

"Nada es permanente, excepto el cambio."

Heráclito



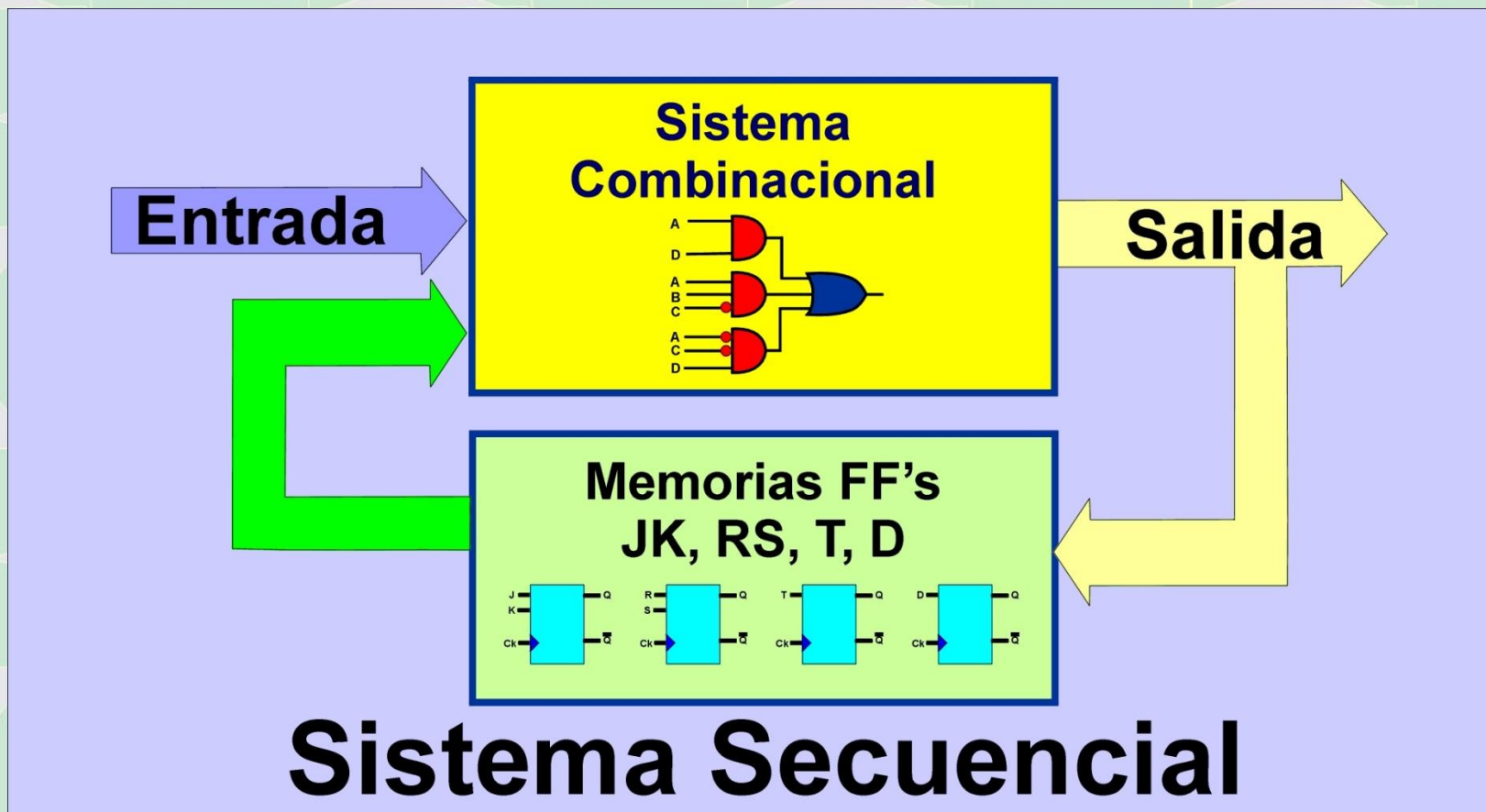
	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	<i>Miércoles 8 de octubre</i>
PF7	Diseño Combinacional con HDL	F	<i>Viernes 10 de octubre</i>
PF8	Multiplexor	5	<i>Miércoles 15 de octubre</i>
AF3	Decodificador con Display	10	<i>Lunes 18 de octubre</i>
PF9	Flip Flops	F	<i>Lunes 27 de octubre</i>
PF10	Pulsos de sincronía	5	<i>Viernes 7 de noviembre</i>
AF4	Diseño Secuencial	10	
AF5	PIA	40	Día del examen

Fundamentos del Diseño de Sistemas Secuenciales Síncronos y su descripción en ABEL-HDL

- 1. Introducción conceptual.***
- 2. diferencia entre sistemas combinacionales y secuenciales.***
- 3. Descripción de un sistema secuencial mediante un diagrama de transición de estados.***
- 4. Flujo de diseño.***
- 5. Uso de la señal de sincronía reloj CLK.***
- 6. Concepto de estados.***
- 7. Primeros comandos de descripción de la secuencia en ABEL-HDL***
 - a) GOTO***
 - b) IF THEN***
 - c) IF THEN ELSE.***

Sistema Secuencial

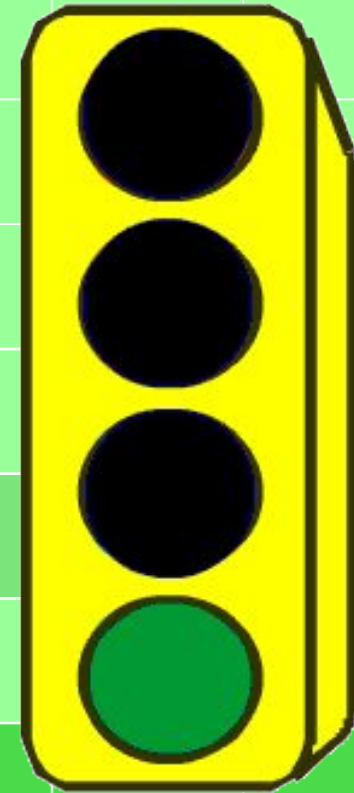
Un Sistema Secuencial es aquel sistema en donde los valores de salida no dependen únicamente de las combinaciones de entrada, *sino también de la salida misma.*



Sistema Secuencial

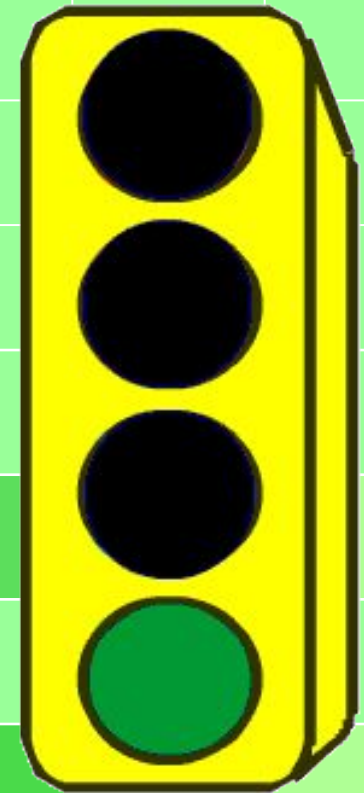
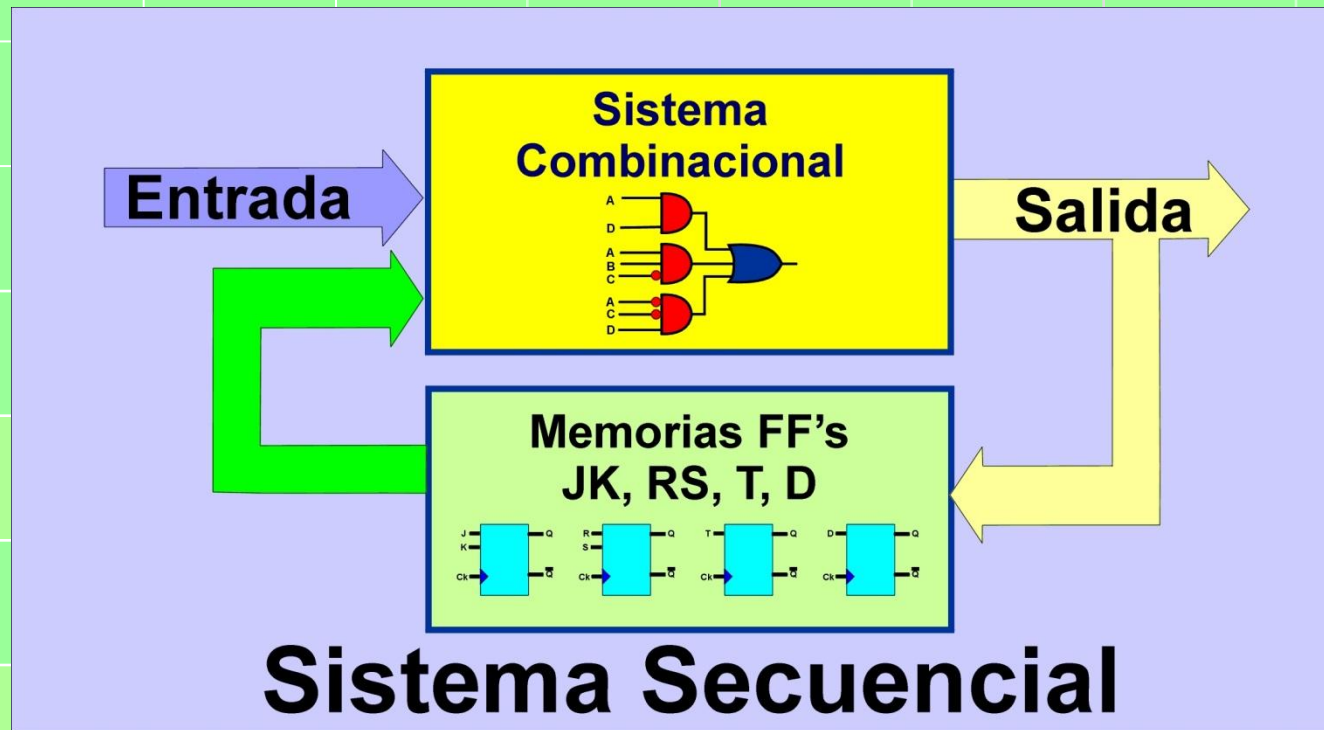
Un Sistema Secuencial es aquel Sistema en donde los valores de salida no dependen únicamente de las combinaciones de entrada, sino también de la salida misma.

**En un semáforo de que
depende decidir la luz que
sigue en encender
?
Cual es la luz que sigue**



Sistema Secuencial

Un Sistema Secuencial es aquel Sistema en donde los valores de salida no dependen únicamente de las combinaciones de entrada, sino también de la salida misma.

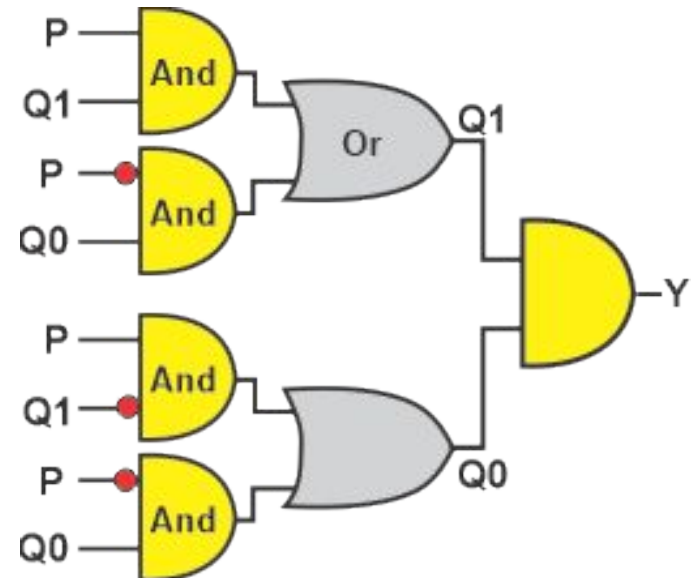
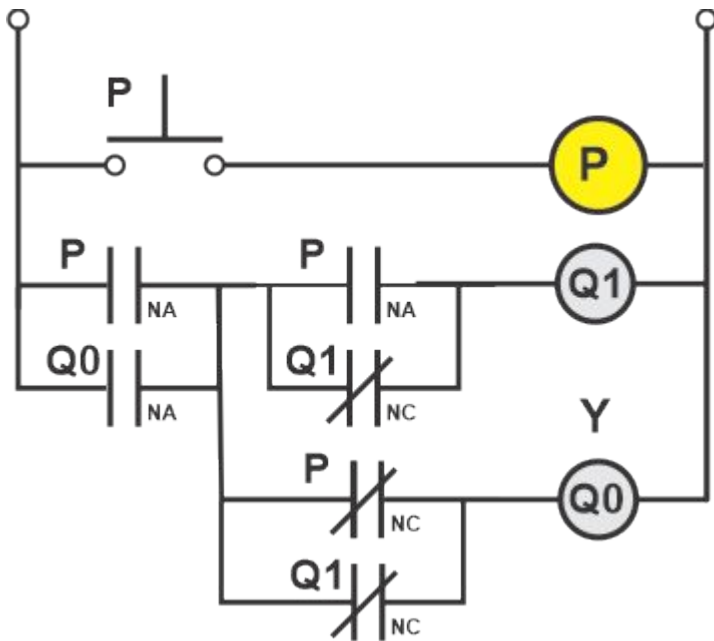


clasificación

Asíncrono: Depende solo de un cambio en las entradas

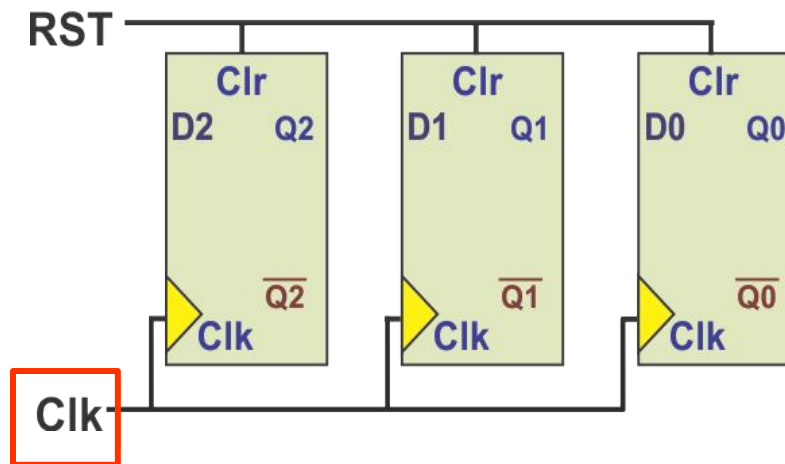
No depende de una señal de sincronía

Modo nivel



clasificación

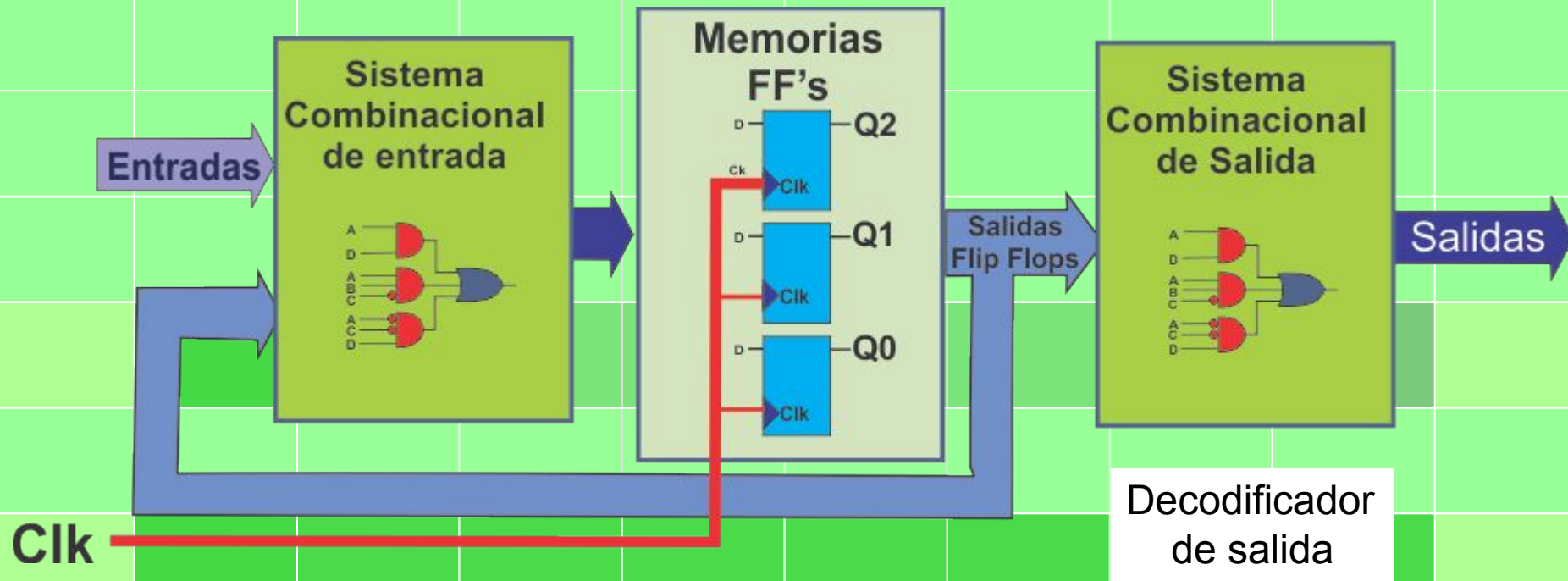
- **Síncrono** en donde los cambios de estado dependen de una señal de sincronía de los Flip Flops llamada reloj, **Ck o Clk**.



Modelos secuenciales síncronos y sus representaciones

Máquina de Moore

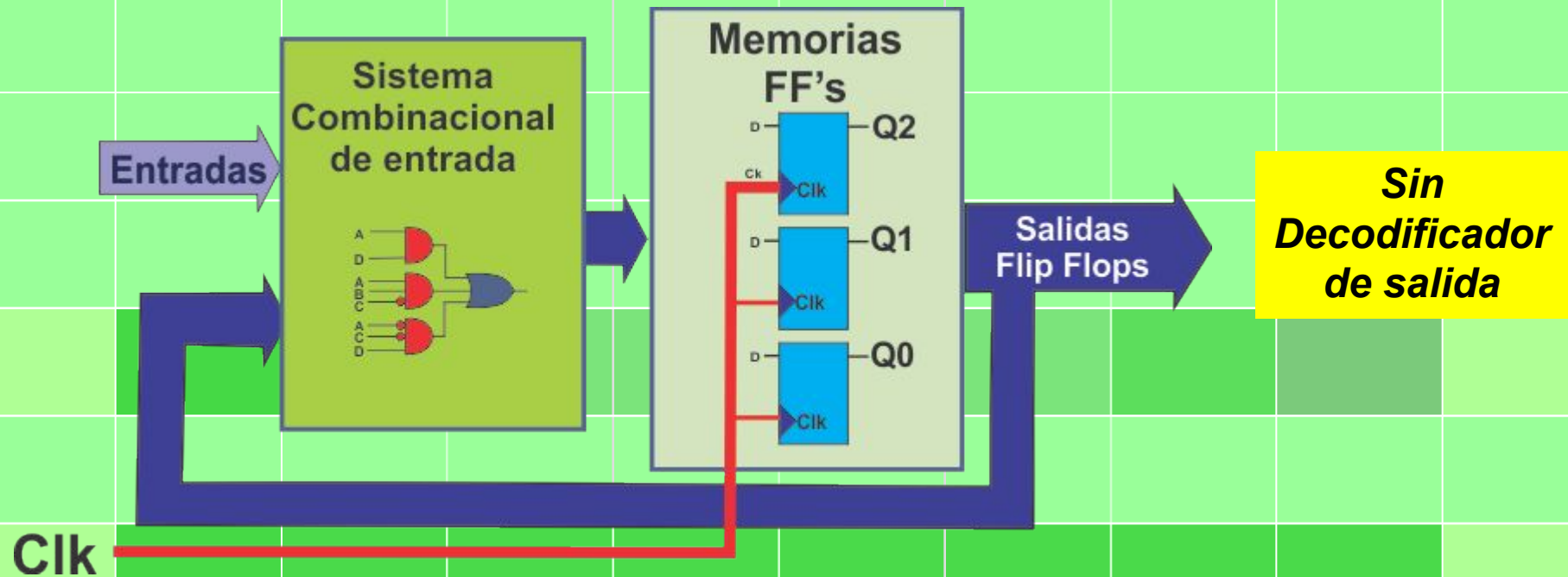
la salida solo depende del estado presente.



