ISim (P.20131013) - [Default.wcfg]

File   Edit   View   Simulation   Window   Layout   Help

Instance... ↔ □ ⊟ ×   Objects ↔ □ ⊟ ×

Simulation Objects...

Instance and Process

▷ adder_tb
▷ glbl

Object Name

▷ sum[4:0]
   cout
▷ a[4:0]
▷ b[4:0]

Name	Value
▷ sum[4:0]	00000
cout	1
▷ a[4:0]	10000
▷ b[4:0]	10000

0.000 ns

	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30 ns	35 ns
sum[4:0]	00000	11110	00000	11111	00110		10000	
a[4:0]	10000	11111		10101	00011		10000	
b[4:0]	10000	11111	00001	01010	00011		00000	

No overflow. Testing normal functionality.

Our adder module is just a simple sum, so there's nothing too special to be tested. We ran 6 tests to assess core functionality and different overflow cases.

Cases with overflow. Note how the cout (carry) bit is 1.

ADDER

X1: 0.000 ns

Default.wcfg

Console

ISim P.20131013 (signature 0x8ef4fb42)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
Output is 00000 and carry is 1, we expected 00000 and 1
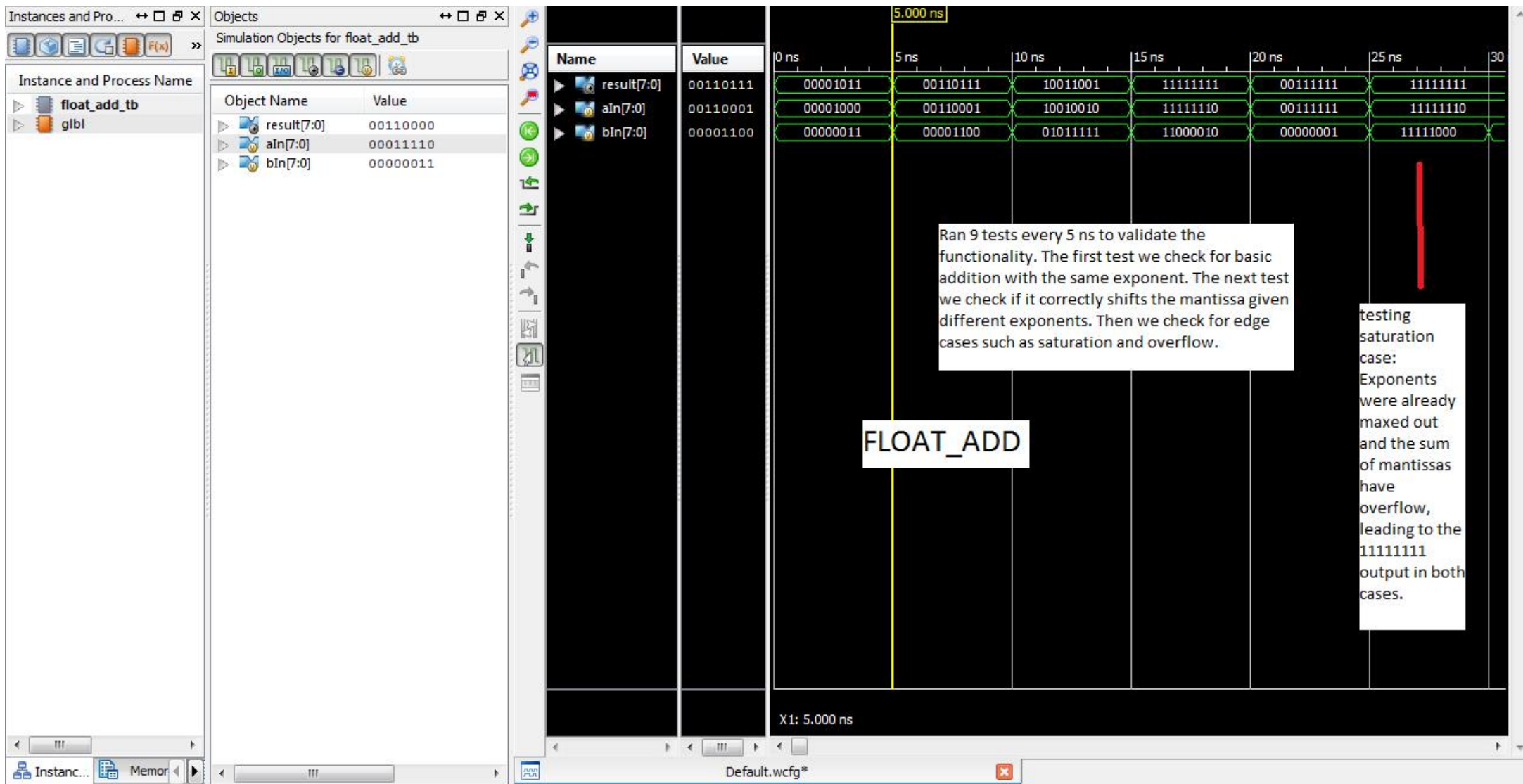Output is 11110 and carry is 1, we expected 11110 and 1
Output is 00000 and carry is 1, we expected 00000 and 1
Output is 11111 and carry is 0, we expected 11111 and 0
Output is 00110 and carry is 0, we expected 00110 and 0
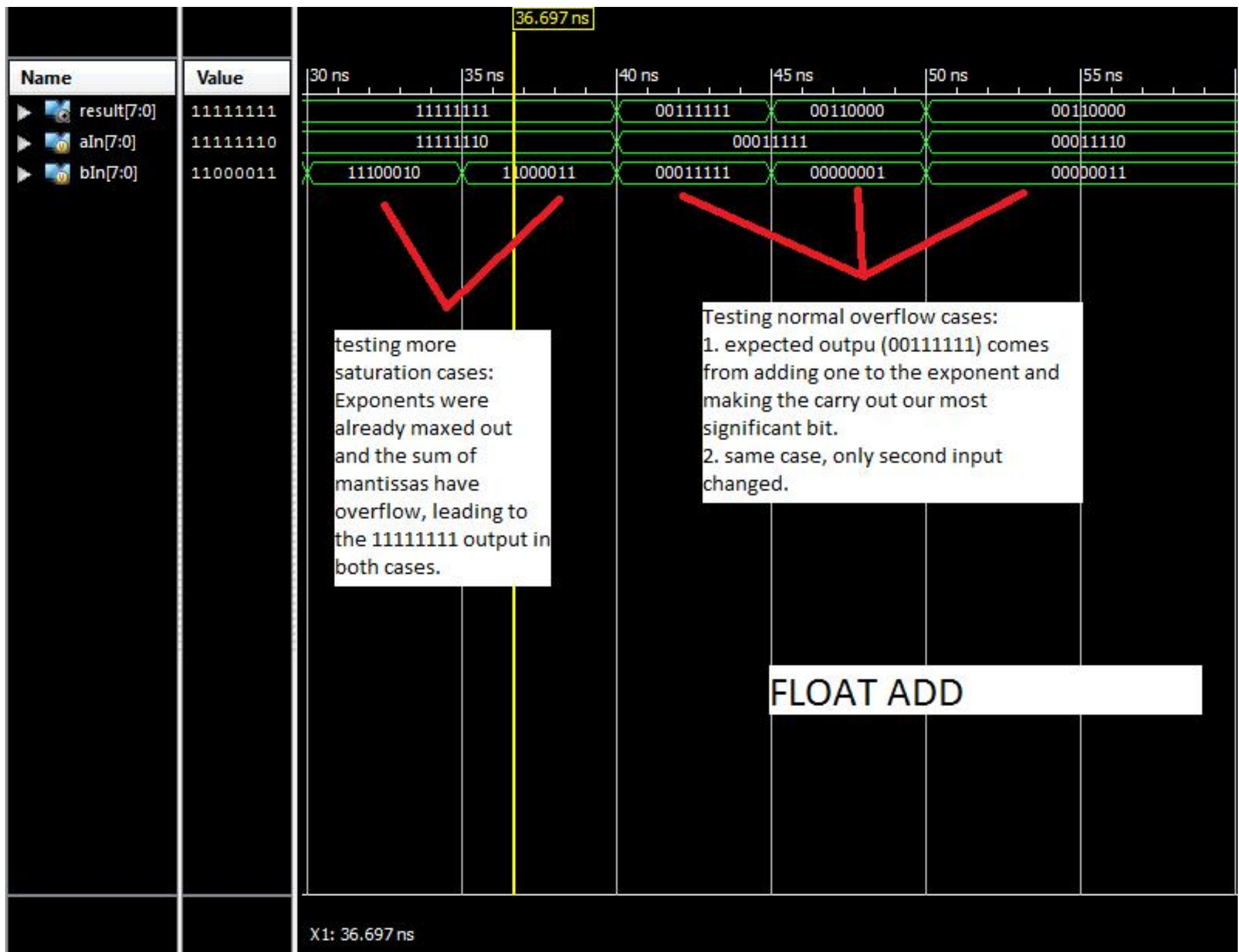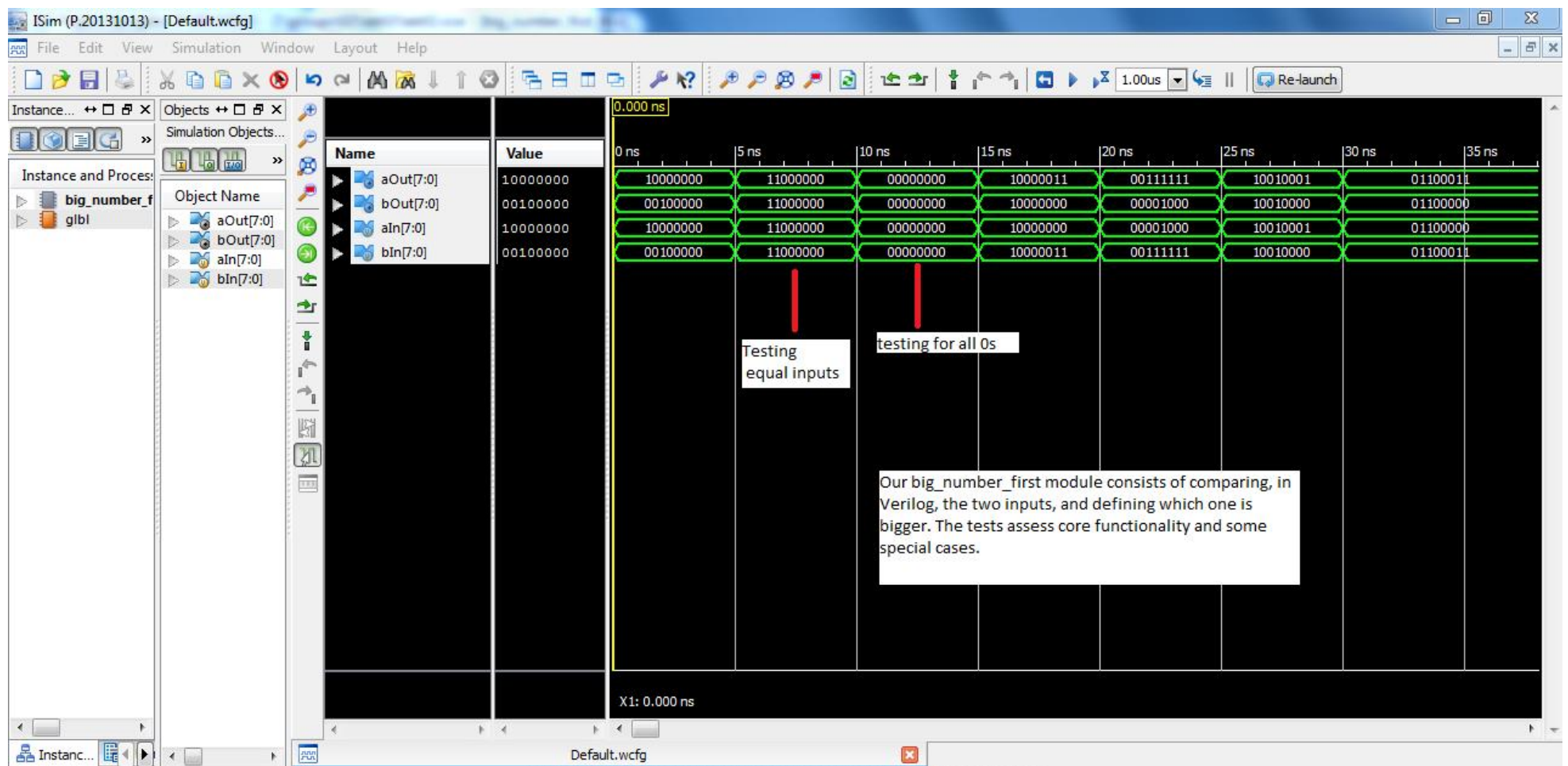Output is 10000 and carry is 0, we expected 10000 and 0
ISim>

