THE UNIVERSITY OF KANSAS

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EECS 645 – Computer Architecture

Spring 2023

Homework 09 (Single-Cycle MIPS)

Student Name: Student ID:

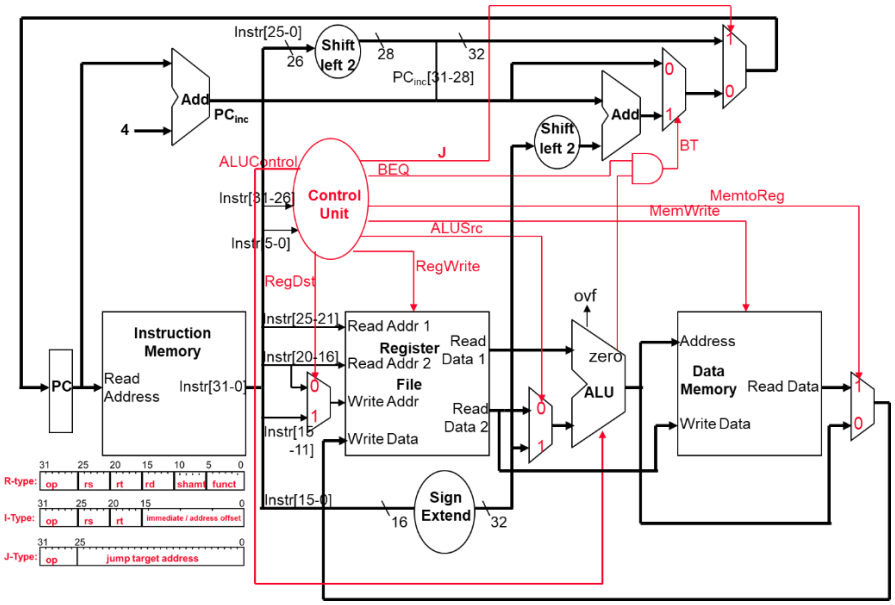
Single-Cycle MIPS

Describe in structural and behavioral VHDL the Single-Cycle (non-pipelined) version of the MIPS processor that supports the following subset, i.e., 11 instructions, of MIPS ISA:

a) 7 Arithmetic/Logical instructions: *add, sub, and, or, nor, slt, addi*
b) 2 Memory reference: *lw, sw*
c) 2 Control transfer: *beq, j*

The microarchitecture datapath and control path are shown in Figure 1. Your memory address translation/mapping should follow the convention shown in Figure 2. The processor has the following interface:

- Inputs
  - Clock (clk → 1 bit)
  - Asynchronous reset for processor initialization and for mimicking program load (rst → 1 bit)