Modelos secuenciales síncronos y sus representaciones

Máquina de Moore

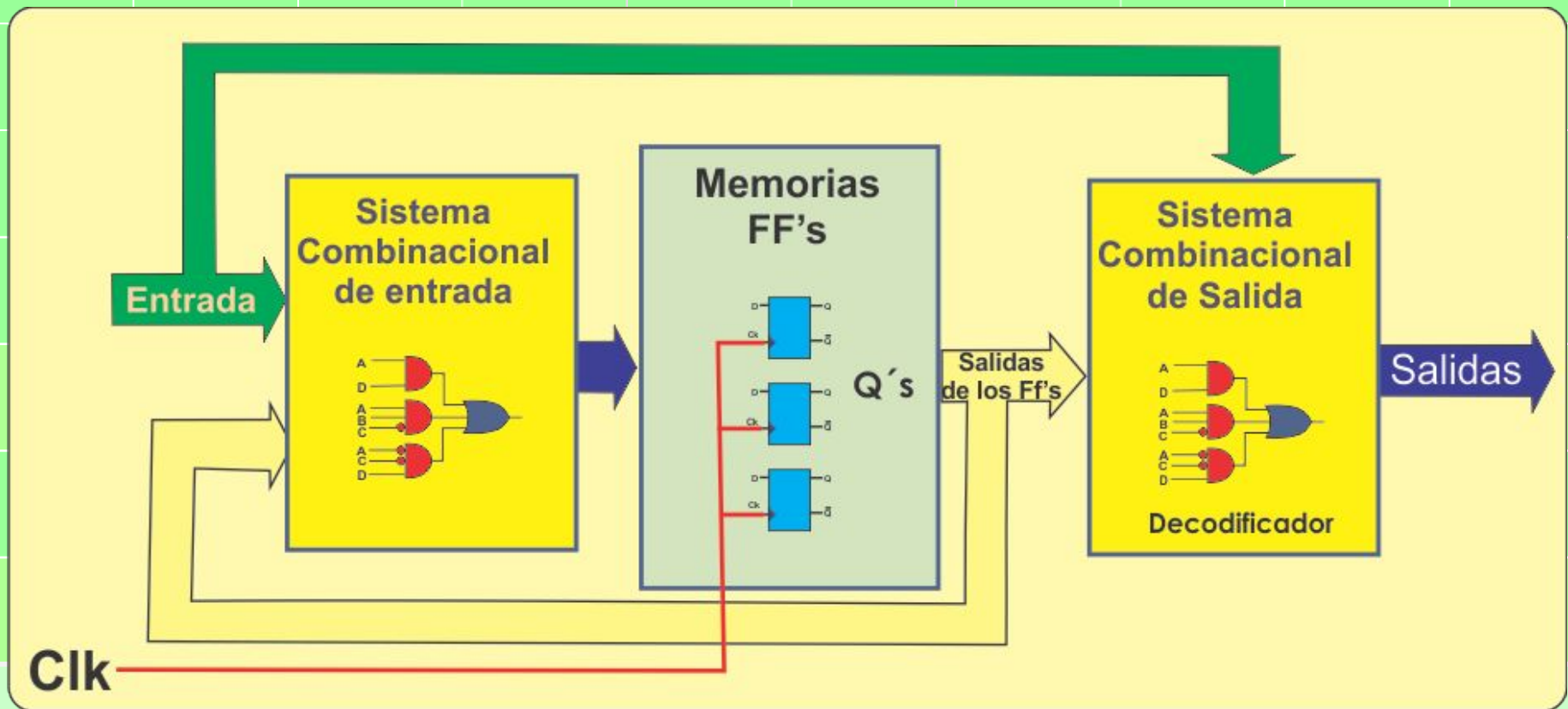
la salida solo depende del estado presente.



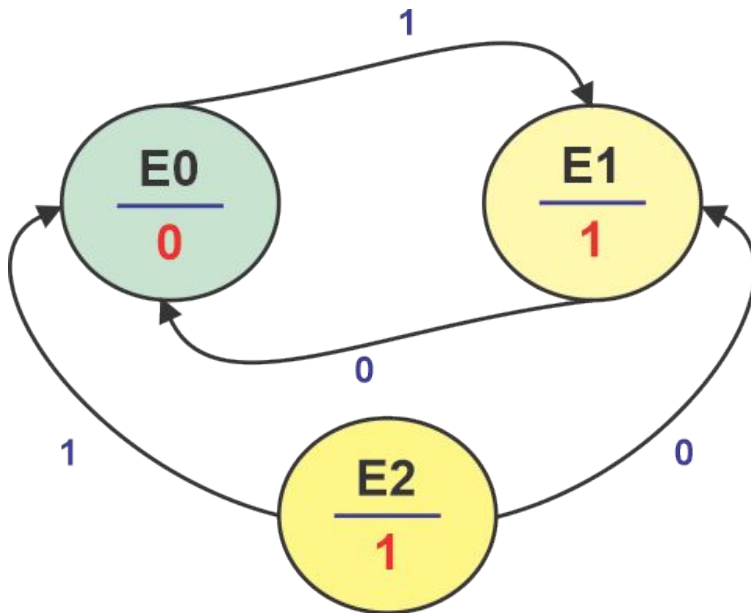
Modelos secuenciales síncronos y sus representaciones

Máquina de Mealy

en el que la salida depende tanto de la entrada como del estado presente.

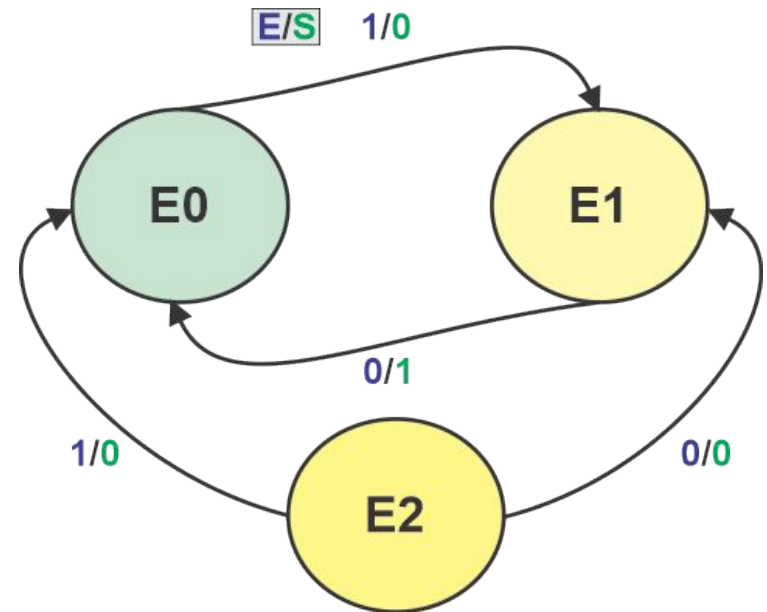


Moore



La salida asociada
al estado

Mealy



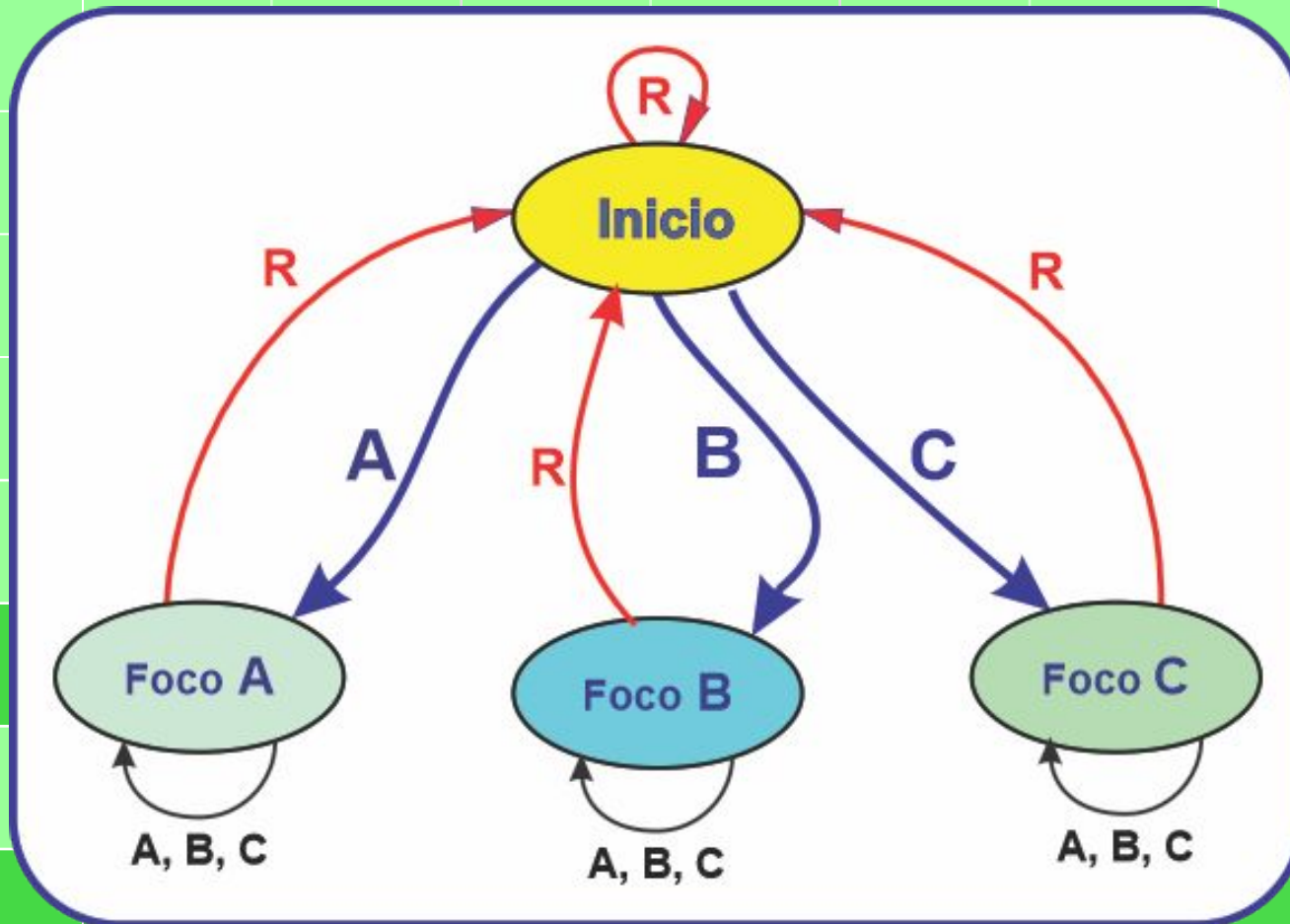
La salida asociada
al la transición

Cual es parte más difícil de la solución de un problema



Describirlo, Plantearlo, Identificarlo

Diagrama de Transición de Estados

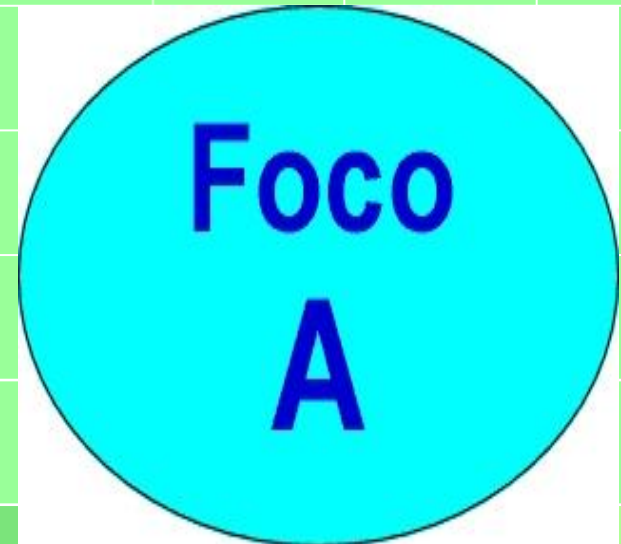


Grafos, Autómatas

Un diagrama de transición se compone de:

Estados o Eventos:

Una condición o situación de un objeto, durante la cual satisface una condición, realiza una actividad o está esperando un evento.



Un diagrama de transición se compone de:

Transición en el mismo estado:

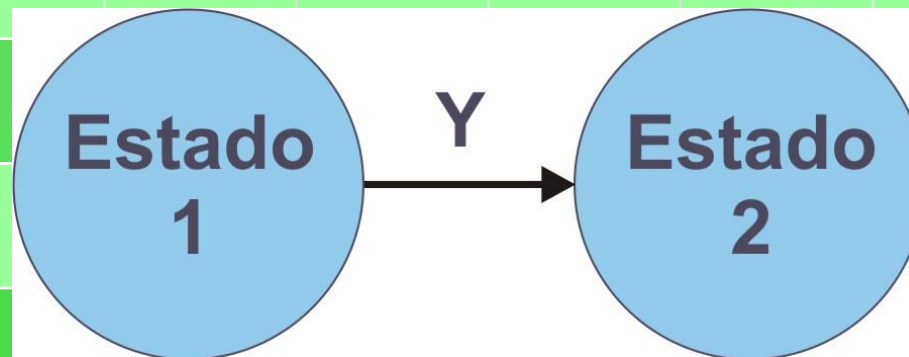
Una entrada X cuyo estado próximo es el mismo que el anterior.



Un diagrama de transición se compone de:

Transición entre dos estados:

Una relación entre estados que indica que un objeto que está en el primer estado realizará una acción especificada y, entrará en el segundo estado cuando un evento Y especificado ocurra y unas condiciones especificadas sean satisfechas.



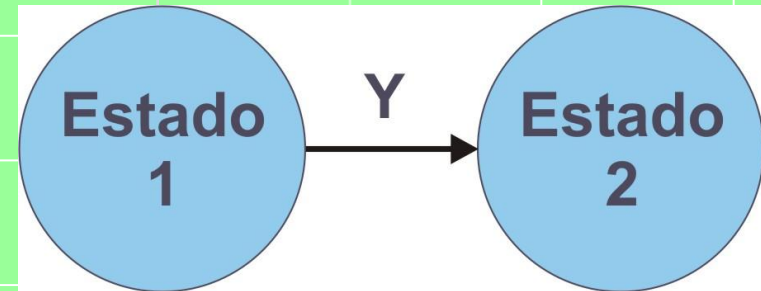
Un diagrama de transición se compone de:

Entradas:

Combinaciones que establecen un cambio de evento.

Salidas:

Valores combinacionales que determinan un evento



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

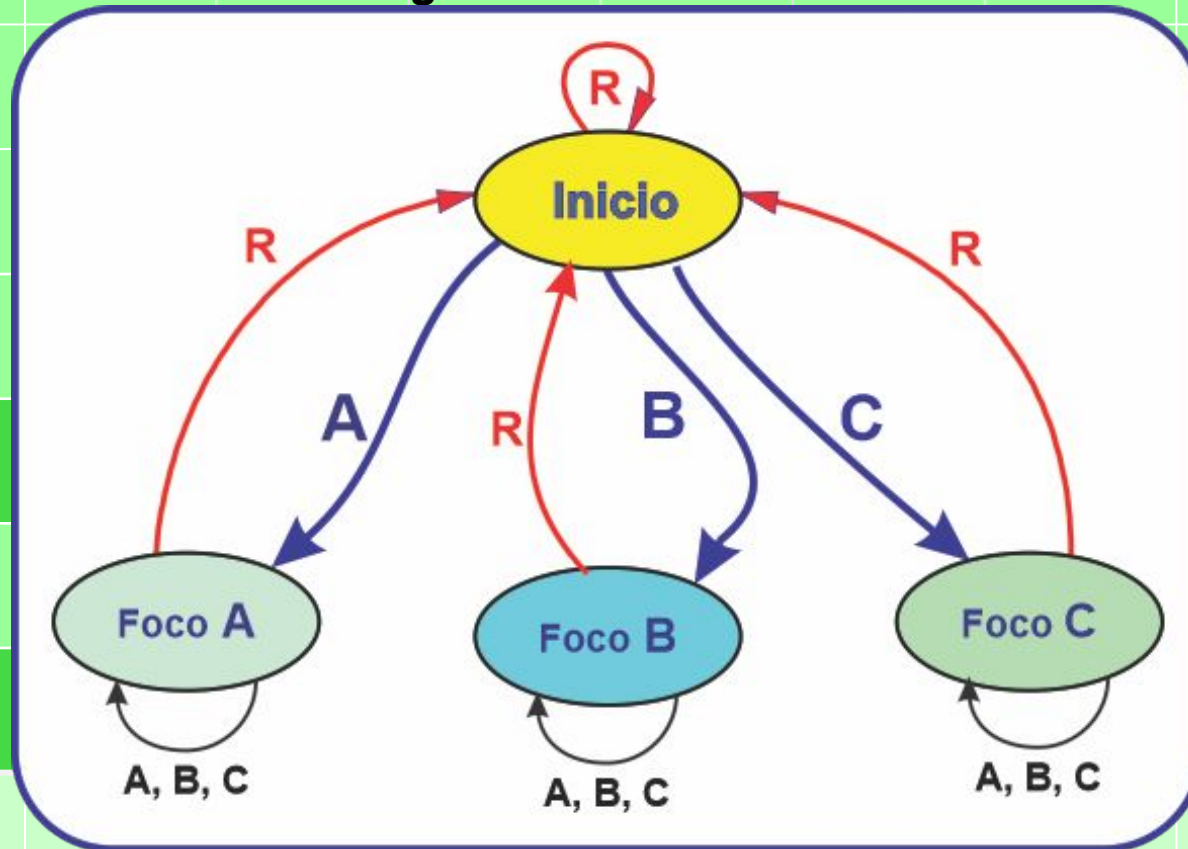
- 1.- Especificar el sistema (Diagrama de transición)
- 2.- Determinar la cantidad de Flip Flops
- 3.- Asignar valores a los estados
- 4.- Diagrama de Bloque (entradas y salidas)
- 5.- Construir la tabla de estado siguiente
- 6.- Codificación en ABEL-HDL
 - a) Entradas y Salidas
 - b) Sincronización de los Flip Flops
 - c) Asignación de valores a los estados
 - d) definir la secuencia (state_diagram o Truth_table)
- 7.- Simulación
- 8.- Implementación

Base para
describir y
Diseñar
un sistema
secuencial

Especificar el Sistema

Para especificar el comportamiento del sistema se puede hacer uso del diagrama de transición, en donde se indica la secuencia deseada además de las entradas, salidas y estados.

Diagrama de Transición



Relación entre Estados y Flip-Flops

La cantidad de Flip-Flops necesarios en el diseño de un sistema secuencial depende directamente del número de estados definidos en el diagrama de transición.

