Group 2- Patrick Balcombe, Harry Austin, Devashish Bishnoi

**ECS615U-** **Laboratory Session 2**

# **Abstract**

In this lab we implement an ALU and shifter in VHDL code using several lower level devices. We then go on to test them using a test bench file that cycles through several possible combinations of inputs. Then using the Modelsim software to simulate the output results we confirm that the output is as expected for that particular device. Then finally we upload the devices to the physical board and verify that our devices and simulations are correct.

**Method**

The first stage to create the devices was taken in ISE project navigator in which the VHDL code was written for the actual devices and their test benches. Through this we then also simulated the devices in ISim to gain timing diagrams for our test benches to confirm that the devices were working as expected according to the derived truth table. Plan Ahead was then used to define user constraints files that map the inputs out outputs of the top level devise to the FPGM board that we would later upload the devices to. We configure the FPGM board and upload the devise (which in this lab were the ALU and shifter) using iPACT tools after which the devices actual inputs and outputs where then tested.

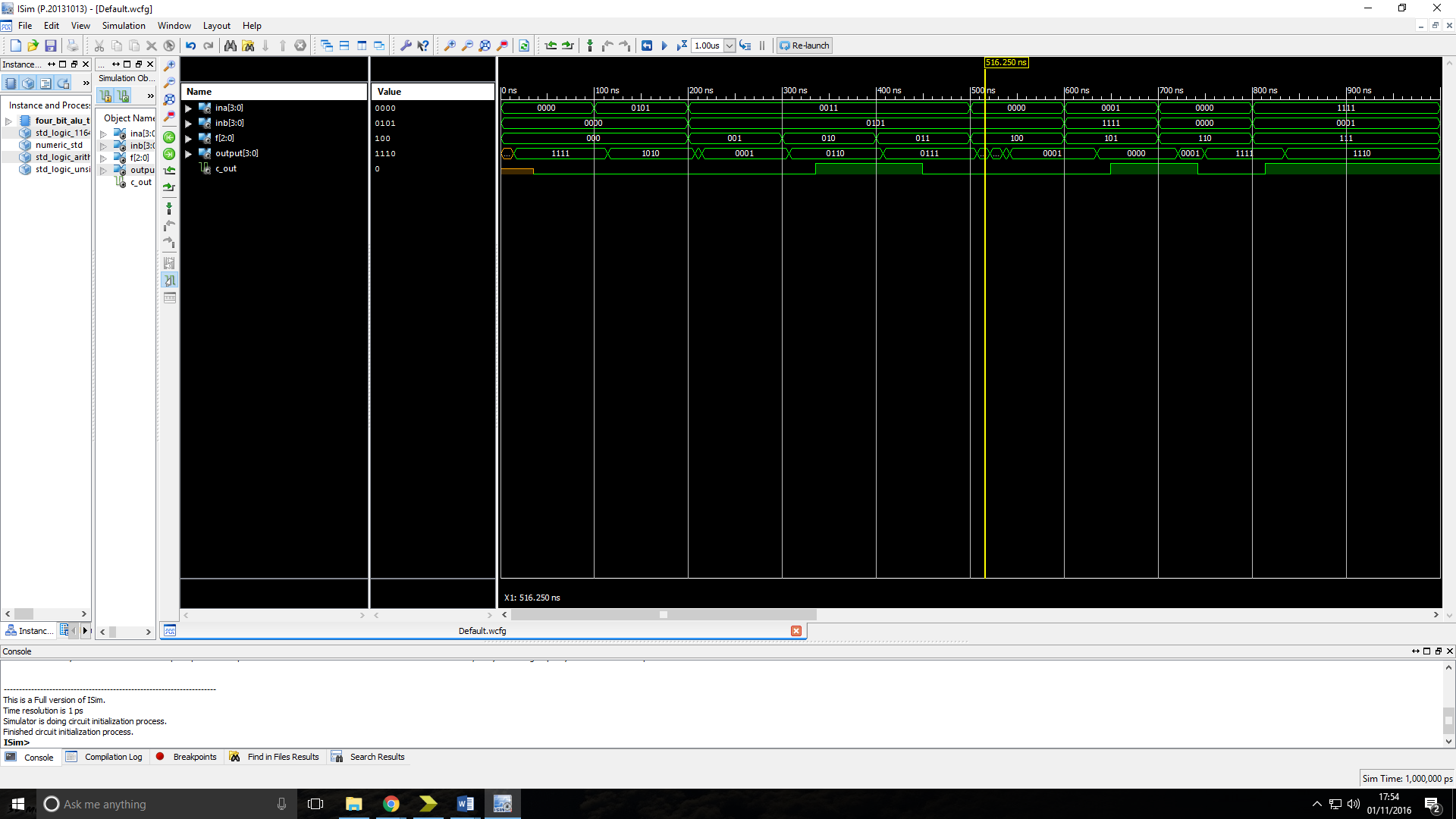
The devices higher up in the design such as the ALU and Shifter use the functionality of the devises lower down in the design such as the Adder/subtractor to create a hierarchical design. Devices that are further down are port mapped, defining their connections within the device, into the higher level device so that it can be passed the signals necessary to carry out its function.

# Simulation

In this section we individually run the test bench files for each device in Modelsim to check the timing diagram of the I/O matches its truth table. The Truth tables list possible combinations of inputs and the relevant output which are compared to the results gained from the simulation.

# Timing Diagrams

**4-bit ALU:**

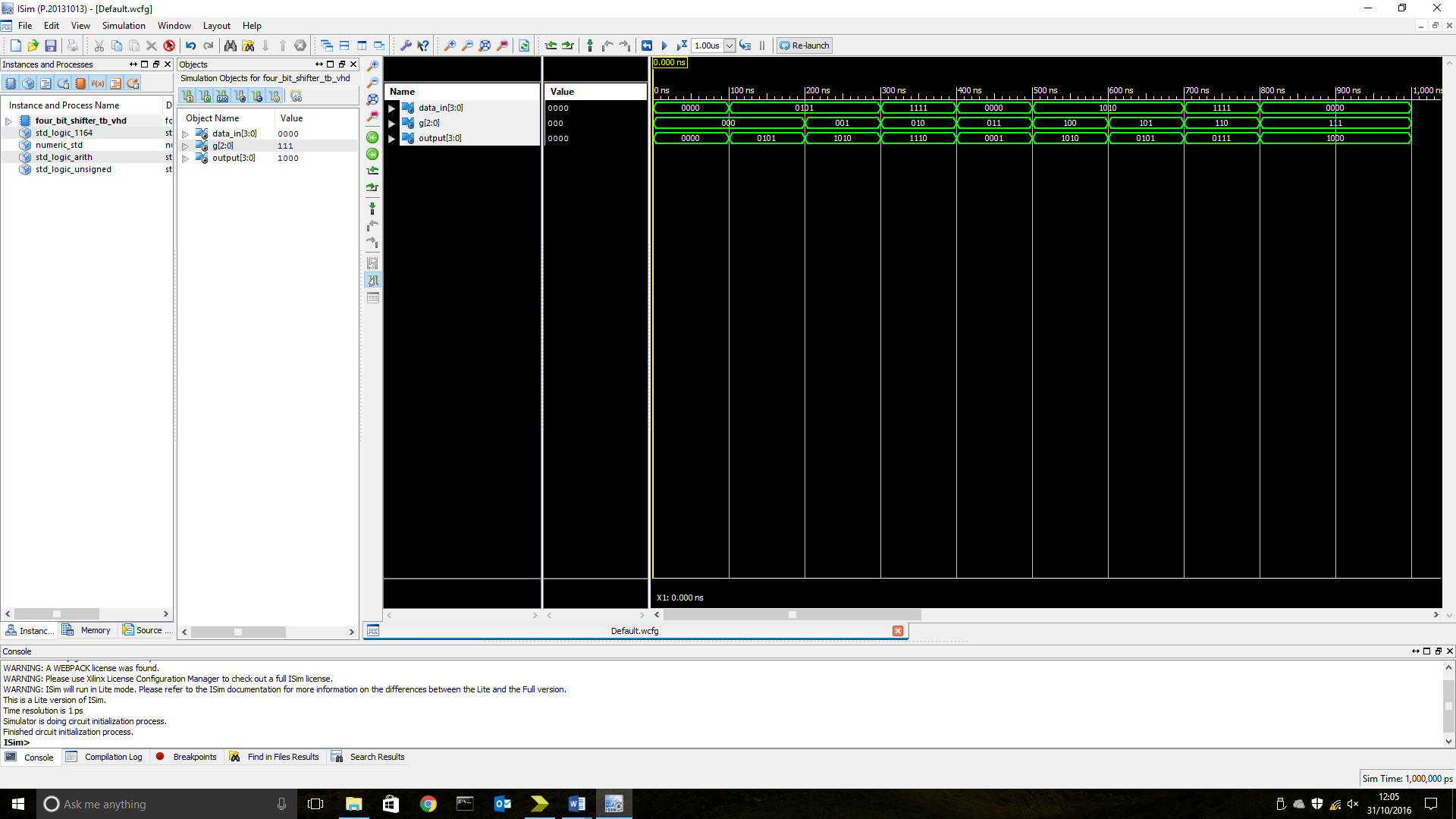
Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | | | | | | | | Output | | | | |
| F | | | InA | | | | InB | | | | Output | | | | C\_out |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | x | x | x | x | 1 | 0 | 1 | 0 | X |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | X |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | X |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | X |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

Figure 1: Simulation timing diagram of 4-bit ALU showing its behaviour when tested with a range of possible inputs.

Analysis: The final output of the simulation was as expected for the devices behaviour but with with quite a long delay for C\_out. Originally there was a mistake in the design provided that meant that the first four functions where switched with the second four, this was rectified using a not gate on that control line. As well as that there was a mistake in the truth table provided in the last two rows of C\_out that was found and rectified when we got into the lab.

**4-bit Shifter:**

Truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| Function | G | Data\_In | Output |
| Pass | 000 | 0101 | 0101 |
| Rotate Left | 001 | 0101 | 1010 |
| Shift Left (0) | 010 | 1111 | 1110 |
| Shift Left (1) | 011 | 0000 | 0001 |
| Pass | 100 | 1010 | 1010 |
| Rotate Right | 101 | 1010 | 0101 |
| Shift Right (0) | 110 | 1111 | 0111 |
| Shift Right (1) | 111 | 0000 | 1000 |

Figure 2: Simulation timing diagram of 4-bit Shifter showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the simulation was as expected for the devices behaviour

**4-bit Arithmetic Unit:**

Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | | Outputs | | | | |
| InA | | | | InB | | | | Control | | Sum | | | | C\_out |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

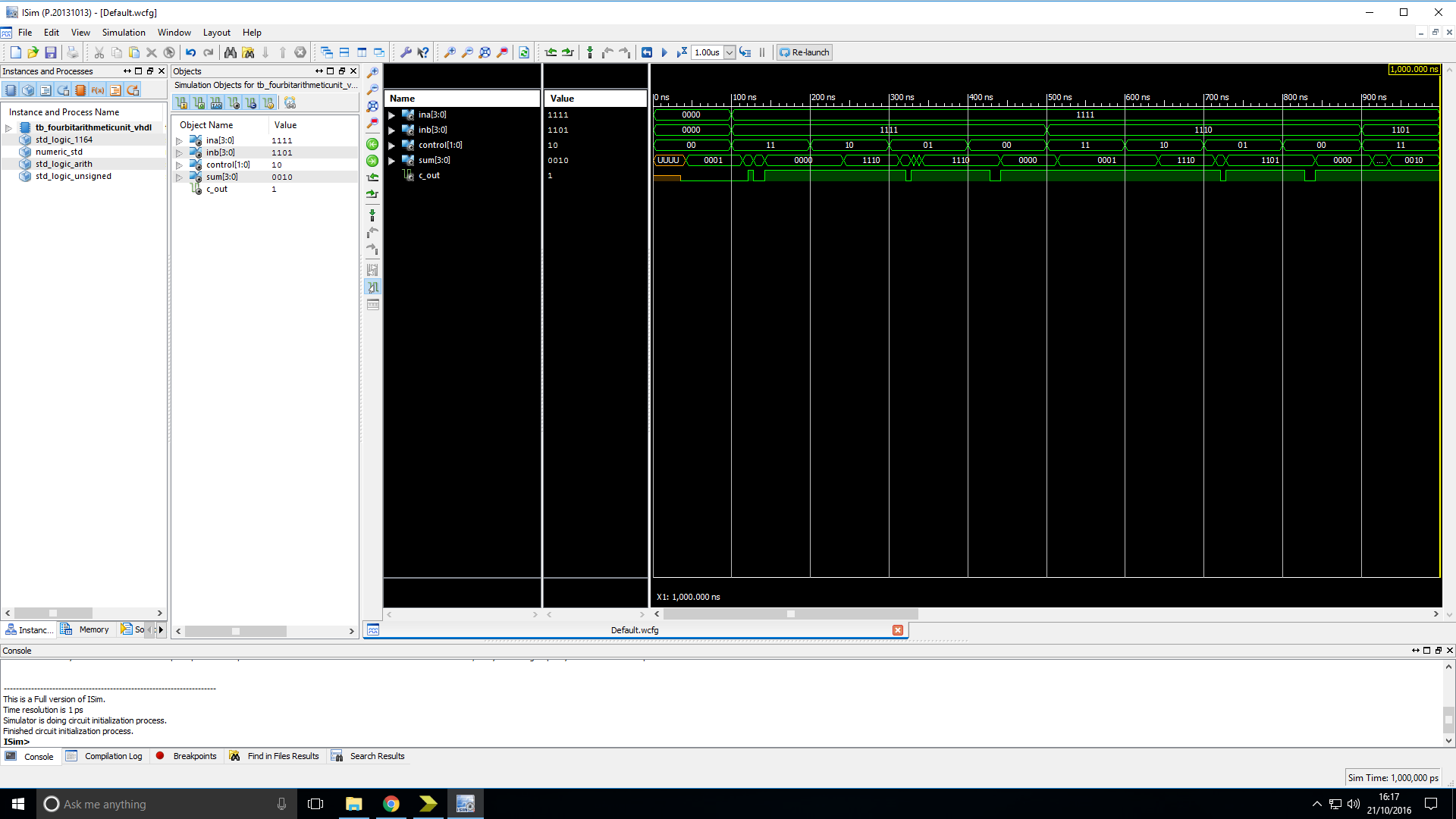
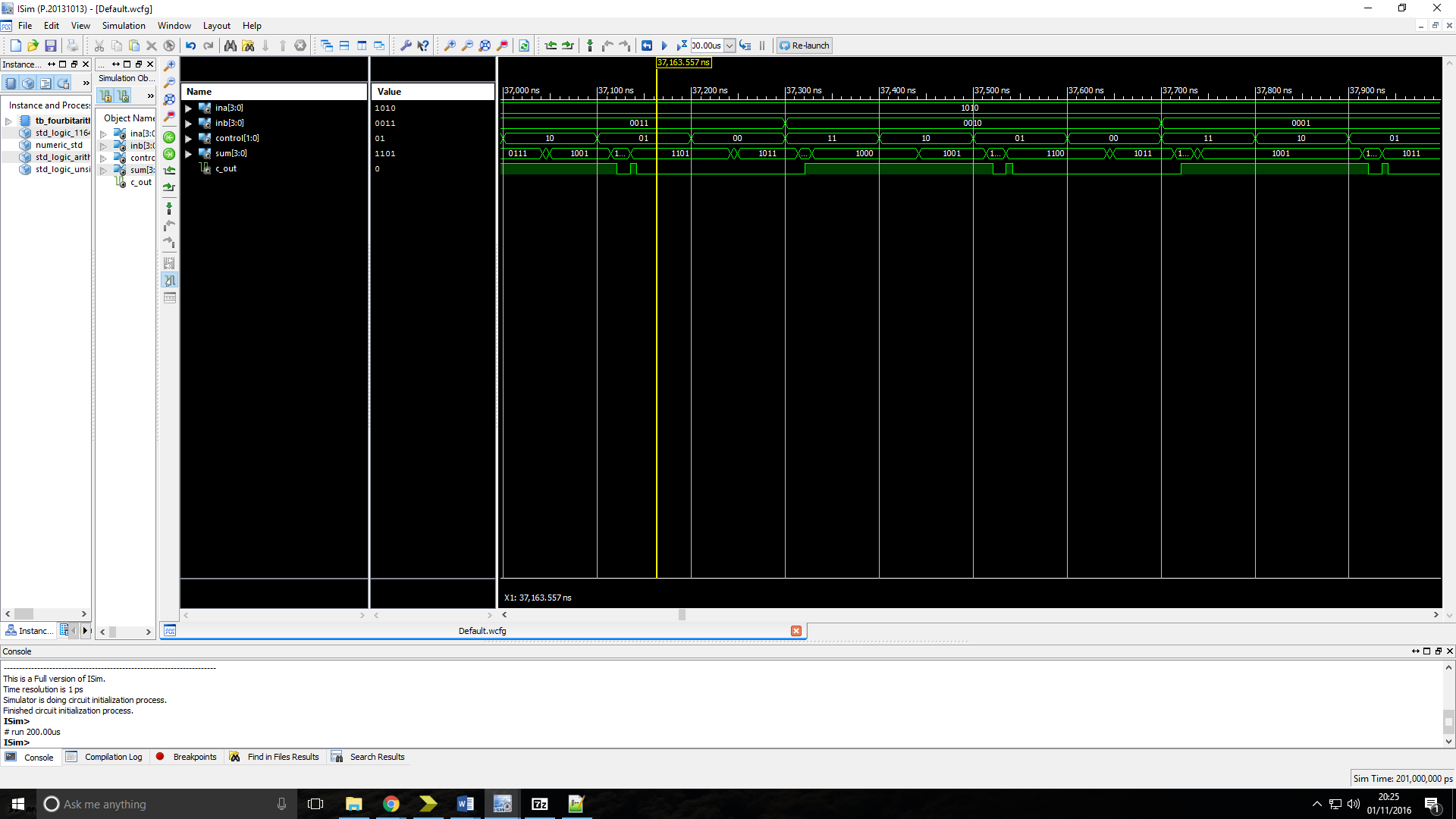


