

Diseño de Sistemas Secuenciales Síncronos



TODOS SOMOS
FAMILIA

En los momentos de crisis, sólo la imaginación y el esfuerzo es más importante que el conocimiento.

Albert Einstein



El Pinal
Chipinque, N, L,
Sábado 2 Nov. 2013

Sábado 2 agosto 2025



Paraje el Pinal Jueves 13 Abril 2017



Paraje el Epazote Sábado 22 Abril 2017



Paraje el Epazote, Domingo 4 nov. 2018



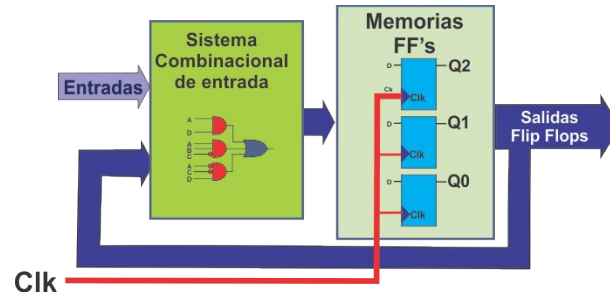
Paraje el Pinal Domingo 14 enero 2019

Actividades y proyectos en proceso

	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	Miércoles 26 Marzo
PF7	Diseño Combinacional con HDL	F	Lunes 31 de Marzo
PF8	Multiplexor	5	Viernes 4 de Abril
AF3	Decodificador con Display	10	Miércoles 9 de Abril
PF9	Flip Flops	F	Martes 29 de Abril
PF10	Pulsos de sincronía	5	Lunes 12 de Mayo
AF4	Diseño Secuencial	10	
AF5	PIA	40	Día del examen

sesión de Hoy

1.- La máquina de Moore usando solo salidas Registradas (FFs)



2.- Tabla de estado siguiente simplificada en la descripción de la secuencia usando **Don't Care (X)**.

3.-Uso del **Truth_Table** :> en la descripción de una secuencia.

4.- Descripción de **Stae_Diagram** , Else IF Encadenamiento con Prioridad

a) Reset **Asíncrono** como prioridad

b) Prioridad LT Preset **síncrono** (SP).

Método de Diseño de Sistemas Secuenciales síncronos con el uso de HDL y su implementación en un PLD

1.- Especificar el sistema (Diagrama de transición)

2.- Determinar la cantidad de Flip Flops

a) **Dependiendo de los estados**

b) **Usando solo salidas Registradas (FFs)**

3.- Asignar valores a los estados

4.- Diagrama de Bloque (entradas y salidas)

5.- Construir la tabla de estado siguiente.

Base para
describir un
sistema
secuencial

6.- Codificación en ABEL-HDL

a) Entradas y salidas

b) Sincronizar los Flip Flops

c) Asignar de valores a los estados

d) definir la secuencia (state_diagram o Truth_table)

7.- Simulación

8.- Implementación.

Descripción de la secuencia en el lenguaje ABEL-HDL

Goto

"Diagrama de transición

State_Diagram Sinc

STATE Verde:

Ve=1; Fl=0; Am=0; Ro=0;

Goto Flecha;

STATE Flecha:

Ve=0; Fl=1; Am=0; Ro=0;

goto Ambar;

STATE Ambar:

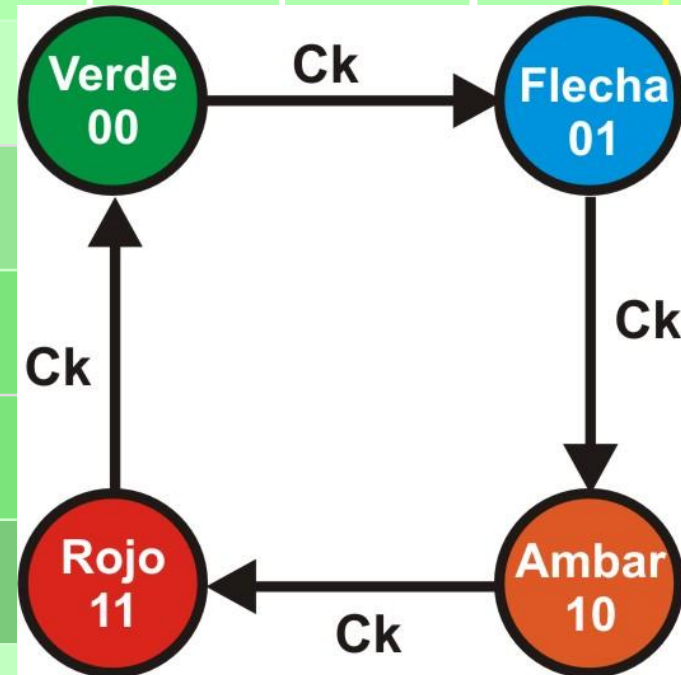
Ve=0; Fl=0; Am=1; Ro=0;

goto Rojo;

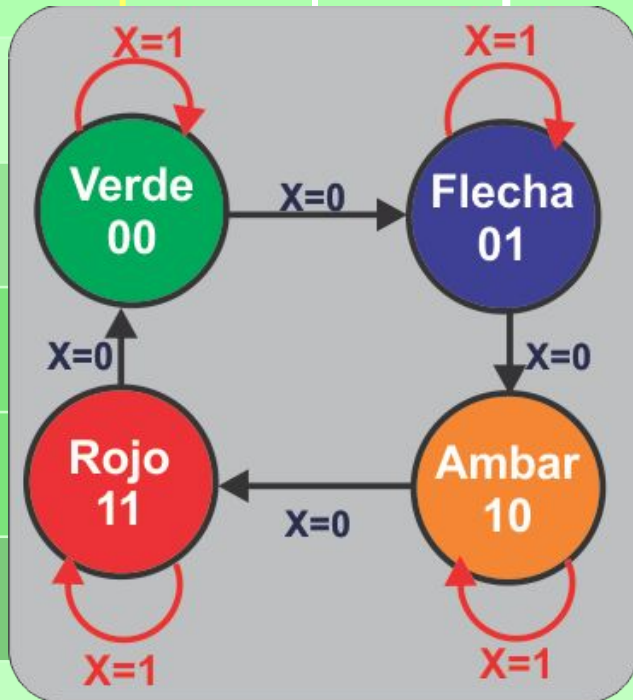
STATE Rojo:

Ve=0; Fl=0; Am=0; Ro=1;

goto Verde;



IF, Then, Else



"Diagrama de transición

State_Diagram Sinc

STATE Verde:

Ve=1; Fl=0; Am=0; Ro=0;

If X then Verde else Flecha;

STATE Flecha:

Ve=0; Fl=1; Am=0; Ro=0;

If !X then Ambar;

If X then Flecha;

STATE Ambar:

Ve=0; Fl=0; Am=1; Ro=0;

If X==0 then Rojo else Ambar;

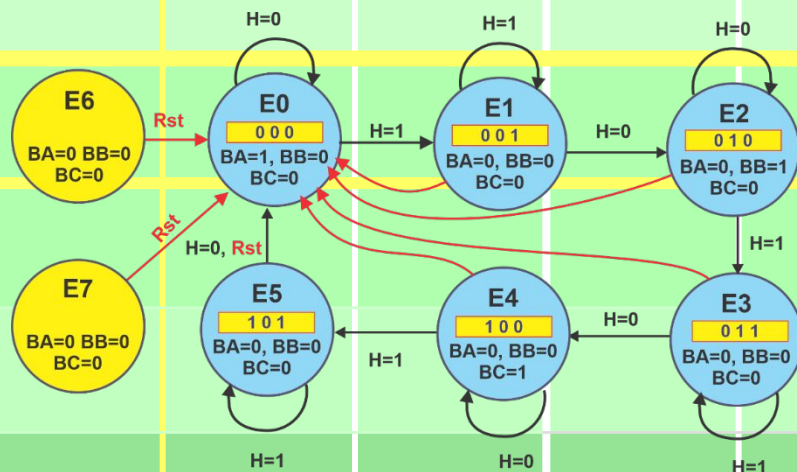
STATE Rojo:

Ve=0; Fl=0; Am=0; Ro=1;

If X==1 then Rojo;

If X==0 then Verde;

IF, Then, else



E. P	Estado próximo		Salidas		
	H=0	H=1	BA	BB	BC
E0	E0	E1	1	0	0
E1	E2	E1	0	0	0
E2	E2	E3	0	1	0
E3	E4	E3	0	0	0
E4	E4	E5	0	0	1
E5	E0	E5	0	0	0

State_diagram E

State E0:

BA=1;BB=0;BC=0;

IF H then E1;

IF !H then E0;

State E1:

BA=0;BB=0;BC=0;

IF H then E1;

IF !H then E2;

State E2:

BA=0;BB=1;BC=0;

IF H then E3;

IF !H then E2;

State E3:

BA=0;BB=0;BC=0;

IF H then E3;

IF !H then E4;

State E4:

BA=0;BB=0;BC=1;

IF H then E5;

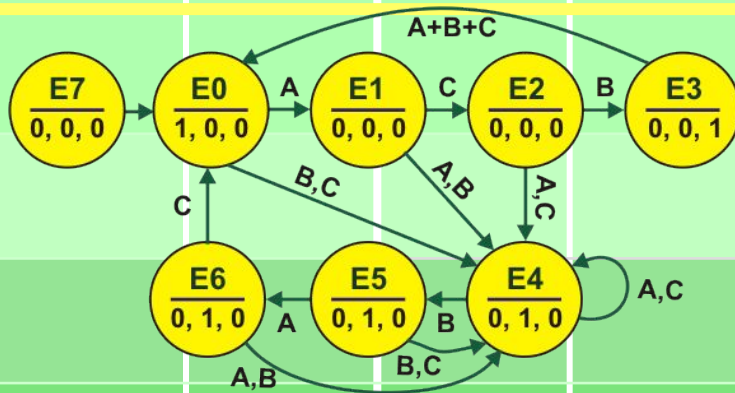
IF !H then E4;

State E5:

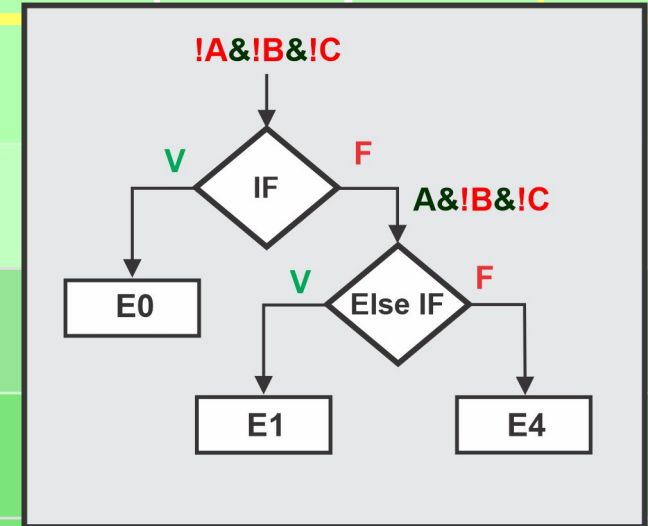
BA=0;BB=0;BC=0;

IF H then E5 else E0;

If-Then-Else Encadenado (Chained) por repetición



A	0	0	0	0	1	1	1	1			
B	0	0	1	1	0	0	1	1			
C	0	1	0	1	0	1	0	1	CI	AL	PA
E0	E0	E4	E4	E4	E1	E4	E4	E4	1	0	0



IF **!A&!B&!C** THEN E0 Else IF **A&!B&!C** THEN E1 Else **E4**;

M=[A,B,C];

IF **M==0** THEN E0 Else IF **M==4** THEN E1 Else **E4**;

If-Then-Else

Encadenamiento con prioridad

State E1:

IF !Rst&!LT&P&S then E2;

IF !Rst&!LT&P&S then E2;

IF !Rst&!LT&P then E1;

IF !Rst< then E7;

IF Rst then E0;

State E1:

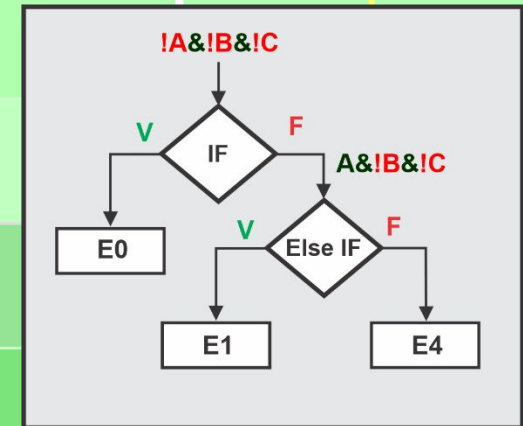
IF !Rst&!LT&P then E2;

IF !Rst&!LT&P then E1;

IF !Rst< then E7;

IF Rst then E0;

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E1	E0	E7	E1	E2	E2



State E1:

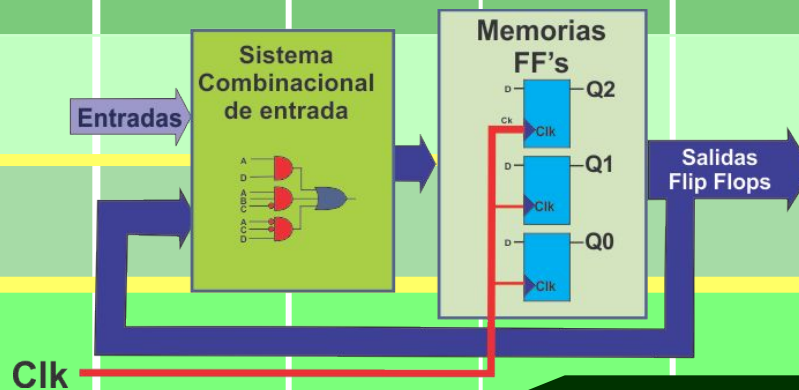
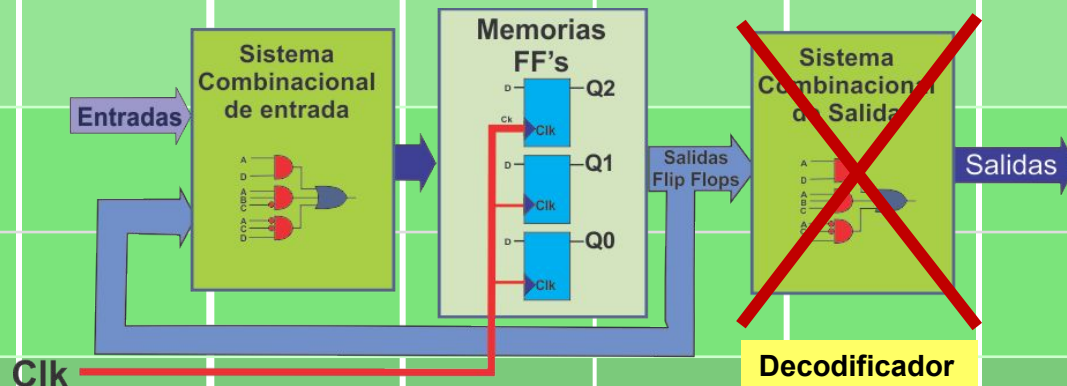
IF Rst then E0 else if LT then E7 else if P then E1 else E2;

2.- Determinar la cantidad de Flip Flops

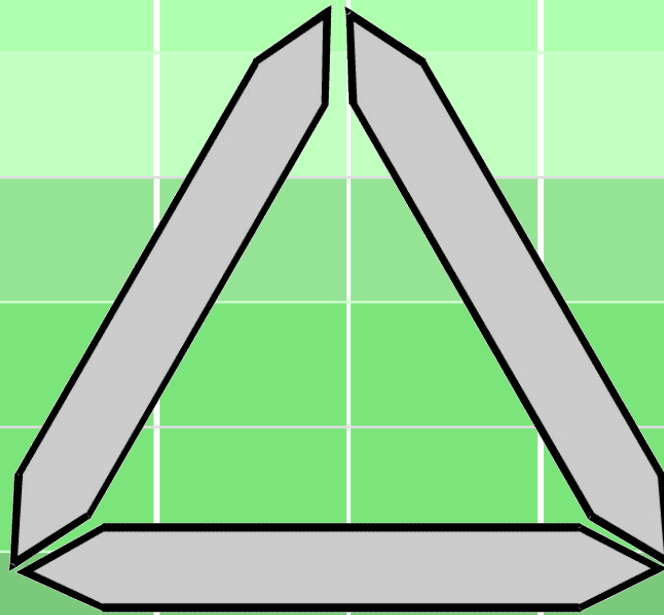
a) Dependiendo de los estados

b) Usando solo salidas Registradas (FFs)

Máquina de Moore

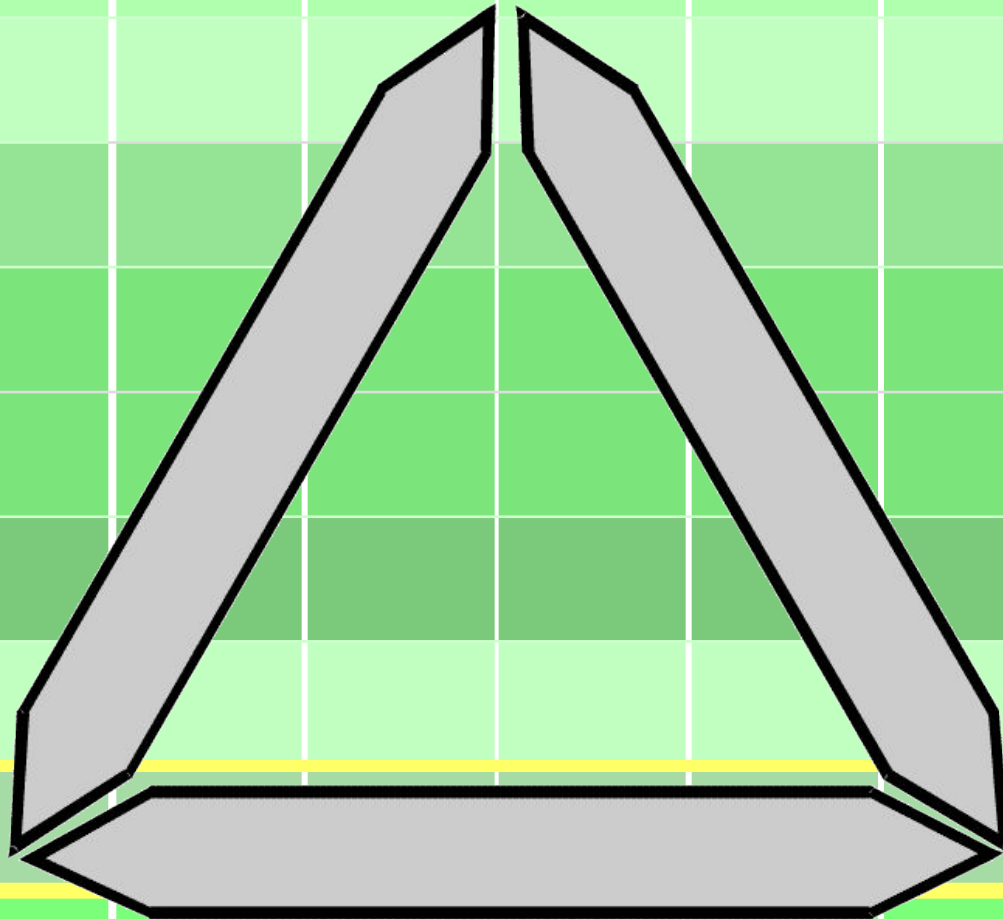


Diseño de Sistemas Secuenciales síncronos



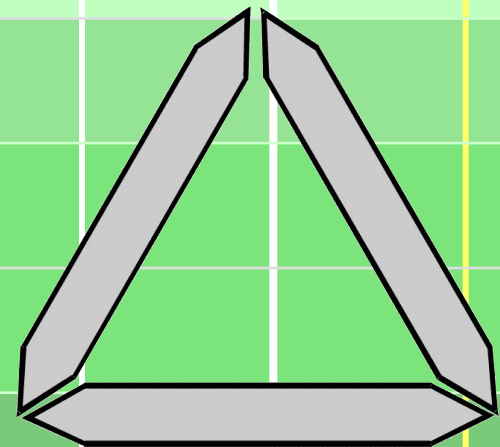
Triángulo Led Luz de Seguridad de Emergencia de avería de Automóvil

