

# **Diseño Secuencial de un circuito usando flip-flops con dispositivos lógicos programables**

Demetrio Manuel Roa Perdomo

## **Notas del autor**

Demetrio Manuel Roa Perdomo, Facultad de Ingeniería Mecánica y Eléctrica, Universidad

Autónoma de Nuevo León

Esta investigación ha sido financiada por el propio alumno

La correspondencia relacionada con esta investigación debe ser dirigida a Demetrio Roa

Universidad Autónoma de Nuevo León, Pedro de Alba S/N, Niños Héroes, Ciudad Universitaria,

San Nicolás de los Garza, N.L.

Contacto: [demetrio.roap@uanl.edu.mx](mailto:demetrio.roap@uanl.edu.mx)

## **Diseño Secuencial de un circuito usando flip-flops con dispositivos lógicos programables**

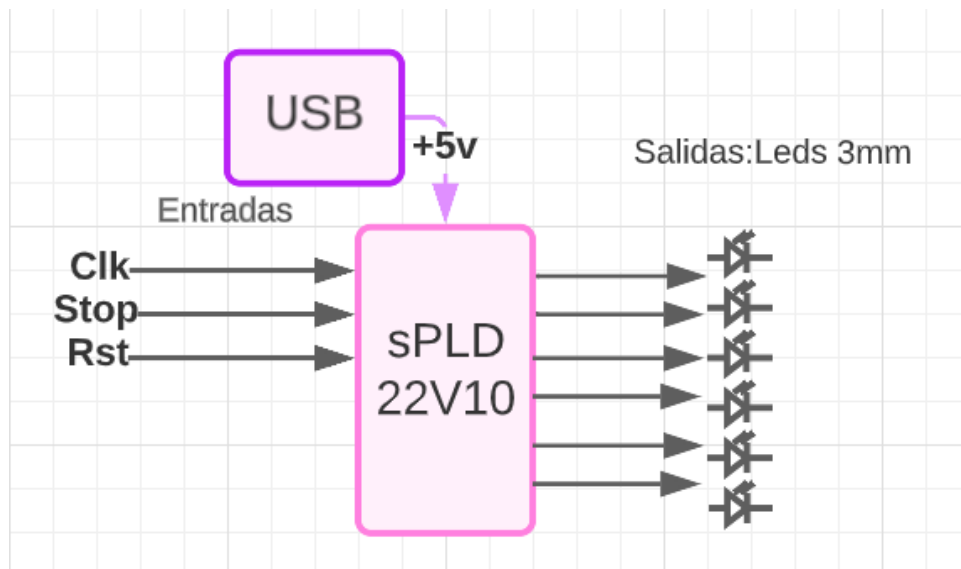
### **Introducción**

En electrónica, un driver o controlador (aunque en español es más habitual usar el término en inglés) es un circuito eléctrico u otro componente electrónico utilizado para controlar otro circuito o componente. Suele ser muy común usarlos para controlar transistores alta potencia y pantallas de cristal líquido (LCD), entre muchos otros.

Normalmente, se utilizan para regular la corriente que fluye a través de un circuito o para controlar otros componentes. El término se usa a menudo, por ejemplo, para un circuito integrado especializado que controla interruptores de alta potencia en convertidores de potencia de modo conmutado. Un amplificador también puede ser considerado un controlador para altavoces. O incluso, un regulador de tensión que mantenga la tensión de un componente conectado a él operando dentro de un amplio rango de tensiones de entrada, se puede considerar un driver.

Los controladores de led serie no son más que un arreglo de fuentes de corriente que permite encender, apagar y controlar la ganancia de cada fuente. Mediante ellos se pueden utilizar varios leds y controlarlos independientemente con una mínima cantidad de pines del microcontrolador, ahorrando así no sólo circuitería adicional y espacio sino también el conteo de pines del microcontrolador. El protocolo utilizado para controlar este integrado es un protocolo serie con línea de control de datos, clock, activación de salidas y latcheo de datos (guarda en memoria); lo que en total necesita 4 pines del microcontrolador. En un arreglo paralelo la cantidad de pines es proporcional a la cantidad de leds a controlar.

