SDMAY25-28

CAMDEN FERGEN

JOHN HUARACHA

NICHOLAS LYNCH

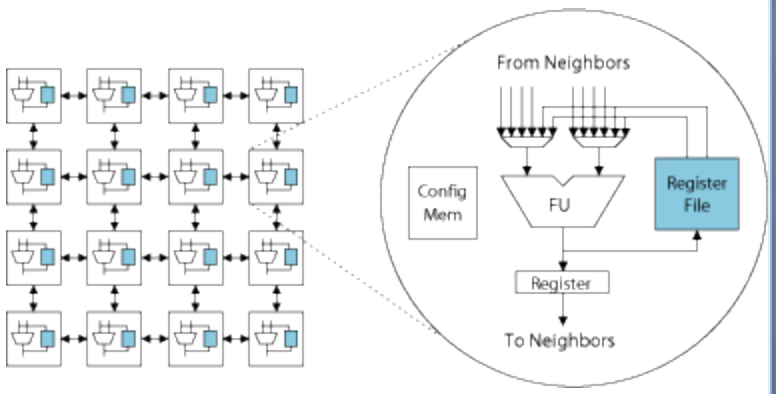
CALVIN SMITH

LEVI WENCK

DIGITAL ASIC DESIGN

FACULTY PANEL PRESENTATION

Client and
Faculty Advisor:
Professor Duwe

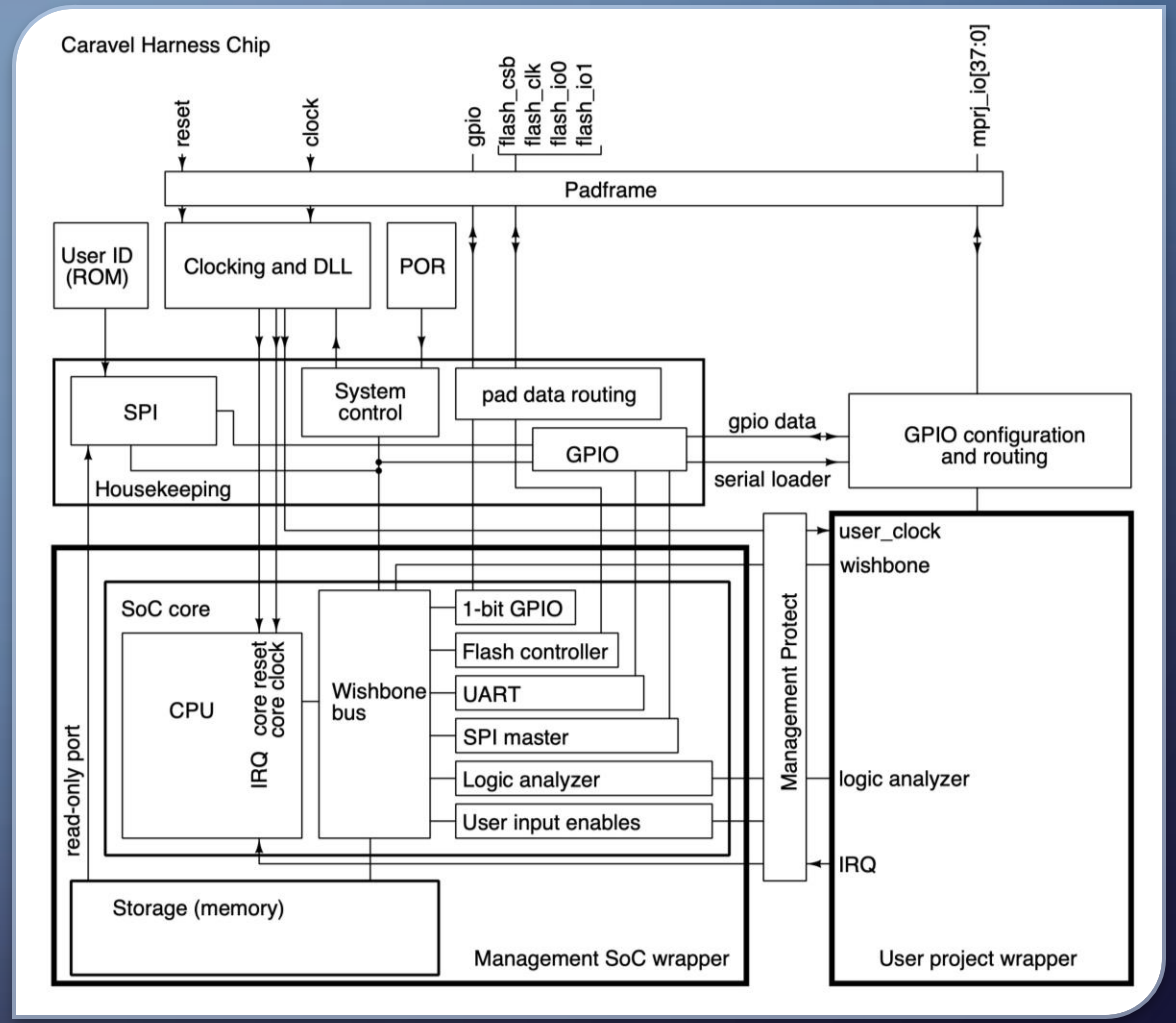


PROJECT OVERVIEW

- Problem
  - Non-Flexible ISAs
    - Re-fabricate if you want to test/use a new instruction
  - Lack of physical processors for student to learn from
    - Hardware student learn through software simulations or FPGAs
- Solution: Reprogrammable RISC-V Processor
  - Extending the RISC-V ISA to support custom instructions
  - Using a CGRA called "CyGRA" for programmable instruction
  - Support for memory and I/O

EFABLESS

- OpenLANE
  - Compilation of open-source tools
  - Verilog HDL to IC Layout
  - Provides test for generated ICs
- Caravel Harness
  - User Project Area
  - Management Area
    - Signals
      - Clock
      - Logic Analyzer
      - Wishbone
      - IRQ
    - VexRISC-V



From Efabless

USERS

CHIP FORGE CLUB MEMBERS

HARDWARE STUDENTS

PROFESSORS

FUNCTIONAL REQUIREMENTS

- Overall design contains a RISC-V processor
  - Can execute RISC-V machine code
  - Must be open source
- Custom instruction support
  - User should be able to program custom instruction into a custom instruction module
  - Users should be able to call and execute the custom instruction loaded
- Users can load in programs in memory and execute

