Report 7: Logic gates and digital electronics

Group MER3: Ziglio Simone, Furlan Riccardo*

November 29, 2021

1 Introduction

In this lab experience we have studied the basics of digital electronic through the usage of various circuits based on logic gates.
Firstly we have implemented a basic **NOR gate** with a transistor, then a **XOR gate** made up of four NAND gates and visually verified both truth tables with the help of LED diodes. Then we have built a memory using a **RS flip-flop** and studied its operation. We have used **D-type flip-flops** to implement asynchronous frequency dividers (2,4,8 and 16 times), reverse counters and counters (up to 31). In the end we have studied the time delay in signal propagation through **NOT gates**, measuring the mean delay per gate.

*E-mail: simone.ziglio-1@studenti.unitn.it - riccardo.furlan@studenti.unitn.it

2 NOR

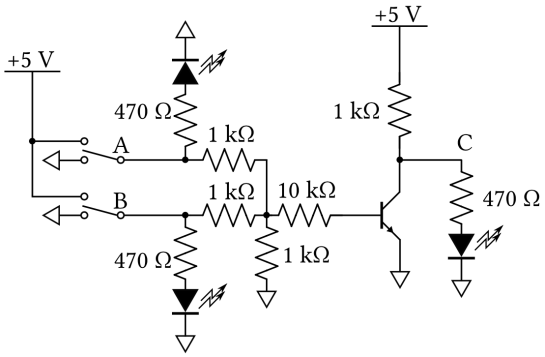


Figure 1: NOR circuit

In Figure 1 we see the NOR circuit, based on a 2N2222 transistor. A and B are the inputs of our gate, they can be 5 V (logic 1) or 0 V (logic 0). When at least one of those inputs is high, the transistor is opened and the output C is at 0 V; when both inputs are low, the transistor is cut off and the output is directly connected to 5 V.
We studied the circuit using three LEDs to visualize inputs' (green and yellow) and output's states (red). We clearly recognise the truth table of a NOR gate in the results shown in Figure 2.

