Single Cycle RISC_V Processor REPORT

Ziad Ahmed

8//27//2024

—

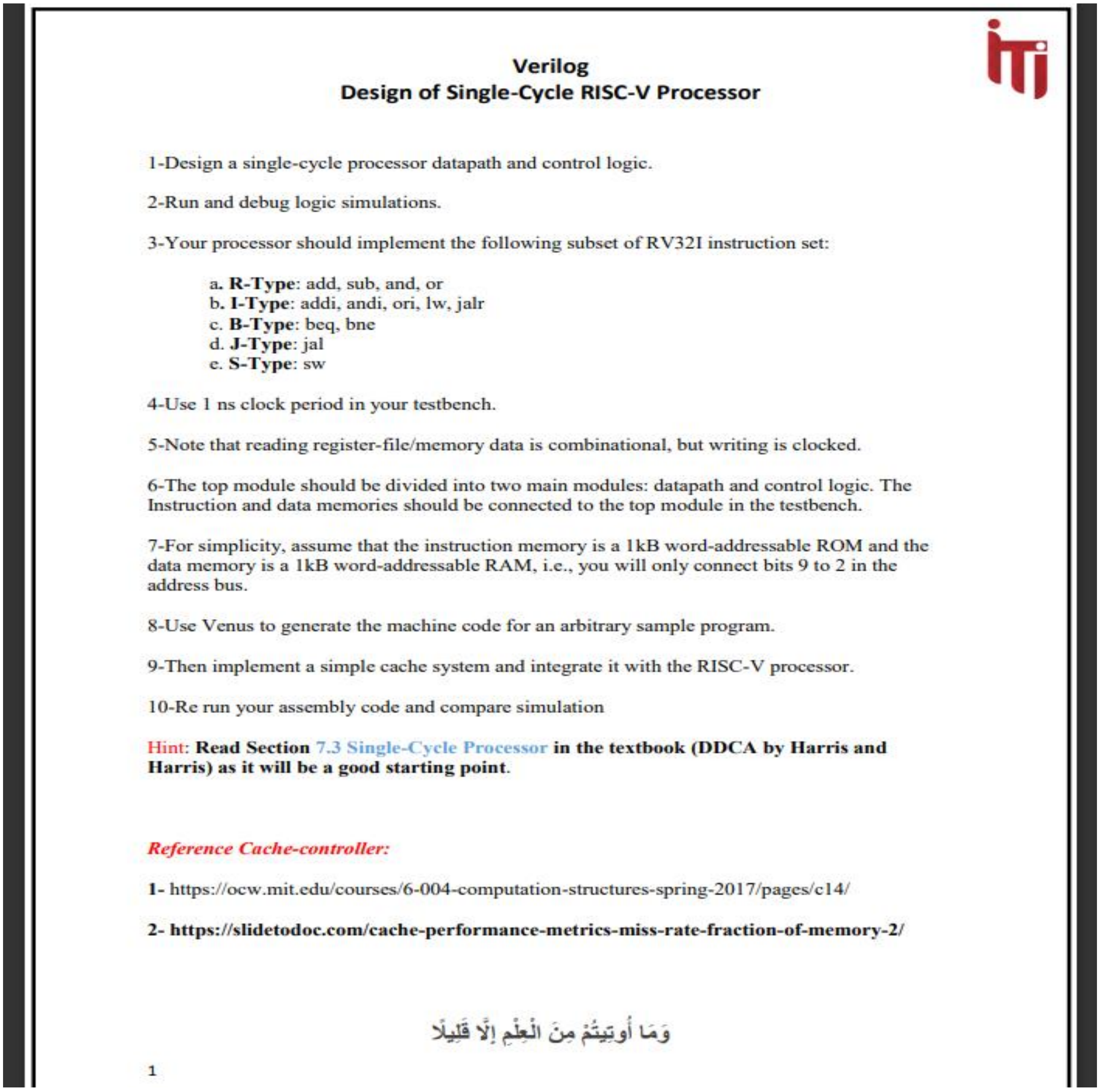
RTL & Functional_Verfication

—

GitHub REPO

ITI

Specs



Verilog
Design of Single-Cycle RISC-V Processor

1-Design a single-cycle processor datapath and control logic.

2-Run and debug logic simulations.

3-Your processor should implement the following subset of RV32I instruction set:

a. **R-Type**: add, sub, and, or
b. **I-Type**: addi, andi, ori, lw, jalr
c. **B-Type**: beq, bne
d. **J-Type**: jal
e. **S-Type**: sw

4-Use 1 ns clock period in your testbench.

5-Note that reading register-file/memory data is combinational, but writing is clocked.

6-The top module should be divided into two main modules: datapath and control logic. The Instruction and data memories should be connected to the top module in the testbench.

7-For simplicity, assume that the instruction memory is a 1kB word-addressable ROM and the data memory is a 1kB word-addressable RAM, i.e., you will only connect bits 9 to 2 in the address bus.

8-Use Venus to generate the machine code for an arbitrary sample program.

9-Then implement a simple cache system and integrate it with the RISC-V processor.

10-Re run your assembly code and compare simulation

Hint: **Read Section 7.3 Single-Cycle Processor in the textbook (DDCA by Harris and Harris) as it will be a good starting point.**

Reference Cache-controller:

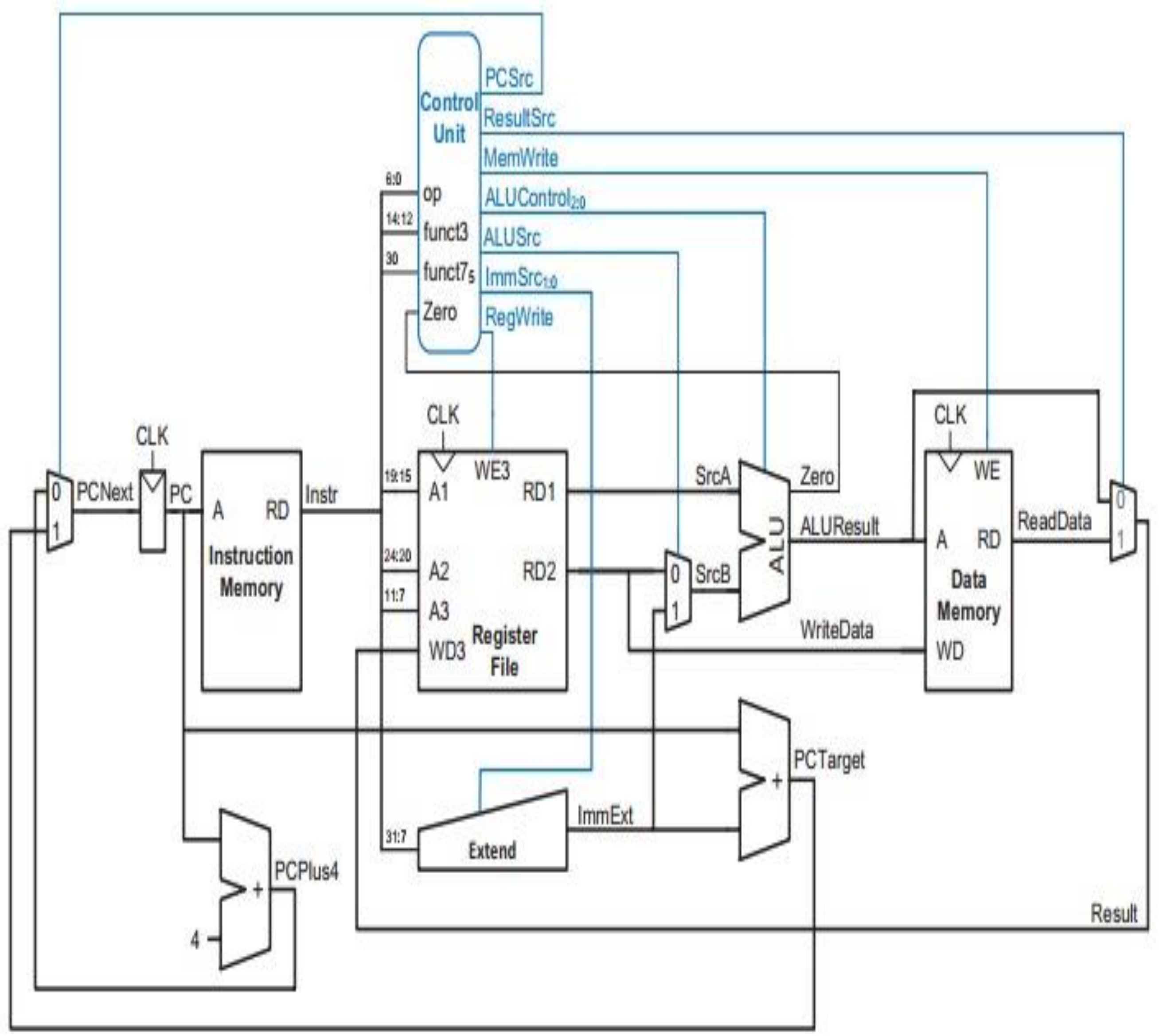
1- https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c14/

