ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY EXPERIMENT REPORT

EXPERIMENT NO	:	4
EXPERIMENT DATE	:	07.04.2023
LAB SESSION	:	FRIDAY - 14.00
GROUP NO	:	G3

GROUP MEMBERS:

150200054 : ASLI YEL

150190028 : SEVİM EFTAL AKŞEHİRLİ

SPRING 2023

Contents

1 INTRODUCTION

The study of data storage components specifically latches and flip-flops, is the primary aim of this experiment. Flip-flops and latches are essential parts of digital systems that allow binary data to be maintained and stored. Designing and analyzing complex digital systems requires an in-depth knowledge of their principles, functionality, and characteristics. This experiment attempts to investigate the functionality, uses, and performance of latches and flip-flops through implementation and analysis. Participants will improve their knowledge of data storage methods and their capacity to create effective digital systems by gaining knowledge about these key elements.

2 MATERIALS AND METHODS

2.1 PRELIMINARY

2.1.1 How Latches and Flip-Flops Work?

Binary data is stored using sequential logic circuits which include latches and flip-flops. They use feedback loops to operate, which enables them to remember information over time.

The most basic type of sequential circuit is a latch. Two cross-coupled NOR or NAND gates make up the device. The latch can remember its former state thanks to the feedback loop created by the gates. The behavior of the latch is controlled by its inputs, also referred to as the set (S) and reset (R) inputs. The latch stores a high (1) output when the set input is triggered and a low (0) output when the reset input is activated. Until a new input signal is applied and the latch changes state, it holds its output.

A flip-flop is a more complex sequential circuit made up of many control inputs and multiple latches. The D flip-flop, which has a data (D) input, a clock (CLK) input, and an output (Q), is the form of flip-flop that is most frequently used. The value to be stored in the flip-flop is determined by the D input. The flip-flop records the value of the D input and saves it in its internal state as the CLK signal changes from low to high. The output (Q), which is unchanged until the following clock edge, presents the stored value.

Latch and flip-flop circuits, in summary, are sequential circuits that store binary data. Flip-flops store data based on clock edges and the value of the data input, and latches store data that depends on the activation of set and reset inputs.

2.1.2 The Circuit Designs For Experiment Parts

For each part of the experiment, we designed and drew the circuit to be implemented. You can find each of them in the corresponding section under the heading 'Experiments'.

2.1.3 Inputs for Shift Register

Before the 4th part of the experiment we determined which data input should be loaded to the shift register for each specific pulse generation case. You can find them in Section 2.2.4.

2.2 EXPERIMENTS

2.2.1 Part1 - SR Latch without an Enable Input

In the first part of the experiment we were asked to design and implement an SR Latch only with NOR Gates. For this purpose, two cross-coupled NOR Gates which form a feedback loop as in the following design were sufficient.

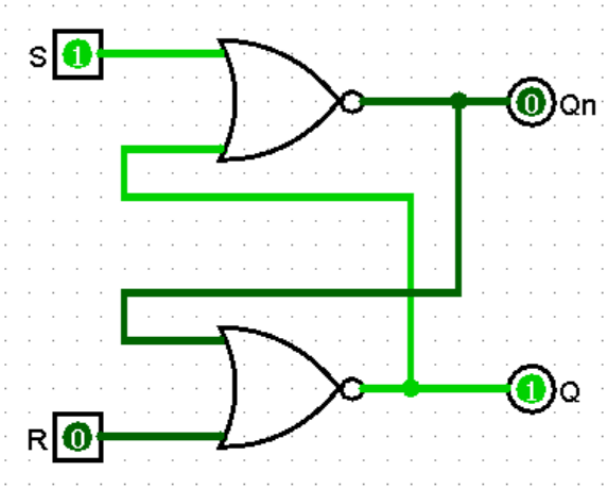

Figure 1: Design of an SR Latch without an Enable Input

For the implementation of this circuit,

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx02 - Quadruple 2-input Positive NOR Gates

are used.

2.2.2 Part2 - SR Latch with Enable Input

In the second part of the experiment an SR Latch with an Enable input that consists of only NAND Gates was required to be designed and implemented. Design of a latch with these properties looks like the following.

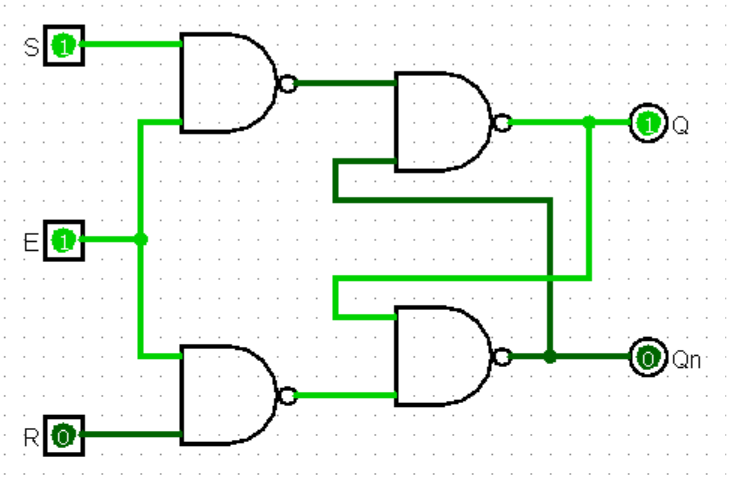

Figure 2: Design of an SR Latch with Enable Input

For the implementation of this circuit,

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx00 - Quadruple 2-input Positive NAND Gates

are used.