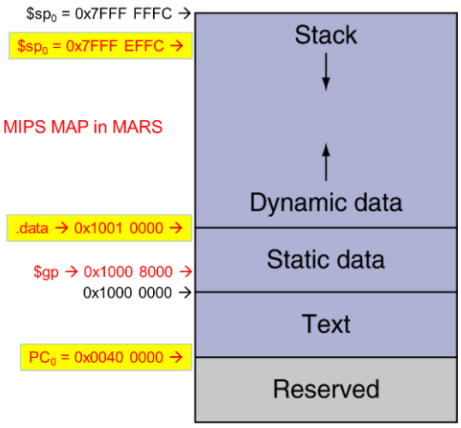


(1-a) Single-Cycle (non-pipelined) Datapath

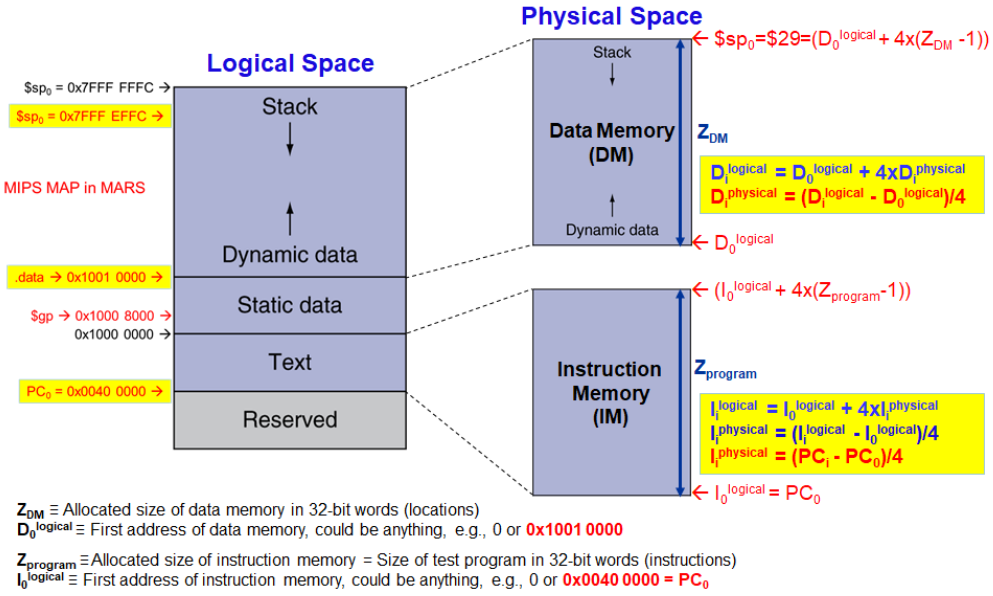
opcode	funct	ALUControl	RegDst	ALUSrc	MemToReg	RegWr	MemWr	BEQ	J
R-type ≡ 000000	AND 100100	0000	1	0	0	1	0	0	0
	OR 100101	0001							
	add 100000	0010							
	sub 100010	0110							
	slt 101010	0111							
	NOR 100111	1100							
lw ≡ 100011	xxxxxx	0010	0	1	1	1	0	0	0
sw ≡ 101011	xxxxxx	0010	0	1	0	0	1	0	0
beq ≡ 000100	xxxxxx	0110	0	0	0	0	0	1	0
j ≡ 000010	xxxxxx	0000	0	0	0	0	0	0	1
addi ≡ 001000	xxxxxx	0010	0	1	0	1	0	0	0

(1-b) Control Unit Functionality

Figure 1. Single-Cycle (non-pipelined) MIPS Processor



(2-a) MIPS Memory Layout/Map



(2-b) Formulae for Address Translation/Mapping

Figure 2. Convention for Memory Address Translation/Mapping

test_program_assembly.asm

```
1         addi    $a0, $zero, 3   # Instruction 00
2         addi    $t0, $zero, 10  # Instruction 01
3         addi    $t1, $zero, 6   # Instruction 02
4         and     $t2, $t1, $t0   # Instruction 03
5         or      $t3, $t1, $t0   # Instruction 04
6         nor     $t4, $t1, $t0   # Instruction 05
7   start: slt    $t5, $t1, $t0   # Instruction 06
8         sw      $t0,  0($sp)    # Instruction 07
9         sw      $t1, -4($sp)    # Instruction 08
10        lw      $s0,  0($sp)    # Instruction 09
11        lw      $s1, -4($sp)    # Instruction 10
12        beq     $s0, $s1, else  # Instruction 11
13        add     $s3, $s0, $s1   # Instruction 12
14        j       exit            # Instruction 13
15  else: sub     $s3, $s0, $s1   # Instruction 14
16  exit: add     $s0, $s0, $s3   # Instruction 15
17        or      $s1, $s1, $s3   # Instruction 16
18        addi    $t0, $t0,  3    # Instruction 17
19        addi    $t1, $t1,  3    # Instruction 18
20        addi    $sp, $sp, -8    # Instruction 19
21        addi    $a0, $a0, -1    # Instruction 20
22        slt     $a1, $zero, $a0 # Instruction 21
23        beq     $a1, $zero, end # Instruction 22
24        j       start           # Instruction 23
25  end:  j       end             # Instruction 24
```

