ECE 474 Final Project

Rand Black

June 8, 2024

1 Introduction

This document will showcase the functionality of the microprocessor outlined for the final project of ECE 474. Below is the block diagram of the entire system. Each one of those blocks will be briefly described and it's functionality will be proven with the results of each module's testbench.

2 ALU

2.1 Design

The desired functionality of this ALU is outlined in the two tables below. The simplified ALU supports four basic operations of 8-bit operands: NAND, NOR, XOR, and ADD. These operations are done in combinational logic and output to the out pin.

Pin List

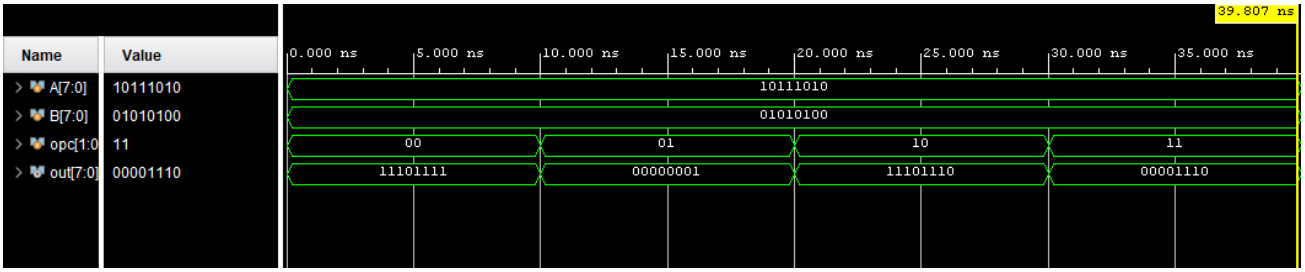
Name	I/O	Width	Description
A	input	8-bit	Operand 1
B	input	8-bit	Operand 2
opc	input	2-bit	operational code
out	output	8-bit	operation output

Operational Codes

Opcode	Operation
00	NAND
01	NOR
10	XOR
11	ADD

2.2 Testbench

The figure included below is the waveform of the ALU testbench. The cases have been outlined in the table. There are four tested cases. One test case for each of the operational codes. The input operands remains constant through the cases.



Test Cases
Operands for all cases: A = 10111010 and B = 01010100

Case	OP code	Expected Output	Actual Output
1	00	11101111	11101111
2	01	00000001	00000001
3	10	11101110	11101110
4	11	00001110	00001110

As seen in the table the expected outputs are equal to the actual outputs in the waveform. This means that the test is successful and the ALU is operational.

3 SRU

3.1 Design

The SRU provides four shifting operations: ROTL, SLA, SRA, and SRL. The associated operational code are and module pins are outlined below. Based on combinational logic, it operates without a clock signal.

Pin List

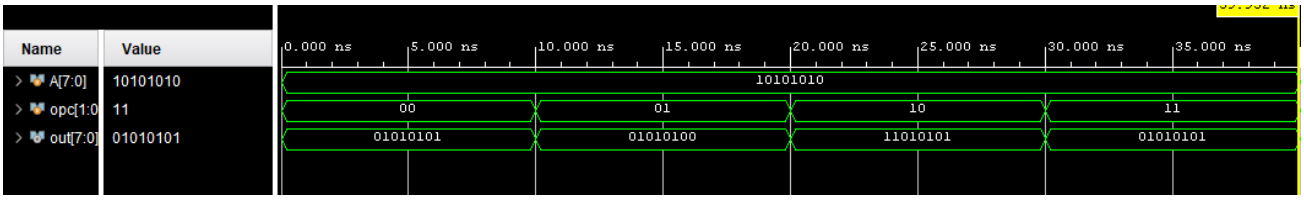
Name	I/O	Width	Description
A	signed input	8-bit	Shifting Operand
opc	input	2-bit	operational code
out	output	8-bit	operation output

Operational Codes

Opcode	Operation
00	ROTL
01	SLA
10	SRA
11	SRL

3.2 Testbench

The figure included below is the waveform of the SRU testbench. The cases have been outlined in the table. There are four tested cases, one test case for each of the operational codes. The input operand remains constant through the cases.



Test Cases
Operands for all cases: A = 10101010

Case	OP code	Expected Output	Actual Output
1	00	01010101	01010101
2	01	01010100	01010100
3	10	11010101	11010101
4	11	01010101	01010101

As seen in the table the expected outputs are equal to the actual outputs in the waveform. This means that the test is successful and the SRU is operational.

