.ExtendedConfigParser.ExtendedConfigParser attribute), 318
- remove_option() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- remove_section() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- RemoveSolution() (DataBase.Solution.Repository method), 306
- ReplacePattern (Compiler.ReplaceTask attribute), 281
- ReplacePattern (Parser.RulesCodeDOM.ReplaceStatement attribute), 328
- ReplacePattern (Parser.RulesParser.ReplaceRuleMixIn attribute), 331
- ReplaceRuleMixIn (class in Parser.RulesParser), 331
- ReplaceStatement (class in Parser.RulesCodeDOM), 328
- ReplaceTask (class in Compiler), 280
- ReportStatement (class in Parser.FilesCodeDOM), 321
- Repository (class in DataBase.Solution), 305
- Result (DataBase.Entity.CocoTestbench attribute), 301
- Result (DataBase.Entity.Testbench attribute), 300
- Result (DataBase.Entity.VHDLTestbench attribute), 301
- reverse() (Base.Executable.CommandLineArgumentList method), 254
- RevListParameters (ToolChain.Git.GitRevList attribute), 407
- RevParseParameters (ToolChain.Git.GitRevParse attribute), 406
- RF (DataBase.Config.Packages attribute), 289
- RightChild (lib.CodeDOM.AndExpression attribute), 502
- RightChild (lib.CodeDOM.BinaryExpression attribute), 499
- RightChild (lib.CodeDOM.CompareExpression attribute), 500
- RightChild (lib.CodeDOM.EqualExpression attribute), 500
- RightChild (lib.CodeDOM.GreaterThanEqualExpression attribute), 501
- RightChild (lib.CodeDOM.GreaterThanExpression attribute), 501
- RightChild (lib.CodeDOM.InExpression attribute), 502
- RightChild (lib.CodeDOM.LessThanEqualExpression attribute), 501
- RightChild (lib.CodeDOM.LessThanExpression attribute), 501
- RightChild (lib.CodeDOM.LogicalExpression attribute), 500
- RightChild (lib.CodeDOM.NotInExpression attribute), 503
- RightChild (lib.CodeDOM.OrExpression attribute), 502
- RightChild (lib.CodeDOM.UnequalExpression attribute), 500
- RightChild (lib.CodeDOM.XorExpression attribute), 502
- RightChild (Parser.FilesCodeDOM.ConcatenateExpression attribute), 319
- RightChild (Parser.FilesCodeDOM.SubDirectoryExpression attribute), 318
- RivieraPRO (class in ToolChain.Aldec.RivieraPRO), 355
- RivieraPROException, 355
- Root() (DataBase.Entity.FQN method), 305
- RootDirectory (Base.Project.Project attribute), 260
- RootDirectory (DataBase.Solution.VirtualProject attribute), 309
- RootDirectory (ToolChain.Altera.Quartus.QuartusProject attribute), 377
- RootDirectory (ToolChain.Xilinx.ISE.ISEProject attribute), 474
- RootDirectory (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
- RS (DataBase.Config.Packages attribute), 289
- Rule (class in Parser.RulesParser), 330
- RulesFile (class in DataBase.Solution), 310
- RulesFile (DataBase.Entity.CoreGeneratorNetlist attribute), 304
- RulesFile (DataBase.Entity.LatticeNetlist attribute), 303
- RulesFile (DataBase.Entity.Netlist attribute), 301
- RulesFile (DataBase.Entity.QuartusNetlist attribute), 303
- RulesFile (DataBase.Entity.VivadoNetlist attribute), 304
- RulesFile (DataBase.Entity.XstNetlist attribute), 302
- RulesParserMixIn (class in Parser.RulesParser), 331
- Run() (Compiler.Compiler method), 282
- Run() (Compiler.ISECompiler.Compiler method), 267
- Run() (Compiler.LSECompiler.Compiler method), 269
- Run() (Compiler.QuartusCompiler.Compiler method), 271
- Run() (Compiler.VivadoCompiler.Compiler method), 273
- Run() (Compiler.XCICompiler.Compiler method), 274
- Run() (Compiler.XCOCCompiler.Compiler method), 274

- 276
- Run() (Compiler.XSTCompiler.Compiler method), 278
- Run() (lib.pyAttribute.ArgParseAttributes.ArgParseMixin method), 519
- Run() (Simulator.ActiveHDL Simulator.Simulator method), 333
- Run() (Simulator.Cocotb Simulator.Simulator method), 334
- Run() (Simulator.GHDL Simulator.Simulator method), 335
- Run() (Simulator.ISE Simulator.Simulator method), 337
- Run() (Simulator.ModelSim Simulator.Simulator method), 338
- Run() (Simulator.Questa Simulator.Simulator method), 340
- Run() (Simulator.Simulator method), 346
- Run() (Simulator.Vivado Simulator.Simulator method), 342
- Run() (ToolChain.GHDL.GHDLRun method), 394
- RunAll() (Compiler.ISE Compiler.Compiler method), 267
- RunAll() (Compiler.LSE Compiler.Compiler method), 269
- RunAll() (Compiler.Quartus Compiler.Compiler method), 271
- RunAll() (Compiler.Vivado Compiler.Compiler method), 273
- RunAll() (Compiler.XCI Compiler.Compiler method), 274
- RunAll() (Compiler.XCO Compiler.Compiler method), 276
- RunAll() (Compiler.XST Compiler.Compiler method), 278
- RunAll() (Simulator.ActiveHDL Simulator.Simulator method), 333
- RunAll() (Simulator.Cocotb Simulator.Simulator method), 334
- RunAll() (Simulator.GHDL Simulator.Simulator method), 336
- RunAll() (Simulator.ISE Simulator.Simulator method), 337
- RunAll() (Simulator.ModelSim Simulator.Simulator method), 339
- RunAll() (Simulator.Questa Simulator.Simulator method), 340
- RunAll() (Simulator.Simulator method), 346
- RunAll() (Simulator.Vivado Simulator.Simulator method), 342
- RunCocotb() (ToolChain.GNU.Make method), 397
- RunOptions (ToolChain.GHDL.GHDL attribute), 386
- RunOptions (ToolChain.GHDL.GHDLAnalyze attribute), 389
- RunOptions (ToolChain.GHDL.GHDLElaborate attribute), 392
- RunOptions (ToolChain.GHDL.GHDLRun attribute), 395
- RunPostConfigurationTasks() (ToolChain.Aldec.ActiveHDL.Configuration method), 350
- RunPostConfigurationTasks() (ToolChain.Aldec.Configuration method), 363
- RunPostConfigurationTasks() (ToolChain.Aldec.RivieraPRO.Configuration method), 356
- RunPostConfigurationTasks() (ToolChain.Altera.Configuration method), 380
- RunPostConfigurationTasks() (ToolChain.Altera.ModelSim.AlteraEditionConfiguration method), 368
- RunPostConfigurationTasks() (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration method), 370
- RunPostConfigurationTasks() (ToolChain.Altera.ModelSim.Configuration method), 366
- RunPostConfigurationTasks() (ToolChain.Altera.Quartus.Configuration method), 374
- RunPostConfigurationTasks() (ToolChain.Configuration method), 490
- RunPostConfigurationTasks() (ToolChain.GHDL.Configuration method), 383
- RunPostConfigurationTasks() (ToolChain.Git.Configuration method), 403
- RunPostConfigurationTasks() (ToolChain.GTKWave.Configuration method), 400
- RunPostConfigurationTasks() (ToolChain.Intel.Configuration method), 424
- RunPostConfigurationTasks() (ToolChain.Intel.ModelSim.Configuration method), 413
- RunPostConfigurationTasks() (ToolChain.Intel.ModelSim.IntelEditionConfiguration method), 415
- RunPostConfigurationTasks() (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration method), 418
- RunPostConfigurationTasks() (ToolChain.Intel.Quartus.Configuration method), 421
- RunPostConfigurationTasks() (ToolChain.Lattice.Configuration method), 434
- RunPostConfigurationTasks() (ToolChain.Lattice.Diamond.Configuration method), 427
- RunPostConfigurationTasks() (ToolChain.Lattice.Synplify.Configuration method), 431
- RunPostConfigurationTasks()

