Twine: A Chisel Extension for Component-Level Heterogeneous Design

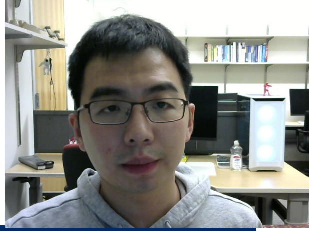
Shibo Chen, Yonathan Fisseha, Jean-Baptiste Jeannin, Todd Austin

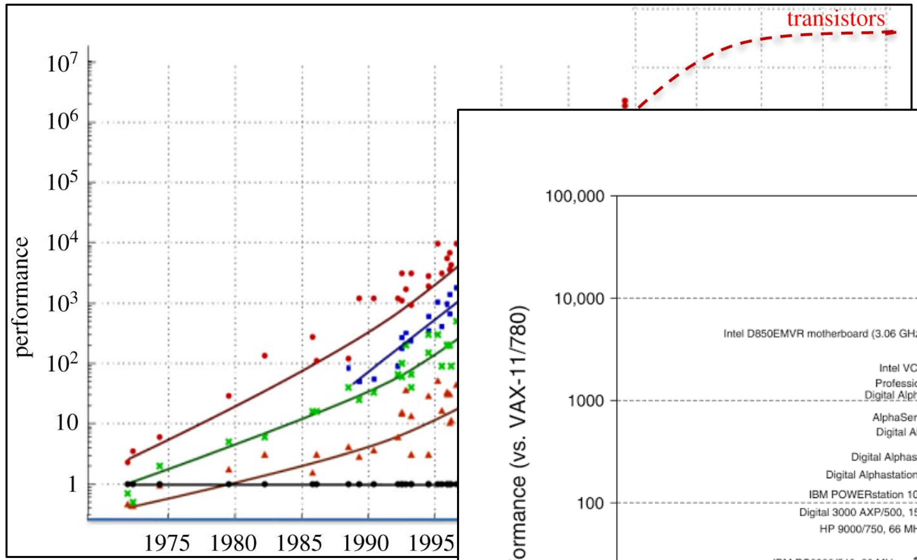
University of Michigan

Applications Driving Architectures

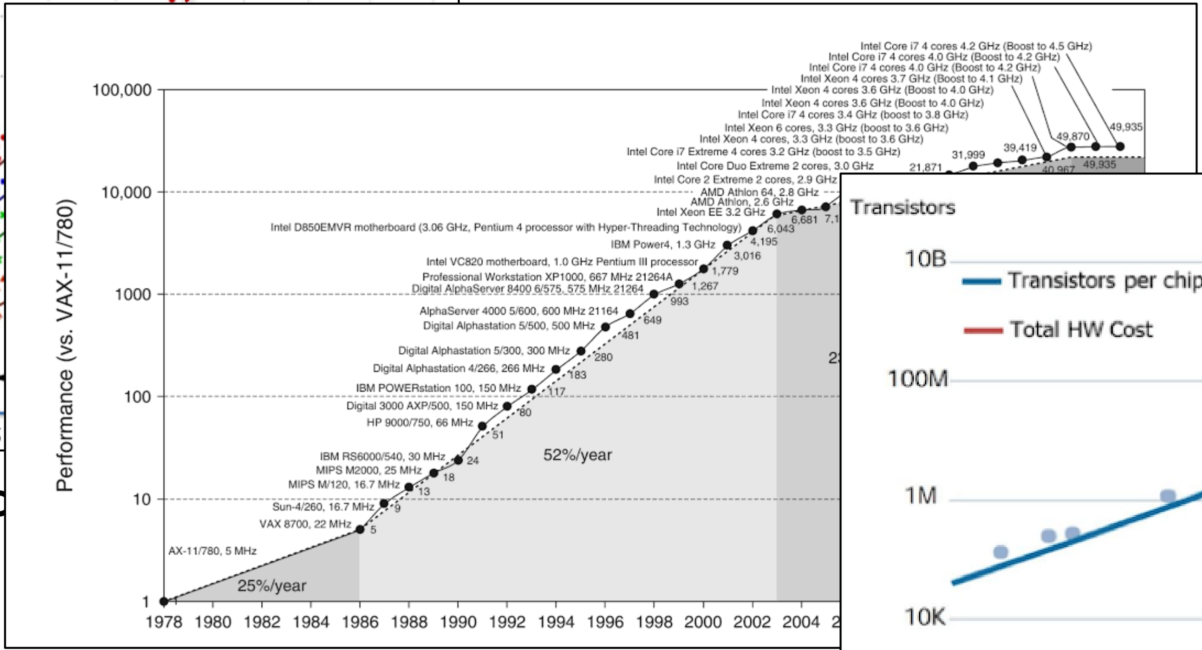
COLLEGE OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

