Title

Twine: A Chisel Extension for Component-Level Heterogeneous Design

PRESENTER:

Shibo Chen

BACKGROUND: To achieve the swift development of heterogeneous designs, designers reuse existing hardware components to craft their systems. Twine aims to make heterogeneous design and design space exploration more accessible.

METHODS

1. Reusable Standard Module Interfaces

- ❖ Four standard interfaces: *TightlyCoupledIO, ValidIO, DecoupledIO, OutOfOrderIO*
- ❖ Parameterizable buffers
- ❖ Flexible mix & match

2. High-level Operators & Semantics

- ❖ Component-level connection operations
- ❖ Producer/Consumer relations

3. Automations for Heterogeneous Design

- ❖ System-level control logic synthesis
- ❖ Built-in data format conversions
  - Different forms of INT and FP
  - Parallel ←→ Serial

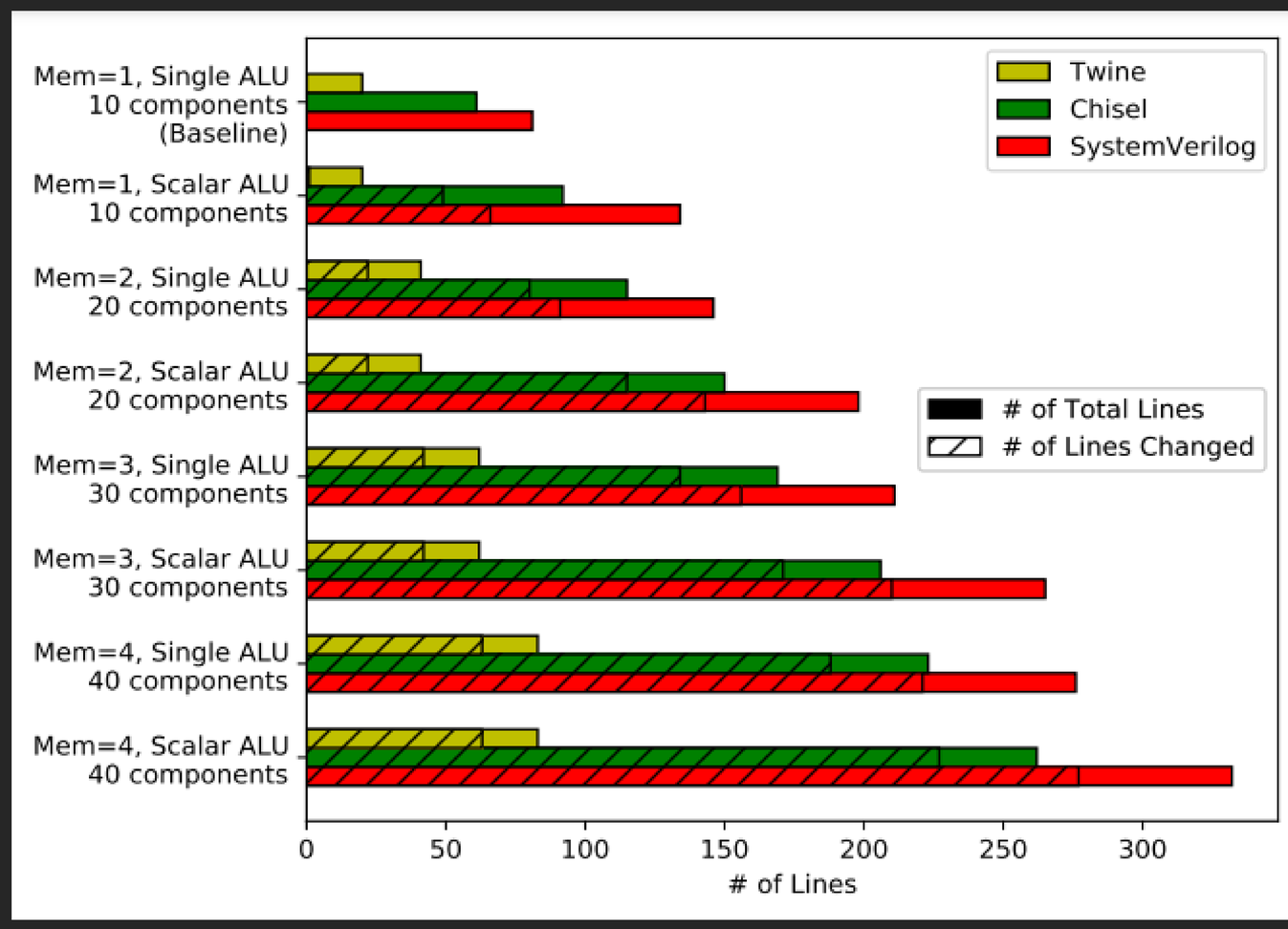
4. Chisel-Compatible

- ❖ Access to low-level abstractions & meta-programming capabilities
- ❖ Smooth migration process

