Contents

1. Introduction

The main objective of this project is to test an existing design called Calc2. The test should be done through a verification system that consists of different blocks which work together in order to perform the required verify tasks. In our system, we have built it with the essential blocks, which are the transaction, generator, driver, interface, monitor, and the scoreboard and checker. In brief, the generator will generate the random values to be tested and send them through mailbox to the driver, and then the driver will be responsible for driving the signals into the Design Under Test (DUT). Lastly, the monitor will sample the actual output from the DUT along with input values to be calculated and compared in the scoreboard/checker.

The design consists of two internal arithmetic pipelines, one for add/shift, and the other one for shift left/right. The command input bus for each port will choose the operation that should be executed on the two inserted operands of data for that port. In this design, the possible operations for inserted operands 1 and 2 can be either adding, subtracting, shifting left, or shifting right. Beside all mentioned inputs, there is also the clock and the reset. The clock is used to organize setting and inserting the values as required, and the reset is required to reset the internal state of the design. In addition to that, there is the tag input that should be set in order to correlate the responses to the correct command.

On the other hand, the outputs of the design consist of three busses for each port. One bus will give the output result, the other to show response data, and the tag out that correlates to the response of correct command. And in this design, the response will represent either no response, successful response, overflow/underflow, or unused response value.

In order to test the design, a detailed test plan, test cases, and spreadsheet will be represented in the upcoming sections in order to clarify the results.

2. Test Plan

The test plan for this design is mainly focusing on randomized testing. Randomize testing is very efficient way to test a design since it can cover many possibilities, which in return will lead in discovering more unanticipated errors and problems in a system. The system will wait until the output is calculated before sending new input operands for all port at the same time.

In Calc2, the input data operands are randomized for all ports on a wide range of possibilities, we have limited the input data operands to be between 0 and 32’hFFFFFFFF in order to avoid having many overflow cases. In addition to that, the commands are randomized to be on a constraint for two scenarios, one scenario is for invalid commands, and the other is with the valid commands which will be the scenario to focus on for analyzing the bugs. Moreover, the tag

reference is one of the required things to be set in the verification, so we have planned to give tags for the addition command as a 2'b00, subtract as a 2'b01, shift left as a 2'b10, and shift right 2'b11. Furthermore, the reset is a parameter to be highlighted in the plan by checking the output after resetting the DUT as required in the specification; therefore, it is one of the aspects that will be analyzed in the test too. Another scenario in the test plan is to test the behavior of the system when only sending specific command to all port at a time without randomizing the arithmetic requests. For example, one scenario will send only addition command to all ports for all iterations and so on.

The table below summarizes the essential test points that will be discussed later along with section numbers.

Specification/Feature	Reference	Test Points	Test Scenarios	Expected Results
Invalid Command	Section 3.0	Random testing with invalid commands for port 1	Testing all possible invalid commands	1-Output of 0 with response '10'b
	Section 4.0	Random testing with invalid commands for port 2		
	Section 5.0	Random testing the result with invalid commands for port 3		
	Section 6.0	Random testing with invalid commands for port 4		
Add	Section 3.1	Random testing the addition for port 1	Random values for: 1-Addition without overflow 2-Addition with overflow	1-Without overflow, output as calculated in scoreboard and response '01'b and tag '00'b. 2-With overflow, output is disregarded and response '10'b and tag 2'b00
	Section 4.1	Random testing the addition for port 2		
	Section 5.1	Random testing the addition for port 3		
	Section 6.1	Random testing the addition for port 4		
Subtract	Section 3.2	Random testing the subtraction for port 1	Random values for: 1-Subtraction without underflow	1-Without underflow, output as calculated in scoreboard and response '01'b and tag '01'b.
	Section 4.2	Random testing the subtraction for port 2		
	Section 5.2	Random testing the subtraction for port 3		

	Section 6.2	Random testing the subtraction for port 4	2-Subtraction with underflow	2-With underflow, output is disregarded and response '10'b with tag 2'b01.
Shift Left	Section 3.3.1	Random testing left shifting for port 1	Left shift	1- Output as calculated in scoreboard, with response '01'b with tag 2'b10
	Section 4.3.1	Random testing left shifting for port 2		
	Section 5.3.1	Random testing left shifting for port 3		
	Section 6.3.1	Random testing left shifting for port 4		
Shift Right	Section 3.3.2	Random testing the right shifting for port 1	Right shift	1- Output as calculated in scoreboard, with response '01'b with tag 2'b11
	Section 4.3.2	Random testing the right shifting for port 2		
	Section 5.3.2	Random testing the right shifting for port 3		
	Section 6.3.2	Random testing the right shifting for port 4		
Testing specific commands without randomizing	Section 7	Testing result without randomizing commands	Testing the results with only addition/subtract /shift left/shift right	Output should be as in scoreboard

3. Port 1 Test Results

The following sections have the results for all repeated cases in port1.

3.0 Invalid command

Figure 1 below shows simulation for random invalid command.

```
[CHECKING VALUES IN SCOREBOARD]

INPUT VALUES
PORT_1 ::: Data_in1_1 = fd81d242 , PORT_2 ::: Data_in2_1= 183fe066, PORT_3 ::: Data_in3_1= 7bfec0ed , PORT_4 ::: Data_in4_1= fca84b6d
PORT_1 ::: Data_in1_2 = e552da8c , PORT_2 ::: Data_in2_2= 78b4edd2, PORT_3 ::: Data_in3_2= 4409022 , PORT_4 ::: Data_in4_2= ecd02a40
Cmd1=1001,                         Cmd2= 0111,                       Cmd3 =1010,                        Cmd4 =1001

OUTPUT VALUES
                1650port1:output::0,  port2:output::0, port3:output::0, port4:output::0
                1650port1:resp::2,       port2:resp::2,  port3:resp::2,    port4:resp::2
```

Figure 1: Invalid command

3.1 Addition

3.1.1 Addition without overflow

Figure 2 and 3 below shows simulation for addition of two operands without having overflow.

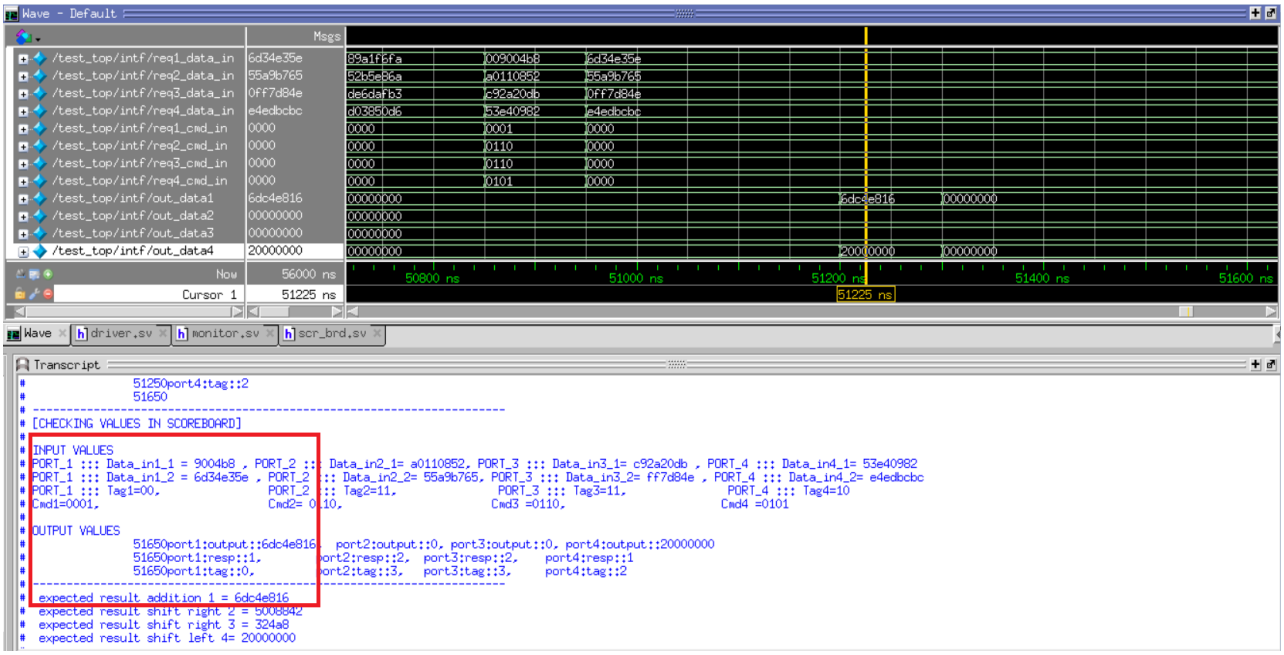

Figure 2: Addition without overflow

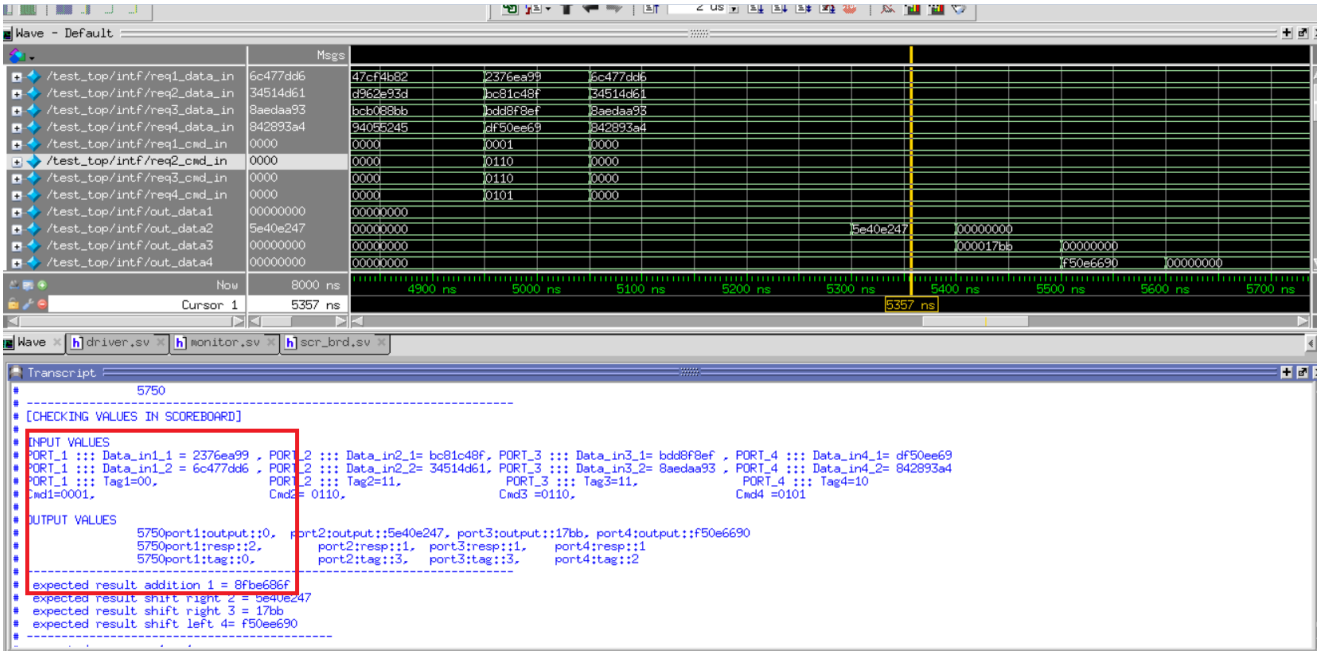

Figure 3: Addition without overflow

3.1.2 Addition with overflow

Figure 4 below shows simulation for addition of two operands with overflow.

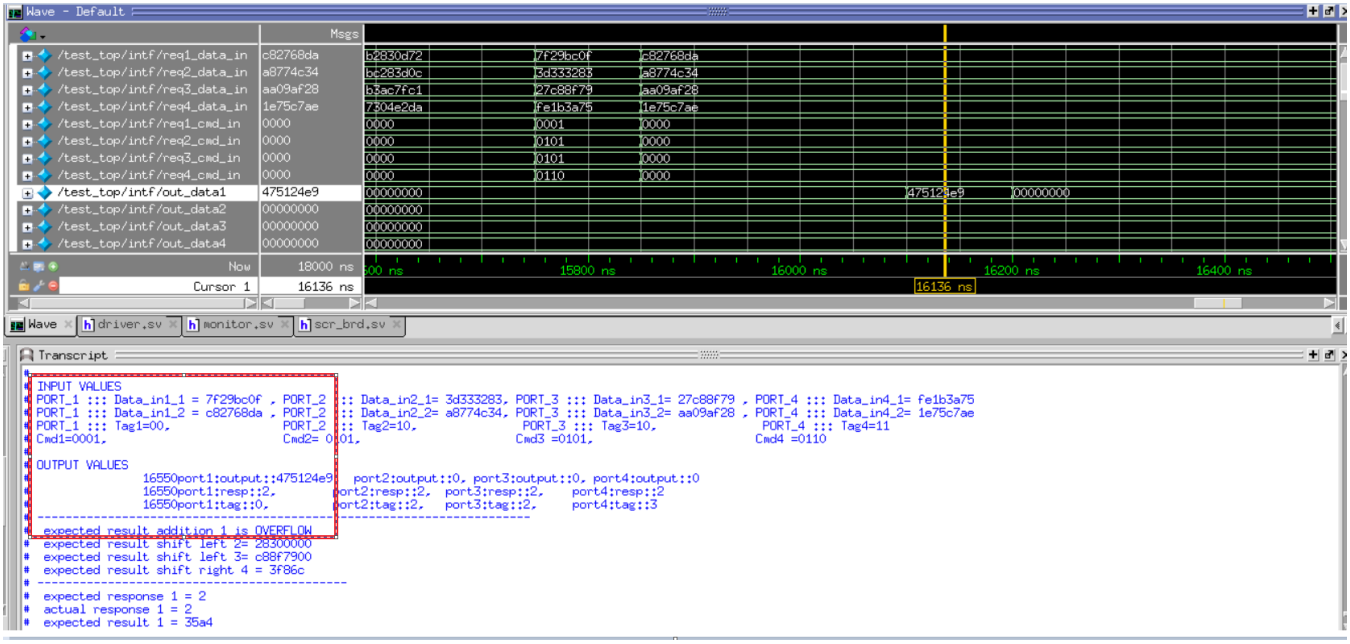

Figure 4: Addition with overflow

3.2 Subtraction

3.2.1 Subtraction without underflow

Figure 5 and 6 below shows simulation for subtraction of two operands without underflow.

