Description of RISC-V

A single cycle processor executes each instruction in one clock cycle. This means that all stages of instruction execution (fetch, decode, execute, memory access, write back) are completed within one clock cycle. This simplicity leads to straightforward control logic but can limit clock speed due to the long critical path.

Components

1. **Program Counter (PC)**:
    - Holds the address of the current instruction.
    - Incremented by 4 (assuming each instruction is 32 bits).
    - Can also be updated for branch instructions.
2. **Instruction Memory**:
    - Stores the program instructions.
    - Outputs the instruction located at the address given by the PC.
3. **Register File**:
    - Contains 32 registers for storing intermediate data.
    - Provides two read ports (rs1 and rs2) and one write port (rd).
    - Supports register write-back for instructions like ADD, LOAD.
4. **ALU (Arithmetic Logic Unit)**:
    - Performs arithmetic and logical operations.
    - Takes two operands (a and b) and a control signal to specify the operation.
    - Outputs the result and a zero flag (for branch operations).
5. **Data Memory**:
    - Used for load and store operations.
    - Takes an address and data to be written or read.
    - Controlled by mem_read and mem_write signals.
6. **Control Unit**:
    - Decodes the instruction opcode to generate control signals for the various components.
    - Generates signals for branch (branch), memory read (mem_read), memory write (mem_write), ALU source (alu_src), register write (reg_write), and whether to use ALU result or memory read data for register write-back (mem_to_reg).

Instruction Execution Flow

1. **Instruction Fetch**:
    - The instruction at the address specified by the PC is fetched from instruction memory.
2. **Instruction Decode and Register Fetch**:
    - The fetched instruction is decoded to determine the operation.
    - The source registers (rs1 and rs2) are read from the register file.

3. **Execute**:
    - The ALU performs the operation specified by the instruction.
    - The operands can be either two registers or a register and an immediate value (for instructions like ADDI).
4. **Memory Access**:
    - For LOAD instructions, data is read from data memory.
    - For STORE instructions, data is written to data memory.
5. **Write Back**:
    - The result from the ALU or data memory is written back to the destination register in the register file.

DATAPATH OF EXECUTION: