SCIENCE
PASSION
TECHNOLOGY

Lightweight Block-Level Formal Verification

A Graphical Guide

Meinhard Kissich

ORConf 2023

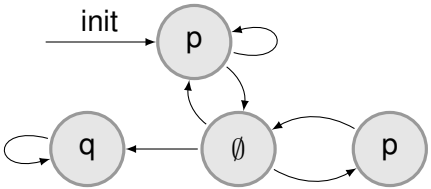
Meinhard Kissich

Location PhD Student @ Graz University of Technology

Interests FPGA architectures
FPGA backend flows
Applied formal verification

My Initial View on Formal Verification

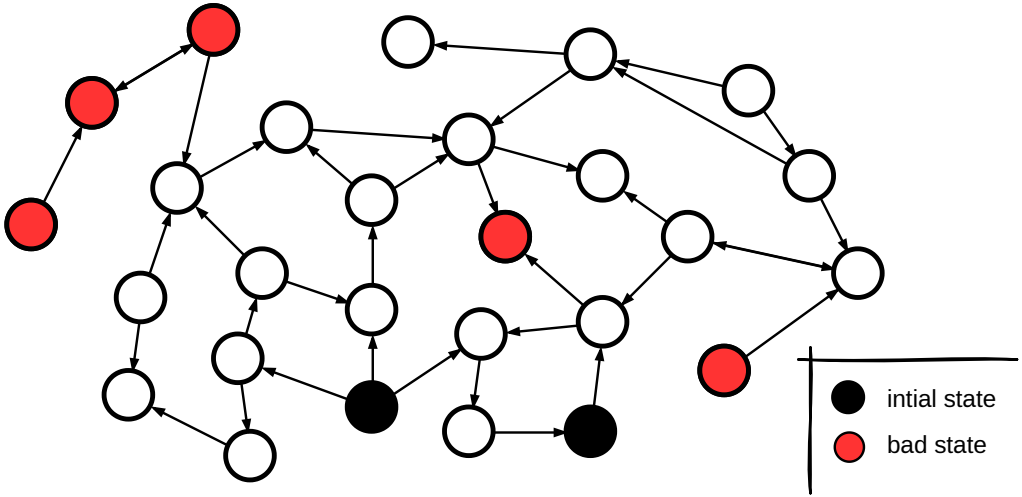
- **Kripke structure**
- $M = (S, S_0, R, AP, L)$
- $M, s \overset{?}{\models} \square(p \rightarrow \neg \bigcirc q)$
- ...



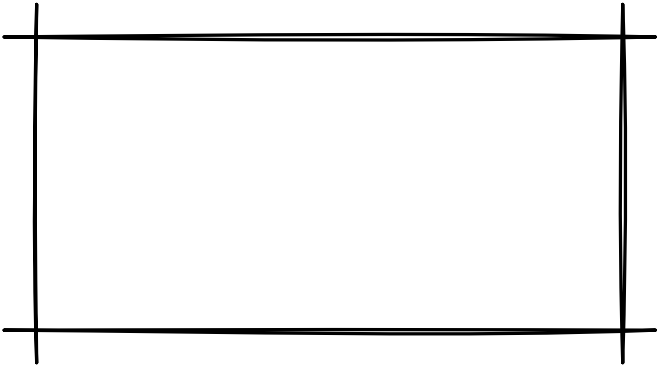
"not practical" $\longrightarrow$ preferred verification method for *suitable* designs

What is this presentation (not) about?

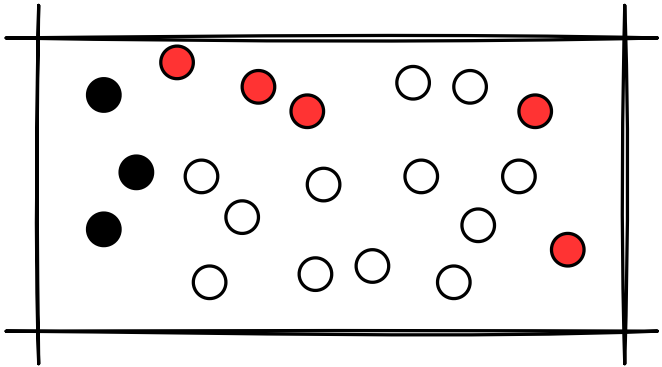
- ⊕ Practical application
- ⊕ Graphical
- ⊕ Designer's guide
- ⊖ Theoretical background
- ⊖ Tools, languages, ...