Figure 3: Simulation timing diagram of 4-bit Arithmetic Unit showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the simulation was as expected for the devices behaviour.

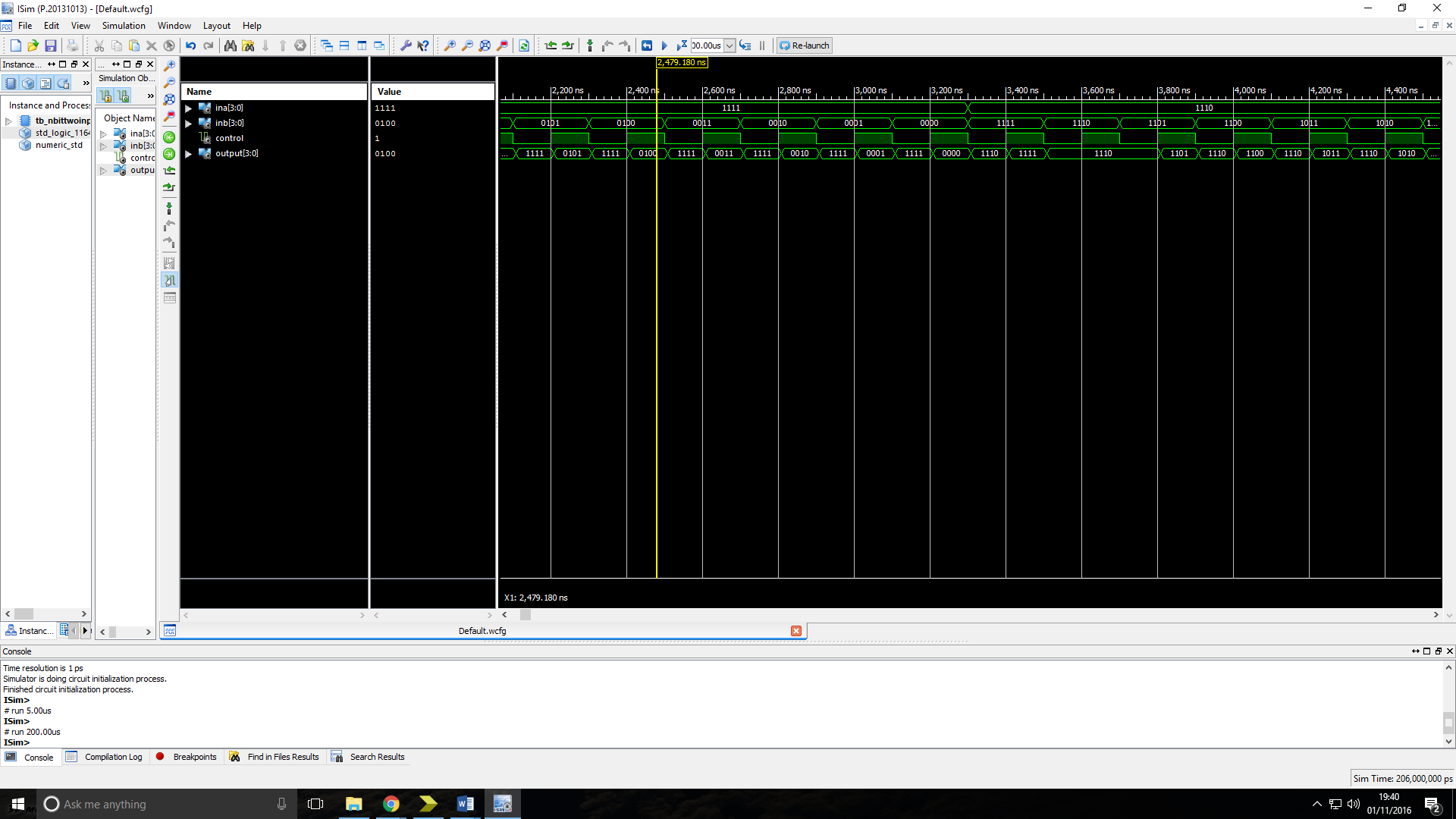
**Logic Unit:**

Truth table:

Figure 4: Simulation timing diagram of Logic Unit showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the simulation was as expected for the devices behaviour