Señal de alerta

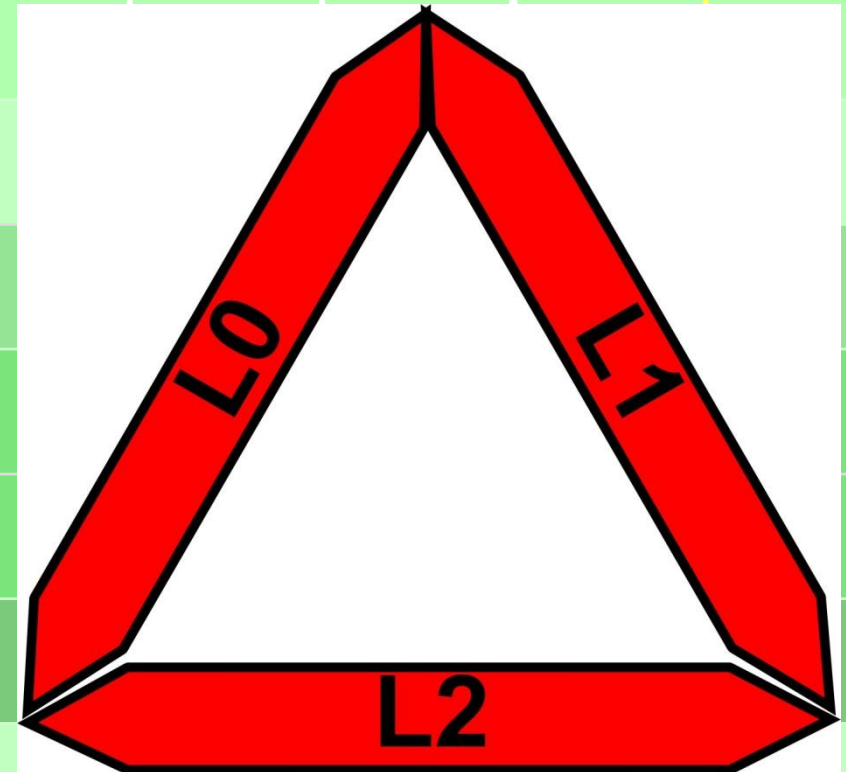


Control de Señal de Alerta

**Diseñe un sistema secuencial
síncrono y cíclico, usando la
Máquina de Moore que sea el
control de una señal de alerta**

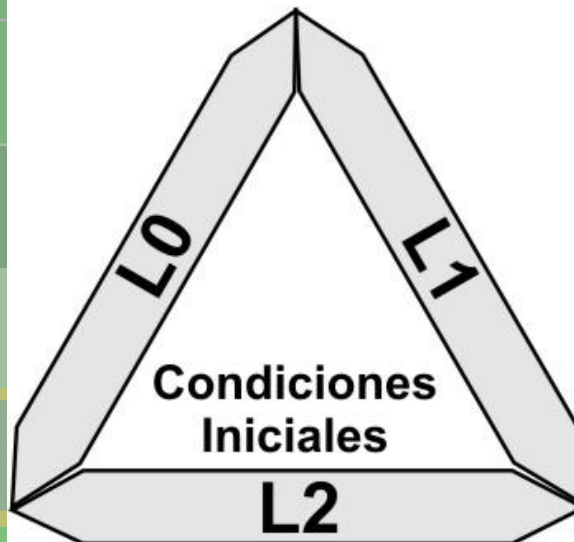


**Dicha señal consta de
3 luces en forma de
triángulo llamadas
L2, L1 y L0.**



Se requieren de dos diferentes secuencias que son seleccionadas por medio un interruptor **S** de modo que:

- a) Ambas parten de condiciones iniciales en donde todas las lámparas están apagadas .

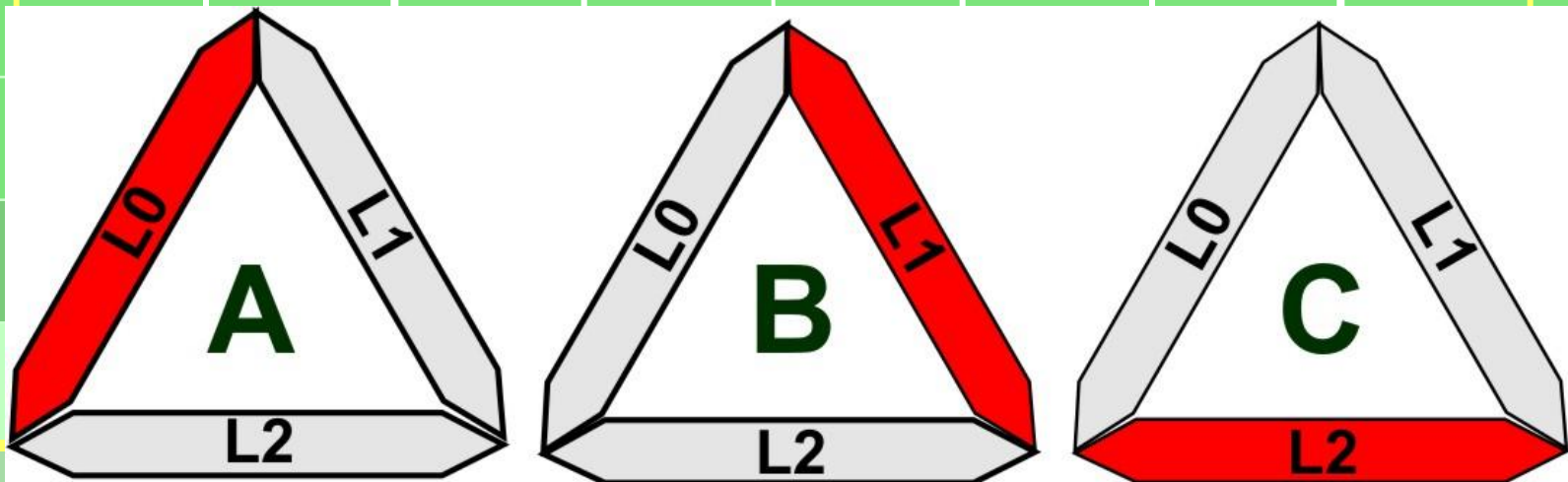


b) Si $S=0$ Ocurrirá la secuencia Uno

A, B, C, A, B, C

Repetidamente en forma cíclica

Una luz a la vez



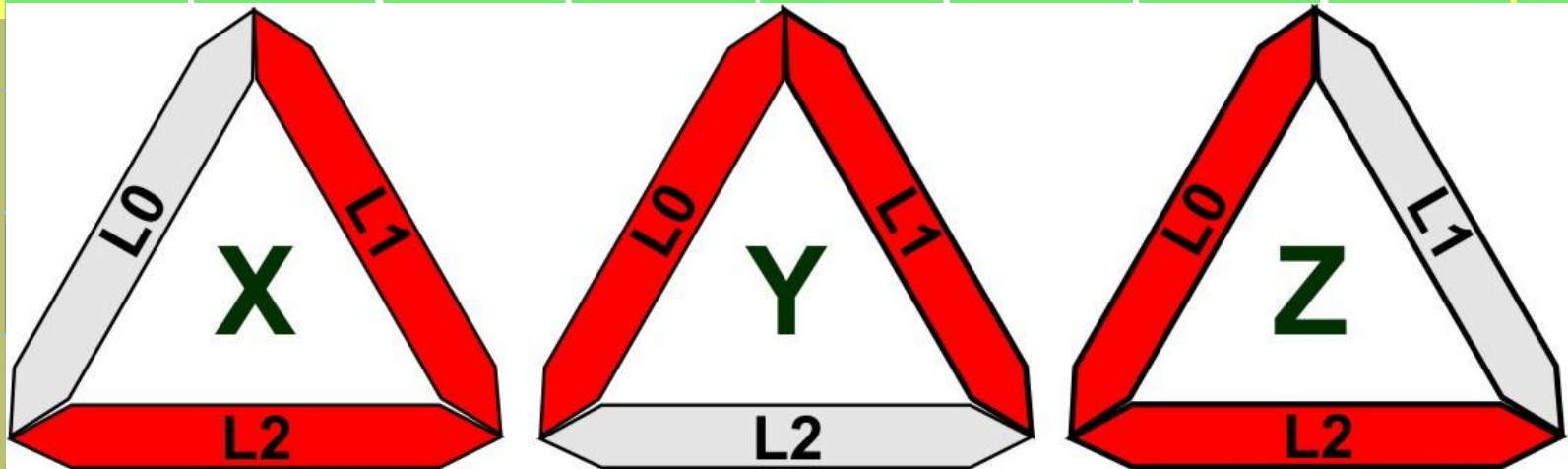
Secuencia Uno

c) Si $S=1$ ocurrirá la secuencia Dos

X, Y, Z, X, Y, Z

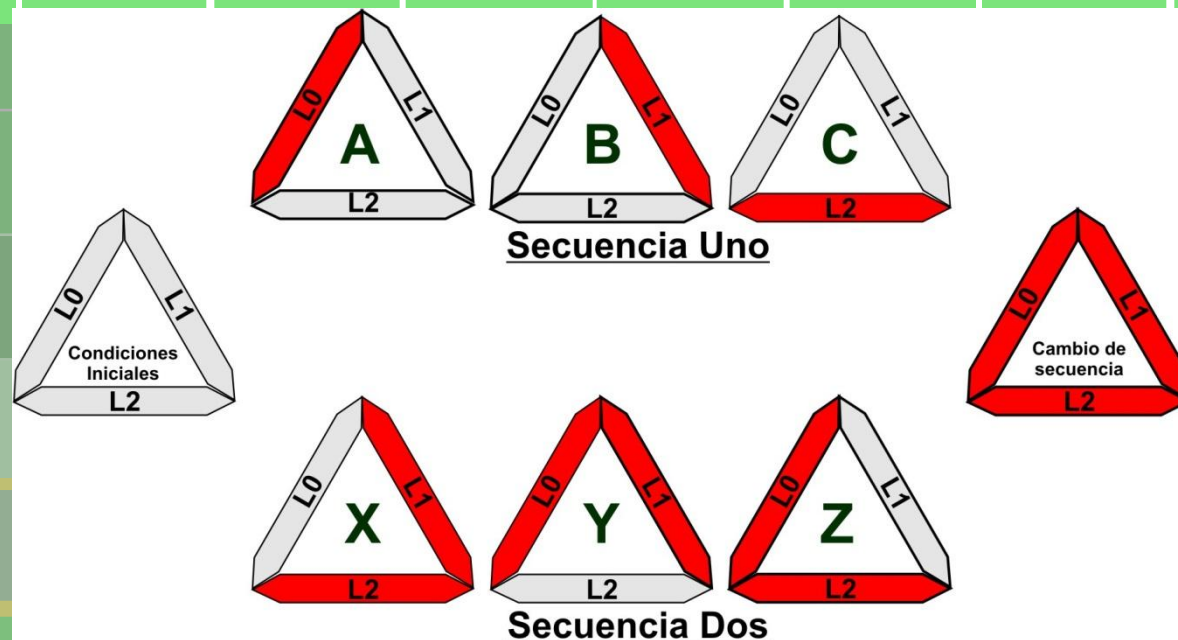
Repetidamente en forma cíclica

Dos luces a la vez



Secuencia Dos

Nota: Cuando usted cambia el selector S en medio de una secuencia, las luces continuarán con la secuencia actual hasta terminar (C o Z) y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S



Por Ejemplo:

Si $S=0$ y un selector S se hace 1, en el momento que la secuencia de luces es ($L_0=0$, $L_1=1$ y $L_2=0$), correspondiente a la condición B de la secuencia Uno, continuará hasta C y de ahí a todos encendidos ($L_0=1$, $L_1=1$ y $L_2=1$) y al siguiente pulso de reloj a condiciones iniciales ($L_0=0$, $L_1=0$ y $L_2=0$) y de ahí en adelante continuará con la secuencia Dos.

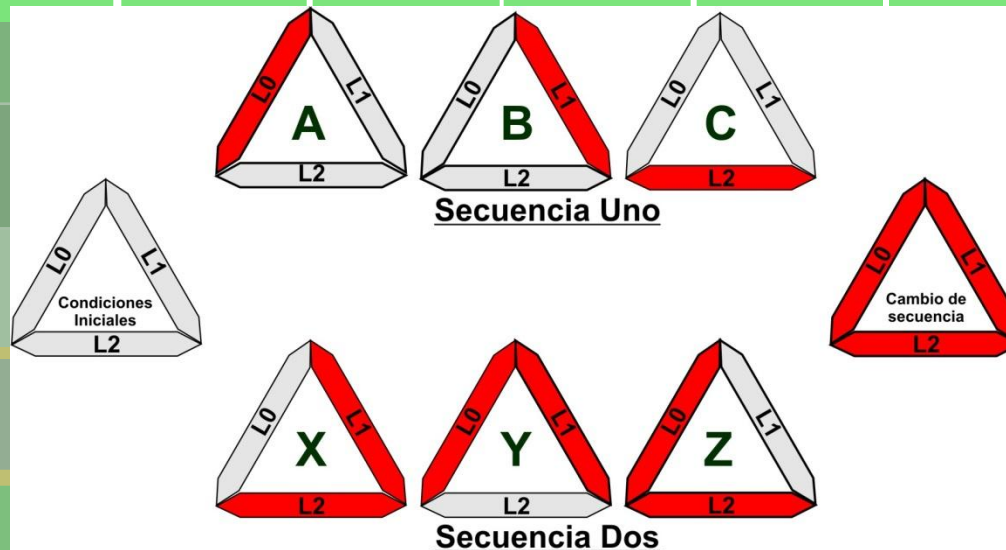
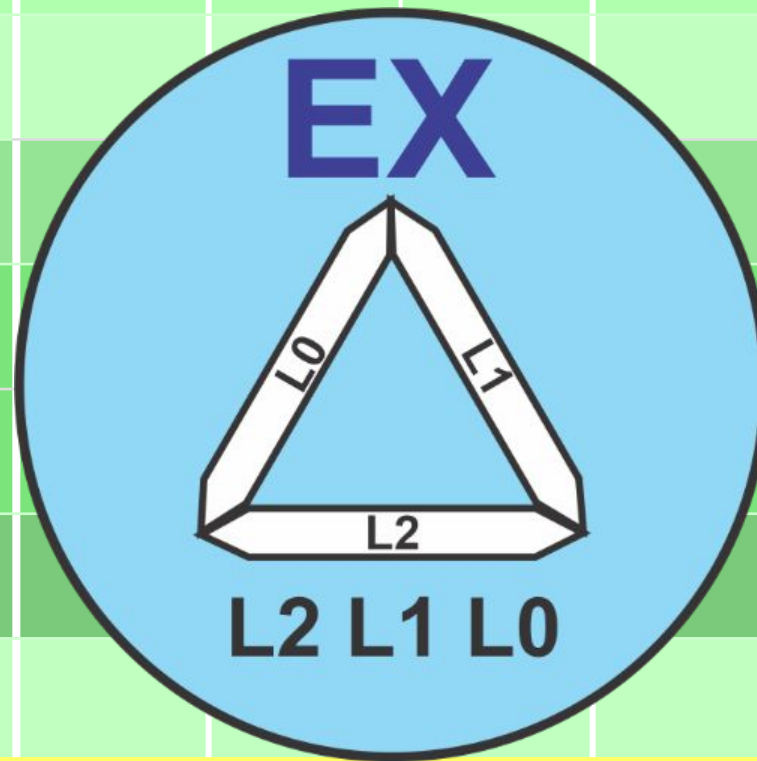
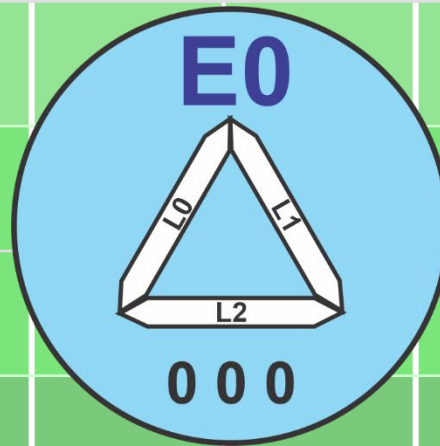


