Musa ÇIBIK

Mustafa SARAÇ

COMP 303

12.01.2017

PROJECT REPORT

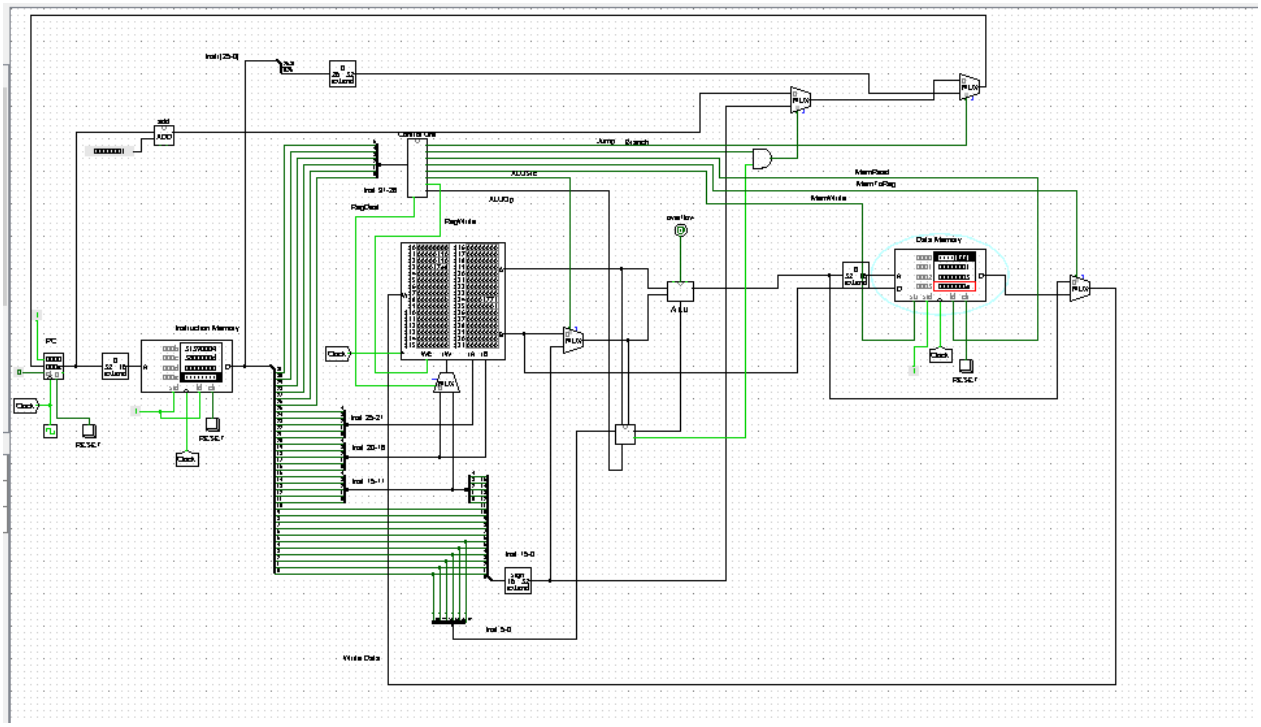

Figure 1: The Circuit on Logisim

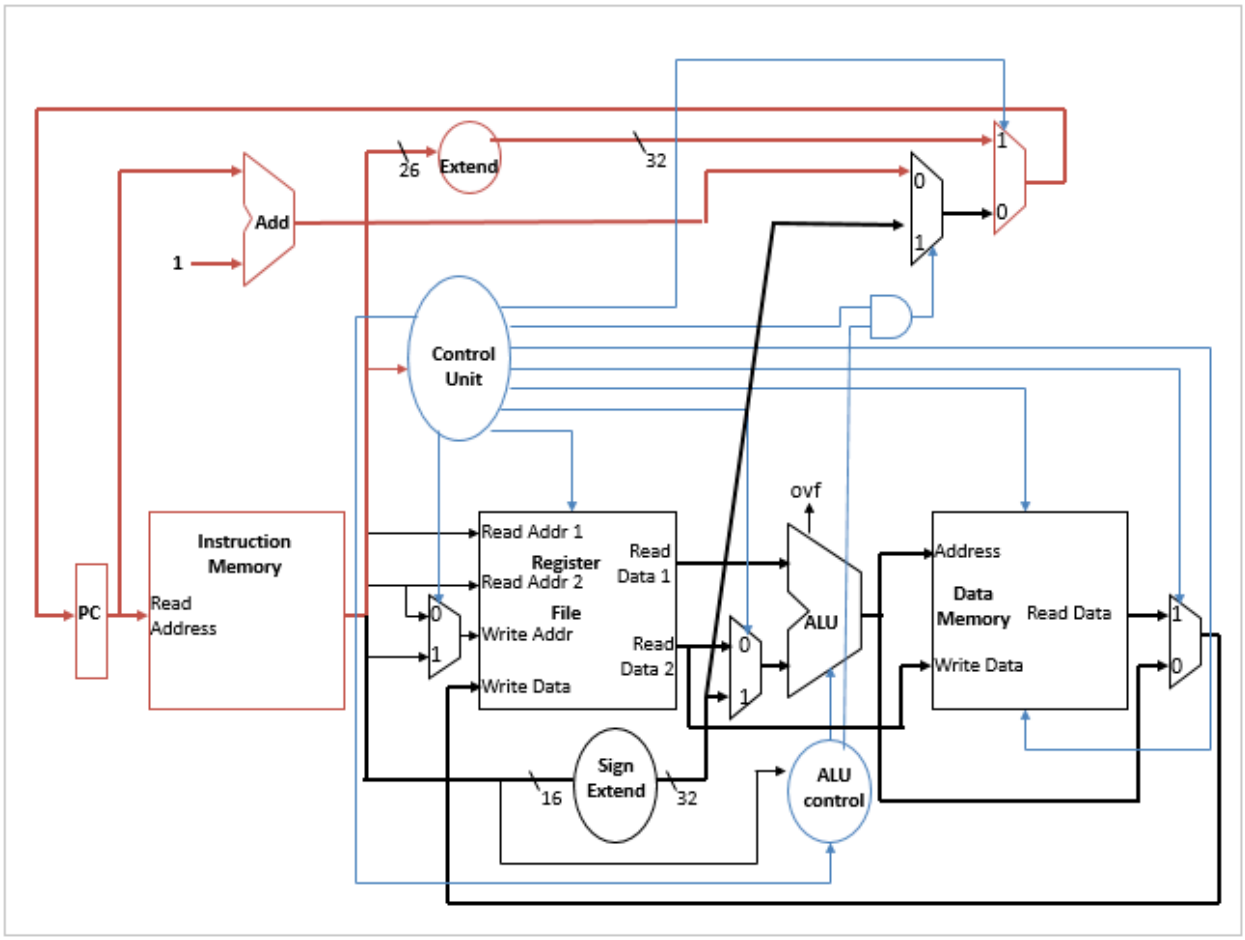

Figure 2: Circuit Schematic of Our Design

During the design process of the single-cycle processor, we followed the same strategy to design our circuit as we learned in the lectures. We modified the given circuit in the class for a single-cycle processor. We changed the circuit's structure, as we needed. The program counter is updated in every clock cycle and appropriate instruction is fetched for the instruction memory. The instruction is processed in control unit and the output for the given inputs is generated and if needed written in the data memory. We also can use the data memory to get the values to perform our operations. Control unit and the ALU control are the key elements of our circuit as they help us to work on the instruction. 