RISC-V RV32I Processor

Anika Mahesh, Henry Tejada Deras

Code: GitHub - H-TejadaDeras/32-bit-RISC-V-Processor: 32-bit RISC-V Processor Integer Microprocessor with a Harvard Architecture for Computer Architecture Course (ENGR3410-01.25FA) in the 2025 Fall Semester.

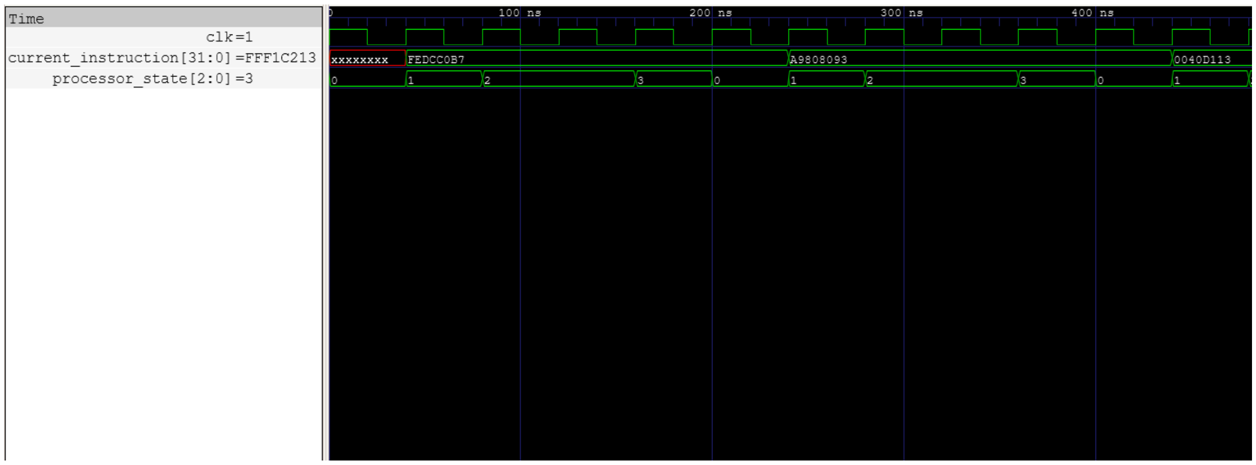
Processor Design Overview

This report details our design for an unpipelined, multicycle (i.e., the processor executes only one instruction at a time), 32-bit RISC-V integer microprocessor with a Harvard architecture. This design is contained within the top.sv, alu.sv, decoder.sv, and the memory.sv files, and the testbench is in test_bench_top.sv. In top.sv, all the registers are defined (x0-x31, pc, and current_instruction), the processor state machine is defined (this is how we keep track of which state the processor is currently in), data memory operations, and register memory operations. In decoder.sv, the decoder is defined along with all the logic used to sign-extend all the immediates. alu.sv defines all the logic related to all logic and arithmetic operations performed by instructions. Note: we do have various arithmetic operations sprinkled around in the top module, this is implemented as very tiny ALUs that perform a limited set of operations to reduce timing complexities. In terms of instructions, the ALU implements the `addi`, `slti`, `sltiu`, `xori`, `ori`, `andi`, `slli`, `srli`, `srai`, `add`, `sub`, `sll`, `slt`, `sltu`, `zor`, `srl`, `sra`, `or`, and `and` instructions. All the other instructions are implemented in top.sv.

In terms of memories, we have a few registers. The x0-x31 registers are for storing data that is going to be accessed frequently or soon. The pc register is for storing the address in data memory for the next instruction, and the current_instruction register is for storing the current instruction that is being executed by the processor.

The processor has 4 different states. Fetch Instruction is the first state, and it executes when the previous instruction is finished running. It reads the next instruction from the instruction memory (defined by the address stored on the pc register) and changes the state to Fetch Registers. The Fetch Registers state stores the instruction in the current_instruction register and changes the state to Execute instruction, which executes the instruction defined by the current instruction. During the fetch registers stage, the decoder simultaneously decodes the new instruction saved to the current_instruction register. In the Execute Instruction state, the processor uses the decoded values from decoder.sv to execute it. Since all values from the decoder are properly sign-extended, the top module just decides where to route the data. After the instruction is executed, the state is Write Back, which signals the instruction is completed, and the value of the program counter changes depending on the instruction that has been executed to get the next instruction, and switches the state back to Fetch instruction. This process repeats to process all the given instructions in the instruction memory.

The processor states can be seen here, where 0 is the Fetch Instruction state, 1 is Fetch Registers, 2 is Execute Instruction, and 3 is Write Back.

