



Universidad Nacional Autónoma de México
Facultad de Ingeniería
Diseño Digital VLSI



Examen Parcial

Integrantes:

- Aguilar Luna Gabriel Daniel
- Jiménez Juárez Jesús
- Torres Galván José Antonio

Grupo teoría: 4

Profesor: Dr. Roberto Giovanni Ramirez Chavarria

Fecha de entrega: Septiembre 12, 2019

Semestre 2020-1

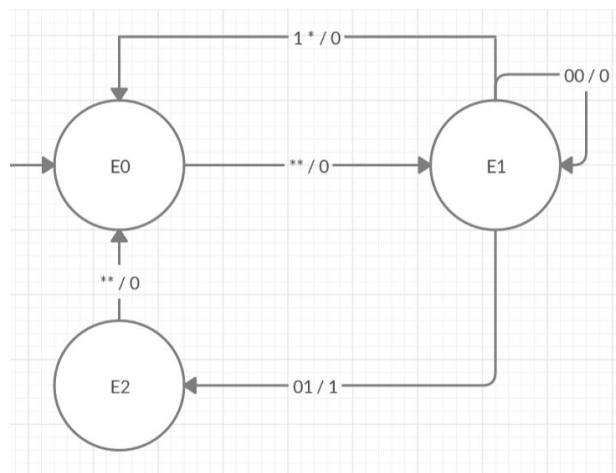
Introducción

Las máquinas de estado finito, más conocidas por su acrónimo en inglés FSM (Finite State Machine), se utilizan ampliamente en el diseño de circuitos digitales (además de en otros ámbitos de la ingeniería, como la programación), para describir el comportamiento de un sistema según el valor de sus entradas y de cómo van cambiando en el tiempo.

Es decir que vamos a tener un conjunto de estados, unas entradas y unas salidas. Según la entrada que tengamos iremos recorriendo los estados y asignando las salidas, dependiendo estas de la secuencia de entradas que haya habido en el sistema.

Planteamiento del problema

1. Diagrama de estados



2. Tablas de transiciones

Estado Presente		Entradas		Salida	Estado Siguiente	
Q1	Q0	s	t	p	Q1+	Q0+
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	1	1	1
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

3. Funciones lógicas

a. $Q_1^+ = P = \neg Q_1 Q_0 \neg s \ t$

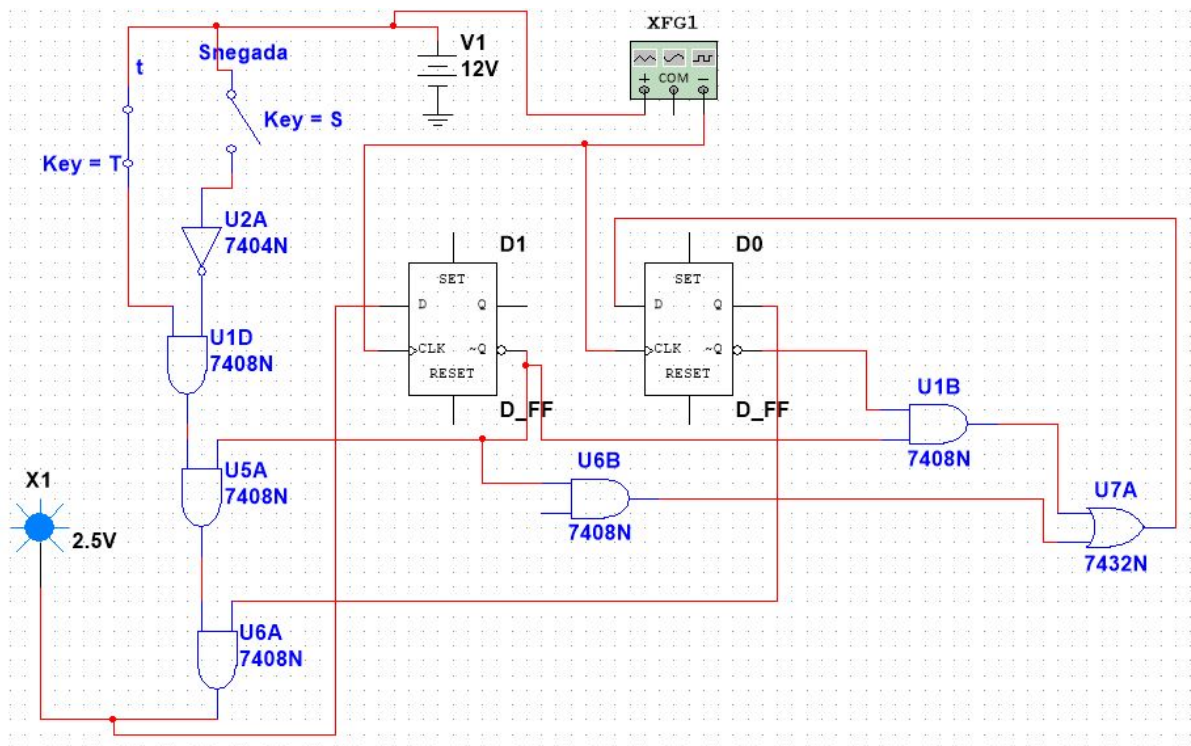
$\begin{matrix} Q_1 \\ s \\ t \end{matrix} \backslash Q_0$	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	0	0	0
10	0	0	0	0

