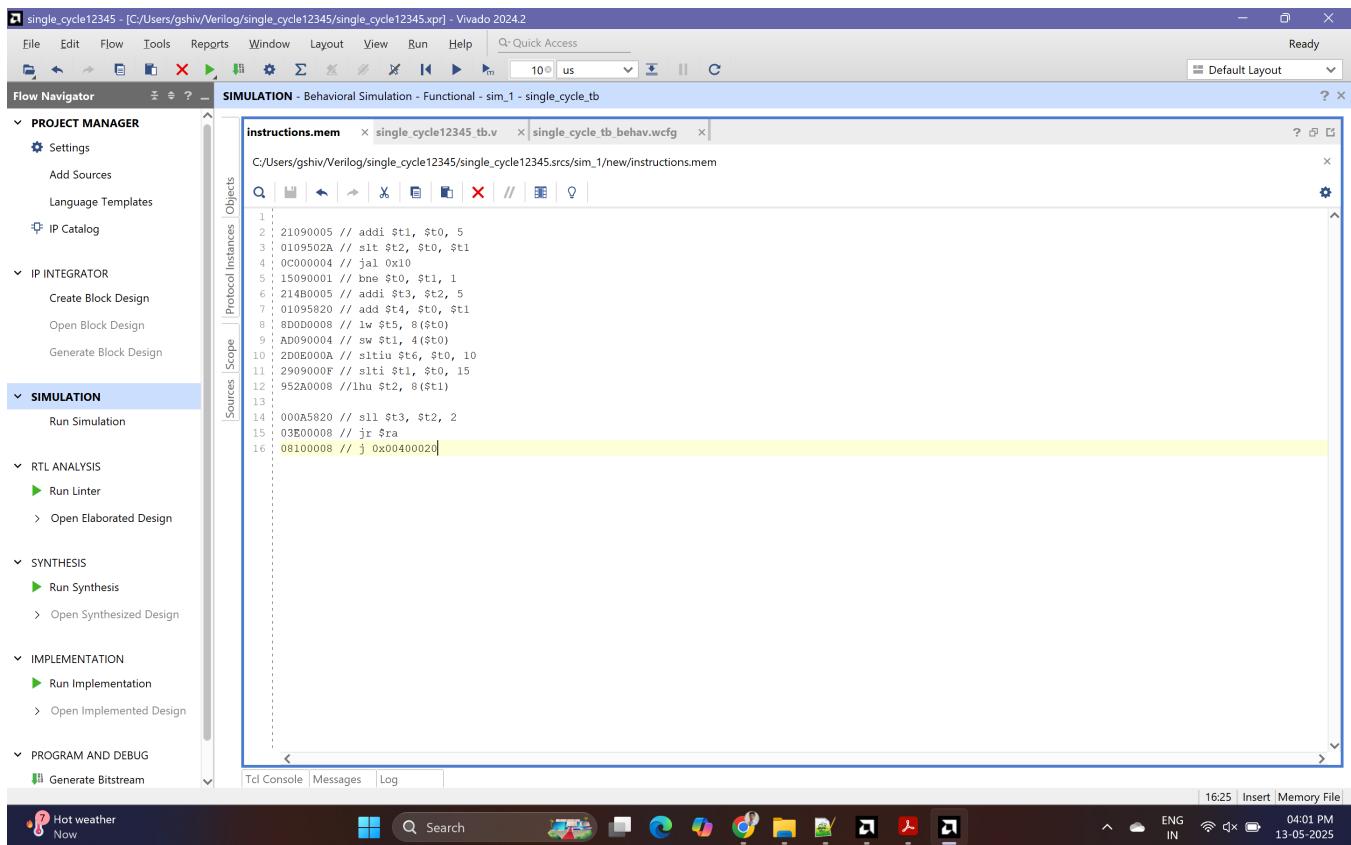


Computer Organisation Architecture Project Report

Shivam Gupta, 2024PVL0120
 M.tech VLSI 2024-2026, Electrical Department
 Indian Institute of Technology, Jammu

I Instructions



The screenshot shows the Vivado 2024.2 interface with the project 'single_cycle12345' open. The left sidebar contains the 'PROJECT MANAGER' with sections for Settings, Add Sources, Language Templates, IP Catalog, IP INTEGRATOR, Create Block Design, Open Block Design, Generate Block Design, and SIMULATION. Under SIMULATION, 'Run Simulation' is selected. The main workspace shows the 'SIMULATION - Behavioral Simulation - Functional - sim_1 - single_cycle_tb' window. Inside, the 'instructions.mem' file is displayed, containing the following assembly-like code:

```

1: 21090005 // addi $t1, $t0, 5
2: 0109502A // slt $t2, $t0, $t1
3: 0C000004 // jal 0x10
4: 15090001 // bne $t0, $t1, 1
5: 214B0005 // addi $t3, $t2, 5
6: 01095820 // add $t4, $t0, $t1
7: 8DD00008 // lw $t5, 8($t0)
8: AD090004 // sw $t1, 4($t0)
9: 2D0E000A // sltiu $t6, $t0, 10
10: 2909000F // sli $t1, $t0, 15
11: 952A0008 // lhu $t2, 8($t1)
12: 000A5820 // sll $t3, $t2, 2
13: 03E00000 // jr $ra
14: 08100008 // j 0x00400020
15:
16:

```

Fig. 1: Instructions

same instructions, memory file or set is used for multi cycle too.