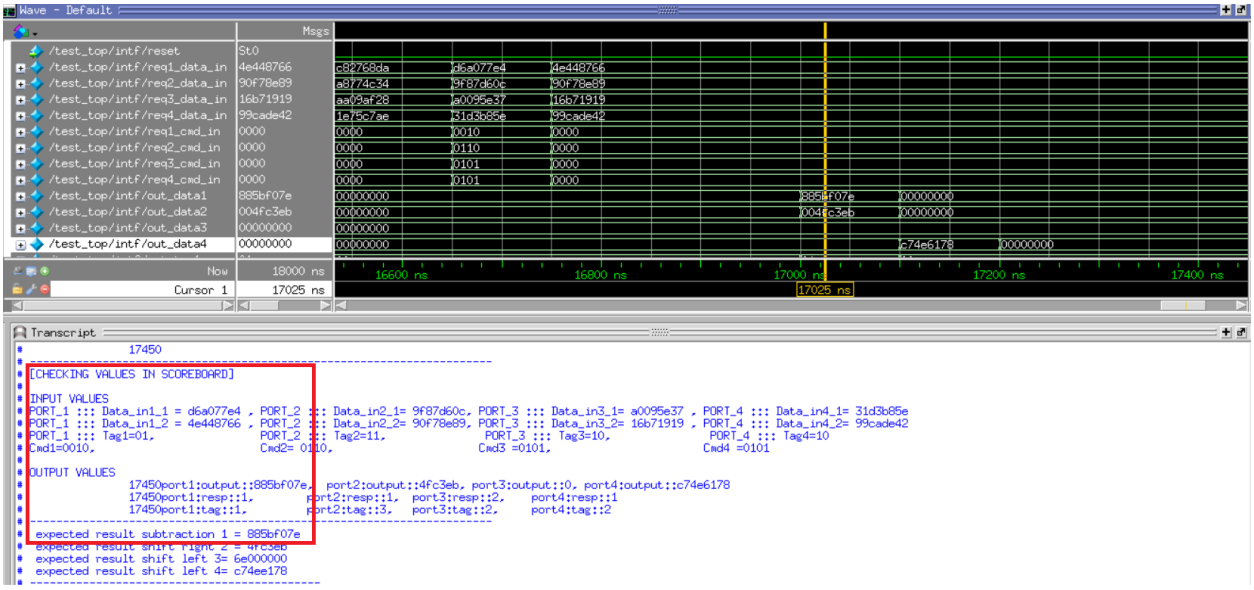

Figure 5: Subtraction without underflow

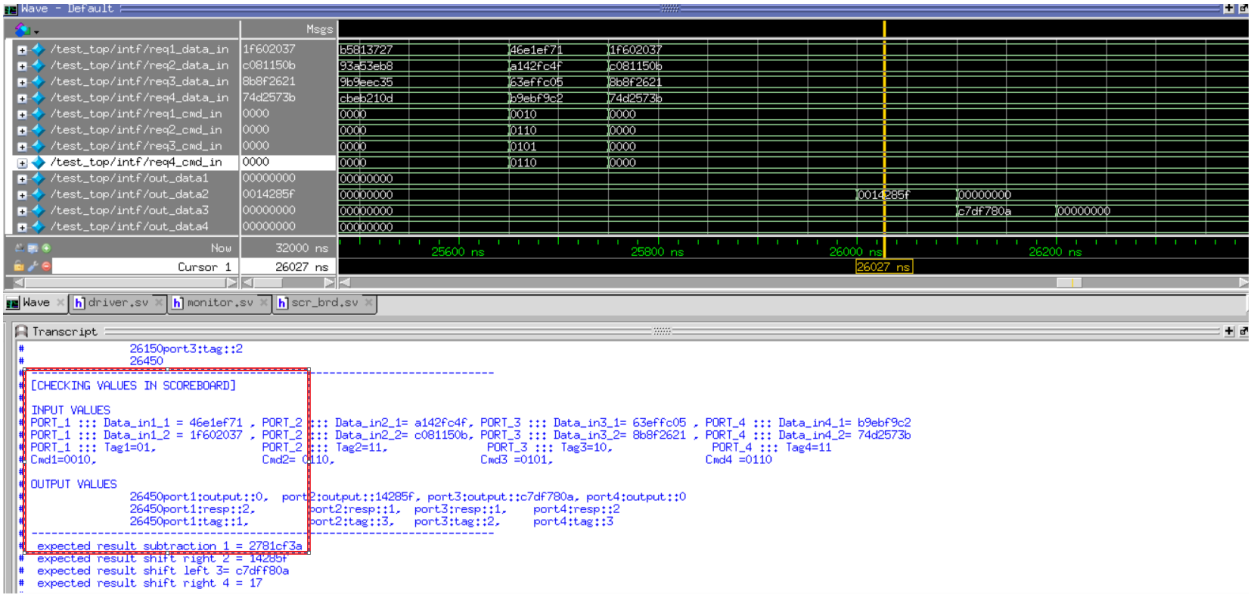

Figure 6: Subtraction without underflow

3.2.2 Subtraction with underflow

Figure 7 below shows simulation for subtraction of two operands with underflow.

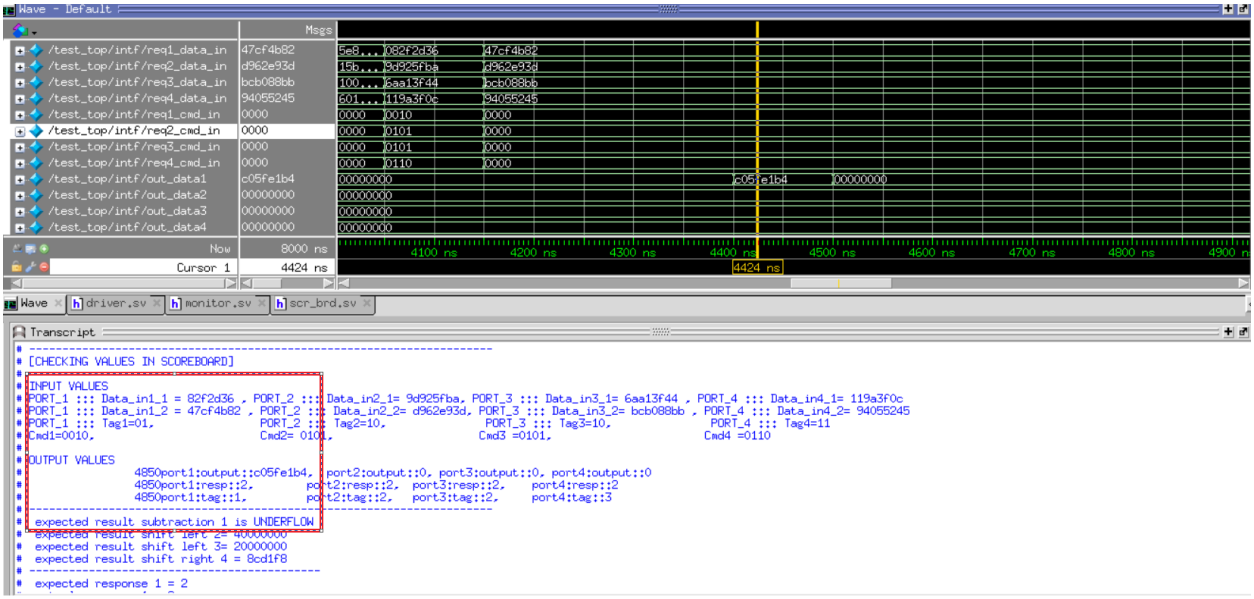

Figure 7: Subtraction with underflow

3.3 Shifting

3.3.1 Shift Left

Figure 8 and 9 below shows simulation for shifting left of port 1 inputs.

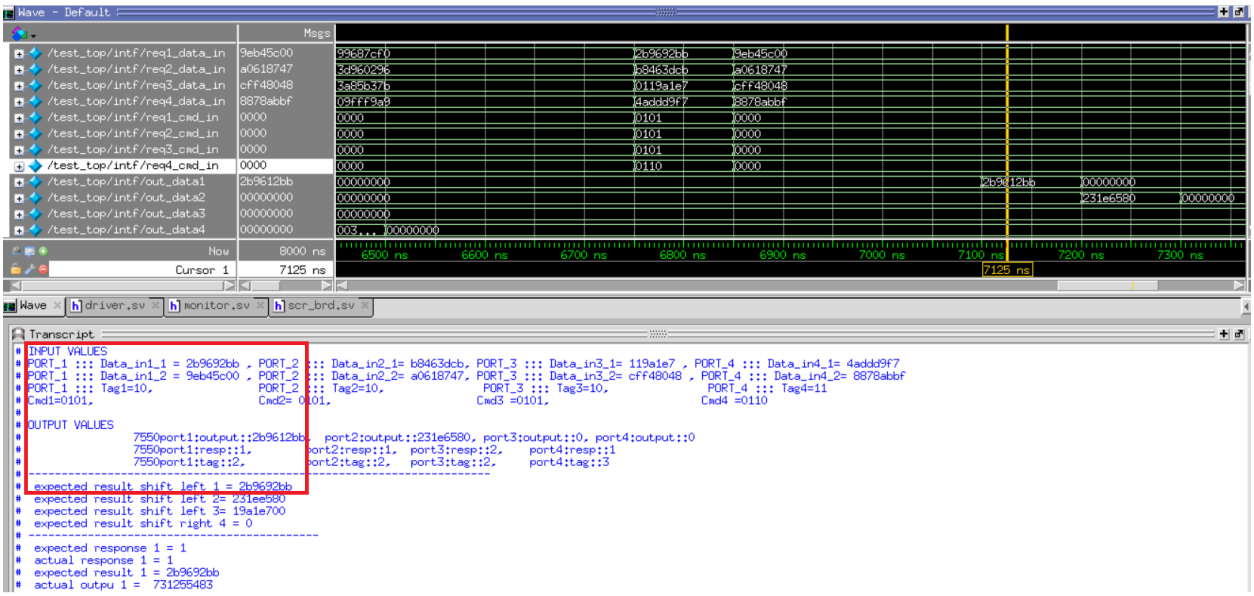

Figure 8: Shift left

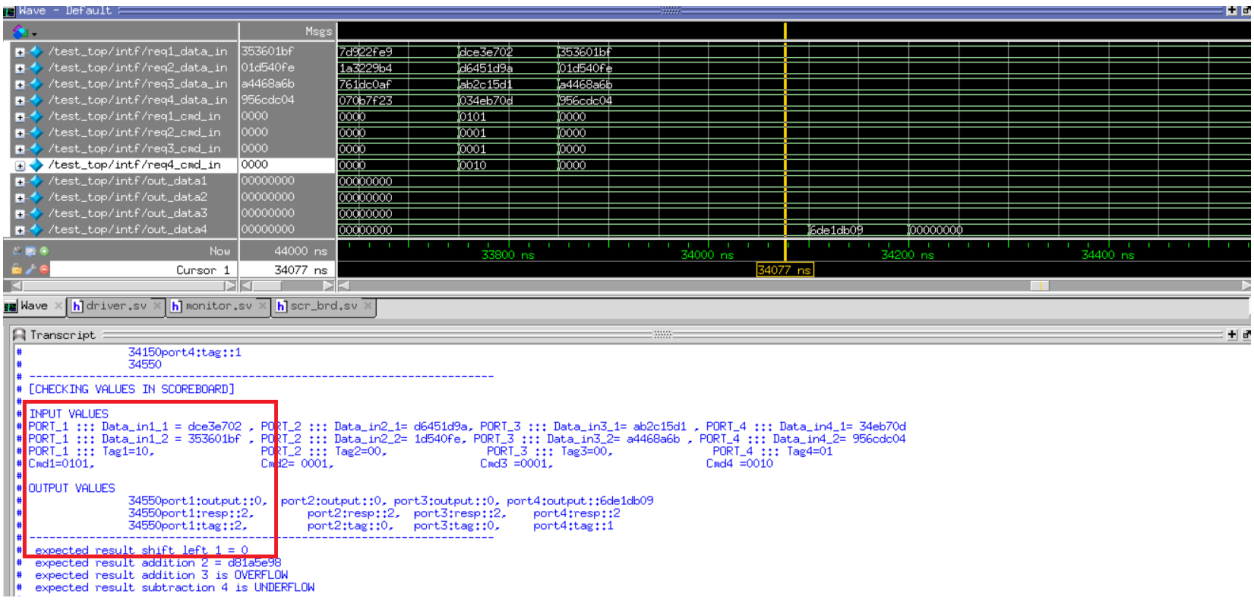

Figure 9: Shift left

3.3.2 Shift Right

Figure 10 and figure 11 below show simulation for shifting right of port 1 inputs.

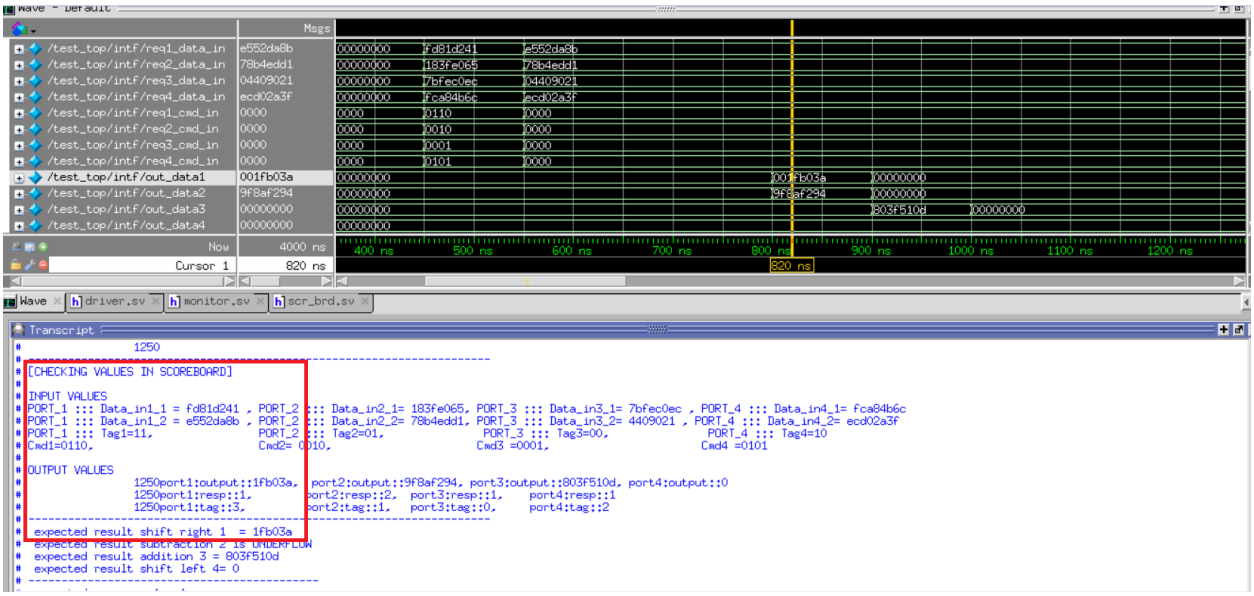

Figure 10: Shift right

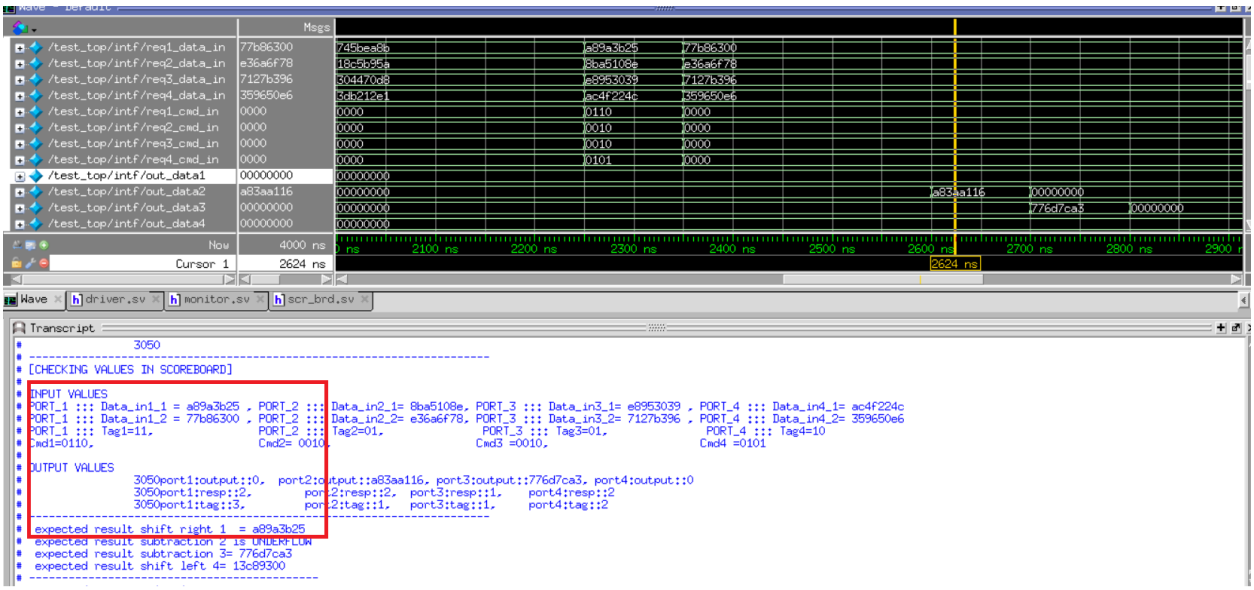

Figure 11: Shift right

4 Port 2 Test Results

The following sections have the results for all repeated cases in port2.