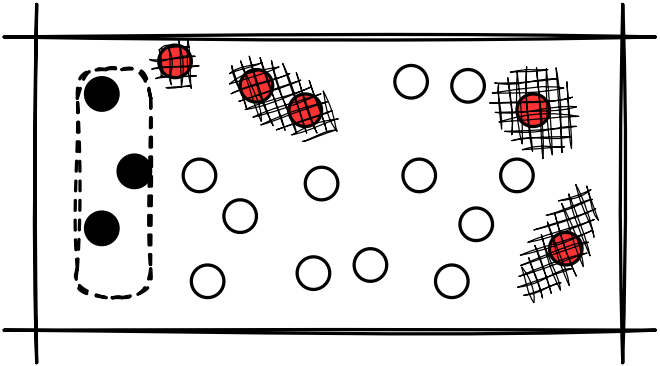
See also [12, 13]



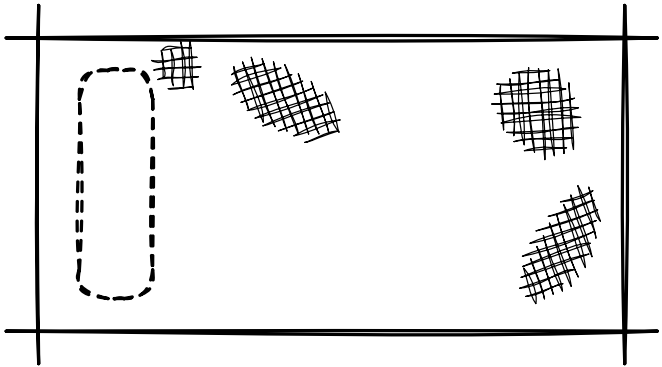
Partly adapted from [8]



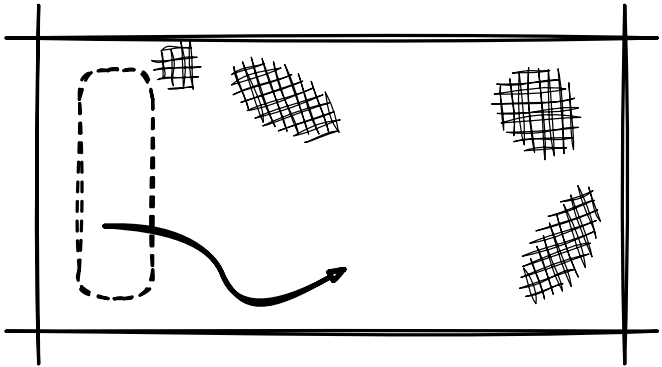
Partly adapted from [8]



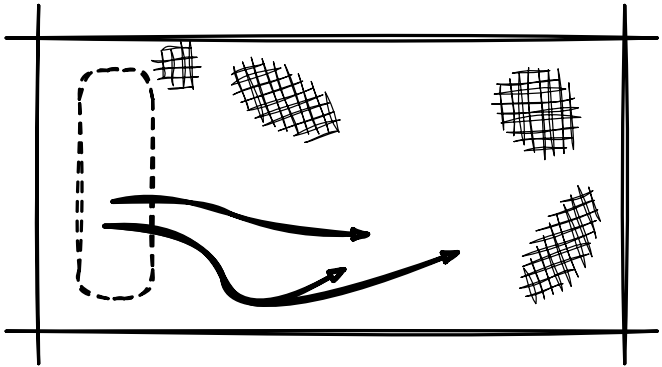
Partly adapted from [8]



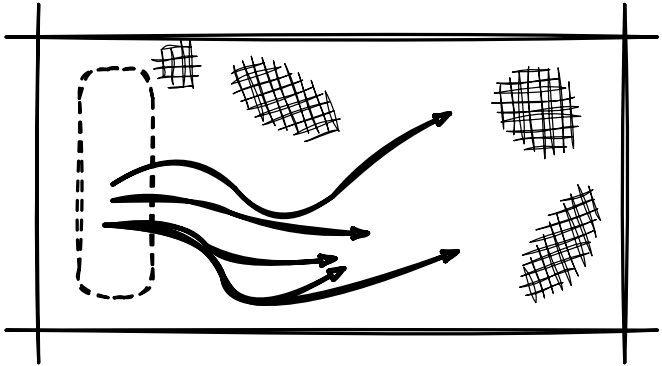
Partly adapted from [8]



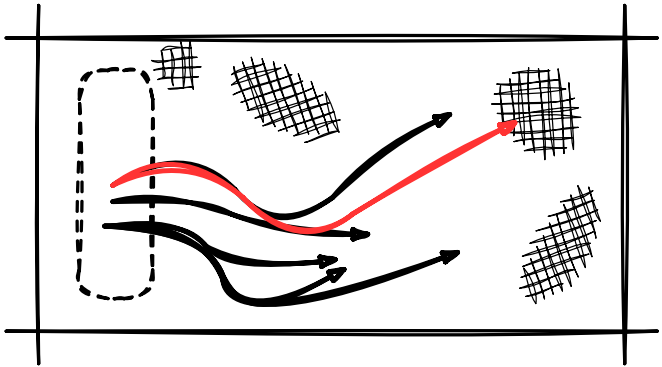
Partly adapted from [8]



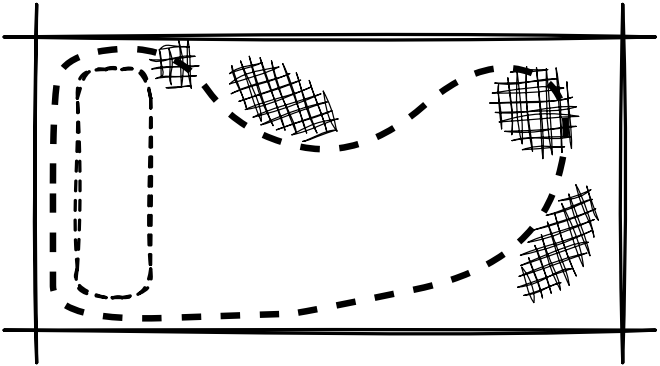
Partly adapted from [8]



