Experiment 2 - Frequency Regulation

Ali Bahari,
810196688

Abstract— This document gives the final results of the second experiment of digital logic design laboratory.

I. Introduction

The goal of this experiment is to design an adjustable clock generator that can fix the output frequency at the desired value meaning that we will construct a component which by changing the initial values of the frequency divider that was created in the last experiment gradually, can generate the desirable frequency at the end. Finally in the end it's tested and shown that using FPGA and setting the correct parallel loads for frequency divider will result in desired frequency.

II. Design Description and Synthesis

In this part we will make and code the component that was mentioned above.

1. The code is provided in the sent files under the folder named 1. In the code everything is done by the help of instructions in the experiment.
2. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus.
3. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus.
4. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus. Also 74LS191 is changed in this experiment and is replaced by 74LS193 as mentioned in the experiment's description.
5. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus.
6. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus. There is input by the name FPGA_clk for this purpose and it'll receive 50 MHz clock signal in the test bench.
7. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus.
8. Done in Quartus. The files for Quartus is provided in sent files under the folder named 1 Quartus.

III. Design Simulation in Modelsim

In this part we are going to test the design so to understand whether it is working correctly or not. Also the circuit is brought into Modelsim from Quartus.

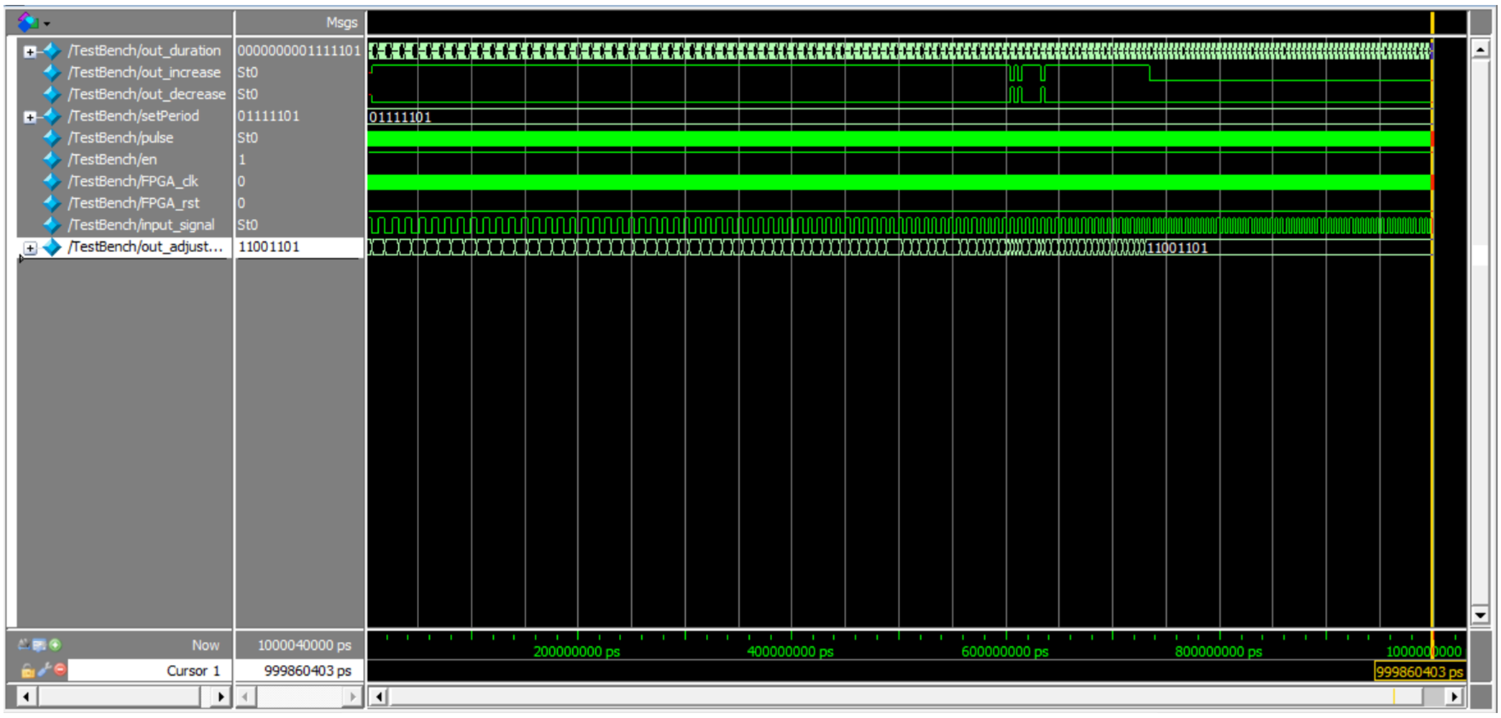
1. Test bench is provided in sent files under the folder named 1 ModelSim and there are descriptions in the code commented as well for any needed clarification. In the test bench ring oscillator code is provided by TA and it is used for frequency divider clock. Also in the test bench there is a 50 MHz clock signal for FPGA, since the frequency is 50 MHz so clock cycle length will be 20 ns. Other components are also placed correctly for testing such as FPGA, ring oscillator, FPGA_clk and so on. Also there is an AND gate for frequency divider load so that we can set load value for divider like the last experiment.
2. The results are shown below in Table 1.

