

Industry-Grade SystemVerilog IPs And The Open Flow: How We Synthesized Iguana

Integrated Systems Laboratory (ETH Zürich)

Thomas Benz

tbenz@iis.ee.ethz.ch

Paul Scheffler

paulsc@iis.ee.ethz.ch

Jannis Schönleber

janniss@iis.ee.ethz.ch

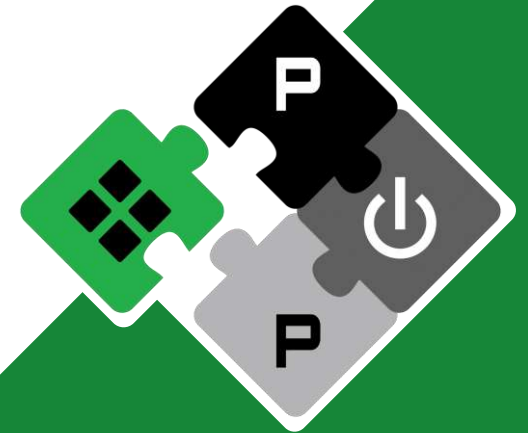
Philippe Sauter

phsauter@ethz.ch

FSiC2023, Paris, July 2023

PULP Platform

Open Source Hardware, the way it should be!



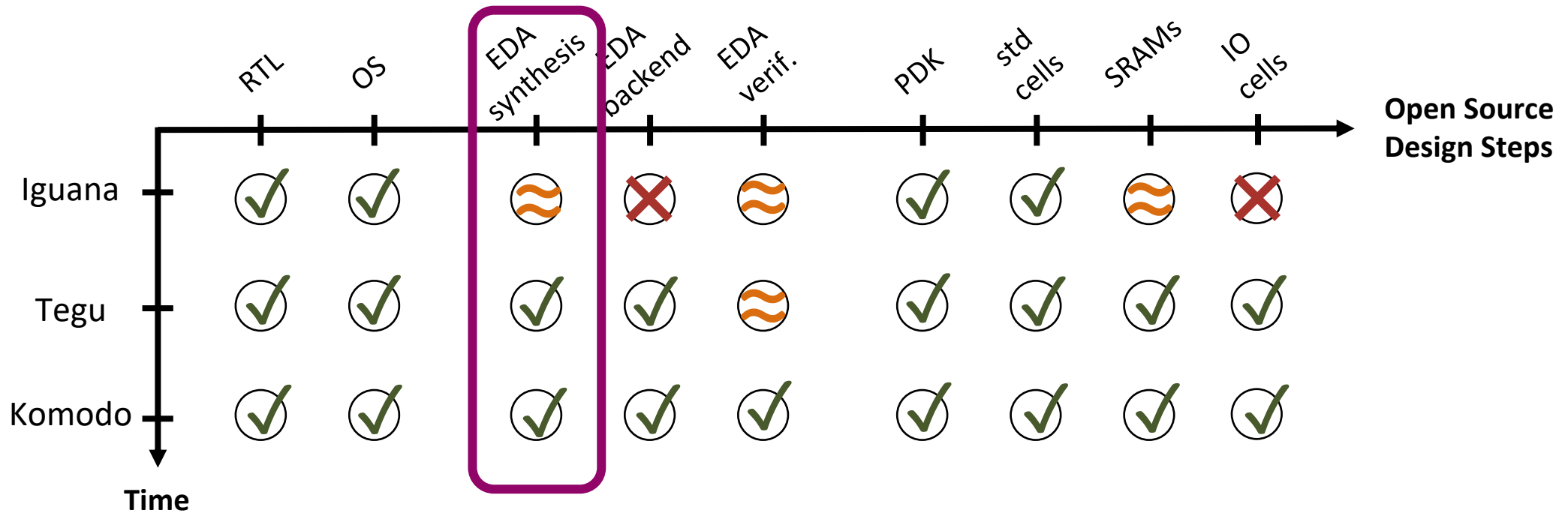
@pulp_platform 

pulp-platform.org 

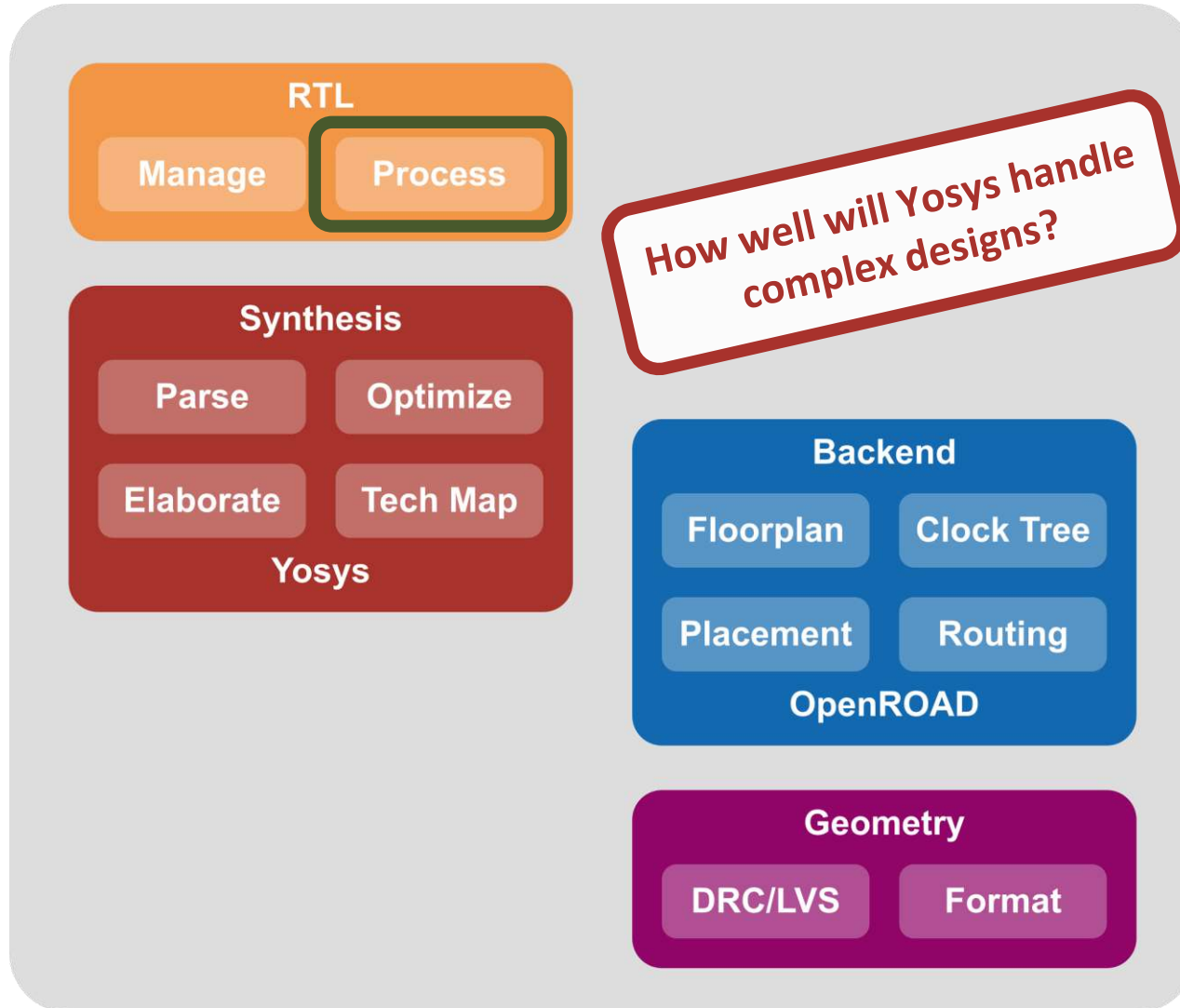
youtube.com/pulp_platform 

Our Goal

- **An end-to-end open Linux-capable ASIC (IHP 130nm)**
 - Open hardware (RTL), software/OS, EDA tools, PDK, standard cells
- **During the course of this project: increase openness**



Current Support In Open Synthesis



☰ README.md

yosys – Yosys Open SYNthesis Suite

This is a framework for RTL synthesis tools. It currently has extensive Verilog-2005 support and provides a set of synthesis reports for various applications.

