

Diseño de Sistemas Secuenciales Síncronos

**"La tecnología ya es parte del presente educativo;
usemos su potencial para transformar
la enseñanza y el aprendizaje."**

Salón 1-201, UANL, FIME marzo 2003



**"Innovar en la enseñanza implica más que
incorporar tecnologías; requiere
adaptarse continuamente a las
necesidades para formar ingenieros
acorde al perfil de egreso y las tendencias
globales."**

"JAGG"

JAGG

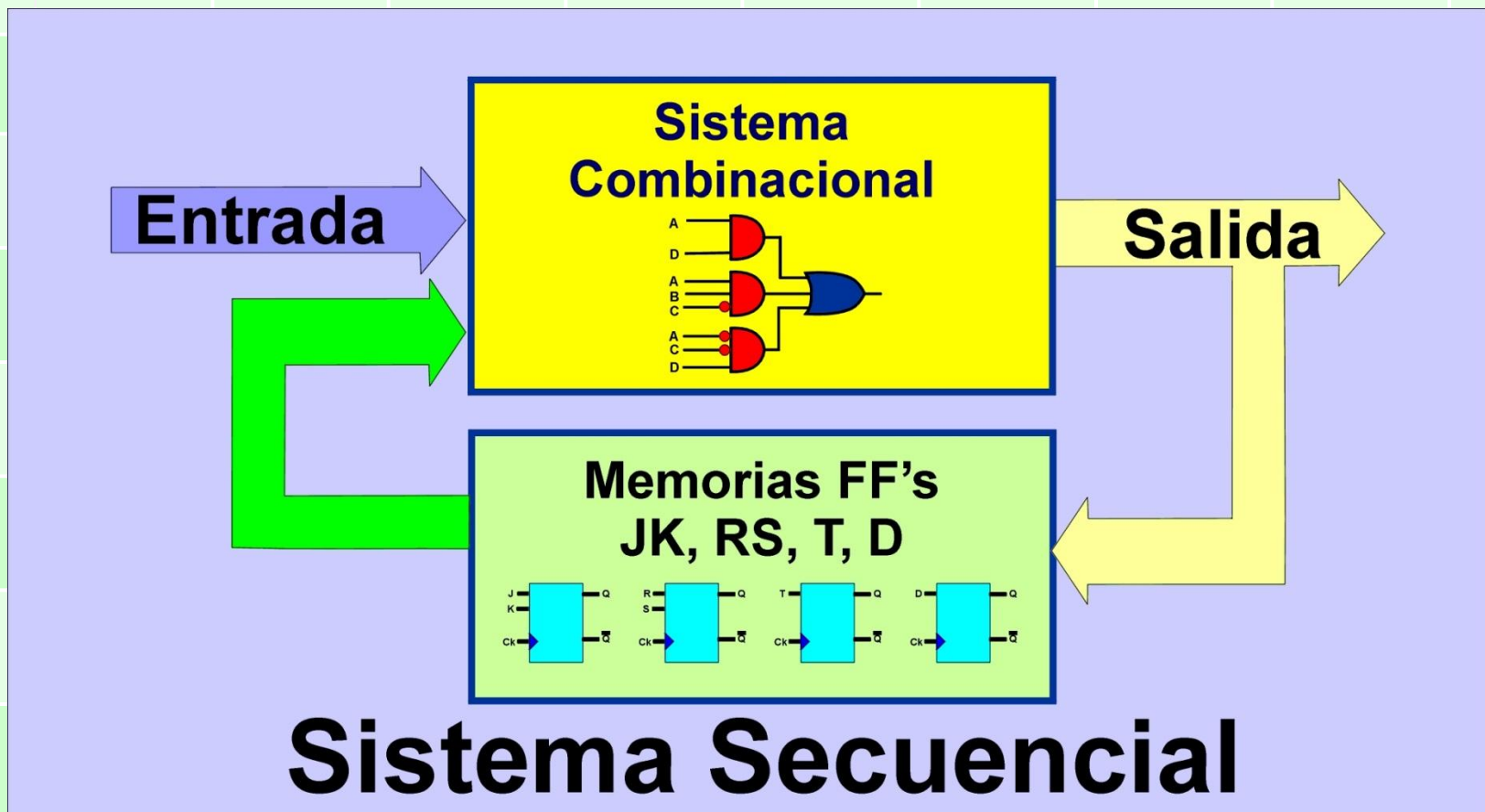
	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	<i>Miércoles 8 de octubre</i>
PF7	Diseño Combinacional con HDL	F	<i>Viernes 10 de octubre</i>
PF8	Multiplexor	5	<i>Miércoles 15 de octubre</i>
AF3	Decodificador con Display	10	<i>Lunes 18 de octubre</i>
PF9	Flip Flops	F	<i>Lunes 27 de octubre</i>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

Diagramas de Estado y Construcción de la Tabla de Estado Siguiente

- 1. Representación de secuencias mediante
Diagramas de Transición de Estados***
- 2. Asignación binaria de estados***
- 3. Codificación en ABEL-HDL usando :***
 - a) State_Diagram***
 - b) Truth_Table.***
- 4. Sistema Secuencial Asíncrono***

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.

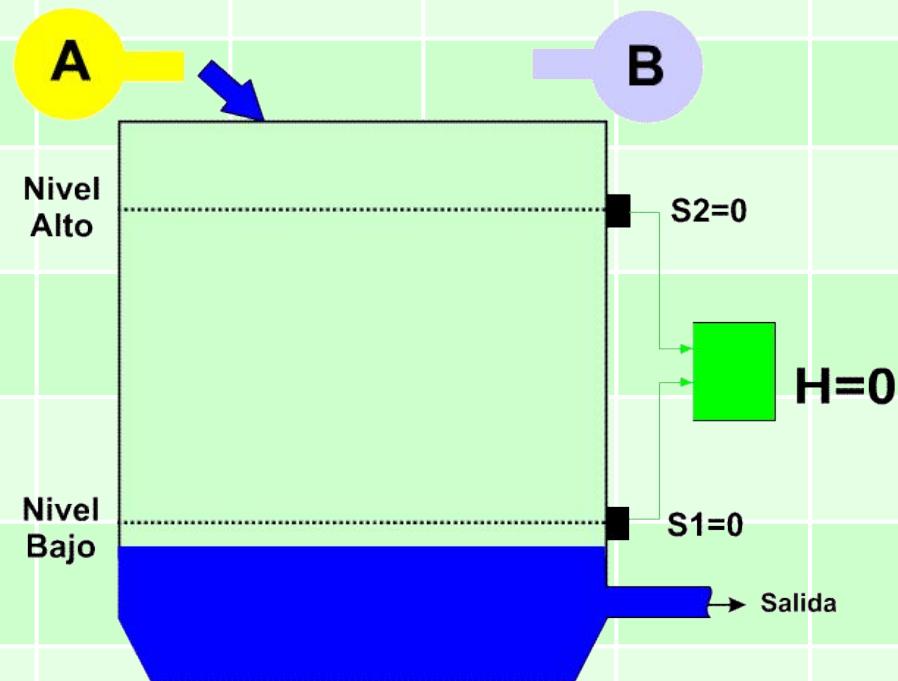


Método para el Diseño de Sistemas Secuenciales síncronos con el uso de HDL y su implementación en un PLD