Figure 3. Assembly Test Program

- In Vivado
  - Create a blank project
  - Add design and simulation source files
  - Run behavioral simulation
  - Your waveform configuration should be identical to the provided waveform snapshots, see Figure 3.
- Steps:
  1) Download the file "HW09_MIPS_Single_Cycle.zip" from Canvas and extract its contents.
  2) Rename the folder "HW09_MIPS_Single_Cycle" to "HW09_MIPS_Single_Cycle_<your last name>", for example "HW09_MIPS_Single_Cycle_El-Araby".
  3) Launch Vivado and create a new project, for example "vivado_project", with the default settings under the following directory "\HW09_MIPS_Single_Cycle_<your last name>" resulting in the following project directory "\HW09_MIPS_Single_Cycle_<your last name>\vivado_project\"
  4) Add to the project the VHDL design and simulation source files from the folders; "\HW09_MIPS_Single_Cycle_<your last name>\design_sources" and "\HW09_MIPS_Single_Cycle_<your last name>\simulation_sources" respectively.
  5) Edit the VHDL files in the folder "\HW09_MIPS_Single_Cycle_<your last name>\design_sources\" according to your design such that it describes the required *MIPS microarchitecture*.
  6) Set the simulation time to the proper time, e.g., ***750 ns***, and then launch Vivado Simulator.
  7) Verify the correctness of your design by comparing your simulation results, shown in Figure 4, with those of MARS runs, shown in Figure 5, using the provided assembly test program "\HW09_MIPS_Single_Cycle\test_program\test_program_assembly.asm" shown in Figure 3. Your waveform configuration should be identical to the waveform snapshots shown in Figure 4. You may go back to step 5 to correct your code until your design works properly as required.
  8) After you are done, compress the folder "\HW09_MIPS_Single_Cycle_<your last name>" to "HW09_MIPS_Single_Cycle_<your last name>.zip", for example "HW09_MIPS_Single_Cycle_El-Araby.zip" and upload it to Canvas before the due date and time.

***Important Hint*:**

When you use the components "PC_register", "InstrMem", "RegFile", "ALU", "DataMem", and "CU", in your top-level design "top_mips_single_cycle.vhd", name their instances by appending "_inst" to the end of the component name as follows "PC_register_inst", "InstrMem_inst", "RegFile_inst", "ALU_inst", "DataMem_inst", and "CU_inst". This will guarantee the correct setup of your waveform configuration for showing all needed signals. For example:

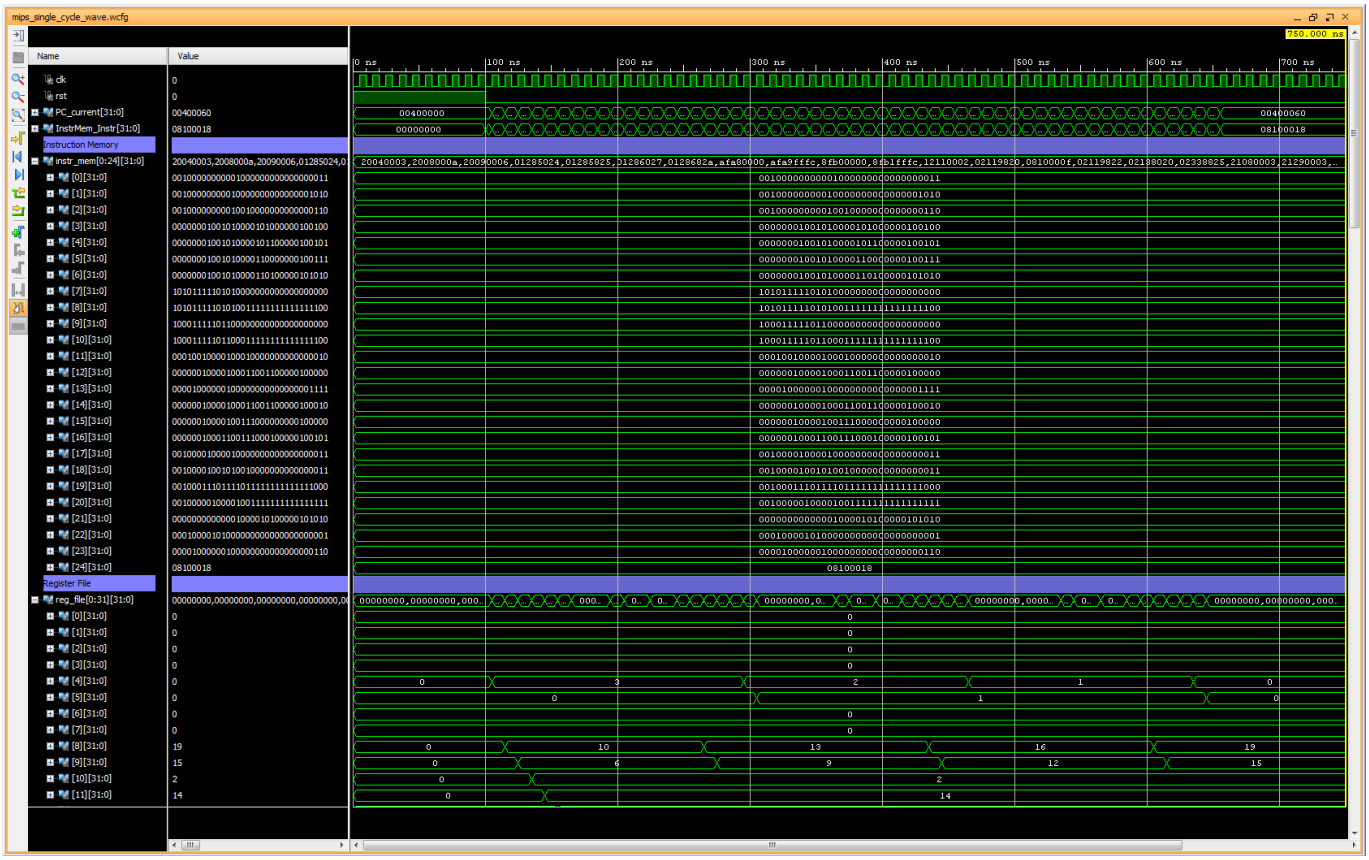
```
PC_register_inst : PC_register
  PORT MAP (
              PC_next    => . . . ,
              clk        => . . . ,
              rst        => . . . ,
            PC_current   =>  . . .
  );
```