**Natively: very solid Verilog support.
(But only limited SV)**

**There are solutions out there!
RTL pre-processing**



Parallel Ultra Low Power (PULP) Platform



- Research on open-source energy-efficient computing architectures



We Have Designed And Tested More Than 50 PULP ICs




2013 (3)	2014 (5)	2015 (10)	2016 (3)	2017 (2)	2018 (6)	2019 (7)	2020 (3)	2021 (7)	2022 (9)
 	  	   	 	 	  	  	 	  	    
PULPv1 <i>STM 28FDSOI</i> Multi-core processor	Diana <i>UMC 65</i> 4-core system with approximate FPU's	Fulmine <i>UMC 65</i> 4-core system with ML and Crypto accelerators	VivoSoC 2.001 <i>SMIC 130</i> Mixed signal system for biosignal acquisition	Mr. Wolf <i>TSMC 40</i> 8+1 core IoT processor	Poseidon <i>GF 22FDX</i> 64bit RISC-V core, 32bit Microcontroller system, ML accelerator	Baikonur <i>GF 22FDX</i> Dual 64bit RISC-V core, 3x 8core snitch clusters, Body biasing test vehicle	Dustin <i>TSMC 65</i> IoT processor with 16 cores and QNN enhancements	Kraken <i>GF 22FDX</i> IoT processor with Spiking Neural and Ternary Inference Engines	Occamy <i>GF 12LPP</i> ML accelerator with 216 + 1 cores and HBM interface

Check <http://asic.ethz.ch> for all our chips



- <https://github.com/pulp-platform>




- 

pulp-platform


[Overview](#)
[Repositories 239](#)
[Projects 1](#)
[Packages](#)
[People 14](#)

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
☆ 312
👤 93



[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
☆ 288
👤 137


[snitch](#)
Public

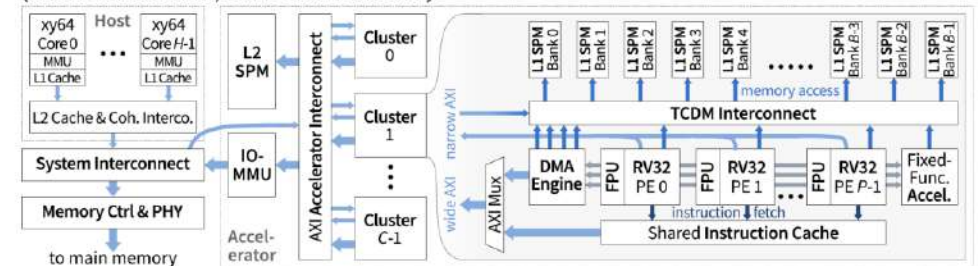
Lean but mean RISC-V system!


[hero](#)
Public

Heterogeneous Research Platform (HERO) for exploration of

HERO is an FPGA-based research platform that enables accurate and fast exploration of heterogeneous computers consisting of programmable many-core accelerators and an application-class host CPU. Currently, 32-bit RISC-V cores are supported in the accelerator and 64-bit ARMv8 or RISC-V cores as host CPU. HERO allows to seamlessly share data between host and accelerator through a unified heterogeneous programming interface based on OpenMP 4.5 and a mixed-data-model, mixed-ISA heterogeneous compiler based on LLVM.

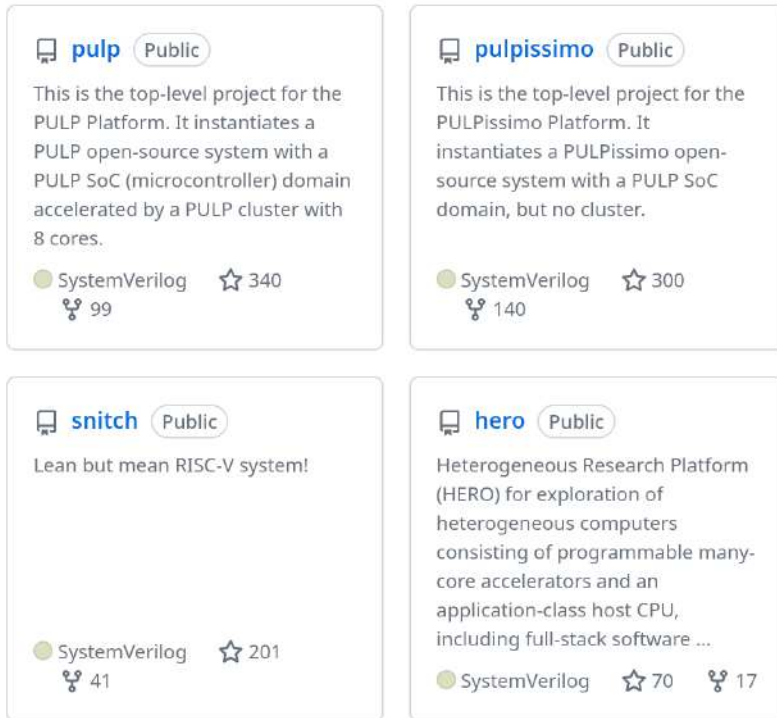
HERO's **hardware architecture**, shown below, combines a general-purpose host CPU (in the upper left corner) with a domain-specific programmable many-core accelerator (on the right side) so that data in the main memory (in the lower left corner) can be shared effectively.



SystemVerilog, Why Bother?