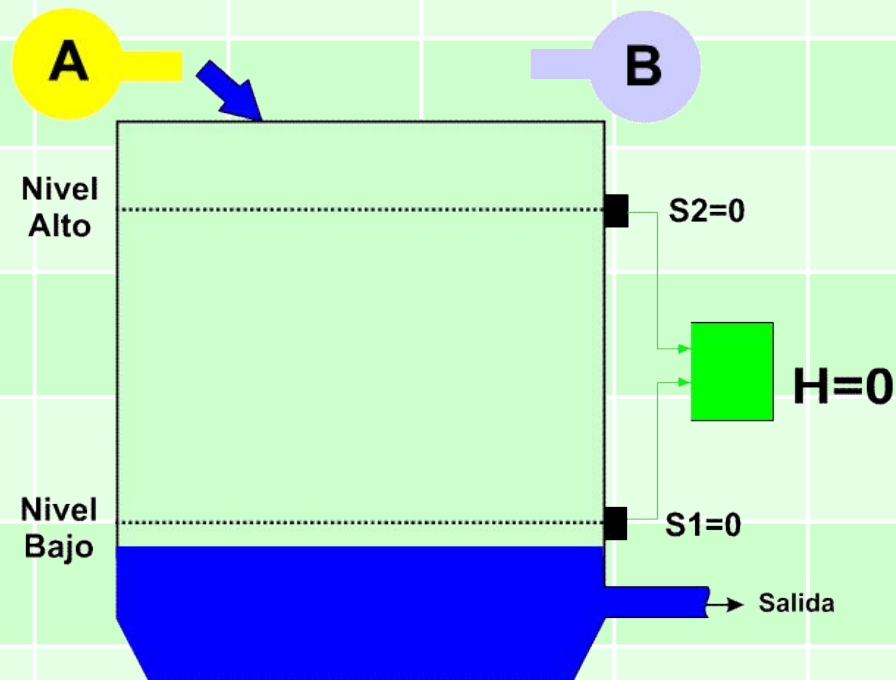
- 1.- Especificar el sistema (Diagrama de transición)
- 2.- Determinar la cantidad de Flip Flops
- 3.- Asignar valores a los estados
- 4.- Diagrama de Bloque (entradas y salidas)
- 5.- Construir la tabla de estado siguiente
- 6.- Codificación en ABEL-HDL
 - a) Entradas y Salidas
 - b) Sincronización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

Base para describir
un sistema
secuencial

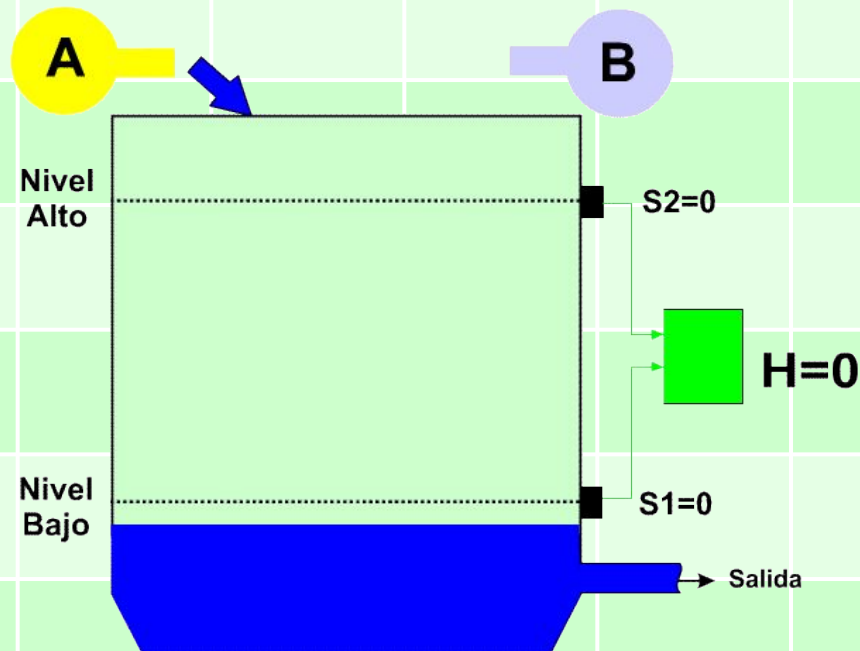
Sistema Alternativo de Llenado de Tanque con Dos Bombas



Diseñar un sistema secuencial síncrono que controle el llenado de un tanque utilizando dos bombas alternadas con las siguientes características:



- a) El sistema consta de dos bombas llamadas "A" y "B"
- b) Un sensor de nivel "H", $H=1$ Tanque lleno y $H=0$ Tanque vacío.
- c) Partiendo de que el Tanque se encuentra vacío ($H=0$), el llenado deberá iniciarse encendiendo la bomba "A" hasta llenar el Tanque ($H=1$), y entonces se apaga.



- d) Si de nuevo se vacía el Tanque ($H=0$), el llenado deberá hacerse encendiendo la bomba "B", hasta llenar el Tanque ($H=1$), y entonces se apaga.
- e) Si se vacía de nuevo el Tanque, el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que las bombas se alternen en su funcionamiento.

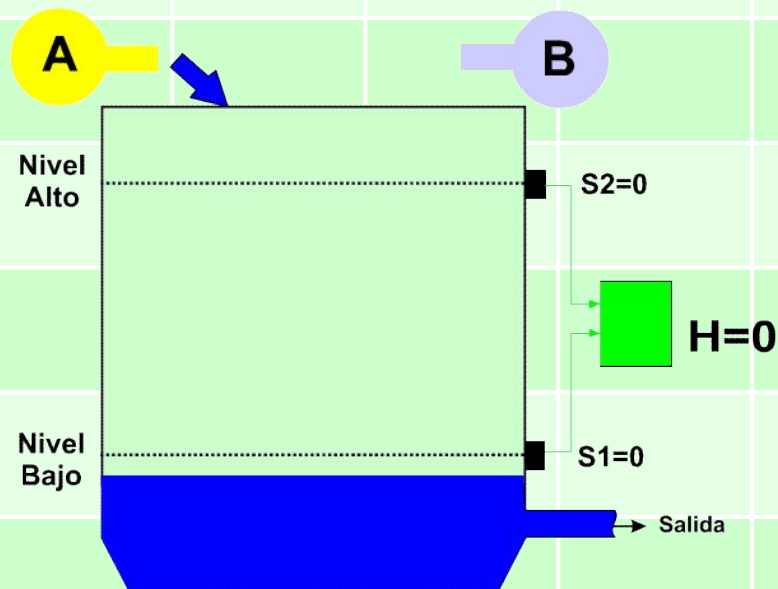


Diagrama de Transición de estados



Diagrama de Transición de estados



Diagrama de Transición de estados

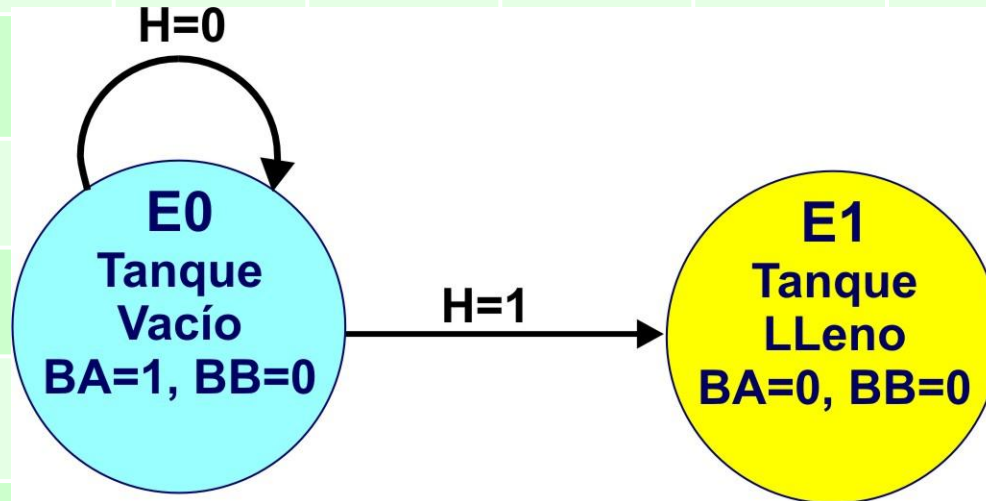


Diagrama de Transición de estados

