

CPE 470 - CARP Core & History



CARP

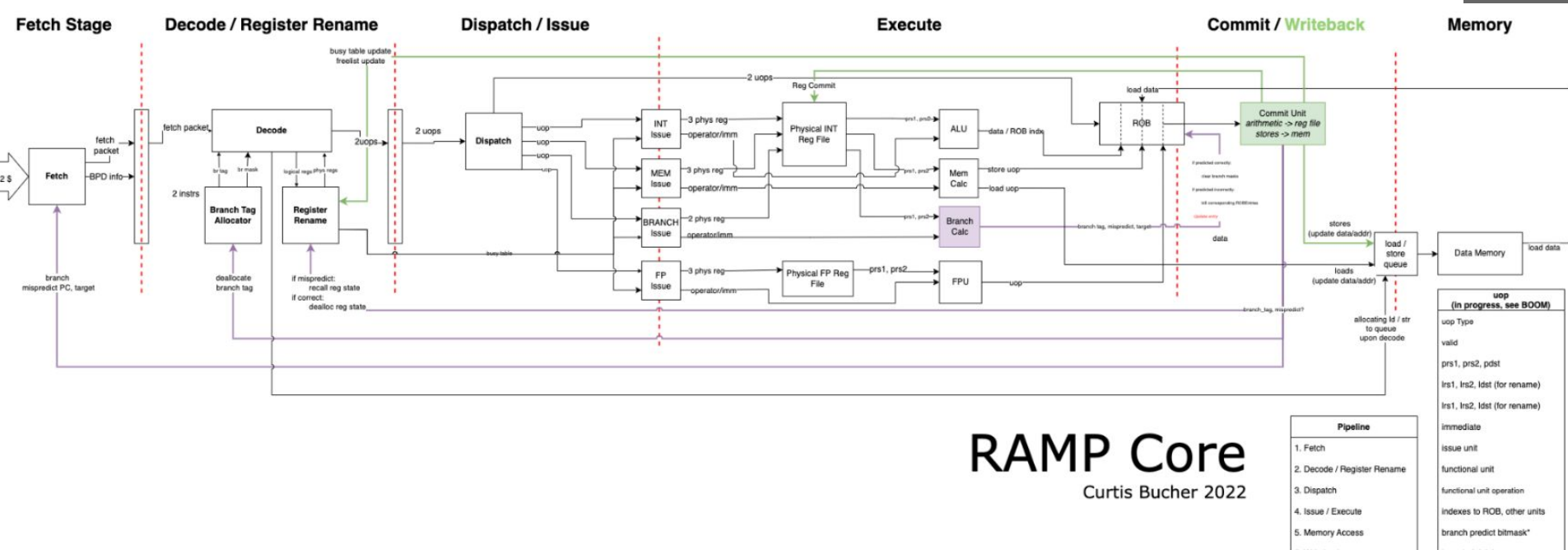
CARP: Computer Architecture Research Project

SURP: Summer Undergraduate Research Program

- **CARP:** a research Group at Cal Poly
 - Active from Late 2022 through December 2024
 - Directed by the late prof. Joseph Callenes-Sloan
- Focused on RISC-V development
 - Oriented at computer architecture more than VLSI
 - Used fully open source tools, primarily `verilator`
- Originally began with a **SURP** project
 - Implementation of an Out-Of-Order processor in PyMTL
 - Open Source Hardware Modeling in Python
 - High Level Synthesis (HLS)