TABLE I
ACHIEVED RESULTS FOR DIFFERENT RING OSCILLATOR FREQUENCIES

Ring Oscillator Frequency	Desired Frequency	Final Parallel Loads	Initial Parallel Loads	SetPeriod
20000 KHz (clock = 50 ns)	400 KHz	205	127	125
20080 KHz (clock = 49.8 ns)	400 KHz	205	127	125
19841 KHz (clock = 50.4 ns)	400 KHz	205	127	125
20550 KHz (clock = 48.66 ns)	400 KHz	205	127	125
20375 KHz (clock = 49.08 ns)	400 KHz	205	127	125
19960 KHz (clock = 50.1 ns)	400 KHz	205	127	125
19912 KHz (clock = 50.22 ns)	400 KHz	205	127	125

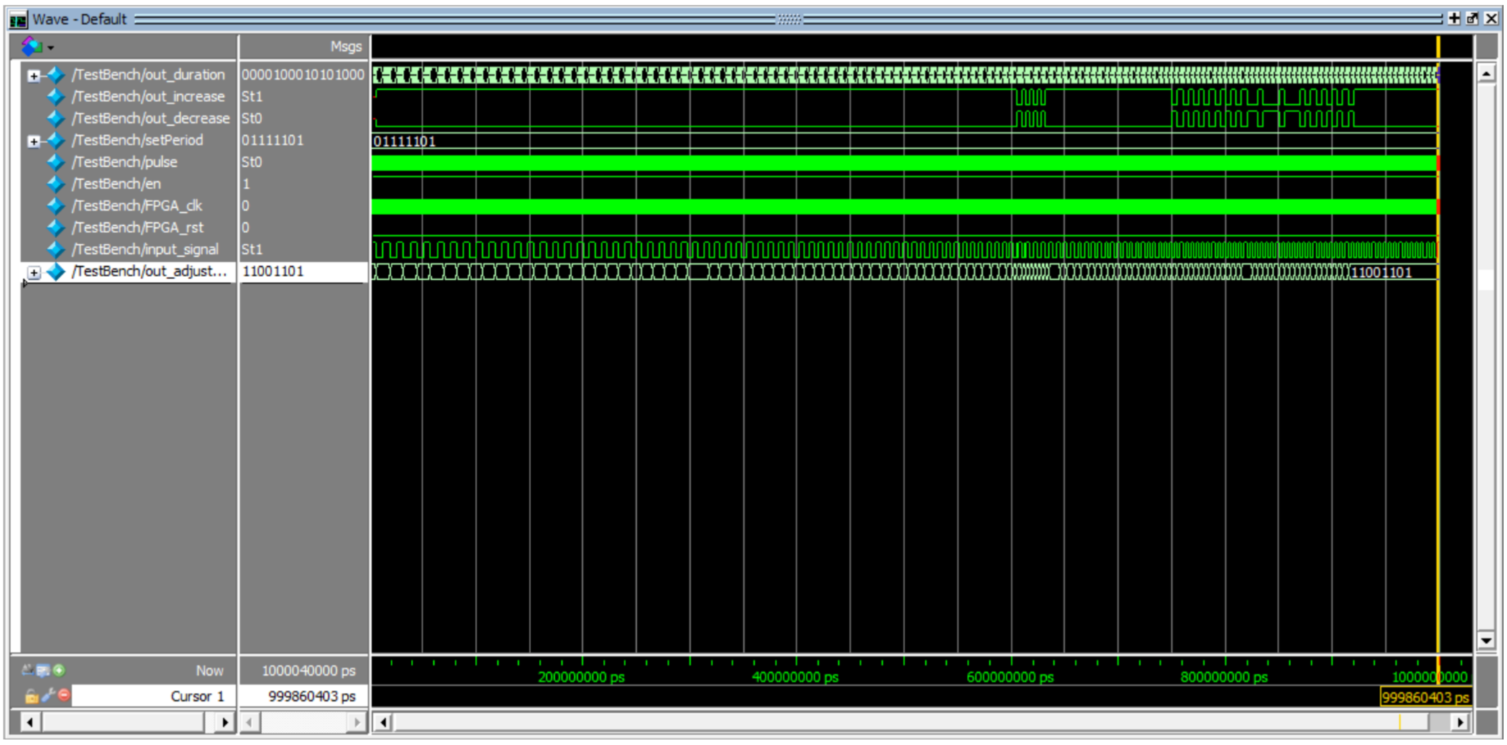
Since the experiment description states that the desired frequency must be 400 KHz and also FPGA frequency must be 50 MHz at all times, setPeriod will always be 125. Also in FPGA code the initial parallel loads always start from 127 and must reach 205 because setPeriod is always the same. Ring oscillator frequency is changed by a small amount for multiple tests as shown above, frequency is shown clearly and also clock parameter in Table 1 shows the clock cycle length of ring oscillator, this parameter is changed in ring oscillator code by changing INV_DELAY_ns which is delay of each inverter and since there are three inverters this parameter will equal clock values in Table 1 divided by 6.

3. All the waveforms are provided below in figures 1 to 6. In some of the waveforms since the exact frequency is not reachable, it'll fluctuate between increasing and decreasing but the final loads will always be 205 and it'll be centred on it. That is the reason for some of the figures fluctuating between increase and decrease when FPGA reaches the end result. But in some cases since there is a part for being equal in the code the end result will exactly be 205 and there won't be any fluctuation.



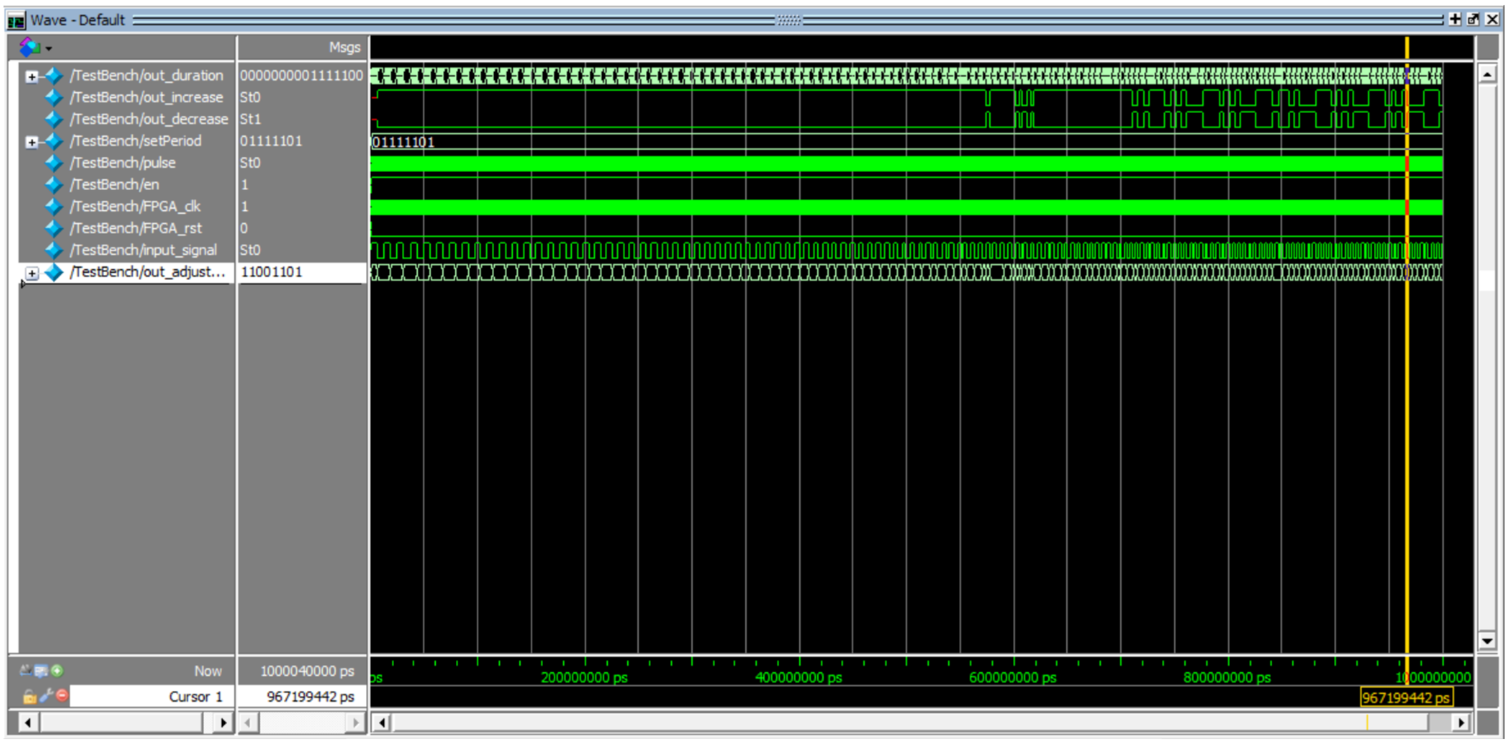
```
                                Msgs
/TestBench/out_duration      0000000001111101
/TestBench/out_increase      St0
/TestBench/out_decrease      St0
/TestBench/setPeriod         01111101
/TestBench/pulse             St0
/TestBench/en                1
/TestBench/FPGA_clk          0
/TestBench/FPGA_rst          0
/TestBench/input_signal      St0
/TestBench/out_adjust...     11001101

Now       1000040000 ps
Cursor 1  999860403 ps
```

Fig. 1 All the wanted waveforms for ring oscillator with frequency 20080 KHz (49.8 ns)