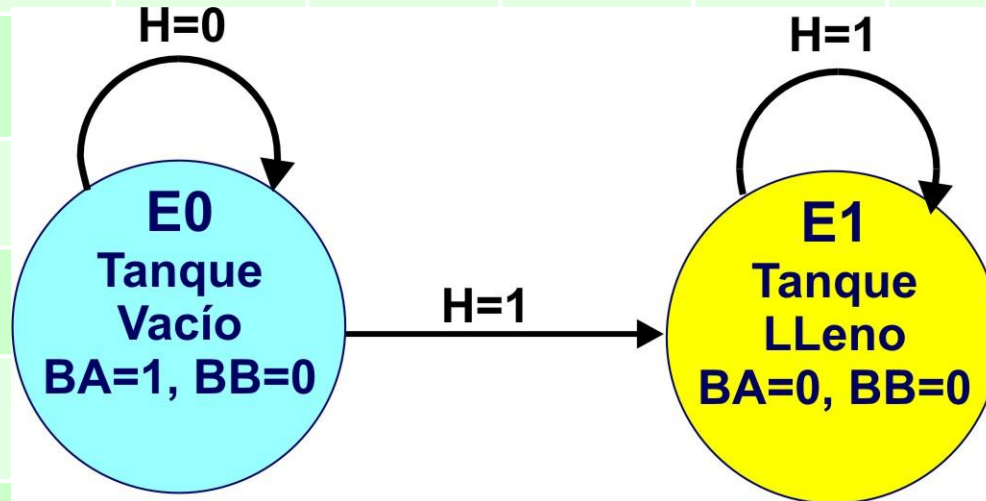


Diagrama de Transición de estados

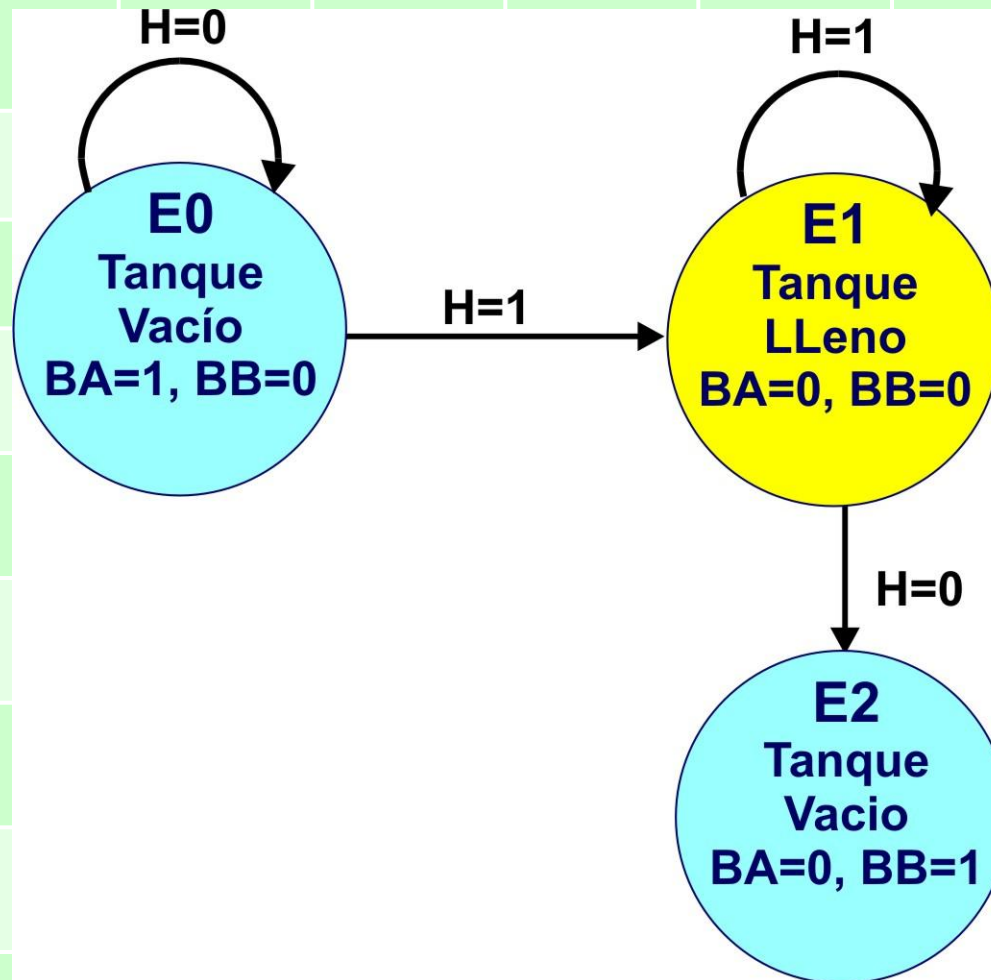


Diagrama de Transición de estados

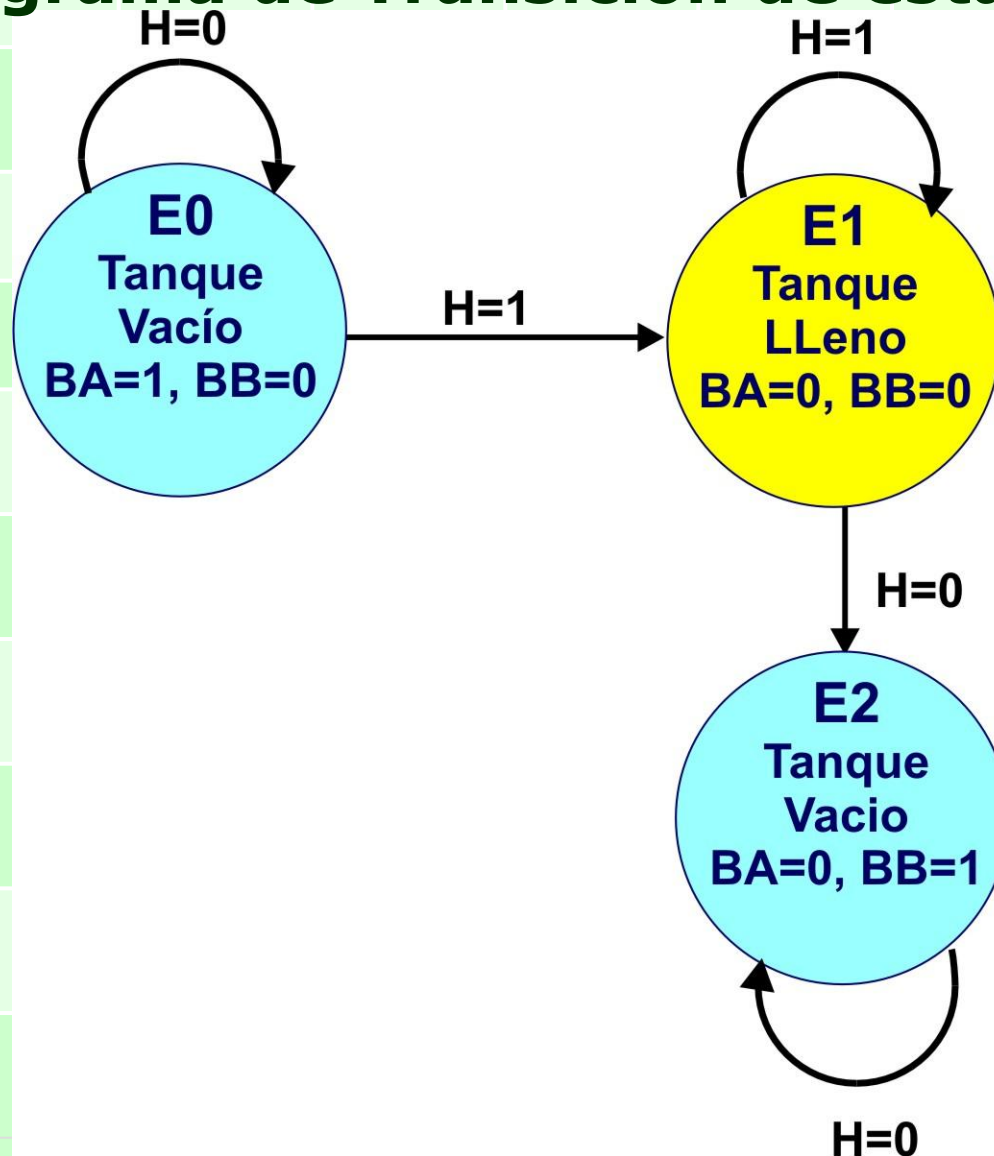
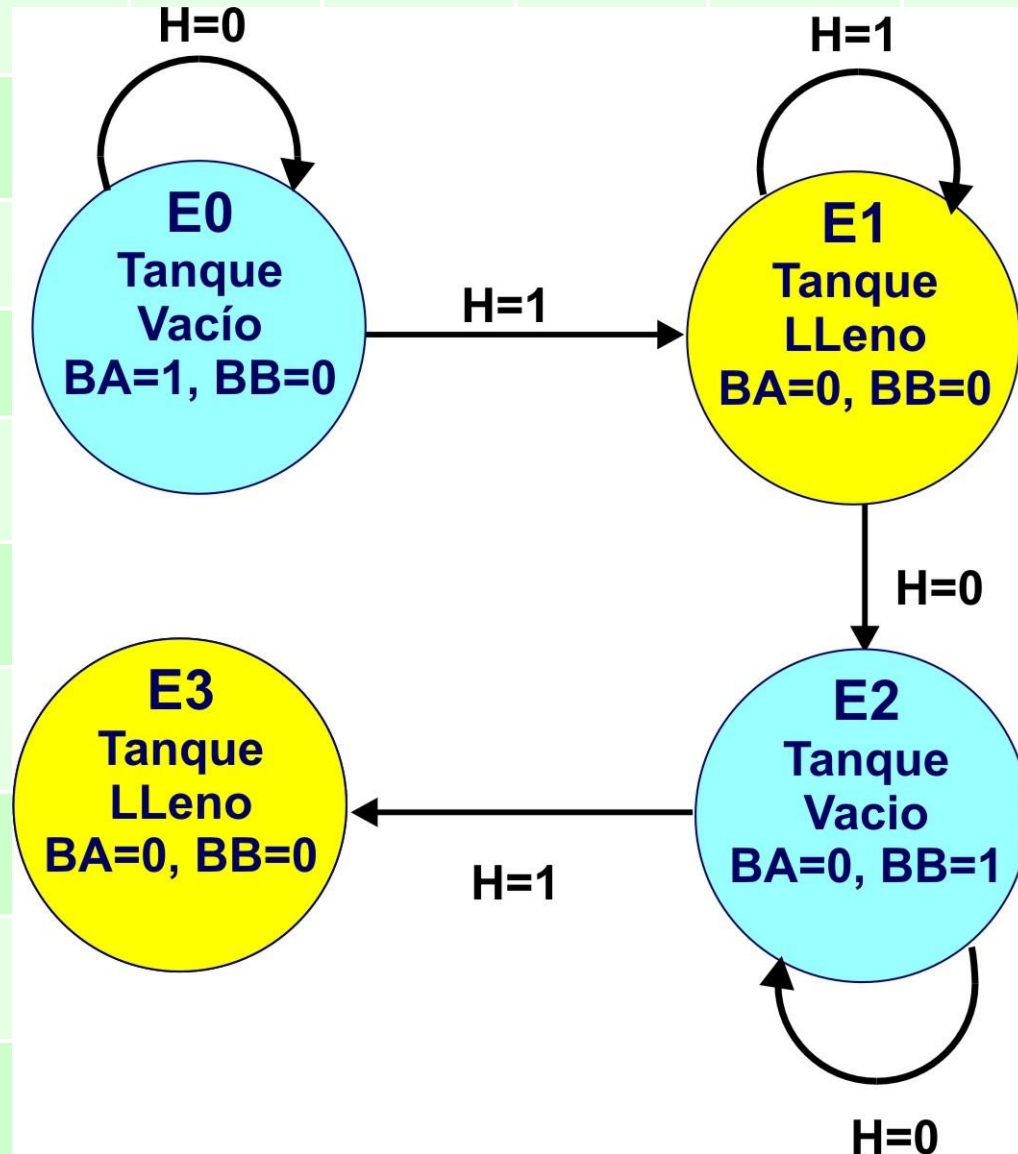


