Experiment #1 - Clock and Periodic Signal Generation

Daneshvar Amrollahi

Student ID: 810197685

***Abstract*— This document is a student report to experiment #1 of Digital Logic Laboratory course at ECE Department, University of Tehran. In this series of experiments, different methods for clock generation are tested. In the first section, LTspice is used as a simulation tool using ICs and Analog components. In the second section, LM555 timer is used for clock generation and simulated in Modelsim. Finally, in the third section a frequency divider is synthesized by Quartus and simulated using Modelsim operating with one of the clocks implemented in the previous section. In each section, relevant parameters such as frequency, duty cycle, and etc are calculated.**

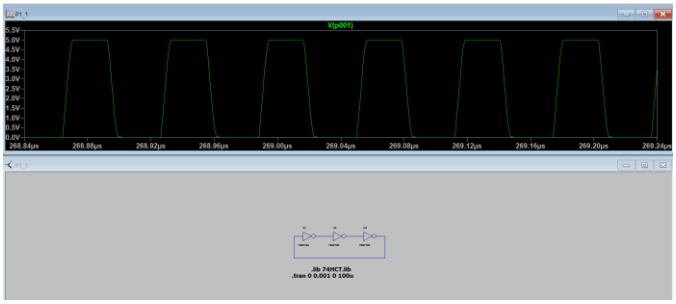
***Keywords*— Clock, Ring Oscillator, LM555, Schmitt Trigger Oscillator, Frequency Divider, 74HCT, MAXPLUS2 74, Modelsim, Quartus, LTspice**

I. Introduction

A high number of of digital circuits in different scales primarily depend on a constantly oscillating signal, aka clock cycle to work properly. Different methods have been devised and introduced for providing such signals. In this report, 3 different methods are discussed: Ring Oscillator, NE555 IC in astable mode, and Schmitt Trigger Oscillator.

II. Clock Generation using ICs and Analog Components

A. *Ring Oscillator*



Here is a simulation of a ring oscillator in LTspice using 3 inverters (74HCT04).

$$T = 2N.delay_{inv}$$

T is calculated using the two points which their x distance indicates the main period of the signal.

$$x = 267.934\ \mu s,\ y = 0.26\ mV$$
$$x = 267.996\ \mu s,\ y = 0.92\ mV$$
$$2\times3\times delay_{inv} = 0.062\ \mu s = 62\ ns$$
$$delay_{inv} = 10ns$$

B. *NE555 Timer*

An NE555 IC can used to produce a clock signal. The duty cycle can be modified by changing a resistor value in the circuit.

3 different values for *R1* is tested and the frequency and duty cycle for each one is calculated.

$$T = 0.693 \times (R_1 + 2R_2) \times c$$
$$\text{Duty Cycle} = (R_1 + R_2) / (R_1 + 2R_2)$$

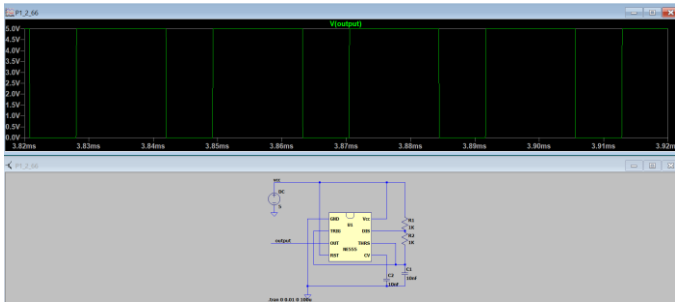
A 10nF capacitor is used for all simulations in this section.