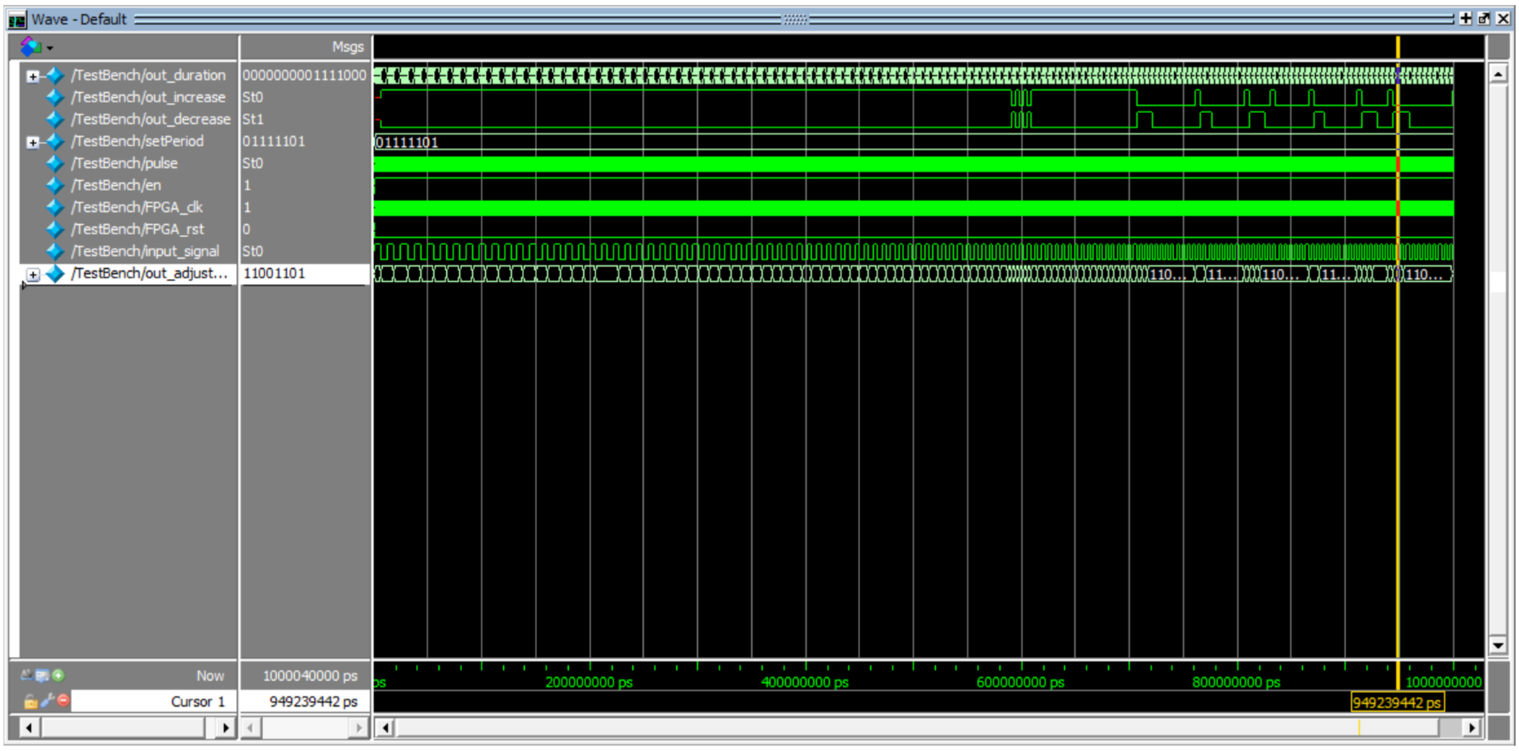
```
Wave - Default
                                Msgs
/TestBench/out_duration      0000100010101000
/TestBench/out_increase      St1
/TestBench/out_decrease      St0
/TestBench/setPeriod         01111101
/TestBench/pulse             St0
/TestBench/en                1
/TestBench/FPGA_clk          0
/TestBench/FPGA_rst          0
/TestBench/input_signal      St1
/TestBench/out_adjust...     11001101

Now       1000040000 ps
Cursor 1  999860403 ps
```

Fig. 2 All the wanted waveforms for ring oscillator with frequency 19841 KHz (50.4 ns)



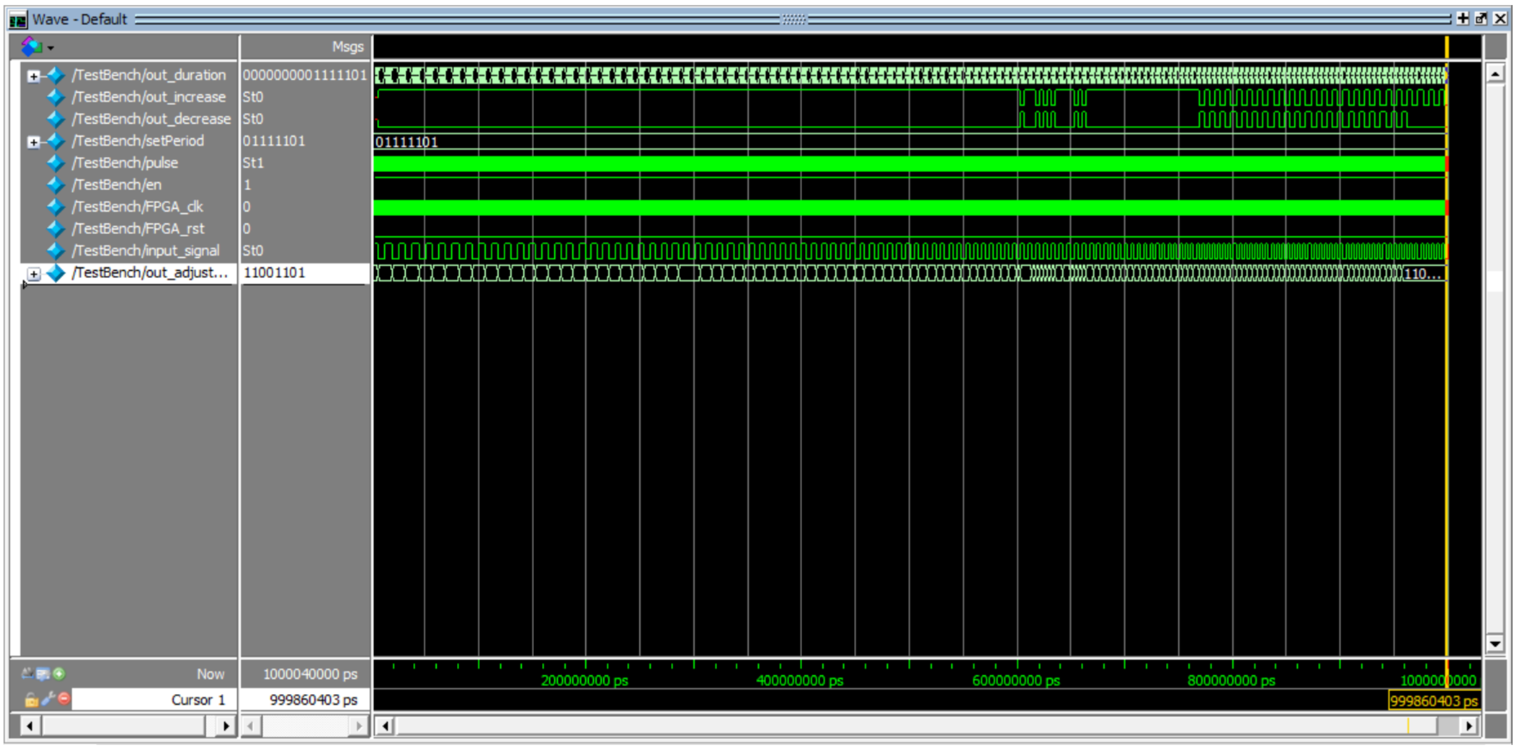
	Msgs
/TestBench/out_duration	0000000001111100
/TestBench/out_increase	St0
/TestBench/out_decrease	St1
/TestBench/setPeriod	01111101
/TestBench/pulse	St0
/TestBench/en	1
/TestBench/FPGA_clk	1
/TestBench/FPGA_rst	0
/TestBench/input_signal	St0
/TestBench/out_adjust...	11001101
Now	1000040000 ps
Cursor 1	967199442 ps

Fig. 3 All the wanted waveforms for ring oscillator with frequency 20550 KHz (48.66 ns)