2- https://slidetodoc.com/cache-performance-metrics-miss-rate-fraction-of-memory-2/

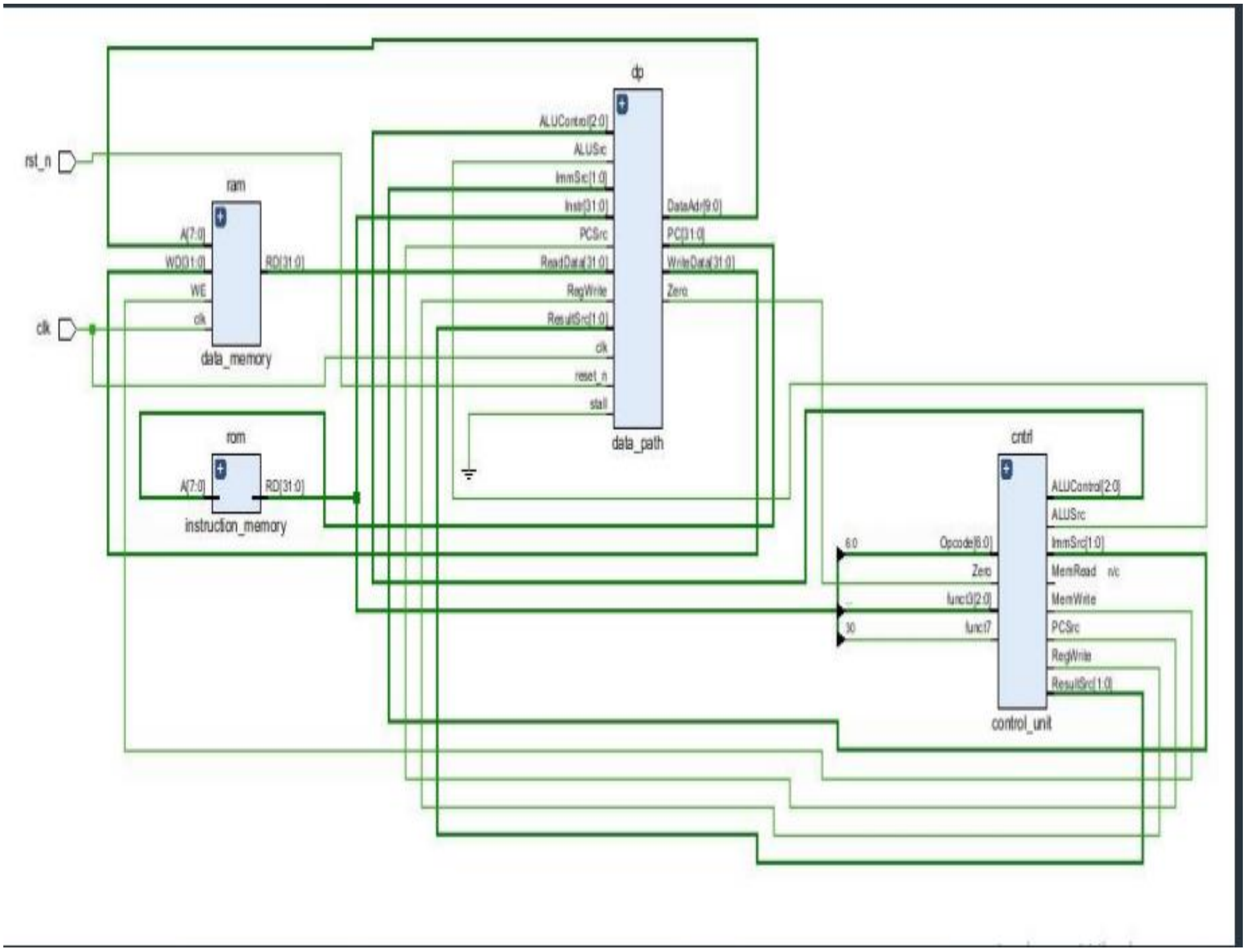
وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