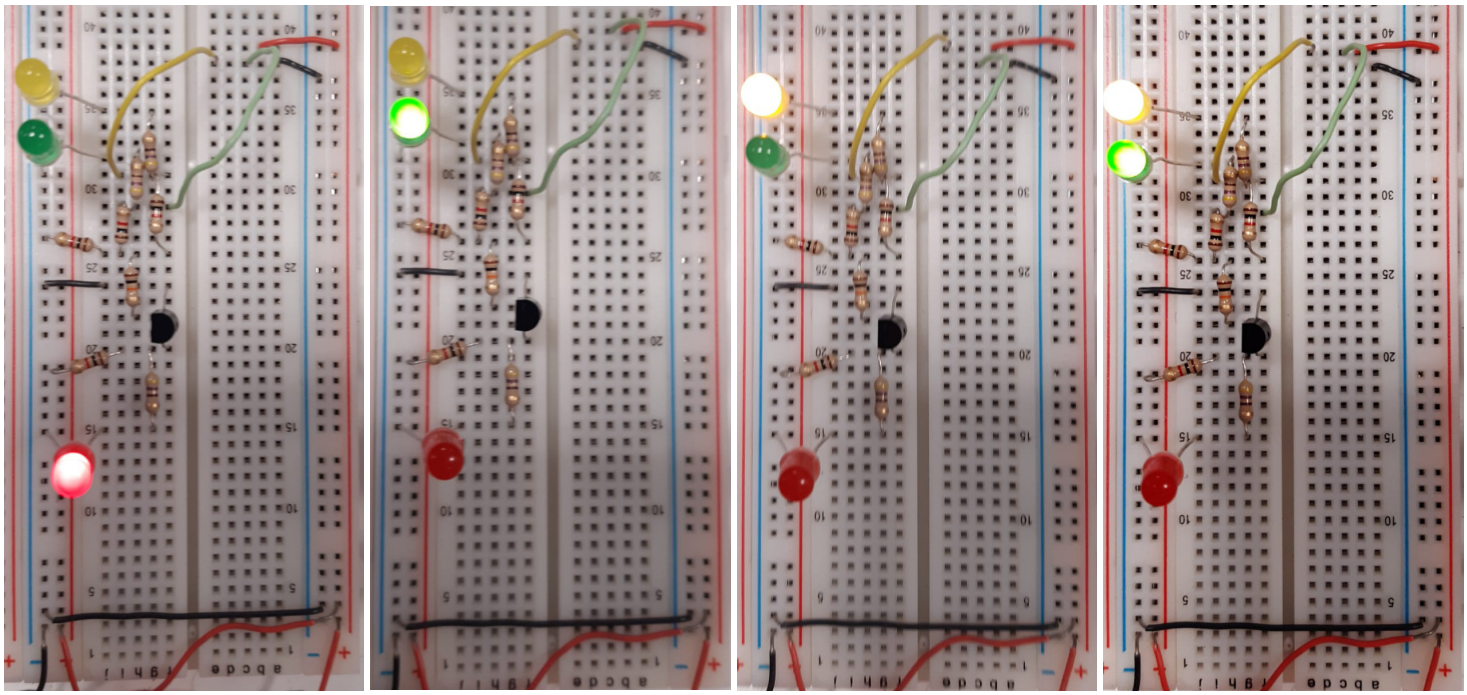


Figure 2: NOR circuit with configurations corresponding to input states AB: 00, 01, 10, 11

3 XOR

We built the XOR gate combining the 4 NAND gates contained in the IC SN74LS00 in the following manner:

$$A \oplus B = \overline{\overline{A \cdot \overline{A \cdot B}} \cdot \overline{B \cdot \overline{A \cdot B}}} \tag{1}$$

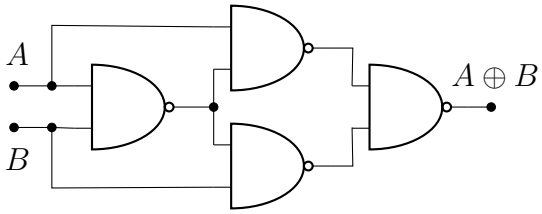


Figure 3: Flowchart of our XOR gate

We proceeded to verify this circuit's truth table using LEDs once more. The results shown in Figure 4 prove that this circuit operates as a XOR gate as expected.