Con las combinaciones generadas con las salidas de los Flip Flops se puede identificar forma única cada uno de los estados del sistema secuencial

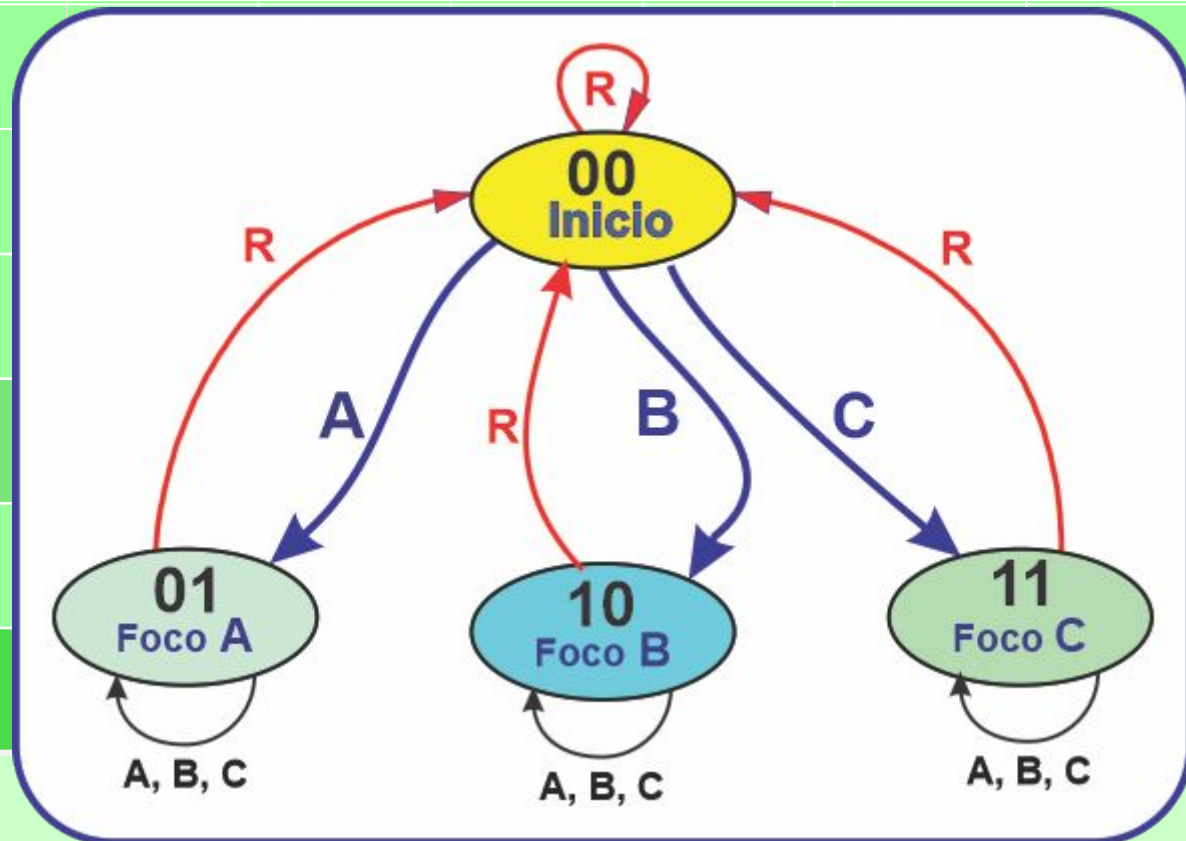
Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3
9 a 16	4
17 a 32	5
33 a 64	6
65 a 128	7
129 a 256	8
257 a 512	9
513 a 1024	10

Asignar los valores a los estados

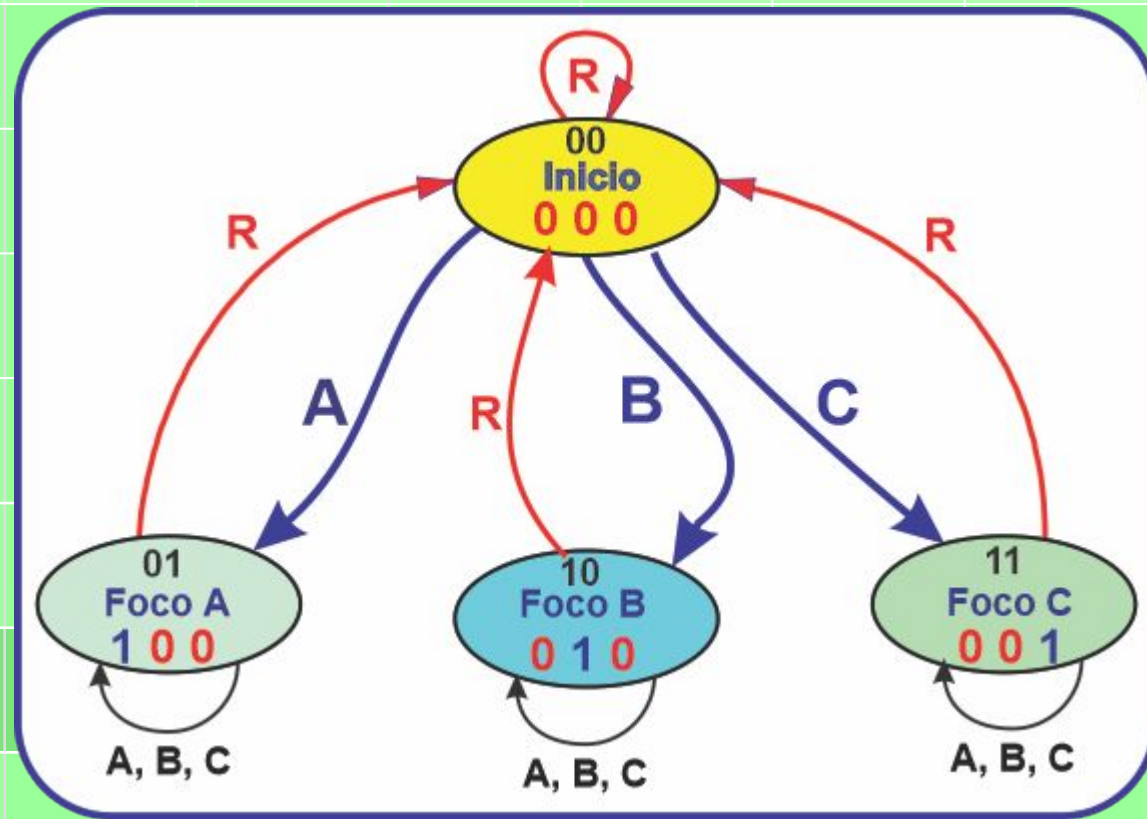
La asignación de códigos binarios a los estados puede hacerse de forma arbitraria, siempre y cuando cada estado reciba una combinación única.

Estas combinaciones corresponden a las posibles salidas de los Flip-Flops (Q), lo que permite identificar inequívocamente cada estado dentro del sistema.

Estados	Salidas FF's	
	Q1	Q0
Inicio	0	0
Foco A	0	1
Foco B	1	0
Foco C	1	1



Estados	Salidas FF's	Salidas Combinacional es Concursantes		
	Q1 Q0	A	B	C
Inicio	0 0	0	0	0
Foco A	0 1	1	0	0
Foco B	1 0	0	1	0
Foco C	1 1	0	0	1



Determinar las entradas y salidas

En esta parte se recomienda identificar las entradas y salidas del sistema secuencial, usando un diagrama de bloques como lo muestra la siguiente figura.

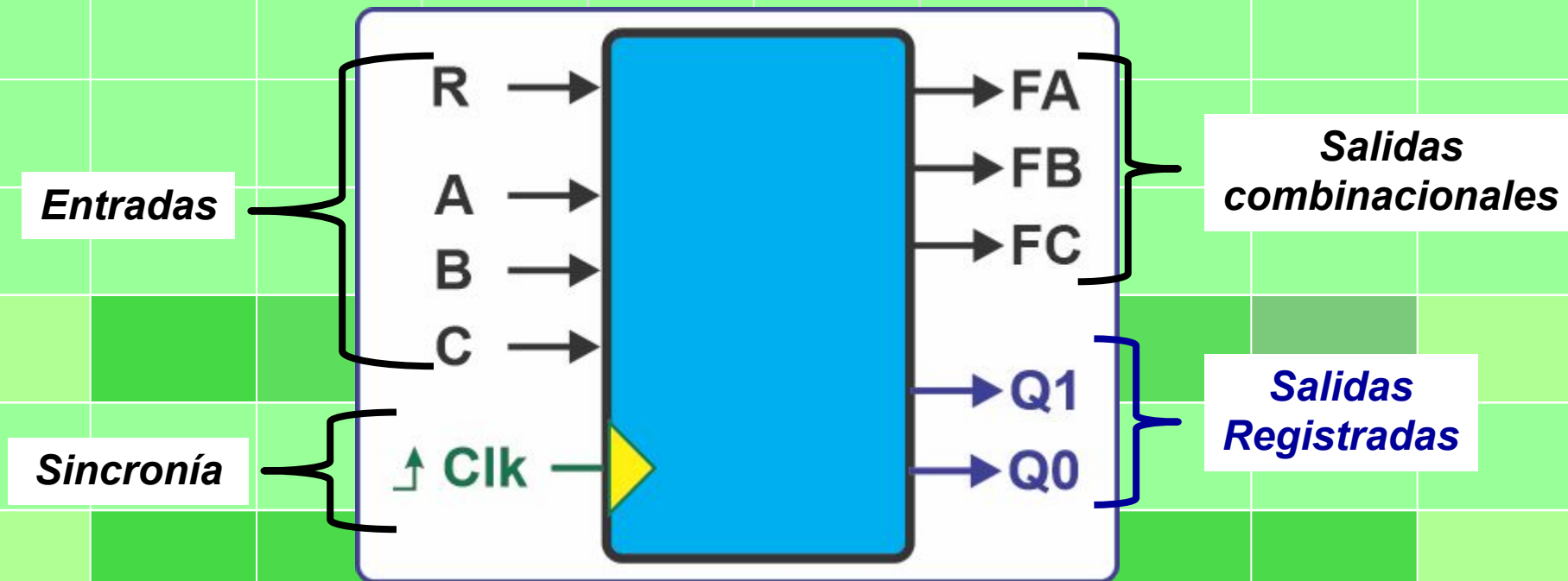


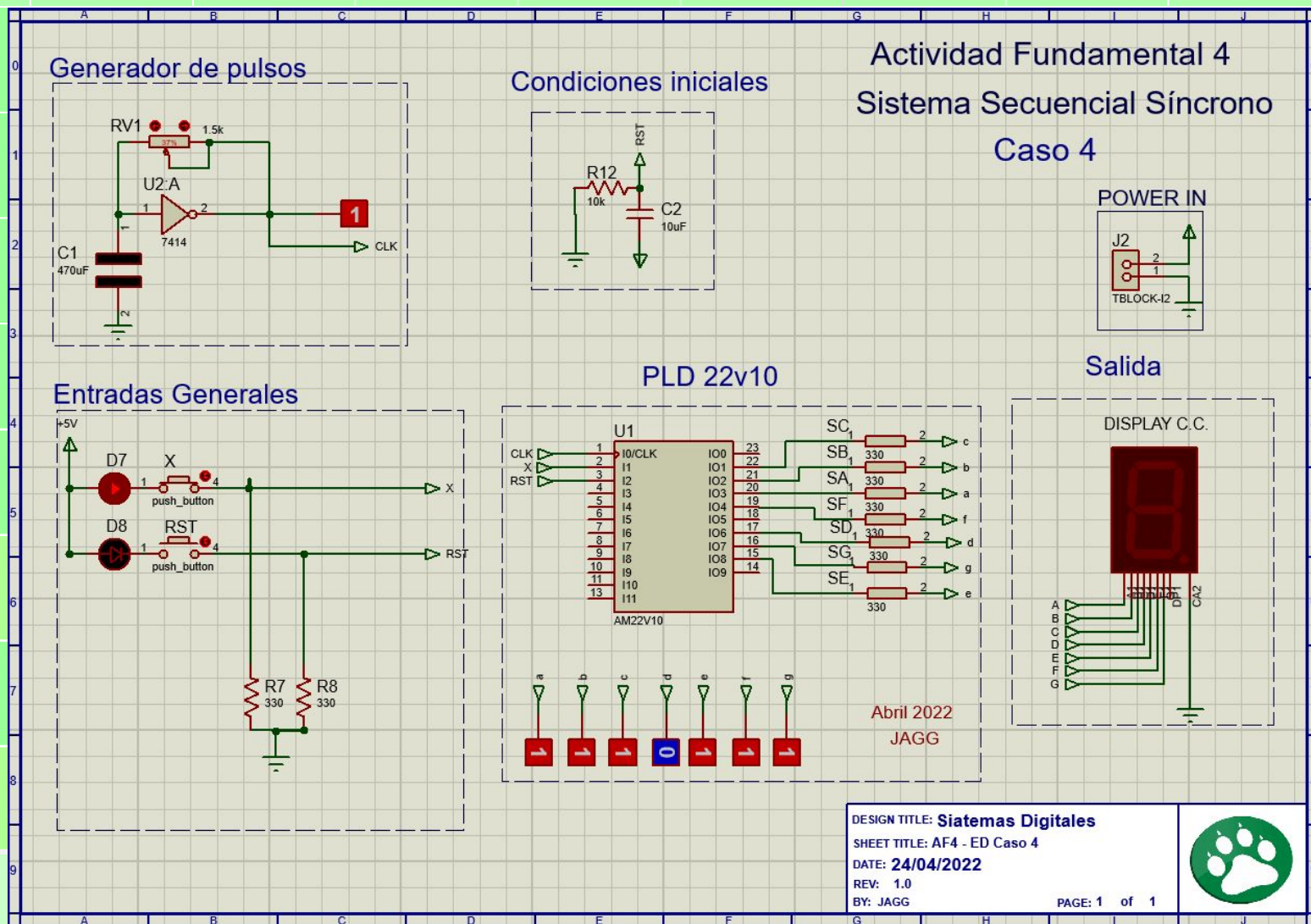
Tabla de Estado Siguiente

		Estado Próximo					Salidas Combinacionales			Salidas Registradas	
Entradas	R	0	0	0	0	1					
	BA	0	0	0	1	X					
	BB	0	0	1	X	X					
	BC	0	1	X	X	X					
Estado Presente	CI	CI	EC	EB	EA	CI	FA	FB	FC	Q1	Q0
	EA	EA	EA	EA	EA	CI	1	0	0	0	1
	EB	EB	EB	EB	EB	CI	0	1	0	1	0
	EC	EC	EC	EC	EC	CI	0	0	1	1	1

Código en ABEL-HDL

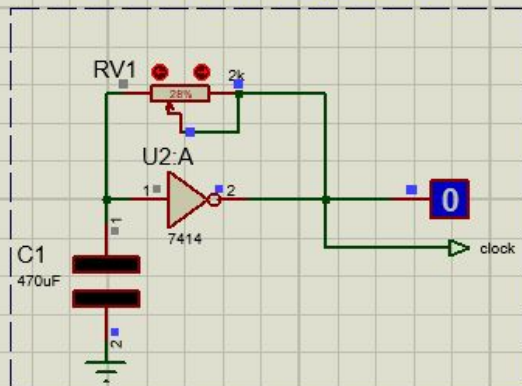
- a) Entradas y Salidas.
- b) Sincronización de los Flip Flops.
- c) Asignación de valores a los estados.
- d) Definir la secuencia (State_Diagram o Truth_table)

Diagrama Esquemático

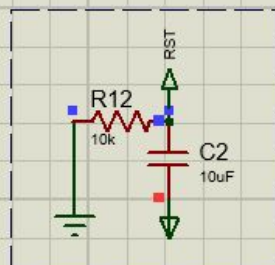


Simulación

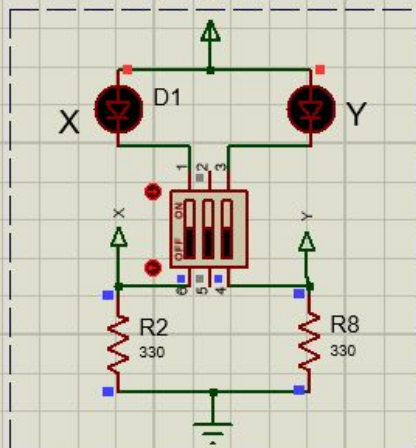
Generador de pulsos



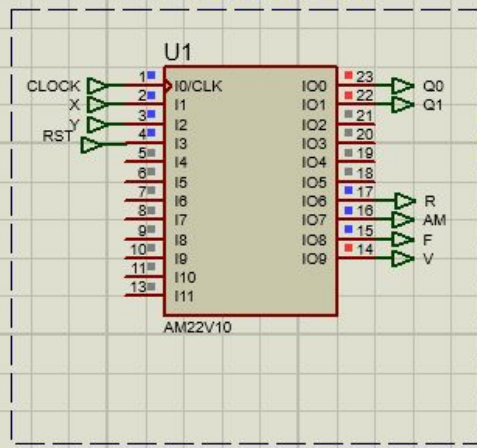
Condiciones iniciales



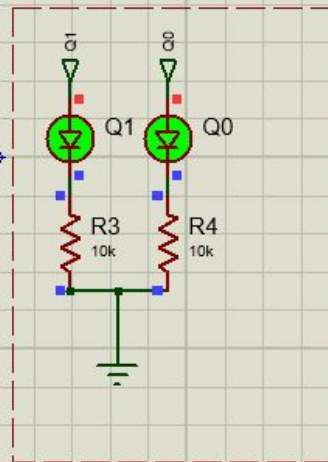
Entradas



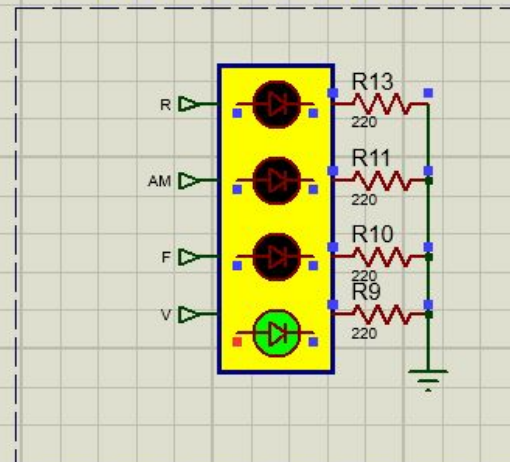
PLD 22v10



Salidas Registradas

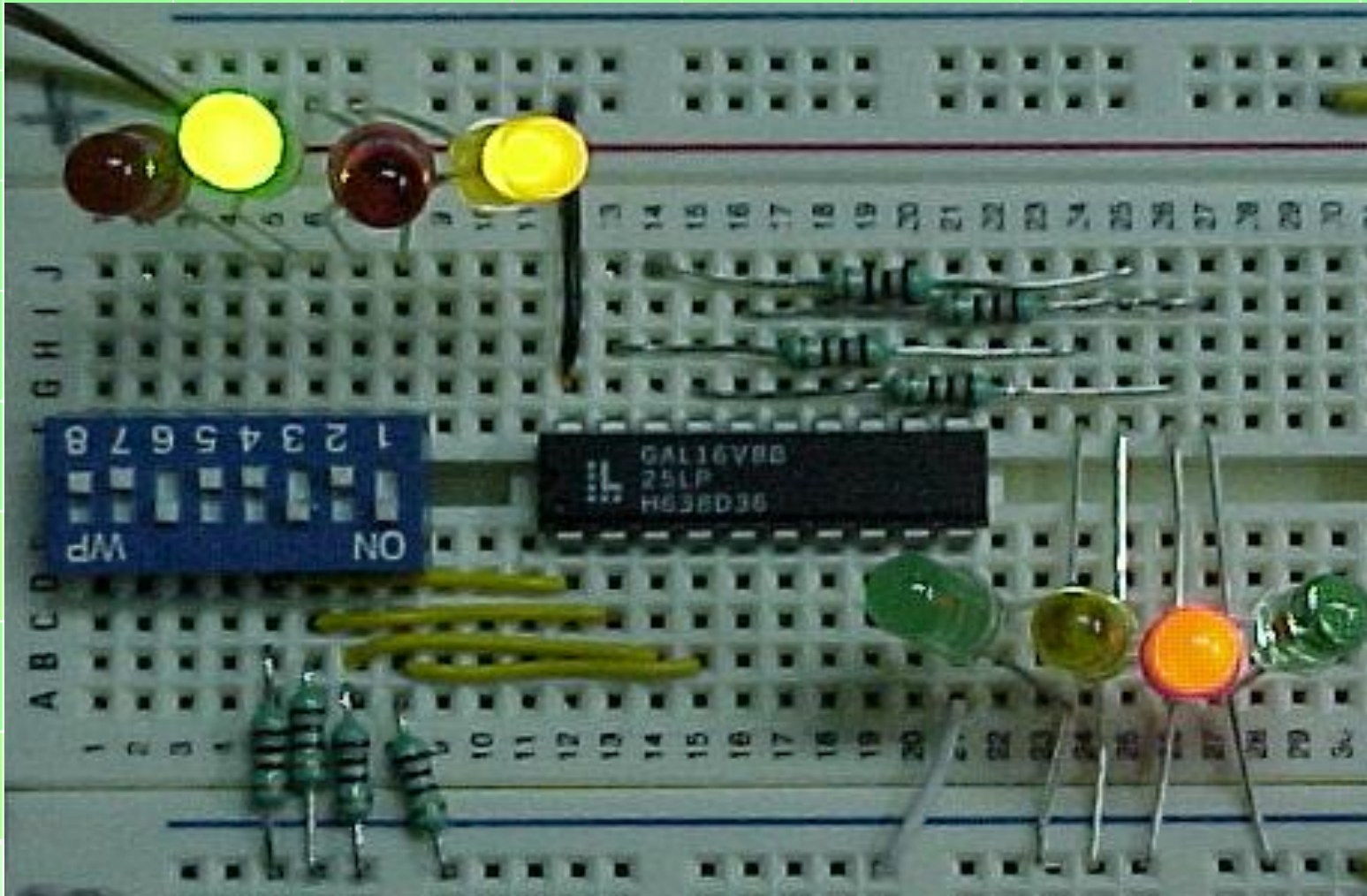


Salidas Combinacionales



Sistema Secuencial para un semaforo

Construir el prototipo



Método para el 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
- 3.- Asignar valores a los estados
- 4.- Diagrama de Bloque (entradas y salidas)
- 5.- Construir la tabla de estado siguiente
- 6.- Codificación en ABEL-HDL
 - a) Entradas y Salidas
 - b) Sincronización de los Flip Flops
 - c) Asignación de valores a los estados
 - d) definir la secuencia (state_diagram o Truth_table)
- 7.- Simulación
- 8.- Implementación

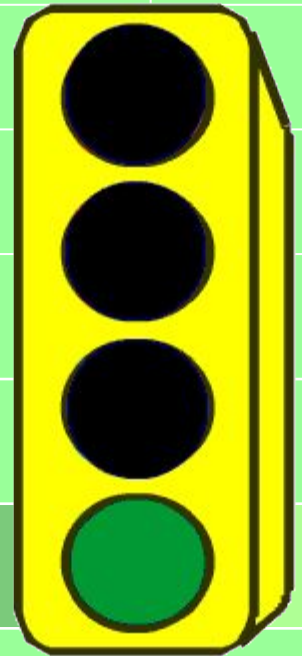
Base para
describir y
Diseñar
un sistema
secuencial

Ejemplo 1

Diseñe un ***Sistema Secuencial Síncrono*** que represente la operación de un semáforo de cuatro estados que se presentan en el siguiente orden:

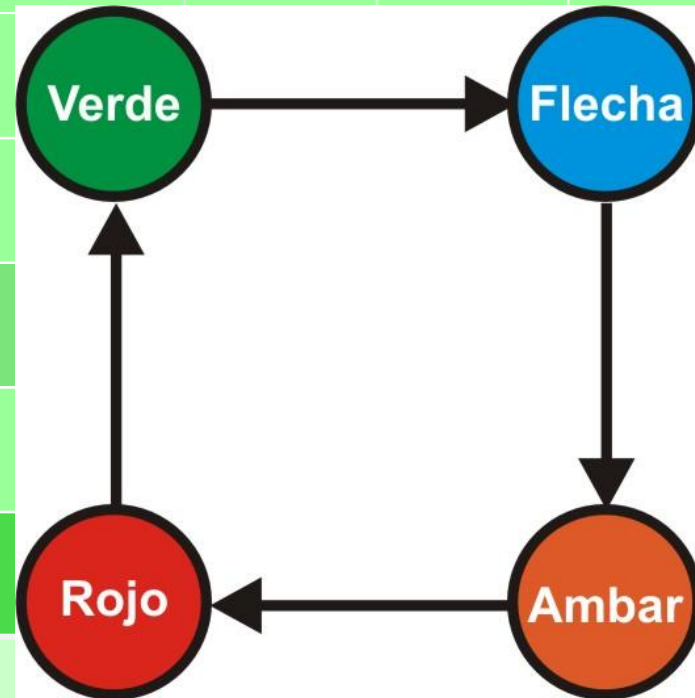
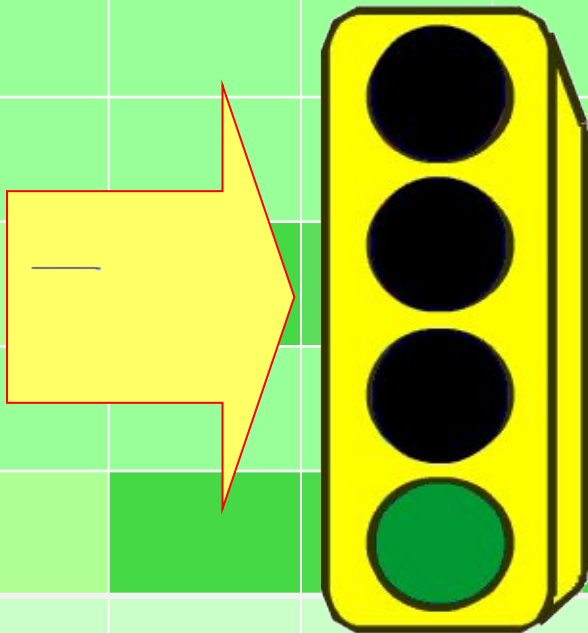
Verde, **Flecha**, **Ámbar** y **Rojo**

que cambie de estado con una señal de transición positiva llamada Clk, el sistema debe de ser cíclico



Especificar el Sistema

En este diagrama de transición se indica la secuencia del semáforo en donde los cambios de estado se realizarán cada vez que le proporcionemos un pulso de sincronía (Ck) a los Flip Flops.



Determinar la cantidad de Flip Flops

Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3
9 a 16	4
17 a 32	5

Nuestro ejemplo consta de cuatro estados **Verde**, **Flecha**, **Ámbar** y **Rojo**, por lo que requeriremos de **dos *Flip Flops***, para identificarlos los llamaremos **Q1 y Q0**.

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde		
Flecha		
Ámbar		
Rojo		

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha		
Ámbar		
Rojo		

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar		
Rojo		

Asignar los valores a los estados

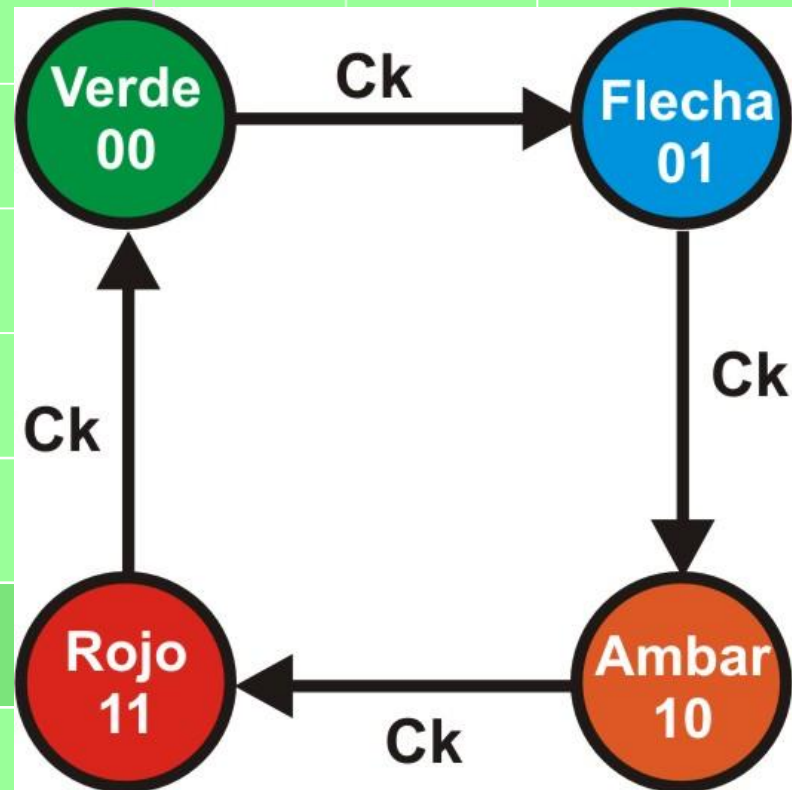
Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo		

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1



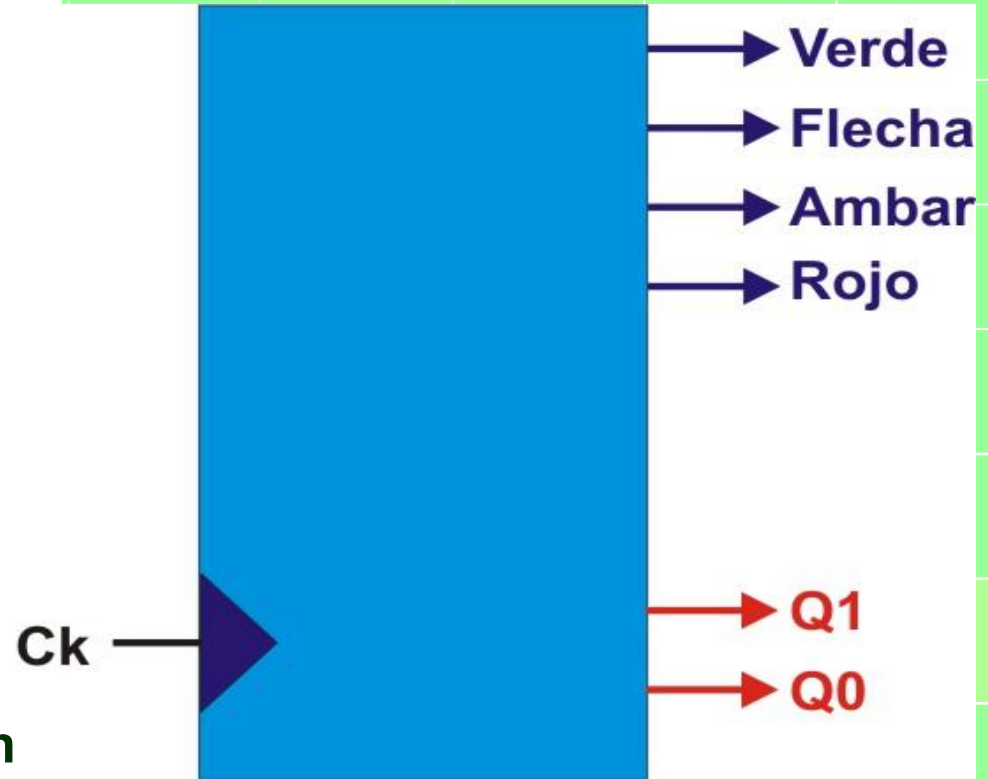
Determinar las entradas y salidas

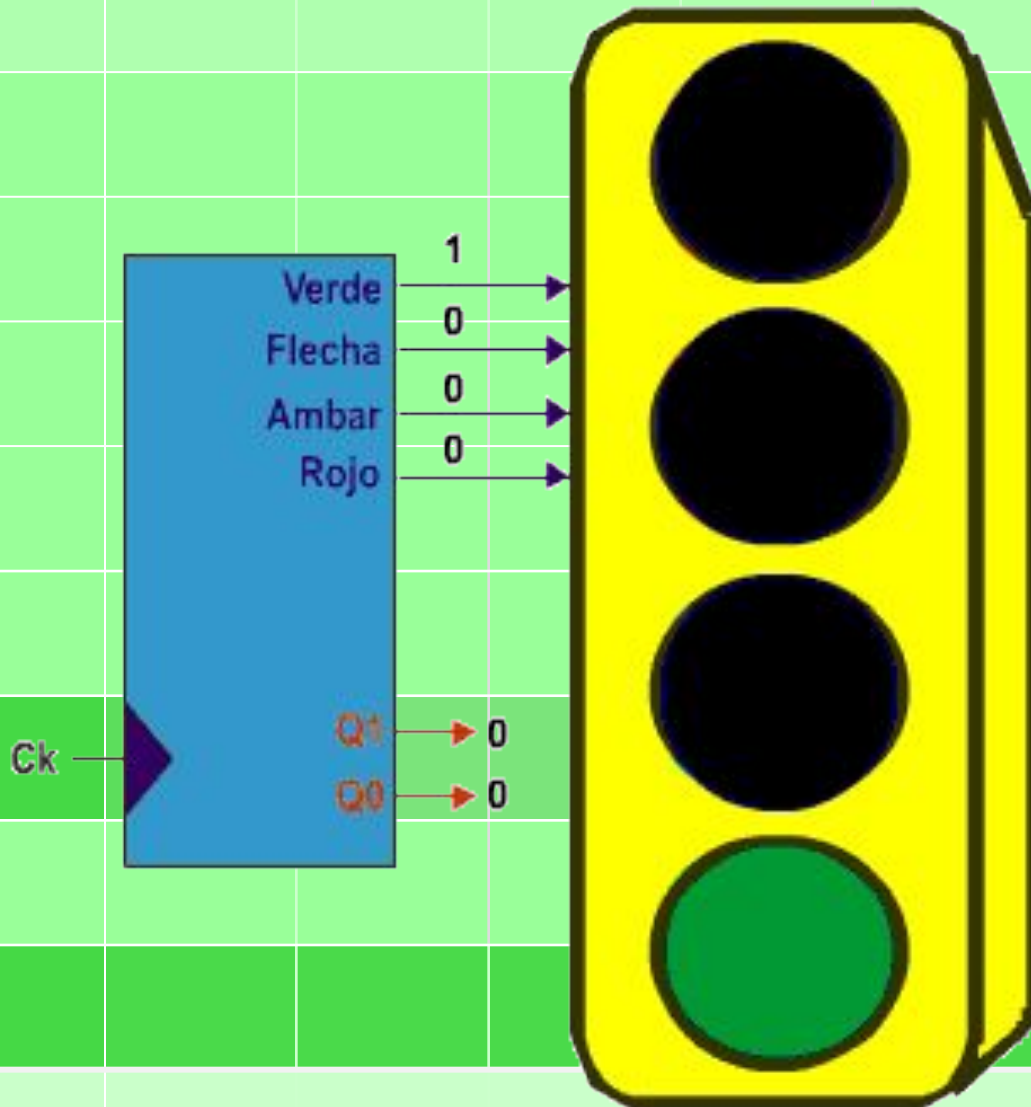
Como se observa en la figura, el sistema tiene una sola entrada

llamada **Ck**

Seis salidas de las cuales **Verde**, **Flecha**, **Ámbar** y **Rojo**, son **Combinacionales**.

Además de **Q1** y **Q0** son las salidas de los Flip Flops o también llamadas registradas (reg).



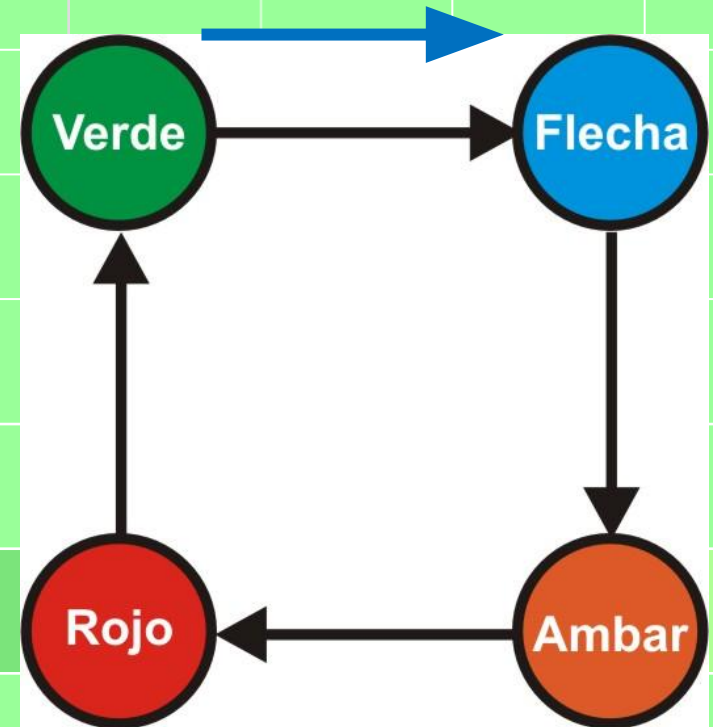


Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	
Flecha	
Ámbar	
Rojo	

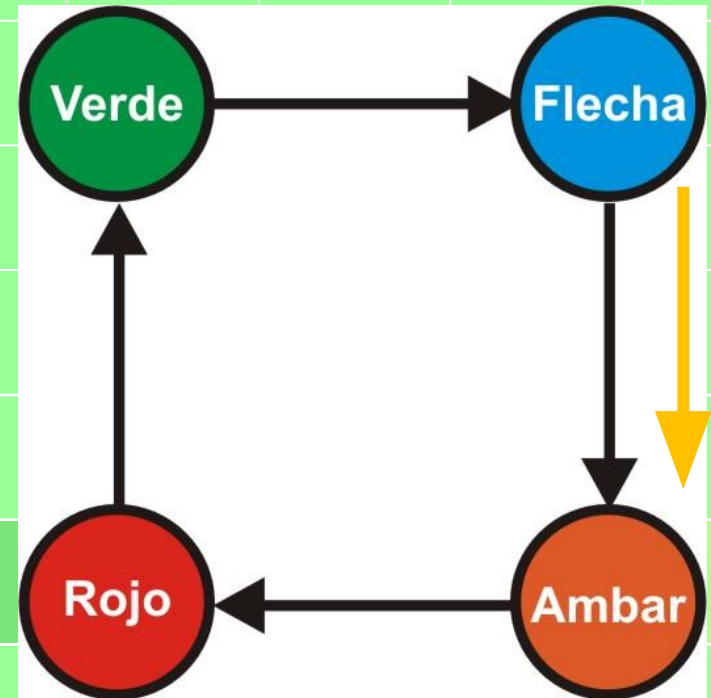
Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	
Flecha	
Ámbar	
Rojo	