(1) 20080003 - addi \$t0,\$zero,3 (I type)

0010 0001 0000 1000 0000 0000 0000 0011
 opcode TS rt immediate 16 bits
 8 8 9 3
 \$0 \$t0

(2) 21090005 - addi \$t1,\$t0,5

0010 0001 0000 1001 0000 0000 0000 0101
 opcode \$t0 \$t1 immediate
 8 8 9 5 (16 bits)

(3) 0109502A - slt \$t2,\$t0,\$t1

0000 0001 0000 100 | 0101 0000 0010 1010
 opcode rs rt rd shamt function
 0 \$t0 \$t1 \$t2 0 42
 8 9 10

(4) 0C006004 - jal 0x10

0000 1100 0000 0000 0000 0000 0000 1000
 opcode Target address
 3-decimal 4
 26 bits

(5) 15090001 - bne \$t0,\$t1,1

0001 0101 0000 1001 0000 0000 0000 0001
 opcode rs rt immediate
 5 \$t0 \$t1 1
 decimal value 8 9

(6) 214B0005 - addi \$t3,\$t2,5

0010 0001 0100 1011 0000 0000 0000 0101
 opcode rs rt immediate
 8 \$t2 \$t3 5
 decimal value 10 11
 16 bits

(7) 01095820 - add \$t4,\$t0,\$t1

0000 0001 0000 1001 0101 1000 0010 0000
 opcode rs rt rd shamt function
 0 \$t0 \$t1 \$t4 0 32
 decimal value 8 9 12

Fig. 2: Instruction Decode

(6) 8D0D0008 - lw \$t5, 8(\$t0)

1000 1101 0000 1101 0000 0000 0000 1000
 opcode rs rt immediate
 35 \$t0 \$t5 8
 (decimal) 8 13

(7) A0090004 - sw \$t1, 4(\$t0)

1010 1101 0000 1001 0000 0000 0000 0100
 opcode rs rt immediate
 43 \$t0 \$t1 4
 (decimal) 8 9

(8) 2D0E000A - sltiu \$t6, \$t0, 10

0010 1101 0000 1110 0000 0000 0000 1010
 opcode rs rt immediate
 11 \$t0 \$t6 10
 (decimal) 9 14

(9) 2909000F - slti \$t1, \$t0, 15

001010 01000 01001 0000 0000 0000 1111
 opcode rs rt immediate
 10 \$t0 \$t1 15
 (decimal) 8 9

(10) 952A0008 - lhu \$t2, 8(\$t1)

100101 01001 01000 0000 0000 0000 1000
 opcode rs rt immediate
 37 \$t1 \$t2 8
 (decimal) 9 10

(11) 08100008 - j 0x00400020

000010 0001 000000000000 00000001000
 opcode 2 26-bit address
 (PC = 0x00400020)

(12) 000A5820 - SLL \$t3, \$t2, 2

000000 00000 01010 01011 00010 0000000
 opcode rs rt rd funct
 0 0 \$t2 \$t3 2 0
 (decimal) 0 0 11 11 2 0

(13) 03E00009

000000 1111 00000 00000 00000 0010000
 opcode \$t9 31
 0
 (PC = \$t9a)