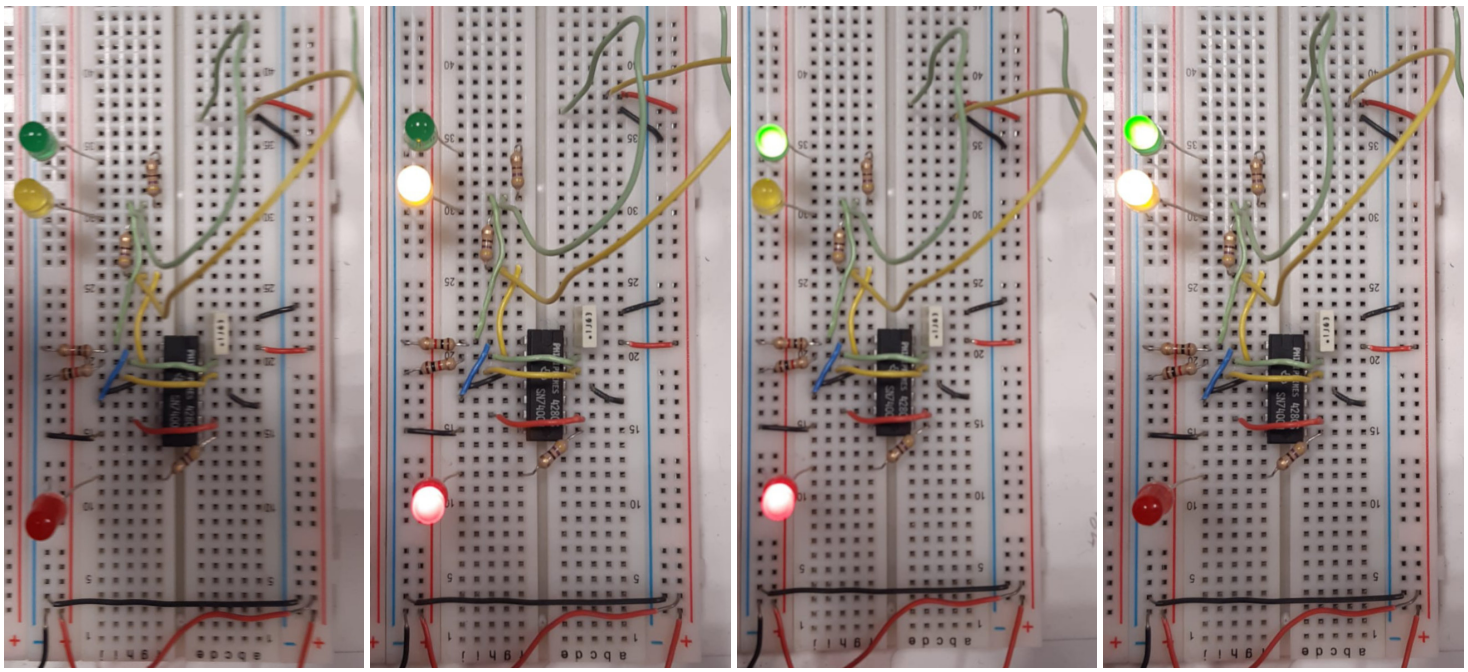


Figure 4: XOR circuit with configurations corresponding to input states AB: 00, 01, 10, 11

4 RS Flip-flop

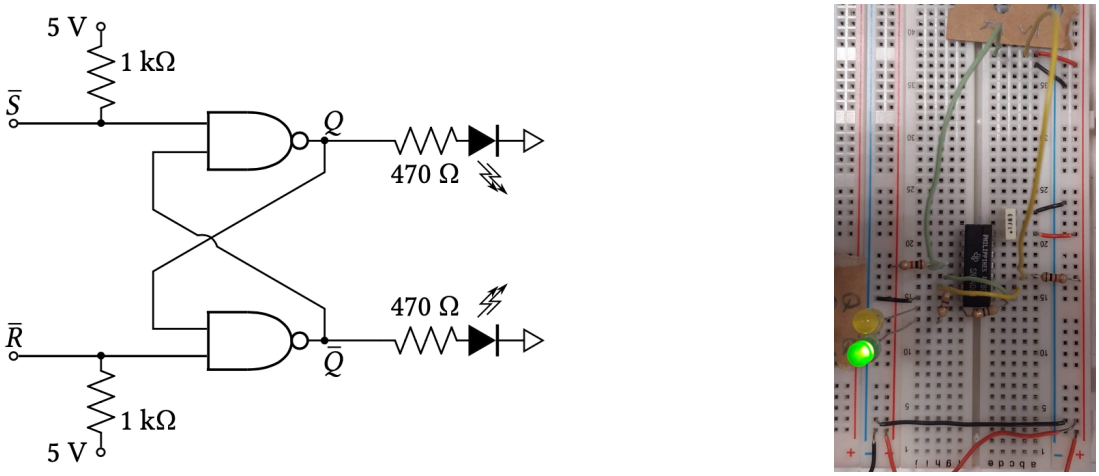


Figure 5: RS flip-flop circuit and its implementation

We used two NAND gates from the IC SN74LS00 to build the RS flip-flop shown in Figure 5. We implemented the pull-up configuration connecting our inputs $\bar{R}$ and $\bar{S}$ to 5 V.
This circuit operates like a memory: its two outputs Q and $\bar{Q}$ maintain their state indefinitely in time until we put one input to ground. After doing so we observe the so called "flipping": Q and $\bar{Q}$ switch their states.
The way we implemented this action was to briefly touch a loose wire from the ground with another one coming from $\bar{R}$ (reset) or $\bar{S}$ (set). After doing this the inputs return both to high and our circuit maintains the outputs' states that was previously set: we have built a basic memory.

If we gather together the three wires and suddenly release them we enter the realm of the forbidden state: the outputs aren't well defined because $\bar{Q}$ is no longer the negation of Q. We expect to observe both outputs high, but this is not the case. The last wire to be released, even a fraction of second later, sets the outputs and we don't enter the forbidden state.
In Table 1 we present our RS latch truth table.

