Experiment #1

Group 3

Nazanin Sabri

810194346

*nazanin.sabrii@gmail.com*

Nima Jarrahiyan

810194292

*jarrahian.nima76@gmail.com*

Abstract—***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. We got familiar with ring oscillators and oscillation, generating clock signals using LM555 timer and Schmitt inverter. We also implemented a counter and practiced dividing frequencies using the counter.***

***We also got to know the requirements of the FPGA input signal and created such a signal using voltage level convertors and an IC to create square signals. Last but not least we got to use a FPGA and how to assign it’s pins and program one.***

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

* Methodology

We started the experiment from part 2 as instructed by the TA due to the absence of signal generators, therefore we will also start the methodology from part 2.

A ring oscillator as shown in figure 1 is made up of an odd number of inverters. The output of the last inverter is connected to the input of the first one. The time at which a value feeds back to the same node is the time period of the ring oscillator which equals 2N\*Delay(inv).

In this formula N is the number of invertors and Delay(inv) is the delay each single invertor.

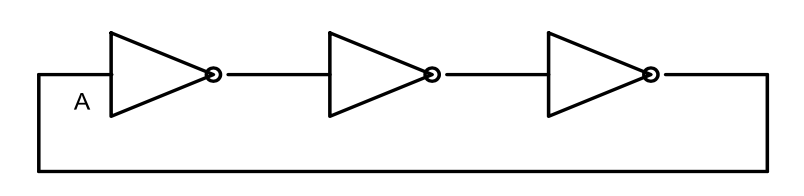


Fig. 1 Ring Oscillator

A ring oscillator can be used to measure the delay of logic gates as measuring them directly could introduce errors and large bandwidth.

In order to implement a ring oscillator we used a 74ls04 IC. Wiring it as shown in the figure bellow.

We made sure to assign one bar on the breadboard to the input signals and one bar to the output signals in order to avoid mistakes and confusions.

The output delay shown on the oscilloscope was that of the entire ring oscillators meaning all 3 inverters in the circuit that we had wired together.

We then also downloaded the datasheet of the used IC (74ls04) and used it to compare the delay we got to that available in the datasheet. The data sheet is available in this link: [*http://www.futurlec.com/74LS/74LS04.shtml*](http://www.futurlec.com/74LS/74LS04.shtml)

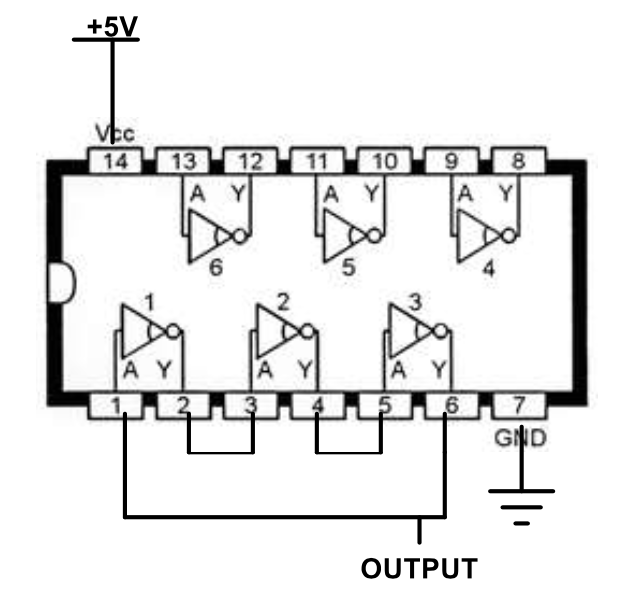


Fig. 2 Implementing a Ring Oscillator using 74ls04 IC

We then moved on to the second experiment of part 2, in which we were introduced to LM555 timer.

LM555 is among devices for generating clock signal or time delay. The pin layout of this IC is as shown in figure 3.

