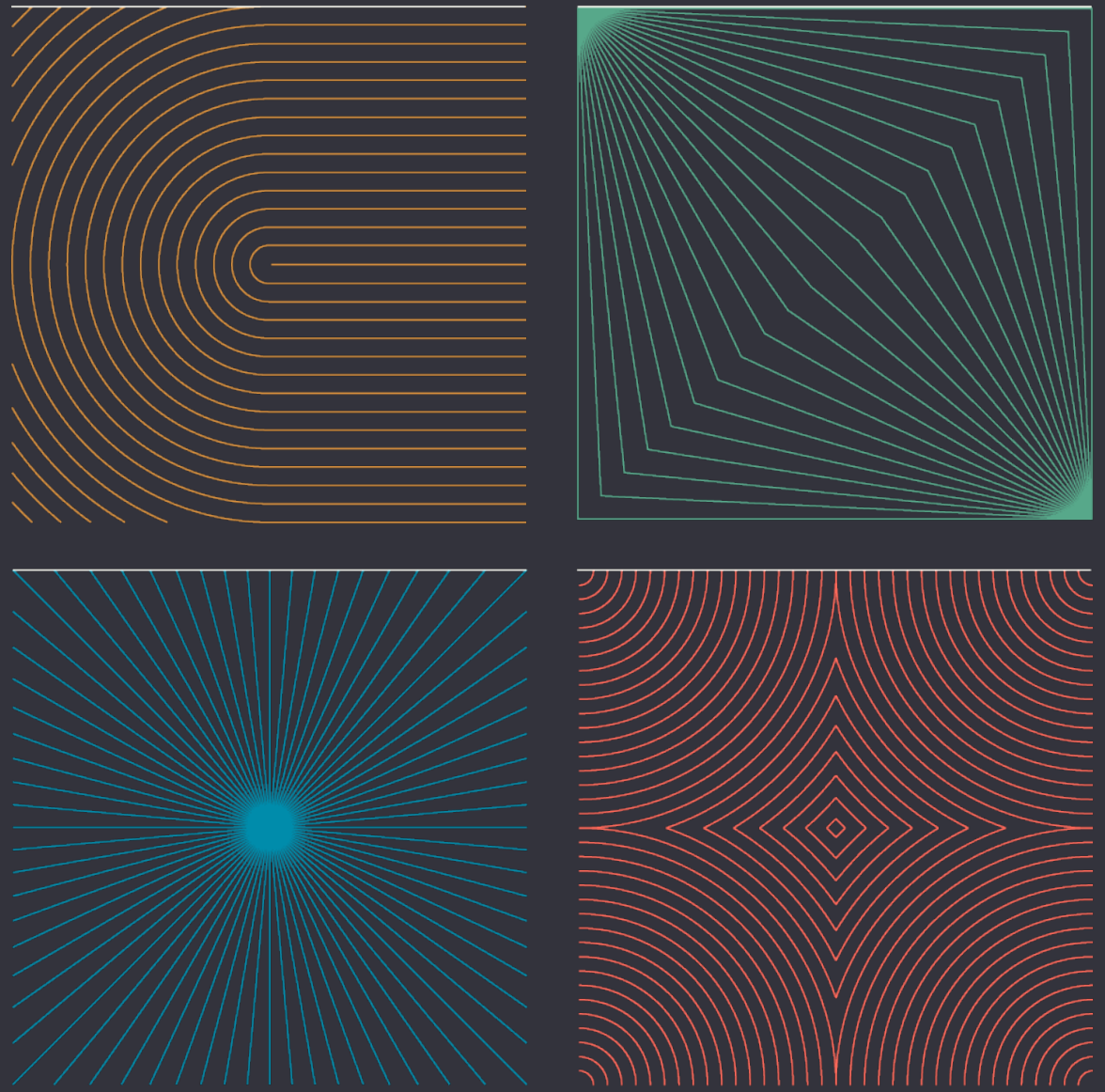
Ocelot Vector Unit and Integrating SV-based Modules in BOOM