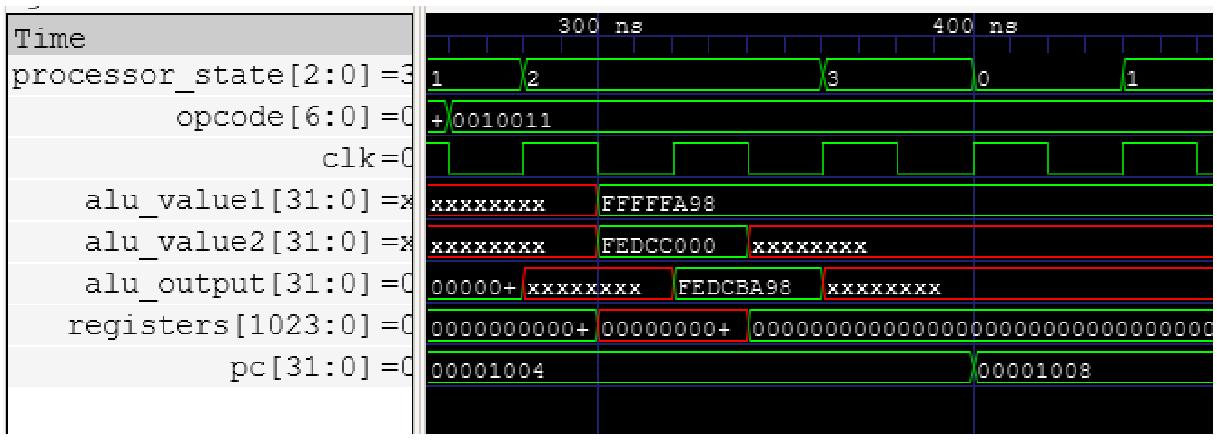


Instruction Functionality Validation Tests

ALU Instructions (rv32i_test.s)

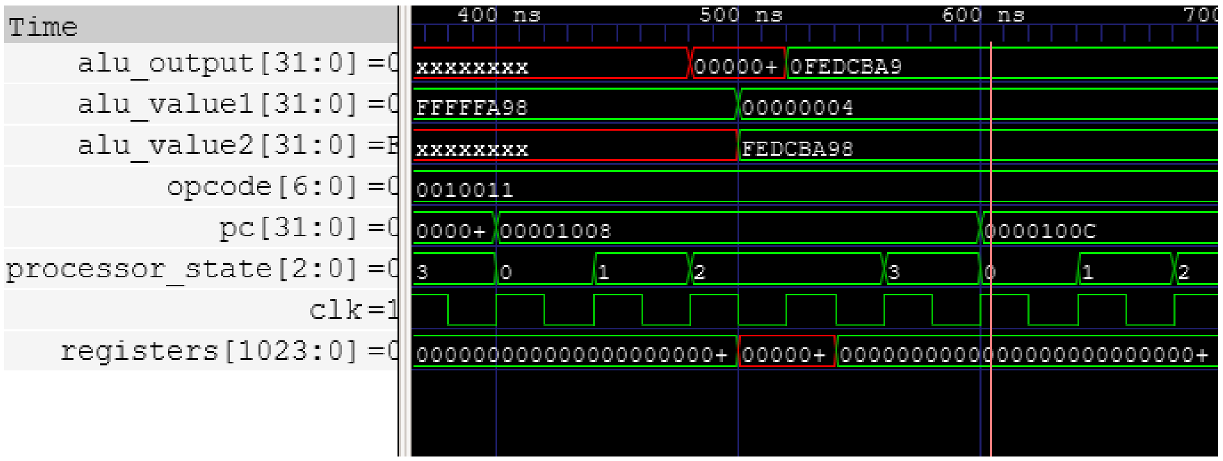
Testing `addi`, `srli`, `srai`, `xori`, `add`, `sub`, `sll`, `ori`, `slt`, and `sltu` instructions.

addi x1, x1, 0xA98   x1 = 0xFEDCBA98



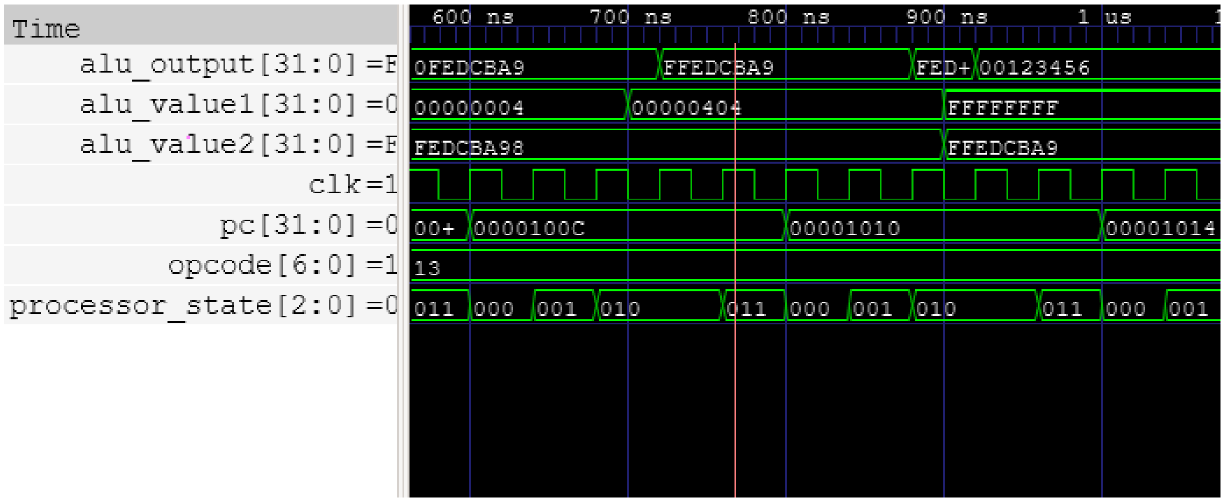
The result is correct (alu_output); alu and registers are updated correctly; and the program counter increments by 4.