S

- tribute), 486
- SectionName (ToolChain.Xilinx.ISE.Configuration attribute), 468
- SectionName (ToolChain.Xilinx.Vivado.Configuration attribute), 477
- sections() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- Select() (ToolChain.Mentor.ModelSim.Selector method), 446
- Selector (class in ToolChain.Mentor.ModelSim), 446
- Send() (Base.Executable.Executable method), 255
- Send() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- Send() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 352
- Send() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 354
- Send() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 360
- Send() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- Send() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361
- Send() (ToolChain.Altera.Quartus.Map method), 376
- Send() (ToolChain.Altera.Quartus.TclShell method), 377
- Send() (ToolChain.GHDL.GHDL method), 387
- Send() (ToolChain.GHDL.GHDLAnalyze method), 389
- Send() (ToolChain.GHDL.GHDL Elaborate method), 392
- Send() (ToolChain.GHDL.GHDLRun method), 395
- Send() (ToolChain.Git.GitConfig method), 410
- Send() (ToolChain.Git.GitDescribe method), 409
- Send() (ToolChain.Git.GitRevList method), 408
- Send() (ToolChain.Git.GitRevParse method), 407
- Send() (ToolChain.Git.GitSCM method), 406
- Send() (ToolChain.GNU.Make method), 398
- Send() (ToolChain.GTKWave.GTKWave method), 402
- Send() (ToolChain.Intel.Quartus.Map method), 422
- Send() (ToolChain.Lattice.Diamond.Synth method), 429
- Send() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 451
- Send() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 448
- Send() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454
- Send() (ToolChain.Windows.Cmd method), 465
- Send() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- Send() (ToolChain.Xilinx.ISE.Fuse method), 470
- Send() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- Send() (ToolChain.Xilinx.ISE.Xst method), 472
- Send() (ToolChain.Xilinx.Vivado.Synth method), 482
- Send() (ToolChain.Xilinx.Vivado.XElab method), 480
- Send() (ToolChain.Xilinx.Vivado.XSim method), 481
- SendBoundary() (Base.Executable.Executable method), 255
- SendBoundary() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- SendBoundary() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 352
- SendBoundary() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 354
- SendBoundary() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 360
- SendBoundary() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- SendBoundary() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361
- SendBoundary() (ToolChain.Altera.Quartus.Map method), 376
- SendBoundary() (ToolChain.Altera.Quartus.TclShell method), 377
- SendBoundary() (ToolChain.GHDL.GHDL method), 387
- SendBoundary() (ToolChain.GHDL.GHDLAnalyze method), 389
- SendBoundary() (ToolChain.GHDL.GHDL Elaborate method), 392
- SendBoundary() (ToolChain.GHDL.GHDLRun method), 395
- SendBoundary() (ToolChain.Git.GitConfig method), 410
- SendBoundary() (ToolChain.Git.GitDescribe method), 409
- SendBoundary() (ToolChain.Git.GitRevList method), 408
- SendBoundary() (ToolChain.Git.GitRevParse method), 407
- SendBoundary() (ToolChain.Git.GitSCM method), 406
- SendBoundary() (ToolChain.GNU.Make method), 398
- SendBoundary() (ToolChain.GTKWave.GTKWave method), 402
- SendBoundary() (ToolChain.Intel.Quartus.Map method), 422
- SendBoundary() (ToolChain.Lattice.Diamond.Synth method), 429
- SendBoundary() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 451
- SendBoundary() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 448
- SendBoundary() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454
- SendBoundary() (ToolChain.Windows.Cmd method), 465
- SendBoundary() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- SendBoundary() (ToolChain.Xilinx.ISE.Fuse method), 470
- SendBoundary() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- SendBoundary() (ToolChain.Xilinx.ISE.Xst method), 472

- SendBoundary() (ToolChain.Xilinx.Vivado.Synth method), 482
- SendBoundary() (ToolChain.Xilinx.Vivado.XElab method), 480
- SendBoundary() (ToolChain.Xilinx.Vivado.XSim method), 481
- Series (DataBase.Config.Device attribute), 290
- set() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- setdefault() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- setdefault() (lib.ExtendedConfigParser.ExtendedSectionParser method), 507
- SettingsFile (class in Base.Project), 263
- Severity (Base.Logging.LogEntry attribute), 256
- Severity (class in Base.Logging), 256
- Shared (class in Base.Shared), 265
- ShortCommandArgument (class in Base.Executable), 246
- ShortFlagArgument (class in Base.Executable), 248
- ShortName (DataBase.Config.Device attribute), 290
- ShortTupleArgument (class in Base.Executable), 253
- ShortValuedFlagArgument (class in Base.Executable), 250
- ShortValuedFlagListArgument (class in Base.Executable), 251
- Simulate (Simulator.SimulationState attribute), 345
- Simulate() (ToolChain.Aldec.ActiveHDL.VHDLStandaloneSimulator method), 353
- Simulate() (ToolChain.Aldec.RivieraPRO.VHDLStandaloneSimulator method), 360
- Simulate() (ToolChain.Mentor.ModelSim.VHDLStandaloneSimulator method), 453
- Simulate() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- Simulate() (ToolChain.Xilinx.Vivado.XSim method), 481
- Simulation (Base.Project.Environment attribute), 259
- SimulationError (DataBase.TestCase.SimulationStatus attribute), 311
- SimulationFailed (DataBase.TestCase.SimulationStatus attribute), 311
- SimulationGUIRun (DataBase.TestCase.SimulationStatus attribute), 311
- SimulationNoAsserts (DataBase.TestCase.SimulationStatus attribute), 311
- SimulationResult (class in Simulator), 345
- SimulationState (class in Simulator), 344
- SimulationStatus (class in DataBase.TestCase), 311
- SimulationSteps (class in Simulator), 344
- SimulationSuccess (DataBase.TestCase.SimulationStatus attribute), 311
- Simulator (class in Simulator), 345
- Simulator (class in Simulator.ActiveHDLStandaloneSimulator), 332
- Simulator (class in Simulator.CocotbSimulator), 333
- Simulator (class in Simulator.GHDLStandaloneSimulator), 335
- Simulator (class in Simulator.ISESimulator), 337
- Simulator (class in Simulator.ModelSimSimulator), 338
- Simulator (class in Simulator.QuestaSimulator), 340
- Simulator (class in Simulator.VivadoSimulator), 341
- Simulator (module), 332
- Simulator Adapters
- Pre-compilation, 55
- Simulator.ActiveHDLStandaloneSimulator (module), 332
- Simulator.CocotbSimulator (module), 333
- Simulator.GHDLStandaloneSimulator (module), 335
- Simulator.ISESimulator (module), 337
- Simulator.ModelSimSimulator (module), 338
- Simulator.QuestaSimulator (module), 340
- Simulator.RivieraPROStandaloneSimulator (module), 341
- Simulator.VivadoSimulator (module), 341
- SimulatorException, 343
- SimulatorFilter() (in module ToolChain.Xilinx.ISE), 475
- SimulatorFilter() (in module ToolChain.Xilinx.Vivado), 484
- SkipableCommonException, 243
- SkipableCompilerException, 280
- SkipableException, 243
- SkipableSimulatorException, 343
- SkipConfigurationException, 488
- Solution (class in DataBase.Solution), 306
- SolutionNames (DataBase.Solution.Repository attribute), 306
- Solutions (DataBase.Solution.Repository attribute), 306
- sort() (Base.Executable.CommandLineArgumentList method), 254
- Source (DataBase.Entity.EntityTypes attribute), 291
- SourceCodePosition (class in lib.Parser), 513
- SourceFile (class in Base.Project), 263
- SourcePath (Compiler.CopyTask attribute), 280
- SourcePath (Parser.RulesCodeDOM.CopyStatement attribute), 328
- SourcePath (Parser.RulesParser.CopyRuleMixIn attribute), 331
- SpaceChars (lib.Parser.Tokenizer.TokenKind attribute), 515
- SpaceToken (class in lib.Parser), 514
- Spartan (DataBase.Config.XilinxFamilies attribute), 285
- Spartan3 (DataBase.Config.XilinxDevices attribute), 287
- Spartan6 (DataBase.Config.XilinxDevices attribute), 287
- SpeedGrade (DataBase.Config.Device attribute), 290
- SpeedGrade (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 429
- ST (DataBase.Config.AlteraSubTypes attribute), 288
- StandardEdition (ToolChain.Aldec.ActiveHDL.ActiveHDLEditions attribute), 348
- StandardEdition (ToolChain.Aldec.ActiveHDL.AldecActiveHDLEditions attribute), 348
- StartOfDocumentToken (class in lib.Parser), 514

