A Guide for Rapid Creation of New HDLs

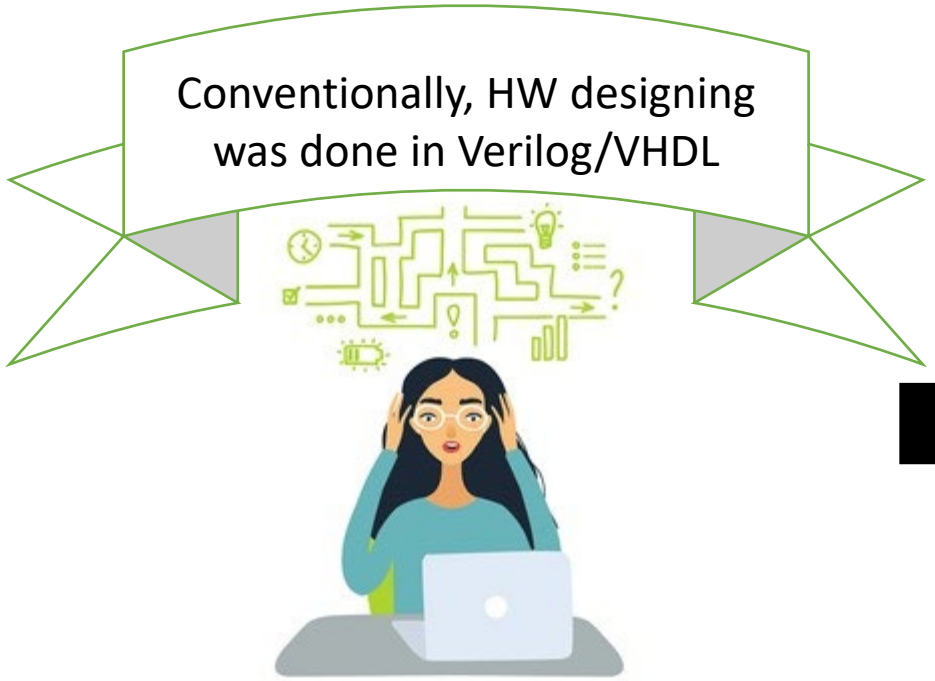
Sakshi Garg, Sheng-Hong Wang, Jose Renau

Department of CSE

University of California, Santa Cruz

WOSET 2021

Hardware Designing is getting easier...



Functional Programming

Module Reusability

Object Oriented programming

Digital Hardware Designing

Agile Hardware designing

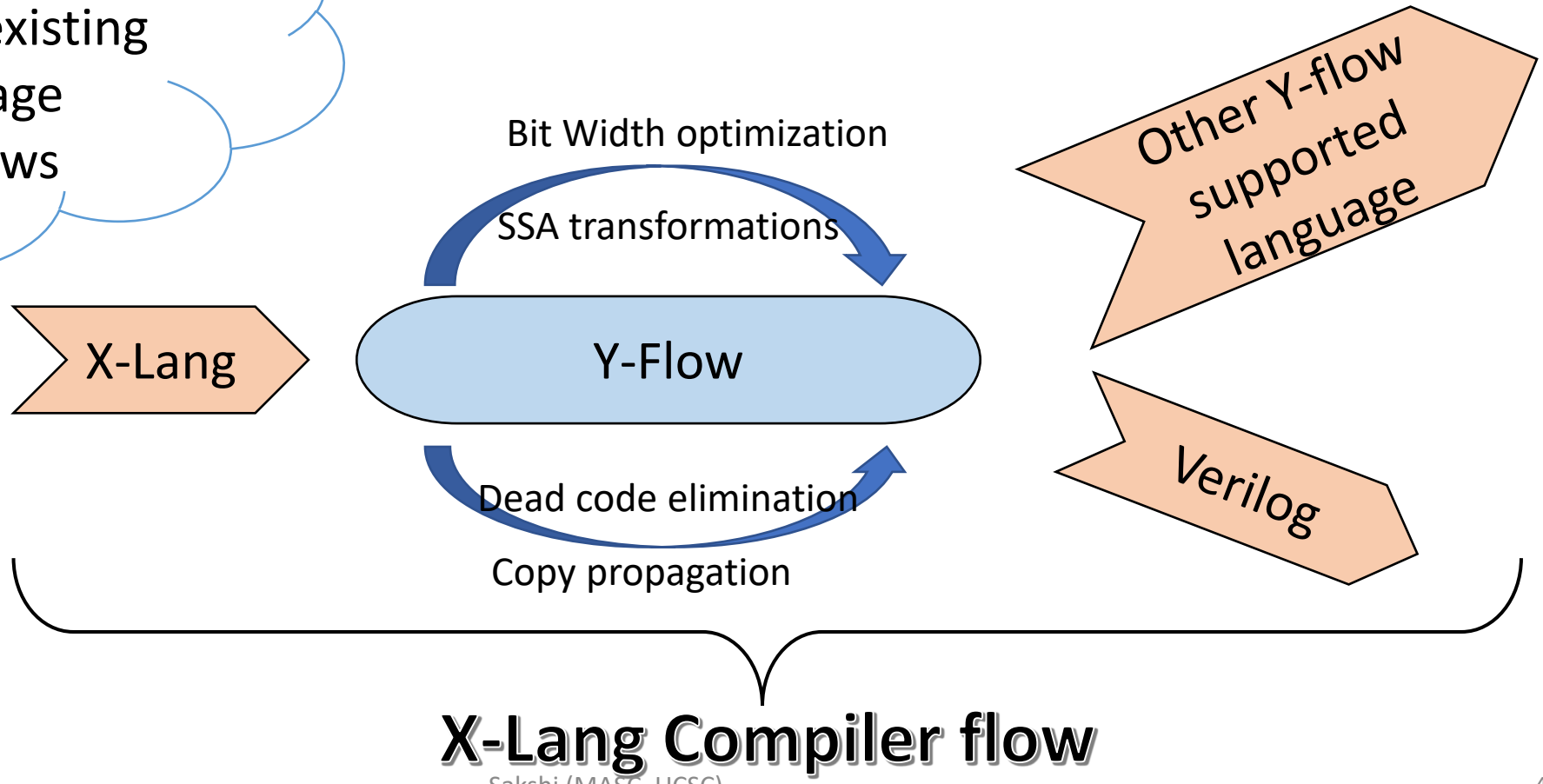
New HDLs like Chisel, PyRTL, Mamba, etc.

