DashRTL

ACCELERATE RTL DESIGN CYCLES: 10X TO 1000X SPEEDUP WITH ULTRA-FAST MULTICORE COMPILER TECHNOLOGY

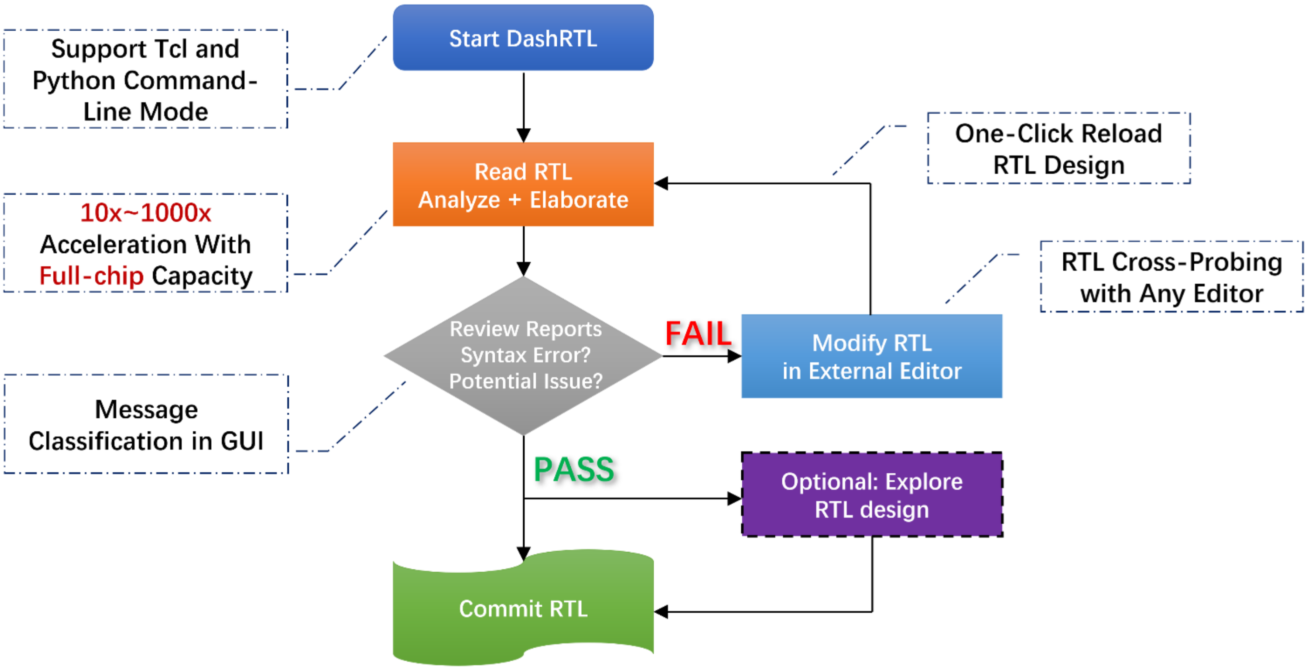
WHY DASHRTL?

DashRTL is a high-performance RTL design platform engineered for large-scale chip designs. As projects grow in complexity, RTL development often requires frequent compilations to identify and resolve issues during iterative design cycles. Traditional approaches, such as using linting, synthesis, or simulation tools, can be highly time-consuming and inefficient.

DashRTL overcomes these challenges with its ultra-fast multicore compiler, which enables parallel RTL processing for rapid design iterations. For large-scale SoC projects, DashRTL's robust capacity and scalability supports full-chip RTL compilation and comprehensive checks, eliminating the need for time-intensive hierarchical RTL checking.

Whether you are verifying syntax, checking synthesizability, or exploring design hierarchies, DashRTL ensures that your RTL code is error-free and ready for design integration, reducing iteration cycles from hours to seconds.

RTL Design Flow Using DashRTL



Start DashRTL

Read RTL
Analyze + Elaborate

Review Reports
Syntax Error?
Potential Issue?

FAIL

Modify RTL
in External Editor

PASS

Optional: Explore
RTL design

Commit RTL

Support Tcl and Python Command-Line Mode

10x~1000x Acceleration With Full-chip Capacity

Message Classification in GUI

One-Click Reload RTL Design

RTL Cross-Probing with Any Editor

Contact Us

contact@dashthru.com

https://www.dashthru.com

Resources

DashRTL User Guide

DashRTL Video

Start Free Trial Now

DashThru

PERFORMANCE

DashRTL leverages innovative multicore compiler technology to achieve 10x-1000x improvements over traditional RTL checking tools.

