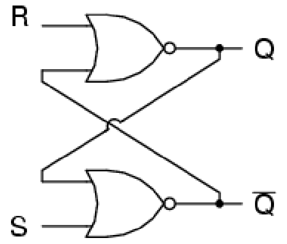
FLIP-FLOP

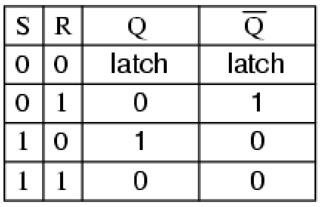
Eduardo Estrada González, Marco Baez González, Moisés Jahir González Márquez, Manuel Fernández Mercado

Introduction

Biestable circuit is a type of digital circuit capable of storing Aone bit information. This kind of circuit has two stable states, meaning it can maintain one of these states indefinitely until it receives an input signal that causes it to change. Bistable circuits are the foundation of sequential digital systems, as they allow for temporary storage of information and syncrhonization of operations.

Biestable circuits are fundamental in digital electronics because the form the basis of memory devices, such as





S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

registers and counters. A common example of a bistable circuit is the flip-flop, whic is a syncrhonous bistable circuit, meaning it requires a clock signal (CLK) to change states.

To understand how we arrive at a flip-flop, we must first discuss sequential circuits. These circuits depend not only on the current inputs but also on the previus state of the system. Unlike combinational circuits, where the output dependeds solely on the current inputs, sequential circuits have internal memory that allows them to "remember" previous states.

The flip-flop is a bistable circuit that is triggered by a clock signal (CLK). This clock signal can be generated by an oscillator, such as the LM555, whis is a very popular timer used to generate precise clock pulses.

I. NOR LATCH

A NOR Latch is another type of basic bistable circuit that uses NOR logic gates to store one bit of information. Like the NAND Latch, this circuit has two inputs and two complementary outputs, typically labeled Q and Q. The inputs are also labeled as S (Set) and R (Reset), and they control the state of the latch.

The NOR Latch operates similary to the NAND Latch, but with a difference: the activation conditions for the inputs are inverted due to the nature of the NOR gates.

Here is the behavior:

1. When S = 1 and R = 0:
   a. The latch is "set" and Q = 1 (Set state)
2. When R = 1 and S = 0
   a. The latch is "reset" and Q = 0 (Reset state)
3. When S = 0 and R = 0:
   a. The latch maintains its previous state (Qprev)
4. When S = 1 and R = 1:
   a. This condition is not allowed because it generates an incositent output (Q = Q´ = 0)

B. SR Flip-Flop with Clock Signal

An SR Flip-Flop is a synchronous bistable circuit that uses an SR Latch as its base and adds a clock signal (CLK) to control when the inputs S (Set) and R (Reset) affect the circuit's state. The clock signal introduces a level of synchronization, allowing the flip-flop to operate within more complex digital systems.

A. Operation of the SR Flip-Flop with CLK

The SR Flip-Flop with a clock signal uses two SR Latches connected in cascade to implement synchronization. This design is commonly known as the Master-Slave Flip-Flop . Here is the step-by-step operation:

1. Flip-Flop Inputs:z
   - S: Set input (activates the Q=1 state).
   - R: Reset input (activates the Q=0 state).
   - CLK: Clock signal that controls when the flip-flop updates its state.

2. Synchronization via CLK:
   - When the clock signal (*CLK*) is active (e.g., high), the Master Latch captures the *S* and *R* inputs, but the Slave Latch remains locked.
   - When the clock signal changes state (e.g., goes low), the Slave Latch transfers the captured state from the Master Latch to the output.

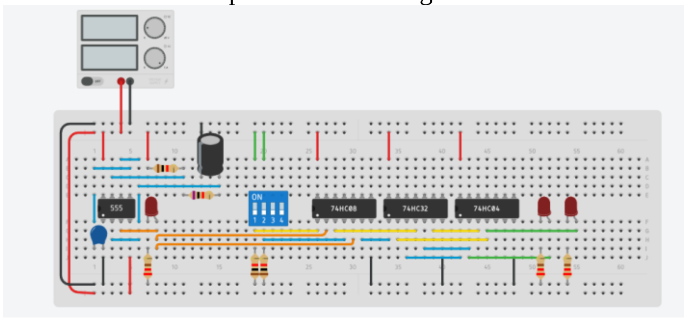
CLK	S	R	Q	Q(t+1)
0	X	X	0	0
0	X	X	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X

*

1	1	1	1	X

When CLK = 0: The flip-flop maintains its previous state (Qn).
CLK = 1: The flip-flop updates its states based on the S and R inputs.

We can watch the implementation using Tinkercad:



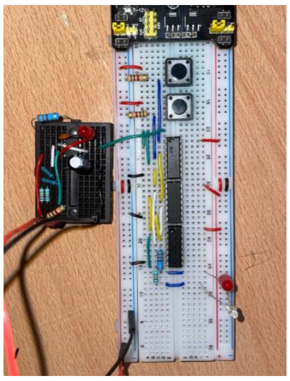
Bill of materials:
1. Timer
2. 100 nF Capactiro
3. 1 kΩ Resistor
4. 7 kΩ Resistor
5. 100 uF, 16 V Polarized Capacitor
6. 220 Ω Resistor
7. Red LED
8. 5,5 Power Supply
9. DIP Switch SPST x 4
10. Quad AND gate
11. Quad OR gate
12. Hex inverter

Curiosities:
- Noise in the oscillator Signal LM555:
  - Although the LM555 is designed to generate a clean signal, in practice you might observe small fluctuations or noise in the output due to external interference or the quality of the components. This cane cause unexpected transitions in the latch.
- Propagation Delay in Transitions:
  - Logic gates have a small propagation delay (in nanoseconds). This can cause the outputs Q and Q´ not change instantly, potentially leading to unexpected intermediate states during transitions.
- Inaccurate Oscillator Frequency:
  - Although the LM555 is configured to generated a 1 Hz signal, the actual frequency might vary slightly due tolerances in the resistor and capacitor values. This can cause the oscillator to run faster or slower than expected.

Finally, we will show the physical implementation made on a protoboard,
based on the previous simulation.

Conclusion

The design and implementation of an S-R latch controlled by a signal generated with an LM555 not only reinforces the theoretical concepts of sequential circuits but also provides insight into how these components interact in real-world applications. This type of practice is essential for learning to work with clock signals, state storage, and



synchronization—key concepts in digital systems such as microprocessors, memory devices, and programmable logic controllers (PLCs). Additionally, observing unexpected phenomena during the physical implementation helps develop critical skills for diagnosing and solving problems in digital circuits. In summary, this exercise not only strengthens theoretical knowledge but also prepares students to tackle practical challenges in electronic system design.

References:
Jiménez, R. (s.f.). *Biestables* . Recuperado el [fecha de acceso], de https://www.uhu.es/raul.jimenez/MICROELECTRONICA/biestables.pdf
Build Electronic Circuits. (s.f.). *S-R Latch* . Recuperado el [fecha de acceso], de https://www.build-electronic-circuits.com/s-r-latch/

[1]
[2] Universidad de la República, FING. (s.f.). *Notas sobre Arquitectura de Computadoras: Latches y Flip-Flops* . Recuperado el [fecha de acceso], de https://www.fing.edu.uy/tecnoinf/paysandu/cursos/1er/arq/material/Notas/7.pdf