3 types of the instructions (R – I - J) are processed correctly thanks to control unit.

Instruction	Opcode	Type	RegDst	RegWrite	ALUSrc	MemWrite	MemToReg	MemRead	Branch	Jump	ALUOp
Sub	000000	R	1	1	0	0	0	0	0	0	000000
Add	000001	R	1	1	0	0	0	0	0	0	000001
Mult	000010	R	1	1	0	0	0	0	0	0	000010
And	000011	R	1	1	0	0	0	0	0	0	000011
Or	000100	R	1	1	0	0	0	0	0	0	000100
Xor	000101	R	1	1	0	0	0	0	0	0	000101
Sll	000110	R	1	1	0	0	0	0	0	0	000110
Slt	000111	R	1	1	0	0	0	0	0	0	000111
Sra	001000	R	1	1	0	0	0	0	0	0	001000
Srl	001001	R	1	1	0	0	0	0	0	0	001001
Lw	001010	I	0	1	1	0	1	1	0	0	001010
Sw	001011	I	0	1	1	1	1	0	0	0	001011
Beq	001100	I	0	0	0	0	0	0	1	0	001100
Bne	001101	I	0	0	0	0	0	0	1	0	001101
J	001110	J	0	0	0	0	0	0	0	1	001110
myIns	001111										001111