This runtime data compares DashRTL with mainstream linting, synthesis, and simulation tools, all based on a direct comparison of the total time spent on the analyze and elaborate steps.

RTL Reading Runtime	DashRTL	Linting Tool	Synthesis Tool	Simulation Tool
Design A Cell Count ~0.1M	0.6s	69s	23s	7s
Design B Cell Count ~0.4M	1.2s	169s	54s	12s
Design C Cell Count ~2.5M	3.0s	237s	122s	61s

KEY BENEFITS

Items	DashRTL	Linting Tool	Synthesis Tool	Simulation Tool
Checking Performance	Ultra-Fast	Very Slow	Slow	Fast
Checking Scalability	Block / Full Chip	Block / Hierarchical	Block / Hierarchical	Block / Hierarchical
Checking Strategy	Moderate	Over Check	Under Check	Under Check
Synthesizability Check	Yes	Partial	Yes	No
SV Compatibility Solution	Yes	No	No	No
Message Classification in GUI	Yes	Yes	No	No
Command-Line Mode	Tcl / Python / Mix	Tcl	Tcl / Python	Tcl
RTL Read Processing	Multi Core	Single Core	Single Core	Single Core
RTL Cross-Probing with External Editor	Yes	No	No	No

Contact Us

contact@dashthru.com
https://www.dashthru.com

Resources

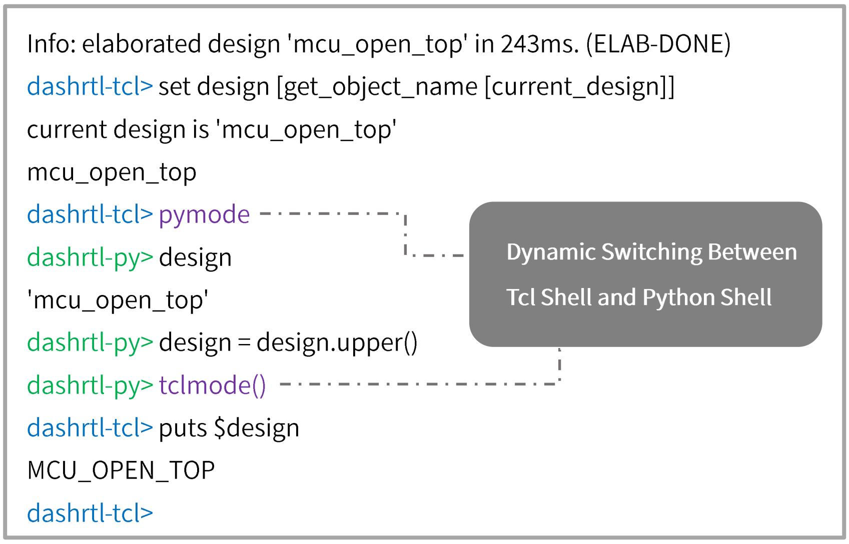
DashRTL User Guide
DashRTL Video
Start Free Trial Now

DashThru

KEY FEATURES

Dynamic Workflow

Offers dual launch modes (Simulation-Style and Synthesis-Style) and supports Tcl/Python scripting for customizable workflows.

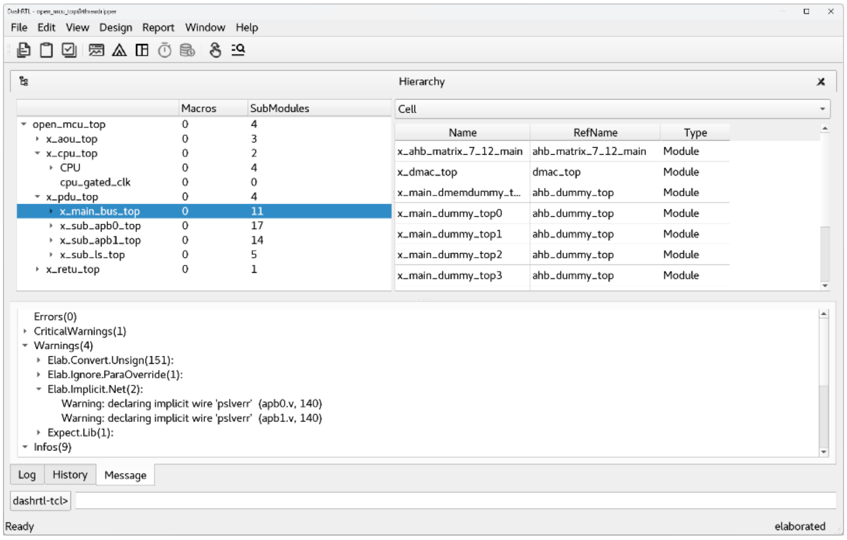


```
Info: elaborated design 'mcu_open_top' in 243ms. (ELAB-DONE)
dashrtl-tcl> set design [get_object_name [current_design]]
current design is 'mcu_open_top'
mcu_open_top
dashrtl-tcl> pymode
dashrtl-py> design
'mcu_open_top'
dashrtl-py> design = design.upper()
dashrtl-py> tclmode()
dashrtl-tcl> puts $design
MCU_OPEN_TOP
dashrtl-tcl>
```