**n-bit 2-input multiplexer:**

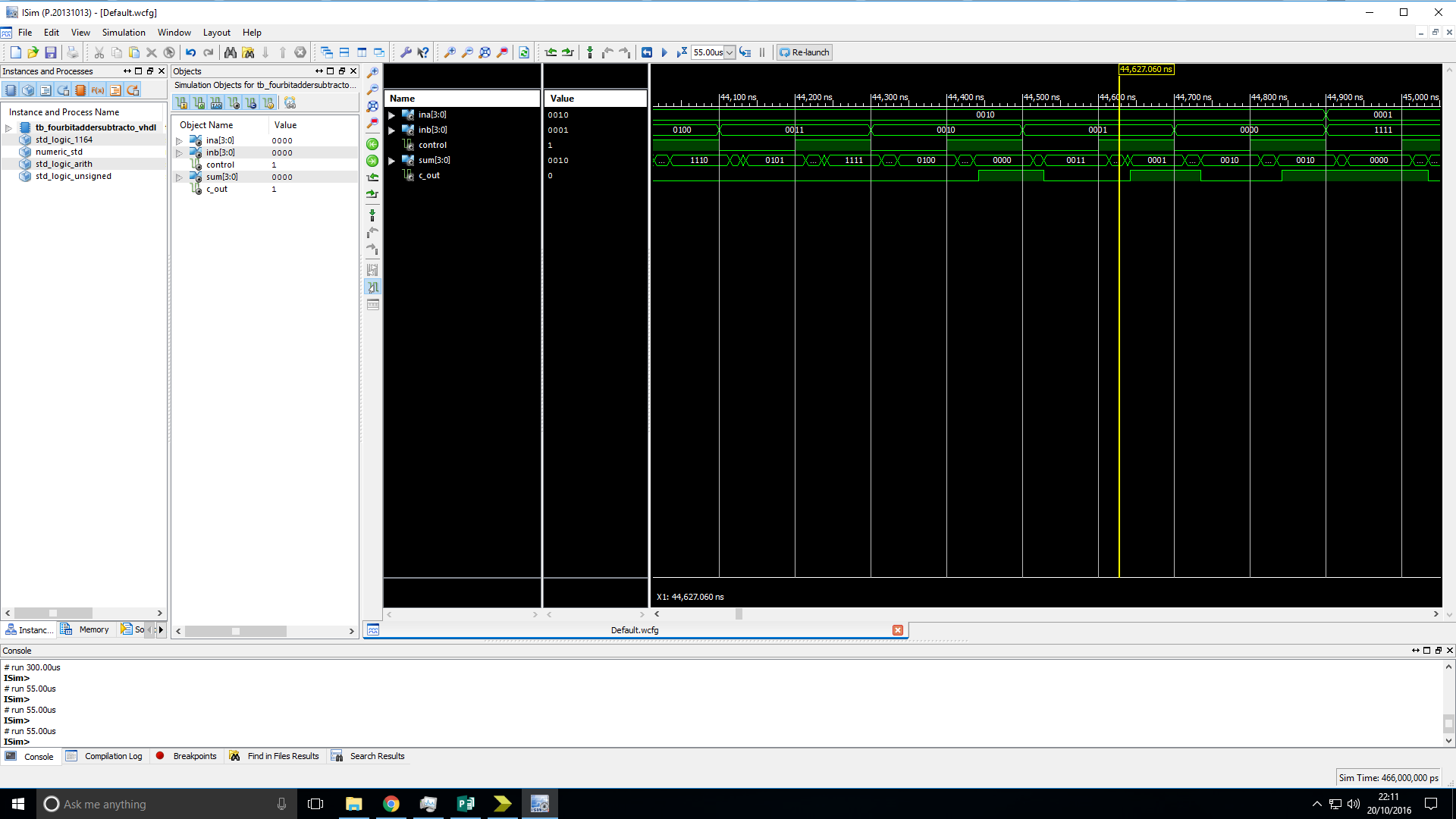
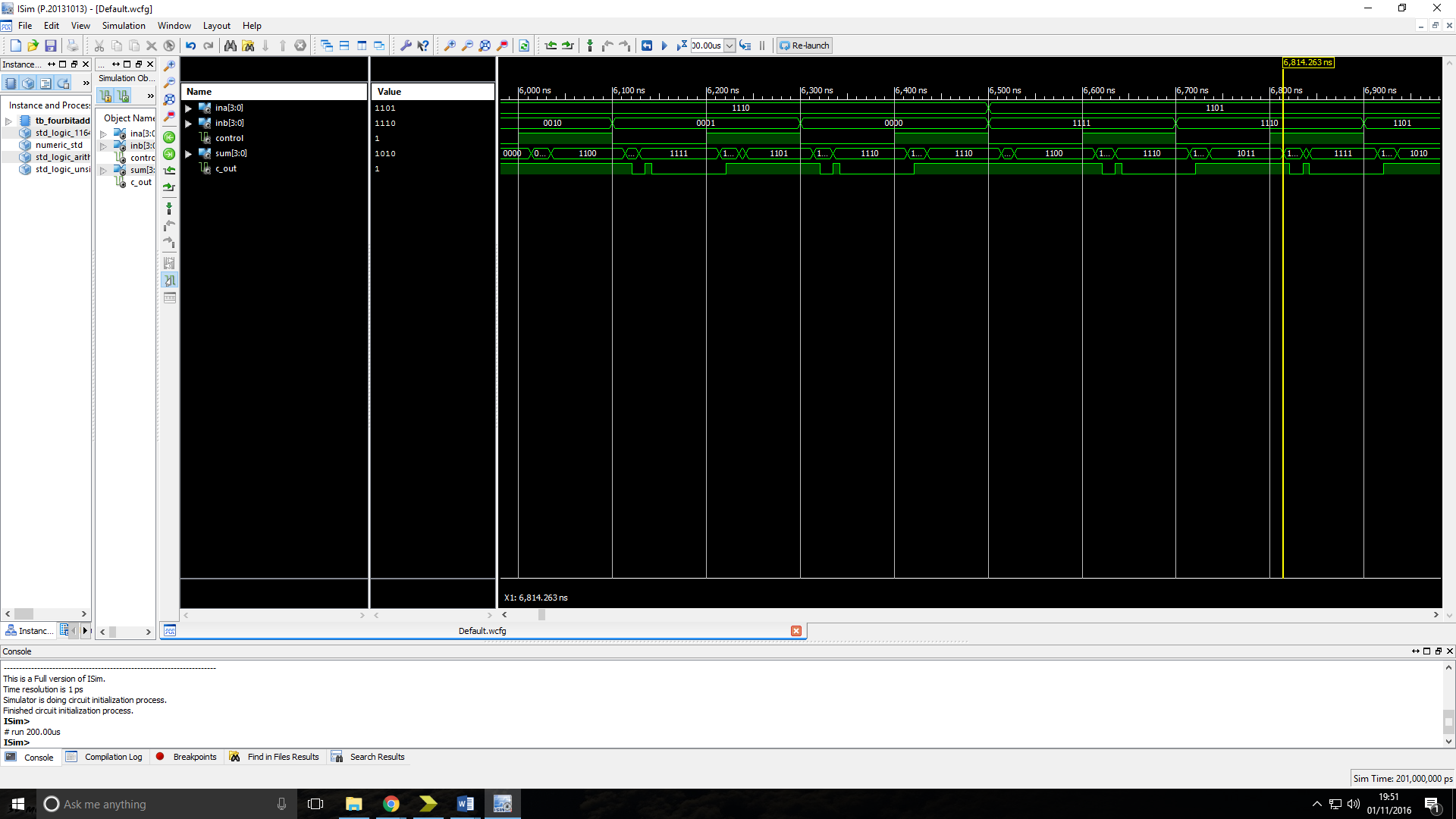
Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | Outputs | | | |
| InA | | | | InB | | | | Control | Sum | | | |
| 1 | 1 | 1 | 1 | x | x | x | x | 0 | 1 | 1 | 1 | 1 |
| x | x | x | x | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | x | x | x | x | 0 | 0 | 1 | 1 | 1 |
| x | x | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

Figure 5: Simulation timing diagram of n-bit 2-input multiplexer Unit showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the simulation was as expected for the devices behaviour.