4.0 Invalid command

Figure 12 below shows simulation for random invalid command.

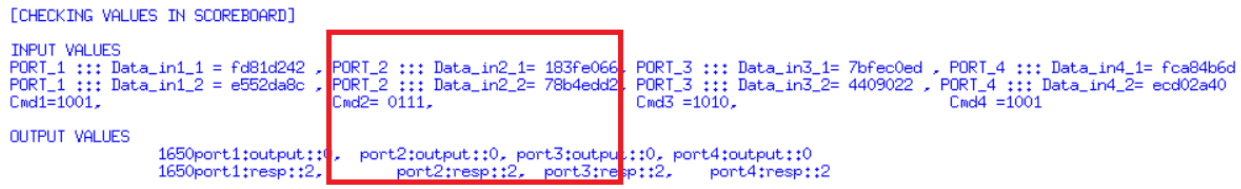

Figure 12: Invalid command

4.1 Addition

4.1.1 Addition without overflow

Figure 13 and 14 below shows simulation for addition of two operands without having overflow.

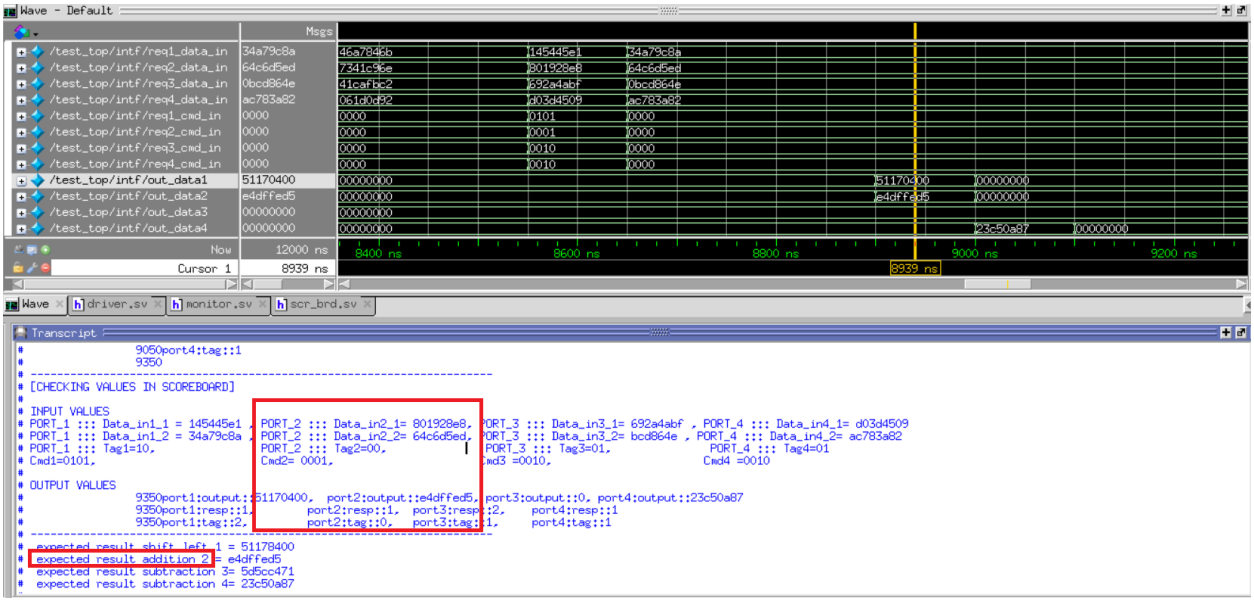

Figure 13: Addition without overflow

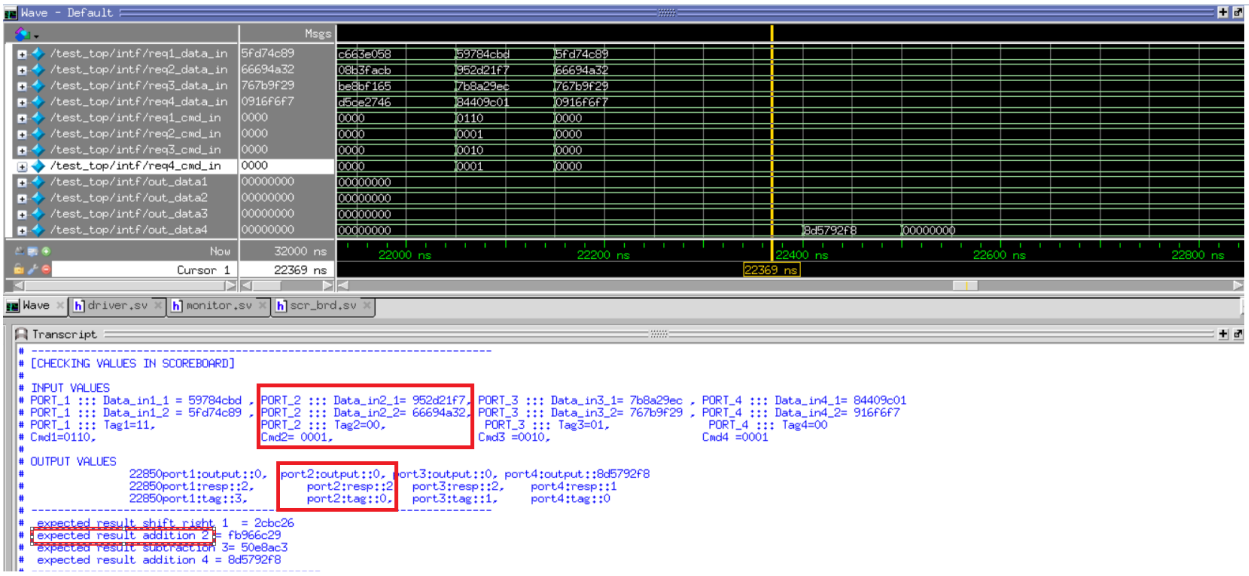

Figure 14: Addition without overflow

4.1.2 Addition with overflow

Figure 15 below shows simulation for addition of two operands with overflow.

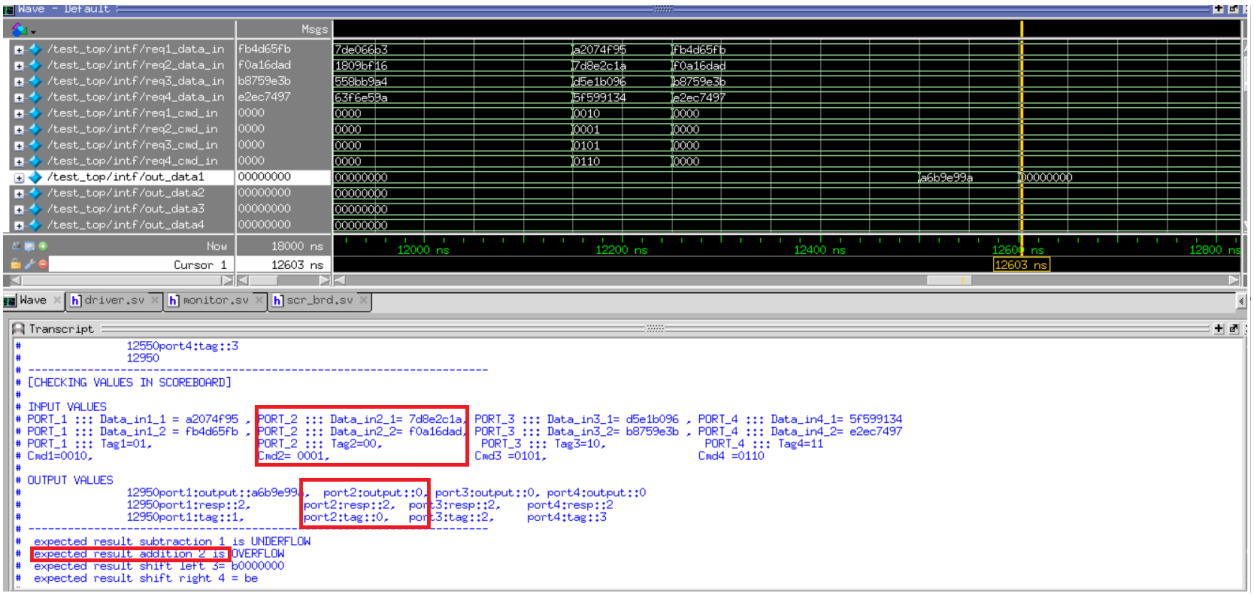

Figure 15: Addition with overflow

4.2 Subtraction

4.2.1 Subtraction without underflow

Figure 16 and 17 below shows simulation for subtraction of two operands without underflow.

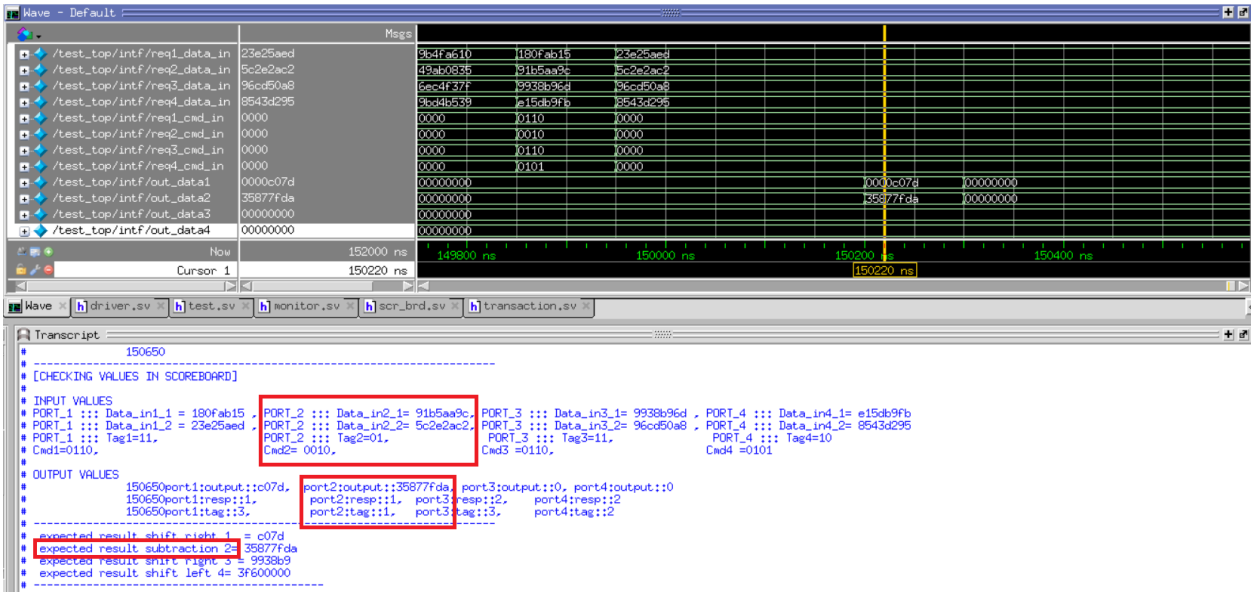

Figure 16: Subtraction without underflow

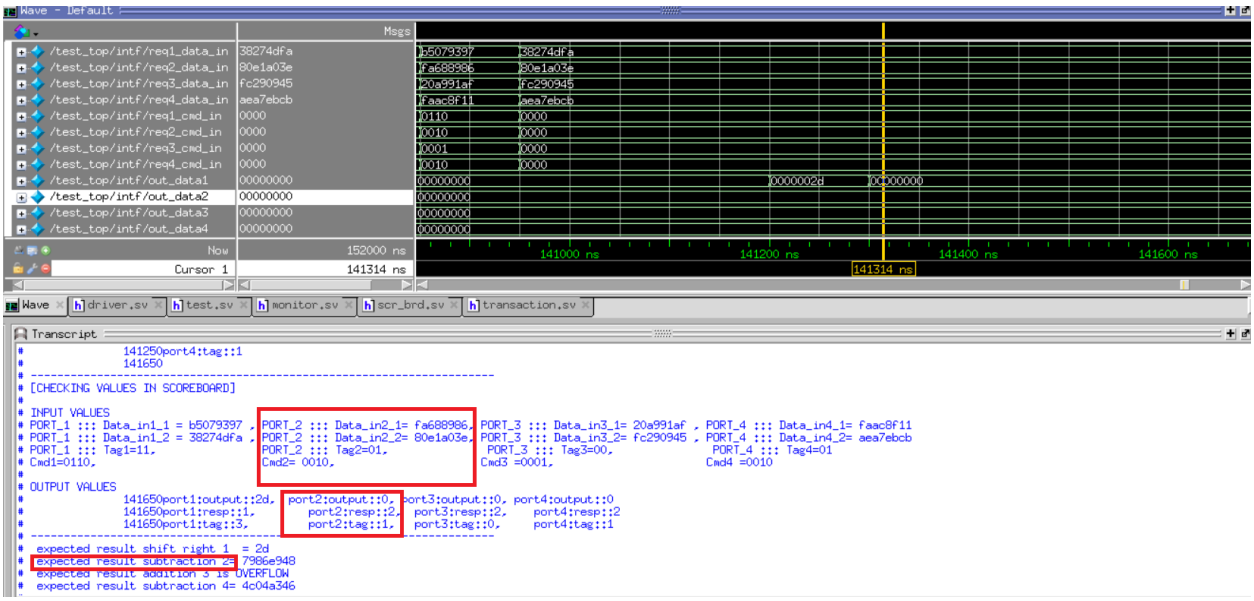

Figure 17: Subtraction without underflow

4.2.2 Subtraction with underflow

Figure 18 below shows simulation for subtraction of two operands with underflow.

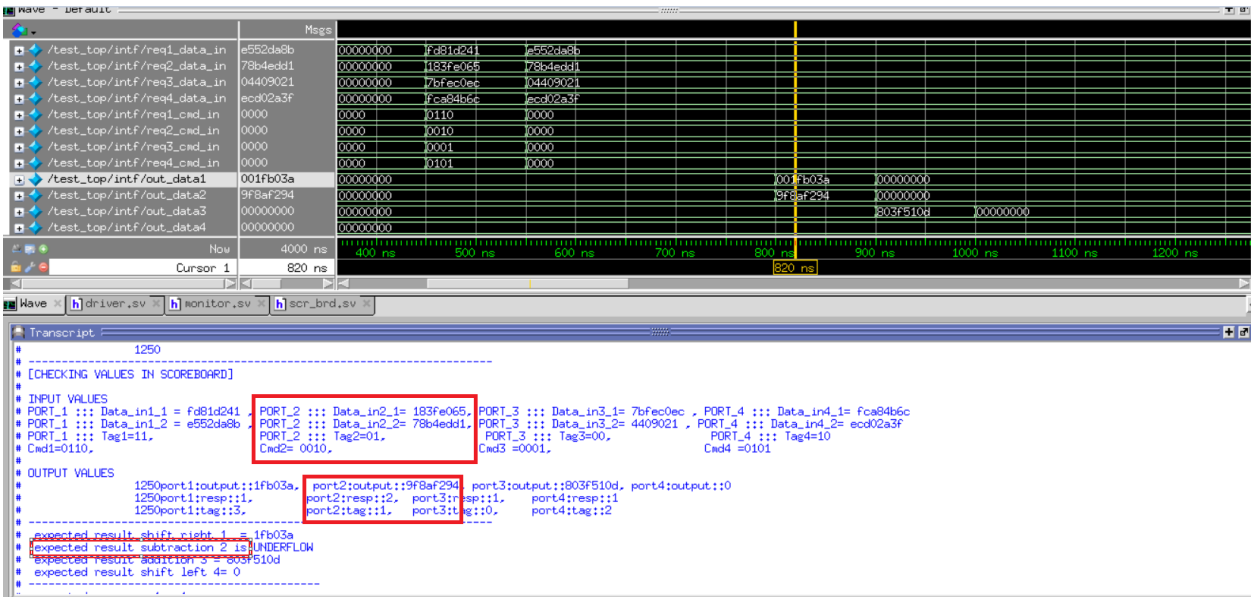

Figure 18:Subtraction with underflow

4.3 Shifting

4.3.1 Shifting Left

Figure 19 and 20 below shows simulation for shifting left of port 2 inputs.