a) $R_1 = R_2 = 1K\Omega$

$T = 0.693 \times (1 + 2) \times 10^3 \times 10 \times 10^{-9} = 20ms$

$f = 1 / T = 0.05mHz$

Duty Cycle = 66%

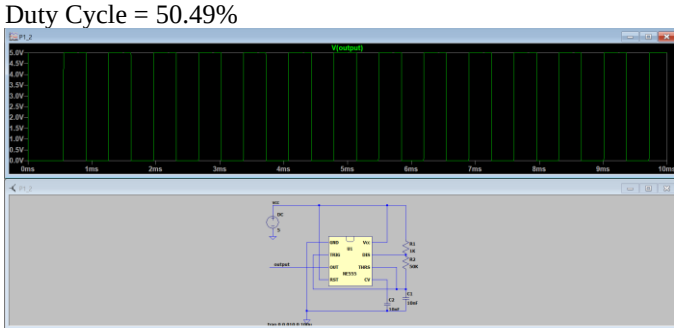


b) $R_1 = 1K\Omega$, $R_2 = 50K\Omega$

$T = 0.693 \times (1 + 100) \times 10^3 \times 10 \times 10^{-9} = 0.69ms$

$f = 1 / T = 1.44mHz$

Duty Cycle = 50.49%



c) $R_1 = 1\ K\Omega$, $R_2 = 10K\Omega$

$T = 145ms$, $f = 0.006mHz$, Duty Cycle = 52.38%

d) $R_1 = 1\ K\Omega$, $R_2 = 100K\Omega$

$T = 1392ms$, $f = 0.0007mHz$, Duty Cycle = 50.2%

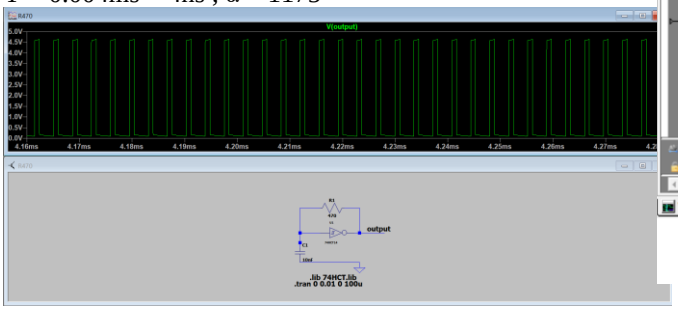
C. *Schmitt Trigger Oscillator*

A 74HCT14 is used for a Schmitt inverter oscillator.

$$f = \alpha / RC \rightarrow \alpha = RC / T$$

A 10nF capacitor is used for all simulations in this section.

a) R = 470Ω
x = 4.216ms y = 0.158V , x = 4.220ms y = 0.162V
T = 0.004ms = 4ns , α = 1175

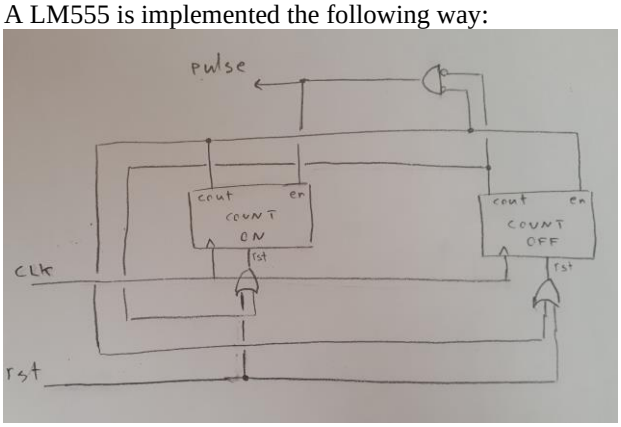


b) R = 1KΩ
x = 4.088ms y = 0.73V , x = 4.096ms y = 0.80V
T = 0.008ms = 8ns , α = 312.5

c) R = 2.2KΩ
x = 2.807ms y = 0.16V , x = 2.822ms y = 0.16V
T = 0.015ms = 15ns , α = 1466

III. CLOCK GENERATION USING VERILOG HDL

A LM555 is implemented the following way:

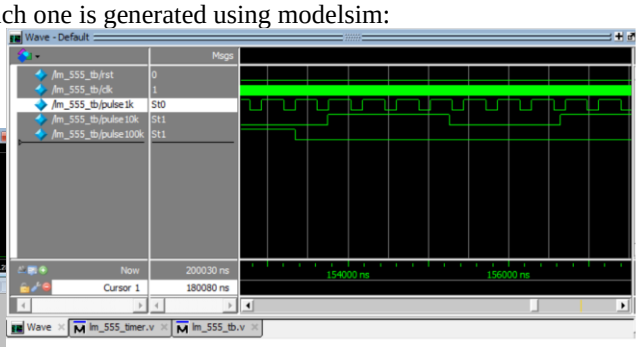


There are 2 counters. Only one of them is counting at each moment. COUNT ON counts up to *on_duration* which is calculated by the module given the values of R_1 and R_2 and c. The same thing applies to COUNT OFF which counts up to *off_duration.* These values values are calculated with these formulas:

$$on\_duration = 0.693 \times (R_1 + R_2) \times c$$
$$off\_duration = 0.693 \times (R_2) \times c$$

As soon as a counter reaches it's final value (carry out = high), it becomes disabled and the enable port of the other counter becomes high and it is resetted. Therefore it starts counting.

3 different values for R_2 are simulated and the pulse for each one is generated using modelsim:



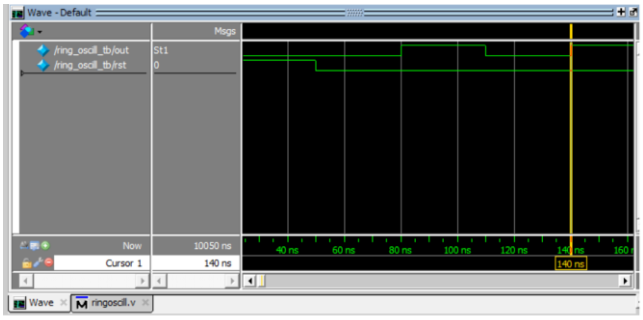
As you can see, the pulse using a 1KΩ resistor is giving us a duty cycle of 66% which we showed in the previous setion. As the value of R_2 increases, the duty cycle becomes closer to 50%.

The duty cycles were calculated in sections a, c, d of part II.B which can be verified using these waveforms.

IV. FPGA DESIGN

A. *Ring Oscillator*

A module is implemented inputting the number of inverters and delay for each inverter representing a ring oscillator by cascading the inverters. This module will be used for a clock signal in the next sections.

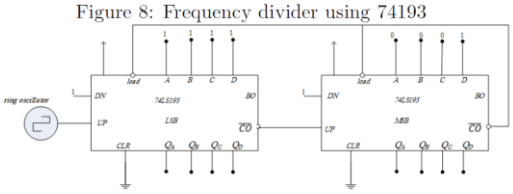


Here a ring oscillator with N = 3 inverters with $Delay_{inv}$ = 10ns is tested. Using the formula $T = 2N.Delay_{inv}$ we expect T = 60ns

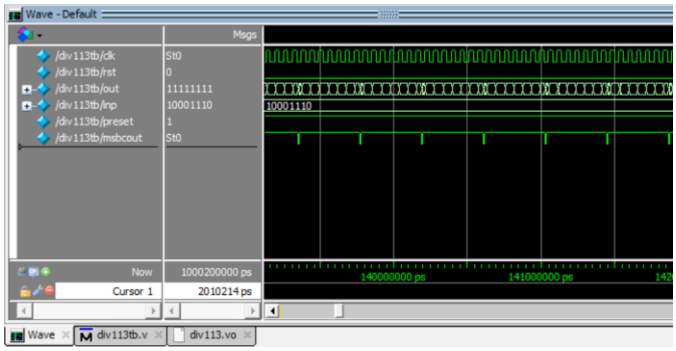
This can be verified using the waveforms. A period of the clock is from 80ns to 140ns which it's length is 60ns which is close to section I where T was concluded to be 62ns.

