RISC-V 32-bit Pipelined CPU (Logisim) - Project Report

This report describes the design and implementation of a basic 32-bit RISC-V CPU built using Logisim. The CPU implements a 5-stage pipeline (Instruction Fetch, Instruction Decode/Register Fetch, Execute, Memory, Write Back) and supports core arithmetic, logical, load/store, and move instructions. The project demonstrates a complete datapath including Program Counter (PC), Instruction Register (IR), Decoder, Register File (RF), Arithmetic Logic Unit (ALU), Data Memory, and Writeback logic.

Architecture Overview:
• Data Width: 32 bits for registers, ALU, and memory data.
• Address Width: 16-bit PC and memory addressing.
• Register File: 8 × 32-bit general purpose registers with two read ports and one write port.
• Memory: 16-bit address × 32-bit data RAM used for both instructions and data.
• Pipeline Registers: RA, RB, RY, RZ, RM for stage-by-stage isolation.
• Control Signals: IR_ENABLE, PC_ENABLE, ALU_SELECT, SELECT_RA, SELECT_RB, RY_SELECT, MEM_READ, MEM_WRITE, MUXY_SELECT, MUXMEM_SELECT.
• ALU Operations Supported: ADD, SUB, AND, OR (as labelled in the design).

Module Descriptions:

Instruction Address Generator (IAG): Holds and updates the Program Counter (PC). A 16-bit PC register feeds the memory address input. PC increments under control of PC_ENABLE.

Instruction Decoder (DECODER): Contains the Instruction Register (IR) to latch the current instruction. A splitter extracts OPCODE, source register indices (SELECT_RA, SELECT_RB), destination register index (RY_SELECT), and immediate fields. It also drives control signals (ALU_SELECT, MEM_READ, MEM_WRITE, MUXY_SELECT).

Register File (RF): Contains 8 general purpose registers, each 32 bits wide. Two read ports output RA_OUT and RB_OUT based on SELECT_RA and SELECT_RB. Data is written to the destination register via DATA_IN and SELECT_RO (or RY_SELECT) under control of EN_Rn enables.

ALU: Accepts two 32-bit operands (IN_A, IN_B) and performs ADD, SUB, AND, OR depending on ALU_SELECT. The result is placed on RES and forwarded to the next pipeline stage.

Memory: A single 16-bit address × 32-bit data RAM used for instruction fetch and data load/store. MEM_READ and MEM_WRITE control load and store operations.

Pipeline Registers: RA and RB hold operand values from the register file. RY holds ALU results and control bits into the MEM stage. RM holds data from memory reads. RZ holds data to be written back.

Pipeline Flow (per instruction):
• **IF:** PC from IAG addresses RAM; 32-bit instruction fetched into IR on IR_ENABLE.
• **ID:** DECODER splits IR to generate ALU_SELECT, register selects, immediate. RF outputs RA_OUT, RB_OUT. Values latched into RA and RB pipeline registers.
• **EX:** ALU executes with RA (and RB or Immediate) producing RES. RES latched into RY.
• **MEM:** If load/store, memory is accessed using address in RY. Data read latched into RM. For ALU-only ops, RY forwards result.
• **WB:** Multiplex between ALU result (RY/RZ) or memory data (RM) into DATA_IN, then write to RF destination register indicated by RY_SELECT.

Example Instruction Execution (ADD R2, R3, R4):
1. IF: PC addresses RAM; instruction fetched into IR.
2. ID: DECODER outputs SELECT_RA=3, SELECT_RB=4, RY_SELECT=2. RF outputs R3 and R4 into RA and RB.
3. EX: ALU performs ADD of RA and RB. Result latched into RY.
4. MEM: No memory action for ADD. RY forwards result.
5. WB: DATA_IN = RY result. SELECT_RO=2 enables RF to write result into R2.

Conclusion:
This Logisim implementation demonstrates a complete 5-stage pipelined RISC-V-like CPU with an 8×32 register file, 32-bit ALU, unified 32-bit memory, and basic instruction set. It is suitable as a teaching/learning tool and as a portfolio project to showcase understanding of processor datapath, control signals, and pipelining concepts.