NON-FUNCTIONAL REQUIREMENTS

- Project should be user-friendly
  - User-friendly interface for loading instructions and programming the CyGRA
  - Comprehensive and clear documentation
- Custom instruction should run within 1x-10x the CPI of a standard add instruction.
- Custom instruction should adhere to RISC-V ISA standards
- Efabless
  - Designs written in Verilog
  - Firmware written in C
  - Skywater 130nm process

EFABLESS DESIGN CONSTRAINTS

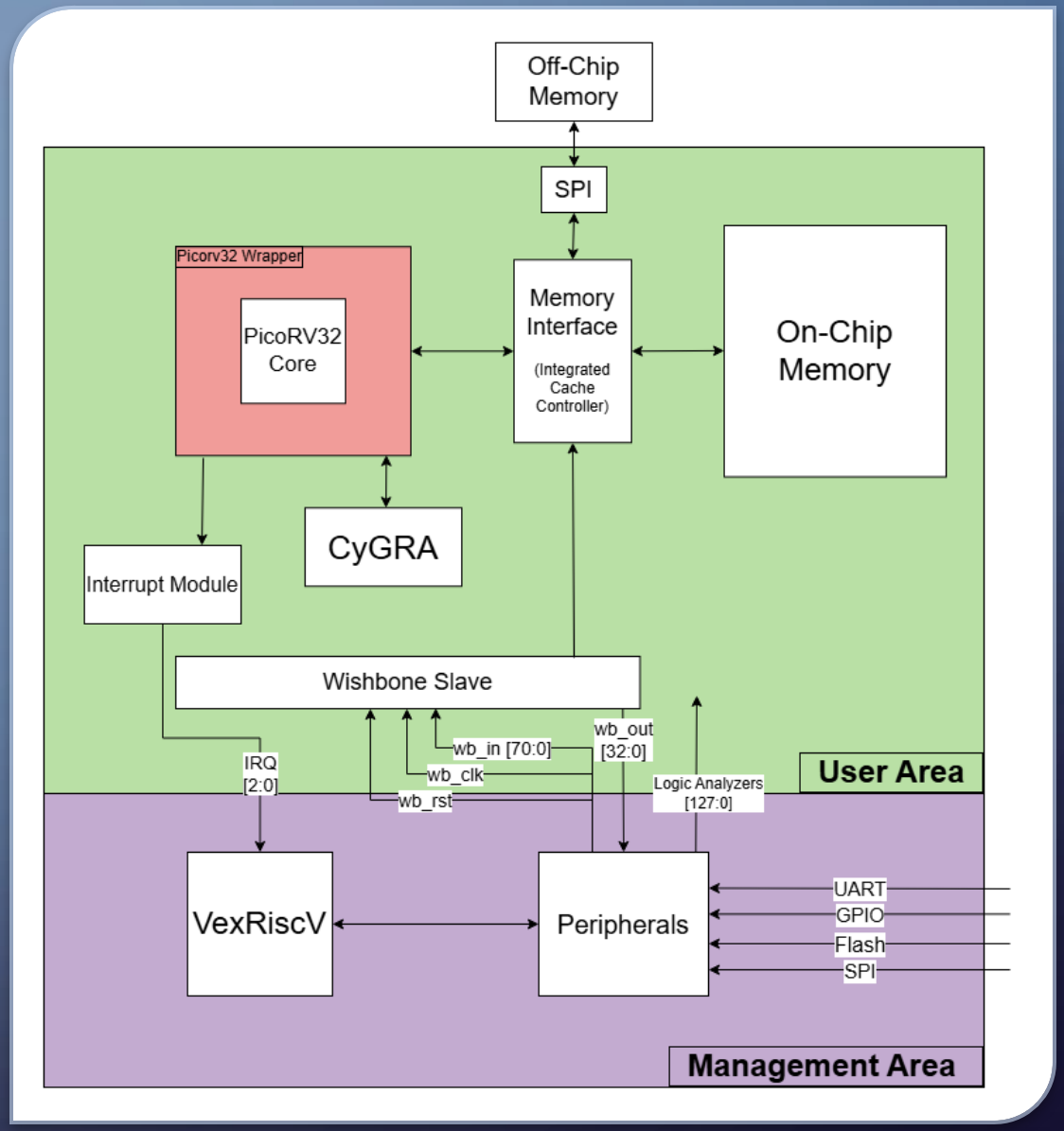
- 2920 µm x 3520 µm wrapper area (10 mm^2)
  - Dictates how much internal memory we can fit
- 40 MHz clock rate max
- Maximum of 34 I/O pins
  - Four used to communicate with SPI memory
  - Others used for I/O devices

ENGINEERING STANDARDS

- IEEE 1754-1994: IEEE Standard for a 32-bit Microprocessor Architecture
- IEEE 1364-2001: IEEE Standard Verilog Hardware Description Language
- IEEE 1364.1-2002: IEEE Standard for Verilog Register Transfer Level Synthesis
- Wishbone Bus
- SPI Protocol

