



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**  
**ACADEMIA DE SISTEMAS DIGITALES**



## **DISEÑO DE SISTEMAS DIGITALES**

### **Práctica de Laboratorio No. 1**

#### **“Flip-Flops”**

**Profesores:**

**VICTOR HUGO GARCIA ORTEGA**  
**JULIO CESAR SOSA SAVEDRA**

**OBJETIVO:** Realizar un programa que implemente el funcionamiento de los Flip-Flops JK, SR, T y D mediante un lenguaje de descripción de hardware (HDL) en un PLD 22V10.

### MATERIAL Y EQUIPO:

Mesa de instrumentación del laboratorio de sistemas digitales

1 PLD 22v10

Además de lo anterior, se puede optar por alguna de estas dos opciones:

1 Fuente de 5V

1 Generador de funciones

1 Interruptor de presión

1 DIP switch de 8

9 Resistencias de 1K $\Omega$

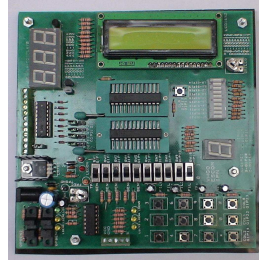
7 Resistencias de 330 $\Omega$

1 Display de Ánodo común

1 Protoboard

Pinzas y cable para alambrar

1 TEDDi (Tarjeta Educativa para Diseño Digital).



## INTRODUCCIÓN TEÓRICA

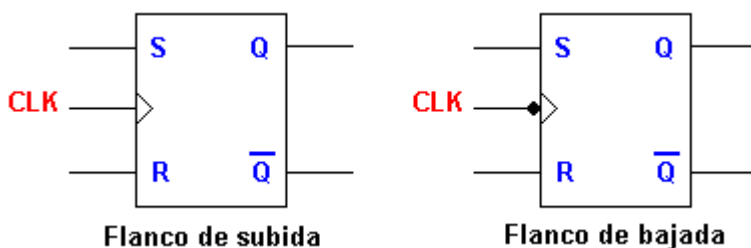
Un Flip – Flop (FF) es un dispositivo binario capaz de almacenar un bit de información, cuyo estado se determina en cada flanco de la señal de control. A esa señal de control se le llama “señal de reloj”. A esa señal de reloj generalmente se le denota como CLK, CK o CP [1]. Existen 4 tipos de Flip – Flops, los cuales se mencionan a continuación.

### 1. Flip - Flop-SR.

Su funcionamiento se define por la siguiente tabla de verdad:

S	R	CLK	Q(t+1)
0	0	$\uparrow$ ó $\downarrow$	$Q(t)$
0	1	$\uparrow$ ó $\downarrow$	0
1	0	$\uparrow$ ó $\downarrow$	1
1	1	$\uparrow$ ó $\downarrow$	X

Su símbolo es:



Si consideramos como una entrada el estado de salida actual del FF, obtenemos la siguiente tabla de verdad extendida:

S	R	Q(t)	CLK	Q(t+1)
0	0	0	$\uparrow$ ó $\downarrow$	0
0	0	1	$\uparrow$ ó $\downarrow$	1
0	1	0	$\uparrow$ ó $\downarrow$	0
0	1	1	$\uparrow$ ó $\downarrow$	0
1	0	0	$\uparrow$ ó $\downarrow$	1
1	0	1	$\uparrow$ ó $\downarrow$	1
1	1	0	$\uparrow$ ó $\downarrow$	X
1	1	1	$\uparrow$ ó $\downarrow$	X

Obtener la ecuación para  $Q(t+1)$  usando la tabla de verdad extendida mediante un mapa de Karnaugh:

	00	01	11	10
0				
1				

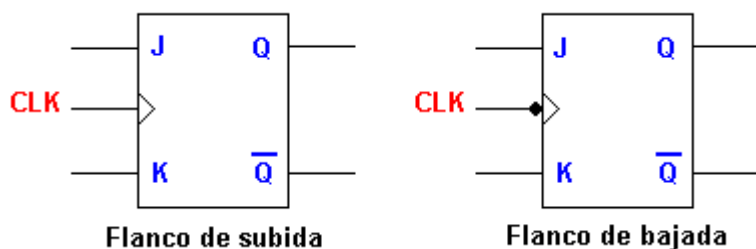
$$Q(t+1) =$$

## 2. Flip - Flop-JK.

Su funcionamiento se define por la siguiente tabla de verdad:

J	K	CLK	$Q(t+1)$
0	0	$\uparrow$ ó $\downarrow$	$Q(t)$
0	1	$\uparrow$ ó $\downarrow$	0
1	0	$\uparrow$ ó $\downarrow$	1
1	1	$\uparrow$ ó $\downarrow$	$\overline{Q(t)}$

Su símbolo es:



Si consideramos como una entrada el estado de salida actual del FF, obtenemos la siguiente tabla de verdad extendida:

J	K	$Q(t)$	CLK	$Q(t+1)$
0	0	0	$\uparrow$ ó $\downarrow$	0
0	0	1	$\uparrow$ ó $\downarrow$	1
0	1	0	$\uparrow$ ó $\downarrow$	0
0	1	1	$\uparrow$ ó $\downarrow$	0
1	0	0	$\uparrow$ ó $\downarrow$	1
1	0	1	$\uparrow$ ó $\downarrow$	1
1	1	0	$\uparrow$ ó $\downarrow$	1
1	1	1	$\uparrow$ ó $\downarrow$	0

Obtener la ecuación para  $Q(t+1)$  usando la tabla de verdad extendida mediante un mapa de Karnaugh:

	00	01	11	10
0				
1				

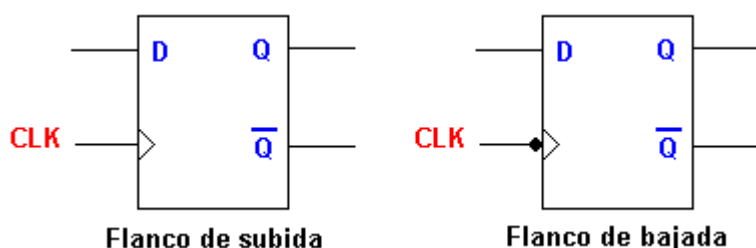
$$Q(t+1) =$$

### 3. Flip - Flop-D.

Su funcionamiento se define por la siguiente tabla de verdad:

D	CLK	Q(t+1)
0	↑ ó ↓	0
1	↑ ó ↓	1

Su símbolo es:



Si consideramos como una entrada el estado de salida actual del FF, obtenemos la siguiente tabla de verdad extendida:

D	Q(t)	CLK	Q(t+1)
0	0	↑ ó ↓	0
0	1	↑ ó ↓	0
1	0	↑ ó ↓	1
1	1	↑ ó ↓	1

Obtener la ecuación para  $Q(t+1)$  usando la tabla de verdad extendida:

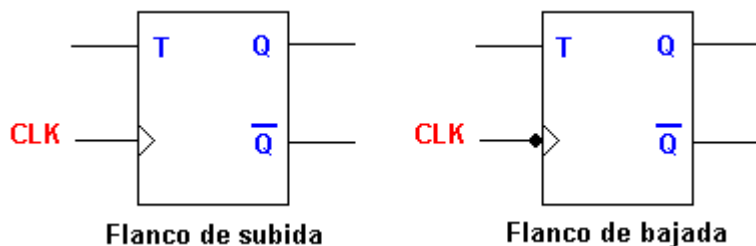
$$Q(t+1) =$$

### 4. Flip - Flop-T.

Su funcionamiento se define por la siguiente tabla de verdad:

T	CLK	Q(t+1)
0	↑ ó ↓	$Q(t)$
1	↑ ó ↓	$\overline{Q(t)}$

Su símbolo es:



Si consideramos como una entrada el estado de salida actual del FF, obtenemos la siguiente tabla de verdad extendida:

T	Q(t)	CLK	Q(t+1)
0	0	↑ ó ↓	0
0	1	↑ ó ↓	1
1	0	↑ ó ↓	1
1	1	↑ ó ↓	0

Obtener la ecuación para  $Q(t+1)$  usando la tabla de verdad extendida:

$$Q(t+1) =$$

## PROCEDIMIENTO.

Antes de asistir al laboratorio:

1. Realizar un programa en HDL que permita implementar el diseño de la ilustración 1.

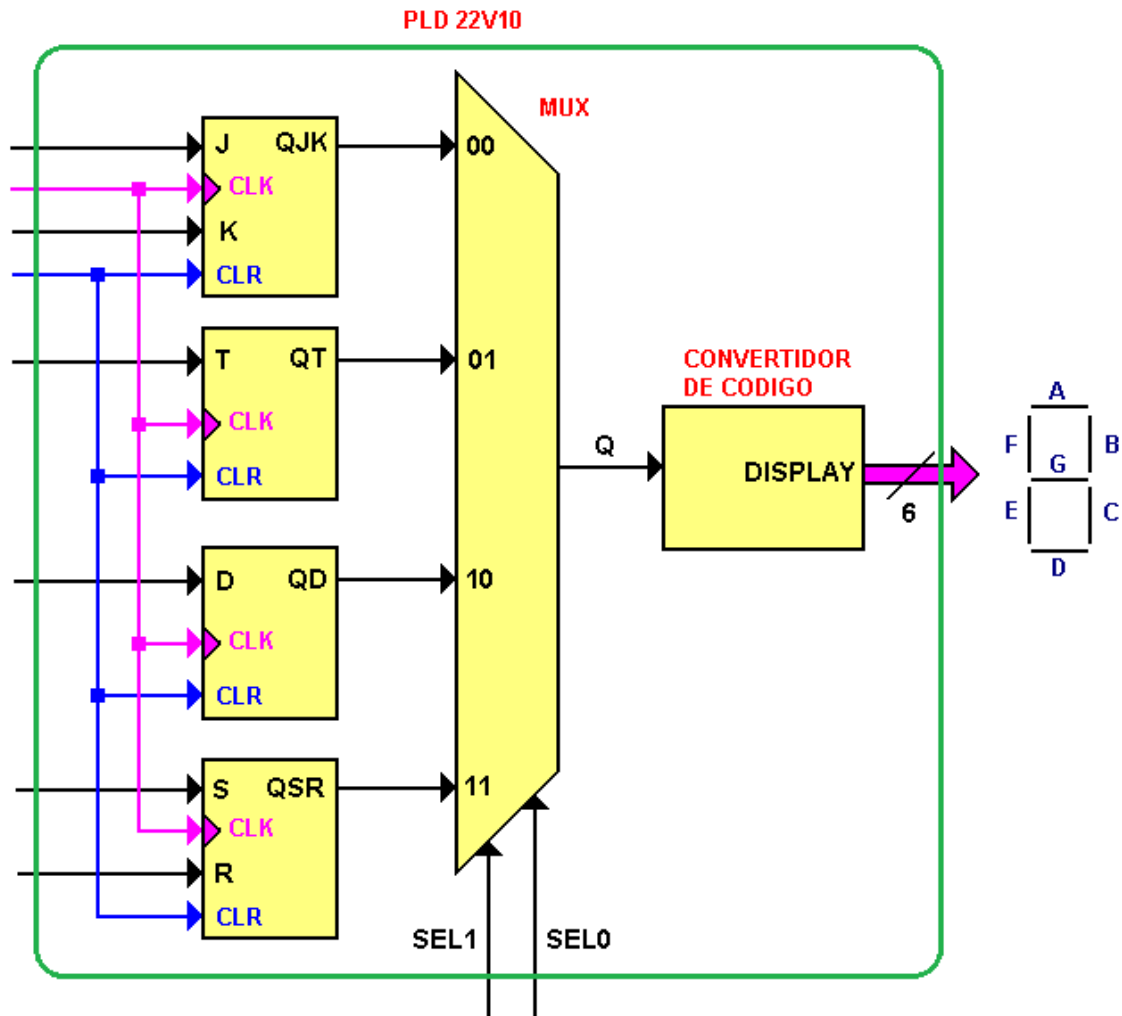


Ilustración 1 Diagrama a bloques del diseño a realizar.