srli x2, x1, 4   pc = 0x08, x2 = 0x0FEDCBA9



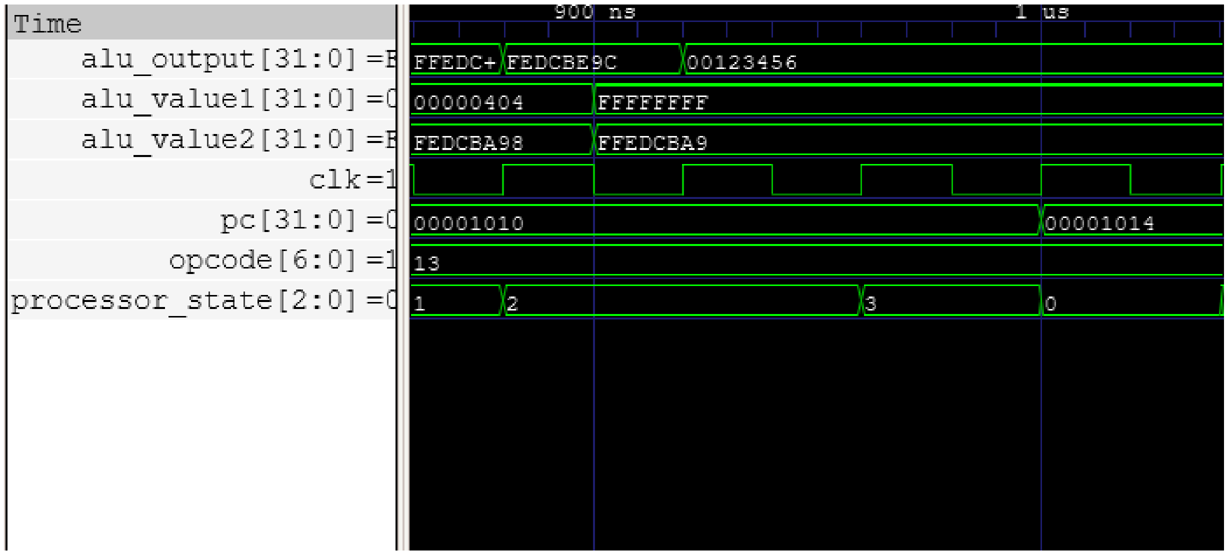
Result is correct and alu and registers are updated and program counter updates by 4

srai x3, x1, 4   pc = 0x0C, x3 = 0xFFEDCBA9



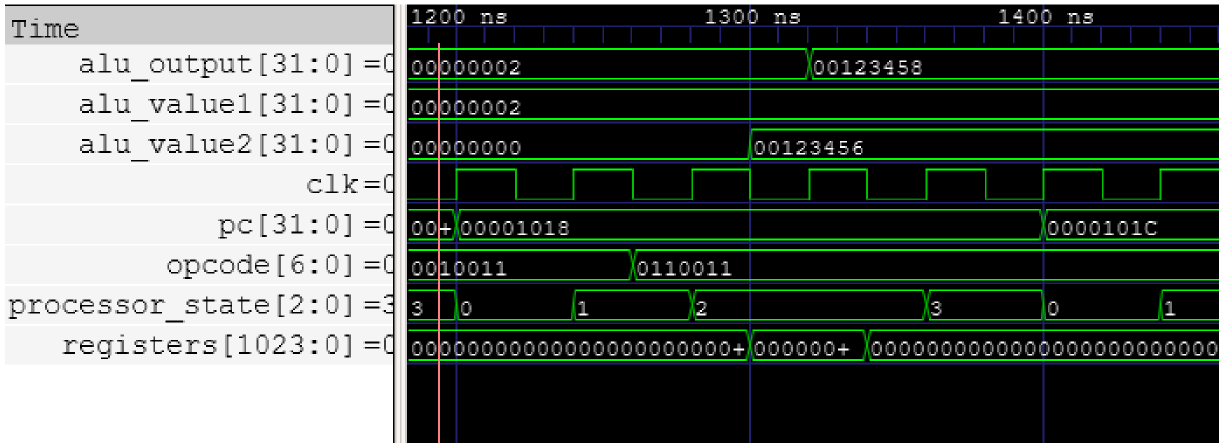
Result is correct and alu and registers are updated and program counter updates by 4

