Iowa State University
College of Engineering

CyGRA

Open Coarse-Grained Reconfigurable Architecture Chip

sdmay25-28

| 05/07/25

Team Introduction

Camden Fergen
Project Lead and DevOps

Software Engineering

John Huaracha
Testing Lead

Computer Engineering

Nicholas Lynch
Hardening and Verification Lead

Computer Engineering

Calvin Smith
Accelerator Design Lead

Computer Engineering

Levi Wenck
Communication Interfaces Lead

Computer Engineering

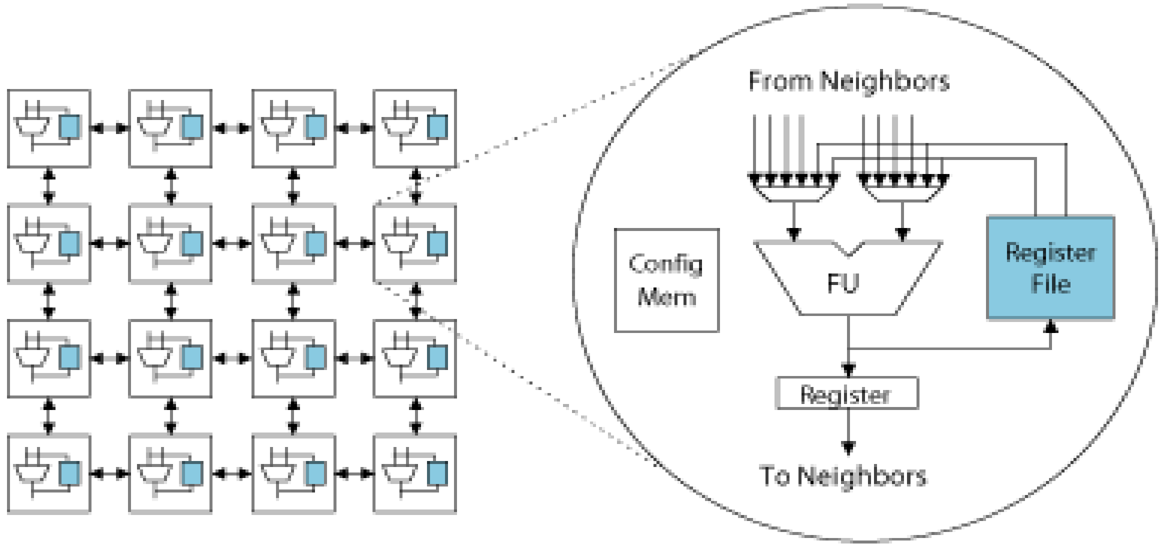
Project Overview

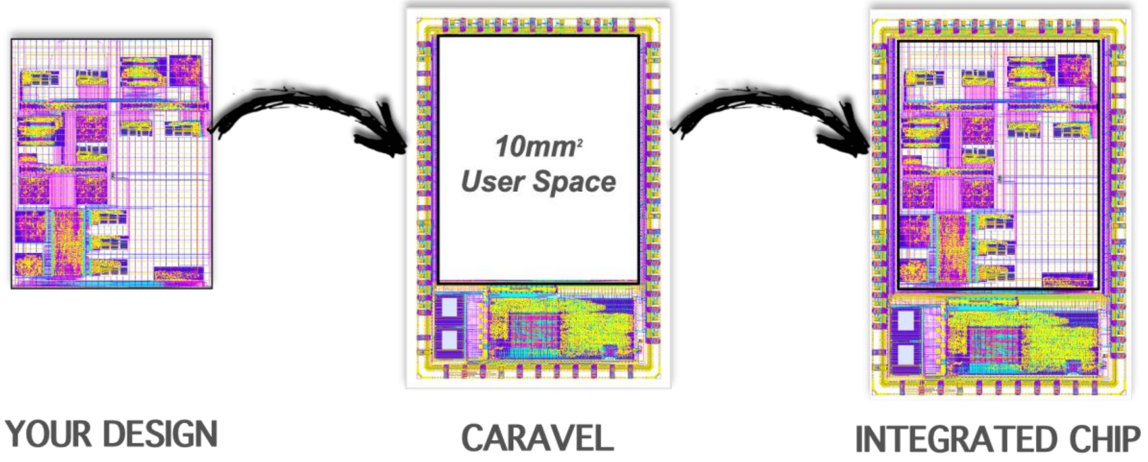
Problem

- Processors constrained by typical ISAs
- No open-source CGRAs (SNAFU)
- Interest in reconfigurable computing

Solution

- Fully integrated SoC (system on chip)
    - Full memory system (off chip and on chip)
    - Support for I/O
- Coarse-Grained Reconfigurable Architecture
    - CyGRA co-processor