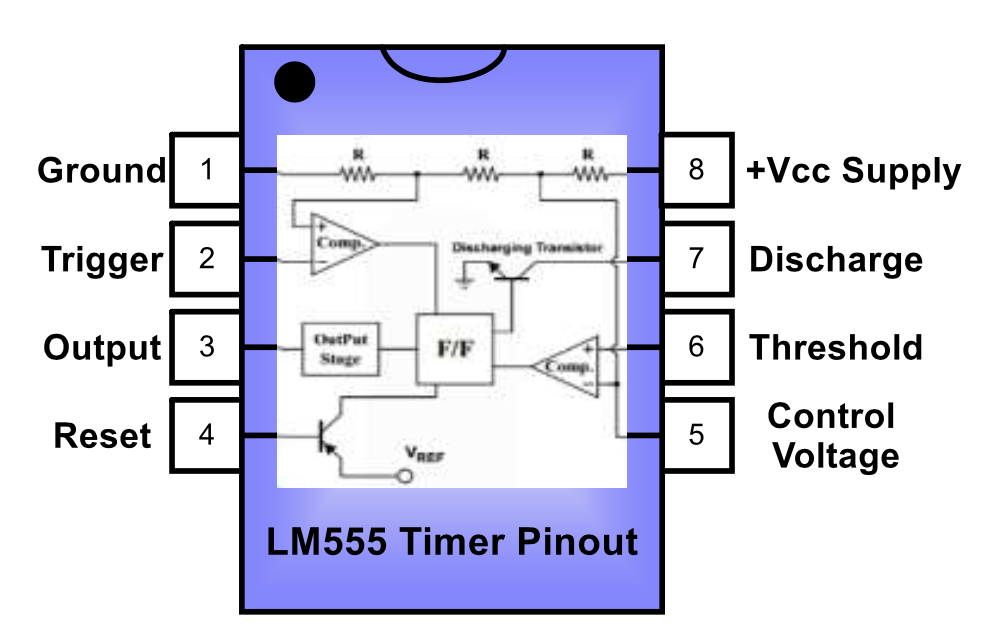


Fig. 3 Pin assignment in LM555

The experiment layout them continues to inform us of how LM555 works in 3 modes, we however focused on the astable mode and created our circuit using the wiring shown in figure 4.

The time during which the output is low or high is determined by the external elements connected to the IC.

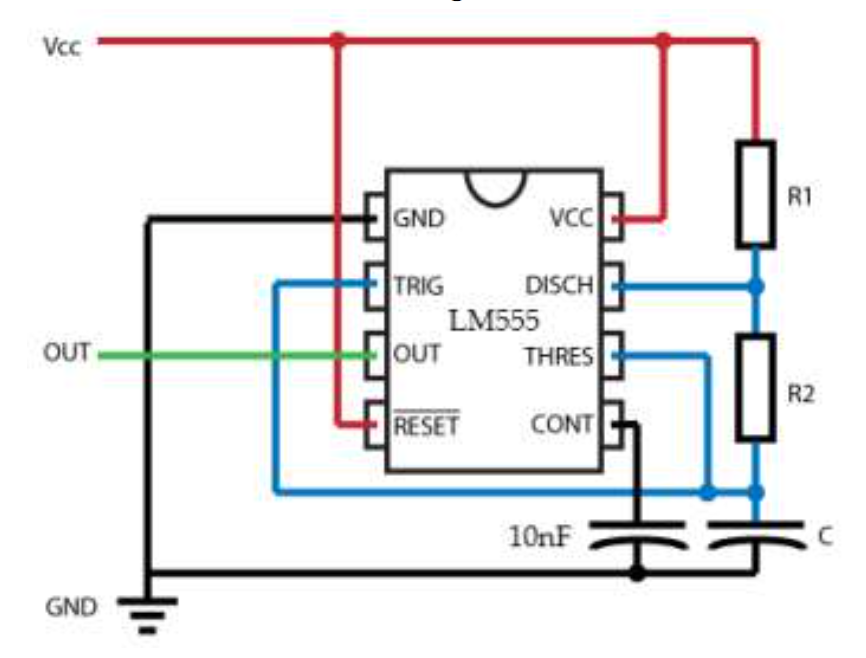


Fig. 4 LM555 in astable mode

We connected the ground pin of the IC and one end of the capacitors to the circuits ground bar, we then connected the 5v input to the reset bar and VCC pins of the IC and one of the resistors (R1). The rest of the wiring was also done based on the figure shown above (which was also in the lab instruction).

The pin OUT is used to get output we can show on the oscilloscope.

Then the experiment moves on to a new method of clock generation, the Schmitt inverter oscillator. We used the 74ls14 IC to do this part of the experiment; figure 5 shows how this IC works. (This image was obtained by a search on the Internet)

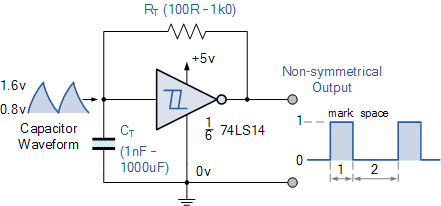


Fig. 5 An example Schmitt invertor

After we are familiarized with methods of clock generation we move on to frequency division and counters and working with the FPGA. In working with the counter we had some issues with connecting the inputs to the counter loads, we learned the logic behind the number we were giving the counter as initial value based on what number we are dividing by. Another important issue was the difference of the 12th and 13th pins, which both displayed the same logical meaning but one was active low and one active high.

We also got familiar with coding the FPGA, programming it, assigning pins and using the 7-segment display.

* Results

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.

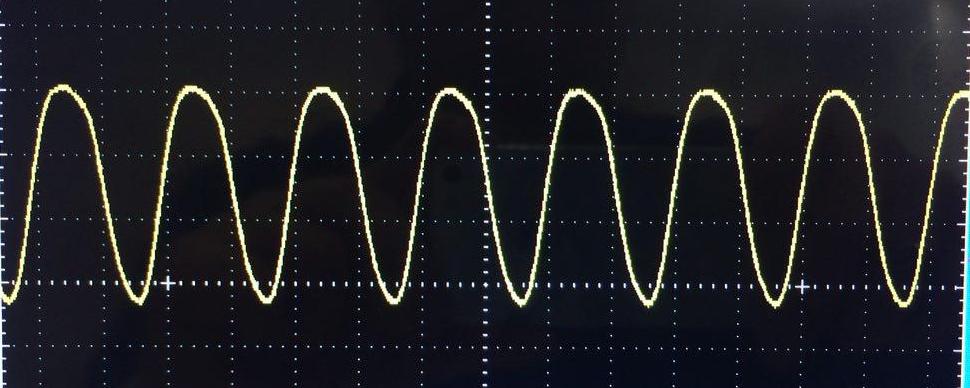


Fig. 1 Example of a ring oscillator output signal

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

Total Delay Time: 50ns

Delay time of each invertor: 50ns / 6 = 8.333 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.