**4-bit Adder/Subtractor:**

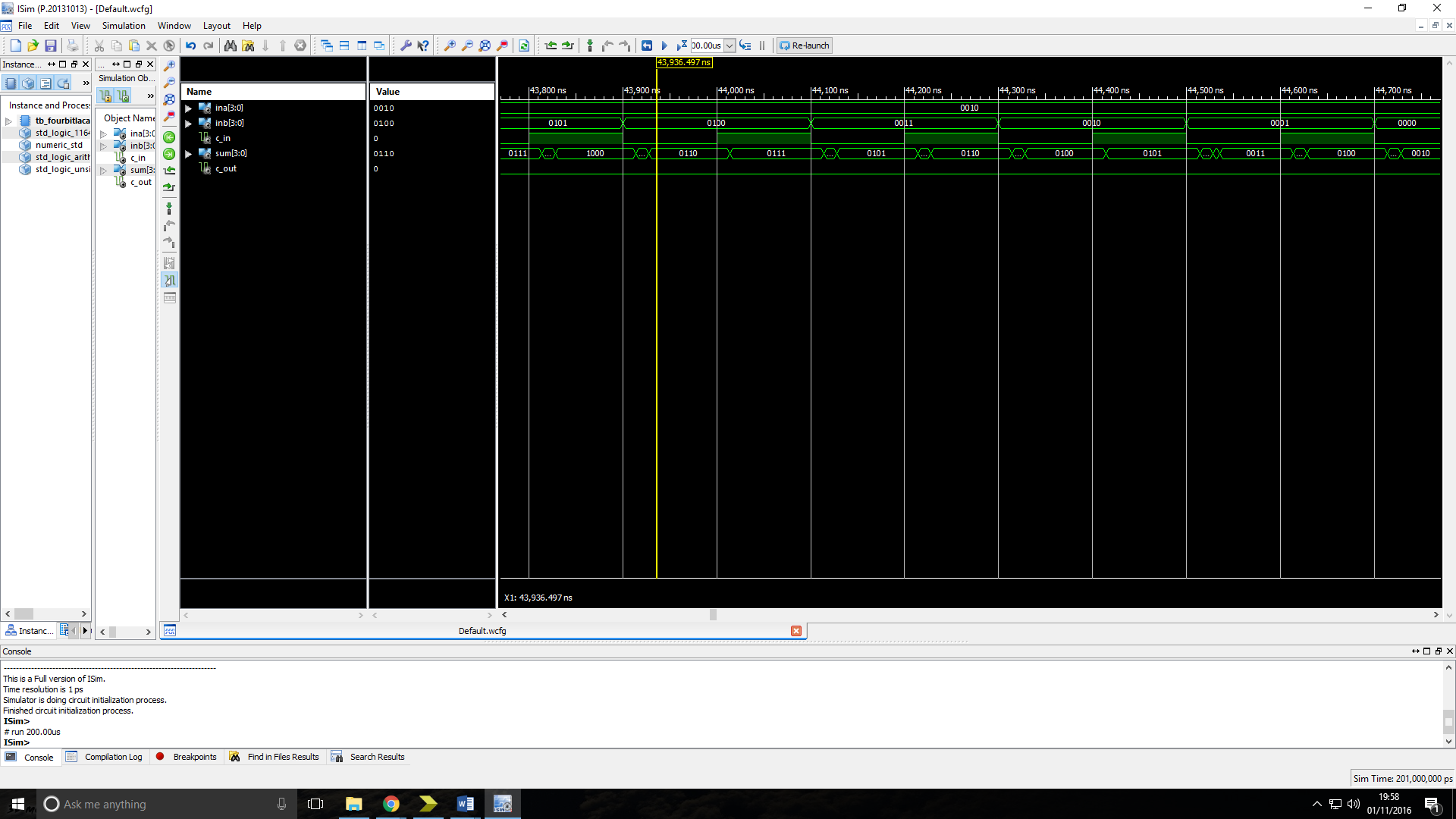
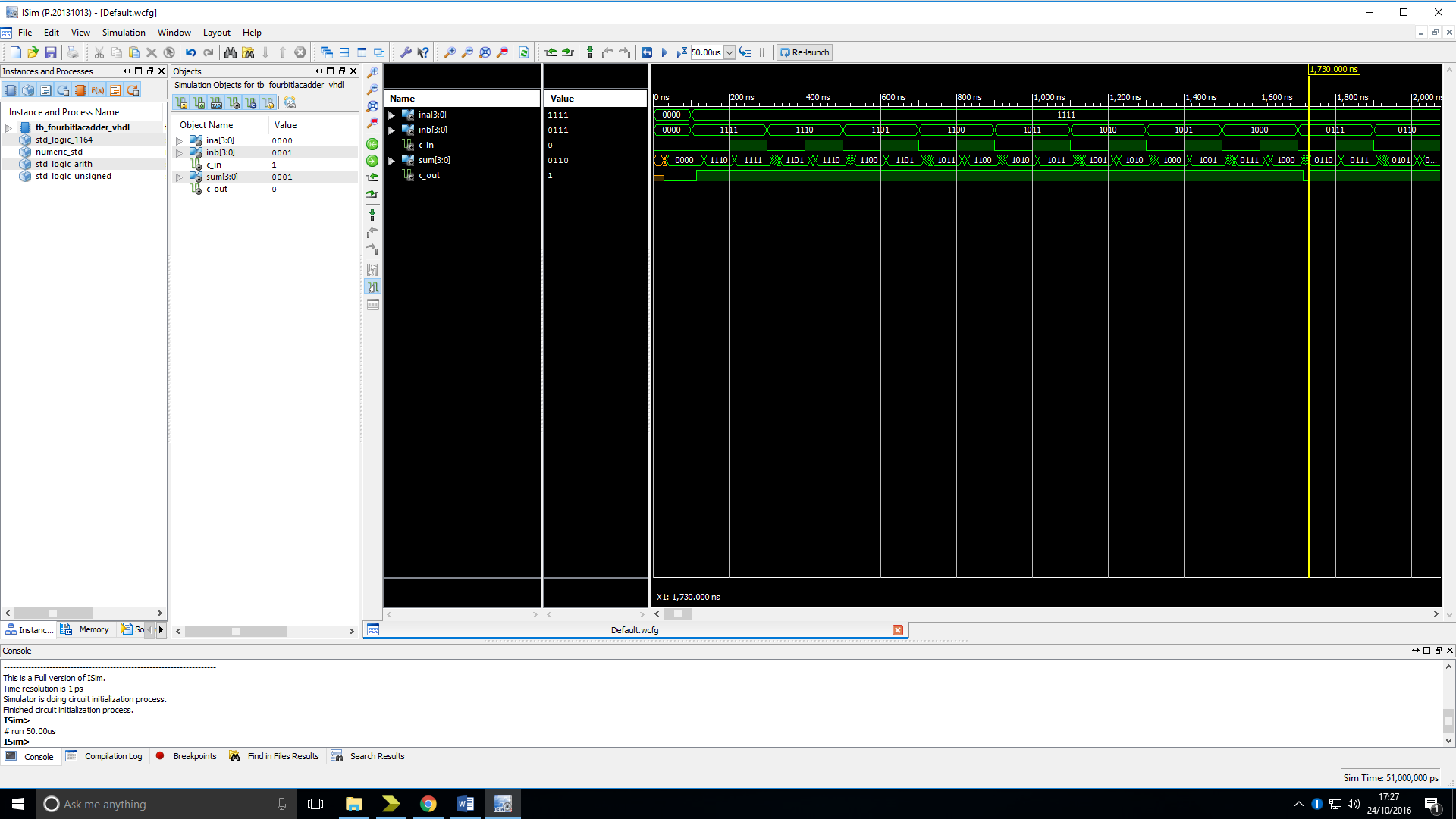
Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | Outputs | | | | |
| InA | | | | InB | | | | Control | Sum | | | | C\_out |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

Figure 6: Simulation timing diagram of 4-bit Adder/Subtractor showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the simulation was as expected for the devices behaviour.