Project Users

ISU Chip Forge Club

Hardware bring up and verification tool for new users

Hardware Design Students

Learn and understand the processes of HDL to fabrication

Professors

Used as a teaching aid in classes working with ISAs

Iowa State University
College of Engineering

Project Requirements

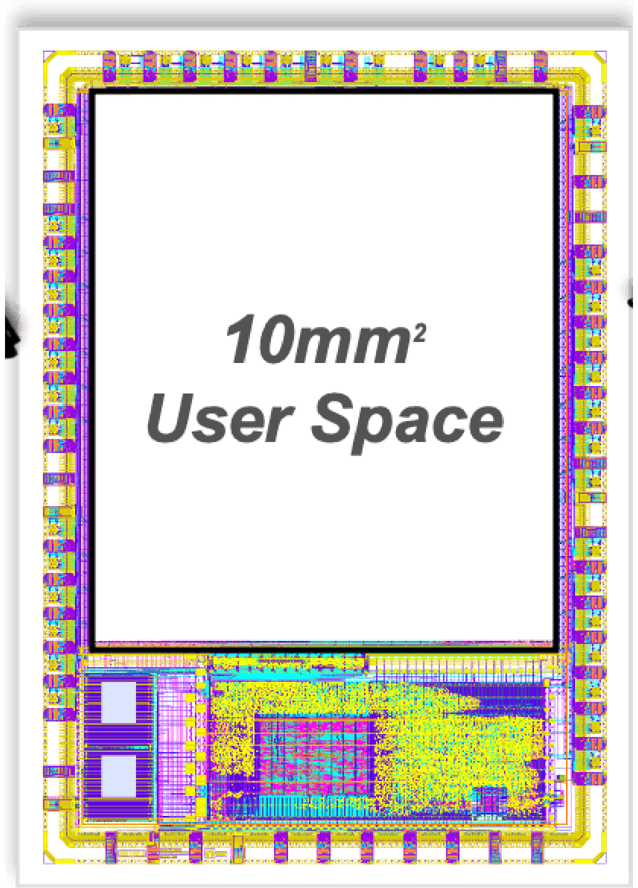
Functional Requirements

- Must support standard instructions
  - Design contains an open-source general-purpose processor
- Must support custom instructions
  - Users should be able to define custom configurations for the CyGRA
  - Users must be able to run computations on the CyGRA
  - Must interface with RISC-V processor with little to no overhead
- CyGRA unit should increase performance
  - Expected speedup of 1x-10x faster using the CyGRA

RISC-V: The Free and Open RISC Instruction Set Architecture

Non-functional Requirements

- Project should be user friendly
  - Clear documentation and interface
- Custom instructions should adhere to open-source ISA standards
- Chip Fabrication via Efabless
  - Verilog
  - Firmware in C
  - OpenLANE
  - Skywater 130 nm Process
  - Must fit within a ~$10mm^2$ wrapper (2920 µm x 3520 µm)



