

Robot Electronics - Lab4: Simple oscillator and clock divider using a TE0722 propeller SoC board

Group 2:

Thor Gunnlaugsson Jensen - thorj14@student.sdu.dk

Jorge Cebollada Serrano - joceb18@student.sdu.dk

Pauline Steiner - paste18@student.sdu.dk

October 22, 2018

Introduction

Assignment 1: simple oscillator

A simple oscillator built from 3 Schmitt trigger inverter inside the 74HC14 was build according to the schematic in figure 1.

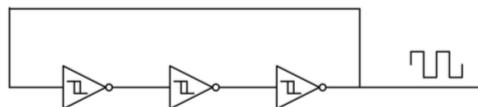


Figure 1: A simple oscillator and clock generator using three Schmitt trigger inverters.

The generated clock signal is shown in figure 2 and does not appear to be very much like a square wave. However this figure is misleading because the signal is as "ugly" as it appears in the figure. The signal appears so because of the inductance while measure with the oscilloscope probes.

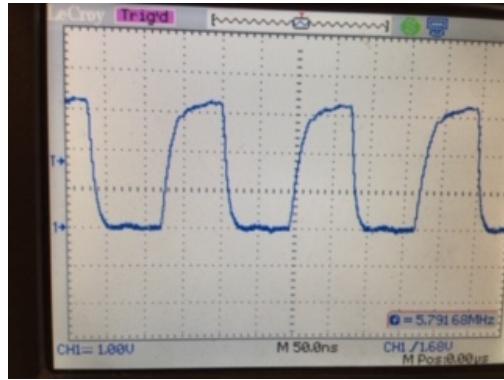
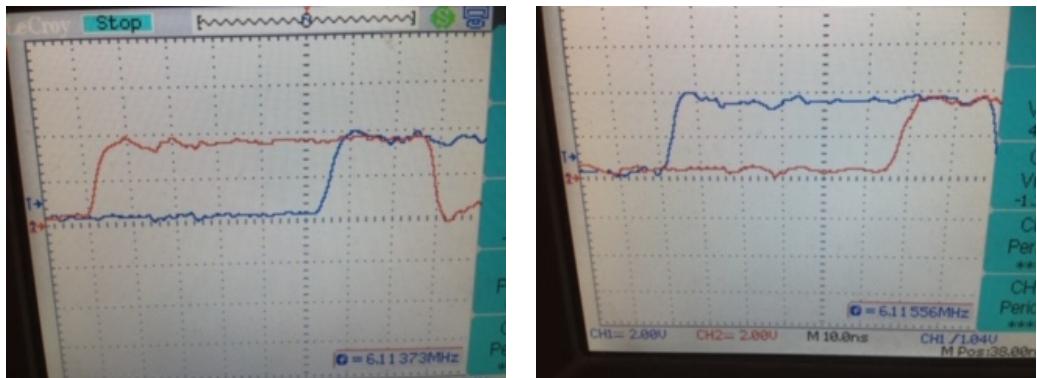


Figure 2: This is the "square" wave signal created by the three Schmitt trigger inverters.

The frequency and rise time were measured on this clock generator and are shown in figure 3 and a picture of the propagation delay between two of the Schmitt trigger inverters can be seen in figure 6. All of these values can also be seen in table 1.



Figure 3: A picture showing the propagation delay and also the values of rise time frequency and more in the right hand side.



(a) Propagation delay between the first schmitt trigger inverter.

(b) Propagation delay between the second schmitt trigger inverter.

Figure 4: Pictures of the propagation delay between each schmitt trigger inverter in the 74HC14 IC.

Frequency	≈ 5.8	MHz
Rise time	≈ 4	ns
Propagation delay pr. Schmitt trigger	≈ 55	ns

Table 1: *A table containing circuits features.*

Here is the simulation picture of an inverter in Vivado. As it can be seen in the figure 5, the input signal was forced in a 'clock signal' and each time the input is equal to 0, then the output goes to 1 and inversely when it's 1, it will be 0.



Figure 5: *Simulation of an inverter gate in Vivado*

Assignment 2: clock divider

In this part, the assignment is to create a clock divider that takes the previous output oscillator as the input. And the clock divider should return a 4-bit vector containing the clock signal divided by 2, 4, 8 and 16.

For this, we created in Vivado a VHDL file with input and outputs, these different signals as seen in the figure 6.

```
entity clk_divider is
Port (
    clk_in      : in STD_LOGIC;
    divider_o   : out STD_LOGIC_VECTOR(3 DOWNTO 0)
);
end clk_divider;
```

(a)
VHDL code of the clock divider.



