

Miguel Guevara – 4968505 – U01

Control Unit, Registers, Single Cycle Processor

Dr. Atoussa Tehrani

Ms. Sudarat Tangnimitchok

April 18, 2016

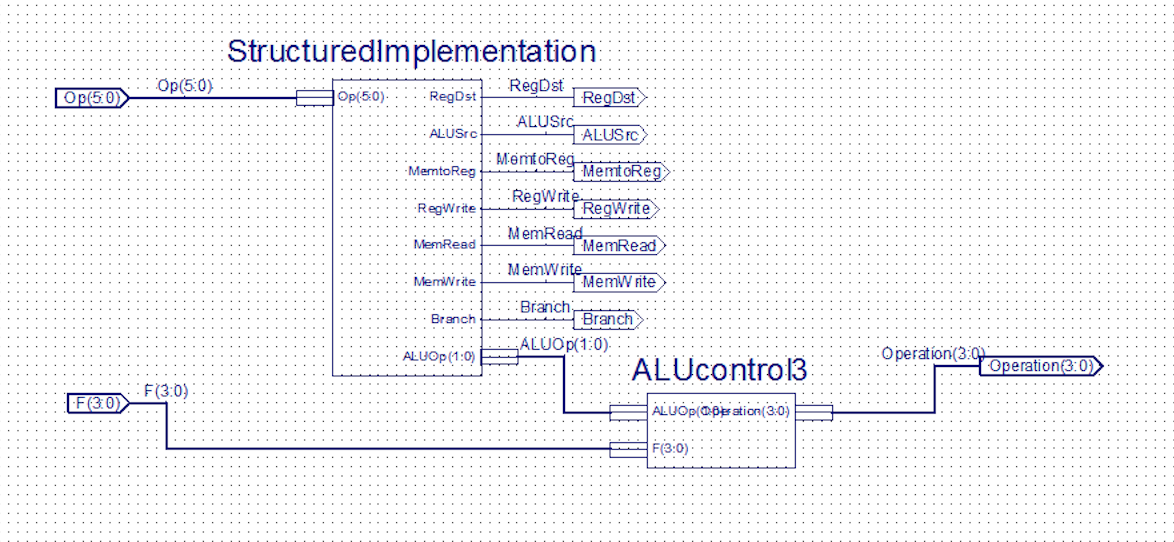
Lab 8 – 10

**Introduction:**

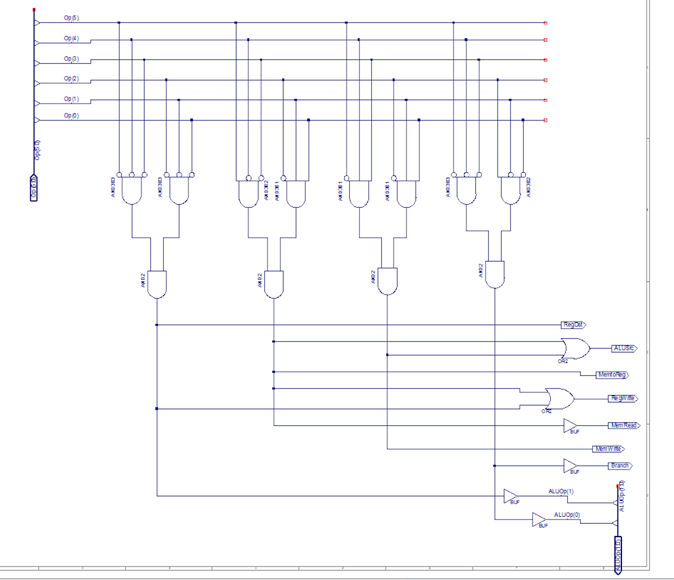
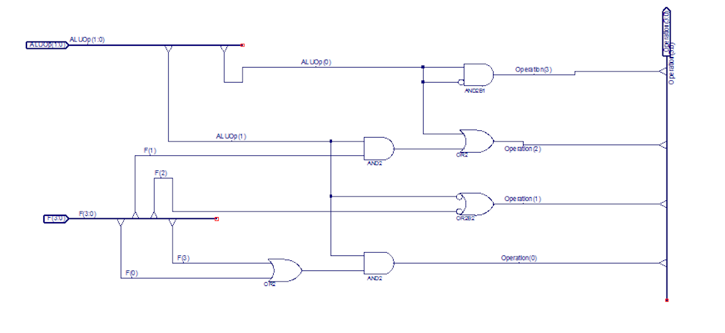
In our journey to build a single cycle processor, we have gone from creating a arithmetic logic unit to register modules. In these labs we have build this registers, as well as the control unit, and that concludes our simple tools to design and implement out single cycle processor.

**Design and Code:**

First, the ALU controller consists of two modules shown in figure 1.1, the structured implementation and the alu control. The SI (Structured implementation) module, receives operation instructions of 6-bits. The 6-bit operation controls whether the system writes into memory, or writes back to registers, to branching and so forth. In this system the the operations are register destination, ALUSrc, Memory to register, register write, memory read, memory write, and branch. The last is the ALU Operation that plays a pivot point with the function to determine what operation to perform in the ALU. Figure 1.2 and Figure 1.3 shows the schematic of these two modules that are included in our ALU control unit. To determine which Operation to conduct when the ALU is working on a given task. Table 1.1 shows what the output of these combinations are and that determines the output of the ALU control unit and the input of the ALU to operate on.