Then what's the problem?

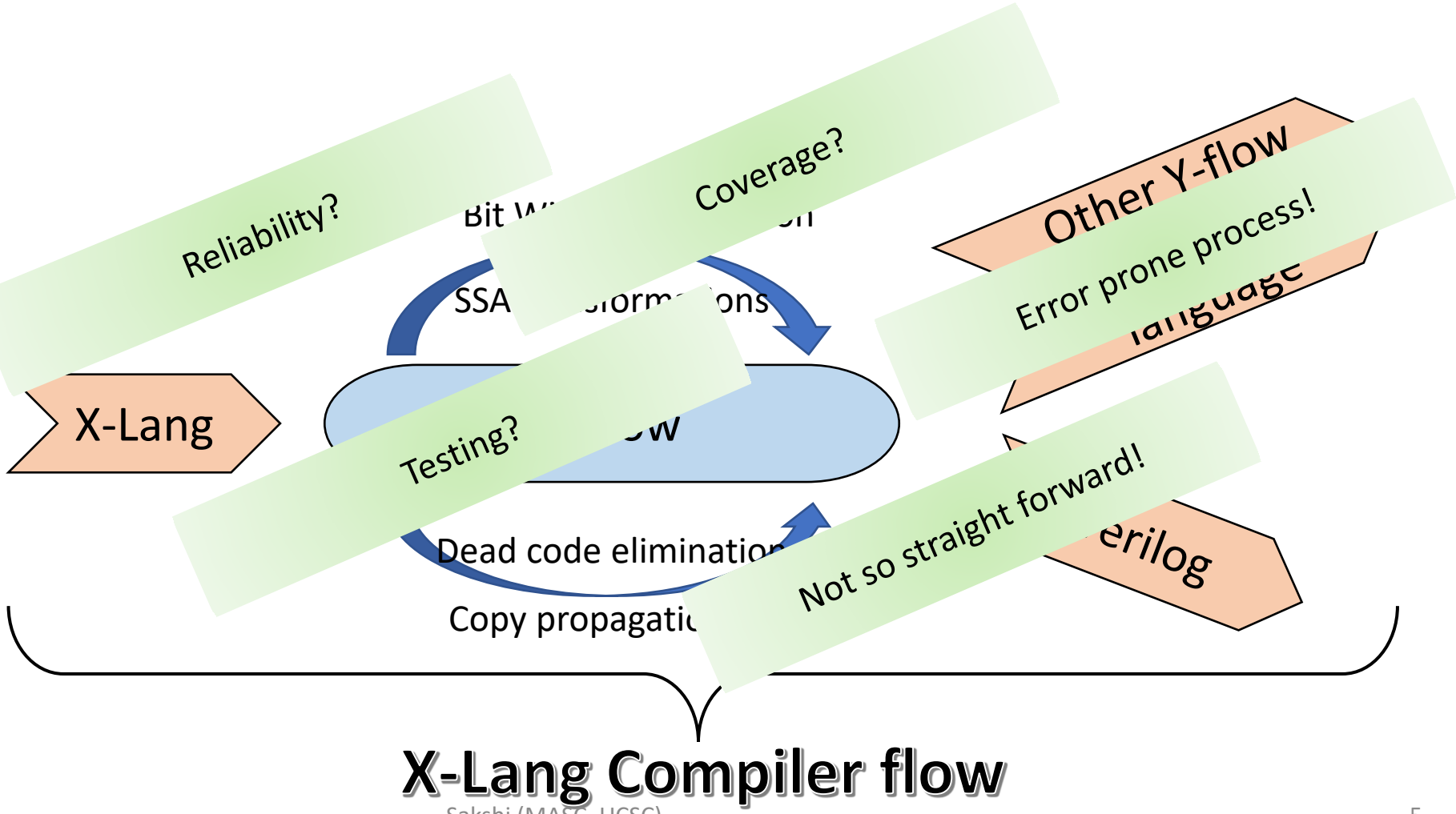
- Every new HDL has its own compiler flow.
- Hence corresponding compiler optimization passes were designed.
- Verification and testing for each of these language-compiler infrastructure were carried out.
- Too much redundancy!