Fig. 3: Instruction Decode

II Single Cycle

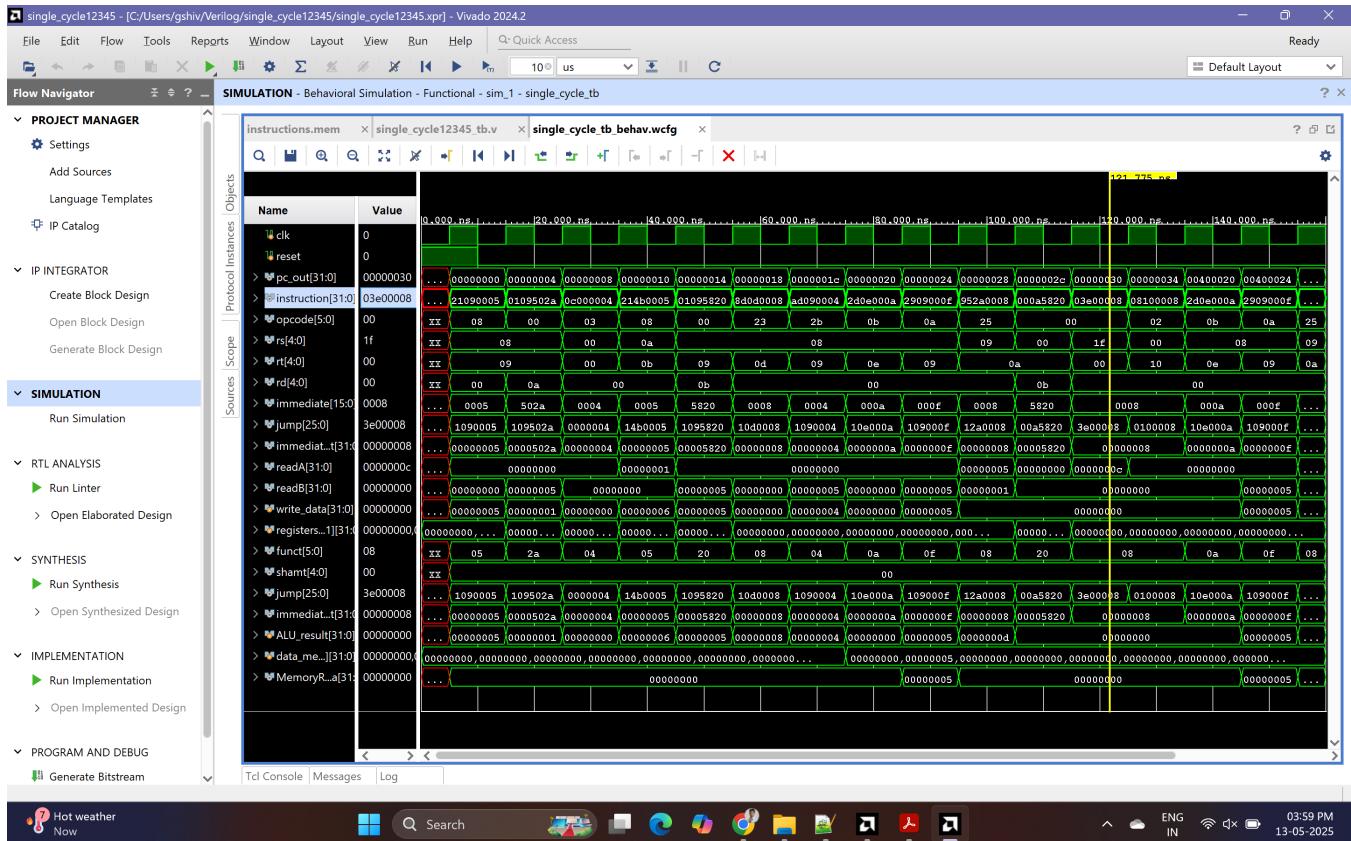


Fig. 4

III Multi Cycle

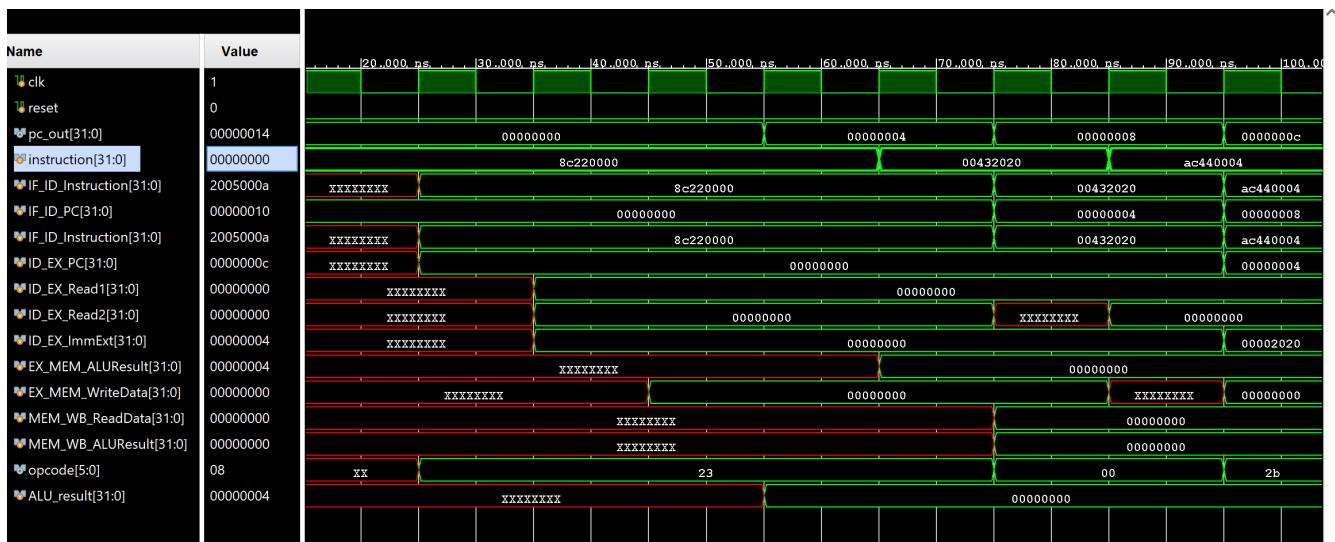


Fig. 5: Output Waveform part1