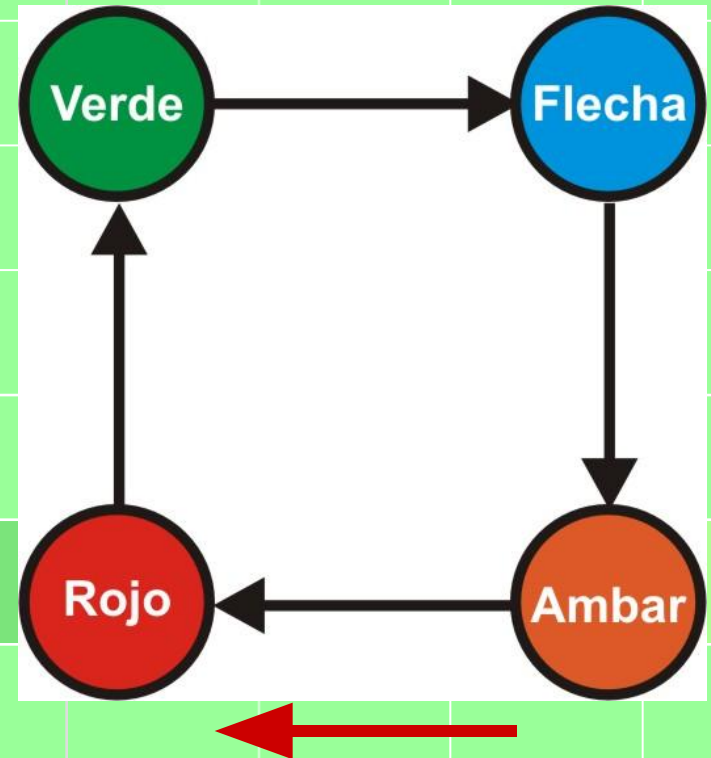
Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	Flecha
Flecha	
Ámbar	
Rojo	



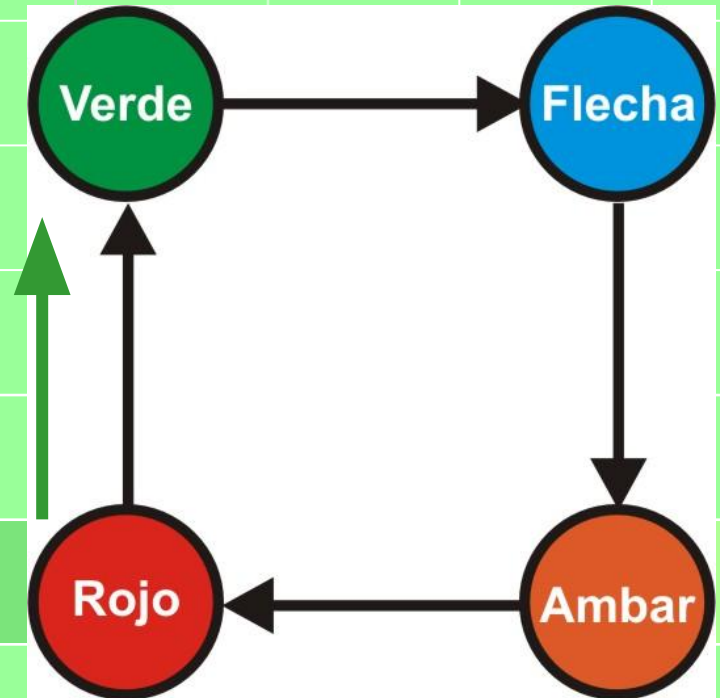
Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	Flecha
Flecha	Ámbar
Ámbar	
Rojo	



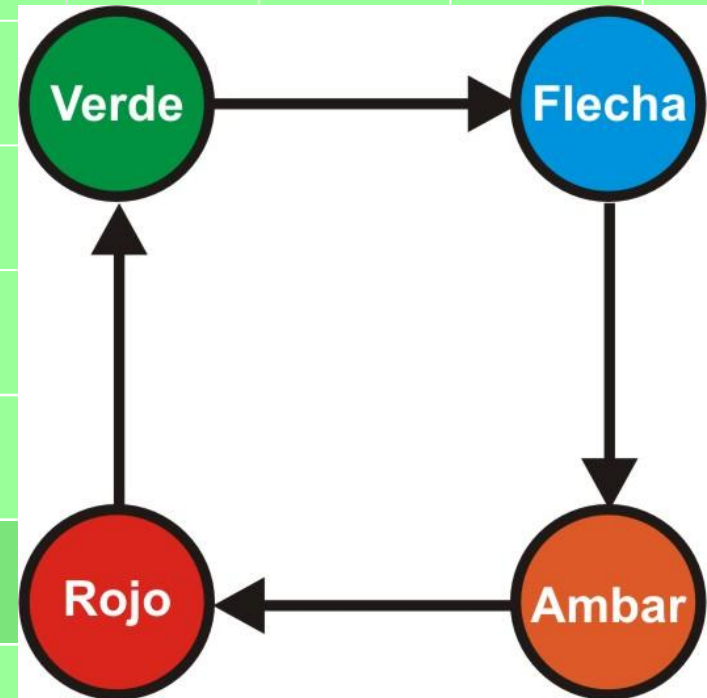
Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	Flecha
Flecha	Ámbar
Ámbar	Rojo
Rojo	



Construir una Tabla de Estados

Estado Presente	Estado Siguiente
Verde	Flecha
Flecha	Ámbar
Ámbar	Rojo
Rojo	Verde



		Estado Presente		Estado Siguiente		Entradas de Control		Salidas			
m		Q1	Q0	Q1+1	Q0+1	T1	T0	V	F	A	R
0	Verde	0	0	0	1	0	1	1	0	0	0
1	Flecha	0	1	1	0	1	1	0	1	0	0
2	Ámbar	1	0	1	1	0	1	0	0	1	0
3	Rojo	1	1	0	0	1	1	0	0	0	1

Tabla de Excitación

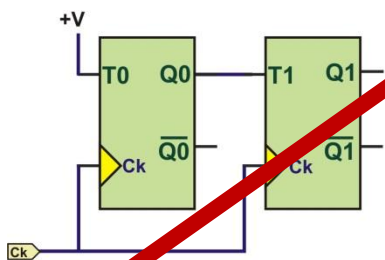
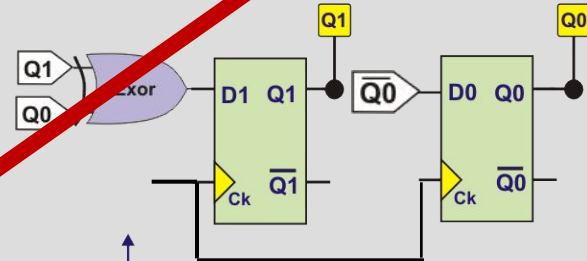
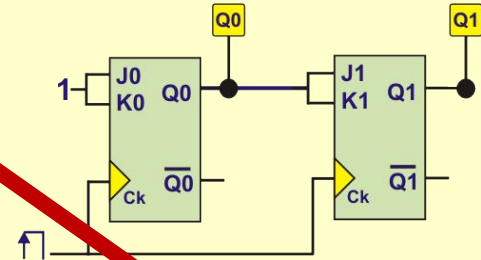
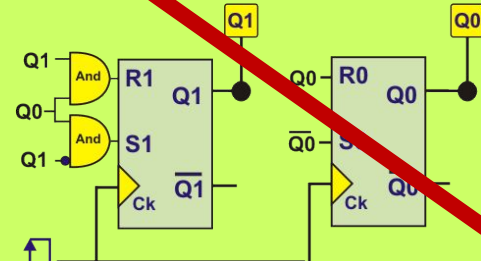
Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

Para Flip Flop D

		Estado Presente		Estado Siguiente		Entradas de Control		Salidas			
m		Q1	Q0	Q1+1	Q0+1	D1	D0	V	F	A	R
0	Verde	0	0	0	1	0	1	1	0	0	0
1	Flecha	0	1	1	0	1	0	0	1	0	0
2	Ámbar	1	0	1	1	1	1	0	0	1	0
3	Rojo	1	1	0	0	0	0	0	0	0	1

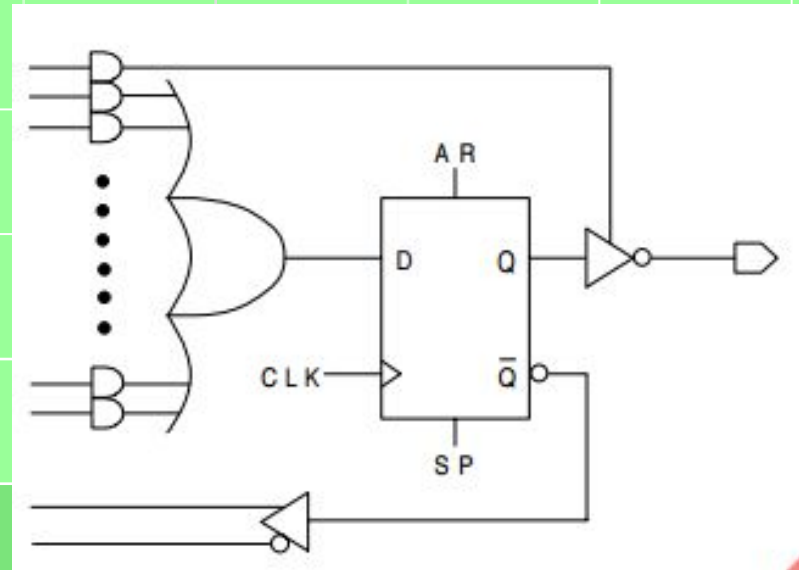
Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
1	1	1

Comparación

FF T	$T = Q_0$		$T_0 = 1$		
FF D	$D_1 = Q_1 \oplus Q_0$		$D_0 = Q_0$		
FF JK	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$	
FF RS	$R_1 = Q_1 Q_0$	$S_1 = Q_1' Q_0$	$R_0 = Q_0$	$R_0 = Q_0'$	



El GAL22V10 tiene 10 FFs tipo D



Istype 'reg'

Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

2.- Sincronización

3.- Asignación de valores a los estados

4.- Diagrama de transición (State_Diagram, Truth Table)

Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

Module sem

“Entrada

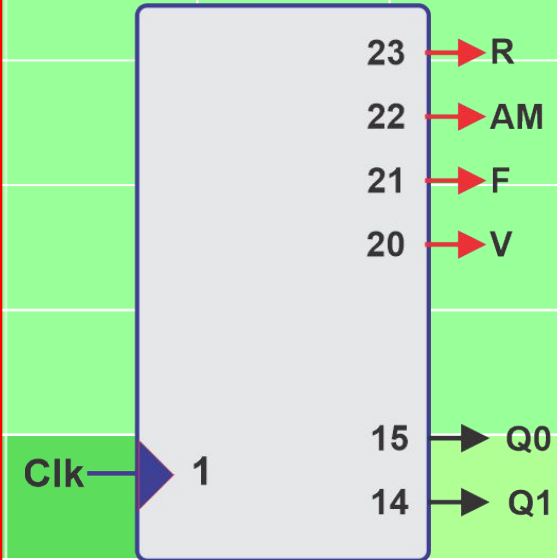
Ck pin 1;

“Salidas Combinacionales

V, F, AM, R pin 20..23 istype ‘com’;

“Salidas Registradas

Q1,Q0 pin 14,15 istype ‘reg’;



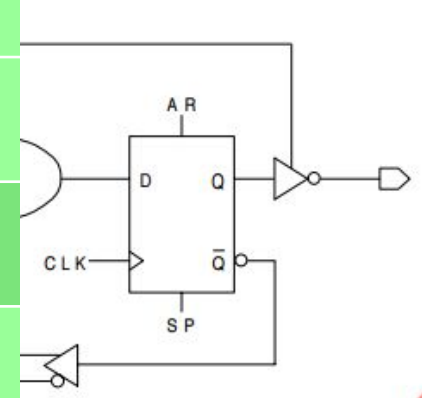
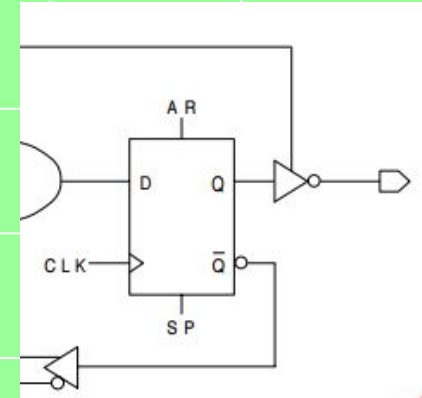
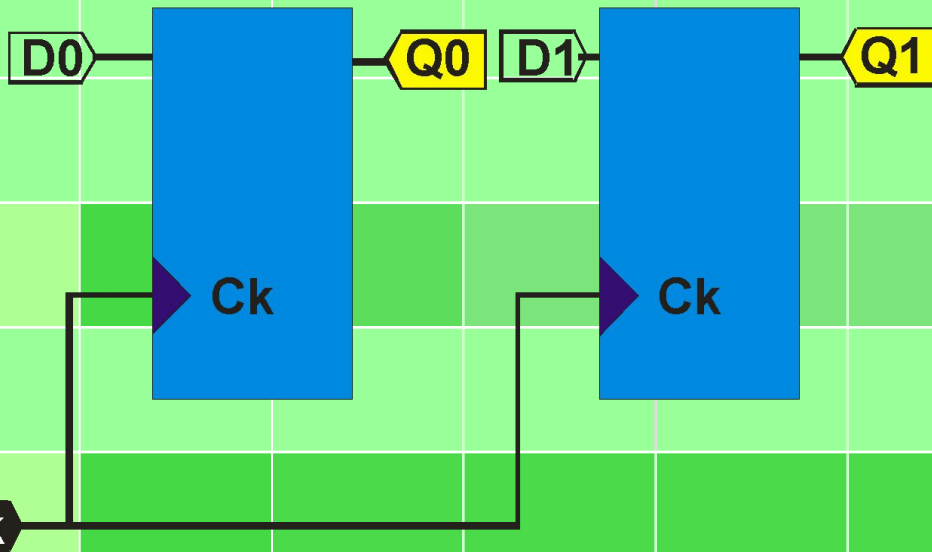
2.- Sincronización

"sincronizacion de los Flip Flops

Sinc=[Q1,Q0];

Equations

Sinc.clk=Ck;



3.- Asignación de valores a los estados

"Asignación de valores a los estados declarations

Verde = [0,0];

Flecha = [0,1];

Ambar = [1,0];

Rojo = [1,1];

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1

4.- Diagrama de transición

"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:

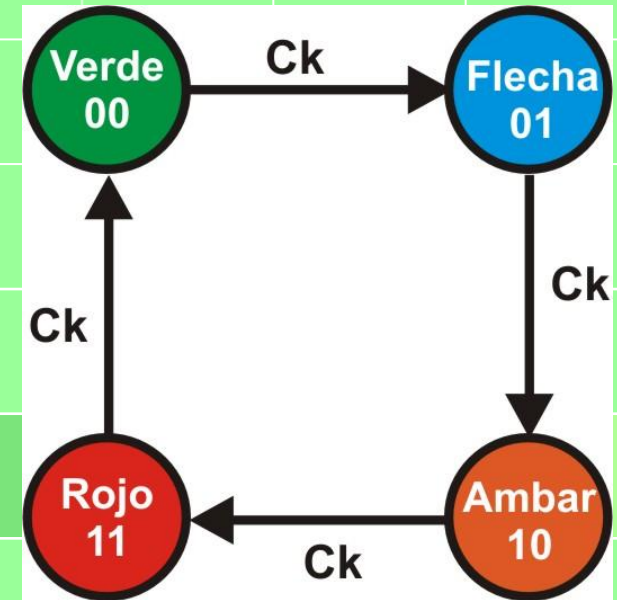
Ve=0; Fl=0; Am=1; Ro=0;

goto Rojo;

STATE Rojo:

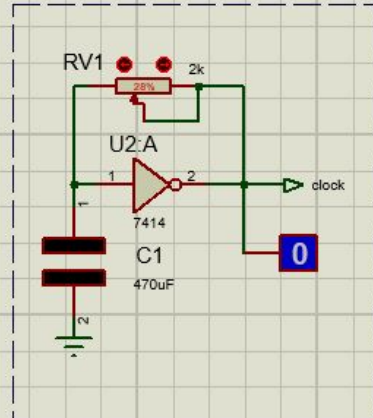
Ve=0; Fl=0; Am=0; Ro=1;

goto Verde;

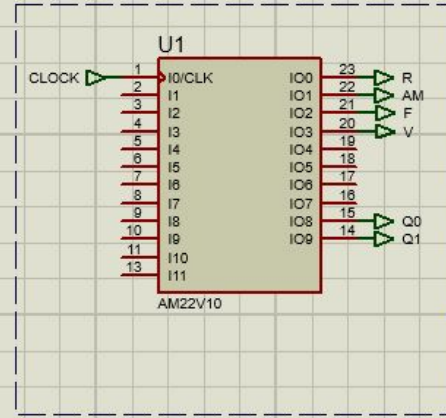


Proteus

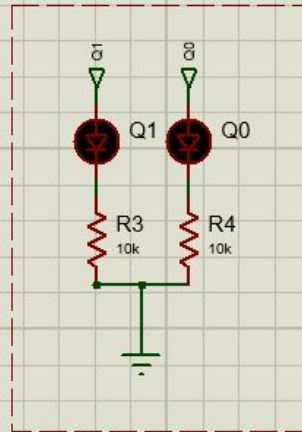
Generador de pulsos



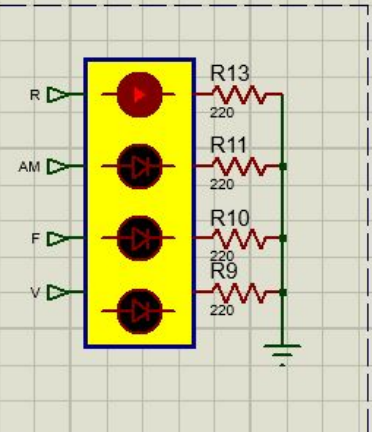
PLD 22v10



Salidas Registradas

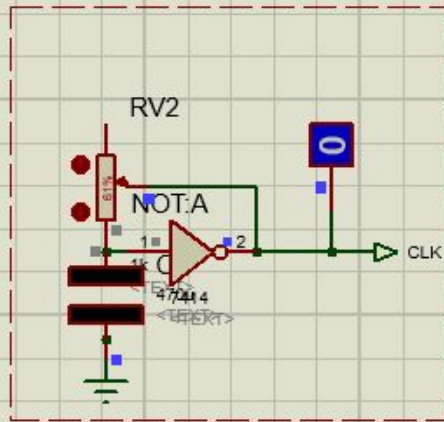


Salidas Combinacionales

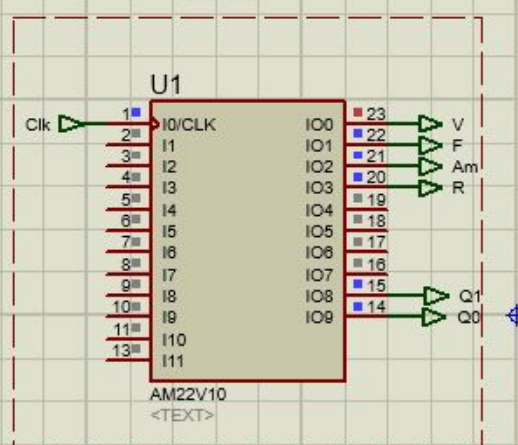


Semaforo

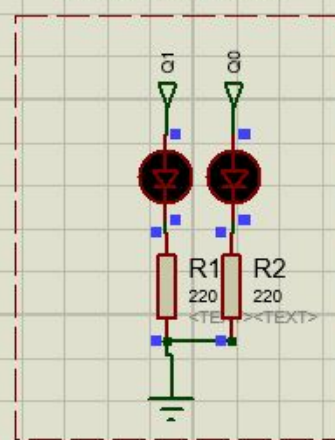
Entrada



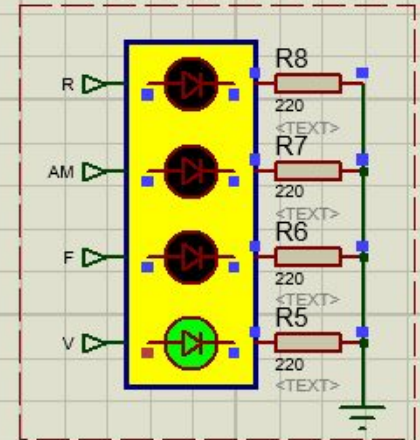
PLD



Salidas Registradas



Salidas Combinacionales



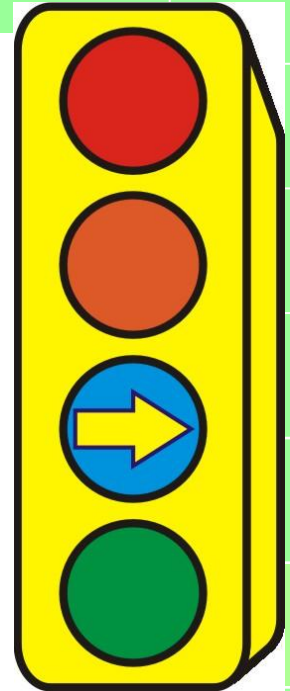
Tradicional	Programación
1.- Especificar el Sistema (Diagrama de Transición).	
2.- Determinar la cantidad de Flip Flops.	
3.- Asignar los valores a los estados.	
4.- Determinar las entradas y salidas. a) Entrada de sincronía reloj. b) Entradas combinacionales. c) Salidas combinacionales. d) Salidas registradas (FF's).	
5.- Construir una Tabla de Estado siguiente. valores de los estados en código binario y la obtención de las entradas de control que están basadas en las tablas de excitación RS, JK, T y D.	5.- Construir una Tabla de Estado siguiente. Solo nombres.
6.- Obtener las ecuaciones mínimas. Y seleccionar el tipo o tipos de FF's mas económico, Máximo 5 variables.	6.- Código en Abel-HDL a) Entradas y salidas b) Sincronización de los Flip Flops c) Asignación de valores a los estados d) Definir la secuencia (state_diagram o Truth_table) e) Simulación
7.- Diagrama Esquemático	
8.- Implementación varios CI de Función fija	7.- Implementación. DLP (1 solo CI)

Método para el Diseño de un sistema Secuencial

- 1.- Especificar el Sistema (Diagrama de Transición).
- 2.- Determinar la cantidad de Flip Flops.
- 3.- Asignar los valores a los estados.
- 4.- Determinar las entradas y salidas.
 - a) Entrada de sincronía reloj.
 - b) Entradas combinacionales.
 - c) Salidas combinacionales.
 - d) Salidas registradas (FF's).
- 5.- Construir una Tabla de Estados.
- 6.- Código en ABEL_HDL
- 7.- Diagrama Esquemático.
- 8.- Simulación
- 9.- Implementación.

