



Formal Verification Adoption Made Easy

DVClub World Online 2025

Matt Venn, YosysHQ

Topics for today

- Who is YosysHQ?
- Sby is a free, open source formal verification tool
- Demonstration
- GitHub actions with oss-cad-suite
- Formal Verification on every push
- Limitations, Tabby CAD suite
- Where to get more information and a free evaluation license

YosysHQ – The place for Open-Source EDA

Synthesis

Place-and-Route

Verification

- Home of the creators and maintainers of Yosys, nextpnr, and related tools.
 - We push the boundaries of EDA in interesting and unexpected ways.
- Our mission is to increase access to state-of-the-art EDA design and (formal) verification tools.
 - One bundle with all our tools, competitively priced (starting at 500 EUR per month)
 - Integrated with industry-proven SystemVerilog/VHDL parser.
- Support and Project Work
 - Get support for our tools directly from the team that wrote them.
 - Get your custom features developed by the team most knowledgeable with the code base.

[Installation guide](#)[Getting started](#)[Using *sby*](#)[Reference for .sby file format](#)[Autotune: Automatic Engine Selection](#)[Formal extensions to Verilog](#)[SystemVerilog, VHDL, SVA](#)

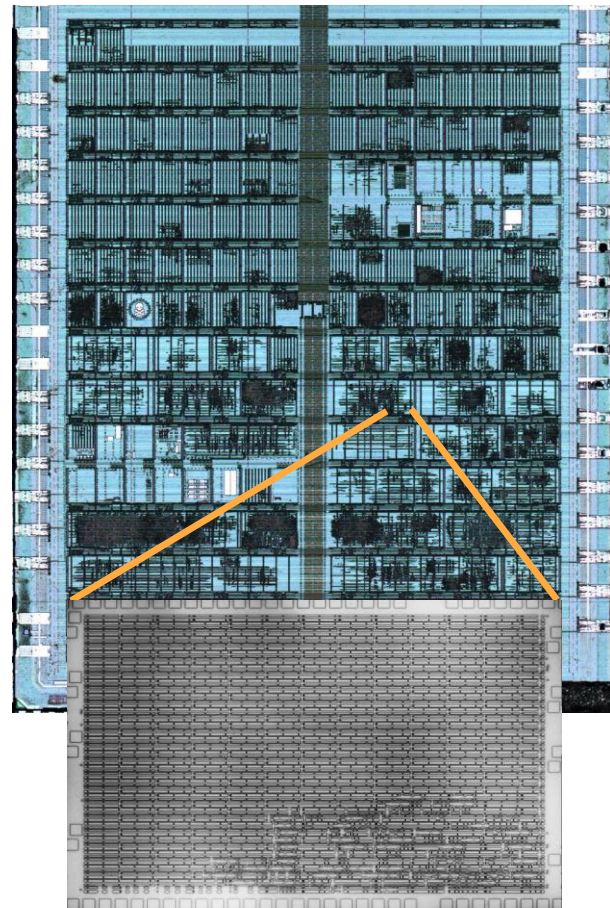
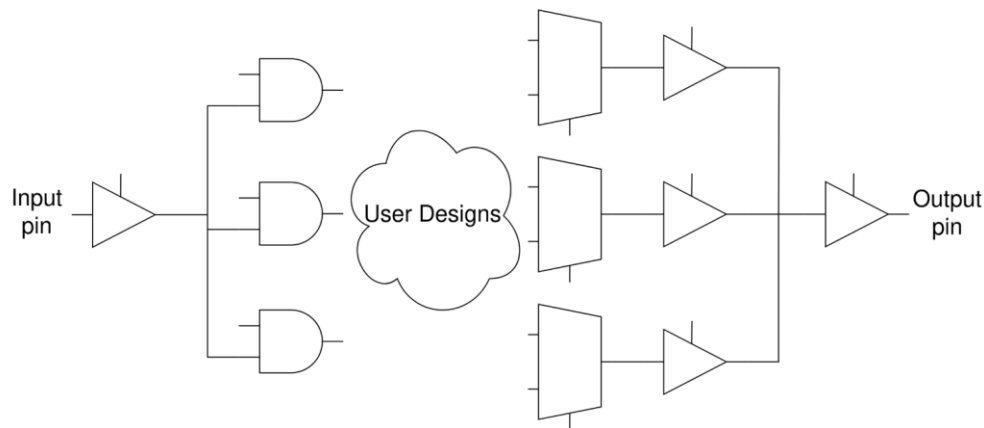
SymbiYosys (sby) Documentation

SymbiYosys (sby) is a front-end driver program for Yosys-based formal hardware verification flows. SymbiYosys provides flows for the following formal tasks:

- Bounded verification of safety properties (assertions)
- Unbounded verification of safety properties
- Generation of test benches from cover statements
- Verification of liveness properties