Para implementar los Flip-Flops (FF's) se deben usar las ecuaciones obtenidas de las tablas de verdad extendida. Cabe notar que estas ecuaciones describen a cada FF mediante el FF-D. Esto permite implementar los FF's en el PLD 22V10 ya que este PLD tiene en las macroceldas lógicas de salida solo FF-D. Estos 4 FF's ocupan 4 macroceldas de salida (OLMC – Output Logic Macrocell) del PLD 22V10 y por lo tanto 4 terminales de salida, por lo que solamente quedan disponibles 6 terminales de salida para conectar el display de 7 segmentos.

En la ilustración 1 se muestra el diseño completo a programar donde interviene un FF-JK, un FF-T, un FF-D, un FF-SR, un multiplexor y un convertidor de código. El multiplexor funciona de acuerdo a la tabla 1.

SEL1	SEL0	Q
0	0	$QJK$
0	1	$QT$
1	0	$QD$
1	1	$QSR$

Tabla 1 Funcionamiento del multiplexor.

El convertidor de código convierte el valor 0 ó 1 en un valor de siete bits para mostrar en un display de siete segmentos de ánodo común los números 0 y 1, los cuales se dibujarán como se muestra en la ilustración 2.

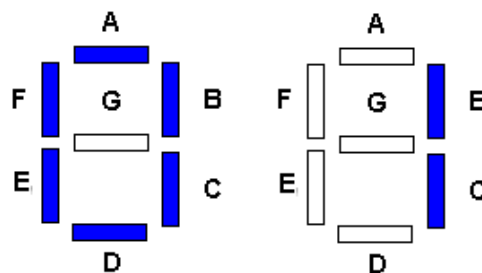


Ilustración 2 Números 0 y 1.

El funcionamiento del convertidor de código se muestra en la tabla 2.

Q	Símbolo	A Display(5)	B Display(4)	C Display(3)	D Display(2)	E Display(1)	F Display(0)	G
0	0	0	0	0	0	0	0	1
1	1	1	0	0	1	1	1	1

Tabla 2 Códigos de los símbolos de 0 y 1.

En la tabla 2 se observa que el segmento G siempre debe de estar apagado, esto se logra de dos formas:

- Mandando un “1” lógico a la terminal de salida del PLD 22V10 al que se conecta el segmento G.
- Colocando la terminal de salida del PLD 22V10 al que se conecta el segmento G, en alta impedancia.

En esta práctica se tiene que usar la segunda forma. Esto se logra al usar el FF de la macrocelda de salida de esa terminal como una señal interna. De esta forma solamente tenemos que conectar 6 segmentos del display al PLD 22V10 (A, B, C, D, E y F).

La asignación de pines la deben de hacer de acuerdo a la disponibilidad que se muestra en la hoja de especificaciones del PLD 22V10. Las señales del diseño pueden distribuirse de acuerdo a la ilustración 3.



---

C22V10

clk =  1	24  * not used
j =  2	23 = (qsr)
k =  3	22 = (qd)
t =  4	21 = display(5)
s =  5	20 = display(4)
r =  6	19 = display(3)
d =  7	18 = display(2)
sel(0) =  8	17 = display(1)
sel(1) =  9	16 = display(0)
not used * 10	15 = (qjk)
not used * 11	14 = (qt)
not used * 12	13 = clr

---

Ilustración 3 Distribución de señales en el PLD 22V10.