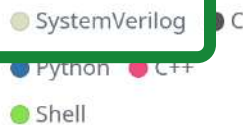
Pinned



People



Top languages



Most used topics

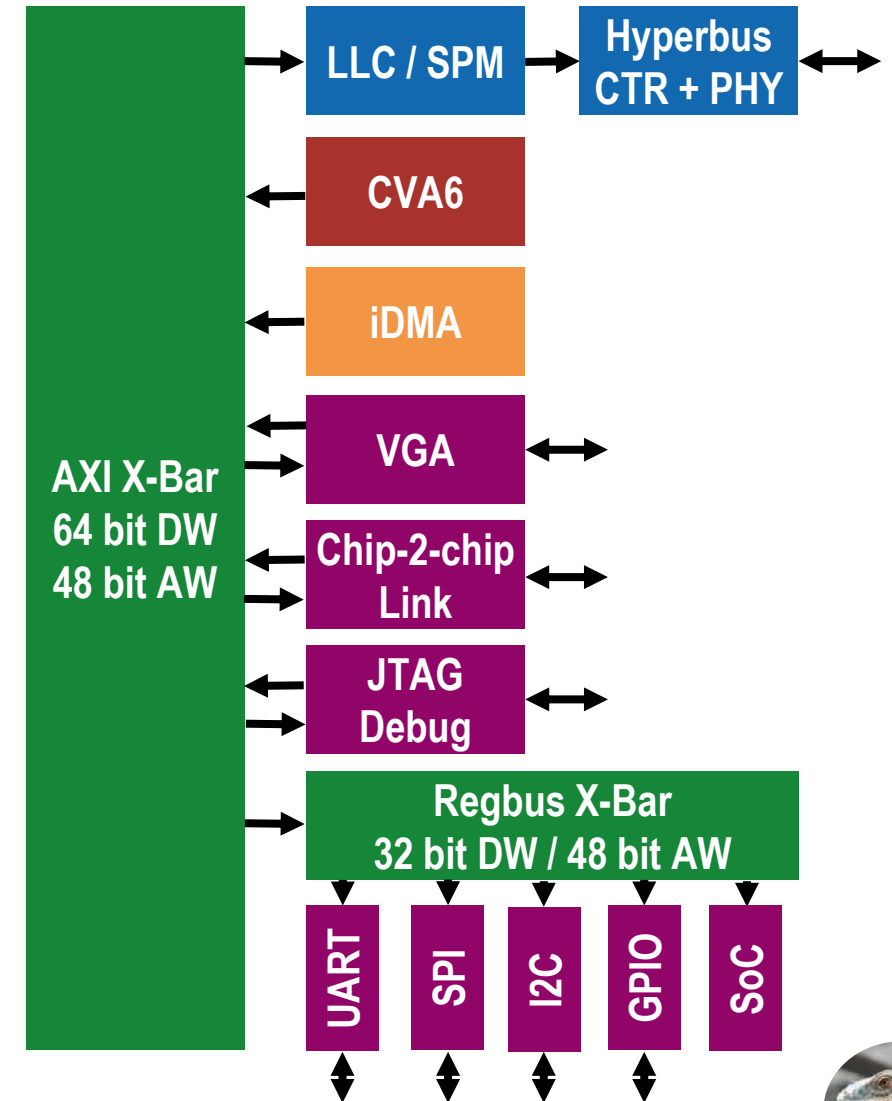


- There are alternatives
 - Chisel, SpinalHDL, Amaranth
- **We use it, a lot**
 - The top language in our repos
 - **1M+ lines** of code
 - All our **systems** in SV
 - Silicon-proven in **50+** PULP tapeouts
 - **Logical choice for Iguana/Tegu/Komodo**
- **Widespread** in industry
 - Many **commercial** IPs in SV
- Works in **proprietary** tools

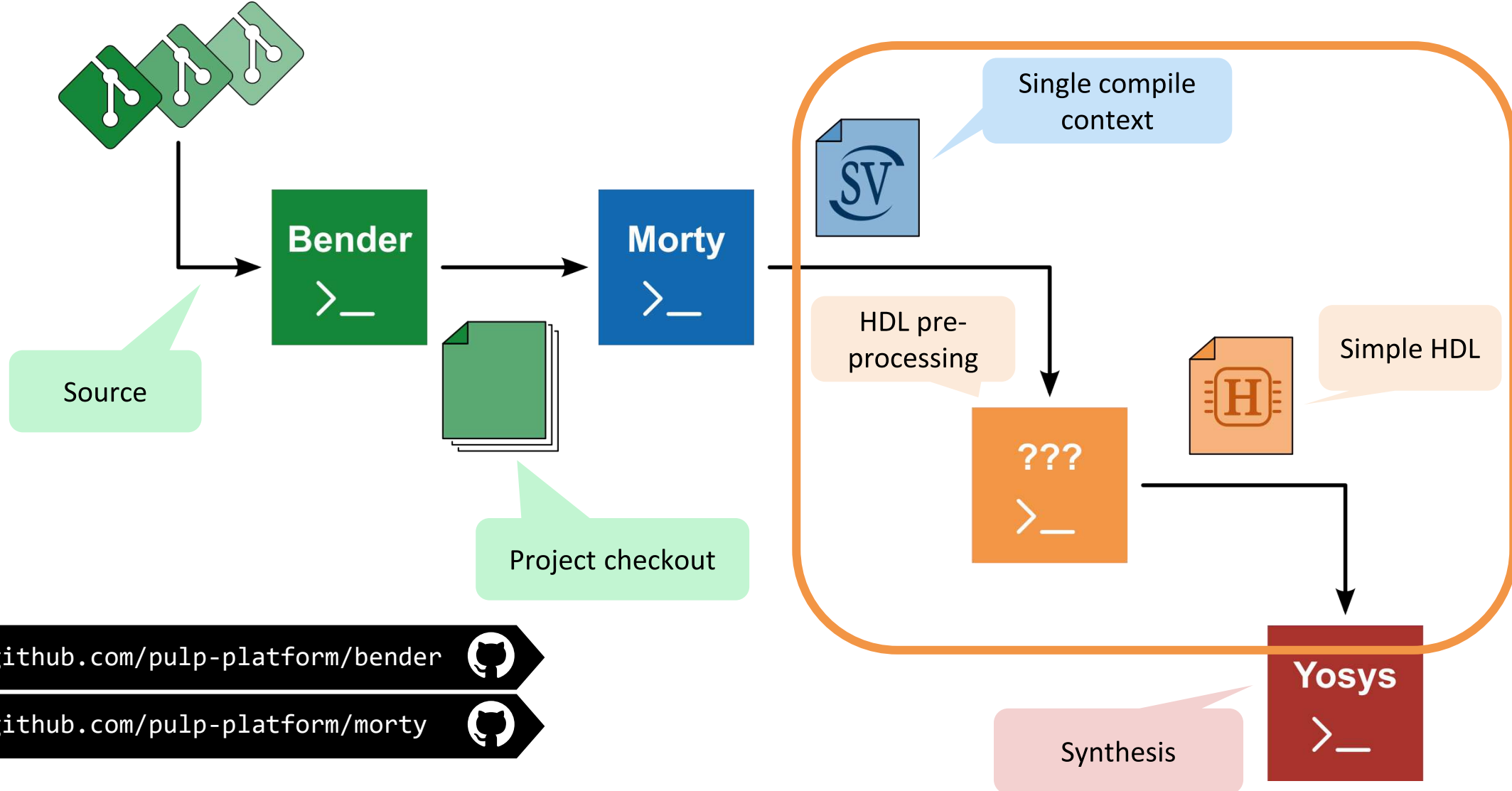


Iguana: A Linux-Capable Design

- **We choose a complex SoC design**
 - Academy & industry: complexity **increases**
 - More **compute** & **complex** interfaces
- **Evaluating the open flow requires complexity**
 - **Sufficient** hardware **complexity**
 - Full Linux-capable **CVA6** SoC: ~2 MGE
 - Arithmetic operands, std-cell memory, ROMs
 - A **capable** demonstrator chip
 - **Multiple** clock domains (HyperBus, C2C link)
 - Implementation & constraining of CDCs
 - **High-speed**, high-complexity async interfaces
 - Synthesis of DDR/asynchronous logic
 - Constraints and proper setup/hold fixings



