Experiment #1

Group 3

Nazanin Sabri

810194346

*nazanin.sabrii@gmail.com*

Nima Jarrahiyan

810194292

*jarrahian.nima76@gmail.com*

**Abstract**—working with FPGAs, ring oscillators, creating a steady rectangular (square) pulse.

Keywords— clock divider, oscillator, frequency divider, FPGA, voltage converter

1. Introduction

The goal of this experiment is to introduce the concepts of static characteristics of digital logic gates, delay times, clock frequency generation and digital system using schematic diagram and Verilog HDL.

1. Part 1

This part was omitted from the experiment due to the absence of signal generators.

1. Part 2

In this part, we understood different methods of clock generation in digital systems.

1. Ring Oscillator

IC used in this part is 74LS04.

Our duty cycle is about 50%, the number of inverters used in our circuit needs to be an odd number. Different ICs of same family produce different frequencies.

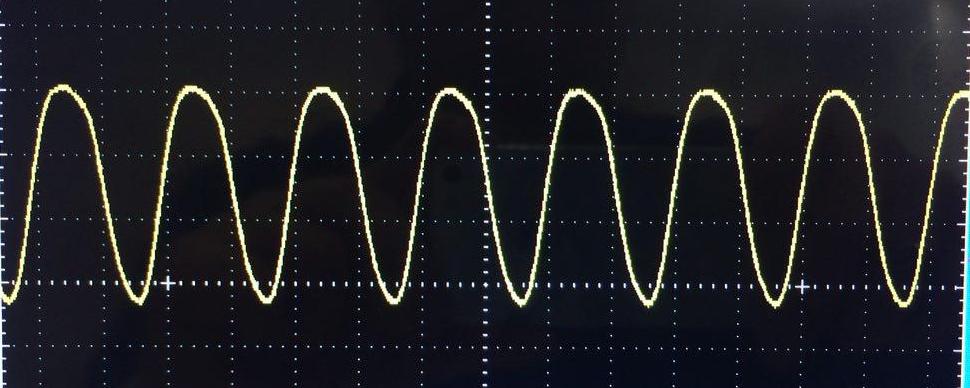


Fig. Example of a ring oscillator output signal

In order to measure the delay of each inverter in the ring oscillator, we divide the frequency of the output signal by 3.

Delay Time: 50ns / 3 = 16.6 ns .

Delay Time of 7404TTL : 8ns for HV as input and 12ns for LV input.

Comparing with 74LS04, since 74LS04 is dependent on an inside capacitance (15ns MAX) and looking at our delay time we can say that 7404TTL is faster.

1. LM555 Timer

This IC generates a square (rectangular) signal with different duty cycles. The duty cycle is computed using the following formula:

As the resistance of R2 increases, the duty cycle of the output pulse increases toward the limit of 50%.

The required frequencies are:

For R1 = 1KOhm and

R2 :

1 kΩ, f = 45 kHz and Duty Cycle of 0.66 .

10 kΩ, f = 6.9 kHz and Duty Cycle of 0.52.

100 kΩ, f =735 Hz and Duty Cycle of 0.502.

1. Schmitt Inverter Oscillator

In this inverter, the Duty Cycle decreases as resistance increases unlike the previous one .

The formula for the duty cycle and the frequency are as follows:

Where α is a constant number and unknown at this point.

To find the value of α, we change and observe different frequencies and calculate the number by using of value of capacitor and resistor used.

For C=10nF and R:

470 Ω, f =147 kHz

1 kΩ , f= 75kHz

2 kΩ, f= 31kHz

Using different frequencies, α = 72 using a linear equation between three different α so it may not be as precise as the true number.

Used IC is 74ls14.

Due to absence of 470 Ω resistor we used two paralleled 1 kΩ resistors.

1. Synchronous counter as a Frequency divider

Another method to build a clock with specific low frequency, is to divide a higher frequency clock with lower timing error.

Since different ICs build different frequencies from same family, ours was not able to build 20 kHz and it was 29kHz signal.

It is important to mention that pin 12 and 13 create opposite signals at the end so we must be careful when installing the generated signal on FPGA.

Preparing the ICs is important. We must note that for creating a signal divided by 113, first we need to know how the ICs work:

First, 74LS191 IC creates a carry out whenever the clocks cross 255. To measure a signal divided by a number, we should be starting from a point to measure the dividend which would be 255 – dividend.

Second, since this is a 8-bit divider, we should be starting from least most significant 4-bits and propagate the carry to the second IC. The carry of the second IC is our exit signal.

This introduction is for up count and for down count we do vise versa.

Entry pins are as A would be least significant bit and D most significant bit.

Calculation:

29M/113 = 256 kHz = 0.256 MHz

Result:

T= 2\*2 µs = 4 \* 10e-06

F=0.25\*10e+06 = 250 kHz

As shown the results are equal to calculations.

1. Part 3

The goal of this part is to measure the relation between the clock frequency and the frequency of FPGA.

In order to work with the FPGA, we need to make sure that the input signal is a steady square signal of 3.3 volts.

To build such a signal we went through the following steps:

1. Oscillator: we used the oscillator made in part 2 (A), to generate a clock frequency.
2. Counter: Using the counter of part 2 (D), we divided the oscillator frequency. More on how that was done is available in the mentioned section of the report. (Part 2 – D)
3. 74ls74: to build a steady square signal
4. Voltage Converter: to convert the 5v signal to a 3.3v one.