1

Bock Diagram for RISC V Processor

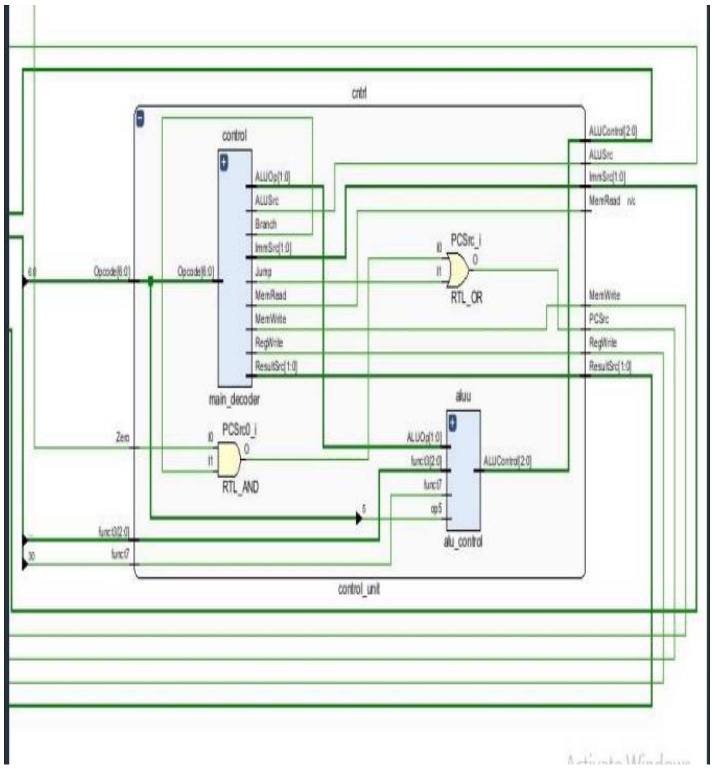


RISC-V Processor Architecture

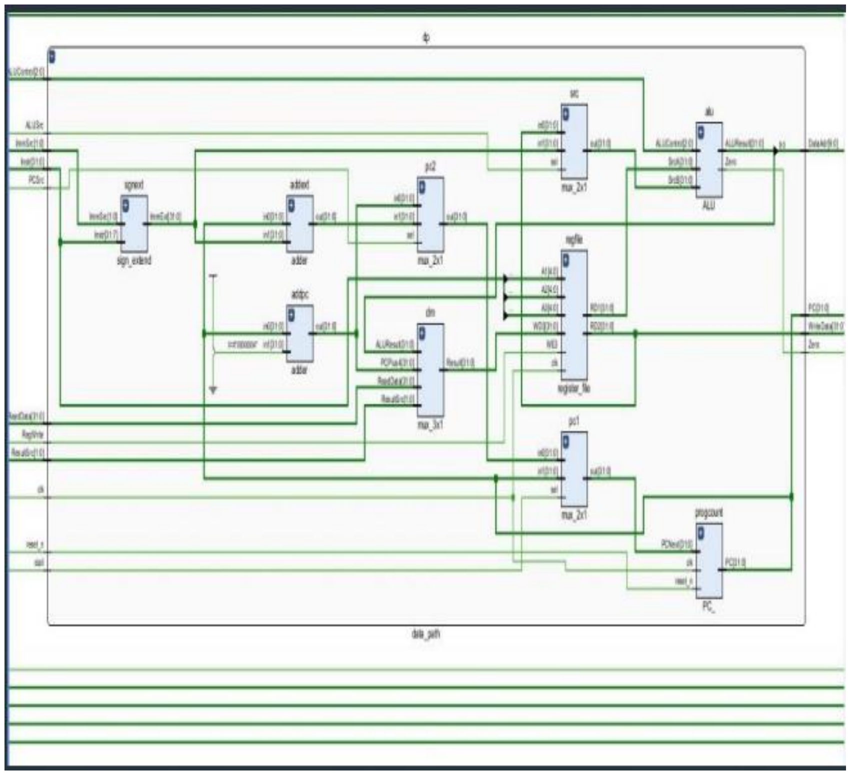


Main Blocks for The System

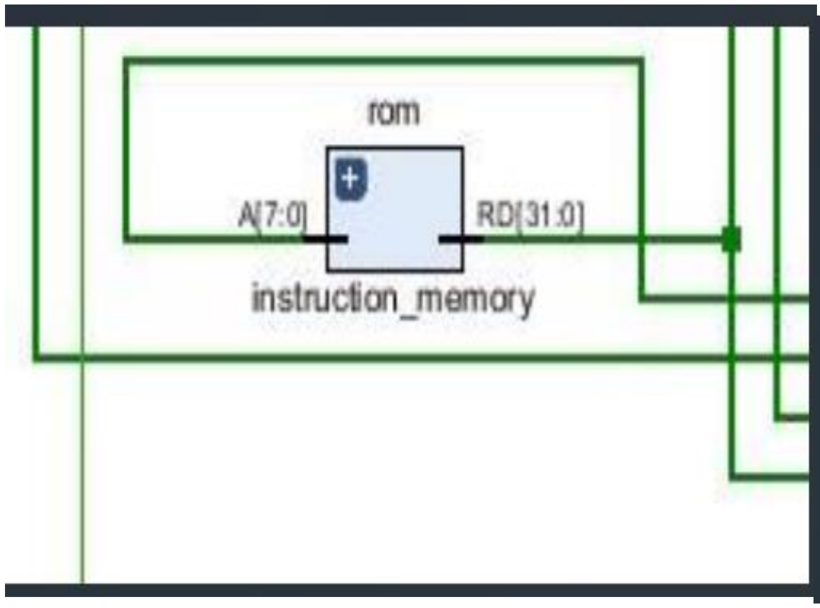
Control unit



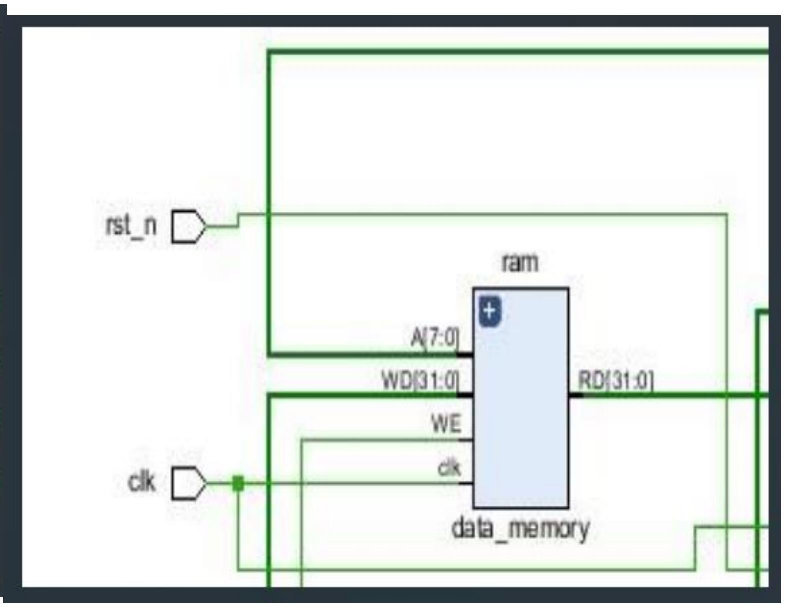
Datapath



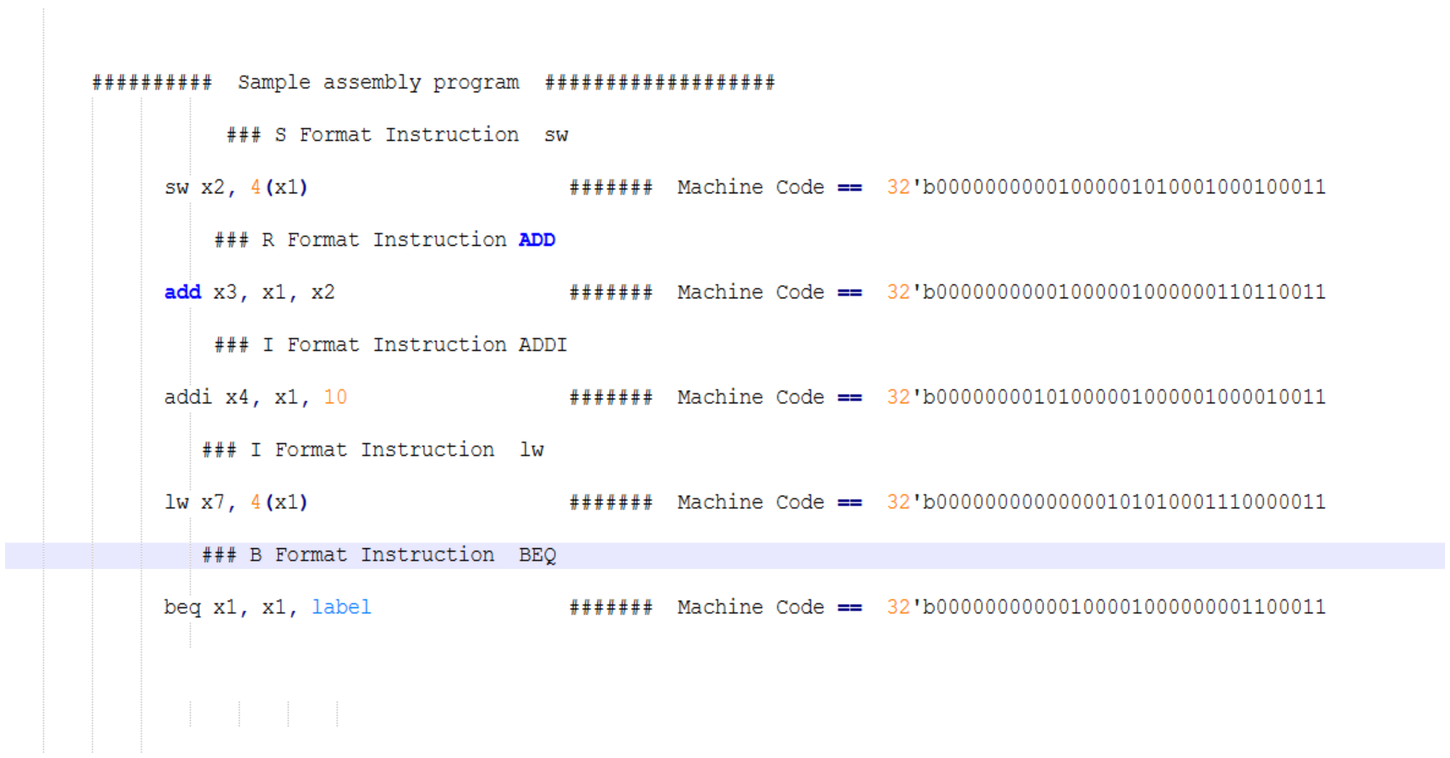
Instruction memory (ROM)



Data memory (RAM)

