RISCV-Business: A Configurable, Extensible RISC-V Core

Electrical and Computer Engineering

John Skubic, Jacob R. Stevens, Chuan Yean Tan, Dr. Mark Johnson, Dr. Matthew Swabey

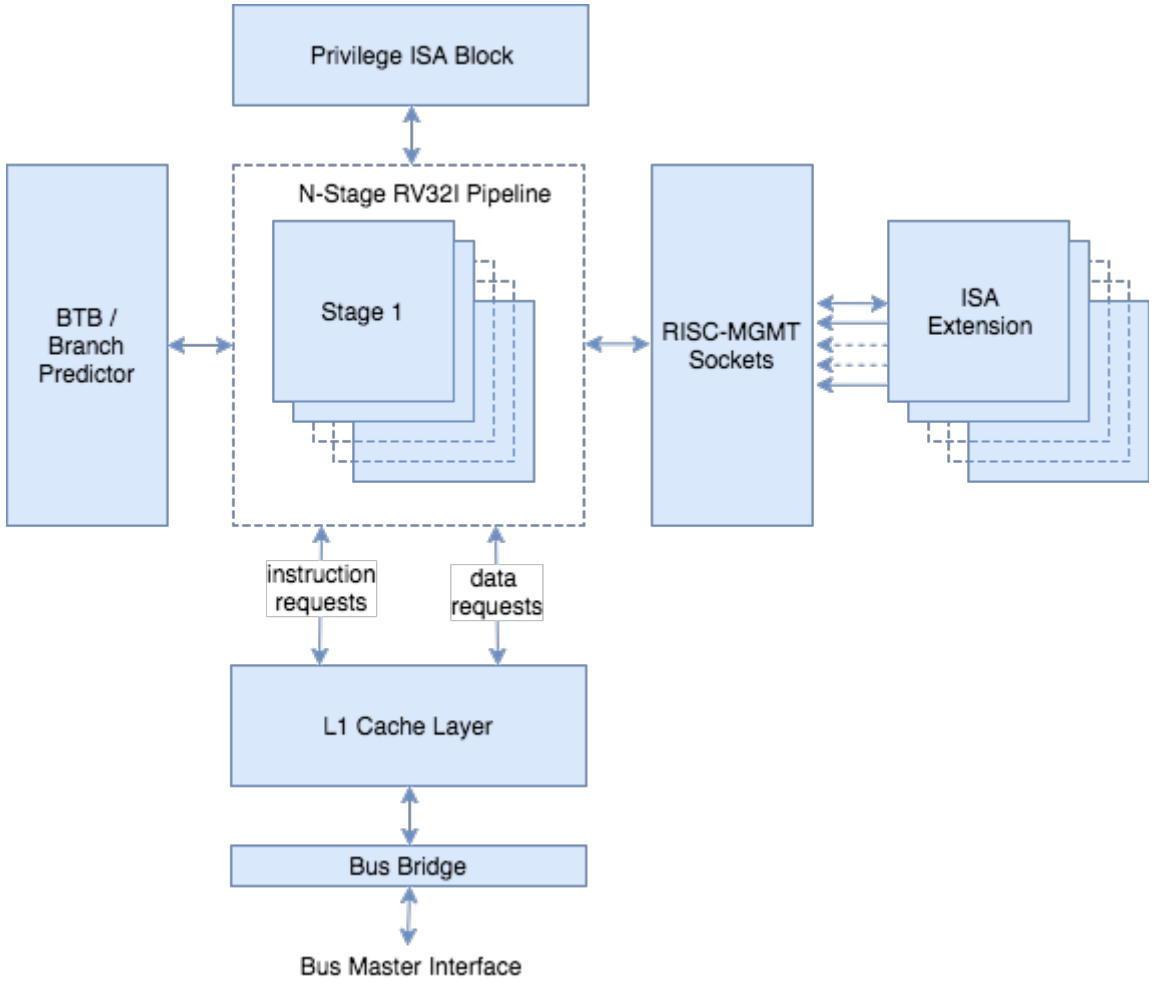
PURDUE UNIVERSITY®

Motivation

- Easily accessible core for wide array of hardware designers
  - Open source
  - SystemVerilog
  - WAF build system
  - Automation scripts
- Prototype ASIPs and customize supported ISA
  - Standard extensions
  - Nonstandard extensions