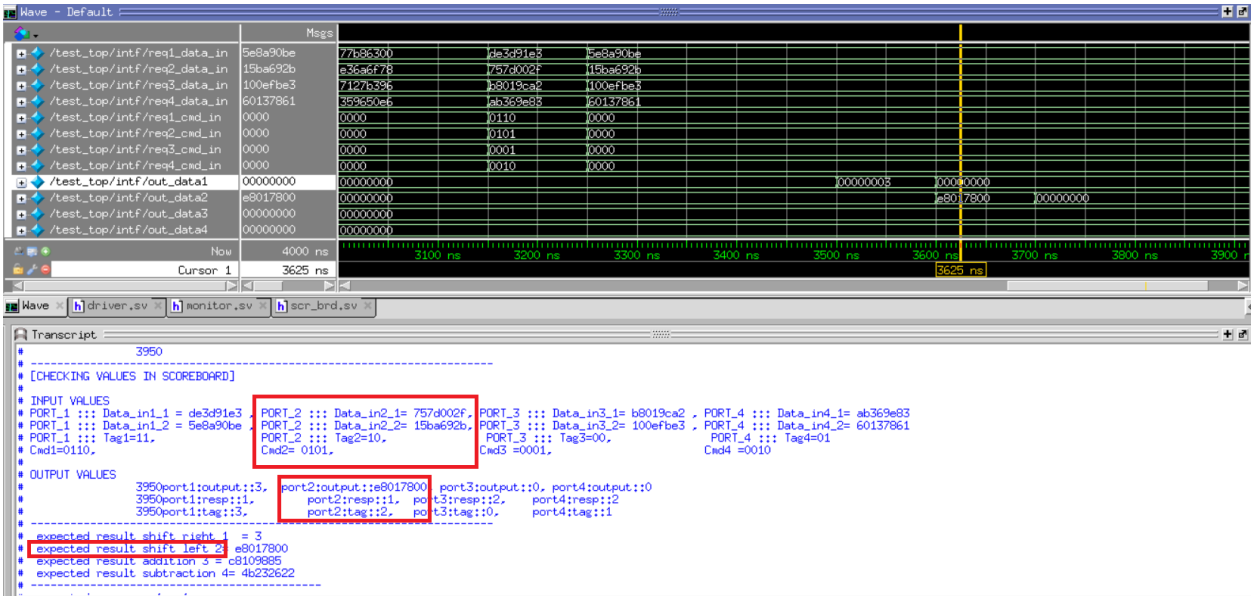

Figure 19: Shift left

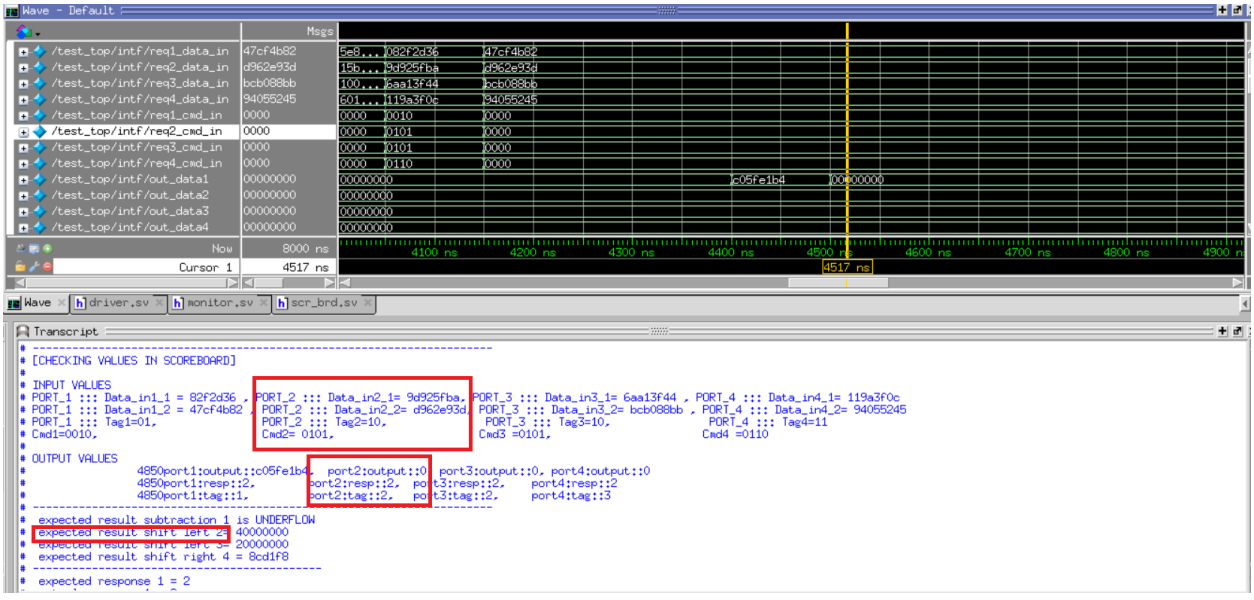

Figure 20: Shift left

4.3.2 Shifting Right

Figure 21 and figure 22 below show simulation for shifting right of port 2 inputs.

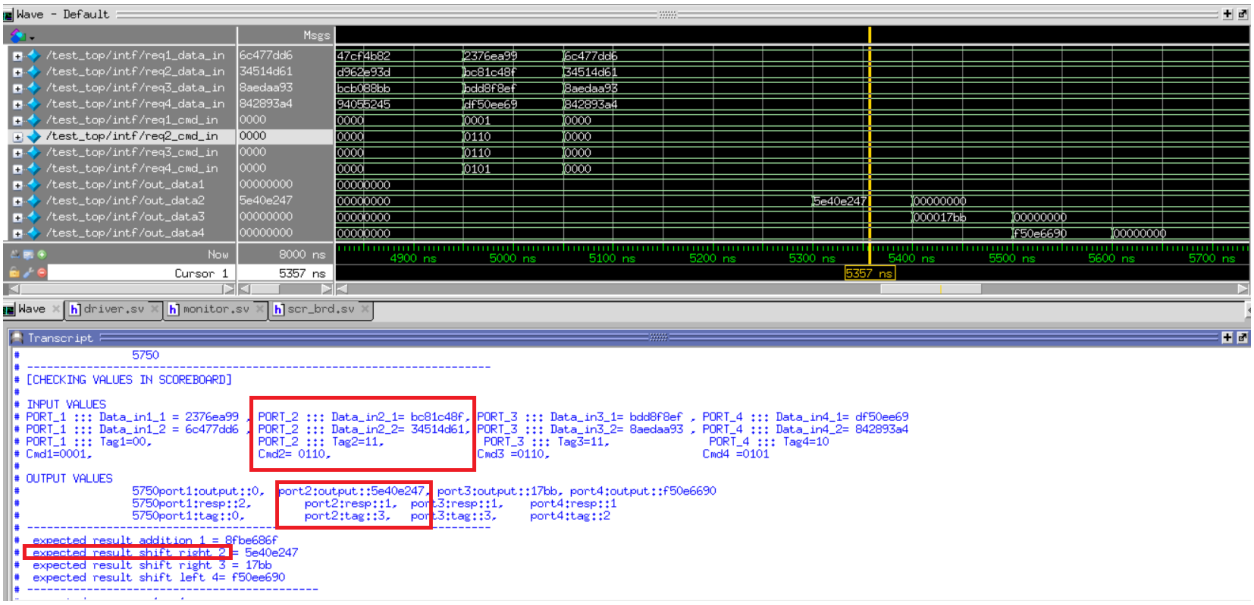

Figure 21: Shift right

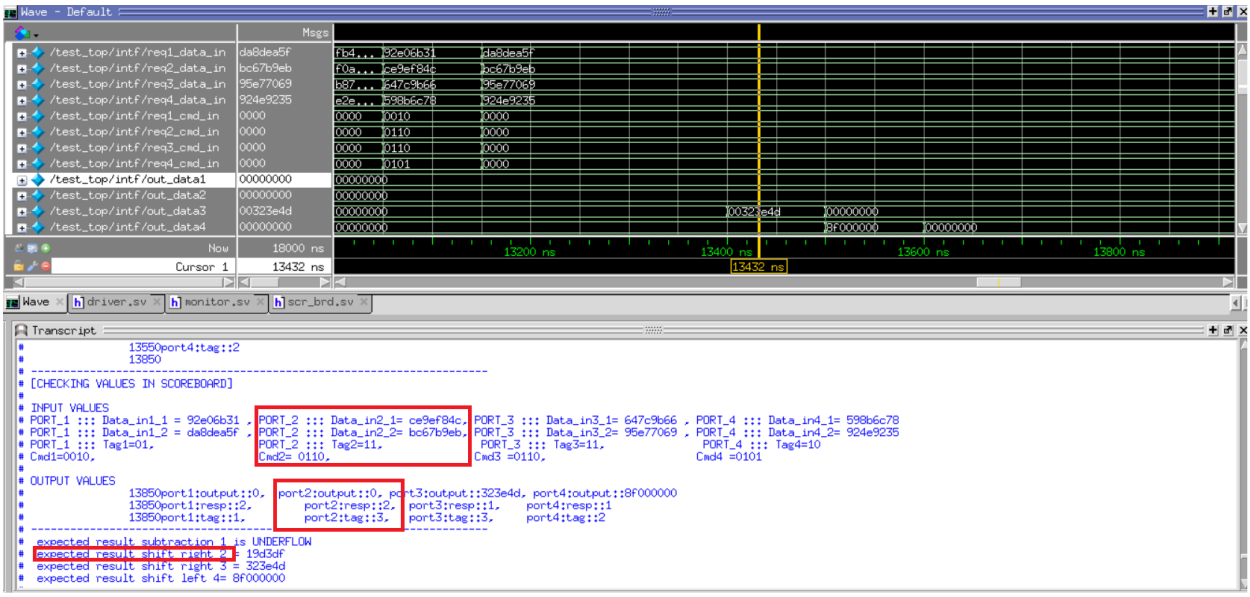

Figure 22:Shift right

5 Port 3 Test Results

The following sections have the results for all repeated cases in port3.

5.0 Invalid command

Figure 23 below shows simulation for random invalid command.

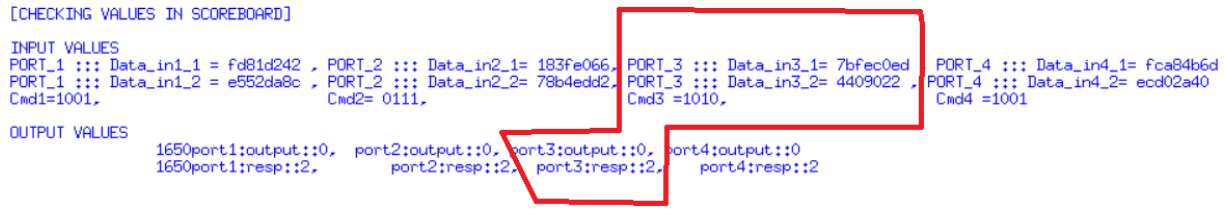

Figure 23: Invalid command

5.1 Addition

5.1.1 Addition without overflow

Figures 24 and 25 below show simulation for addition of two operands without having overflow.

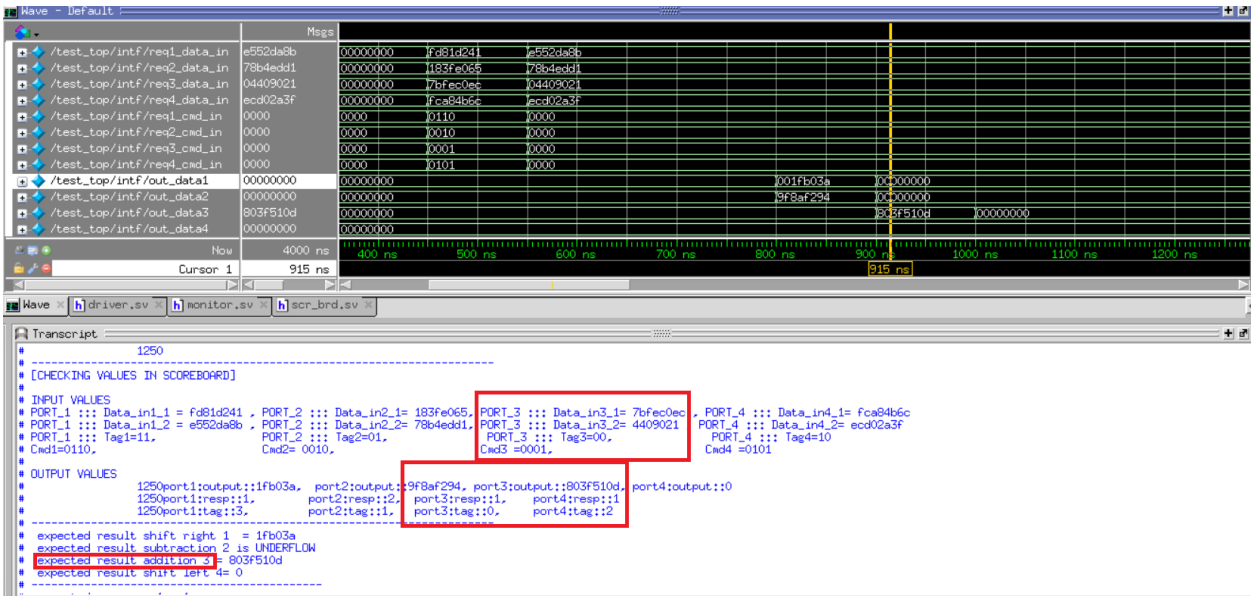

Figure 24: Addition without overflow

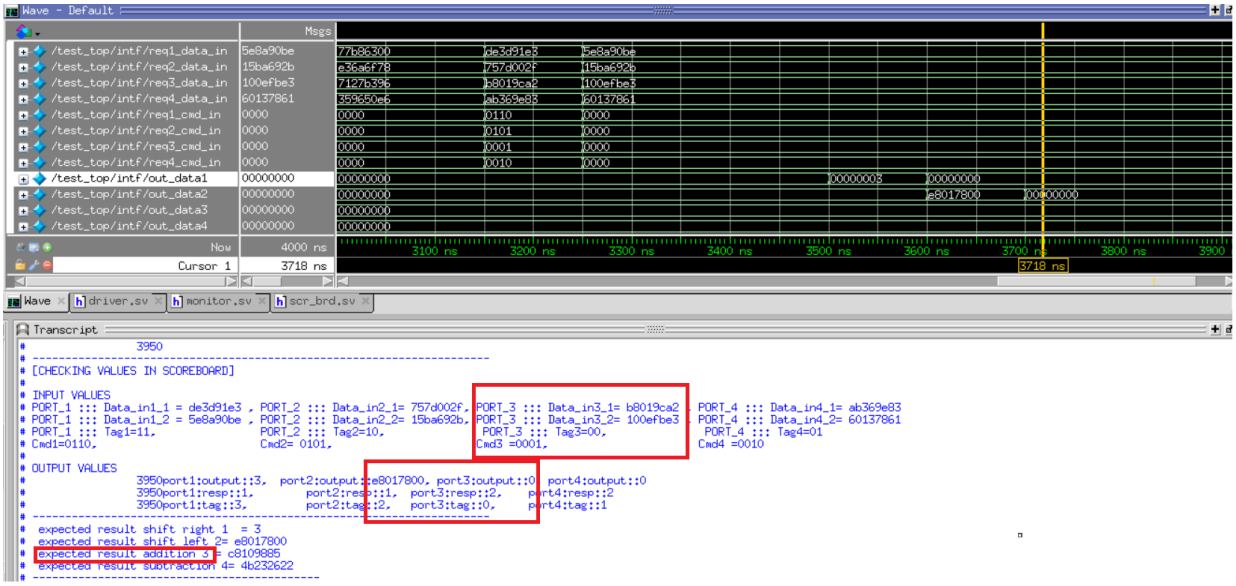

Figure 25: Addition without overflow

5.1.2 Addition with overflow

Figure 26 below shows simulation for addition of two operands with overflow.