- StartProcess() (Base.Executable.Executable method), 255
- StartProcess() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353
- StartProcess() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 352
- StartProcess() (ToolChain.Aldec.ActiveHDL.VHDLStandardsSimulator method), 354
- StartProcess() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 360
- StartProcess() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358
- StartProcess() (ToolChain.Aldec.RivieraPRO.VHDLSimulation method), 361
- StartProcess() (ToolChain.Altera.Quartus.Map method), 376
- StartProcess() (ToolChain.Altera.Quartus.TclShell method), 377
- StartProcess() (ToolChain.GHDL.GHDL method), 387
- StartProcess() (ToolChain.GHDL.GHDLAnalyze method), 389
- StartProcess() (ToolChain.GHDL.GHDLElaborate method), 392
- StartProcess() (ToolChain.GHDL.GHDLRun method), 395
- StartProcess() (ToolChain.Git.GitConfig method), 410
- StartProcess() (ToolChain.Git.GitDescribe method), 409
- StartProcess() (ToolChain.Git.GitRevList method), 408
- StartProcess() (ToolChain.Git.GitRevParse method), 407
- StartProcess() (ToolChain.Git.GitSCM method), 406
- StartProcess() (ToolChain.GNU.Make method), 398
- StartProcess() (ToolChain.GTKWave.GTKWave method), 402
- StartProcess() (ToolChain.Intel.Quartus.Map method), 422
- StartProcess() (ToolChain.Lattice.Diamond.Synth method), 429
- StartProcess() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 451
- StartProcess() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 448
- StartProcess() (ToolChain.Mentor.ModelSim.VHDLSimulation method), 454
- StartProcess() (ToolChain.Windows.Cmd method), 465
- StartProcess() (ToolChain.Xilinx.ISE.CoreGenerator method), 473
- StartProcess() (ToolChain.Xilinx.ISE.Fuse method), 470
- StartProcess() (ToolChain.Xilinx.ISE.ISESimulator method), 471
- StartProcess() (ToolChain.Xilinx.ISE.Xst method), 472
- StartProcess() (ToolChain.Xilinx.Vivado.Synth method), 483
- StartProcess() (ToolChain.Xilinx.Vivado.XElab method), 480
- StartProcess() (ToolChain.Xilinx.Vivado.XSim method), 481
- StartTime (DataBase.TestCase.SuiteMixIn attribute), 313
- StartTime (DataBase.TestCase.SynthesisSuite attribute), 314
- StartTime (DataBase.TestCase.TestSuite attribute), 314
- StartTimer() (DataBase.TestCase.SuiteMixIn method), 313
- StartTimer() (DataBase.TestCase.Synthesis method), 315
- StartTimer() (DataBase.TestCase.SynthesisSuite method), 314
- StartTimer() (DataBase.TestCase.TestBase method), 314
- StartTimer() (DataBase.TestCase.TestCase method), 315
- StartTimer() (DataBase.TestCase.TestSuite method), 314
- StarWildCard (class in DataBase.Entity), 297
- State (ToolChain.Aldec.ActiveHDL.Configuration attribute), 350
- State (ToolChain.Aldec.Configuration attribute), 363
- State (ToolChain.Aldec.RivieraPRO.Configuration attribute), 356
- State (ToolChain.Altera.Configuration attribute), 380
- State (ToolChain.Altera.ModelSim.AlteraEditionConfiguration attribute), 368
- State (ToolChain.Altera.ModelSim.AlteraStarterEditionConfiguration attribute), 371
- State (ToolChain.Altera.ModelSim.Configuration attribute), 366
- State (ToolChain.Altera.Quartus.Configuration attribute), 374
- State (ToolChain.Configuration attribute), 489
- State (ToolChain.GHDL.Configuration attribute), 383
- State (ToolChain.Git.Configuration attribute), 404
- State (ToolChain.GTKWave.Configuration attribute), 400
- State (ToolChain.Intel.Configuration attribute), 425
- State (ToolChain.Intel.ModelSim.Configuration attribute), 413
- State (ToolChain.Intel.ModelSim.IntelEditionConfiguration attribute), 415
- State (ToolChain.Intel.ModelSim.IntelStarterEditionConfiguration attribute), 418
- State (ToolChain.Intel.Quartus.Configuration attribute), 421
- State (ToolChain.Lattice.Configuration attribute), 434
- State (ToolChain.Lattice.Diamond.Configuration attribute), 427
- State (ToolChain.Lattice.Synplify.Configuration attribute), 431
- State (ToolChain.Mentor.Configuration attribute), 459
- State (ToolChain.Mentor.ModelSim.Configuration attribute), 438
- State (ToolChain.Mentor.ModelSim.ModelSimPEConfiguration attribute), 440
- State (ToolChain.Mentor.ModelSim.ModelSimSE32Configuration attribute), 440