Método para el Diseño de un sistema Secuencial

Diseñe un Sistema Secuencial síncrono que represente la operación de un semáforo de cuatro estados que se presentan en el siguiente orden: **Verde**, **Flecha**, **Ámbar** y **Rojo**, además incluya una Entrada **X** de modo que:



Si $X=1$, el sistema deberá de permanecer en el mismo estado
Si $X=0$, el sistema deberá de cambiar al estado siguiente

X

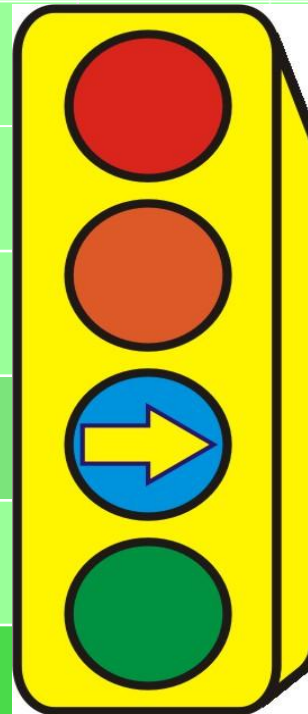
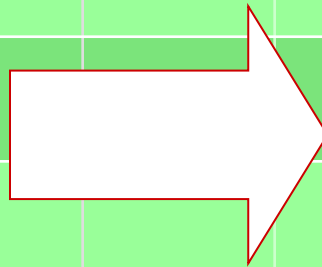
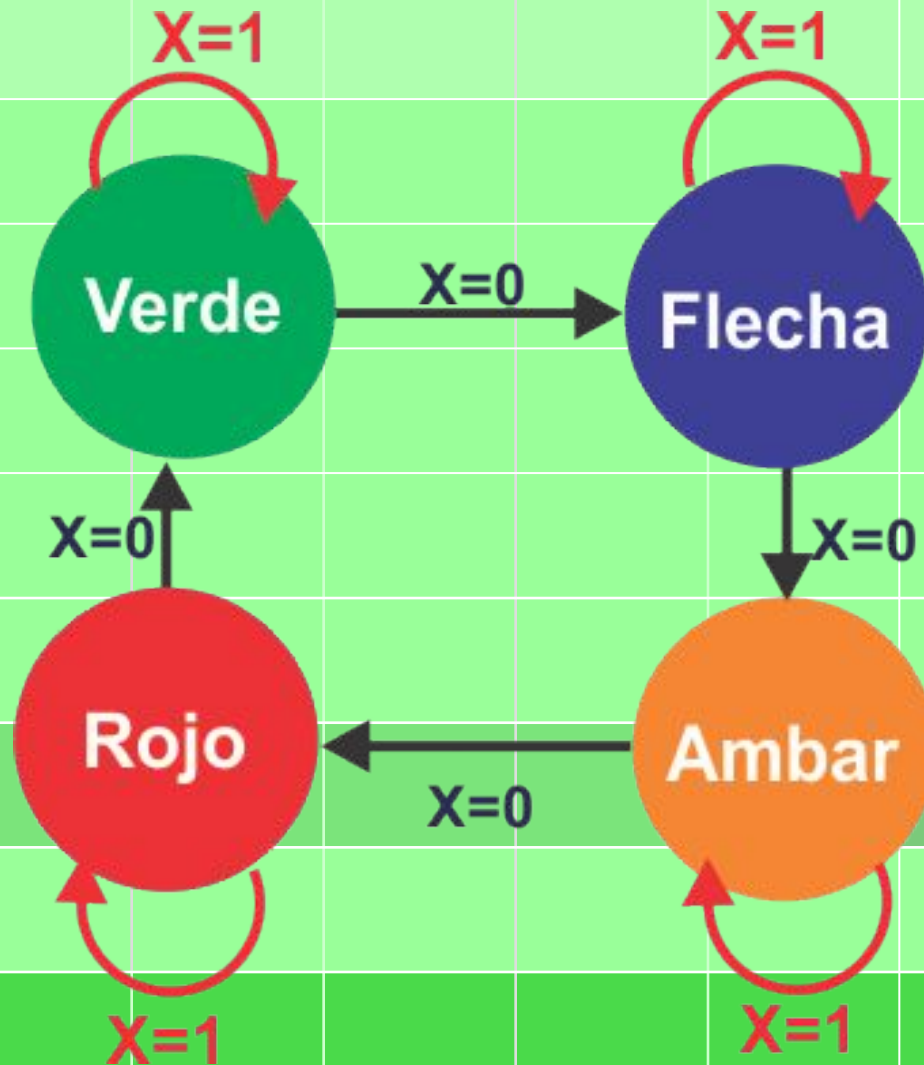
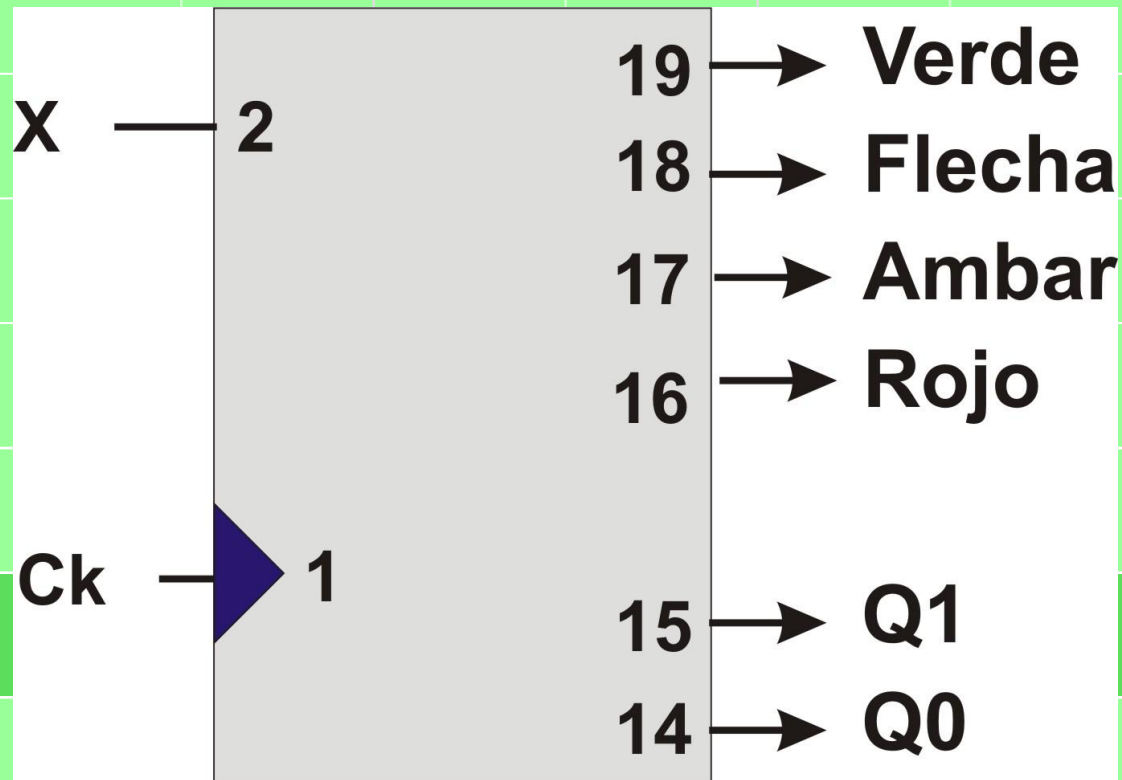


Diagrama de transición





Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1

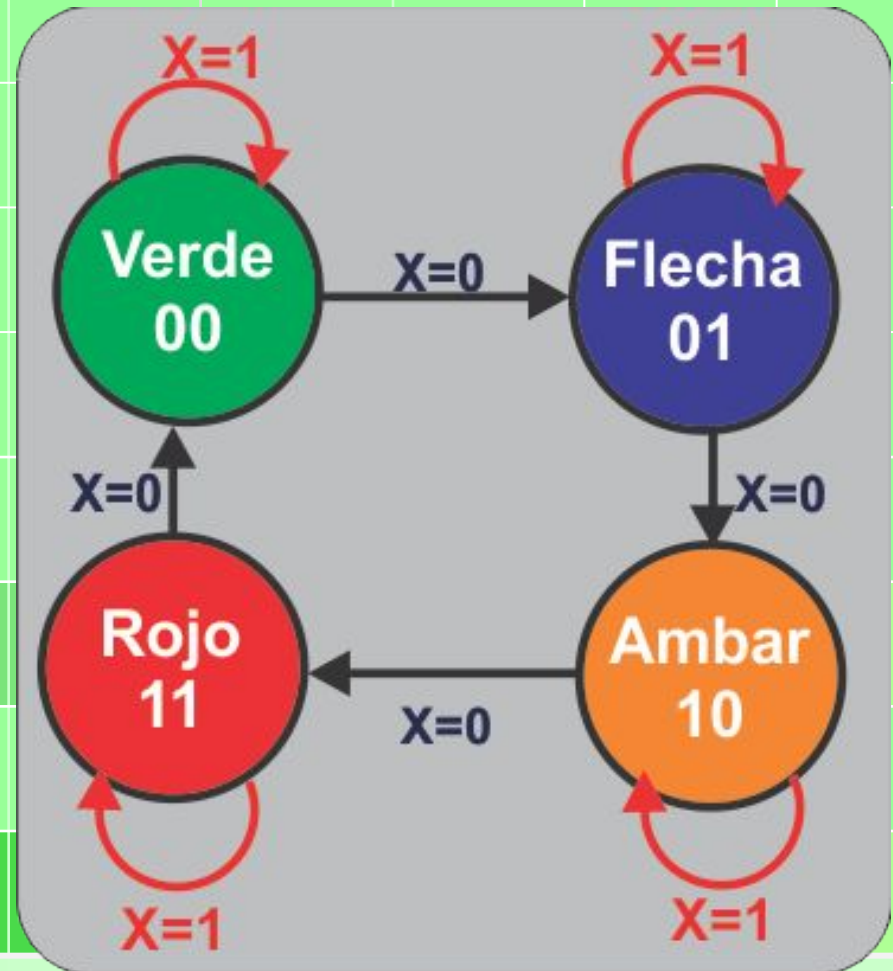


Tabla de estados

Estado Presente	Estado Siguiente	
	X=0	X=1
Verde	Flecha	Verde
Flecha	Ámbar	Flecha
Ámbar	Rojo	Ámbar
Rojo	Verde	Rojo

Tabla de estados

Estado Presente	Estado Siguiente		Flip Flops		Salidas Combinacionales			
	X=0	X=1	Q1	Q0	VE	FL	AM	RO
Verde	Flecha	Verde	0	0	1	0	0	0
Flecha	Ámbar	Flecha	0	1	0	1	0	0
Ámbar	Rojo	Ámbar	1	0	0	0	1	0
Rojo	Verde	Rojo	1	1	0	0	0	1

Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

Module sem

“Entrada

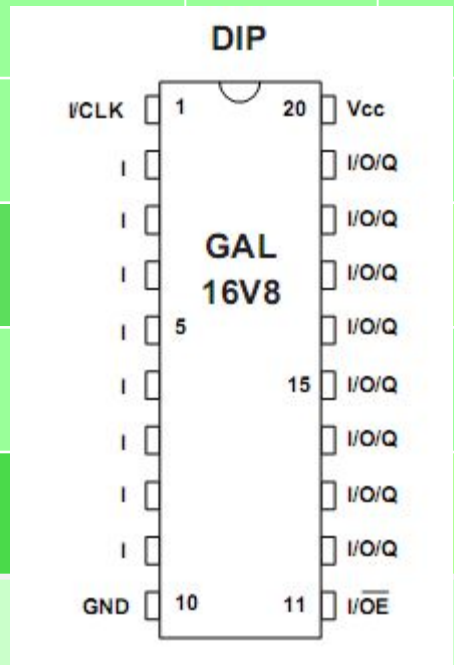
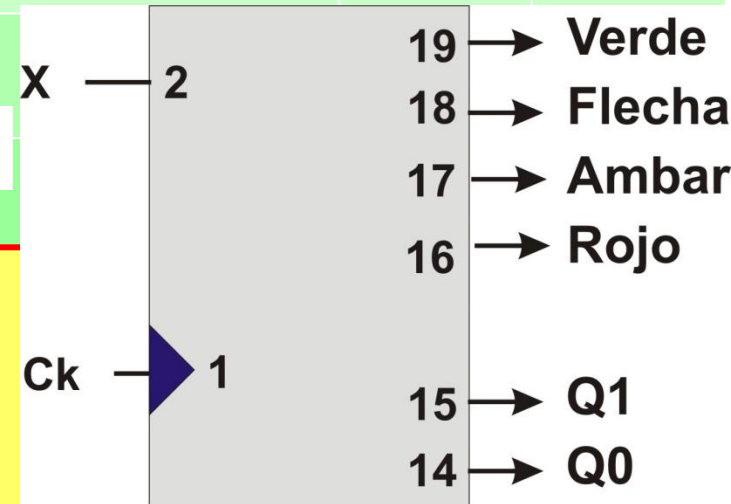
Ck, X, Rst pin 1,2,3;

“Salidas Combinacionales

Ve, Fl, Am, Ro pin 19..16 istype ‘com’;

“Salidas Registradas

Q1,Q0 pin 15,14 istype ‘reg’;



Sincronización

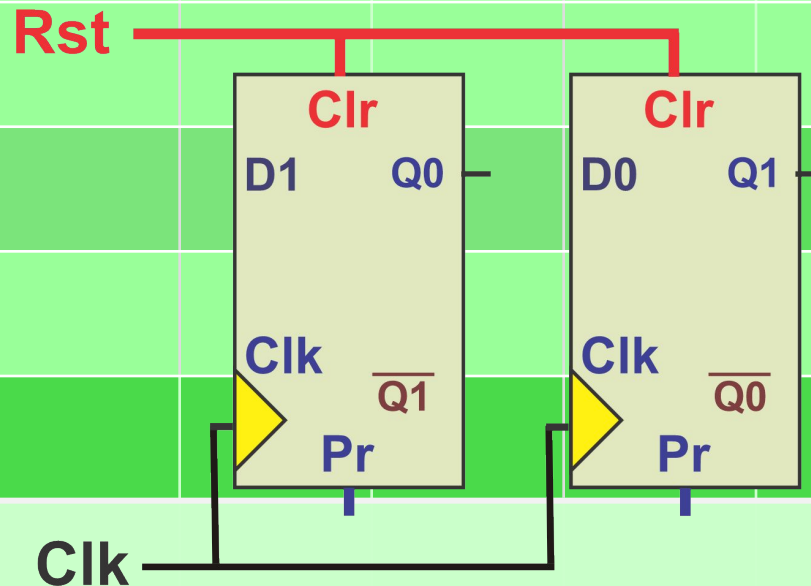
"sincronización de los Flip Flops

Sinc=[Q1,Q0];

Equations

Sinc.clk=Ck;

Sinc.ar=Rst;



Asignación de valores a los estados

"Asignación de valores a los estados declarations

Verde = [0,0];

Flecha = [0,1];

Ambar = [1,0];

Rojo = [1,1];

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1

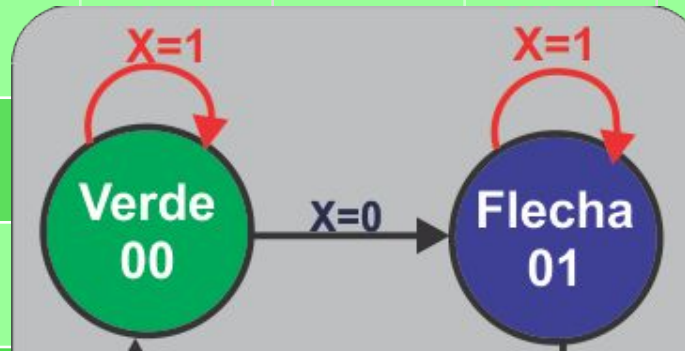
Descripción de la Secuencia

State_diagram

If-Then-Else Syntax

IF exp THEN state_exp [ELSE
state_exp]

Las sentencias **if-then-else** se utilizan en la sección **state_diagram** para describir la progresión de un estado a otro



Descripción de la Secuencia

State_diagram

"Diagrama de transición

State_Diagram Sinc

STATE Verde:

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

If X then Verde else Flecha

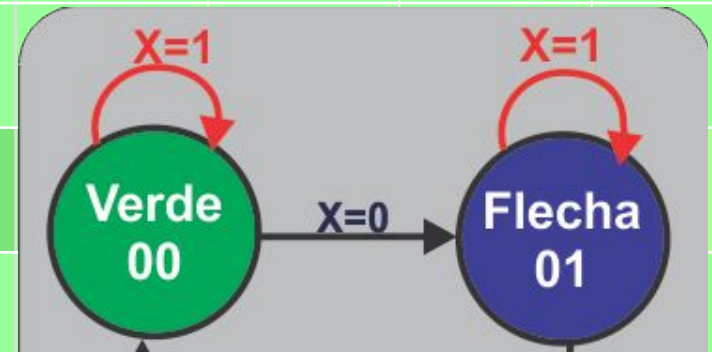
"Diagrama de transición

State_Diagram Sinc

STATE Verde:

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

If !X then Flecha else Verde



Descripción de la Secuencia

State_diagram

"Diagrama de transición

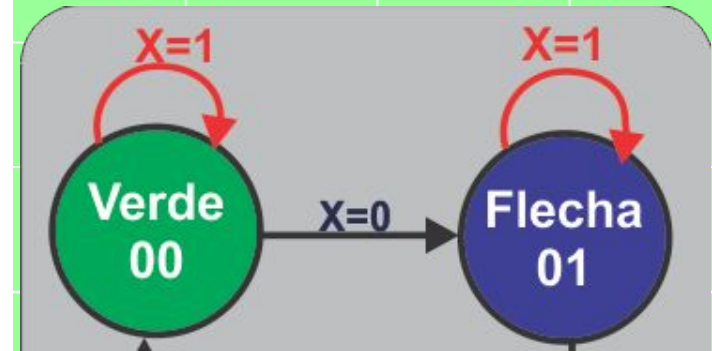
State_Diagram Sinc

STATE Verde:

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

If X then Verde;

IF !X then Flecha;



Descripción de la Secuencia

State_diagram

"Diagrama de transición

State_Diagram Sinc

STATE Verde:

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

If X==1 then Verde;

IF X==0 then Flecha;

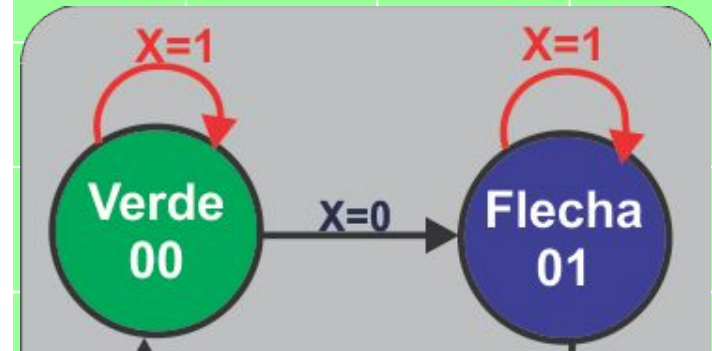


Diagrama de transición

"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;

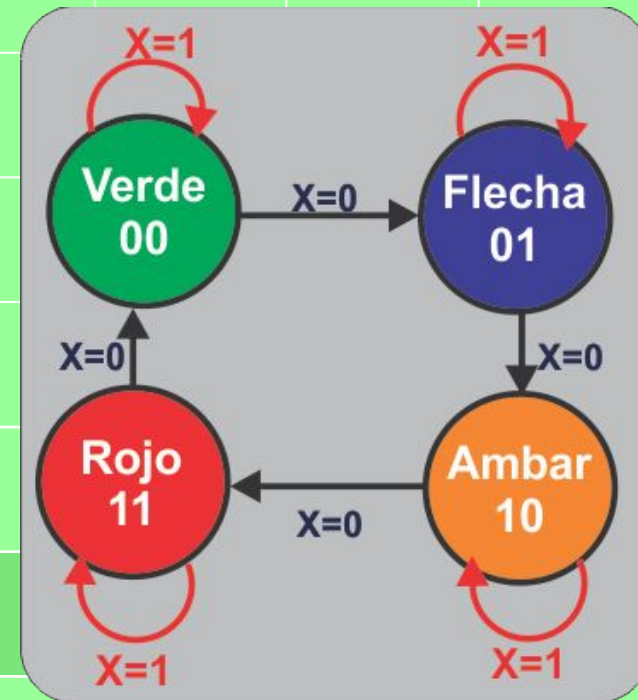
If X==0 then Rojo else Ambar;

STATE Rojo:

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

If X==1 then Rojo;

If X==0 then Verde;



Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

Module sem

“Entrada

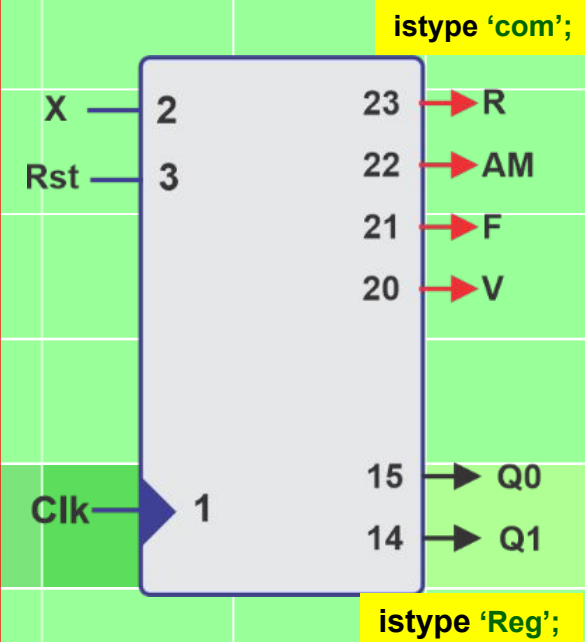
Ck, X, Rst pin 1..3;

“Salidas Combinacionales

Ve, Fl, Am, Ro pin 20..23 istype ‘com’;

“Salidas Registradas

Q1,Q0 pin 14,15 istype ‘reg’;



Sincronización

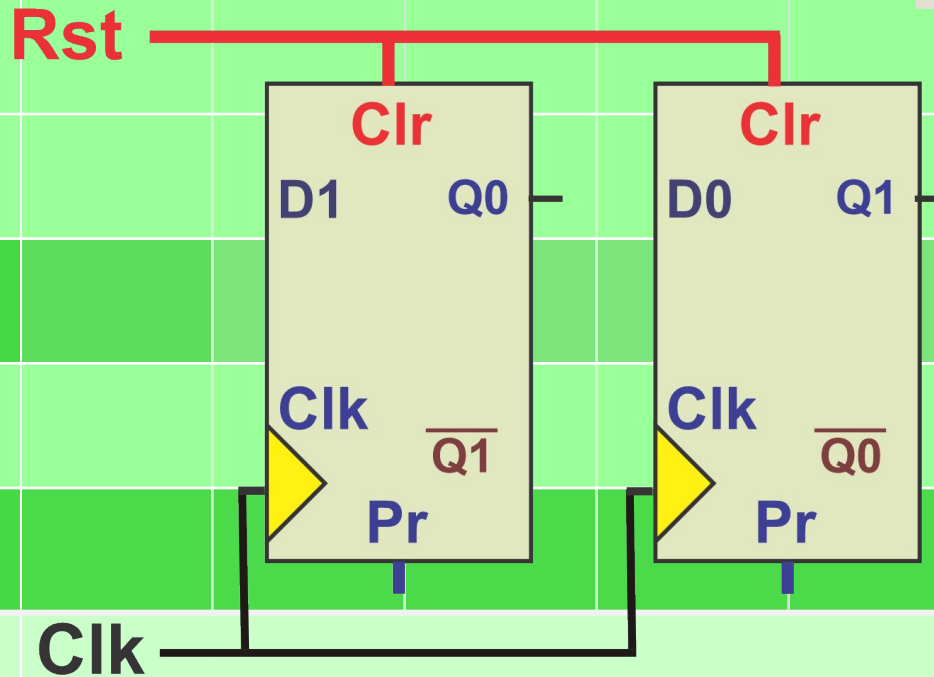
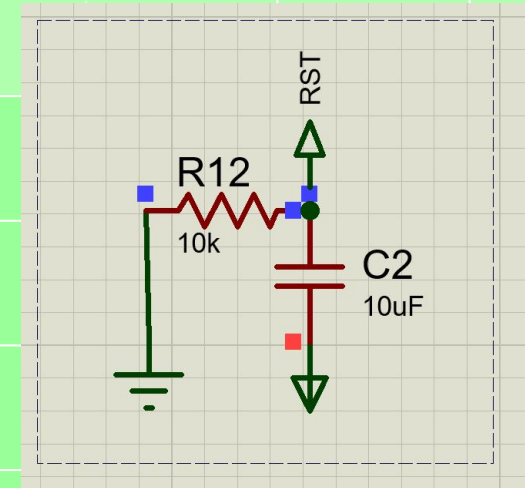
"sincronización de los Flip Flops

$\text{Sinc}=[Q1,Q0];$

Equations

$\text{Sinc.clk}=\text{Ck};$

$\text{Sinc.ar}=\text{Rst};$



Sincronización

Module sem

“Entrada

Ck, X, Rst, P pin 1..4;

“Salidas Combinacionales

Ve, Fl, Am, Ro pin 20..23 istype ‘com’;

“Salidas Registradas

Q1,Q0 pin 14,15 istype ‘reg’;

"sincronización de los Flip Flops

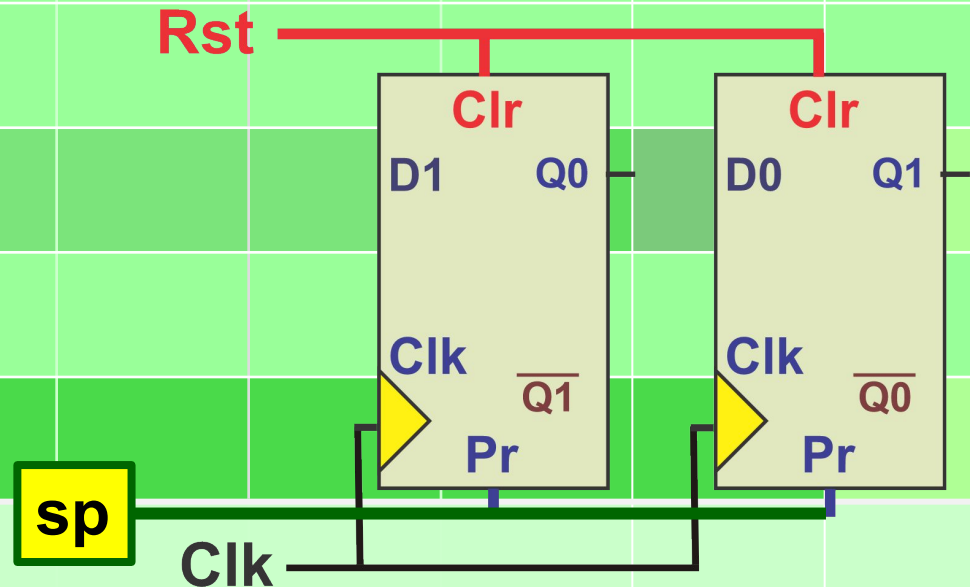
Sinc=[Q1,Q0];

Equations

Sinc.clk=Ck;

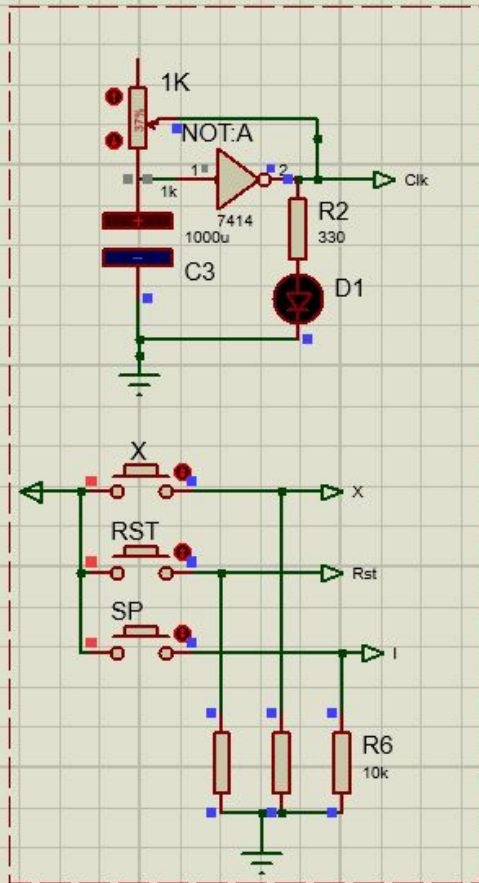
Sinc.ar=Rst;

Sinc.sp=P;

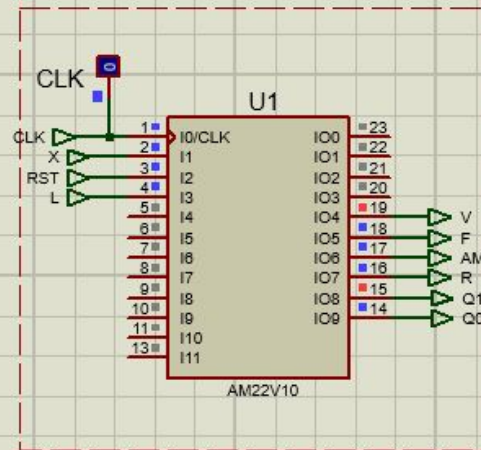


PROTEUS

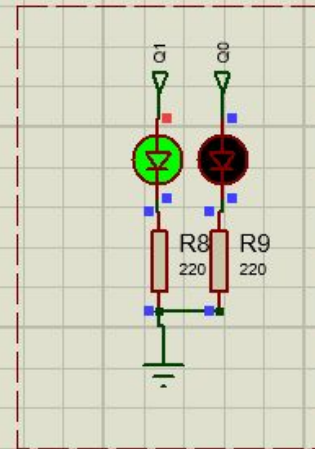
Entradas



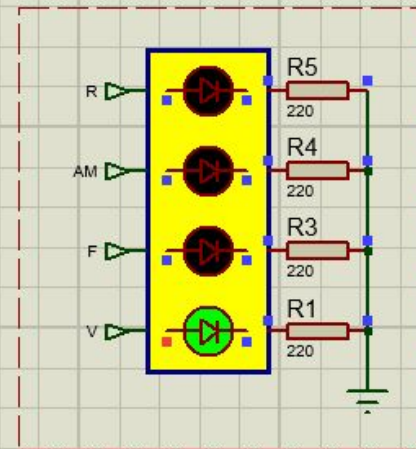
PLD



Salidas Registradas



Salidas Combinacionales



Equations
Sinc.clk=Ck;
Sinc.ar=Rst;

Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

2.- Sincronización

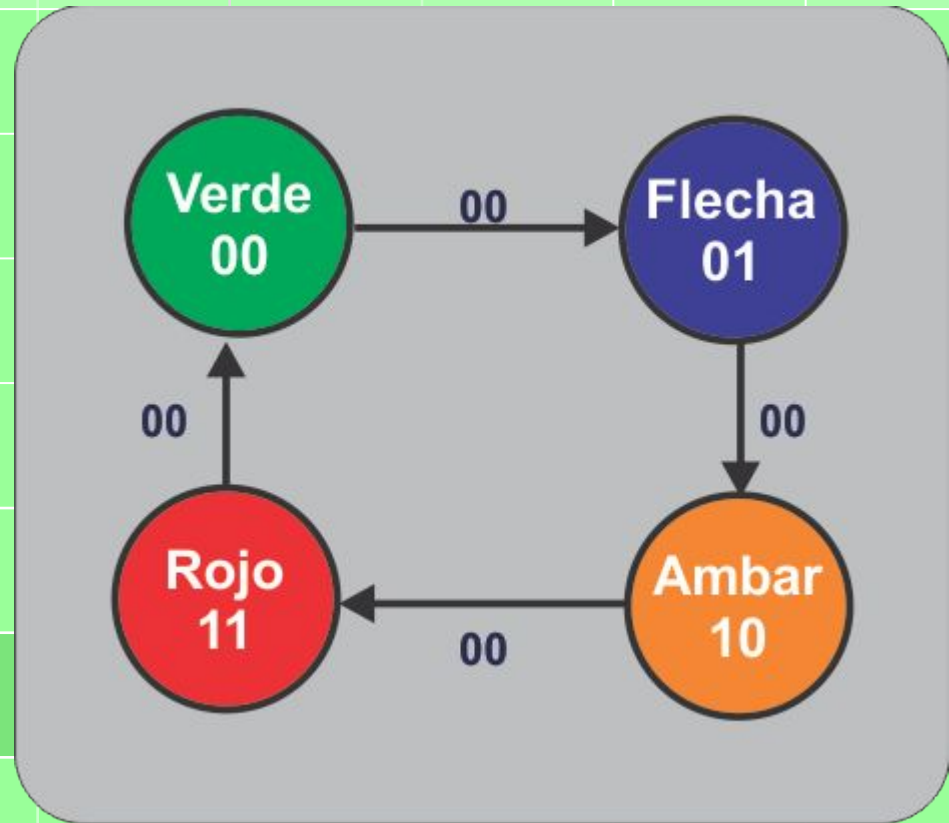
3.- Asignación de valores a los estados

4.- Diagrama de transición (State_Diagram, Truth Table)

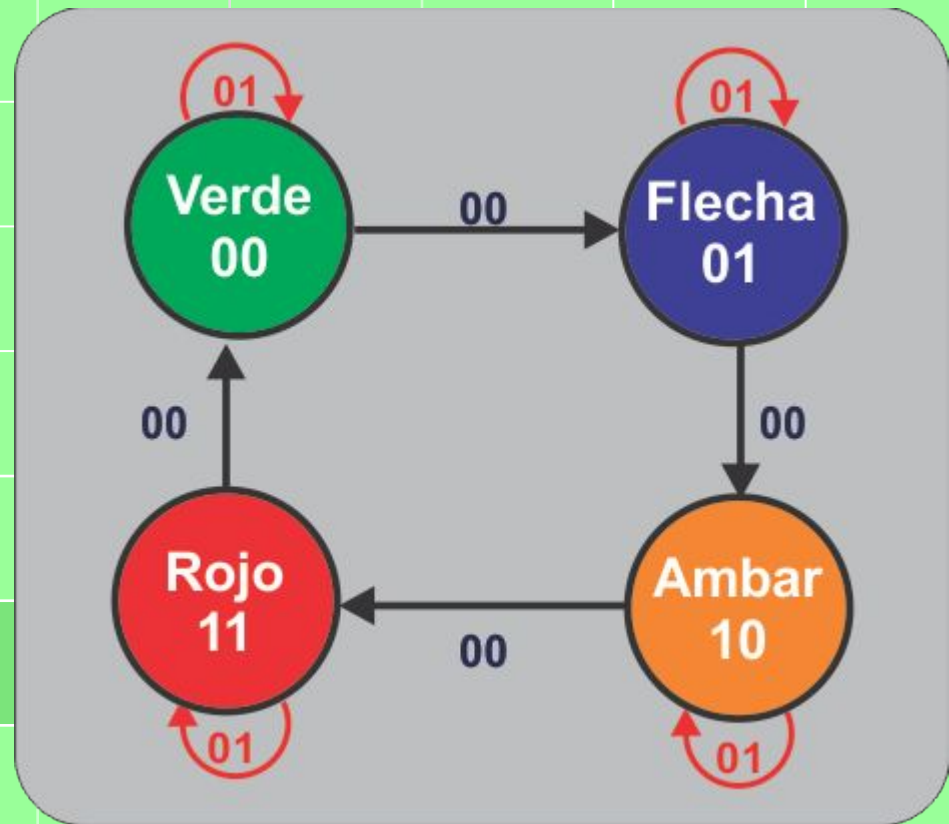
Ejemplo 3

X	Y	Acción
0	0	Cambiar al estado siguiente
0	1	Permanecer en el mismo estado
1	0	Avanzar dos estados
1	1	Retroceder un estado