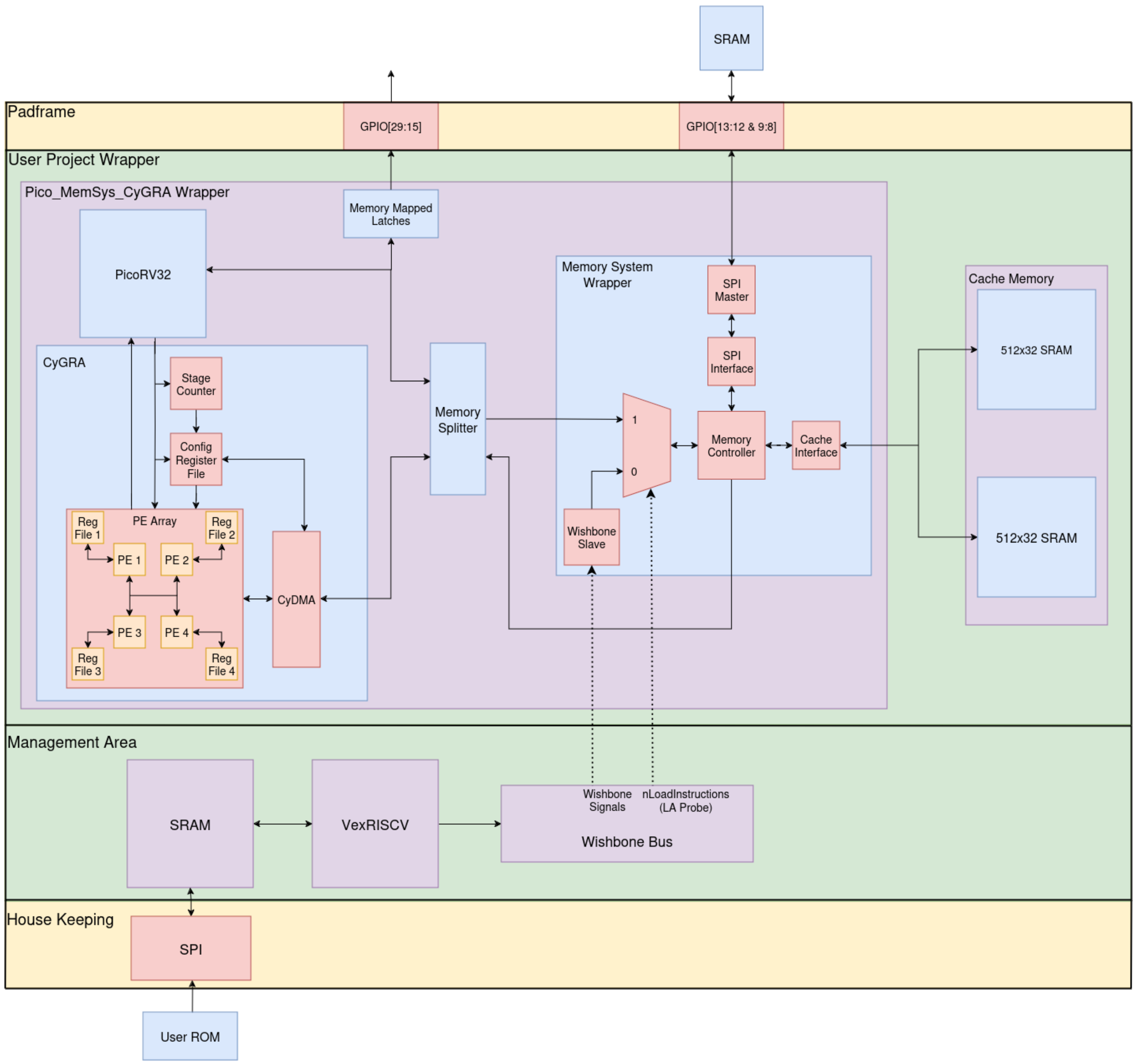
CARAVEL

Iowa State University
College of Engineering

Project Design

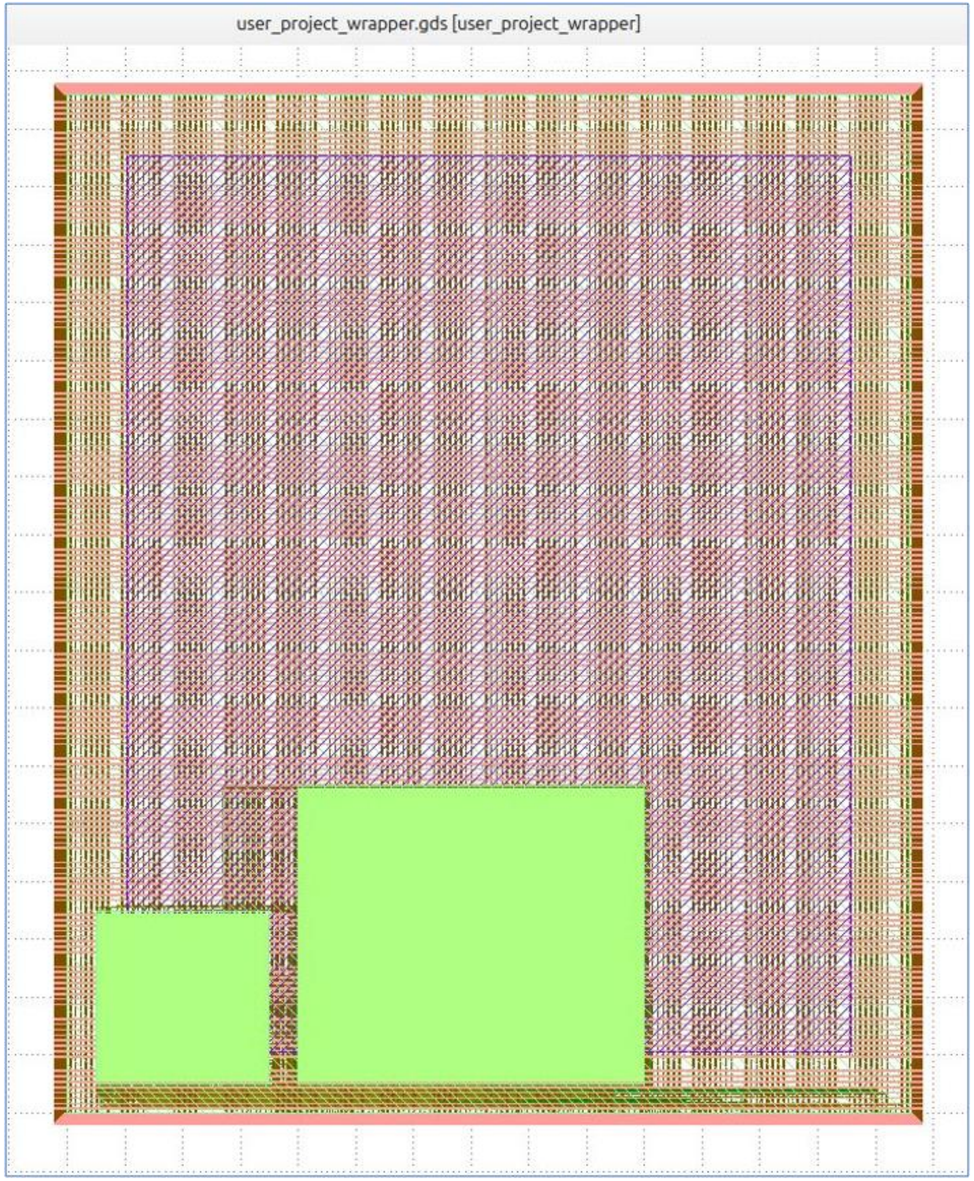
High Level Design

- PicoRV32
  - Staged RISC-V processor
- CyGRA co-processor
  - CGRA based acceleration unit
- Cache controller
  - Interfaces with both on- and off-chip memory
- Cache/SPI Master interface
- Memory splitter



Design Tradeoffs

- Space constraints
  - Max user project area ~10mm^2
  - Influenced softcore decision
- Off-chip memory
  - Needed to ensure we had ample memory for running programs
  - Much higher memory access times
- Limited CyGRA size
  - Had to settle for a 2x2 opposed to a 3x3