- attribute), 443
- State (ToolChain.Mentor.ModelSim.ModelSimSE64Configuration attribute), 445
- State (ToolChain.Mentor.QuestaSim.Configuration attribute), 456
- State (ToolChain.PoC.Configuration attribute), 461
- State (ToolChain.Synopsys.Configuration attribute), 463
- State (ToolChain.ToolConfiguration attribute), 494
- State (ToolChain.VendorConfiguration attribute), 492
- State (ToolChain.Xilinx.Configuration attribute), 486
- State (ToolChain.Xilinx.ISE.Configuration attribute), 468
- State (ToolChain.Xilinx.Vivado.Configuration attribute), 477
- Statement (class in lib.CodeDOM), 504
- Statements (lib.CodeDOM.BlockStatement attribute), 504
- Statements (lib.CodeDOM.ConditionalBlockStatement attribute), 504
- Statements (Parser.FilesCodeDOM.Document attribute), 323
- Statements (Parser.FilesCodeDOM.ElseIfStatement attribute), 323
- Statements (Parser.FilesCodeDOM.ElseStatement attribute), 323
- Statements (Parser.FilesCodeDOM.IfStatement attribute), 322
- Statements (Parser.RulesCodeDOM.Document attribute), 330
- Statements (Parser.RulesCodeDOM.FileStatement attribute), 329
- Statements (Parser.RulesCodeDOM.PostProcessRulesStatement attribute), 330
- Statements (Parser.RulesCodeDOM.PreProcessRulesStatement attribute), 329
- Statements (Parser.RulesCodeDOM.ProcessRulesBlockStatement attribute), 329
- Status (DataBase.TestCase.Synthesis attribute), 315
- Status (DataBase.TestCase.TestBase attribute), 314
- Status (DataBase.TestCase.TestCase attribute), 315
- StopTimer() (DataBase.TestCase.SuiteMixIn method), 313
- StopTimer() (DataBase.TestCase.Synthesis method), 315
- StopTimer() (DataBase.TestCase.SynthesisSuite method), 314
- StopTimer() (DataBase.TestCase.TestBase method), 314
- StopTimer() (DataBase.TestCase.TestCase method), 315
- StopTimer() (DataBase.TestCase.TestSuite method), 314
- Stratix (DataBase.Config.AlteraFamilies attribute), 285
- Stratix10 (DataBase.Config.AlteraDevices attribute), 286
- Stratix2 (DataBase.Config.AlteraDevices attribute), 286
- Stratix4 (DataBase.Config.AlteraDevices attribute), 286
- Stratix5 (DataBase.Config.AlteraDevices attribute), 286
- StringArgument (class in Base.Executable), 247
- StringListArgument (class in Base.Executable), 247
- StringLiteral (class in lib.CodeDOM), 503
- StringToken (class in lib.Parser), 515
- StudentEdition (ToolChain.Aldec.ActiveHDL.AldecActiveHDLVersions attribute), 348
- SubDirectoryExpression (class in Parser.FilesCodeDOM), 318
- SubParsers (lib.pyAttribute.ArgParseAttributes.ArgParseMixin attribute), 519
- SubTypes (class in DataBase.Config), 287
- Success (Compiler.CompileResult attribute), 282
- SuccessCount (DataBase.TestCase.SynthesisGroup attribute), 313
- SuccessCount (DataBase.TestCase.SynthesisSuite attribute), 314
- SuiteMixIn (class in DataBase.TestCase), 313
- SuperToken (class in lib.Parser), 514
- Supported Simulators
 - Pre-compilation, 50
- SwitchArgumentAttribute (class in lib.pyAttribute.ArgParseAttributes), 518
- SX (DataBase.Config.AlteraSubTypes attribute), 288
- SXT (DataBase.Config.XilinxSubTypes attribute), 289
- SynopsysDesignConstraintFile (class in ToolChain.Synopsys), 464
- SynopsysException, 462
- SynplifyException, 430
- Synth (class in ToolChain.Lattice.Diamond), 428
- Synth (class in ToolChain.Xilinx.Vivado), 481
- Synth.Executable (class in ToolChain.Lattice.Diamond), 428
- Synth.Executable (class in ToolChain.Xilinx.Vivado), 482
- Synth.SwitchLogFile (class in ToolChain.Xilinx.Vivado), 482
- Synth.SwitchMode (class in ToolChain.Xilinx.Vivado), 482
- Synth.SwitchProjectFile (class in ToolChain.Lattice.Diamond), 428
- Synth.SwitchSourceFile (class in ToolChain.Xilinx.Vivado), 482
- Synthesis (Base.Project.Environment attribute), 259
- Synthesis (class in DataBase.TestCase), 315
- SynthesisArgumentFile (class in ToolChain.Lattice.Diamond), 429
- Synthesises (DataBase.TestCase.SynthesisGroup attribute), 313
- Synthesises (DataBase.TestCase.SynthesisSuite attribute), 314
- SynthesisGroup (class in DataBase.TestCase), 312
- SynthesisSuite (class in DataBase.TestCase), 314
- SystemError (DataBase.TestCase.CompileStatus attribute), 311