xori x4, x3, -1   pc = 0x10, x4 = 0x00123456



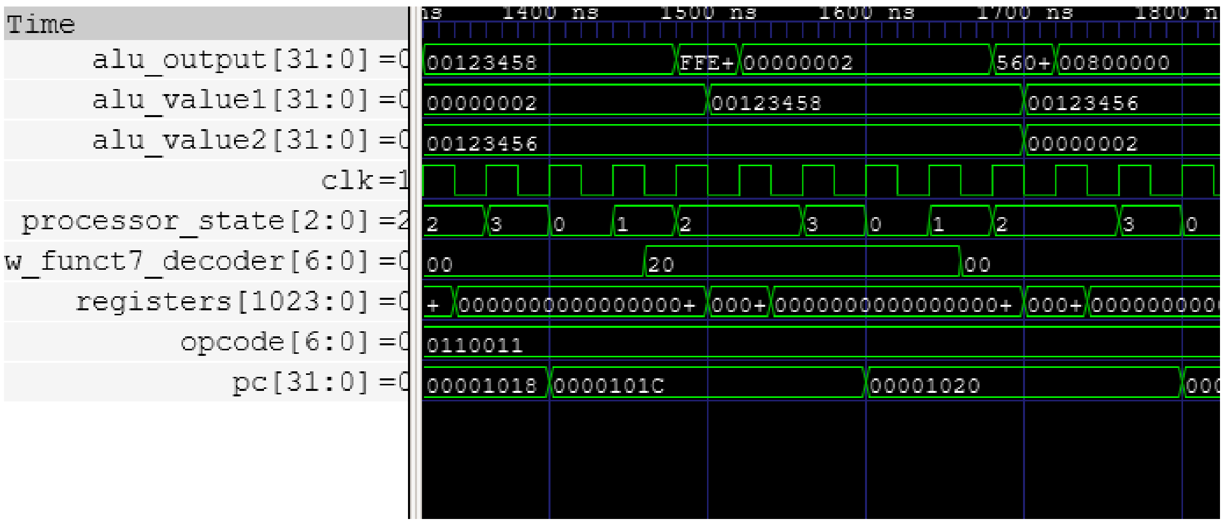
Result is correct and program counter updates by 4

add x6, x5, x4 pc = 0x18, x6 = 0x00123458



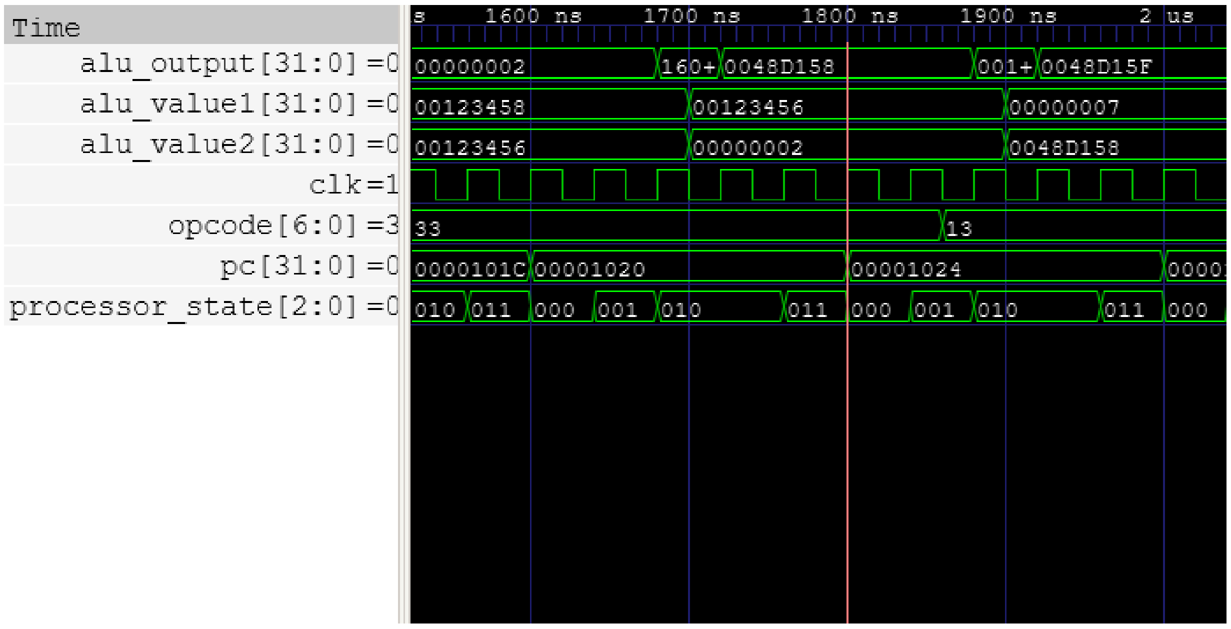
Result is correct and program counter updates by 4

