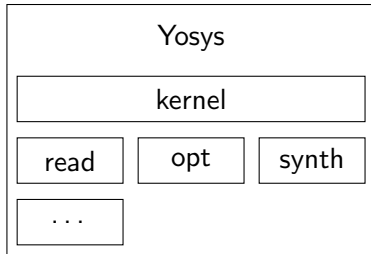


# Python Wraps Yosys for Rapid Open-Source EDA Application Development

**Benedikt Tutzer**, Christian Krieg, Clifford Wolf, Axel Jantsch

29<sup>th</sup> of March 2019, Workshop on Open Source Design Automation (OSDA), Florence, Italy

# What is pyosys?

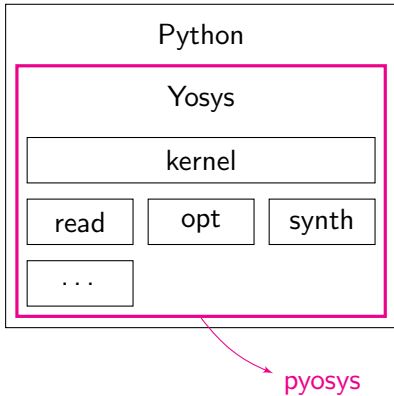


Yosys:

- Kernel
- Default Passes
- Custom Passes
- C++ Interface



# What is pyosys?



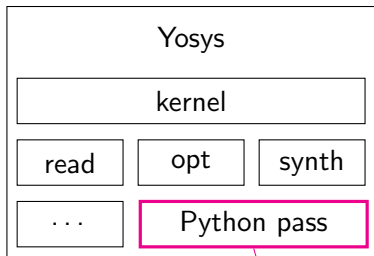
## Yosys:

- Kernel
- Default Passes
- Custom Passes
- C++ Interface

## pyosys

- Python Interface

# What is pyosys?



pyosys

Yosys:

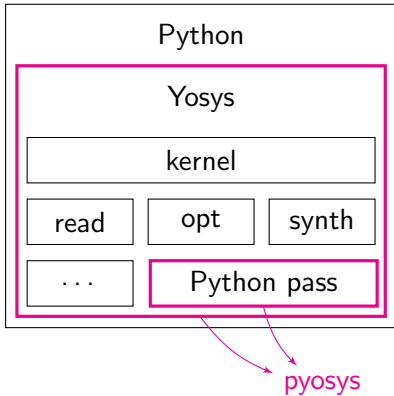
- Kernel
- Default Passes
- Custom Passes
- C++ Interface

pyosys

- Python Passes



# What is pyosys?



Yosys:

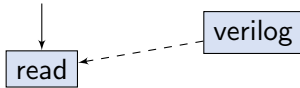
- Kernel
- Default Passes
- Custom Passes
- C++ Interface

pyosys

- Python Interface
- Python Passes



# Why pyosys?

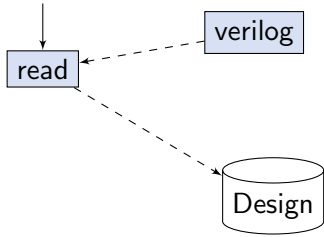


Sample TCL Flow:

- read HDL design



# Why pyosys?

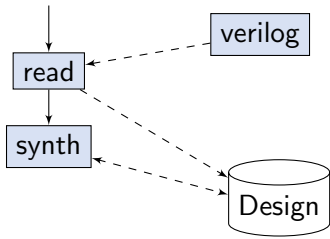


Sample TCL Flow:

- read HDL design
- depicted to internal data structure



# Why pyosys?



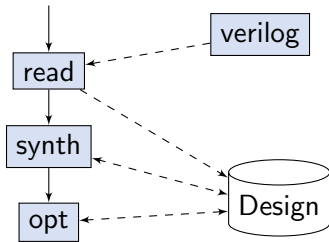
## Sample TCL Flow:

- synthesize
- read and write from/to internal data structure





# Why pyosys?

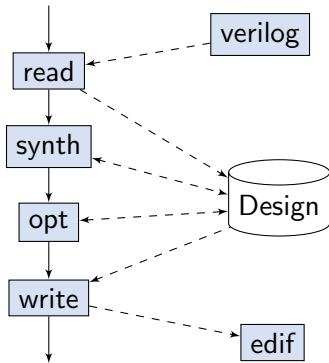


## Sample TCL Flow:

- optimize
- read and write from/to internal data structure



# Why pyosys?

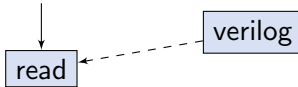


## Sample TCL Flow:

- read from internal data structure
- write to edif file



# Why pyosys?

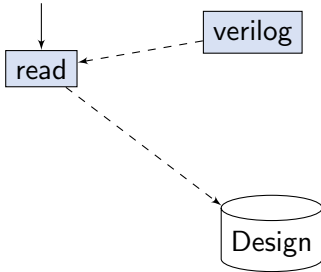


Sample C++ based Yosys Flow:

- read HDL design



# Why pyosys?

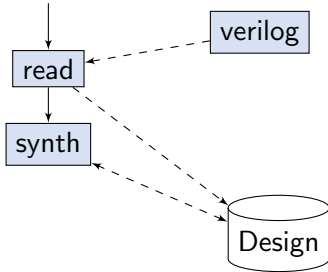


Sample C++ based Yosys Flow:

- read HDL design
- depicted to *accessible* internal data structure



# Why pyosys?

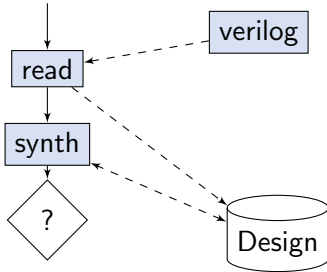


Sample C++ based Yosys Flow:

- synthesize
- read and write from/to internal data structure



# Why pyosys?

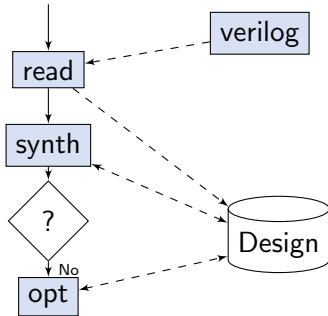


Sample C++ based Yosys Flow:

- C++ code
- Elaborate computations with full design access



# Why pyosys?

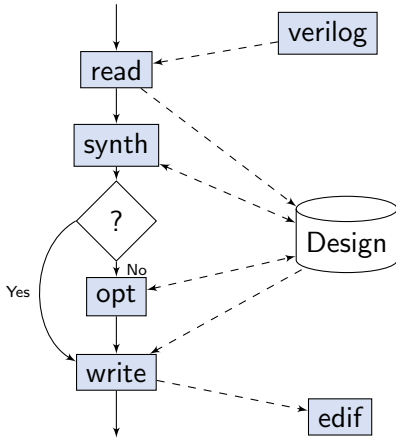


Sample C++ based Yosys Flow:

- if design does not fit: optimize
- read and write from/to internal data structure



# Why pyosys?



Sample C++ based Yosys Flow:

- write netlist file
- read from internal data structure
- write to edif file





- TCL<sup>1</sup> sufficient for synthesis flow

---

<sup>1</sup>Tool Command Language

# Motivation

- TCL<sup>1</sup> sufficient for synthesis flow
- C++ based Yosys

---

<sup>1</sup>Tool Command Language



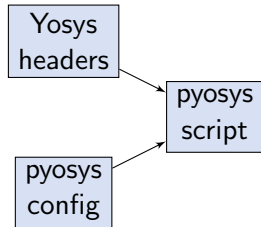
- TCL<sup>1</sup> sufficient for synthesis flow
- C++ based Yosys
- Functionality focused, fast prototyping through Python