SystemError (DataBase.TestCase.SimulationStatus attribute), 311

T

T (DataBase.Config.XilinxSubTypes attribute), 289

T_SORTNET_IMPL (C type), 188

TclFile (DataBase.Entity.VivadoNetlist attribute), 304

TclShell (class in ToolChain.Altera.Quartus), 376

TclShell.Executable (class in ToolChain.Altera.Quartus), 376

TclShell.SwitchShell (class in ToolChain.Altera.Quartus), 376

Terminate() (Base.Executable.Executable method), 255

Terminate() (ToolChain.Aldec.ActiveHDL.VHDLCompiler method), 353

Terminate() (ToolChain.Aldec.ActiveHDL.VHDLLibraryTool method), 352

Terminate() (ToolChain.Aldec.ActiveHDL.VHDLStandAloneSimulator method), 354

Terminate() (ToolChain.Aldec.RivieraPRO.VHDLCompiler method), 360

Terminate() (ToolChain.Aldec.RivieraPRO.VHDLLibraryTool method), 358

Terminate() (ToolChain.Aldec.RivieraPRO.VHDL Simulator method), 361

Terminate() (ToolChain.Altera.Quartus.Map method), 376

Terminate() (ToolChain.Altera.Quartus.TclShell method), 377

Terminate() (ToolChain.GHDL.GHDL method), 387

Terminate() (ToolChain.GHDL.GHDLAnalyze method), 390

Terminate() (ToolChain.GHDL.GHDL Elaborate method), 393

Terminate() (ToolChain.GHDL.GHDLRun method), 396

Terminate() (ToolChain.Git.GitConfig method), 410

Terminate() (ToolChain.Git.GitDescribe method), 409

Terminate() (ToolChain.Git.GitRevList method), 408

Terminate() (ToolChain.Git.GitRevParse method), 407

Terminate() (ToolChain.Git.GitSCM method), 406

Terminate() (ToolChain.GNU.Make method), 398

Terminate() (ToolChain.GTKWave.GTKWave method), 402

Terminate() (ToolChain.Intel.Quartus.Map method), 423

Terminate() (ToolChain.Lattice.Diamond.Synth method), 429

Terminate() (ToolChain.Mentor.ModelSim.VHDLCompiler method), 451

Terminate() (ToolChain.Mentor.ModelSim.VHDLLibraryTool method), 448

Terminate() (ToolChain.Mentor.ModelSim.VHDL Simulator method), 454

Terminate() (ToolChain.Windows.Cmd method), 465

Terminate() (ToolChain.Xilinx.ISE.CoreGenerator method), 473

Terminate() (ToolChain.Xilinx.ISE.Fuse method), 470

Terminate() (ToolChain.Xilinx.ISE.ISE Simulator method), 471

Terminate() (ToolChain.Xilinx.ISE.Xst method), 472

Terminate() (ToolChain.Xilinx.Vivado.Synth method), 483

Terminate() (ToolChain.Xilinx.Vivado.XElab method), 480

Terminate() (ToolChain.Xilinx.Vivado.XSim method), 481

TestBase (class in DataBase.TestCase), 314

Testbench (class in DataBase.Entity), 300

Testbench (DataBase.Entity.EntityTypes attribute), 291

Testbench (DataBase.TestCase.TestCase attribute), 315

Testbenches (DataBase.Entity.AskWildCard attribute), 298

Testbenches (DataBase.Entity.StarWildCard attribute), 297

Testbenches (DataBase.Entity.WildCard attribute), 296

TestbenchKind (class in DataBase.Entity), 292

TestCase (class in DataBase.TestCase), 315

TestCases (DataBase.TestCase.TestGroup attribute), 312

TestCases (DataBase.TestCase.TestSuite attribute), 314

TestGroup (class in DataBase.TestCase), 312

TestGroup (DataBase.TestCase.Synthesis attribute), 315

TestGroup (DataBase.TestCase.TestBase attribute), 314

TestGroup (DataBase.TestCase.TestCase attribute), 315

TestSuite (class in DataBase.TestCase), 313

TestSuite (Simulator.ActiveHDL Simulator.Simulator attribute), 333

TestSuite (Simulator.Cocotb Simulator.Simulator attribute), 334

TestSuite (Simulator.GHDL Simulator.Simulator attribute), 336

TestSuite (Simulator.ISE Simulator.Simulator attribute), 337

TestSuite (Simulator.ModelSim Simulator.Simulator attribute), 339

TestSuite (Simulator.Questa Simulator.Simulator attribute), 340

TestSuite (Simulator.Simulator attribute), 345

TestSuite (Simulator.Vivado Simulator.Simulator attribute), 342

Text (lib.CodeDOM.CommentLine attribute), 505

Third-Party Libraries, 202

Cocotb, 203

OSVVM, 203

Pre-compilation, 53

UVVM, 203

VUnit, 204

to_simple_str() (Base.Project.FileTypes method), 259

to_simple_str() (DataBase.Entity.BaseFlags method), 292

to_simple_str() (DataBase.Entity.NetlistKind method), 294

to_simple_str() (DataBase.Entity.TestbenchKind method), 293

- `to_simple_str()` (`Simulator.SimulationSteps` method), 344
- `to_time()` (in module `Base.Shared`), 266
- `ToArgumentList()` (`Base.Executable.CommandLineArgumentList` method), 254
- `Token` (class in `lib.Parser`), 514
- `Tokenizer` (class in `lib.Parser`), 515
- `Tokenizer.TokenKind` (class in `lib.Parser`), 515
- `Tool` (`Base.Project.Project` attribute), 261
- `TOOL` (`Base.Shared.Shared` attribute), 265
- `Tool` (class in `Base.Project`), 259
- `TOOL` (`Compiler.Compiler` attribute), 283
- `TOOL` (`Compiler.ISECompiler.Compiler` attribute), 267
- `TOOL` (`Compiler.LSECompiler.Compiler` attribute), 269
- `TOOL` (`Compiler.QuartusCompiler.Compiler` attribute), 271
- `TOOL` (`Compiler.VivadoCompiler.Compiler` attribute), 272
- `TOOL` (`Compiler.XCICompiler.Compiler` attribute), 274
- `TOOL` (`Compiler.XCOCCompiler.Compiler` attribute), 276
- `TOOL` (`Compiler.XSTCompiler.Compiler` attribute), 278
- `Tool` (`DataBase.Solution.VirtualProject` attribute), 309
- `TOOL` (`Simulator.ActiveHDL Simulator.Simulator` attribute), 332
- `TOOL` (`Simulator.Cocotb Simulator.Simulator` attribute), 334
- `TOOL` (`Simulator.GHDL Simulator.Simulator` attribute), 335
- `TOOL` (`Simulator.ISE Simulator.Simulator` attribute), 337
- `TOOL` (`Simulator.ModelSim Simulator.Simulator` attribute), 338
- `TOOL` (`Simulator.Questa Simulator.Simulator` attribute), 340
- `TOOL` (`Simulator.Simulator` attribute), 346
- `TOOL` (`Simulator.Vivado Simulator.Simulator` attribute), 341
- `Tool` (`ToolChain.Altera.Quartus.QuartusProject` attribute), 378
- `Tool` (`ToolChain.Xilinx.ISE.ISEProject` attribute), 474
- `Tool` (`ToolChain.Xilinx.Vivado.VivadoProject` attribute), 483
- `TOOL_CHAIN` (`Base.Shared.Shared` attribute), 265
- `TOOL_CHAIN` (`Compiler.Compiler` attribute), 283
- `TOOL_CHAIN` (`Compiler.ISECompiler.Compiler` attribute), 267
- `TOOL_CHAIN` (`Compiler.LSECompiler.Compiler` attribute), 269
- `TOOL_CHAIN` (`Compiler.QuartusCompiler.Compiler` attribute), 271
- `TOOL_CHAIN` (`Compiler.VivadoCompiler.Compiler` attribute), 272
- `TOOL_CHAIN` (`Compiler.XCICompiler.Compiler` attribute), 274
- `tribute`), 274
- `TOOL_CHAIN` (`Compiler.XCOCCompiler.Compiler` attribute), 276
- `TOOL_CHAIN` (`Compiler.XSTCompiler.Compiler` attribute), 278
- `TOOL_CHAIN` (`Simulator.ActiveHDL Simulator.Simulator` attribute), 332
- `TOOL_CHAIN` (`Simulator.Cocotb Simulator.Simulator` attribute), 334
- `TOOL_CHAIN` (`Simulator.GHDL Simulator.Simulator` attribute), 335
- `TOOL_CHAIN` (`Simulator.ISE Simulator.Simulator` attribute), 337
- `TOOL_CHAIN` (`Simulator.ModelSim Simulator.Simulator` attribute), 338
- `TOOL_CHAIN` (`Simulator.Questa Simulator.Simulator` attribute), 340
- `TOOL_CHAIN` (`Simulator.Simulator` attribute), 346
- `TOOL_CHAIN` (`Simulator.Vivado Simulator.Simulator` attribute), 341
- `ToolChain` (`Base.Project.Project` attribute), 260
- `ToolChain` (class in `Base.Project`), 259
- `ToolChain` (`DataBase.Solution.VirtualProject` attribute), 309
- `ToolChain` (module), 347
- `ToolChain` (`ToolChain.Altera.Quartus.QuartusProject` attribute), 378
- `ToolChain` (`ToolChain.Xilinx.ISE.ISEProject` attribute), 474
- `ToolChain` (`ToolChain.Xilinx.Vivado.VivadoProject` attribute), 483
- `ToolChain.Aldec` (module), 347
- `ToolChain.Aldec.ActiveHDL` (module), 347
- `ToolChain.Aldec.RivieraPRO` (module), 354
- `ToolChain.Altera` (module), 363
- `ToolChain.Altera.ModelSim` (module), 364
- `ToolChain.Altera.Quartus` (module), 372
- `ToolChain.GHDL` (module), 381
- `ToolChain.Git` (module), 402
- `ToolChain.GNU` (module), 397
- `ToolChain.GTKWave` (module), 398
- `ToolChain.Intel` (module), 411
- `ToolChain.Intel.ModelSim` (module), 411
- `ToolChain.Intel.Quartus` (module), 419
- `ToolChain.Lattice` (module), 425
- `ToolChain.Lattice.ActiveHDL` (module), 425
- `ToolChain.Lattice.Diamond` (module), 425
- `ToolChain.Lattice.Synplify` (module), 430
- `ToolChain.Mentor` (module), 435
- `ToolChain.Mentor.ModelSim` (module), 435
- `ToolChain.Mentor.QuestaSim` (module), 454
- `ToolChain.PoC` (module), 459
- `ToolChain.Synopsys` (module), 462
- `ToolChain.Windows` (module), 464
- `ToolChain.Xilinx` (module), 466
- `ToolChain.Xilinx.ISE` (module), 466

