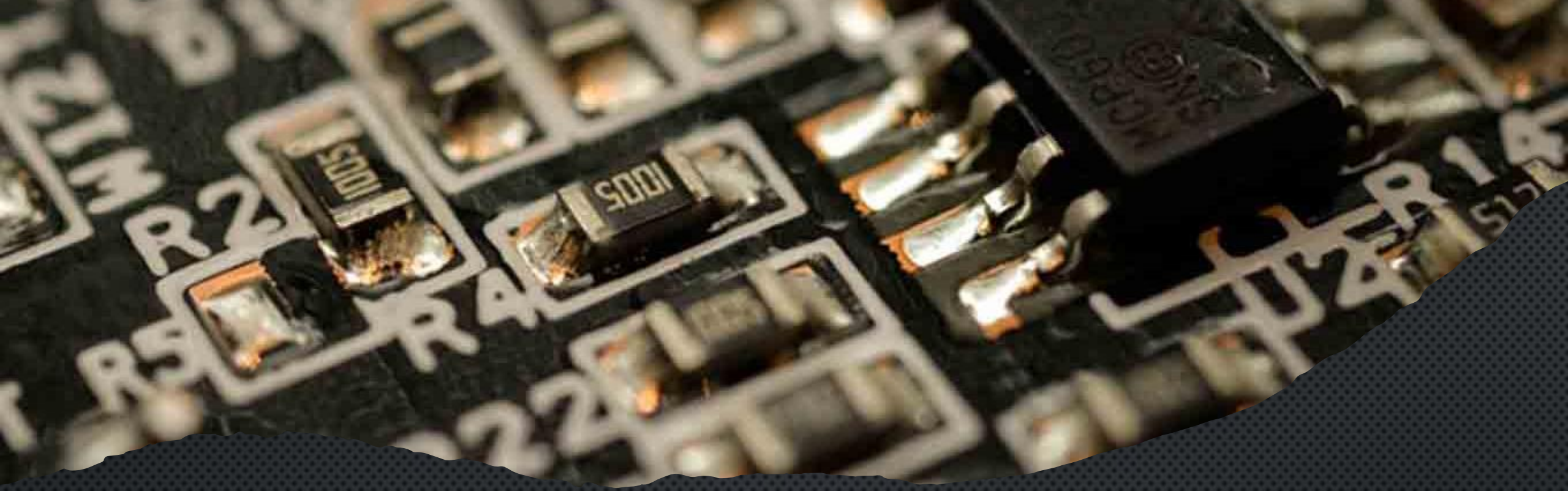
RISC-V PROCESSOR WITH SCOREBOARDING AND REGISTER RENAMING

ARYAN | ABHIRAM



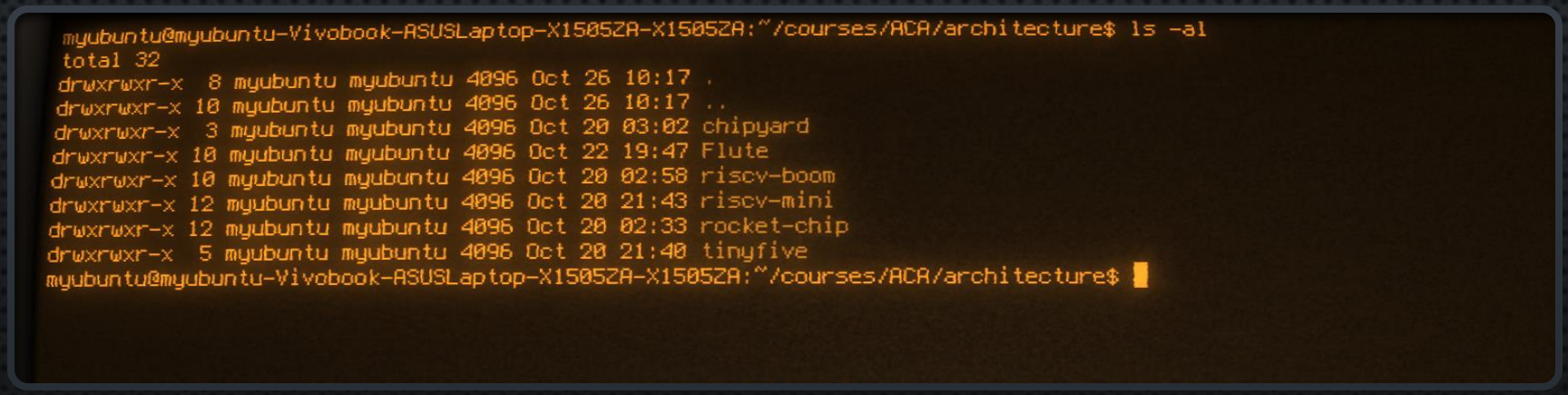
PROJECT DESCRIPTION

- To implement Scoreboarding and Register renaming in Risc-V core.
- To evaluate Performance based on metrics like instruction throughput, execution latency.

Options explored and issues encountered

Before Choosing BSV based cpu core, we've tried several cpu cores. Each of them have their own set of limitations

- BOOM:
  - The documentation seems really good but may not be idea for our project
  - Features like register renaming and reorder buffers are pre-implemented, so doesn't fit for our project
- Chipyard:
  - The project is very big and well maintained but it is really complex and is pretty out of scope for simple things that we are trying to do.
- RISC-V Mini
  - Their github has been archived since a few years and there is no official support.
- Rocket Chip
  - This processor is the base for a lot of other processors based on chisel BUT
  - The codebase is archived since a few years and is unmaintained as of right now.



WHY FLUTE WITH BLUESPEC?

- **BLUESPEC** IS PREFERRED FOR ITS EASE OF USE COMPARED TO OTHER LANGUAGES LIKE CHISEL. SINCE ITS BEING TAUGHT IN CLASS, IT WILL BE EASIER FOR US TO IMPLEMENT THE LOGIC.
- **FLUTE** ONE OF THE MOST POPULAR RISC-V CORES BASED ON BLUESPEC VERILOG. OTHER ALTERNATIVES ARE PICCOLO AND ROCKET (BERKLEY). BUT THIS ONE SEEMS TO BE MORE RECENTLY MAINTAINED.

