

SVase: A SystemVerilog pre-Elaborator

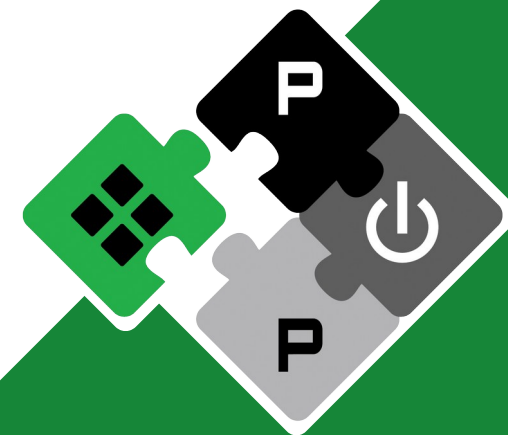
Yosys User's Group Meeting

Philippe Sauter

phsauter@ethz.ch

PULP Platform

Open Source Hardware, the way it should be!



@pulp_platform



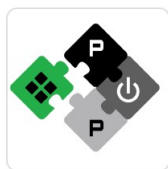
pulp-platform.org



youtube.com/pulp_platform



SystemVerilog



pulp-platform

Unfollow

Pinned

 **pulp** Public

This is the top-level project for the PULP Platform. It instantiates a PULP open-source system with a PULP SoC (microcontroller) domain accelerated by a PULP cluster with 8 cores.

 SystemVerilog  340
 99

 **pulpissimo** Public

This is the top-level project for the PULPissimo Platform. It instantiates a PULPissimo open-source system with a PULP SoC domain, but no cluster.

 SystemVerilog  300
 140

 **snitch** Public

Lean but mean RISC-V system!

 SystemVerilog  201
 41

 **hero** Public






Heterogeneous Research Platform (HERO) for exploration of heterogeneous computers consisting of programmable many-core accelerators and an application-class host CPU, including full-stack software ...

 SystemVerilog  70  17

People



Top languages

 SystemVerilog 
 Python  C++
 Shell

Most used topics

 riscv  asic  fpga
 risc-v  pulp

- **We use it, a lot**
 - The top language
 - **1M+ lines** of code
- Code is Silicon-proven
- Works in commercial tools



PULP goes Fully-Open!



Frontend

Backend

Closed
Synopsys

Closed
Innovus

Open
Yosys

Open
OpenROAD

Open Verification
Verilator

github.com/pulp-platform/iguana



PULP goes Fully-Open!



Frontend

Backend

Closed
Synopsys

Closed
Innovus

Open
Yosys

Open
OpenROAD

Open Verification
Verilator



Iggy Iguana

github.com/pulp-platform/iguana



PULP goes Fully Open!



From

Cadence
Synopsys

Open
Yosys

Open
OpenROAD

Open Verification
Verilator

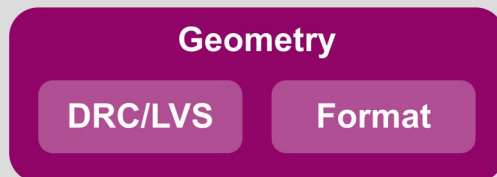
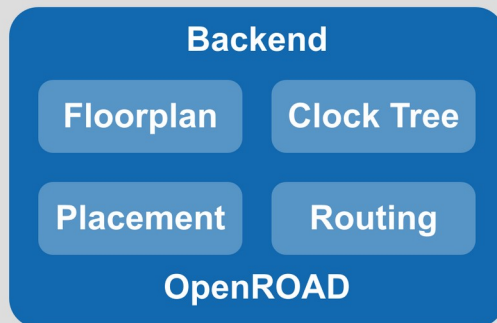
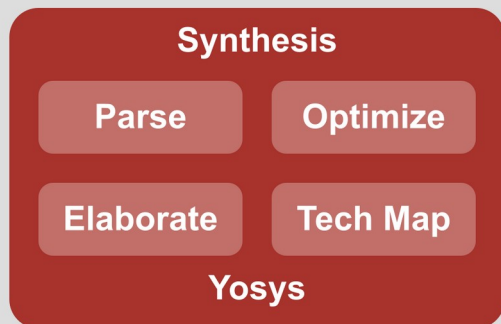
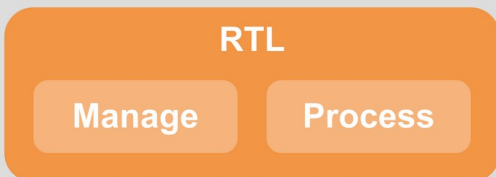
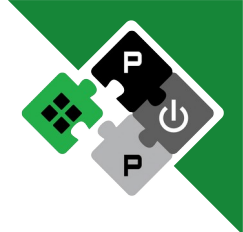


