RISC-V Processor Verification Report

Lab 25 - Complete Pipelined RISC-V with Tracer

Ahmad Mukhtar

National University of Sciences and Technology (NUST)
Chip Design Centre (NCDC), Islamabad, Pakistan

September 30, 2024

Pipeline Design Overview

The following diagram illustrates the integration of pipeline registers into the RISC-V processor, showing the separation of stages to achieve instruction-level parallelism and improve instruction throughput:

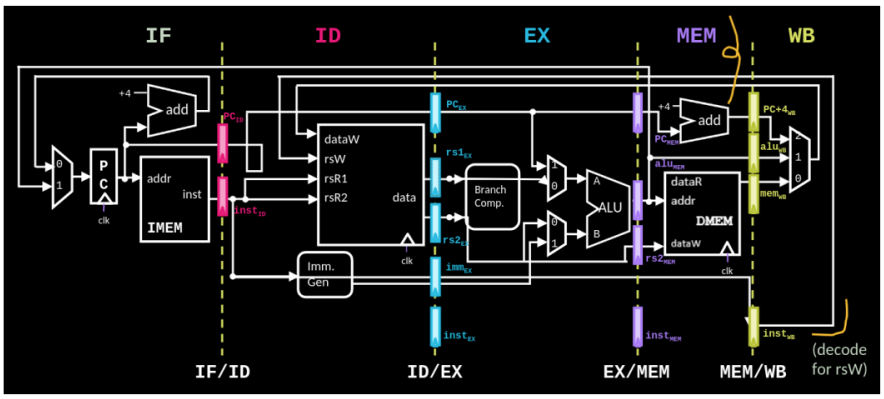

Figure 1: Pipelined Datapath with Registers between Stages

Control Hazards in Pipelining

Control hazards arise when the flow of execution becomes unpredictable, such as with branch instructions. In the RISC-V pipeline, control hazards occur due to the delayed decision of branch outcomes, causing instructions fetched afterward to potentially be invalid.

The following diagram depicts the pipeline datapath where control hazards, particularly those caused by branch instructions, are managed. Specifically, when a branch needs to be taken, the processor must flush subsequent instructions to ensure correct program flow.

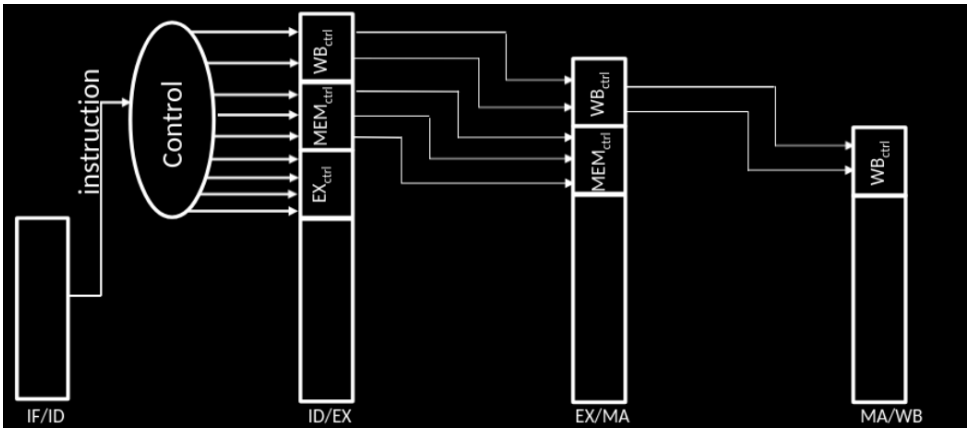

Figure 2: Handling Control Hazards in the Pipeline Stages

Single Cycle Datapath Design

This diagram presents the single-cycle version of the datapath, where all instruction stages are executed in one clock cycle. While simple, it lacks the efficiency gained from pipelining and cannot handle control hazards effectively without significant performance penalties.

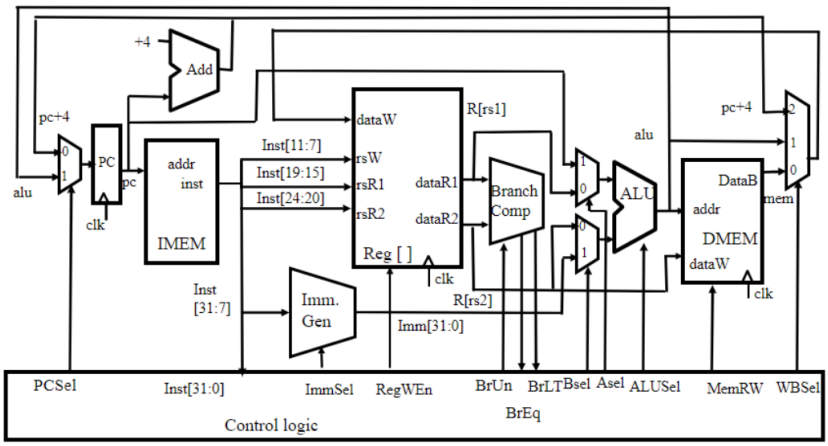

Figure 3: Single Cycle Datapath

Simulation Results: Register File

The following figure shows the detailed results of the register file states during the execution of the given assembly program. The assembly instructions tested include arithmetic operations and memory load/store instructions. This simulation, run using Cadence Xcelium, shows that the register values are updated correctly during each clock cycle, reflecting the data flow across the pipeline stages.