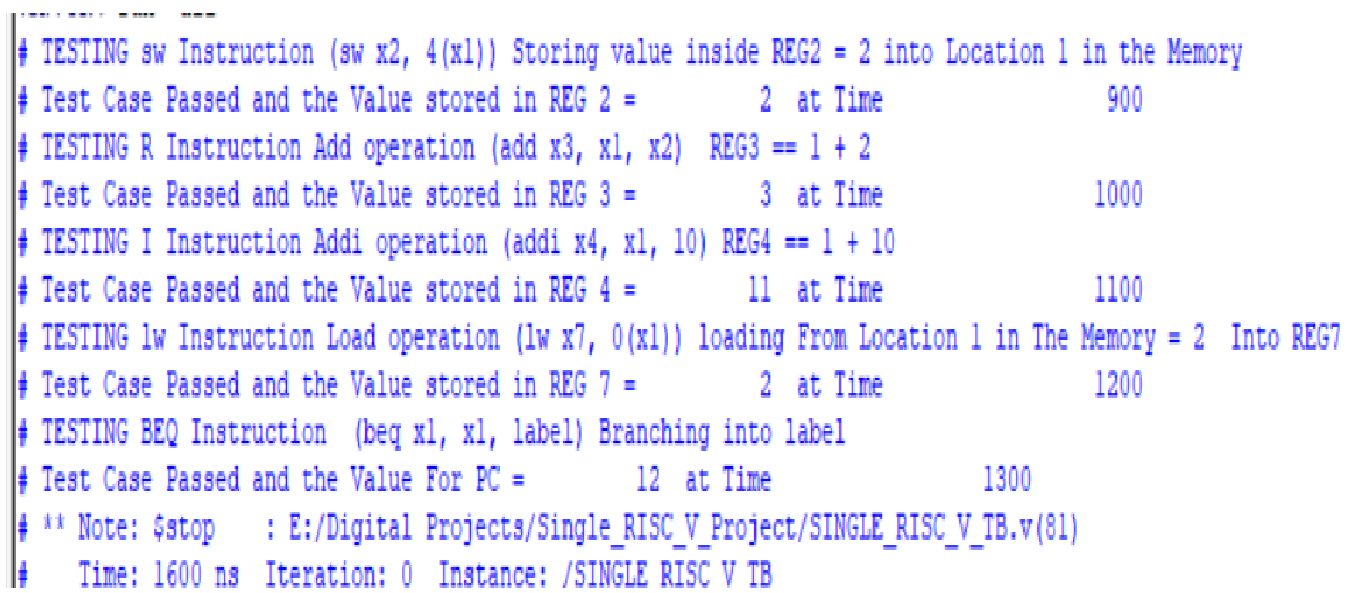


Sample assembly program



```
##########  Sample assembly program  ###################

           ### S Format Instruction  sw

      sw x2, 4(x1)                  #######  Machine Code ==  32'b00000000001000001010001000100011

          ### R Format Instruction ADD

      add x3, x1, x2                #######  Machine Code ==  32'b00000000001000001000000110110011

          ### I Format Instruction ADDI

      addi x4, x1, 10               #######  Machine Code ==  32'b00000000101000001000001000010011

         ### I Format Instruction  lw

      lw x7, 4(x1)                  #######  Machine Code ==  32'b00000000000000101010001110000011

         ### B Format Instruction  BEQ

      beq x1, x1, label             #######  Machine Code ==  32'b00000000000100001000000001100011
```

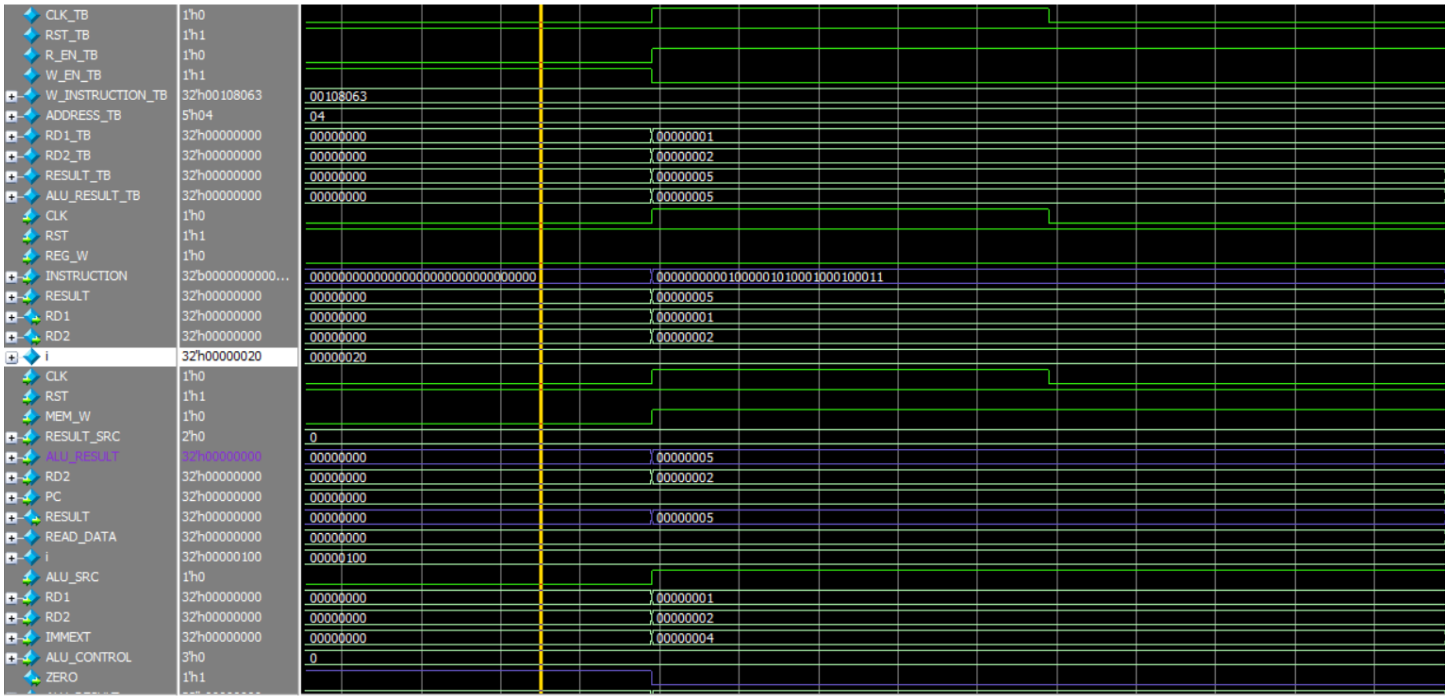
Transcript Snip Showing Passed Test Cases



```
# TESTING sw Instruction (sw x2, 4(x1)) Storing value inside REG2 = 2 into Location 1 in the Memory
# Test Case Passed and the Value stored in REG 2 =          2  at Time                   900
# TESTING R Instruction Add operation (add x3, x1, x2)  REG3 == 1 + 2
# Test Case Passed and the Value stored in REG 3 =          3  at Time                  1000
# TESTING I Instruction Addi operation (addi x4, x1, 10) REG4 == 1 + 10
# Test Case Passed and the Value stored in REG 4 =         11  at Time                  1100
# TESTING lw Instruction Load operation (lw x7, 0(x1)) loading From Location 1 in The Memory = 2  Into REG7
# Test Case Passed and the Value stored in REG 7 =          2  at Time                  1200
# TESTING BEQ Instruction  (beq x1, x1, label) Branching into label
# Test Case Passed and the Value For PC =         12  at Time                  1300
# ** Note: $stop    : E:/Digital Projects/Single_RISC_V_Project/SINGLE_RISC_V_TB.v(81)
#    Time: 1600 ns  Iteration: 0  Instance: /SINGLE_RISC_V_TB
```