sub x7, x6, x4 pc = 0x1C, x7 = 0x00000002



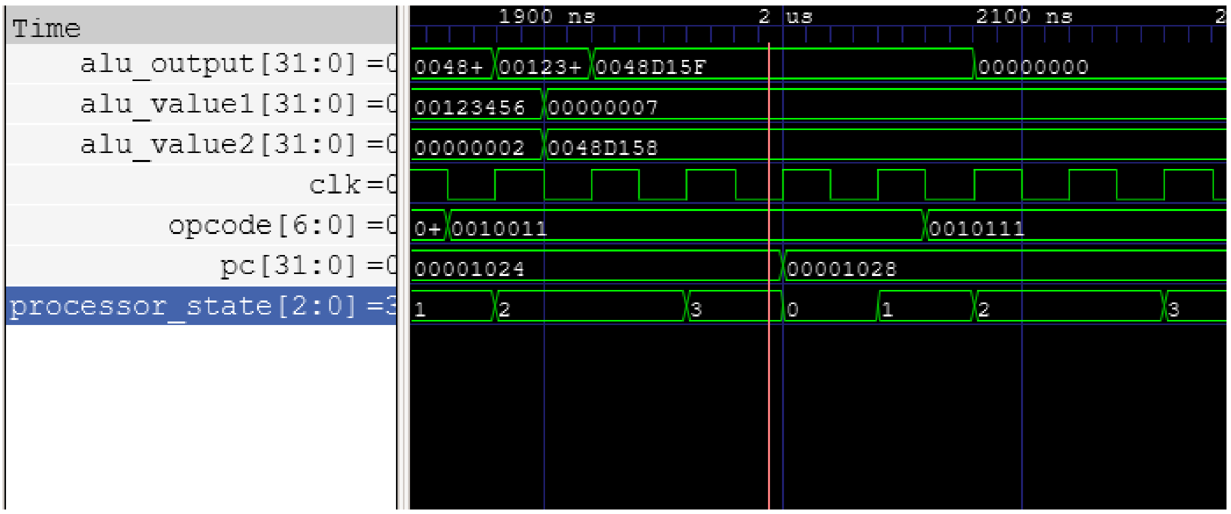
Result is correct and alu and registers are updated and program counter updates by 4

sll x8, x4, x5 **pc = 0x20, x8 = 0x0048D158**



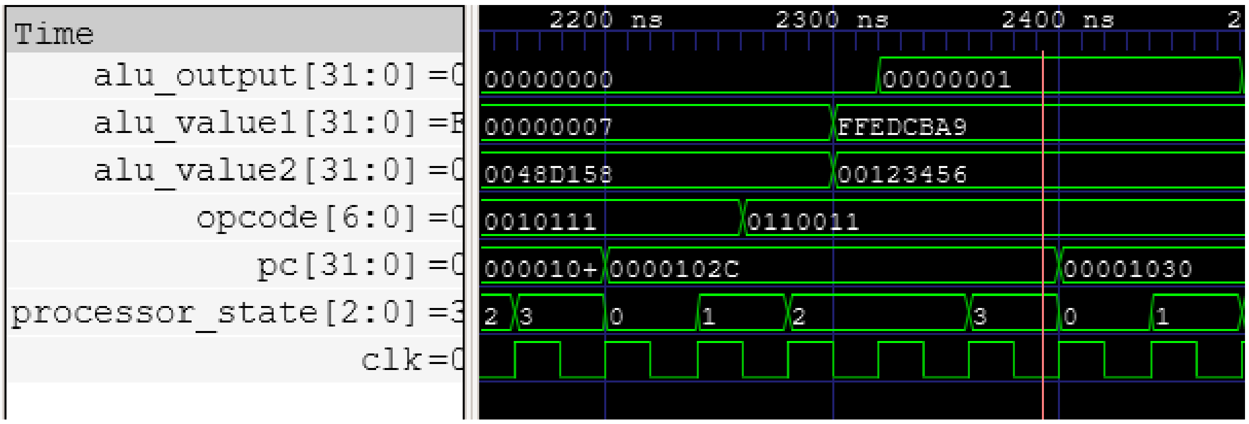
Result is correct and alu and registers are updated and program counter updates by 4