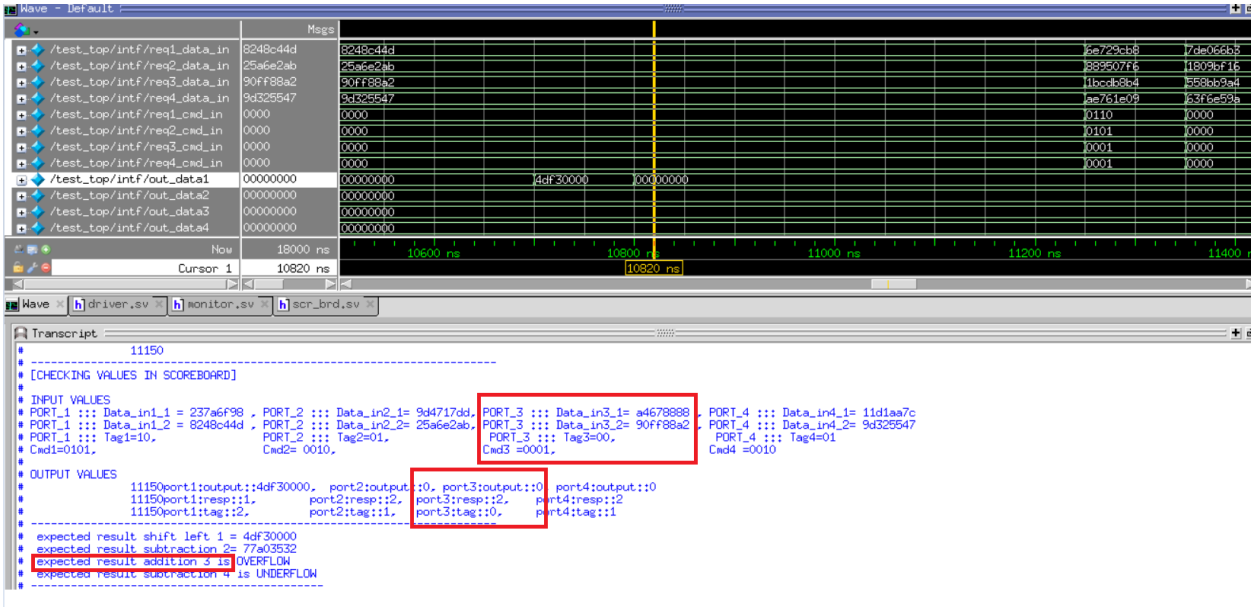

Figure 26: Addition with overflow

5.2 Subtraction

5.2.1 Subtraction without underflow

Figures 27 and 28 below show simulation for subtraction of two operands without underflow.

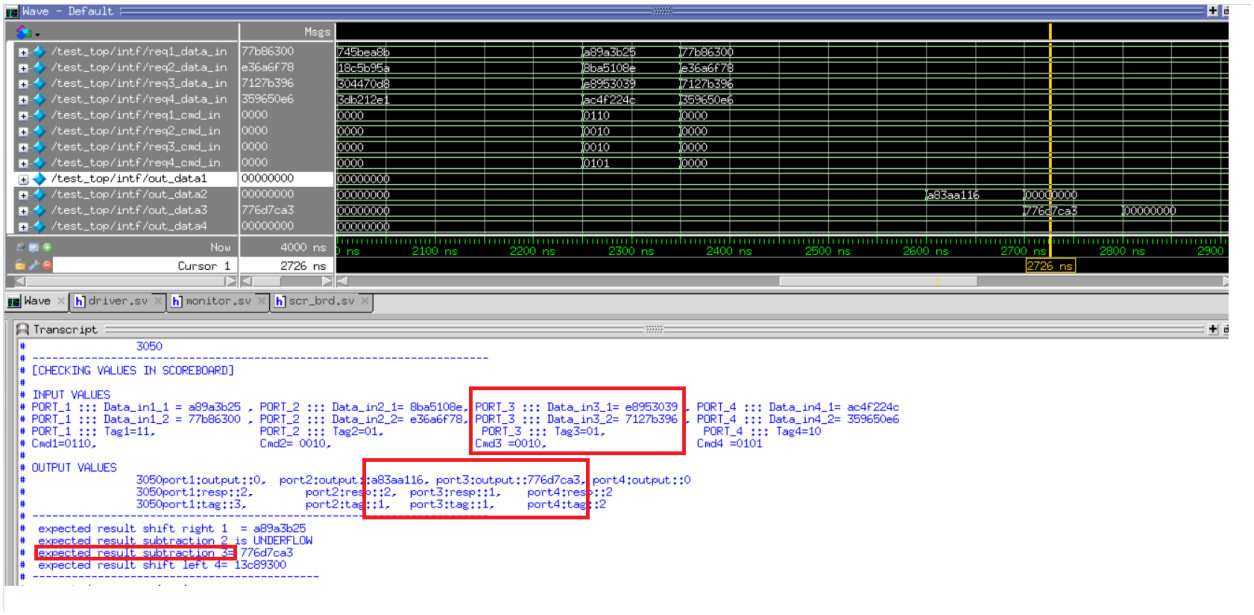

Figure 27: Subtraction without underflow

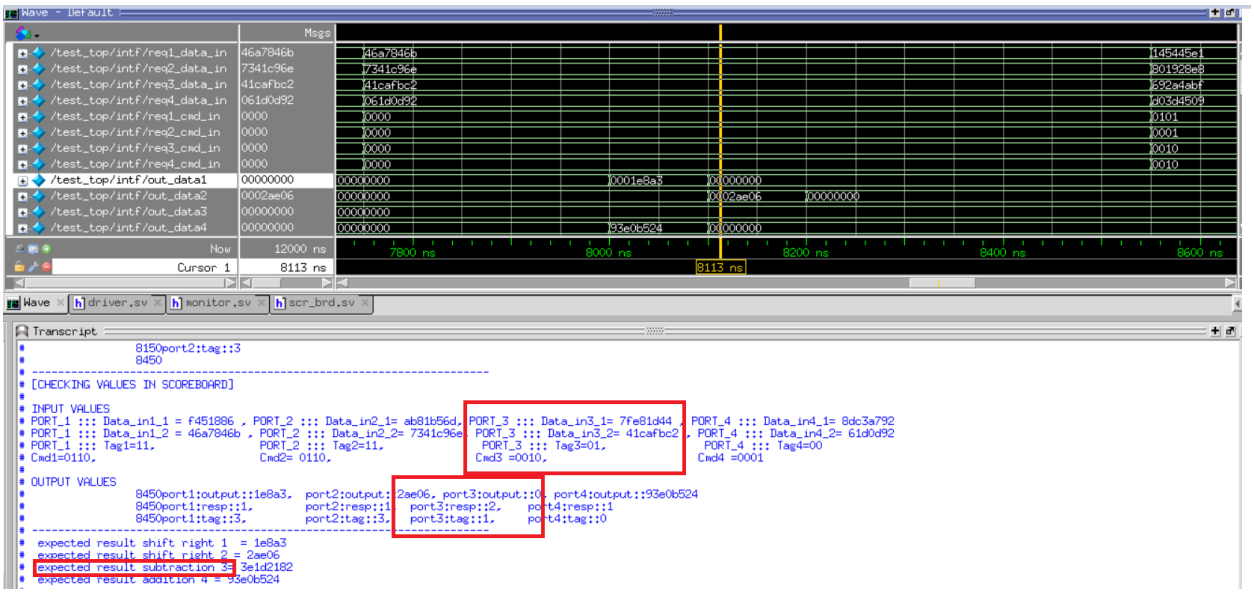

Figure 28: Subtraction without underflow

5.2.2 Subtraction with underflow

Figure 29 below shows simulation for subtraction of two operands with underflow.

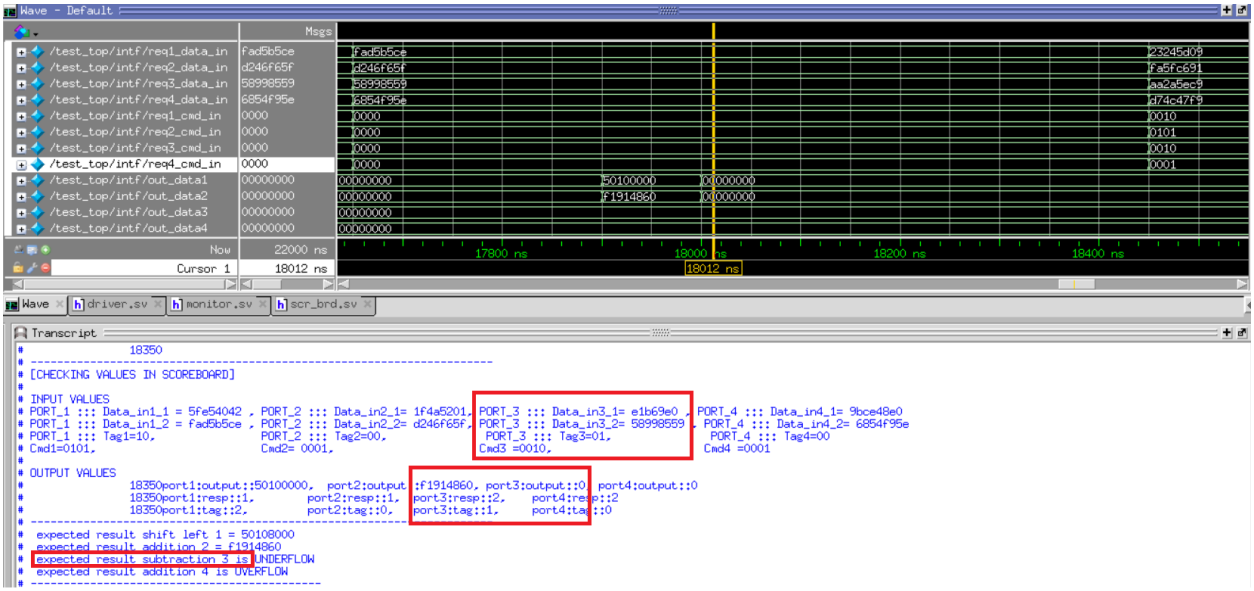

Figure 29: Subtraction with underflow

5.3 Shifting

5.3.1 Shifting Left

Figures 30 and 31 below show simulation for shifting left of port 3 inputs.

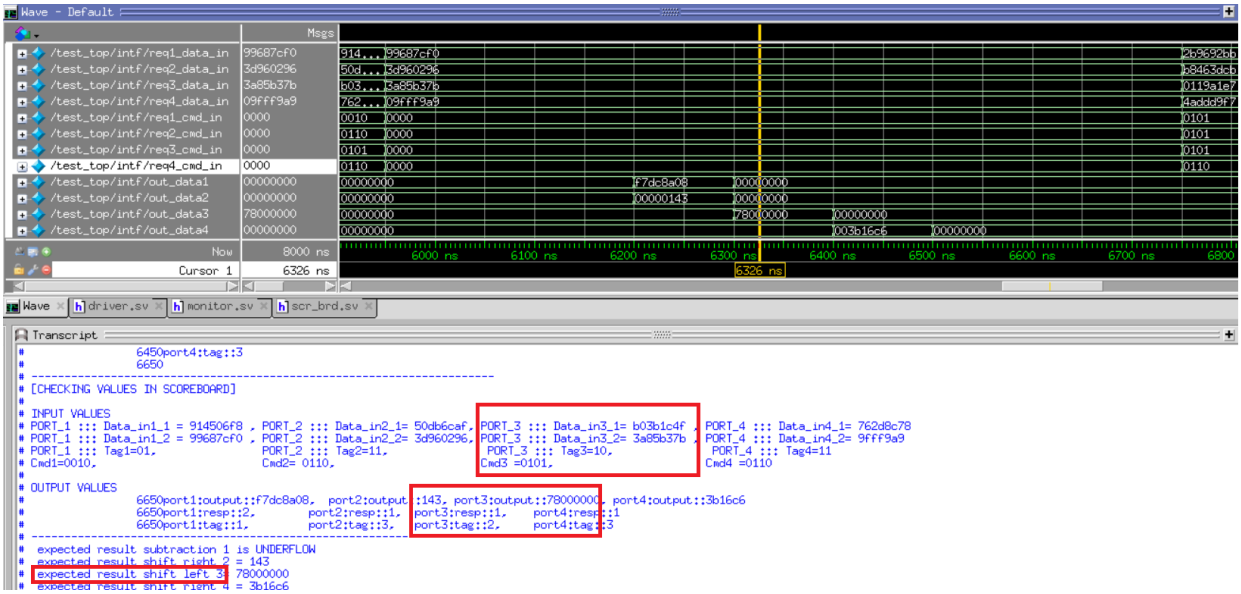

Figure 30:Shift left

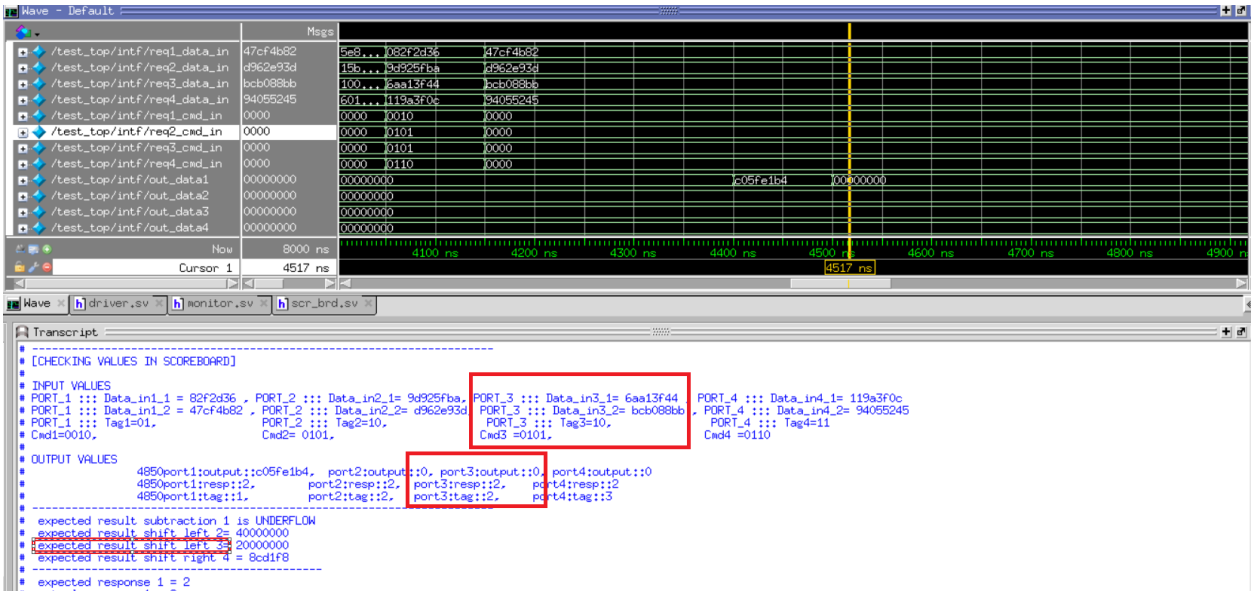

Figure 31:Shift left

5.3.2 Shifting Right

Figure 32 and figure 33 below show simulation for shifting right of port 3 inputs.

