



INSTITUTO POLITÉCNICO
NACIONAL
ESCUELA SUPERIOR DE
COMPUTO



DISEÑO DE SISTEMAS DIGITALES

PRÁCTICA 1: FLIP FLOPS

PROFESOR:

Testa Nava Alexis

ALUMNO:

Ramírez Cotonieto Luis Fernando

GRUPO:

2CV18

GRUPO:

25/Marzo/2021

DESARROLLO EXPERIMENTAL

La práctica consistió en la programación de un multiplexor en el lenguaje VHDL que contuviera los flip-flops vistos anteriormente en clase (SR,T,D y JK), para este programa se utilizó en simulador de ALTERA ModelSim pues se encontraba ya instalado en la computadora donde se realizó la página, la compilación del código fue correcta, a continuación se puede observar el código utilizado para la compilación y su resultado en el programa.

```
--Practica 1 DSD ... Ramirez Cotonieto Luis Fernando
```

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
entity FlipFlop is port (
```

```
    --Entradas de FlipFlops
```

```
    D, T, S, R, J, K, clock, clear : in std_logic;
```

```
    --Selector del Multiplexor
```

```
    selector : in std_logic_vector ( 1 downto 0 );
```

```
    --Salidas
```

```
    display : out std_logic_vector ( 5 downto 0 )
```

```
);
```

```
end FlipFlop;
```

```
architecture a_FlipFlop of FlipFlop is
```

```
    signal qD, qT, qSR, qJK : std_logic;
```

```
    signal MUX : std_logic;
```

```
begin
```

```
    --FlipFlop D
```

```
    process ( clock, clear )
```

```
    begin
```

```
        if ( clear = '1' ) then
```

```

    qD <= '0';

--Clock
    elsif ( clock'event and clock = '1' ) then
        qD <= D;
    end if;
end process;

--FlipFlop T
process ( clock, clear )
begin
    if ( clear = '1' ) then
        qT <= '0';
    elsif ( clock'event and clock = '1' ) then
        qT <= T xor qT;
    end if;
end process;

--FlipFlop SR
process ( clock, clear )
begin
    if ( clear = '1' ) then
        qSR <= '0';
    elsif ( clock'event and clock = '1' ) then
        qSR <= ( ( not R ) and qSR ) or S;
    end if;
end process;

--FlipFlop JK
process ( clock, clear )
begin
    if ( clear = '1' ) then
        qJK <= '0';
    elsif ( clock'event and clock = '1' ) then
        qJK <= ( J and ( not qJK ) ) or ( ( not K ) and qJK );
    end if;
end process;

--Multiplexor
process ( selector, qD, qT, qSR, qJK )
begin
    if ( selector = "00" ) then

```

```

MUX <= qJK;

elsif ( selector = "01" ) then

    MUX <= qT;

elsif ( selector = "10" ) then

    MUX <= qD;

else

    MUX <= qSR;

end if;