F(x) »

Instance and Process Name

▷ float_add_tb
▷ glbl

Instanc... | Memor ◂ ▶

Objects ↔ □ ⊟ ×

Simulation Objects for float_add_tb

Object Name	Value
▷ result[7:0]	00110000
▷ aIn[7:0]	00011110
▷ bIn[7:0]	00000011

5.000 ns

Name	Value	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30
▷ result[7:0]	00110111	00001011	00110111	10011001	11111111	00111111	11111111	
▷ aIn[7:0]	00110001	00001000	00110001	10010010	11111110	00111111	11111110	
▷ bIn[7:0]	00001100	00000011	00001100	01011111	11000010	00000001	11111000	

Ran 9 tests every 5 ns to validate the functionality. The first test we check for basic addition with the same exponent. The next test we check if it correctly shifts the mantissa given different exponents. Then we check for edge cases such as saturation and overflow.

testing saturation case: Exponents were already maxed out and the sum of mantissas have overflow, leading to the 11111111 output in both cases.

FLOAT_ADD

X1: 5.000 ns

Default.wcfg*

Console ↔ □ ⊟ ×

Output is 00001011, we expected 00001011
Output is 00110111, we expected 00110111
Output is 10011001, we expected 10011001
Output is 11111111, we expected 11111111
Output is 00111111, we expected 00111111
Output is 11111111, we expected 11111111
Output is 11111111, we expected 11111111