We then passed on the output of the counter to the 74ls74 as a clock generator. The output of 74ls74 was then converted to a 3.3v signal using the voltage converter and then passed to the FPGA as an entry signal.

We faced some difficulties in converting the output voltage. To solve the problem we used FPGA as a source for the 3.3v.

Following the steps of the appendix available in the end of the lab 1 we created a project in Quartus, using the VHDL code and the data sheet of out FPGA we assigned the pins and then compiled and ran the program.it is important to mention that Oscilloscope does have a load effect so to have a precise answer we have to disconnect all probes and their grounds.

The 7 segments displayed how many local (FPGA) clock cycles our signal was. The calculation that proves the display was correct is shown bellow:

Delay time of the oscillator = 1.7 \* 20ns

Frequency of the oscillator = 29 MHz

This frequency was supposed to be divided of 113 making the divided result 0.26 MHz. The delay time of the output of the counter was 2 \* 2 microseconds. Making the frequency of the counter 0.25 MHz.

The inside clock frequency of the FPGA is 50 Hz. Making our counter output frequency 200 of the FPGA clock frequency.

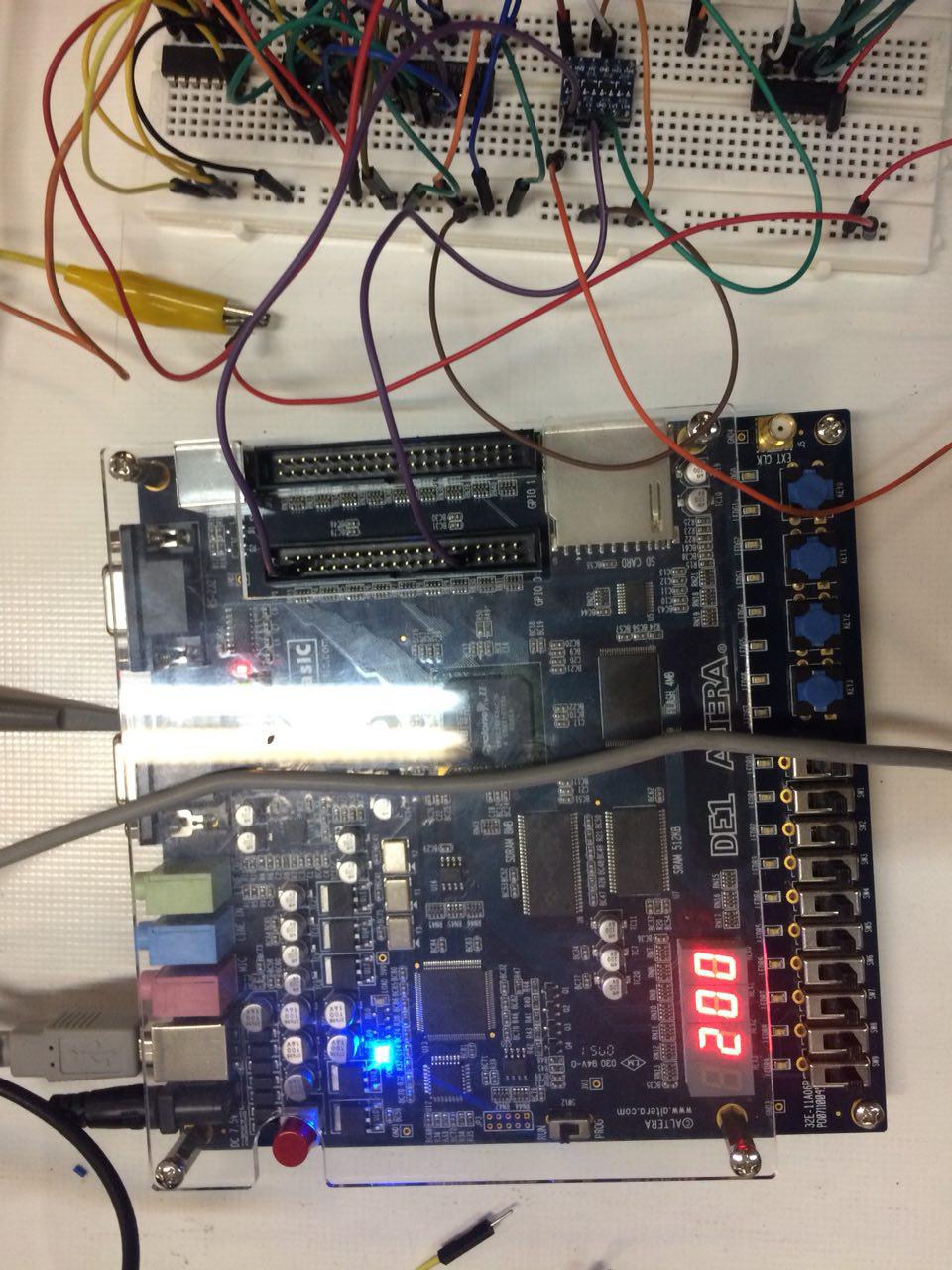


Fig. Display of FPGA seven-segment