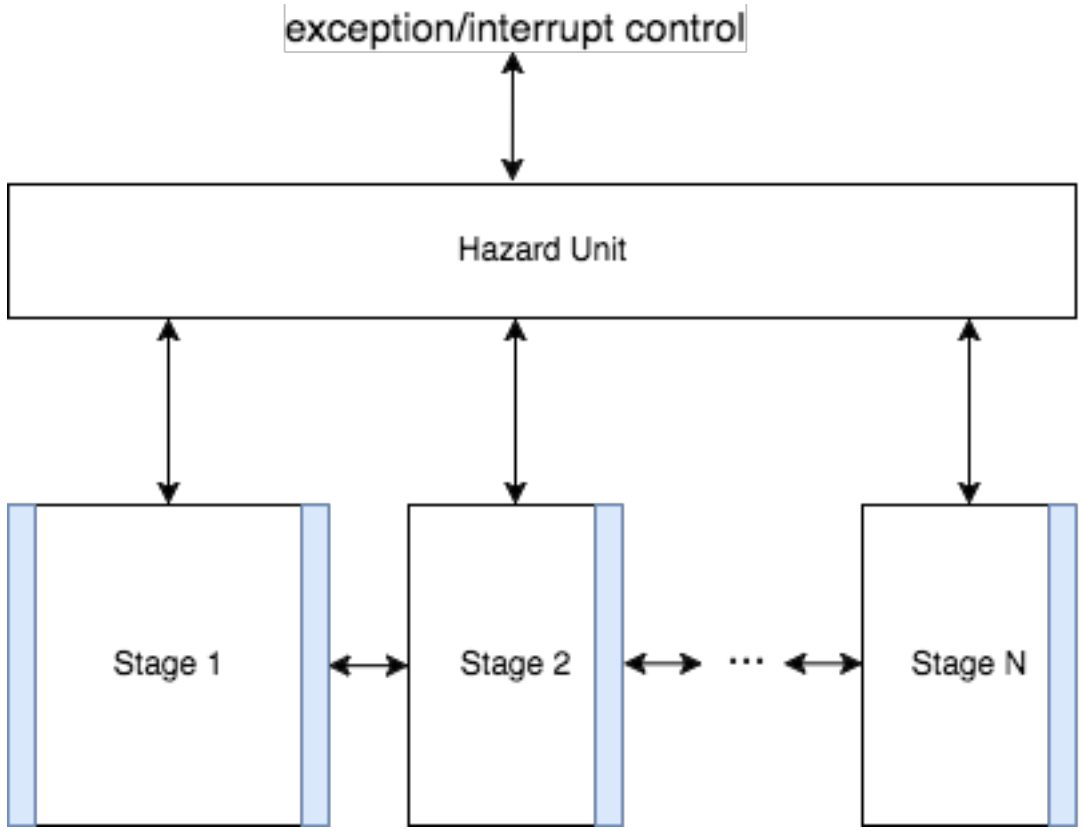
Architectural Overview

- Broken up into **Configurable Components (CCs)**
- Well defined interfaces between CCs
- Each CC independent of the version of other CCs