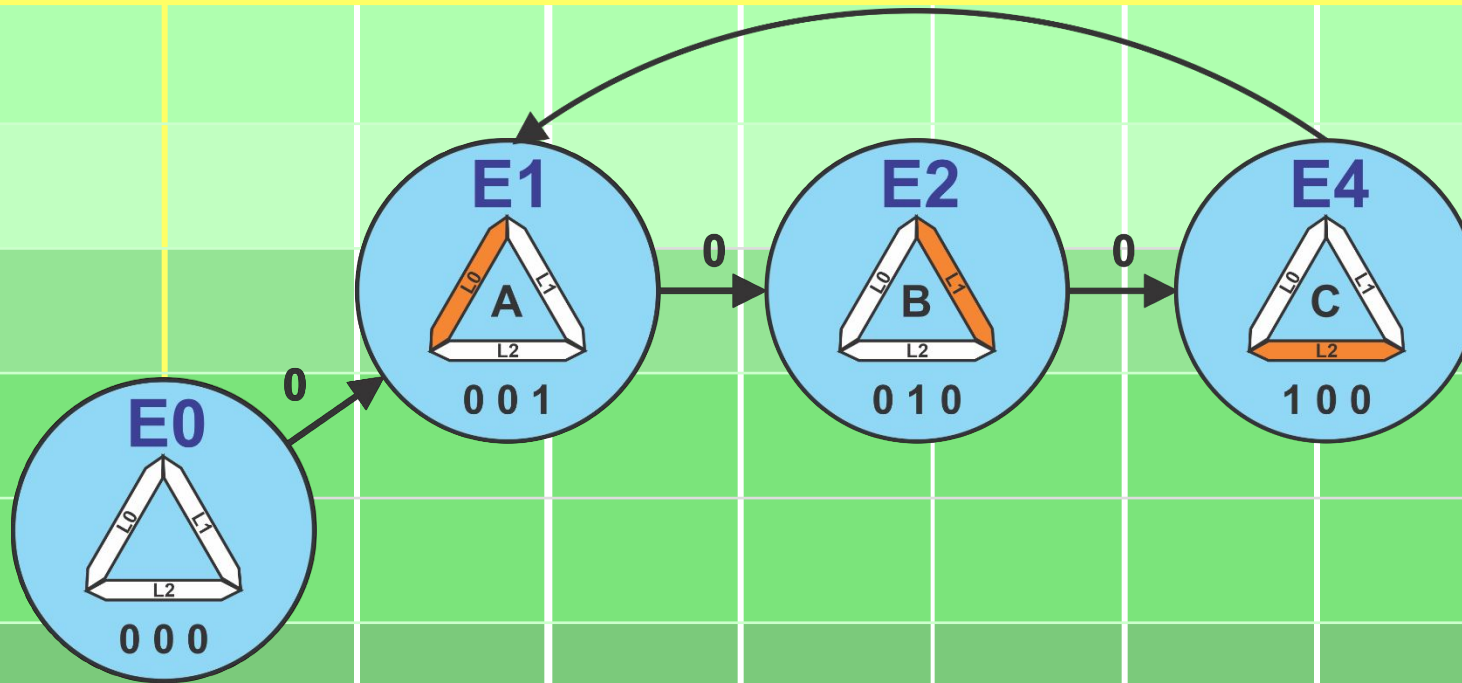
Diagrama de transición



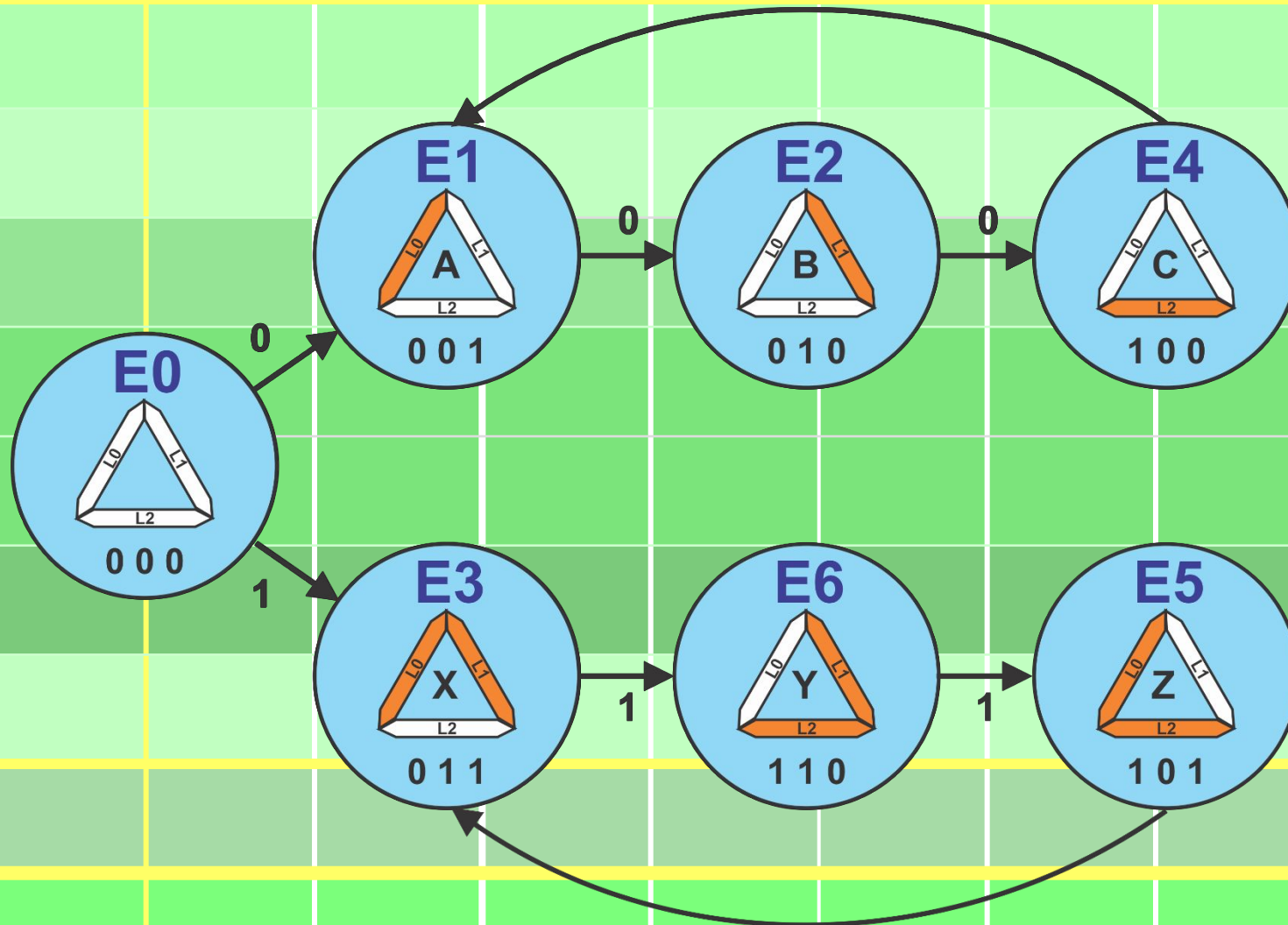
a) Ambas parten de condiciones iniciales en donde todas las lámparas están apagadas.



b) Si $S=0$ Ocurrirá la secuencia Uno (A, B, C, A, B, C etc.)

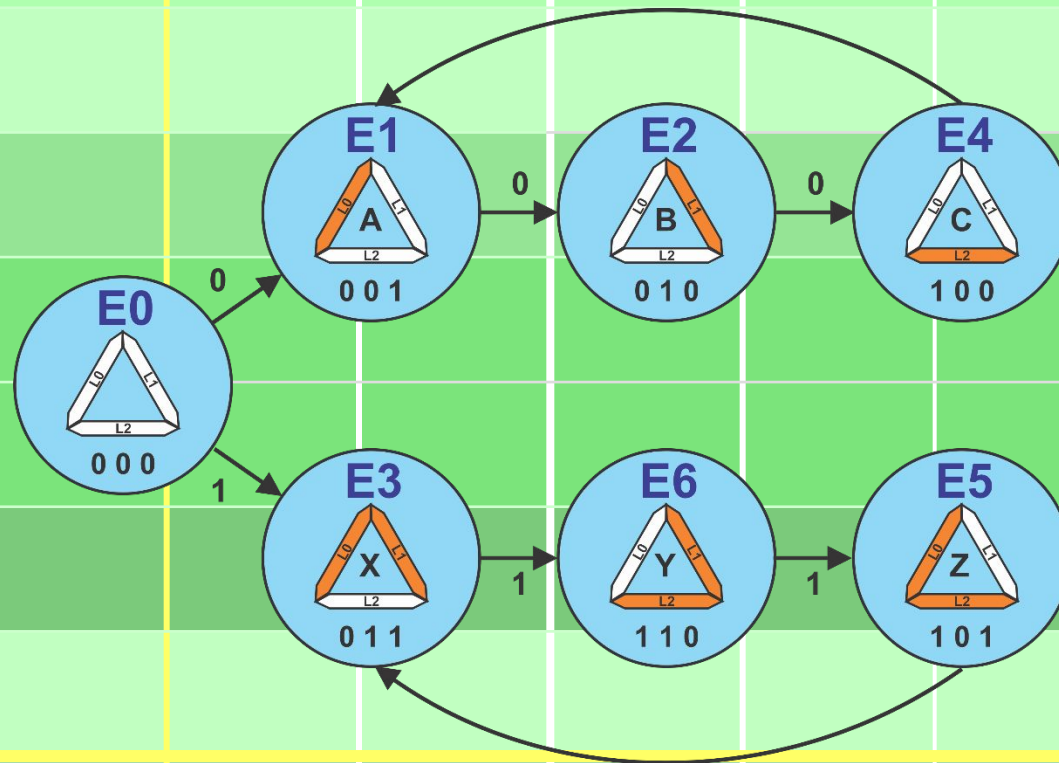


c) Si $S=1$ ocurrirá la secuencia Dos (X, Y, Z, X, Y, Z, etc.)



c) Si $S=1$ ocurrirá la secuencia Dos (X, Y, Z, X, Y, Z, etc.)

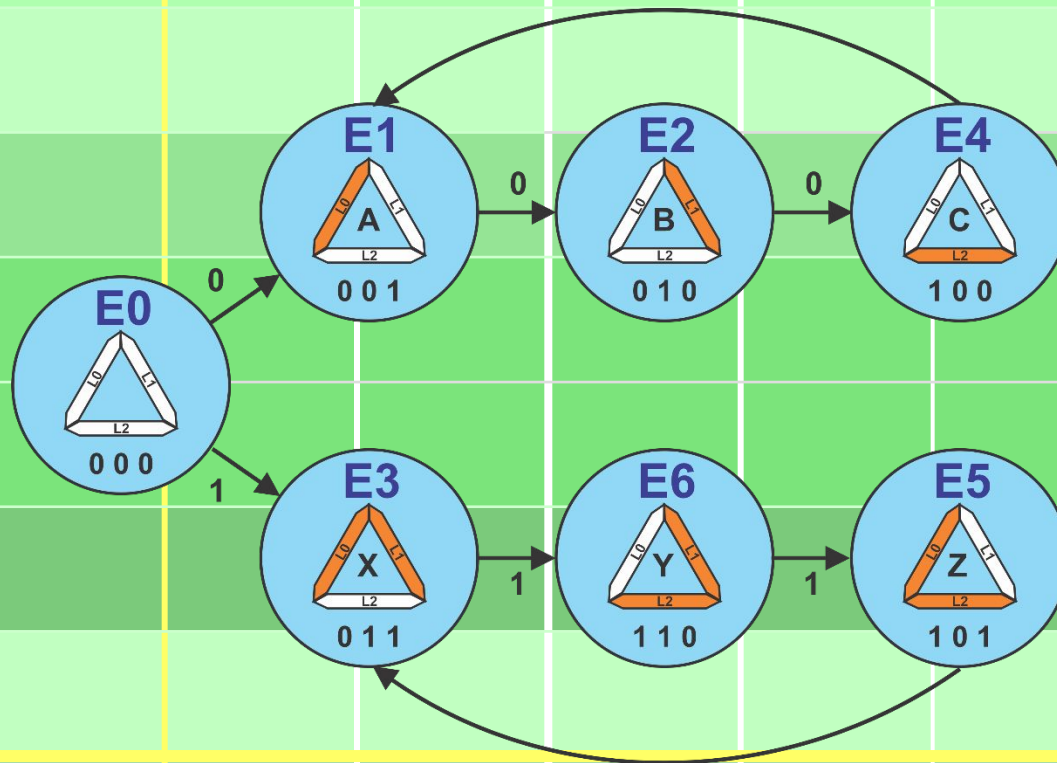
Tabla de estado siguiente



	S=0	S=1
E0	E1	
E1	E2	
E2	E4	
E3		
E4	E1	
E5		
E6		
E7		

c) Si $S=1$ ocurrirá la secuencia Dos (X, Y, Z, X, Y, Z, etc.)

Tabla de estado siguiente



	S=0	S=1
E0	E1	E3
E1	E2	
E2	E4	
E3		E6
E4	E1	
E5		E3
E6		E5
E7		

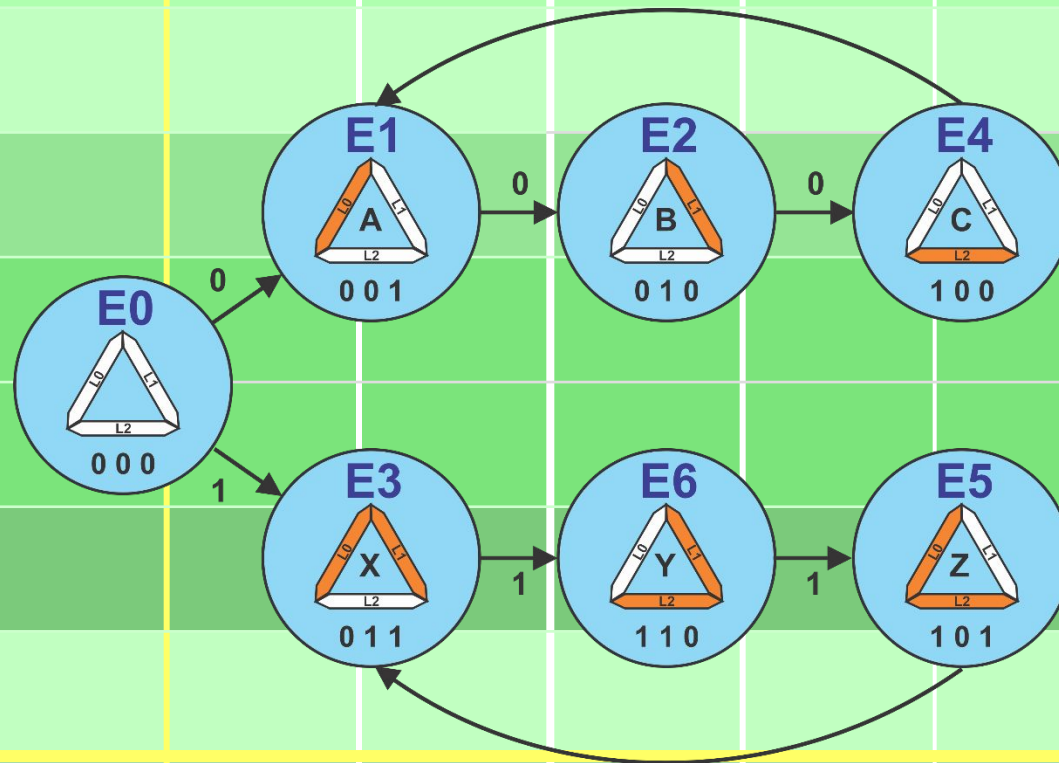
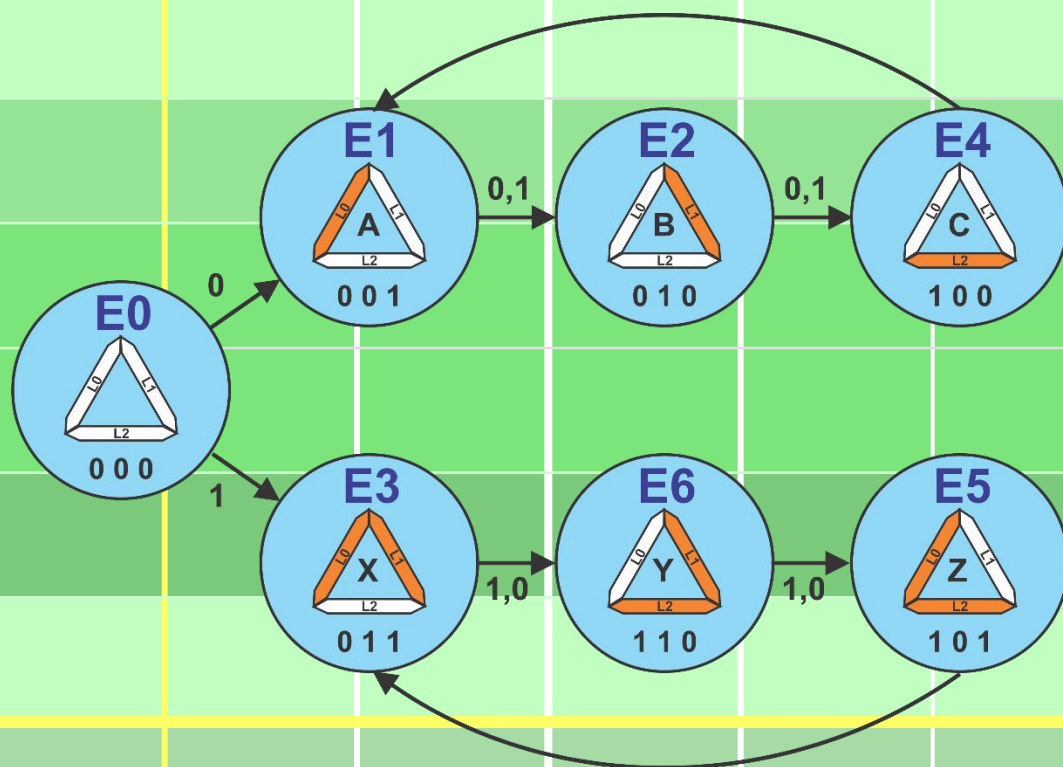


Tabla de estado siguiente

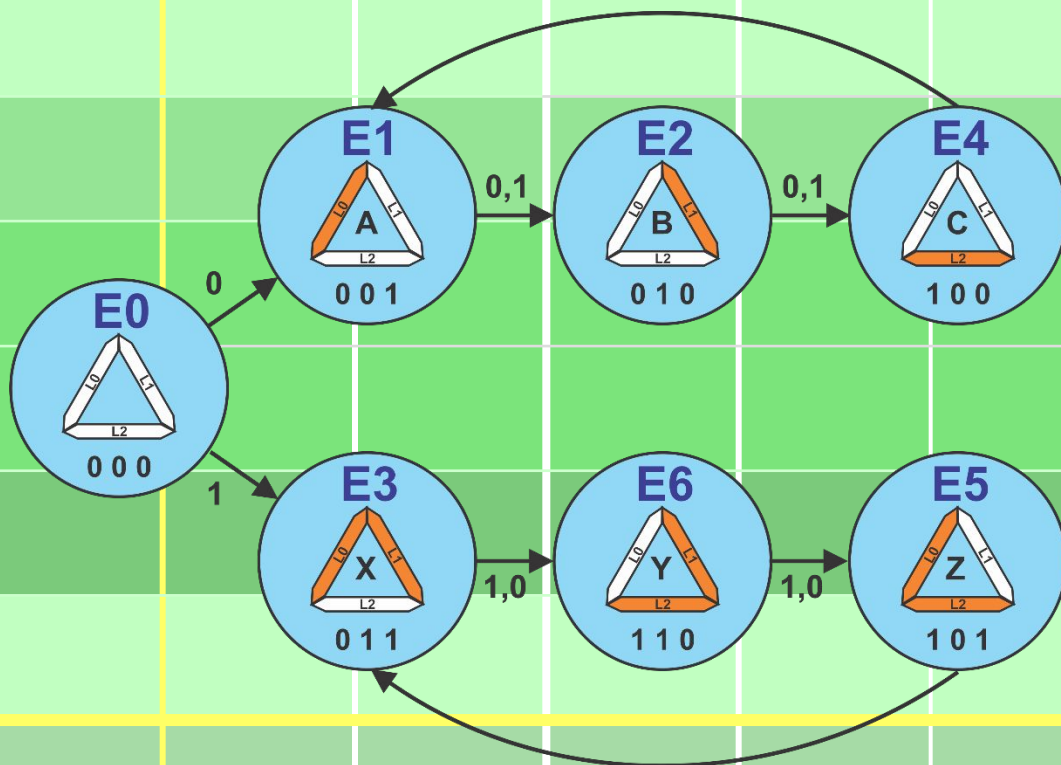
	S=0	S=1
E0	E1	E3
E1	E2	?
E2	E4	?
E3	?	E6
E4	E1	
E5	?	E3
E6	?	E5
E7	?	?