2KB DFFRAM

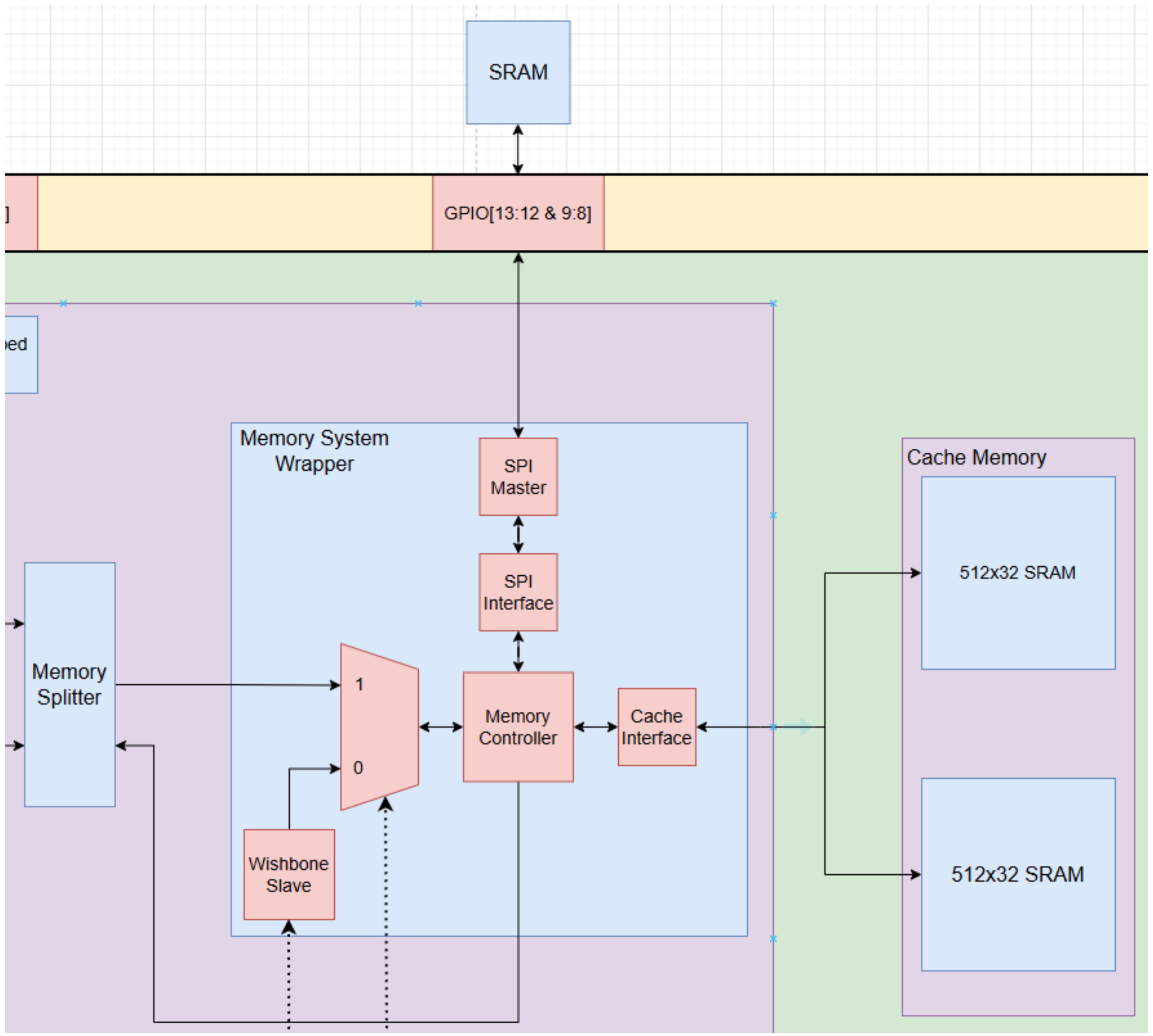
Microprocessor: PicoRV32

- Open-source RISC-V processor
- 32-bit ISA, further space optimized
- Built-in support for custom instructions
  - Utilizes a co-processor interface, easy integration
- Written in Verilog, easy integration with Caravel project
- Built-in wishbone interface

Rocket Chip	Neorv32	Vex RISC-V	CVA6	**PicoRV32**
- RISC-V 64 bit - Feature rich - Not size optimized - Written in chisel	- RISC-V 32 bit - Designed as a microcontroller - Not size optimized - Written in VHDL	- RISC-V 32 bit - Highly customizable - Not size optimized - Written in SpinalHDL	- RISC-V 64 bit - Support for UNIX-like operating systems - Not size optimized - Written in System Verilog	- RISC-V 32 bit - Wishbone interface and high clock frequency - Size optimized - Written in Verilog