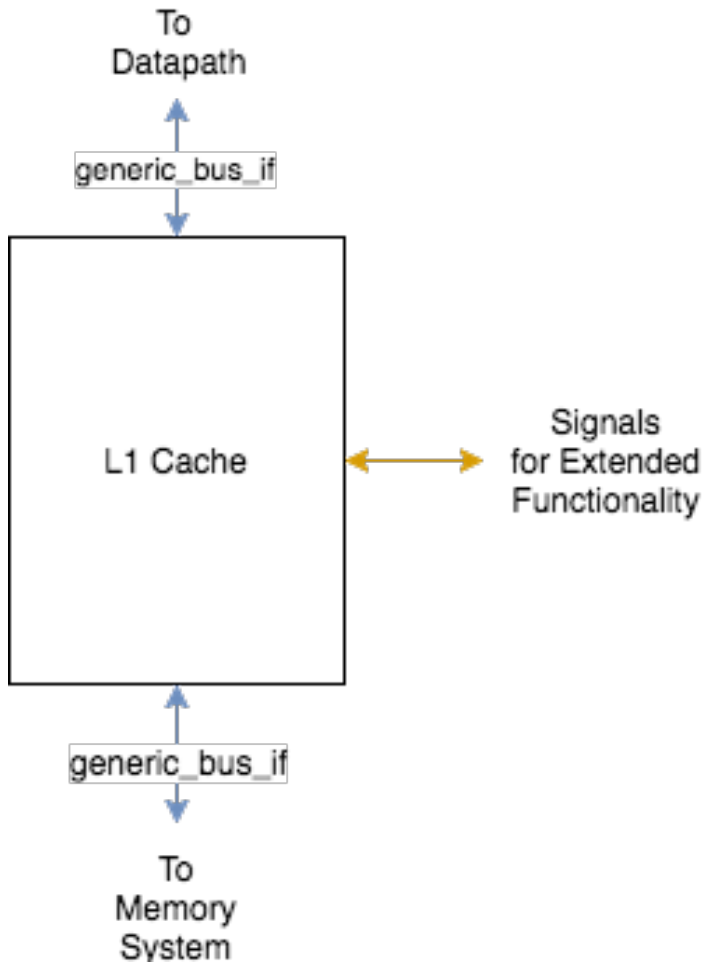
Configurable Components: Pipeline

- Each pipeline depth is its own module
- Hazard unit will handle hazards and forwarding
- Hazard unit will interface with privilege ISA implementation



Configurable Components: Caches

- Consists of two generic bus interfaces from datapath
  - Data requests
  - Instruction requests
- One or two outgoing generic bus interfaces
- Extra signals may be added for more advanced caches



Configurable Components: Bus

- Takes in one generic bus interface
  - pipelined bus interface
- Allows pipelined and non-pipelined busses

Signal Name	Bit Width	Description
addr	32	Address for RAM transaction
busy	1	Busy signal ind