$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
1	1	x	$\bar{x}$
1	0	0	1
0	1	1	0
0	0	forbidden state	

Table 1: RS flip-flop truth table, the terms x and $\bar{x}$ depict that in this state the circuit maintains the value of the previous state

5 Frequency divider and counter

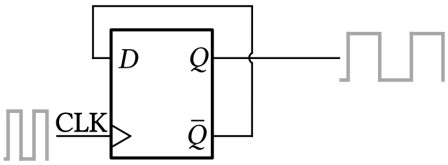


Figure 6: Divider circuit made out of a D-type flip-flops

We implemented the basic asynchronous 2 times frequency divider shown in Figure 6 using a IC SN74LS74, which contains two D-type flip-flops positive edge triggered.
We put both the $\bar{R}$ and $\bar{S}$ pins high in order to avoid set and reset; then we used a square wave $0-5\,\mathrm{V}$ at $1\,\mathrm{kHz}$ from the function generator as a clock and observed the output Q on the oscilloscope. Its frequency is $f = 500.01\,\mathrm{Hz}$, almost exactly half the clock and the relative error is several orders of magnitude lower than one.
We observed a delay between the input and the output $\Delta t \sim 27\,\mathrm{ns}$, caused by the propagation of the signal through the gates, which is responsible for the functioning of this circuit.
Using the output of each divider as the clock for the next D-type flip-flop, we implemented a 4 ($f = 250\,\mathrm{Hz}$), 8 ($f = 125\,\mathrm{Hz}$) and 16 times ($f = 62.501\,\mathrm{Hz}$) frequency divider.
We lowered the first clock frequency to $1\,\mathrm{Hz}$ and connected it together with the outputs of every frequency divider to an integrated circuit implementing LEDs. In this way we had built a reverse binary counter from 31 to 0, which counts a number per second.
Using the $\bar{Q}$ outputs from the dividers we obtained an ascending counter from 0 to 31. Then we removed the integrated circuit with the LEDs and used one with a seven-segment display showing numbers in hexadecimal.