Grade Distribution:

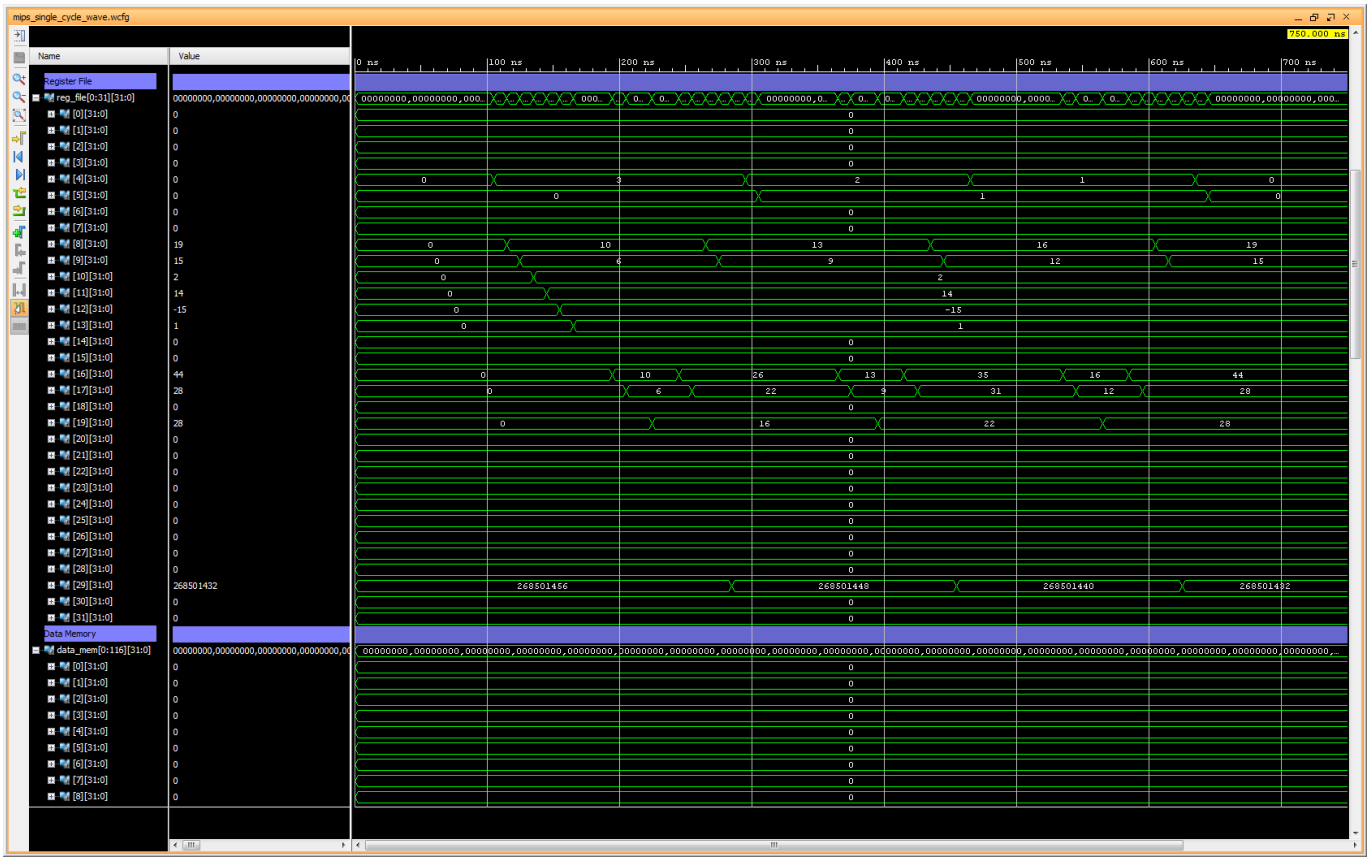
- Functional Correctness, i.e., correct source code → 75 / 100
- Proper Setup of Vivado Project → 25 / 100