ToolChain.Xilinx.Vivado (module), [475](#)
ToolChainException, [487](#)
ToolConfiguration (class in ToolChain), [492](#)
ToolMixIn (class in ToolChain), [488](#)
ToolMixIn (class in ToolChain.Xilinx.Vivado), [478](#)
ToolName (ToolChain.Mentor.ModelSim.Selector attribute), [446](#)
ToolName (ToolChain.ToolSelector attribute), [495](#)
ToolSelector (class in ToolChain), [495](#)
TopLevel (DataBase.Entity.CocoTestbench attribute), [301](#)
TopLevel (ToolChain.Lattice.Diamond.SynthesisArgument attribute), [429](#)
TQG (DataBase.Config.Packages attribute), [289](#)
TryRun() (Compiler.Compiler method), [282](#)
TryRun() (Compiler.ISECompiler.Compiler method), [268](#)
TryRun() (Compiler.LSECompiler.Compiler method), [270](#)
TryRun() (Compiler.QuartusCompiler.Compiler method), [271](#)
TryRun() (Compiler.VivadoCompiler.Compiler method), [273](#)
TryRun() (Compiler.XCICompiler.Compiler method), [275](#)
TryRun() (Compiler.XCOCompiler.Compiler method), [277](#)
TryRun() (Compiler.XSTCompiler.Compiler method), [278](#)
TryRun() (Simulator.ActiveHDL Simulator.Simulator method), [333](#)
TryRun() (Simulator.Cocotb Simulator.Simulator method), [334](#)
TryRun() (Simulator.GHDL Simulator.Simulator method), [336](#)
TryRun() (Simulator.ISESimulator.Simulator method), [337](#)
TryRun() (Simulator.ModelSim Simulator.Simulator method), [339](#)
TryRun() (Simulator.Questa Simulator.Simulator method), [340](#)
TryRun() (Simulator.Simulator method), [346](#)
TryRun() (Simulator.Vivado Simulator.Simulator method), [342](#)
TryWrite() (Base.Logging.Logger method), [257](#)
TupleArgument (class in Base.Executable), [252](#)
TXT (DataBase.Config.XilinxSubTypes attribute), [289](#)

U

U (DataBase.Config.LatticeSubTypes attribute), [288](#)
U (DataBase.Config.Packages attribute), [289](#)
UCFSourceFileMixIn (class in Parser.FilesParser), [325](#)
UCFStatement (class in Parser.FilesCodeDOM), [320](#)
UM (DataBase.Config.LatticeSubTypes attribute), [288](#)
UnaryExpression (class in lib.CodeDOM), [499](#)
Unchanged (ToolChain.ChangeState attribute), [488](#)
Unconfigured (ToolChain.ConfigurationState attribute), [488](#)

UnequalExpression (class in lib.CodeDOM), [500](#)
Unknown (DataBase.Config.GenericDevices attribute), [286](#)
Unknown (DataBase.Config.GenericFamilies attribute), [285](#)
Unknown (DataBase.Config.GenericSubTypes attribute), [288](#)
Unknown (DataBase.Config.Packages attribute), [289](#)
Unknown (DataBase.Config.Vendors attribute), [284](#)
Unknown (DataBase.Entity.EntityTypes attribute), [291](#)
Unknown (DataBase.Entity.Visibility attribute), [294](#)
Unknown (DataBase.TestCase.CompileStatus attribute), [311](#)
Unknown (DataBase.TestCase.SimulationStatus attribute), [311](#)
Unregister() (DataBase.Solution.Solution method), [306](#)
update() (lib.ExtendedConfigParser.ExtendedConfigParser method), [510](#)
update() (lib.ExtendedConfigParser.ExtendedSectionProxy method), [507](#)
UpdateCache() (lib.ExtendedConfigParser.ExtendedInterpolation method), [507](#)
UpdateConfiguration() (ToolChain.Configurator method), [496](#)
UpdateStatus() (DataBase.TestCase.Synthesis method), [315](#)
UpdateStatus() (DataBase.TestCase.TestCase method), [315](#)
UserConstraintFile (class in ToolChain.Xilinx.ISE), [474](#)
UVVM
 Pre-compilation, [54](#)
 Third-Party Libraries, [203](#)

V

Value (Base.Executable.CommandArgument attribute), [246](#)
Value (Base.Executable.ExecutableArgument attribute), [245](#)
Value (Base.Executable.FlagArgument attribute), [248](#)
Value (Base.Executable.LongCommandArgument attribute), [246](#)
Value (Base.Executable.LongFlagArgument attribute), [249](#)
Value (Base.Executable.LongTupleArgument attribute), [253](#)
Value (Base.Executable.LongValuedFlagArgument attribute), [250](#)
Value (Base.Executable.LongValuedFlagListArgument attribute), [252](#)
Value (Base.Executable.PathArgument attribute), [248](#)
Value (Base.Executable.ShortCommandArgument attribute), [246](#)
Value (Base.Executable.ShortFlagArgument attribute), [248](#)
Value (Base.Executable.ShortTupleArgument attribute), [253](#)

- Value (Base.Executable.ShortValuedFlagArgument attribute), 250
- Value (Base.Executable.ShortValuedFlagListArgument attribute), 251
- Value (Base.Executable.StringArgument attribute), 247
- Value (Base.Executable.StringListArgument attribute), 247
- Value (Base.Executable.TupleArgument attribute), 253
- Value (Base.Executable.ValuedFlagArgument attribute), 249
- Value (Base.Executable.ValuedFlagListArgument attribute), 251
- Value (Base.Executable.WindowsCommandArgument attribute), 247
- Value (Base.Executable.WindowsFlagArgument attribute), 249
- Value (Base.Executable.WindowsTupleArgument attribute), 254
- Value (Base.Executable.WindowsValuedFlagArgument attribute), 251
- Value (Base.Executable.WindowsValuedFlagListArgument attribute), 252
- Value (lib.CodeDOM.IntegerLiteral attribute), 504
- Value (lib.CodeDOM.StringLiteral attribute), 503
- value (lib.Parser.EmptyChoiceParserResult attribute), 513
- value (lib.Parser.GreedyMatchingParserResult attribute), 513
- value (lib.Parser.MatchingParserResult attribute), 513
- value (lib.Parser.MismatchingParserResult attribute), 513
- ValuedFlagArgument (class in Base.Executable), 249
- ValuedFlagListArgument (class in Base.Executable), 251
- ValuedToken (class in lib.Parser), 514
- values() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
- values() (lib.ExtendedConfigParser.ExtendedSectionProxy method), 507
- Variable (Parser.FilesCodeDOM.PathStatement attribute), 321
- VComFilter() (in module ToolChain.Aldec.ActiveHDL), 354
- VComFilter() (in module ToolChain.Aldec.RivieraPRO), 361
- VComFilter() (in module ToolChain.Mentor.ModelSim), 454
- Vendor (DataBase.Config.Device attribute), 290
- Vendor Primitives
 - Pre-compilation, 50
- VendorConfiguration (class in ToolChain), 491
- Vendors (class in DataBase.Config), 284
- Verbose (Base.Logging.Severity attribute), 256
- VerilogSourceFile (class in Base.Project), 264
- VerilogSourceFileMixin (class in Parser.FilesParser), 324
- VerilogStatement (class in Parser.FilesCodeDOM), 319
- Version (ToolChain.GHDL.GHDL attribute), 384
- Version (ToolChain.GHDL.GHDLAnalyze attribute), 390
- Version (ToolChain.GHDL.GHDLElaborate attribute), 393
- Version (ToolChain.GHDL.GHDLRun attribute), 396
- Version (ToolChain.GTKWave.GTKWave attribute), 401
- versionCheck() (lib.Functions.Exit class method), 511
- VhCompFilter() (in module ToolChain.Xilinx.ISE), 475
- VHDL2002 (Base.Project.VHDLVersion attribute), 260
- VHDL2008 (Base.Project.VHDLVersion attribute), 260
- VHDL87 (Base.Project.VHDLVersion attribute), 260
- VHDL93 (Base.Project.VHDLVersion attribute), 260
- VHDL_VERSION (Base.Shared.Shared attribute), 265
- VHDL_VERSION (Compiler.Compiler attribute), 282
- VHDL_VERSION (Compiler.ISECompiler.Compiler attribute), 268
- VHDL_VERSION (Compiler.LSECompiler.Compiler attribute), 270
- VHDL_VERSION (Compiler.QuartusCompiler.Compiler attribute), 272
- VHDL_VERSION (Compiler.VivadoCompiler.Compiler attribute), 273
- VHDL_VERSION (Compiler.XCICompiler.Compiler attribute), 275
- VHDL_VERSION (Compiler.XCOCCompiler.Compiler attribute), 277
- VHDL_VERSION (Compiler.XSTCompiler.Compiler attribute), 279
- VHDL_VERSION (Simulator.ActiveHDL Simulator.Simulator attribute), 333
- VHDL_VERSION (Simulator.Cocotb Simulator.Simulator attribute), 334
- VHDL_VERSION (Simulator.GHDL Simulator.Simulator attribute), 336
- VHDL_VERSION (Simulator.ISE Simulator.Simulator attribute), 338
- VHDL_VERSION (Simulator.ModelSim Simulator.Simulator attribute), 339
- VHDL_VERSION (Simulator.Questa Simulator.Simulator attribute), 341
- VHDL_VERSION (Simulator.Simulator attribute), 345
- VHDL_VERSION (Simulator.Vivado Simulator.Simulator attribute), 342
- VHDLCompiler (class in ToolChain.Aldec.ActiveHDL), 352
- VHDLCompiler (class in ToolChain.Aldec.RivieraPRO), 358
- VHDLCompiler (class in