2.2.3 Part3 - Negative Edge Triggered D Type Flip-Flop

In this part of the experiment, we designed and implemented a negative edge triggered D type Flip-Flop using two D type latches.

In order to use as a block in the design of D Flip Flop, first we designed a D Latch as following. It is basically obtained by sending D signal to the S input, inverted signal of D to the R input of an SR Latch with enable input.

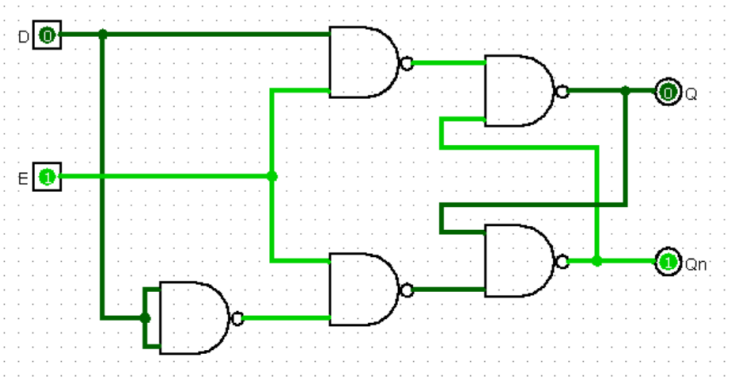

Figure 3: Design of a D Type Latch

To design a D Fip Flop, two D Latches are connected so that first one's output Q feeds the input D of the second one. In order to make it negative edge triggered, first one's enable input is connected directly to the clock signal while the second one's is connected to the inverted clock signal. How this configuration is effective is shown later in section 3.3.

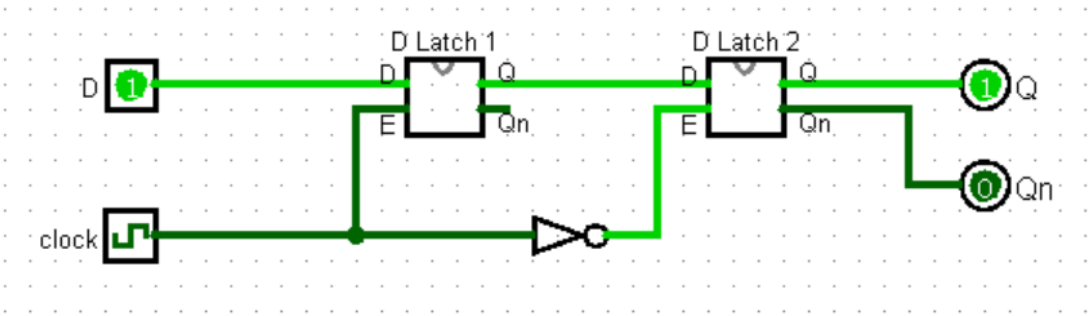

Figure 4: Design of Negative Edge Triggered D Type Flip-Flop

For the implementation of this circuit, we used

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx04 - Hex Inverters
  - 74xx75 - Quadruple Bistable D Type Latches

2.2.4 Part4 - Pulse Generator from a Shift Register

In this part of the experiment a pulse generator that supports variable pulse frequencies and durations is implemented by using an 8-bit Parallel-Load Shift Register. The circuit for this purpose is designed like below. The IC serie 74xx165 - 8-Bit Parallel Input/Serial Output Shift Register is used as a block.

