# 22583 Special Topics in Computer Engineering (2)<br>Advanced Digital Systems Design and Verification

## Lab 2: RTL Model of Two-Stage RISC-V Processor

March 13, 2024

## 1 Objectives

For the second lab assignment, you will write an RTL model of a two-stage pipelined RISC-V-v2 (a subset of the RISC-V instruction set) processor.

## 2 Tools

Use one of the following Verilog simulators (or any one of your choice):

- Intel® Quartus®
- www.edaplayground.com
- Download Verilog from: http://bleyer.org/icarus/

## 3 Design Activity

The two-stage pipeline should perform instruction fetch in the first stage, while the second pipeline stage should do everything else including data memory access. If you need to refresh your memory about pipelining and the RISC-V instruction set, we recommend Computer Organization and Design: The Hardware/Software Interface, by Patterson and Hennessey.

Make sure to separate out datapath and memory components from control circuitry. The system diagram in Figure 1 can be used as an initial template for your two-stage RISC-V-v2 processor implementation, but please treat it as a suggestion. Your objective in this lab is to implement the RISC-V-v2 ISA, not to implement the system diagram so feel free to add new control signals, merge modules, or make any other modifications to the system. You will need to turn in a diagram of your datapath anyway, so it is highly recommended that you draw your datapath from the beginning in a program such as Visio, and keep it updated as you design. This reference will be very useful to speed up debugging.

For this lab assignment, you will only be implementing a subset of the RISC-V specification. Figure 2 shows the instructions that you must support.

## 4 Design Flow

Strictly follow the following order:

1. Write the Verilog code for the circuits described above.
2. Write Verilog testbenches to test your modules.
3. Perform Functional Simulation (Run Simulation → Run Behavioral Simulation) and take snapshots of the results.

## 5 Deliverables

You will need to submit to E-learning (as a .zip file) including the following:

- The Verilog code: Your design should be modular.
- Verilog testbenches: test each module separately.
- Run logs: the run result for each test case (text and waveforms)