The Death of Homogeneous Designs

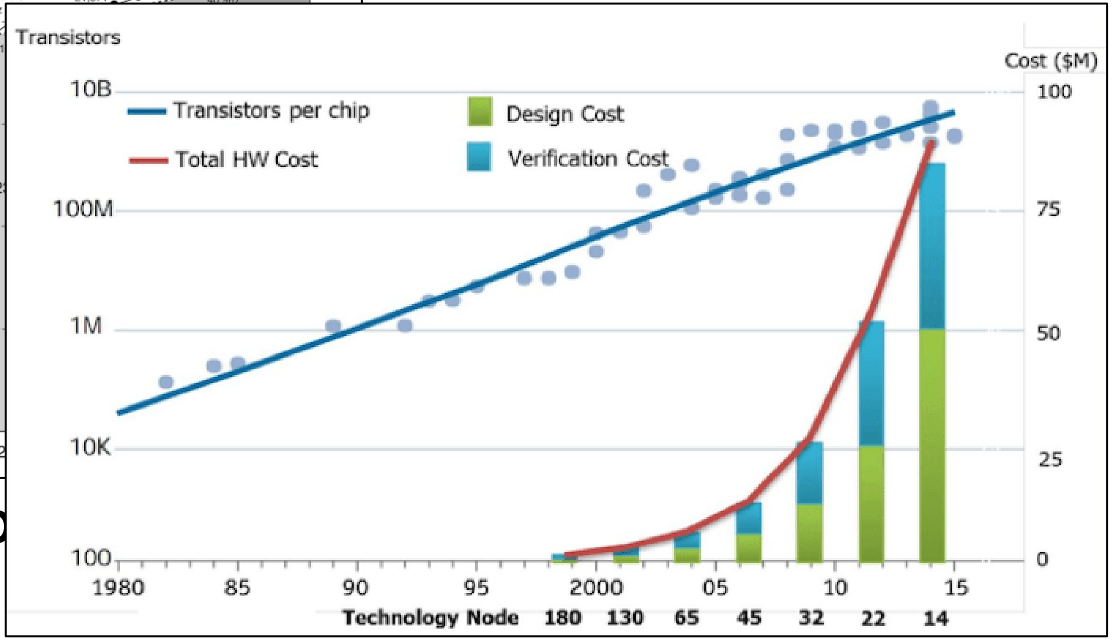




Classical technologic



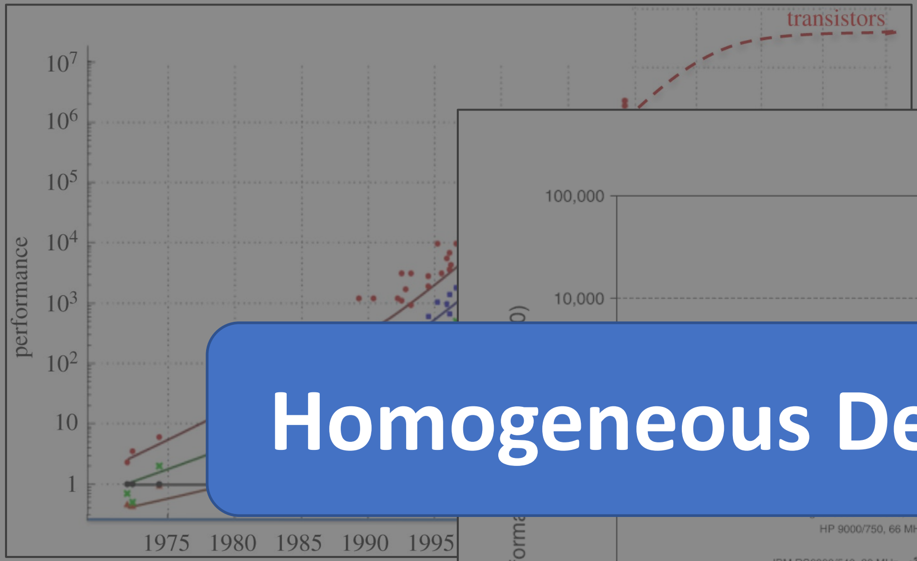
CPU Performance Scaling is D



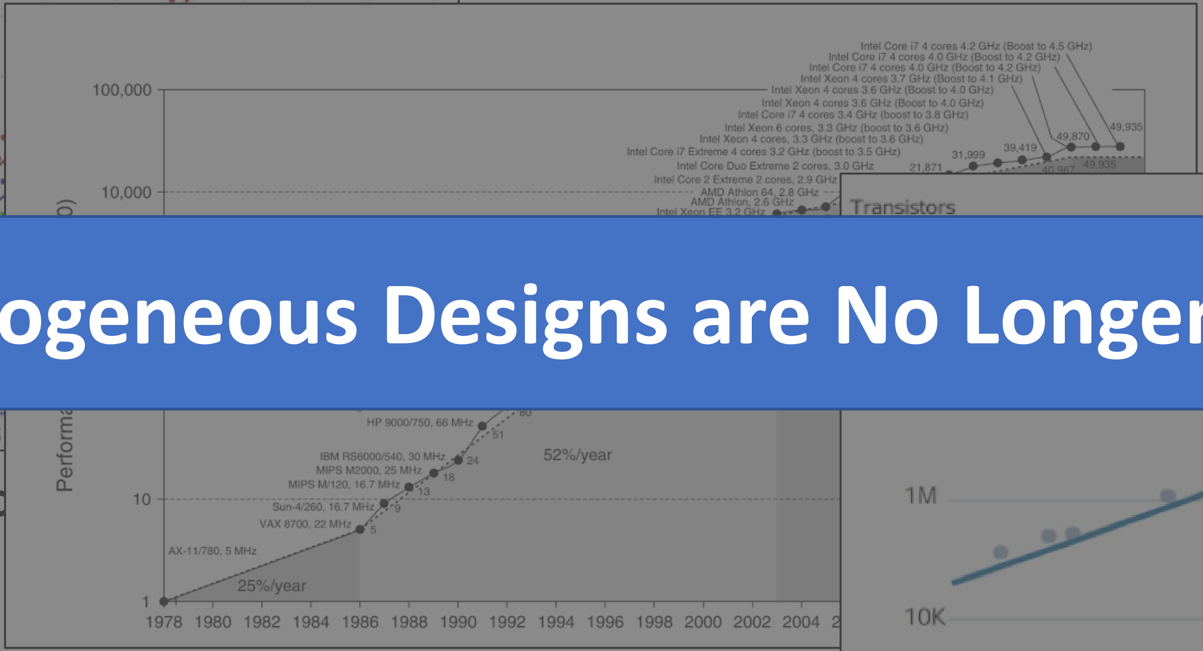
Cost of Design is Exploding.

The future of computing beyond Moore's Law, Volume: 378, Issue: 2166, DOI: (10.1098/rsta.2019.0061)
J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach, 6th edition, Morgan Kauffman, San Francisco, 2019.