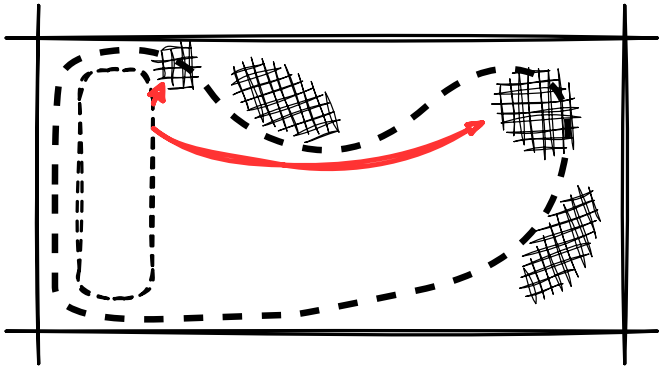
Partly adapted from [8]



Partly adapted from [8]



Partly adapted from [8]

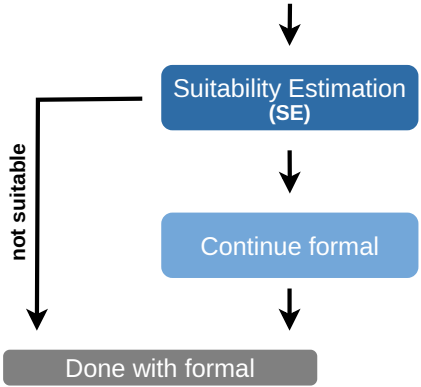


Partly adapted from [8]

Methodology [7]

- For designers
- Formalize the endeavor
- Concise
- Lightweight

Design Suitable for Formal?



Suitability Estimation (SE)

- ⊕ Control-oriented designs
- ⊕ Data transportation
- ⊖ (Deep) sequential designs
- ⊖ Data transformation

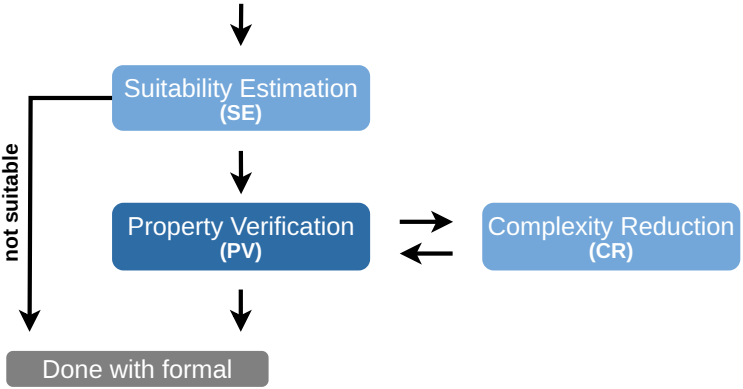
Based on [2, 3]

Formal Friendly Designs

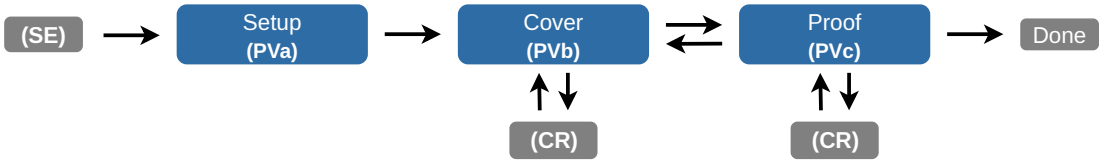
- ⊕ Arbiters, scheduler
- ⊕ DMA controller
- ⊕ Interrupt controller
- ⊖ Floating point unit
- ⊖ Convolution unit
- ⊖ Graphics shader

Based on [2, 3]

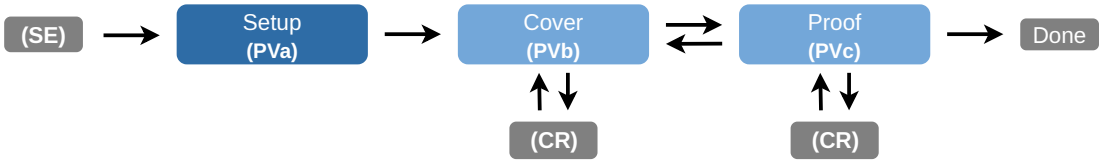
Continue Formal Endeavor



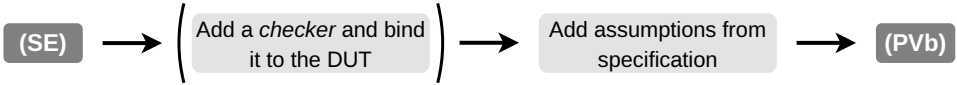
Property Verification (PV)