NOTE:

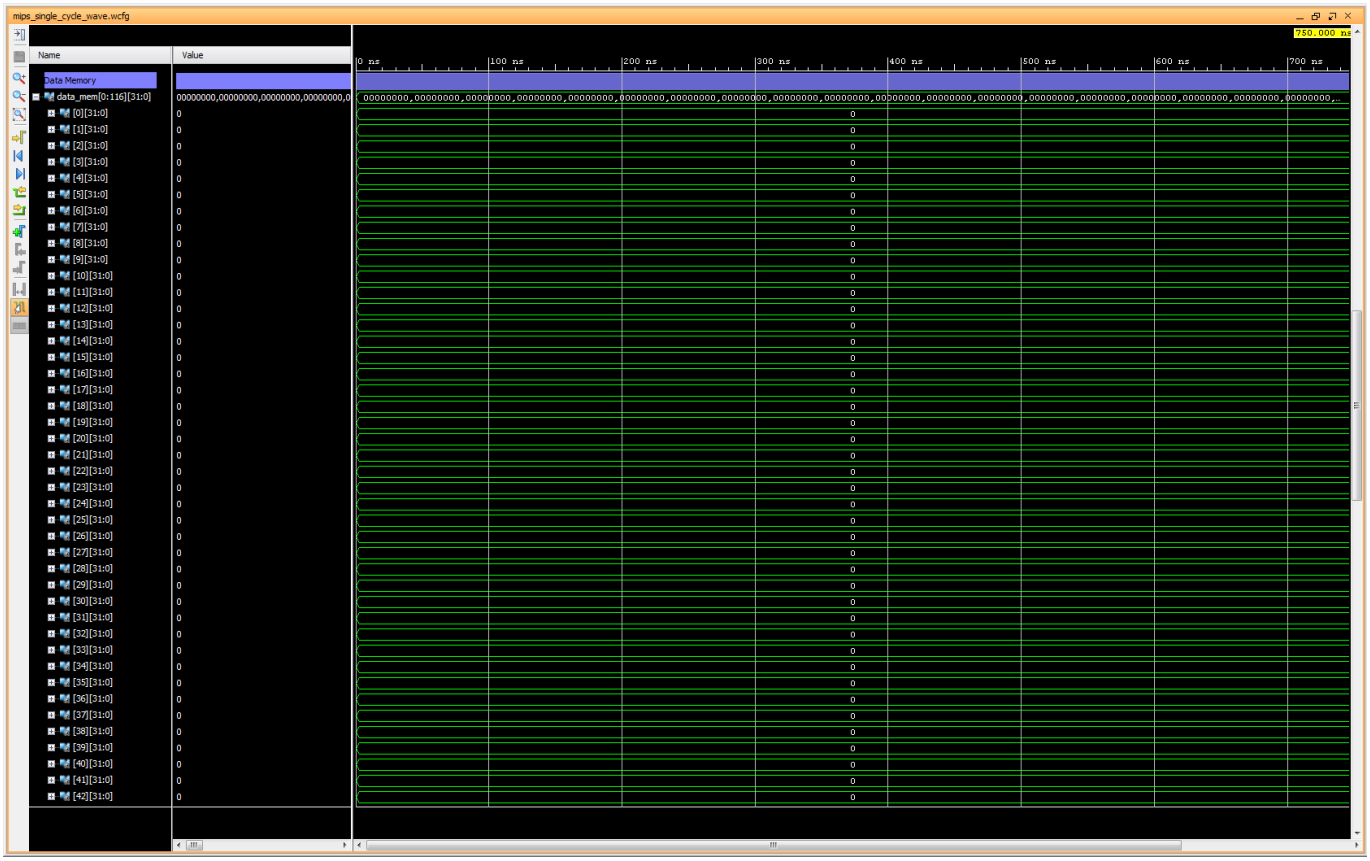
Homework submission is a "Single Attempt", i.e., carefully review everything that you want to submit before hitting the "submit" button and make sure that you have uploaded all documents you want to submit and have not missed anything.