Original OOO Core

- Partially working python model
- Never fully completed

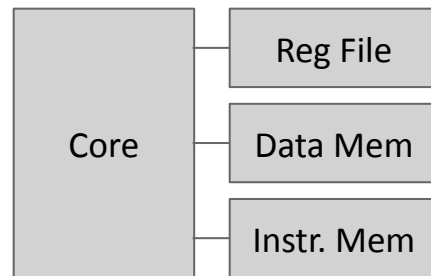


In Order Processor: Goals

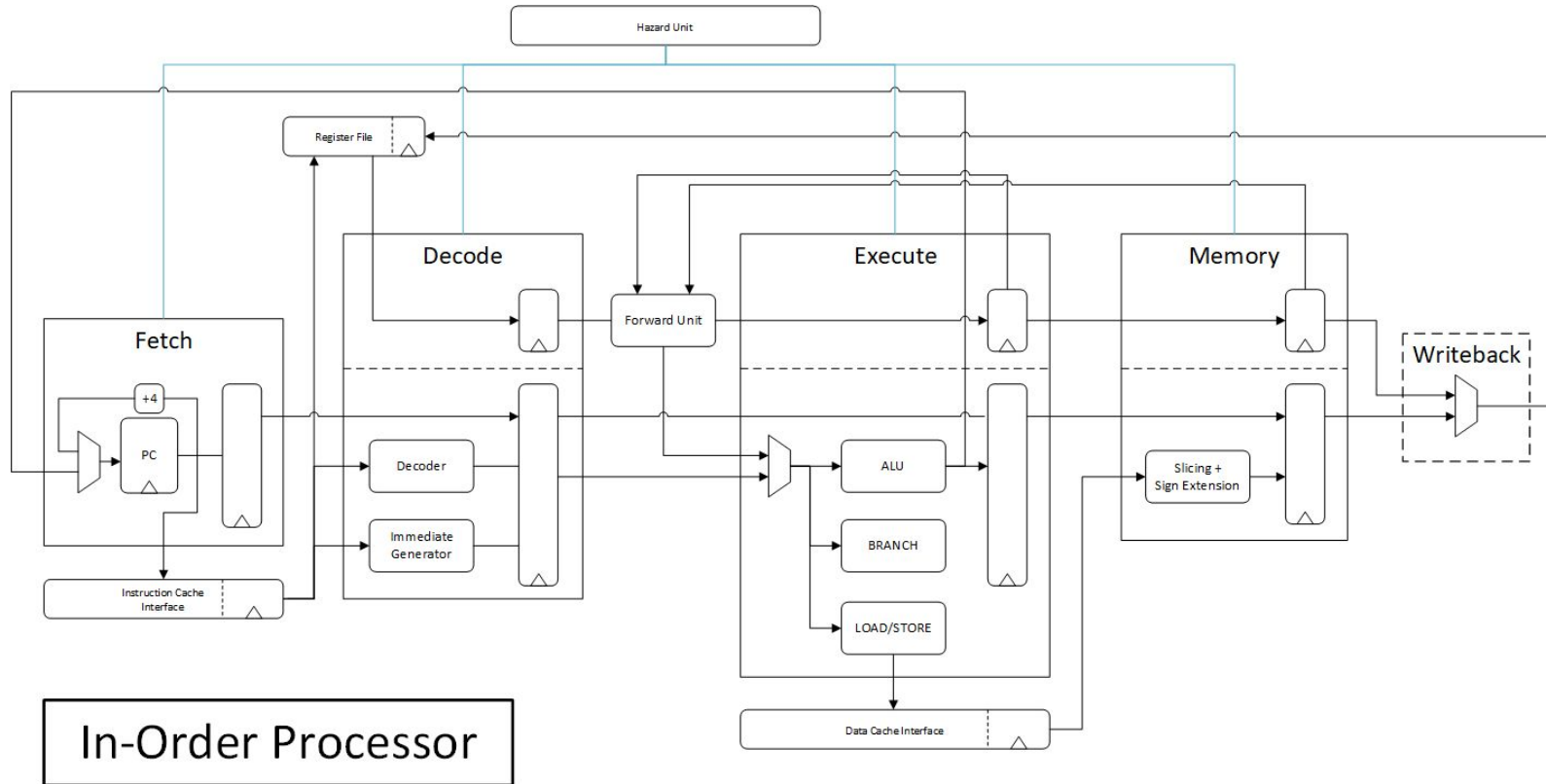
Glossary

RTOS: Lightweight OS capable of switching tasks with tight time constraints

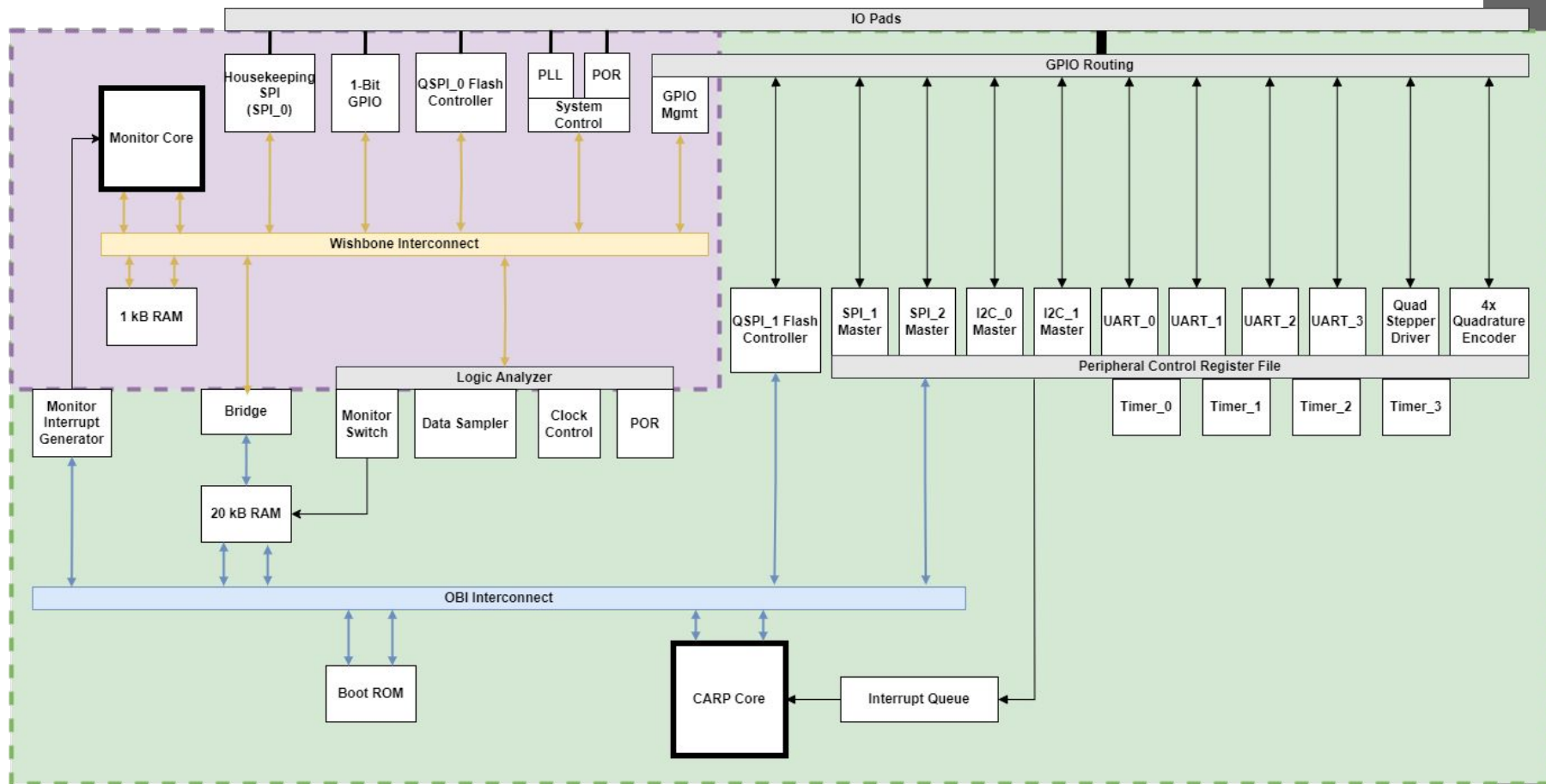
- Goal: make an embedded processor capable of running an **RTOS**
 - RISC V CPU with interrupts & IO peripherals
- Effectively CPE 333 Pipelined Otter
 - Redesigned from scratch for readability & modularity
- Each pipeline stage defined in its own module
 - Explicit struct used to pass data between each stage
 - Fetch to Decode, Decode to Execute each have a unique struct defining all signals between each stage
- Register File, Instruction Memory, and Data Memory kept separate from core
 - Abstract memories from design → easier synthesis