ori x9, x8, 7 **pc = 0x24, x9 = 0x0048D15F**



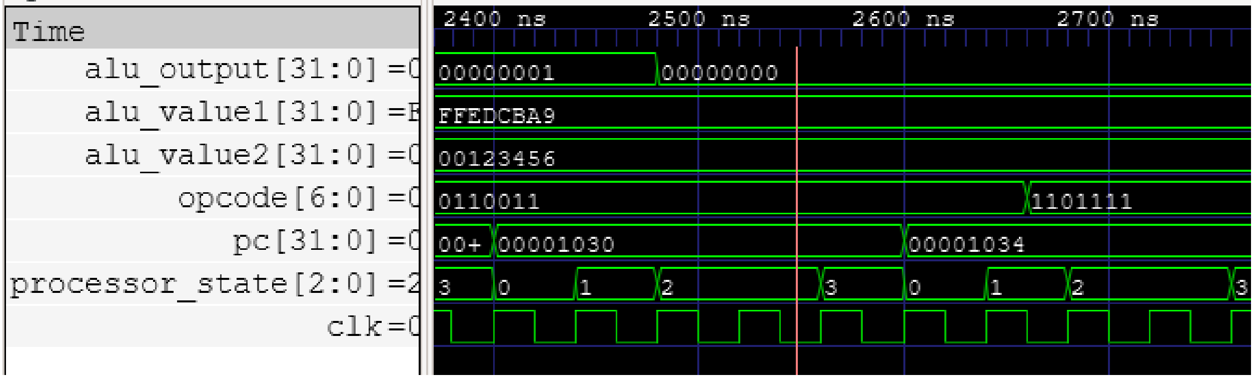
Result is correct and alu and registers are updated and program counter updates by 4

slt x11, x3, x4   pc = 0x2C, x11 = 0x00000001



Result is correct and alu and registers are updated and program counter updates by 4

sltu x12, x3, x4   pc = 0x30, x12 = 0x00000000



Result is correct and alu and registers are updated and program counter updates by 4

Branch Instructions (branch_instruct_test.s + tester_dmem.hex)
Testing bne and blt instructions. Note: This test makes use of tester_dmem.hex.

tester_dmem.hex

```
01234567
76543210
11223344
55667788
99101011
12346548
15975346
26489173
16AF2345
0316497D
00001010
00000000
…
```

branch_instruct_test.s

```
add x2, x0, x0 # Set stack pointer
lw x5, 0(x2) # 01234567 -> x5
lw x6, 4(x2) # 76543210 -> x6
bne x5, x6, 8
sw x6, 1000(x2) # x6 -> 00002000 (address)
blt x6, x5, 24
```

Processor Status (when running the above program)		
Program Counter: 00001000 Current Instruction: 00000133 [add x2, x0, x0] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 00000000 Register x6: 00000000 Register x7: 00000000 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000	Program Counter: 00001004 Current Instruction: 00012283 [lw x5, 0(x2)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00000000 Register x7: 00000000 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000	Program Counter: 00001008 Current Instruction: 00412303 [lw x6, 4(x2)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000

Register x20: 00000000	Register x20: 00000000	Register x20: 00000000
Register x21: 00000000	Register x21: 00000000	Register x21: 00000000
Register x22: 00000000	Register x22: 00000000	Register x22: 00000000
Register x23: 00000000	Register x23: 00000000	Register x23: 00000000
Register x24: 00000000	Register x24: 00000000	Register x24: 00000000
Register x25: 00000000	Register x25: 00000000	Register x25: 00000000
Register x26: 00000000	Register x26: 00000000	Register x26: 00000000
Register x27: 00000000	Register x27: 00000000	Register x27: 00000000
Register x28: 00000000	Register x28: 00000000	Register x28: 00000000
Register x29: 00000000	Register x29: 00000000	Register x29: 00000000
Register x30: 00000000	Register x30: 00000000	Register x30: 00000000
Register x31: 00000000	Register x31: 00000000	Register x31: 00000000
Program Counter: 0000100c	Program Counter: 00001014	Program Counter: 00001018
Current Instruction: 00629463	Current Instruction: 00534c63	Current Instruction: 00000000
[bne x5, x6, 8]	[blt x6, x5, 24]	[No Instruction]
Register x0: 00000000	Register x0: 00000000	Register x0: 00000000
Register x1: 00000000	Register x1: 00000000	Register x1: 00000000
Register x2: 00000000	Register x2: 00000000	Register x2: 00000000
Register x3: 00000000	Register x3: 00000000	Register x3: 00000000
Register x4: 00000000	Register x4: 00000000	Register x4: 00000000
Register x5: 01234567	Register x5: 01234567	Register x5: 01234567
Register x6: 76543210	Register x6: 76543210	Register x6: 76543210
Register x7: 00000000	Register x7: 00000000	Register x7: 00000000
Register x8: 00000000	Register x8: 00000000	Register x8: 00000000
Register x9: 00000000	Register x9: 00000000	Register x9: 00000000
Register x10: 00000000	Register x10: 00000000	Register x10: 00000000
Register x11: 00000000	Register x11: 00000000	Register x11: 00000000
Register x12: 00000000	Register x12: 00000000	Register x12: 00000000
Register x13: 00000000	Register x13: 00000000	Register x13: 00000000
Register x14: 00000000	Register x14: 00000000	Register x14: 00000000
Register x15: 00000000	Register x15: 00000000	Register x15: 00000000
Register x16: 00000000	Register x16: 00000000	Register x16: 00000000
Register x17: 00000000	Register x17: 00000000	Register x17: 00000000
Register x18: 00000000	Register x18: 00000000	Register x18: 00000000
Register x19: 00000000	Register x19: 00000000	Register x19: 00000000
Register x20: 00000000	Register x20: 00000000	Register x20: 00000000
Register x21: 00000000	Register x21: 00000000	Register x21: 00000000
Register x22: 00000000	Register x22: 00000000	Register x22: 00000000
Register x23: 00000000	Register x23: 00000000	Register x23: 00000000
Register x24: 00000000	Register x24: 00000000	Register x24: 00000000

Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: bne instruction properly increments the pc by 8 and arrives at the correct answer from the comparison.	Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: blt instruction properly updates pc by incrementing it by 4 (and not 24) and arrives at the correct answer from the comparison.	Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: There are no instructions past this.