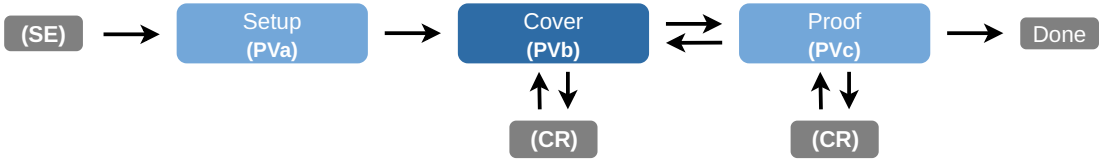
Property Verification (PV)

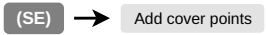


Setup (PVa)

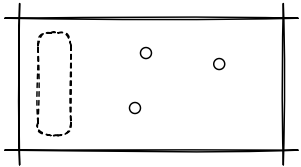

(SE)
Add a *checker* and bind it to the DUT
Add assumptions from specification
(PVb)

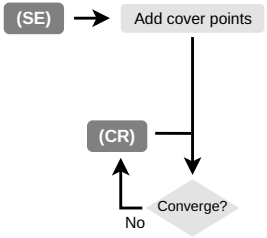
Property Verification (PV)

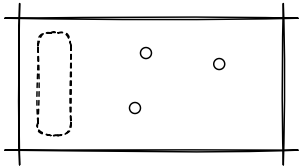


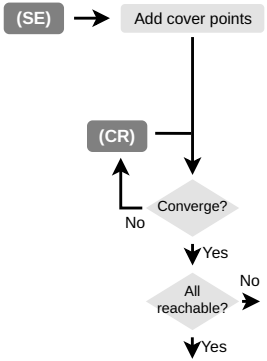

(SE)
Add cover points

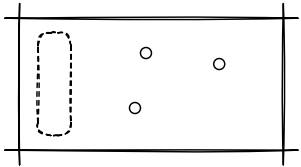
(SE) → Add cover points

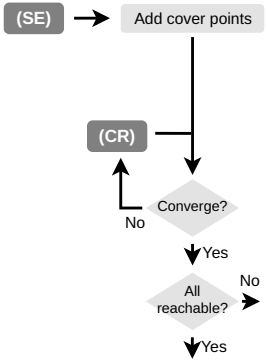


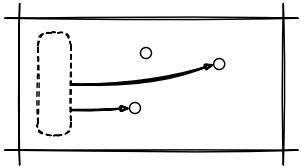

(SE)
Add cover points
(CR)
Converge?
No

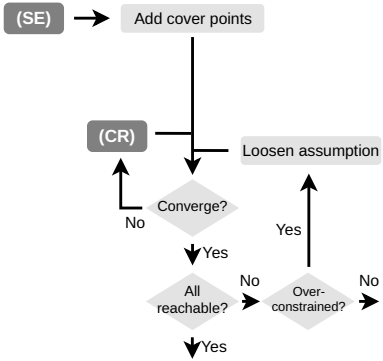


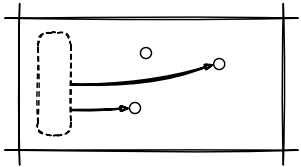

(SE)
Add cover points
(CR)
Converge?
No
Yes
All reachable?
No
Yes

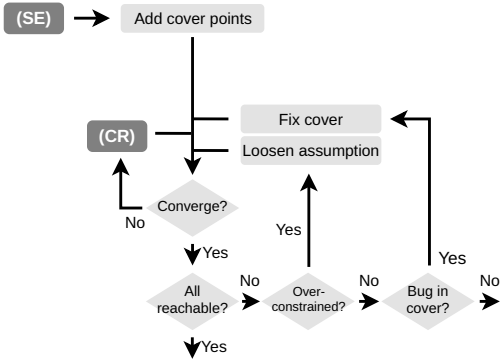


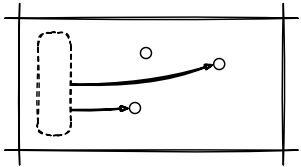

(SE)
Add cover points
(CR)
Converge?
No
Yes
All reachable?
No
Yes

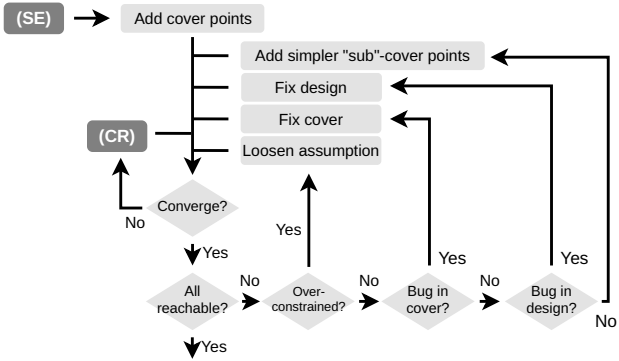


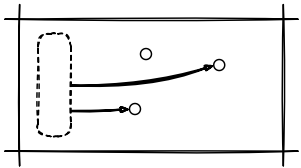

(SE)
Add cover points
(CR)
Loosen assumption
Converge?
No
Yes
All reachable?
No
Over-constrained?
No
Yes
Yes