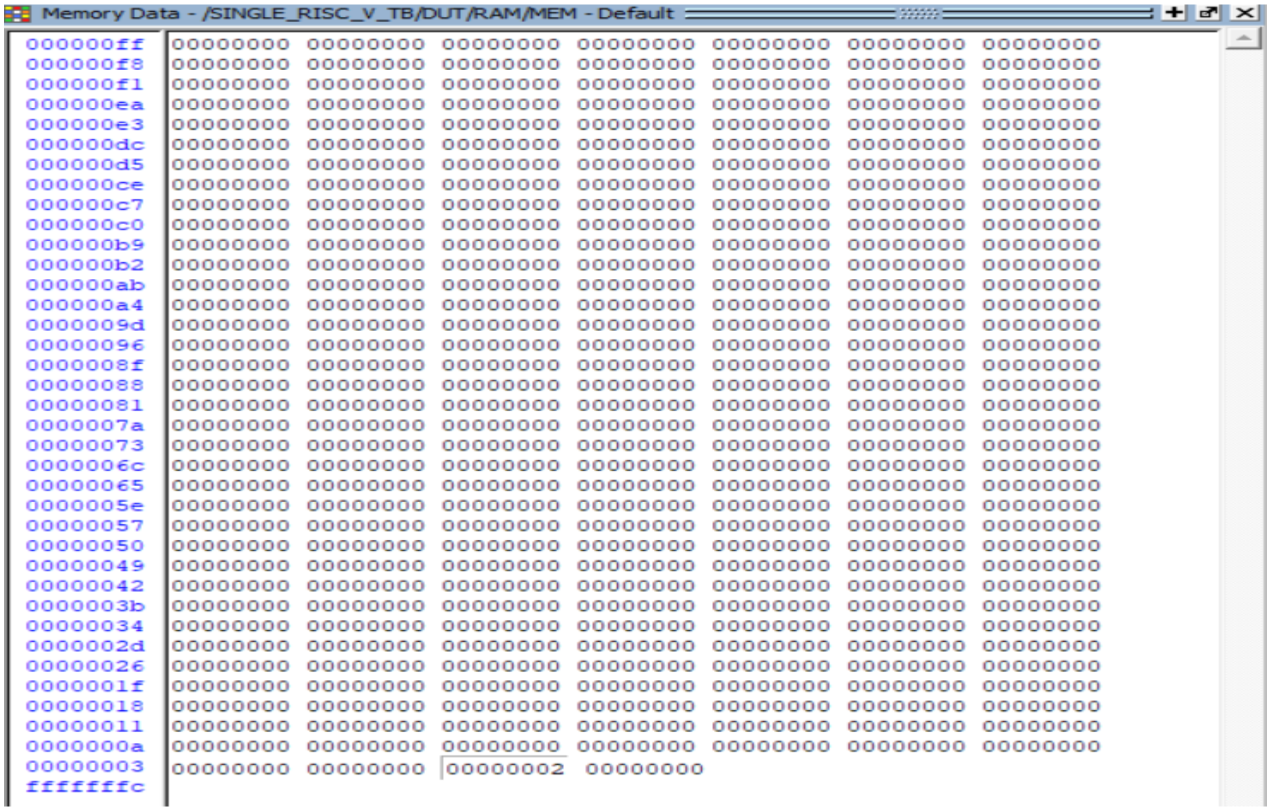
WAVE_FORM FOR SW instruction

sw x2, 4(x1)



```
CLK_TB               1'h0
RST_TB               1'h1
R_EN_TB              1'h0
W_EN_TB              1'h1
W_INSTRUCTION_TB     32'h00108063      00108063
ADDRESS_TB           5'h04             04
RD1_TB               32'h00000000      00000000   00000001
RD2_TB               32'h00000000      00000000   00000002
RESULT_TB            32'h00000000      00000000   00000005
ALU_RESULT_TB        32'h00000000      00000000   00000005
CLK                  1'h0
RST                  1'h1
REG_W                1'h0
INSTRUCTION          32'b0000000000...  00000000000000000000000000000000   00000000001000001010001000100011
RESULT               32'h00000000      00000000   00000005
RD1                  32'h00000000      00000000   00000001
RD2                  32'h00000000      00000000   00000002
i                    32'h00000020      00000020
CLK                  1'h0
RST                  1'h1
MEM_W                1'h0
RESULT_SRC           2'h0              0
ALU_RESULT           32'h00000000      00000000   00000005
RD2                  32'h00000000      00000000   00000002
PC                   32'h00000000      00000000
RESULT               32'h00000000      00000000   00000005
READ_DATA            32'h00000000      00000000
i                    32'h00000100      00000100
ALU_SRC              1'h0
RD1                  32'h00000000      00000000   00000001
RD2                  32'h00000000      00000000   00000002
IMMEXT               32'h00000000      00000000   00000004
ALU_CONTROL          3'h0              0
ZERO                 1'h1
```

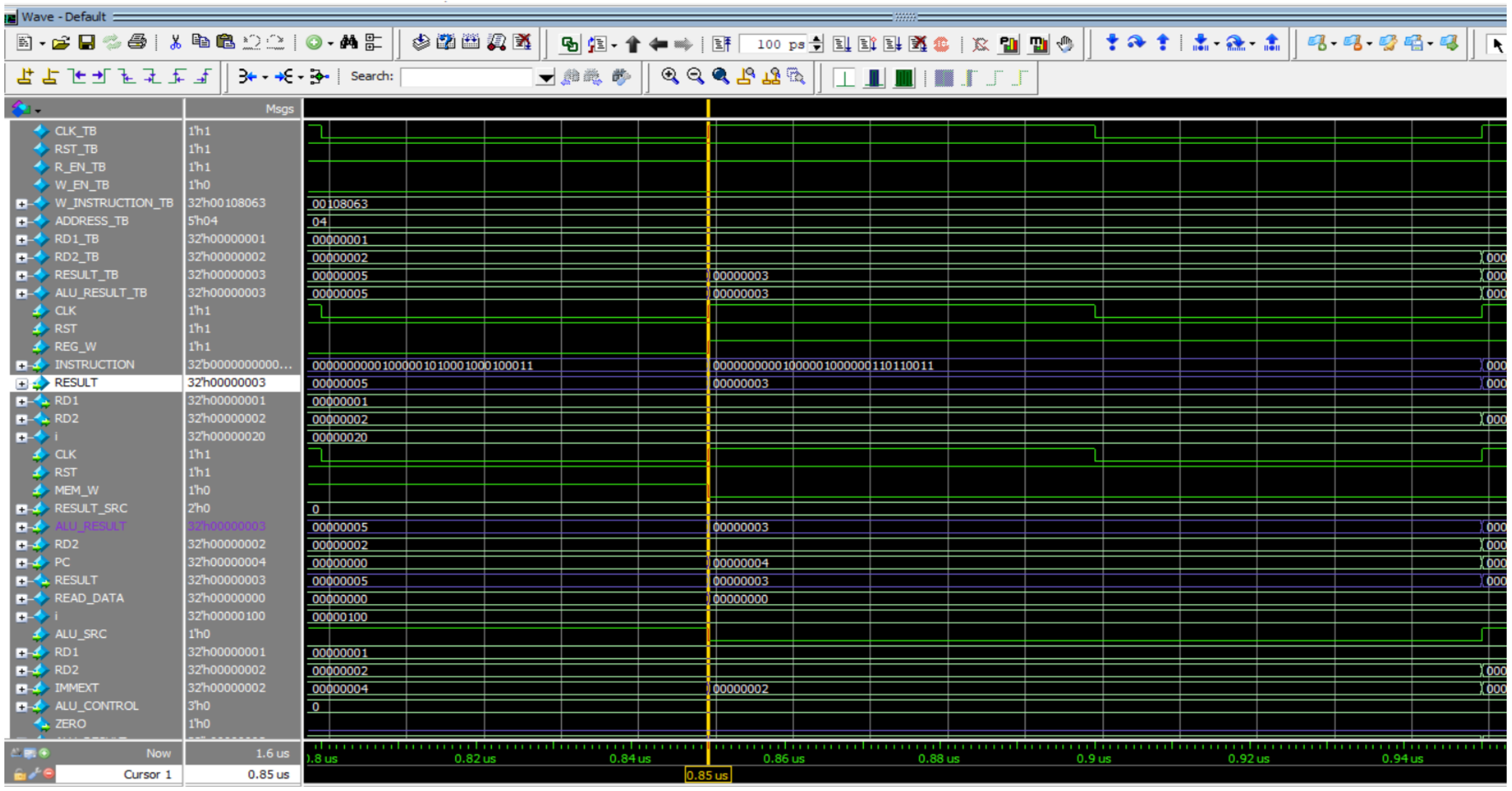
Value Stored In loctaion 1 In the Mem (Should = Value inside REG2 which is 2)



```
Memory Data - /SINGLE_RISC_V_TB/DUT/RAM/MEM - Default
000000ff  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000f8  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000f1  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000ea  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000e3  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000dc  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000d5  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000ce  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000c7  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000c0  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000b9  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000b2  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000ab  00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000a4  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000009d  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000096  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000008f  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000088  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000081  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000007a  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000073  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000006c  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000065  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000005e  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000057  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000050  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000049  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000042  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000003b  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000034  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000002d  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000026  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000001f  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000018  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000011  00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000000a  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000003  00000000 00000000 00000002 00000000
fffffffc
```