Output is 11111111, we expected 11111111
Output is 00111111, we expected 00111111
Output is 00110000, we expected 00110000
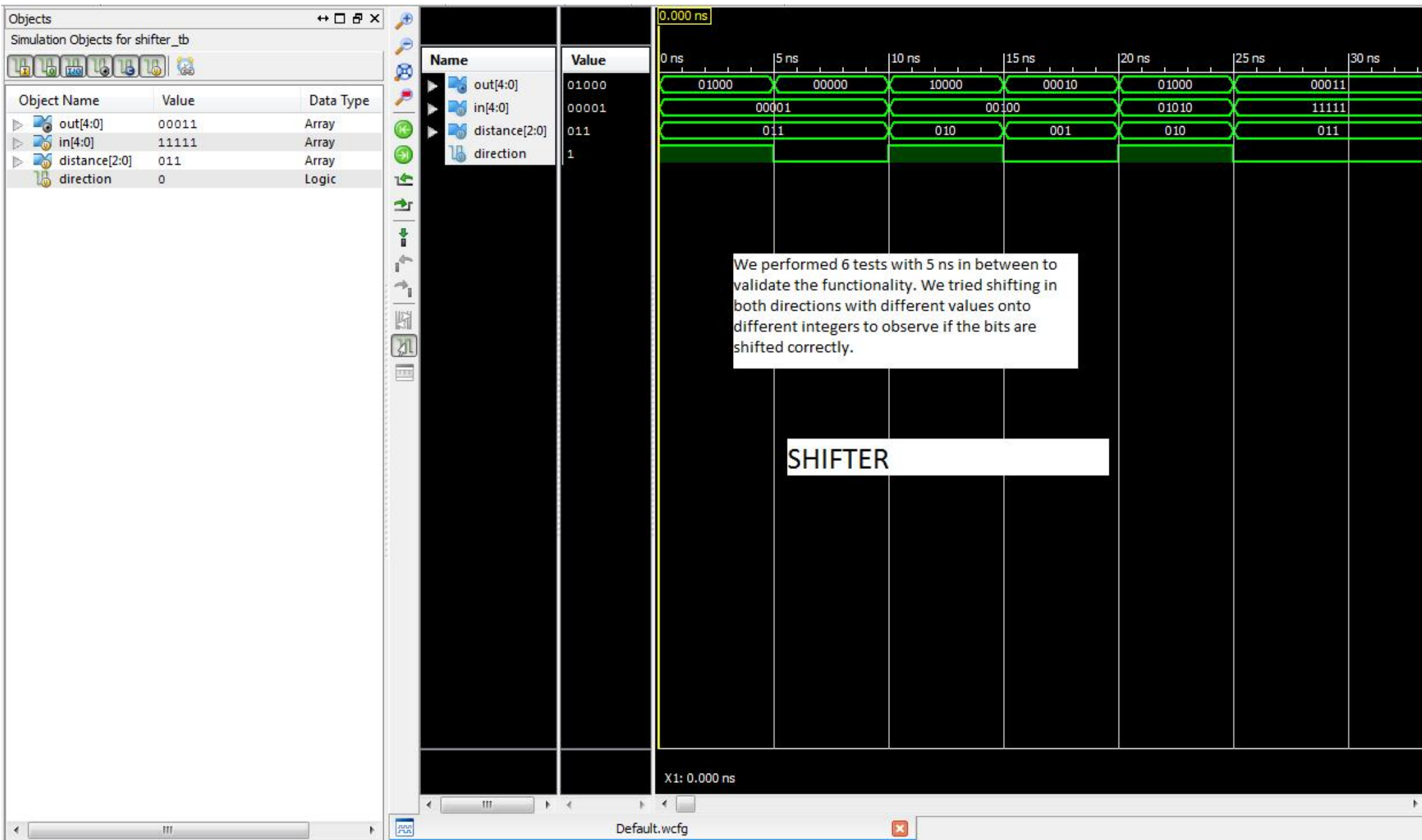Output is 00110000, we expected 00110000
ISim>

Name	Value						
		30 ns	35 ns	40 ns	45 ns	50 ns	55 ns
▶ result[7:0]	11111111	11111111		00111111	00110000	00110000	
▶ aIn[7:0]	11111110	11111110		00011111		00011110	
▶ bIn[7:0]	11000011	11100010	11000011	00011111	00000001	00000011	

36.697 ns

testing more saturation cases: Exponents were already maxed out and the sum of mantissas have overflow, leading to the 11111111 output in both cases.

Testing normal overflow cases:
1. expected outpu (00111111) comes from adding one to the exponent and making the carry out our most significant bit.
2. same case, only second input changed.

FLOAT ADD

X1: 36.697 ns

We performed 6 tests with 5 ns in between to validate the functionality. We tried shifting in both directions with different values onto different integers to observe if the bits are shifted correctly.

SHIFTER

Output is 01000, we expected 01000
Output is 00000, we expected 00000
Output is 10000, we expected 10000
Output is 00010, we expected 00010
Output is 01000, we expected 01000
Output is 00011, we expected 00011
ISim>