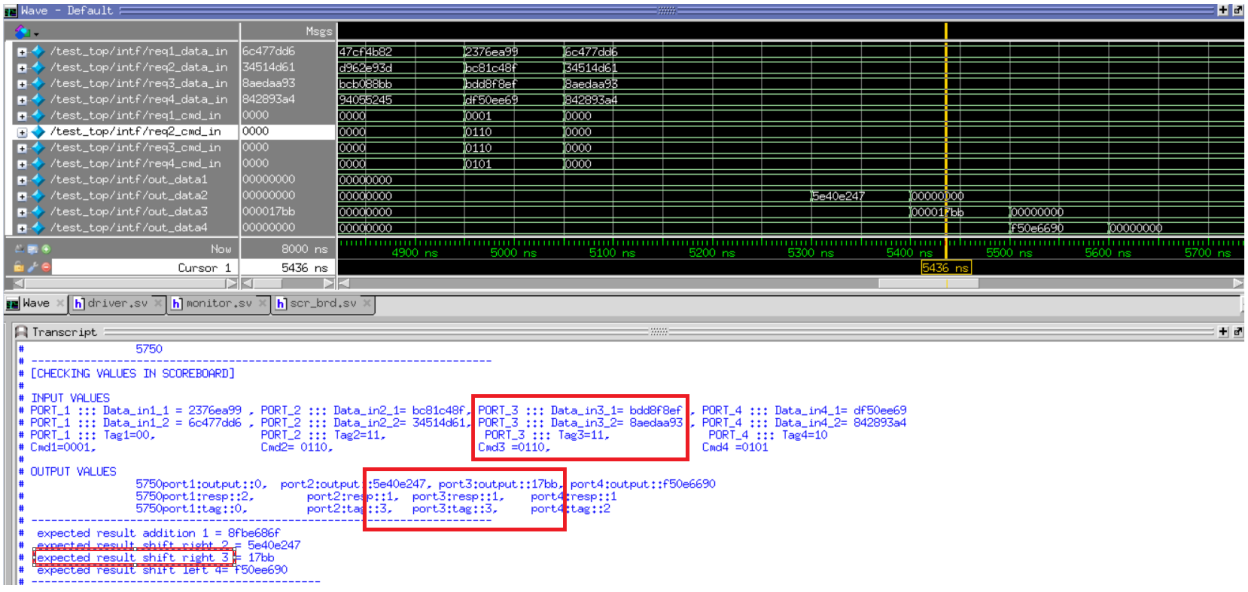

Figure 32:Shift right

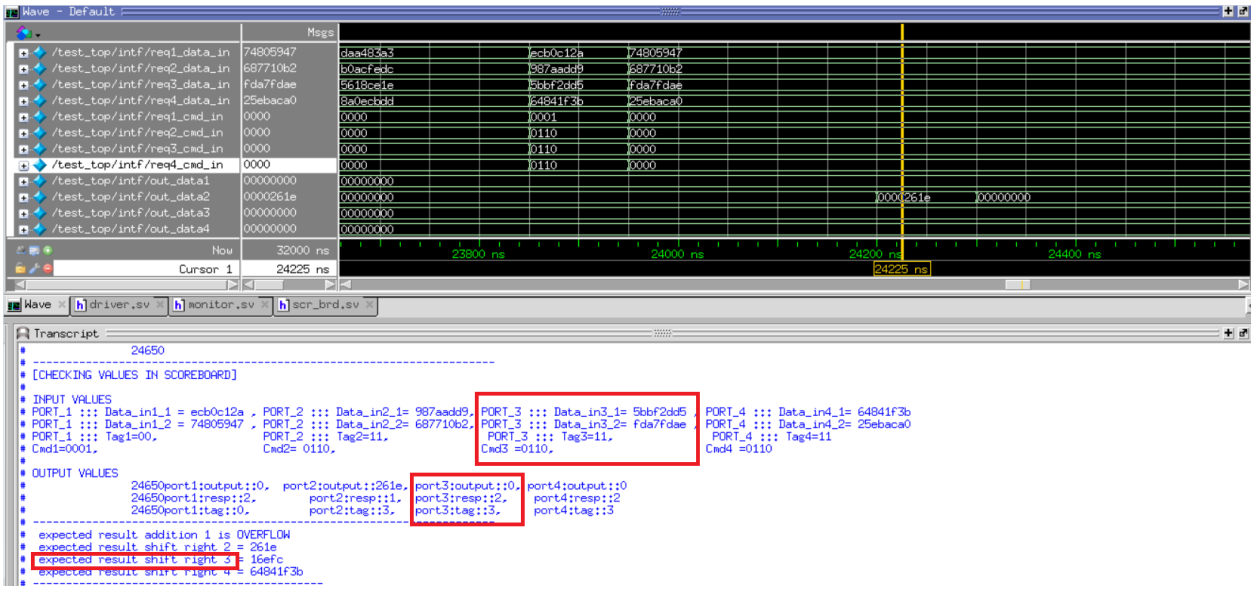

Figure 33:Shift right

6 Port 4 Test Results

The following sections show the results for all cases in Calc2 port4.

6.0 Invalid command

Figure 34 below shows the results with invalid command.

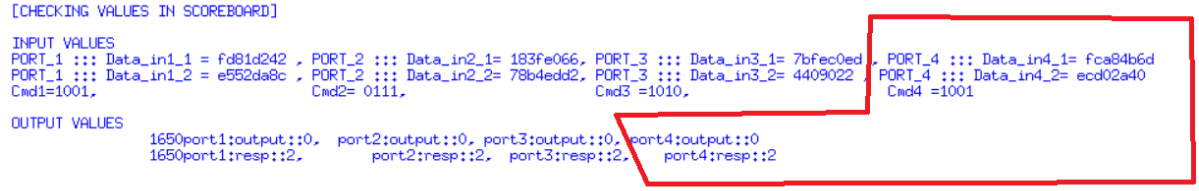

Figure 34: Invalid command

6.1 Addition

6.1.1 Addition without overflow

Figures 35 and 36 below show simulation for addition of two operands without having overflow.

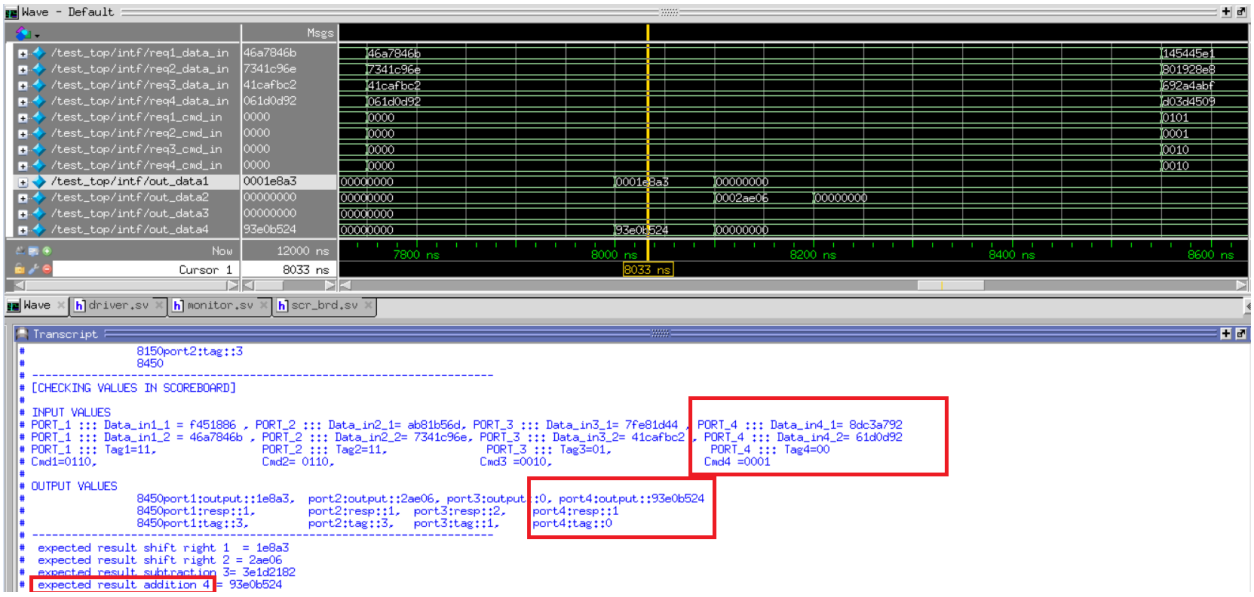

Figure 35: Addition without overflow

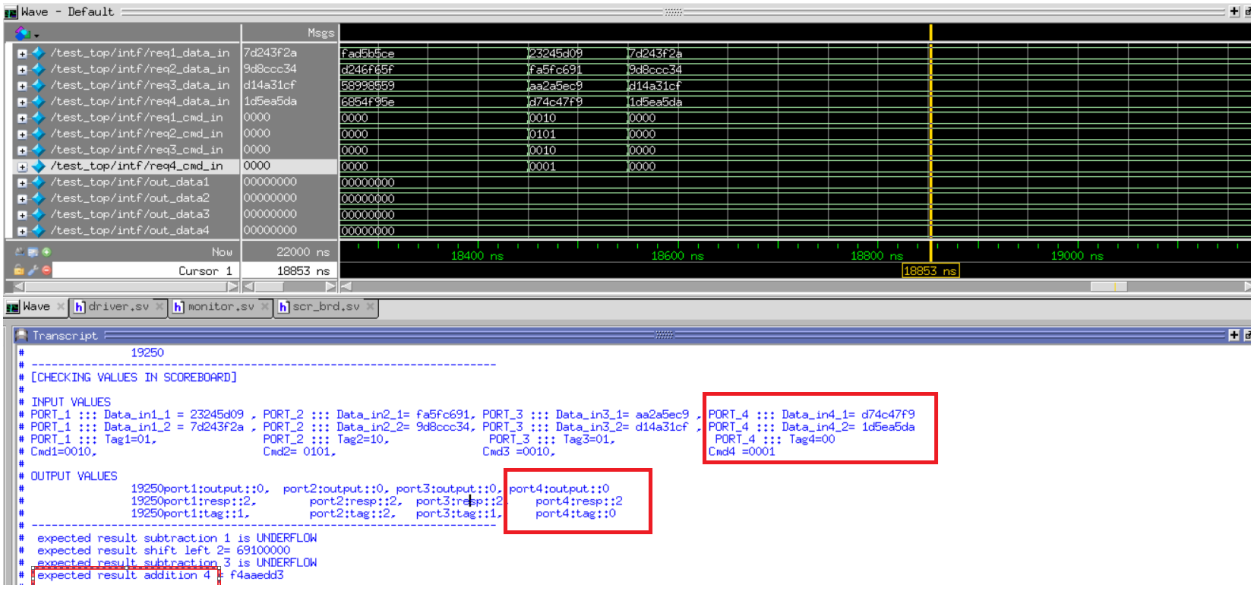

Figure 36:Addition without overflow

6.1.2 Addition with overflow

Figure 37 below shows simulation for addition of two operands with overflow.

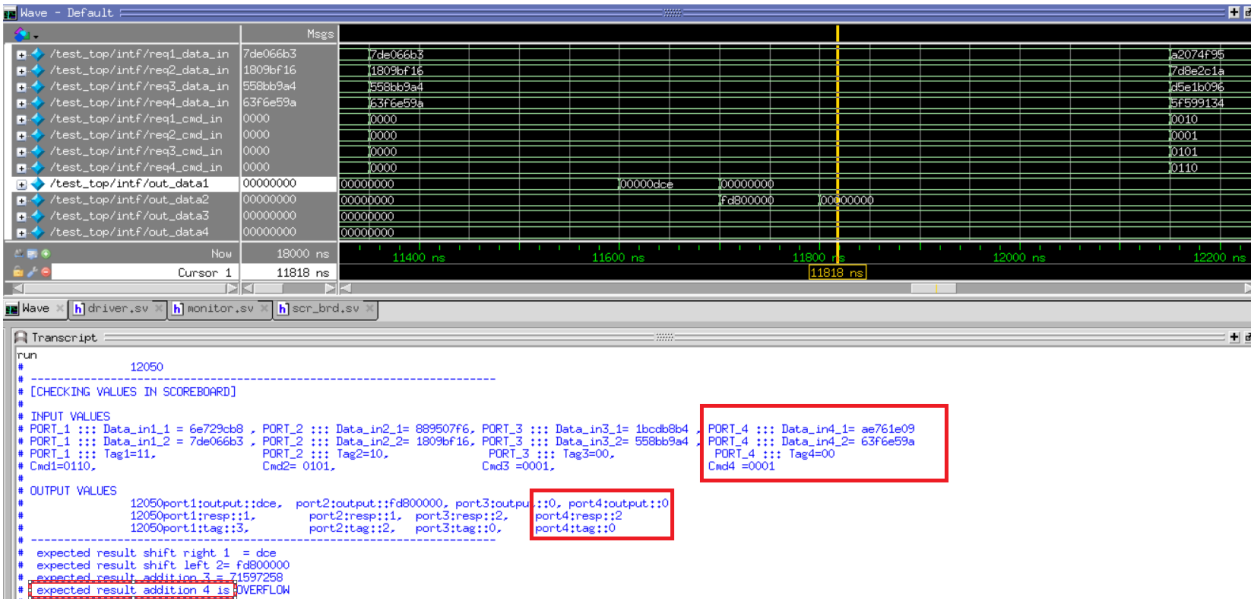

Figure 37:Addition with overflow

6.2 Subtraction

6.2.1 Subtraction without underflow

Figures 38 and 39 below show simulation for subtraction of two operands without underflow.