Thus, we propose:

Create a new HDL by leveraging pre-existing multi-language compiler flows



Uh oh!


Reliability?
Coverage?
Other Y-flow
Error prone process!
X-Lang
Testing?
Not so straight forward!
Verilog
X-Lang Compiler flow

The proposal!

- Let’s have a way to **verify** the X-lang compiler flow by having high **coverage**.
- Let’s not implement reference compilers.
- Let’s leverage Y-flow **cost-effective**ly!
- No more extensive Verilog simulations needed to verify the system.

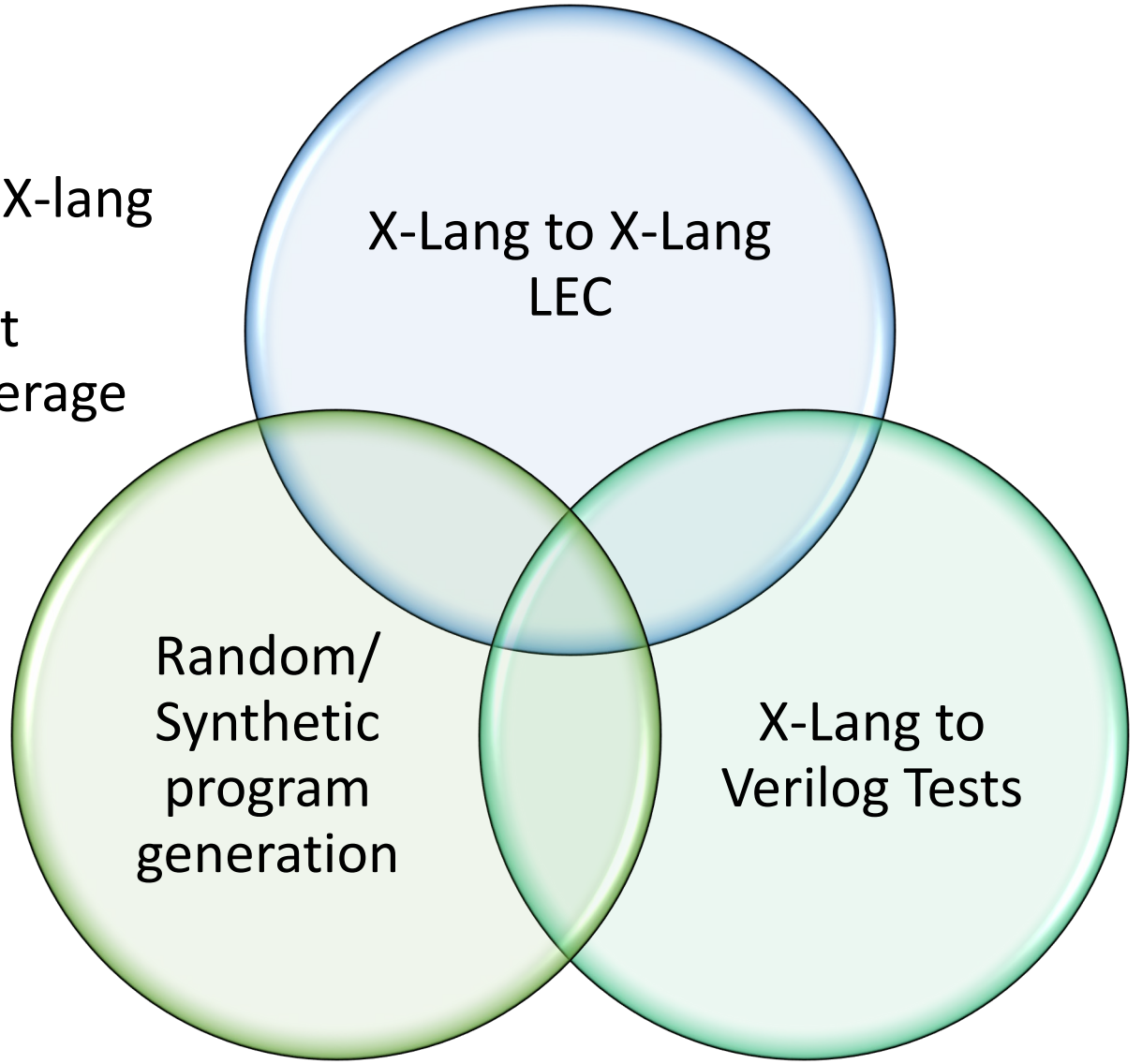
The proposal!