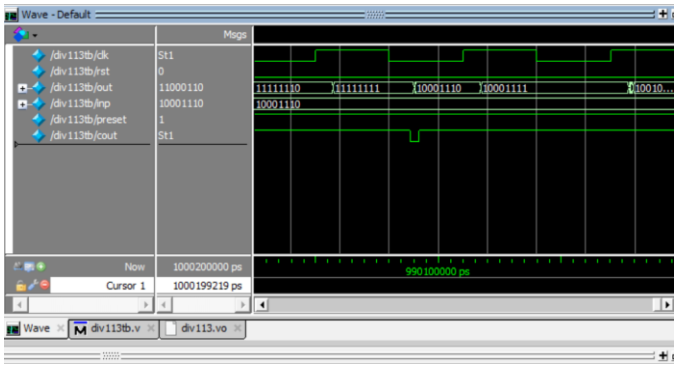
B. *Synchronous Counter as Frequency Divider*

Figure 8: Frequency divider using 74193

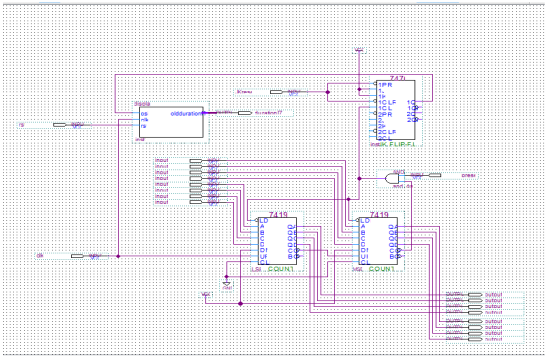


Here is a block diagram of a frequency divider using 74193. Two counters are cascaded together. When one of the 4 bit counters reaches 1111, the other one is incremented. Parallel loading is used to make the counter start counting from 10001110. Therefore it takes 11111111-10001110 = 113 clock cycles for it to reach the end. When the MSB counter reaches 1111, we have a pulse on it's carry out. This happens with every 113 clock pulses.

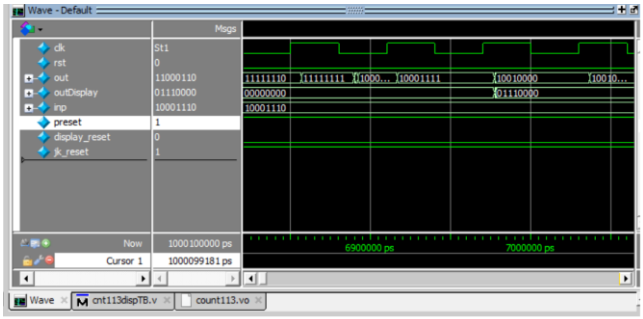




C. *Synchronous Counter as Frequency Divider with an FPGA for display*



Here is the block diagram of the 113 frequency divider with display. The display block was previously synthesized and only a block symbol of it is used in this schematic. The carry out output of the MSB counter is used as a clock of a T flip flop (A JK flip flop is converted to a T Flip Flop by connecting J to K). A high voltage is put on J and K which makes it toggle each time there is a posedge on it's clock (carry out becomes high). The output of the T flip flop is passed to display which counts up to 112 which indicates the fraction of the divided frequency whenever the counter count up to $(11111111)_2$.



V. CONCLUSIONS

Through these experiments, different approaches of producing periodic clock signals are studied. A ring oscillator was tested as a clock signal for a synchronous frequency divider circuit which worked well enough for the circuit to operate correctly. The clock signals were also parameterized so it could be used to generate different clock signals with different duty cycles. The synchronous frequency was synthesized using Quartus so the results would be as closely possible to a real implementation on an FPGA.

REFERENCES

[1] SN7407 Datasheet, Provided by Texas Instruments
[2] DM74LS191 Datasheet, Provided by Fairchild Semiconductor