Figure 3: The Table Summarizing Control Signals

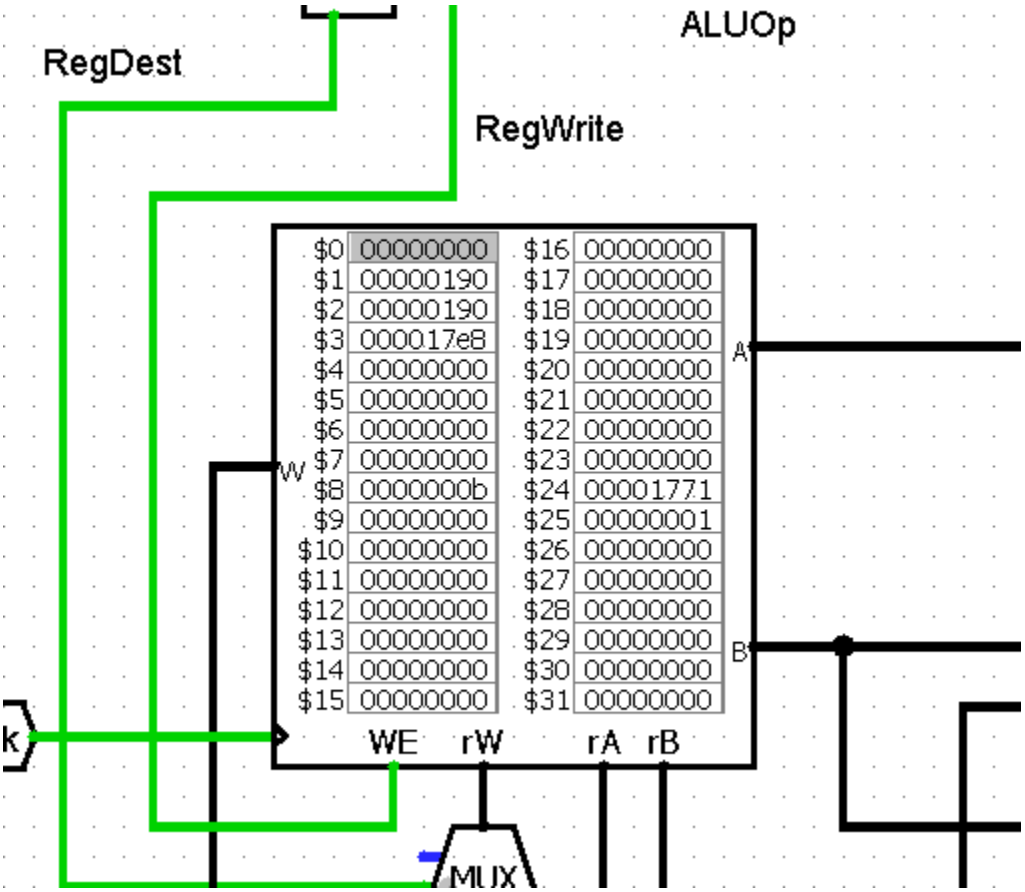

Figure 4: The Final Values in Register

In order to test whether our design works in a correct way, we began with implementing the assembly version of the given pseudo code. Then, according to this assembly code, we generated the instruction machine codes to give them as an input to the Logisim. In addition to these machine codes, we gave the data inputs, which we wanted to work on during the test, to the Logisim. On the first try, we had some errors, so we tried to find the reason that why the processor did not work properly. Then, we realized that when generating the machine codes of the instructions in assembly implementation, we made some mistakes and we actually got a wrong machine code that does something different from the one that we would like to do. Thus, we fixed them and tested the single-cycle processor again. As a result of our test, we expected the value on the register 3 as 6120 in decimal. As you can see the value on the register 3 is $(17e8)_{16}$ which is equal to 6120 in decimal. The other test results and intermediate values were also correct as we traced the instructions and the values on the register, program memory and data memory on the Logisim.