## Diagrama de Bloques completo



## Lista de materiales utilizados en formato BOM

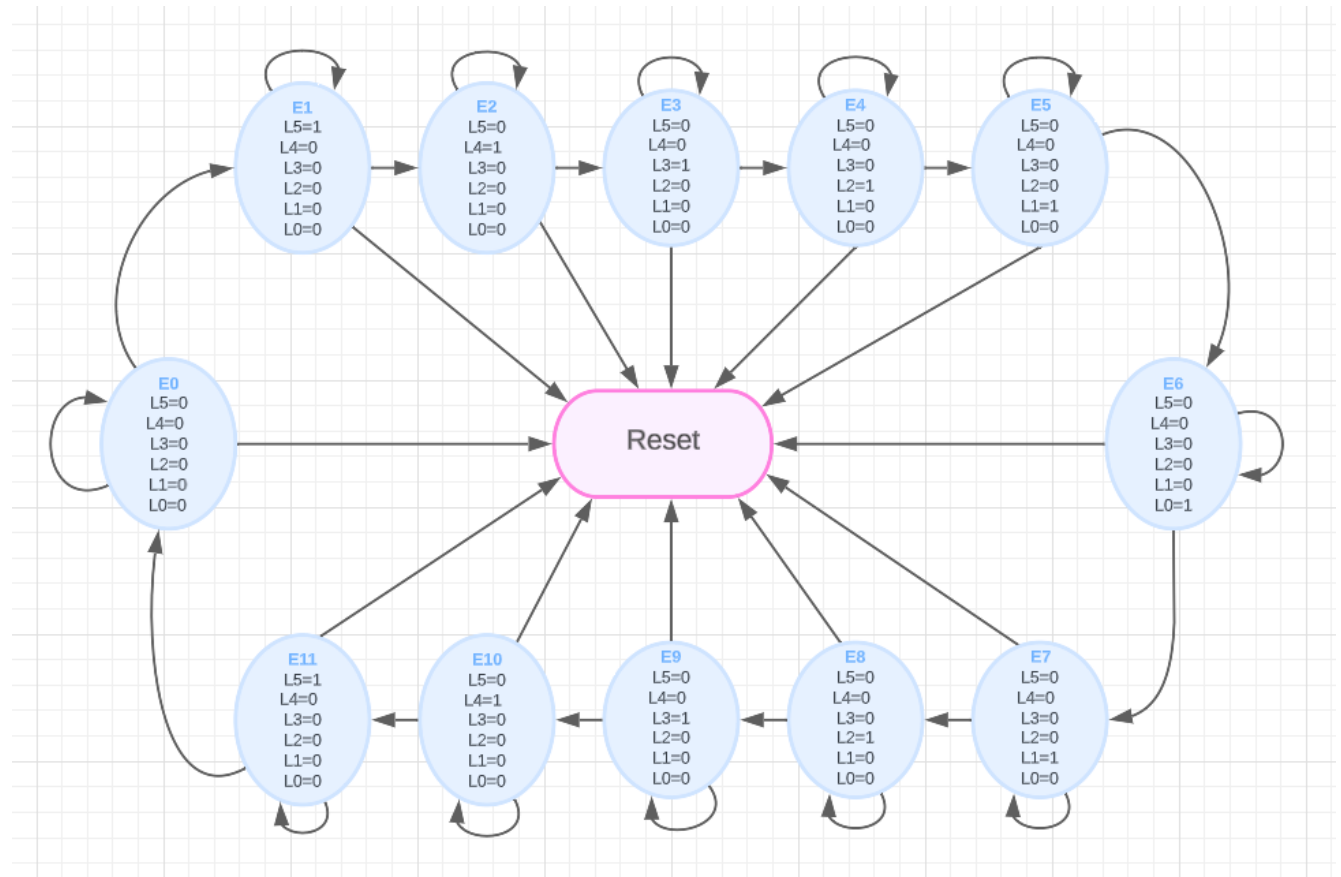
Nombre	Cantidad	Componente	Num. De Parte	Precio Unitario	Costo Total	Datasheet
C	1	Capacitor 1000uF		\$15	\$15	
Pot	1	Potenciometro 1k		\$12	\$12	
R	8	Resistancias 220 ohms		\$1	\$8	
LED	6	LEDs 3mm		\$2	\$12	
LED	2	LEDs 5mm		\$2	\$4	
PLD	1	PLD	ATF22V10C	\$80	\$80	
NOT	1	Compuerta NOT	SN74LS90N	\$25	\$25	
Alambre	1m	Alambre calibre 22 negro y rojo			\$20	