FLUTE RV32/64

5-Stage In-Order

Bluespec

CURRENT STATUS

- Installed BSC (Bluespec System Compiler)
- Compiled FLUTE from source
- Ran Tests and explored File Structure

```
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-srlw.log
Worker 3: Test: rv64ui-p-srlw PASS [So far: total 60, executed 14, PASS 14, FAIL 0]
    Exec: /home/myubuntu/courses/ACA/Flute/Tests/elf_to_hex/elf_to_hex /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-add Mem
    Exec: /home/myubuntu/courses/ACA/Flute/builds/Flute_RV32CI_MU_WT_L1_bluesim_tohost/exe_HW_sim +tohost
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-add.log
Worker 2: Test: rv64ui-p-add PASS [So far: total 60, executed 14, PASS 14, FAIL 0]
    Exec: /home/myubuntu/courses/ACA/Flute/Tests/elf_to_hex/elf_to_hex /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-lw Mem
    Exec: /home/myubuntu/courses/ACA/Flute/builds/Flute_RV32CI_MU_WT_L1_bluesim_tohost/exe_HW_sim +tohost
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-lw.log
Worker 1: Test: rv64ui-p-lw PASS [So far: total 60, executed 15, PASS 15, FAIL 0]
    Exec: /home/myubuntu/courses/ACA/Flute/Tests/elf_to_hex/elf_to_hex /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-srli M
    Exec: /home/myubuntu/courses/ACA/Flute/builds/Flute_RV32CI_MU_WT_L1_bluesim_tohost/exe_HW_sim +tohost
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-srli.log
Worker 0: Test: rv64ui-p-srli PASS [So far: total 60, executed 15, PASS 15, FAIL 0]
    Exec: /home/myubuntu/courses/ACA/Flute/Tests/elf_to_hex/elf_to_hex /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-auipc
    Exec: /home/myubuntu/courses/ACA/Flute/builds/Flute_RV32CI_MU_WT_L1_bluesim_tohost/exe_HW_sim +tohost
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-auipc.log
Worker 3: Test: rv64ui-p-auipc PASS [So far: total 60, executed 15, PASS 15, FAIL 0]
    Exec: /home/myubuntu/courses/ACA/Flute/Tests/elf_to_hex/elf_to_hex /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-ori Me
    Exec: /home/myubuntu/courses/ACA/Flute/builds/Flute_RV32CI_MU_WT_L1_bluesim_tohost/exe_HW_sim +tohost
    Writing log: /home/myubuntu/courses/ACA/Flute/Tests/Logs/rv64ui-p-ori.log
Worker 2: Test: rv64ui-p-ori PASS [So far: total 60, executed 15, PASS 15, FAIL 0]
Worker 0 executed 15 tests, of which 15 passed
Worker 1 executed 15 tests, of which 15 passed
Worker 2 executed 15 tests, of which 15 passed
Worker 3 executed 15 tests, of which 15 passed
Total tests: 60 tests
Executed:    60 tests
PASS:        60 tests
FAIL:        0 tests
Finished running regressions; saved logs in Logs/
myubuntu@myubuntu-Vivobook-ASUSLaptop-X1505ZA-X1505ZA:~/courses/ACA/Flute/Tests$
```

```
myubuntu@myubuntu-Vivobook-ASUSLaptop-X1505ZA-X1505ZA:~/courses/ACA/Flute/Tests/Logs$ ls
rv64mi-p-access.log      rv64ui-p-addiw.log  rv64ui-p-bltu.log     rv64ui-p-lui.log     rv64ui-p-slliw.log  rv64ui-p-sraw.log
rv64mi-p-breakpoint.log  rv64ui-p-add.log    rv64ui-p-bne.log      rv64ui-p-lw.log      rv64ui-p-sll.log    rv64ui-p-srli.log
rv64mi-p-csr.log         rv64ui-p-addw.log   rv64ui-p-fence_i.log  rv64ui-p-lwu.log     rv64ui-p-sllw.log   rv64ui-p-srliw.log
rv64mi-p-illegal.log     rv64ui-p-andi.log   rv64ui-p-jal.log      rv64ui-p-ori.log     rv64ui-p-slti.log   rv64ui-p-srl.log
rv64mi-p-ma_addr.log     rv64ui-p-and.log    rv64ui-p-jalr.log     rv64ui-p-or.log      rv64ui-p-sltiu.log  rv64ui-p-srlw.log
rv64mi-p-ma_fetch.log    rv64ui-p-auipc.log  rv64ui-p-lb.log       rv64ui-p-sb.log      rv64ui-p-slt.log    rv64ui-p-sub.log
rv64mi-p-mcsr.log        rv64ui-p-beq.log    rv64ui-p-lbu.log      rv64ui-p-sd.log      rv64ui-p-sltu.log   rv64ui-p-subw.log
rv64mi-p-sbreak.log      rv64ui-p-bge.log    rv64ui-p-ld.log       rv64ui-p-sh.log      rv64ui-p-srai.log   rv64ui-p-sw.log