From Repos To Netlist



github.com/pulp-platform/bender



github.com/pulp-platform/morty

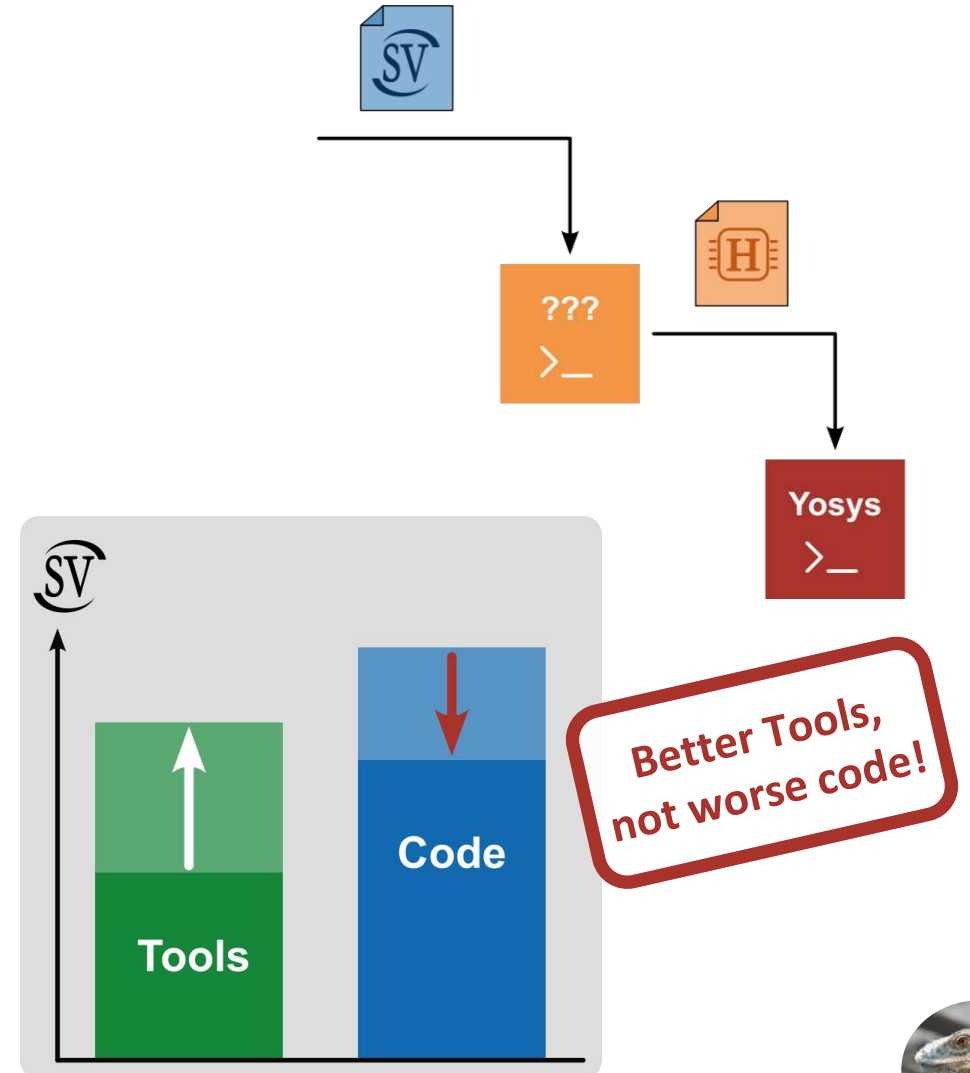


Our Experiences Synthesizing Iguana



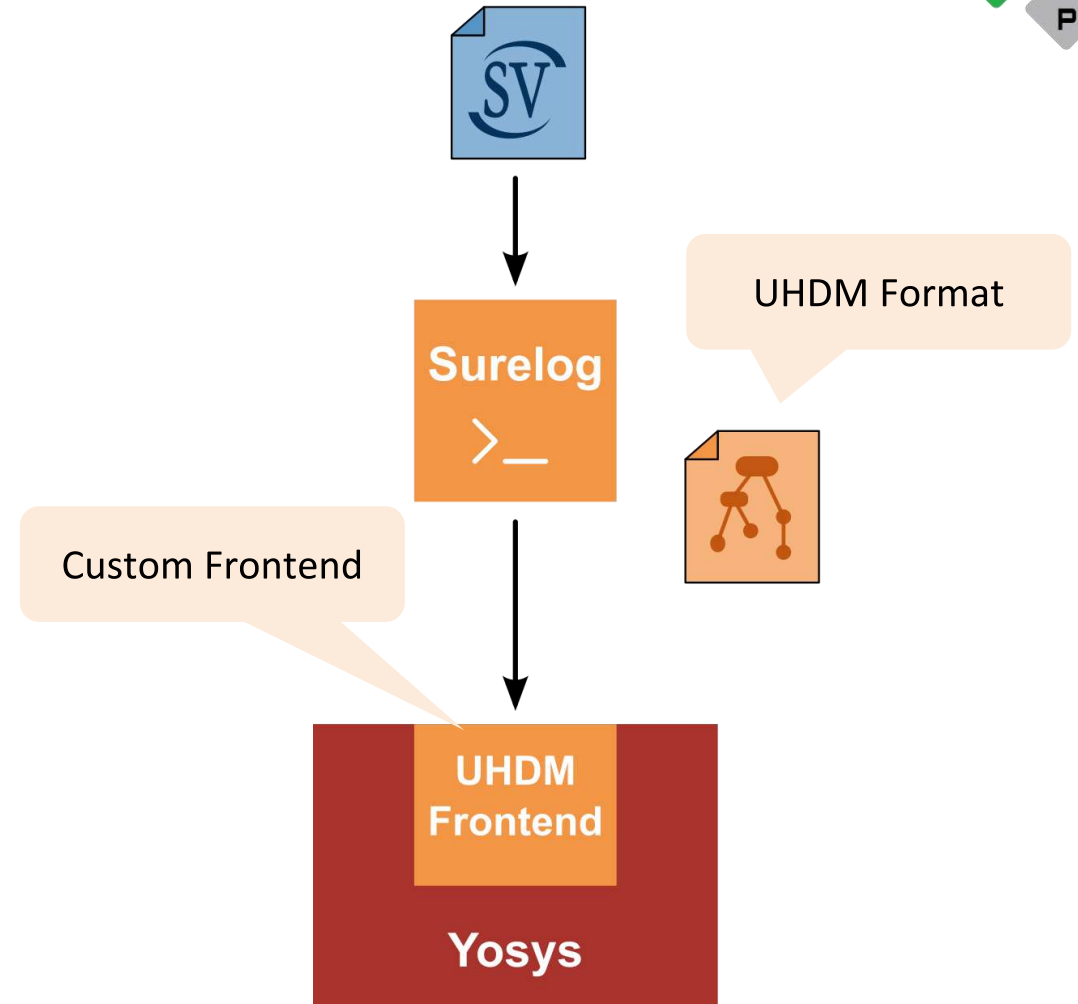
- **First challenge: RTL**

- How to bring SystemVerilog into Yosys?
- PULP codebase is **spec-conform**
- As few manual fixes as possible
 - **Better** tools, not **worse** code!



UHDM-Yosys

- CHIPS Alliance
- Surelog SystemVerilog parser
- Yosys plugin to read UHDM
- UHDM as an intermediate model
 - Similar: LLHD or LLVM IR
 - Simplification possible at UHDM level
 - One parser for all tools



github.com/chipsalliance/surelog



UHDM-Yosys: First Approach

- **Many *loud* errors**
 - Helpful error messages
 - RTL simplification or improving the parser
 - Time consuming



```
ERROR: 2nd expression of procedural for-loop is not constant
ERROR: Couldn't find search elem: resp in struct
ERROR: Failed to detect width of signal access
ERROR: Failed to detect width of memory access
ERROR: Signal `Exp_a_D' with non-constant width
ERROR: Latch inferred for signal ... from always_comb process
ERROR: Failed to detect width for identifier
ERROR: Failed to resolve identifier \EQ for width detection
ERROR: Invalid bit-select on memory access
ERROR: Don't know how to detect sign and width for ...
ERROR: Don't know how to detect sign and width for ...
ERROR: Assert 'children[0]->type == AST_WIRETYPE' failed in ...
ERROR: Unsupported node type: AST_WIRE
ERROR: enum item already exists
....
Segmentation Fault
....
```

UHDM-Yosys: First Approach

- **Many *loud* errors**
 - Helpful error messages
 - RTL simplification or improving the parser
 - Time consuming
- **More severe problems appeared**
 - Scope resolution issues



```
package base_pkg;  
    localparam logic BASE = 1'b0;  
    localparam logic USED = BASE | 1'b0;  
endpackage
```

```
package error_pkg;
```

Error!

```
    localparam logic DERIVED = base_pkg::USED;  
endpackage
```

```
module top import error_pkg::*; #() ();  
    // ERROR: BASE implicitly declared?  
endmodule
```

UHDM-Yosys: First Approach

- **Many *loud* errors**
 - Helpful error messages
 - RTL simplification or improving the parser
 - Time consuming
- **More severe problems appeared**
 - Scope resolution issues
 - **Silent errors: extremely** hard to detect
 - UHDM is **hard to verify**
 - Fix **Surelog** or **SV-Plugin**?