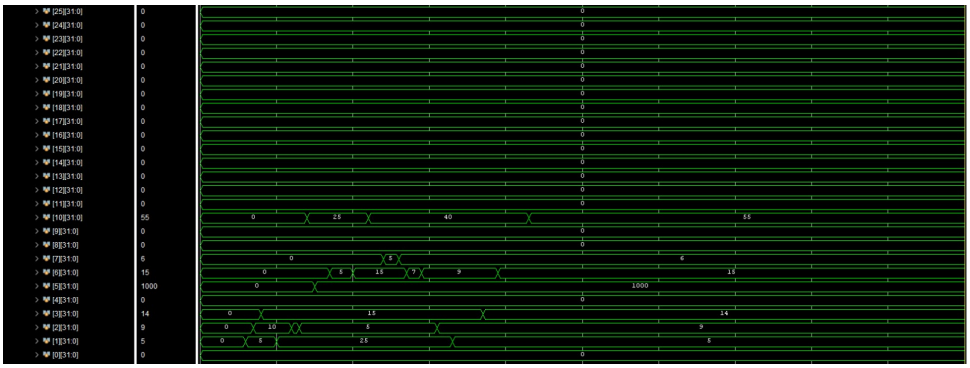

Figure 4: Register File Simulation Showing Data Flow During Execution

Tracer Verification with Cadence Xcelium

A tracer has been implemented to verify the complete functionality of the pipelined RISC-V processor. Using Cadence Xcelium, the tracer recorded the execution of over 98,880 instructions to ensure the correctness of the pipeline, especially under various hazard

conditions, including data hazards and control hazards. The following figure presents the tracer output, demonstrating the real-time instruction flow and the correct behavior of the pipeline during comprehensive testing.

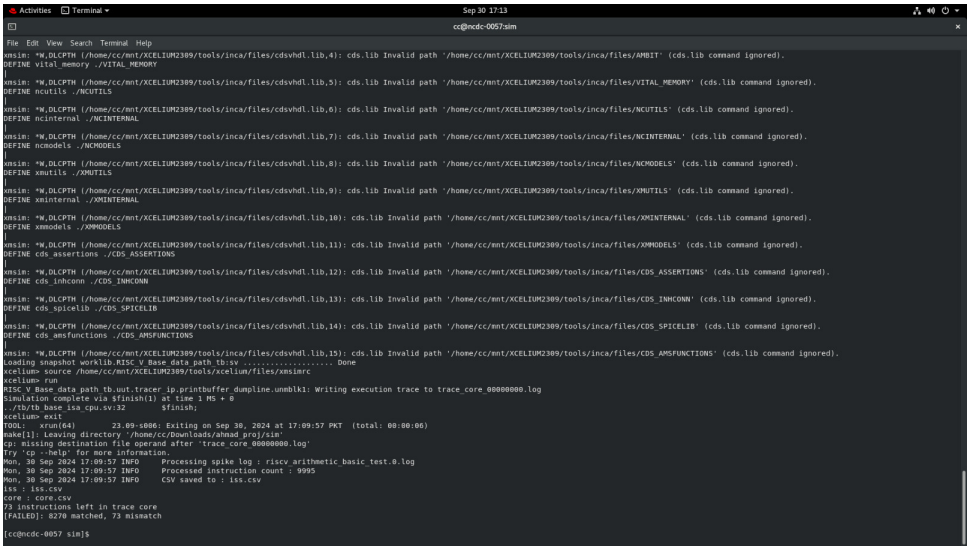

Figure 5: Tracer Output Capturing Full Instruction Execution Flow

Control Path and Hazard Detection

Control hazards, especially those involving branch instructions, are mitigated by implementing additional logic to predict branches and selectively flush instructions. The logic includes branch prediction and flushing mechanisms to prevent incorrect state changes.

The diagram below illustrates the control path responsible for generating signals to manage multiplexer control, ALU operations, memory accesses, and register file writes. This path has been augmented with additional modules, such as forwarding and hazard detection logic, to handle both data and control hazards effectively.

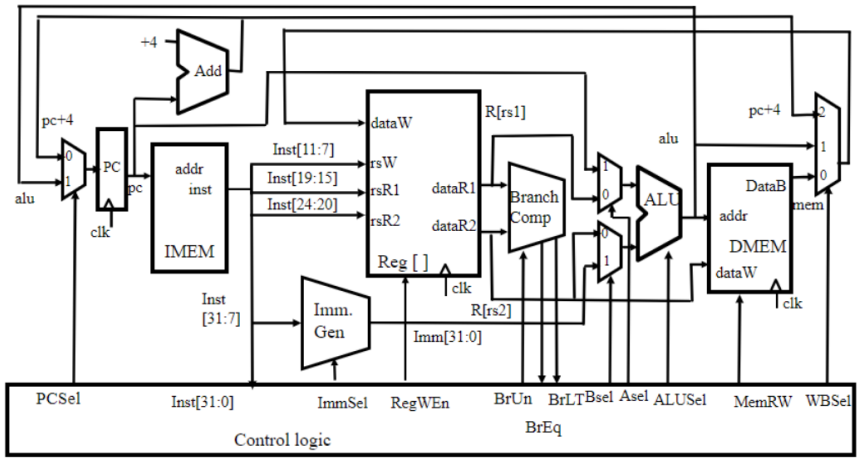

Figure 6: Pipelined Control Path in the Datapath with Hazard Detection Logic

Conclusion

In this lab, a complete pipelined RISC-V processor with a tracer was implemented and verified. Control hazards, primarily caused by delayed branch decisions, have been mitigated through careful hazard detection and control logic. The tracer provided valuable insights into the data flow across the pipeline, confirming the effectiveness of hazard management for a large number of instructions. The correct simulation of the provided assembly program demonstrates the processor's reliable functionality. Future improvements may include more sophisticated branch predictors to further reduce pipeline stalls and flushes.