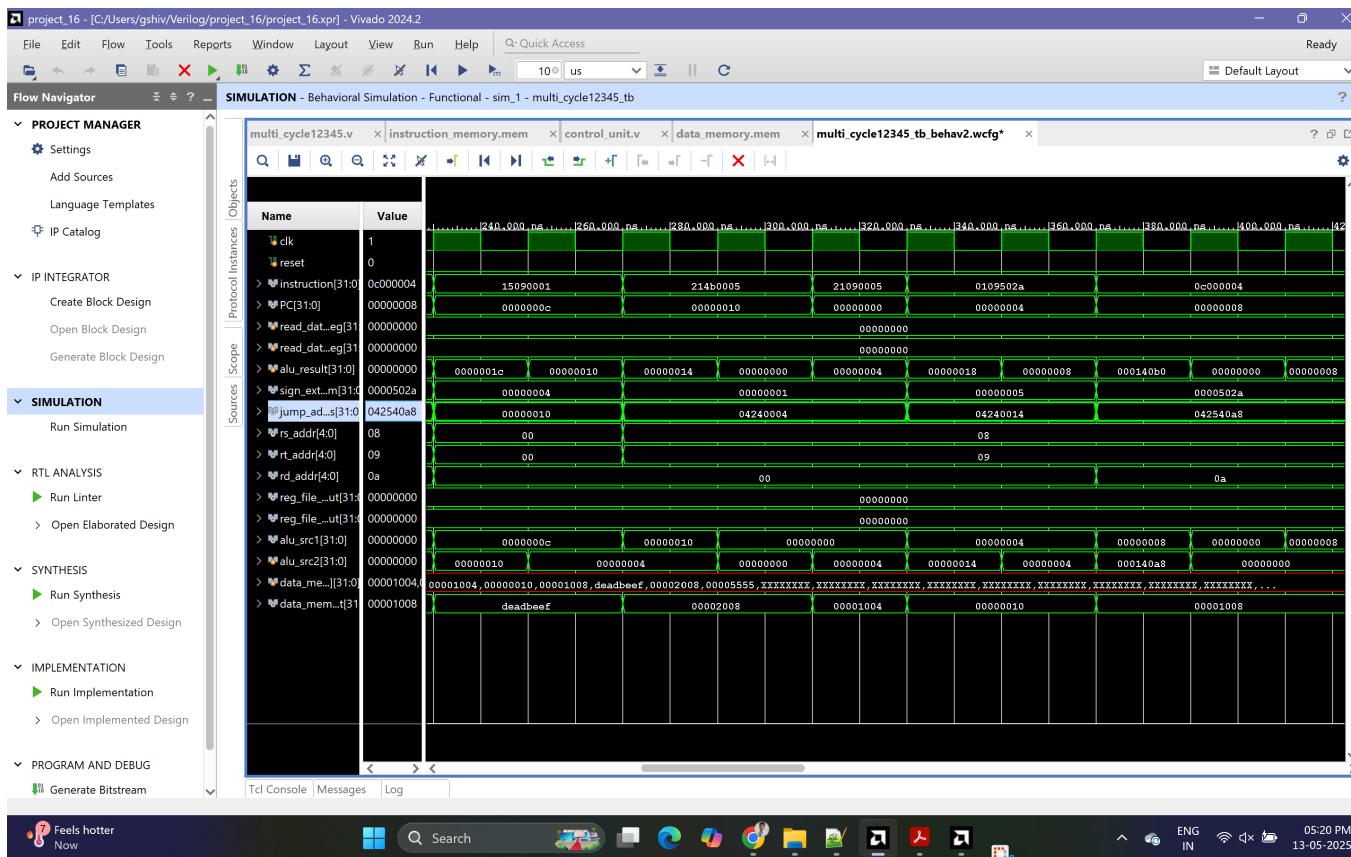


Fig. 6: Output Waveform part 2

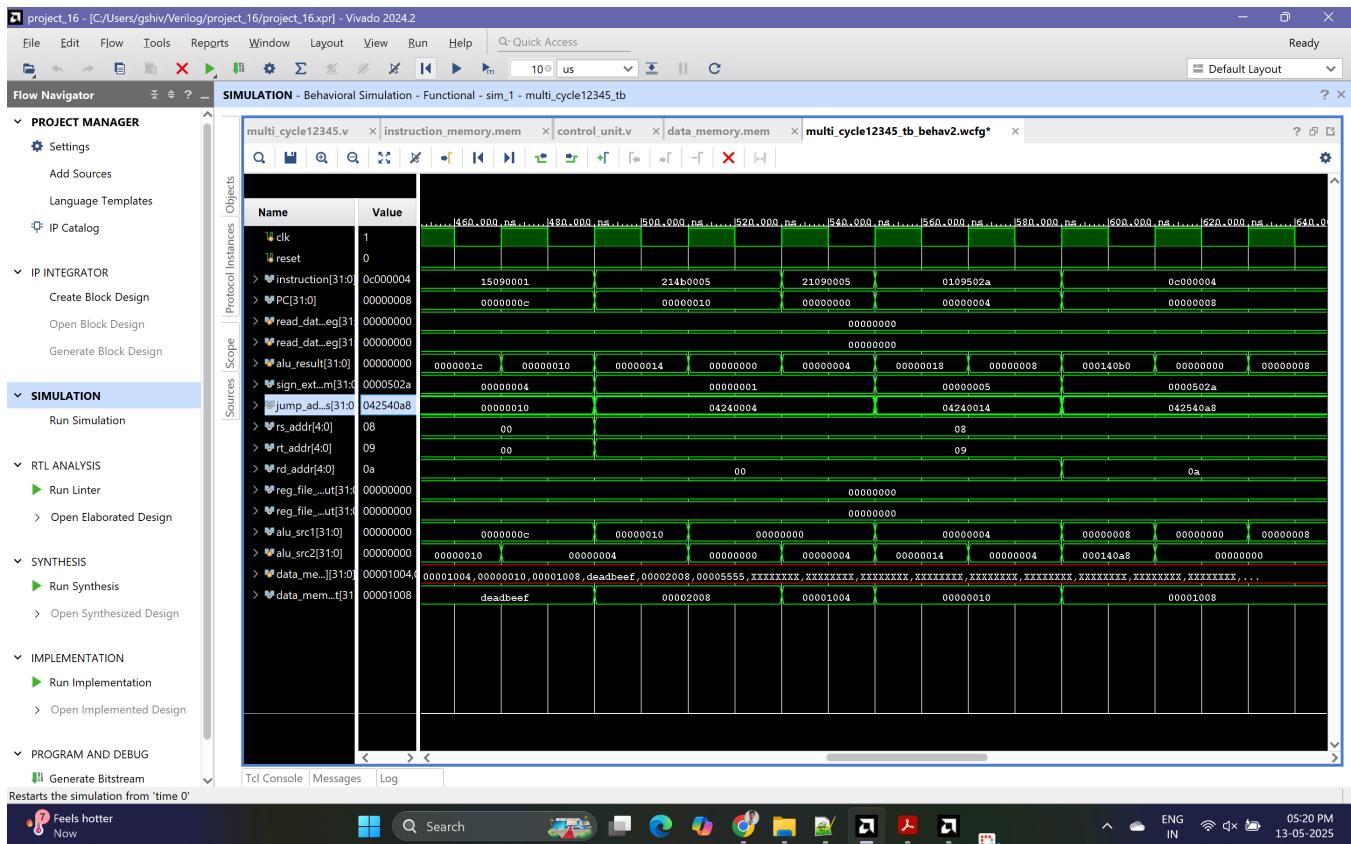


Fig. 7: Output waveform part 3

