Experiment 1 - Clock and Periodic Signal Generation

Student Name : Shahzad Momayez , Mohammad Amanlou
Student ID : 810100272 , 810100084

***Abstract*— This document is a report for experiment #1 of Digital Logic Design Laboratory at ECE department, University of Tehran. The purpose of this experiment is to provide different methods for clock generation.**

***Keywords*— Clock, Ring Oscillator, LM555 Timer, Schmitt Triger Oscillator, Frequency Divider, T Flip-Flop**

I. 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.

II. Clock Generation using ICs and Analog Components

A. RING OSCILLATOR

(As the TA said, we did the experiment by using 3 inverters.) Fig. 1 shows the circuit's implementation and Fig. 2 shows the output waveform.

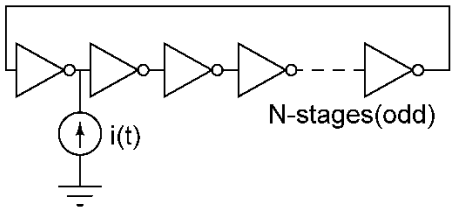


Fig. 1 Ring oscillator circuit

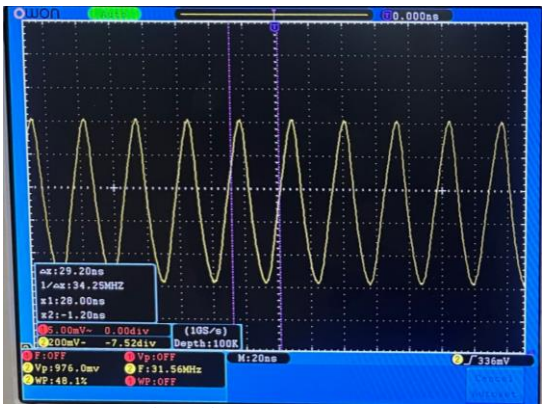


Fig2. The waveform of the ring oscillator (using 3 not gates)

As shown in the Fig2. We have the following data:
vp= 976.0 mv , Wp= 48.1% , F = 31.56 MHz

1) *Propagation Delay*: We know that the propagation delay is defined as the time from the 50% point of input to the 50% point of output. For example:

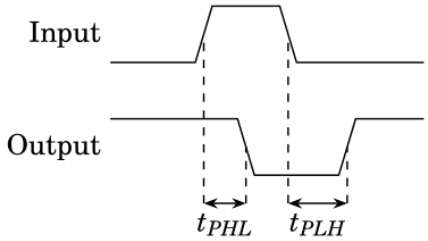


Fig3. Example of propagation delay

Propagation delay in this ring oscillator is equal to the time that output value changes from 1 to 0 or vice versa because of that the half time of a propagation from 1 to 0 is equal to half time of propagation from 0 to 1 and two half of every one is equals to complete delay of each one so that means our propagation delay is equals to half of a period time equals to T/2.

$T = 1/f = 1/(31.56 * 10^6) = 31.6$ ns
Propagation delay = $T/2 = 31.6/2 = 15.8$ ns

2) Inverter Delay: Measuring the period time of the output requires two consecutive points with same phase

time period of the ring oscillator = $2N * Delay_{inv}$

$T_{period\ time} = 2*3* Delay_{inv}$
$T_{period\ time} = 1/f = 1/(31.56 * 10^6) = 31.6$ ns
$T_{period\ time} = 31.6$ ns $= 6 * Delay_{inv}$
$Delay_{inv} = 5.3$ ns

B. LM555 Timer

Fig4. shows the LM555 Timer implementation in astable mode

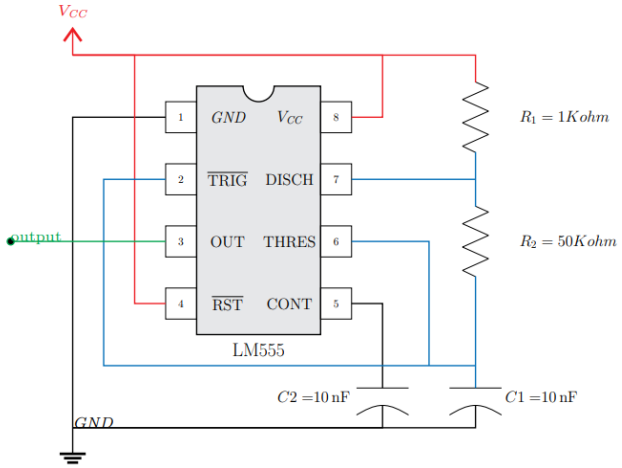


Fig4. LM555 Timer implementation

1) Change the value of R_2 resistors to produce different clock frequencies.)