Cuando usted cambia el selector S en medio de una secuencia, las luces *continuarán con la secuencia actual hasta terminar (C, E4 o Z, E5)* y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



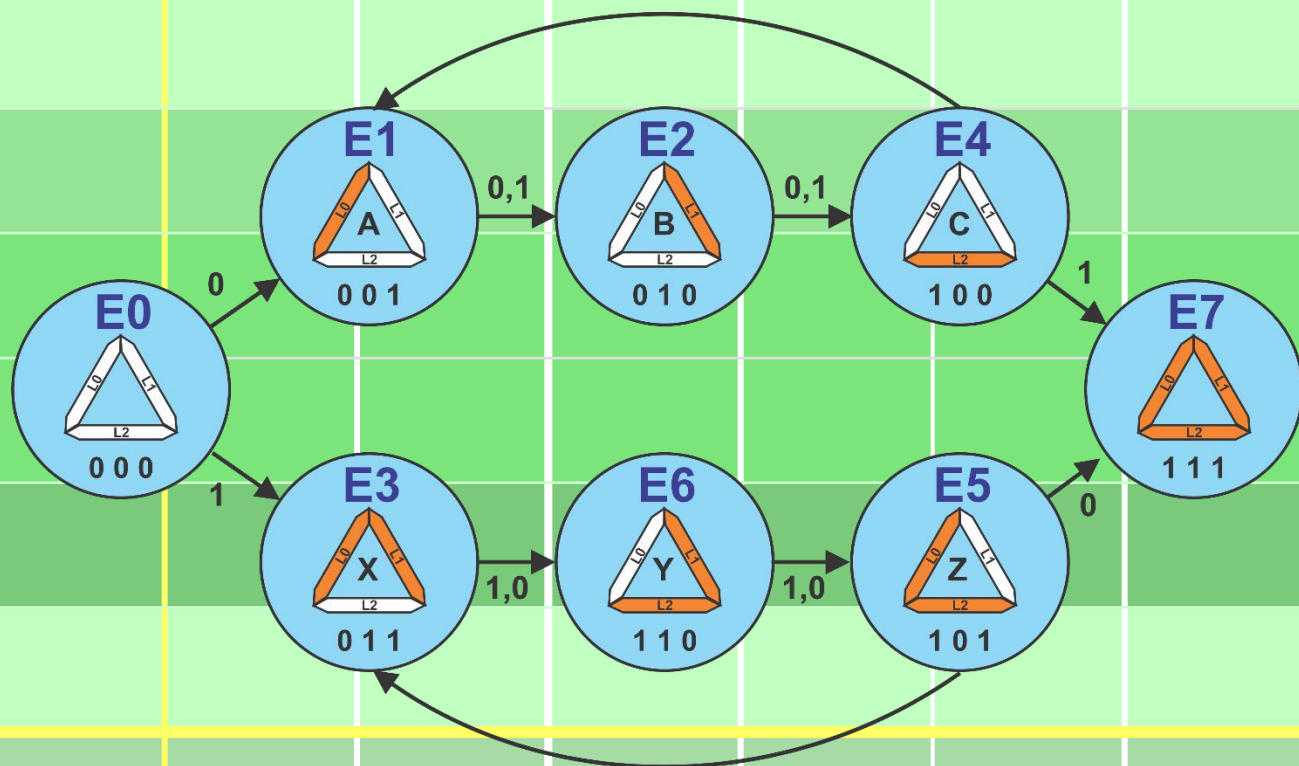
	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3		E6
E4	E1	
E5		E3
E6		E5
E7		

Cuando usted cambia el selector S en medio de una secuencia, las luces *continuarán con la secuencia actual hasta terminar (C, E3 o Z, E6)* y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



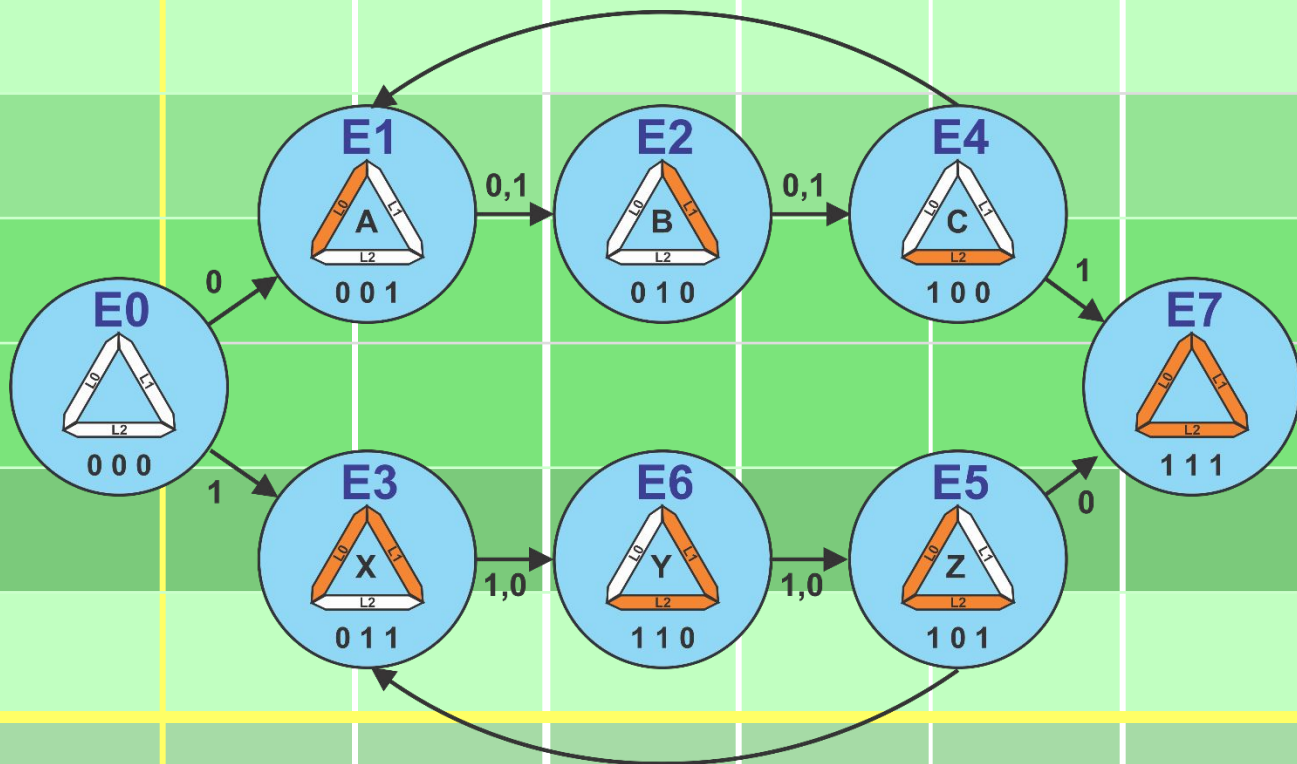
	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	
E5		E3
E6	E5	E5
E7		

Cuando usted cambia el selector S en medio de una secuencia, las luces *continuarán con la secuencia actual hasta terminar (C, E3 o Z, E6)* y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



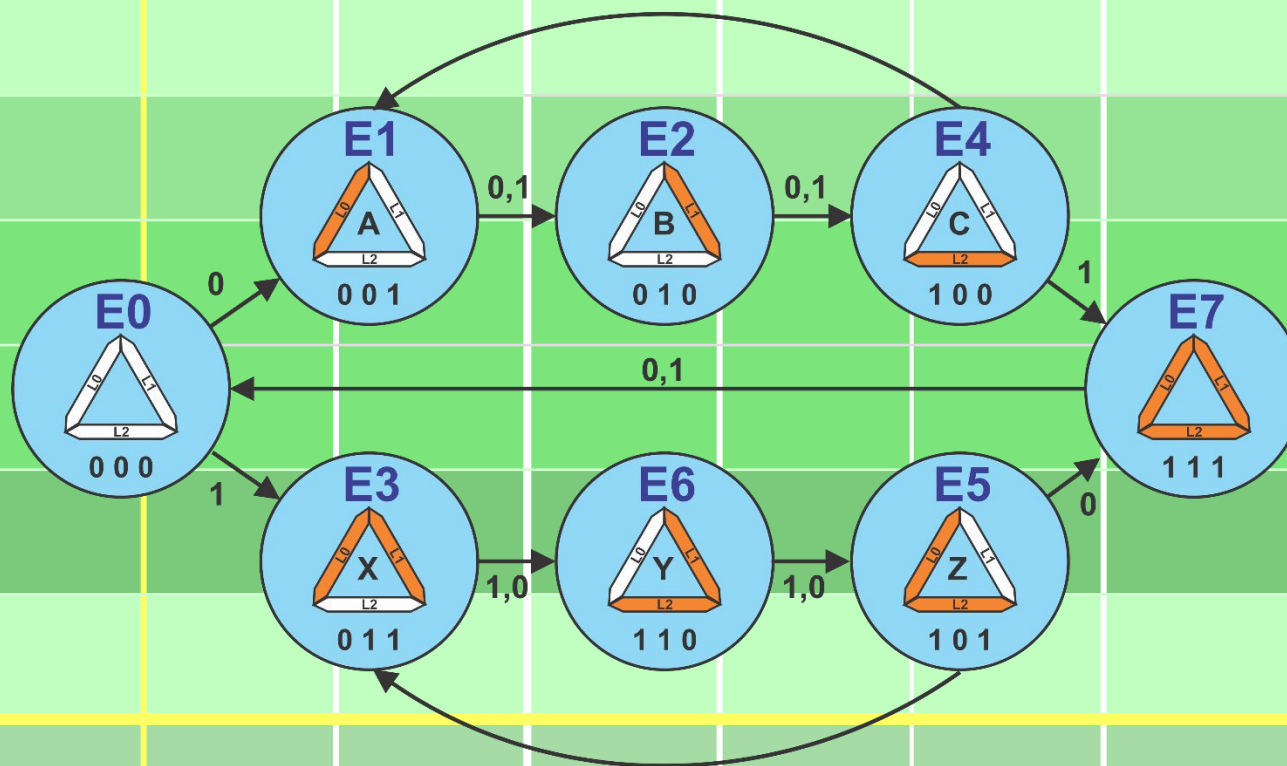
	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7		

Cuando usted cambia el selector S en medio de una secuencia, las luces *continuarán con la secuencia actual hasta terminar (C, E3 o Z, E6)* y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7		

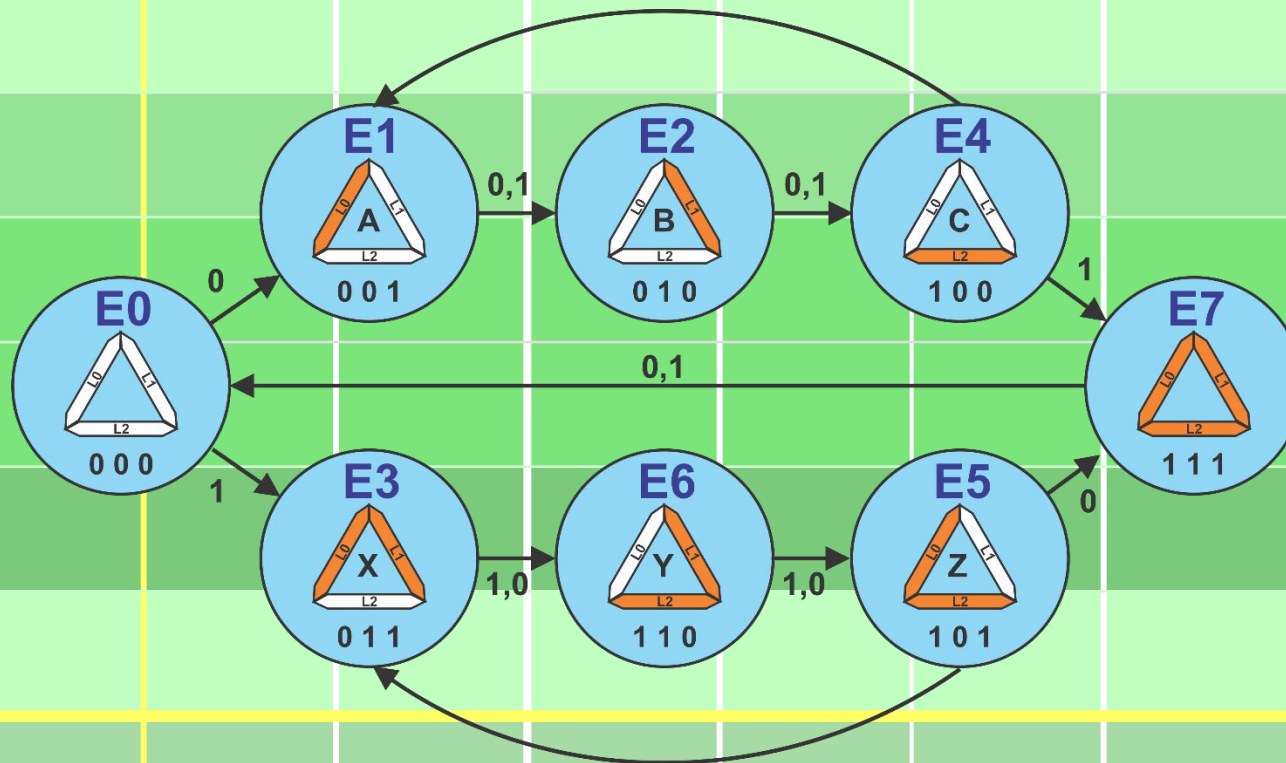
Cuando usted cambia el selector S en medio de una secuencia, las luces *continuarán con la secuencia actual hasta terminar (C, E3 o Z, E6)* y pasará a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuará con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7	E0	E0

¿Cuántos Flip Flops se requieren ?

Tabla de estado siguiente



	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7	E0	E0

3 Flip Flops

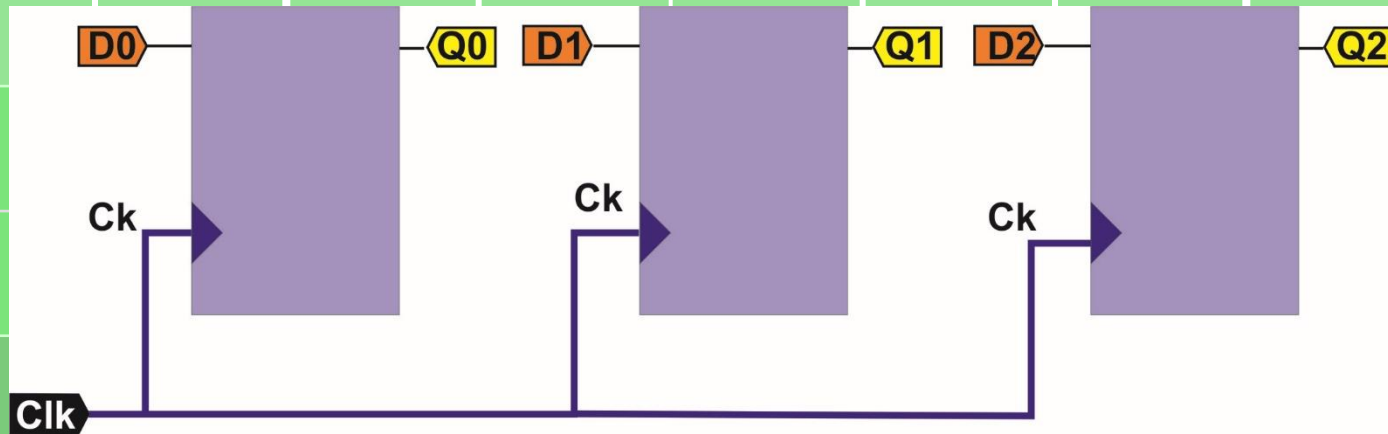
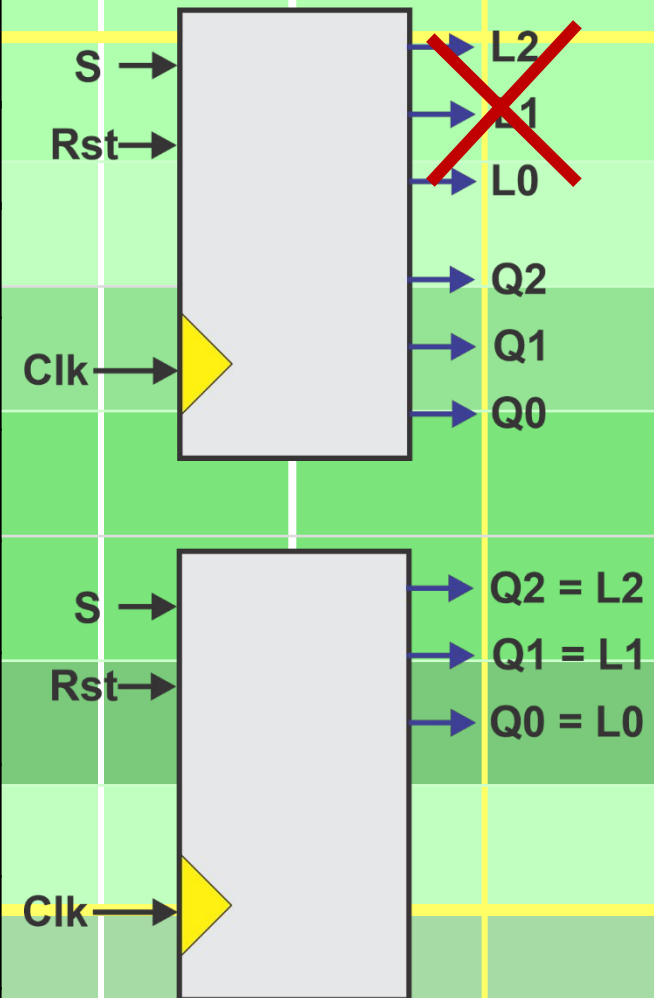


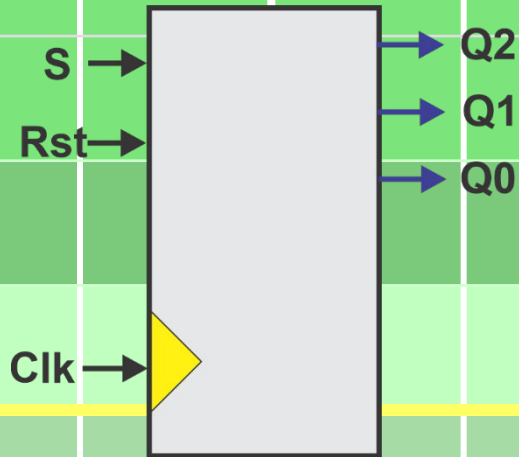
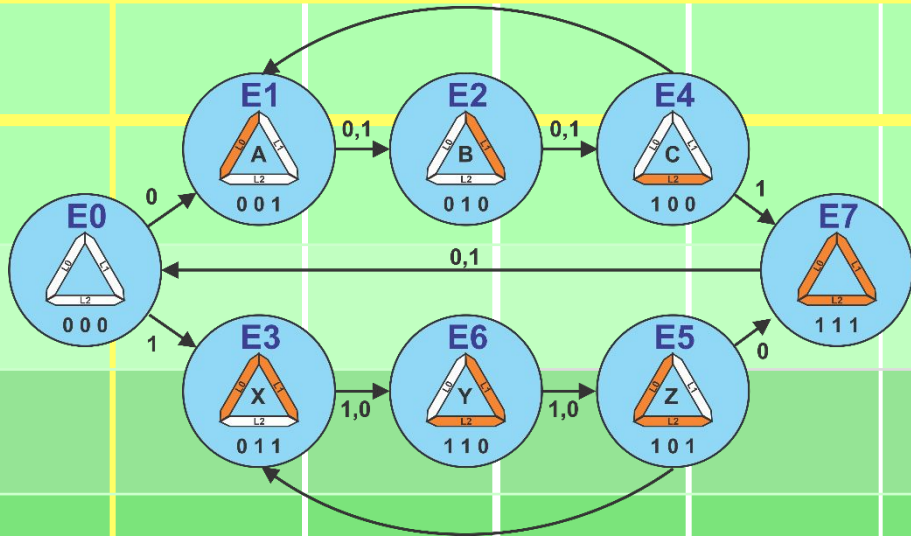
Tabla de estado siguiente

Estado Próximo			Salidas Combinacionales		
	S=0	S=1	L2	L1	L0
E0	E1	E3	0	0	0
E1	E2	E2	0	0	1
E2	E4	E4	0	1	0
E3	E6	E6	0	1	1
E4	E1	E7	1	0	0
E5	E7	E3	1	0	1
E6	E5	E5	1	1	0
E7	E0	E0	1	1	1

Tabla de estado siguiente

	Estado Próximo		Salidas Combinacionales			Salidas Registradas		
	S=0	S=1	L2	L1	L0	Q2	Q1	Q0
E0	E1	E3	0	0	0	0	0	0
E1	E2	E2	0	0	1	0	0	1
E2	E4	E4	0	1	0	0	1	0
E3	E6	E6	0	1	1	0	1	1
E4	E1	E7	1	0	0	1	0	0
E5	E7	E3	1	0	1	1	0	1
E6	E5	E5	1	1	0	1	1	0
E7	E0	E0	1	1	1	1	1	1





	Estado Próximo		Salidas Registradas		
	S=0	S=1	Q2	Q1	Q0
E0	E1	E3	0	0	0
E1	E2	E2	0	0	1
E2	E4	E4	0	1	0
E3	E6	E6	0	1	1
E4	E1	E7	1	0	0
E5	E7	E3	1	0	1
E6	E5	E5	1	1	0
E7	E0	E0	1	1	1

Archivo ABEL-HDL

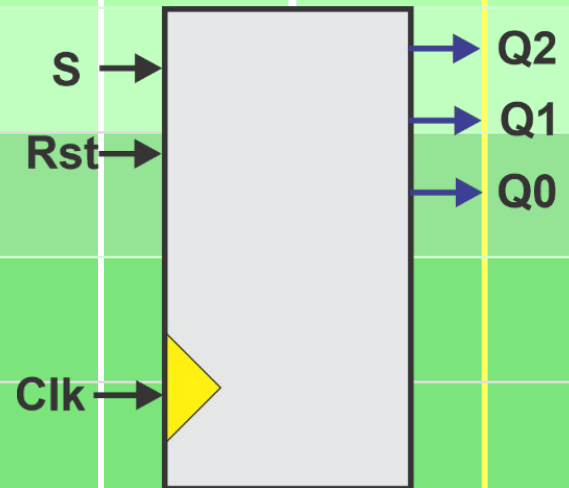
MODULE alerta

"Entrada

Clk,S, Rst Pin 1,2,3;

"salidas Registradas

Q2..Q0 pin 19..17 istype 'reg';