Iggy Iguana

github.com/pulp-platform/iguana



From Repos to Netlist

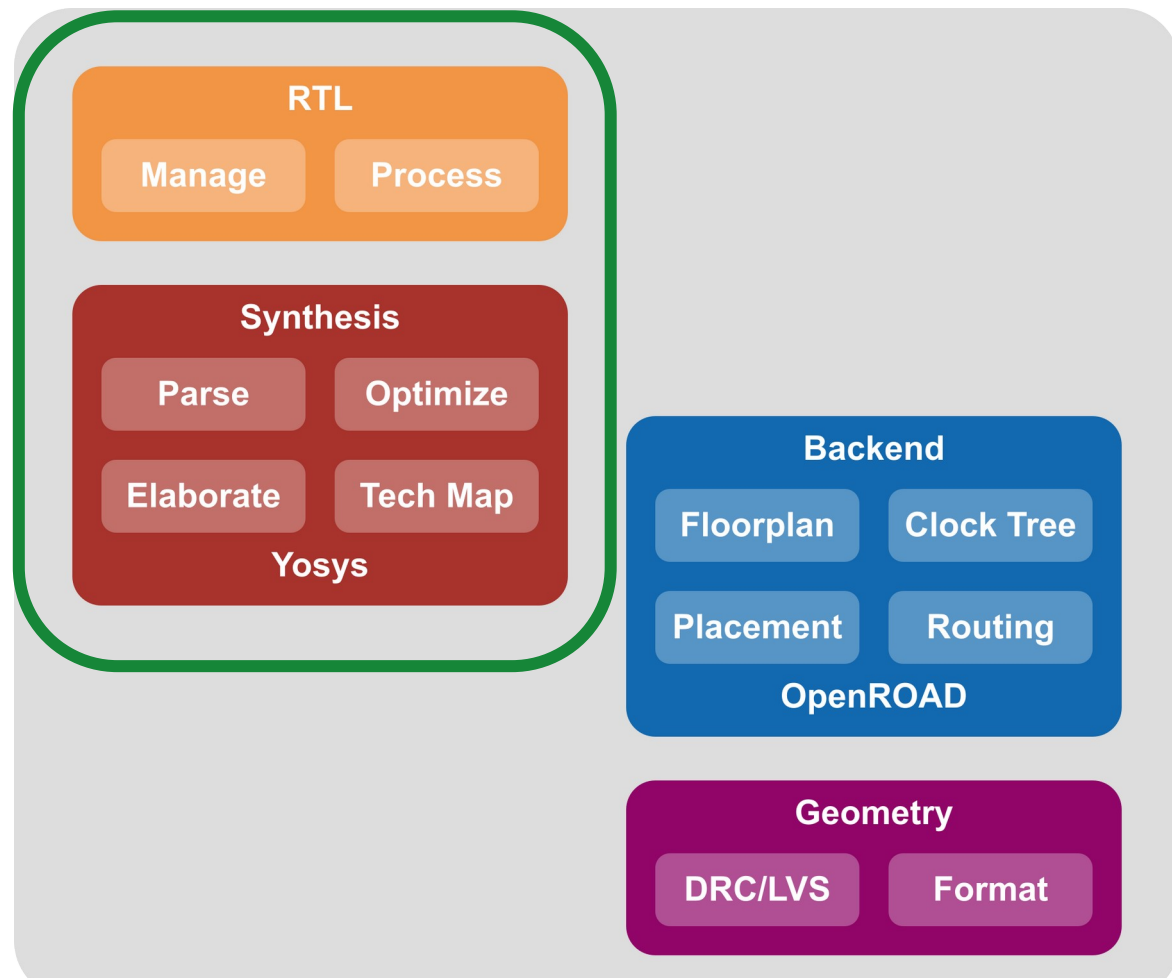


Iggy Iguana

github.com/pulp-platform/iguana



From Repos to Netlist

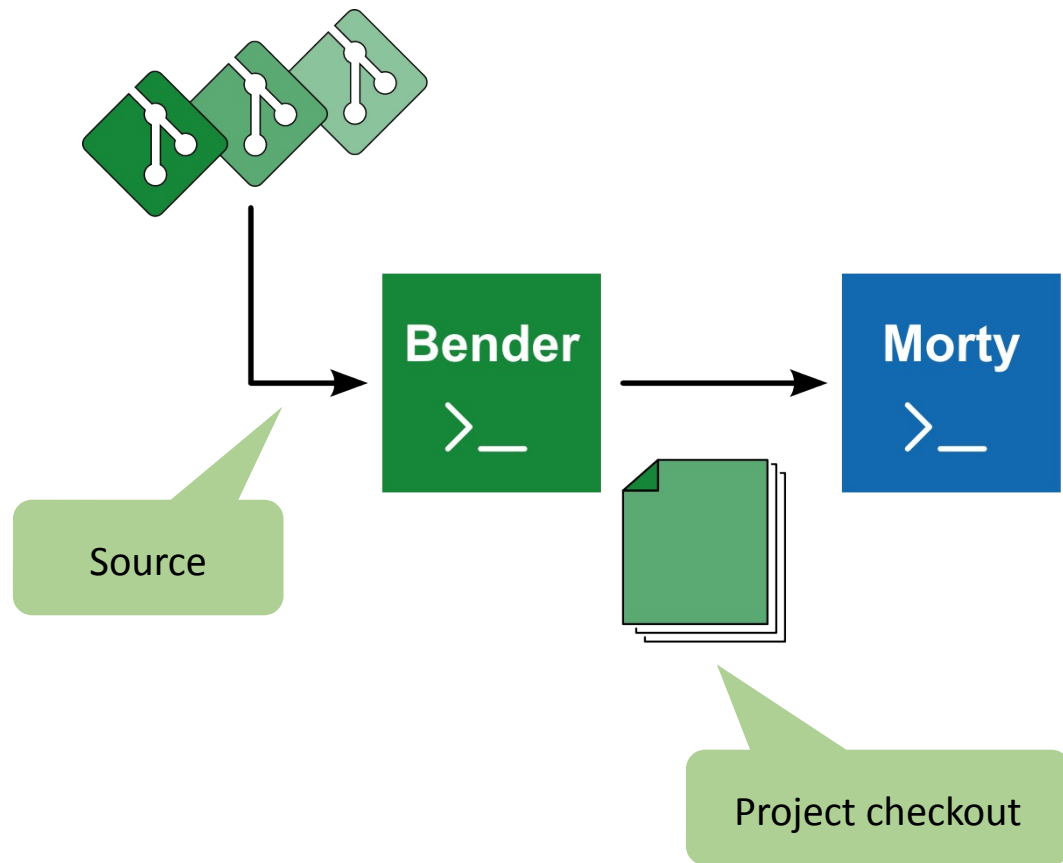


Iggy Iguana