rv64mi-p-scall.log       rv64ui-p-bgeu.log   rv64ui-p-lh.log       rv64ui-p-simple.log  rv64ui-p-sraiw.log  rv64ui-p-xori.log
rv64ui-p-addi.log        rv64ui-p-blt.log    rv64ui-p-lhu.log      rv64ui-p-slli.log    rv64ui-p-sra.log    rv64ui-p-xor.log
myubuntu@myubuntu-Vivobook-ASUSLaptop-X1505ZA-X1505ZA:~/courses/ACA/Flute/Tests/Logs$ cat rv64ui-p-xor.log
c_mem_load_elf: /home/myubuntu/courses/ACA/Flute/Tests/isa/rv64ui-p-xor is a 64-bit ELF file
Section .text.init      : addr         80000000 to addr         8000077c; size 0x     77c (= 1916) bytes
Section .tohost         : addr         80001000 to addr         80001048; size 0x      48 (= 72) bytes
Section .riscv.attributes: Ignored
Section .symtab         : Searching for addresses of '_start', 'exit' and 'tohost' symbols
Writing symbols to:    symbol_table.txt
    No 'exit' label found
Section .strtab         : Ignored
Section .shstrtab       : Ignored
Min addr:                  80000000 (hex)
Max addr:                  80001047 (hex)
Writing mem hex to file 'Mem.hex'
Subtracting 0x80000000 base from addresses
Warning: file 'Mem.hex' for memory 'rf' has a gap at addresses 131 to 8388606.
Warning: RegFile 'top.mem_model.rf' -- Read address is out of bounds: 0xaaaaaaaaaaaaaaaa
================================================================
Bluespec RISC-V WindSoC simulation v1.2
Copyright (c) 2017-2020 Bluespec, Inc. All Rights Reserved.
================================================================
1: top.soc_top.core.cpu.near_mem.icache.ma_ddr4_ready: Enabling MMU_Cache
```

```
Max addr:                       80001047 (hex)
Writing mem hex to file 'Mem.hex'
Subtracting 0x80000000 base from addresses
Warning: file 'Mem.hex' for memory 'rf' has a gap at addresses 131 to 8388606.
Warning: RegFile 'top.mem_model.rf' -- Read address is out of bounds: 0xaaaaaaaaaaaaaaaa
================================================================
Bluespec RISC-V WindSoC simulation v1.2
Copyright (c) 2017-2020 Bluespec, Inc. All Rights Reserved.
================================================================
1: top.soc_top.core.cpu.near_mem.icache.ma_ddr4_ready: Enabling MMU_Cache
1: top.soc_top.core.cpu.near_mem.dcache.ma_ddr4_ready: Enabling MMU_Cache
INFO: watch_tohost = 1, tohost_addr = 0x80001000
1: top.soc_top.core.cpu.near_mem.dcache.set_watch_tohost: watch 1, addr 80001000
2:top.soc_top.rl_reset_start_initial ...
3: Core.rl_cpu_hart0_reset_from_soc_start
================================================================
CPU: Bluespec  RISC-V  Flute  v3.0 (RV32)
Copyright (c) 2016-2023 Bluespec, Inc. All Rights Reserved.
================================================================
6: D_MMU_Cache: cache size 8 KB, associativity 2, line size 64 bytes (= 16 XLEN words)
6: I_MMU_Cache: cache size 8 KB, associativity 2, line size 64 bytes (= 16 XLEN words)
CPU: Restart at PC = 0x1000
514: Core.rl_cpu_hart0_reset_complete
515: Mem_Controller.set_addr_map: addr_base 0x80000000 addr_lim 0x90000000
515:top.soc_top.rl_reset_complete_initial
instret:0  PC:0x1000  instr:0x297  priv:3
837: D_MMU_Cache: cache size 8 KB, associativity 2, line size 64 bytes (= 16 XLEN words)
****************************************************************
964: top:.rl_terminate_tohost: tohost_value is 0x1 (= 0d1)
    PASS
Simulation speed: 963 cycles, 25799981 nsecs  = 37325 cycles/sec
myubuntu@myubuntu-Vivobook-ASUSLaptop-X1505ZA-X1505ZA:~/courses/ACA/Flute/Tests/Logs$
```

GOALS FOR NEXT PRESENTATION

1. **Understand RISC-V Structures**
    - Underrstand File Structure And Alraedy Available Code.
    - Focus on essential components (Register file, ALU, Control Units)
2. **Implement Key Features**

- **Register Renaming:**
    - Implement Register Renaming and Analyse Performance

3. **Scoreboarding (Pseudo Code):**
    - Pseudo Code

THANK YOU