****

**Figure 1.1 – ALU control**

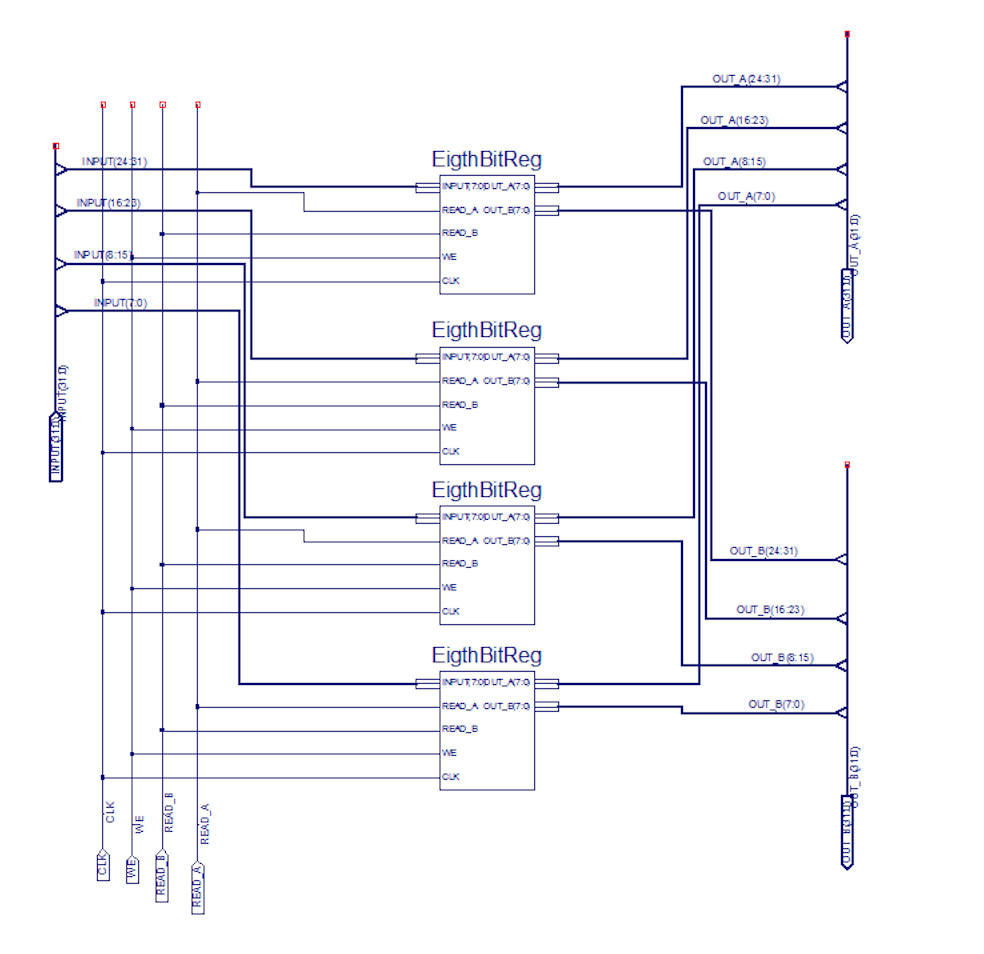
****

**Figure 1.2 Structured Figure 1.3 ALU control**

****

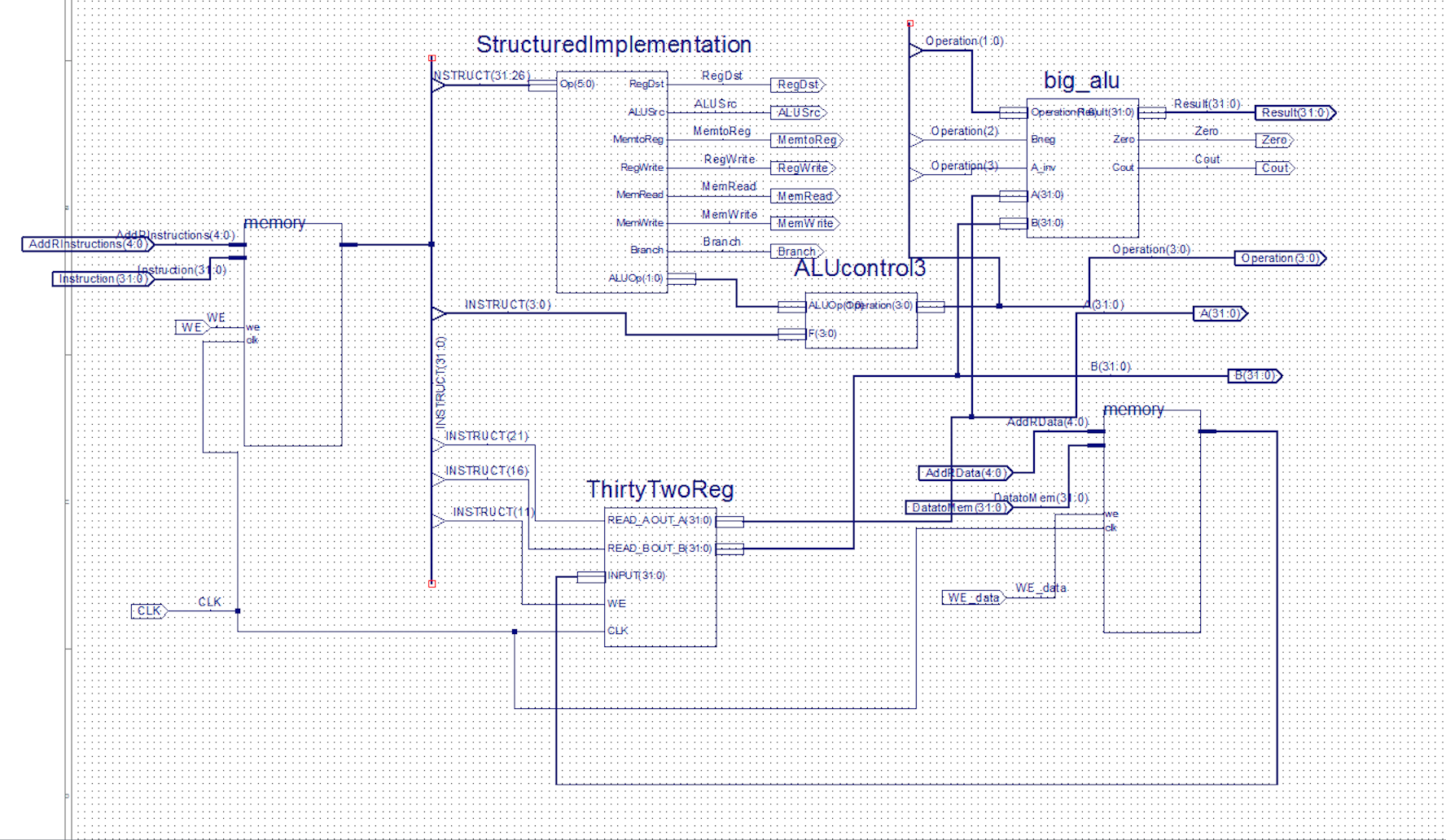
**Table 1.1 – ALUOP in combination with FUNCT**

Second, the 32-bit register is the volatile memory that is inside the processor. In this case, we have build a 32-bit register with only one register for simplicity but it should be obvious to see how this functions. Figure 2.1 shows the schematic of this particular deign.