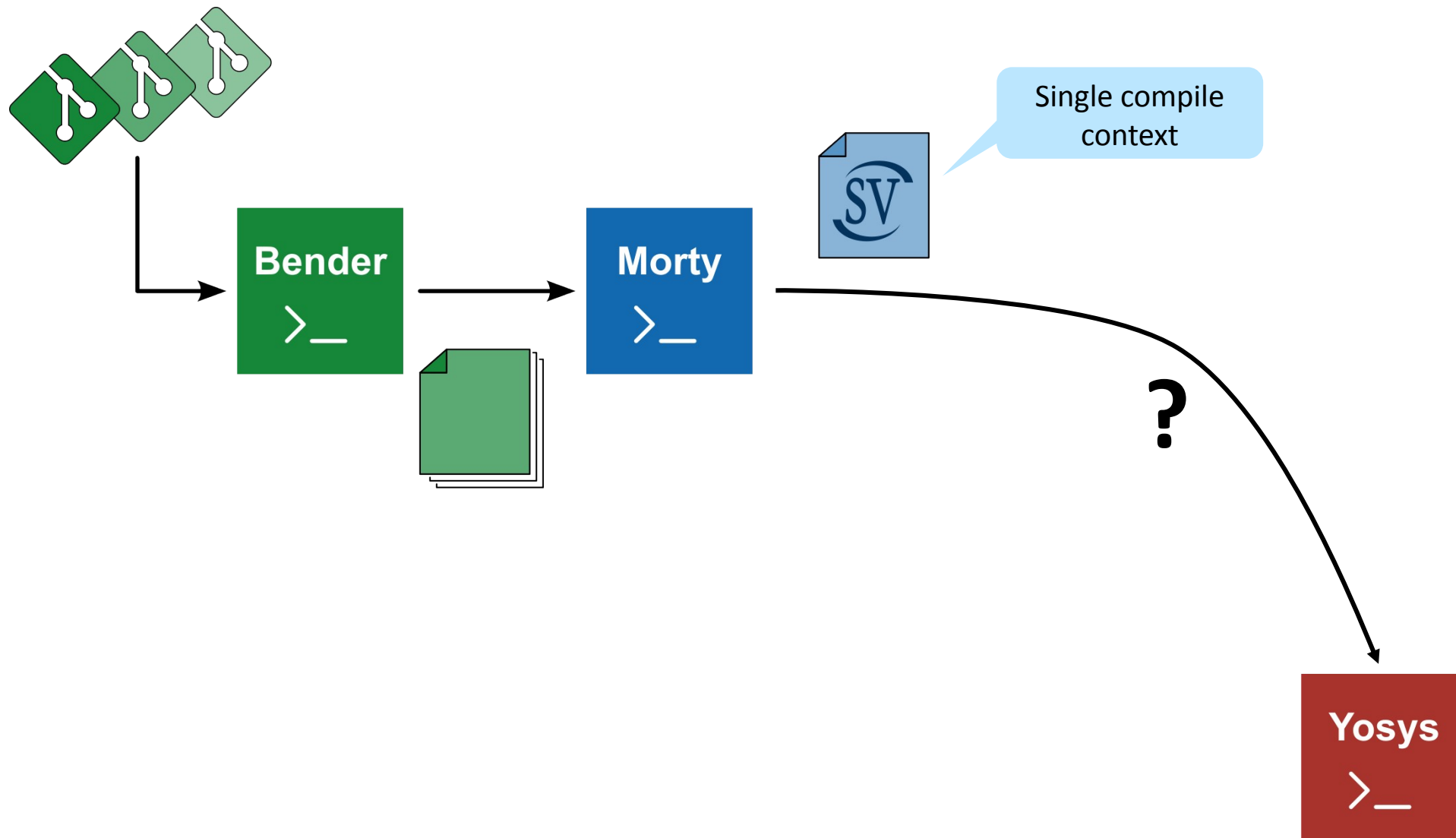
github.com/pulp-platform/iguana



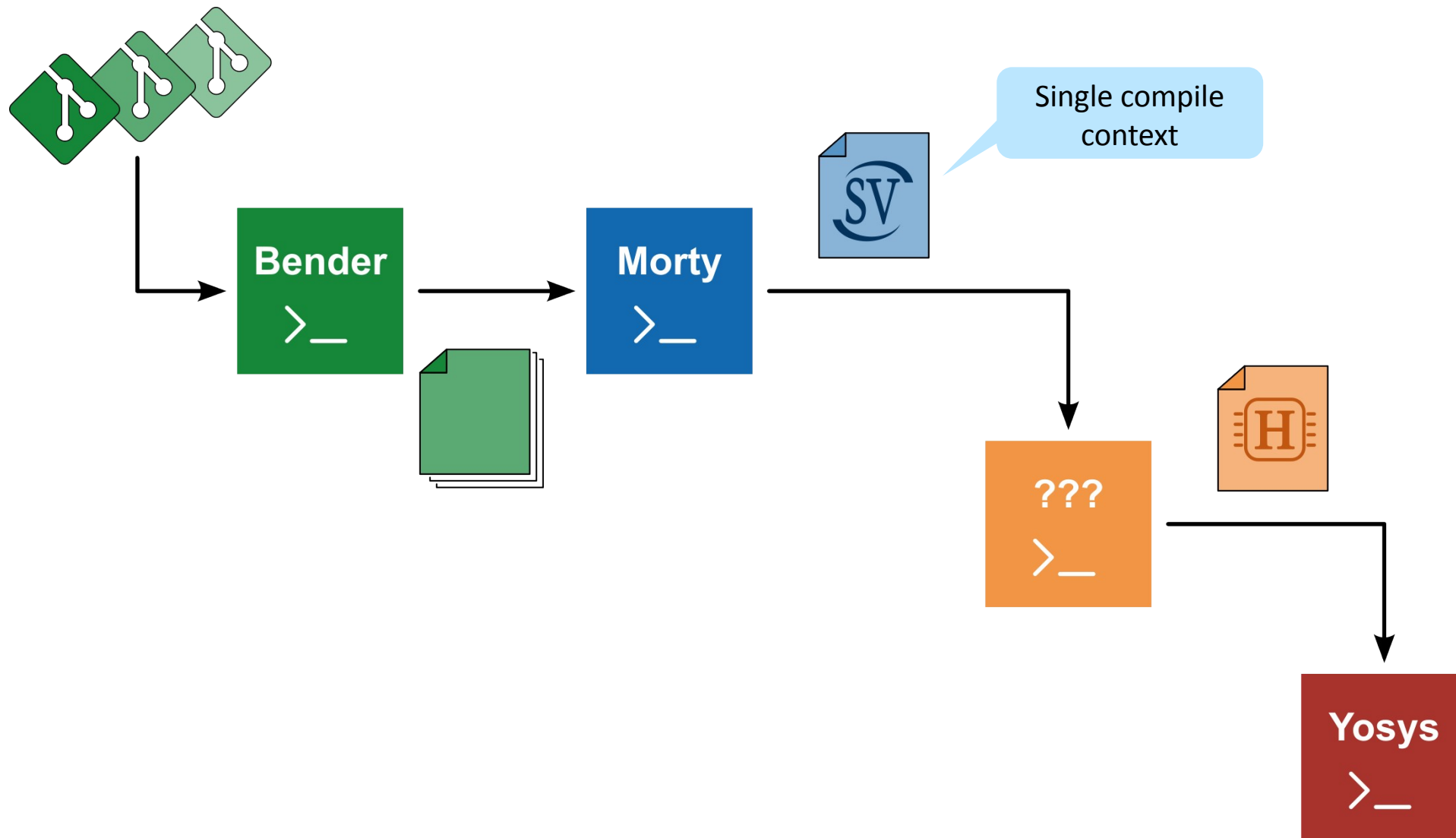
From Repos to Netlist



From Repos to Netlist



From Repos to Netlist



Which Frontend-Tool?

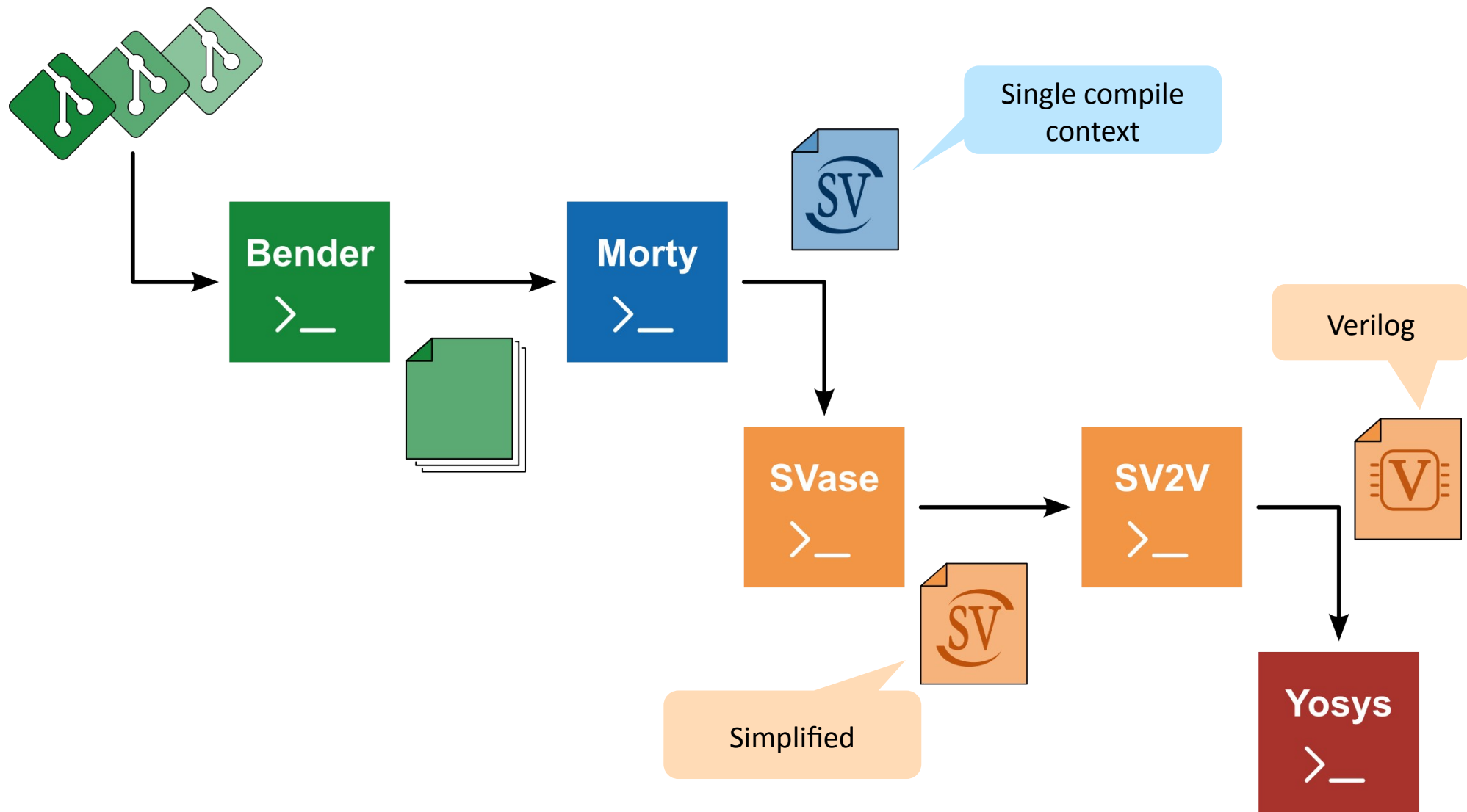


	Synlig (Surelog)	SV2V	Slang/SVase
Language features	1055/1619	1169/1604	1604/1604!!!
RAM usage	24GB	4GB	1GB
Runtime	19min	14min	11s
Development	Custom format + example driven	Verilog parser + example driven	SystemVerilog Spec
Language	C++	Haskell	C++