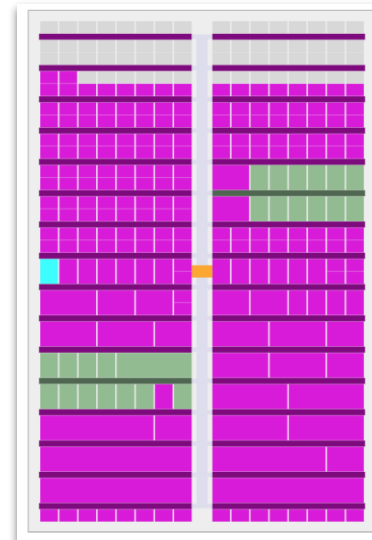
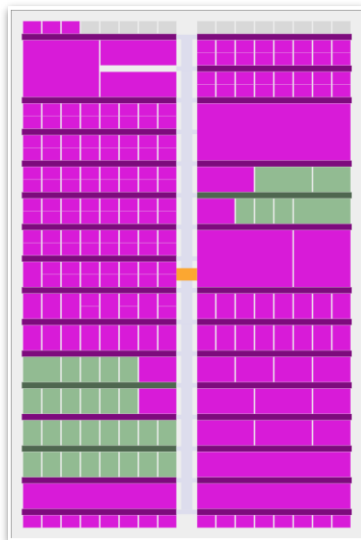
Tiny Tapeout

- Supports 512 designs
- Digital, mixed signal, [analog](#)
- Every design is power gated
- Multi tile support for bigger designs
- [Read the IEEE paper](#)



ASIC educational project, must work 1st time

- [10+ chips](#)
- 30+ countries
- 50+ universities
- 100+ [silicon proven](#)
- 1500+ designs
- 5000+ people



Formal connectivity

```
module tt_um_formal (
    input wire [7:0] ui_in,      // Dedicated inputs
    output wire [7:0] uo_out,    // Dedicated outputs
    input wire [7:0] uio_in,     // IOs: Input path
    output wire [7:0] uio_out,   // IOs: Output path
    output wire [7:0] uio_oe,    // IOs: Enable path (active high: 0)
    input wire ena,
    input wire clk,
    input wire rst_n
);

// let solver drive the outputs
rand reg [7:0] anyseq1; assign uo_out = anyseq1;

always @(*) begin
    if(ena) begin
        // if design is enabled, looped back inputs must = outputs
        assert(ui_in == uo_out);

    end else begin
        // otherwise inputs must be 0
        assert(ui_in == 0);

        // design is in reset
        assert(rst_n == 0);

        // no clock
        assert(clk == 0);

    end
end

endmodule // tt_um_formal
```

<https://github.com/mattvenn/tt-multiplexer/tree/yosyshq-demo>

```
// loop back dedicated outs to ins
assign { io_in[13], io_in[6:0] } = io_out[31:24];

// loop back bidirectional outs to ins, depending on output enable
assign io_in[23:16] = io_out[23:16] & (~io_oe[23:16]);
```

GitHub action

```
matt-office:2065 [yosyshq-demo]: cat .github/workflows/formal.yaml
name: formal
# either manually started, or on a schedule
on: [ push, workflow_dispatch, pull_request ]
jobs:
  formal:
    # ubuntu
    runs-on: ubuntu-latest
    steps:
      # need the repo checked out
      - name: checkout repo
        uses: actions/checkout@v3

      # install oss fpga tools for cocotb and iverilog
      - name: install oss-cad-suite
        uses: YosysHQ/setup-oss-cad-suite@v2
        with:
          github-token: ${ secrets.GITHUB_TOKEN }

      - run: |
          yosys --version

      - name: formal connectivity
        run: |
          cd formal
          sby -f tt_connectivity.sby
```

Run on push, PR

Checkout the repo

Install oss-cad-suite

Run the formal proof

Results

formal

succeeded 46 minutes ago in 59s

Q Search logs

↺

⚙

>

✓

Set up job

0s

>

✓

checkout repo

2s

>

✓

install oss-cad-suite

17s

>

✓

Run yosys --version

0s

>

✓

formal connectivity

37s

>

✓

Post checkout repo

0s

>

✓

Complete job

0s

Tabby CAD Suite

- **Contains Proprietary Code**
- **Industrial-strength Language Support**
 - **Verilog 1995/2001/2005**
 - **SystemVerilog 2005/2009/2012**
 - **VHDL 87/93/2000/2008/2019**
- **Comprehensive formal SVA support**
- **Commercial Support, Training, and Services**
- **Custom builds to fit customer needs**
- **Tested stable releases**

OSS CAD Suite

- **Pure Open Source Software**
- **Open Source Verilog Parser**
- **Immediate assume/assert**
- **Support via Community Slack**
- **Bug Reports via GitHub Issues**
- **Nightly builds (untested)**

For more information...

- Docs: yosyshq.readthedocs.io/en/latest/
- Eval license: www.yosyshq.com/contact
- Pricing: www.yosyshq.com/pricing
 - From €500 PCM for freelancers, €800 otherwise
 - 32 cores, as many users as you want
- Join our newsletter blog.yosyshq.com/newsletter/
- We're hiring! www.yosyshq.com/jobs