rdata	32	Read data coming from RAM
ren	1	Read request
wdata	32	Write data going to RAM
wen	1	Write request

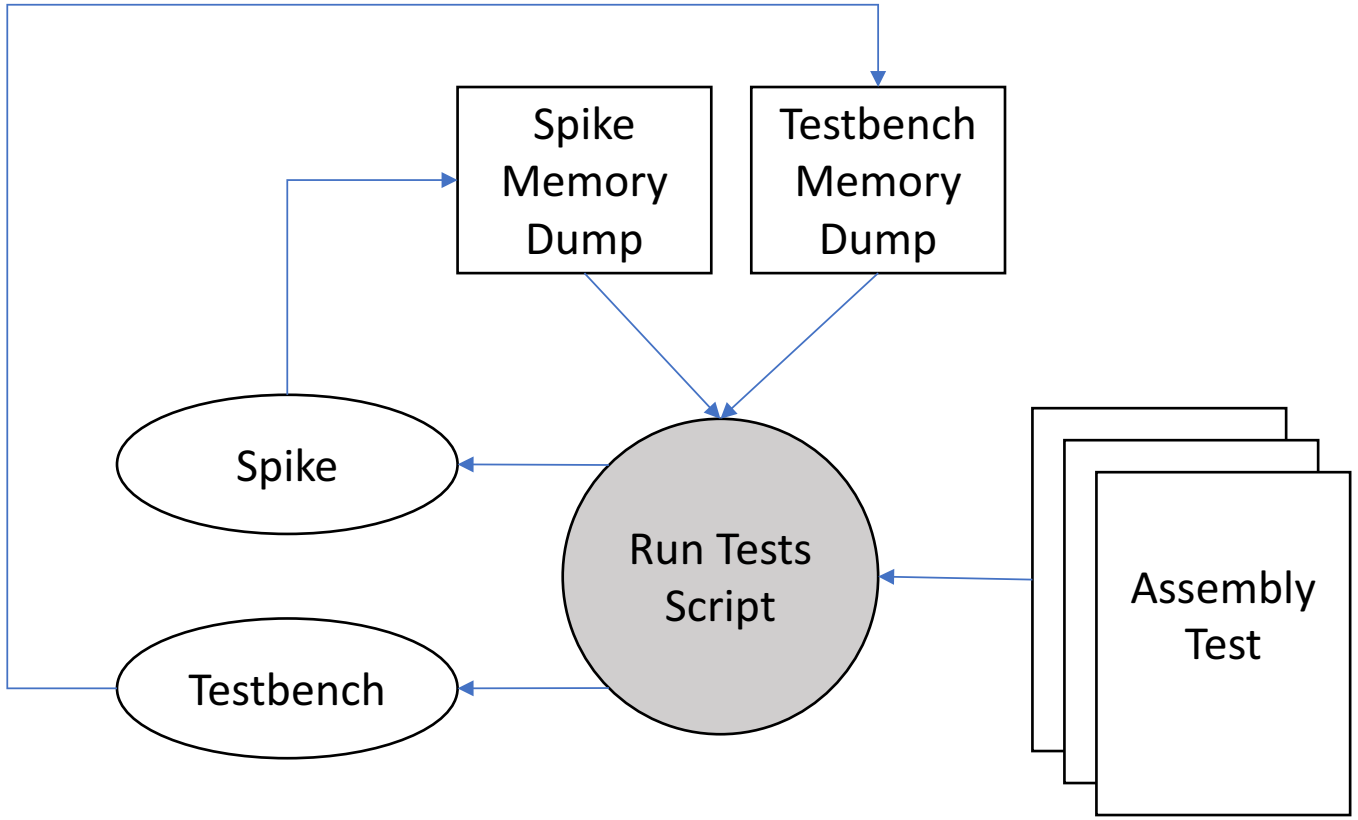
Configuring a Core: Example YAML

```
# Microarchitectural Configurations
microarch_params:
  # Branch/Jump Configurations
  br_predictor_type : "not_taken"

  # Cache configurations
  cache_config   : "separate"
  dcache_type    : "pass_through"
  icache_type    : "pass_through"

  # Bus configurations
  bus_endianness       : "big"
  bus_interface_type   : "generic_bus_if"
```

```
# ISA Configurations
isa_params:
  xlen : 32


# RISC-MGMT Extension Configuration
risc_mgmt_params:
  standard_extensions:
    -rv32m
```