**Figure 2.1 – 32-bit Register module**

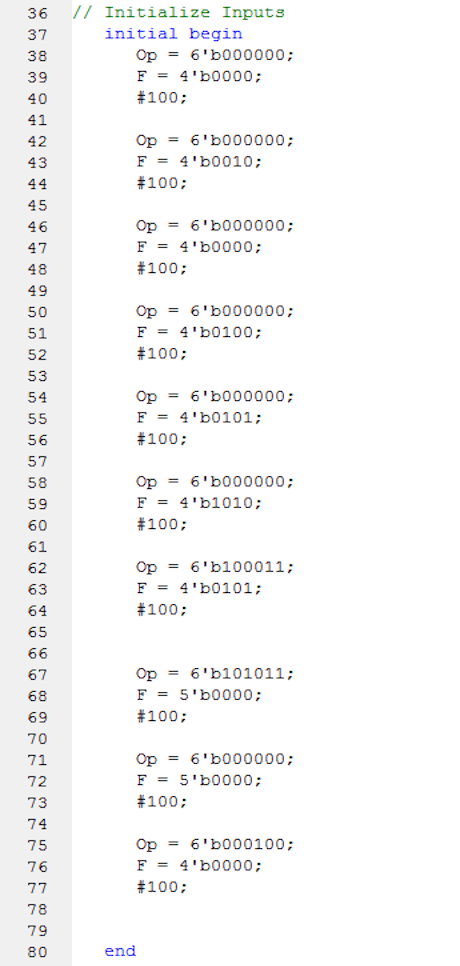
Last we get into the final design of this single cycle processor. After all the different modules created, we can finally combine them to create a single cycle processor. This processor only uses a 32-bit register and the ALU can add, subtract, AND, OR operations, compare values for branching. The functions are very strong since are the fundamental functions of any processor. We also created in this past labs the ALU controller and the ALU was build a few labs ago. Figure 3.1 shows our 32-bit single cycle processor.

****

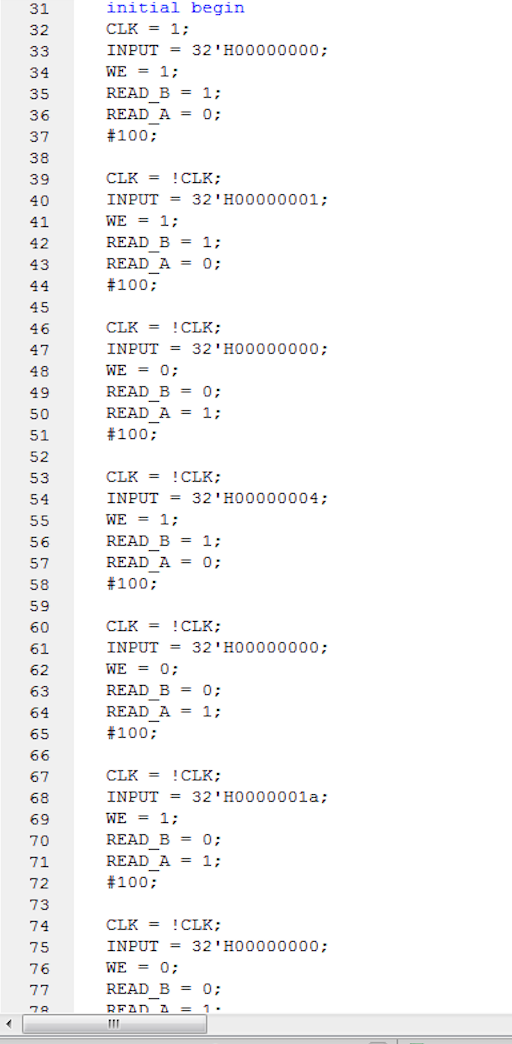
**Figure 3.1 – 32-bit single cycle processor**