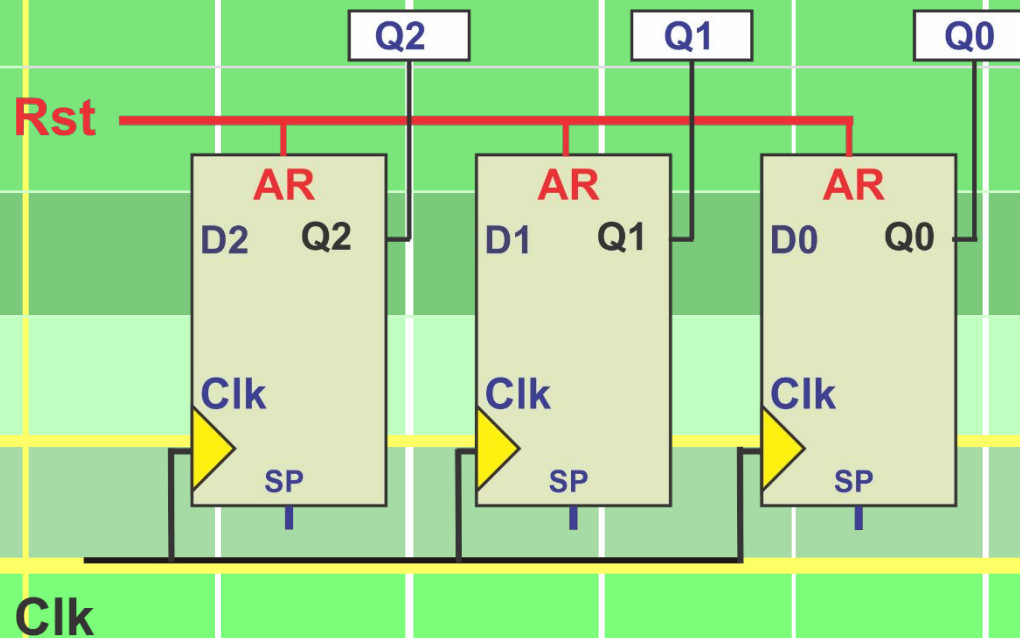
"sincronizacion de los flip flops

Sinc=[Q2..Q0];

Equations

Sinc.Clk=Clk;

Sinc.ar=Rst;



Archivo ABEL-HDL

**"asignación de valores
a los estados**

Declarations

E0=[0,0,0];

E1=[0,0,1];

E2=[0,1,0];

E3=[0,1,1];

E4=[1,0,0];

E5=[1,0,1];

E6=[1,1,0];

E7=[1,1,1];

	L2	L1	L0	N(10)
	Q2	Q1	Q0	
E0	0	0	0	0
E1	0	0	1	1
E2	0	1	0	2
E3	0	1	1	3
E4	1	0	0	4
E5	1	0	1	5
E6	1	1	0	6
E7	1	1	1	7

Archivo ABEL-HDL

State_diagram Sinc

State E0:

If **!S** then E1 else E3;

State E1:

goto E2;

State E2:

goto E4;

State E3:

goto E6;

State E4:

If **!S** then E1 else E7;

State E5:

If **!S** then E7 else E3;

State E6:

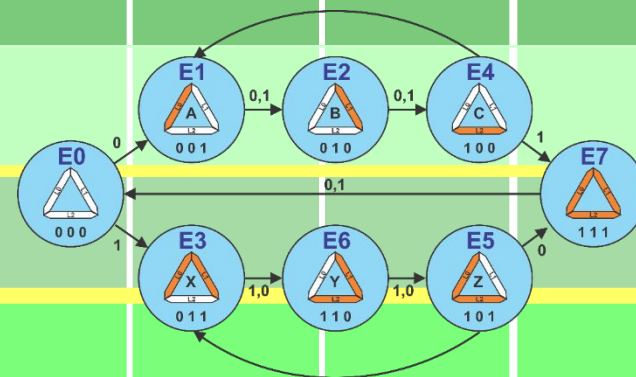
goto E5;

State E7:

goto E0;

Tabla de estado
siguiente

	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7	E0	E0



Programación con Truth_table en modo secuencial :>

Truth_Table

([Entrada, Estado presente]:>[Estado próximo])

Programación con Truth_table

```
MODULE luces
"Entrada
Clk,S Pin 1,2;
"salidas Registradas
Q2..Q0 pin 19..17
istype 'reg';
"sincronizacion
Sinc=[Q2..Q0];
Equations
Sinc.Clk=Clk;
```

No es necesaria
La asignación

Truth_table
([**S**,Q2,Q1,Q0]:>[Q2,Q1,Q0])

[0 , 0,0,0] :> [0,0,1];
[0 , 0,0,1] :> [0,1,0];
[0 , 0,1,0] :> [1,0,0];
[0 , 0,1,1] :> [1,1,0];
[0 , 1,0,0] :> [0,0,1];
[0 , 1,0,1] :> [1,1,1];
[0 , 1,1,0] :> [1,0,1];
[0 , 1,1,1] :> [0,0,0];
[1 , 0,0,0] :> [1,1,0];
[1 , 0,0,1] :> [0,1,0];
[1 , 0,1,0] :> [1,0,0];
[1 , 0,1,1] :> [1,1,0];
[1 , 1,0,0] :> [1,1,1];
[1 , 1,0,1] :> [0,1,1];
[1 , 1,1,0] :> [1,0,1];
[1 , 1,1,1] :> [0,0,0];

Sinc=[Q2..Q0];

	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E3
E6	E5	E5
E7	E0	E0

Programación con Truth_table

```
MODULE luces
"Entrada
Clk,S Pin 1,2;
"salidas Registradas
Q2..Q0 pin 19..17
istype 'reg';
"sincronizacion
Sinc=[Q2..Q0];
Equations
Sinc.Clk=Clk;
```

No es necesaria
La asignación

```
Truth_table
([S,Sinc]:>[Sinc])
[0, 0]:>[1];
[0, 1]:>[2];
[0, 2]:>[4];
[0, 3]:>[6];
[0, 4]:>[1];
[0, 5]:>[7];
[0, 6]:>[3];
[0, 7]:>[0];
[1, 0]:>[3];
[1, 1]:>[2];
[1, 2]:>[4];
[1, 3]:>[5];
[1, 4]:>[7];
[1, 5]:>[3];
[1, 6]:>[5];
[1, 7]:>[0];
```

Sinc=[Q2..Q0];

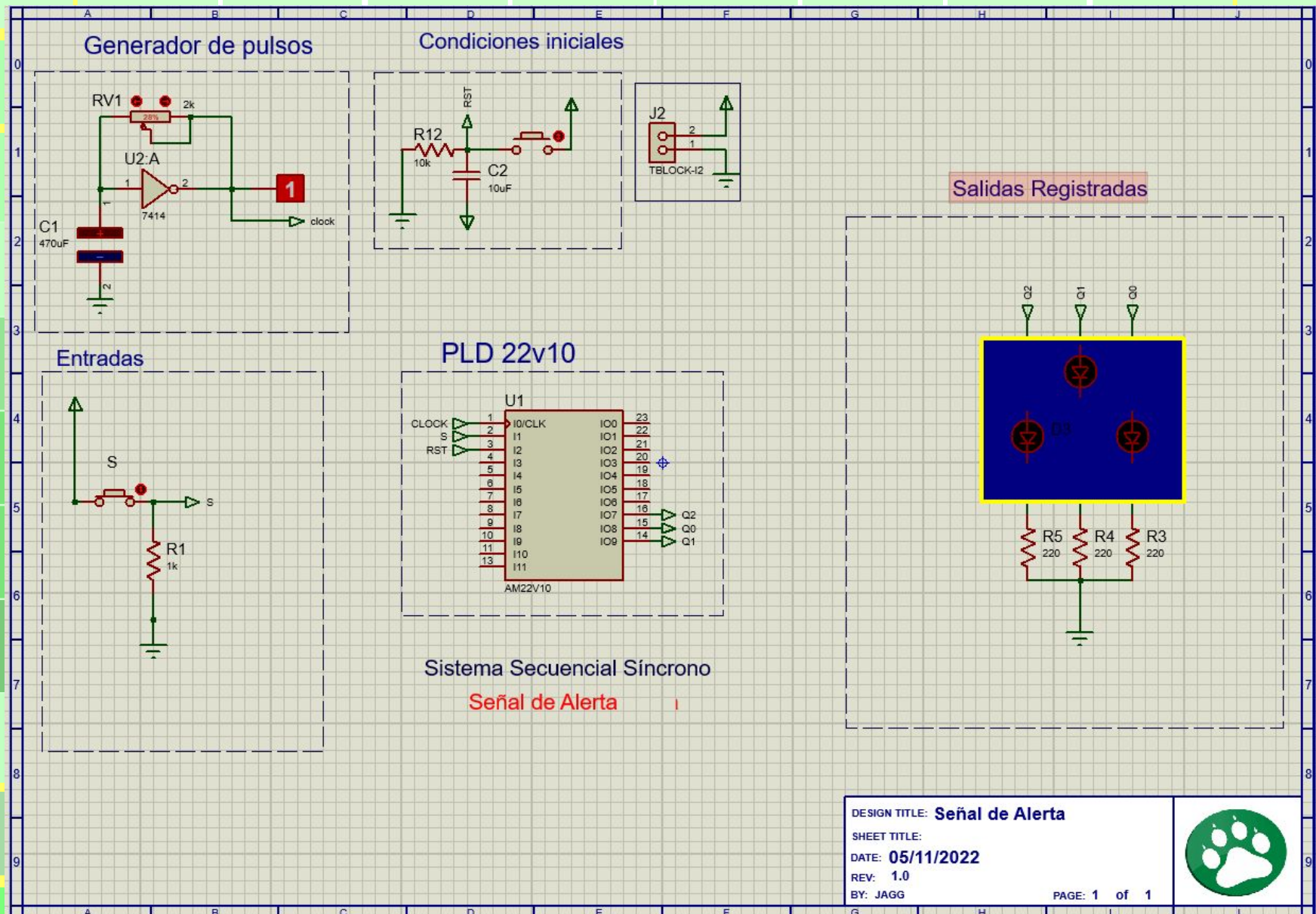
	S=0	S=1
E0	E1	E3
E1	E2	E2
E2	E4	E4
E3	E6	E6
E4	E1	E7
E5	E7	E0
E6	E5	E5
E7	E0	E0

```
MODULE TRI
"5 Nov 2022 ver 1.0
"Luz de Seguridad
"JAGG
"Entradas
Clk,S, Rst Pin 1,2,3;
"salidas Registradas
Q2..Q0 pin 16..14 istype 'reg';

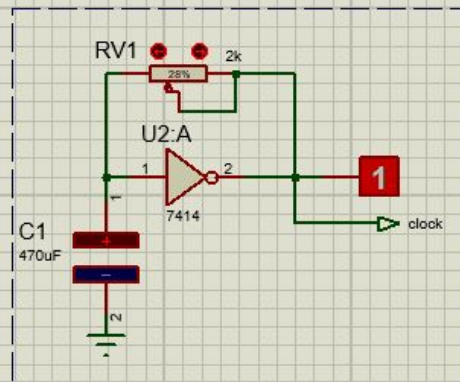
"Sincronizacion de los flip flops y Rst asincrono
Sinc=[Q2..Q0];
Equations
Sinc.Clk=Clk;
Sinc.ar=Rst;
```

"Descripción de la secuencia

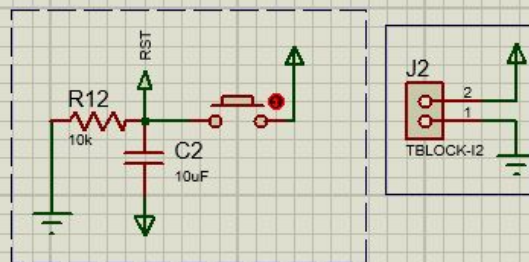
```
Truth_table
([S,Q2,Q1,Q0]:>[Q2,Q1,Q0])
[ 0, 0,0,0]  =>  [0,0,1];
[ 0, 0,0,1]  =>  [0,1,0];
[ 0, 0,1,0]  =>  [1,0,0];
[ 0, 0,1,1]  =>  [1,1,0];
[ 0, 1,0,0]  =>  [0,0,1];
[ 0, 1,0,1]  =>  [1,1,1];
[ 0, 1,1,0]  =>  [1,0,1];
[ 0, 1,1,1]  =>  [0,0,0];
[ 1, 0,0,0]  =>  [1,1,0];
[ 1, 0,0,1]  =>  [0,1,0];
[ 1, 0,1,0]  =>  [1,0,0];
[ 1, 0,1,1]  =>  [1,1,0];
[ 1, 1,0,0]  =>  [1,1,1];
[ 1, 1,0,1]  =>  [0,1,1];
[ 1, 1,1,0]  =>  [1,0,1];
[ 1, 1,1,1]  =>  [0,0,0];
END
```



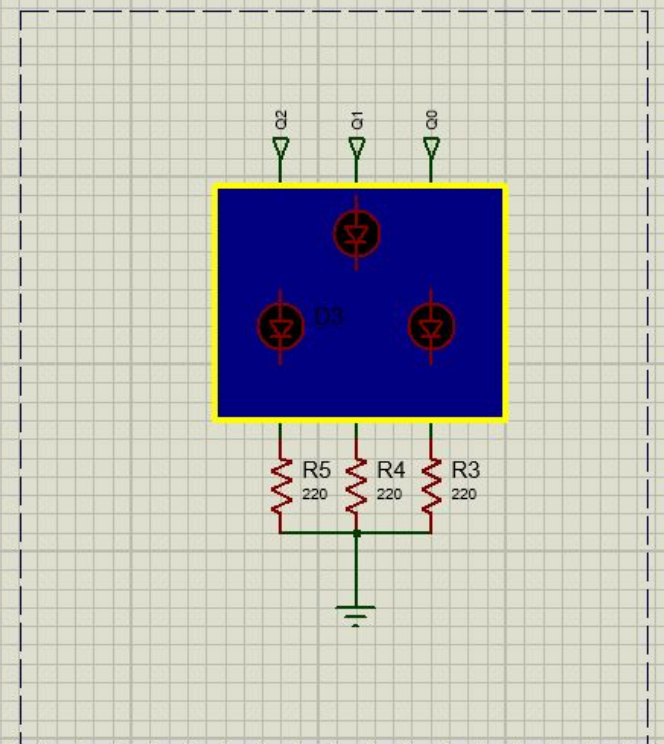
Generador de pulsos



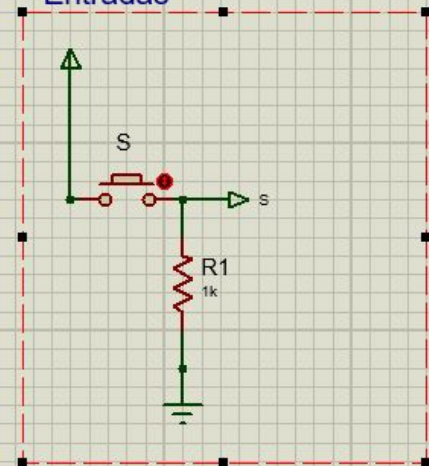
Condiciones iniciales



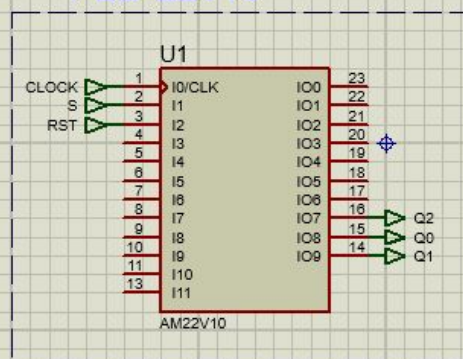
Salidas Registradas



Entradas



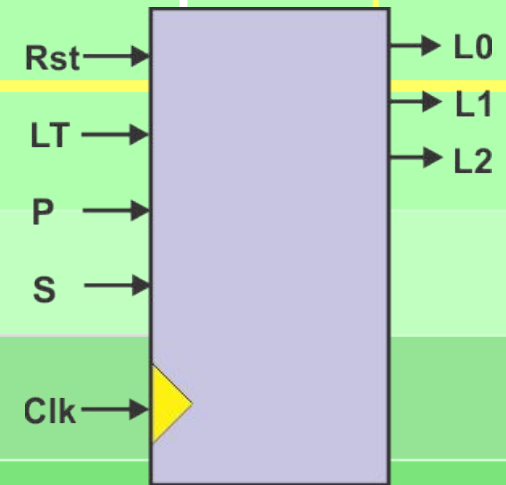
PLD 22v10



Sistema Secuencial Síncrono Señal de Alerta

Al diseño secuencial síncrono anterior incluya 3 entradas:

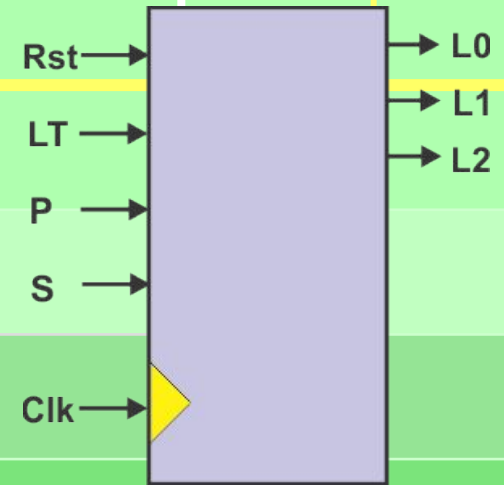
1.- **Paro** que, al oprimirlo sin importar el valor de **S** al llegar la señal de **Clk** el sistema deberá de permanecer en el mismo estado y al soltarlo continuar partiendo del estado actual hacia la secuencia correspondiente al valor de **S**.



Al diseño secuencial síncrono anterior incluya 3 entradas:

1.- **Paro** que, al oprimirlo sin importar el valor de **S** al **llegar la señal de Clk** el sistema deberá de permanecer en el mismo estado y al soltarlo continuar partiendo del estado actual hacia la secuencia correspondiente al valor de **S**.

2.- **LT** (prueba de lampara) que al oprimirlo **al llegar la señal de Clk** se deberán de encender todas las lámparas sin importar el valor de **S** o **P** y al soltarlo regresar al estado en donde están todas las lampara apagadas.