4 IF Module

4.1 Design

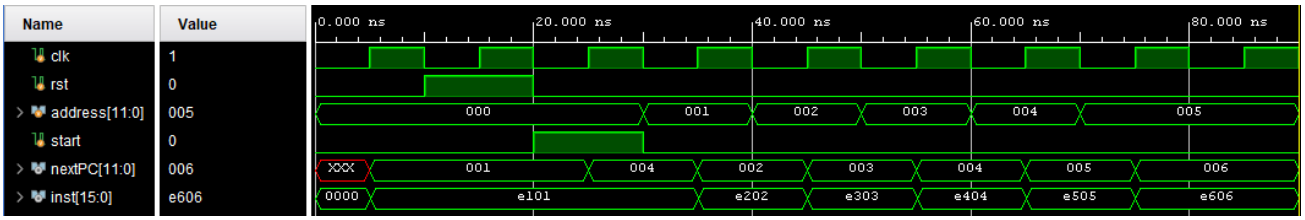
The IF module takes both increments the PC and contains a single port BROM that contains the system's instructions. The current address is attained by a wire input of the module.

Pin List

Name	I/O	Width	Description
clk	input	1-bit	clock signal
rst	input	1-bit	reset
address	input	12-bit	The current PC
start	input	1-bit	A start signal that sets current PC to 3
nextPC	output	8-bit	The current PC incremented
inst	output	8-bit	The instruction from the BROM that is passed to the EX module

4.2 Testbench

The testbench inputs a series of addresses that to test both the incrementing the PC and the instruction fetch.



The image shows the simulation of the IF. Every input address is incremented and output to the nextPC register and the corresponding instruction is output to inst.

5 EX Module

5.1 Design

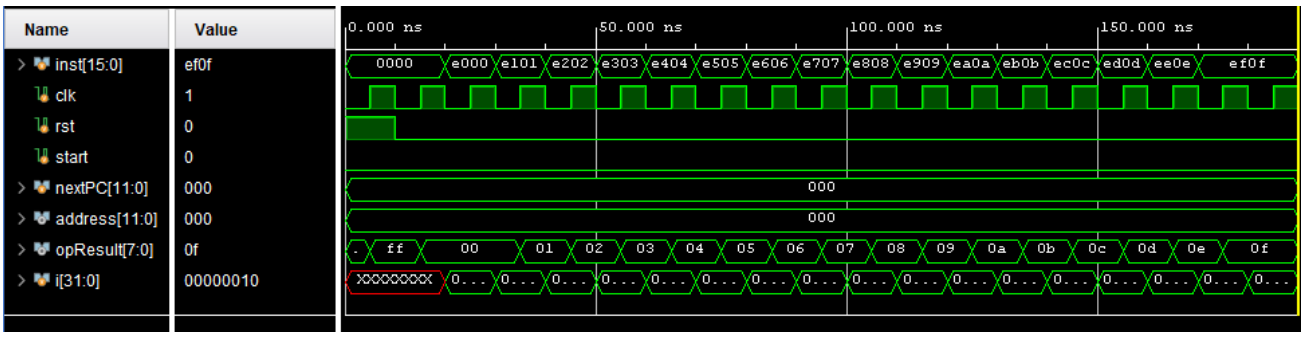
This module deals with all the processing of the instruction output by the IF module. This module also deals with deciding the next address for the PC.

Pin List

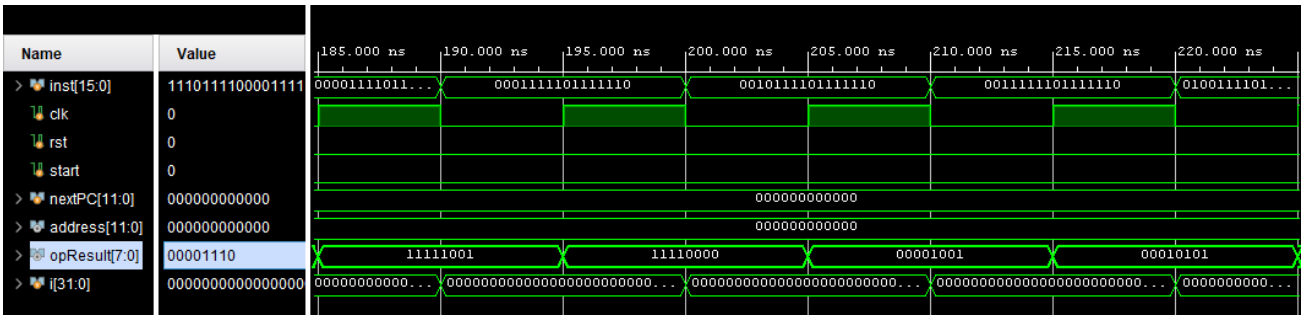
Name	I/O	Width	Description
inst	input	16-bit	The 16-bit instruction being loaded from BRAM
clk	input	1-bit	Clock signal
rst	input	1-bit	Reset
start	input	1-bit	Start signal
nextPC	input	12-bit	The next PC
address	output	12-bit	Address
opResult	output	8-bit	Operation result