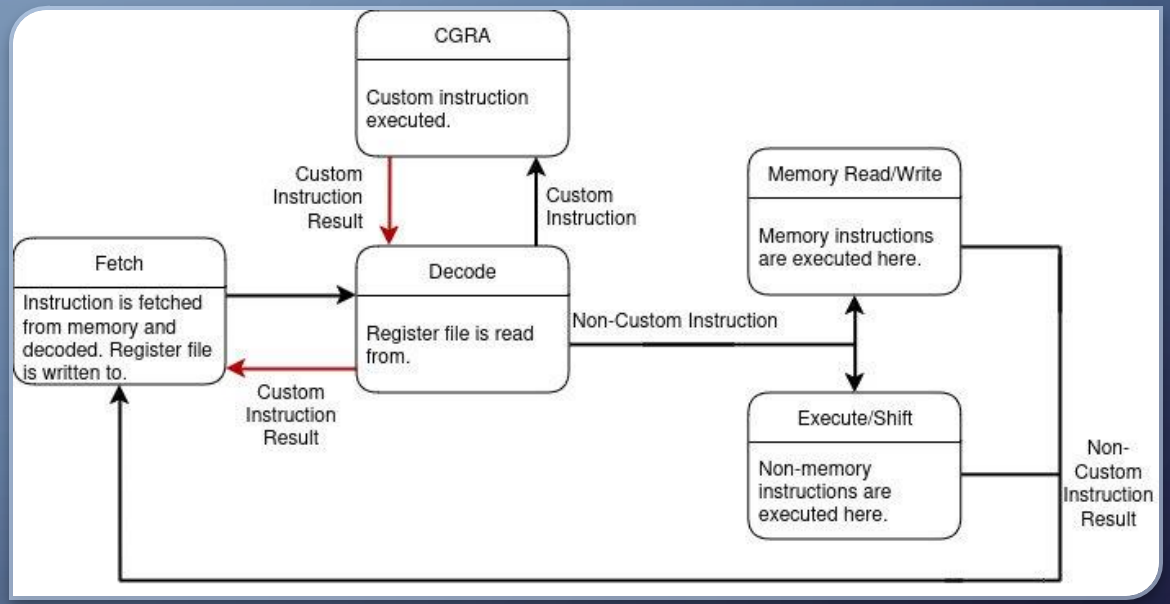
HIGH-LEVEL DESIGN

- PicoRV32 is a staged RISC-V processor.
- CyGRA is a CGRA that can be programmed and used by the PicoRV32 to execute custom instructions.
- Instruction memory written by management micro-controller through the wishbone bus.
- On-Chip Memory is used as cache for off-chip
- Off-Chip Memory is slower SPI memory.
- Logic Analyzers used to probe key parts of the project.
- IRQ module communicates with management core (VexRISC-V)



RISC-V CORE: PICORV32

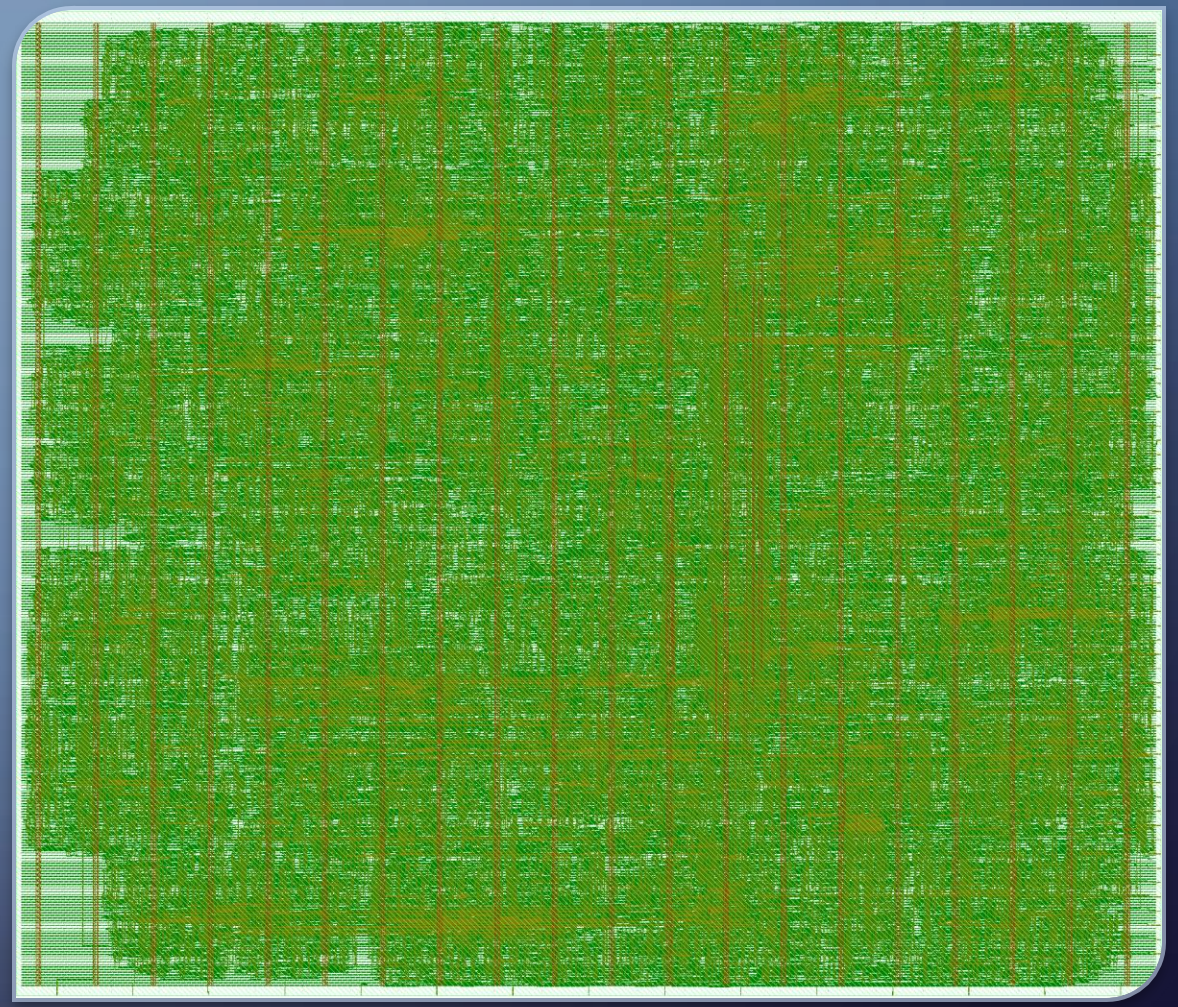
- Open-Source
- Low-Size 32bit RISC-V processor
- Supports custom non-branching instruction using a Co-Processor Interface.
  - Allows us to easily integrate our CyGRA
- Written in Verilog
  - Same as caravel project which allows for easy integration



Simplified stages of the PicoRV32 processor with the CyGRA. Custom instruction branch with normal instruction at the decode stage and the custom instruction result follow the red arrows back to the Fetch stage.