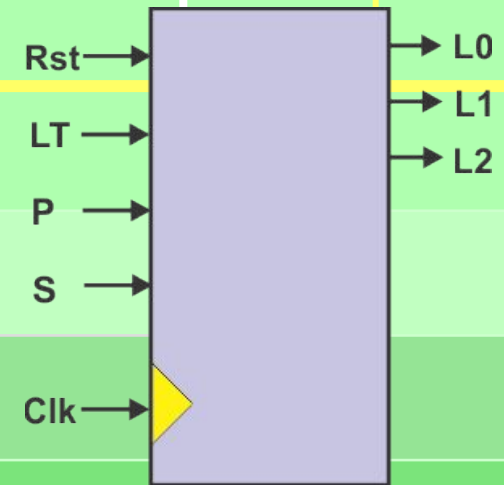


Al diseño secuencial síncrono anterior incluya 3 entradas:

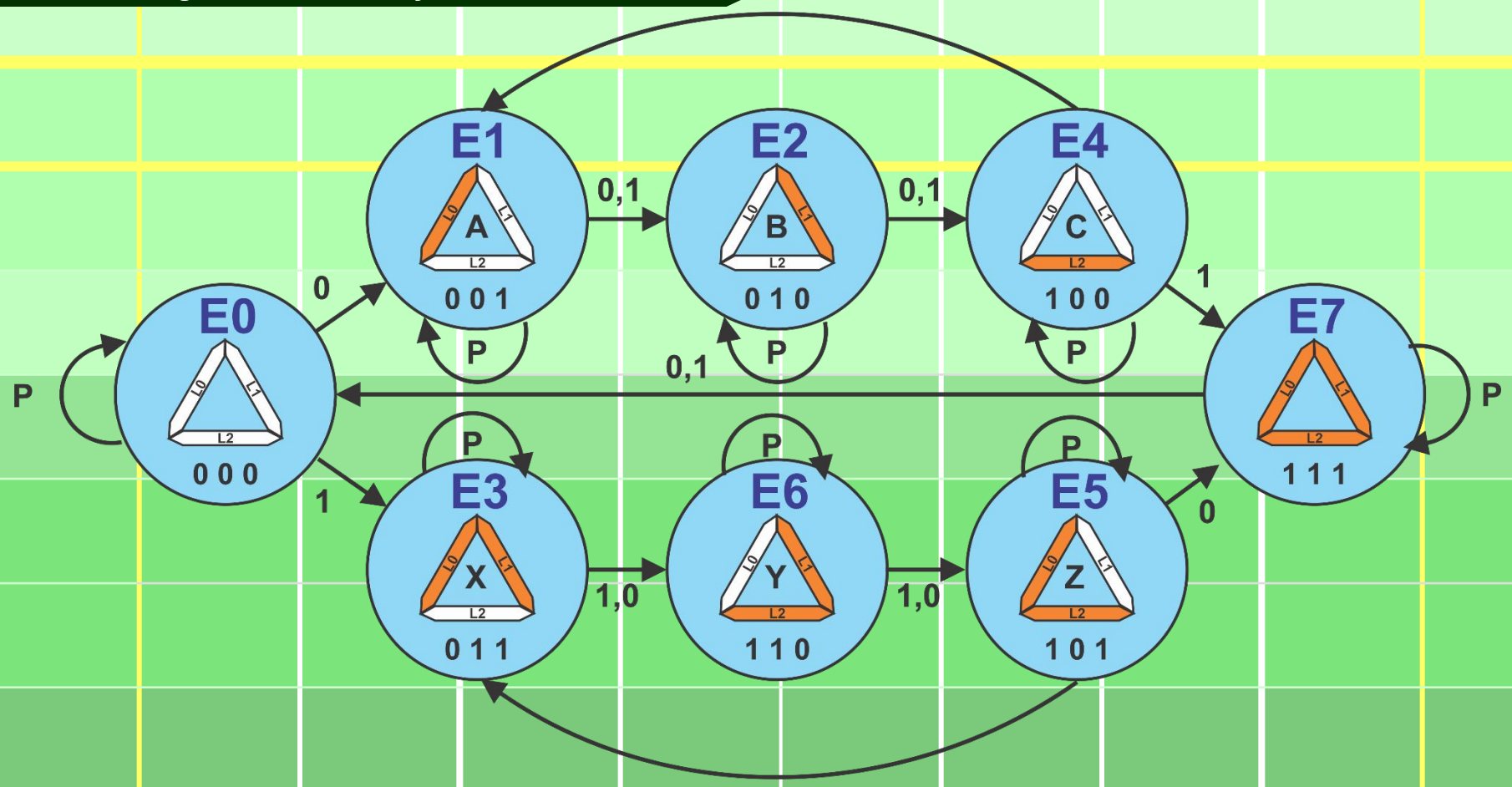
1.- **Paro** que, al oprimirlo sin importar el valor de **S** **al llegar la señal de Clk** el sistema deberá de permanecer en el mismo estado y al soltarlo continuar partiendo del estado actual hacia la secuencia correspondiente al valor de S.

2.- **LT** (prueba de lampara) que al oprimirlo **al llegar la señal de Clk** se deberán de encender todas las lámparas sin importar el valor de **S** o **P** y al soltarlo regresar al estado en donde están todas las lampara apagadas.

3.- **Rst** (Restablecer) de modo que al oprimirlo **al llegar la señal de Clk** **sin importar la condición actual** el sistema regrese al estado en donde están todas las lampara apagadas y al soltarlo continuar con la secuencia correspondiente al valor de S.



**El reset es síncrono
y depende
del pulso de Clk**



Notas: Al oprimir **LT** al llegar la señal de **Clk** se deberán de encender todas las lámparas E7 sin importar el valor de **S** o **P** y al soltarlo regresar al estado en donde están todas las lampara apagadas.

Al oprimir **Rst** al llegar la señal de Clk **sin importar la condición actual** el sistema regrese al estado en donde están todas las lampara apagadas E0.

0  15

Tabla de uiente

EP	Estado Siguiente					Salidas Registradas		
Rst	0	0	0	0	1			
LT	0	0	0	1	X			
P	0	0	1	X	X			
S	0	1	X	X	X	L2	L1	L0
E0						0	0	0
E1						0	0	1
E2						0	1	0
E3						0	1	1
E4						1	0	0
E5						1	0	1
E6						1	1	0
E7						1	1	1

**El reset es síncrono
y depende
del pulso de Clk**

Rst  **S**

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0						0	0	0
E1						0	0	1
E2						0	1	0
E3						0	1	1
E4						1	0	0
E5						1	0	1
E6						1	1	0
E7						1	1	1

**Estableciendo la
prioridad**

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E0					0	0	0
E1	E0					0	0	1
E2	E0					0	1	0
E3	E0					0	1	1
E4	E0					1	0	0
E5	E0					1	0	1
E6	E0					1	1	0
E7	E0					1	1	1

**Estableciendo la
prioridad**

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E0	E7				0	0	0
E1	E0	E7				0	0	1
E2	E0	E7				0	1	0
E3	E0	E7				0	1	1
E4	E0	E7				1	0	0
E5	E0	E7				1	0	1
E6	E0	E7				1	1	0
E7	E0	E7				1	1	1

Estableciendo la prioridad

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E0	E7	E0			0	0	0
E1	E0	E7	E1			0	0	1
E2	E0	E7	E2			0	1	0
E3	E0	E7	E3			0	1	1
E4	E0	E7	E4			1	0	0
E5	E0	E7	E5			1	0	1
E6	E0	E7	E6			1	1	0
E7	E0	E7	E7			1	1	1

Estableciendo la prioridad

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E0	E7	E0	E3	E1	0	0	0
E1	E0	E7	E1	E2	E2	0	0	1
E2	E0	E7	E2	E4	E4	0	1	0
E3	E0	E7	E3	E6	E6	0	1	1
E4	E0	E7	E4	E7	E1	1	0	0
E5	E0	E7	E5	E3	E7	1	0	1
E6	E0	E7	E6	E5	E5	1	1	0
E7	E0	E7	E7	E0	E0	1	1	1

Estableciendo la prioridad

EP	Estado Siguiente							
Rst	1	0	0	0	0			
LT	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E0	E7	E0	E3	E1	0	0	0
E1	E0	E7	E1	E2	E2	0	0	1
E2	E0	E7	E2	E4	E4	0	1	0
E3	E0	E7	E3	E6	E6	0	1	1
E4	E0	E7	E4	E7	E1	1	0	0
E5	E0	E7	E5	E3	E7	1	0	1
E6	E0	E7	E6	E5	E5	1	1	0
E7	E0	E7	E7	E0	E0	1	1	1

Estableciendo la prioridad

Código ABEL-HDL

MODULE TRIPLR

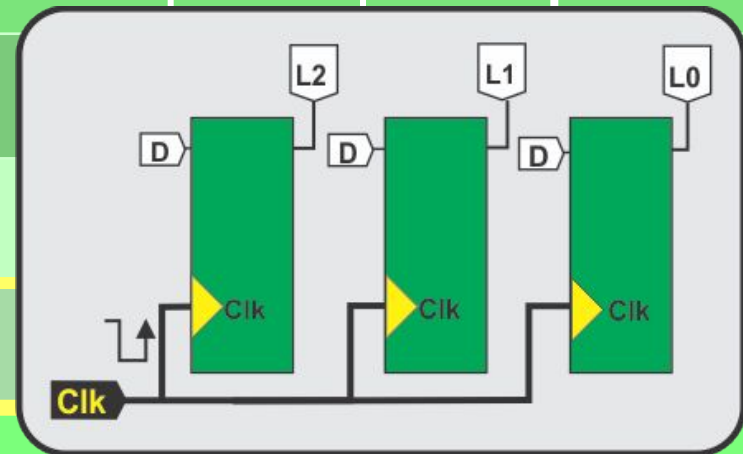
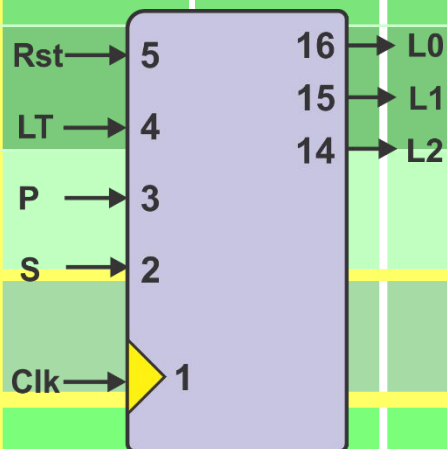
"7 Nov 2022

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

```
L=[L2..L0];  
equations  
L.clk=Clk;
```



Código ABEL-HDL

Declarations

E0=[0,0,0];

E1=[0,0,1];

E2=[0,1,0];

E3=[0,1,1];

E4=[1,0,0];

E5=[1,0,1];

E6=[1,1,0];

E7=[1,1,1];

Código ABEL-HDL

EP	Estado Siguiente				
Rst	0	0	0	0	1
LT	0	0	0	1	X
P	0	0	1	X	X
S	0	1	X	X	X
E0	E1	E3	E0	E7	E0

State_diagram L

State E0:

IF !Rst&!LT&!P&!S then E1;

IF !Rst&!LT&!P&S then E3;

IF !Rst&!LT&P then E0;

IF !Rst & LT then E7;

IF Rst then E0;

Código ABEL-HDL

State_diagram L

State E0:

IF !Rst&!LT&!P&!S then E1;

IF !Rst&!LT&!P&S then E3;

IF !Rst&!LT&P then E0;

IF !Rst & LT then E7;

IF Rst then E0;

E0=[0,0,0];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E0	E0	E7	E0	E3	E1

Else If encadenamiento con Prioridad

State E0:

IF Rst then E0 else if LT then E7 else if P then E0 else if S then E3 else E1;

Código ABEL-HDL

State E1:

IF !Rst&!LT&!P&!S then E2;

IF !Rst&!LT&!P&S then E2;

IF !Rst&!LT&P then E1;

IF !Rst< then E7;

IF Rst then E0;

State E1:

IF !Rst&!LT&!P then E2;

IF !Rst&!LT&P then E1;

IF !Rst< then E7;

IF Rst then E0;

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E1	E0	E7	E1	E2	E2

E1=[0,0,1];

State E1:

IF Rst then E0 else if LT then E7 else if P then E1 else E2;

Código ABEL-HDL

State E2:

IF !Rst&!LT&!P then E4;

IF !Rst&!LT&P then E2;

IF !Rst< then E7;

IF Rst then E0;

E2=[0,1,0];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E2	E0	E7	E2	E4	E4

State E2:

IF Rst then E0 else if LT then E7 else if P then E2 else E4;

Código ABEL-HDL

State E3:

IF !Rst&!LT&!P then E6;

IF !Rst&!LT&P then E3;

IF !Rst< then E7;

IF Rst then E0;

E3=[0,1,1];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E3	E0	E7	E3	E6	E6

State E3:

IF Rst *then* E0 *else if* LT *then* E7 *else if* P *then* E3 *else* E6;

Código ABEL-HDL

State E4:

IF !Rst&!LT&!P&!S then E1;

IF !Rst&!LT&!P&S then E7;

IF !Rst&!LT&P then E4;

IF !Rst< then E7;

IF Rst then E0;

E4=[1,0,0];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E4	E0	E7	E4	E7	E1

State E4:

IF Rst then E0 else if LT then E7 else if P then E4 else if S Then E7 else E1;

Código ABEL-HDL

State E5:

IF !Rst&!LT&!P&!S then E7;

IF !Rst&!LT&!P&S then E3;

IF !Rst&!LT&P then E5;

IF !Rst< then E7;

IF Rst then E0;

E5=[1,0,1];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E5	E0	E7	E5	E3	E7

State E5:

IF Rst *then* E0 *else if* LT *then* E7 *else if* P *then* E5 *else if* S *Then* E3 *else* E7;

Código ABEL-HDL

State E6:

IF !Rst&!LT&!P then E5;

IF !Rst&!LT&P then E6;

IF !Rst< then E7;

IF Rst then E0;

E6=[1,1,0];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E0	E0	E7	E6	E5	E5

State E6:

IF Rst then E0 else if LT then E7 else if P then E6 else E5;

Código ABEL-HDL

State E7:

IF !Rst&!LT&!P then E0;

IF !Rst&!LT&P then E7;

IF !Rst< then E7;

IF Rst then E0;

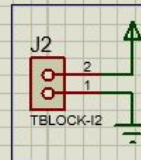
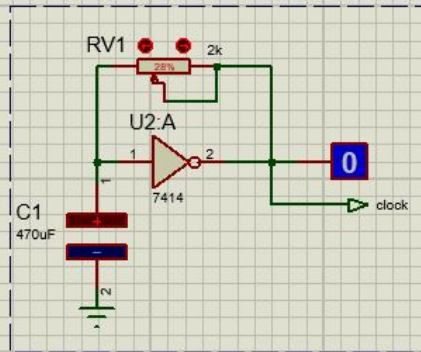
E7=[1,1,1];

EP	Estado Siguiente				
Rst	1	0	0	0	0
LT	X	1	0	0	0
P	X	X	1	0	0
S	X	X	X	1	0
E7	E0	E7	E7	E0	E0

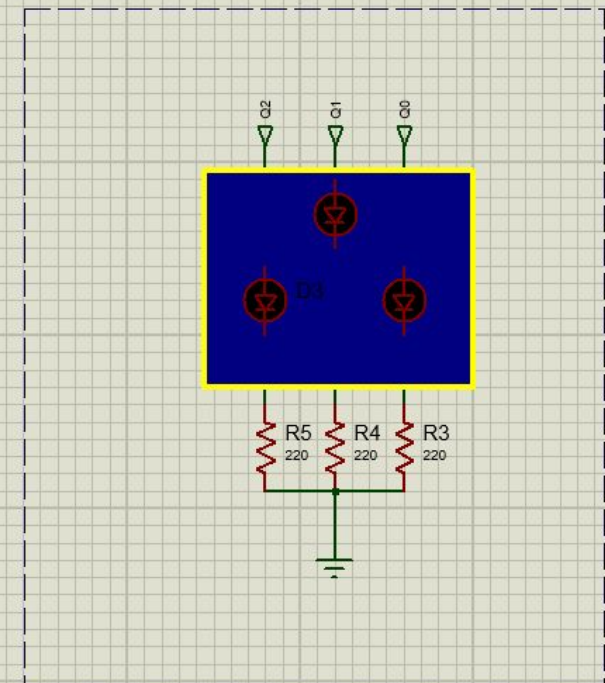
State E7:

IF Rst then E0 else if LT then E7 else if P then E7 else E0;

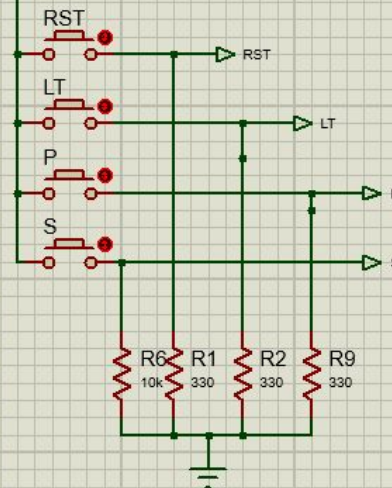
Generador de pulsos



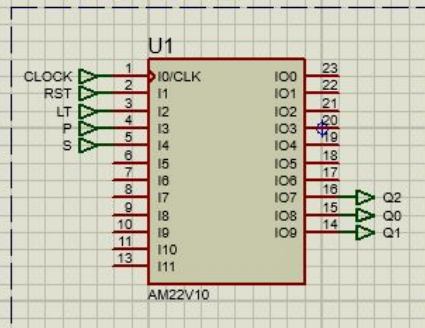
Salidas Registradas



Entradas



PLD 22v10



Sistema Secuencial Síncrono Señal de Alerta

DESIGN TITLE: **Señal de Alerta**

SHEET TITLE:

DATE: **05/11/2022**

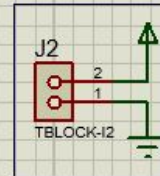
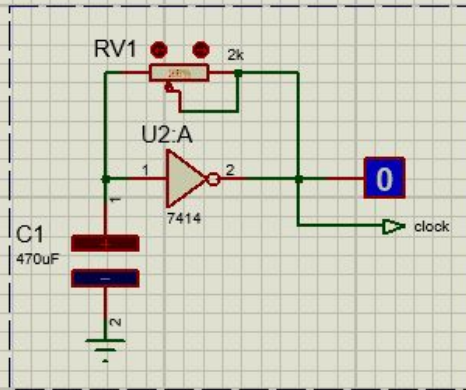
REV: **1.0**

BY: **JAGG**

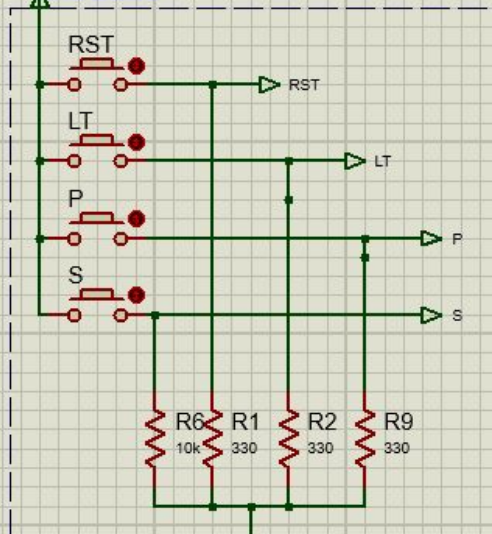
PAGE: **1** of **1**



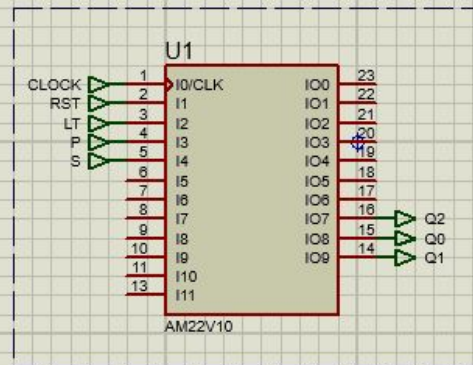
Generador de pulsos



Entradas

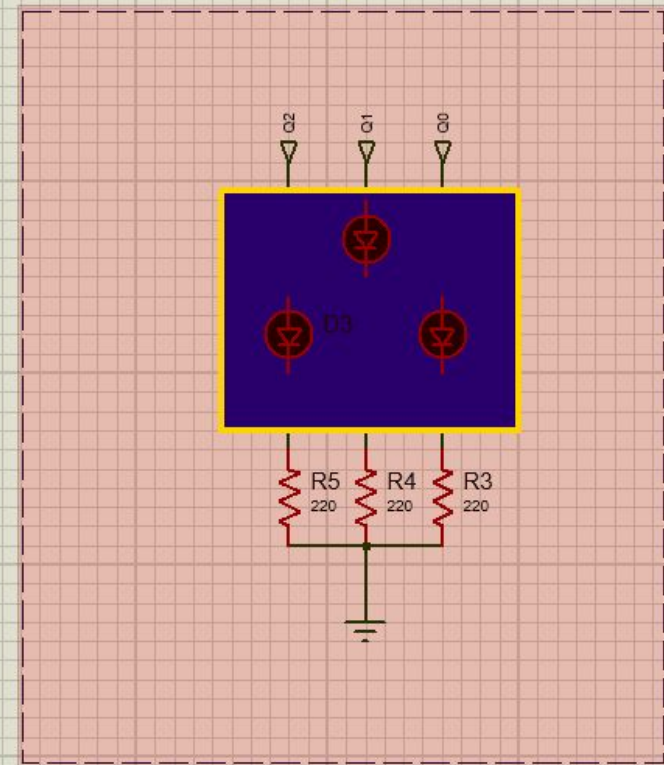


PLD 22v10



Sistema Secuencial Síncrono
Señal de Alerta

Salidas Registradas



Truth_Table

MODULE TRIPLRTT

"17 Nov 2020

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

X=.x.;

L=[L2..L0];

equations

L.clk=Clk;

Rst,
LT

Truth_Table

([Rst,LT,P,S,L]:>L)

[1,X,X,X,X]:>0;

[0,1,X,X,X]:>7;

Truth_Table

P=1

[0,0,1,X,0]:>0;
[0,0,1,X,1]:>1;
[0,0,1,X,2]:>2;
[0,0,1,X,3]:>3;
[0,0,1,X,4]:>4;
[0,0,1,X,5]:>5;
[0,0,1,X,6]:>6;
[0,0,1,X,7]:>7;

S=0

[0,0,0,0,0]:>1;
[0,0,0,0,1]:>2;
[0,0,0,0,2]:>4;
[0,0,0,0,3]:>6;
[0,0,0,0,4]:>1;
[0,0,0,0,5]:>7;
[0,0,0,0,6]:>5;
[0,0,0,0,7]:>0;

S=1

[0,0,0,1,0]:>3;
[0,0,0,1,1]:>2;
[0,0,0,1,2]:>4;
[0,0,0,1,3]:>6;
[0,0,0,1,4]:>7;
[0,0,0,1,5]:>3;
[0,0,0,1,6]:>5;
[0,0,0,1,7]:>0;

3.- **Rst** (Restablecer) de modo que al oprimirlo **sin importar la condición actual** el sistema regrese al estado en donde están todas las lampara apagadas y al soltarlo continuar con la secuencia correspondiente al valor de S.