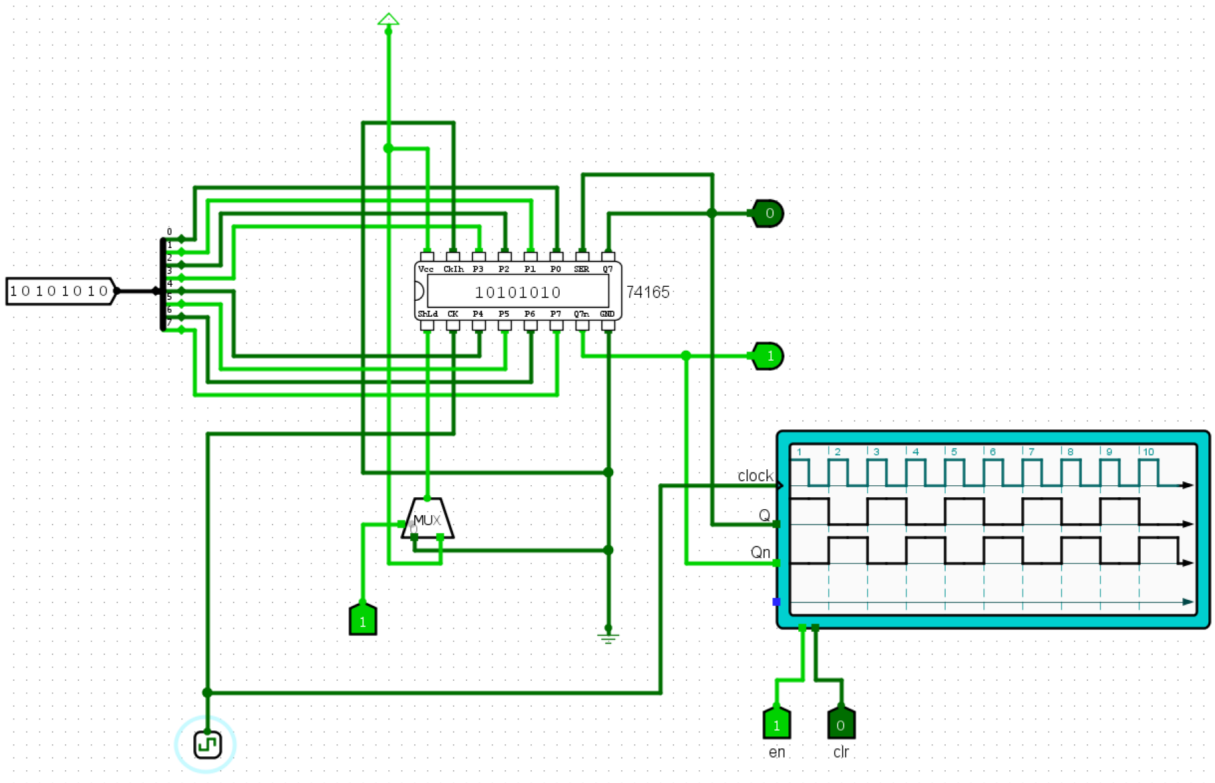

Figure 5: Design of Pulse Generator

This shift register takes an 8 bit data input. When SH/$\bar{LD}$ input (SH for Shift and LD for Load) is low, parallel load of 8 bit data occurs.

On the other hand, when SH/$\bar{LD}$ is high, the 8 bit data stored in the register is shifted toward a serial output Q at every rising clock edge. MSB of the data becomes the serial output and a serial input provided to the register becomes the LSB.

If the serial output is sent back to the register as the serial input, then the register performs a circular shift. This is what we will do, since we want to generate some special periodic signals.

According to the input loaded to the register before starting the shift operation, succession of serial outputs at each clock creates a signal pattern. And a relation between the frequency of this signal and the clock signal can be easily established. In order to observe this signals and the relation between them, both are sent to an oscilloscope.

To observe the signals with properties as following, a special input for each case should be parallel loaded to the register and then serial shift should be started.

- Case1: with the 1/2 frequency of input
  Input: 10101010
- Case2: with the 1/4 frequency of input
  Input: 11001100
- Case3: with the 1/8 frequency of input
  Input: 11110000
- Case4: with 1/3 pulse–gap duration rate
  Input: 10001000
- Case5: with 1/7 pulse–gap duration rate
  Input: 10000000

For the implementation of this circuit,

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx165 - 8-Bit Parallel Input/Serial Output Shift Register
- Oscilloscope

are used.

2.2.5 Part5 - Circular Up Counter

In the last part of the experiment an up counter that counts 0 to 5 in a circular way is implemented. The circuit for this purpose is designed like below. The IC serie 74161 - BCD Decade Counter / 4-Bit Binary Counter is used as a block.