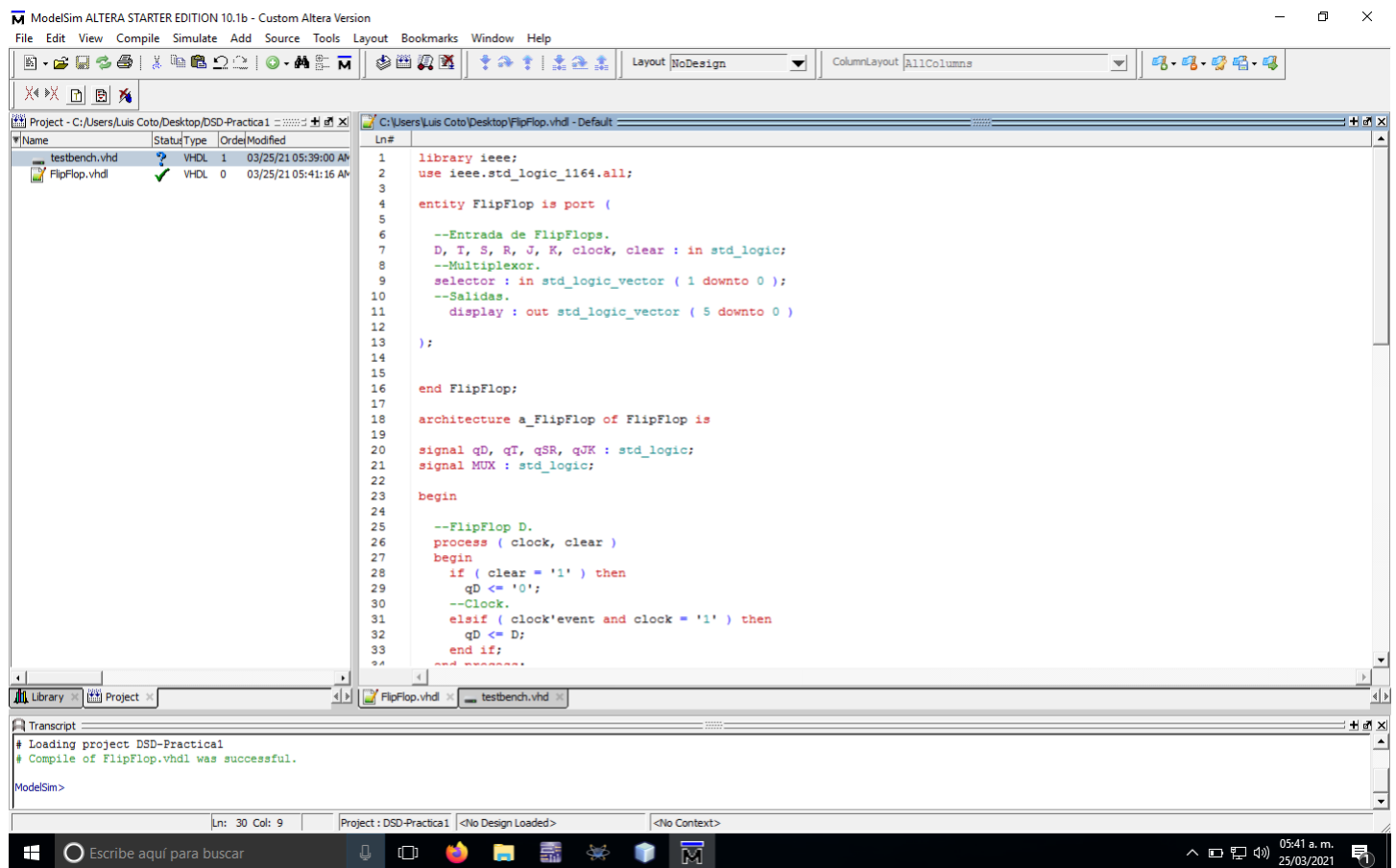
end process;

display <= "000000" when ( MUX = '0' ) else

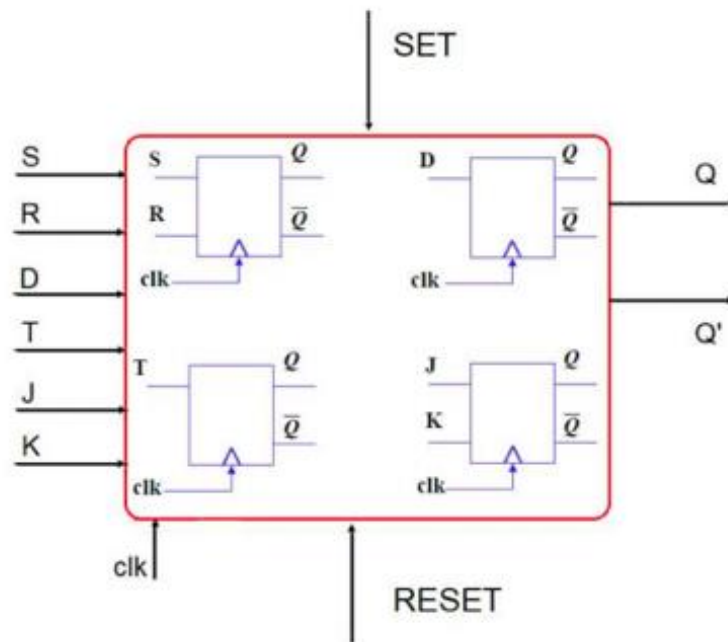
    "100111";