mips_single_cycle_wave.wcfg
Name
Value
clk
rst
PC_current[31:0]
InstrMem_Instr[31:0]
Instruction Memory
instr_mem[0:24][31:0]
Register File
reg_file[0:31][31:0]
750.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns

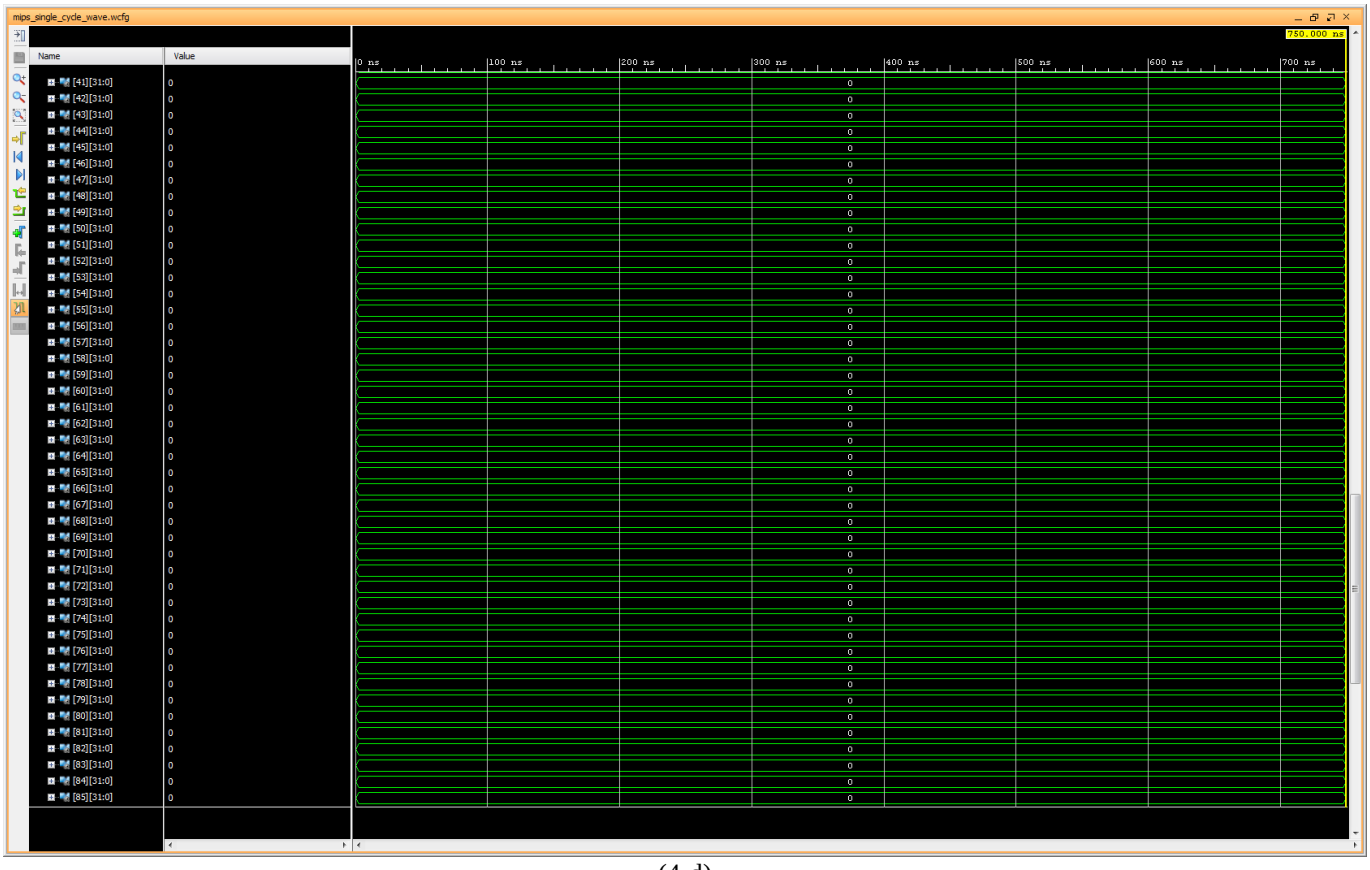
(4-a)