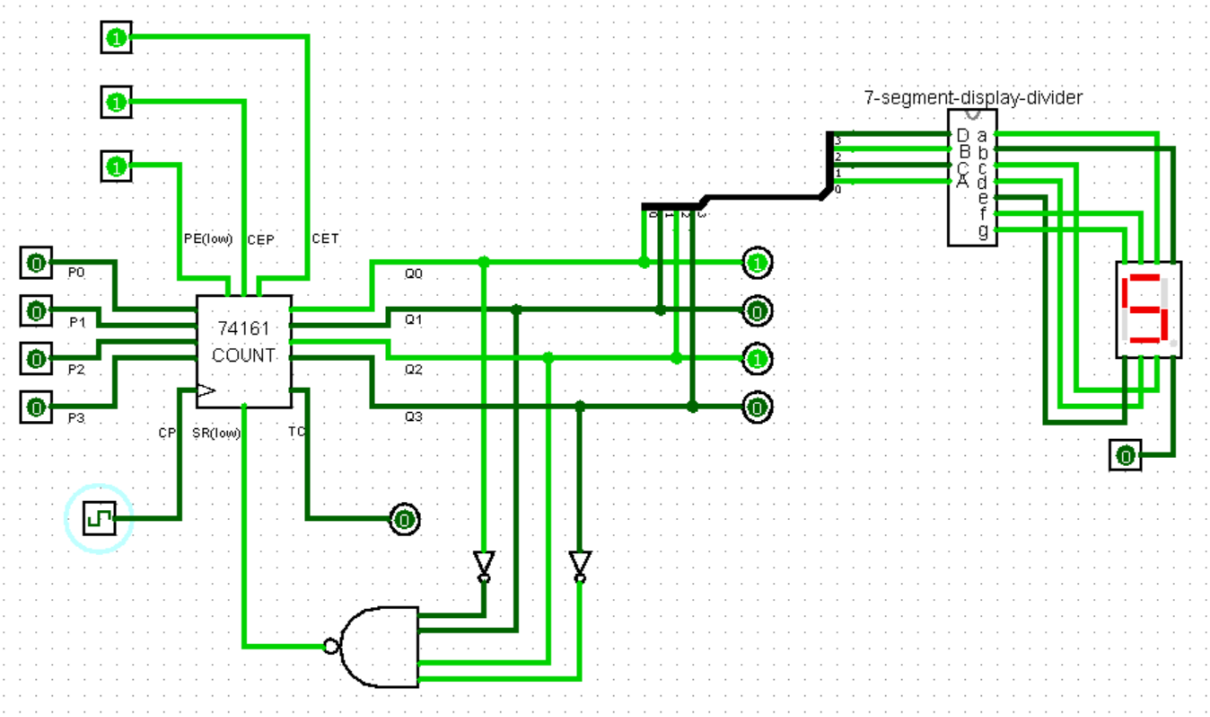

Figure 6: Design of Circular Up Counter

This counter has 4bit parallel inputs(P0-P3), a Count Enable Parallel Input (CEP), Count Enable Trickle Input (CET), Parallel Enable (Active LOW) Input ($\bar{PE}$), Synchronous Reset (Active LOW) Input ($\bar{SR}$), Clock (Active HIGH Going Edge) Input (CP), Terminal Count Output (TC), 4bit parallel outputs (Q0-Q3).

When SR is high and PE is low, parallel load occurs. Each Pn is loaded to the corresponding Qn.

When SR and PE are high and at least one of the CEP and CET input is low, it holds the current state, there is no change.

When SR, PE, CEP and CET inputs are all high, counter starts to count. It actually increments from the current state.

In order to observe the counted numbers in decimal, parallel outputs (Q0-Q3) are passed through a 7-segment-display-driver and sent to a 7-segment-display. Here, numbers can be displayed digitally.

When SR input is low counter resets to 0. Since we want our counter to count up to 5 and then turn back to 0 and then again count to 0 etc., we needed to design a mechanism that stops and resets the counter when it is about to count to 6. Parallel outputs corresponding to the decimal number 6 are Q3=0, Q2=1, Q1=0, Q0=0. So if we take the inversion of Q3, Q2, Q1 and inversion of Q0, send them to a 4-input NAND gate and connect the output of the NAND gate to the SR input, we will accomplish our object. Because the output of the NAND gate, so the SR input, will be low if and only if the output combination of the counter is 6 in decimal. Once the counter is about to reach to 6, the output signals immediately cause it to reset and not count after 5.

For the implementation of this circuit,

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx00 - Quadruple 2-input Positive NAND Gates
  - 74161 - BCD Decade Counter / 4-Bit Binary Counter
- Seven Segment Display

are used.

3 RESULTS

3.1 Part1 - SR Latch without an Enable Input

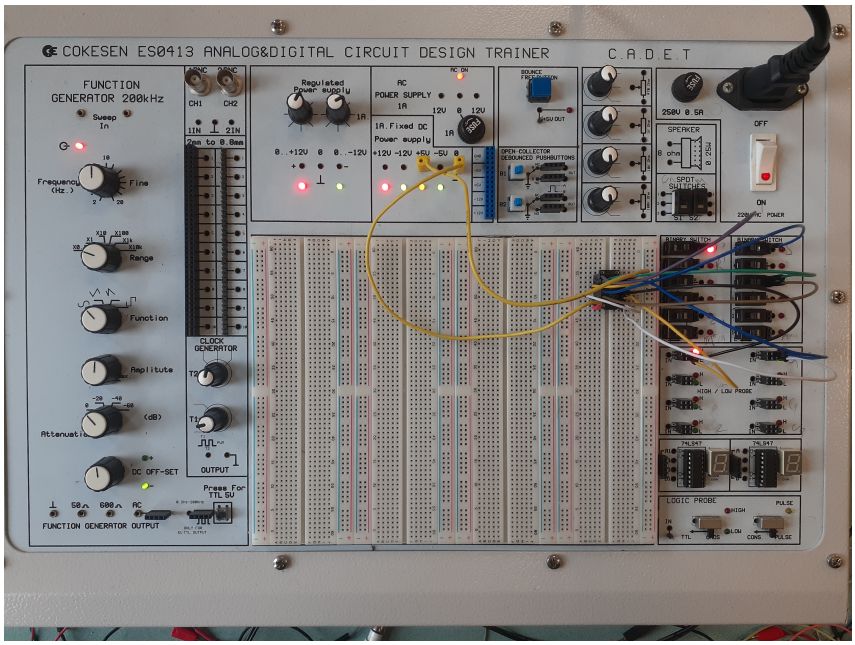

Figure 7: Implementation of SR Latch without an Enable Input

After implementation, we tested our circuit and obtained the results for all possible inputs and ensured that a truth table can be filled as following.

S	R	Q(t)	Q(t+1)		
0	0	0	0	Q(t)	No Change
0	0	1	1		
0	1	0	0	0	Reset
0	1	1	0		
1	0	0	1	1	Set
1	0	1	1		
1	1	0	ϕ	ϕ	Forbidden Inputs
1	1	1	ϕ		

Table 1: Truth table for an SR Latch without an Enable Input

The characteristic equation of an SR Latch without an Enable Input:

$Q(t+1) = f(S, R, Q(t)) = S + Q(t)\bar{R}$ $(SR = 0)$

3.2 Part2 - SR Latch with Enable Input

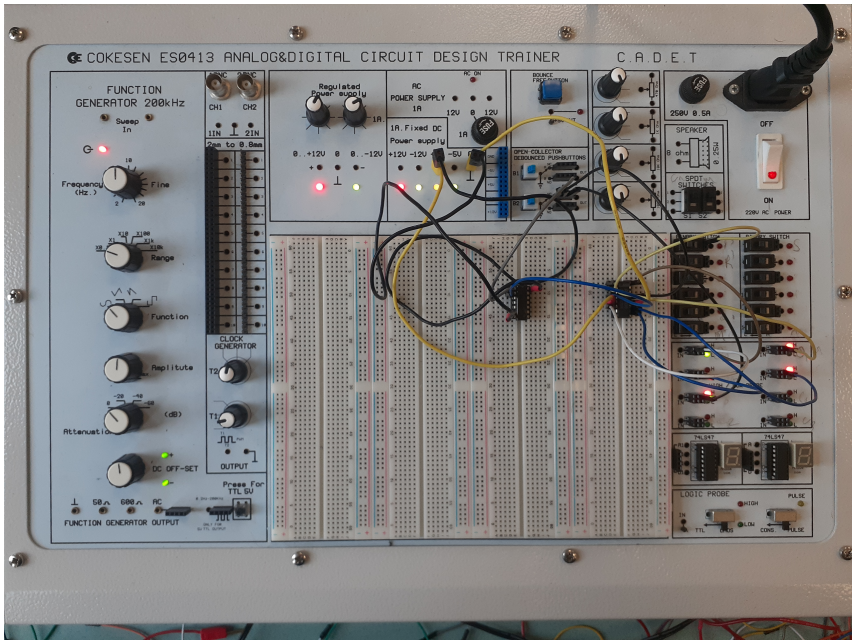

Figure 8: Implementation of SR Latch with Enable Input

After implementation, we tested our circuit and obtained the results for all possible inputs and ensured that a truth table can be filled as following.

E	S	R	Q(t)	Q(t+1)		
0	ϕ	ϕ	0	0	Q(t)	Not Enabled
0	ϕ	ϕ	1	1		
1	0	0	0	0	Q(t)	No Change
1	0	0	1	1		
1	0	1	0	0	0	Reset
1	0	1	1	0		
1	1	0	0	1	1	Set
1	1	0	1	1		
1	1	1	0	ϕ	ϕ	Forbidden Inputs
1	1	1	1	ϕ		

Table 2: Truth table for an SR Latch with an Enable Input

The characteristic equation of an SR Latch with Enable Input:

$$Q(t+1) = f(S, R, E, Q(t)) = ES + Q(t)\bar{R} + \bar{E}Q(t)$$

3.3 Part3 - Negative Edge Triggered D Type Flip-Flop

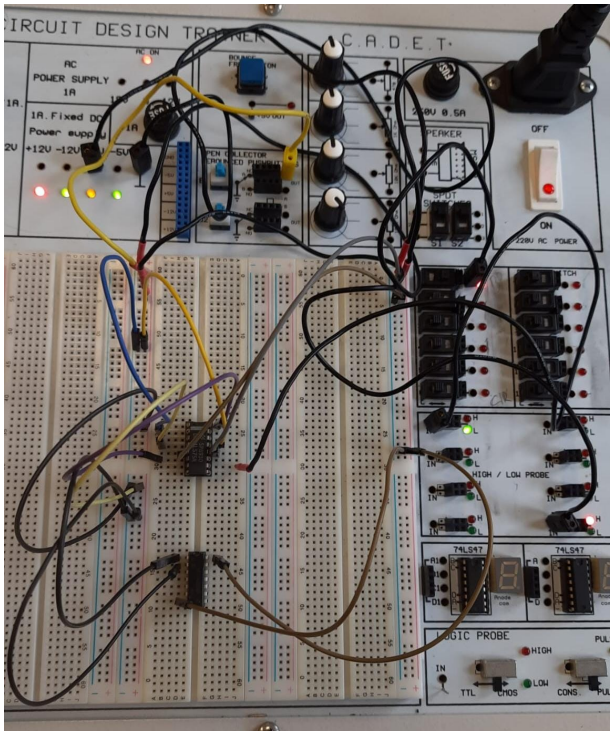

Figure 9: Implementation of Negative Edge Triggered D Type Flip-Flop

After implementation, we tested our circuit and obtained the results for all possible inputs and ensured that a truth table can be filled as following.

