**THIS IS THE FORMAT OF HOW I AM TYPING IT OUT**

**PORT NAME**

**PORT SIZE**

**PORT TYPE**

**DESCRIPTION**

PC

CLK

1

input

This is one of the two user inputs for the processor. It is used set b the clock portion of the pc and reg memory, when clk is on positive edge the pc increments and the red memory reads.

RST

1

input

This is the other one of the two user inputs for the processor. This is used to reset the PC to 0 when RST is assigned to 1. It also initializes the PC to 0.

Pout

6

output

This is the output/result of the PC. It can count up to 63 in binary, 111111.

Instruction\_mem

count

6

input

This is an input from the output of the PC. Pout -> count. This is used to access the next memory address in the instruction memory.

instruction

32

output

This is an output that comes from the 32 bit binary code that is hard coded in this sv file. The 32 bit binary code is chosen based on the count number (0-63).

Controller

Instruction

32

input

This is the same 32 bit binary code from the instruction memory.

ri

1

output

This is the most significant of the 32 bit instruction memory binary code. This value determines if the processor uses R-type(0) or I-type(1) instructions.

rs

6

output

This is bits [30:25] of the 32 bit instruction memory binary code. This is one of the two source registers used in the instructions.

rd

6

output

This is bits [24:19] of the 32 bit instruction memory binary code. This is the destination register used in the instructions. This register is where memory gets loaded into, ADD **rd** <- rs + rt

mode

1

output

This is bit 18 of the 32 bit instruction memory binary code. This is used to determine whether the ALU computes arithmetic or logic operations.

F\_Code

3

output

This is bit [17:15] of the 32 bit instruction memory binary code. This is used to tell the ALU which function to use.

rt

6

output

This is bits [14:9] of the 32 bit instruction memory binary code. This is one of the two source registers used in the instructions. This register is only used if instruction is R-type.

imm

15

output

This is bits [14:0] of the 32 bit instruction memory binary code. This is used when I-type instruction is selected. This is just a value in signed 15 bits.

RegWrite

1

output

This is used to tell the processor to load memory into a register. This value becomes 1 if the corresponding F\_Code is selected.

MemWrite

1

output

This is used to tell the processor to store data into memory. This value becomes 1 if the corresponding F\_Code is selected.

**THIS WILL BE WHERE REG MEM IS BUT I DONT HAVE THE FILE RN**

SignEXT

A

15

input

This is the 15 bit imm value taken from the instruction memory. We are simply extending it to 32 bits.

EXT

32

output

This is the result of the SignEXT module. It is the sign extended 32 bit imm value.

**THIS WILL BE WHERE THE 2TO1 MUX IS BUT I DONT HAVE THE FILE RN**

**THIS WILL BE WHERE THE 32 ALU IS BUT IDK IF WE ARE PUTTING IT IN HERE?**

Data\_mem

MemWrite

1

input

This is either a 1 or 0 depending on what the Controller assigned. If the corresponding F\_Code matched the if statement in the Controller then this value is 1 and Data\_mem will run.

address

6

input

This is the 6 bit register address that comes from the rd value assigned in the Controller. This is where the processor stores data to.

WriteData

31

input

**ANDREW NEEDS TO WRITE THIS PART CAUSE IDK**

ReadData

1

output

**ANDREW WHAT IS THIS**