Diagrama de Transición de estados



JAGG

Diagrama de Transición de estados

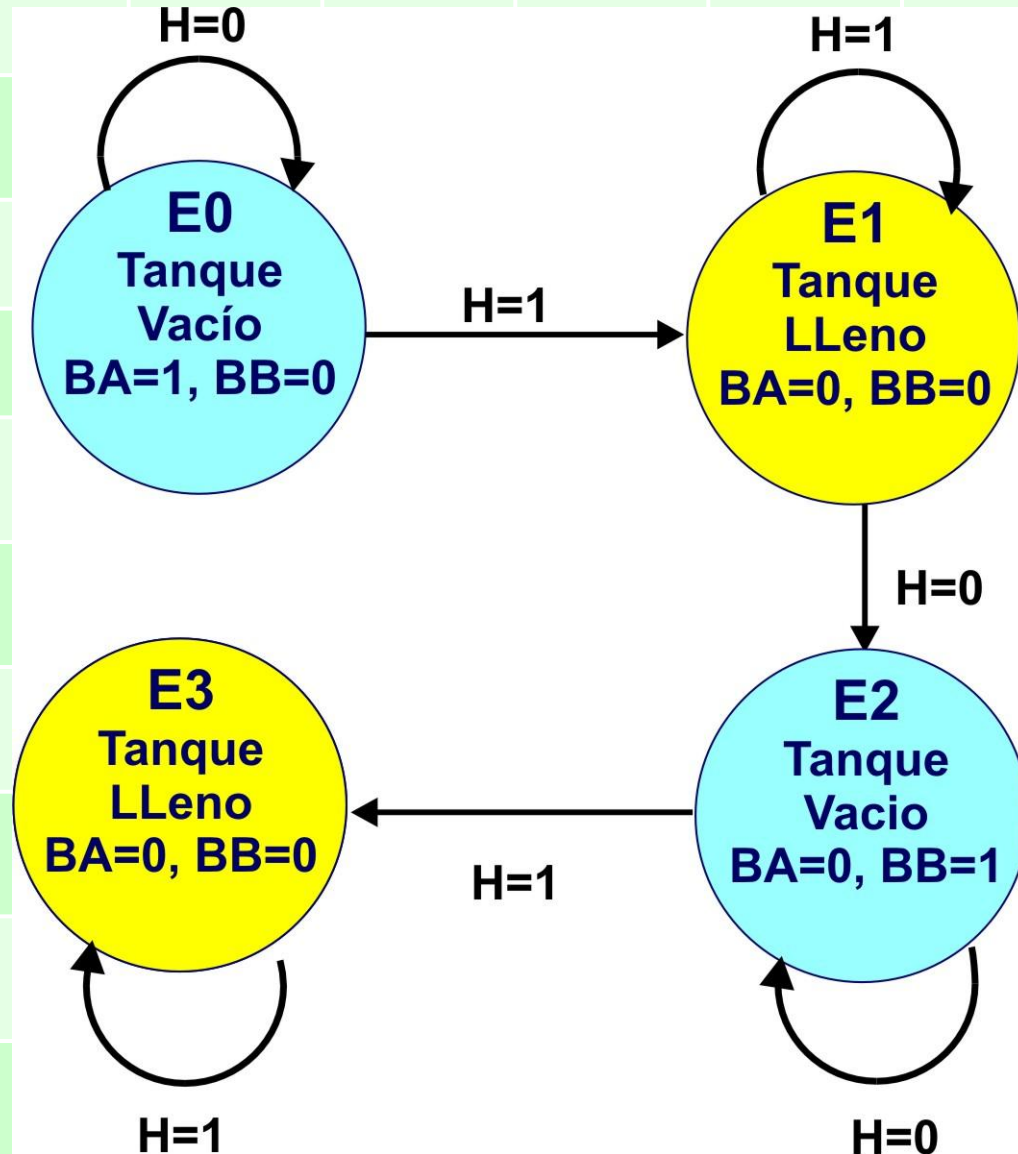


Diagrama de Transición de estados

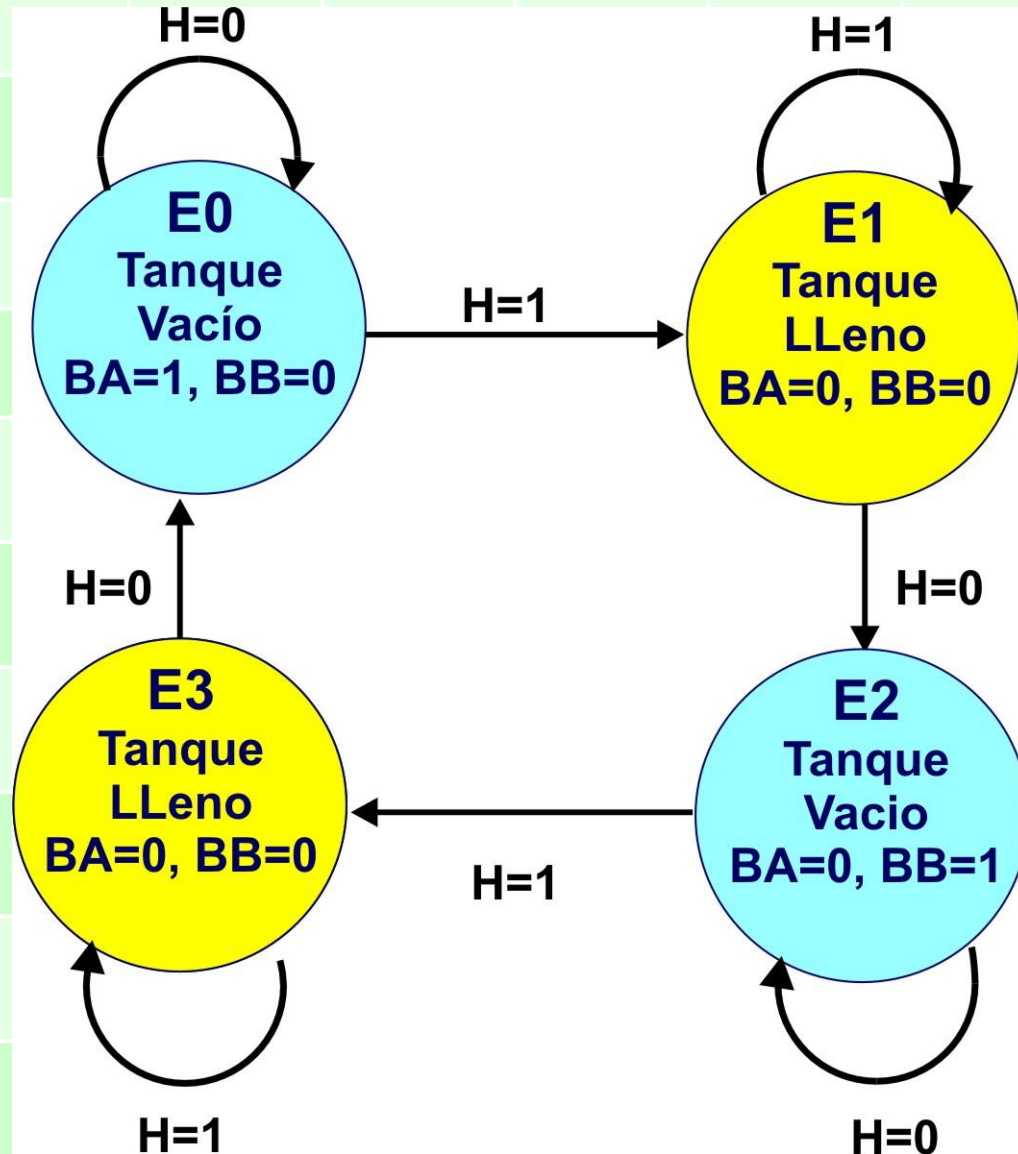