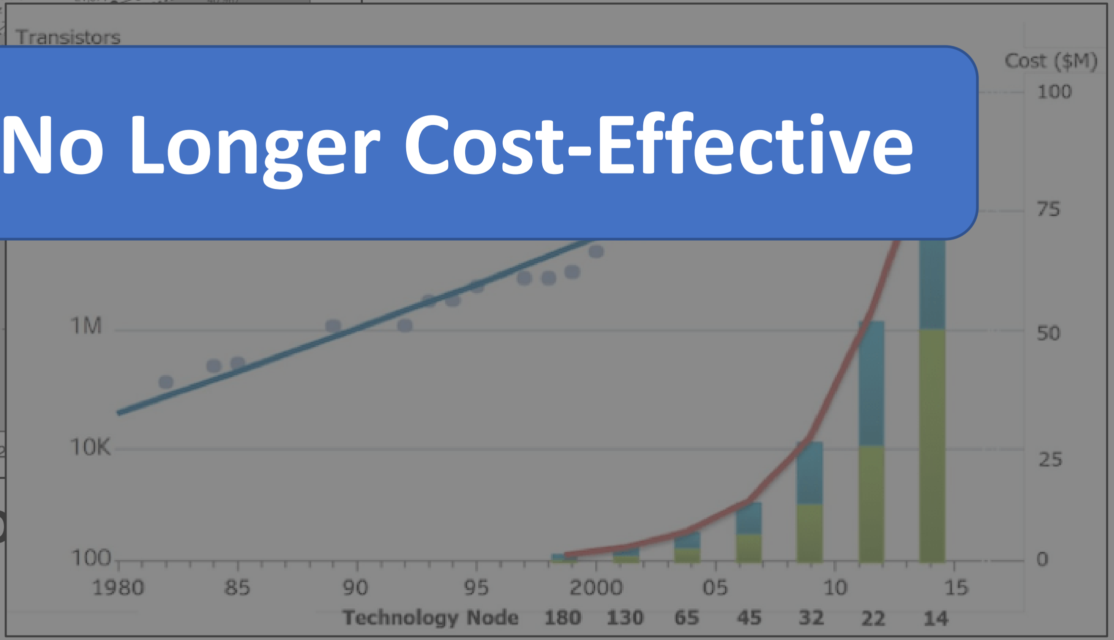
The Death of Homogeneous Designs



Classical technologic



CPU Performance Scaling is D



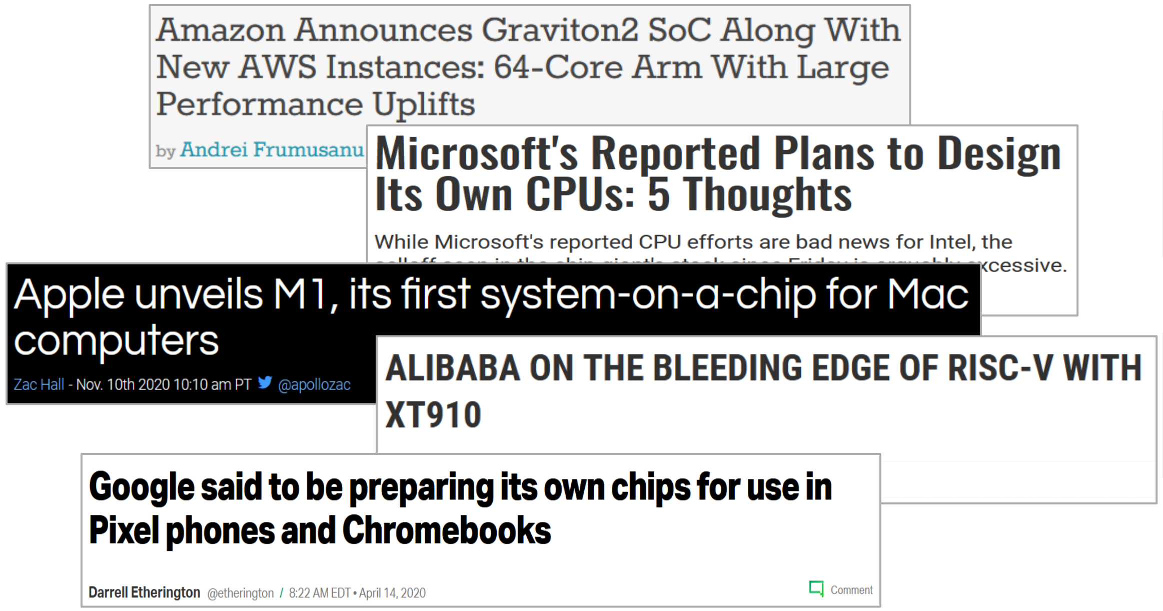
Homogeneous Designs are No Longer Cost-Effective

Cost of Design is Exploding.

The future of computing beyond Moore's Law, Volume: 378, Issue: 2166, DOI: (10.1098/rsta.2019.0061)
J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach, 6th edition, Morgan Kauffman, San Francisco, 2019.

The Era of Heterogeneous Designs

Increasing Amount of Hardware Designed,
Customized, and Tailored for Specific Applications.



Amazon Announces Graviton2 SoC Along With New AWS Instances: 64-Core Arm With Large Performance Uplifts

by Andrei Frumusanu

Microsoft's Reported Plans to Design Its Own CPUs: 5 Thoughts

While Microsoft's reported CPU efforts are bad news for Intel, the xcessive.

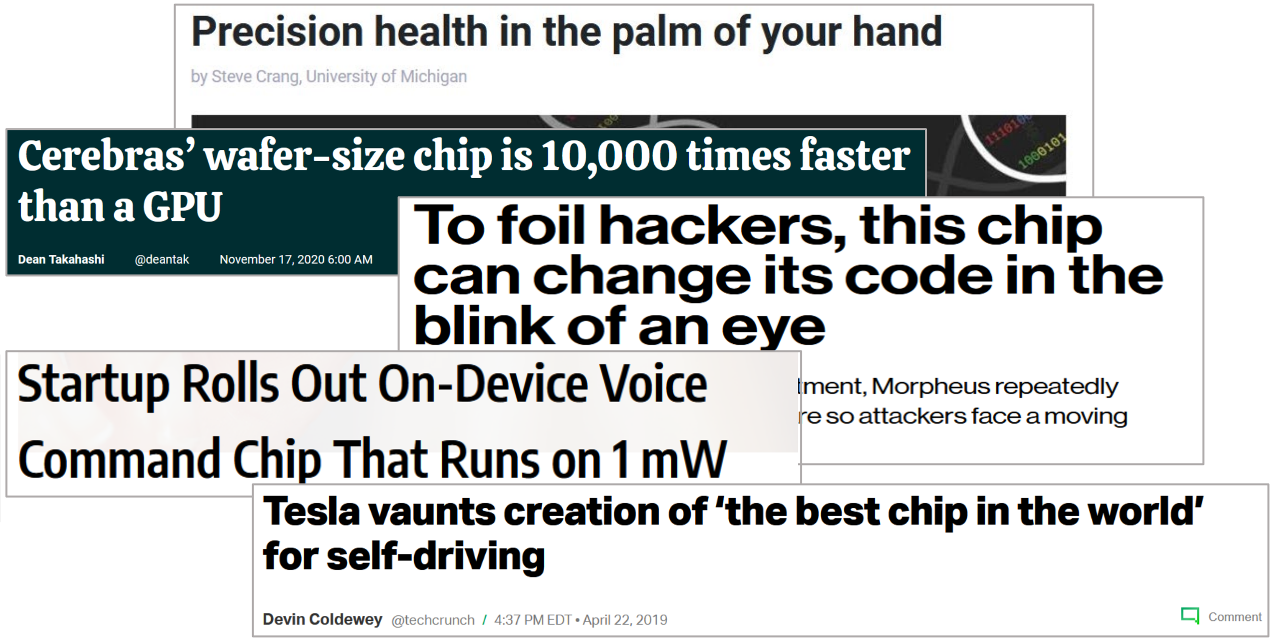
Apple unveils M1, its first system-on-a-chip for Mac computers

Zac Hall - Nov. 10th 2020 10:10 am PT @apollozac

ALIBABA ON THE BLEEDING EDGE OF RISC-V WITH XT910

Google said to be preparing its own chips for use in Pixel phones and Chromebooks

Darrell Etherington @etherington / 8:22 AM EDT • April 14, 2020 Comment



Precision health in the palm of your hand

by Steve Crang, University of Michigan

Cerebras' wafer-size chip is 10,000 times faster than a GPU

Dean Takahashi @deantak November 17, 2020 6:00 AM

To foil hackers, this chip can change its code in the blink of an eye

tment, Morpheus repeatedly
re so attackers face a moving

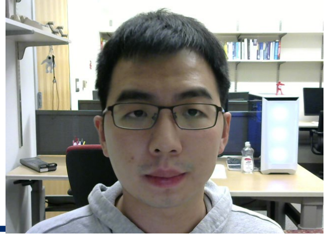
Startup Rolls Out On-Device Voice Command Chip That Runs on 1 mW

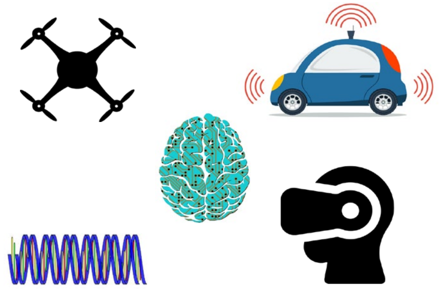
Tesla vaunts creation of 'the best chip in the world' for self-driving

Devin Coldewey @techcrunch / 4:37 PM EDT • April 22, 2019 Comment

Customized SoC *Application-specific Hardware*

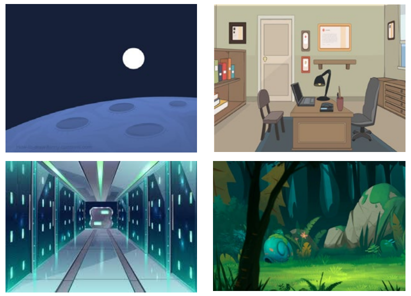
Meeting Distinct Requirements





Various Algorithms

+



Diverse Settings

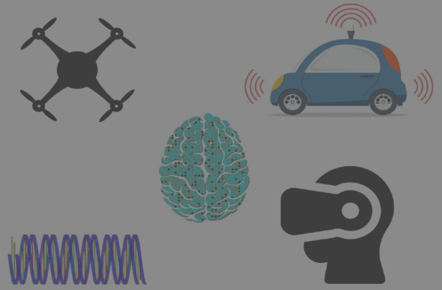
+

Different Technologies

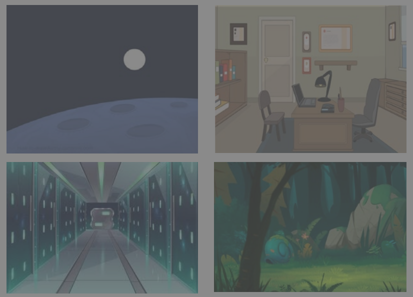
Distinct Performance, Area, Power, and Cost Requirements