b. $Q_0^+ = \neg Q_1 (Q_0 + \neg s)$

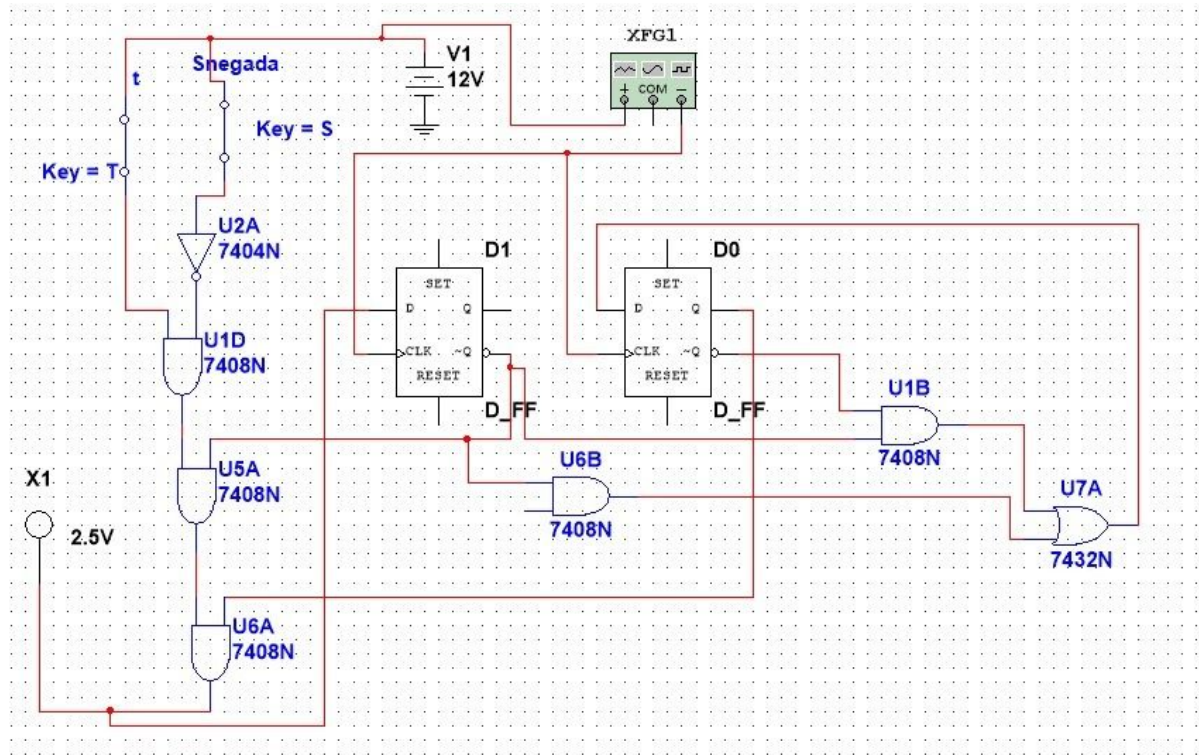
$\begin{matrix} Q_1 \\ s \\ t \end{matrix} \backslash Q_0$	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	0	0	0
10	1	0	0	0