mips_single_cycle_wave.wcfg
Name
Value
Register File
reg_file[0:31][31:0]
Data Memory
data_mem[0:116][31:0]
750.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns

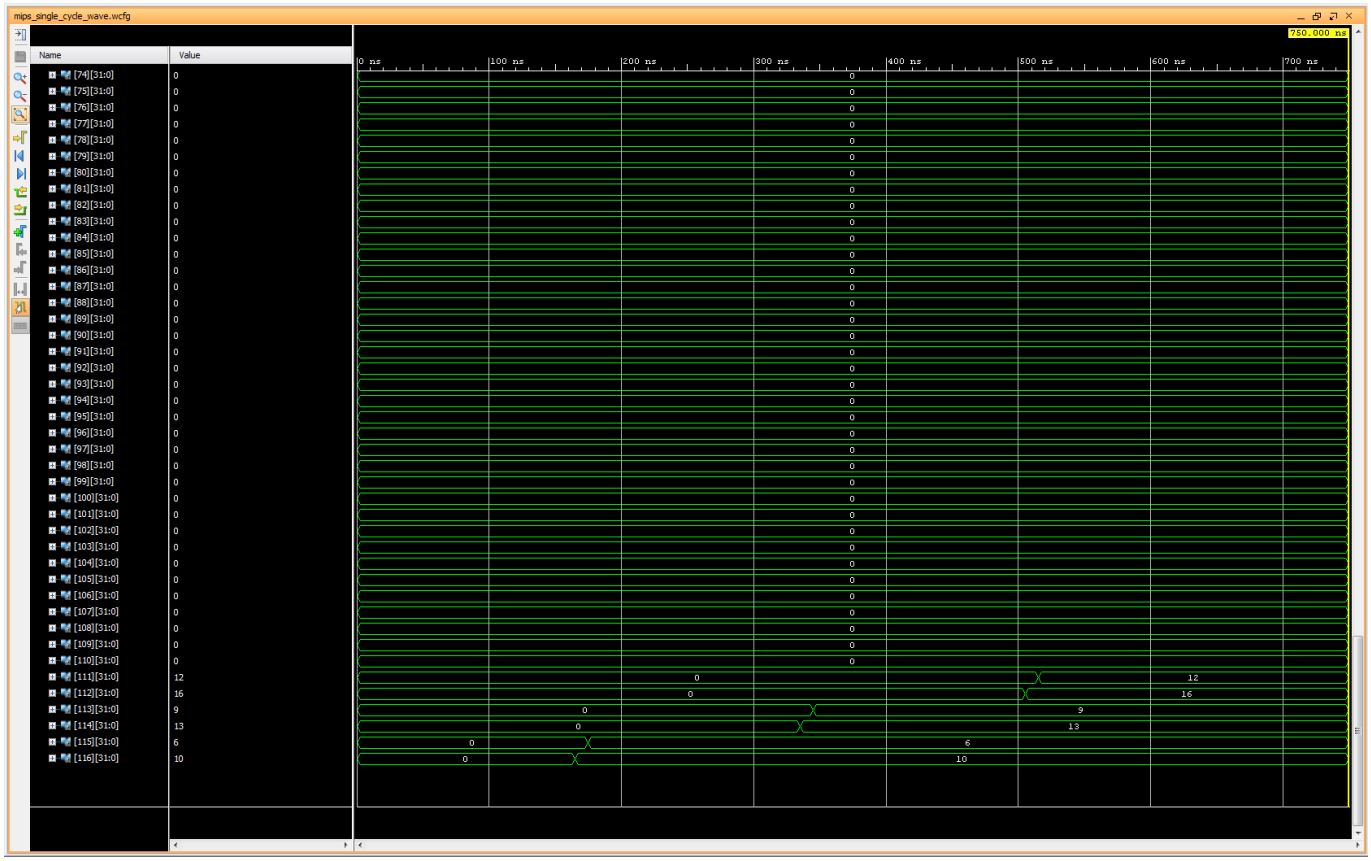
(4-b)