EP	Estado Siguiente						
LT	0	0	0	1	Salidas Registradas		
P	0	0	1	X			
S	0	1	X	X			
					L2	L1	L0
E0	E1	E3	E0	E7	0	0	0
E1	E2	E2	E1	E7	0	0	1
E2	E4	E4	E2	E7	0	1	0
E3	E6	E6	E3	E7	1	0	0
E4	E1	E7	E4	E7	1	1	0
E5	E7	E0	E5	E7	0	1	1
E6	E5	E5	E6	E7	1	0	1
E7	E0	E0	E7	E7	1	1	1

El reset es asíncrono y no depende del pulso de Clk

Código ABEL-HDL

MODULE TRIPLR

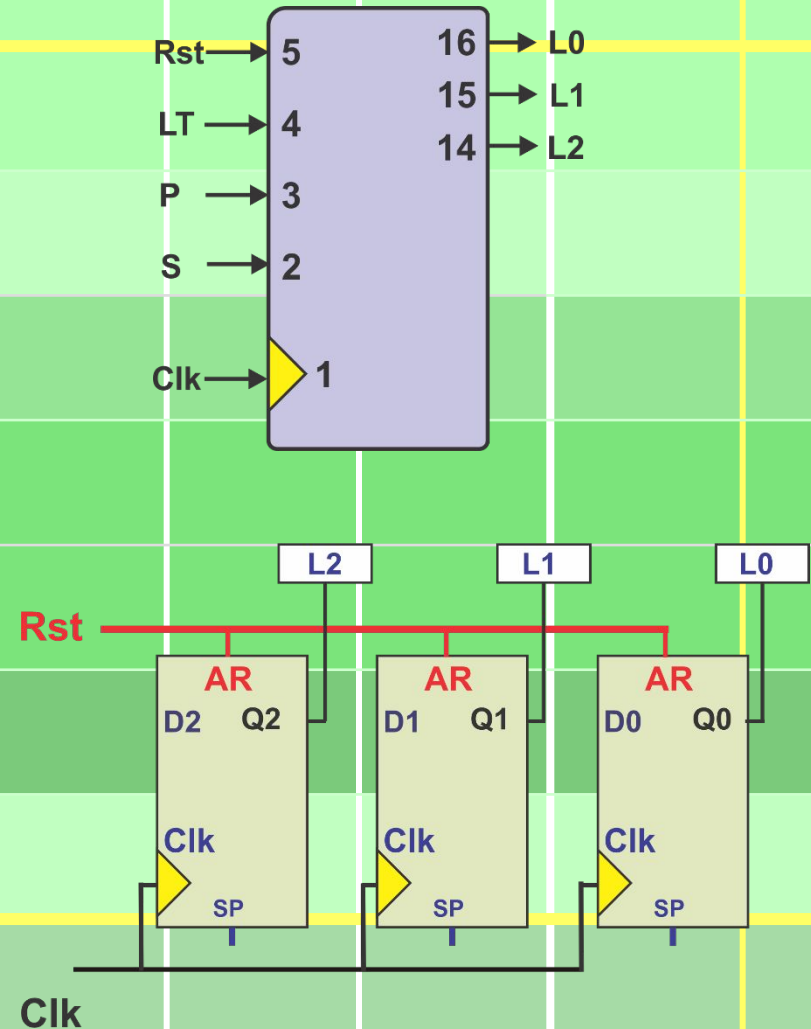
"17 Nov 2020

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

```
L=[L2..L0];  
equations  
L.clk=Clk;  
L.ar = Rst;
```



Código ABEL-HDL

Declarations

E0=[0,0,0];

E1=[0,0,1];

E2=[0,1,0];

E3=[0,1,1];

E4=[1,0,0];

E5=[1,0,1];

E6=[1,1,0];

E7=[1,1,1];

Código ABEL-HDL

State_diagram L

State E0:

IF !LT&!P&!S then E1;

IF !LT&!P&S then E4;

IF !LT&P then E0;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E0	E7	E0	E3	E1

E0=[0,0,0];

Else If con Prioridad

State E0:

IF LT then E7 *else if* P then E0 *else if* S then E3 *else* E1;

Código ABEL-HDL

State E1:

IF **!LT**&**!P** then E2;

IF **!LT**&P then E1;

IF LT then E7;

E1=[0,0,1];

State E1:

IF LT **then** E7 **else if** P **then** E1 **else** E2;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E1	E7	E1	E2	E2

Código ABEL-HDL

State E2:

IF !LT&!P then E4;

IF !LT&P then E2;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E2	E7	E2	E4	E4

E2=[0,1,0];

State E2:

IF LT then E7 else if P then E2 else E4;

Código ABEL-HDL

State E3:

IF **LT**&**!P** then E6;

IF **!LT**&**P** then E3;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E3	E7	E3	E6	E6

E3=[0,1,1];

State E3:

IF LT then E7 **else if** P **then** E3 **else** E6;

Código ABEL-HDL

State E4:

IF **LT**&**!P**&**!S** then E1;

IF **!LT**&**!P**&**S** then E7;

IF **!LT**&**P** then E4;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E4	E7	E4	E7	E1

E4=[1,0,0];

State E4:

IF LT **then** E7 **else if** P **then** E4 **else if** S **Then** **E7** **else** E1;

Código ABEL-HDL

State E5:

IF !LT&!P&!S then E7;

IF !LT&!P&S then E3;

IF !LT&P then E5;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E5	E7	E5	E3	E7

E5=[1,0,1];

State E5:

IF LT *then* E7 *else if* P *then* E5 *else if* S *Then* E3 *else* E7;

Código ABEL-HDL

State E6:

IF !LT&!P then E5;

IF !LT&P then E6;

IF LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E0	E7	E6	E5	E5

E6=[1,1,0];

State E6:

IF LT *then* E7 *else if* P *then* E6 *else* E5;

Código ABEL-HDL

State E7:

IF !LT&!P&!S then E0;

IF !LT&!P&S then E0;

IF !LT&P then E7;

IF !LT then E7;

EP				
LT	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E7	E7	E7	E0	E0

E7=[1,1,1];

State E7:

IF LT *then* E7 *else if* P *then* E7 *else* E0;

Truth_Table

MODULE TRIPLRTT

"17 Nov 2020

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

X=.x.;

L=[L2..L0];

equations

L.clk=Clk;

L.ar=Rst;

Truth_Table

([LT,P,S,L]:>L)

[**1**,X,X,X]:>7;

L

T

Truth_Table

P=1

[0,1,X,0]:>0;
[0,1,X,1]:>1;
[0,1,X,2]:>2;
[0,1,X,3]:>3;
[0,1,X,4]:>4;
[0,1,X,5]:>5;
[0,1,X,6]:>6;
[0,1,X,7]:>7;

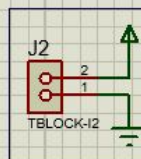
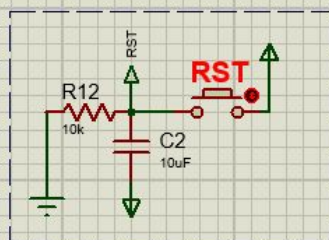
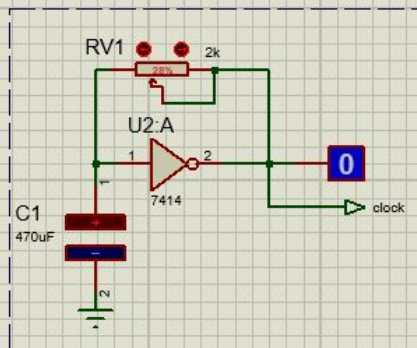
S=0

[0,0,0,0]:>1;
[0,0,0,1]:>2;
[0,0,0,2]:>4;
[0,0,0,3]:>6;
[0,0,0,4]:>1;
[0,0,0,5]:>7;
[0,0,0,6]:>5;
[0,0,0,7]:>0;

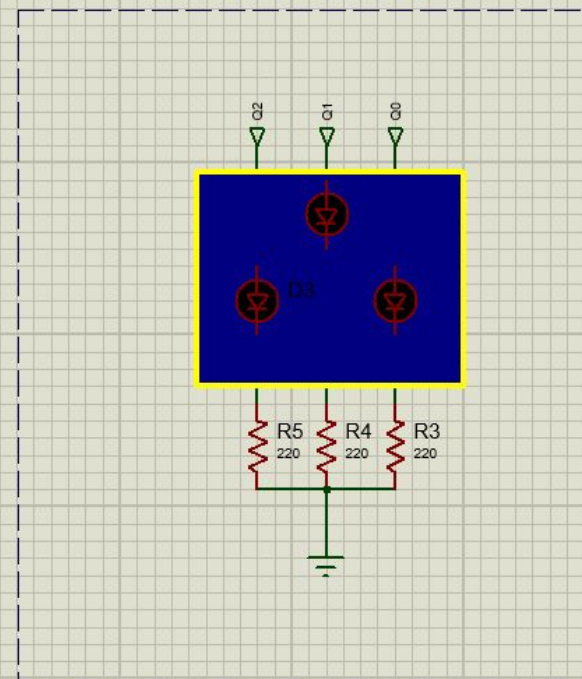
S=1

[0,0,1,0]:>3;
[0,0,1,1]:>2;
[0,0,1,2]:>4;
[0,0,1,3]:>6;
[0,0,1,4]:>7;
[0,0,1,5]:>3;
[0,0,1,6]:>5;
[0,0,1,7]:>0;

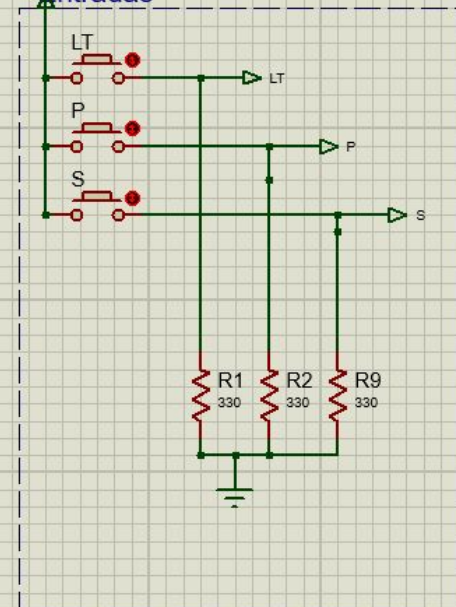
Generador de pulsos



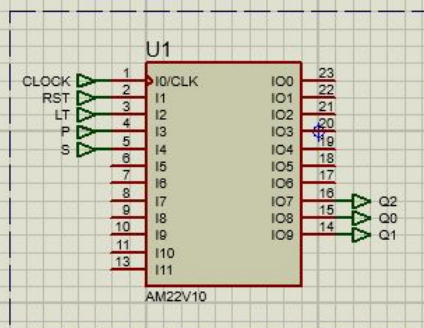
Salidas Registradas



Entradas



PLD 22v10



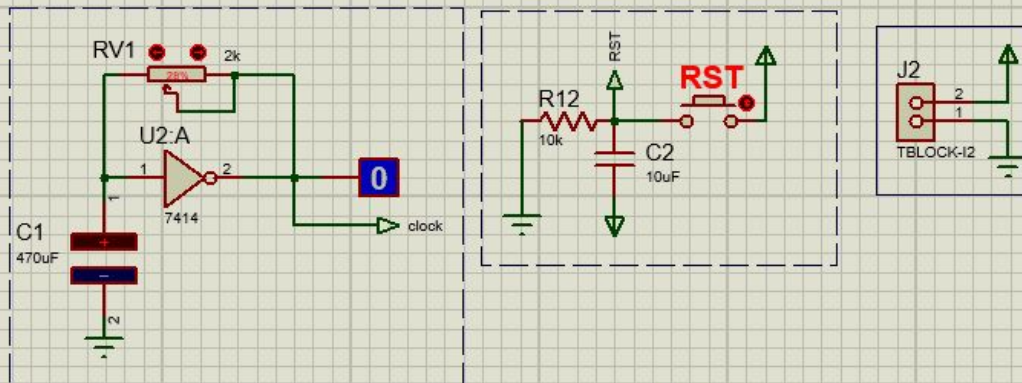
Sistema Secuencial Síncrono Señal de Alerta

DESIGN TITLE: **Señal de Alerta Rst asincron**
SHEET TITLE:
DATE: **06/11/2022**
REV: 1.0
BY: JAGG

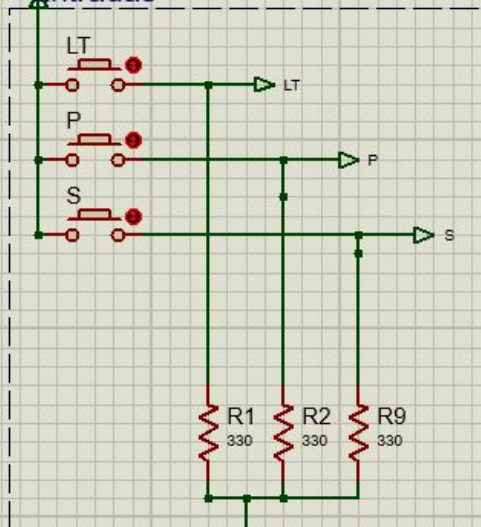
PAGE: 1 of 1



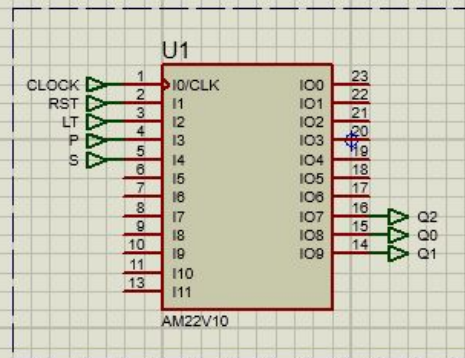
Generador de pulsos



Entradas

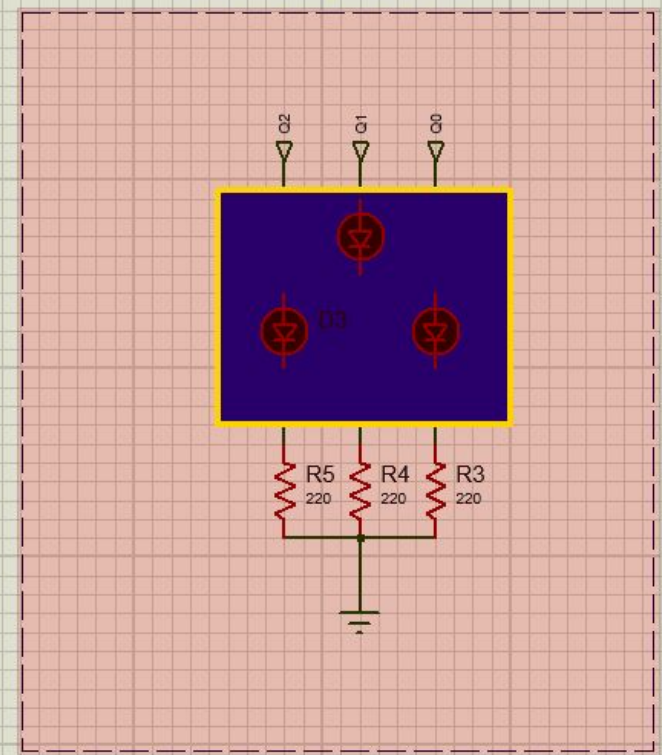


PLD 22v10



Sistema Secuencial Síncrono Señal de Alerta

Salidas Registradas



Prioridad LT

- 1.- **Paro** que, al oprimirlo sin importar el valor de S al llegar la señal de Clk el sistema deberá de permanecer en el mismo estado y al soltarlo continuar partiendo del estado actual hacia la secuencia correspondiente al valor de S.
- 2.- **Rst** (Restablecer) de modo que al oprimirlo al llegar la señal de Clk sin importar los valores de S o P el sistema regrese al estado en donde están todas las lampara apagadas y al soltarlo continuar con la secuencia correspondiente al valor de S.
- 3.- **LT** (prueba de lampara) que al oprimirlo al llegar la señal de Clk sin importar la condición actual se deberán de encender todas las lámparas y al soltarlo regresar al estado en donde están todas las lampara apagadas.

EP	Estado Siguiente							
LT	1	0	0	0	0			
Rst	X	1	0	0	0	Salidas Registradas		
P	X	X	1	0	0			
S	X	X	X	1	0	L2	L1	L0
E0	E7	E0	E0	E3	E1	0	0	0
E1	E7	E0	E1	E2	E2	0	0	1
E2	E7	E0	E2	E4	E4	0	1	0
E3	E7	E0	E3	E6	E6	0	1	1
E4	E7	E0	E4	E7	E1	1	0	0
E5	E7	E0	E5	E3	E7	1	0	1
E6	E7	E0	E6	E5	E5	1	1	0
E7	E7	E0	E7	E0	E0	1	1	1

Código ABEL-HDL

MODULE TRIPLR

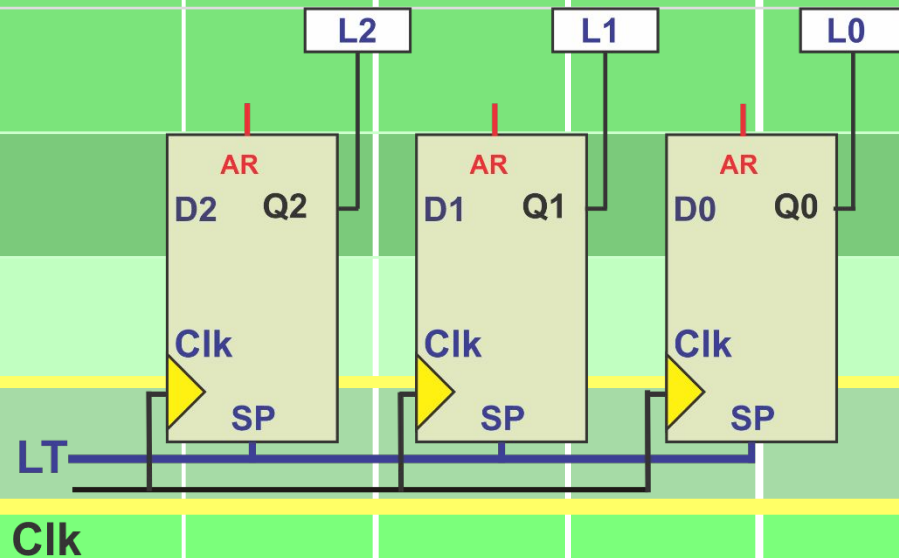
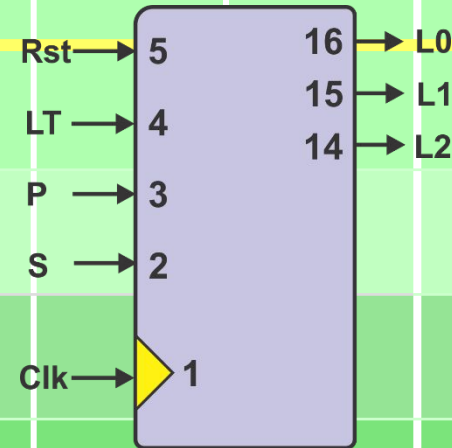
"17 Nov 2020

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

L=[L2..L0];
equations
L.clk=Clk;
L.SP = LT;



Código ABEL-HDL

Declarations

E0=[0,0,0];

E1=[0,0,1];

E2=[0,1,0];