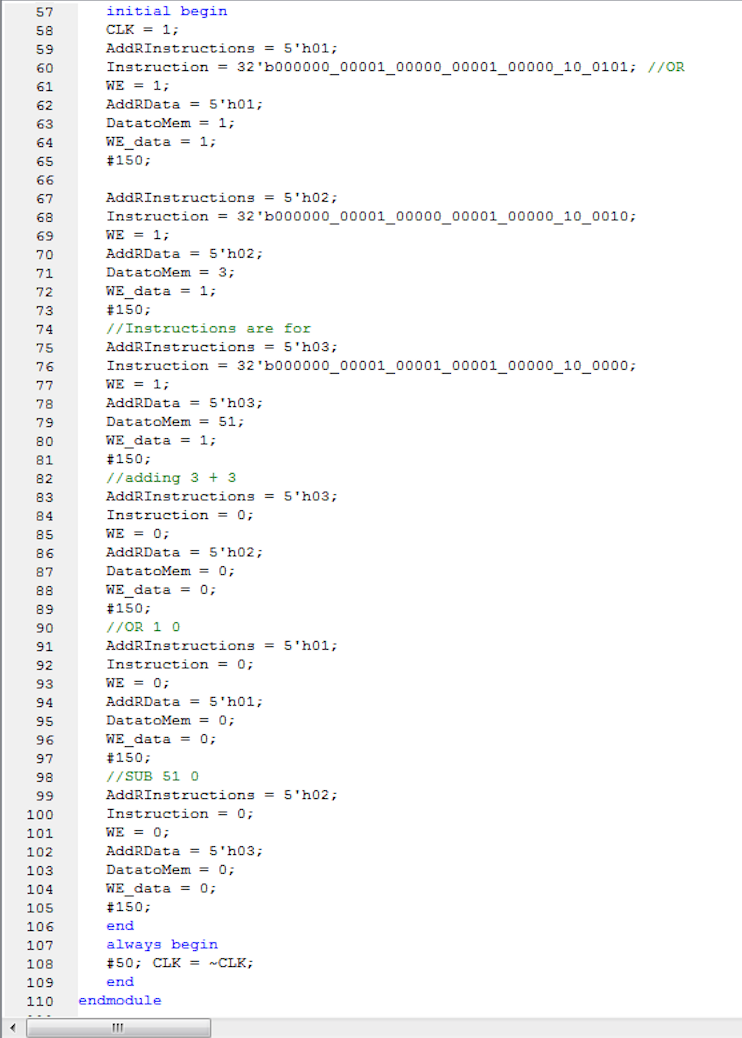
**Results:**

Now, we focus out discussion on the Verilog code that controls all these different modules in running time. To begin, Figure 4.1 shows the inputs of the register modules, it works at the falling edge and therefore, if the Write input is “1” we input value into register and we can read either the value inside the register or the value “0” if reading the B output. Figure 4.2 shows the different combinations of the operation and the function input and later will analyses the result. Then, Figure 4.3 shows the inputs every 150 milliseconds to the processor. We start by adding instructions to instruction memory. The first instruction added is addition instruction, and also added to main memory is the value 1. Then we continue these format for two blocks more, we added or instructions to instruction memory and to main the value “3”. Last, subtraction to instruction memory with value to main of “51”. Then, we go about to read the values in the instruction memory according to where the instructions are and WE is “0” meaning we’ll be reading from them and reading to memory as well. These will give us outputs shown later.



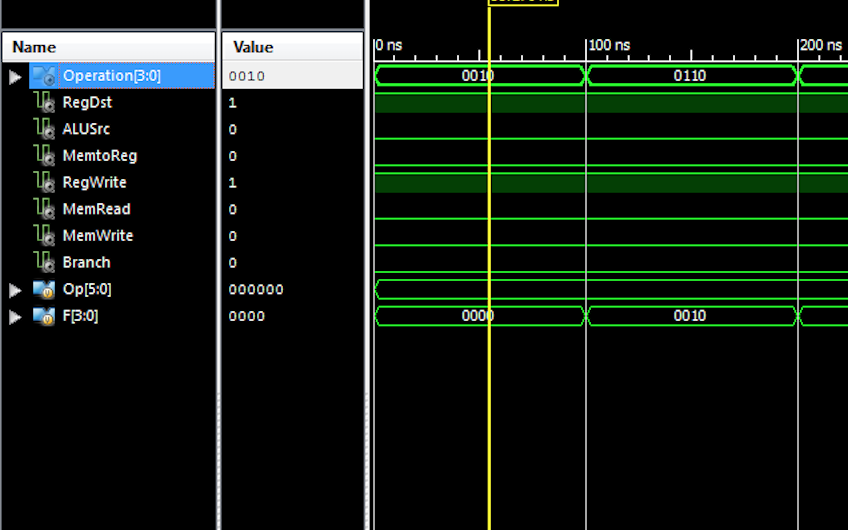
**Figure 4.1 Register Verilog Figure 4.2 Control unit Verilog**