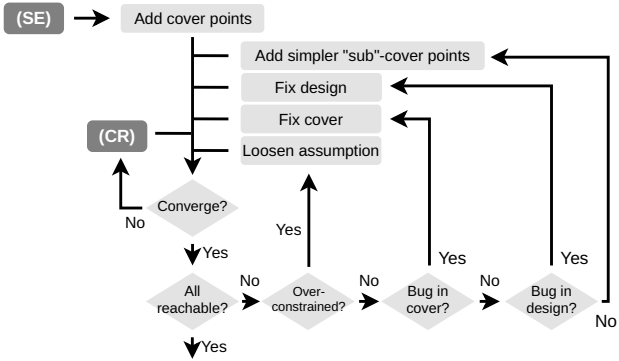


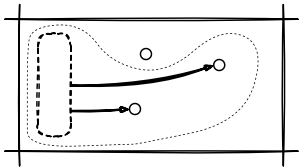

(SE)
Add cover points
(CR)
Fix cover
Loosen assumption
Converge?
No
Yes
All reachable?
No
Yes
Over-constrained?
No
Yes
Bug in cover?
No
Yes

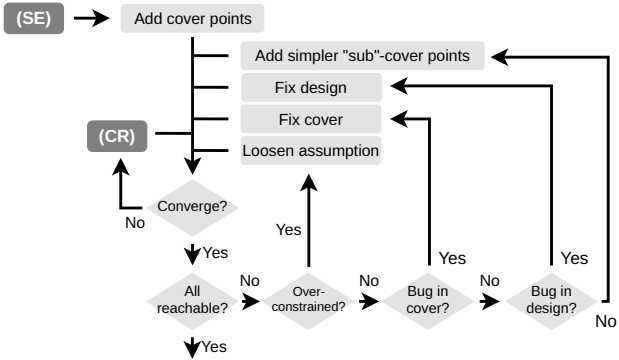


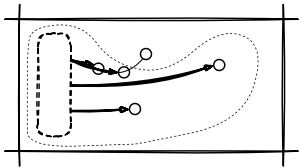

(SE)
Add cover points
Add simpler "sub"-cover points
Fix design
Fix cover
Loosen assumption
(CR)
Converge?
No
Yes
All reachable?
No
Yes
Over-constrained?
No
Yes
Bug in cover?
No
Yes
Bug in design?
No
Yes

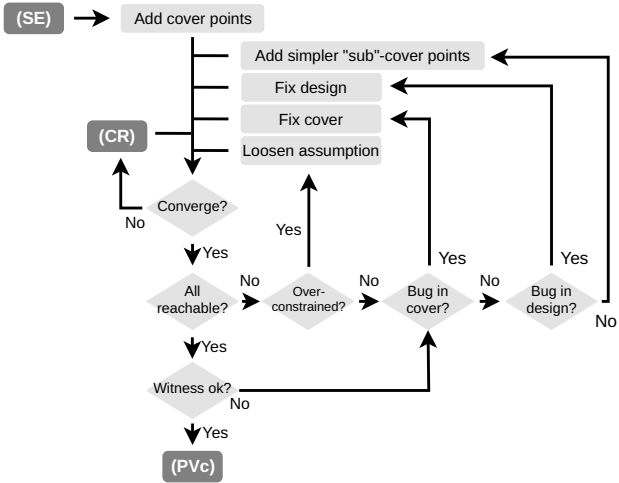


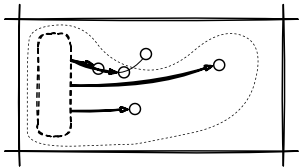

(SE)
Add cover points
Add simpler "sub"-cover points
Fix design
Fix cover
Loosen assumption
(CR)
Converge?
No
Yes
All reachable?
No
Yes
Over-constrained?
No
Yes
Bug in cover?
No
Yes
Bug in design?
No
Yes



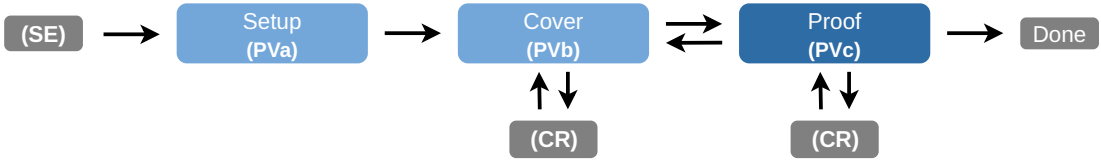

(SE)
Add cover points
Add simpler "sub"-cover points
Fix design
Fix cover
Loosen assumption
(CR)
Converge?
No
Yes
All reachable?
No
Yes
Over-constrained?
No
Yes
Bug in cover?
No
Yes
Bug in design?
No
Yes

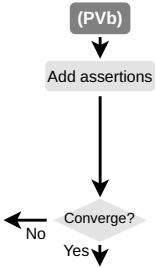


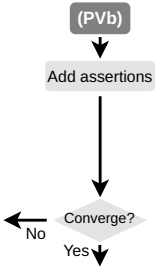

(SE)
Add cover points
Add simpler "sub"-cover points
Fix design
Fix cover
Loosen assumption
(CR)
Converge?
No
Yes
All reachable?
No
Over-constrained?
No
Bug in cover?
No
Bug in design?
No
Yes
Yes
Yes
Yes
Witness ok?
No
Yes
(PVc)

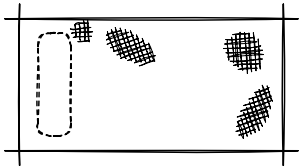


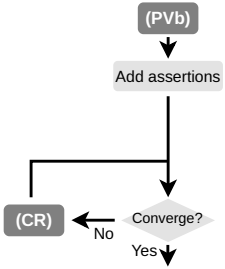
Property Verification (PV)

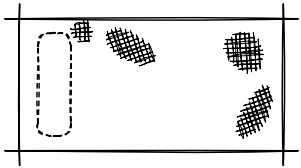


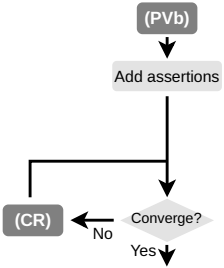

(PVb)
Add assertions
Converge?
No
Yes

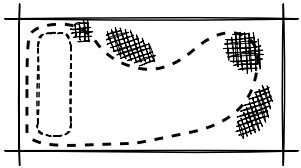

(PVb)
Add assertions
Converge?
No
Yes

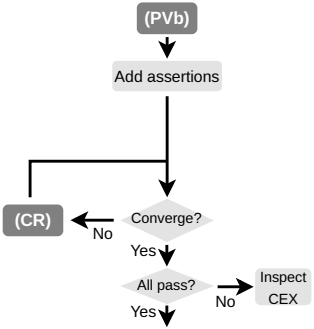


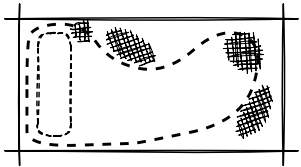

(PVb)
Add assertions
(CR)
No
Converge?
Yes

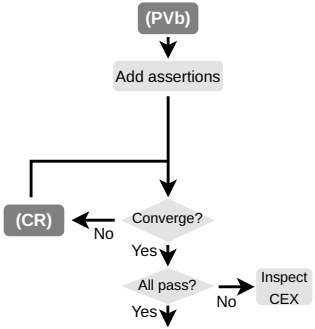


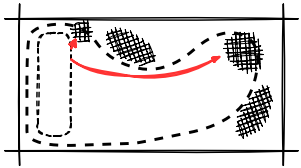

(PVb)
Add assertions
(CR)
No
Converge?
Yes

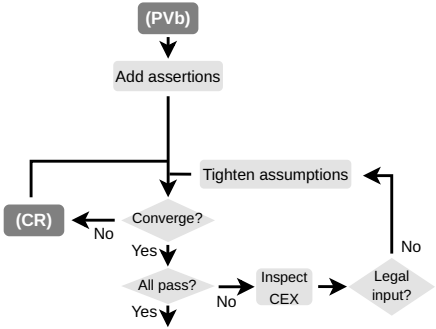


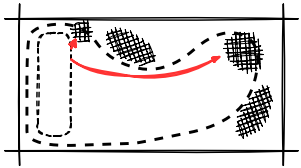

(PVb)
Add assertions
(CR)
No
Converge?
Yes
All pass?
No
Inspect CEX
Yes

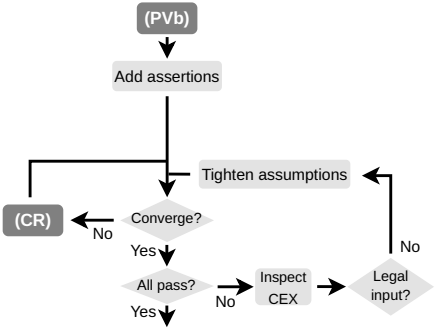


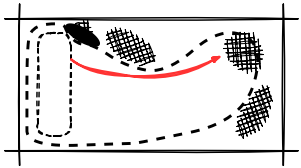

(PVb)
Add assertions
(CR)
No
Converge?
Yes
All pass?
No
Inspect CEX
Yes

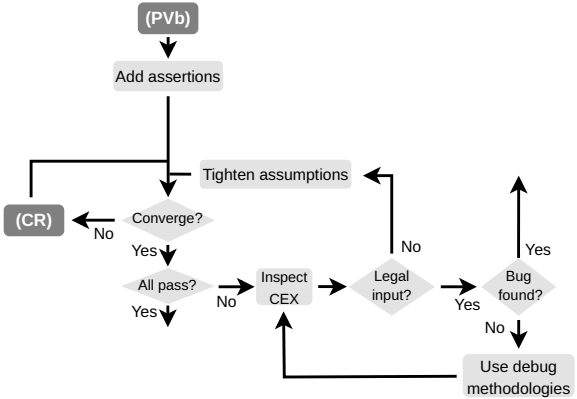


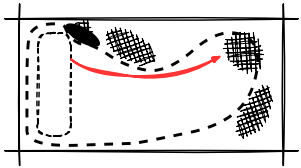

(PVb)
Add assertions
Tighten assumptions
(CR)
No
Converge?
Yes
All pass?
No
Inspect CEX
Legal input?
No
Yes