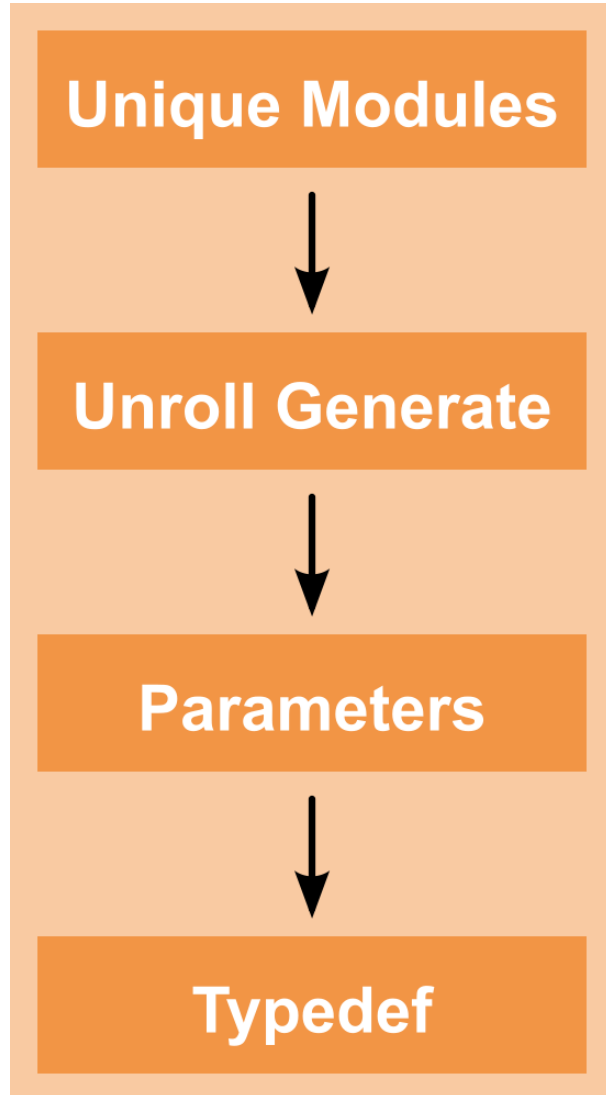
github.com/chipsalliance/sv-tests 



From Repos to Netlist



SVase

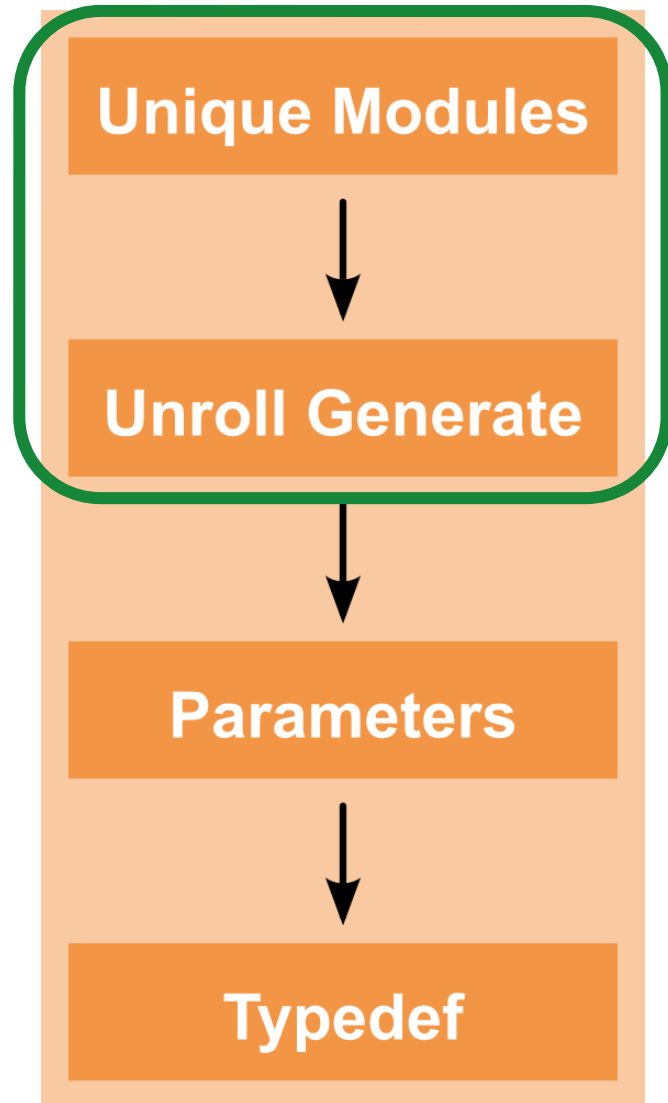


```
module test #(
    parameter int NoDefaultParam,
    parameter type PortTypeParam = logic
) ( );
endmodule
```

```
module top #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

    test #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

SVase

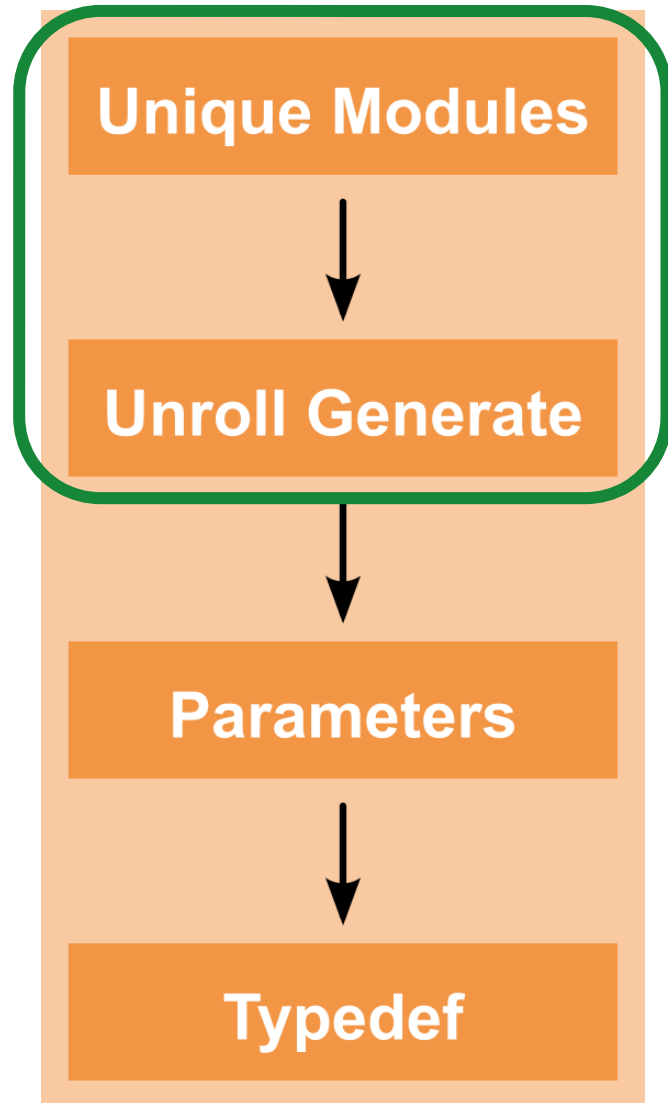