RESULTS

- **Reusability & Productivity:**
  - Reconfigurable & modular architecture based on Q100[1]
  - Results shown on the center panel.
- **Design Quality:**
  - 3-stage RISCV-MINI Core[2]
  - vs. Chisel: 99.9% area, 103% clock frequency[3]
- **Early Adopters & Onboarding Experience:**
  - Seven graduate students with different levels of hardware design background
  - Learned and reconfigured Twine design in 35 minutes

Twine makes component-level hardware design more accessible and reusable in the age of heterogeneous design.



- **3x fewer lines of code for same design**
- **Fewer lines changed between configurations**
- **Smooth onboarding experience**

poster

source code

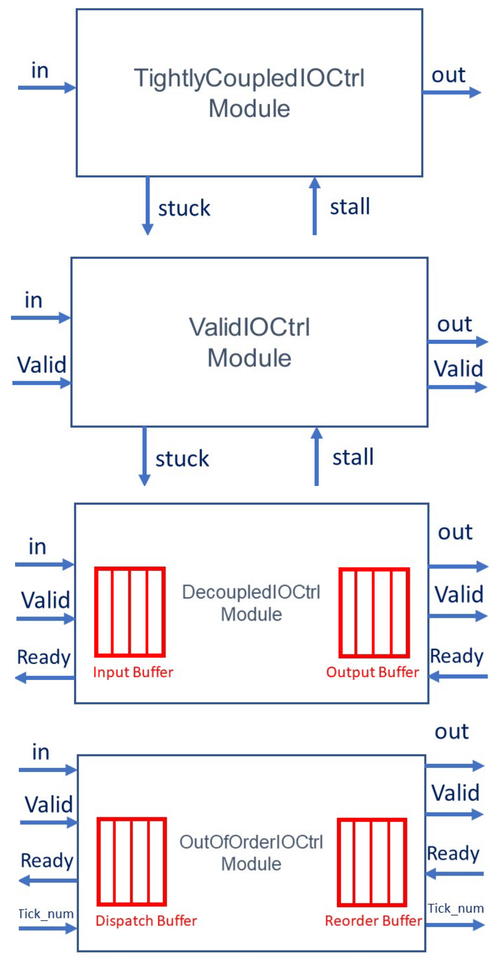
Take a picture to **download** the **poster** or **source code**

[1] Lisa Wu et al, 2014. Q100: the architecture and design of a database processing unit. SIGPLAN Not. 49, 4 (April 2014), 255-268. DOI:https://doi.org/10.1145/2644865.2541961

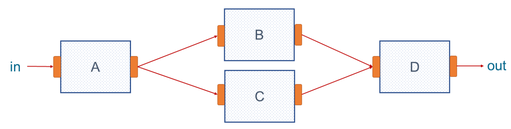
[2] Donggyu Kim. RISCV-MINI.https://github.com/ucb-bar/riscv-mini

[3] Synthesized with IBM 45nm SOI12S0 CMOS Process

Four Standard Control Interfaces



High-level Syntax & Semantics

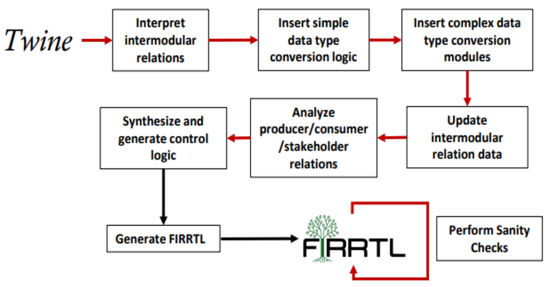


in >>> A >>> B >>> D.in.data1

A >>> C >>> D.in.data2

D >>> out

Twine Elaboration Process



Acknowledgement

We would like to thank the reviewers for their valuable feedback.
This work was supported by the Applications Driving Architectures (ADA) Research Center, a JUMP Center co-sponsored by SRC and DARPA.

Shibo Chen,
Yonathan Fisseha,
Jean-Baptiste Jeannin,
Todd Austin

UNIVERSITY OF MICHIGAN

SRC® JUMP ada Applications Driving Architectures