2. Simular el diseño en el ambiente de desarrollo.
3. Una vez simulado el sistema construir el circuito mostrado en la ilustración 4 para probarlo en el laboratorio. **En caso de usar la TEDDi este paso no es necesario.**

En el laboratorio:

1. Programar el PLD 22V10 usando el programador disponible del laboratorio.
2. Colocar la frecuencia de la señal de reloj a 1 HZ. En caso de usar TEDDi ajustar la frecuencia con el potenciómetro “FREC”. En caso de haber armado el circuito en protoboard, usar el generador de funciones.
3. Verificar el correcto funcionamiento de cada Flip - Flop.

**CUESTIONARIO**

1. ¿Cuántos dispositivos PLD 22V10 son necesarios para el desarrollo de esta práctica?
2. ¿Cuántos dispositivos de la serie 74xx (TTL) ó 40xx (CMOS) hubieras necesitado para el desarrollo de esta práctica?
3. ¿Cuántos pines de entrada/salida del PLD 22V10 se usan en el diseño?
4. ¿Cuántos términos producto ocupan las ecuaciones para cada señal de salida y que porcentaje se usa en total del PLD 22V10?
5. ¿Por qué aparecen las señales QJK, QT, QD Y QSR entre paréntesis en los pines de salida?
6. ¿Cuáles son las señales que funcionan de manera síncrona y cuales de manera asíncrona?
7. ¿Qué puedes concluir de esta práctica?

**REFERENCIAS**

[1] Ronald J. Tocci, “Sistemas digitales, principios y aplicaciones”, Prentice Hall, 2007.

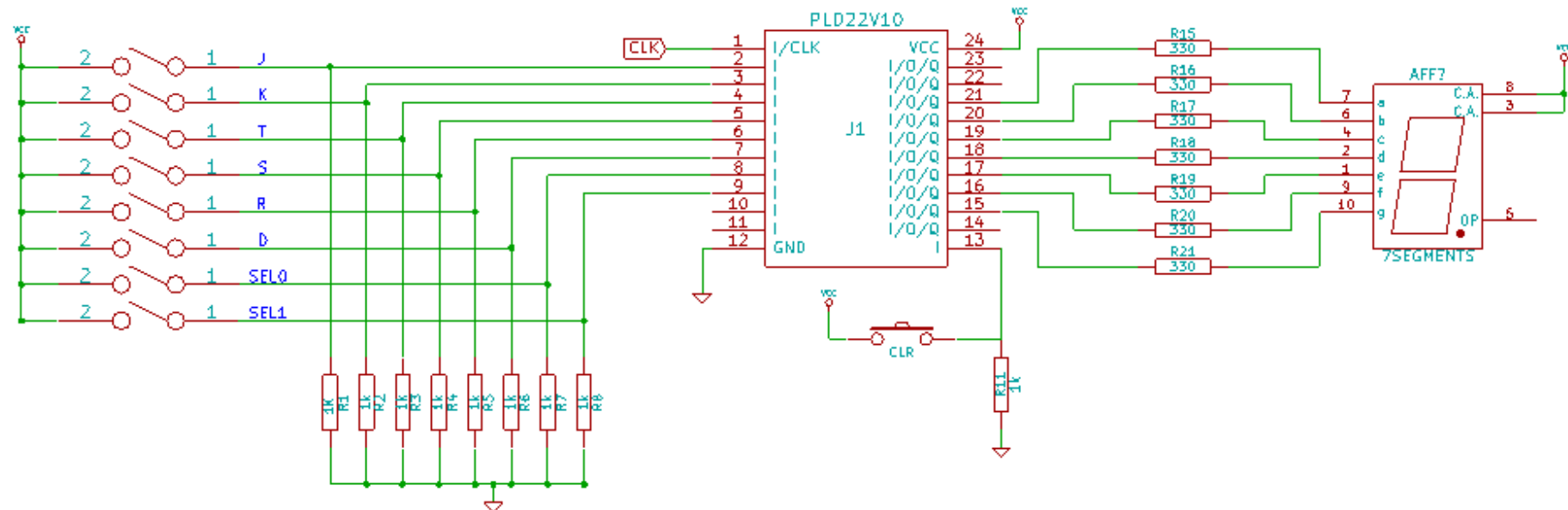


Ilustración 4 Diagrama esquemático.