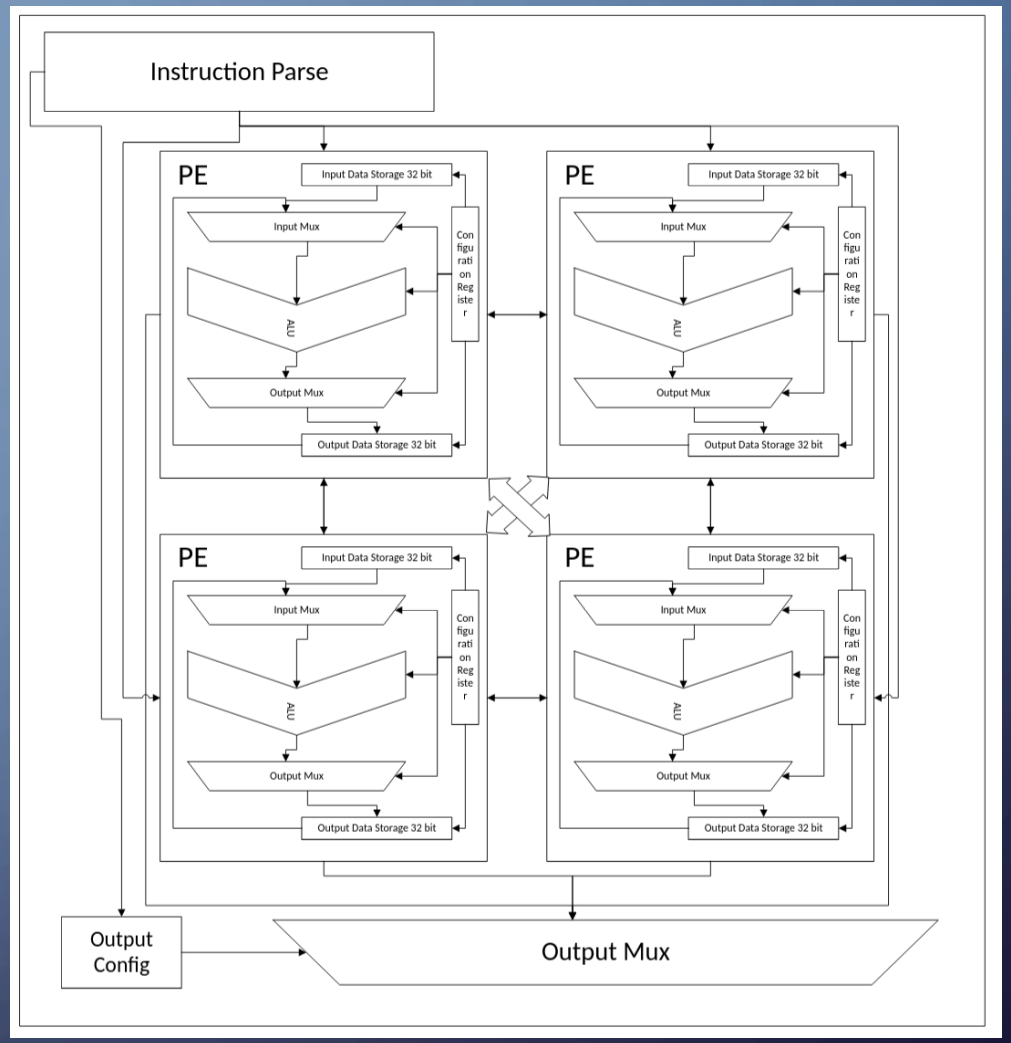
ON-CHIP MEMORY

- Currently using Open-Source pre-hardened 512x32 bit DFFRAM
- Area ≈1mm^2
- May experiment with other open-source memory solutions
  - OpenRAM
  - DFFRAM Compiler



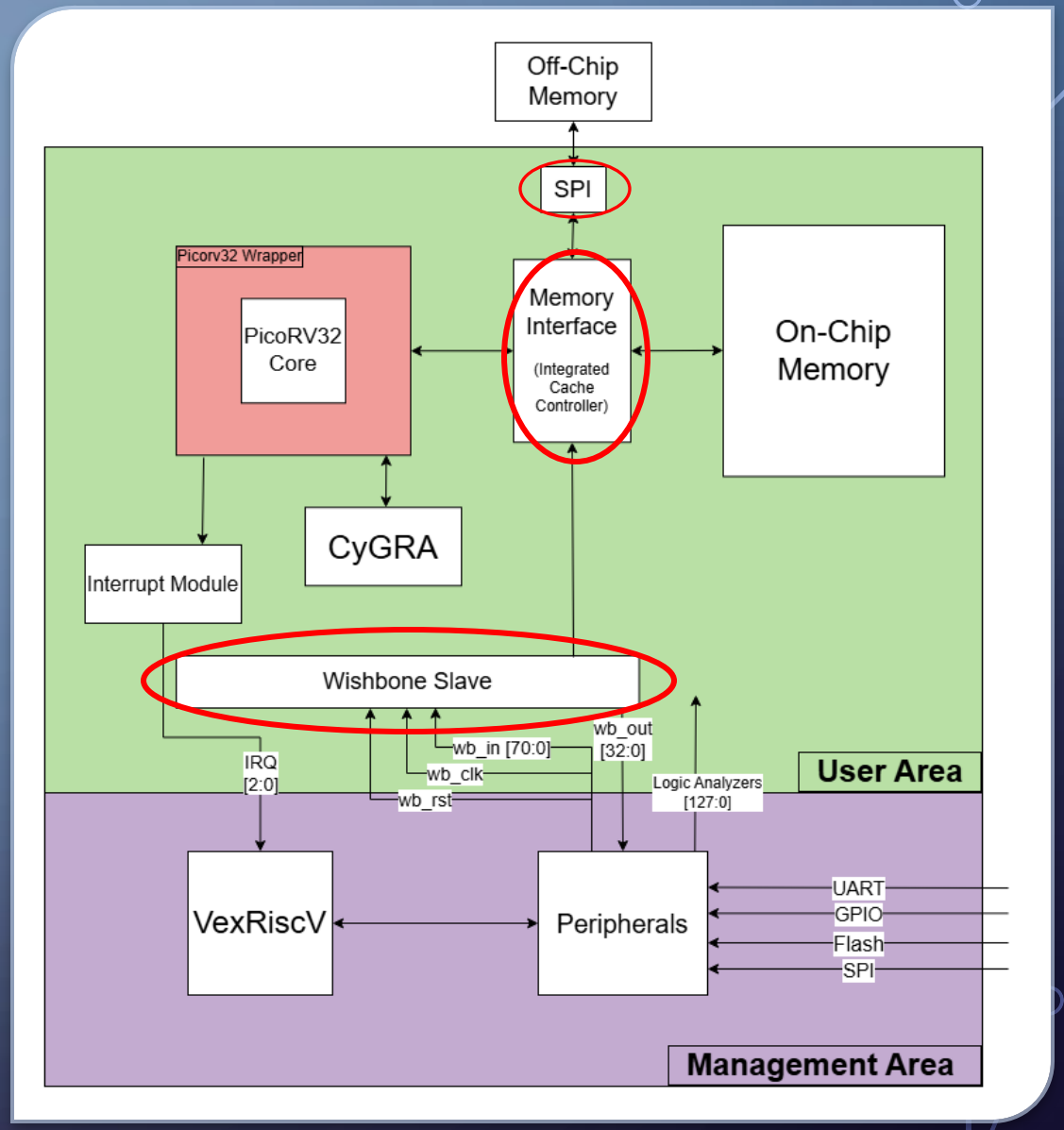
CURRENT CYGRA DESIGN

- 2x2 fully configurable CGRA implementation
- 32-bit fixed point and integer arithmetic and bitwise operations
- Tightly coupled with picorv32
- Math.h, convolution, matrix multiplication



