Experiment 4

Single Cycle RISC-V Processor

We will design a microarchitecture that executes R-, I-, S- and B-type instructions in a single cycle. A RISC-V instruction normally goes through different phases starting with the instruction fetch phase (Figure 4.1).

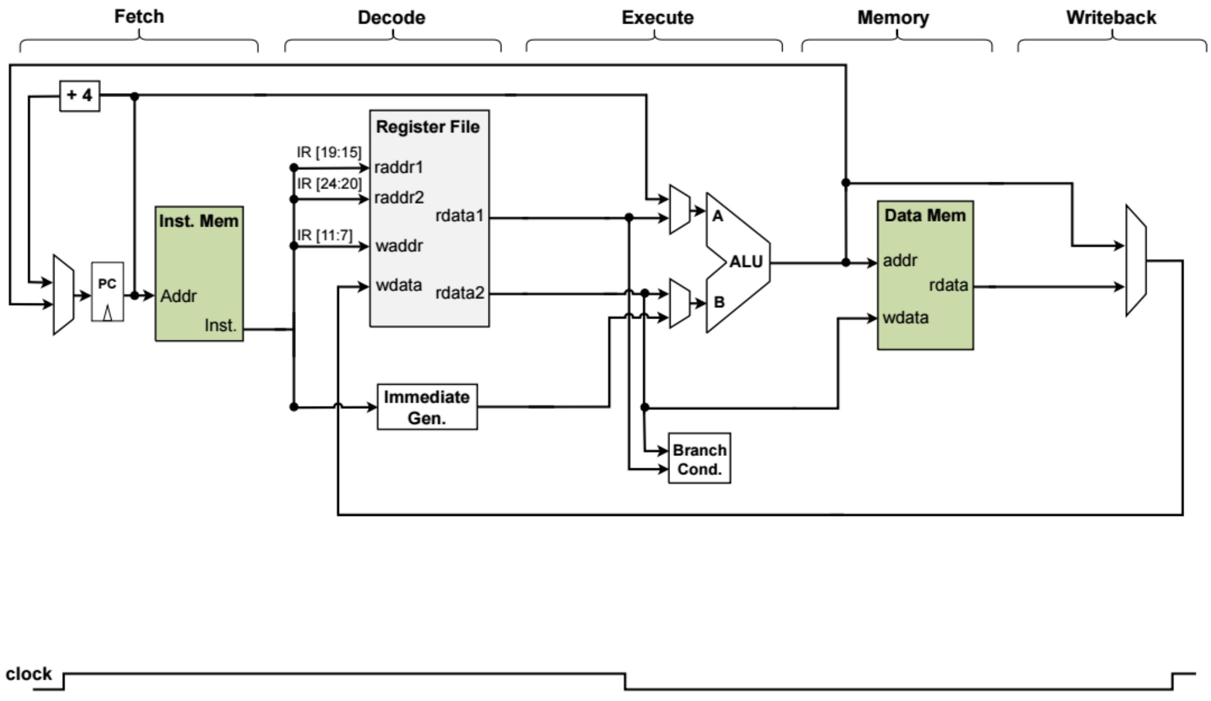

Figure 4.1: RISC-V Single Cycle Datapath for R-,I-,S- and B- type Instructions

Fetch Phase

An instruction is fetched by providing an appropriate address to the instruction memory.

Instruction Memory

Instruction memory can be viewed as a read-only memory buffer with address inputs and data outputs. Since we are following the RV 32I instruction set, all the instructions are encoded using 32-bit machine codes. As a result the data bus width will be 32-bits. The address bus size depends on the size of the memory, however the addresses generated by the processor ALU will be 32-bits.

Decode Phase

The three key operations performed at this stage are:

1. Preparing the immediate values
2. Fetching the operands from the register file

Immediates are sign-extended because RISC-V developers did not observe a benefit to using zero-extension for some immediates as in the MIPS ISA and wanted to keep the ISA as simple as possible. Only shift amounts are zero extended.

Execute Phase

The required operation by the instruction is performed in the execution phase. The first step is the selection of the two operands followed by the implementation of the operation to be performed by the ALU. *Controller controls the mux selection for operands and lets the ALU know which operation needs to be performed by checking* ***funct3 and funct7****.* (Refer to *RISC-V Instruction Set Summary* in *Harris & Harris, Digital Design and Computer Architecture RISC-V.*)

Memory Phase

The load/store operations are performed for data transfer from/to data memory. The decode operation for load and store instructions use func3 bit-field of the instruction machine code to define the size and type of the data variable that is to be exchanged with the data memory. We will only be implementing word size memory accesses for now.
Load instructions are I-type while store operations follow S-type encoding format. The memory address for load/store operations is constructed using register-offset memory addressing mode. The address generation for load/store operations during the instruction execution phase simply uses the ADD operation for register-immediate offset addressing.
Controller provides mem_rd signal for load instructions and mem_wr signal for store to the memory. Can be separate or the same signal.

Write-back Phase

During the write back phase the result of execution is written back to the destination register.
Controller enable reg_wr signal for the register file.

Branch Instructions

Branch instructions can be dealt in two ways. Either a separate module checks branch conditions and ALU generates the new PC. Or ALU provides flags after executing the condition and a separate adder generates the new PC.