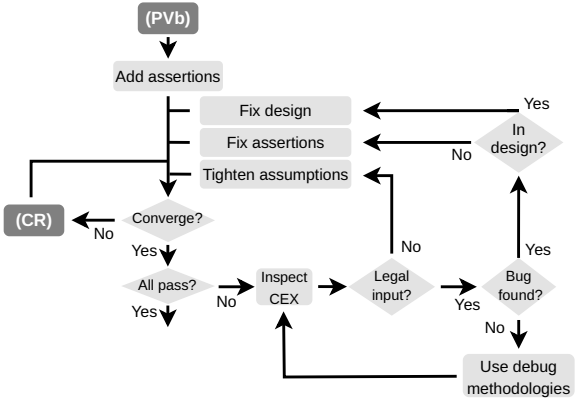


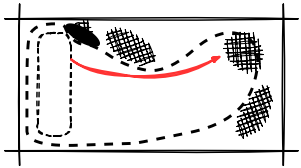

(PVb)
Add assertions
Tighten assumptions
(CR)
No
Converge?
Yes
All pass?
No
Inspect CEX
Legal input?
No
Yes

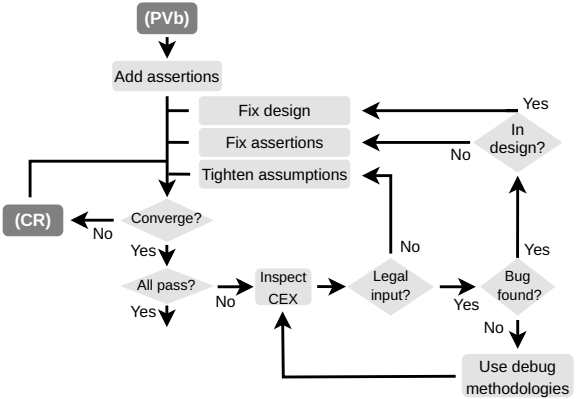


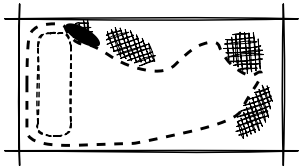

(PVb)
Add assertions
Tighten assumptions
(CR)
Converge?
No
Yes
All pass?
Yes
No
Inspect CEX
Legal input?
No
Yes
Bug found?
Yes
No
Use debug methodologies

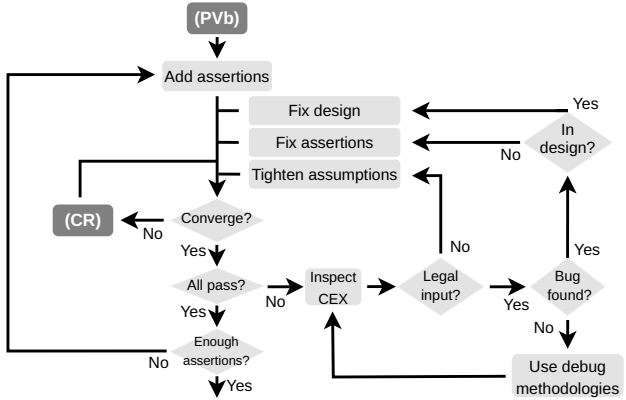


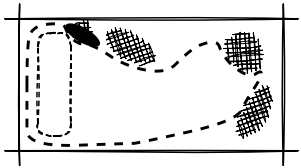

(PVb)
Add assertions
Fix design
Fix assertions
Tighten assumptions
Yes
In design?
No
(CR)
Converge?
No
Yes
All pass?
No
Yes
Inspect CEX
Legal input?
No
Yes
Bug found?
Yes
No
Use debug methodologies

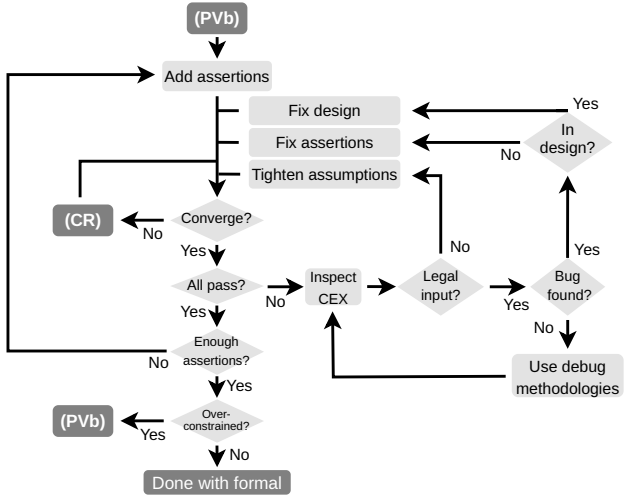


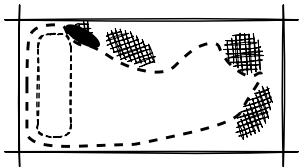

(PVb)
Add assertions
Fix design
Fix assertions
Tighten assumptions
(CR)
Converge?
No
Yes
All pass?
Yes
No
Inspect CEX
Legal input?
No
Yes
In design?
Yes
No
Bug found?
Yes
No
Use debug methodologies