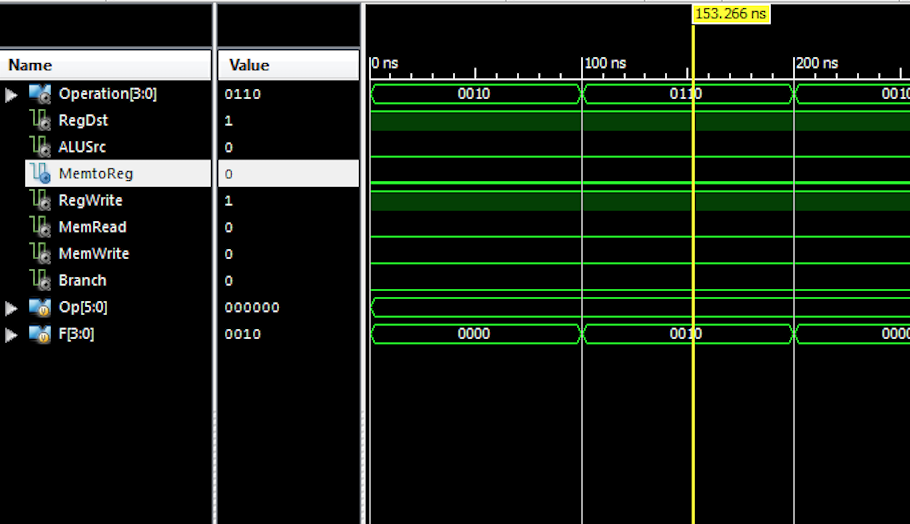


**Figure 4.3 Single Cycle processor Verilog code**

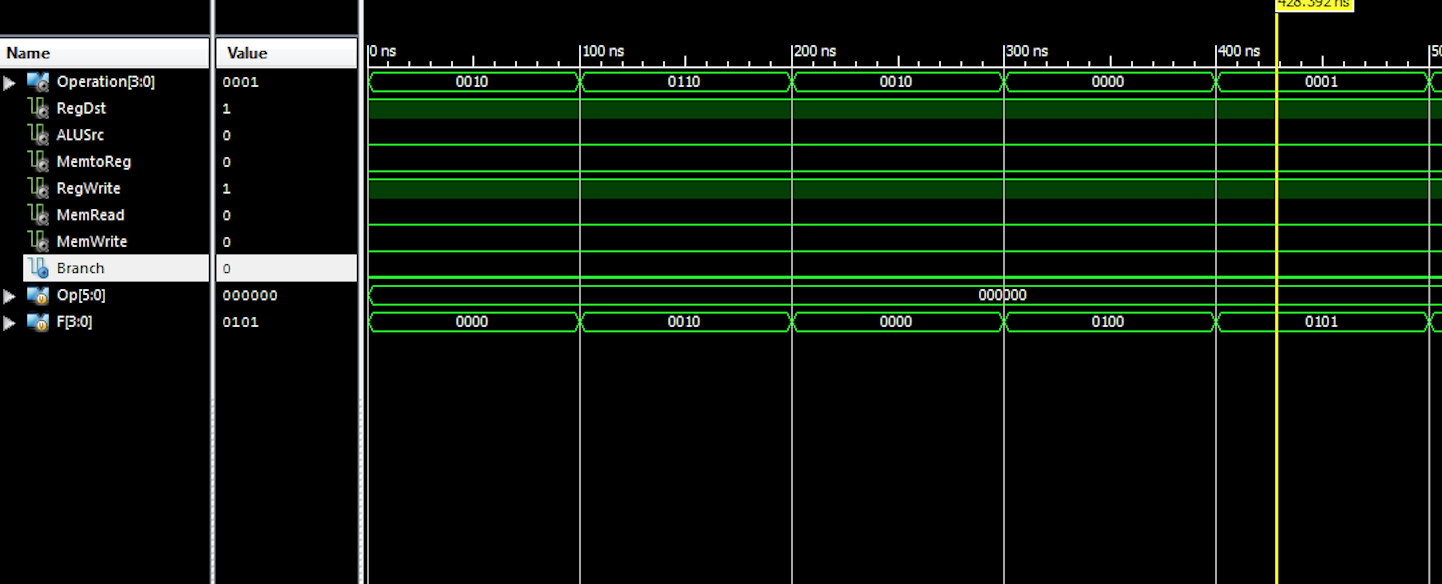
We now look at out Test bench to check out the results. We begin at the ALU control, Figure 5.1 – Figure 5.3 shows the different functionalities of the operations these numbers will be read at the Operation block which is in the last 6-bits of an R-type operation. We show the addition, subtraction and or operations going to the ALU for processing. Then we have the Register modules outputs. These outputs can be appreciated looking at Figures 6.1 to Figure 6.4. We start by writing to register the number “1” and then reading the value on output A. Then to conclude, we write value “4” and then read the value on output A. Finally, the operations of the Single Cycle Processor are somewhat complex, but since we have study and analyze every single component of this system, then our study is simplify to only worry about what the inputs are and what the outputs are. First we take a look at Figure 7.1 to Figure 7.3. This reveals the addition or writing into the two memory that we included in our design. The first is the instruction memories which we will write instructions in every single block of these figures. The first instruction we are writing are the addition of register 1 with the same register. So whatever register one holds it will be added or multiply by two if you want to be a little more technical. In the same block we are also adding a number into Data memory, the number is “1”. We continue this model in the next two blocks and we write subtraction of register A with register B which it is “0”, and write number to Data memory “3”. Also write or function of register A and register B with a value to Data memory of “3”. After these three operations, we use out instructions and out values in the Data memory to read and get an output. First we get the value of the memory with the addition instructions in the first part and add this values to get 6 as an output. Then, Or function with the number in memory which is “1” and OR the value with register 0 which is “0”. Last, Subtract the value in memory which is 51 (this was added in the third operation when we added 51 to memory location 3) to register 0 to get an output value of 51.