Different Designs, Topologies, Functionalities

Meeting Distinct Requirements



Various Algori



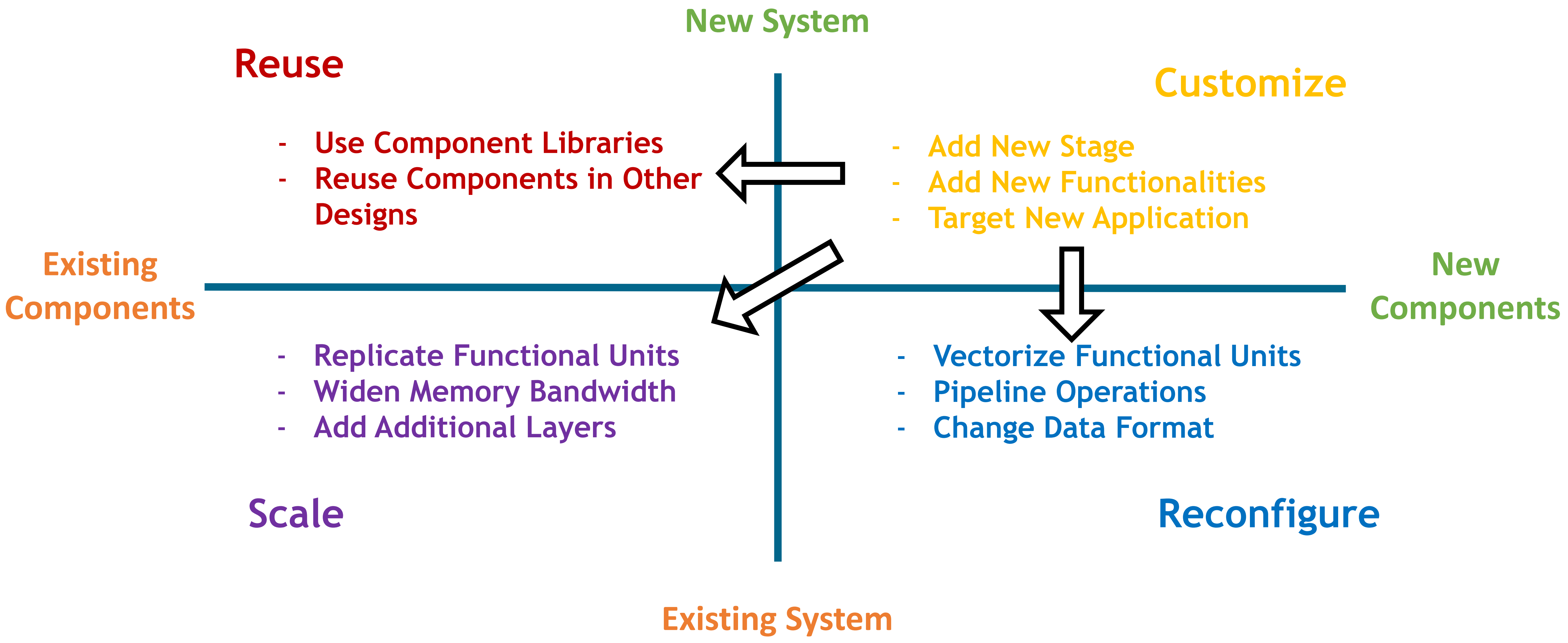
hnologies

How Do Developers Create New Designs?

Distinct Performance, Area, Power, and Cost Requirements

Different Designs, Topologies, Functionalities

The Zen of Heterogeneous Design



New System

Reuse

- Use Component Libraries
- Reuse Components in Other Designs

Customize

- Add New Stage
- Add New Functionalities
- Target New Application

Existing Components

New Components

- Replicate Functional Units
- Widen Memory Bandwidth
- Add Additional Layers

Scale

- Vectorize Functional Units
- Pipeline Operations
- Change Data Format

Reconfigure

Existing System

The Zen of Heterogeneous Design

New System

Reuse

- Use Component Libraries
- Reuse Components in Other

Customize

- Add New Stage
- Add New Functionalities

Existing Componen

Modern Hardware Design Languages Should Help Developers Efficiently Complete These Jobs

New mponents

- Widen Memory Bandwidth

- Pipeline Operations
- Change Data Format

Scale

Reconfigure

Existing System

Our Solution: Twine

Twine is a Chisel extension for *component-level* heterogeneous designs.

Twine

Twine supports essential features for heterogeneous design:

- **Standardize Control Interfaces (reusability, scalability)**
- **High-level Operator for Composability (scalability, reconfigurability, customizability)**
- **Automate Control Coordination & Data Type Conversion (scalability, reconfigurability)**
- **Low Level Access to Chisel Primitives (reconfigurability, customizability)**

Content

Content

Twine Standard Control Interfaces

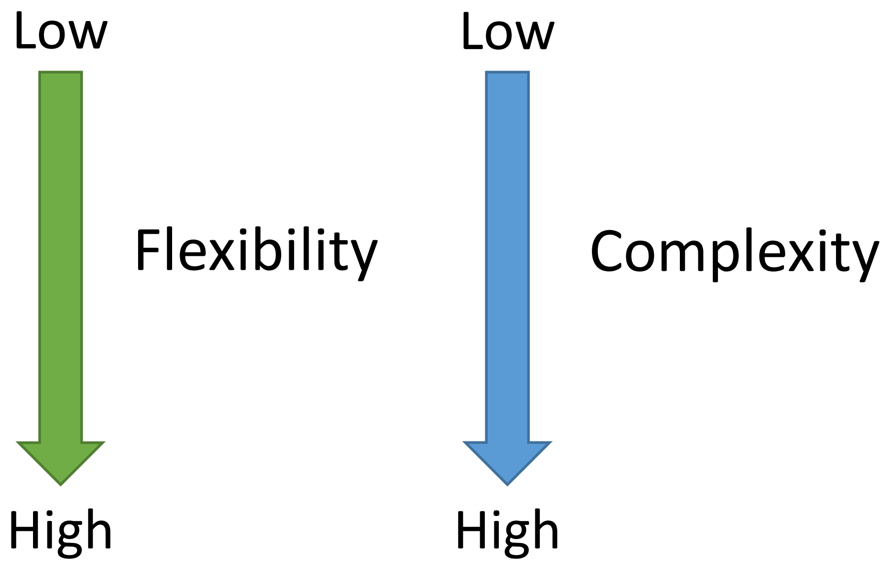
- **Interfaces define how a component communicates.**
- **Standardizing interfaces is a common practice.**
  - *Many standard interfaces for coarse-grained components (e.g., AXI, PCIe).*
  - *Too heavy for intra-accelerator communication.*
- **Naive approach: without standard control interfaces**
  - *Inspect, examine, and adapt component interfaces every time.*
  - *Automation is not straightforward, requiring significant designer effort and debugging*
- **Better approach: standard control interfaces**
  - *Make component behaviors more predictable.*
  - *Enable high-level automation.*