**Tabla de estado siguiente**

Entradas					Estado	Estados Siguiente		
						If clk	If stop	If rst
	<b>Start</b>	1	0	0	E0	E1	E0	E0
	<b>Test</b>	0	1	0	E1	E2	E1	E0
	<b>Sensor</b>	0	0	1	E2	E3	E2	E0
	<b>Reset</b>							
					E3	E4	E3	E0
					E4	E5	E4	E0
					E5	E6	E5	E0
					E6	E7	E6	E0
					E7	E8	E7	E0
					E8	E9	E8	E0
					E9	E10	E9	E0
					E10	E10	E10	E0

Estados	Salidas					
	LED 1	LED 2	LED 3	LED 4	LED 5	LED 6
<b>E0</b>	1	0	0	0	0	0
<b>E1</b>	0	1	0	0	0	0
<b>E2</b>	0	0	1	0	0	0
<b>E3</b>	0	0	0	1	0	0
<b>E4</b>	0	0	0	0	1	0
<b>E5</b>	0	0	0	0	0	1
<b>E6</b>	0	0	0	0	1	0
<b>E7</b>	0	0	0	1	0	0
<b>E8</b>	0	0	1	0	0	0
<b>E9</b>	0	1	0	0	0	0
<b>E10</b>	1	0	0	0	0	0

## Máquina de estados finitos (FSM)





## Imagen del código en lenguaje HDL

```
MODULE AFU4

TITLE 'Leds'

DECLARATIONS

"Inputs"
clk PIN 1;
stop PIN 2;
rst PIN 3;

"Outputs"
//Combinacionales
led1,led2,led3,led4,led5,led6 PIN 23,22,21,20,19,18 ISTYPE 'COM';

//Registradas
q3,q2,q1,q0 PIN 17,16,15,14 ISTYPE 'REG';

"Set"
FF = [q3,q2,q1,q0];
E0 = [0,0,0,0];
E1 = [0,0,0,1];
E2 = [0,0,1,0];
E3 = [0,0,1,1];
E4 = [0,1,0,0];
E5 = [0,1,0,1];
E6 = [0,1,1,0];
E7 = [0,1,1,1];
E8 = [1,0,0,0];
E9 = [1,0,0,1];
E10 = [1,0,1,0];
E11 = [1,0,1,1];
E12 = [1,1,0,0];
E13 = [1,1,0,1];
E14 = [1,1,1,0];
E15 = [1,1,1,1];

EQUATIONS
FF.clk = clk;//Sincronizar el reloj
FF.sr = rst;

STATE_DIAGRAM FF

STATE E0:
led1 = 1;
led2 = 0;
led3 = 0;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E0;
IF !stop THEN E1;
IF rst THEN E0;
```

```
STATE E1:
led1 = 0;
led2 = 1;
led3 = 0;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E1;
IF !stop THEN E2;
IF rst THEN E0;
```

```
STATE E2:
led1 = 0;
led2 = 1;
led3 = 0;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E2;
IF !stop THEN E3;
IF rst THEN E0;
```

```
STATE E3:
led1 = 0;
led2 = 0;
led3 = 1;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E3;
IF !stop THEN E4;
IF rst THEN E0;
```

```
STATE E4:
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 1;
led5 = 0;
led6 = 0;
IF stop THEN E4;
IF !stop THEN E5;
IF rst THEN E0;
```

```
STATE E5:
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 0;
led5 = 1;
led6 = 0;
IF stop THEN E5;
IF !stop THEN E6;
IF rst THEN E0;
```