```
module test #(
    parameter int NoDefaultParam,
    parameter type PortTypeParam = logic
) ( );
endmodule

module top #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

    test #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

SVase

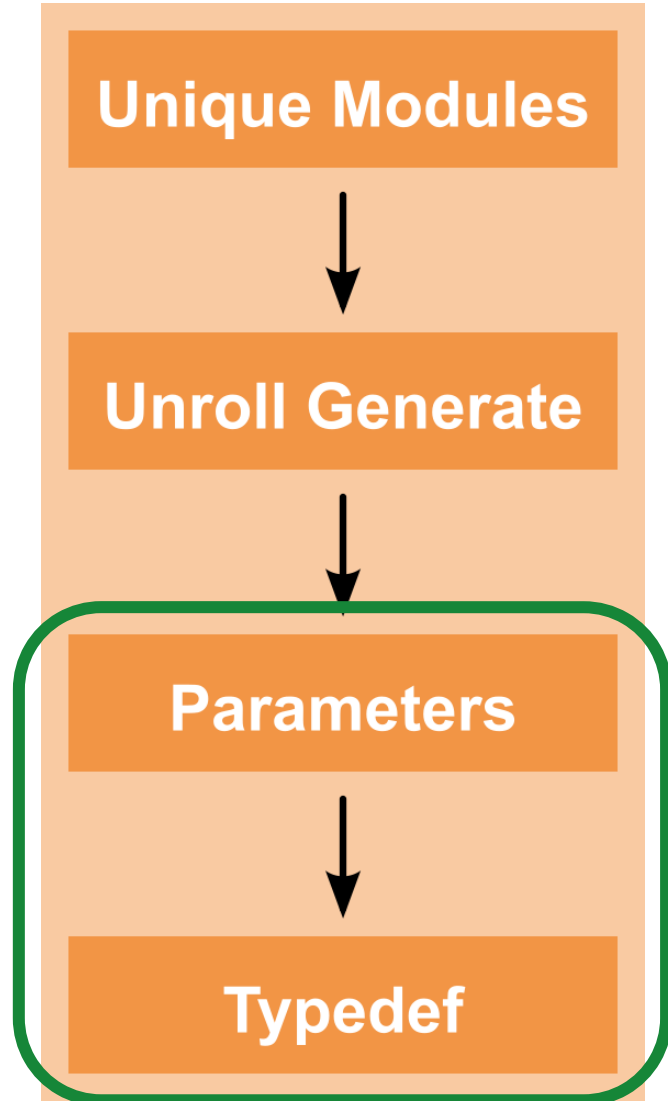


```
module test__1699695787177254841 #(
    // unique parametrization
) ( );
endmodule

module top__6142509188972423790 #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

    test__1699695787177254841 #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

SVase



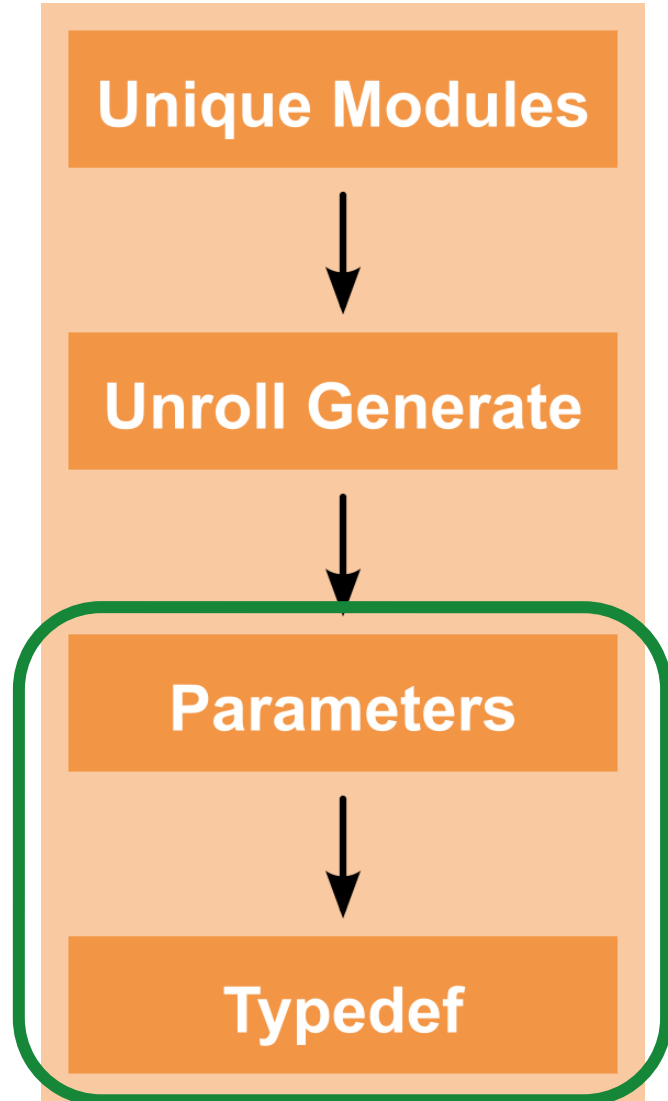
```
module test__1699695787177254841 #(
    parameter int NoDefaultParam=32'd5,
    parameter type PortTypeParam=struct packed {...}
) ( );
endmodule
```