Dynamic Switching Between Tcl Shell and Python Shell

Smart GUI

Features cross-probe debugging, message classification, and one-click navigation to code lines in third-party editors for instant issue resolution.



DashRTL - open_mcu_top@threadripper
File Edit View Design Report Window Help
Hierarchy
Cell
Log
History
Message

	Macros	SubModules
open_mcu_top	0	4
x_aou_top	0	3
x_cpu_top	0	2
CPU	0	4
cpu_gated_clk	0	0
x_pdu_top	0	4
x_main_bus_top	0	11
x_sub_apb0_top	0	17
x_sub_apb1_top	0	14
x_sub_ls_top	0	5
x_retu_top	0	1

Name	RefName	Type
x_ahb_matrix_7_12_main	ahb_matrix_7_12_main	Module
x_dmac_top	dmac_top	Module
x_main_dmemdummy_t...	ahb_dummy_top	Module
x_main_dummy_top0	ahb_dummy_top	Module
x_main_dummy_top1	ahb_dummy_top	Module
x_main_dummy_top2	ahb_dummy_top	Module
x_main_dummy_top3	ahb_dummy_top	Module

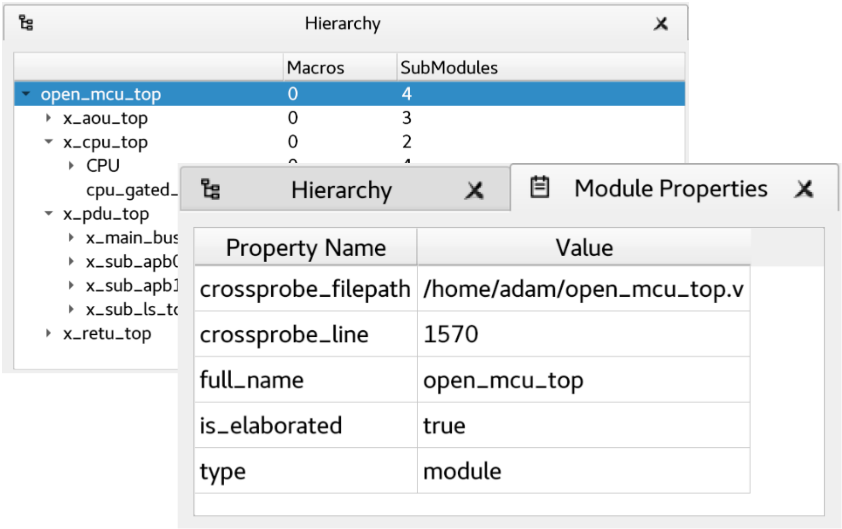
```
Errors(0)
CriticalWarnings(1)
Warnings(4)
  Elab.Convert.Unsign(151):
  Elab.Ignore.ParaOverride(1):
  Elab.Implicit.Net(2):
    Warning: declaring implicit wire 'pslverr'  (apb0.v, 140)
    Warning: declaring implicit wire 'pslverr'  (apb1.v, 140)
  Expect.Lib(1):
Infos(9)
```

dashrtl-tcl>

Ready elaborated

Design Exploration

Enables hierarchical exploration of RTL designs, including design object and property analysis, for deeper insights into design structures.



	Macros	SubModules
open_mcu_top	0	4
x_aou_top	0	3
x_cpu_top	0	2
CPU		
cpu_gated_		
x_pdu_top		
x_main_bus		
x_sub_apb(		
x_sub_apb1		
x_sub_ls_tc		
x_retu_top		

Property Name	Value
crossprobe_filepath	/home/adam/open_mcu_top.v
crossprobe_line	1570
full_name	open_mcu_top
is_elaborated	true
type	module

Contact Us

contact@dashthru.com

https://www.dashthru.com

Resources

DashRTL User Guide

DashRTL Video

Start Free Trial Now

DashThru