Computer Architecture
CSCE 3301-01
Fall 2019
MS1 Report
October 28th, 2019
For
Dr. Cherif Salama

Background:

Project: femtoRV32

Author(s): Abdelhakim Badawy - abdelhakimbadawy@aucegypt.edu
Marwan Eid - marwanadel99@aucegypt.edu
Mohammed Abuelwafa - mohammedabuelwafa@aucegypt.edu

Description: Verilog modules that constructs the full single cycle datapath supporting all of RV32I 47 instructions with basic test cases.

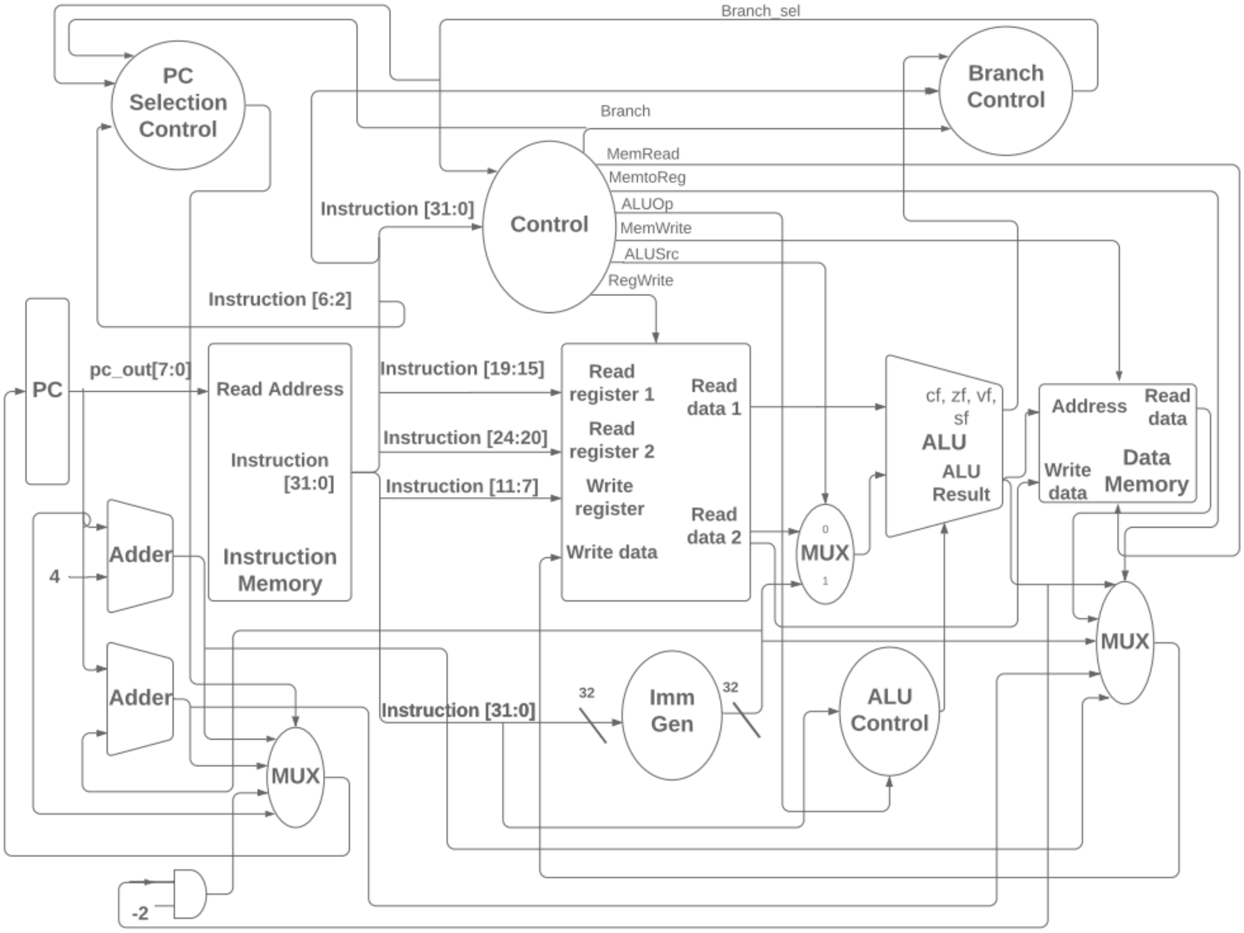
Technical Description:

- We used the Verilog modules we implemented in the lab for the single cycle datapath supporting only seven instructions, modified and added some aspects to support all the 47 instructions.
- We changed the instruction memory and the data memory to be both byte-addressable instead of word-addressable. We changed the size of the memory to be 4 GB.
- We determine the type of the instruction in case of load or store instructions using the func3 bits of the instruction.
- We changed the size of the control signal (MemToReg) to three bits for it to be the selection line of the 8x1 multiplexer whose inputs are the ALU result, the data memory output, the immediate generator output, the output of the adder of the program counter and the immediate generator output, and the output of the adder of the program counter and the hard coded value 4. The output of this multiplexer goes to the register file write data.
- The carry flag, zero flag, overflow flag, and sign flag generated by the ALU, along with the current instruction, are inputs to the branching control unit which in turn generates a signal (pc_sel) which is an input to the control unit; this signal is an input to a module .
- The instruction needed to be implemented as a nop instruction are implemented so by setting all the control signals to 0.
- EBREAK instruction is determined by checking its opcode, func3, and the 20th bit of the instruction and is implemented as a halting instruction by

setting all control signals to 0 except for the branch signal which is set to 1 and is an input to a module that generates the selection line for the multiplexer whose output is the next program counter; additionally, this module has the branch signal and the instruction as inputs. The multiplexer has four inputs: the output of the adder of the program counter and the hard coded value 4, the output of the adder of the program counter and the immediate generator output, the value of the next program counter in case of a JALR instruction, and the same program counter. Accordingly, we can choose the next program counter based on the current instruction, the branch signal, and the pc_sel signal.

- The value of the next program counter in case of a JALR instruction is determined by ANDing the ALU result with -2, (32’b111…110) which thus sets the least significant bit to zero, since jumps are allowed to only even addresses.

Block Diagram:



Schematics:

The schematic is attached in the submission folder.