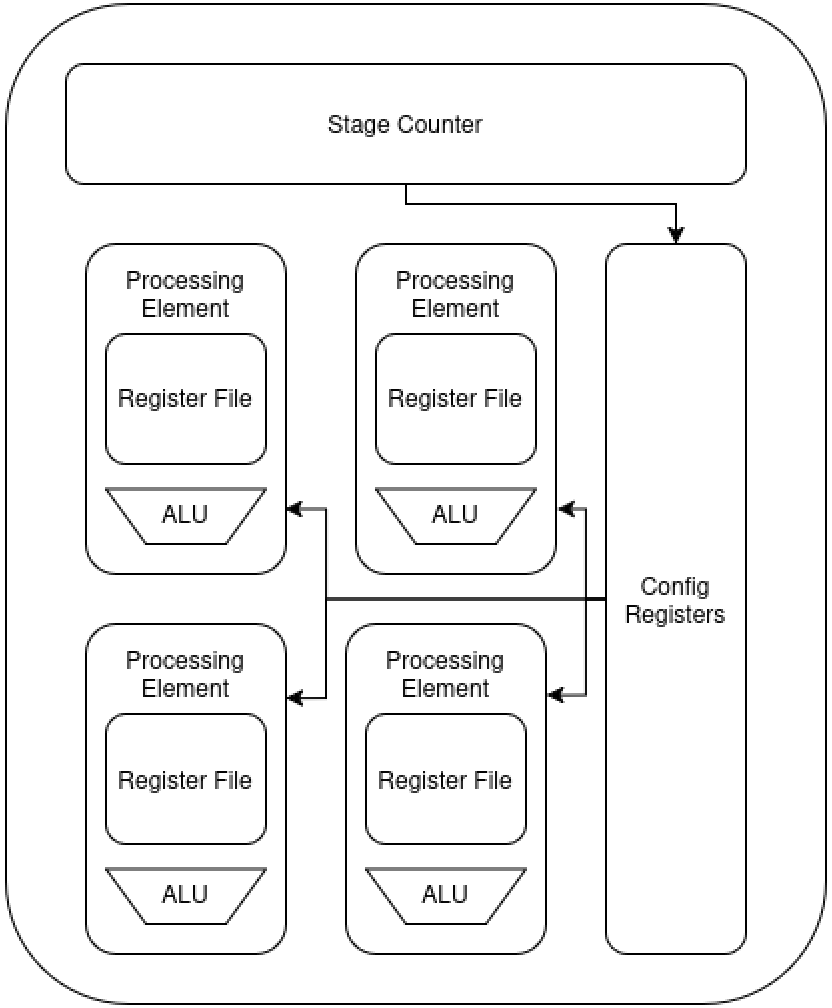
Memory

- Cache Controller
  - Write-Through
  - Greatly decreases overall memory read times
- Cache memory (on-chip)
  - Two 512x32 bit DFFRAM modules
  - Read hit : 4 clocks
- SRAM (off-chip)
  - 512 Kbit SRAM PMOD module
  - Read miss and writes : > 128 clocks



CyGRA Co-processor

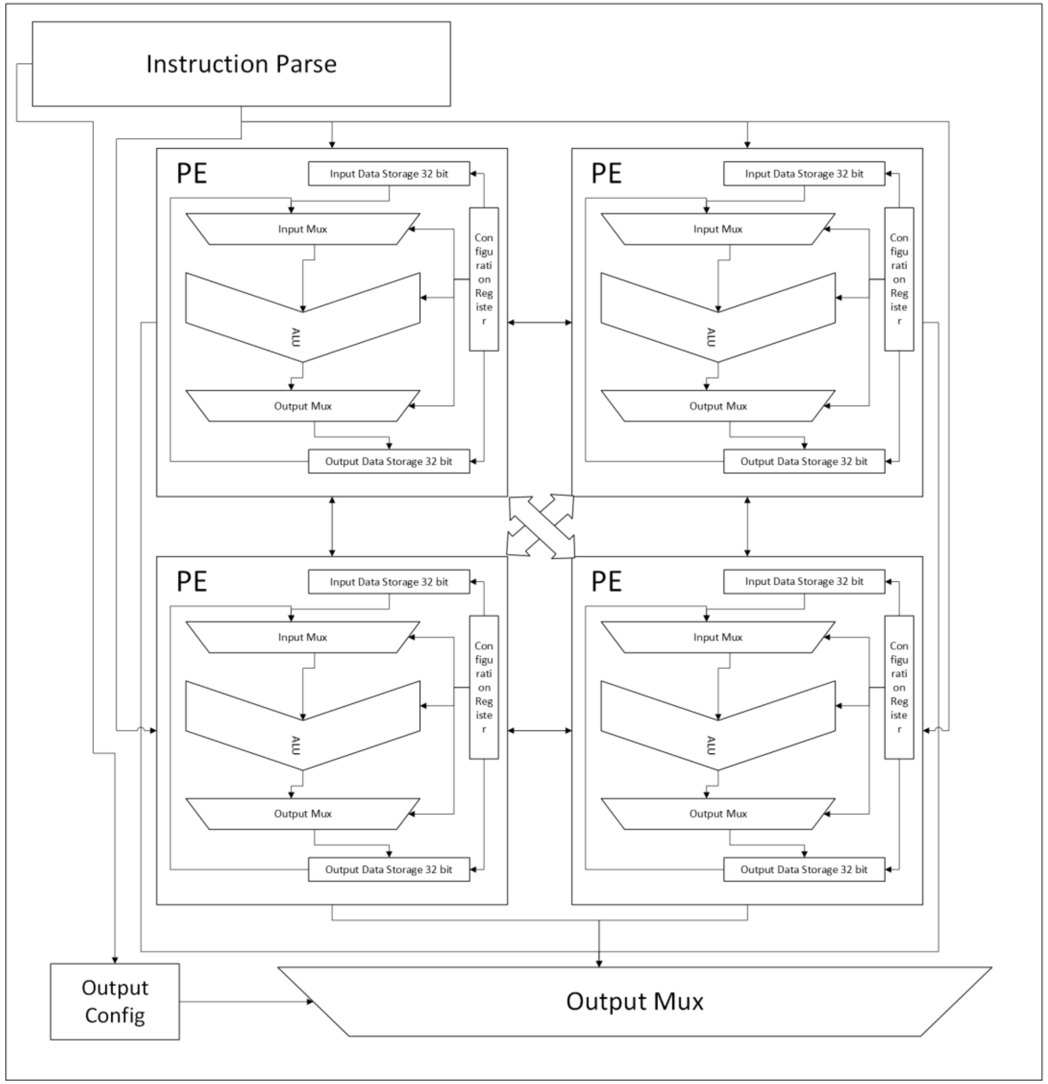
- 2x2 ALU + Register File array
- 4 Arithmetic operations per cycle (*, +, -)
- Fixed point arithmetic
- 16 Cycle dataflow + operation configurations
- Current mapped applications include
  - Fast Fourier Transform
  - Dot Product
  - Multiply-Accumulate
  - Matrix Multiply