Dongjie (DJ) Xie

Srikanth Arekapudi

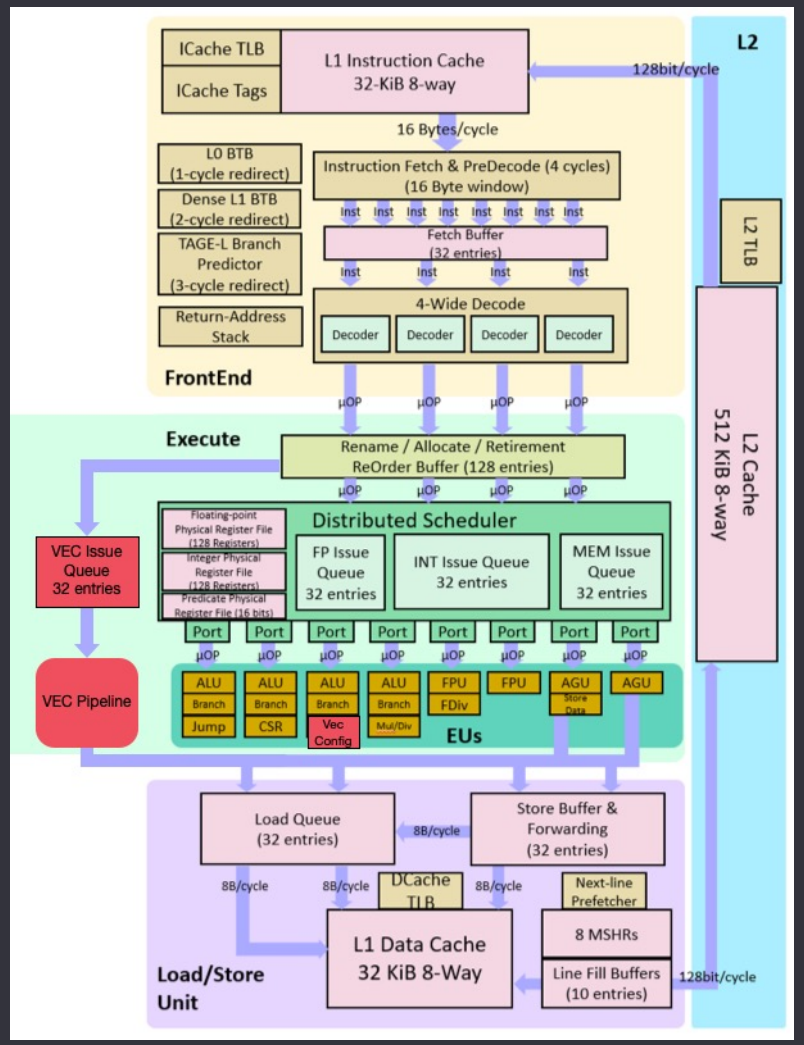
tenstorrent



Outline

- Overview
  - BOOM Core
  - Chisel
  - Vector Unit
- Integration Process
  - Chisel-SystemVerilog Integration
  - In-Order Pipeline in Out-of-Order Core
  - LSU Retrofitting
- Debugging Harness

BOOM Core with Ocelot Vector Unit



tenstorrent

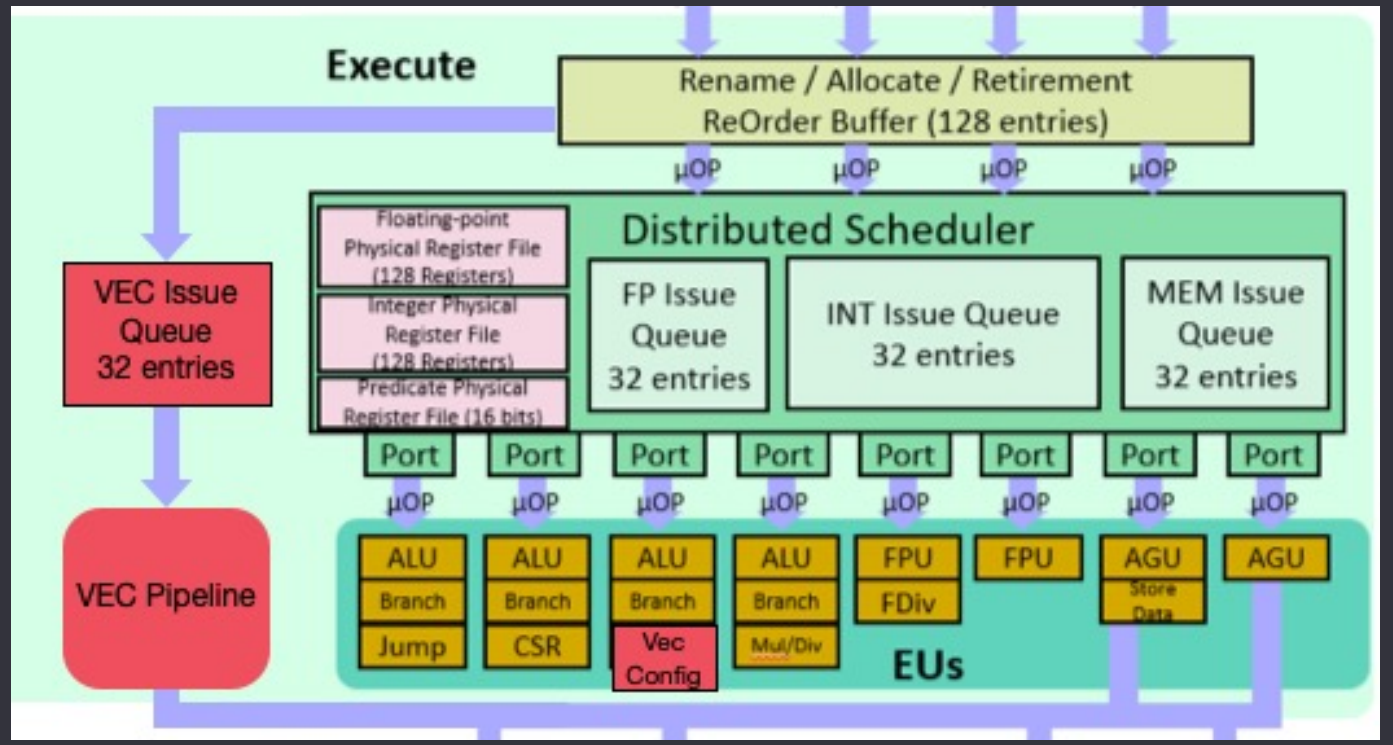
https://carrv.github.io/2020/papers/CARRV2020_paper_15_Zhao.pdf