**4-bit LAC Adder:**

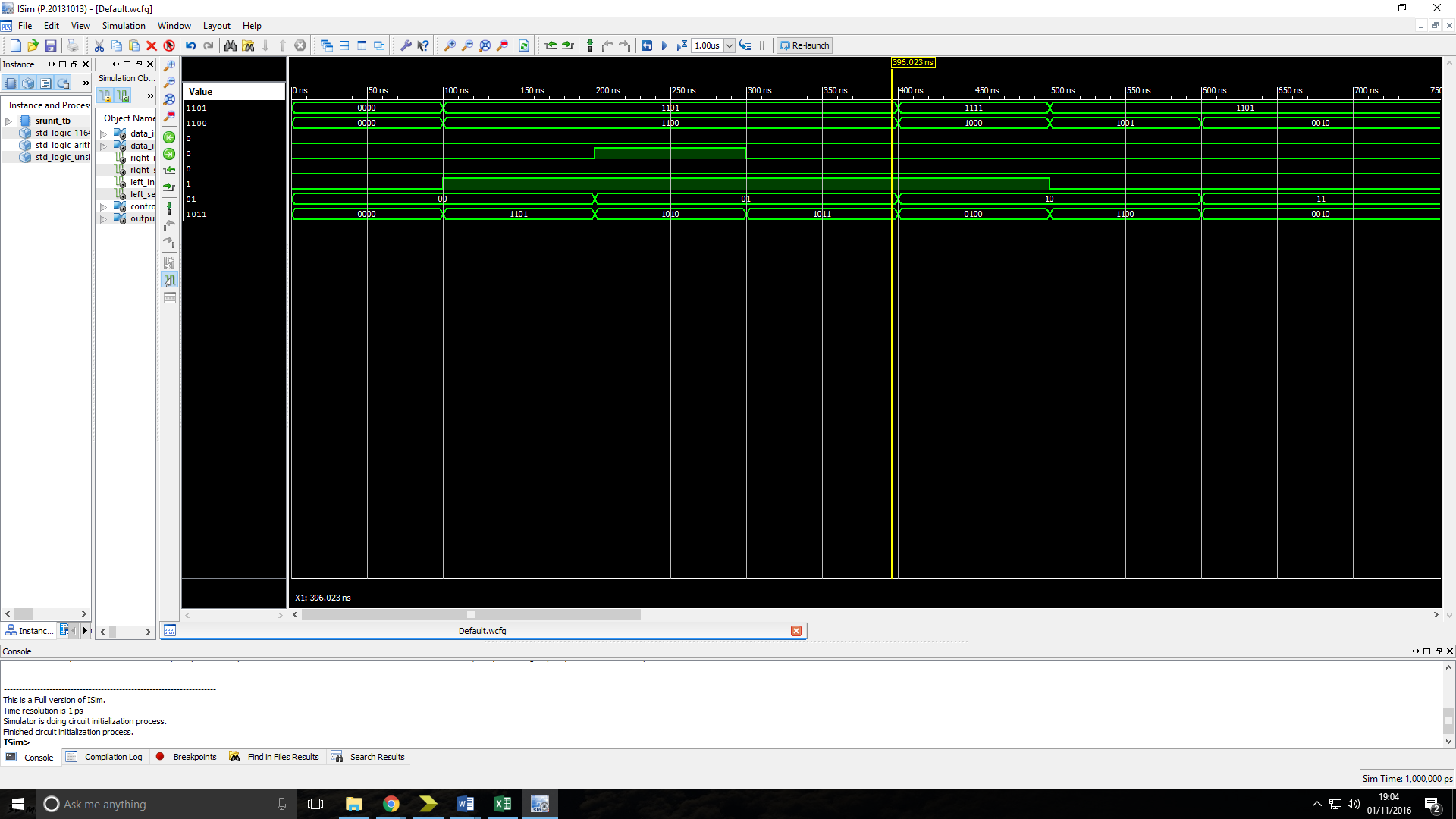
Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | Outputs | | | | |
| InA | | | | InB | | | | C\_in | Sum | | | | C\_out |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 7: Simulation timing diagram of 4-bit LAC Adder showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**n-bit Shift/Rotator:**