ToolChain.Mentor.ModelSim), 453

VHDL Simulator.Executable (class in ToolChain.Aldec.RivieraPRO), 360

VHDL Simulator.Executable (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.FlagBatchMode (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.FlagCommandLineMode (class in ToolChain.Aldec.RivieraPRO), 360

VHDL Simulator.FlagCommandLineMode (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.FlagDisableCoverage (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagDisableKeepAssertionCountsForCoverage (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagDisableOptimization (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagDisablePSL (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnableCoverage (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnableFSMDebugging (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnableKeepAssertionCountsForCoverage (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnableOptimization (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnableOptimizationVerbosity (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagEnablePSL (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagForceLanguageChecks (class in ToolChain.Mentor.ModelSim), 453

VHDL Simulator.FlagGuiMode (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.FlagQuietMode (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.FlagRelaxLanguageChecks (class in ToolChain.Mentor.ModelSim), 453

VHDL Simulator.FlagReportAsError (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagReportAsFatal (class in ToolChain.Mentor.ModelSim), 453

VHDL Simulator.FlagReportAsNote (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.FlagReportAsWarning (class in ToolChain.Mentor.ModelSim), 452

VHDL Simulator.SwitchBatchCommand (class in ToolChain.Aldec.RivieraPRO), 360

VHDL Simulator.SwitchBatchCommand (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.SwitchModelSimIniFile (class in ToolChain.Mentor.ModelSim), 451

VHDL Simulator.SwitchTimeResolution (class in ToolChain.Aldec.RivieraPRO), 360

VHDL Simulator.SwitchTimeResolution (class in ToolChain.Mentor.ModelSim), 453

VHDL Simulator.SwitchTopLevel (class in ToolChain.Aldec.RivieraPRO), 360

VHDL Simulator.SwitchTopLevel (class in ToolChain.Mentor.ModelSim), 453

VHDL SourceFile (class in Base.Project), 263

VHDL SourceFileMixIn (class in Parser.FilesParser), 324

VHDL Standalone Simulator (class in ToolChain.Aldec.ActiveHDL), 353

VHDL Standalone Simulator.Executable (class in ToolChain.Aldec.ActiveHDL), 353

VHDL Standalone Simulator.SwitchBatchCommand (class in ToolChain.Aldec.ActiveHDL), 353