****

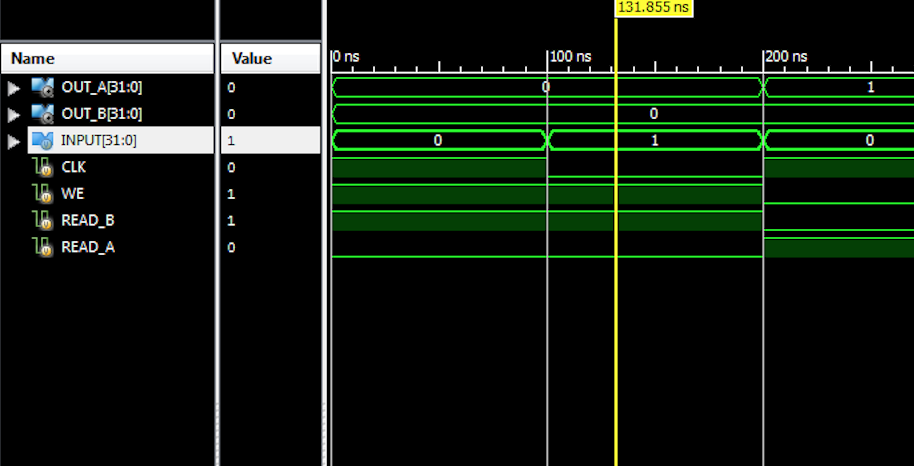
**Figure 5.1 – Control: Addition**

****

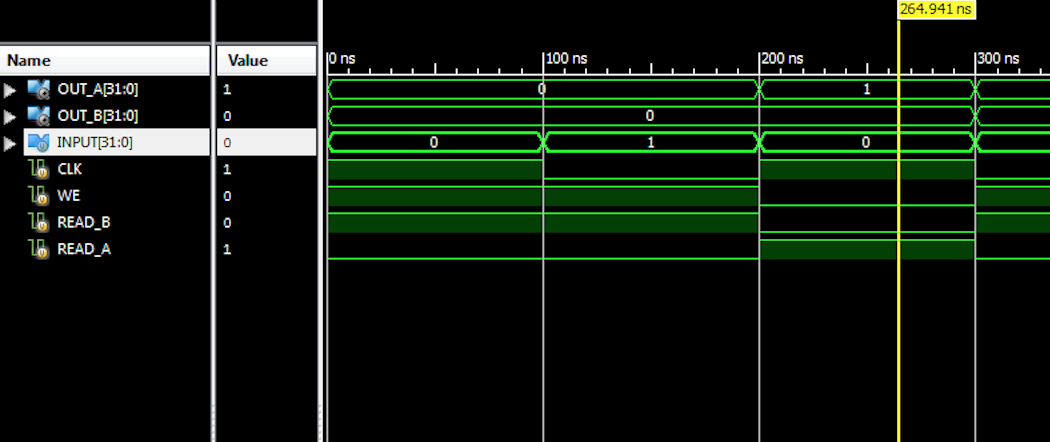
**Figure 5.2 – Control: Subtraction**

****

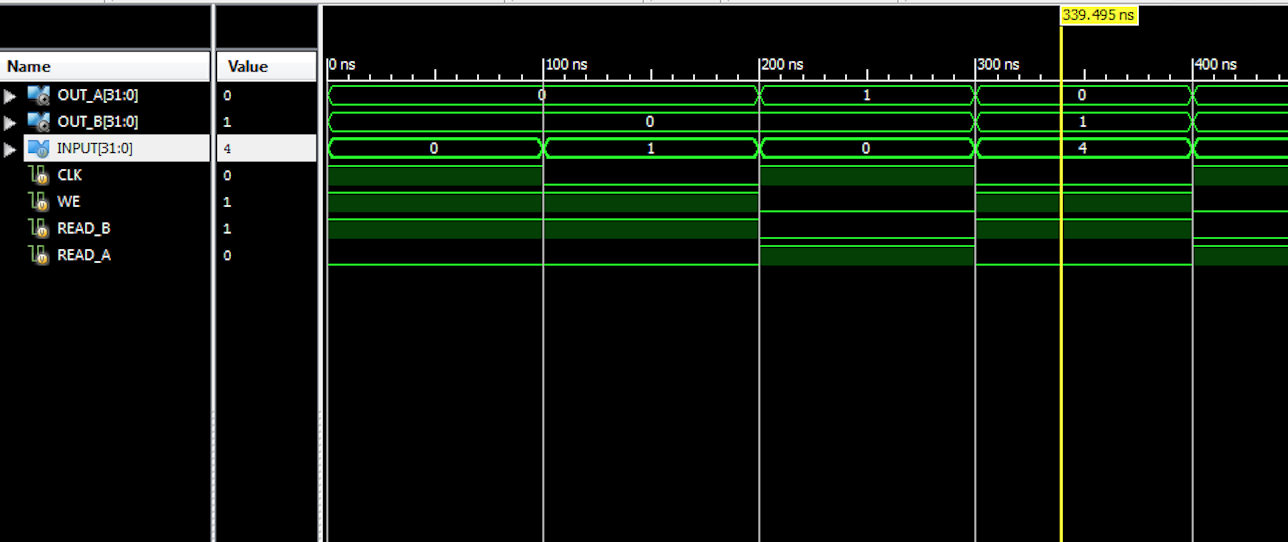
**Figure 5.3 Control: Or**

****

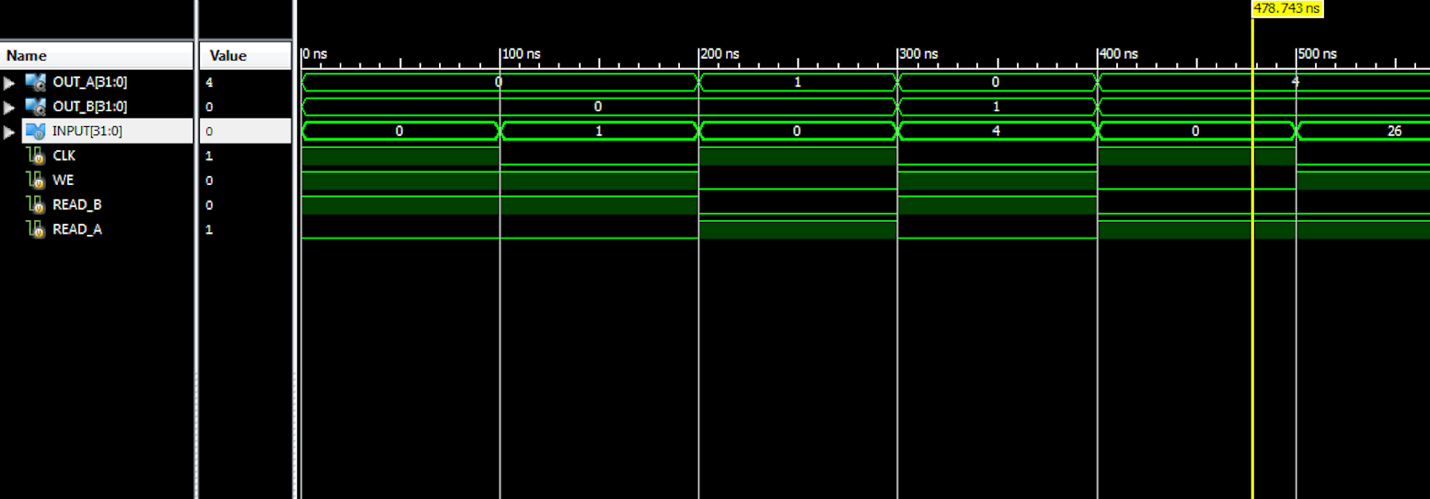