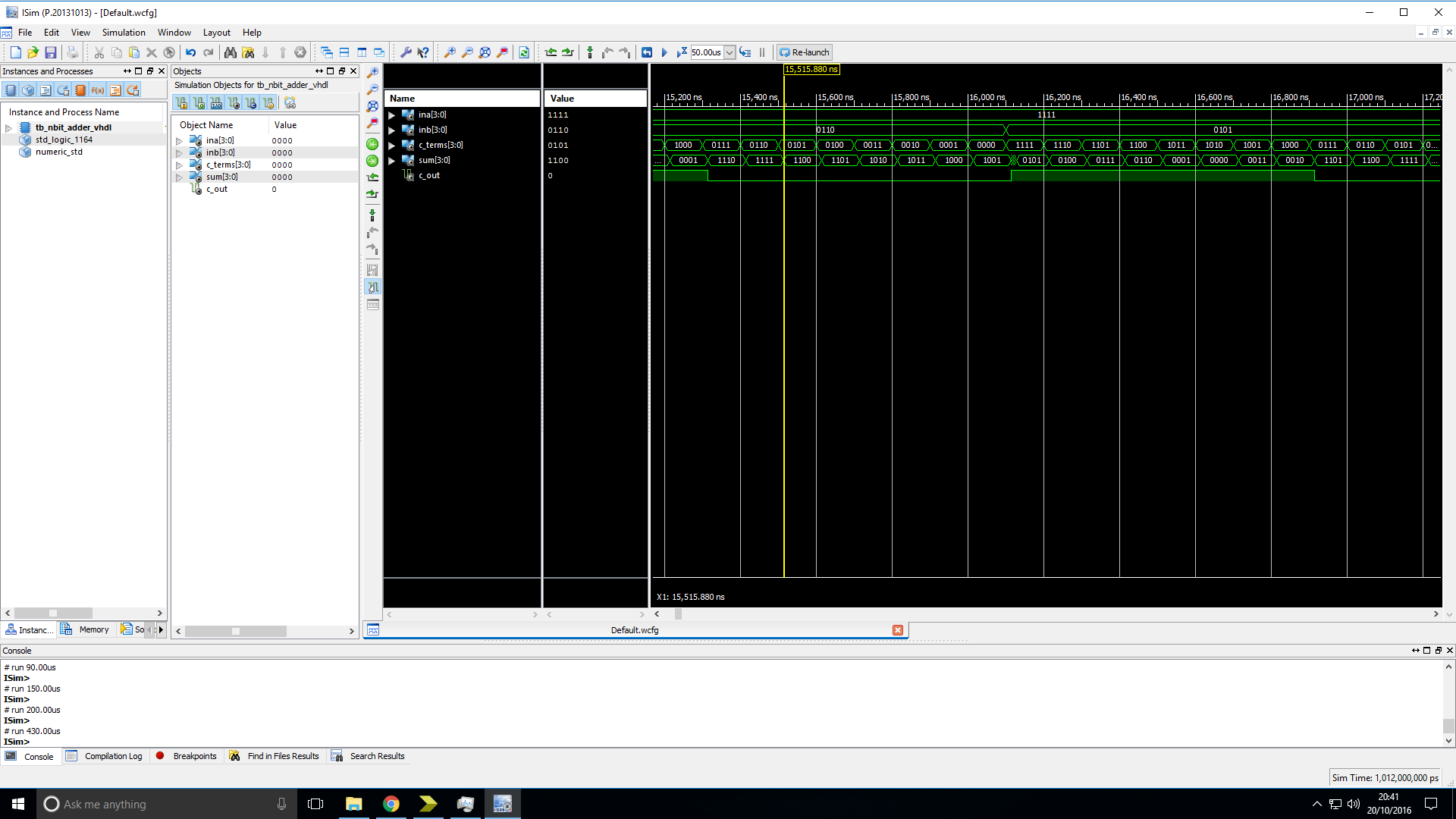
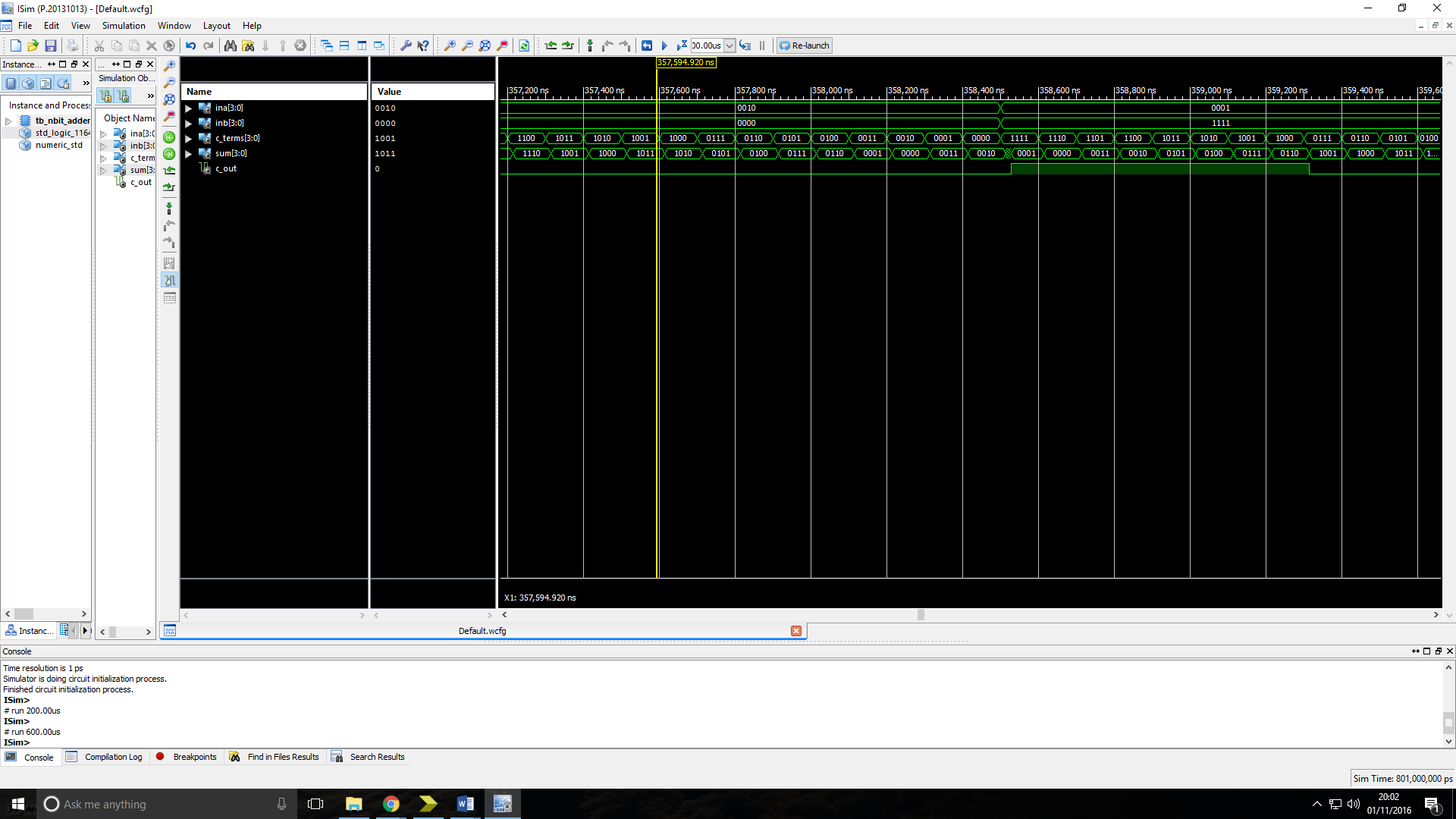
Truth table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | Output |
| Data\_ln1 | Data\_ln2 | Right\_In | Right\_Select | Left\_In | Left\_Select | Control |
| 1101 | xxxx | x | x | x | x | 00 | 1101 |
| 1101 | xxxx | 0 | 1 | x | x | 01 | 1010 |
| 1101 | xxxx | x | 0 | x | x | 01 | 1011 |
| xxxx | 1000 | x | x | 0 | 1 | 10 | 0100 |
| xxxx | 1001 | x | x | x | 0 | 10 | 1100 |
| xxxx | 0010 | x | x | x | x | 11 | 0010 |

Figure 8: Simulation timing diagram of n-bit Shift/Rotator showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**n-bit Adder:**

Truth table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | | | | Outputs | | | | |
| InA | | | | InB | | | | C\_terms | | | | Sum | | | | C\_out |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

Figure 9: Simulation timing diagram of n-bit Adder showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**logic function bit-slice:**

Truth table:

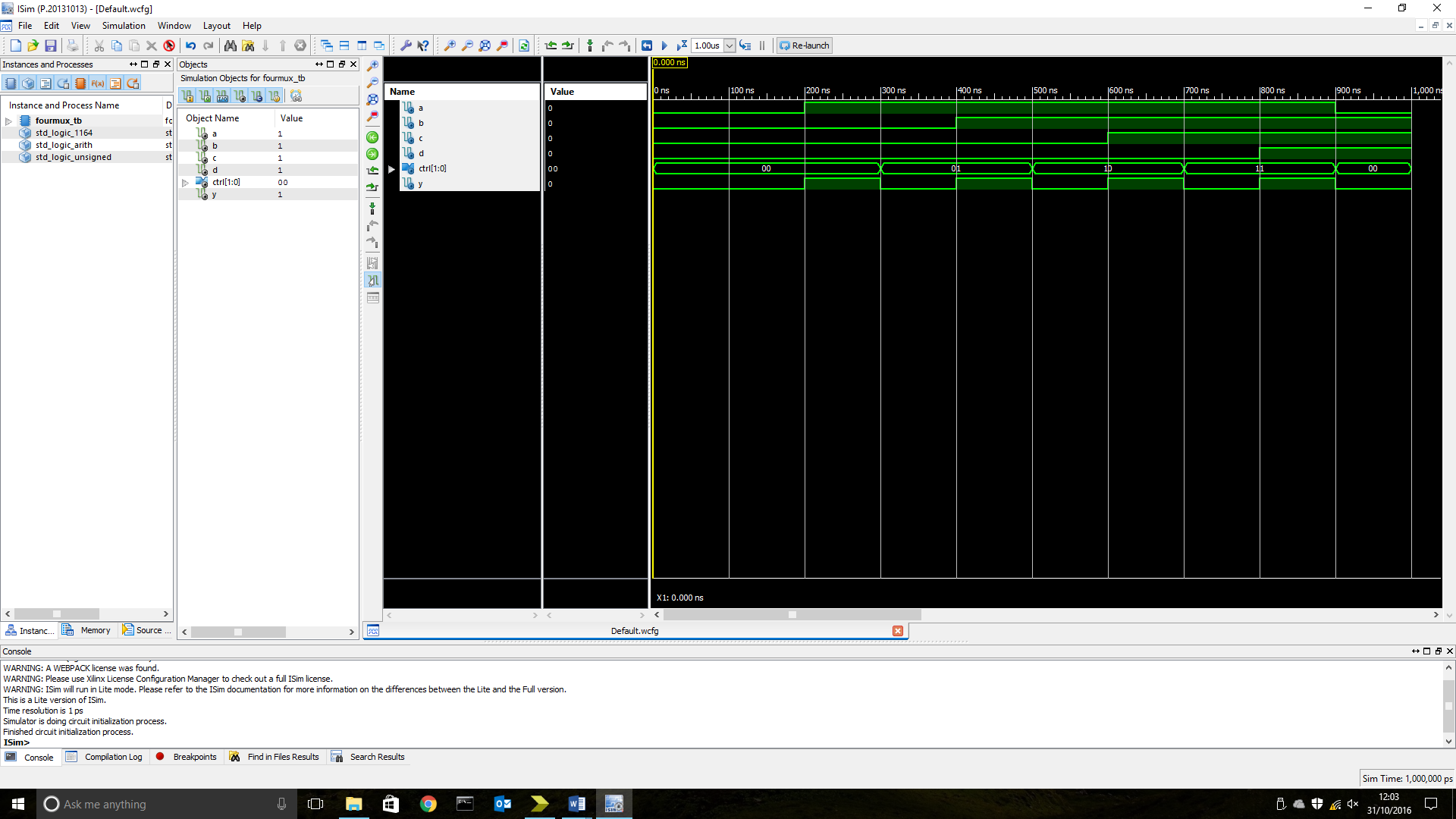
Figure 10: Simulation timing diagram of logic function bit-slice showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**4-input multiplexer:**