Chisel-Based Design

Object-oriented programming

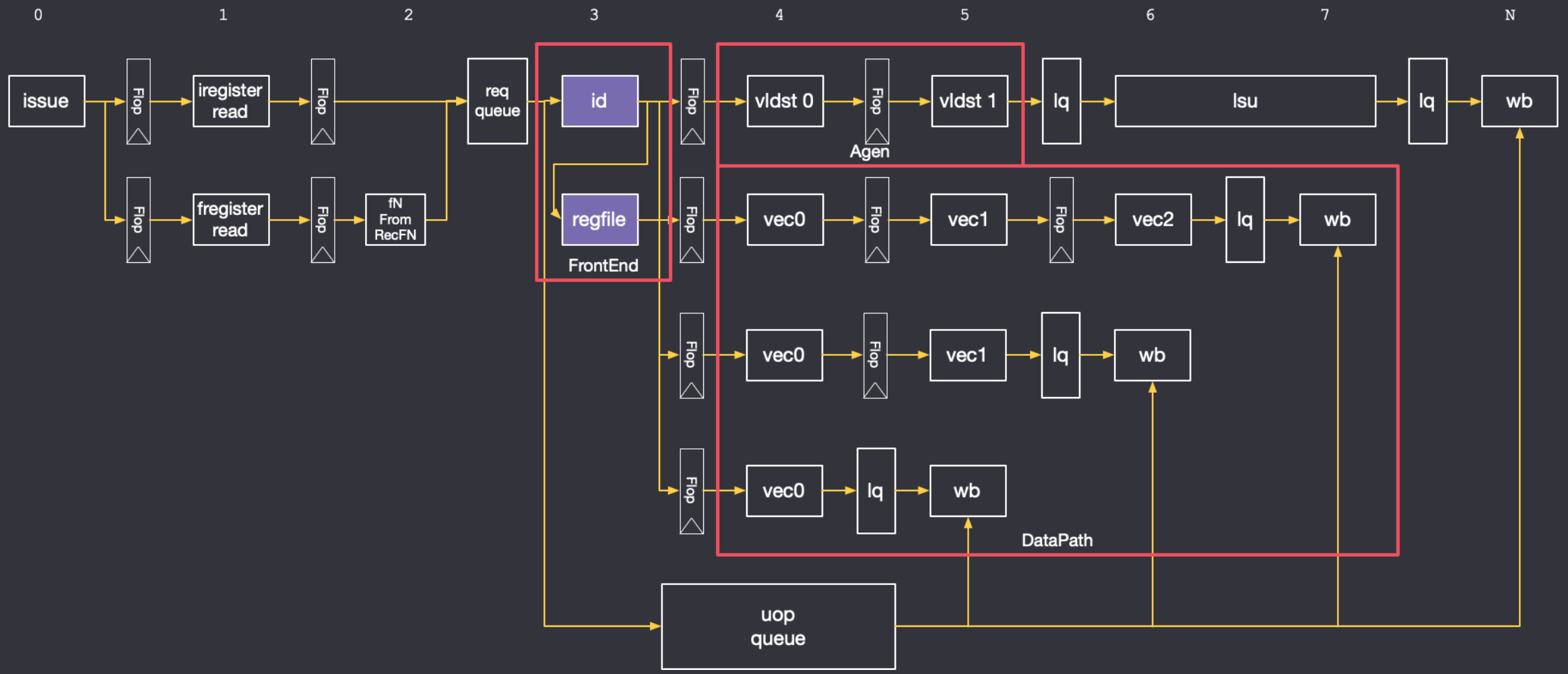
Well-defined class hierarchy

Highly configurable



https://carrv.github.io/2020/papers/CARRV2020_paper_15_Zhao.pdf

In-order Vector Unit



tenstorrent

Integration Challenges

1. Integrating a SystemVerilog module into a Chisel-based design
2. Incorporating an in-order pipeline into an out-of-order core
3. Retrofitting the Load-Store Unit (LSU)

Chisel-SystemVerilog Integration

Chisel Bundle

```
class ToyCommitSignals(val coreMaxAddrBits: BigInt, val retireWidth: BigInt) extends
Bundle
{
  val arch_valids = Vec(retireWidth, Bool())
  val uops        = Vec(retireWidth, new ToyMicroOp(coreMaxAddrBits))
}

class ToyMicroOp(val coreMaxAddrBits: BigInt)
extends Bundle
{
  val debug_pc = UInt(coreMaxAddrBits.W)
}
```