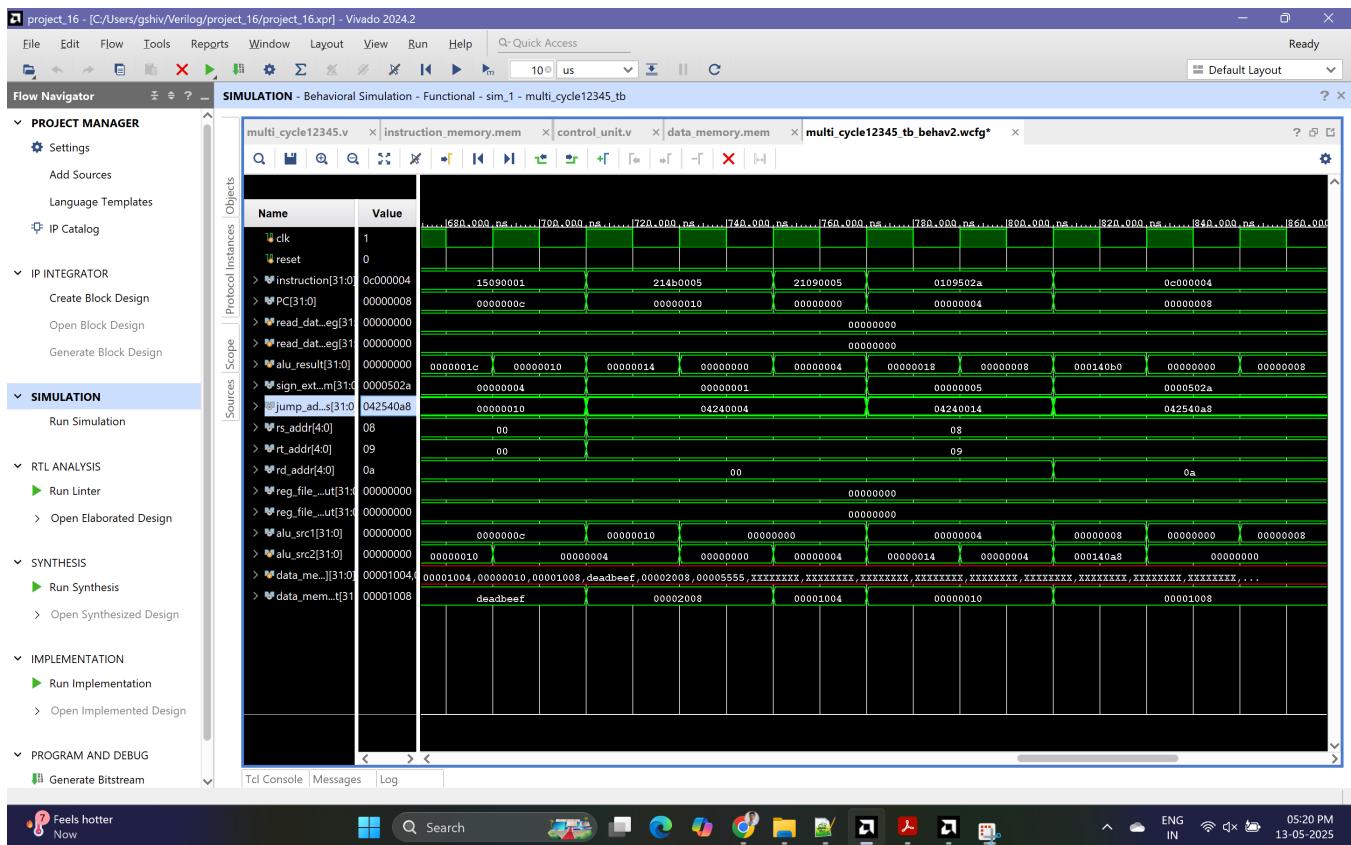


Fig. 8: Output waveform part 4

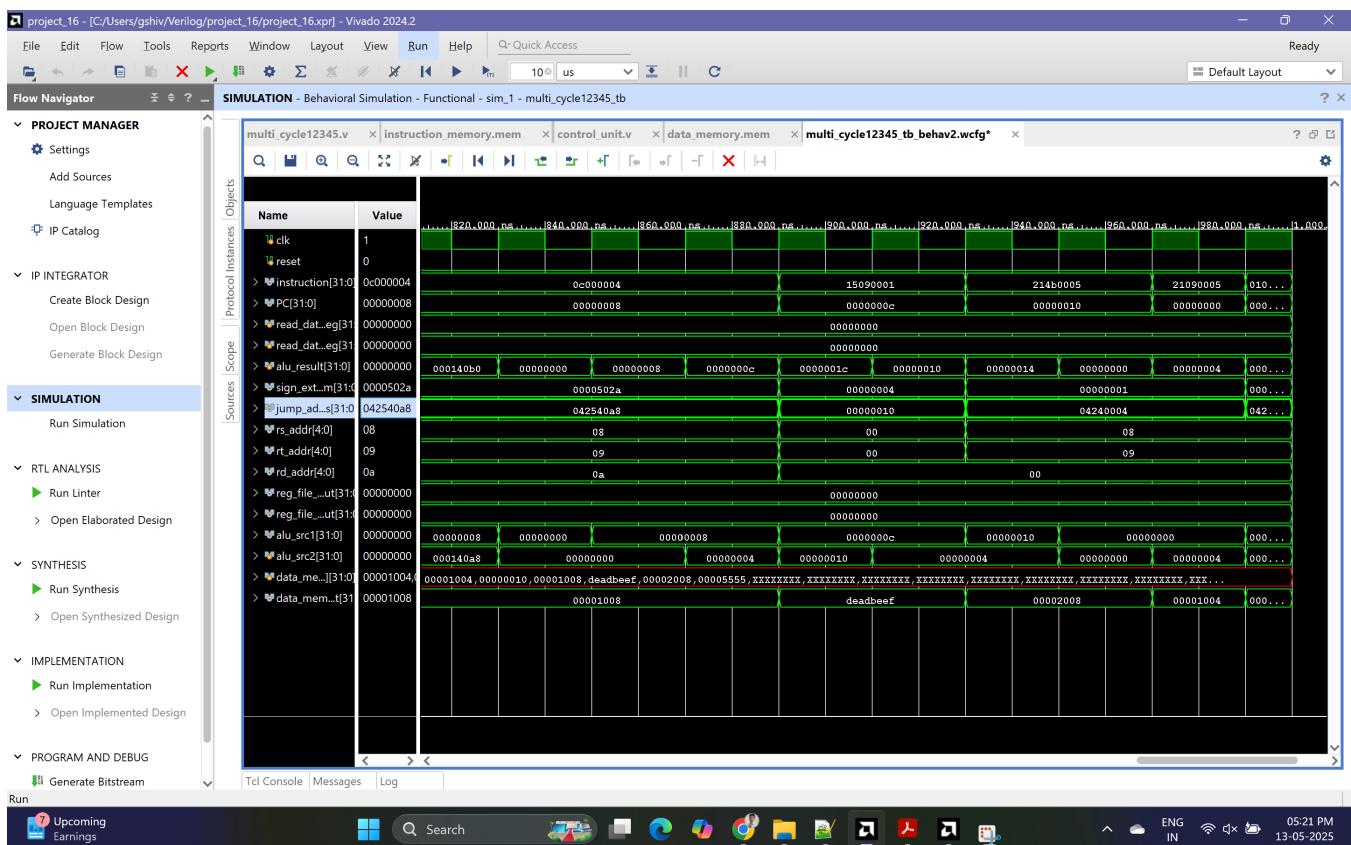


Fig. 9: Output waveform part 5

IV Pipeline

1	20010005	# addi \$1, \$0, 5
2	2002000A	# addi \$2, \$0, 10
3	00221820	# add \$3, \$1, \$2
4	00612022	# sub \$4, \$3, \$1
5	00622824	# and \$5, \$3, \$2
6	00853025	# or \$6, \$4, \$5
7		