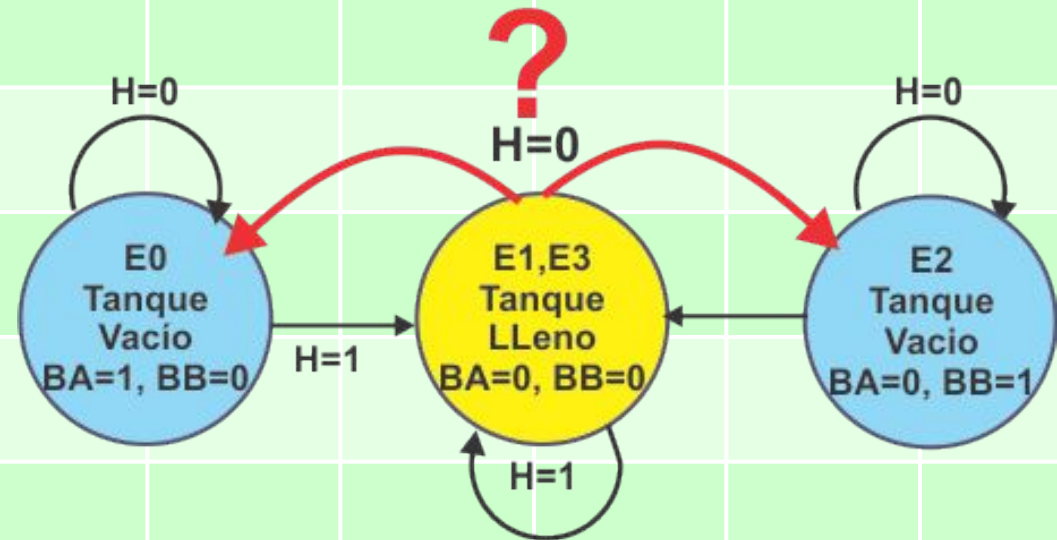
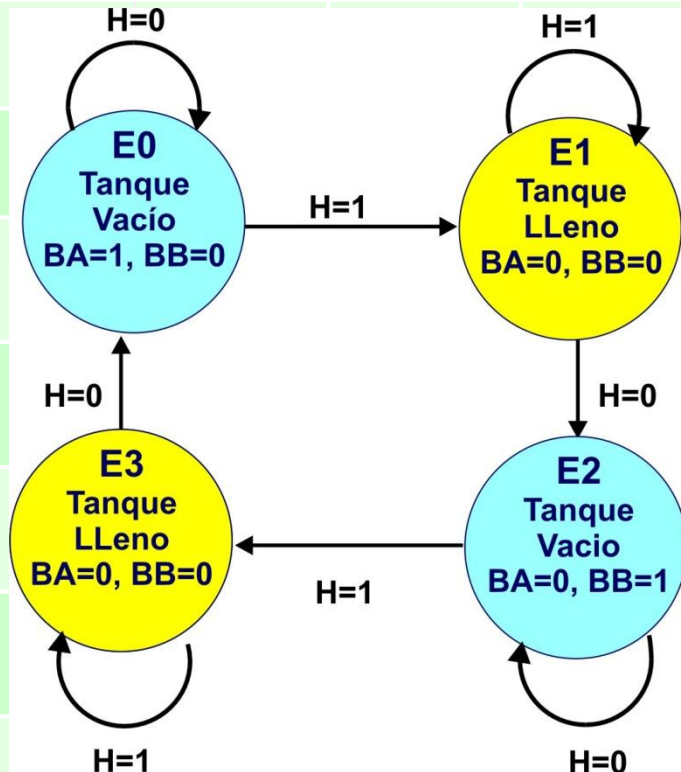
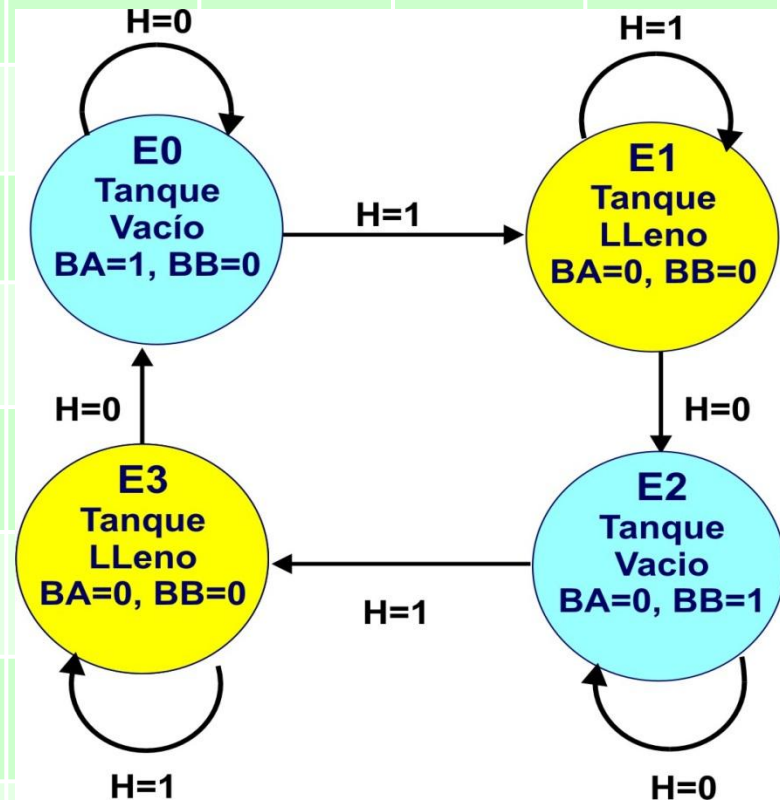


Diagrama de transición de estados



Determinar la cantidad de Flip Flops.

Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3



Para 4 estados E0, E1, E2 y E3 se requieren mínimo 2 Flip_Flops

Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
E0	0	0
E1	0	1
E2	1	0
E3	1	1

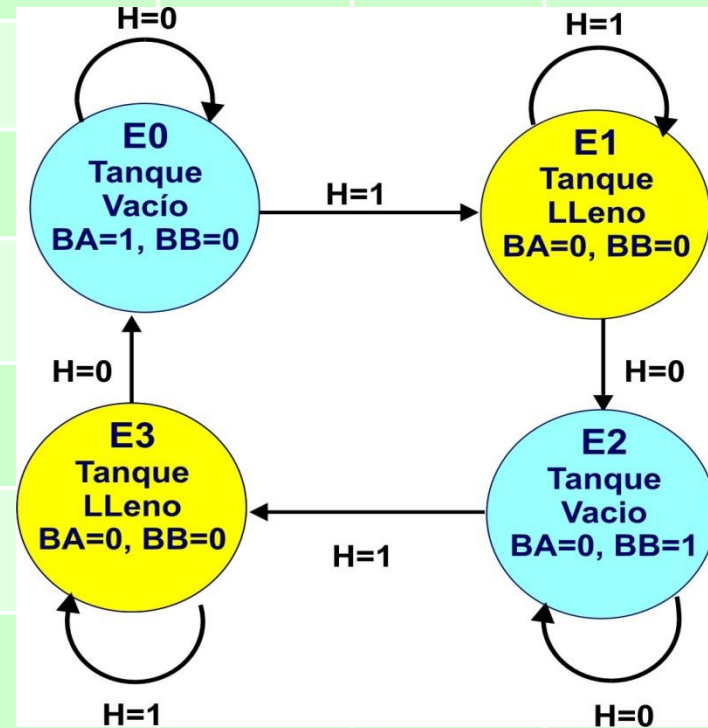


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0		
E1		
E2		
E3		

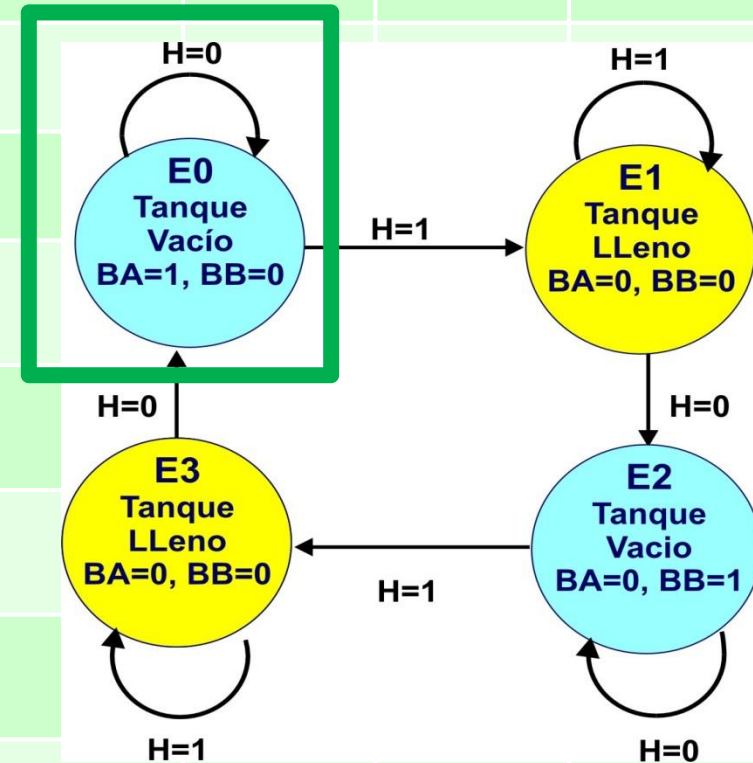
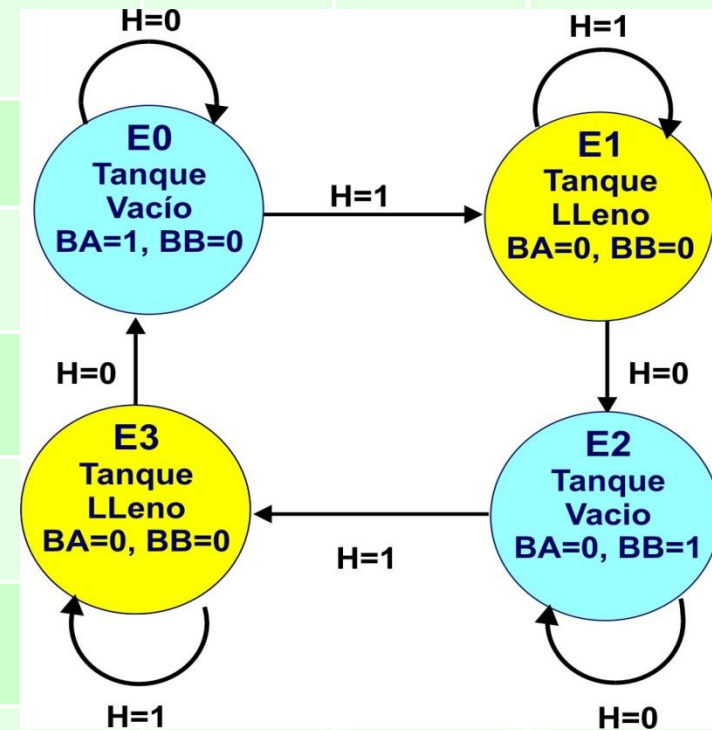