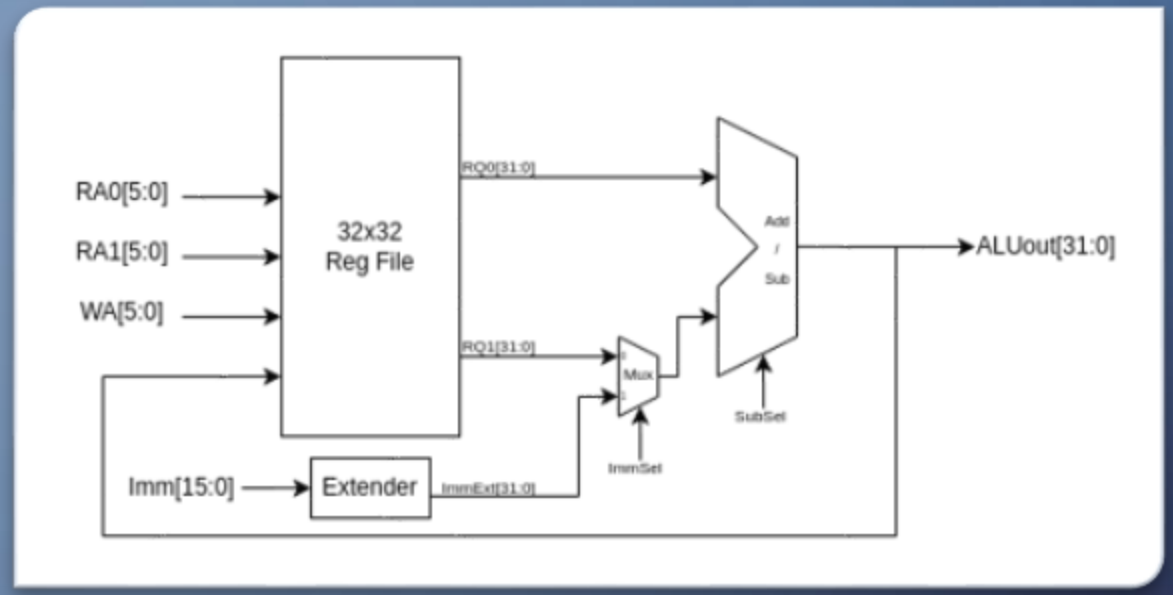
ADDITIONAL COMPONENTS

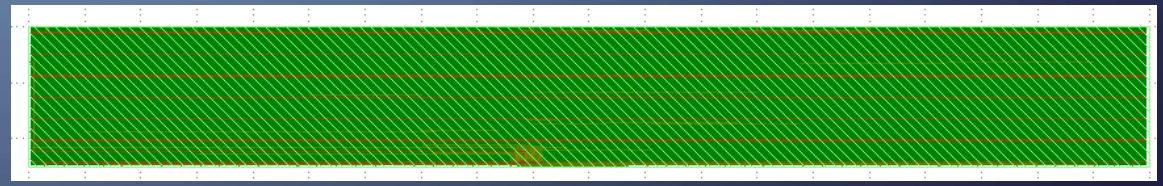
- Wishbone Slave: Used to interface with management area MCU to write instruction memory On-Chip.
- SPI Main Unit: Used to communicate with external SPI memory.
- Memory Interface Unit: Manges internal and external memory and provides memory inputs and outputs. Used to manage the cache (on-chip).



NOVEMBER TEST CHIP

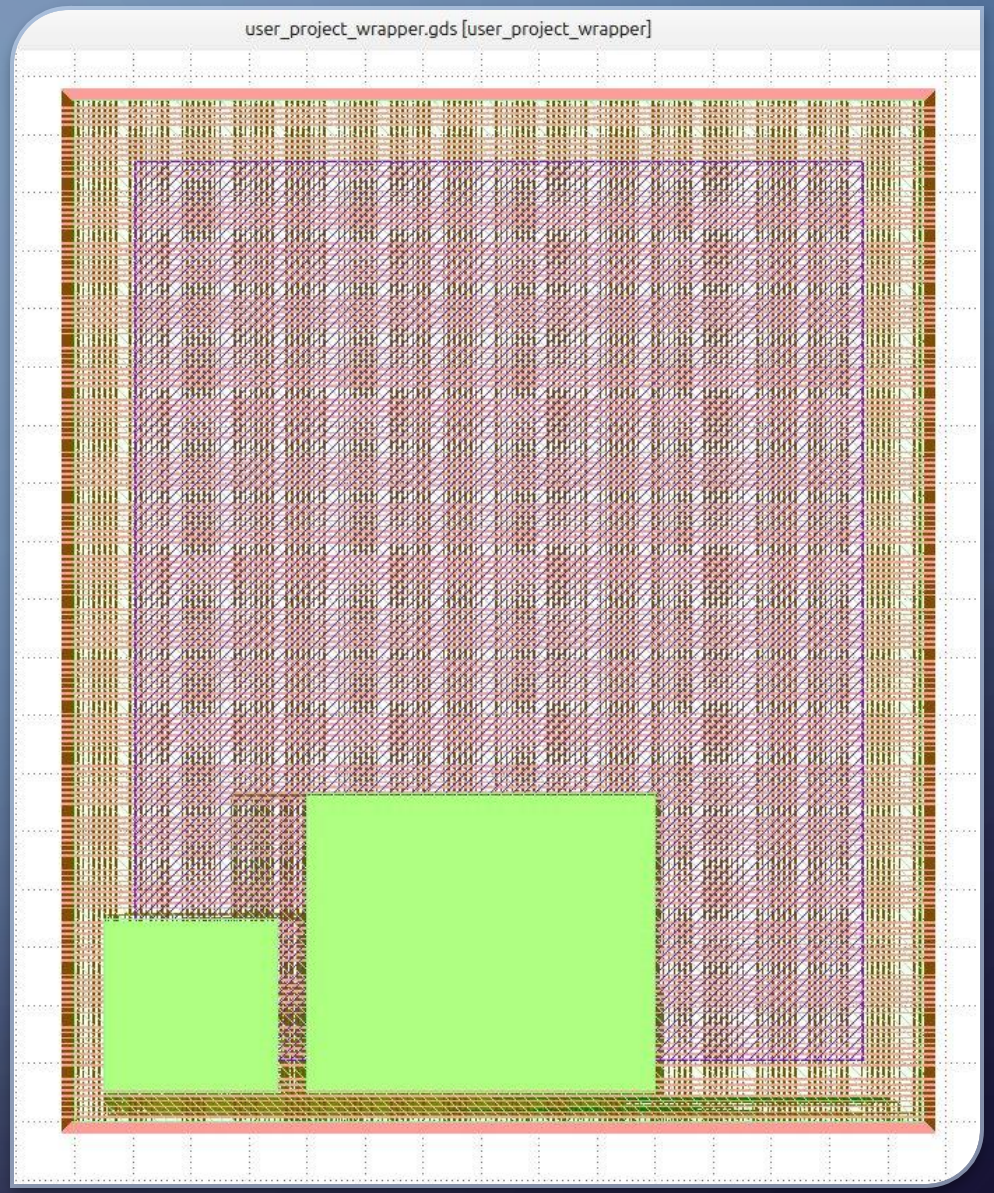
- Design made for team sddec24-12 (Last semester's Digital ASIC project)
  - Goal: Put several projects on one Caravel chip
- Opportunity to gain experience tools
- Module hardened
- Module in Caravel wrapper hardened
- Testing
  - RTL/GL test written
    - Passes RTL
  - DRC and LVS checks pass
  - Pre-Check (Check before sending chip to be fabricated) passes