SystemVerilog Interface

```
interface BoomCoreHarnessIntf
#(parameter
   coreMaxAddrBits = 40,
   retireWidth     = 3  );

   typedef struct packed {
       logic [coreMaxAddrBits-1:0] debug_pc;
   } MicroOp_t;

   typedef struct packed {
       logic     [retireWidth-1:0] arch_valids;
       MicroOp_t [retireWidth-1:0] uops;
   } CommitSignals_t;

   logic           clock;
   logic           reset;
   CommitSignals_t commit;

endinterface
```

Chisel-SystemVerilog Integration

SystemVerilog Wrapper

```
module BoomCoreHarnessWrapper_A
(
   input clock,
   input reset,

   input commit_arch_valids_0,
   input commit_arch_valids_1,
   input commit_arch_valids_2,

   input [39:0] commit_uops_0_debug_pc,
   input [39:0] commit_uops_1_debug_pc,
   input [39:0] commit_uops_2_debug_pc
);

   BoomCoreHarnessIntf #(.coreMaxAddrBits(40),
                         .retireWidth    (3 ) ) i0();

   assign i0.clock = clock;
   assign i0.reset = reset;

   assign i0.commit.arch_valids[0] = commit_arch_valids_0;
   assign i0.commit.arch_valids[1] = commit_arch_valids_1;
   assign i0.commit.arch_valids[2] = commit_arch_valids_2;

   assign i0.commit.uops[0].debug_pc = commit_uops_0_debug_pc;
   assign i0.commit.uops[1].debug_pc = commit_uops_1_debug_pc;
   assign i0.commit.uops[2].debug_pc = commit_uops_2_debug_pc;

   BoomCoreHarness
   #(.coreMaxAddrBits(40),
     .retireWidth    (3 ) )
   SV_instance
   (i0);

endmodule
```

```
module BoomCoreHarnessWrapper_B
(
   input clock,
   input reset,

   input commit_arch_valids_0,
   input commit_arch_valids_1,
   input commit_arch_valids_2,
   input commit_arch_valids_3,

   input [39:0] commit_uops_0_debug_pc,
   input [39:0] commit_uops_1_debug_pc,
   input [39:0] commit_uops_2_debug_pc,
   input [39:0] commit_uops_3_debug_pc
);

   BoomCoreHarnessIntf #(.coreMaxAddrBits(40),
                         .retireWidth    (4 ) ) i0();

   assign i0.clock = clock;
   assign i0.reset = reset;

   assign i0.commit.arch_valids[0] = commit_arch_valids_0;
   assign i0.commit.arch_valids[1] = commit_arch_valids_1;
   assign i0.commit.arch_valids[2] = commit_arch_valids_2;
   assign i0.commit.arch_valids[3] = commit_arch_valids_3;

   assign i0.commit.uops[0].debug_pc = commit_uops_0_debug_pc;
   assign i0.commit.uops[1].debug_pc = commit_uops_1_debug_pc;
   assign i0.commit.uops[2].debug_pc = commit_uops_2_debug_pc;
   assign i0.commit.uops[3].debug_pc = commit_uops_3_debug_pc;

   BoomCoreHarness
   #(.coreMaxAddrBits(40),
     .retireWidth    (4 ) )
   SV_instance
   (i0);

endmodule
```