Also comparing the 74ls04 delay in the datasheet and the one we got in the experiment. The delay of each inverter was said to have a minimum of 4ns and a maximum of 15 ns and the delay of each inverter based on our experiment was 8.333 ns, making the delay close to the average amount of delay mentioned in the datasheet.

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.

As we can see the frequency also decreases as the resistance of R2 increases. An example of the output wave we got is shown in the bellow figures.

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.

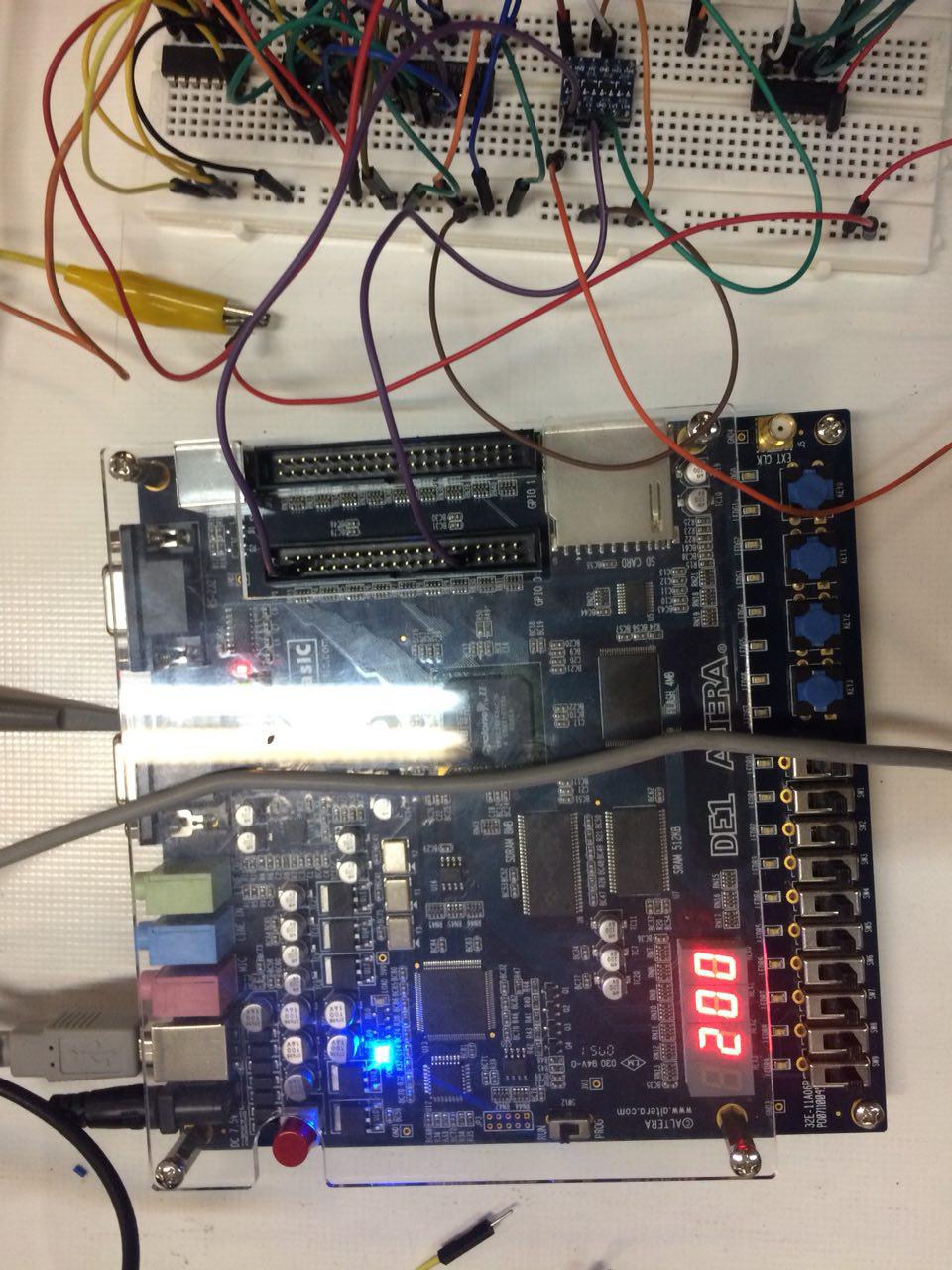


Fig. 2 Display of FPGA seven-segment

* CONCULOSION

In this experiment we learned the importance of clocks, methods of clock generation, we also got some basic understanding of what FPGA is and how to work with one.