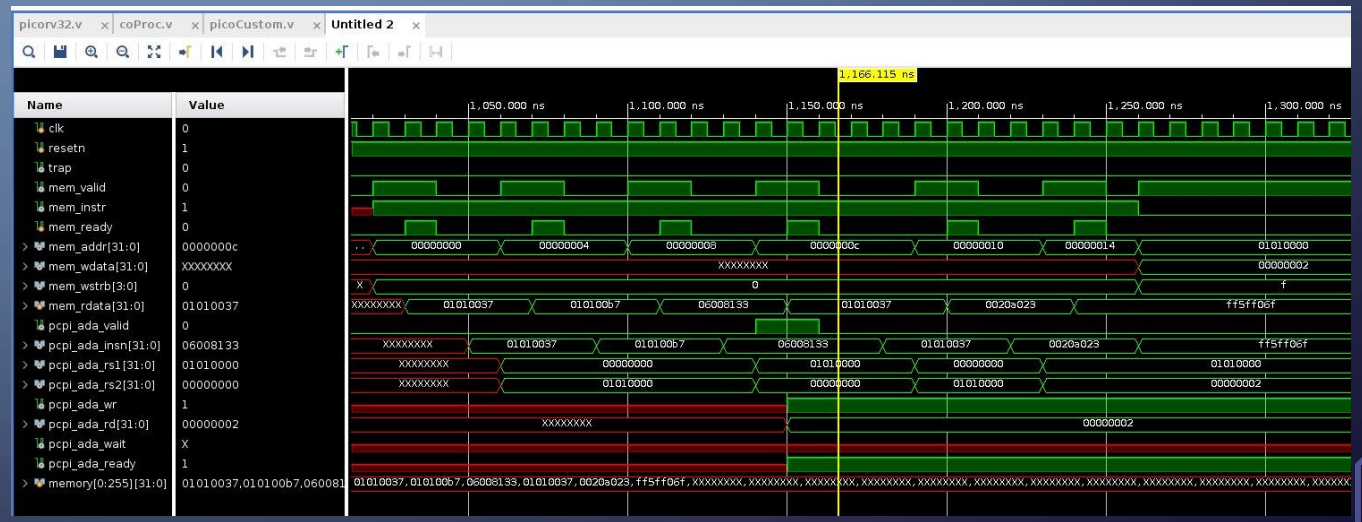
CURRENT PROGRESS: HARDENED WRAPPER WITH PICORV32 AND MEMORY

- Fully Synthesized
- 2KB of DFFRAM
- Passes Pre-check
- Can add other components onto wrapper once they are hardened and tested



CUSTOM INSTRUCTION RUNNING ON SIMULATED PICORV32

- Co-Processor interface proven
- Implemented R-type instruction which performs unsigned addition of all bytes in a word
- Behavioral simulation tested



BASIC C CODE RUNNING ON PICORV32

- RISCV toolchain scripts created for running C code on the picorv32
- Future accelerator code implementation will involve inline bytecode
- Needs incorporation with Efabless

scr > C main.c

```
1  void main(){
2      *(unsigned int*)(0x3FC) = 0xDEADBEEF;
3  }
4
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
vm-user@sdmay25-28-alpha:~/picoTools/picorv32$ ./scr/cr.sh
iverilog -o testbench_ez.vvp -DCOMPRESSED_ISA testbench_ez.v picorv32.v
chmod -x testbench_ez.vvp
vvp -N testbench_ez.vvp
output[255] = deadbeef
testbench_ez.v:20: $finish called at 11000000 (1ps)
vm-user@sdmay25-28-alpha:~/picoTools/picorv32$
```