```
module top__6142509188972423790 #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

    test__1699695787177254841 #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

An arrow points from the `TypeParam` definition in the second code block to its usage in the instantiation of `test__1699695787177254841`.

SVase



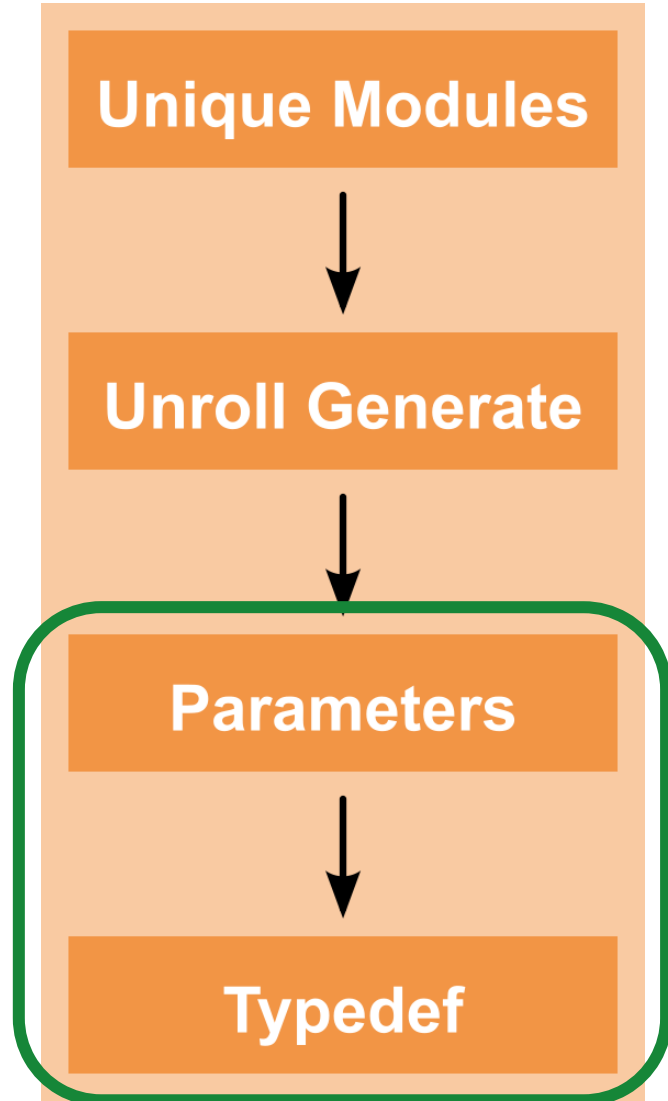
```
module test__1699695787177254841 #(
    parameter int NoDefaultParam=32'd5,
    parameter type PortTypeParam=struct packed {...}
) ( );
endmodule

module top__6142509188972423790 #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

    test__1699695787177254841 #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

A white arrow points from the `PortTypeParam` parameter in the first module to the `TypeParam` local parameter in the second module, illustrating the parameter passing mechanism.

SVase

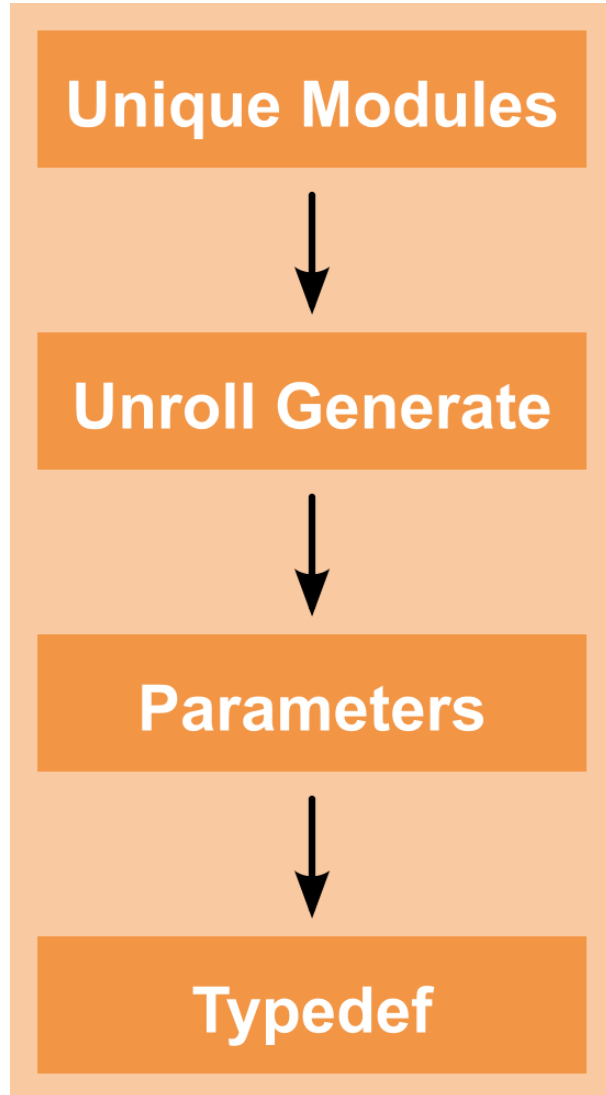