D	CLK	Q(t)	Q_N(t)	Q(t+1)		Q_N(t+1)	
0	falling	0	1	0	0	1	1
0	falling	1	0	0		1	
1	falling	0	1	1	1	0	0
1	falling	1	0	1		0	
ϕ	0	0	1	0	Q(t)	1	Q_N(t)
ϕ	0	1	0	1		0	
ϕ	1	0	1	0	Q(t)	1	Q_N(t)
ϕ	1	1	0	1		0	

Table 3: Truth table for a Negative Edge Triggered D Type Flip-Flop

The characteristic equation of a D Flip-Flop:

Q(t+1) = D

We also tested and clarified that this circuit is effective only at the falling edge of the clock. Remind that first D latch's enable input was fed by the clock signal directly and the second one's was by the inversion of the clock signal. In this regard, the following conclusion can be reached:

When clock is high; first latch is enabled and it stores the input D, second latch is disabled and the output of the flip flop, the whole system, does not change.

When clock is low; first latch is disabled and it preserves the data that is stored when clock was high, second latch is enabled and the output of the first latch is forwarded to the second latch, the value that previously stored in the first latch becomes the output of the flip flop.

And this shows that the flip flop changes its output only when a transition of clock signal from 1 to 0 happens.

3.4 Part4 - Pulse Generator from a Shift Register

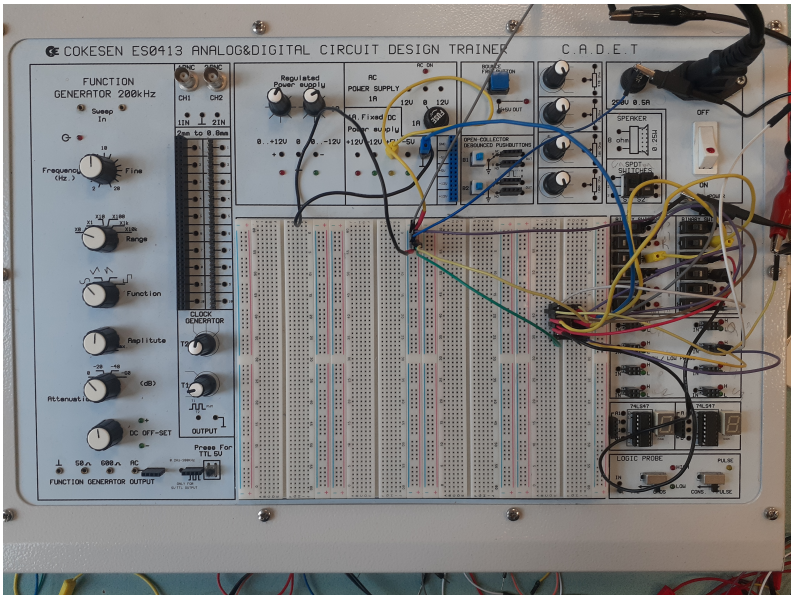

Figure 10: Implementation of a Pulse Generator from a Shift Register

- Case1: with the 1/2 frequency of input
  Input: 10101010

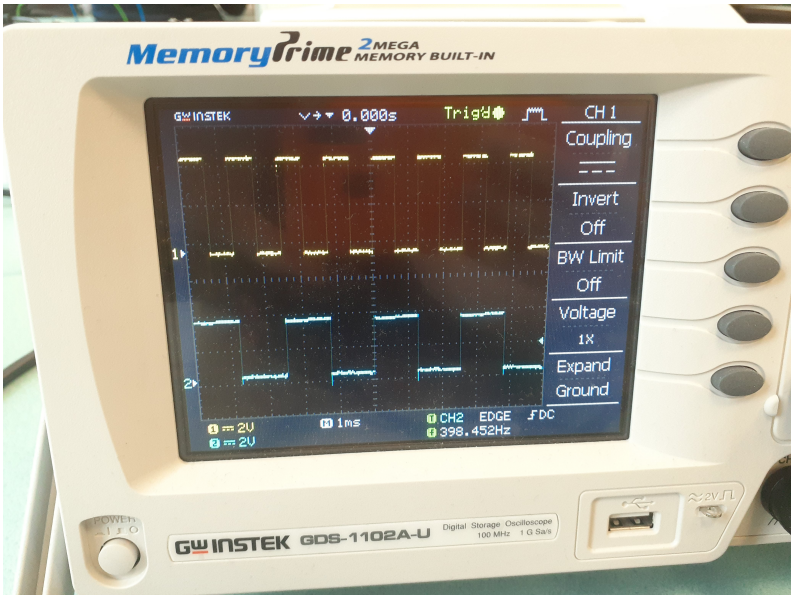

Figure 11: Oscilloscope Display of the signal with the 1/2 Frequency of Input

- Case2: with the 1/4 frequency of input
  Input: 11001100

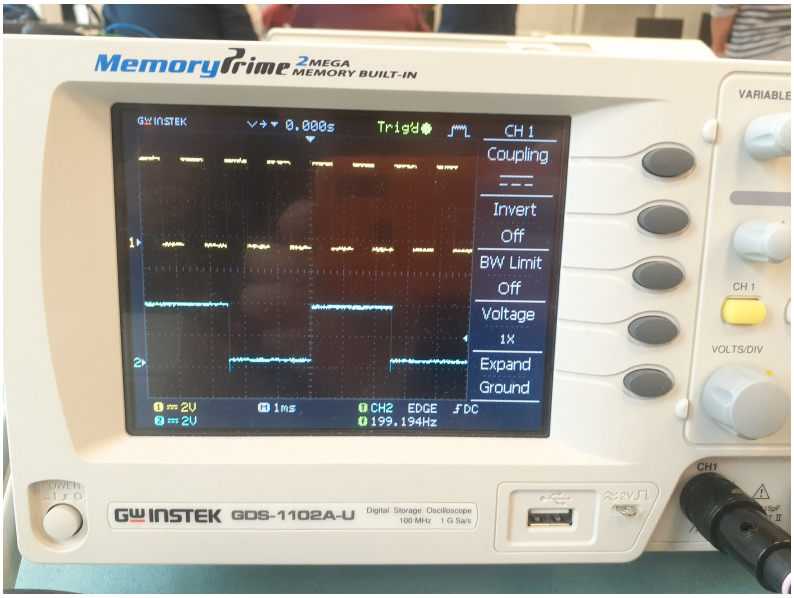

Figure 12: Oscilloscope Display of the signal with the 1/4 Frequency of Input

- Case3: with the 1/8 frequency of input
  Input: 11110000

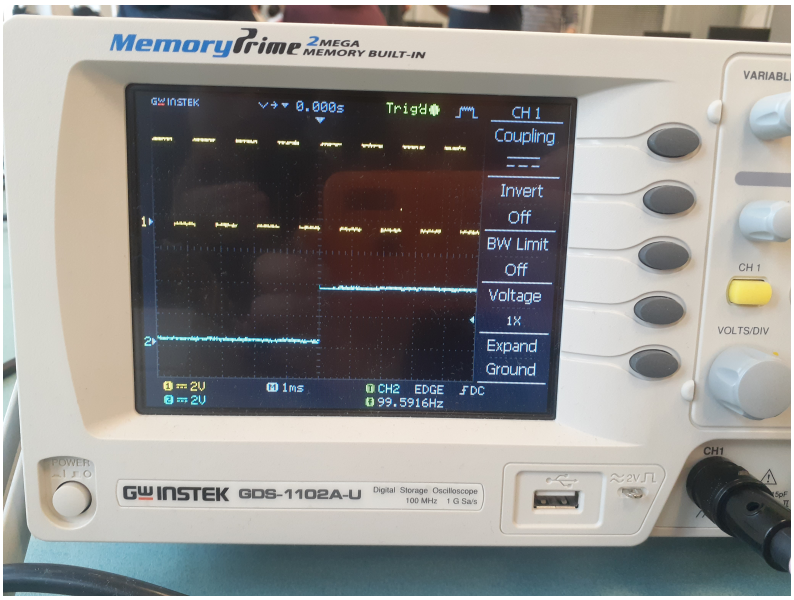

Figure 13: Oscilloscope Display of the signal with the 1/8 Frequency of Input

- Case4: with 1/3 pulse–gap duration rate
  Input: 10001000

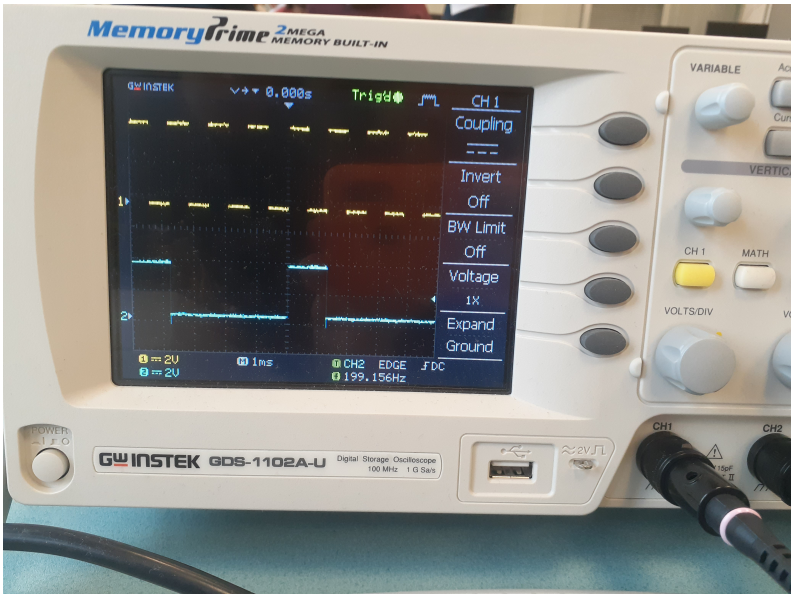

Figure 14: Oscilloscope Display of the signal with 1/3 Pulse–Gap Duration Rate

- Case5: with 1/7 pulse–gap duration rate
  Input: 10000000

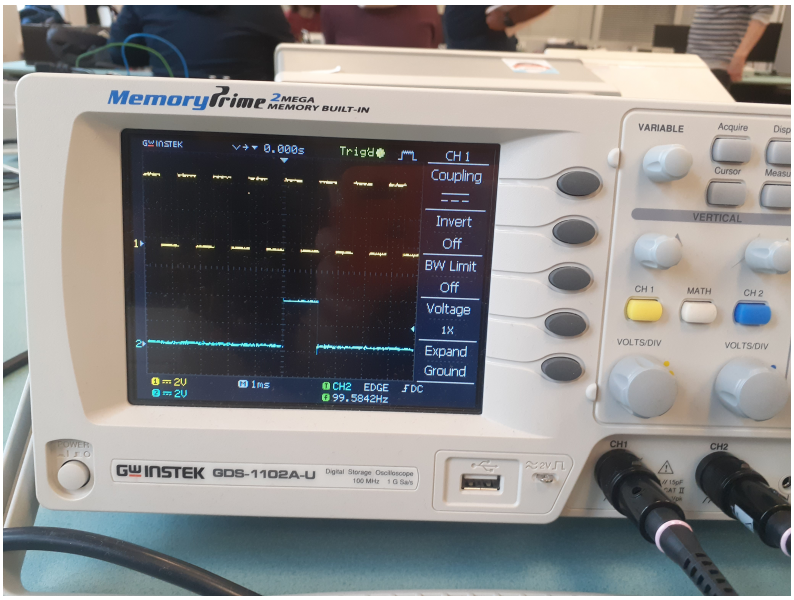

Figure 15: Oscilloscope Display of the signal with 1/7 Pulse–Gap Duration Rate

3.5 Part5 - Circular Up Counter

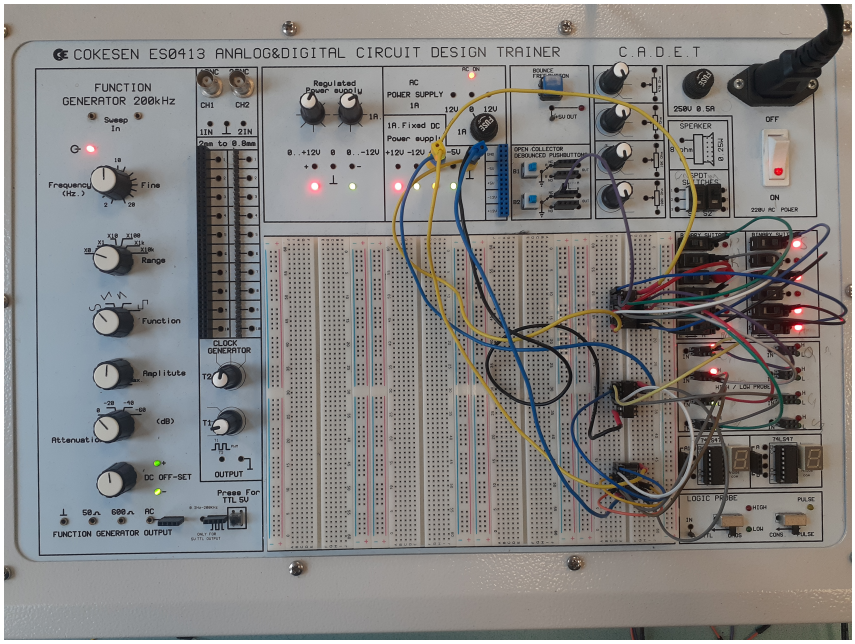

Figure 16: Implementation of Circular Up Counter

4 DISCUSSION

For the first part of the experiment, we took input from the switches for our S(Set) and R(Reset) values. We connected our inputs with the binding method that should be for NOR Gate and created a latch. When we tested all values, we got correct results for values that are not HIGH(1) for both inputs. Since we used NOR gate when both values were 1, the error we got was as follows: The outputs of the two NOR gates in the latch are 0 when both S and R inputs are 1 (active). Additionally, this produces a feedback loop with both of the NOR gates' inputs set to 0. It results in an undetermined output condition.