Too much effort
for all PULP IPs



```
package base_pkg;
  localparam logic BASE = 1'b0;
  localparam logic USED = BASE | 1'b0;
endpackage

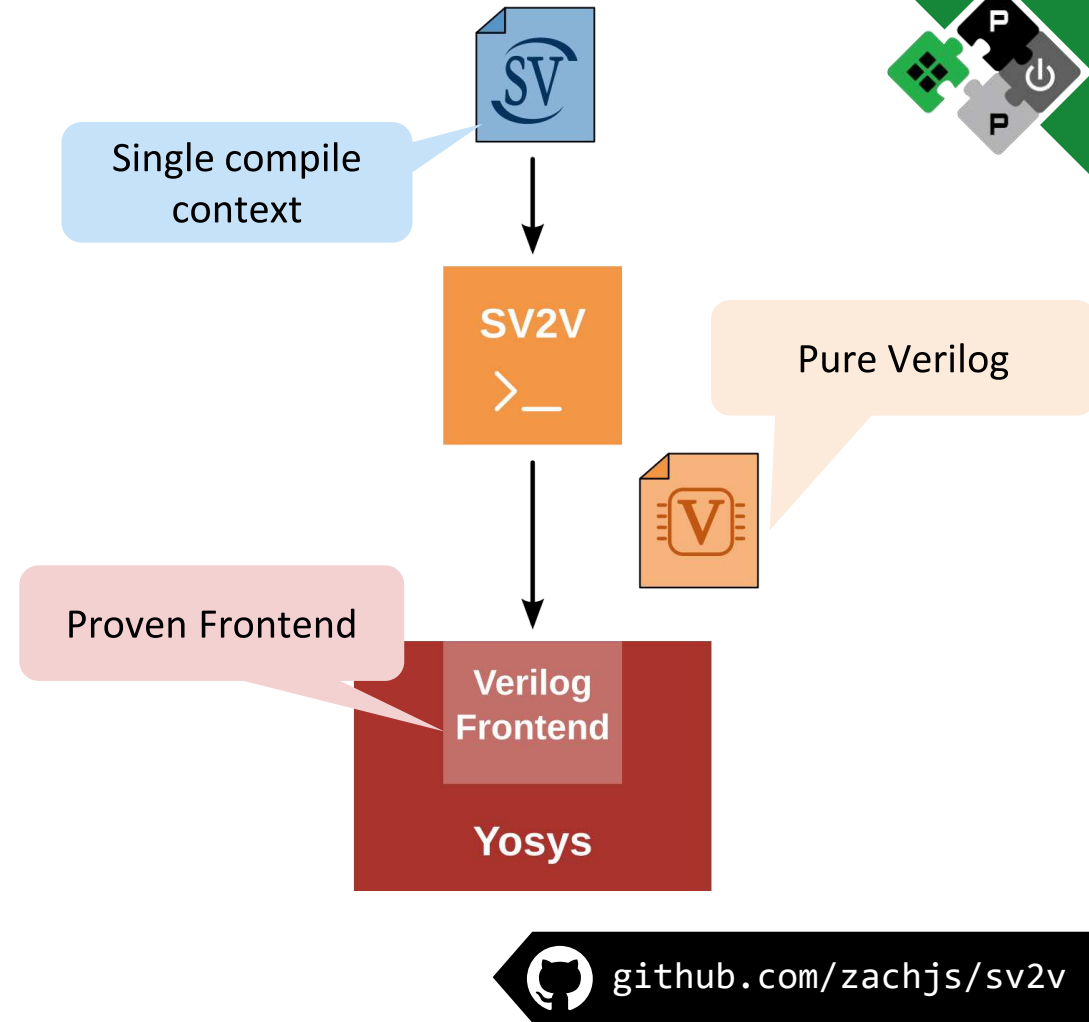
package error_pkg;
  localparam logic BASE = 1'b1;
  localparam logic DERIVED = base_pkg::USED;
Endpackage           // = BASE | 1'b0;

module top import error_pkg::*; #() ();
  // ERROR: DERIVED is 1'b1 (from BASE)!
endmodule
```

Silent error

SV2V: Second Approach

- **SV2V**
 - Translation to Verilog 2005
 - Yosys can read the output
 - Symbolic resolution of the parameters
- **Does not support the entire SV spec**
 - Helpful error messages
 - Haskell requires niche knowledge to extend
- **Symbolic parameter resolution**
 - Very difficult for humans to comprehend
 - Emitted code is hard to simulate/trace

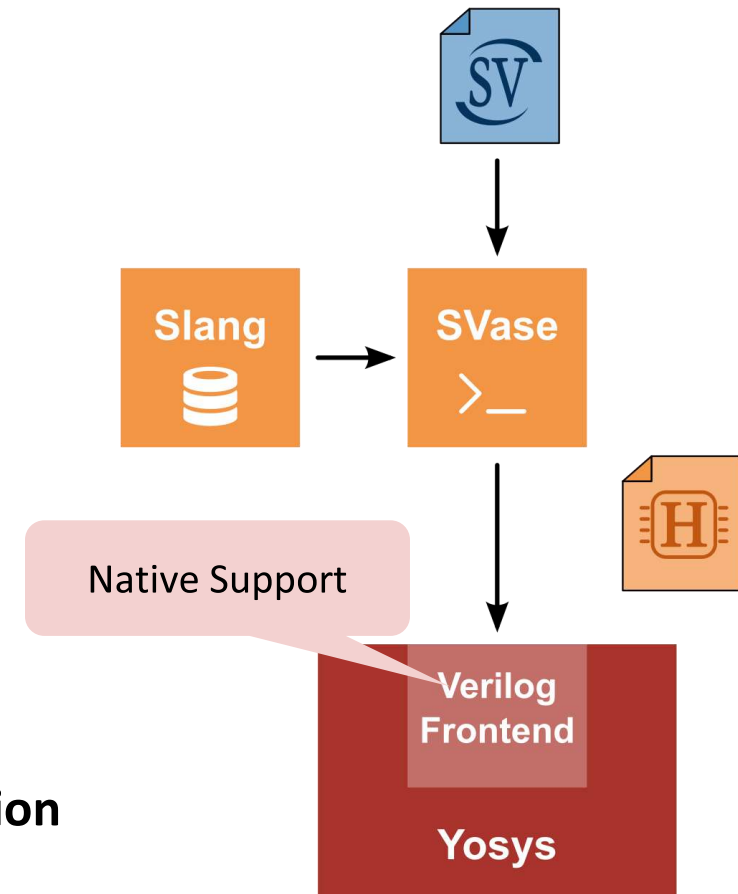


```
localparam [31:0] NumLevels = $unsigned($clog2(NumIn));  
wire [(((2 ** NumLevels) - 2) >= 0 ? (((2 ** NumLevels) - 1) * 91) - 1 : ((3 - (2 ** NumLevels)) * 91) +  
      (((2 ** NumLevels) - 2) * 91) - 1)):(((2 ** NumLevels) - 2) >= 0 ? 0 : ((2 ** NumLevels) - 2) * 91)] data_nodes;
```



SVase: Third Approach

- **Slang has a perfect SV score**
 - Can parse and **elaborate** the code
 - Spec-conform, **excellent linter**, rock-stable, **very fast**
 - So far: Not much is done with the elaborated AST
- **Idea: We simplify SV using Slang: *SVase***
 - Can parse and elaborate the **entire** PULP codebase
 - Only simplify **unsupported** constructs -> Simple SV
 - **Uniquification, generate** resolution, parameter **elaboration**
 - Simplification is implemented in **passes**
 - Easy-to-extend (C++) **framework**