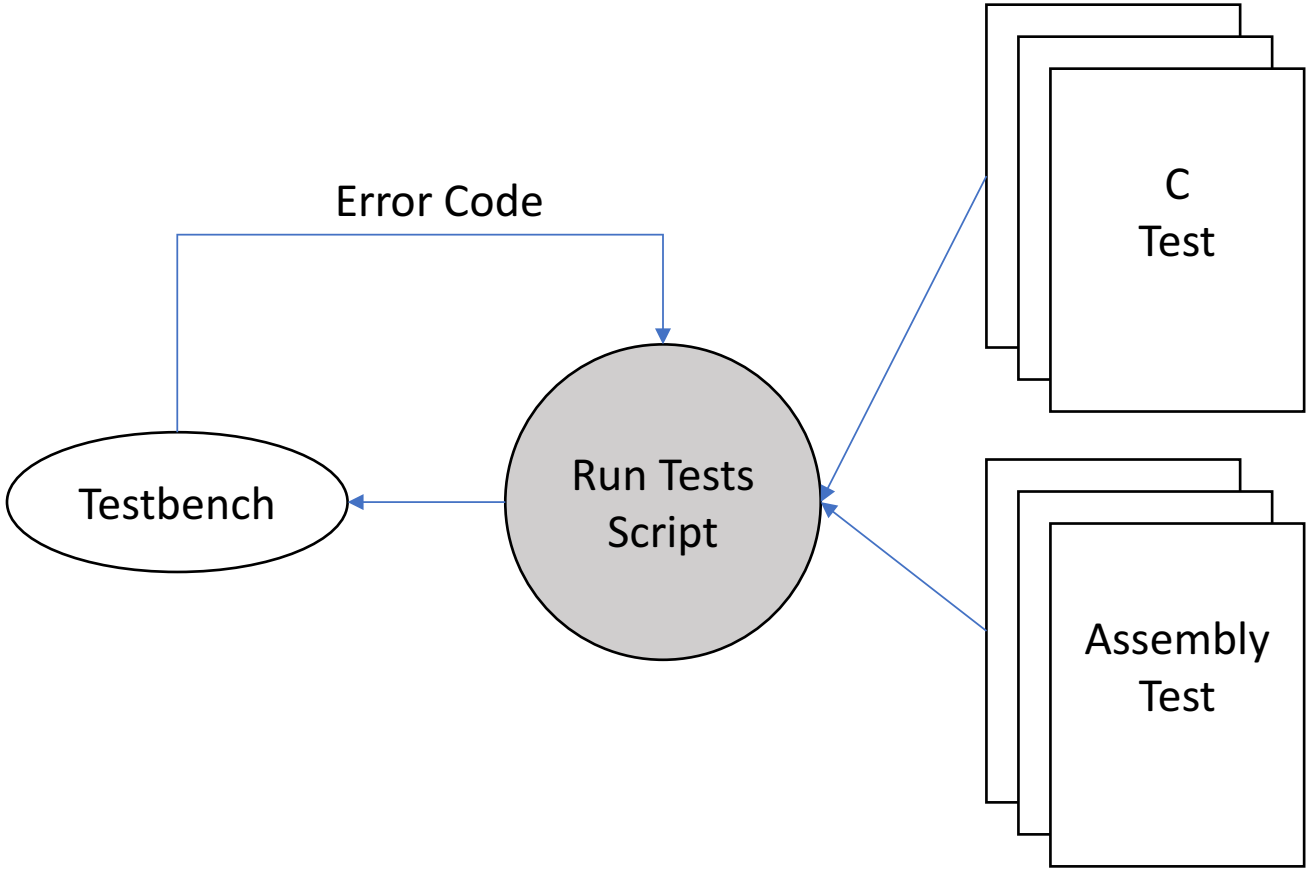
Verification and Testing: Memory Based

- Tests basic functionality



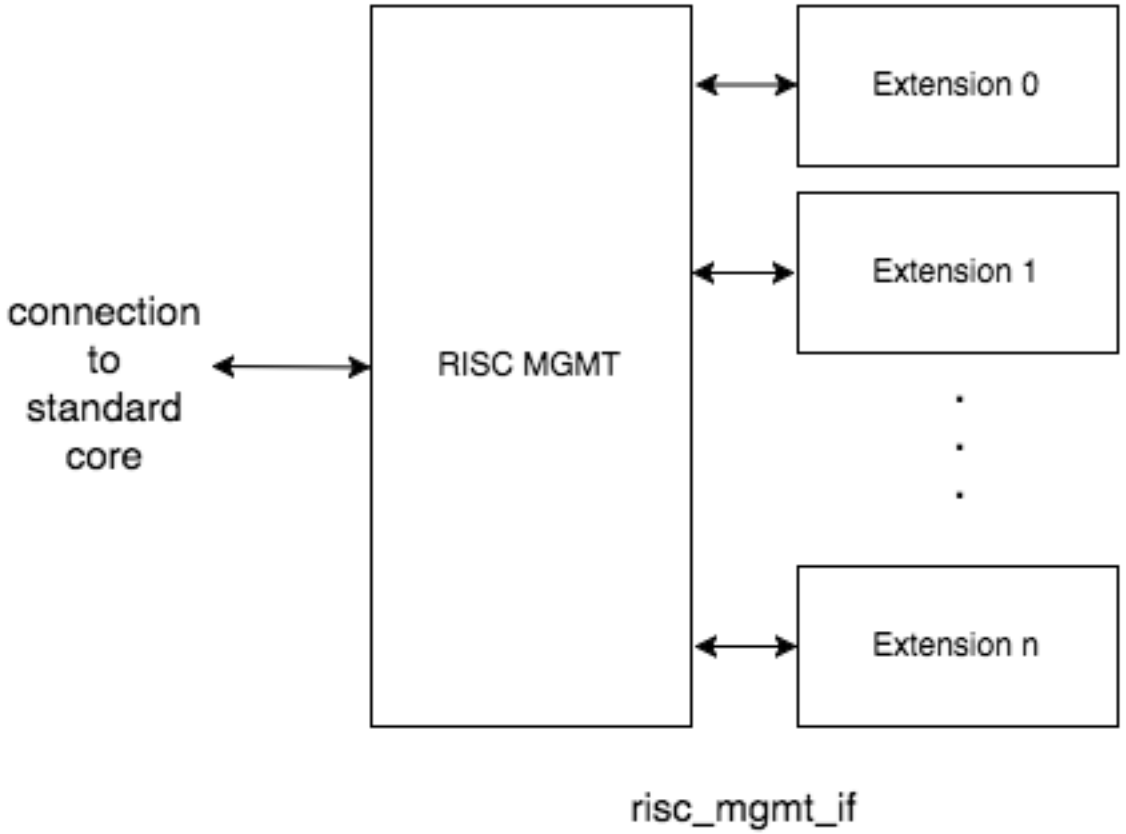
Verification and Testing: Self Tests

- Self testing C and Assembly
- Writes error code to register



RISC-MGMT Overview

- RISC Massively Generic Modification Tie-in
- Simple interface for extensions
- Supports standard and nonstandard extensions
- Abstracts away pipeline depth