CyGRA Version 3, simplified diagram

CyGRA Revisions

- CyGRA Version 1
  - Single Cycle Configuration
  - Small local storage
- CyGRA Version 2
  - Multi-cycle applications in RTL
  - Register Files (expensive multiple write ports)
- CyGRA Version 3
  - Multi-Cycle Configuration
  - Greater memory access capabilities (DMA I/O)
  - Single write port Register Files
  - Retains complex mappings



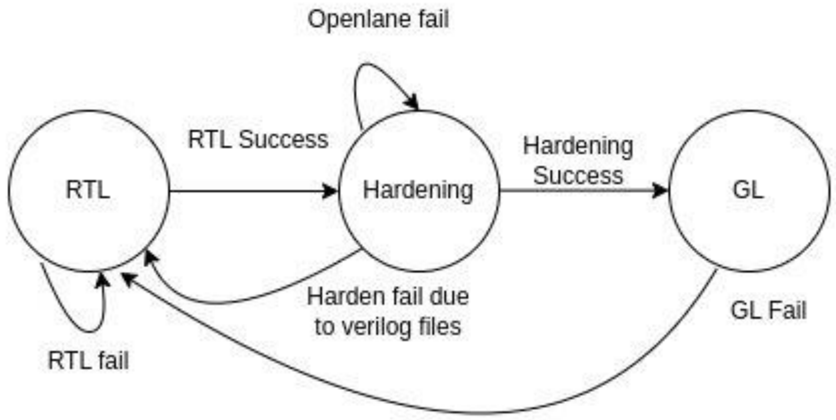
CyGRA Version 1

Iowa State University
College of Engineering

Testing

Testing Plan

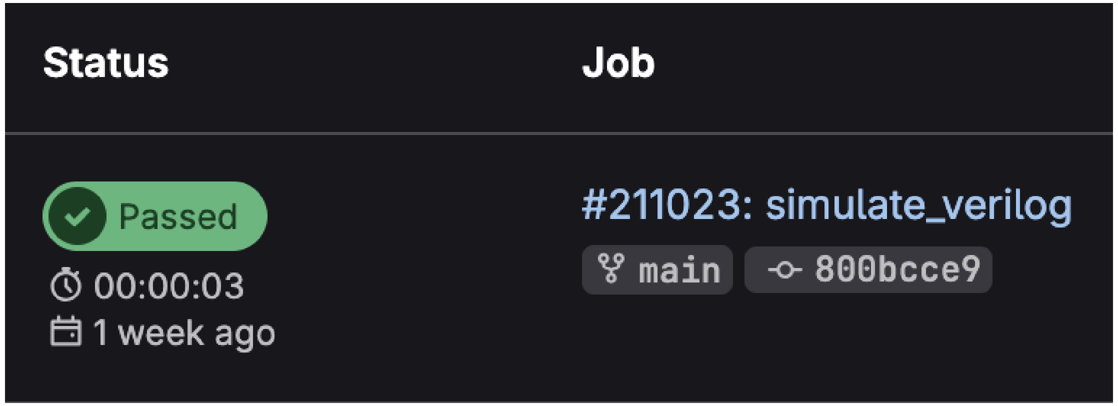
- Verilog testbenches for major components
  - Simulate with Questasim and check waveforms
- RTL tests
  - Synthesize design and run C files using Cocotb
- Gate-level tests
- FPGA testing
  - Use the ChipForge tooling to run C code on their FPGAs



State diagram showing testing flow for complete system

Regression Testing

- Custom CICD pipeline for all modules
  - Comprehensive testbenches on all Verilog files
  - Pipeline fails if any testbench does not complete
- CICD pipeline for entire system
  - Runs basic assembly tests on PicoRV32 to ensure nothing breaks
  - Can be utilized to test the CyGRA and cache interface to ensure integrations works as expected