end architecture;

```

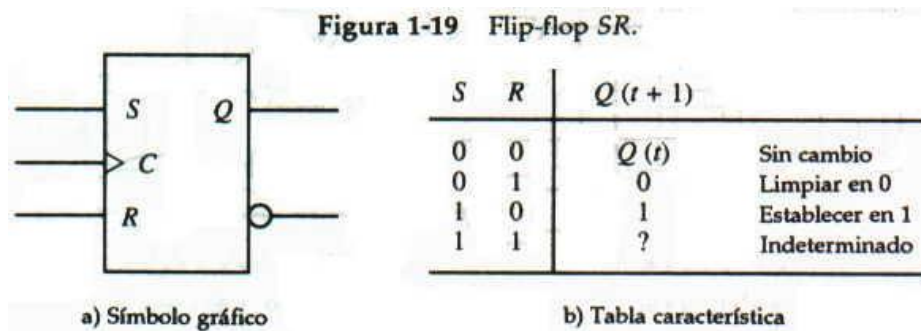


Como se observa, el programa compiló sin ningún problema. Este tiene la siguiente forma:

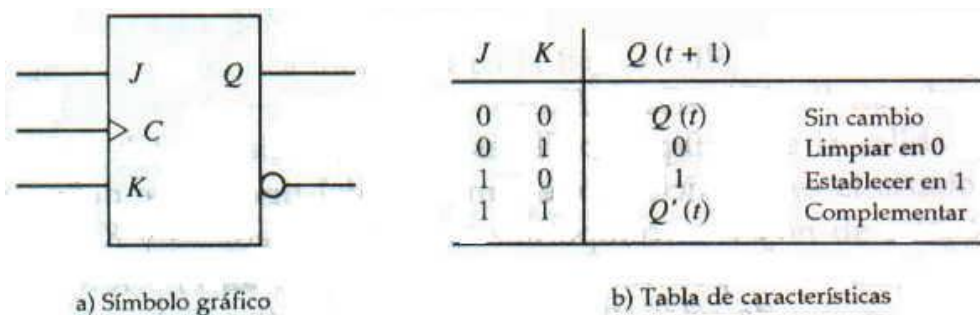


Donde cada señal lógica corresponde a las siguientes tablas de verdad:

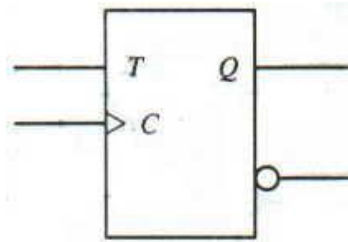
Flip-Flop S / R



Flip-Flop JK



Flip-Flop T



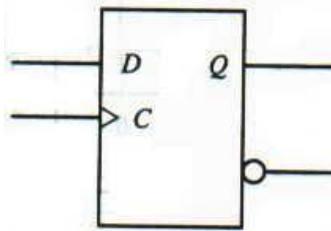
a) Símbolo gráfico

T	$Q(t+1)$	
0	$Q(t)$	Sin cambio
1	$Q'(t)$	Complementar

b) Tabla de características

Flip-Flop D

Figura 1-20 Flip-flop D.



a) Símbolo gráfico

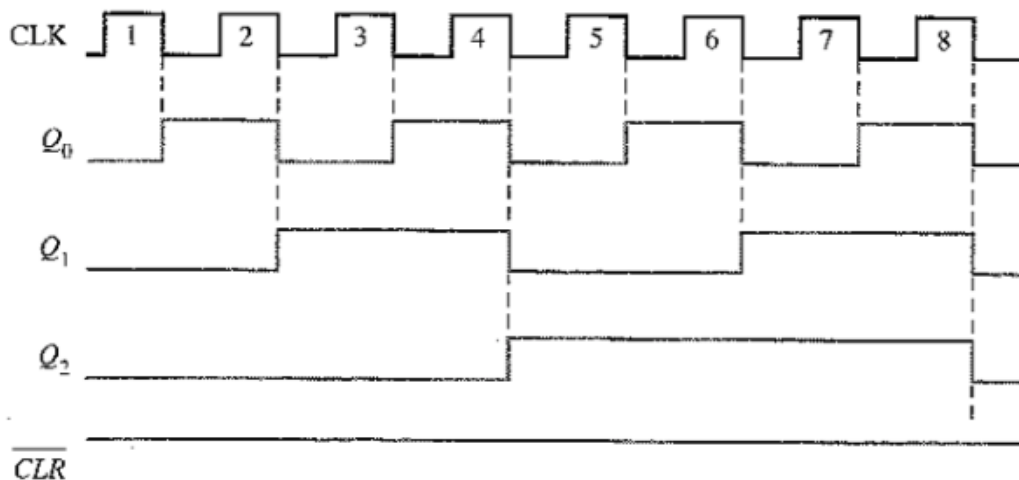
D	$Q(t+1)$	
0	0	Limpiar a 0
1	1	Establecer en 1

b) Tabla característica

La lógica del multiplexor correspondía a:

Entrada de selección de datos		Entrada seleccionada
S1	S2	
0	0	D0
0	1	D1
1	0	D2
1	1	D3

Y su diagrama de la siguiente manera:



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.

Conclusiones y observaciones:

Los multiplexores son circuitos combinacionales con varias entradas y una única salida de datos. Están dotados de entradas de control capaces de seleccionar una, y solo una, de las entradas de datos para permitir su transmisión desde la entrada seleccionada hacia dicha salida.

En el campo de la electrónica el multiplexor se utiliza como dispositivo que puede recibir varias entradas y transmitir las por un medio de transmisión compartido. Para ello lo que hace es dividir el medio de transmisión en múltiples canales, para que varios nodos puedan comunicarse al mismo tiempo.