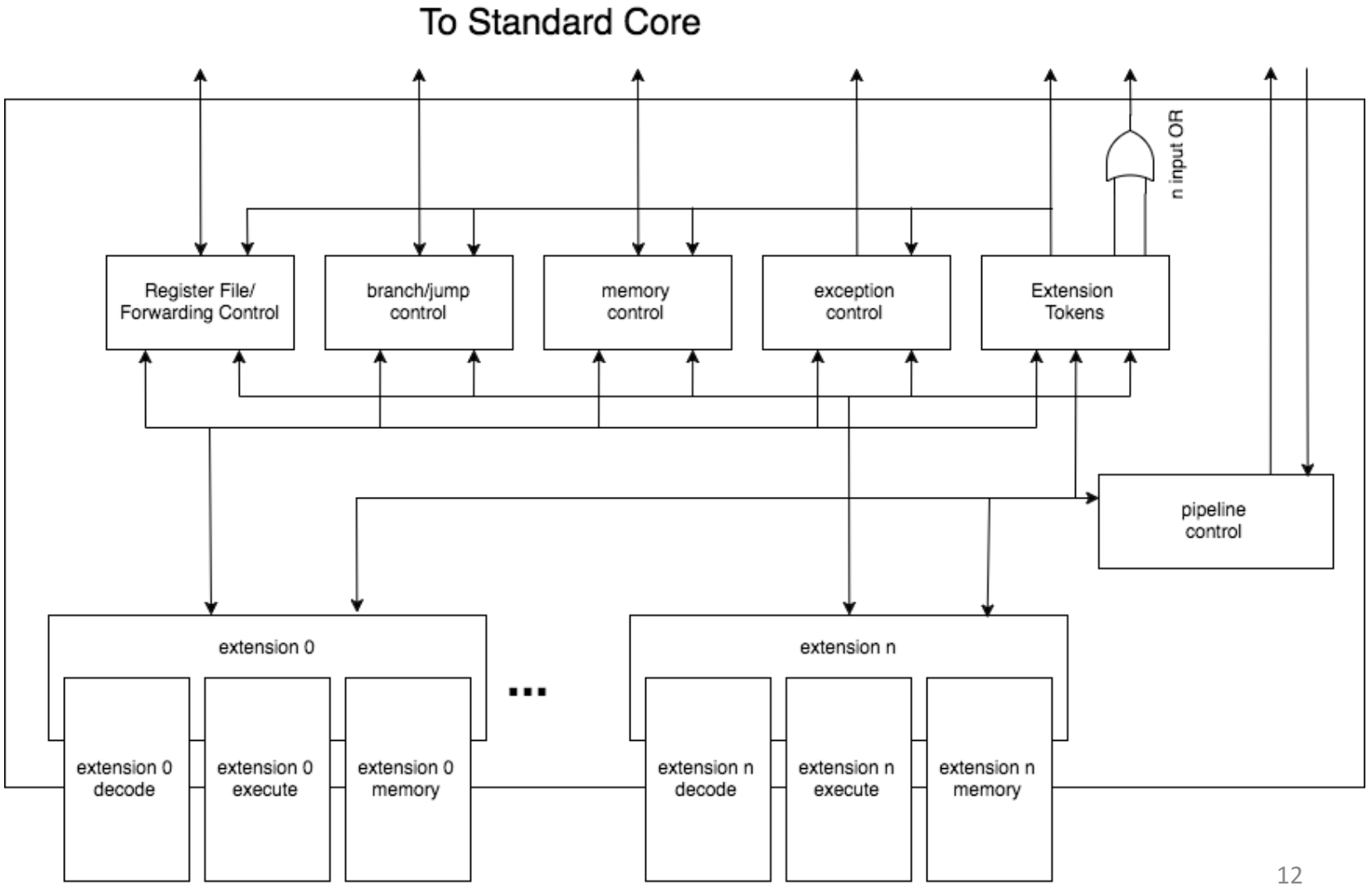


RISC-MGMT Extension Architecture

- 3 stages
  - Decode
  - Execute
  - Memory
- User provides a package containing:
  - Packed struct of signals between decode and execute
  - Packed struct of signals between execute and memory
  - Other structs used by the extension