Fig. 10: Instructions Used in Pipeline to demonstrate data forwarding

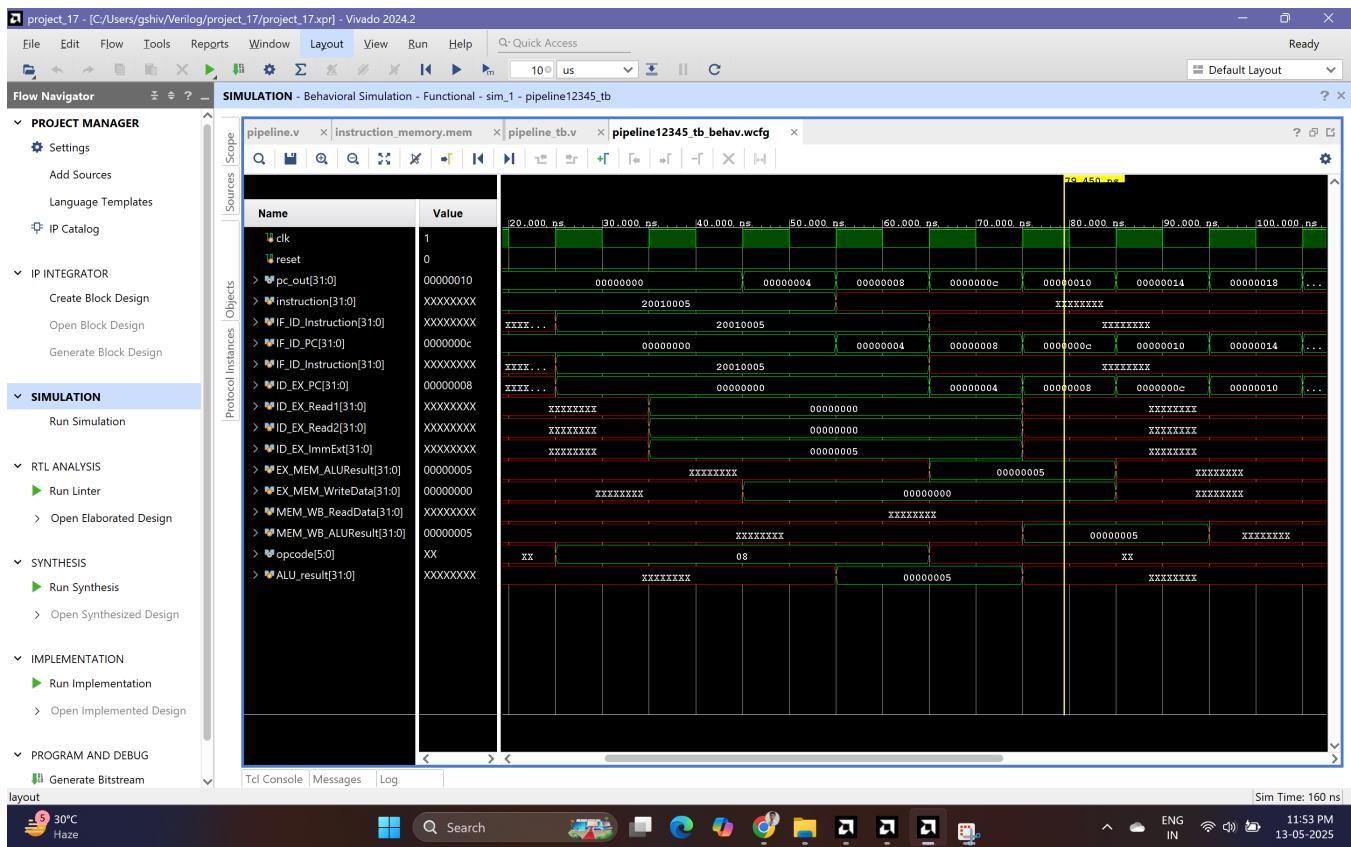


Fig. 11: Output waveform

```

1 8c220000 // lw $2, 0($1)
2 00432020 // add $4, $2, $3