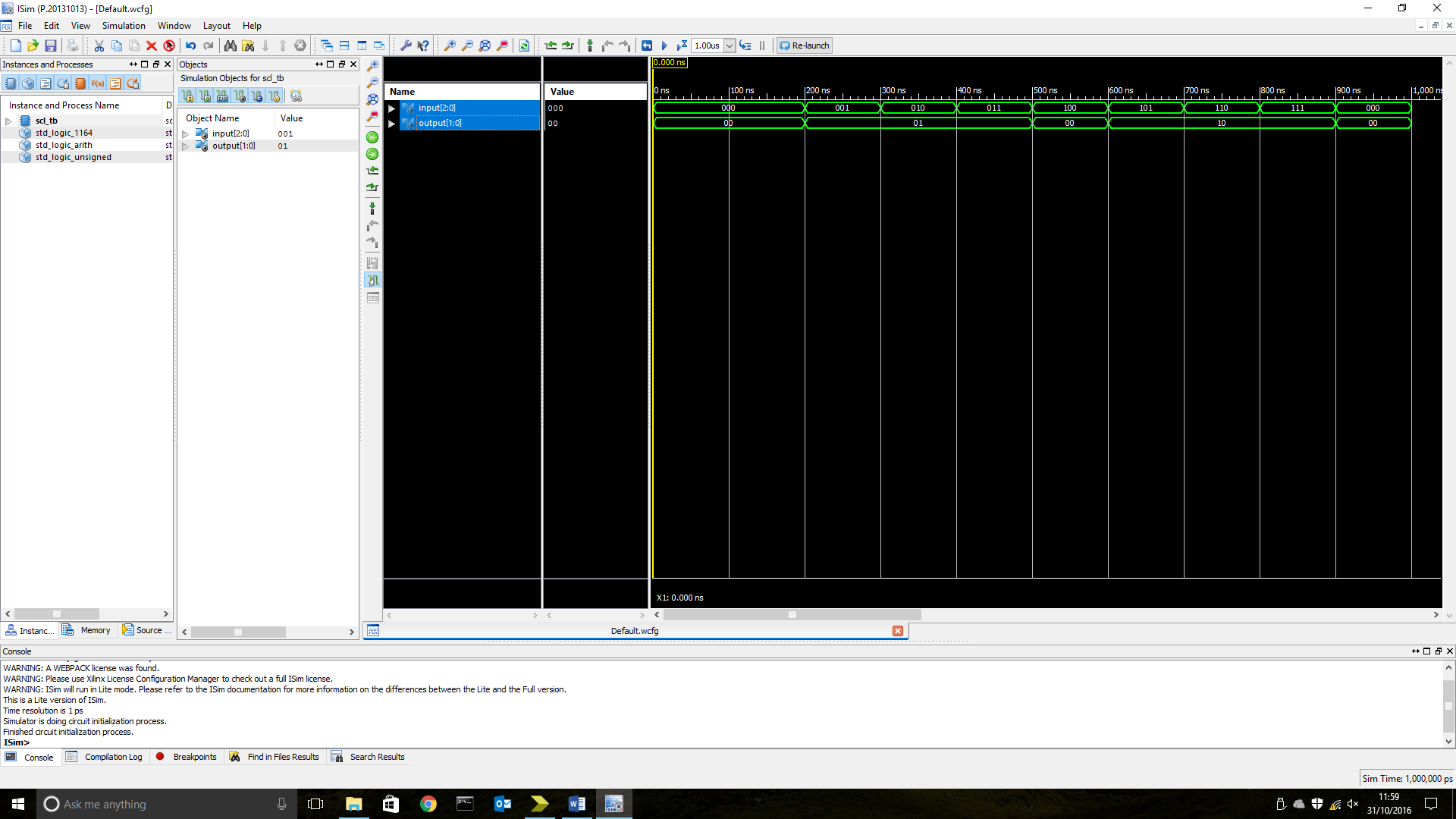
Truth table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | Ctrl | Output |
| 0 | X | X | X | 00 | 0 |
| 1 | X | X | X | 00 | 1 |
| X | 0 | X | X | 01 | 0 |
| X | 1 | X | X | 01 | 1 |
| X | X | 0 | X | 10 | 0 |
| X | X | 1 | X | 10 | 1 |
| X | X | X | 0 | 11 | 0 |
| X | X | X | 1 | 11 | 1 |

Figure 11: Simulation timing diagram of 4-input multiplexer showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**Shift control logic:**

Truth table:

|  |  |
| --- | --- |
| Input | Output |
| 000 | 00 |
| 001 | 01 |
| 010 | 01 |
| 011 | 01 |
| 100 | 00 |
| 101 | 10 |
| 110 | 10 |
| 111 | 10 |

Figure 12: Simulation timing diagram of Shift control logic showing its behaviour when tested with a range of possible inputs

Analysis: The output of the simulation was as expected for the devices behaviour.

**Testing on FPGA**

In the lab we uploaded the ALU and Shifter on to the FPGA board using the method described above. The shifter provided no errors and matched the expected behaviour perfectly however the ALU did not initially seem to be working completely correctly. The C\_out of the last two rows of the truth table were not matching up with the results we were observing, after checking the design and other possible sources of the error it was found that the provided truth table was actually in fact incorrect.

**Discussion**

All the devices designing in this lab according to our simulations and in lab testing are working as required. The ALU and some of the devises under it do have long delays due to the amount of gates that the signal has to pass through. If the ALU is to be used as a component in another device then these delays will have to be taken into consideration to make sure all relevant signals from parallel devices arrive at the same time. A look ahead is already being used in the ALU if this was not used then the delay produced would be substantially longer then they are currently as the carry would have to pass through each full adder in turn.

**Conclusion**

For this lab we implemented an ALU and Shifter which were both made up of lower level devices such as full adders and multiplexers. The design for the ALU had to be adapted slightly from the given schematic to give the requested output, but once rectified worked correctly. All simulations performed correctly and the two top level devices that were uploaded to the FPGA. When tested on the board they matched the expected outputs for all inputs given once the ALU truth table had been corrected. The devices coded in this lab could now be used in other structures to make more complicated systems.