E3=[0,1,1];

E4=[1,0,0];

E5=[1,0,1];

E6=[1,1,0];

E7=[1,1,1];

Código ABEL-HDL

State_diagram L

State E0:

IF **!Rst**&**!P**&**!S** then E1;

IF **!Rst**&**!P**&S then E4;

IF **!Rst**&P then E0;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E0	E0	E0	E3	E1

E0=[0,0,0];

Else If encadenamiento con Prioridad

State E0:

IF Rst then E0 **else if** P then E0 **else if** S then E3 **else** E1;

Código ABEL-HDL

State E1:

IF **!Rst**&**!P** then E2;

IF **!Rst**&**P** then E1;

IF Rst then E0;

E1=[0,0,1];

State E1:

IF Rst **then** E0 **else if** P **then** E1 **else** E2;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E1	E0	E1	E2	E2

Código ABEL-HDL

State E2:

IF !Rst&!P then E4;

IF !Rst&P then E2;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E2	E0	E2	E4	E4

E2=[0,1,0];

State E2:

IF Rst then E0 else if P then E2 else E4;

Código ABEL-HDL

State E3:

IF **!Rst**&**!P**&S then E6;

IF **!Rst**&P then E3;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E3	E0	E3	E6	E6

E3=[0,1,1];

State E3:

IF Rst then E0 **else if** P **then** E3 **else** E6;

Código ABEL-HDL

State E4:

IF **!Rst**&**!P**&**!S** then E1;

IF **!Rst**&**!P**&**S** then E7;

IF **!Rst**&**P** then E4;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E4	E0	E4	E7	E1

E4=[1,0,0];

State E4:

IF Rst **then** E0 **else if** P **then** E4 **else if** S **Then** **E7** **else** E1;

Código ABEL-HDL

State E5:

IF **!Rst**&**!P**&**!S** then E7;

IF **!Rst**&**!P**&**S** then E3;

IF **!Rst**&**P** then E5;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E5	E0	E5	E3	E7

E5=[1,0,1];

State E5:

IF Rst **then** E0 **else if** P **then** E5 **else if** S **Then** E3 **else** E7;

Código ABEL-HDL

State E6:

IF **!Rst**&**!P** then E5;

IF **!Rst**&**P** then E6;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E0	E0	E6	E5	E5

E6=[1,1,0];

State E6:

IF Rst then E0 else if P then E6 else E5;

Código ABEL-HDL

State E7:

IF **!Rst**&**!P** then E0;

IF **!Rst**&**P** then E7;

IF Rst then E0;

EP				
Rst	1	0	0	0
P	X	1	0	0
S	X	X	1	0
E7	E0	E7	E0	E0

E7=[1,1,1];

State E7:

IF Rst then E0 else if P then E7 else E0;

Truth_Table

MODULE TRIPLRTT

"17 Nov 2020

Clk,S,P,LT,Rst pin 1..5;

" salidas Registradas

L2..L0 pin 14..16 istype 'dc,reg';

X=.x.;

L=[L2..L0];

equations

L.clk=Clk;

L.sp=LT;

Truth_Table

([Rst,P,S,L]:>L)

[1,X,X,X]:>0;

Rst

Truth_Table

P=1

[0,1,X,0]:>0;
[0,1,X,1]:>1;
[0,1,X,2]:>2;
[0,1,X,3]:>3;
[0,1,X,4]:>4;
[0,1,X,5]:>5;
[0,1,X,6]:>6;
[0,1,X,7]:>7;

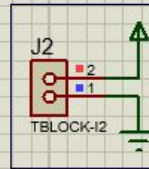
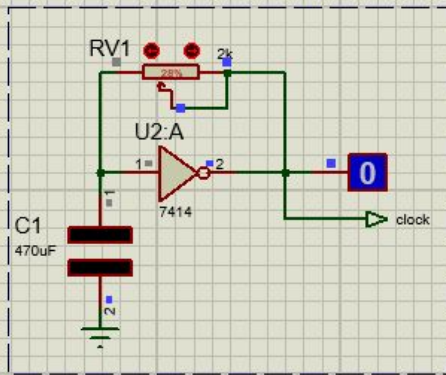
S=0

[0,0,0,0]:>1;
[0,0,0,1]:>2;
[0,0,0,2]:>4;
[0,0,0,3]:>6;
[0,0,0,4]:>1;
[0,0,0,5]:>7;
[0,0,0,6]:>5;
[0,0,0,7]:>0;

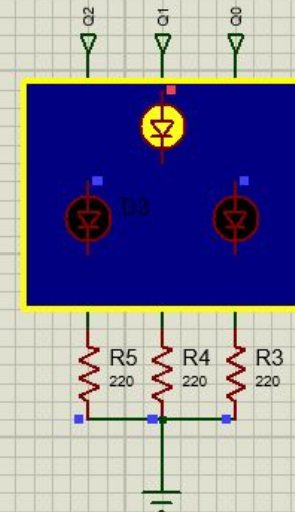
S=1

[0,0,1,0]:>3;
[0,0,1,1]:>2;
[0,0,1,2]:>4;
[0,0,1,3]:>6;
[0,0,1,4]:>7;
[0,0,1,5]:>3;
[0,0,1,6]:>5;
[0,0,1,7]:>0;

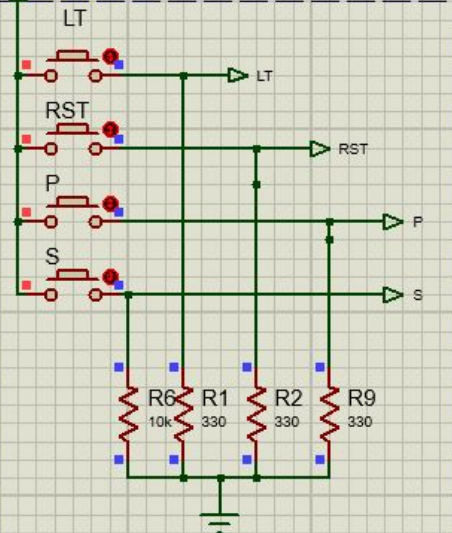
Generador de pulsos



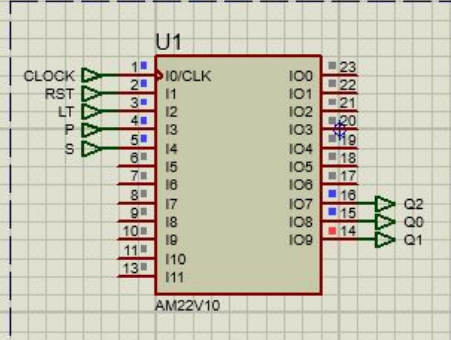
Salidas Registradas



Entradas



PLD 22v10



Sistema Secuencial Síncrono Señal de Alerta

DESIGN TITLE: **Señal de Alerta**

SHEET TITLE:

DATE: **06/11/2022**

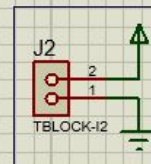
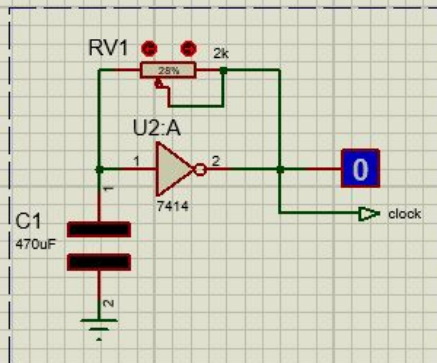
REV: **1.0**

BY: **JAGG**

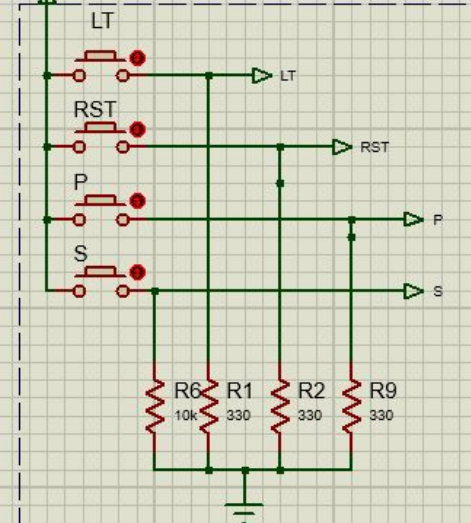
PAGE: **1** of **1**



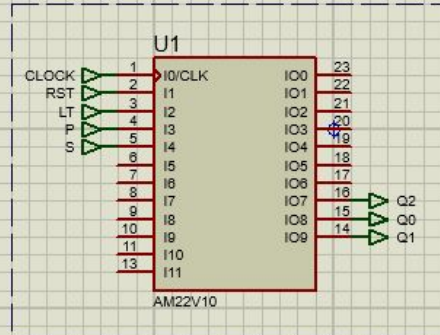
Generador de pulsos



Entradas

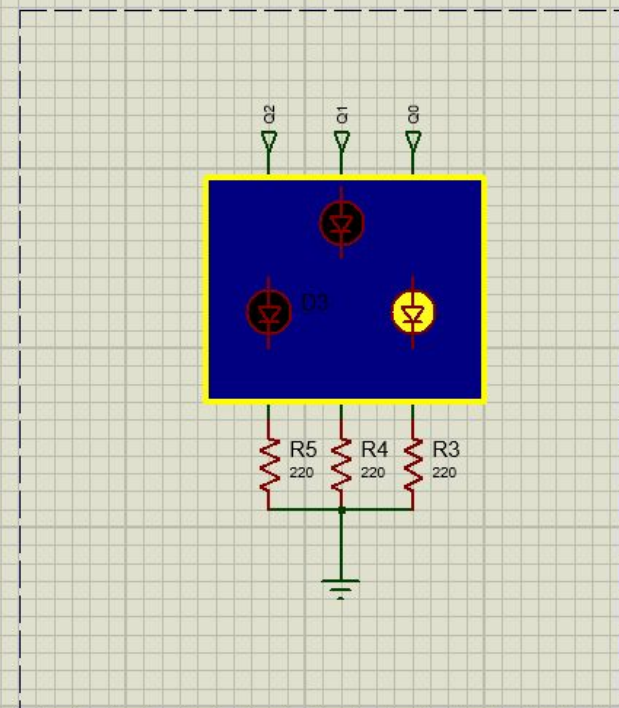


PLD 22v10

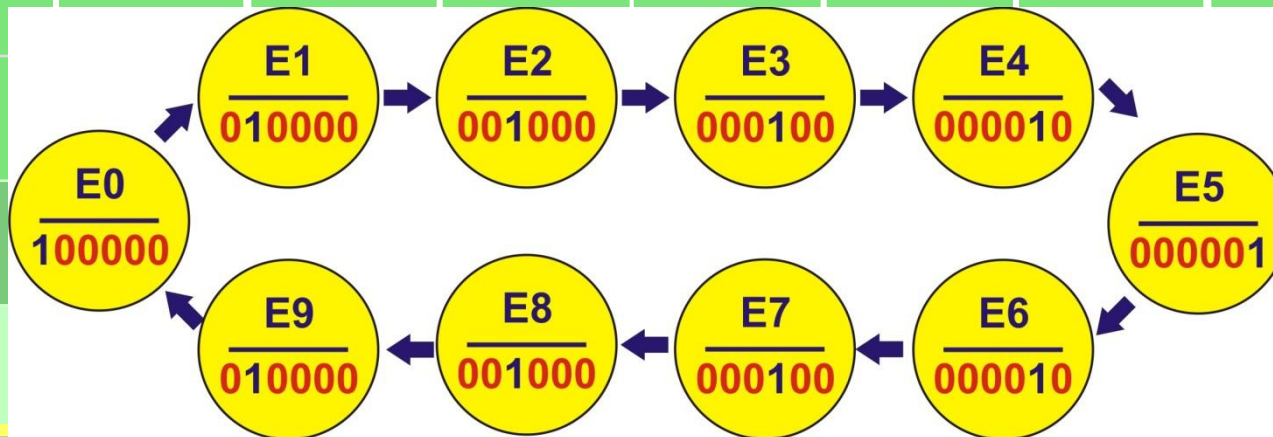


Sistema Secuencial Síncrono Señal de Alerta

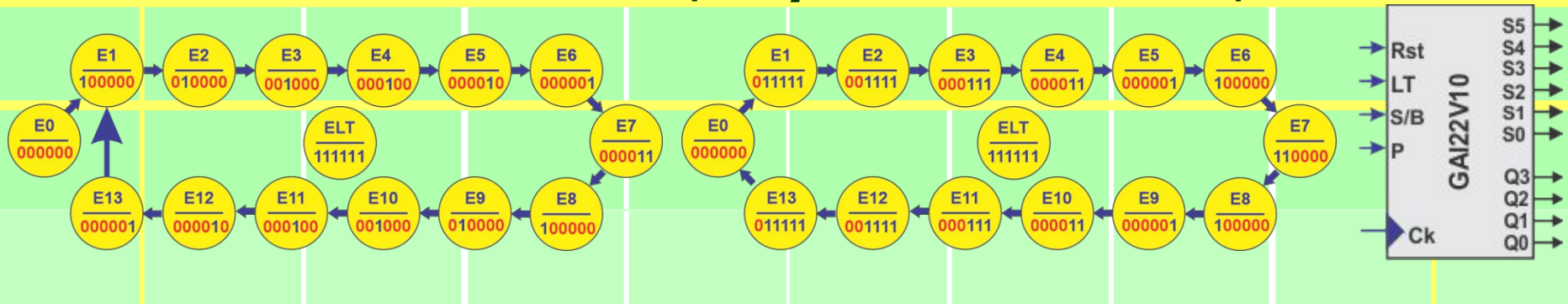
Salidas Registradas



Auto Increíble



Actividad fundamental 4, Asynchronous Reset, AR

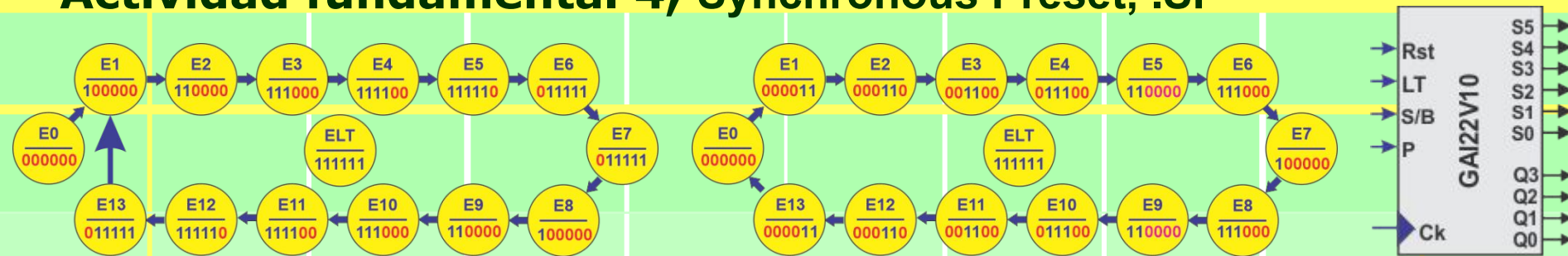


- Clk** (reloj) señal de sincronía con la que se determina el cambio de estado.
- SB** (sube/baja) de modo que si toma el valor de uno (SB=0) la secuencia será en forma ascendente de E0 a E13 y si SB= 1 la secuencia será en forma descendente, el sistema podrá en cualquiera de los estados cambiar de dirección dependiendo del valor de SB.
- P** (paro) de modo que al oprimirlo (P=1) sin importar el valor de SB el sistema debe de permanecer en el mismo estado y al soltarlo continuar con la secuencia seleccionada partiendo del estado actual y en dirección según el valor de valor de SB.
- LT** (Lamp Test) de modo que al oprimirlo si no está oprimido Rst deberán de encender todas las lámparas **ELT** al llegar la señal de Clk, y al soltarlo regresar al E0.
- Rst** (Restablecer) de modo que al oprimirlo sin importar la condición actual deberá de regresar al E0 sin necesidad de esperar la señal de Clk, para tal efecto se recomienda utilizar la instrucción **Asynchronous Reset, AR** y asignar al **E0=[0,0,0,0]**; el valor de cero a todas las salidas registradas E0=[Q3,Q2,Q1,Q0].

Entra das	LT	0	0	0	1	Salidas Combinacionales						Salidas Registradas Código Gray			
	P	0	0	1	X										
	SB	0	1	X	X	S5	S4	S3	S2	S1	S0	Q3	Q2	Q1	Q0
Esta do Pres ente	E0	E1	E13	E0	ELT	0	0	0	0	0	0	0	0	0	0
	E1			E1	ELT							0	0	0	1
	E2			E2	ELT							0	0	1	1
	E3			E3								0	0	1	0
	E4											0	1	1	0
	E5														
	E6														
	E7														
	E8														
	E9														
	E10														
	E11														
	E12														
	E13														
	ELT					1	1	1	1	1	1				

Reset asíncrono
 Asynchronous
 Reset
 AR

Actividad fundamental 4, Synchronous Preset, .SP



1. **Clk** (reloj) señal de sincronía con la que se determina el cambio de estado.
2. **SB** (sube/baja) de modo que si toma el valor de uno (SB=0) la secuencia será en forma ascendente de E0 a E13 y si SB= 1 la secuencia será en forma descendente, el sistema podrá en cualquiera de los estados cambiar de dirección dependiendo del valor de SB.
3. **P** (paro) de modo que al oprimirlo (P=1) sin importar el valor de SB el sistema debe de permanecer en el mismo estado y al soltarlo continuar con la secuencia seleccionada partiendo del estado actual y en dirección según el valor de valor de SB.
4. **Rst** (Restablecer) de modo que, al oprimirlo, siempre y cuando no está oprimido LT al llegar el pulso de Clk deberá de regresar al E0 y al soltarlo continuar con la secuencia correspondiente.
5. **LT** (Lamp Test) de modo que al oprimirlo sin importar la condición actual deberán al llegar la señal de Clk de encender todas las lámparas, y al soltarlo regresar al E0. para tal efecto se recomienda utilizar la instrucción **Synchronous Preset, .SP** y asignar al **ELT=[1,1,1,1]**; el valor de cero a todas las salidas registradas E0=[Q3,Q2,Q1,Q0].

Entradas	LT	0	0	0	0	1	Salidas Combinacionales						Salidas Registradas Código Gray			
	Rst	0	0	0	1	X							Q3	Q2	Q1	Q0
	P	0	0	1	X	X							Q3	Q2	Q1	Q0
	SB	0	1	X	X	X	S5	S4	S3	S2	S1	S0	0	0	0	0
Estado Presente	E0												0	0	0	1
	E1												0	0	1	1
	E2												0	0	1	0
	E3												0	1	1	0
	E4															
	E5															
	E6															
	E7															
	E8															
	E9															
	E10															
	E11															
	E12															
	E13															
	ELT															

Prioridad LT
Synchronous Preset,
.SP

Actividad Fundamental 4

Aplicación del método para el Diseño de sistemas secuenciales síncronos

Aula Invertida (Flipped Classroom):

Es un enfoque diferente de enseñanza, en donde el estudiante antes de la sesión de clase incorpora información (Teoría y aplicación del método) y con ello efectúa los procedimientos iniciales solicitados en la actividad y posteriormente dentro del aula en la sesión de clase (presencial o virtual) concluir su desarrollo asesorado por el profesor .

Es una actividad individual y es necesario efectuar los siguientes pasos:

Plataforma	parte	Actividad Fundamental 4
Share Point	1	AF4 m1 P1.mp4
	2	AF4 M1 P2.mp4
	3	AF4 M1 P3.mp4

Producto Integrador de Aprendizaje (PIA) 40 puntos

Para su evaluación es necesario ***programar una entrevista con el profesor***, en la cual se mostrará el prototipo y se explicarán los procedimientos y los resultados obtenidos con el apoyo de una presentación.

Antes de la entrevista, ***se deben haber subido los archivos entregables solicitados a la plataforma Google Classroom.***

La fecha límite para la entrega de los proyectos es de al menos dos días antes de la fecha programada para el examen ordinario.

Se solicita a los estudiantes ser previsores ya que

no se aceptarán proyectos entregados después de la fecha acordada.

Reflexión

***Con que se quedan
de esta sesión de clase***