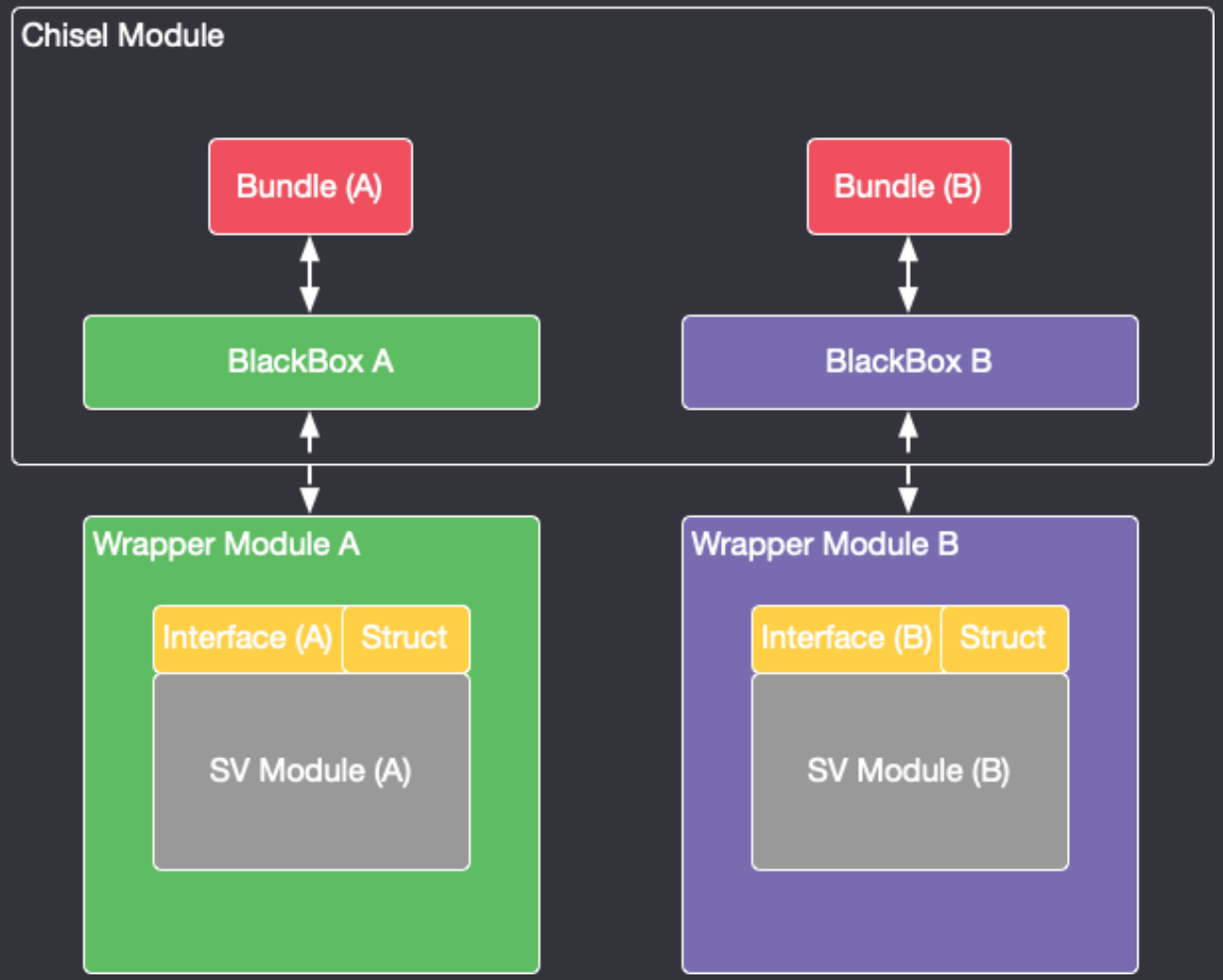
Chisel-SystemVerilog Integration

Chisel BlackBox

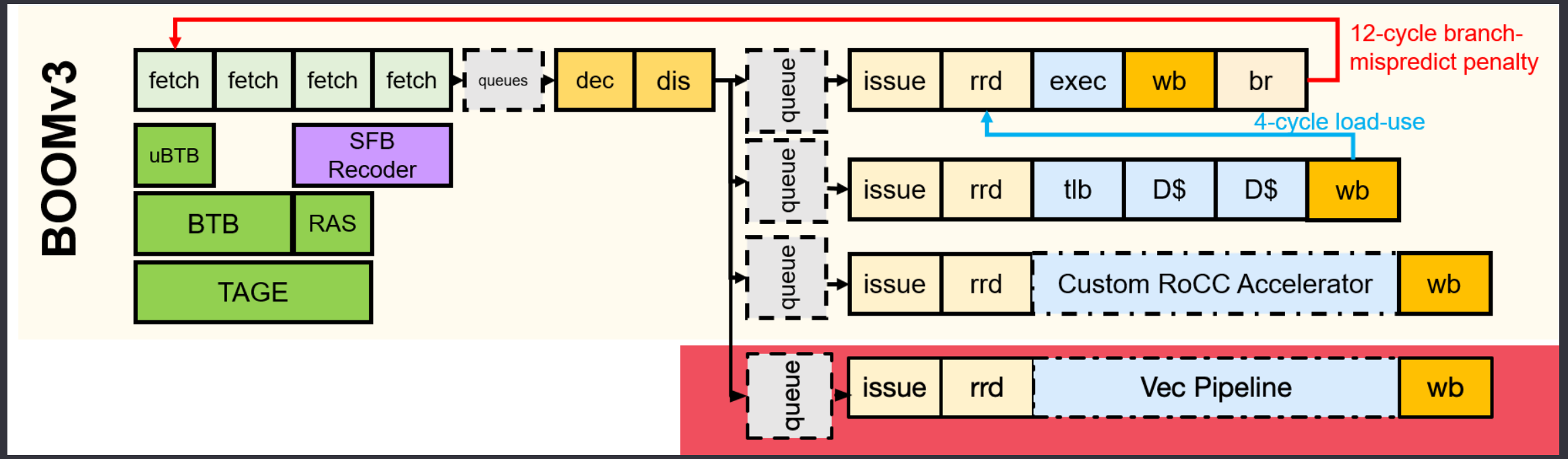
```
class BoomCoreHarnessWrapper_A extends BlackBox
with HasBlackBoxResource {
  val io = IO(new Bundle {
    val clock = Input(Bool())
    val reset = Input(Bool())
    val commit = Input(new ToyCommitSignals(40, 3))
  })
  addResource("/vsrc/core_harness_interface.v")
  addResource("/vsrc/core_harness.v")
  addResource("/vsrc/core_harness_wrapper_A.v")
}
```

```
class BoomCoreHarnessWrapper_B extends BlackBox
with HasBlackBoxResource {
  val io = IO(new Bundle {
    val clock = Input(Bool())
    val reset = Input(Bool())
    val commit = Input(new ToyCommitSignals(40, 4))
  })
  addResource("/vsrc/core_harness_interface.v")
  addResource("/vsrc/core_harness.v")
  addResource("/vsrc/core_harness_wrapper_B.v")
}
```