VHDL Statement (class in Parser.FilesCodeDOM), 319

VHDL Testbench (class in DataBase.Entity), 300

VHDL Testbench (DataBase.Entity.IPCore attribute), 299

VHDL Testbenches (DataBase.Entity.AskWildcard attribute), 298

VHDL Testbenches (DataBase.Entity.StarWildcard attribute), 298

VHDL Testbenches (DataBase.Entity.WildCard attribute), 296

VHDL Version (Base.Project.Project attribute), 261

VHDL Version (Base.Shared.Shared attribute), 265

VHDL Version (class in Base.Project), 260

VHDL Version (Compiler.Compiler attribute), 283

VHDL Version (Compiler.ISECompiler.Compiler attribute), 268

VHDL Version (Compiler.LSECompiler.Compiler attribute), 270

VHDL Version (Compiler.QuartusCompiler.Compiler attribute), 272

VHDL Version (Compiler.VivadoCompiler.Compiler attribute), 273

VHDL Version (Compiler.XCICompiler.Compiler attribute), 275

VHDL Version (Compiler.XCOCCompiler.Compiler attribute), 277

VHDL Version (Compiler.XSTCompiler.Compiler attribute), 279

VHDL Version (DataBase.Solution.VirtualProject attribute), 309

VHDL Version (Simulator.ActiveHDL Simulator.Simulator attribute), 333

VHDL Version (Simulator.Cocotb Simulator.Simulator attribute), 334

VHDL Version (Simulator.GHDL Simulator.Simulator attribute), 336

VHDL Version (Simulator.ISE Simulator.Simulator attribute), 337

VHDL Version (Simulator.ModelSim Simulator.Simulator attribute), 339

VHDL Version (Simulator.Questa Simulator.Simulator attribute), 341

VHDL Version (Simulator.Simulator attribute), 346

VHDL Version (Simulator.Vivado Simulator.Simulator

- attribute), 342
 - VHDLVersion (ToolChain.Altera.Quartus.QuartusProject attribute), 378
 - VHDLVersion (ToolChain.Lattice.Diamond.SynthesisArgumentFile attribute), 430
 - VHDLVersion (ToolChain.Xilinx.ISE.ISEProject attribute), 474
 - VHDLVersion (ToolChain.Xilinx.Vivado.VivadoProject attribute), 483
 - View (Simulator.SimulationState attribute), 345
 - View() (ToolChain.GTKWave.GTKWave method), 401
 - Virtex (DataBase.Config.XilinxFamilies attribute), 285
 - Virtex2 (DataBase.Config.XilinxDevices attribute), 287
 - Virtex4 (DataBase.Config.XilinxDevices attribute), 287
 - Virtex5 (DataBase.Config.XilinxDevices attribute), 287
 - Virtex6 (DataBase.Config.XilinxDevices attribute), 287
 - Virtex7 (DataBase.Config.XilinxDevices attribute), 287
 - VirtexUltraScale (DataBase.Config.XilinxDevices attribute), 287
 - VirtexUltraScalePlus (DataBase.Config.XilinxDevices attribute), 287
 - VirtualProject (class in DataBase.Solution), 308
 - Visibility (class in DataBase.Entity), 294
 - Visibility (DataBase.Entity.AskWildCard attribute), 298
 - Visibility (DataBase.Entity.CocoTestbench attribute), 301
 - Visibility (DataBase.Entity.CoreGeneratorNetlist attribute), 304
 - Visibility (DataBase.Entity.IPCore attribute), 299
 - Visibility (DataBase.Entity.LatticeNetlist attribute), 303
 - Visibility (DataBase.Entity.LazyPathElement attribute), 299
 - Visibility (DataBase.Entity.Library attribute), 296
 - Visibility (DataBase.Entity.Namespace attribute), 295
 - Visibility (DataBase.Entity.Netlist attribute), 302
 - Visibility (DataBase.Entity.PathElement attribute), 295
 - Visibility (DataBase.Entity.QuartusNetlist attribute), 303
 - Visibility (DataBase.Entity.StarWildCard attribute), 298
 - Visibility (DataBase.Entity.Testbench attribute), 300
 - Visibility (DataBase.Entity.VHDLTestbench attribute), 301
 - Visibility (DataBase.Entity.VivadoNetlist attribute), 305
 - Visibility (DataBase.Entity.WildCard attribute), 297
 - Visibility (DataBase.Entity.XstNetlist attribute), 302
 - Vivado (class in ToolChain.Xilinx.Vivado), 478
 - VivadoException, 475
 - VivadoNetlist (class in DataBase.Entity), 304
 - VivadoNetlist (DataBase.Entity.IPCore attribute), 299
 - VivadoNetlists (DataBase.Entity.AskWildCard attribute), 298
 - VivadoNetlists (DataBase.Entity.StarWildCard attribute), 298
 - VivadoNetlists (DataBase.Entity.WildCard attribute), 296
 - VivadoProject (class in DataBase.Solution), 307
 - VivadoProject (class in ToolChain.Xilinx.Vivado), 483
 - VivadoProjectFile (class in ToolChain.Xilinx.Vivado), 483
 - VLibFilter() (in module ToolChain.Aldec.ActiveHDL), 354
 - VLibFilter() (in module ToolChain.Aldec.RivieraPRO), 361
 - VLibFilter() (in module ToolChain.Mentor.ModelSim), 454
 - VSimFilter() (in module ToolChain.Aldec.ActiveHDL), 354
 - VSimFilter() (in module ToolChain.Aldec.RivieraPRO), 361
 - VSimFilter() (in module ToolChain.Mentor.ModelSim), 454
 - VUnit
 - Third-Party Libraries, 204
- ## W
- Warning (Base.Logging.Severity attribute), 256
 - Warnings (DataBase.Solution.FileListFile attribute), 309
 - Warnings (Parser.FilesParser.FilesParserMixIn attribute), 326
 - WildCard (class in DataBase.Entity), 296
 - WindowsCommandArgument (class in Base.Executable), 246
 - WindowsException, 464
 - WindowsFlagArgument (class in Base.Executable), 249
 - WindowsTupleArgument (class in Base.Executable), 254
 - WindowsValuedFlagArgument (class in Base.Executable), 250
 - WindowsValuedFlagListArgument (class in Base.Executable), 252
 - with_traceback() (lib.Parser.EmptyChoiseParserResult method), 513
 - with_traceback() (lib.Parser.GreedyMatchingParserResult method), 513
 - with_traceback() (lib.Parser.MatchingParserResult method), 513
 - with_traceback() (lib.Parser.MismatchingParserResult method), 513
 - with_traceback() (lib.Parser.ParserException method), 512
 - Write() (Base.Logging.Logger method), 257
 - write() (lib.ExtendedConfigParser.ExtendedConfigParser method), 510
 - Write() (ToolChain.Altera.Quartus.QuartusSettings method), 378
 - Write() (ToolChain.Lattice.Diamond.SynthesisArgumentFile method), 430
 - WriteDebug() (Base.Logging.Logger method), 257
 - WriteDryRun() (Base.Logging.Logger method), 257
 - WriteError() (Base.Logging.Logger method), 257
 - WriteFatal() (Base.Logging.Logger method), 257

WriteInfo() (Base.Logging.Logger method), 257
 WriteNormal() (Base.Logging.Logger method), 257
 WriteQuiet() (Base.Logging.Logger method), 257
 WriteVerbose() (Base.Logging.Logger method), 257
 WriteWarning() (Base.Logging.Logger method), 257

X

X (DataBase.Config.XilinxSubTypes attribute), 288
 XcfFile (DataBase.Entity.XstNetlist attribute), 302
 XcoFile (DataBase.Entity.CoreGeneratorNetlist attribute), 304
 XDCSourceFileMixIn (class in Parser.FilesParser), 325
 XDCStatement (class in Parser.FilesCodeDOM), 321
 XElab (class in ToolChain.Xilinx.Vivado), 478
 XElab.ArgTopLevel (class in ToolChain.Xilinx.Vivado), 479
 XElab.Executable (class in ToolChain.Xilinx.Vivado), 478
 XElab.FlagRangeCheck (class in ToolChain.Xilinx.Vivado), 478
 XElab.SwitchDebug (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchLogFile (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchMultiThreading (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchOptimization (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchProjectFile (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchSnapshot (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchTimeResolution (class in ToolChain.Xilinx.Vivado), 479
 XElab.SwitchVerbose (class in ToolChain.Xilinx.Vivado), 479
 Xilinx (DataBase.Config.Vendors attribute), 284
 Xilinx ISE
 Pre-compilation, 52
 Xilinx Vivado
 Pre-compilation, 53
 Xilinx_CoreGen (Base.Project.Tool attribute), 260
 Xilinx_IPCatalog (Base.Project.Tool attribute), 260
 Xilinx_ISE (Base.Project.ToolChain attribute), 259
 Xilinx_iSim (Base.Project.Tool attribute), 260
 Xilinx_PlanAhead (Base.Project.ToolChain attribute), 259
 Xilinx_Synth (Base.Project.Tool attribute), 260
 Xilinx_Vivado (Base.Project.ToolChain attribute), 259
 Xilinx_xSim (Base.Project.Tool attribute), 260
 Xilinx_XST (Base.Project.Tool attribute), 260
 XilinxDesignConstraintFile (class in ToolChain.Xilinx.Vivado), 484
 XilinxDevices (class in DataBase.Config), 287
 XilinxException, 484
 XilinxFamilies (class in DataBase.Config), 285
 XilinxProjectExportMixIn (class in ToolChain.Xilinx), 486

XilinxSubTypes (class in DataBase.Config), 288
 XorExpression (class in lib.CodeDOM), 502
 XSim (class in ToolChain.Xilinx.Vivado), 480
 XSim.Executable (class in ToolChain.Xilinx.Vivado), 480
 XSim.FlagGuiMode (class in ToolChain.Xilinx.Vivado), 480
 XSim.SwitchLogFile (class in ToolChain.Xilinx.Vivado), 480
 XSim.SwitchSnapshot (class in ToolChain.Xilinx.Vivado), 481
 XSim.SwitchTclBatchFile (class in ToolChain.Xilinx.Vivado), 481
 XSim.SwitchWaveformFile (class in ToolChain.Xilinx.Vivado), 481
 Xst (class in ToolChain.Xilinx.ISE), 471
 Xst.Executable (class in ToolChain.Xilinx.ISE), 471
 Xst.SwitchIntStyle (class in ToolChain.Xilinx.ISE), 471
 Xst.SwitchReportFile (class in ToolChain.Xilinx.ISE), 472
 Xst.SwitchXstFile (class in ToolChain.Xilinx.ISE), 471
 XstFile (DataBase.Entity.XstNetlist attribute), 302
 XstFilter() (in module ToolChain.Xilinx.ISE), 475
 XstNetlist (class in DataBase.Entity), 302
 XSTNetlist (DataBase.Entity.IPCore attribute), 299
 XSTNetlists (DataBase.Entity.AskWildcard attribute), 298
 XSTNetlists (DataBase.Entity.StarWildcard attribute), 298
 XSTNetlists (DataBase.Entity.WildCard attribute), 296
 XstTemplateFile (DataBase.Entity.XstNetlist attribute), 302
 XT (DataBase.Config.XilinxSubTypes attribute), 289

Z

Zynq (DataBase.Config.XilinxFamilies attribute), 286
 Zynq7000 (DataBase.Config.XilinxDevices attribute), 287