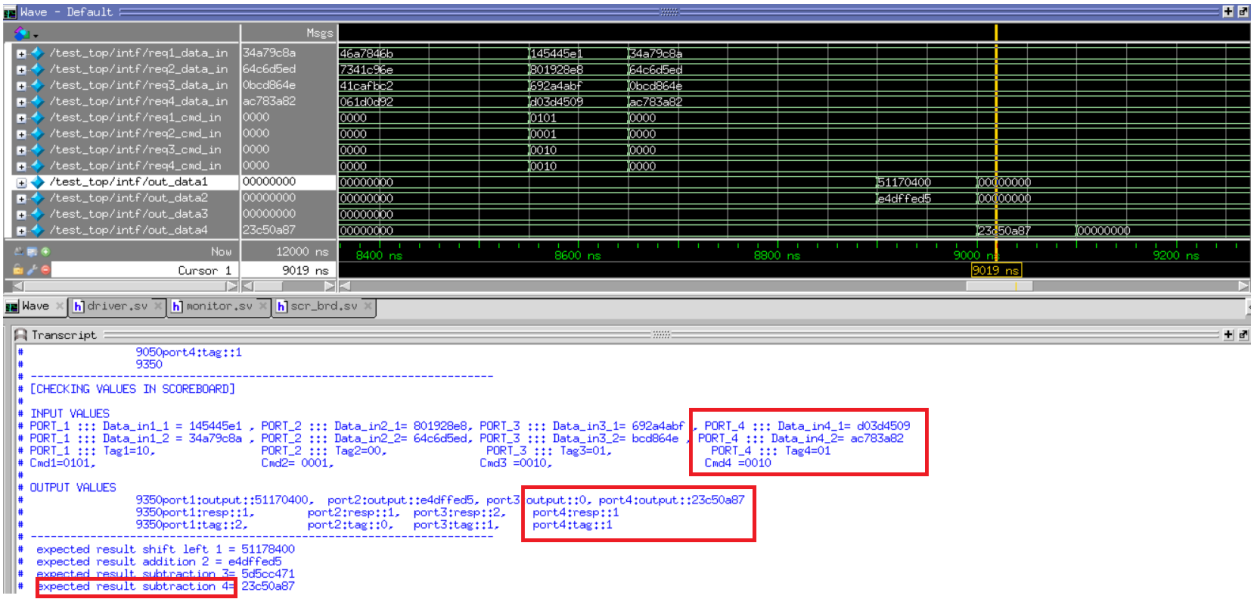

Figure 38:Subtraction without underflow

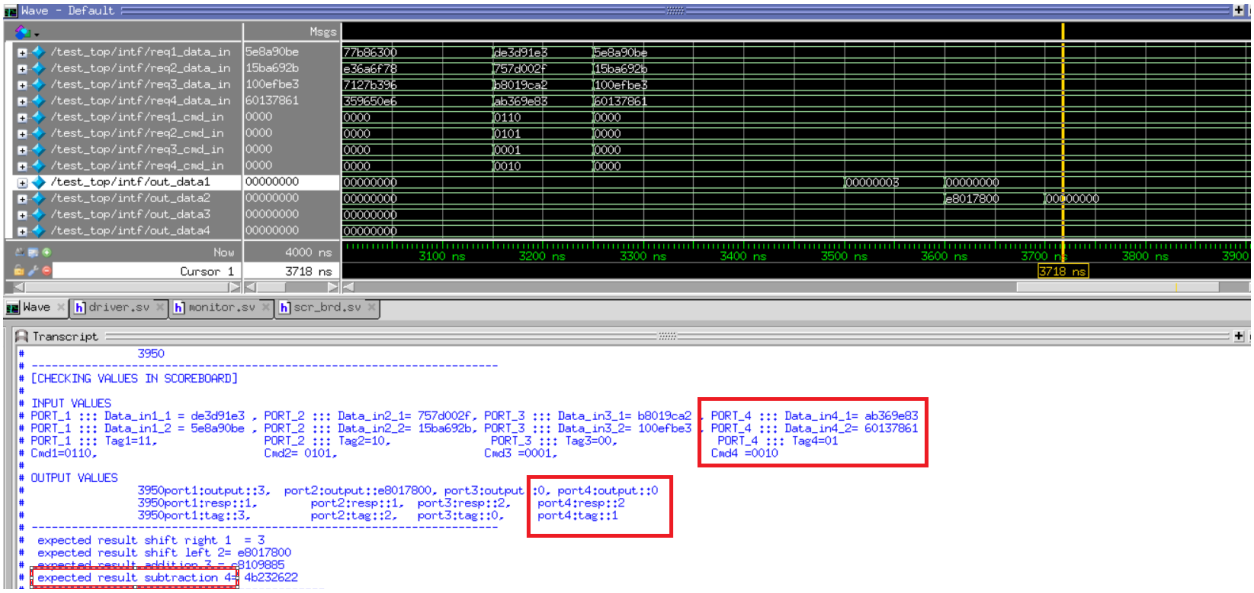

Figure 39:Subtraction without underflow

6.2.2 Subtraction with underflow

Figure 40 below shows simulation for subtraction of two operands with underflow.

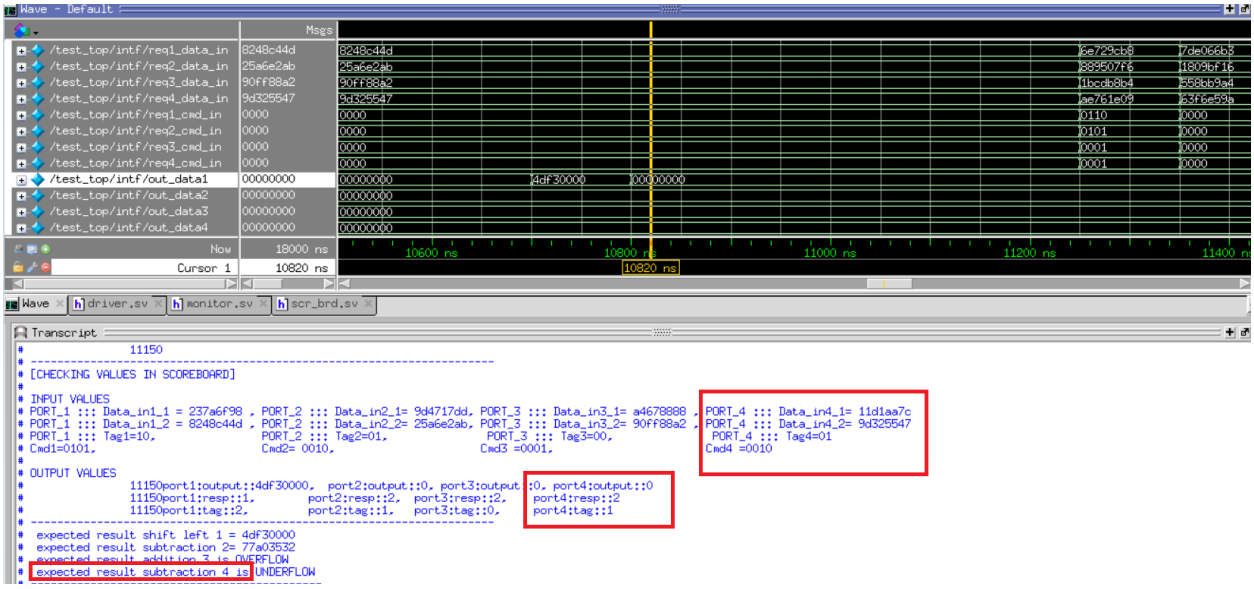

Figure 40:Subtraction with underflow

6.3 Shifting

6.3.1 Shifting Left

Figures 41 and 42 below show simulation for shifting left of port 4 inputs.

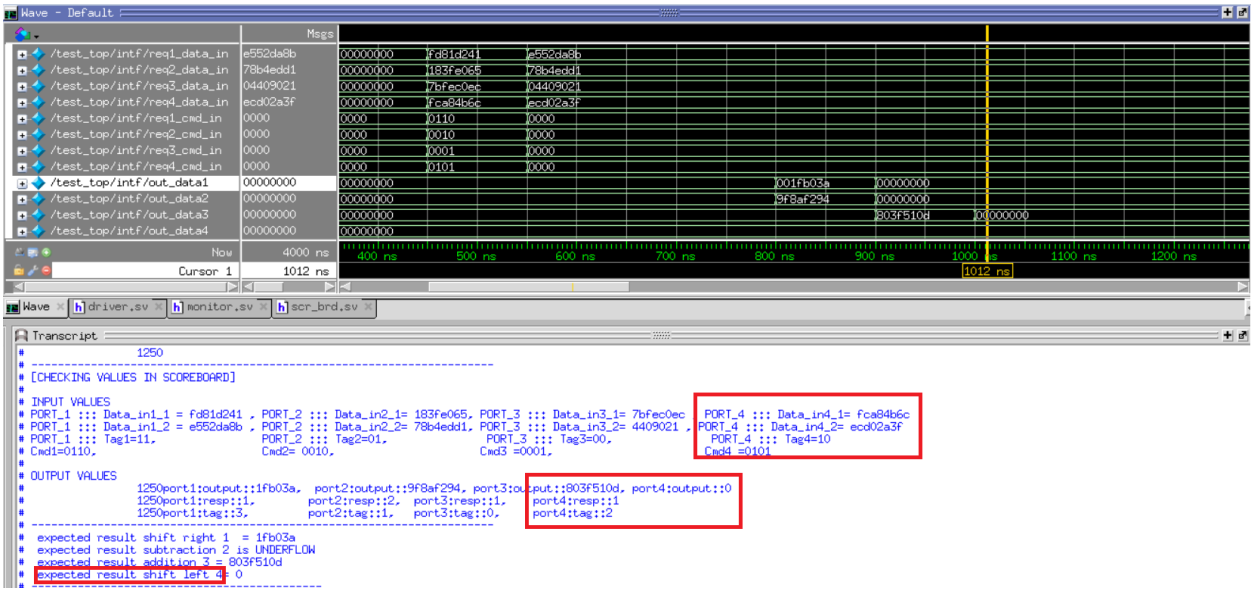

Figure 41:Shift left

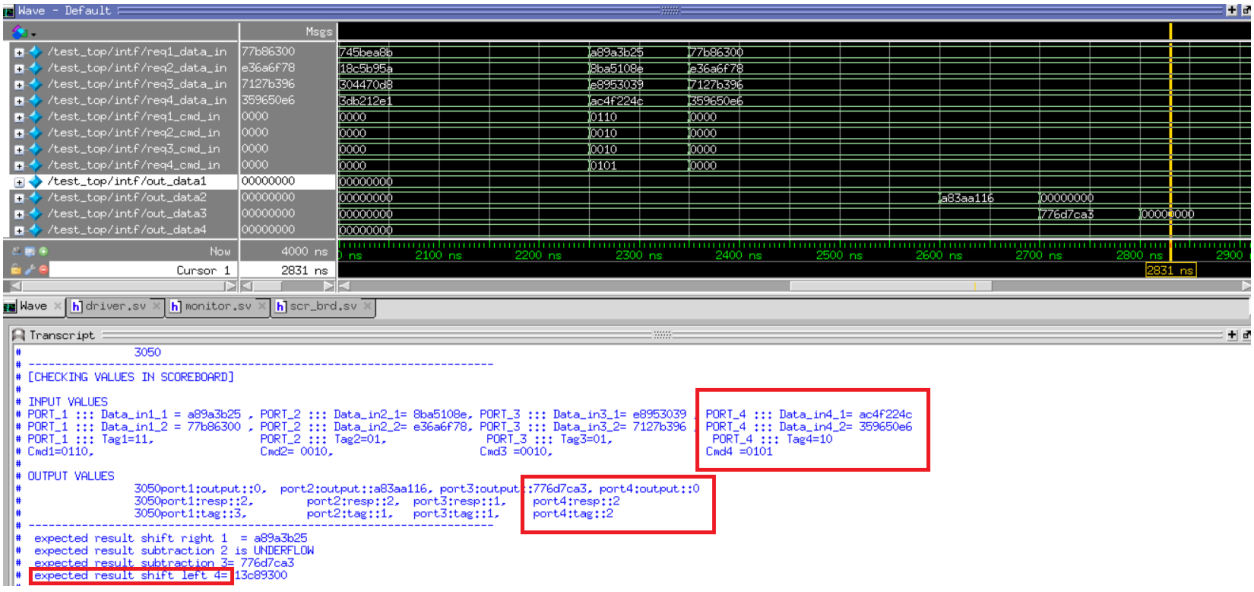

Figure 42:Shift left

6.3.2 Shifting Right

Figures 43 and figure 44 below show simulation for shifting right of port 4 inputs.

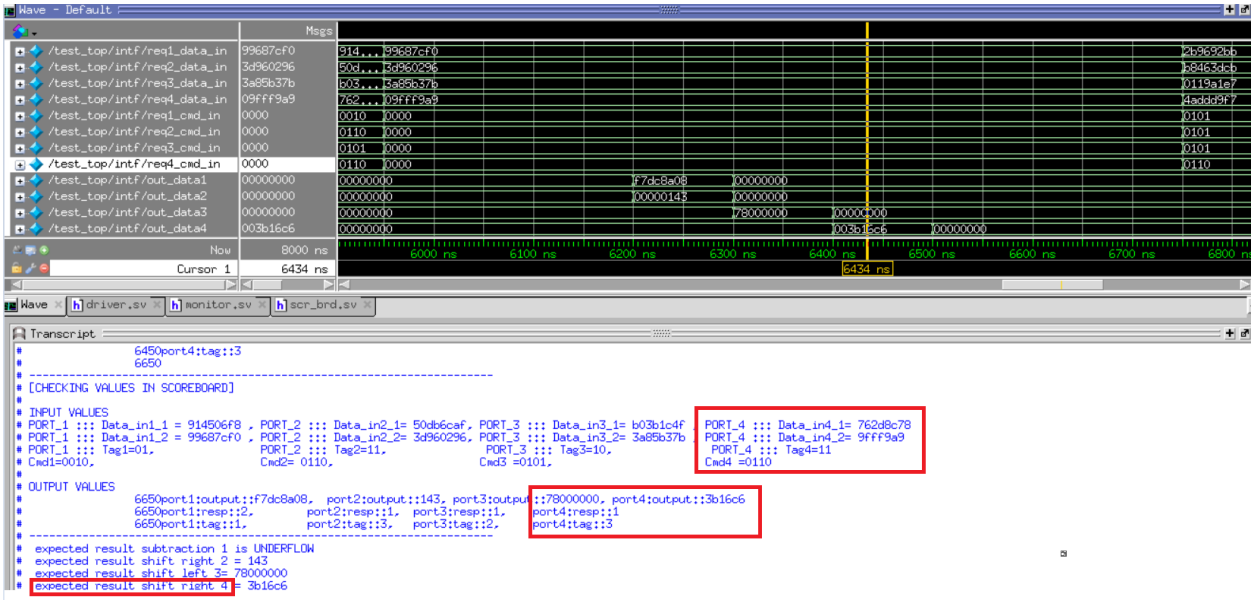

Figure 43:Shift right

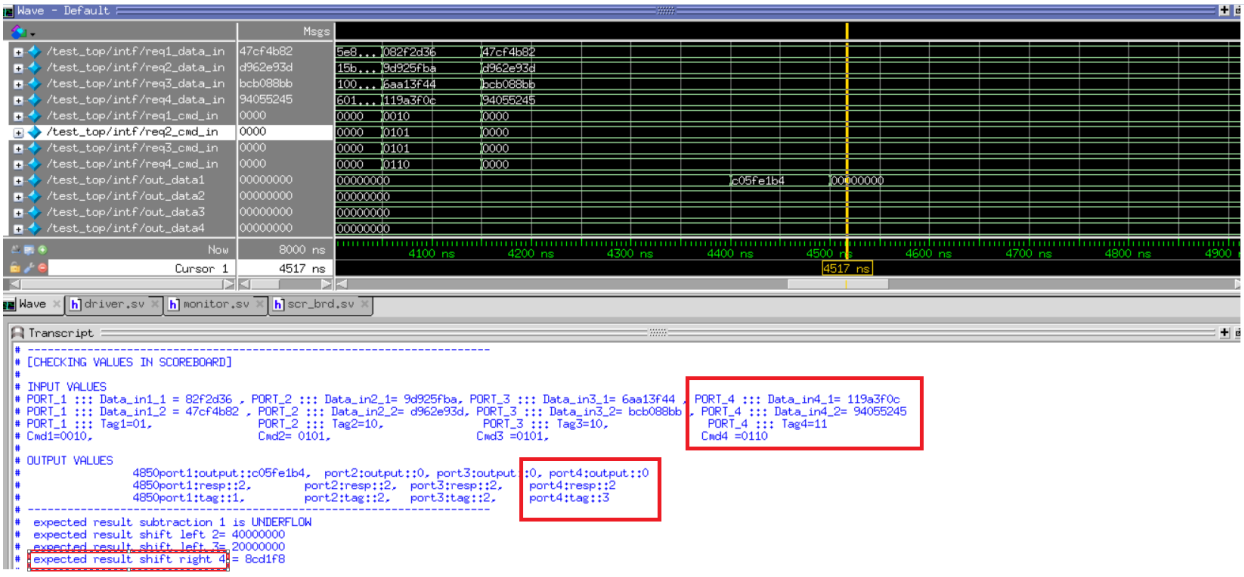

Figure 44:Shift right

7 Testing Ports with Single Command

In this test scenario, we have simulated the results when sending single command type to all ports to check the behavior of the DUT with not randomizing the command. Figures 45 , 46, 47, and 48 below shows the simulation results.