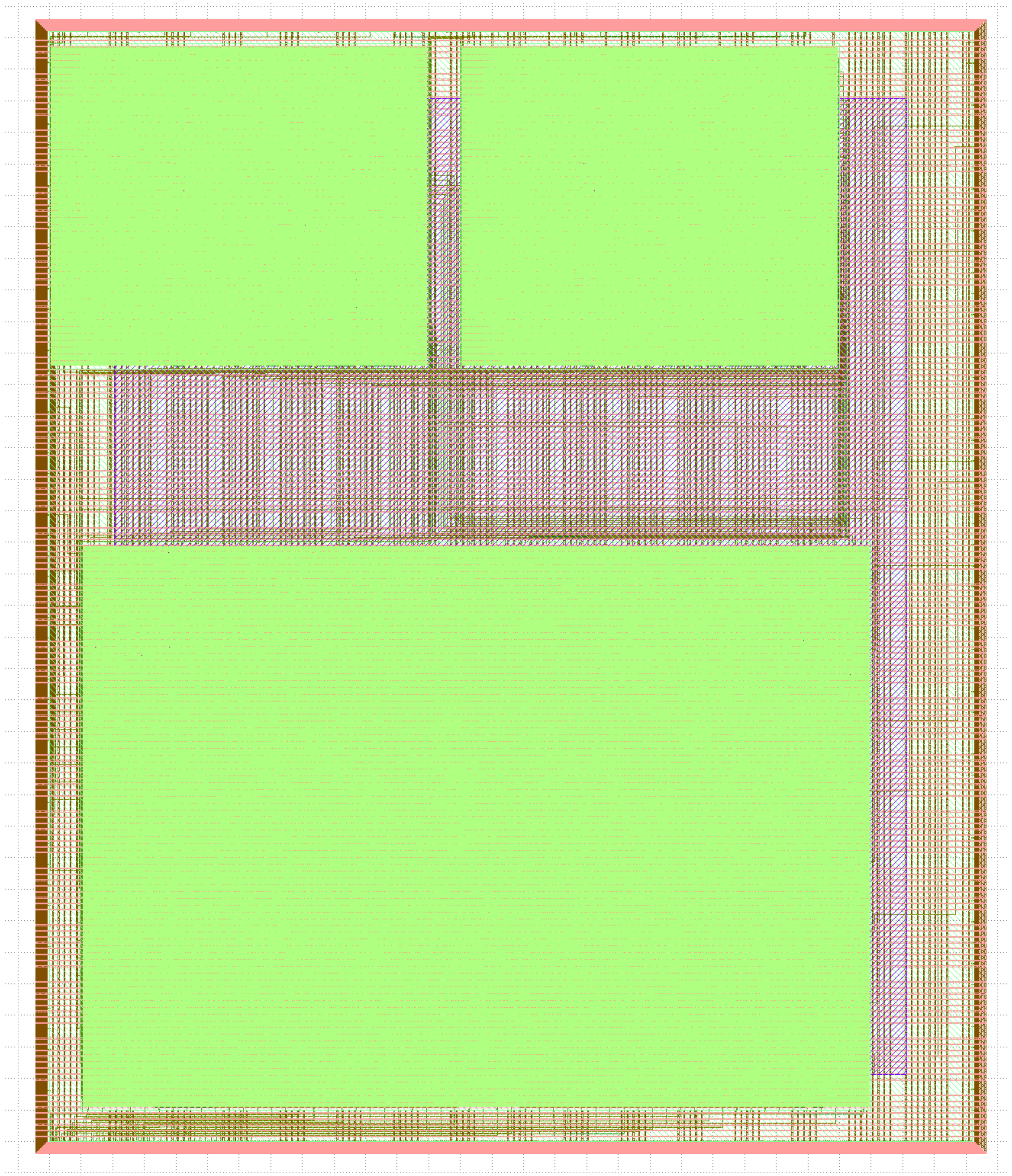

```
simulate_verilog:
  stage: simulate
  image: git.ece.iastate.edu:5050/sdmay25-28/spi_master/iverlog-test-d
  script:
    - mkdir -p output
    - |
      for test_file in tests/*.v; do
        test_name=$(basename "$test_file" .v)
        iverilog -o output/${test_name}.out "$test_file" spi_master.v;
        vvp output/${test_name}.out > output/${test_name}_log.txt;  #
      done
  artifacts:
    paths:
      - output/
      - output/*.txt  # Save all logs for verification
```

IOWA STATE UNIVERSITY
College of Engineering

Project Status

Hardened Design

- Fully synthesized and hardened
- Working RTL tests
- DRC and LVS checks passing
- Currently does not pass STA
  - Slack Violations
    - Decrease Clock Frequency
    - Add Latches/Clock Gates



RTL Simulations

- C code running
- Working bubblesort, mergesort, and FFT running on the PicoRV32
- CyGRA test file
    - MAC operation working
    - All CyGRA operations working

CyGRA Computing MAC
X = 0
MAC X, 3, 4 // X = X + 3 * 4

Merge-sort running on the PicoRV32 in a CocoTB test

Component Status

- **PICORV32**
  - Successfully integrated and running benchmarks in C
- **Memory System**
  - Verified through RTL simulation and functional testing
- **CyGRA**
  - Correctly loads configuration entries from memory
  - Performs data transfers (load/store) correctly to/from memory
  - Executes loaded configurations as expected, including MUL and MAC
- **CyGRA Interface**
  - Provides verified helper functions for simplified usage
  - Fully tested for integration and correctness

Challenges

- Time available/timeline
  - Underestimation of time for certain timeline sections
  - Unexpected delays due to lack of documentation on open-source tooling
- CyGRA Design
  - Complex design built from scratch
  - Multiple iterations close to project completion
- Efabless shutdown
  - Unexpected shutdown March 2nd, 2025
  - Lost technical support