Misc. Instructions (misc_instruct_test.s + tester_dmem.hex)
Testing lui, auipc, jal, and jalr instructions. Note: This test makes use of tester_dmem.hex.

tester_dmem.hex
```
01234567
76543210
11223344
55667788
99101011
12346548
15975346
26489173
16AF2345
0316497D
00001010
00000000
…
```

misc_instruct_test.s
```
add x2, x0, x0 # Set stack pointer
lw x5, 0(x2) # 01234567 -> x5
lw x6, 4(x2) # 76543210 -> x6
lw x28, 40(x2) # 00000010 -> x28
auipc x8, 16
addi x9, x9, 10 # 00000000 -> 0000000A
addi x9, x9, 1 # 0000000A -> 0000000B
jal x10, 8 # Move to 2nd instruction down (lui)
addi x29, x0, x0 # Filler instruction; not meant to do anything
lui x9, 400000
jalr x10, 4002(x28) # Move to auipc instruction
addi x9, x9, 1 # 0000000B -> 0000000C
```

Processor Status (when running the above program)		
Program Counter: 00001000 Current Instruction: 00000133 [add x2, x0, x0] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000	Program Counter: 00001004 Current Instruction: 00012283 [lw x5, 0(x2)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000	Program Counter: 00001008 Current Instruction: 00412303 [lw x6, 4(x2)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000

Register x5: 00000000	Register x5: 01234567	Register x5: 01234567
Register x6: 00000000	Register x6: 00000000	Register x6: 76543210
Register x7: 00000000	Register x7: 00000000	Register x7: 00000000
Register x8: 00000000	Register x8: 00000000	Register x8: 00000000
Register x9: 00000000	Register x9: 00000000	Register x9: 00000000
Register x10: 00000000	Register x10: 00000000	Register x10: 00000000
Register x11: 00000000	Register x11: 00000000	Register x11: 00000000
Register x12: 00000000	Register x12: 00000000	Register x12: 00000000
Register x13: 00000000	Register x13: 00000000	Register x13: 00000000
Register x14: 00000000	Register x14: 00000000	Register x14: 00000000
Register x15: 00000000	Register x15: 00000000	Register x15: 00000000
Register x16: 00000000	Register x16: 00000000	Register x16: 00000000
Register x17: 00000000	Register x17: 00000000	Register x17: 00000000
Register x18: 00000000	Register x18: 00000000	Register x18: 00000000
Register x19: 00000000	Register x19: 00000000	Register x19: 00000000
Register x20: 00000000	Register x20: 00000000	Register x20: 00000000
Register x21: 00000000	Register x21: 00000000	Register x21: 00000000
Register x22: 00000000	Register x22: 00000000	Register x22: 00000000
Register x23: 00000000	Register x23: 00000000	Register x23: 00000000
Register x24: 00000000	Register x24: 00000000	Register x24: 00000000
Register x25: 00000000	Register x25: 00000000	Register x25: 00000000
Register x26: 00000000	Register x26: 00000000	Register x26: 00000000
Register x27: 00000000	Register x27: 00000000	Register x27: 00000000
Register x28: 00000000	Register x28: 00000000	Register x28: 00000000
Register x29: 00000000	Register x29: 00000000	Register x29: 00000000
Register x30: 00000000	Register x30: 00000000	Register x30: 00000000
Register x31: 00000000	Register x31: 00000000	Register x31: 00000000
Program Counter: 0000100c	Program Counter: 00001010	Program Counter: 00001014
Current Instruction: 02812e03	Current Instruction: 00010417	Current Instruction: 00a48493
[lw x28, 40(x2)]	[auipc x8, 16]	[addi x9, x9, 10]
Register x0: 00000000	Register x0: 00000000	Register x0: 00000000
Register x1: 00000000	Register x1: 00000000	Register x1: 00000000
Register x2: 00000000	Register x2: 00000000	Register x2: 00000000
Register x3: 00000000	Register x3: 00000000	Register x3: 00000000
Register x4: 00000000	Register x4: 00000000	Register x4: 00000000
Register x5: 01234567	Register x5: 01234567	Register x5: 01234567
Register x6: 76543210	Register x6: 76543210	Register x6: 76543210
Register x7: 00000000	Register x7: 00000000	Register x7: 00000000
Register x8: 00000000	Register x8: 00011010	Register x8: 00011010
Register x9: 00000000	Register x9: 00000000	Register x9: 0000000a

Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000	Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: auipc properly updates register x8	Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000
Program Counter: 00001018 Current Instruction: 00148493 [addi x9, x9, 1] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00011010 Register x9: 0000000b Register x10: 00000000 Register x11: 00000000	Program Counter: 0000101c Current Instruction: 0080056f [jal x10, 8] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00011010 Register x9: 0000000b Register x10: 00001020 Register x11: 00000000	Program Counter: 00001024 Current Instruction: 61a804b7 [lui x9, 400000] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00011010 Register x9: 61a80000 Register x10: 00001020 Register x11: 00000000

Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000	Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: jal instruction properly saves the next pc value to register x10 and changes the program counter by +8	Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: lui instruction properly changes the value in register x9 by the value in the immediate.
Program Counter: 00001028 Current Instruction: 002e0567 [jalr x10, 4002(x28)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00011010 Register x9: 61a80000 Register x10: 0000102c Register x11: 00000000	Program Counter: 00001012 Current Instruction: 00010417 [Not a real instruction, was moved to half an instruction] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 76543210 Register x7: 00000000 Register x8: 00011012 Register x9: 61a80000 Register x10: 0000102c	The rest of the processor cycles were omitted due to the pc value being at a half-word.

Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: jalr instruction properly updates pc and updated register x10	Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00001010 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000	

Memory Instructions (mem_instruct_test_complete.s + tester_dmem.hex)

Testing lw, lh, lhu, lb, lbu, and sb instructions. Note: This test makes use of tester_dmem.hex.

tester_dmem.hex

```
01234567
76543210
11223344
55667788
99101011
12346548
15975346
26489173
16AF2345
0316497D
00001010
00000000
…
```

mem_instruct_test_complete.s

```
lw x5, 0(x3)
lh x6, 4(x3)
lhu x7, 4(x3)
lb x28, 5(x3)
lbu x29, 5(x3)
sb x28, 250(x3)
lb x30, 250(x3) # verify byte was stored by previous instruction
sh x6, 300(x3)
lh x31, 300(x3) # verify half word was stored by previous instruction
sw x5, 100(x3)
lw x8, 100(x3) # verify word was stored by previous instruction
```

Processor Status (when running the above program)		
Program Counter: 00001000 Current Instruction: 0001a283 [lw x5, 0(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00000000 Register x7: 00000000 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: Word is properly loaded to register x5.	Program Counter: 00001004 Current Instruction: 00419303 [lh x6, 4(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00000000 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: Half word is properly loaded to register x6.	Program Counter: 00001008 Current Instruction: 0041d383 [lhu x7, 4(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000000 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: Unsigned half word is properly loaded to register x7.

Program Counter: 0000100c Current Instruction: 00518e03 [lb x28, 5(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000000 Register x30: 00000000 Register x31: 00000000 Note: Byte is properly loaded to register x28.	Program Counter: 00001010 Current Instruction: 0051ce83 [lbu x29, 5(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000000 Register x31: 00000000 Note: Unsigned byte is properly loaded to register x29.	Program Counter: 00001014 Current Instruction: 0fc18d23 [sb x28, 250(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000000 Register x31: 00000000
Program Counter: 00001018	Program Counter: 0000101c	Program Counter: 00001020

Current Instruction: 0fa18f03 [lb x30, 250(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000032 Register x31: 00000000 Note: Byte was properly stored; this load instruction shows that it was properly stored to the given location.	Current Instruction: 12619623 [sh x6, 300(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000032 Register x31: 00000000	Current Instruction: 12c19f83 [lh x31, 300(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000032 Register x31: 00003210 Note: Half word was properly stored; this load instruction shows that it was properly stored to the given location.

Program Counter: 00001024 Current Instruction: 0651a223 [sw x5, 100(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 00000000 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000032 Register x31: 00003210	Program Counter: 00001028 Current Instruction: 0641a403 [lw x8, 100(x3)] Register x0: 00000000 Register x1: 00000000 Register x2: 00000000 Register x3: 00000000 Register x4: 00000000 Register x5: 01234567 Register x6: 00003210 Register x7: 00003210 Register x8: 01234567 Register x9: 00000000 Register x10: 00000000 Register x11: 00000000 Register x12: 00000000 Register x13: 00000000 Register x14: 00000000 Register x15: 00000000 Register x16: 00000000 Register x17: 00000000 Register x18: 00000000 Register x19: 00000000 Register x20: 00000000 Register x21: 00000000 Register x22: 00000000 Register x23: 00000000 Register x24: 00000000 Register x25: 00000000 Register x26: 00000000 Register x27: 00000000 Register x28: 00000032 Register x29: 00000032 Register x30: 00000032 Register x31: 00003210 Note: Word was properly stored; this load instruction shows that it was properly stored to the given location.	The rest of the processor cycles were omitted due to there being no more instructions.