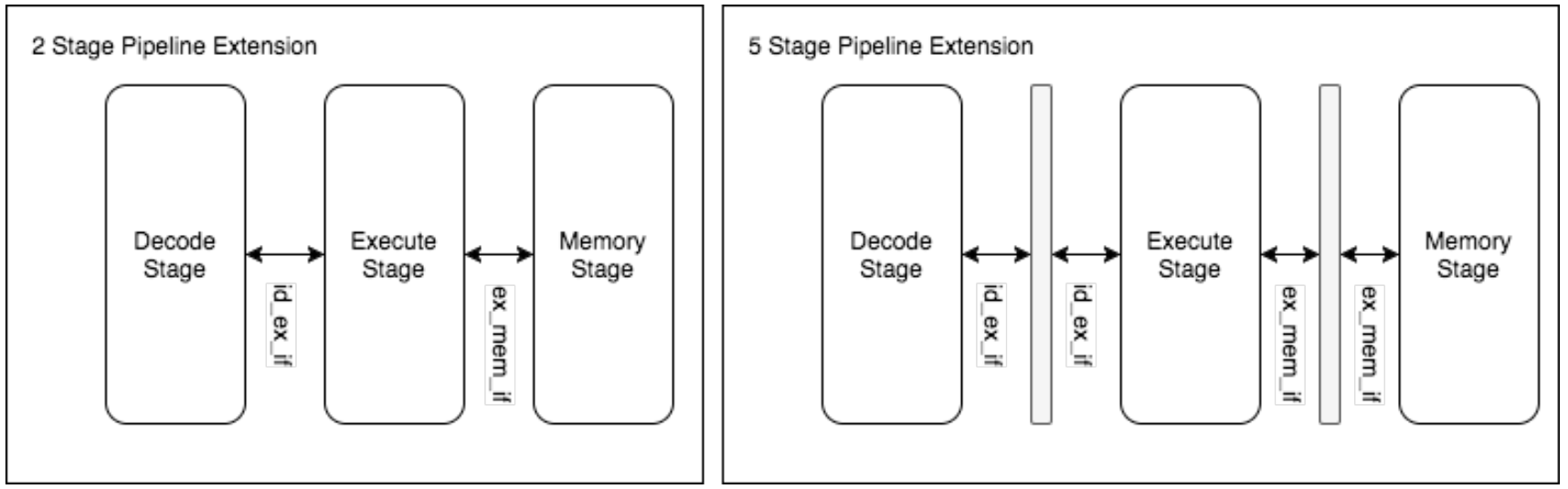
RISC-MGMT Architecture (In-Order)

- Resource sharing
  - Branch/jump unit
  - Register file reads
  - Load/store unit
- Exception handling
- Forwarding
- Register file writes



RISC-MGMT Responsibilities: Latch Insertion

- RISC-MGMT supports 2 to 5 stage in-order pipelines
- RISC-MGMT may treat extensions as functional units for OoO



RISC-MGMT Overhead: RV32M

- RISC-MGMT implementation has a 4.7% decrease in clock speed
- RISC-MGMT implementation has a 1.5% decrease in Logic Elements

	RV32M Direct	RV32M RISC-MGMT	Standard Core
Logic Elements:	5595	5509	4461
Comb Functions:	5557	5473	4422
Registers:	2077	2077	1742
Frequency (MHz)	74.97	71.45	82.59

FPGA: Altera DE2-115, Cyclone IV E

RISCV-Business Status

- Pipeline:
  - 2 stage in-order
- Caches:
  - Pass through
  - N-way set associative
- Buses:
  - AHB
  - Avalon