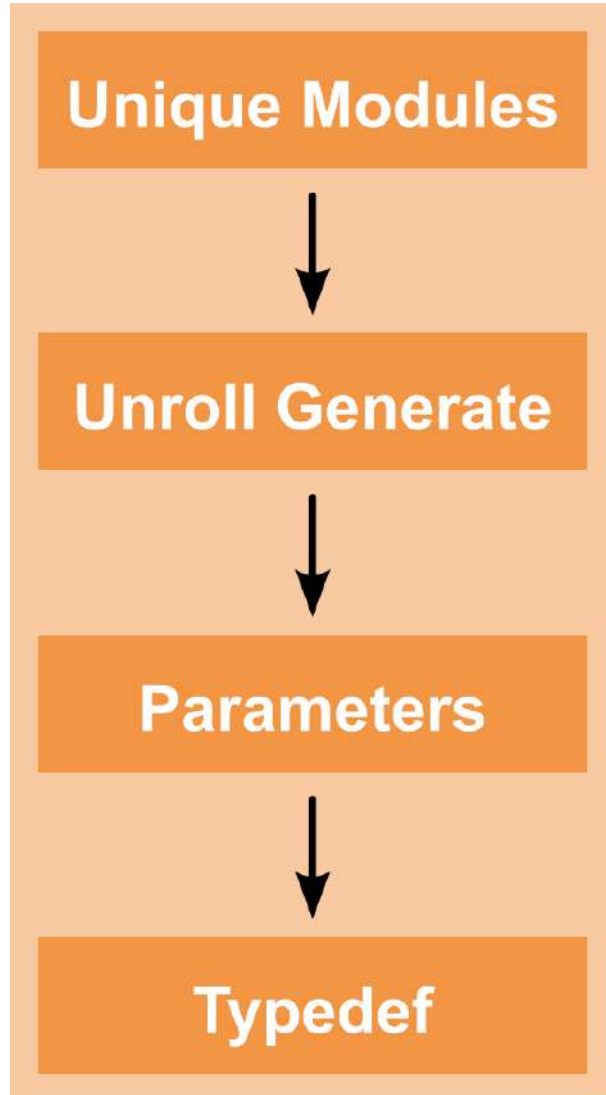


 github.com/MikePopoloski/slang

```
localparam [31:0] NumLevels = $unsigned($clog2(NumIn));
localparam int NumLevels = 4;
logic [81:0] data_nodes;
(((2 ** NumLevels) - 2) * 91) - 1):(((2 ** NumLevels) - 2) >= 0 ? 0 : ((2 ** NumLevels) - 2) * 91] data_nodes;
```



SVase: Example

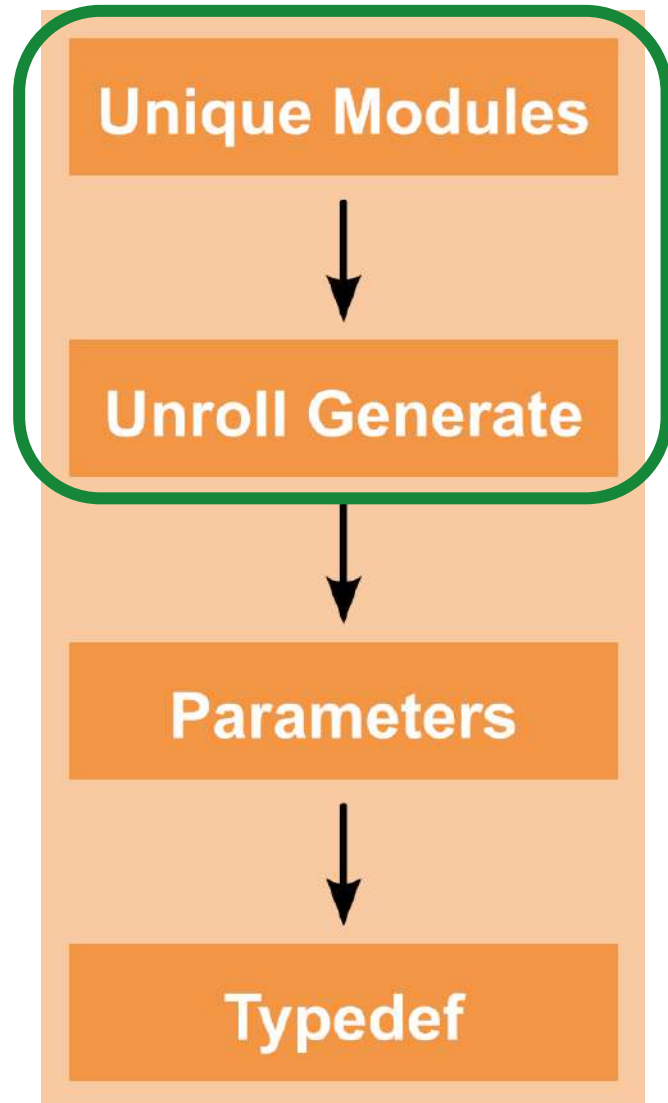


```
module test #(  
    parameter int NoDefaultParam,  
    parameter type PortTypeParam = logic  
) ();  
endmodule
```

```
module top # ( ) ();  
    localparam int TopParam = 32'd5;  
    localparam type TypeParam = struct packed {...};  
  
    test #(  
        .NoDefaultParam(TopParam),  
        .PortTypeParam(TypeParam)  
    ) i_test ( );  
endmodule
```



SVase: Example



```
module test__1699695787177254841 #(
    // unique parametrization

) ();
endmodule

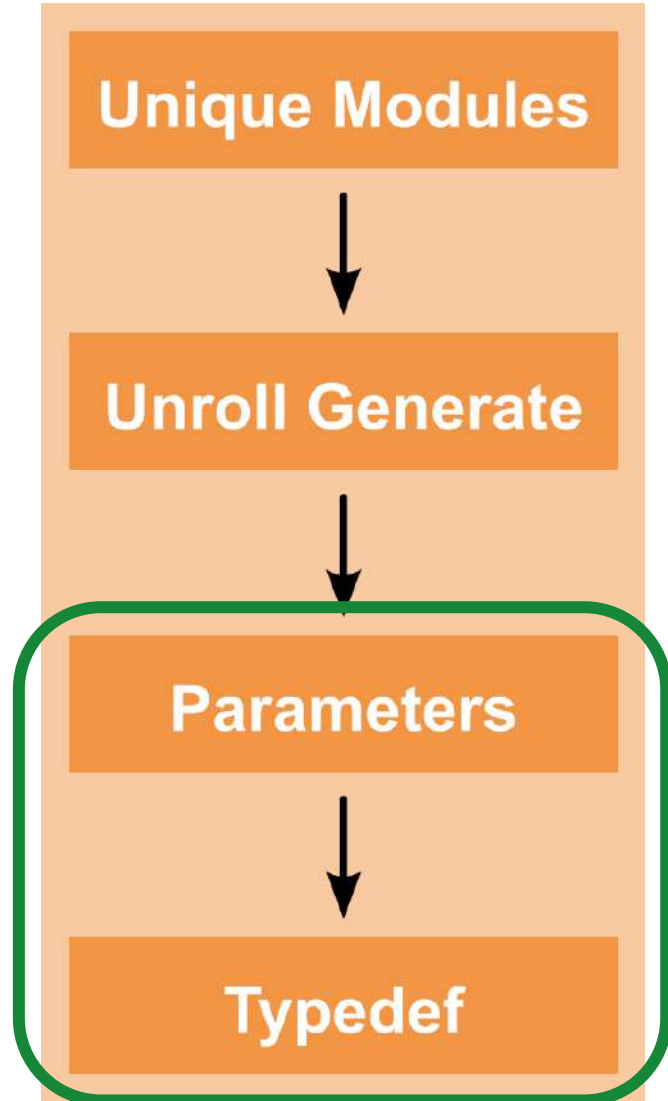
module top__6142509188972423790 #( ) ();
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

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

A large white curved arrow on the right side of the slide points from the `test__1699695787177254841` module instance in the second module to the `test__1699695787177254841` module instance in the first module, indicating a hierarchical or instantiation relationship.



SVase: Example



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

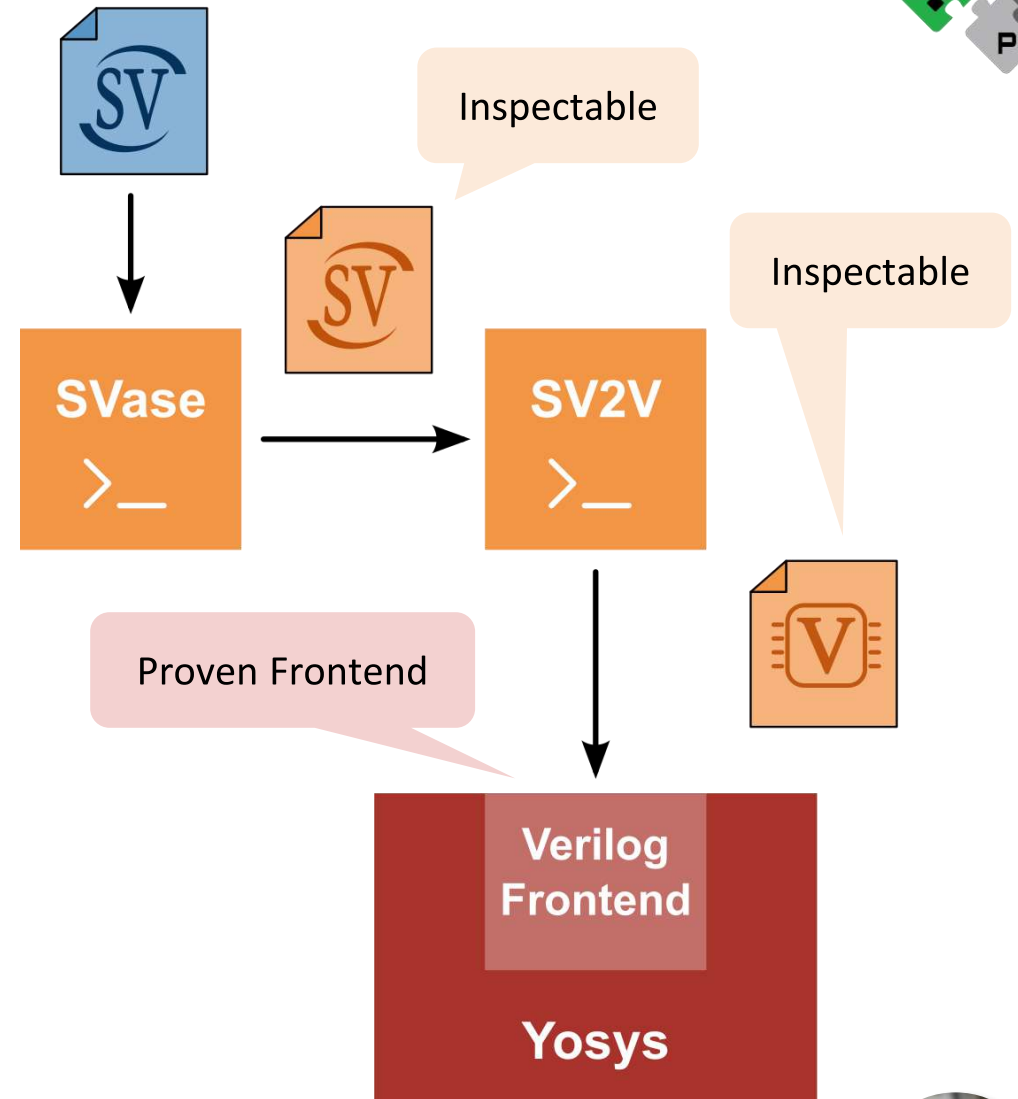
```
module top__6142509188972423790 #( ) ();
    localparam int TopParam = 32'd5;
    localparam type TypeParam = struct packed {...};