5.2 Testbench

This module is very large so the test bench is separated into multiple testbench outputs for different aspects of the module's functionality. The first testbench below shows the LDI function filling all of the register file's registers.



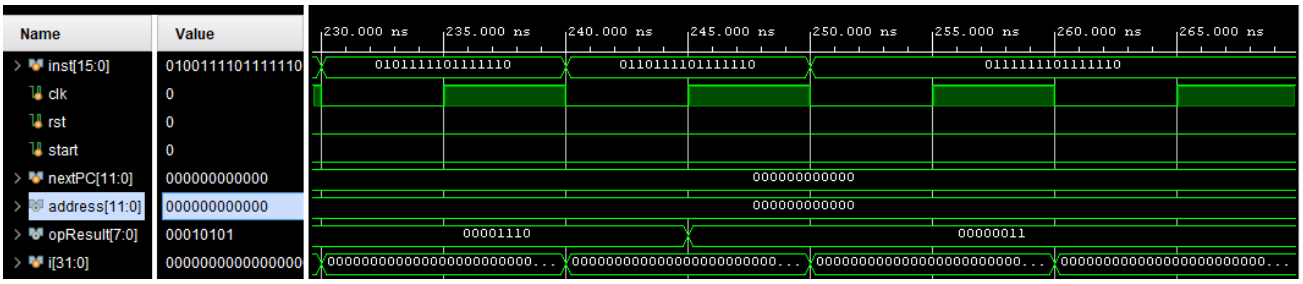
The next portion of the testbench shows the operations of the ALU below is the waveform and the table showing the inputs.



ALU

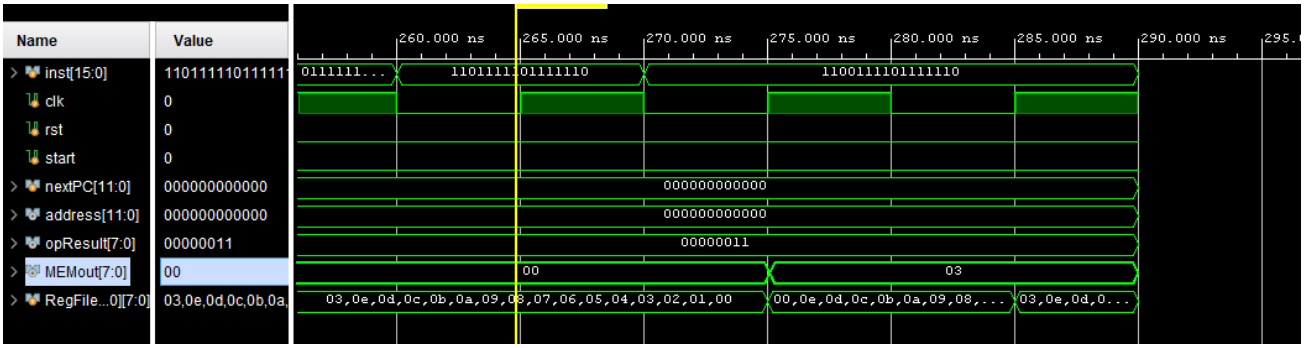
Case	OP code	Expected Output	Actual Output
1	00	11111001	11111001
2	01	11110000	11110000
3	10	00001001	00001001
4	11	00010101	00010101

This is a test of the SRU.



SRU

Case	OP code	Expected Output	Actual Output
1	00	00001110	00001110
2	01	00000011	00000011
3	10	00000011	00000011
4	11	00000011	00000011



This last simulation write the value of 3 in RegFile[15] to MEM[15] and reads it back out. The EX module is fully functional.

6 TOP Module

6.1 Desgin

The design of the top module was just the connection of the IF and EX modules. Only wires were put in between them. All registers were internal to each of the lower modules.

6.2 Testbench

The Top module was mostly functional. Sadly the memory write was working correctly. I believe it was a timing error, but I wasn't able to locate the error. other than that all other functionality appeared to be working correctly.