- Standard Extensions:
  - RV32M
  - RV32C
- Nonstandard Extensions:
  - CRC32

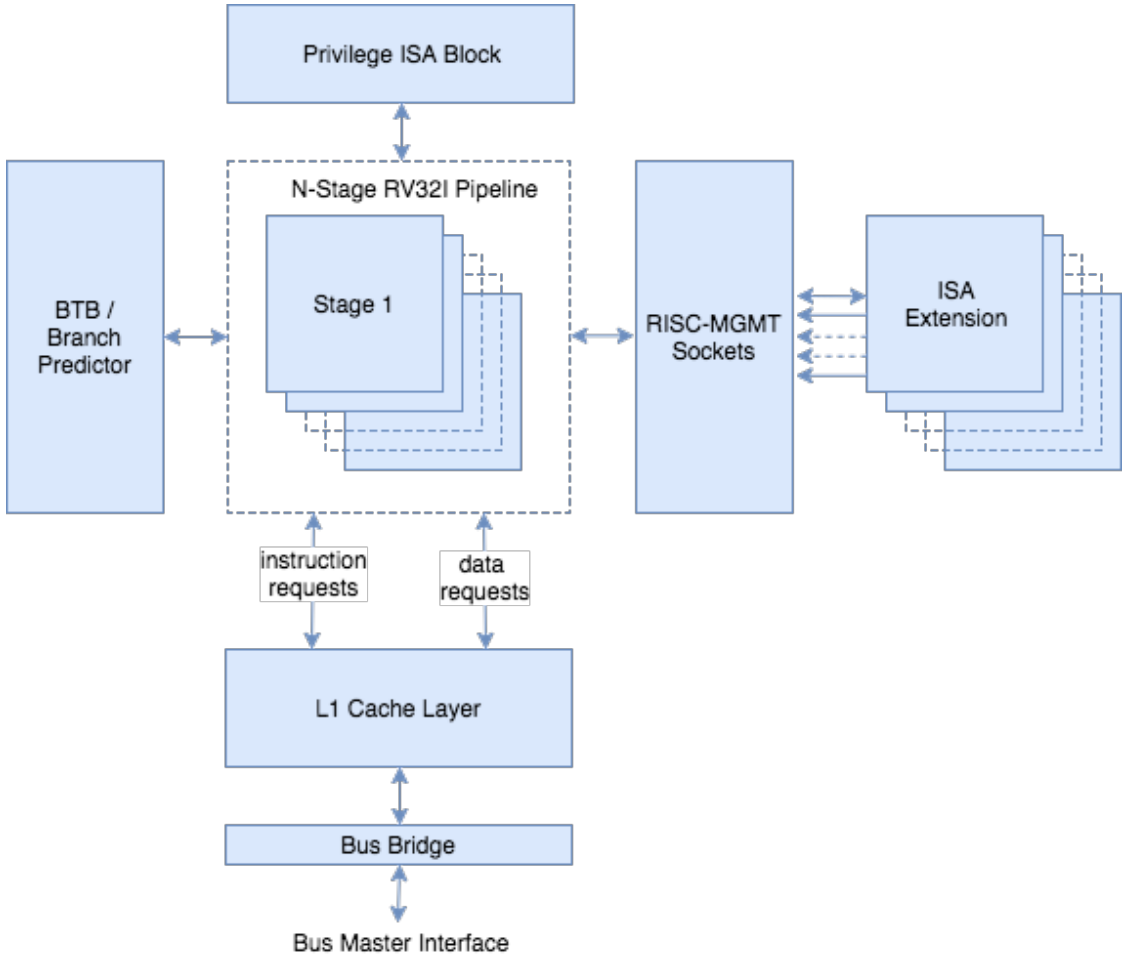
- Not Started

- In Progress

- Completed

- Documentation and source code:
  https://purduesocet.github.io/projects/

Questions?


Privilege ISA Block
N-Stage RV32I Pipeline
BTB / Branch Predictor
Stage 1
RISC-MGMT Sockets
ISA Extension
instruction requests
data requests
L1 Cache Layer
Bus Bridge
Bus Master Interface

Speaker Bio: John Skubic

John Skubic is a masters student at Purdue University studying Electrical and Computer Engineering with a focus in Computer Architecture. He is a member of the Purdue SoCet group which designs and tapes-out custom SoCs. Post graduation, John will be employed as an digital hardware engineer at Qualcomm.