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



Combination: Final Approach

- **SVase is not completed yet**
 - Question of time: Implement **complex** steps
 - **SV2V** is currently required to translate the rest
 - We are open for **collaboration**!
- **Ideally:**
 - Integrate Slang as a **plugin** into Yosys
 - **Native** SystemVerilog support



Our Experiences Synthesizing Iguana



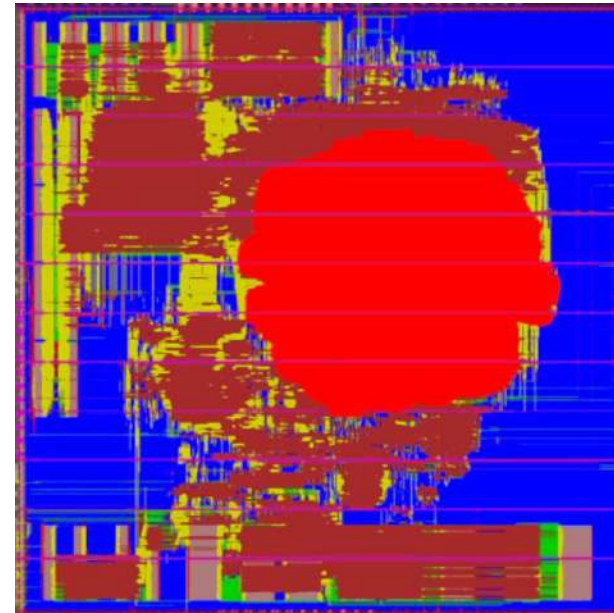
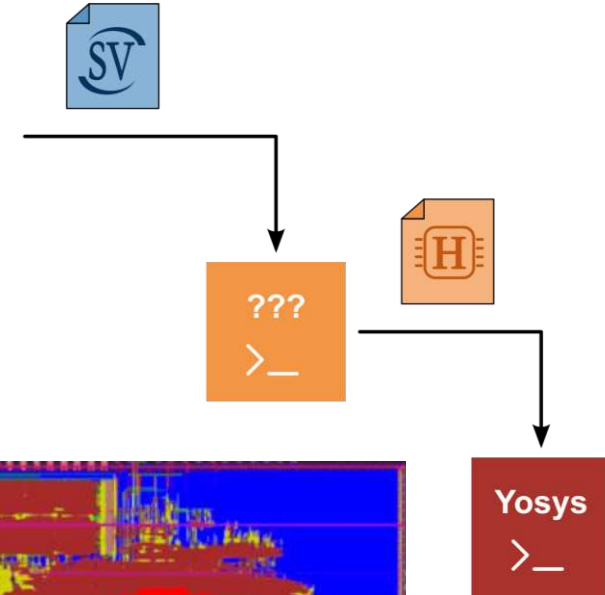
- **First challenge: RTL**

- How to bring SystemVerilog into Yosys?
- PULP codebase is spec-conform
- As few manual fixes as possible
 - **Better tools, not worse code!**

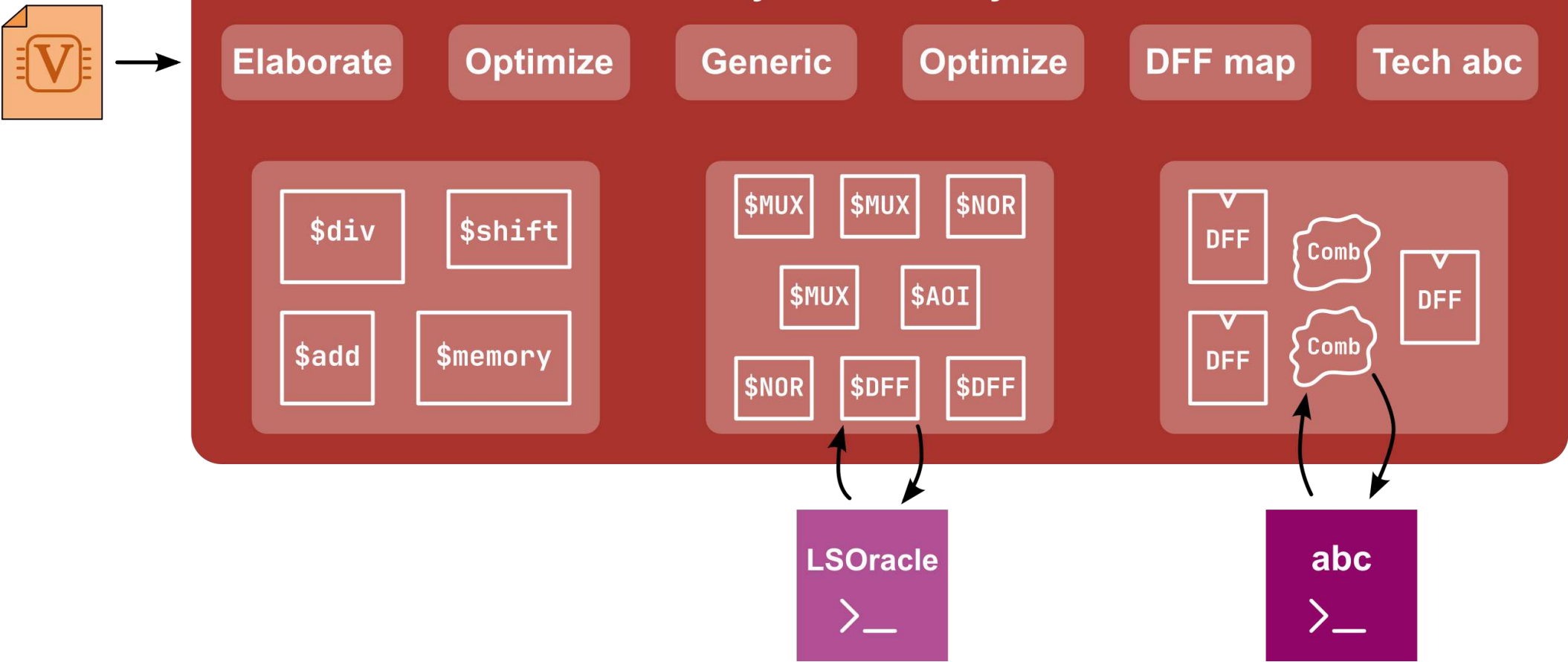
Solved!