4. Circuito con biestables (Simulación)

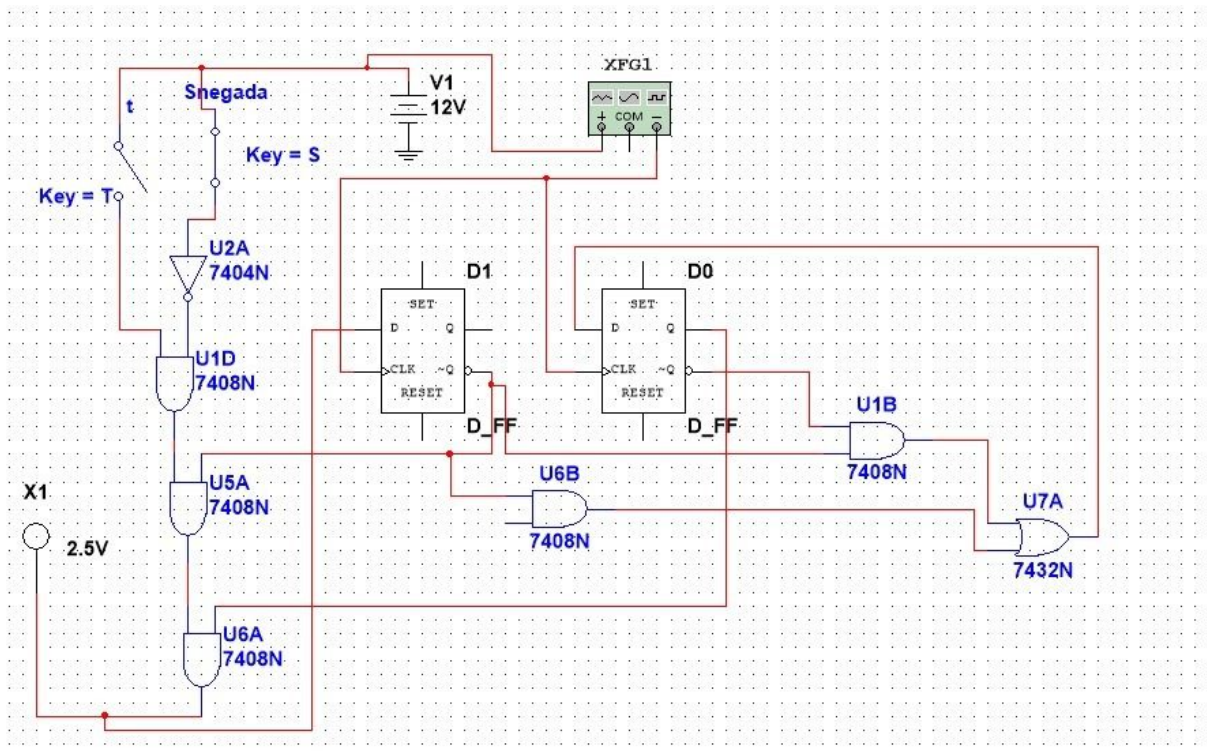
a. Cuando $t=1$ y $s=0$



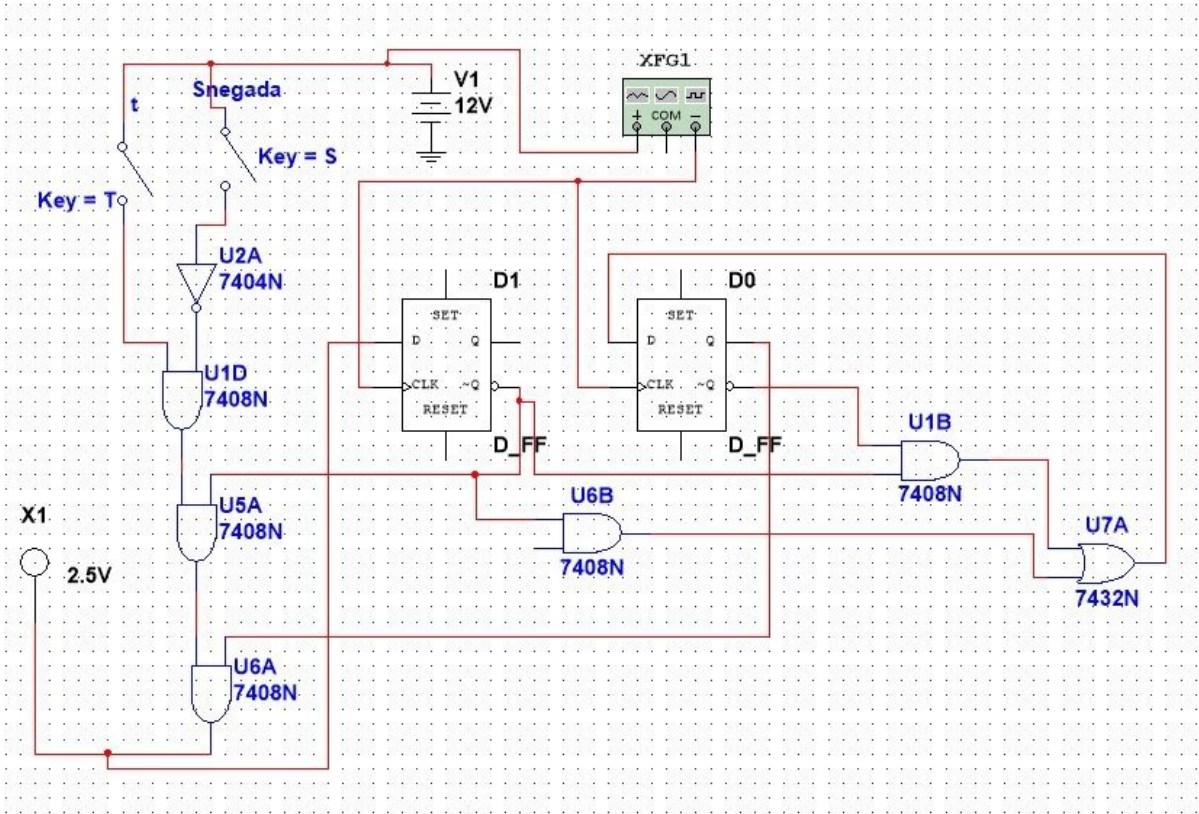
b. Cuando $t=1$ y $s=1$



c. Cuando $t=0$ y $s=1$



d. Cuando $t=0$ y $s=0$



5. Código VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity pulsos is
    Port ( MCLK : in STD_LOGIC;
           t : in STD_LOGIC;
           s : in STD_LOGIC;
           p : out STD_LOGIC);
end pulsos;
```

