Preliminary Design Review for the ECE 449 Project

Aman Nijjar – V00865405

Nic – V00878446

University of Victoria

2021-03-04

Project Summary

The objective is to design and implement a pipelined 16-bit CPU on a Basys3 FPGA board. The goal is to complete the A, B, and L format instructions outlined in the instruction set by the end of the semester. [1] This will be done by completing the A-format by March 4th, and the B-format by March 18th, and the L-format by April 15th. The demonstration of this CPU will be on April 8th or April 12th. The following sections discuss the ISA specification, block-level decomposition, and specification of the data and control paths.

Instruction-Set Architecture Specification

The CPU follows the instruction set document provided by Dr. Nikitas Dimopoulos, however a few custom design decisions have been made.

For the multiplication operation, the CPU produces a 16-bit result, although it takes 2 16-bit inputs. This instruction is implemented with a Booth multiplier. [2] By design, this algorithm produces a 32-bit result, however, we made modifications to the algorithm to output a 16-bit result and an overflow signal in the case the result is too large.

Each register will have an overflow bit allocated to them, which will be stored internally in the ALU as an 8-bit status register. The TEST instruction has been modified so that this register can be read by the user.

The CPU does not support external interrupts as well as the PUSH/POP instructions for the stack buffer, however, this will be reassessed as the progress proceeds.

Block-Level Decomposition

Appendix A is the high-level block diagram of this system. When the processor first boots up, it will need to load the user's program. This is done by setting "RST & LD" to logical high and entering the first 16 instructions, which is when the bios stored in the ROM is activated because the master controller would execute the branch instruction stored at 0x0002. The Bios will store the input into the RAM and then this process is repeated for as long as "RST & LD' is held high.

The program executes when the "RST & EXEC" signal is high by having the CPU branch to the location stored in address 0x0000. The BIOS will fetch the instructions from the memory location there and increment the PC by 2 to get to the next location.

Data and Control Path Specification

The CPU uses 1 master controller, supervising 4 sub-controllers: Decode, Execute, Memory-Access, and Write-back. Moreover, there are 4 latches to help pass along critical information to the next stage.

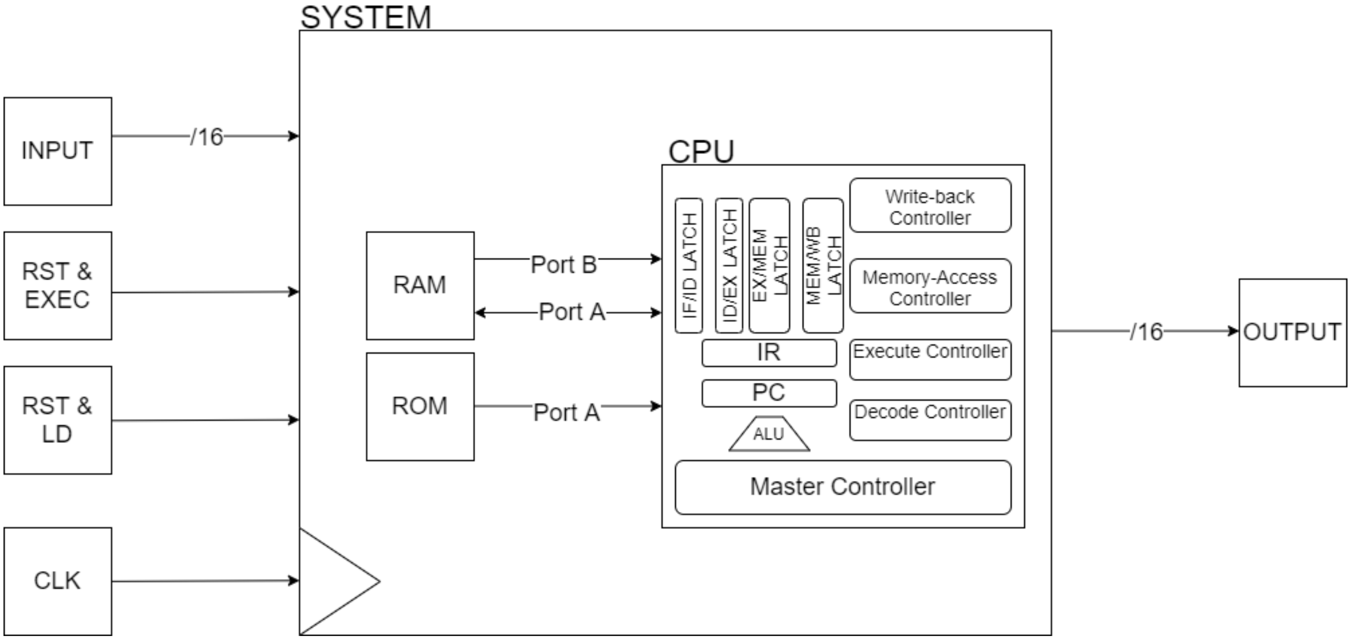
Each controller oversees each stage of the pipeline. For the Decode controller, it analyzes the OP-code from the IF/ID latch and passes forward the ALU mode. For the Execute Controller, it looks at the destination register of the current address and checks if the last instruction used the same register, if so, it predicts that a RAW data hazard will occur and will re-use the ALU's last output. For the Memory-Access controller, it will forward the overflow bits directly to MEM/WB so that they are not stored in the RAM. Finally, the Write-back controller will determine if the instruction outputs the data or store the data locally in a register.

References

[1] N. Dimopoulos, "Instruction set architecture." UVIC, Victoria. https://ece.engr.uvic.ca/~ece449/lab/lab2019/Instruction_Set_16bit_2020.pdf

[2] (Jain & Jaiswal, *Implementation of Booths Algorithm i.e. multiplication of Two 16-bit Signed numbers using VHDL and concept of pipelining* 2016), International Research Journal of Engineering and Technology (IRJET), vol. 3, no. 6, pp.1-5, June 2016

Appendix A – High-level Block Diagram of System


SYSTEM
INPUT
/16
RST &
EXEC
RST &
LD
CLK
RAM
ROM
Port B
Port A
Port A
CPU
IF/ID LATCH
ID/EX LATCH
EX/MEM
LATCH
MEM/WB
LATCH
Write-back
Controller
Memory-Access
Controller
Execute Controller
Decode Controller
IR
PC
ALU
Master Controller
/16
OUTPUT