Chisel-SystemVerilog Integration



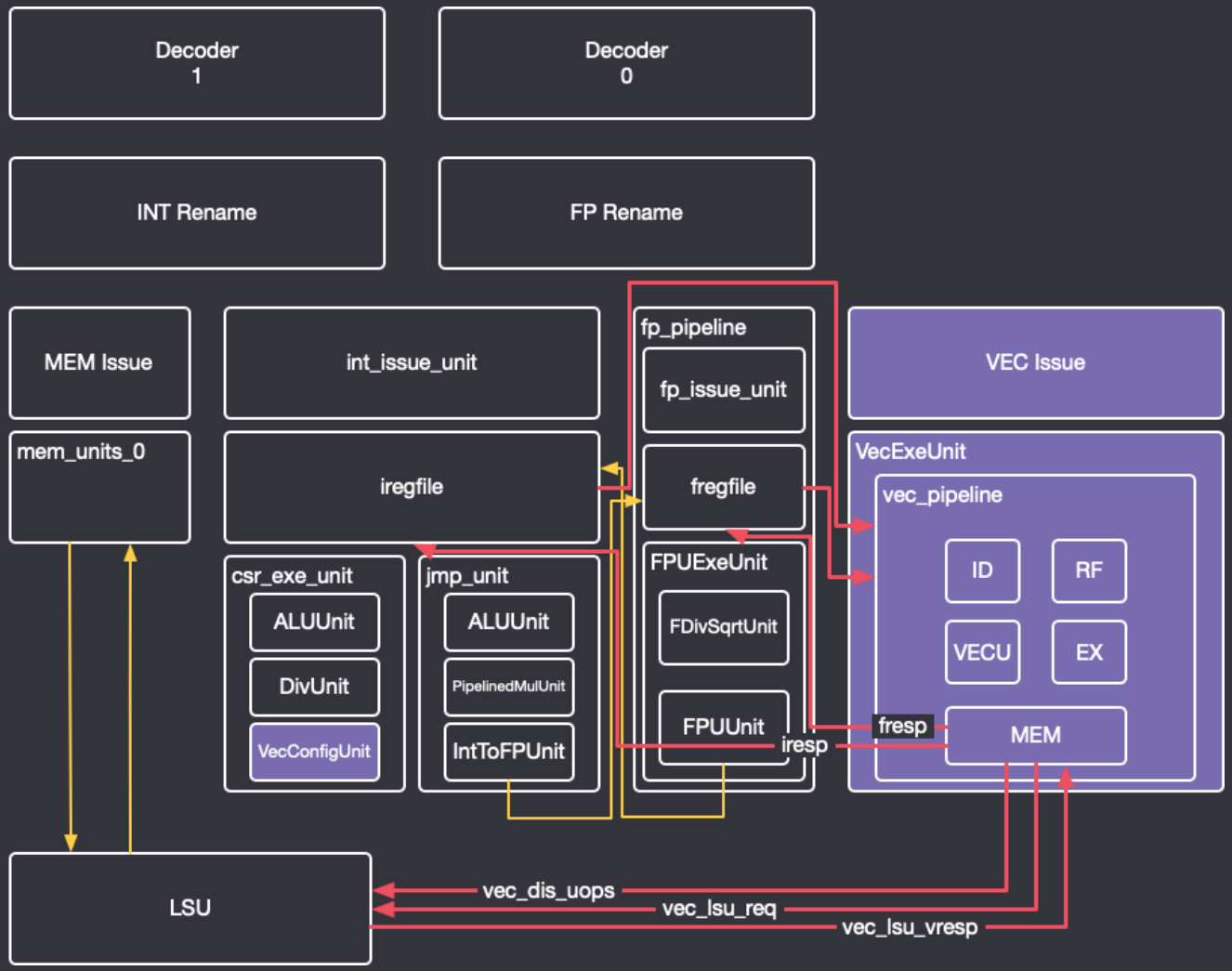
In-Order Pipeline in Out-of-Order Core



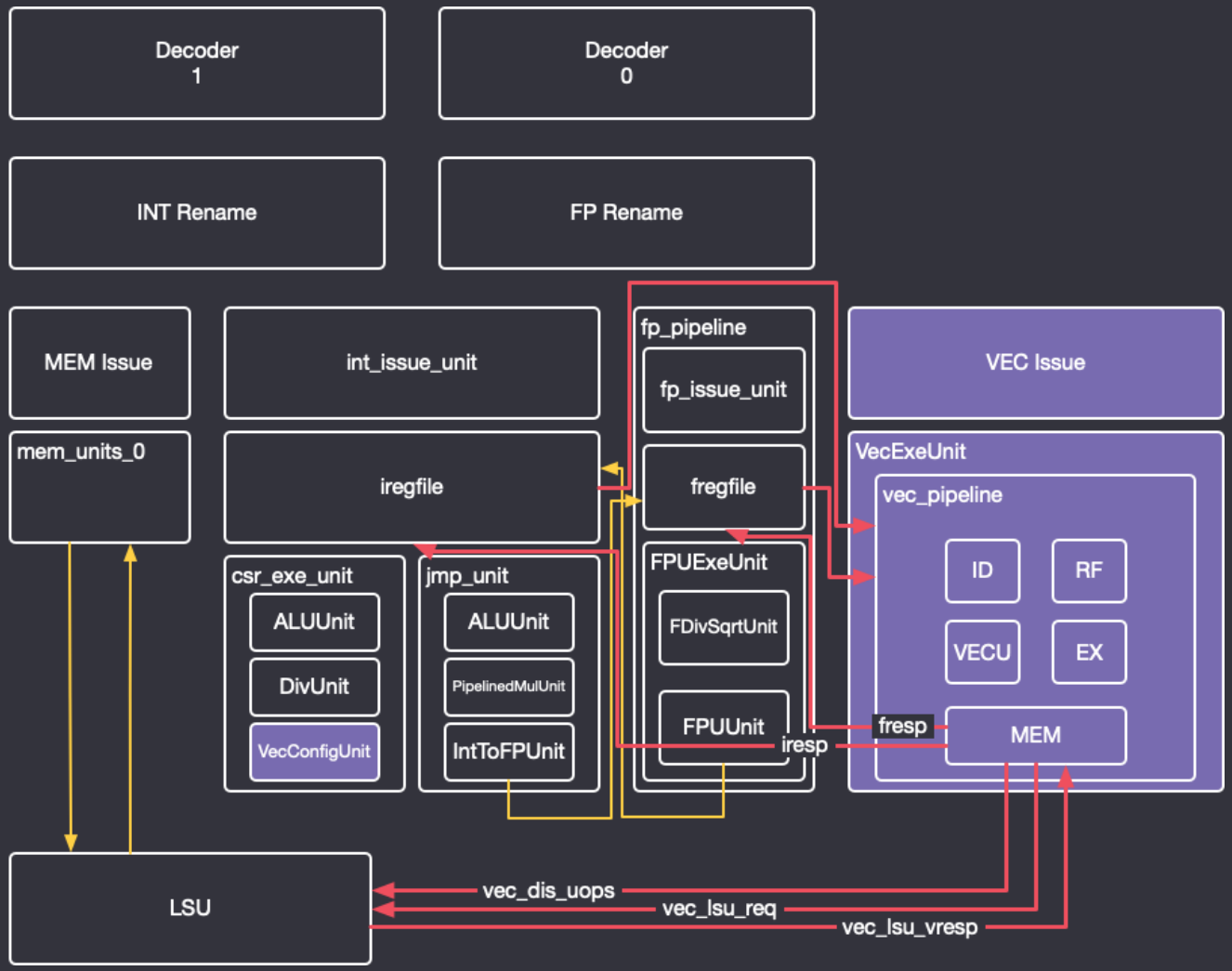
tenstorrent

https://carrv.github.io/2020/papers/CARRV2020_paper_15_Zhao.pdf

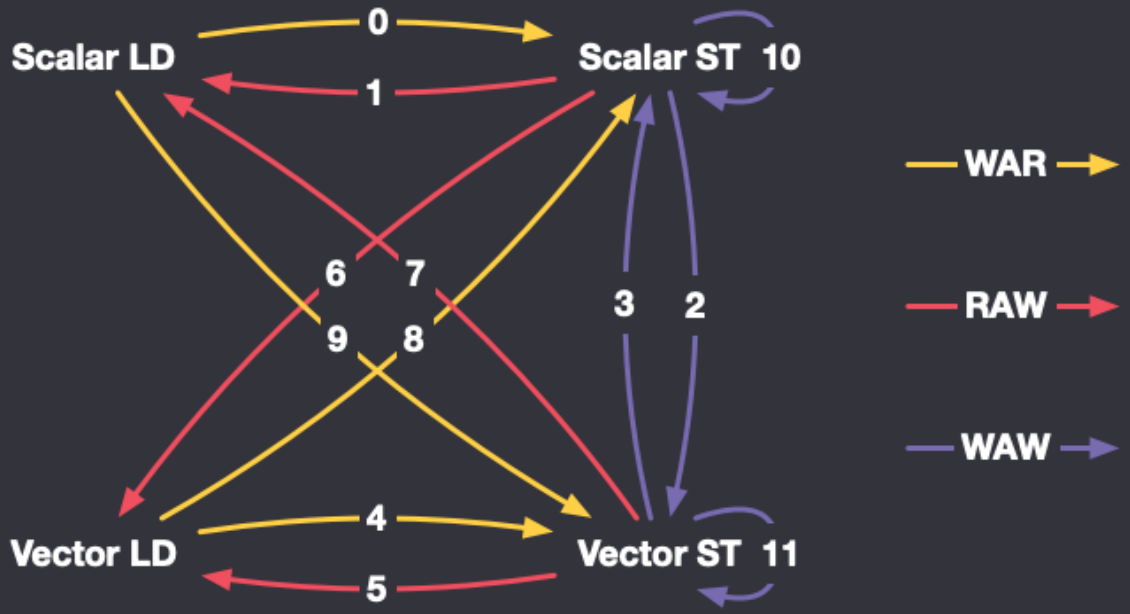
In-Order Pipeline in Out-of-Order Core



LSU Retrofitting

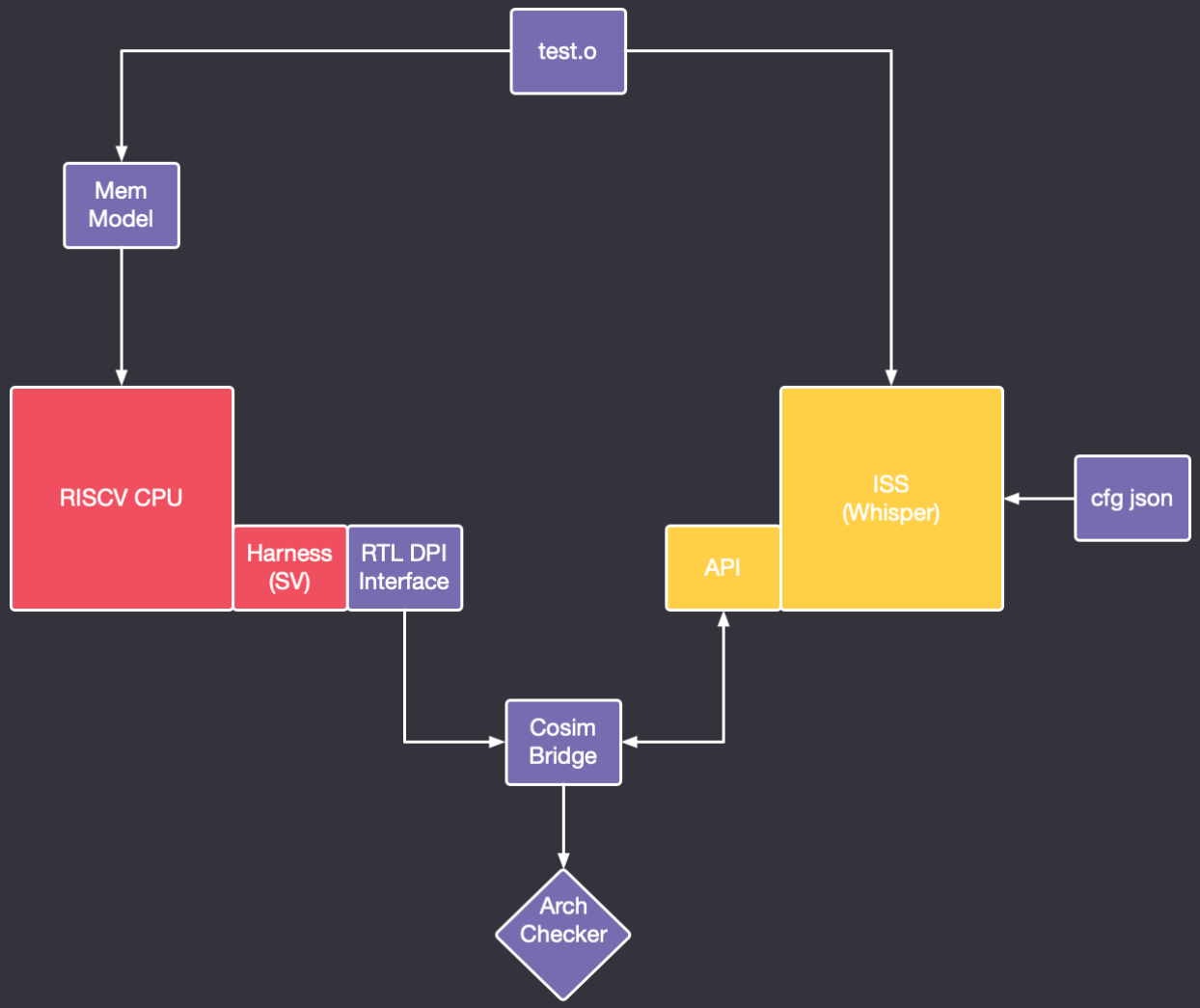


Memory Ordering



Cases	Mechanism
0, 2, 3, 8, 9, 10, 11	stq_commit_head
1	dep_st_mask & addr_match (precise)
7	dep_st_mask (imprecise)
4, 5	Vector Load/Store are serialized by vector unit
6	Missing (FIXME)

Debug Harness



tenstorrent

https://github.com/tenstorrent/cosim-arch-checker

Thank You

- GitHub Repository: https://github.com/tenstorrent/riscv-ocelot
- Forked From: https://github.com/riscv-boom/riscv-boom
- Floating Point Units: http://www.jhauser.us/arithmetic/HardFloat.html
- Presentation prepared with assistance from ChatGPT