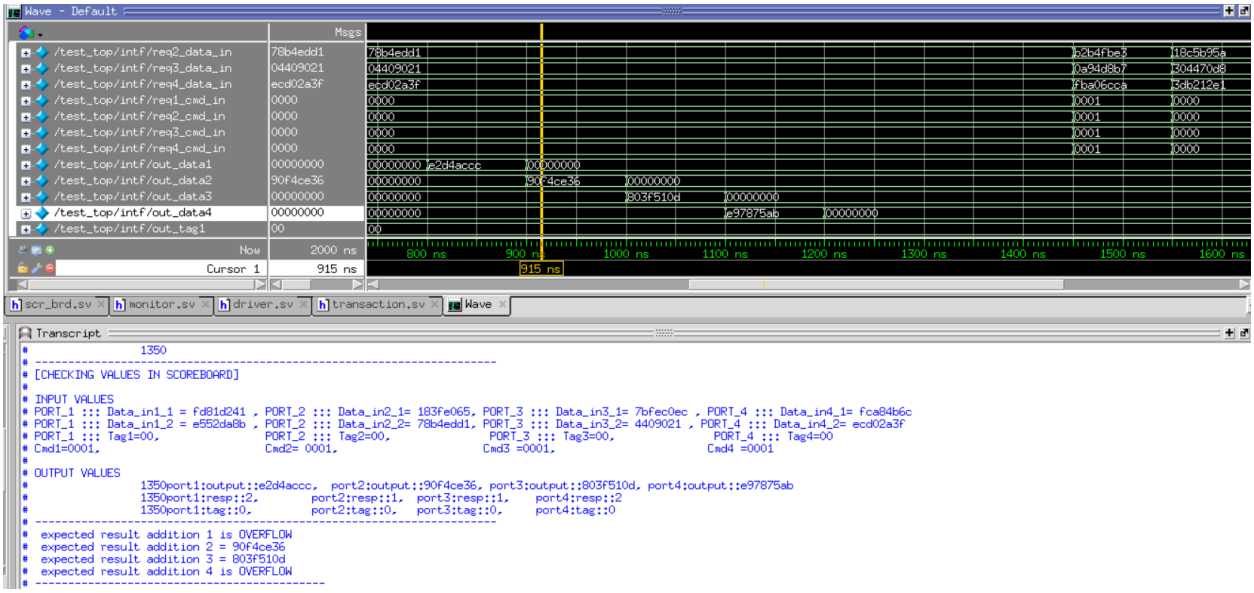

Figure 45: Addition command for all ports

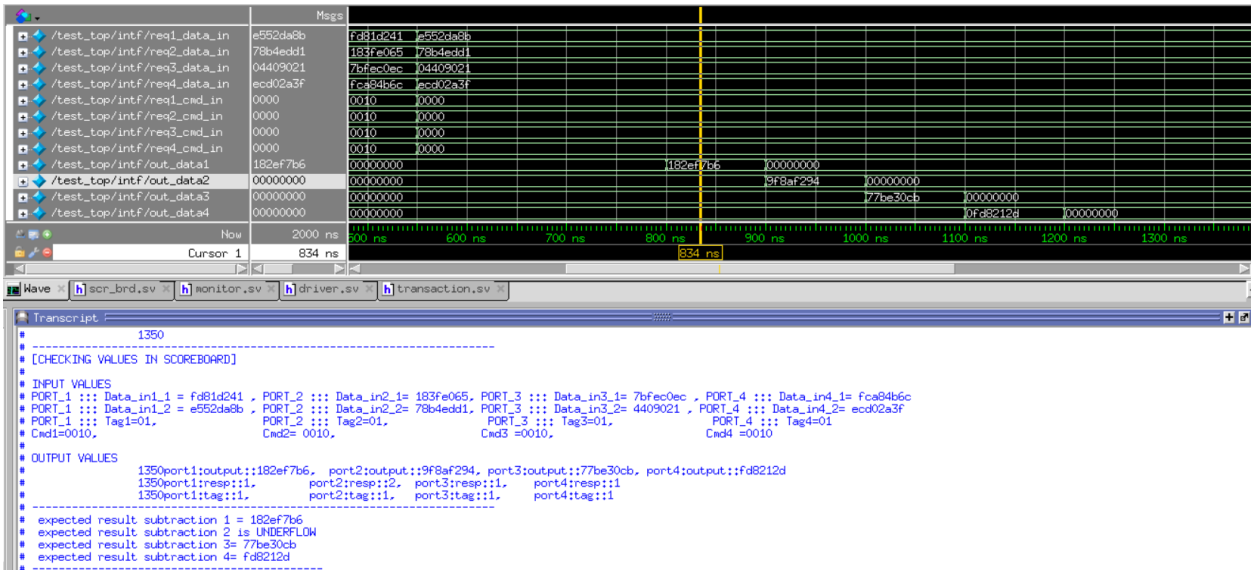

Figure 46: Subtraction command for all ports

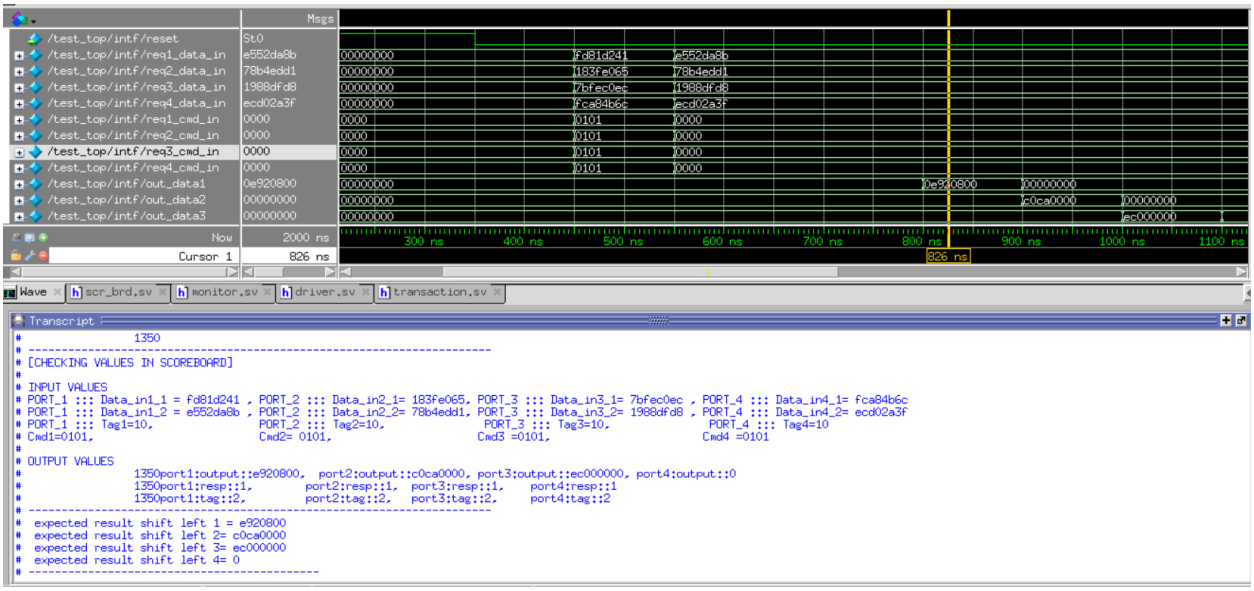

Figure 47: Shift left command for all ports

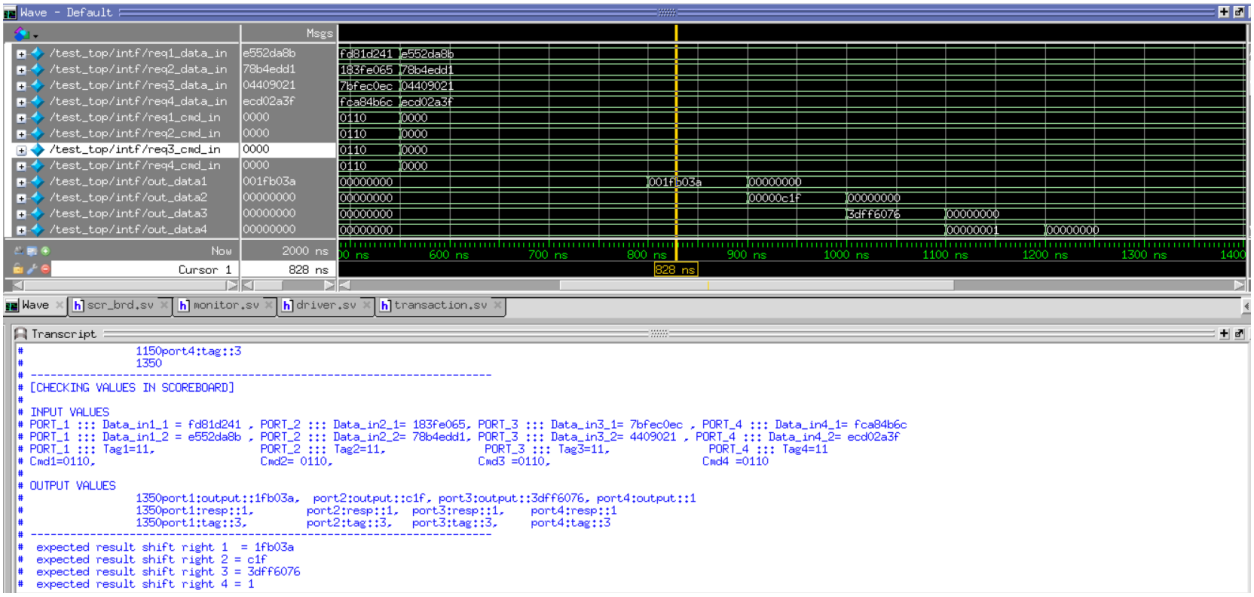

Figure 48: Shift right command for all ports

8 Results and discussion

Firstly, during our analysis of the output we have noticed that the output needs two cycles minimum to be out, so a case was built in the monitor in order to sample the outputs on different cycles needed. In addition to that, since as mentioned earlier the output can be obtained, then we can conclude that the reset is working fine following the reset requirements mentioned in the specification.

Overall, in scenario of randomizing the input and the commands, all ports can do the addition, subtraction, shift left and right and giving correct output, response, and tag. We have noticed that the first iteration always gives correct outputs for all ports for any command; however, all ports are resulting later in having correct iterations and having some cases of 0 output and response of 2'b10, which is a response of underflow, overflow, or invalid command. And this case is a bug since we have noticed this case is happening for ports even though there should be none of the 2'b10 cases. The figure below shows the counting numbers of error for all ports when commands are randomized. It is clear from the result that in general all the ports have almost the same performance.

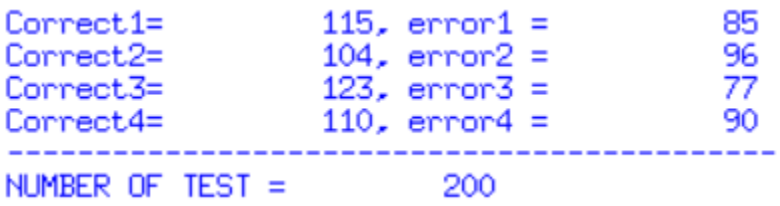

Figure 49: Correct & Error count for all ports

On the other hand, in scenario of not randomizing the command and sending specific arithmetic request, the first iteration will always lead to have correct answers, but the remaining iterations always lead to wrong answer, which is a case that didn't occur when the commands were randomized as mentioned earlier.

The table below summarizes the cases results and bug report.

Specification/Feature	Port	Expected Results	Actual Results
Invalid Command	Port 1	Output of 0 with response '10'b	Output (NO BUG) of 0 with response (NO BUG) '10'b
	Port 2		Output (NO BUG) of 0 with response (NO BUG) '10'b
	Port 3		Output (NO BUG) of 0 with response (NO BUG) '10'b
	Port 4		Output (NO BUG) of 0 with response (NO BUG) '10'b

Add	Port 1	1-Without overflow, output as calculated in scoreboard with response '01'b and tag '00'b 2-With overflow response '10'b	1-Without overflow, output as in scoreboard with response '01'b with tag '00'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With overflow, output of 0 with response '10'b (NO BUG) with tag '00'b (NO BUG)
	Port 2		1-Without overflow, output as in scoreboard with response '01'b with tag '00'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With overflow, output of 0 with response '10'b (NO BUG) with tag '00'b (NO BUG)
	Port 3		1-Without overflow, output as in scoreboard with response '01'b with tag '00'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With overflow, output of 0 with response '10'b (NO BUG) with tag '00'b (NO BUG)
	Port 4		1-Without overflow, output as in scoreboard with response '01'b with tag '00'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With overflow, output of 0 with response '10'b (NO BUG) with tag '00'b (NO BUG)
Subtract	Port 1	1-Without underflow, output as calculated in scoreboard with response '01'b and tag '01'b 2-With underflow response '10'b	1-Without underflow, output as in scoreboard with response '01'b and tag '01'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With underflow, output of 0 with response '10'b with tag '01'b (NO BUG)

	Port 2		1-Without underflow, output as in scoreboard with response '01'b and tag '01'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With underflow, output of 0 with response '10'b with tag '01'b (NO BUG)
	Port 3		1-Without underflow, output as in scoreboard with response '01'b and tag '01'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With underflow, output of 0 with response '10'b with tag '01'b (NO BUG)
	Port 4		1-Without underflow, output as in scoreboard with response '01'b and tag '01'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) with correct tag (NO BUG) 3-With underflow, output of 0 with response '10'b with tag '01'b (NO BUG)
Shift Left	Port 1	1- Output as calculated in scoreboard with response '01'b and tag '10'b	1-Output as in scoreboard with response '01'b and tag '10'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
	Port 2		1-Output as in scoreboard with response '01'b and tag '10'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
	Port 3		1-Output as in scoreboard with response '01'b and tag '10'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)

	Port 4		1-Output as in scoreboard with response '01'b and tag '10'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
Shift Right	Port 1	1- Output as calculated in scoreboard with response '01'b and tag '11'b	1-Output as in scoreboard with response '01'b and tag '11'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
	Port 2		1-Output as in scoreboard with response '01'b and tag '11'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
	Port 3		1-Output as in scoreboard with response '01'b and tag '11'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
	Port 4		1-Output as in scoreboard with response '01'b and tag '11'b (NO BUG) 2-In some cases, output is 0 with response '10'b (BUG) and correct tag (NO BUG)
Addition only for all ports	Port 1,2,3,4	1- Output as calculated in scoreboard with response '01'b	1-Output as in scoreboard with response '01'b for first iteration only (NO BUG) 2-Output is 0 with response '10'b (BUG) for all remaining iterations after the first one.
Subtraction only for all ports	Port 1,2,3,4	1- Output as calculated in scoreboard with response '01'b	1-Output as in scoreboard with response '01'b for first iteration only (NO BUG) 2-Output is 0 with response '10'b (BUG) for all remaining iterations after the first one.

Shift left only for all ports	Port 1,2,3,4	1- Output as calculated in scoreboard with response '01'b	1-Output as in scoreboard with response '01'b for first iteration only (NO BUG) 2-Output is 0 with response '10'b (BUG) for all remaining iterations after the first one.
Shift right only for all ports	Port 1,2,3,4	1- Output as calculated in scoreboard with response '01'b	1-Output as in scoreboard with response '01'b for first iteration only (NO BUG) 2-Output is 0 with response '10'b (BUG) for all remaining iterations after the first one.

9 Conclusion

All in all, the test of calc2 was done through full verification system. The verification was mainly based on randomization in order to cover all possible cases and find unanticipated bugs. Also, the test was done based on different scenarios that analyzes different aspects in the system. All ports were able to execute all arithmetic operation in different iteration, but with some occasions of buggy cases of simulation run.