**Figure 6.1 – Register: Write “1”**

****

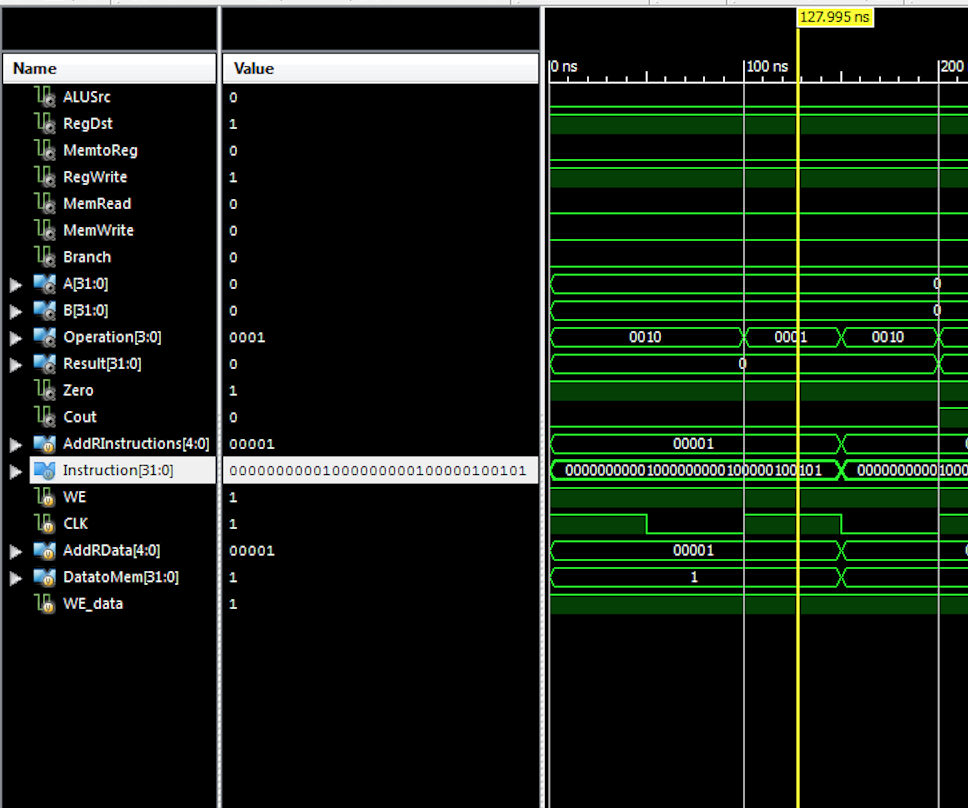
**Figure 6.2 – Register: read “1”**

****

**Figure 6.3 – Register: Write “4”**

****

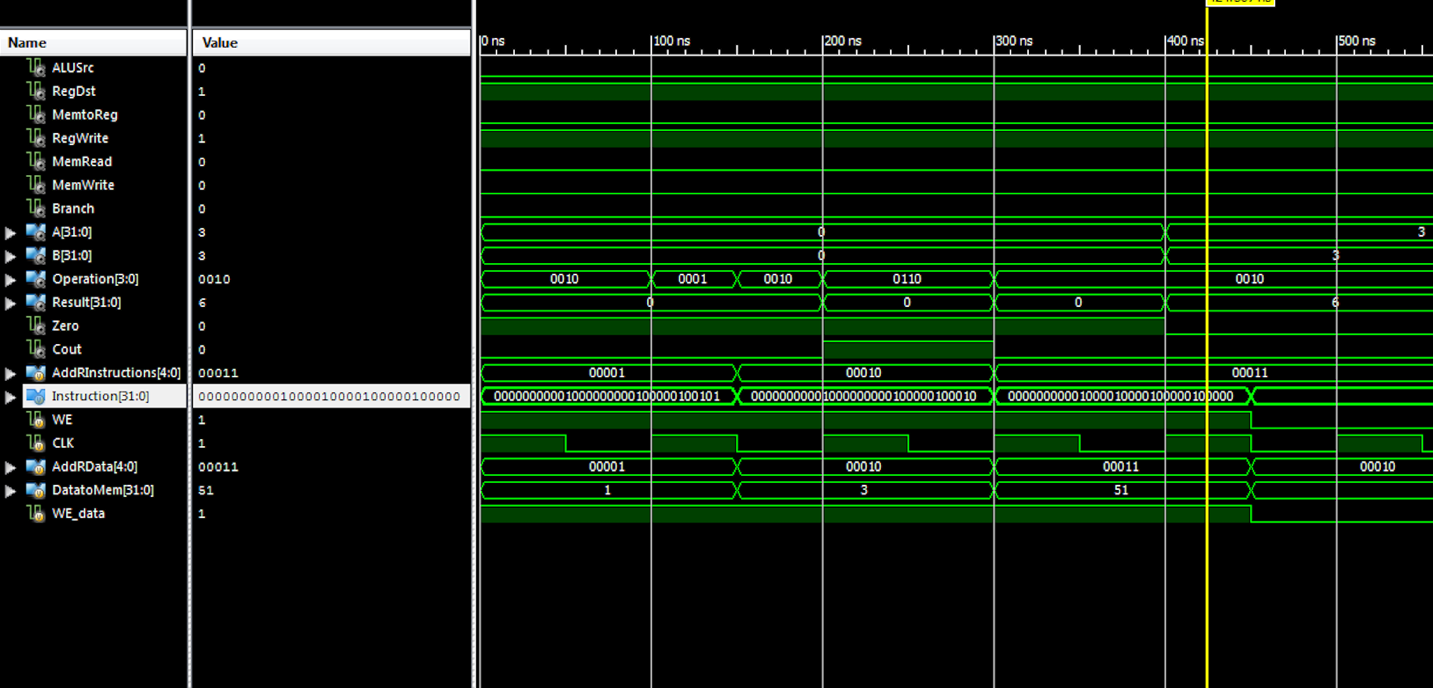
**Figure 6.4 – Register: Read “4”**

****

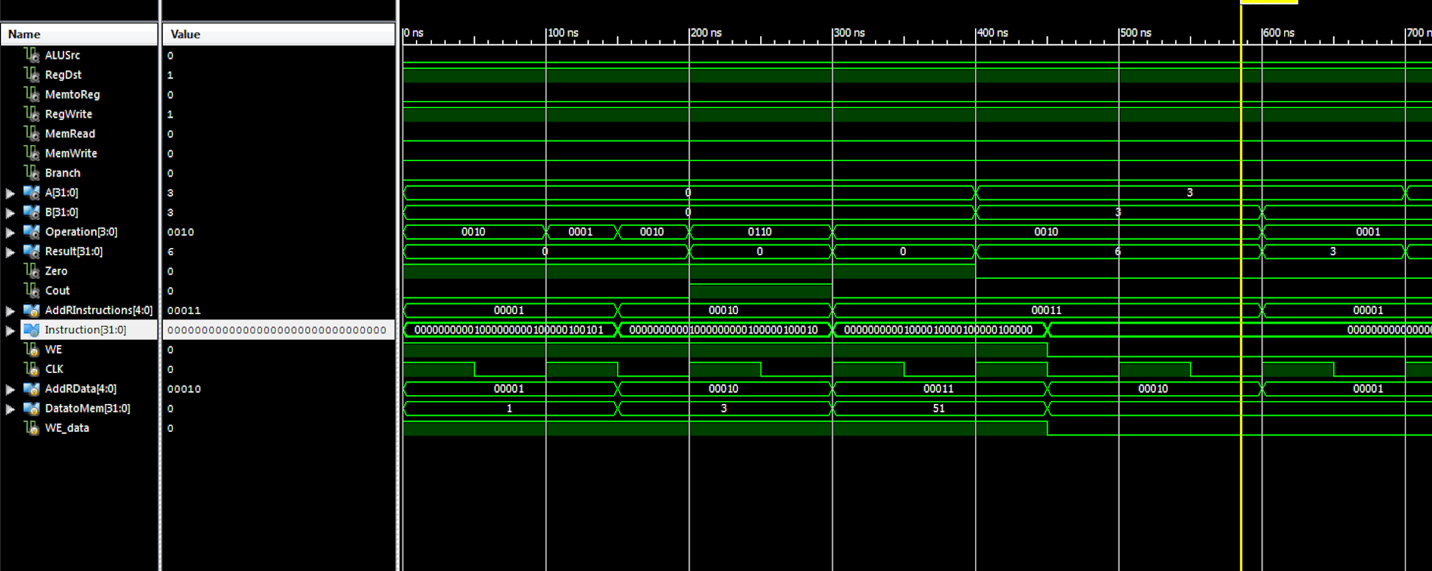
**Figure 7.1 – SCP: Writing to Instruction mem and data mem**