3 ac440004 // sw $4, 4($2)
4 00000000 // nop
5

```

Fig. 12: Instructions used for stalling demonstration

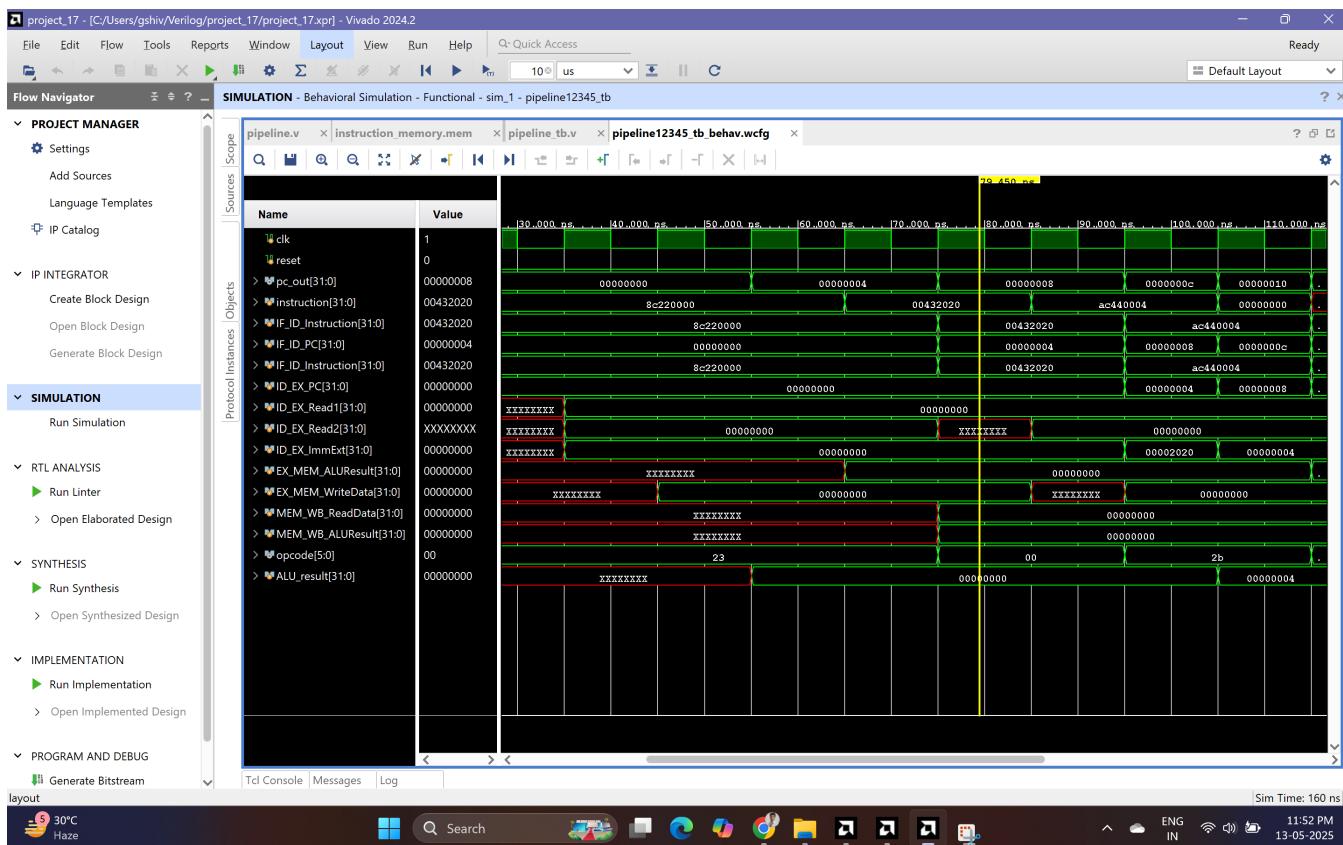


Fig. 13: Ouput waveform