(b) *Diagram of the clock divider*

Figure 6: *VHDL Code and corresponding diagram in Vivado software*

Then, a counter was implemented. And in order to test and check the veracity of our code, we ran the Vivado simulation of our implemented module. The output signal 0 has a 200 ns period whereas the clock signal has a period set-up on 100 ns. This means that that the less significant bit of the vector output signal was divided by 2 as it can be seen in the figure 7.



Figure 7: *Simulation picture that shows the clock signal divide by 2.*

Also the 4 output signals can be observed in figure 8, and it is easy to see that each one has a double period than the previous signal; that is to mean that each signal is divided by 16, 8, 4, and 2.

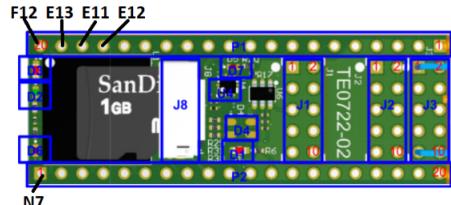


Figure 8: *Simulation picture of the 4 output signals*

Then, the mapping port should be done. The mapping of the different input and output signals to the pins of the Trenz TE0722 were done following the two mapping figure 9.

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco
All ports (5)							
divider_o[4]	OUT			✓	35	LVC MOS33*	3.300
divider_o[1]	OUT		E11	✓	35	LVC MOS33*	3.300
divider_o[0]	OUT		E12	✓	35	LVC MOS33*	3.300
divider_o[2]	OUT		E13	✓	35	LVC MOS33*	3.300
divider_o[3]	OUT		F12	✓	35	LVC MOS33*	3.300
Scalar ports (1)							
clk_in	IN		N7	✓	34	LVC MOS33*	3.300

(a) Mapping port in Vivado



(b) Trenz TE0722 pin ports

Figure 9: *Mapping signals to Trenz TE0722 pins*

In figure 10, can be seen the final circuit, after having generated the bitstream file and uploaded it into the chip.

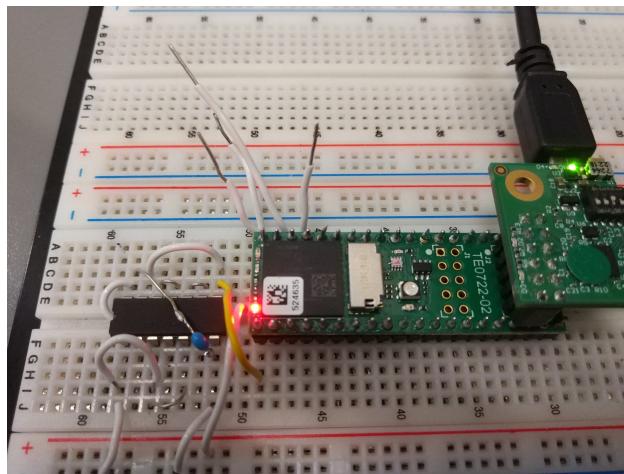


Figure 10: *The final circuit with the yellow cable regarded as the input clock signal, and the 4 white cables considered as the 4 outputs*

Unfortunately the hardware did not perform as well as the simulation even though a *.bit* file was generated. The output from the FPGA can be seen in figure 11. The noisy signal was on all four outputs unfortunately. The source of the noise was not investigated.



Figure 11: *This was supposed to be the original signal divided by eight, however some disturbance seems to be dominating the signal.*

List of Figures

1	<i>A simple oscillator and clock generator using three Schmitt trigger inverters.</i>	1
2	<i>This is the "square" wave signal created by the three Schmitt trigger inverters.</i>	2
3	<i>A picture showing the propagation delay and also the values of rise time frequency and more in the right hand side.</i>	2
4	<i>Pictures of the propagation delay between each schmitt trigger inverter in the 74HC14 IC.</i>	2
5	<i>Simulation of an inverter gate in Vivado</i>	3
6	<i>VHDL Code and corresponding diagram in Vivado software</i>	4
7	<i>Simulation picture that shows the clock signal divide by 2.</i>	4
8	<i>Simulation picture of the 4 output signals</i>	4
9	<i>Mapping signals to Trenz TE0722 pins</i>	5
10	<i>The final circuit with the yellow cable regarded as the input clock signal, and the 4 white cables considered as the 4 outputs</i>	5
11	<i>This was supposed to be the original signal divided by eight, however some disturbance seems to be dominating the signal.</i>	6