Twine Standard Control Interfaces

- **Declaration of a Twine Module Interface**

```
val in = IO(new ModuleInputType) // All data in-flow ports
val out = IO(new ModuleOutputType) // All data out-flow ports
val ctrl = IO(new ModuleCtrlType) // One of four standard control Interfaces
```

- **Four Standard Control Interfaces in Twine**
  - TightlyCoupledIOCtrl
  - ValidIOCtrl
  - DecoupledIOCtrl
  - OutOfOrderIOCtrl



Content

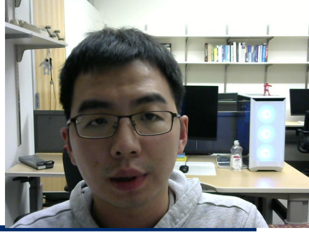
High-level Operator for Composability

- New *flow* operator >>> to distinguish from the original Chisel wire connection
  - Producer >>> Consumer
  - Supports all levels of granularity
  - *moduleA >>> moduleB*, *wireA >>> wireB*, *Bundle(wireA, wireB) >>> moduleA*

- Focus on producer/consumer relations
  - *Producer:* module that outputs completed values
  - *Consumer:* module that takes values as inputs (or needs to know when a value has been taken)

- Automatically inferred from the dataflow of the design

Content



Automate Control Coordination & Data Type Conversion

- Automatically generate system-level control logic
  - Inferred based on dataflow and producer/consumer relations
  - Mix-and-match across different interfaces
  - Ability to manually control preserved

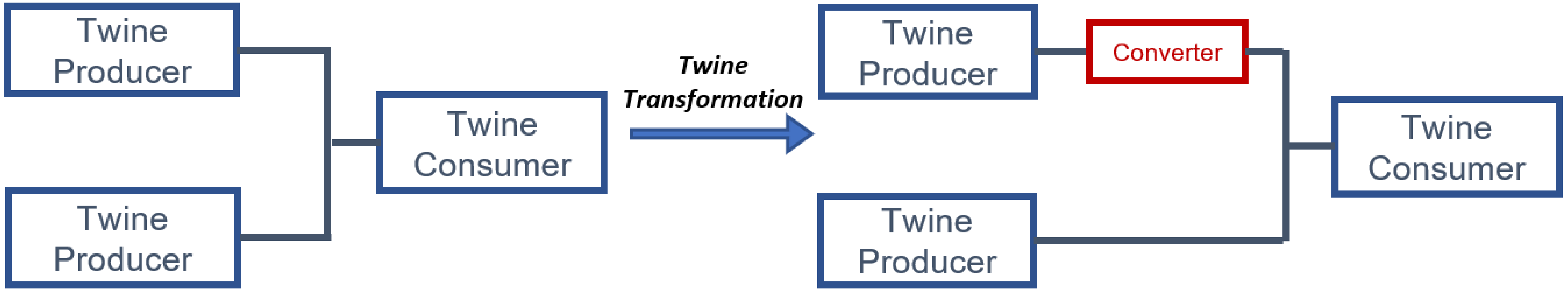
Automate Control Coordination & Data Type Conversion

- Automatically generate system-level control logic
  - Inferred based on dataflow and producer/consumer relations
  - Mix-and-match across different interfaces
  - Ability to manually control preserved

- Data Type Conversion
  - Auto conversion between different data types (*e.g.,* floating points <-> integers)
  - Auto conversion between different port width (useful for vectorized components)

Automate Data Type Conversion

- Simple conversion logic is combinational and transparent
  - *e.g., Unsigned Integers <-> Signed Integers, Bitwidth expansion*
- Complex conversion logic serves as a full converter module
  - Floating point to integer conversion
  - Serializer and de-serializer for vectorized components

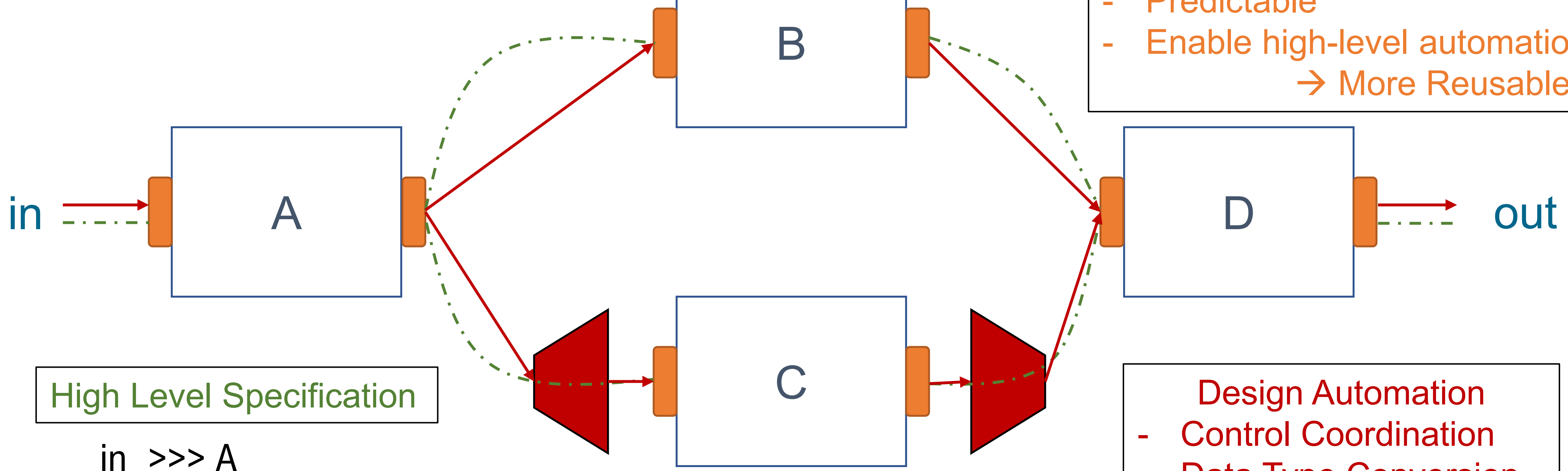


Put Them Together

Assume there are modules A, B, C, and D. Module C is a vector module.

Chisel Compatible

Implement one of four control interfaces →
- Predictable
- Enable high-level automation
  → More Reusable



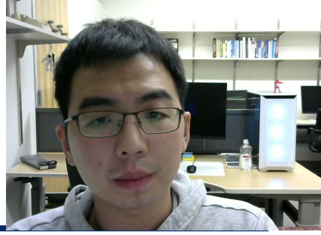
High Level Specification

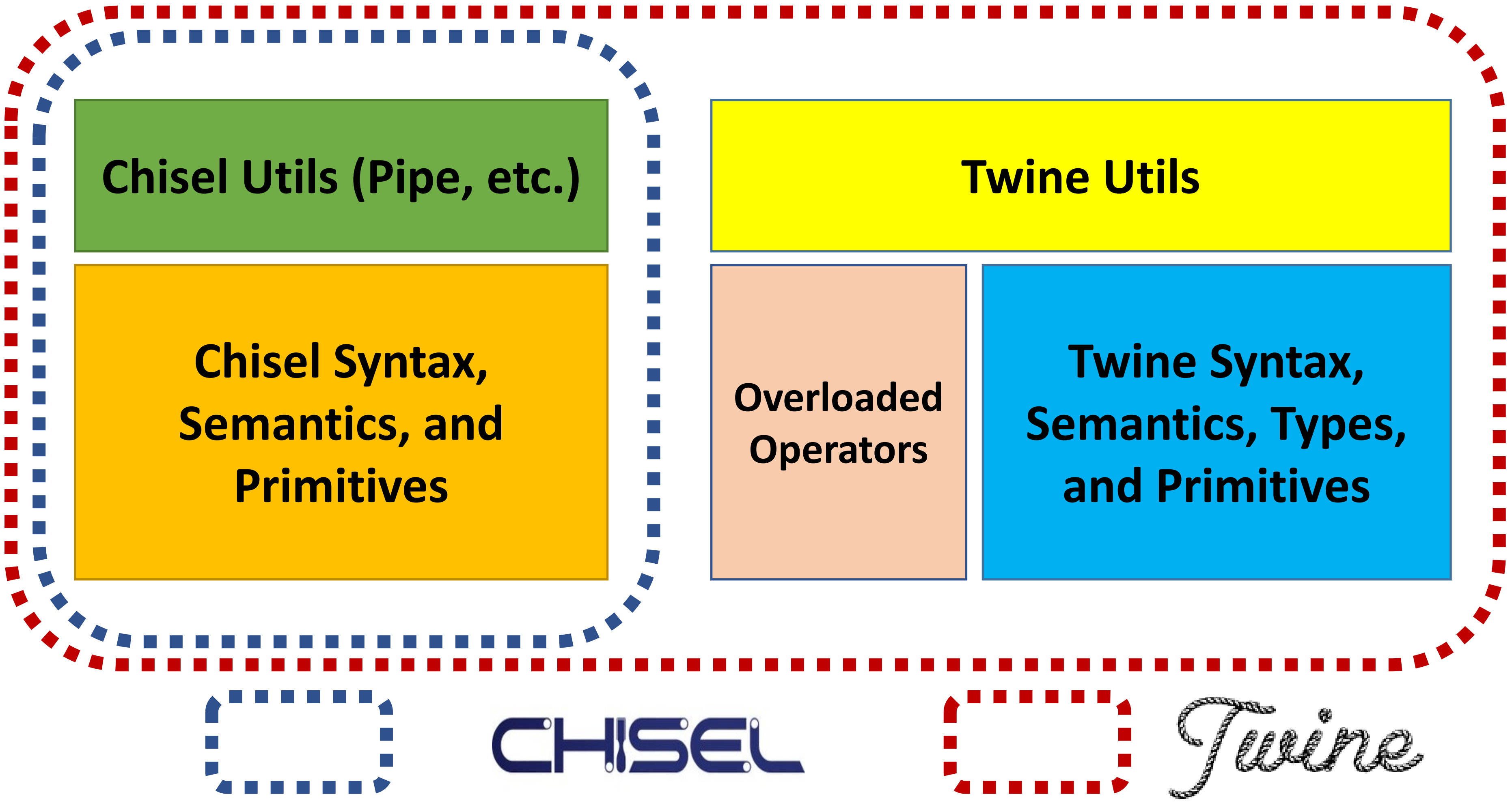
```
in  >>> A
A >>> B >>> D.1
A >>> C >>> D.2
D >>> out
```

Design Automation
- Control Coordination
- Data Type Conversion

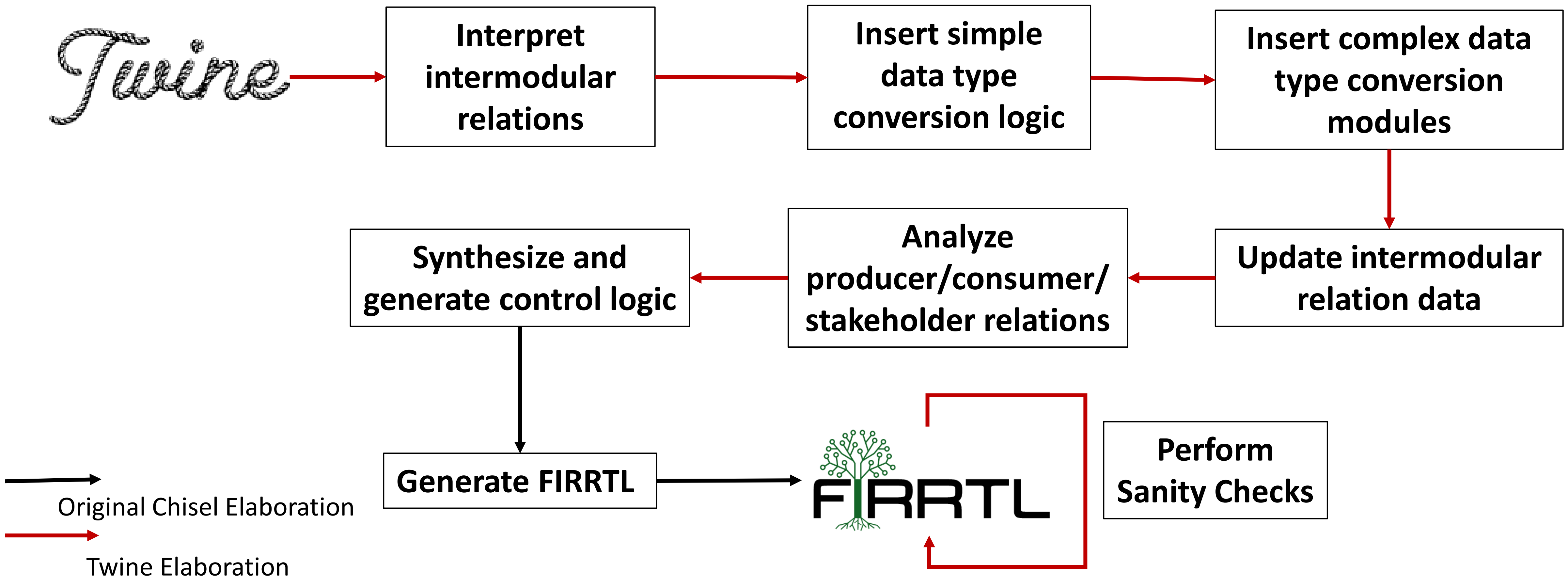
Content

Build Upon Existing Infrastructure & Preserve All Features



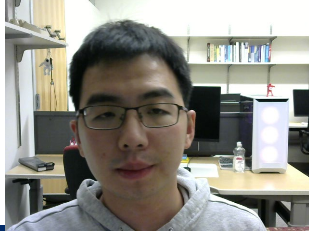

Chisel Utils (Pipe, etc.)
Chisel Syntax, Semantics, and Primitives
Twine Utils
Overloaded Operators
Twine Syntax, Semantics, Types, and Primitives
CHISEL
Twine