We have the following equations to calculate duty cycle and frequency:

(output high) : $T_{charge\ time}=T_1 = 0.693 * (R_1 + R_2) * C$
(output low) :$T_{discharge\ time} = T_2 = 0.693 * R_2 * C$
duty cycle = $(R_1+R_2) \div (R_1+2R_2)$
(charge) : $\tau_1 = (R_1+R_2)\ C$
(discharge) : $\tau_2 = R_2\ C$
C = 10nF

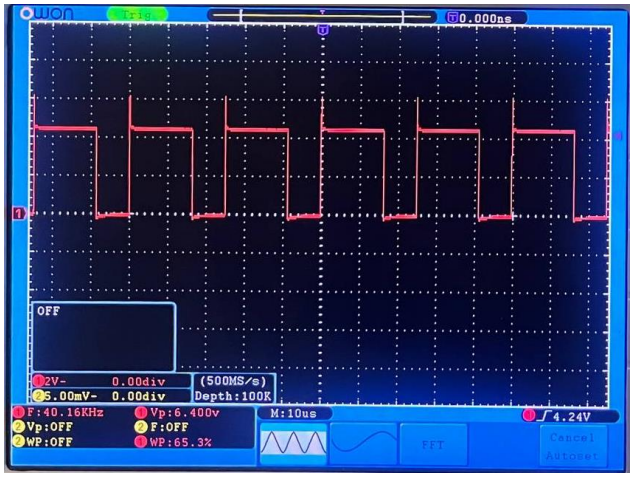


Fig5. The chart of LM555 with $R_1 = R_2 = 1\ k\Omega$

As shown in the Fig5. We have the following data:
F = 40.16 KHz , Vp = 6.400 v , Wp =65.3 %

$T_{charge\ time}=T_1 = 0.693 * (R_1 + R_2) * C$
$T_{charge\ time} = 0.693 * (1000+1000) * 10n = 13860ns$
F = 1/T = 72150 Hz

$T_{discharge\ time} = T_2 = 0.693 * R_2 * C$
$T_{discharge\ time} = T_2 = 0.693 * 1000 * 10n = 6930\ ns$
F= 1/T = 144300 Hz

duty cycle = $(R_1+R_2) \div (R_1+2R_2)$
duty cycle =(2000) / (1000+ 2000) = 67%

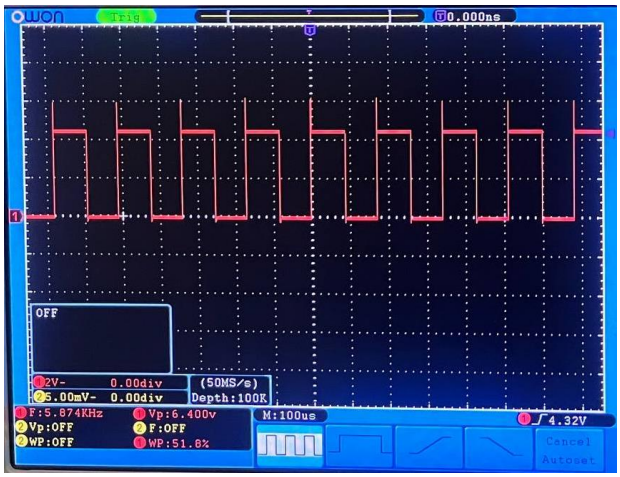


Fig 6. The chart of LM555 with $R_1 = 1\ k\Omega$, $R_2 = 10\ k\Omega$

As shown in Fig 6. We have the following data: F=5.874 KHz ,Vp =6.400v, WP=51.8%

$T_{charge\ time}=T_1 = 0.693 * (R_1 + R_2) * C$
$T_{charge\ time} = 0.693 * (11000) * 10n = 76230ns$
F = 1/T = 13120 Hz

$T_{discharge\ time} = T_2 = 0.693 * R_2 * C$
$T_{discharge\ time} = T_2 = 0.693 * 10^4 * 10^{-8} = 6.93 * 10^{-5}\ s$
F= 1/T = $1.443 * 10^4$ Hz