Iowa State University
College of Engineering

Benchmarking

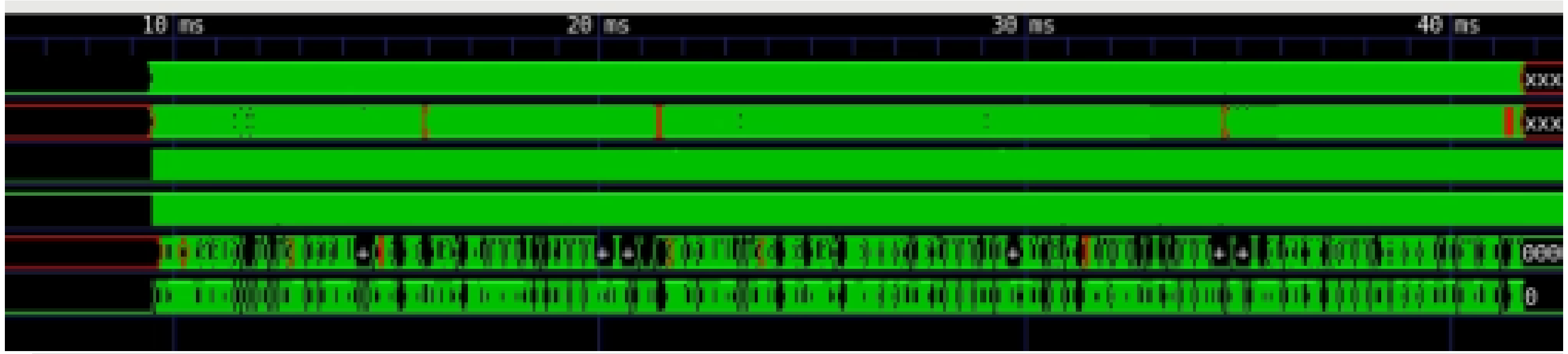
Target Benchmarks

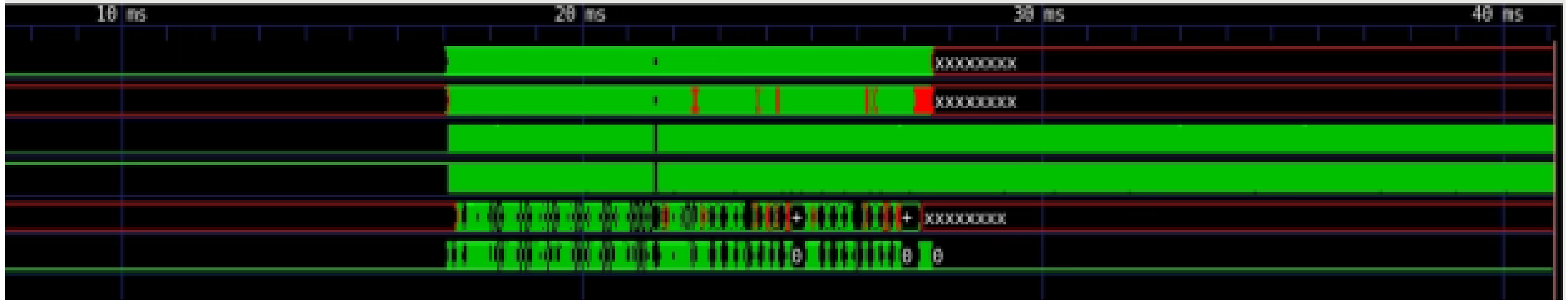
- embench-iot
  - Open-sourced
  - Small compiled size
  - Targets bare metal
  - Future support
- Based off common industry standard benchmarks
  - Measure real performance
- Target two specific operations
  - Fast Fourier Transforms
  - Multiply and Accumulate

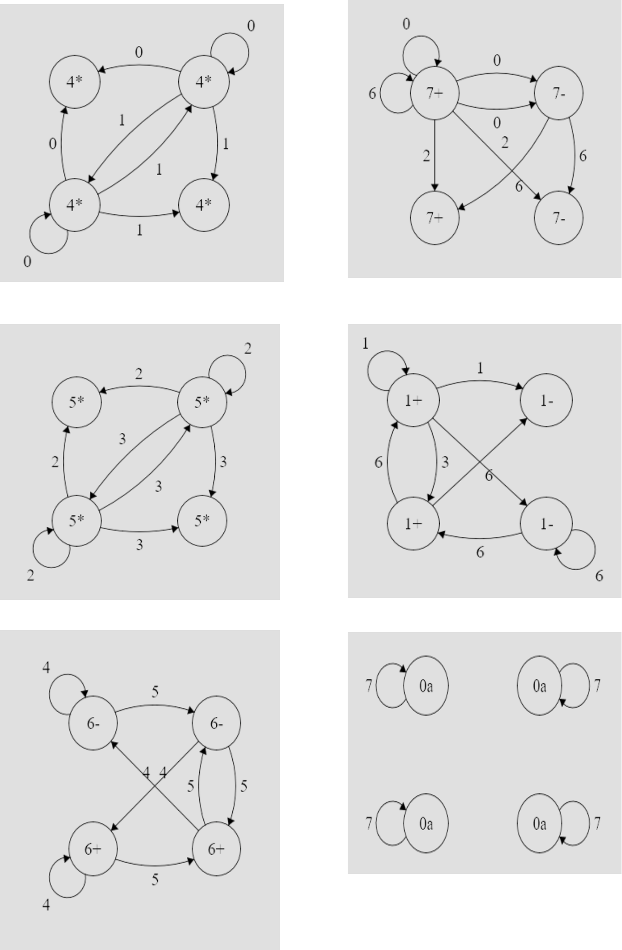
Benchmarks
aha-mont64
crc32
matmult-int
md5sum
nettle-sha256

FFT Benchmark

PicoRV32 (unaccelerated)	PicoRV32 with CyGRA	Speedup
31.63 ms	10.47ms	3.022x







Iowa State University
College of Engineering

Work Environment

Work Environment

Virtual Machine

- Personal VM from ETG for our team do complete our work on
- Ensures that any errors that arise are not due to setup differences
- Central place to do testing, allowing overnight testing in the event a test takes longer to run, scale up and down specs.

ChipForge toolset

- Provided to use by the ISU ChipForge club
- Remotely accessible computer for members to use to flash FPGAs with Verilog projects
- Additional documentation to help understand OpenLANE

Iowa State University
College of Engineering

Conclusions

Final Status

- All Verilog modules created and documented
- CyGRA
  - All custom instructions tested and working in simulation
- Fully integrated system
- Completed majority of milestones
  - Gate level testing incomplete
  - Not 100% ready for fabrication

Future Work

- Finish chip fabrication process
  - Fix timing slack violations (Static timing analysis)
- Improve CyGRA
  - 3x3 array
  - More space efficient
  - Less memory accesses
- Further testing
  - Gate level testing on hardened chip
  - Full FPGA testing

IOWA STATE UNIVERSITY
College of Engineering

THANK YOU

sdmay25-28

| Camden Fergen

| John Huaracha

| Nicholas Lynch

| Calvin Smith

| Levi Wenck

| 05/07/25