Twine Elaboration Pipeline

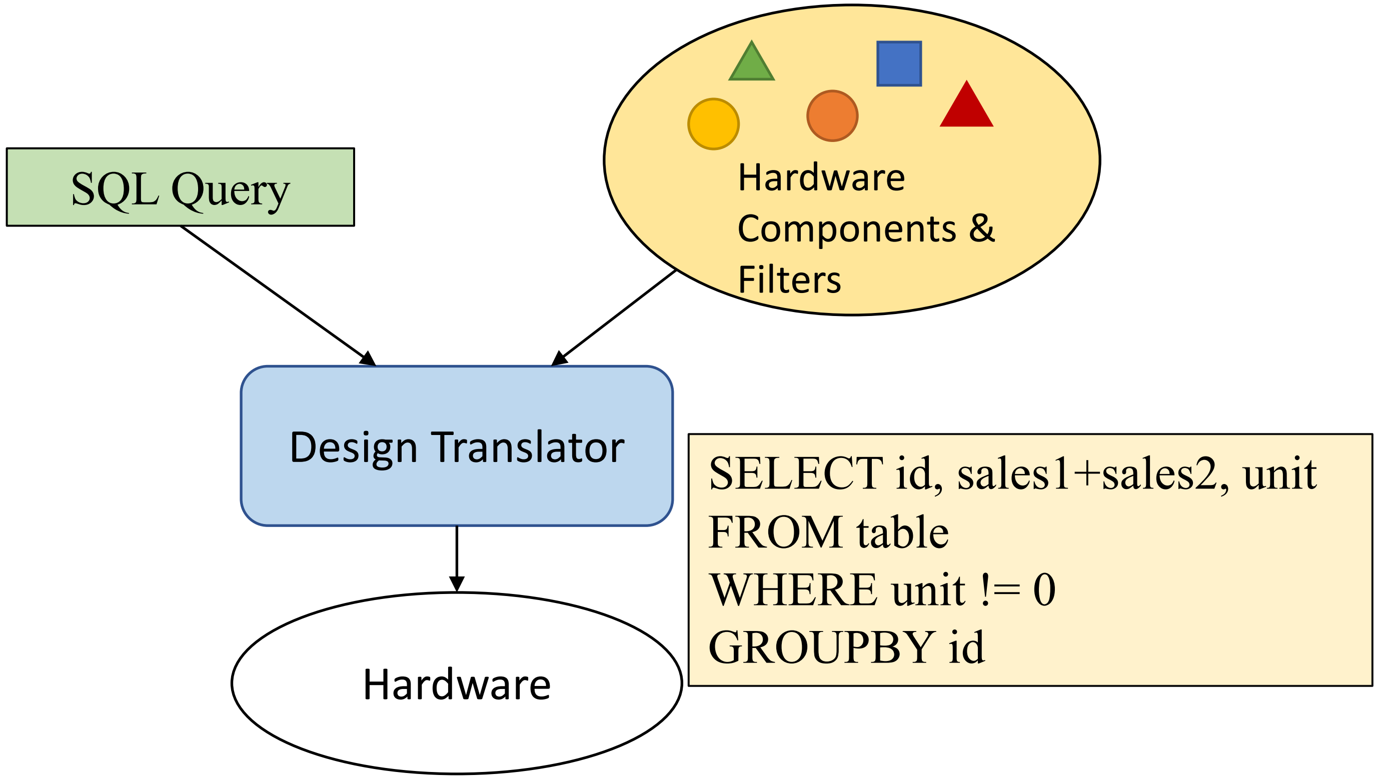

Twine
Interpret intermodular relations
Insert simple data type conversion logic
Insert complex data type conversion modules
Update intermodular relation data
Analyze producer/consumer/ stakeholder relations
Synthesize and generate control logic
Generate FIRRTL
FIRRTL
Perform Sanity Checks
Original Chisel Elaboration
Twine Elaboration

