Group 2- Patrick Balcombe, Harry Austin, Devashish Bishnoi

**ECS615U-** **Laboratory Session 3**

# **Abstract**

In this lab we implemented various devices that used n-bit registers made out of d-flipflops that were design as the first task. 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 top level devises using iPACT after which the devices actual inputs and outputs where then tested.

The devices higher up in the design such as the linear feedback shift register use the functionality of the devises lower down in the design mainly the D-flipflop and n-bit Register 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.

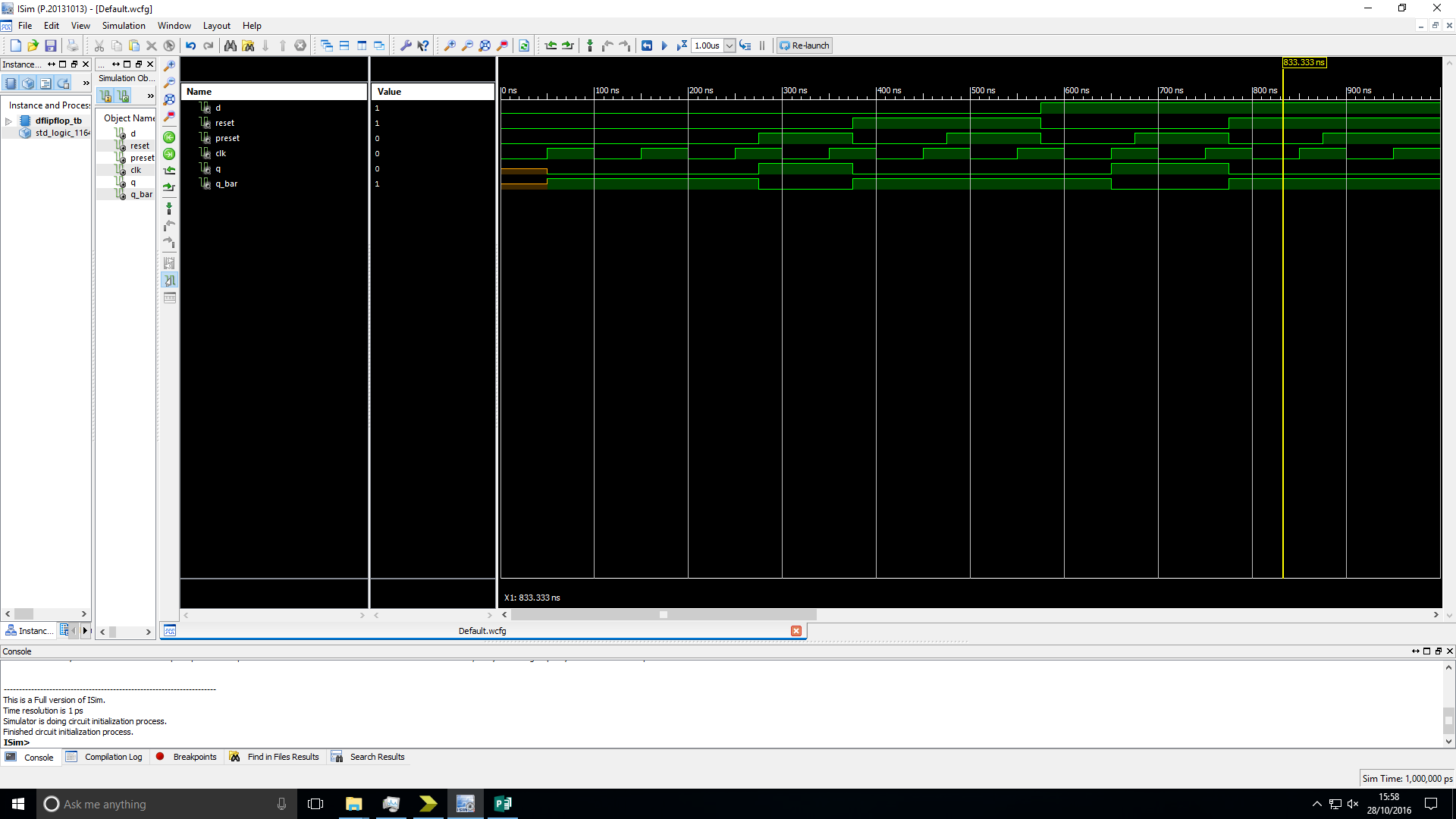
As in this lab clocks were now required for the devices a signal had to be supplied to the FPGA to act as the clock. We used the variable clock that is available on the FPGA board and connected it with wire connecting it from its pin to one of the board’s general IO pins.

# Simulation and FPGA Testing

In this section we individually run the test bench files for each device in Modelsim and on the physical FPGM board to check the timing diagram as well as actual results of the I/O matches its truth table. The Truth tables lists possible combinations of inputs and the relevant outputs which are compared to the results gained from the simulation and tests on the board.

# Timing Diagrams

# D-flipflop:

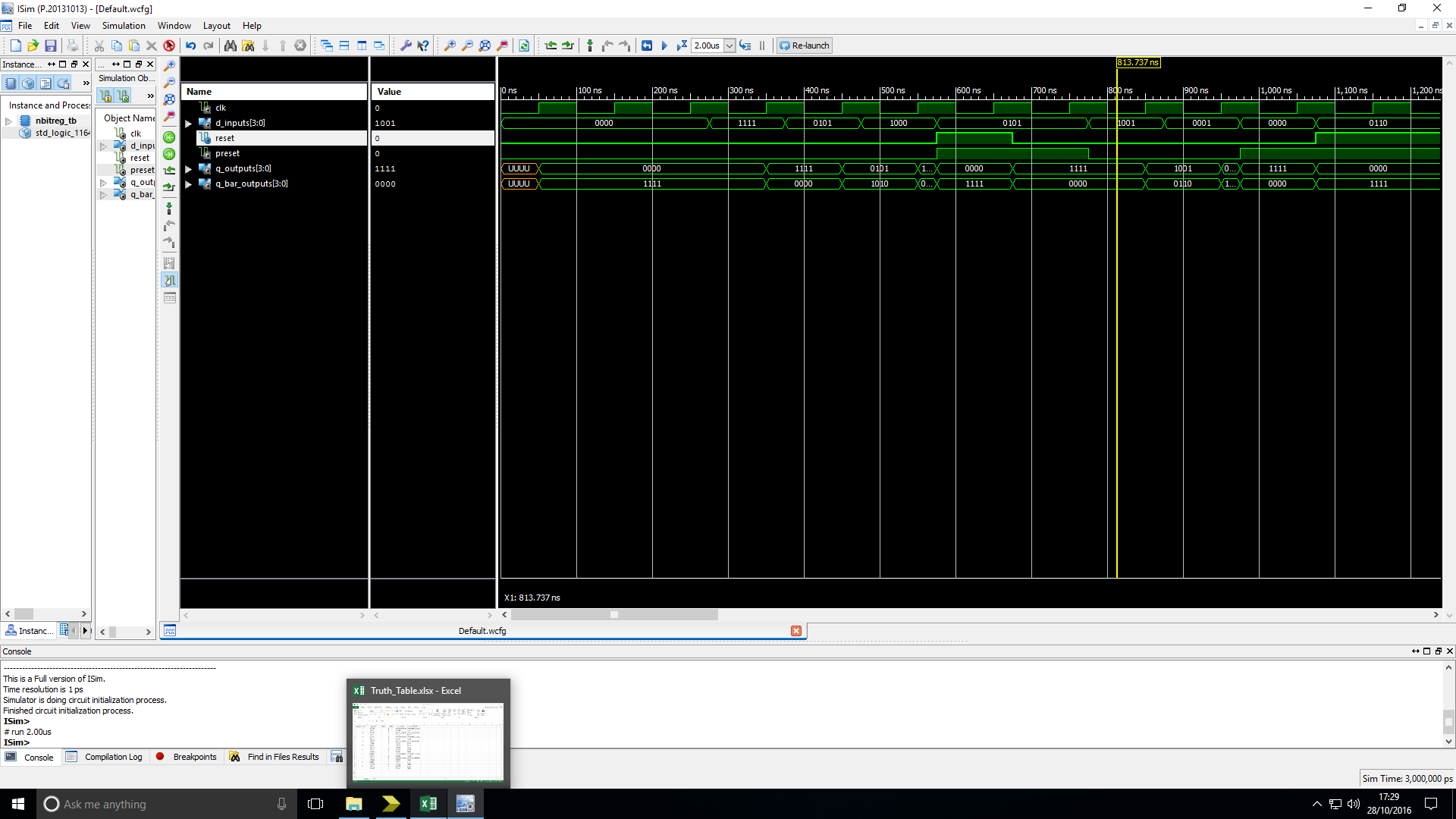
Truth table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CLK | D | Preset | Reset | Q | Q\_Bar |
| ↑ | 0 | 0 | 0 | 0 | 1 |
| ↑ | 1 | 0 | 0 | 1 | 0 |
| 0 | x | 0 | 0 | Last Q | last Q\_bar |
| 1 | x | 0 | 0 | Last Q | last Q\_bar |
| x | x | 1 | 0 | 1 | 0 |
| x | x | x | 1 | 0 | 1 |

Figure 1: Simulation timing diagram of a D-flipflop showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit register:

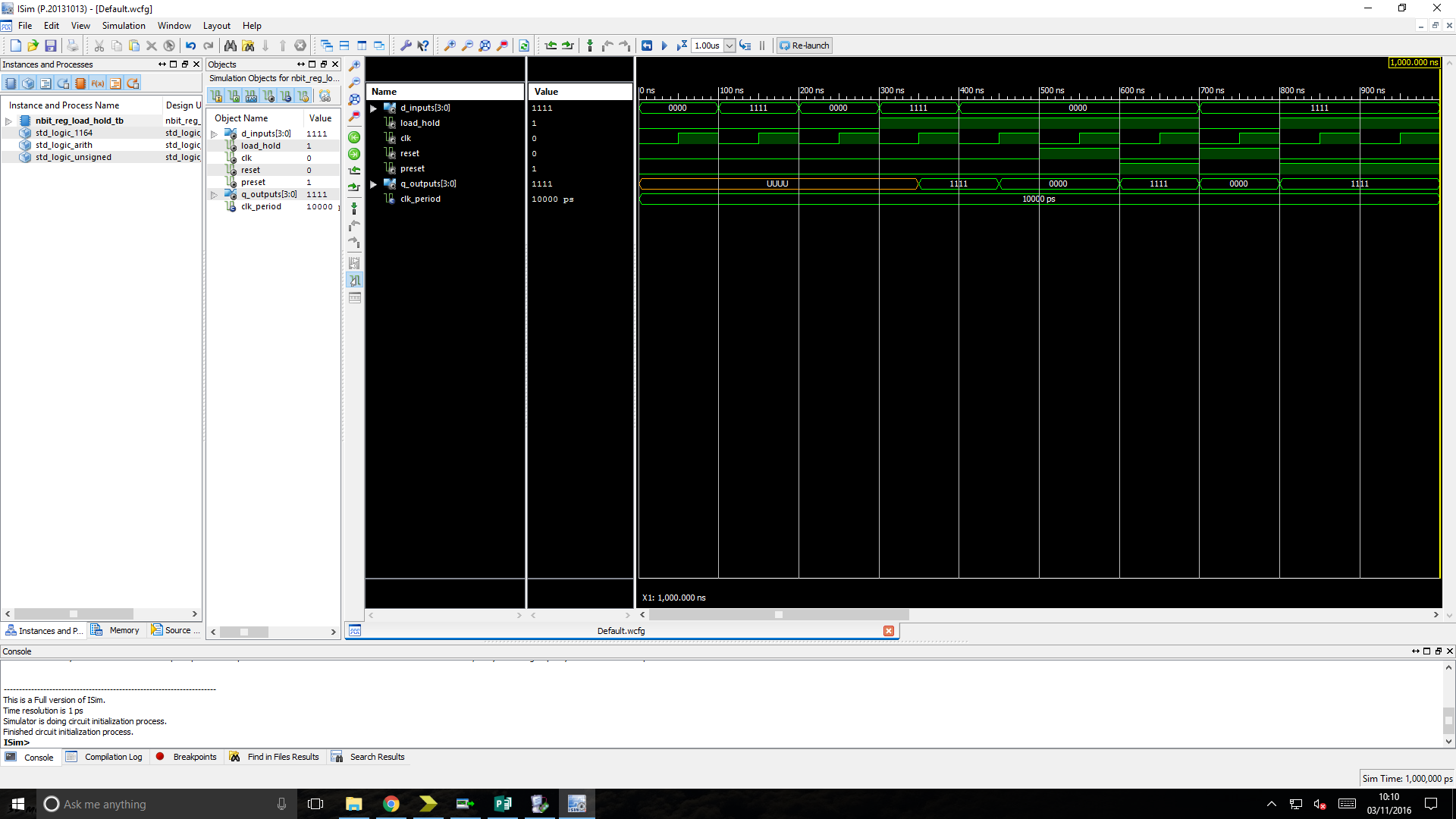
Truth table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CLK | D\_Input | reset | preset | Q\_outputs | Q\_bar\_outputs |
| 0 | 0000 | 0 | 0 | last Q\_outputs | last Q\_bar\_outputs |
| ↑ | 0000 | 0 | 0 | 0000 | 1111 |
| ↑ | 1111 | 0 | 0 | 1111 | 0000 |
| ↑ | 0101 | 0 | 0 | 0101 | 1010 |
| ↑ | 1000 | 0 | 0 | 1000 | 0111 |
| x | 0101 | 1 | 1 | 0000 | 1111 |
| x | 0101 | 0 | 1 | 1111 | 0000 |
| ↑ | 1001 | 0 | 0 | 1001 | 0110 |
| ↑ | 0001 | 0 | 0 | 0001 | 1110 |
| x | 0000 | 0 | 1 | 1111 | 0000 |
| x | 0110 | 1 | 0 | 1111 | 0000 |

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

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit register with LOAD/~HOLD control:

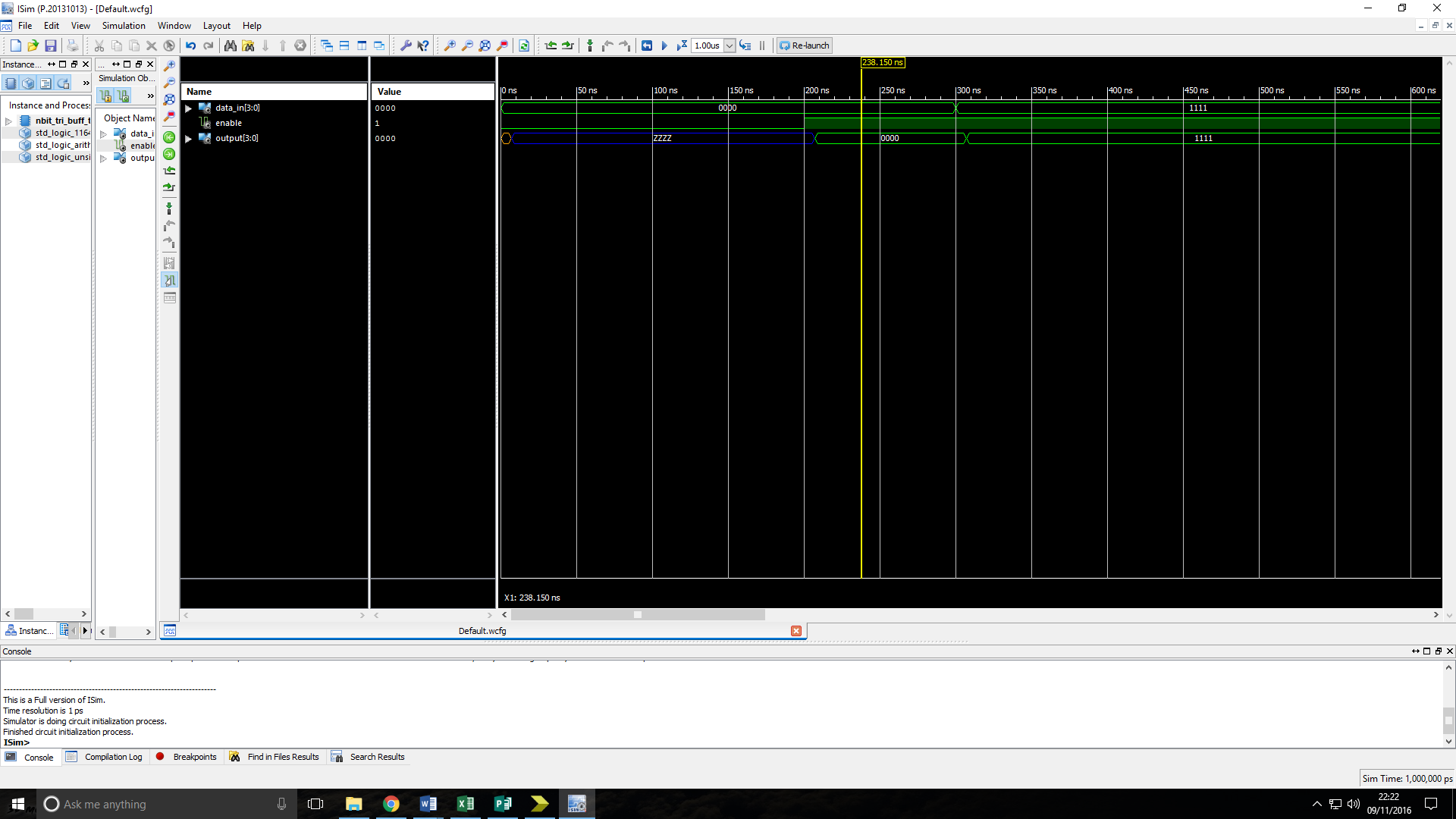
Truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| InA | D\_inputs | Load/hold | Q\_outputs |
| 0101 | 0000 | 00 | 1010 |
| 0101 | 0000 | 01 | 0000 |
| 0101 | 0000 | 10 | 0101 |
| 0101 | 0000 | 11 | 0101 |
| 0101 | 0000 | 00 | 1010 |
| 1010 | 0000 | 01 | 0000 |
| 0000 | 0000 | 10 | 0000 |
| 0000 | 0000 | 11 | 0000 |
| 1110 | 1011 | 00 | 0001 |
| 1110 | 1011 | 01 | 1010 |
| 1110 | 1011 | 10 | 0101 |
| 1110 | 1011 | 11 | 1111 |

Figure 3: Simulation timing diagram of an n-bit register with LOAD/~HOLD control showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit tri-state buffer:

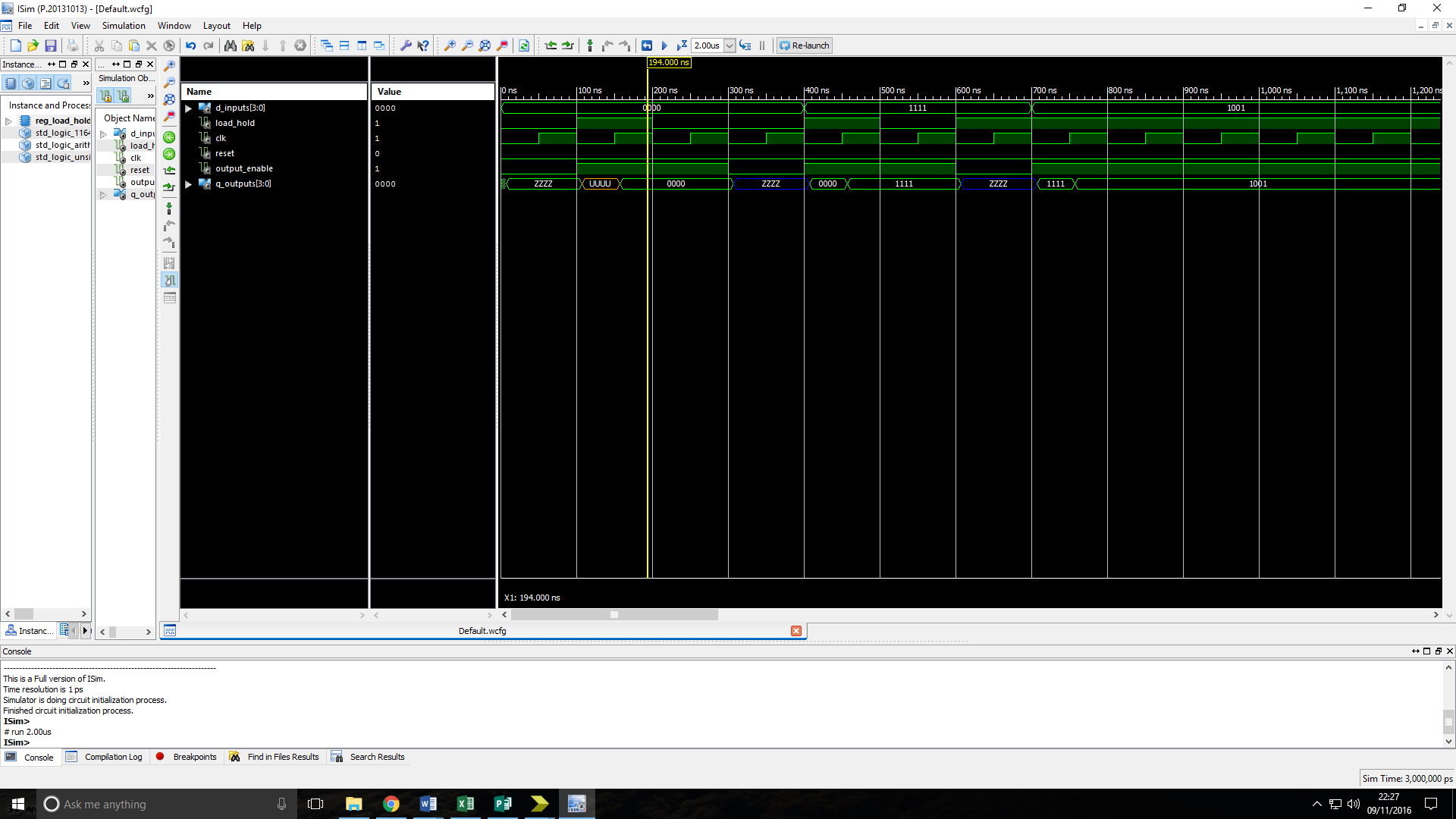
Truth table:

|  |  |  |
| --- | --- | --- |
| Data\_in | Enable | Outputs |
| xxxx | 0 | zzzz |
| 1111 | 1 | 1111 |
| 0000 | 1 | 0000 |
| 0101 | 1 | 0101 |
| 0110 | 1 | 0110 |

Figure 4: Simulation timing diagram of an n-bit tri-state buffer showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit register with LOAD/~HOLD and tristate output enable:

Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| InA | D\_inputs | Load/Hold | Enable | Q\_outputs |
| X | X | X | 0 | Z |
| 0101 | 0000 | 01 | 1 | 0000 |
| X | X | X | 0 | Z |
| 0101 | 0000 | 11 | 1 | 0101 |

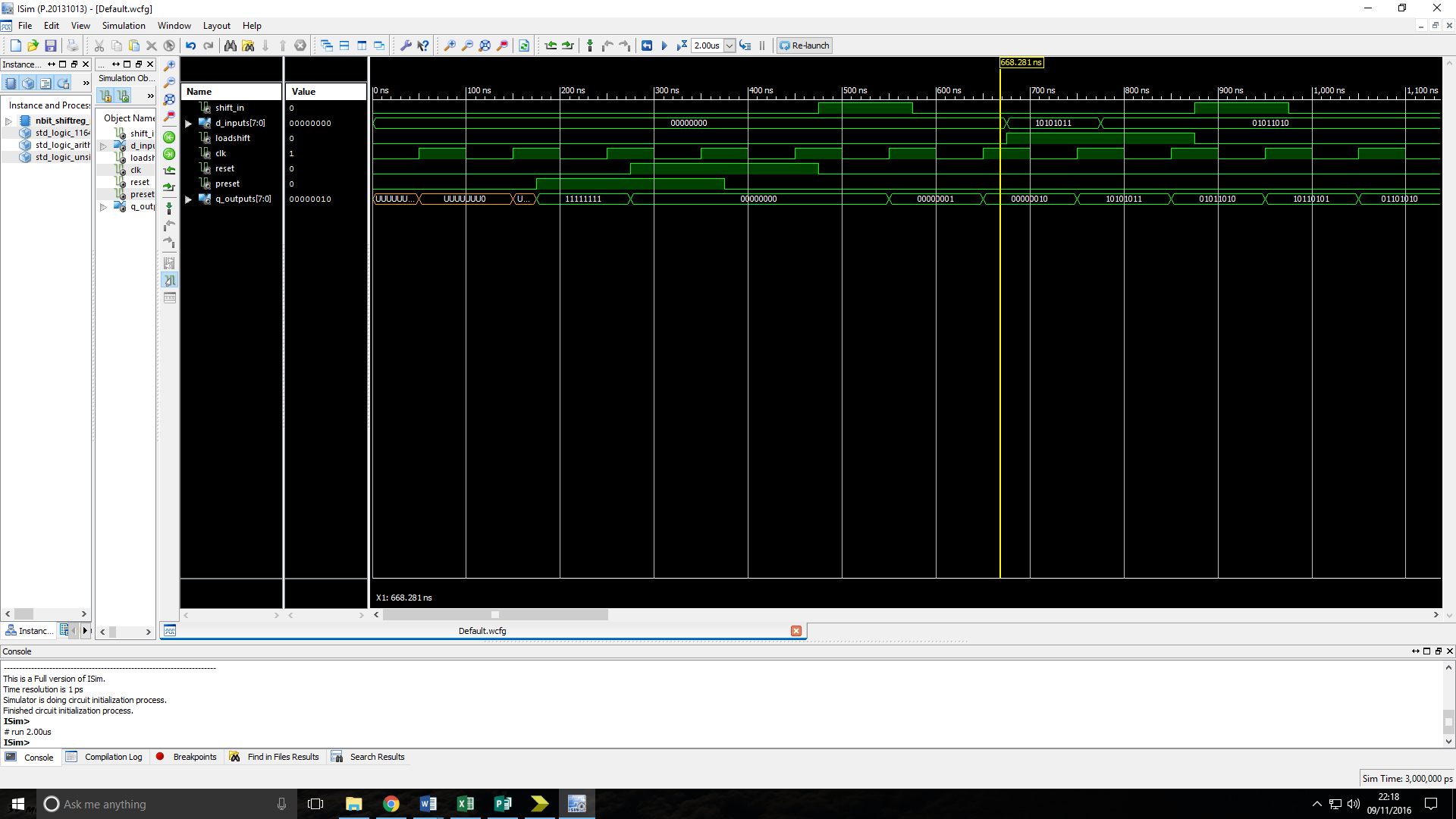
Figure 5: Simulation timing diagram of an n-bit register with LOAD/~HOLD and tristate output enable showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit shift register with parallel load input:

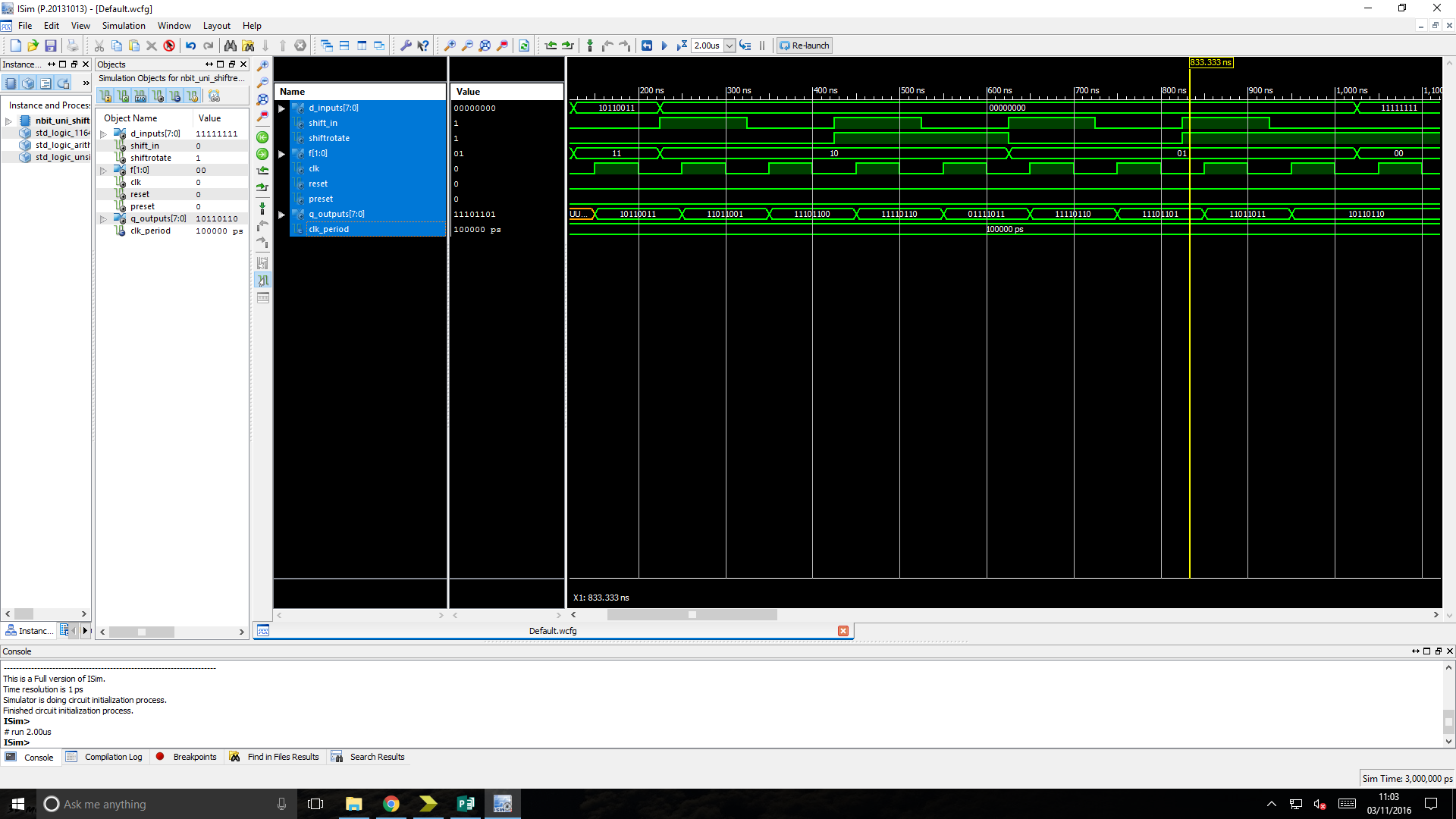
Truth table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Shift in | D\_inputs | load\_shift | CLk | Reset | Preset | Q\_outputs |
| x | x | x | x | 0 | 1 | 11111111 |
| x | x | x | x | 1 | 1 | 00000000 |
| x | x | x | x | 1 | 0 | 00000000 |
| 1 | x | 0 | ↑ | 0 | 0 | 00000001 |
| 0 | x | 0 | ↑ | 0 | 0 | 00000010 |
| x | 10101011 | 1 | ↑ | 0 | 0 | 10101011 |
| x | 01011010 | 1 | ↑ | 0 | 0 | 01011010 |
| 1 | x | 0 | ↑ | 0 | 0 | 10110101 |
| 0 | x | 0 | ↑ | 0 | 0 | 01101010 |

Figure 6: Simulation timing diagram of an n-bit shift register with parallel load input showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# n-bit Universal shift register:

Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F1 | F0 | Shift/~rotate | CLK | Operation |
| x | x | x | 0 | None |
| x | x | x | 1 | None |
| 0 | 0 | x | ↑ | Hold |
| 0 | 1 | 0 | ↑ | Rotate Left |
| 0 | 1 | 1 | ↑ | Shift Left |
| 1 | 0 | 0 | ↑ | Rotate Right |
| 1 | 0 | 1 | ↑ | Shift Right |
| 1 | 1 | x | ↑ | Load |

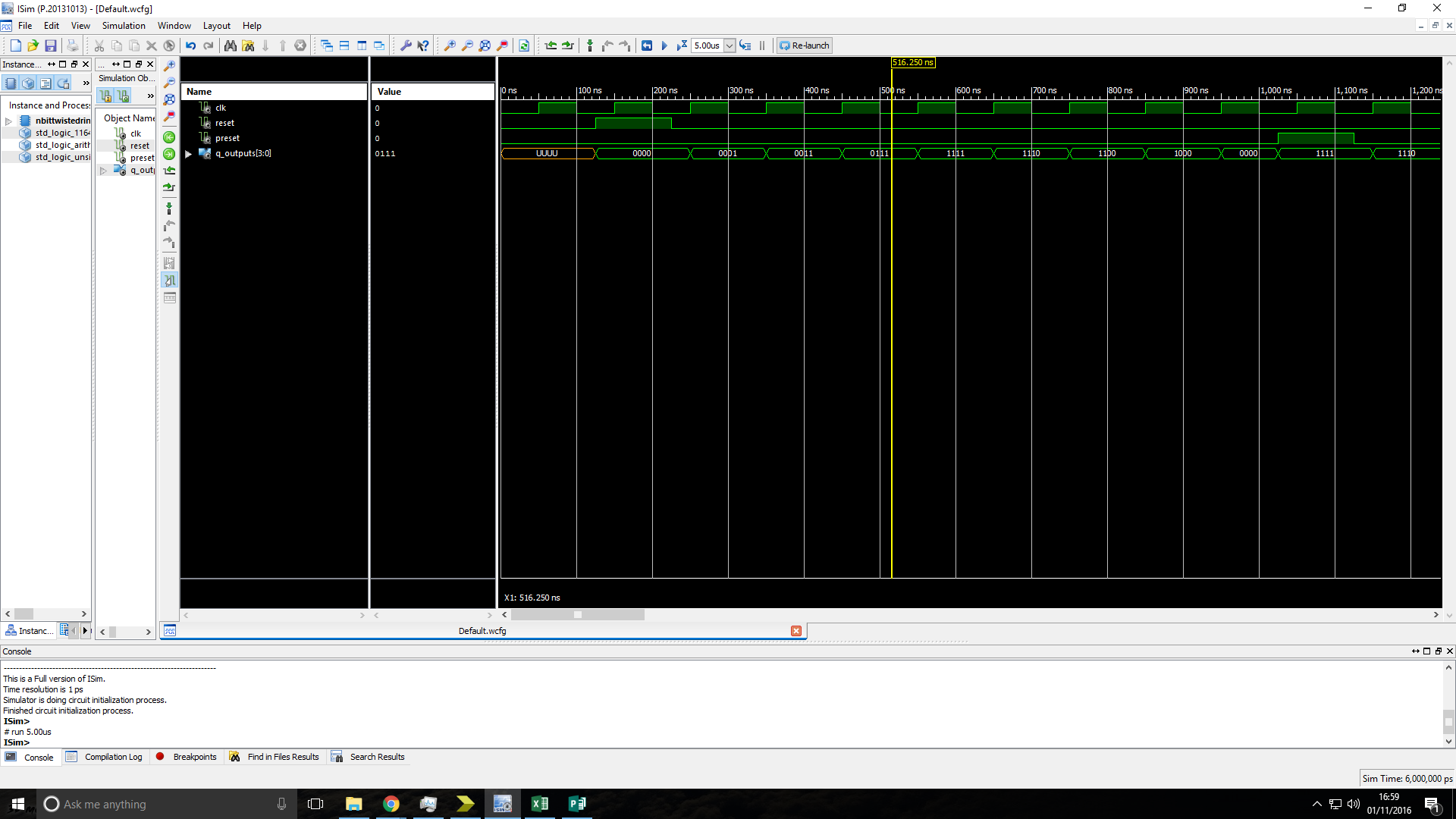
Figure 7: Simulation timing diagram of an n-bit Universal shift register showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board. Initially the test bench for this device did cause some problems as the inputs were tested in the order given in the truth table meaning that a value was only loaded in for the last stimulus and the output was undefined. The first stimulus therefore was changed to load operation so that the other operations could be carried out on the loaded value.

# n-bit twisted-ring counter:

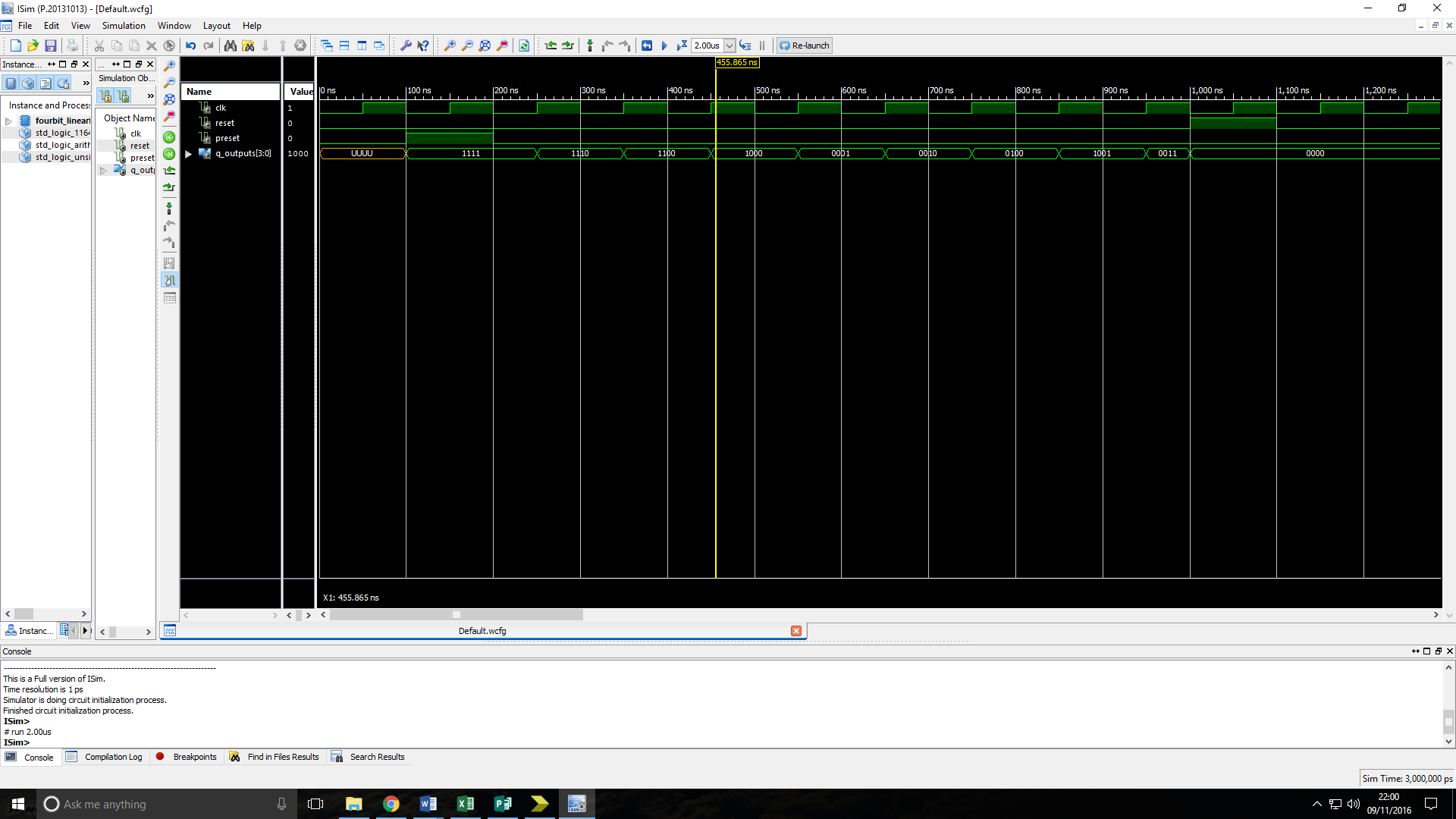
Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLK | reset | preset | Q\_outputs(n-1) | Q\_outputs(n) |
| ↑ | 0 | 0 | 0000 | 1000 |
| ↑ | 0 | 0 | 1000 | 1100 |
| ↑ | 0 | 0 | 1100 | 1110 |
| ↑ | 0 | 0 | 1110 | 1111 |
| ↑ | 0 | 0 | 1111 | 0111 |
| ↑ | 0 | 0 | 0111 | 0011 |
| ↑ | 0 | 0 | 0011 | 0001 |
| ↑ | 0 | 0 | 0001 | 0000 |
| x | 0 | 1 | XXXX | 0000 |
| x | 1 | 0 | XXXX | 1111 |

Figure 8: Simulation timing diagram of an n-bit Universal shift register showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

# 4-bit linear feedback shift register:

Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLk | Shift in (XOR of bits 2 & 3 of previous Q) | Reset | Preset | Q\_Outputs |
| x | x | 0 | 1 | 1111 |
| x | x | 1 | 1 | 0000 |
| x | x | 1 | 0 | 0000 |
| x | x | 0 | 1 | 1111 |
| ↑ | 0 | 0 | 0 | 1110 |
| ↑ | 0 | 0 | 0 | 1100 |
| ↑ | 0 | 0 | 0 | 1000 |
| ↑ | 1 | 0 | 0 | 0001 |
| ↑ | 0 | 0 | 0 | 0010 |
| ↑ | 1 | 0 | 0 | 0100 |
| ↑ | 1 | 0 | 0 | 1001 |

Figure 9: Simulation timing diagram of a 4-bit linear feedback shift register showing its behaviour when tested with a range of possible inputs.

Analysis: The output of the device is as expected for both the simulation and test on FPGM board.

**Discussion**

All devices that were tested matched their truth tables for all stimuluses’ input. In some of the simulation there is a delay in the outputs taking a value (meaning they are uninitialized at the start) for the first few nano seconds. This delay is not always due to gate delays as it usually is but in some cases is due to the fact that no value has been loaded into the flipflop yet. One case of this is the d-flipflop which only assigns a value after the first clock pulses and the LOAD/~HOLD register that has to have the correct control load\_hold input to load an input into the flip flops so the value is assigned. Having an undefined output could be a potential problem if used in a real system as an unexpected value to following circuitry could produce unwanted or unexpected results. In the tests on the FPGM board this lack of assigned value was just shown as zeroes before an input was applied.

# Conclusion

For this lab we implemented various devices that used n-bit registers made out of d-flipflops that were design as the first task. Through testing we then found that all the outputs from the devices where correct when compared with the expected outputs during simulation and on the FPGA boards. Some outputs were undefined for some time before they were assigned a value at the first clock pulse or correct control input which at this stage did not cause any problems apart from a slight initial error on a implementation of a test bench.