Let’s have a way to verify the X-lang compiler flow by having high coverage. Let’s not implement reference compilers. Let’s leverage Y-flow cost-effectively!

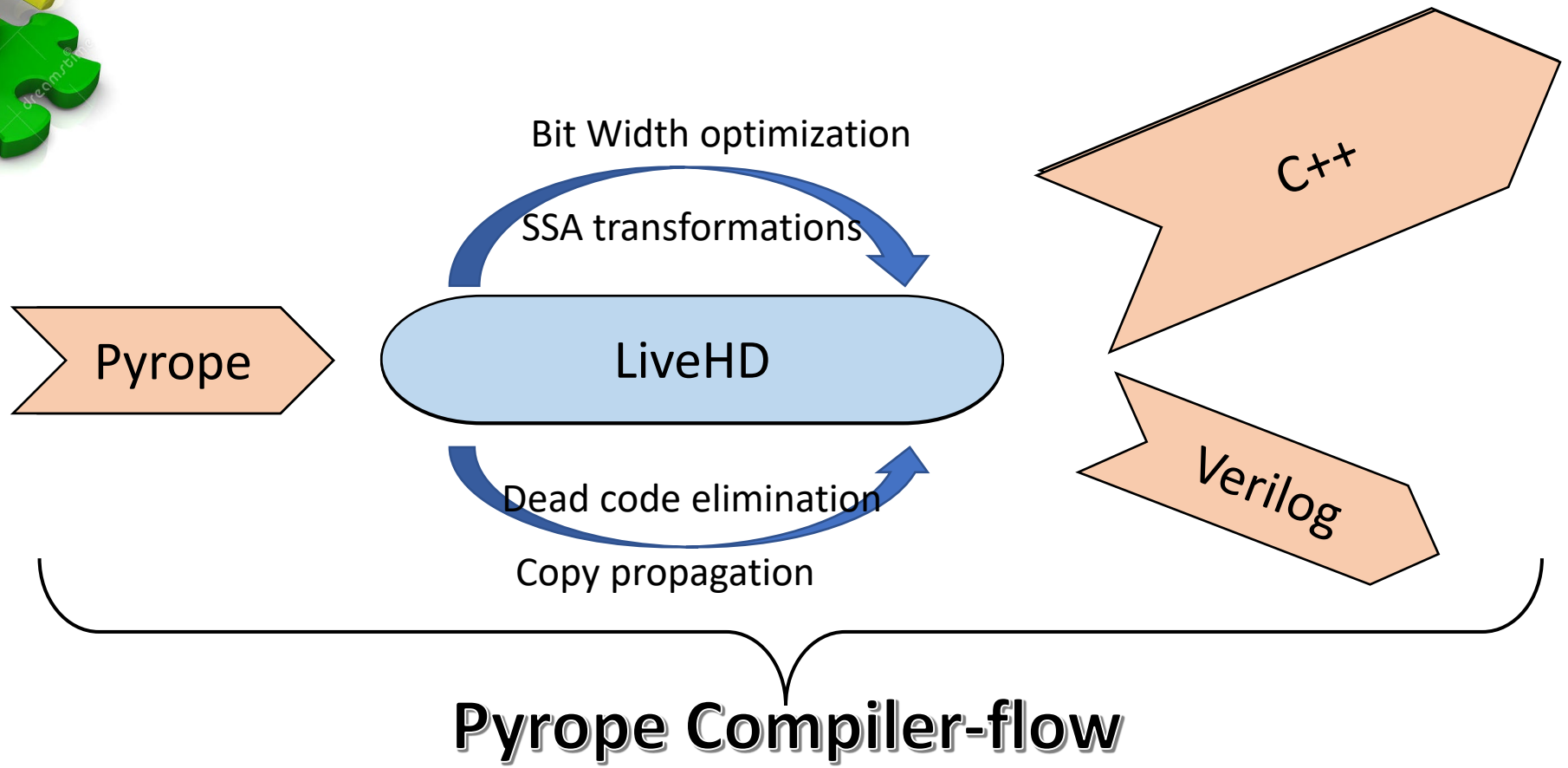
X-lang Input Interfacing

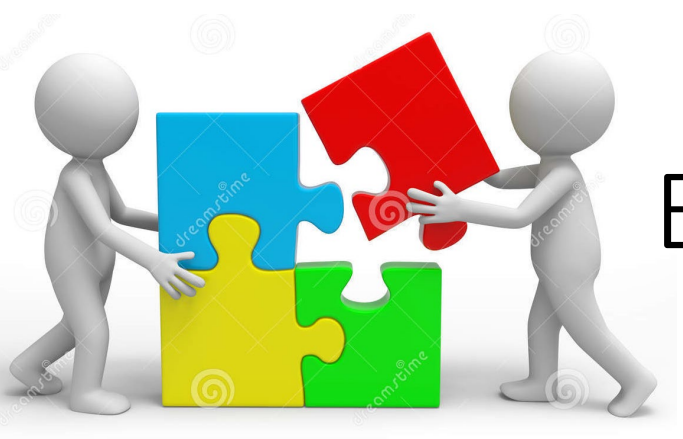
X-lang Generation

Semantically Same programs

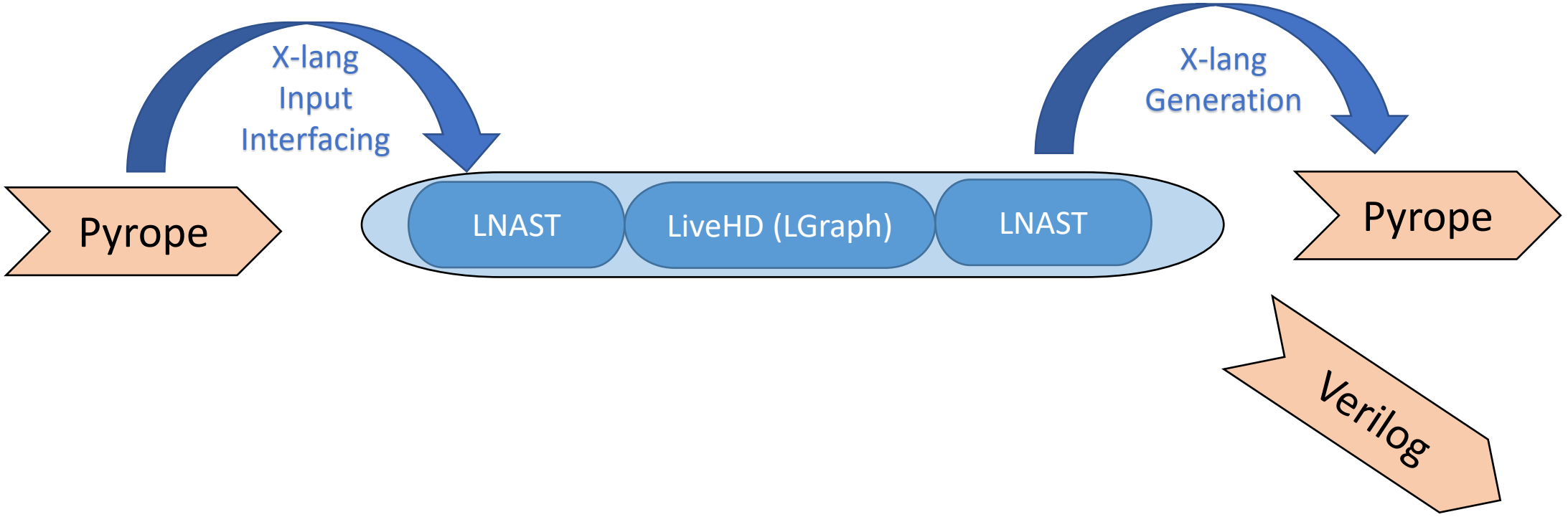


Using an Example Implementation