- **Second challenge: Synthesis**

- Yosys struggles with **some modules**
 - **High** memory footprint, **long** synthesis time
 - Reduced quality of results
- **Early-on** understanding of **timing** is missing



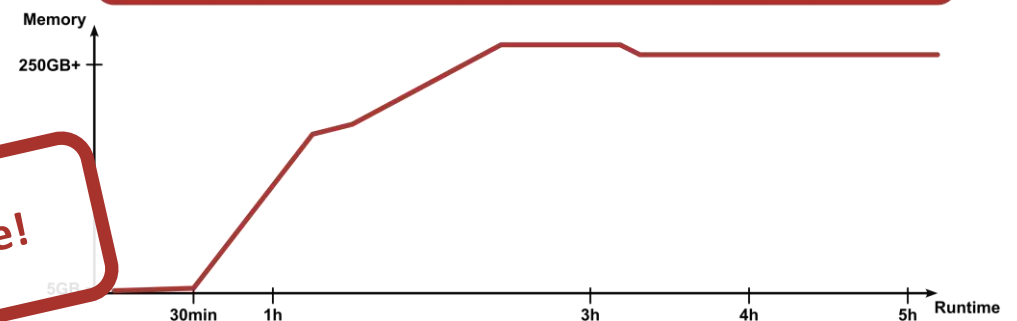
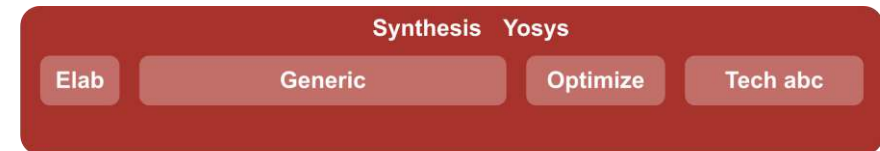
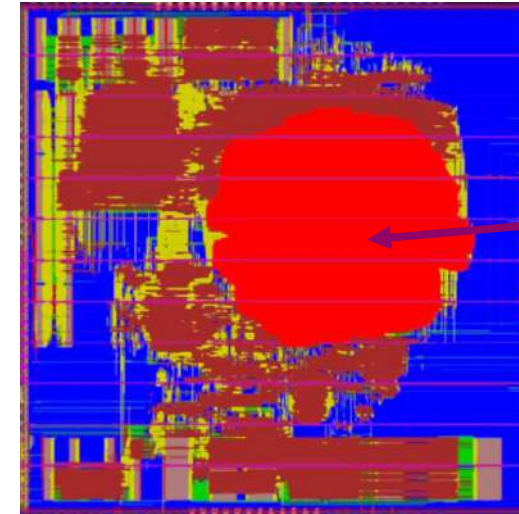
Yosys Synthesis



There Is Something Going Wrong



- A few modules behave weirdly
- Area/timing suboptimal
 - Unexpectedly **large** for their logic complexity
 - Compared to rest of modules: **very long** paths
 - Looks like a **cancer** growing on Iguana
- Explosion in required compute resources
 - Scoreboard alone: **>300 GiB** RAM required
 - Yosys keeps **all** synthesis state **in memory**
 - Very **long** synthesis time



Very hard to manage!



Two Problems

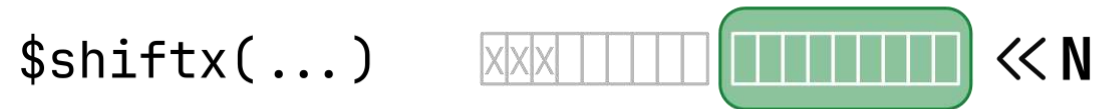


- **Elaboration**

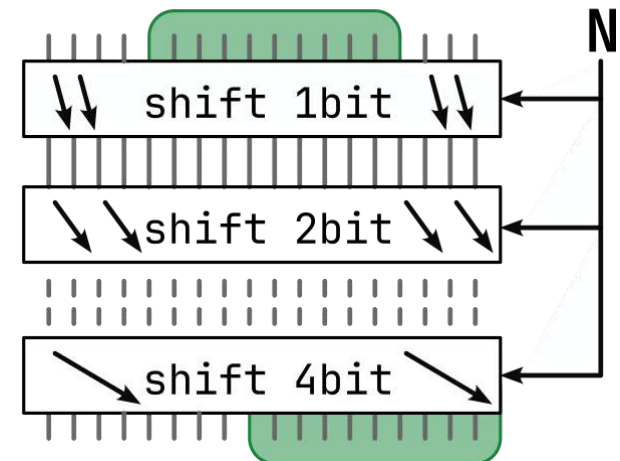
- At this stage: **no timing** information
- Globally **fixed** implementations
- Arithmetic units are **slow**

- **Part-select issue (#3833)**

- CVA6 scoreboard
- Structs not POW2 data types
- At elaboration: infer \$shiftx
- Shift by **N** (keep it generic)
 - Implemented by **log shifter** and select
 - Stages cannot be optimized
- **Fix: Detect this fixed-stride case**



\$shiftx:



Conclusion

- **Yosys can synthesize Iguana**
 - SVase + SV2V RTL pre-processing
 - Iguana is a **complex pipe cleaner** for the flow
- **Yosys can be improved for complex designs**
 - Slang-Yosys plugin or frontend
 - Full **SystemVerilog** support
 - **Timing** information during elaboration
 - More **variants** to select
 - *Library of Arithmetic Units?*
 - **Efficient** part-select
 - More **manageable** memory footprint
 - Option to store temporary synthesis data on **disk**

guest.iis.ee.ethz.ch/~zimmi/arith_lib.html



Collaboration

- **Design and flow will be open ASAP**
 - Some **cleanup** required (SRAM, IOs)
 - A **testcase** for many open tools
 - We can increase the **complexity** if required
- **We are open for collaboration**
 - **SVase** extension
 - Improvements to **Yosys**
 - **OpenRoad**
 - STA & constraints / Timing fixing / DRV fixing
 - **Verification**
 - Logic equivalency check / **post-layout simulation** / DFT



github.com/pulp-platform/iguana



Thomas Benz
Paul Scheffler
Jannis Schönleber
Philippe Sauter

tbenz@iis.ee.ethz.ch
paulsc@iis.ee.ethz.ch
janniss@iis.ee.ethz.ch
phsauter@ethz.ch

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Università di Bologna
Viale del Risorgimento 2
Bologna, Italy



github.com/pulp-platform/iguana



@pulp_platform



pulp-platform.org



youtube.com/pulp_platform