Los flip=flops son n circuitos multivibradores, que tiene dos estados estables y puede almacenar información. Se puede hacer que cambie de estado mediante señales aplicadas a una o más entradas de control y tiene una o dos salidas. Es el elemento de almacenamiento básico en lógica secuencial. Los circuitos biestables son componentes fundamentales de los sistemas electrónicos digitales como las memorias de las computadoras, dispositivos de comunicación digital y muchos otros tipos de sistemas.

Al no haber ocupado este lenguaje en mucho tiempo, fue algo complicada la realización de la práctica, pero interesante a su vez pues el uso de multiplexores es un elemento muy útil en nuestras vidas cotidianas, con estos como se especifica arriba podemos seleccionar alguna de varias señales para poder darles un uso adecuado a nuestra necesidad, en este caso, un selector de Flip-Flops. Los flip-flops son compuertas que en un circuito podría llevarnos gran

tiempo en desarrollar, por lo que implicaría un gran espacio para su implementación y diseño, al sintetizarlo todo y poder hacer uso de algún dispositivo FPGA, simplifica enormemente el costo (a la larga) y el tamaño del circuito. Como algo extra por comentar, la lógica de las compuertas me resulta intrigante, y al implementarlas en un código, se puede ir entendiendo de una mejor manera cómo funcionan y para qué sirven.