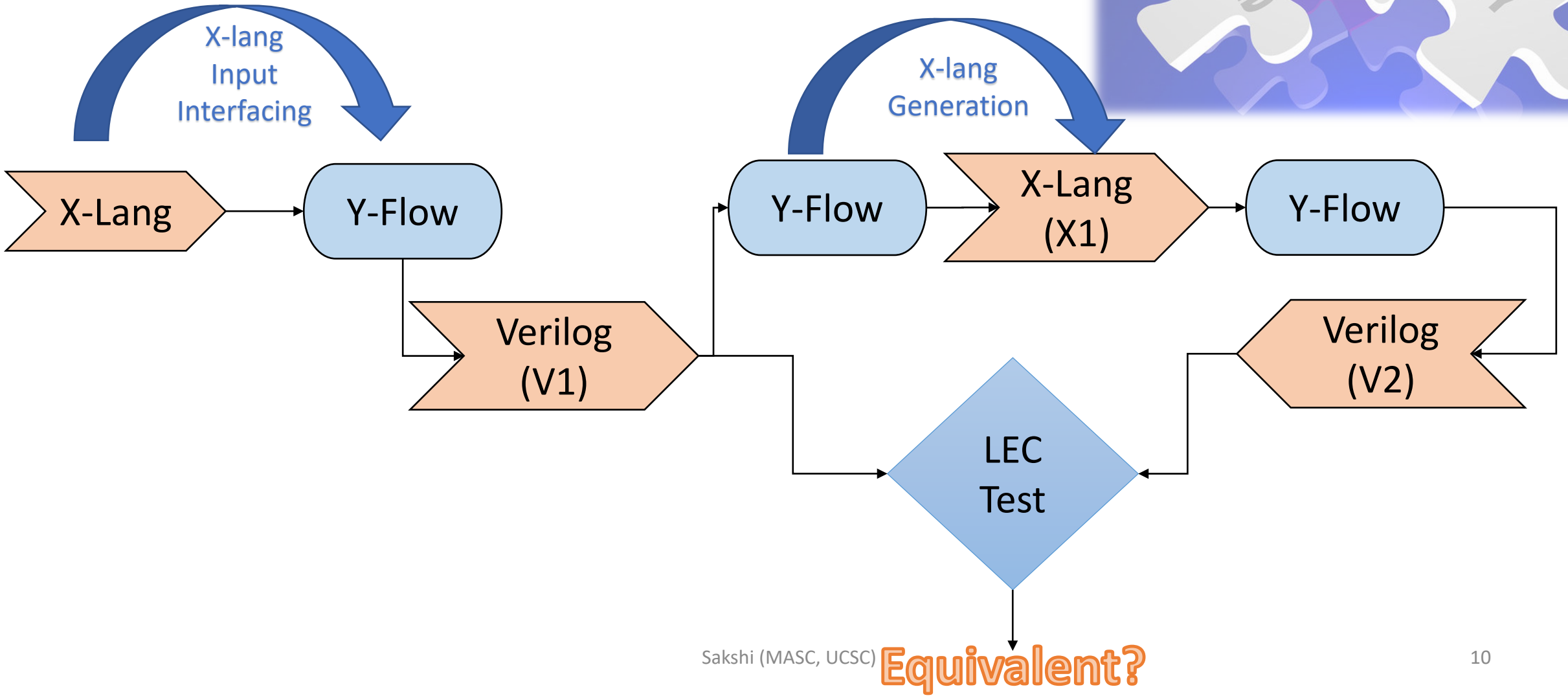


Example Implementation

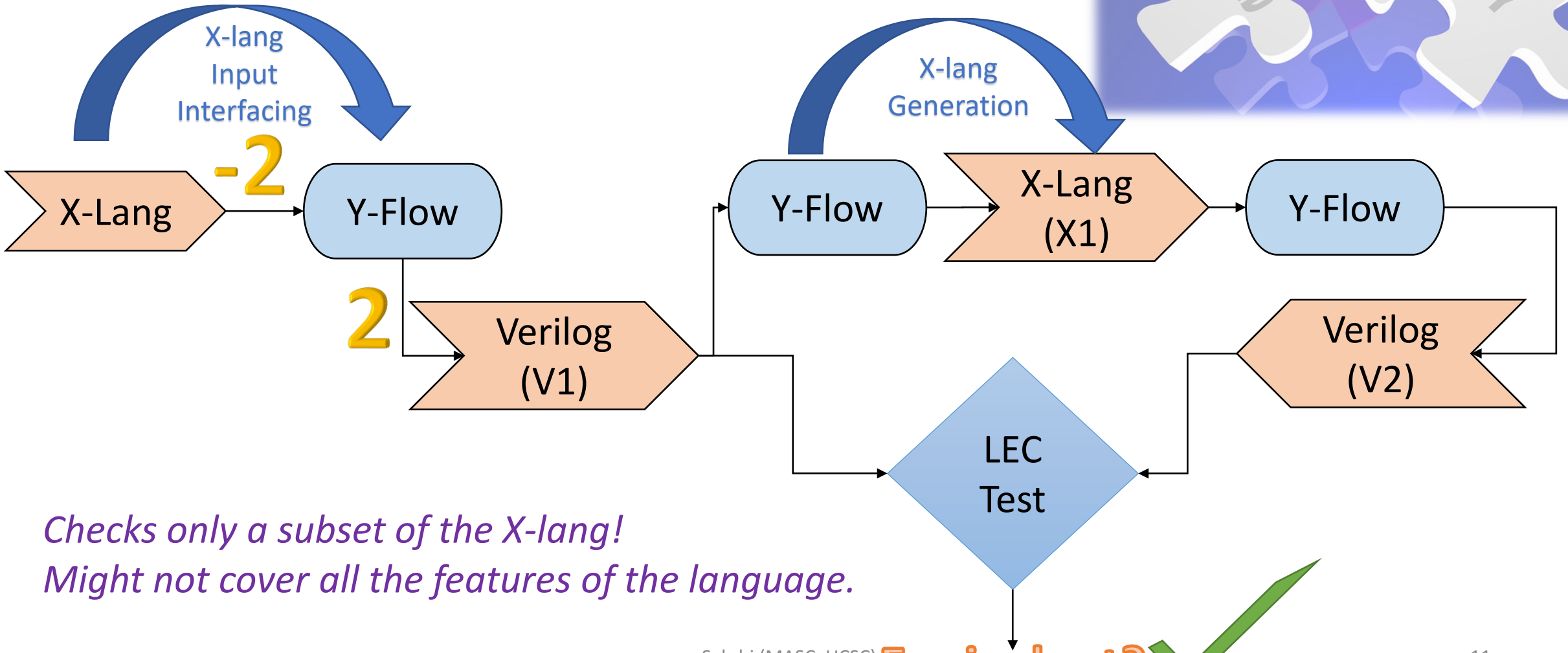


** Code and implementation details can be accessed at https://github.com/masc-ucsc/livehd*

X-Lang to X-Lang LEC

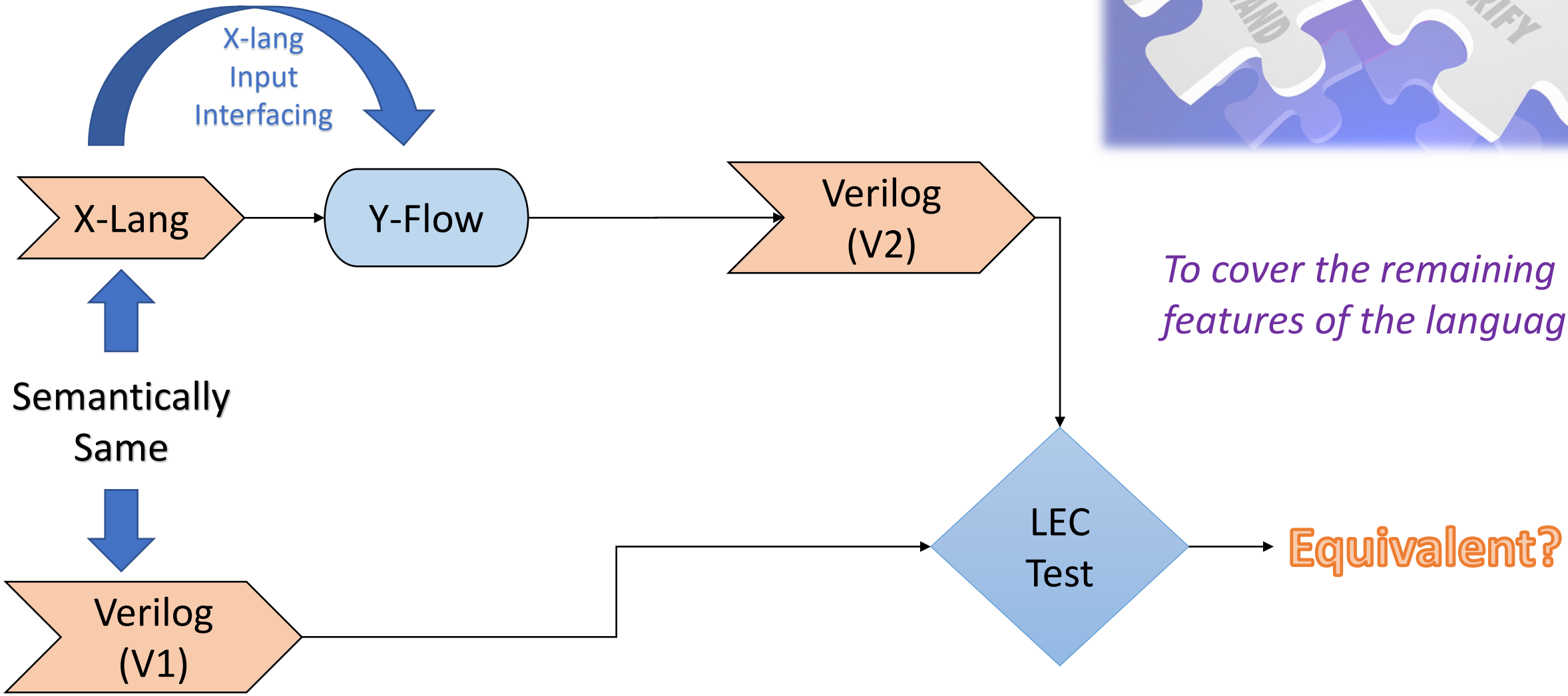


X-Lang to X-Lang LEC



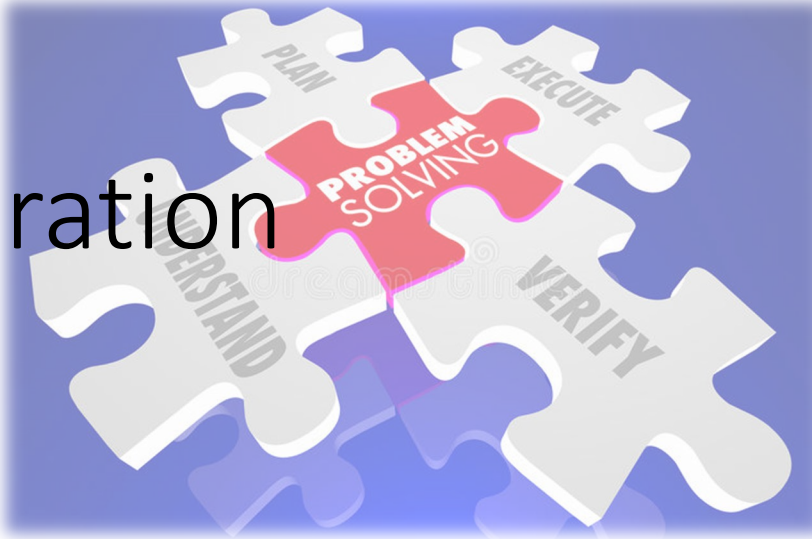
Checks only a subset of the X-lang!
Might not cover all the features of the language.

X-Lang to Verilog Tests



To cover the remaining features of the language.

Random/Synthetic program generation



Bank of randomly/manually generated programs

X-Lang

X-Lang

X-Lang

X-Lang

Verilog

X-Lang to X-Lang LEC

X-Lang to Verilog Tests

High Coverage, High Confidence

Conclusively…

- New HDL developers can leverage exiting multi-language compiler flows for the HDL.
- Provide reliability and trust earlier in the design flow
- Shorter development cycle
- No reference compiler required for flow verification
- Saved tremendous efforts of the new language developers
- Detect issues in any part of the X-Lang compiler infrastructure formed.
- X-lang can now be translated to other Y-flow-supported HDLs.

Acknowledgements

This work has been supported by the Center for Research in Open Source Software (CROSS) at UC Santa Cruz. This material is based upon work supported by, or in part by, the Army Research Laboratory and the Army Research Office under contract/grant W911NF1910466.

Thanks to Sheng-Hong and Dr. Renau for their continuous feedback and support.

Thanks to Mark Zakharov for his contribution to semantically same program generation.

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

Thank You!