```
architecture Behavioral of pulsos is
    type estados is (E0,E1,E3);
    signal edo_pres,edo_sig : estados;
begin
    --Process asignar estados
    process(MCLK)
    begin
        if(rising_edge(MCLK))then
            edo_pres <= edo_sig;
        end if;
    end process;
end Behavioral;
```

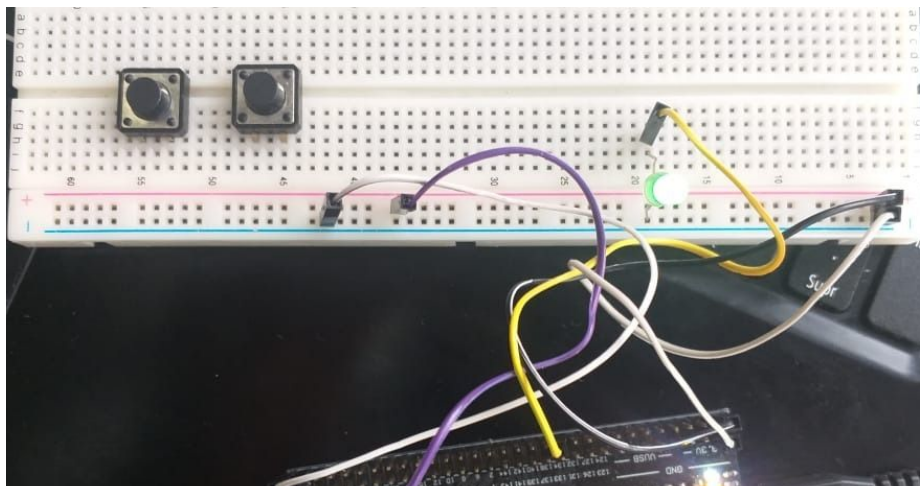
```

end process;
--Process manejo entradas y salidas
process(edo_pres,s,t)
begin
    case edo_pres is
        when E0 =>
            p <='0';
            edo_sig<=E1;
        when E1 =>
            if(s='0') then
                if(t='0') then
                    p<='0';
                    edo_sig<=E1;
                else
                    p<='1';
                    edo_sig<=E3;
                end if;
            else
                p<='0';
                edo_sig<=E0;
            end if;
        when E3 =>
            p <='0';
            edo_sig<=E0;
        end case;
    end process;
end Behavioral;