Content

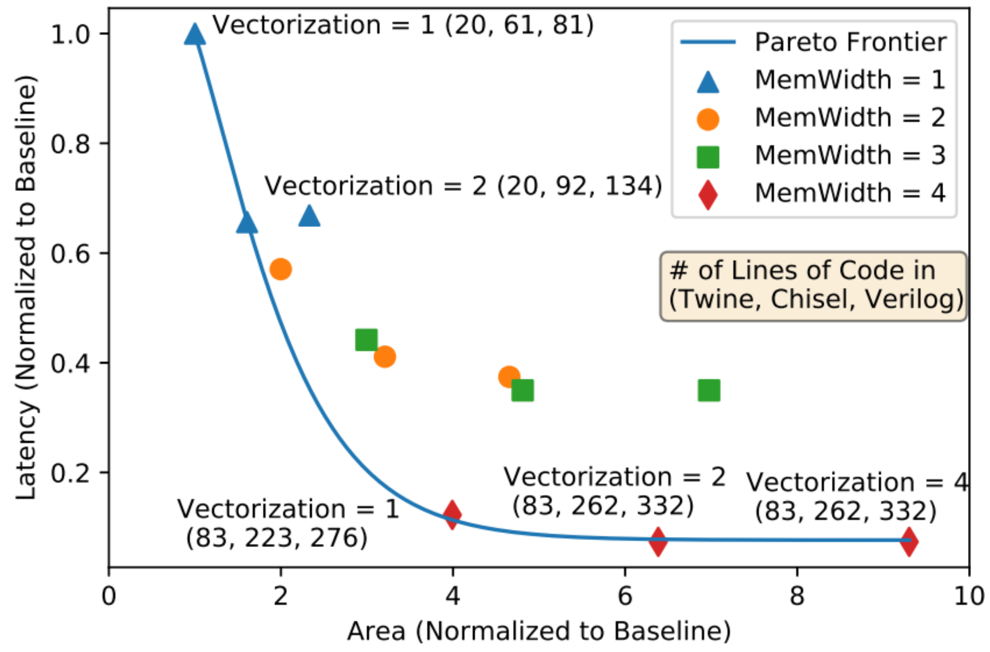
Experiment: Productivity Improvement



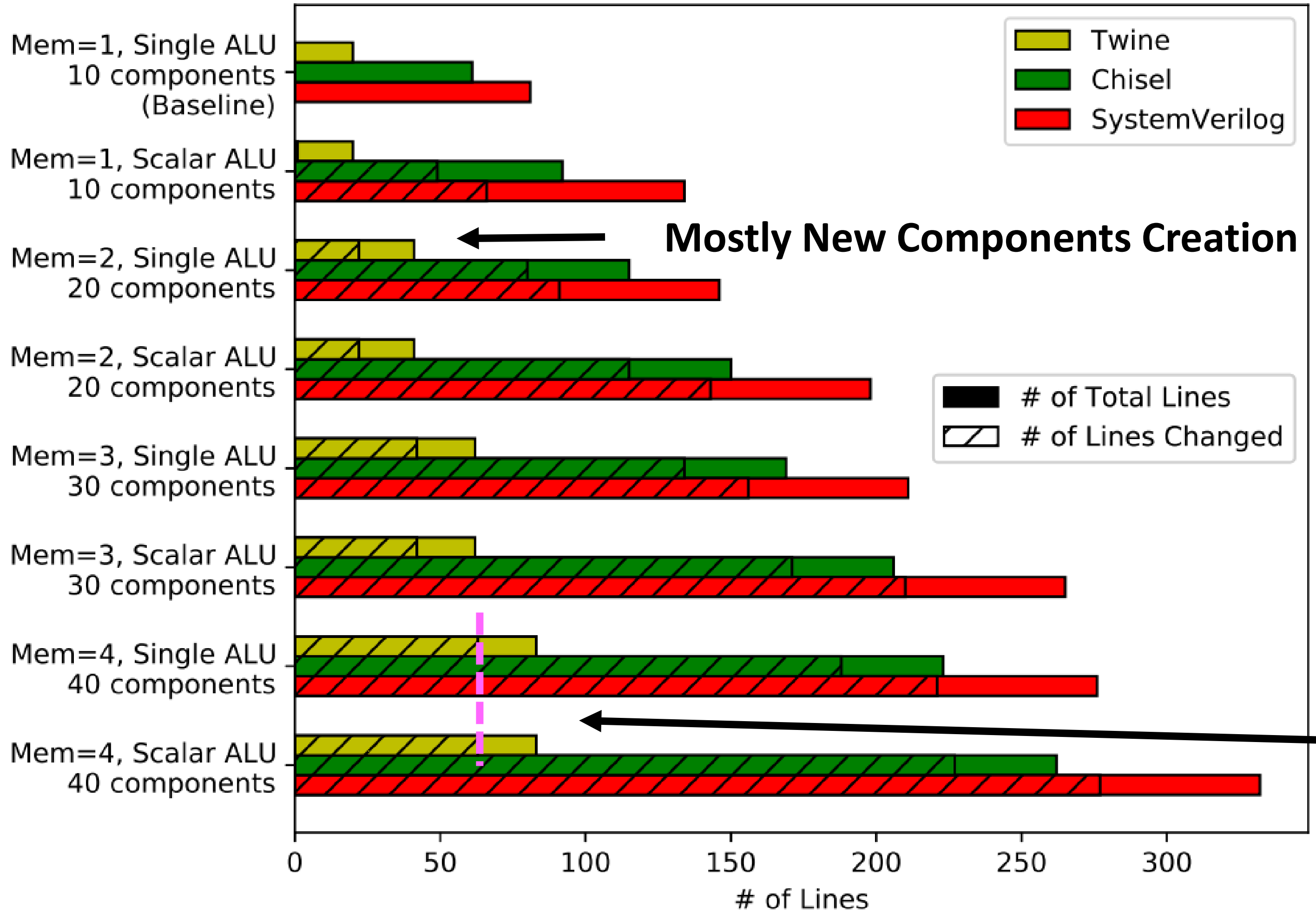
- **Prototyped a database query accelerator similar to Q100 (ASPLOS '14)**
- **Conducted design space exploration in Verilog, Chisel, and Twine**



```
SELECT id, sales1+sales2, unit
FROM table
WHERE unit != 0
GROUPBY id
```



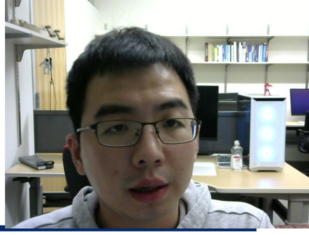
Experiment: Productivity Improvement


Mem=1, Single ALU 10 components (Baseline)
Mem=1, Scalar ALU 10 components
Mem=2, Single ALU 20 components
Mem=2, Scalar ALU 20 components
Mem=3, Single ALU 30 components
Mem=3, Scalar ALU 30 components
Mem=4, Single ALU 40 components
Mem=4, Scalar ALU 40 components
Twine
Chisel
SystemVerilog
# of Total Lines
# of Lines Changed
Mostly New Components Creation
0
50
100
150
200
250
300
# of Lines

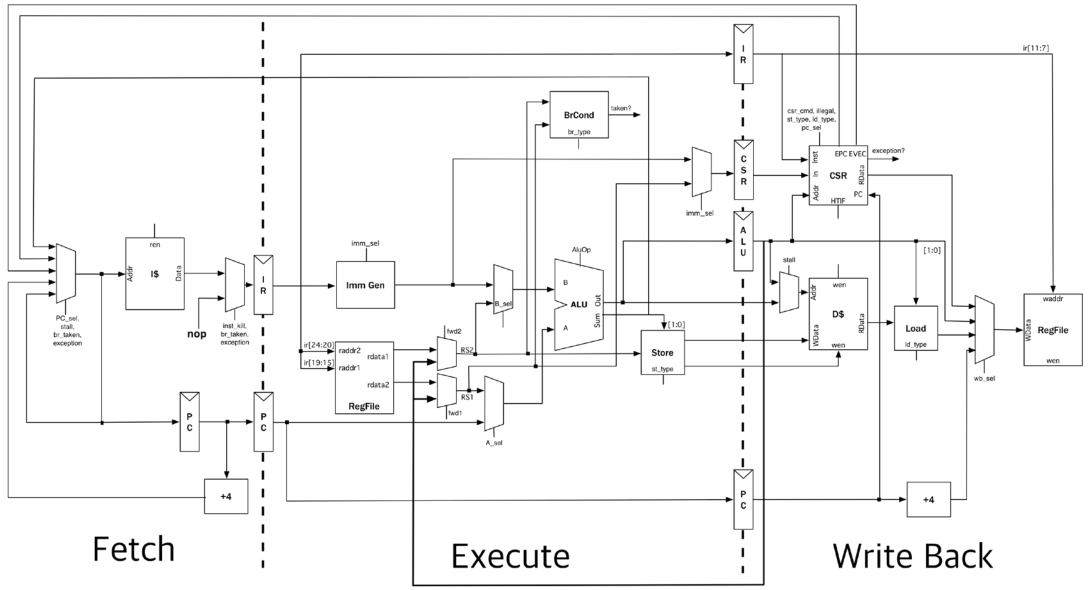
No change between different degree of vectorization

- **Much fewer lines of code (~1/3 of the designs in Chisel)**
- **Number of lines changed between designs is low**

Experiment: Design Quality



- Reproduced RISCV-MINI, a three-stage RISCV core in Twine
- Components interfaced with DecoupledIOCtrl



	Area*	Clock Period*
Chisel	727004.94	0.85 ns
Twine	725937.90	0.82 ns
Change	-0.14%	-3.5%

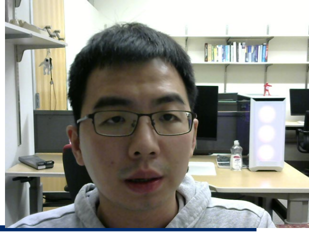
RISCV-MINI: https://github.com/ucb-bar/riscv-mini

*Based on IBM 45nm CMOS Process

Content

- Motivation
- Twine Features
- Implementation & Circuit Generation
- Experiments & Results
- **Limitations & Future work**
- Conclusion

Limitations



- **Inflexible processing granularity for vectorized modules**

- **Missed opportunities in inter-module optimizations**
  - Possible out-of-order execution or forwarding across the module boundary

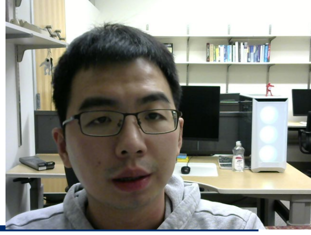
Future Research Directions

- **Better verification and debugging capabilities for Twine**
  - Utilize the producer/consumer relations to speed up verification process
- **Flexible & customizable interface protocol framework**
  - User-defined interfaces and elaboration process

Content

Conclusion

Twine



- **Twine is a Chisel extension that supports**
  - reusable standard component control interfaces
  - high-level operator for composability
  - control coordination & data type conversion automation

- **Twine boosts developer productivity for heterogeneous designs.**
  - 1/3 of lines of codes compared to Chisel

- **Twine provides similar design quality comparing to Chisel.**

- **Visit https://github.com/Twine-Umich/Twine to download Twine.**

Q & A

Twine is an open-source project.

To download Twine, please visit https://github.com/Twine-Umich/Twine

All feedbacks are welcomed!