



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

PRÁCTICA 2: REGISTROS

DISEÑO DE SISTEMAS DIGITALES

ALUMNO:

RAMÍREZ COTONIETO LUIS FERNANDO

PROFESOR:

TESTA NAVA ALEXIS

Fecha: 12 de Abril de 2021

Figura 2: Datasheet de la GAL22V10

Se encontró que su equivalente en el simulador PROTEUS es la AM22V10, por lo que se empezó a trabajar en el diseño partiendo de las entradas:

clk = 1	24 * not used
es = 2	23 * not used
d(6) = 3	22 * not used
d(5) = 4	21 * not used
d(4) = 5	20 = q(6)
d(3) = 6	19 = q(5)
d(2) = 7	18 = q(4)
d(1) = 8	17 = q(3)
d(0) = 9	16 = q(2)
op(0) = 10	15 = q(1)
op(1) = 11	14 = q(0)
not used * 12	13 = clr

Figura 3: Asignación de pines de la GAL22V10

El circuito desarrollado en PROTEUS de primera vista se vió de la siguiente manera:

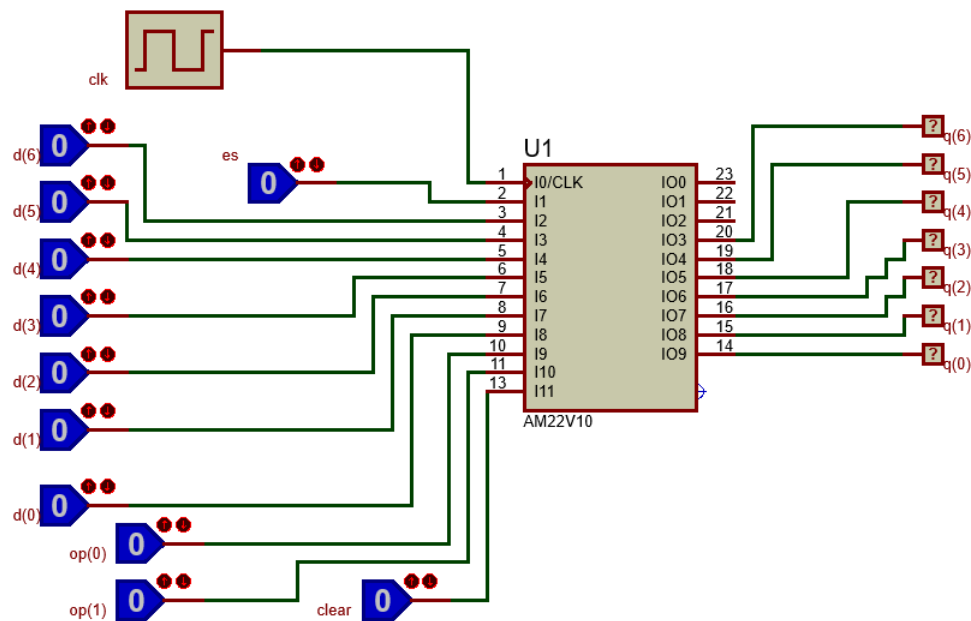


Figura 4: Simulación del circuito

Para este punto lo que procedía era el desarrollo de un código en VHDL que nos permitiera controlar el registro creado para poder visualizar su funcionamiento.

1.1. Código

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 --ENTIDAD
5 entity Prac2 is port(
6     clr, clk, ES: in std_logic;
7     D: in std_logic_vector(6 downto 0);
8     --ENTRADAS
9     Q: inout std_logic_vector(6 downto 0);
10    --ENTRADAS Y SALIDAS
11    OP: in std_logic_vector(1 downto 0)
12    --OPERADOR DEL MUX

```