In Order Processor Implementation



Clam SOC



Clam SOC Ecosystem

- Software ecosystem built around SoC
 - **HAL** built around peripherals to ease configuration
 - SPI, Timers, GPIO, etc.
- Vectored Interrupt controller built into HAL
 - Inspired by STM32 and Arduino libraries
 - Allows attaching specific functions to interrupts
- FreeRTOS builds were still being debugged

```
attach_isr(&spi_intr, ISR_SPI0_TX);
enable_intr(ISR_SPI0_TX);
```

```
//Before
*(SPI0_CONTROL) = 1 | (0x7 << 4)
| (1<<10) | (1<<11)
| (1<<12) | (1<<16);

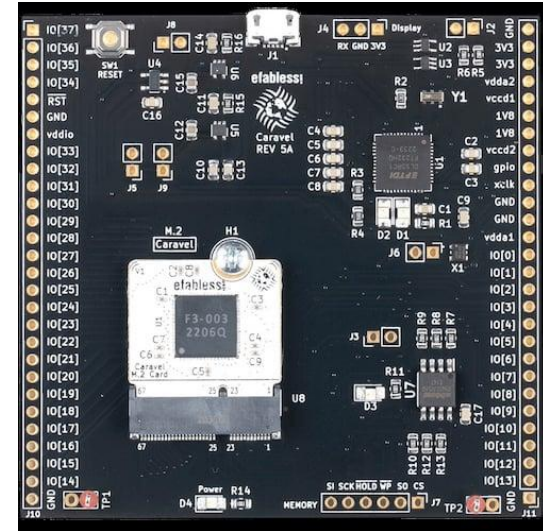
//After
spi_config_t spi_con;
spi_con.c_phase = 0;
spi_con.c_polarity = 0;
spi_con.num_packets = 1;
spi_con.rx_msb_first = 1;
spi_con.tx_msb_first = 1;
spi_con.transaction_length = 8;
spi_con.clock_div = 8;
spi_config(SPI0, &spi_con);

//Before
*(GPIO_0_1) = 0x140 | 0x140 << 16;
*(GPIO_2_3) = 0x140 | 0x040 << 16;

//After
gpio_config(0, GPIO_OUTPUT, 1, 0, 0);
gpio_config(1, GPIO_OUTPUT, 1, 0, 0);
```

Clam SOC Tapeout

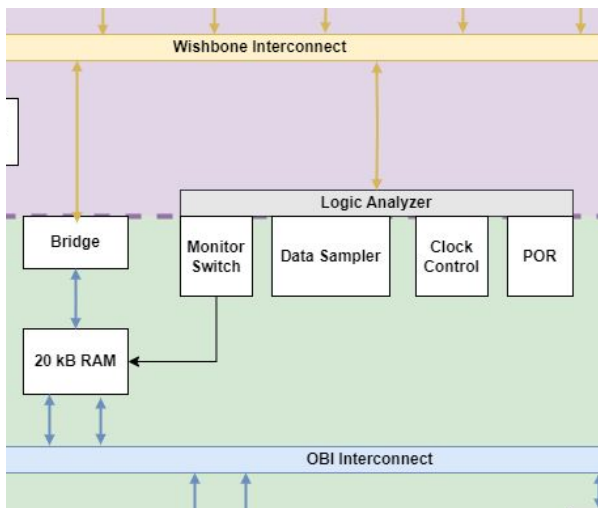
- SoC taped out in November 2023
 - Used Efabless' ChipIgnite Shuttle
 - \$10,000 for 100 chips
 - 1 Test PCB
- Very rushed final stages
 - Effectively verifying things to submission deadline
- Chips Delivered in June 2024



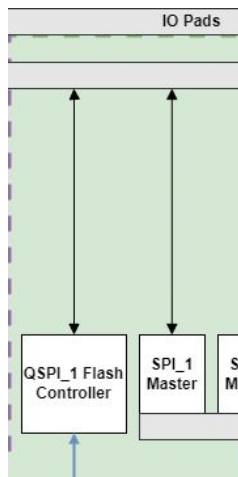
Clam SOC Bringup

- Clam SOC has 2 ways to get flashed with a program
 - Wishbone to OBI bus bridge
 - Different clock domains → Crosses Clock Domains
 - QSPI Flash Controller
 - Bidirectional: has to dynamically configure pins as inputs or outputs
 - Polarity on Input / Output selection seems to be inverted

Wishbone to AXI Bridge



QSPI Controller



- Efabless Management Core works as expected

Learnings

- Outside In Verification
 - Doesn't Matter if Internals work if they cannot reach outside
 - Doesn't matter if RISC-V core works if it cannot be flashed
- Can't assume design that works in sim will work on hardware
 - Gate level testing cannot be skipped
- Add multiple ways to fix possible errors
 - Multiple ways to flash memory
 - Failsafe for any interface that could fail

Clam SOC Ongoing Problems

- Simulation
 - Test with Icarus Verilog
 - Designed for verilator
 - Icarus has improved since the core was made
 - Gate Level Testing
- Synthesis
 - Compatible with Yosys 0.23
 - Not Compatible with Yosys 0.51
 - Possible yosys bug?
- OpenLane
 - Built on Openlane 1
 - Could be ported to Openlane 2
 - SRAM Macros not supported by Openlane 2 (yet?)
- It needs your help!

CARP Future Work

- Besides Fixing all of these issues... New Work
- Compressed Dual-Issue Pipeline
 - Capable of executing compressed, 16-bit instructions
 - Capable of executing more than one instruction in specific cases