(PVb)
Add assertions
Fix design
Fix assertions
Tighten assumptions
(CR)
Converge?
No
Yes
All pass?
No
Yes
Enough assertions?
No
Yes
Inspect CEX
Legal input?
No
Yes
Bug found?
Yes
No
In design?
Yes
No
Use debug methodologies




(PVb)
Add assertions
Fix design
Fix assertions
Tighten assumptions
(CR)
Converge?
No
Yes
All pass?
No
Yes
Inspect CEX
Legal input?
No
Yes
In design?
Yes
No
Bug found?
Yes
No
Use debug methodologies
Enough assertions?
No
Yes
(PVb)
Over-constrained?
Yes
No
Done with formal



Complexity Reduction (CR)

- Highly application specific
- See [1, 8, 3, 2, 4, 5, 10]

Complexity Reduction (CR)

Property Optimization

- Efficient property design
- Auxiliary code
- Assumption optimization

Complexity Reduction (CR)

Abstraction

- Parameter optimization
- Cut-points
- Black-boxing
- Shadow modeling

Complexity Reduction (CR)

Tool Specific & Strategy

- Different engine
- Bounded proof
- Continue with simulation

(Open-Source) Examples

- YosysHQ Read the Docs (great starting point) [14, 17]
- `YosysHQ/sby` [16]
- `riscv-formal` [15]
- The Tiny Tapeout Multiplexer [11]

SystemVerilog Assertions

- SystemVerilog Assertions Basics [9]
- SystemVerilog LRM [6]
- SVA: The Power of Assertions in SystemVerilog [1]
- Formal Verification: An Essential Toolkit for Modern VLSI Design [8]
- Open source support?

www.meinhard-kissich.at/pub/orconf23_slides.pdf

www.meinhard-kissich.at

meinhard.kissich@tugraz.at

References →

Bibliography I

[1] E. Cerny et al. **SVA: The Power of Assertions in SystemVerilog**. Springer International Publishing, 2016.

[2] Keerthikumara Devarajegowda et al. **Formal Verification Methodology in an Industrial Setup**. Euromicro DSD Conf. 2019.

[3] Harry Foster et al. **Guidelines for creating a formal verification testplan**. DVCON (2006).

[4] Joe Hupcey III. **How to Reduce the Complexity of Formal Analysis – Part 3 – Assertion Decomposition**. [Online] https://blogs.sw.siemens.com/verificationhorizons/2018/09/11/how-to-reduce-the-complexity-of-formal-analysis-part-3-assertion-decomposition/ (Apr. 2023). Sept. 2018.

[5] Joe Hupcey III. **How to Reduce the Complexity of Formal Analysis – Part 5 – Memory Abstraction**. [Online] https://blogs.sw.siemens.com/verificationhorizons/2018/10/23/how-to-reduce-the-complexity-of-formal-analysis-part-5-memory-abstraction/ (Apr. 2023). Oct. 2018.

[6] **IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language**. IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012) (2018). DOI: 10.1109/IEEESTD.2018.8299595.

Bibliography II

[7] Meinhard Kissich and Marcel Baunach. **Formal Property Verification for Early Discovery of Functional Flaws in Digital Designs: A Designer's Guide**. DSD Conf. 2023.

[8] Erik Seligman, Tom Schubert, and M V Achutha Kiran Kumar. **Formal Verification: An Essential Toolkit for Modern VLSI Design**. Morgan Kaufmann Publishers Inc., 2015.

[9] Subbdue. **SystemVerilog Assertions Basics**. [Online] https://www.systemverilog.io/verification/sva-basics/ (Aug. 2023).

[10] Anamaya Sullerey. **Design Guidelines for Formal Verification**. DVCON. 2015.

[11] TinyTapeout. **The Tiny Tapeout Multiplexer**. [Online] https://github.com/TinyTapeout/tt-multiplexer (Aug. 2023).

[12] Venn, Matt and Wolf, Clare. **Introduction to Formal Verification with Symbiotic EDA Open Source Tools**. [Online] https://www.youtube.com/watch?v=H3tsP9tjYdY (Jul. 2023).

[13] Wolf, Clare. **Formal Verification of Verilog HDL with Yosys-SMTBMC (33c3)**. [Online] https://media.ccc.de/v/33c3-7922-formal_verification_of_verilog_hdl_with_yosys-smtbmc (Jul. 2023).

Bibliography III

[14] YosysHQ. **Getting started**. [Online] https://symbiyosys.readthedocs.io/en/latest/quickstart.html (Aug. 2023).

[15] YosysHQ. **RISC-V Formal Verification Framework**. [Online] https://github.com/YosysHQ/riscv-formal (Aug. 2023).

[16] YosysHQ. **SymbiYosys (sby) – Front-end for Yosys-based formal verification flows**. [Online] https://github.com/YosysHQ/sby/tree/master/docs/examples (Aug. 2023).

[17] YosysHQ. **Weak precondition cover and witness for SVA properties**. [Online] https://yosyshq.readthedocs.io/projects/ap120/en/latest/ (Jul. 2023).