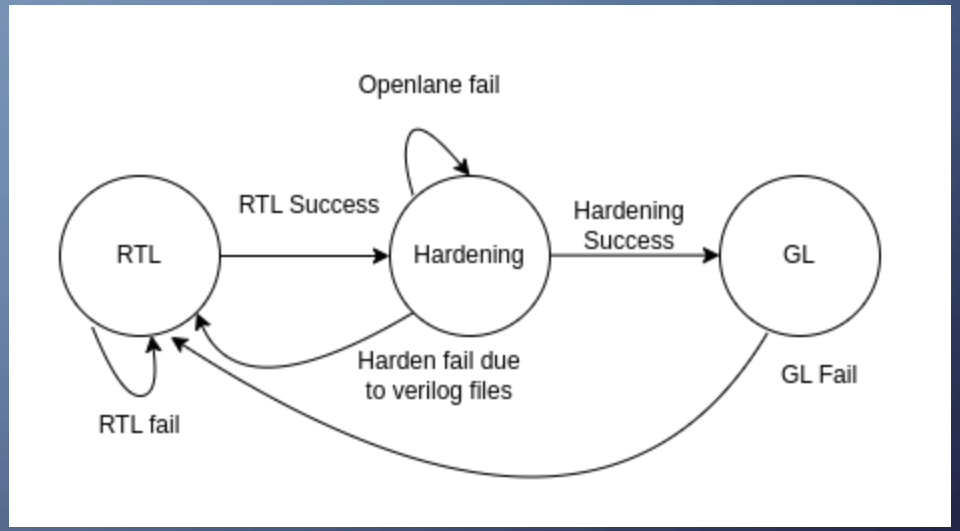
Compilation and running in iverilog via testbench bytecode injection

TEST PLAN

Verilog/Python Testbenches	C Code	FPGA Synthesis	Layout Verification
• Unit Tests • Integration Tests	• Integration Tests • System Tests	• System Tests	• DRC • LVS • STA

TEST FLOW FOR MODULES IN A CARAVEL PROJECT

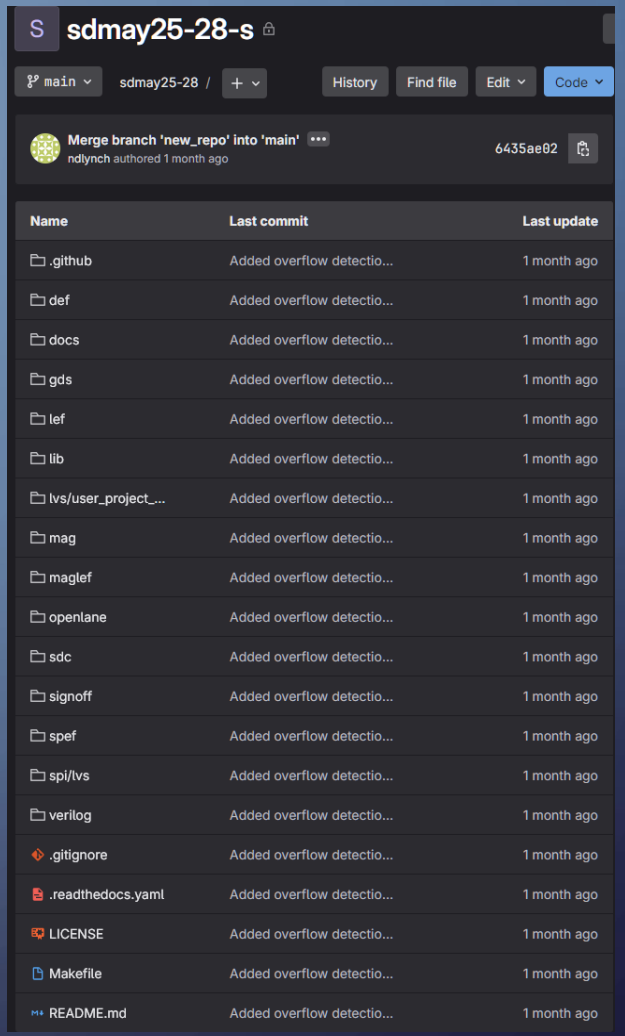
- Caravel RTL Simulation
  - Testbench Tool: CocoTB
  - Test firmware written in C
- OpenROAD Hardening: DRC, LVS, and STA
  - DRC Tool: Magic
  - LVS Tool: Netgen
  - STA Tool: OpenSTA
- Caravel GL Simulation
  - Structural Verilog representation of the layout
  - Reuses RTL testbench and firmware

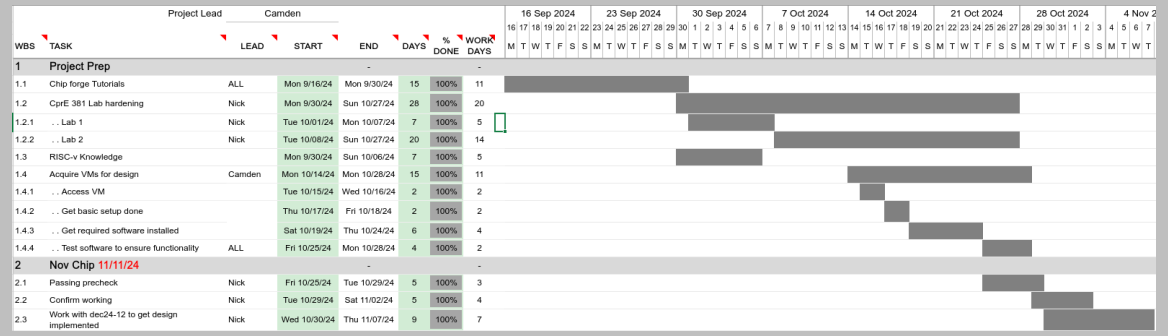


State diagram showing the test flow of a module.

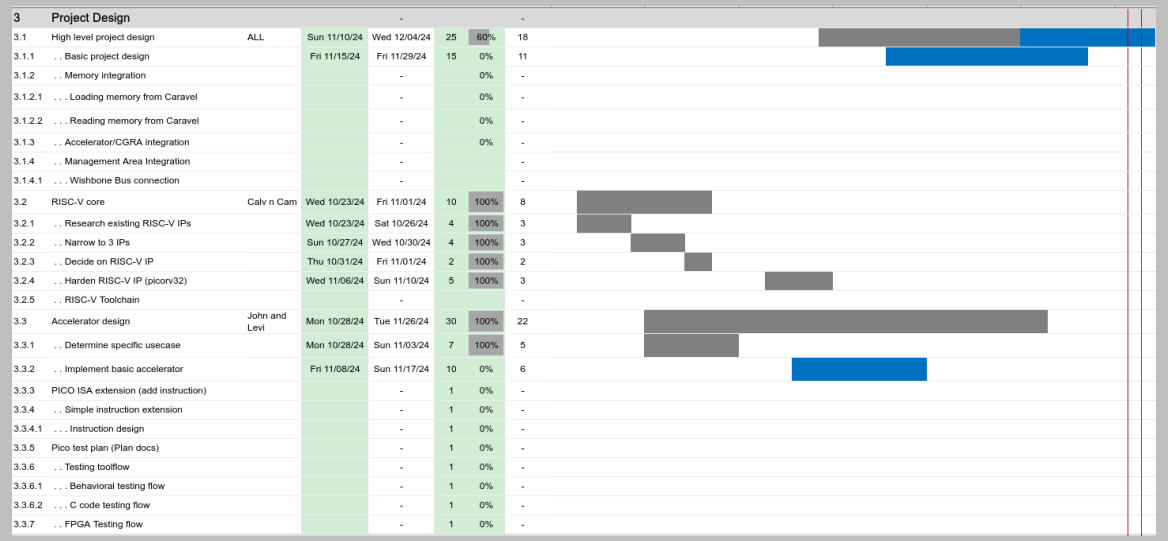
WORKSPACE SETUP

- Gitlab
  - Gitlab group to hold all repositories related to our project
  - Gitlab modules setup
  - Code issue tracking
- Virtual machines
  - Used to eliminate differences in setup
  - Helped streamline tooling setup





WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
	Project Lead	Camden					
1	Project Prep			-			-
1.1	Chip forge Tutorials	ALL	Mon 9/16/24	Mon 9/30/24	15	100%	11
1.2	CprE 381 Lab hardening	Nick	Mon 9/30/24	Sun 10/27/24	28	100%	20
1.2.1	. . Lab 1	Nick	Tue 10/01/24	Mon 10/07/24	7	100%	5
1.2.2	. . Lab 2	Nick	Tue 10/08/24	Sun 10/27/24	20	100%	14
1.3	RISC-v Knowledge		Mon 9/30/24	Sun 10/06/24	7	100%	5
1.4	Acquire VMs for design	Camden	Mon 10/14/24	Mon 10/28/24	15	100%	11
1.4.1	. . Access VM		Tue 10/15/24	Wed 10/16/24	2	100%	2
1.4.2	. . Get basic setup done		Thu 10/17/24	Fri 10/18/24	2	100%	2
1.4.3	. . Get required software installed		Sat 10/19/24	Thu 10/24/24	6	100%	4
1.4.4	. . Test software to ensure functionality	ALL	Fri 10/25/24	Mon 10/28/24	4	100%	2
2	Nov Chip 11/11/24			-			-
2.1	Passing precheck	Nick	Fri 10/25/24	Tue 10/29/24	5	100%	3
2.2	Confirm working	Nick	Tue 10/29/24	Sat 11/02/24	5	100%	4
2.3	Work with dec24-12 to get design implemented	Nick	Wed 10/30/24	Thu 11/07/24	9	100%	7



3	Project Design			-			-
3.1	High level project design	ALL	Sun 11/10/24	Wed 12/04/24	25	60%	18
3.1.1	. . Basic project design		Fri 11/15/24	Fri 11/29/24	15	0%	11
3.1.2	. . Memory integration			-		0%	-
3.1.2.1	. . . Loading memory from Caravel			-		0%	-
3.1.2.2	. . . Reading memory from Caravel			-		0%	-
3.1.3	. . Accelerator/CGRA integration			-		0%	-
3.1.4	. . Management Area Integration			-			-
3.1.4.1	. . . Wishbone Bus connection			-			-
3.2	RISC-V core	Calv n Cam	Wed 10/23/24	Fri 11/01/24	10	100%	8
3.2.1	. . Research existing RISC-V IPs		Wed 10/23/24	Sat 10/26/24	4	100%	3
3.2.2	. . Narrow to 3 IPs		Sun 10/27/24	Wed 10/30/24	4	100%	3
3.2.3	. . Decide on RISC-V IP		Thu 10/31/24	Fri 11/01/24	2	100%	2
3.2.4	. . Harden RISC-V IP (picorv32)		Wed 11/06/24	Sun 11/10/24	5	100%	3
3.2.5	. . RISC-V Toolchain			-			-
3.3	Accelerator design	John and Levi	Mon 10/28/24	Tue 11/26/24	30	100%	22
3.3.1	. . Determine specific usecase		Mon 10/28/24	Sun 11/03/24	7	100%	5
3.3.2	. . Implement basic accelerator		Fri 11/08/24	Sun 11/17/24	10	0%	6
3.3.3	PICO ISA extension (add instruction)			-	1	0%	-
3.3.4	. . Simple instruction extension			-	1	0%	-
3.3.4.1	. . . Instruction design			-	1	0%	-
3.3.5	Pico test plan (Plan docs)			-	1	0%	-
3.3.6	. . Testing toolflow			-	1	0%	-
3.3.6.1	. . . Behavioral testing flow			-	1	0%	-
3.3.6.2	. . . C code testing flow			-	1	0%	-
3.3.7	. . FPGA Testing flow			-	1	0%	-

PROJECT SCHEDULE

High Level Overview:

- Project Prep – **Complete**
- November Chip Deadline – **Complete**
- Project Design – **In-Progress**
- Integration – **Future Step**
- Testing – **Future Step**

PROJECT STATUS

Complete	In-Progress	Future Steps
Virtual machine and Tooling	CyGRA Implementation	Communication with Management Area
November Chip	PICO-RV ISA Extension	High-Level Behavioral Tests
RISC-V Processor	Module Design	FPGA Testing
Accelerator Application		Precheck and Hardening

TEAM ROLES

- Camden Fergen – DevOps and Project Lead
- John Huaracha – Testing Lead
- Nicholas Lynch – Harden and Verification Lead
- Calvin Smith – Accelerator Design Lead
- Levi Wenck – Communication Interfaces Lead

THANK YOU!

QUESTIONS?

REFERENCES

- Benson, Duane. “Efabless Announces Custom SoC Design Platform for Edge Machine Learning.” *All About Circuits*, 16 Oct. 2024. Accessed November 20, 2024
- CGRA Architecture and Tools | AHA Agile Hardware Project. https://aha.stanford.edu/research/cgra-architecture-and-tools. Accessed October 11, 2024.
- Efabless Caravel “Harness” SoC — Caravel Harness Documentation. https://caravel-harness.readthedocs.io/en/latest/. Accessed September 20, 2024.
- “PicoRV32 - A Size-Optimized RISC-V CPU.” GitHub, https://github.com/YosysHQ/picorv32. Accessed October 30, 2024.
- Todd, Dillon. “Tightly Coupling the PicoRV32 RISC-V Processor with Custom Logic Accelerators via a Generic Interface.” All Theses, May 2021, https://open.clemson.edu/all_theses/3552. Accessed November 15, 2024.