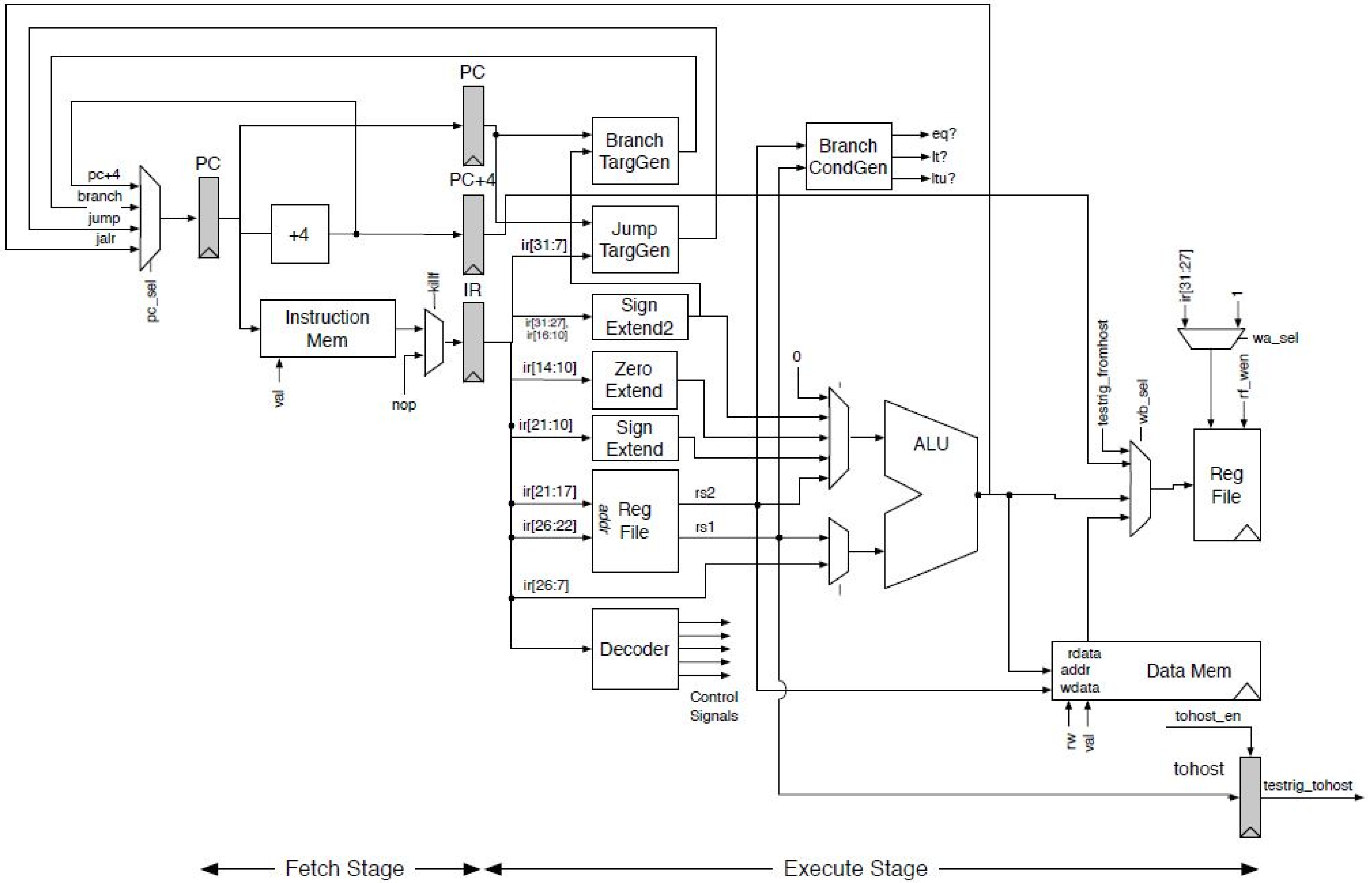

Figure 1: Two-Stage Pipeline for RISC-V-v2 Processor. Shaded state elements need to be correctly loaded on reset

| 31 27 | 26 22 | 21 17 | 16 15 | 14 12 | 11 10 | 9 8 7 | 6 0 | |
|---|---|---|---|---|---|---|---|---|
| jump target | | | | | | | opcode | J-type |
| rd | LUI-immediate | | | | | | opcode | LUI-type |
| rd | rs1 | imm[11:7] | imm[6:0] | | | funct3 | opcode | I-type |
| imm[11:7] | rs1 | rs2 | imm[6:0] | | | funct3 | opcode | B-type |
| rd | rs1 | rs2 | funct10 | | | | opcode | R-type |
| rd | rs1 | rs2 | rs3 | | funct5 | | opcode | R4-type |

### Control Transfer Instructions

| | | | | | | |
|---|---|---|---|---|---|---|
| imm25 | | | | | 1100111 | J imm25 |
| imm25 | | | | | 1101111 | JAL imm25 |
| imm12hi | rs1 | rs2 | imm12lo | 000 | 1100011 | BEQ rs1,rs2,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 001 | 1100011 | BNE rs1,rs2,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 100 | 1100011 | BLT rs1,rs2,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 101 | 1100011 | BGE rs1,rs2,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 110 | 1100011 | BLTU rs1,rs2,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 111 | 1100011 | BGEU rs1,rs2,imm12 |
| rd | rs1 | imm12 | | 000 | 1101011 | JALR.C rd,rs1,imm12 |
| rd | rs1 | imm12 | | 001 | 1101011 | JALR.R rd,rs1,imm12 |
| rd | rs1 | imm12 | | 010 | 1101011 | JALR.J rd,rs1,imm12 |

### Memory Instructions

| | | | | | | |
|---|---|---|---|---|---|---|
| rd | rs1 | imm12 | | 010 | 0000011 | LW rd,rs1,imm12 |
| imm12hi | rs1 | rs2 | imm12lo | 010 | 0100011 | SW rs1,rs2,imm12 |

### Integer Compute Instructions

| | | | | | | |
|---|---|---|---|---|---|---|
| rd | rs1 | imm12 | | 000 | 0010011 | ADDI rd,rs1,imm12 |
| rd | rs1 | 000000 | shamt | 001 | 0010011 | SLLI rd,rs1,shamt |
| rd | rs1 | imm12 | | 010 | 0010011 | SLTI rd,rs1,imm12 |
| rd | rs1 | imm12 | | 011 | 0010011 | SLTIU rd,rs1,imm12 |
| rd | rs1 | imm12 | | 100 | 0010011 | XORI rd,rs1,imm12 |
| rd | rs1 | 000000 | shamt | 101 | 0010011 | SRLI rd,rs1,shamt |
| rd | rs1 | imm12 | | 110 | 0010011 | ORI rd,rs1,imm12 |
| rd | rs1 | imm12 | | 111 | 0010011 | ANDI rd,rs1,imm12 |
| rd | rs1 | rs2 | 0000000 | 000 | 0110011 | ADD rd,rs1,rs2 |
| rd | rs1 | rs2 | 1000000 | 000 | 0110011 | SUB rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 001 | 0110011 | SLL rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 010 | 0110011 | SLT rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 011 | 0110011 | SLTU rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 100 | 0110011 | XOR rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 101 | 0110011 | SRL rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 110 | 0110011 | OR rd,rs1,rs2 |
| rd | rs1 | rs2 | 0000000 | 111 | 0110011 | AND rd,rs1,rs2 |
| rd | imm20 | | | | 0110111 | LUI rd,imm20 |

Figure 2: Instruction listing for RISC-V