---

```
STATE E6:
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 0;
led5 = 0;
led6 = 1;
IF stop THEN E6;
IF !stop THEN E7;
IF rst THEN E0;

STATE E7:
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 0;
led5 = 1;
led6 = 0;
IF stop THEN E7;
IF !stop THEN E8;
IF rst THEN E0;

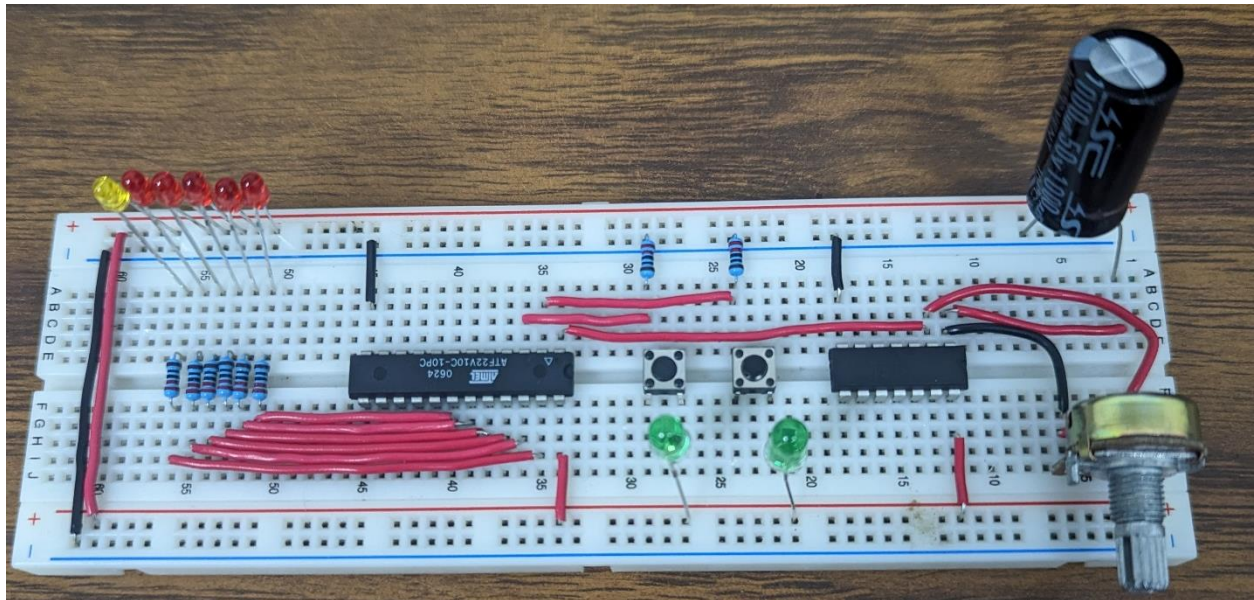
STATE E8:
led1 = 0;
led2 = 0;
led3 = 0;
led4 = 1;
led5 = 0;
led6 = 0;
IF stop THEN E8;
IF !stop THEN E9;
IF rst THEN E0;

STATE E9:
led1 = 0;
led2 = 0;
led3 = 1;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E9;
IF !stop THEN E10;
IF rst THEN E0;

STATE E10:
led1 = 0;
led2 = 1;
led3 = 0;
led4 = 0;
led5 = 0;
led6 = 0;
IF stop THEN E10;
IF !stop THEN E0;
IF rst THEN E0;
```

END

## Fotografía del prototipo armado



## Conclusiones

En esta actividad, hemos comprendido el uso de los flip-flops al crear un diseño secuencial con dispositivos lógicos programables. Hemos diseñado e implementado un sistema electrónico digital secuencial capaz de mostrar una secuencia específica mediante 6 LEDs. El programa consta de 11 estados, y cuenta con entradas para parar y restablecer la secuencia. Esto nos ha permitido aplicar nuestros conocimientos en el diseño de sistemas secuenciales y comprender cómo funcionan en la práctica.

Además, hemos aprendido cómo utilizar entradas para controlar el comportamiento del sistema y cómo implementar un reset asíncrono, así como el mismo botón de pausa. En general, esta actividad nos ha brindado una valiosa experiencia práctica en el diseño e implementación de sistemas electrónicos digitales secuenciales, cosa que será de gran importancia para futuros proyectos que lleguemos a aplicar incluso dentro del área biomédica.

## **Referencias Bibliográficas**

alldatasheet.com. (s. f.). ATF22V10C Datasheet, PDF - Alldatasheet.

<https://www.alldatasheet.com/view.jsp?Searchword=Atf22v10c>

Harris, D., & Harris, S. (2018). Digital design and computer architecture: ARM edition. Morgan Kaufmann.

Roth, C. H., & Kinney, P. J. (2019). Fundamentals of logic design. Cengage Learning.

Tokheim, R. L. (2018). Digital electronics: Principles and applications. McGraw-Hill Education.

Electronics Tutorials. (n.d.). Sequential Logic Circuits and the SR Flip-flop. Retrieved from

[https://www.electronics-tutorials.ws/sequential/seq\\_1.html](https://www.electronics-tutorials.ws/sequential/seq_1.html)

Hanson, K. (2015). Digital Electronics Part II: Sequential Logic. Department of Physics - University of Wisconsin-Madison. Retrieved from

[https://www.physics.wisc.edu/courses/home/spring2021/623/lecture\\_notes/digital/digital-02.html](https://www.physics.wisc.edu/courses/home/spring2021/623/lecture_notes/digital/digital-02.html)