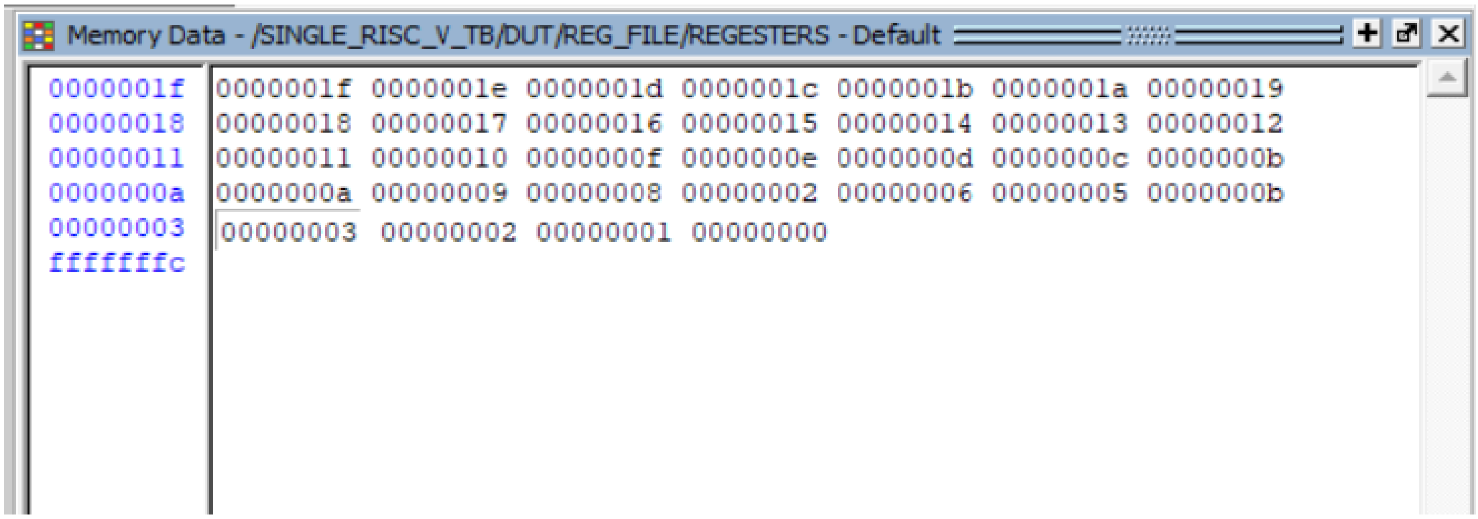
WAVE_FORM FOR ADD instruction

add x3, x1, x2



```
Wave - Default
                      Msgs
CLK_TB                1'h1
RST_TB                1'h1
R_EN_TB               1'h1
W_EN_TB               1'h0
W_INSTRUCTION_TB      32'h00108063
ADDRESS_TB            5'h04
RD1_TB                32'h00000001
RD2_TB                32'h00000002
RESULT_TB             32'h00000003
ALU_RESULT_TB         32'h00000003
CLK                   1'h1
RST                   1'h1
REG_W                 1'h1
INSTRUCTION           32'b0000000000...
RESULT                32'h00000003
RD1                   32'h00000001
RD2                   32'h00000002
i                     32'h00000020
CLK                   1'h1
RST                   1'h1
MEM_W                 1'h0
RESULT_SRC            2'h0
ALU_RESULT            32'h00000003
RD2                   32'h00000002
PC                    32'h00000004
RESULT                32'h00000003
READ_DATA             32'h00000000
i                     32'h00000100
ALU_SRC               1'h0
RD1                   32'h00000001
RD2                   32'h00000002
IMMEXT                32'h00000002
ALU_CONTROL           3'h0
ZERO                  1'h0
Now                   1.6 us
Cursor 1              0.85 us
```

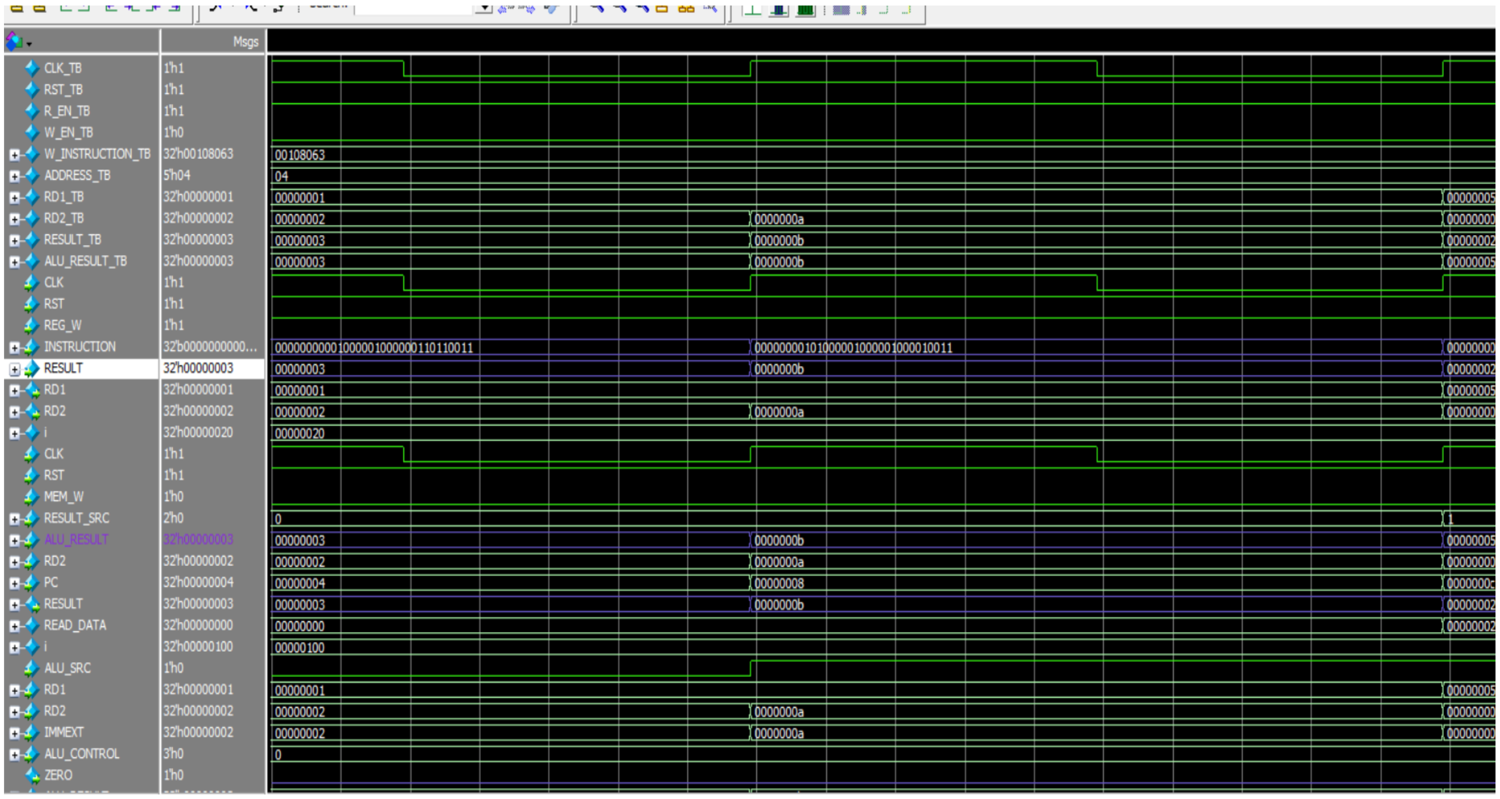
Value Stored In REG3 In the REG_FILE (Should = Value inside REG1 + REG2 = 1 + 2)



```
Memory Data - /SINGLE_RISC_V_TB/DUT/REG_FILE/REGESTERS - Default
0000001f  0000001f 0000001e 0000001d 0000001c 0000001b 0000001a 00000019
00000018  00000018 00000017 00000016 00000015 00000014 00000013 00000012
00000011  00000011 00000010 0000000f 0000000e 0000000d 0000000c 0000000b
0000000a  0000000a 00000009 00000008 00000002 00000006 00000005 0000000b
00000003  00000003 00000002 00000001 00000000
fffffffc
```