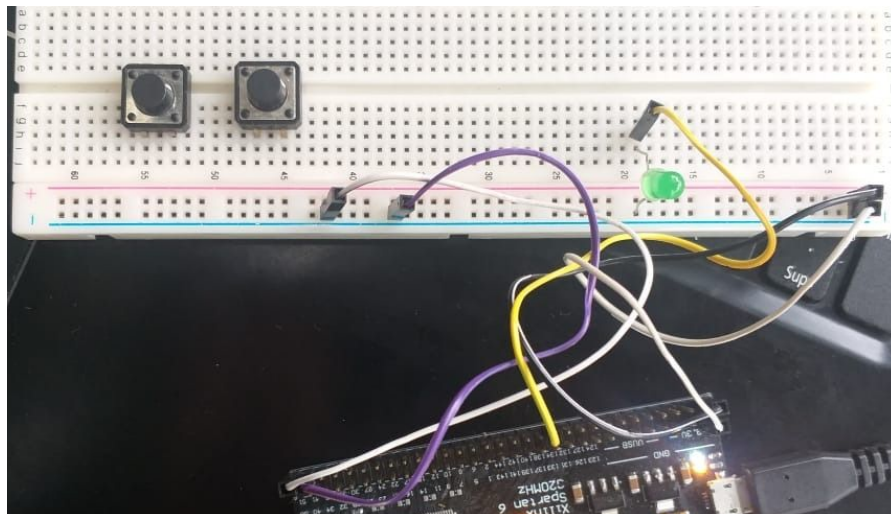
```

6. Funcionamiento en FPGA

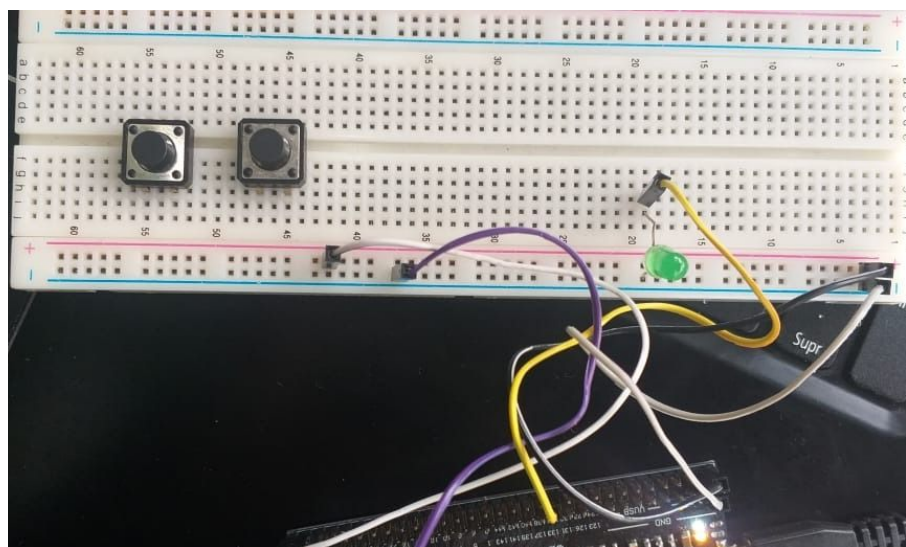
Cuando $t=1$ y $s=0$:



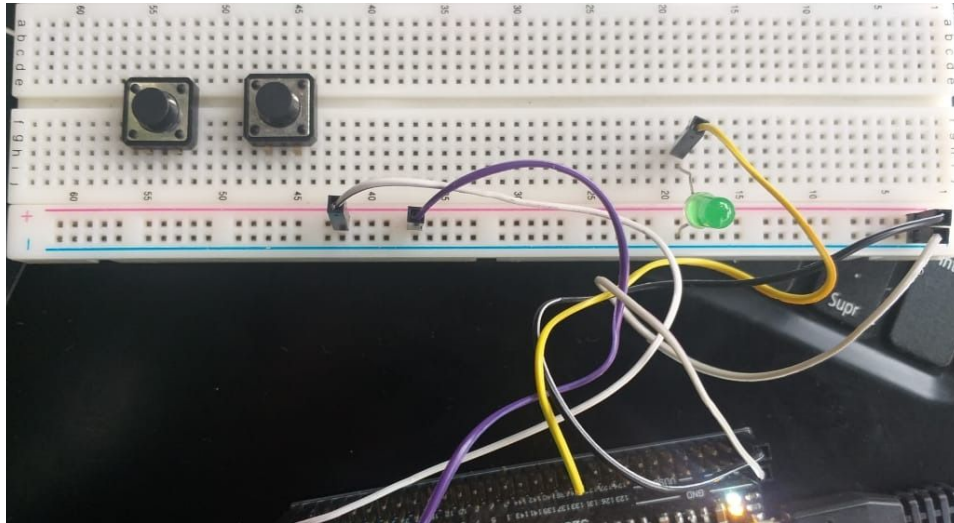
Cuando $t=0$ y $s=0$



Cuando $t=0$ y $s=1$



Cuando $t=1$ y $s=1$



Referencias bibliográficas

- Anónimo (2019). Máquinas de estados en VHDL. *VHDL.es*. Recuperado el 12 de septiembre del 2019 del sitio <https://vhdl.es/maquinas-de-estado-en-vhdl/>
- Garcia, A. (2019). Máquinas de estados finitos en VHDL. *Digilogic*. Recuperado el 12 de septiembre del 2019 del sitio <https://www.digilogic.es/maquinas-de-estado-finito-fsm-vhdl/>