duty cycle = $(R_1+R_2) \div (R_1+2R_2)$
duty cycle =(11k) / (21K) = 52.3%

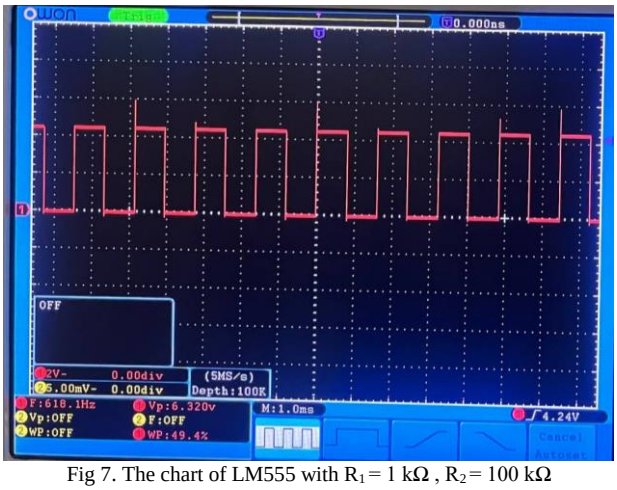


Fig 7. The chart of LM555 with $R_1 = 1\ k\Omega$, $R_2 = 100\ k\Omega$

As shown in Fig 7. We have the following data: F=618.1 KHz, Vp =6.320v, WP=49.4%

$T_{charge\ time}=T_1 = 0.693 * (R_1 + R_2) * C$
$T_{charge\ time} =0.693 * (101000) * 10n =699930ns$
$F = 1/T = 1430$ Hz

$T_{discharge\ time} =T_2 = 0.693 * R_2 * C$
$T_{discharge\ time} =T_2 = 0.693 * 10^5 * 10^{-8} =6.93 *10^{-4}$ s
$F= 1/T = 1.443* 10^3$ Hz

duty cycle = $(R_1+R_2)\div (R_1+2R_2)$
duty cycle =(101k) / (201K) = 50.2%

as we see the duty cycles that have obtain throw oscilloscope are approximately the same as the duty cycles that have obtain throw calculation. Also, the waveform shows that the duty cycle is about 50% and if we increase the resistance to 100 kΩ the duty cycle gets closer to 50%

C. Schmitt Triger Oscillator

Fig. 8 shows the implemented circuit.

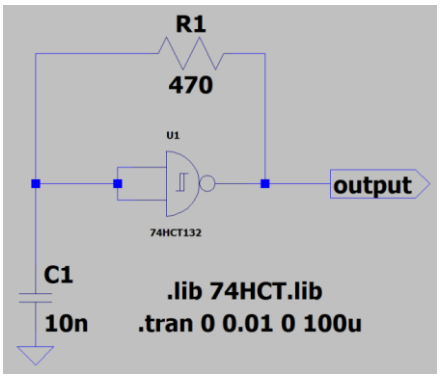


Fig. 8 Schmitt trigger oscillator

We implemented the circuit with different values for the resistor and observe the changes. (C = 10 nF)
We know that f = α/ RC

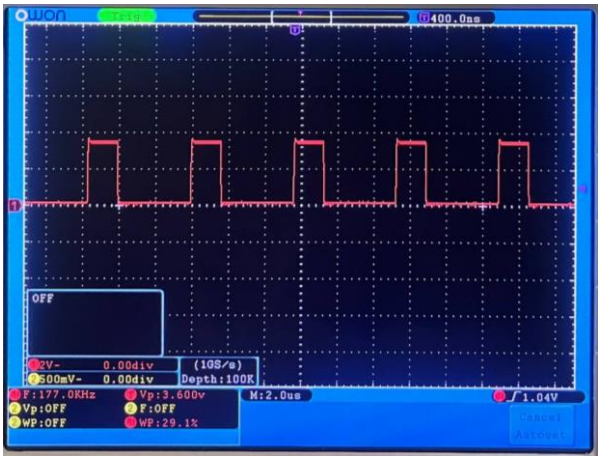


Fig9. R =470 Ω

As shown in the Fig9. We have the following data:
F= 177 KHz , Vp = 3,600 v , Wp =29.1%
α = fRC
$\alpha =177*1000*470 * 10*10^{-9}=8,319 *10^{-1}$

Fig10. R= 1000 Ω

As shown in the Fig10. We have the following data:
F= 86.08 KHz , Vp = 3,760 v , Wp =26.1%

α = fRC
$\alpha = 86000*1000*10\ n = 8.6 * 10^{-1}$

Fig 11. R = 2200 Ω

As shown in the Fig11. We have the following data:
F= 33.96 KHz , Vp = 3,920 v , Wp =16,3%
α = fRC
α = 33960*2200*10 n = 7,4 * 10^{-1}

D. *Synchronous Counter as a Frequency Divider*

In Fig12. Frequency divider using 74193 is pictured.