```

13 );
14 --ASIGNACION DE PINES
15 attribute pin_numbers of Prac2: entity is
16     "Q(0):14 Q(1):15 Q(2):16 Q(3):17 Q(4):18 Q(5):19 Q(6):20 "
17     & "D(0):9 D(1):8 D(2):7 D(3):6 D(4):5 D(5):4 D(6):3 "
18     & "OP(0):10 OP(1):11 "
19     & "clk:1 ES:2 clr:13";
20 end Prac2;
21
22
23 --ARQUITECTURA
24 architecture aPrac2 of Prac2 is
25     signal aux: std_logic_vector(6 downto 0);
26     --signal SAL: std_logic;
27 begin
28     process(clr,clk,Q,OP,D,ES)
29     begin
30         if(clr= '1') then
31             Q <= "0000000";
32         elsif(rising_edge(clk)) then
33             Q <= aux;
34         end if;
35
36         --MUX
37         case (OP) is
38             when "00" => --carga
39                 for i in 0 to 6 loop
40                     aux(i)<=d(i);
41                 end loop;
42             when "01" => --retencion
43                 for i in 0 to 6 loop
44                     aux(i)<=Q(i);
45                 end loop;
46             when "10" => --corrimiento a la izquierda
47                 for i in 0 to 6 loop
48                     if(i>0)then
49                         aux(i)<=Q(i-1);
50                     else
51                         aux(i)<= ES;
52                     end if;
53                 end loop;
54             when others => --corrimiento a la derecha
55                 for i in 0 to 6 loop
56                     if(i<6)then
57                         aux(i)<=Q(i+1);
58                     else
59                         aux(i)<= ES;
60                     end if;
61                 end loop;
62             end case;
63         end process;
64     end aPrac2;

```

De esto obtenemos los siguientes estados en el diagrama:

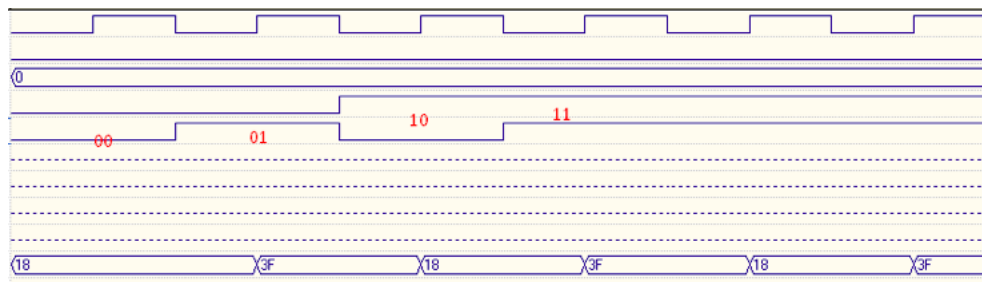


Figura 5: Cuando el pin 7 y 8 se encuentran en alto (1)

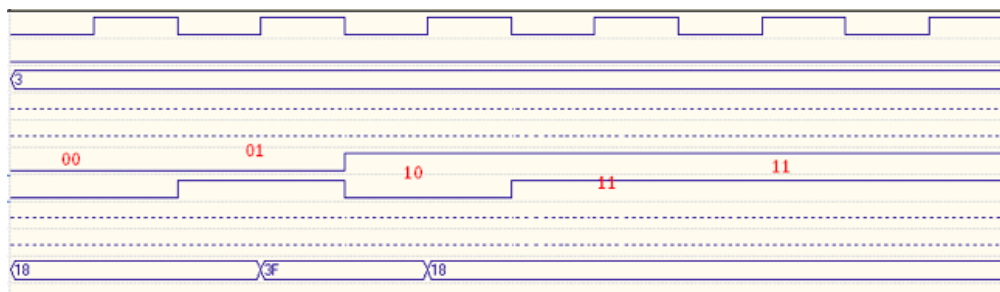


Figura 6: Cuando el pin 3 y 4 se encuentran en alto (1)

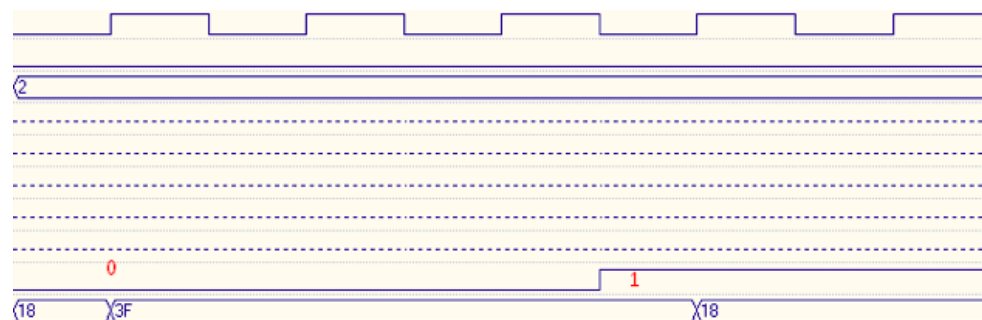


Figura 7: Cuando el pin 9 se encuentran en alto (1)

2. Conclusión

Como nos mostraba el diagrama de la práctica, y la teoría de los registros, estos no son más que una serie de Flip-Flops conectados para que puedan ser almacenadas pequeñas secciones de memoria, pero al ser programados de esta manera, asumimos un ahorro invreible tanto de espacio como seguramente de costo al momento de su fabricación. Para la práctica se utilizó la GAL22V10, que contiene muy poco pines a mi parecer, pues el uso de ellos estuvo casi ocupado en su totalidad por los requerimientos de entrada y salida que se tenían; podemos encontrar una señal síncrona en la mayoría de las entradas-sakidas, pues son dependientes de la señal de reloj asignada en el pin 1, excepto la señal del clear pues este lleva otro proceso. Poco a poco me empiezo a familiarizar de una mejor manera con el lenguaje, y voy comprendiendo que con el avance de cada práctica se van tomando elementos que nos sirvan más a futuro como el funcionamiento de memorias, desplazamientos y otros tipos de procesos de circuitos más avanzados y convencionales en nuestro día a día.

3. Bibliografía

- Stojanovic, V., Oklobdzija, V. G. (1999). Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. IEEE Journal of solid-state circuits, 34(4), 536-548.
- Perry, D. L. (2002). VHDL: programming by example. McGraw-Hill Education.
- Ashenden, P. J. (2010). The designer's guide to VHDL
- Morgan Kaufmann. Lipsett, R., Schaefer, C. F., Ussery, C. (2012). VHDL: Hardware description and design. Springer Science Business Media.