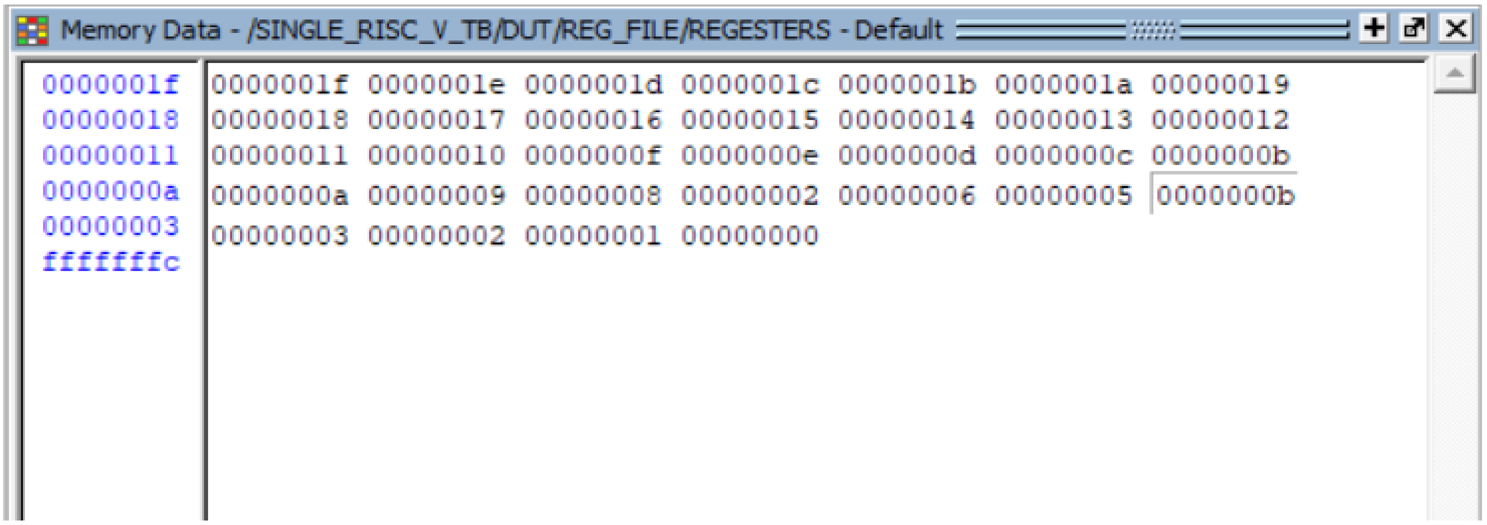
WAVE_FORM FOR ADDI instruction

addI x4, x1, 10



```
                      Msgs
CLK_TB                1'h1
RST_TB                1'h1
R_EN_TB               1'h1
W_EN_TB               1'h0
W_INSTRUCTION_TB      32'h00108063
ADDRESS_TB            5'h04
RD1_TB                32'h00000001
RD2_TB                32'h00000002
RESULT_TB             32'h00000003
ALU_RESULT_TB         32'h00000003
CLK                   1'h1
RST                   1'h1
REG_W                 1'h1
INSTRUCTION           32'b0000000000...
RESULT                32'h00000003
RD1                   32'h00000001
RD2                   32'h00000002
i                     32'h00000020
CLK                   1'h1
RST                   1'h1
MEM_W                 1'h0
RESULT_SRC            2'h0
ALU_RESULT            32'h00000003
RD2                   32'h00000002
PC                    32'h00000004
RESULT                32'h00000003
READ_DATA             32'h00000000
i                     32'h00000100
ALU_SRC               1'h0
RD1                   32'h00000001
RD2                   32'h00000002
IMMEXT                32'h00000002
ALU_CONTROL           3'h0
ZERO                  1'h0
```

Value Stored In REG4 In the REG_FILE (Should = Value inside REG1 + 10 = 1 + 10)

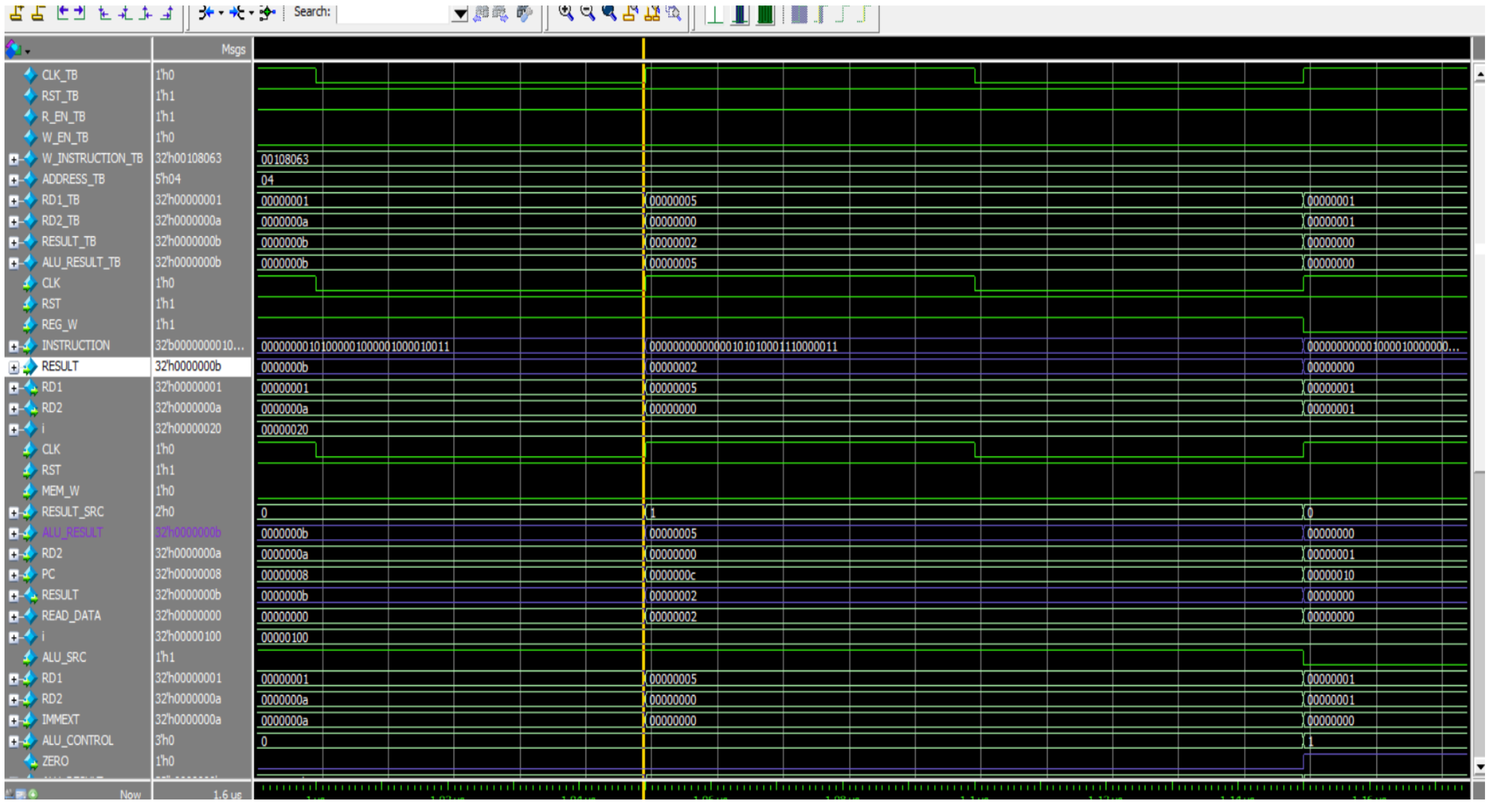


Memory Data - /SINGLE_RISC_V_TB/DUT/REG_FILE/REGESTERS - Default

```
0000001f  0000001f 0000001e 0000001d 0000001c 0000001b 0000001a 00000019
00000018  00000018 00000017 00000016 00000015 00000014 00000013 00000012
00000011  00000011 00000010 0000000f 0000000e 0000000d 0000000c 0000000b
0000000a  0000000a 00000009 00000008 00000002 00000006 00000005 0000000b
00000003  00000003 00000002 00000001 00000000
fffffffc
```