```
module test__1699695787177254841 #(
    parameter int NoDefaultParam=32'd5,
    parameter type PortTypeParam=struct packed {...}
) ( );
endmodule

module top__6142509188972423790 #( ) ( );
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

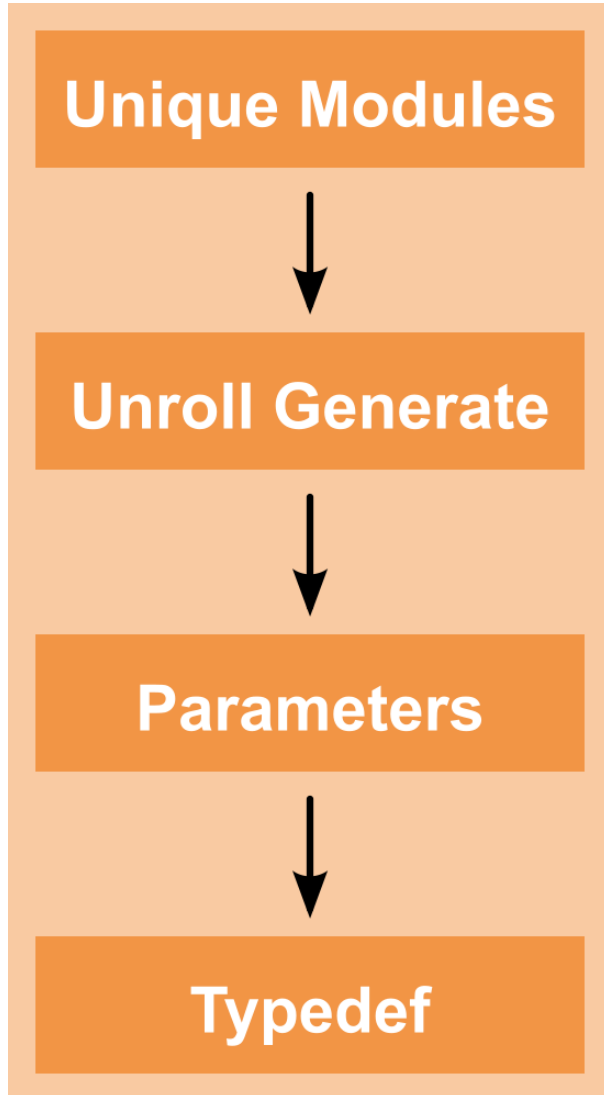
    test__1699695787177254841 #(
        .NoDefaultParam(TopParam),
        .PortTypeParam(TypeParam)
    ) i_test ( );
endmodule
```

A white arrow points from the `PortTypeParam` parameter in the `test__1699695787177254841` module to the `TypeParam` local parameter in the `top__6142509188972423790` module, illustrating the parameter passing mechanism.



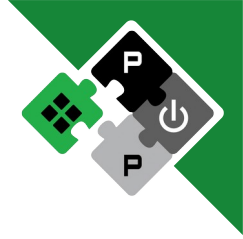
- **Simplifies SystemVerilog**
- We do not want to be SV2V or Synlig
- Meant as a tool





- **Simplifies SystemVerilog**
- We do not want to be SV2V or Synlig
- Meant as a tool
- In the future
 - eg. Interfaces → structs
 - Better documentation
(make it easier to customize passes)





Demo Time!



SystemVerilog



cheshire / hw / cheshire_soc.sv

Code Blame 1669 lines (1478 loc) · 55 KB

```
11 module cheshire_soc import cheshire_pkg::*; #(
12     // Cheshire config
13     parameter cheshire_cfg_t Cfg = '0,
14     // Debug info for external harts
15     parameter dm::hartinfo_t [iomsb(Cfg.NumExtDbgHarts):0] ExtHartinfo = '0,
16     // Interconnect types (must agree with Cheshire config)
17     parameter type axi_ext_llc_req_t = logic,
18     parameter type axi_ext_llc_rsp_t = logic,
19     parameter type axi_ext_mst_req_t = logic,
20     parameter type axi_ext_mst_rsp_t = logic,
21     parameter type axi_ext_slv_req_t = logic,
22     parameter type axi_ext_slv_rsp_t = logic,
23     parameter type reg_ext_req_t = logic,
24     parameter type reg_ext_rsp_t = logic
25 ) (
26     input logic clk_i,
27     input logic rst_ni,
28     input logic test_mode_i,
29     input logic [1:0] boot_mode_i,
30     input logic rtc_i,
```

complex types

functions

parametrized types

interfaces

```
94 interface XBAR_PERIPH_BUS
95     #(
96         parameter ID_WIDTH = `NB_CORES+1
97     );
98     // REQUEST CHANNEL
99     logic req;
100     logic [31:0] add;
101     logic wen;
102     logic [31:0] wdata;
103     logic [3:0] be;
104     logic gnt;
105     logic [ID_WIDTH-1:0] id;
106
107     // RESPONSE CHANNEL
108     logic r_valid;
109     logic r_opc;
110     logic [ID_WIDTH-1:0] r_id;
111     logic [31:0] r_rdata;
112
113     // Master Side
114     //*****
115     modport Master
116     (
117         output req, output add, output wen,
118         input gnt, input r_rdata, input r_opc
119     );
```