For the second part of the experiment, In order to add the Enable input to the S-R Latch we obtained in the first experiment, we processed our E value with our S and R values separately in the NAND gate. Thus, we made it possible to control our Latch with the Enable value. Then, we created our new Latch using the NAND gate with our S and R values obtained after processing with Enable on the AND gate. The effect of the enable input on the circuit was as follows: When it took the value of LOW(0), the change of S and R values did not affect the result and the circuit did not work. We observed that the latch enters an unsure or indeterminate state when the Enable input is active (1) and both the S and R inputs are active (1) simultaneously, which is the forbidden condition. Due to the feedback loop produced by the NAND gates, the output can oscillate or "glitch" between 0 and 1.

For the third part of the experiment, two D-type latches and a single inverter are used to implement a negative edge-triggered D-type flip-flop. We used a debounced pushbutton for the clock input and a switch for the D value. We defined an Enable input to get the result. While sending the Enable value only to the second D Latch, we created the E' value with the inverter. When we evaluated our circuit, we observed that it was working correctly.

5 CONCLUSION

In this experiment, we first created S-R Latch using NOR gate, and in the second part, we examined the S-R Latch we designed with a NAND gate by adding Enable input. We then created the clock signal and evaluated the value change in D Flip-Flop. Then we created a Pulse Generator with an 8-Bit Parallel Input/Serial Output Shift Register and tested the value change according to the clock signal. For the last part of the experiment we created a counter but we couldn't get the right values.

REFERENCES