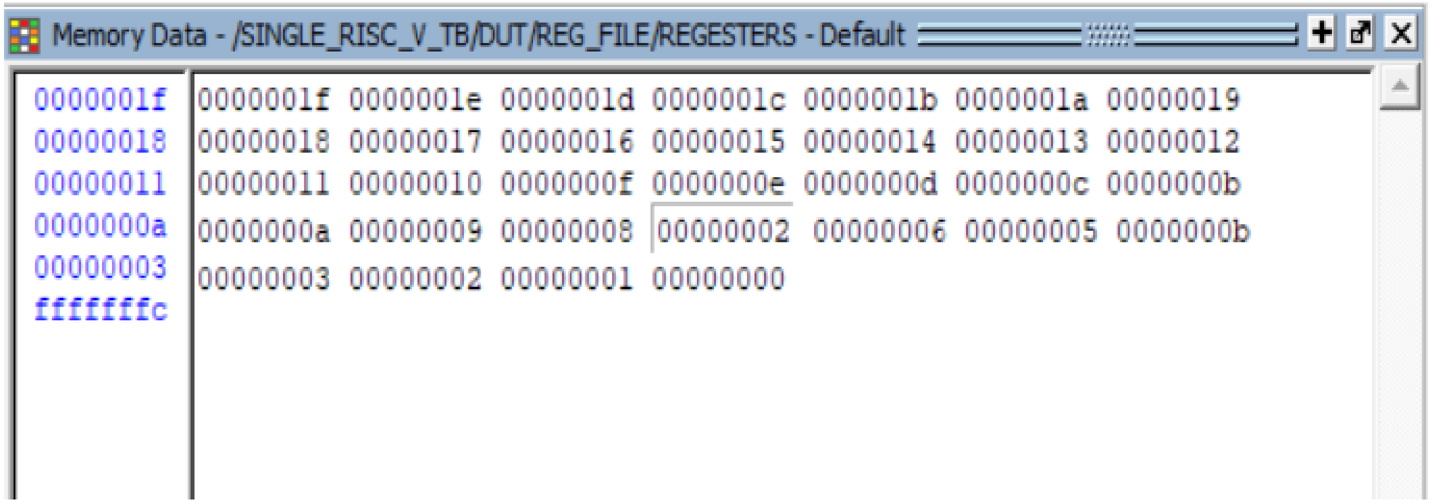
WAVE_FORM FOR LW instruction

lw x7, 4(x1)



```
                     Msgs
CLK_TB               1'h0
RST_TB               1'h1
R_EN_TB              1'h1
W_EN_TB              1'h0
W_INSTRUCTION_TB     32'h00108063    00108063
ADDRESS_TB           5'h04           04
RD1_TB               32'h00000001    00000001   00000005   00000001
RD2_TB               32'h0000000a    0000000a   00000000   00000001
RESULT_TB            32'h0000000b    0000000b   00000002   00000000
ALU_RESULT_TB        32'h0000000b    0000000b   00000005   00000000
CLK                  1'h0
RST                  1'h1
REG_W                1'h1
INSTRUCTION          32'b0000000010...  00000000101000001000001000010011   00000000000000101010001110000011   00000000000100001000000000...
RESULT               32'h0000000b    0000000b   00000002   00000000
RD1                  32'h00000001    00000001   00000005   00000001
RD2                  32'h0000000a    0000000a   00000000   00000001
i                    32'h00000020    00000020
CLK                  1'h0
RST                  1'h1
MEM_W                1'h0
RESULT_SRC           2'h0            0          1          0
ALU_RESULT           32'h0000000b    0000000b   00000005   00000000
RD2                  32'h0000000a    0000000a   00000000   00000001
PC                   32'h00000008    00000008   0000000c   00000010
RESULT               32'h0000000b    0000000b   00000002   00000000
READ_DATA            32'h00000000    00000000   00000002   00000000
i                    32'h00000100    00000100
ALU_SRC              1'h1
RD1                  32'h00000001    00000001   00000005   00000001
RD2                  32'h0000000a    0000000a   00000000   00000001
IMMEXT               32'h0000000a    0000000a   00000000   00000000
ALU_CONTROL          3'h0            0                     1
ZERO                 1'h0
Now                  1.6 us
```

Value Stored In REG7 In REG_File(Should = Loction 1 In the Memory = 2)

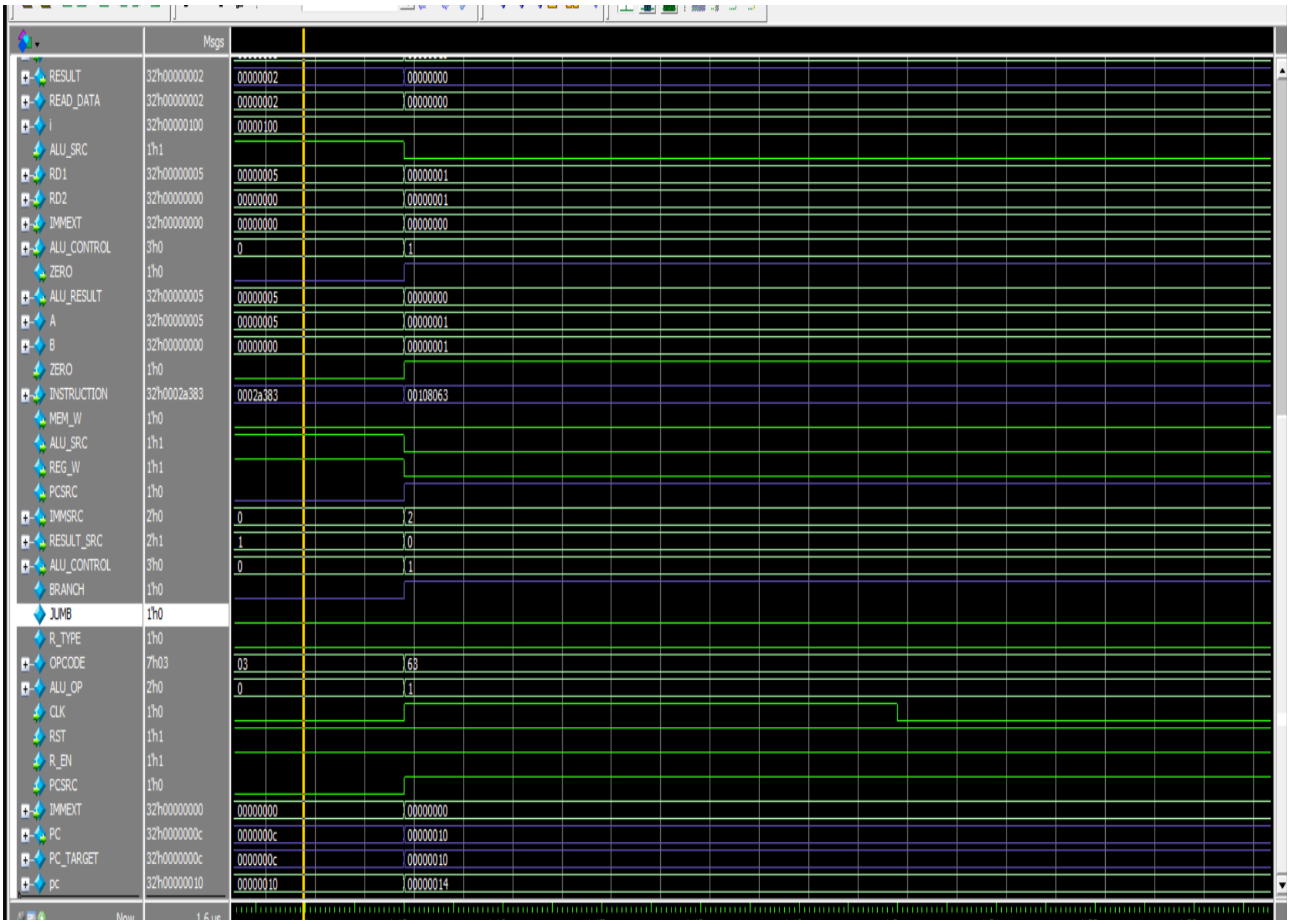


```
Memory Data - /SINGLE_RISC_V_TB/DUT/REG_FILE/REGESTERS - Default

0000001f  0000001f 0000001e 0000001d 0000001c 0000001b 0000001a 00000019
00000018  00000018 00000017 00000016 00000015 00000014 00000013 00000012
00000011  00000011 00000010 0000000f 0000000e 0000000d 0000000c 0000000b
0000000a  0000000a 00000009 00000008 00000002 00000006 00000005 0000000b
00000003  00000003 00000002 00000001 00000000
fffffffc
```

WAVE_FORM FOR BEQ instruction

beq x1, x1, label

Value Stored In PC In PC_COUNTER(Should = PC_TARGET = 0X000C) && PCSRC = 1



	Msgs		
RESULT	32'h00000002	00000002	00000000
READ_DATA	32'h00000002	00000002	00000000
i	32'h00000100	00000100	
ALU_SRC	1'h1		
RD1	32'h00000005	00000005	00000001
RD2	32'h00000000	00000000	00000001
IMMEXT	32'h00000000	00000000	00000000
ALU_CONTROL	3'h0	0	1
ZERO	1'h0		
ALU_RESULT	32'h00000005	00000005	00000000
A	32'h00000005	00000005	00000001
B	32'h00000000	00000000	00000001
ZERO	1'h0		
INSTRUCTION	32'h0002a383	0002a383	00108063
MEM_W	1'h0		
ALU_SRC	1'h1		
REG_W	1'h1		
PCSRC	1'h0		
IMMSRC	2'h0	0	2
RESULT_SRC	2'h1	1	0
ALU_CONTROL	3'h0	0	1
BRANCH	1'h0		
JUMB	1'h0		
R_TYPE	1'h0		
OPCODE	7'h03	03	63
ALU_OP	2'h0	0	1
CLK	1'h0		
RST	1'h1		
R_EN	1'h1		
PCSRC	1'h0		
IMMEXT	32'h00000000	00000000	00000000
PC	32'h0000000c	0000000c	00000010
PC_TARGET	32'h0000000c	0000000c	00000010
pc	32'h00000010	00000010	00000014