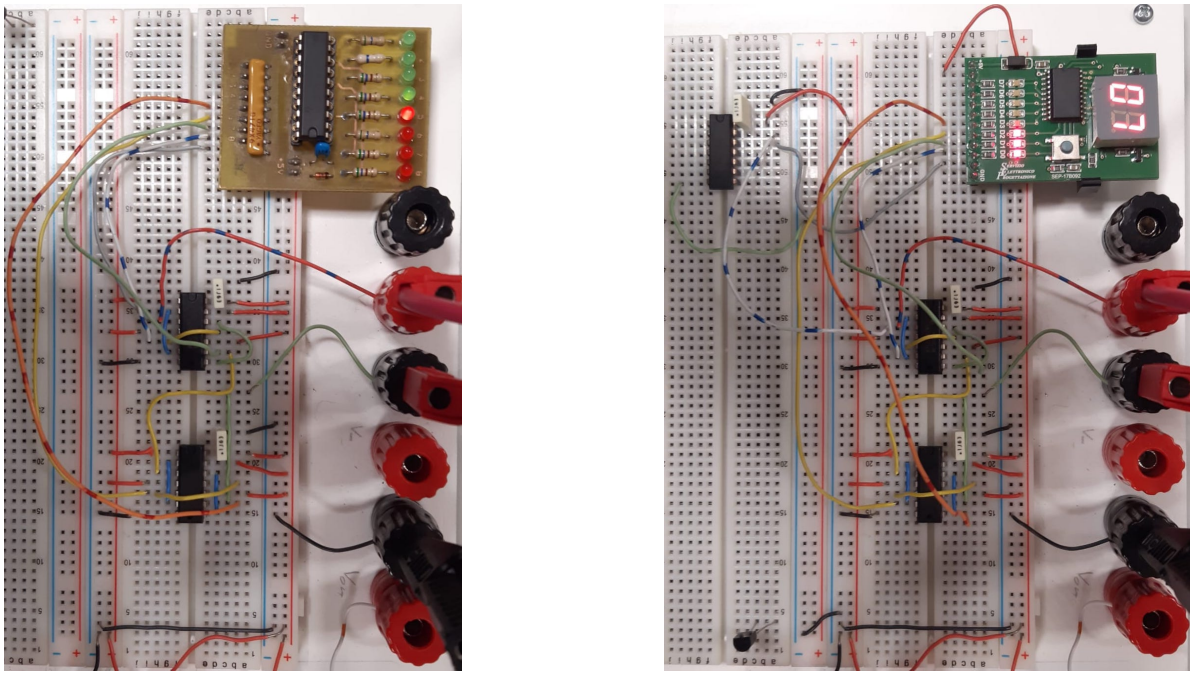


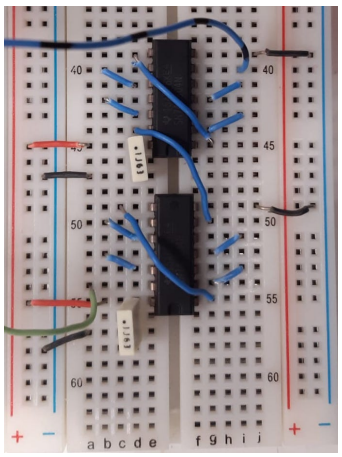
Figure 7: On the left, a descending binary counter with LEDs and on the right an ascending one with a seven-segment display

6 Propagation delay in logic gates

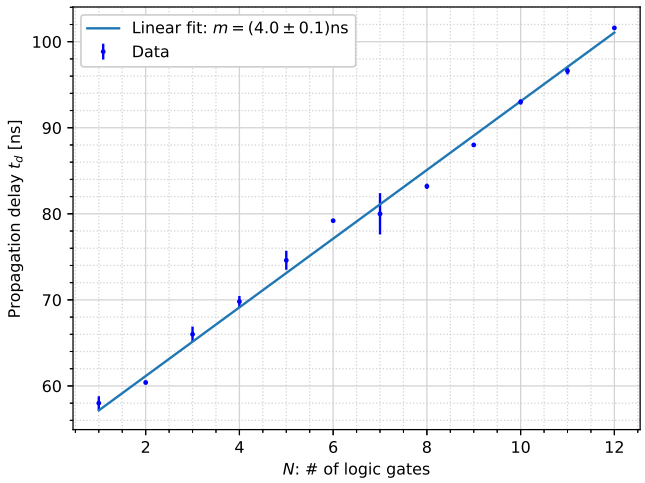
In the last part we wanted to measure the time delay in the signal propagation through logic gates. Even though delay could seem annoying, it's fundamental for the functioning of various circuits such as the frequency divider. In this case in fact, if signal propagation were instantaneous, D should be both high and low simultaneously when the clock becomes high, but a small delay is enough to let the positive edge pass and avoid problems.

We used two IC SN74LS04, both implementing six NOT gates and connected the output of each gate to the input of the next one. Then we applied a square wave $0 - 5\,\mathrm{V}$ in the first input and measured with the oscilloscope the delay between this wave and the output of every gate. We measured the period of time from when the input begins its descent from $5\,\mathrm{V}$ to $0\,\mathrm{V}$ (when we consider an odd number of gates) or its ascent from $0\,\mathrm{V}$ to $5\,\mathrm{V}$ (when we consider an even number of gates) to when the output surpasses $3.5\,\mathrm{V}$, becoming a logic 1.

Afterwards, we calculated a linear fit of the propagation time with respect to the number of gates (see Figure 8b) and estimated the mean delay per gate as the slope of the fit line $t_d = (4.0 \pm 0.1)\,\mathrm{ns}$ which is compatible with the approximate value provided in the datasheet $t_d = (3 \div 10)\,\mathrm{ns}$.
Furthermore, we saw that the line from the linear fit didn't pass through the origin, but rather had an intercept at $t_0 = (53 \pm 1)\,\mathrm{ns}$ due to the length of the BNC cable and the systematic delay of the oscilloscope.



(a) Circuit used to calculate the average propagation delay in not gates



(b) Graph of the time delay as a function of the number of gates with linear fit

7 Conclusions

We have built and verified the functioning of both a NOR and a XOR gate. We have implemented a RS flip-flop as a memory and studied its operation, verifying that we can't achieve the forbidden state. Through the usage of D-type flip-flops we have built 2 and multiples times frequency dividers, observing how good this division is. Using these dividers we have built a counter and a reverse counter. Finally we have characterized an integrated by the mean delay per gate in signal propagation through its NAND gates, finding a result compatible with the one given by the producer.