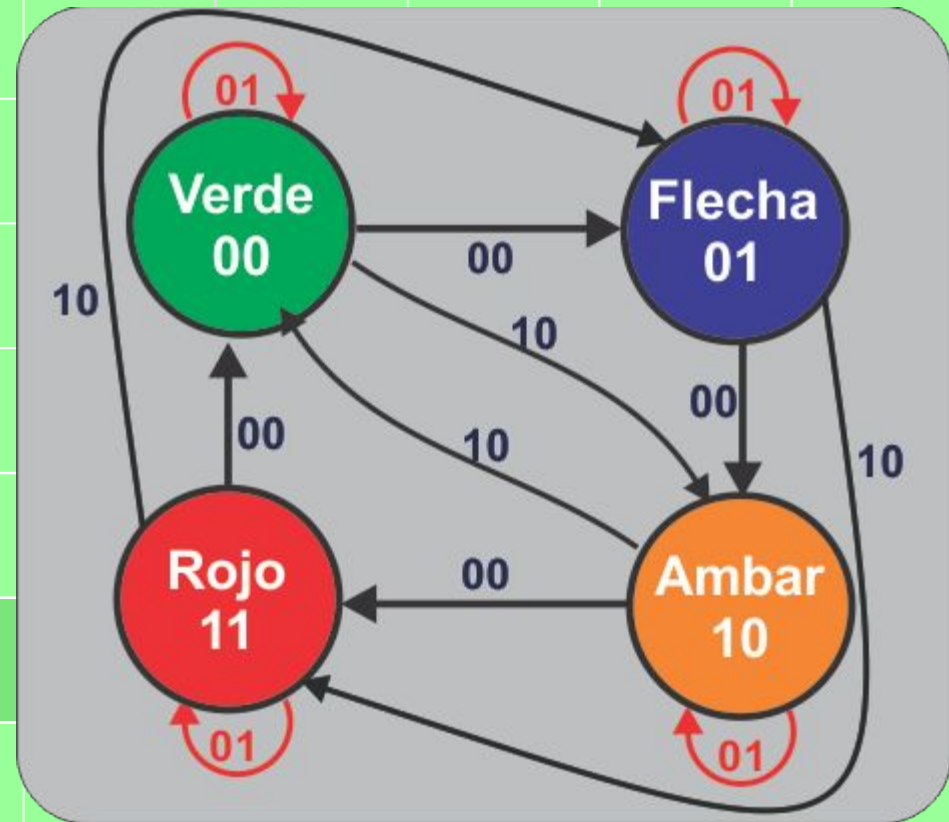
X	Y	ACCION
0	0	Cambiar al estado siguiente
0	1	Permanecer en el mismo estado
1	0	Avanzar dos estados
1	1	Retroceder un estado



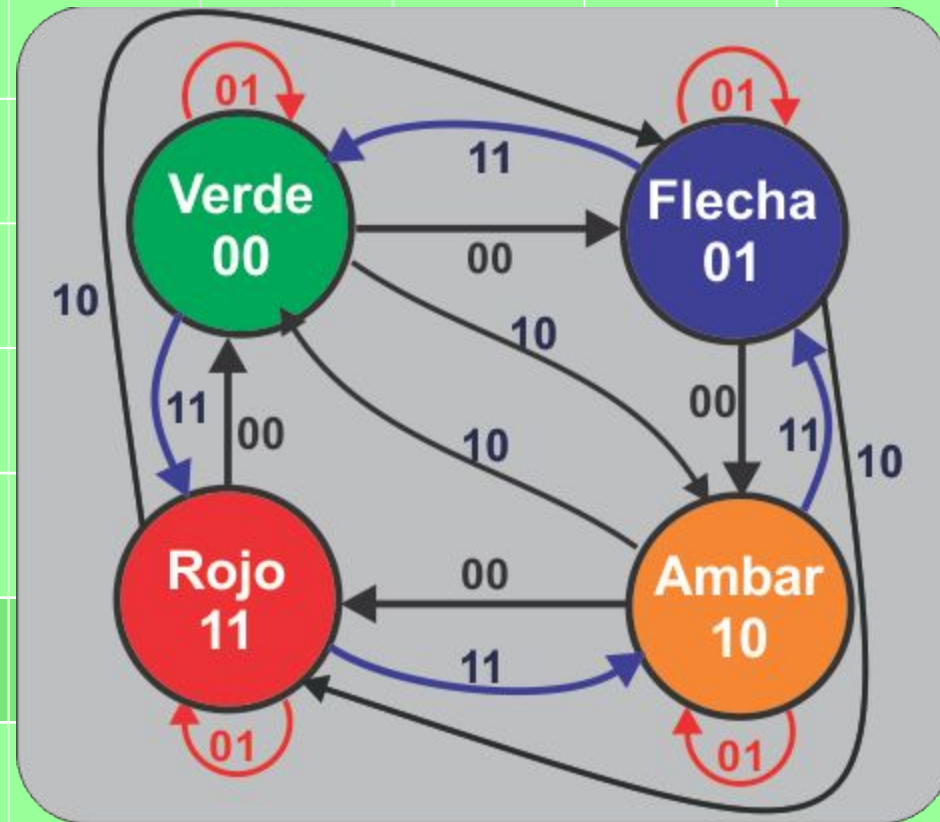
X	Y	ACCION
0	0	Cambiar al estado siguiente
0	1	Permanecer en el mismo estado
1	0	Avanzar dos estados
1	1	Retroceder un estado



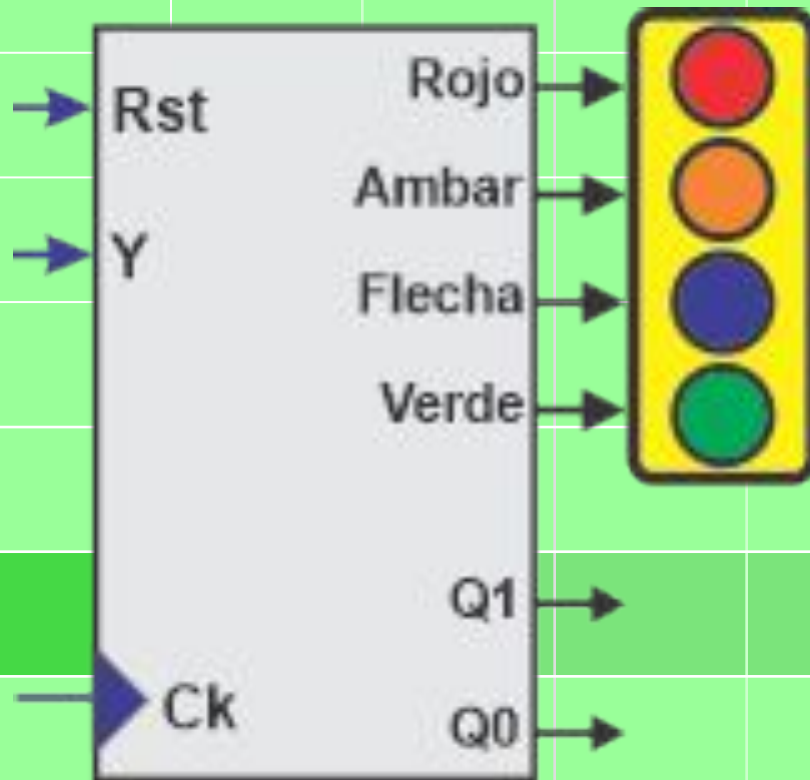
X	Y	ACCION
0	0	Cambiar al estado siguiente
0	1	Permanecer en el mismo estado
1	0	Avanzar dos estados
1	1	Retroceder un estado



X	Y	ACCION
0	0	Cambiar al estado siguiente
0	1	Permanecer en el mismo estado
1	0	Avanzar dos estados
1	1	Retroceder un estado



Entradas y Salidas



Estado Presente	Estado siguiente			
	00	01	10	11
Verde				
Flecha				
Ámbar				
Rojo				

Estado Presente	Estado siguiente			
	00	01	10	11
Verde	Flecha			
Flecha	Ámbar			
Ámbar	Rojo			
Rojo	Verde			

Estado Presente	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde		
Flecha	Ámbar	Flecha		
Ámbar	Rojo	Ámbar		
Rojo	Verde	Rojo		

Estado Presente	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	
Flecha	Ámbar	Flecha	Rojo	
Ámbar	Rojo	Ámbar	Verde	
Rojo	Verde	Rojo	Flecha	

Estado Presente	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	Rojo
Flecha	Ámbar	Flecha	Rojo	Verde
Ámbar	Rojo	Ámbar	Verde	Flecha
Rojo	Verde	Rojo	Flecha	Ámbar

Tabla de estado siguiente

Estado Presente	Estado siguiente				Salidas			
	00	01	10	11	V	F	A	R
Verde	Flecha	Verde	Ámbar	Rojo	1	0	0	0
Flecha	Ámbar	Flecha	Rojo	Verde	0	1	0	0
Ámbar	Rojo	Ámbar	Verde	Flecha	0	0	1	0
Rojo	Verde	Rojo	Flecha	Ámbar	0	0	0	1

Tabla de estado siguiente

Estado Presente	Estado siguiente				Flip Flops		Salidas			
	00	01	10	11	Q1	Q0	V	F	A	R
Verde	Flecha	Verde	Ámbar	Rojo	0	0	1	0	0	0
Flecha	Ámbar	Flecha	Rojo	Verde	0	1	0	1	0	0
Ámbar	Rojo	Ámbar	Verde	Flecha	1	0	0	0	1	0
Rojo	Verde	Rojo	Flecha	Ámbar	1	1	0	0	0	1

Código en Abel-Hdl

1.- Diagrama de Bloques Entradas y Salidas

Module semxy

“Entradas

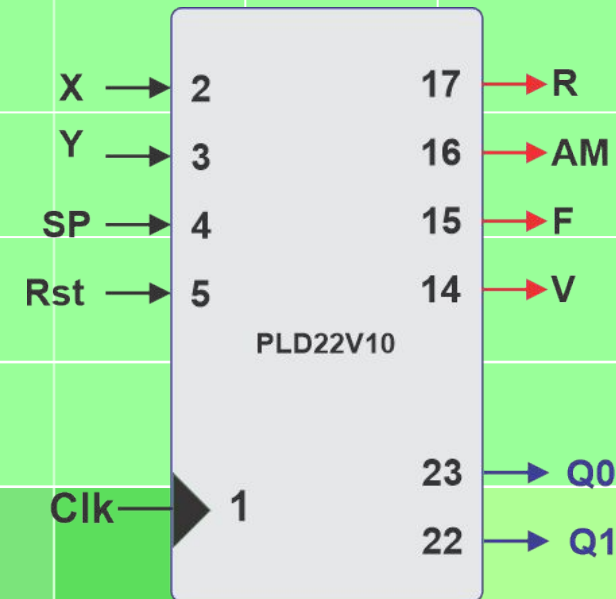
Clk,X,Y,SP,Rst pin 1..5;

“Salidas Combinacionales

R, AM, F, V pin 17..14 istype ‘com’;

“Salidas Registradas

Q1,Q0 pin 22,23 istype ‘reg’;



Sincronización

"sincronizacion de los Flip Flops

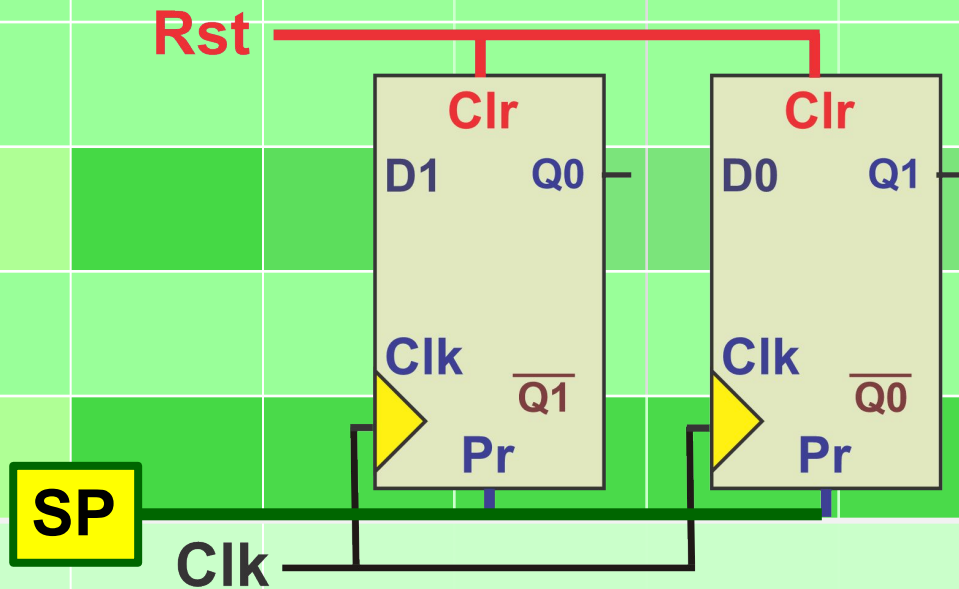
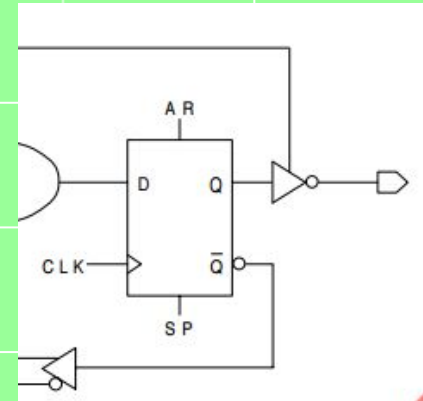
FIME=[Q1,Q0];

Equations

FIME.clk=Clk;

FIME.ar=Rst;

FIME.sp=SP;



Asignación de valores a los estados

"Asignación de valores a los estados declarations

Verde=[0,0];

Flecha=[0,1];

Ambar=[1,0];

Rojo=[1,1];

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ámbar	1	0
Rojo	1	1

State Verde:

State_Diagram FIME

State Verde:

V=1; F=0; AM=0; R=0;

if !X&!Y then Flecha;

if !X & Y then Verde;

if X & !Y then Ambar;

if X & Y then Rojo;

E P	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	Rojo
Flecha	Ámbar	Flecha	Rojo	Verde
Ámbar	Rojo	Ámbar	Verde	Flecha
Rojo	Verde	Rojo	Flecha	Ámbar

State Flecha:

State Flecha:

V=0; F=1; AM=0; R=0;

if !X & !Y then Ambar;

if !X & Y then Flecha;

if X & !Y then Rojo;

if X & Y then Verde;

E P	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	Rojo
Flecha	Ámbar	Flecha	Rojo	Verde
Ámbar	Rojo	Ámbar	Verde	Flecha
Rojo	Verde	Rojo	Flecha	Ámbar

State Ambar:

State Ambar:

V=0; F=0; AM=1; R=0;

if !X & !Y then Rojo;

if !X & Y then Ambar;

if X & !Y then Verde;

if X & Y then Flecha;

E P	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	Rojo
Flecha	Ámbar	Flecha	Rojo	Verde
Ámbar	Rojo	Ámbar	Verde	Flecha
Rojo	Verde	Rojo	Flecha	Ámbar

State Rojo:

State Rojo:

V=0; F=0; AM=0; R=1;

if **!X** & **!Y** then Verde;

if **!X** & Y then Rojo;

if X & **!Y** then Flecha;

if X & Y then Ambar;

E P	Estado siguiente			
	00	01	10	11
Verde	Flecha	Verde	Ámbar	Rojo
Flecha	Ámbar	Flecha	Rojo	Verde
Ámbar	Rojo	Ámbar	Verde	Flecha
Rojo	Verde	Rojo	Flecha	Ámbar

M=[X,Y];

State Verde:

V=1; F=0;

AM=0; R=0;

if **M==0** then Flecha;

if **M==1** then Verde;

if **M==2** then Ambar;

if **M==3** then Rojo;

State Flecha:

V=0; **F=1;**

AM=0; R=0;

if **M==0** then Ambar;

if **M==1** then Flecha;

if **M==2** then Rojo;

if **M==3** then Verde;

State Ambar:

V=0; F=0;

AM=1; R=0;

if **M==0** then Rojo;

if **M==1** then Ambar;

if **M==2** then Verde;

if **M==3** then Flecha;

State Rojo:

V=0; F=0;

AM=0; **R=1;**

if **M==0** then Verde;

if **M==1** then Rojo;

if **M==2** then Rojo;

if **M==3** then Ambar;

Comparación de métodos o flujo de diseño

Método Tradicional

- 1.- Especificar el Sistema
(Diagrama de Transición).
- 2.- Determinar la cantidad de Flip Flops.
- 3.- Asignar los valores a los estados.
- 4.- Determinar las entradas y salidas.
 - a) Entrada de sincronía reloj.
 - b) Entradas combinacionales.
 - c) Salidas combinacionales.
 - d) Salidas registradas (FF's).
- 5.- Construir la Tabla de Estado siguiente.
- 6.- Ecuaciones mínimas.
- 7.- Diagrama Esquemático.
- 8.- Implementación.

Método con HDL

- 1.- Especificar el sistema
(Diagrama de transición)
- 2.- Determinar la cantidad de Flip Flops
- 3.- Asignar valores a los estados
- 4.- Diagrama de Bloque (entradas y salidas)
- 5.- Construir la tabla de estado siguiente
- 6.- Codificación en ABEL-HDL**
 - a) Entradas y salidas**
 - b) Sincronización de los Flip Flops**
 - c) Asignación de valores a los estados**
 - d) Definir la secuencia**
(state_diagram o Truth_table)
 - e) Simulación**
- 7.- Implementación

Actividad Fundamental 4

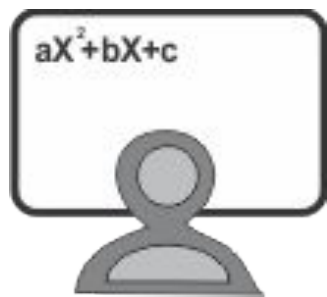
Aplicación del método para el Diseño de sistemas secuenciales síncronos

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) concluye su desarrollo asesorado por el profesor.

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

Aula Invertida



Estudio

Video

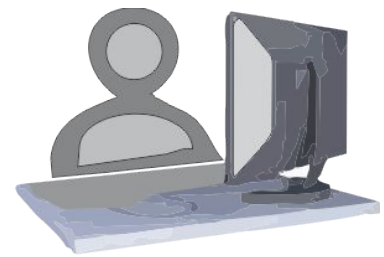


Actividad

Personalizada

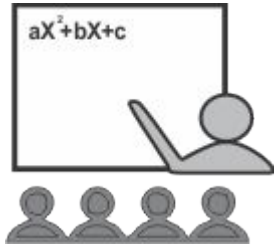


Asesoría

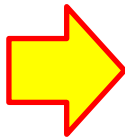


Evaluación

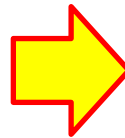
Método Tradicional



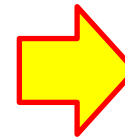
Clase



Actividad

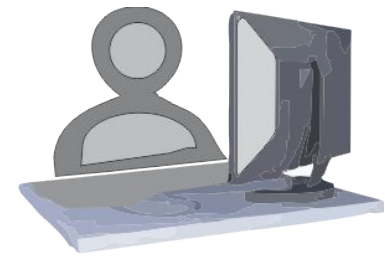
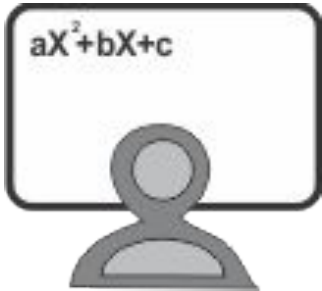


Estudio



Evaluación

Aula Invertida



Estudio



Actividad



Asesoría



Evaluación

Actividad Fundamental 4

Aplicación del método para el Diseño de sistemas secuenciales síncronos

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

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

?

Resumen

Introducción y fundamentos del diseño secuencial

1. Diferencia entre **sistemas combinacionales y secuenciales**.
2. **Sincronía** mediante la señal de reloj (*Clk*) y el papel de los *Flip-Flops*.
3. Concepto de **estado presente y estado siguiente** como base del comportamiento temporal.
4. **Método general de diseño:**
 - a) Especificación
 - b) tabla de estado siguiente
 - c) codificación HDL
 - d) Simulación
 - e) implementación.
- 5.- **Lenguaje ABEL-HDL** para describir secuencias empleando los comandos **GOTO**, **IF THEN** y **IF THEN ELSE**, como recursos fundamentales para definir la transición entre estados y condicionar el flujo lógico del sistema.