Pipelined RISC-V Processor

Milestone 3

CSCE 3301 – Computer Architecture - F'22

Ibrahim Gohar
Anas Ibrahim

The American University in Cairo

Pipelined RISC-V Processor

Milestone 3

by

Ibrahim Gohar
Anas Ibrahim

Student Name	Student ID
Gohar	900203321
Ibrahim	900204611

Faculty: School of Science and Engineering, AUC
Instructor: Dr. Cherif Salama | Fall 2022

Preface

This report is an analysis for an attempt to impelment RISC-V pipelined processer supporting all the RV32I ISA, and takes a deeper look into their hardware implementations.

Ibrahim Gohar
Anas Ibrahim
November 2022

Technical Summary

In the attempt of making a pipelined cycle processer, a new memory was created to handle both the memory of the instructions and the data memory itself. This processer contains two files one for the memory and other for the register file so we can read and write in both of them. We added the appropriate modules to implement the full datapath, and all were used in the top module ”ProcesserMain”. ECALL and FENCE were implemented to jump uncoditionaly to address zero, and EBREAK to halt the program.

Contents

1

Datapath & Schematic Design

we’ve used the simple RISC-V single cycle implementation, adding a few modifications to it in order to support all the instructions, **the Datapath** is shown here:

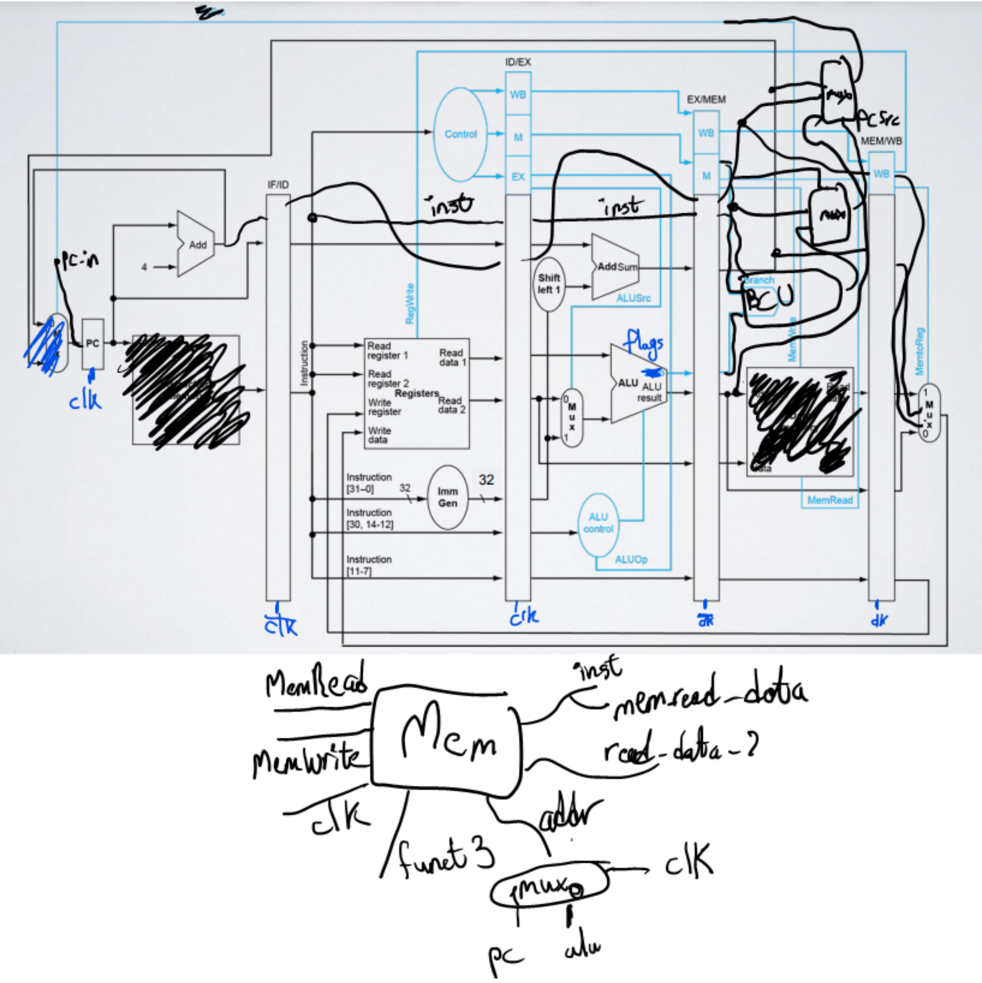

Figure 1.1: The Full Datapath of Pipelined RISC-V 32I

The Schematic Design is shown here:

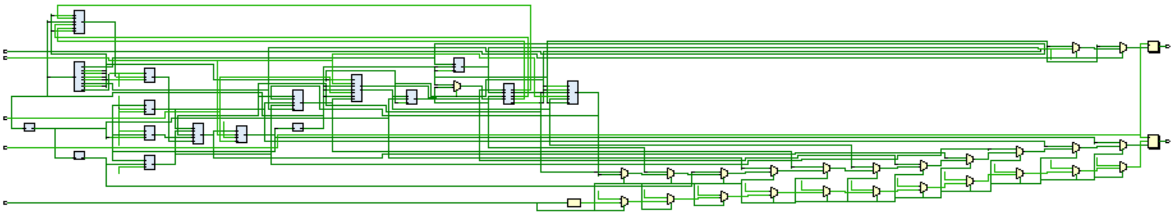

Figure 1.2: The Schematic of Single Cycle RISC-V 32I

.

PC:

- It is the program counter
- Has five options depending on the instruction: added to four, added to the immediate, assigned to the register + the immediate, the same pc value, and zero

Single Memory:

- It has both the data memory and the instructions in it
- It takes a flag that indicates whether the required data is instruction or not. The data is mapped even for data, and odd for the instructions so both can be separated.
- It is considered to be the main memory of the processer and it is byte addressable
- Can write and read data from the register into the memory of the processer

Control Unit:

- It takes the first part of the instruction
- Return different signals and selection lines like ALUOp, ALUSrc, and etc..

Register File:

- It is used read ans write in the registers
- Keep track of all the registers that can be used

Immediate Generator:

- Takes the instruction and fetch the immediate depending on the opcode and both function 3 and 7

ALU Unit & Control Unit:

- applies various operations on the registers depending on the signals sent from the ALU Control Unit

Branch Control Unit:

- It takes the branch signal from Control Unit, Function 3, and some flags to produce the Branch Signal that determine which instruction will be excuted

2

Verilog Files

They are included in the zip file

3

Test Functions & Results

They are is included in the zip file