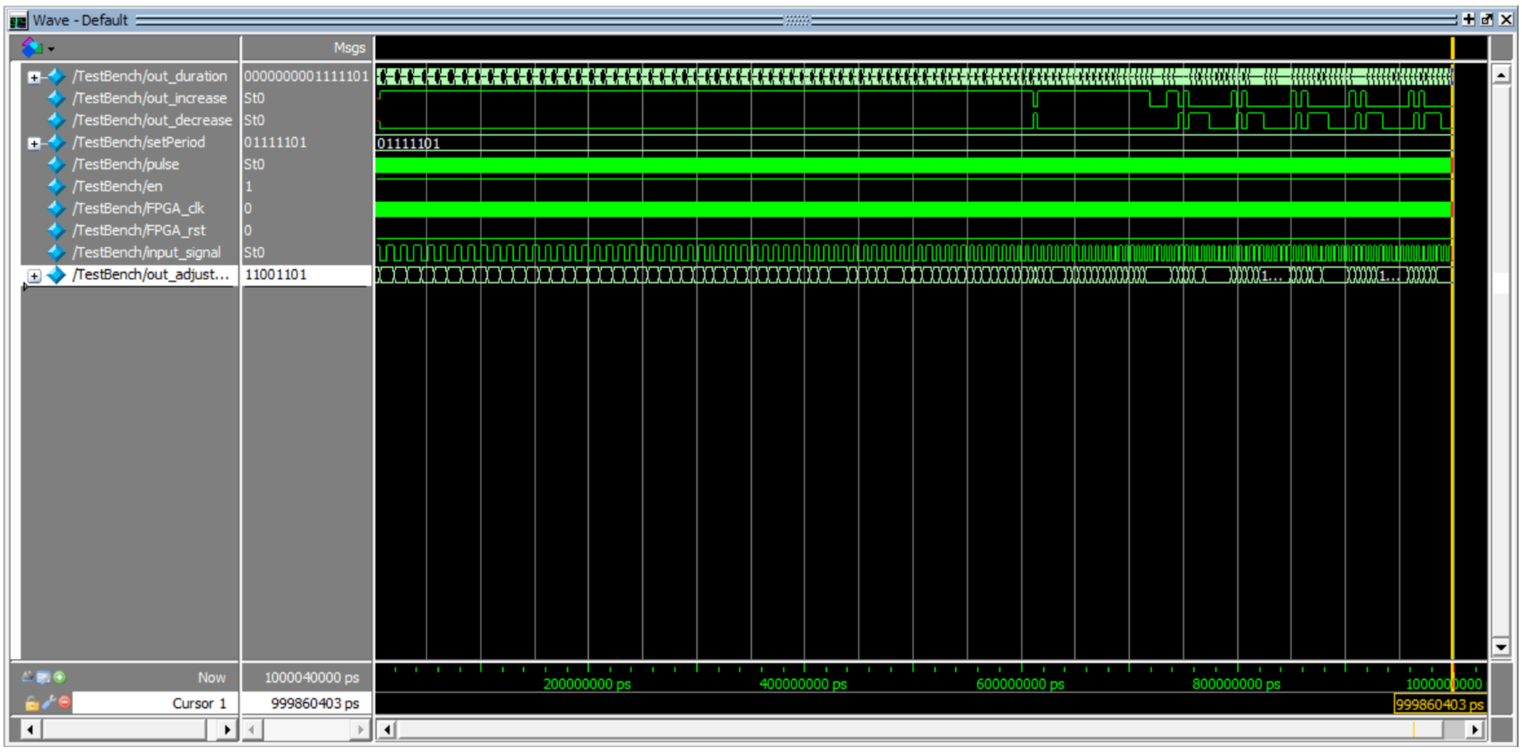
	Msgs
/TestBench/out_duration	0000000001111000
/TestBench/out_increase	St0
/TestBench/out_decrease	St1
/TestBench/setPeriod	01111101
/TestBench/pulse	St0
/TestBench/en	1
/TestBench/FPGA_clk	1
/TestBench/FPGA_rst	0
/TestBench/input_signal	St0
/TestBench/out_adjust...	11001101
Now	1000040000 ps
Cursor 1	949239442 ps

Fig. 4 All the wanted waveforms for ring oscillator with frequency 20375 KHz (49.08 ns)



```
Wave - Default
                                   Msgs
/TestBench/out_duration    0000000001111101
/TestBench/out_increase    St0
/TestBench/out_decrease    St0
/TestBench/setPeriod       01111101
/TestBench/pulse           St1
/TestBench/en              1
/TestBench/FPGA_clk        0
/TestBench/FPGA_rst        0
/TestBench/input_signal    St0
/TestBench/out_adjust...   11001101

Now        1000040000 ps
Cursor 1   999860403 ps
```

Fig. 5 All the wanted waveforms for ring oscillator with frequency 19960 KHz (50.1 ns)



```
Wave - Default
                                   Msgs
/TestBench/out_duration    0000000001111101
/TestBench/out_increase    St0
/TestBench/out_decrease    St0
/TestBench/setPeriod       01111101
/TestBench/pulse           St0
/TestBench/en              1
/TestBench/FPGA_clk        0
/TestBench/FPGA_rst        0
/TestBench/input_signal    St0
/TestBench/out_adjust...   11001101

Now        1000040000 ps
Cursor 1   999860403 ps
```

Fig. 6 All the wanted waveforms for ring oscillator with frequency 19912 KHz (50.22 ns)