****

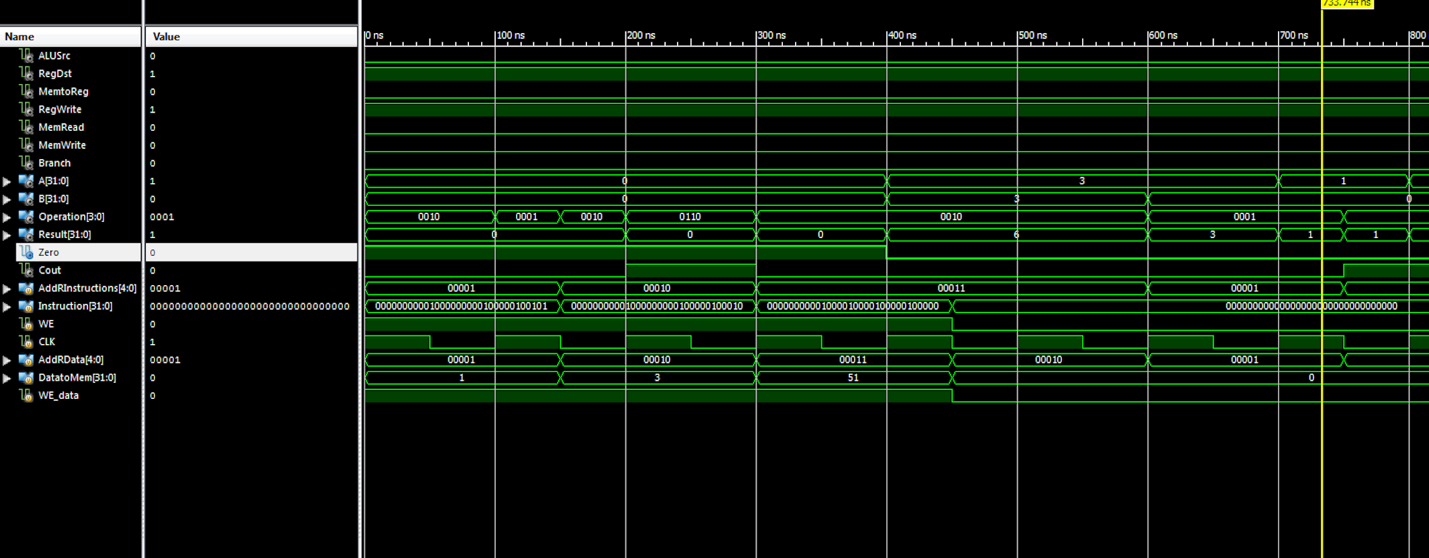
**Figure 7.2 – SCP: Writing to Instruction mem and data mem**

****

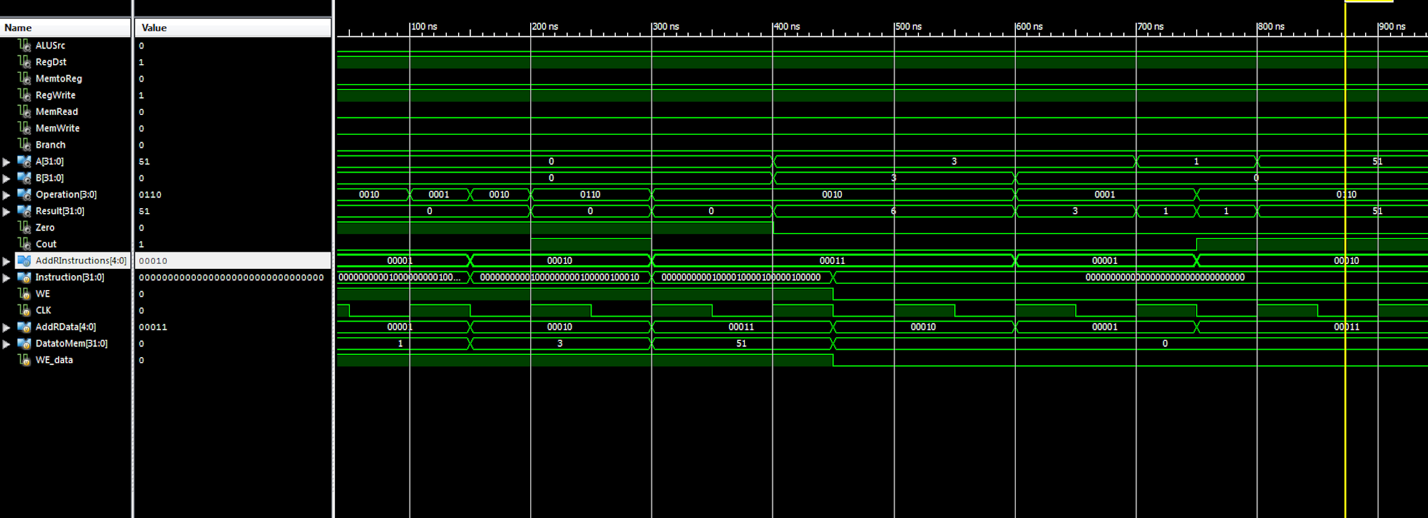
**Figure 7.3 – SCP: Writing to Instruction mem and data mem**

****

**Figure - 7.4 SCP: Adding operation**

****

**Figure – 7.5 SCP: OR Operation**

****

**Figure – 7.6: SCP: Subtraction Operation**

**Conclusion:**

As read in the words and seen in the figures, the single cycle processor has been created using the tools we have worked on from lab 1. These tools or modules includes a 32-bit ALU, registers, ALU control, and we added two memories to save instructions and data in a Harvard model. Our ALU controller controls the operations perform by the ALU, Registers hold date that needs to be read instantly in nanoseconds. This 32-bit single cycle processor is very powerful because with its invention came the facility to make a complex parallel, multi-cycle processing unit which is what the modern processor use in out current time.