mips_single_cycle_wave.wcfg
Name
Value
Data Memory
data_mem[0:116][31:0]
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
750.000 ns

(4-c)


mips_single_cycle_wave.wcfg
Name
Value
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
750.000 ns

(4-d)



(4-e)

Figure 4. Snapshots of Correct Waveform Configuration for Single-Cycle (non-pipelined) MIPS Processor

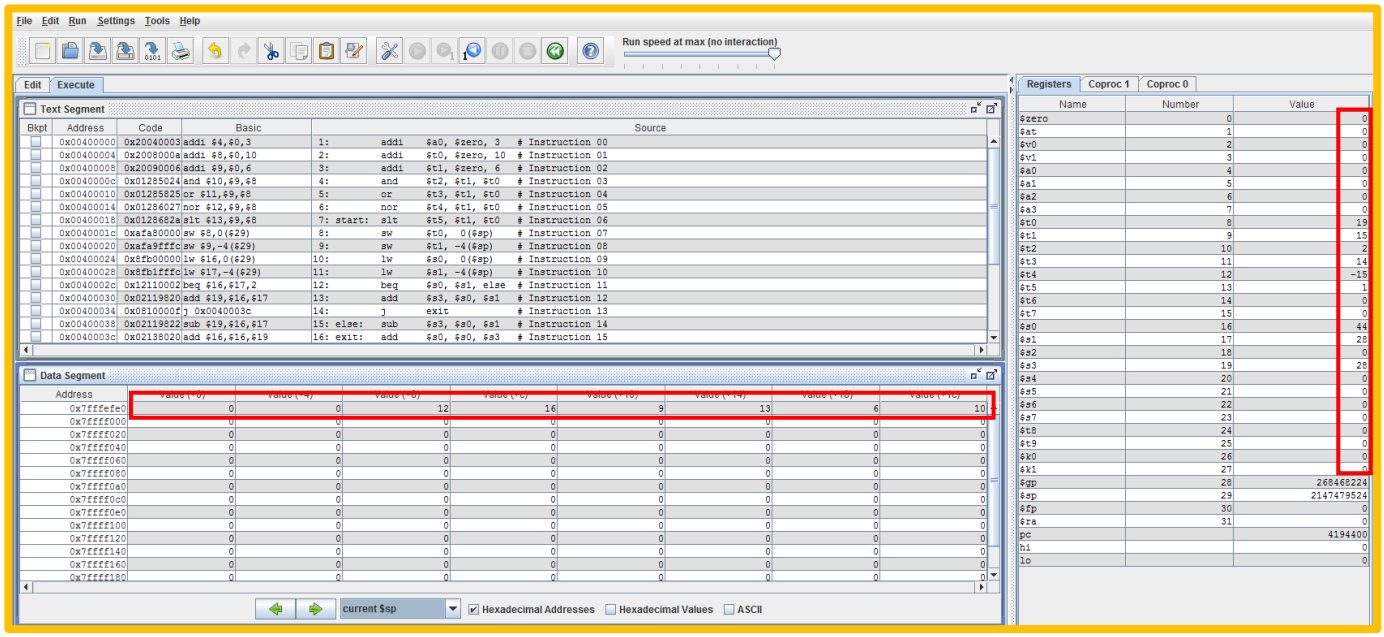


Figure 5: MARS Results of the Assembly Test Program