---

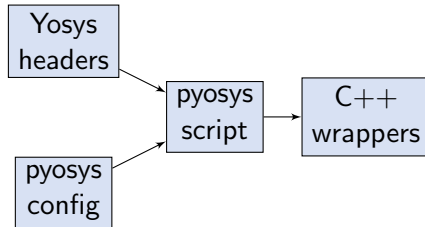
<sup>1</sup>Tool Command Language



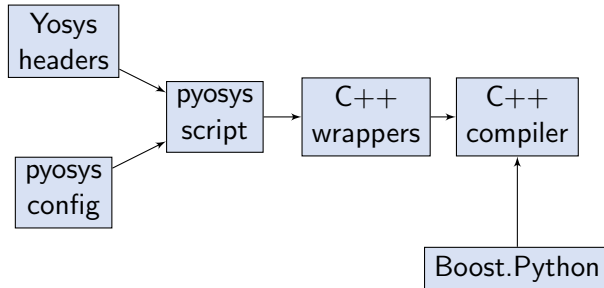
# Implementation



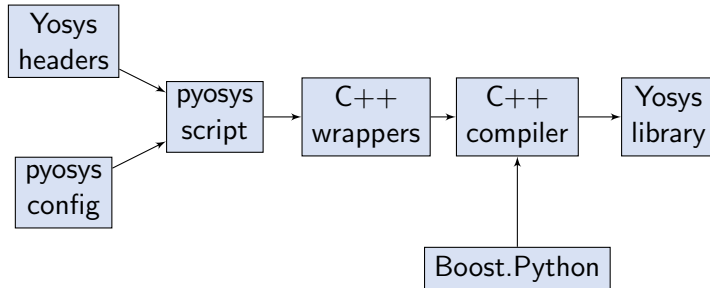
# Implementation



# Implementation

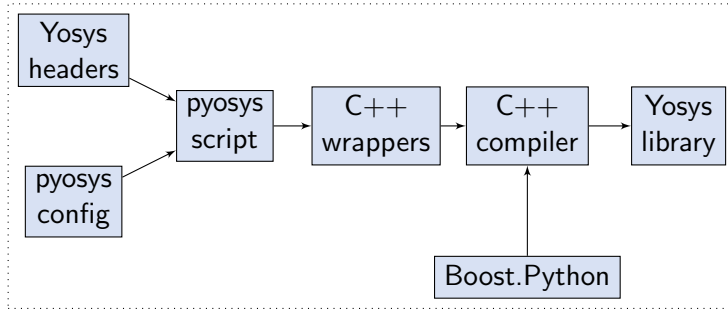


# Implementation



# Implementation

## Yosys Makefile





- ① Interactive usage: *from a Python shell*
- ② Non-interactive usage: *creating a pass*



# Wrap up

- Interactive usage → *immediate feedback*



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*
- Develop Python passes → *abstraction to arbitrary designs*



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*
- Develop Python passes → *abstraction to arbitrary designs*
- Use Python passes in C++ → *flexible pass usage*



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*
- Develop Python passes → *abstraction to arbitrary designs*
- Use Python passes in C++ → *flexible pass usage*
- Use Python passes in Yosys scripts → *reuse existing code*



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*
- Develop Python passes → *abstraction to arbitrary designs*
- Use Python passes in C++ → *flexible pass usage*
- Use Python passes in Yosys scripts → *reuse existing code*
- Currently available at:  
`www.github.com/christian-krieg/yosys/tree/feature/python\_bindings`



# Wrap up

- Interactive usage → *immediate feedback*
- Python scripting → *fast prototyping*
- Develop Python passes → *abstraction to arbitrary designs*
- Use Python passes in C++ → *flexible pass usage*
- Use Python passes in Yosys scripts → *reuse existing code*
- Currently available at:  
`www.github.com/christian-krieg/yosys/tree/feature/python\_bindings`
- Hopefully soon in the official Yosys repository