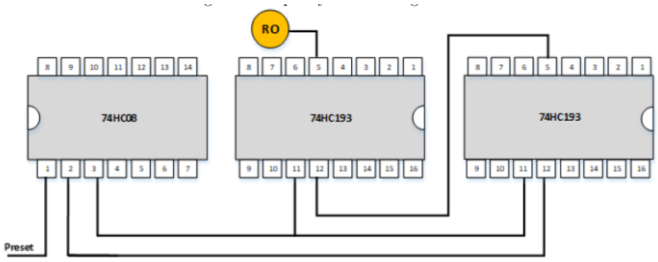


Fig12. Frequency divider using 74193

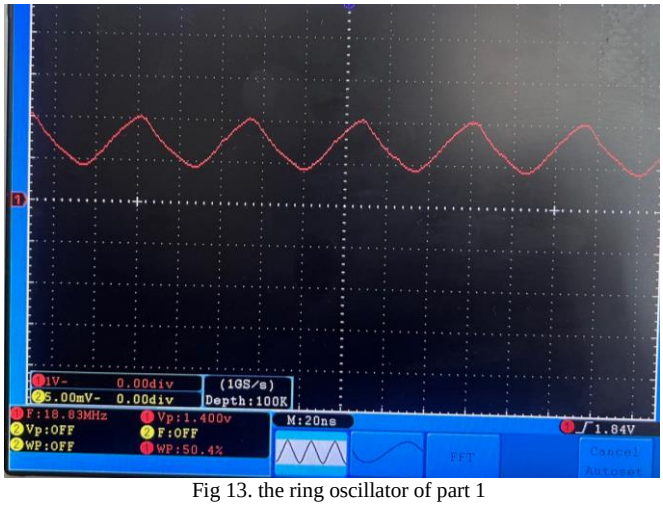
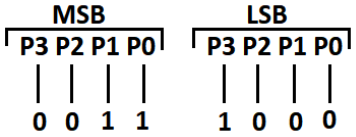


Fig 13. the ring oscillator of part 1

As shown in the Fig13. The ring oscillator has the F= 18.83 MHz and we want to divide its frequency by 200.
For our purpose, we use a 8-bit counters. and for this purpose we cascade 2 4 bit counters. As we know counters can be used as a frequency divider. 74LS193 is a synchronous 4-bit up/down counter. counters can be cascaded when the modulus is more than 4 bits. And we cascade 2 of them together.
By cascading we have 8 bit counter. (maximum value =2^8= 256)
we know that when up counting is desired the initial value is obtained by:
Initial value = Maximum value – Modulus
Initial value = 256 -200 = 56

Now we must give the Initial value(56) to our counters.
$(56)_{decimal}$= $(00111000)_{binary}$



For the MSB we connected P1 and P0 to Vdd and connected P2 and P3 to Gnd.
For the LSB we connected P3 Vdd and connected P2 ,P1 and P0 to Gnd.

The mechanism includes an AND gate for anding a preset input signal with the load inputs of the counter. We perform this presetting by using 7408 AND gate as shown in Fig12.

Fig14. Result frequency

As shown in the Fig14. We have the following data:
F=161.8 KHz , Vp = 4.88 v , Wp= 98.7%

In this case our duty cycle is not 50%. In order to decrease the duty cycle to 50% we should use a TFF.

E.T Flip-Flop

We know in TFF output toggles when input signal is issued. So every two toggles is equal to one period. And that mean each toggle is equal to half of period and we know it is a 50% duty cycle.
In this experiment we use a D Flip-Flop to create a T FlipFlop just by wiring Qbar to the data

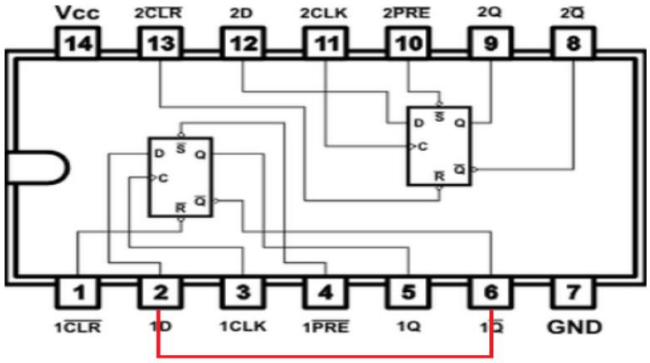


Fig15. DFF connection for making a TFF

Fig 16. Result frequency with TFF

As shown in the Fig14. We have the following data:
F=82.08 KHz , Vp = 4.08 v , Wp= 50.0%

By experimenting this part we see that the initial frequency has almost divided by 200.

F_{input} /200 = 18.83 * 10^6 / 200 = 94150 Hz =94.150 KHz
So we conclude that F_{output} is almost equal to F_{input} /200.

III. Conclusions

A clock generator is an electronic oscillator that produces a clock signal by making series of odd number of invertors and it is used for synchronizing a circuit's operation. Computing the delay of each gate in component which generate the clock by using of clocks frequency.
There is variety of clock generator that includes a counter. We saw that some of them produces 50 percent duty cycle (e.g., ring oscillator and LM555 timer) and some others don't (e.g., Schmitt Trigger). Although duty cycle of these produced signals may differ, but can convert to 50 percent just by adding a TFF. Another property of clock signals is periodicity that can convert to any value by make use of counters.

Acknowledgment

This report was prepared and developed by Mohammad Amanlou(SID:810100084) and Shahzad Momayez(SID:810100272), bachelor students of Computer engineering at University of Tehran, under the supervision of Professor Zain Navabi.