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	
E1		
E2		
E3		



JAGG

Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	
E1		
E2		
E3		

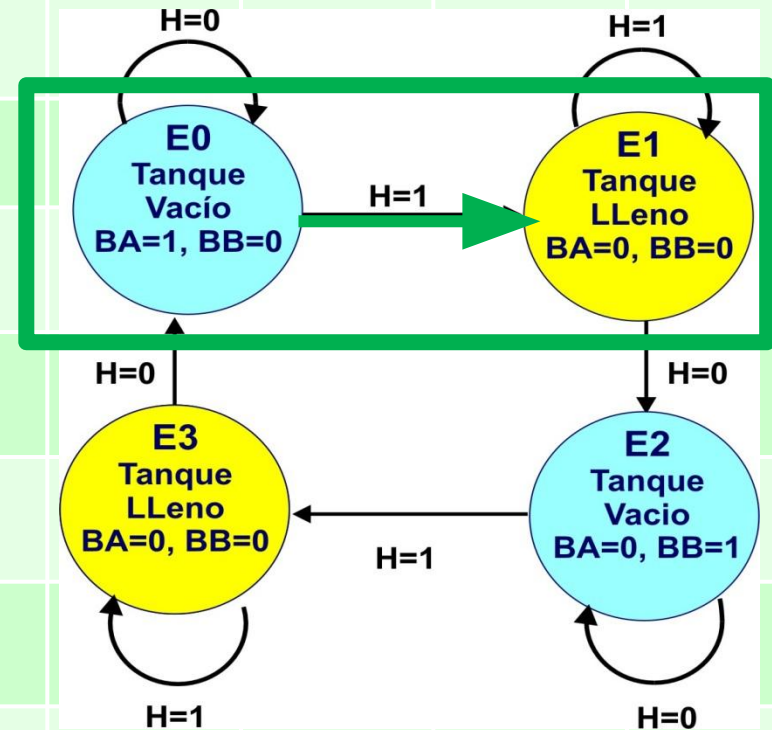


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1		
E2		
E3		

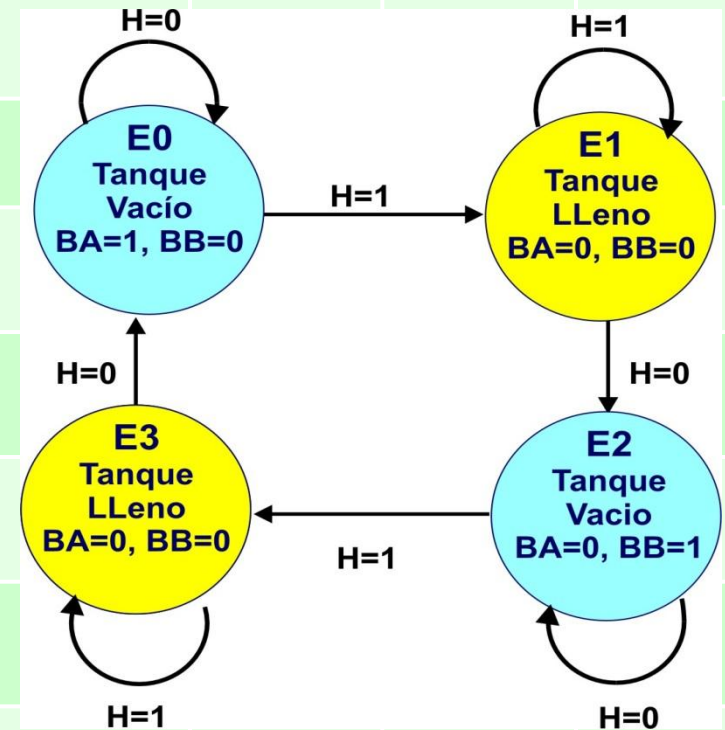


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1		
E2		
E3		

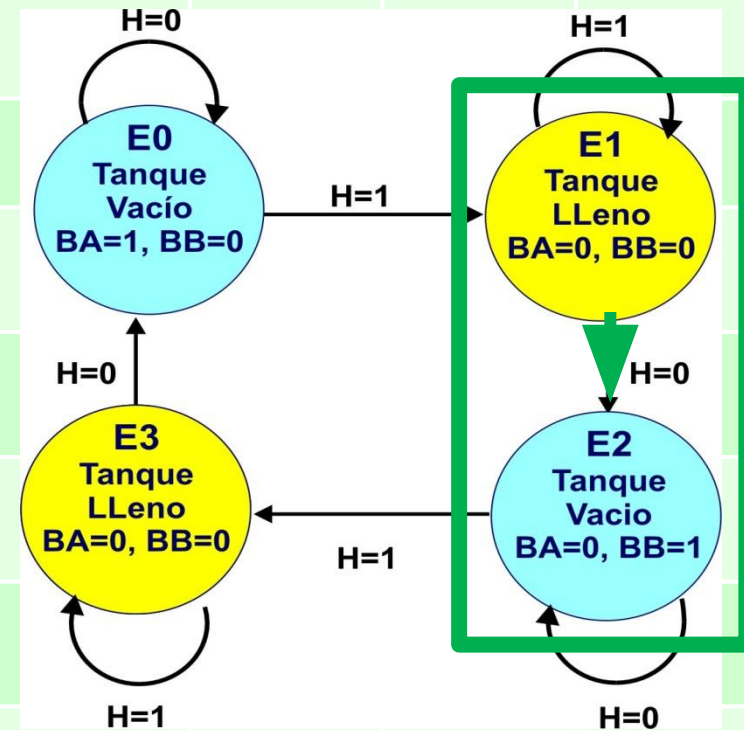


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	
E2		
E3		

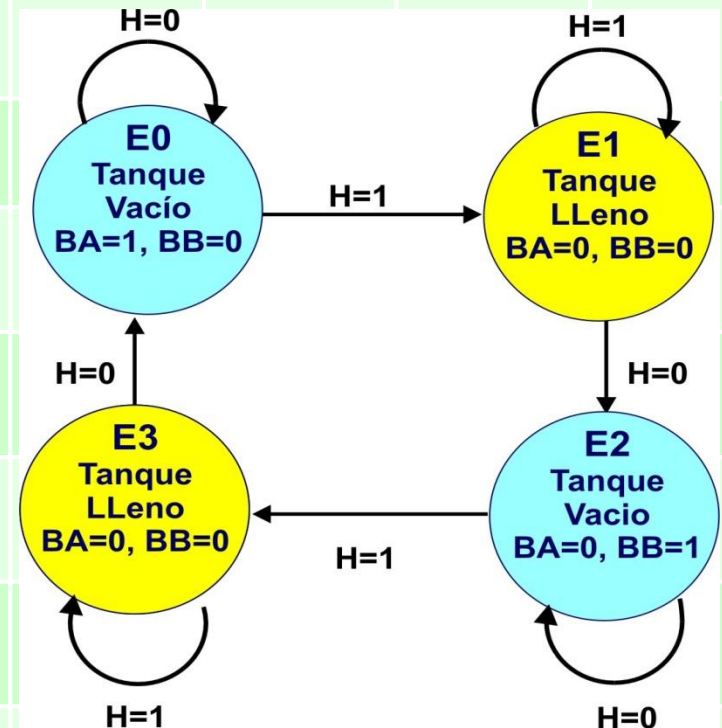


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	
E2		
E3		

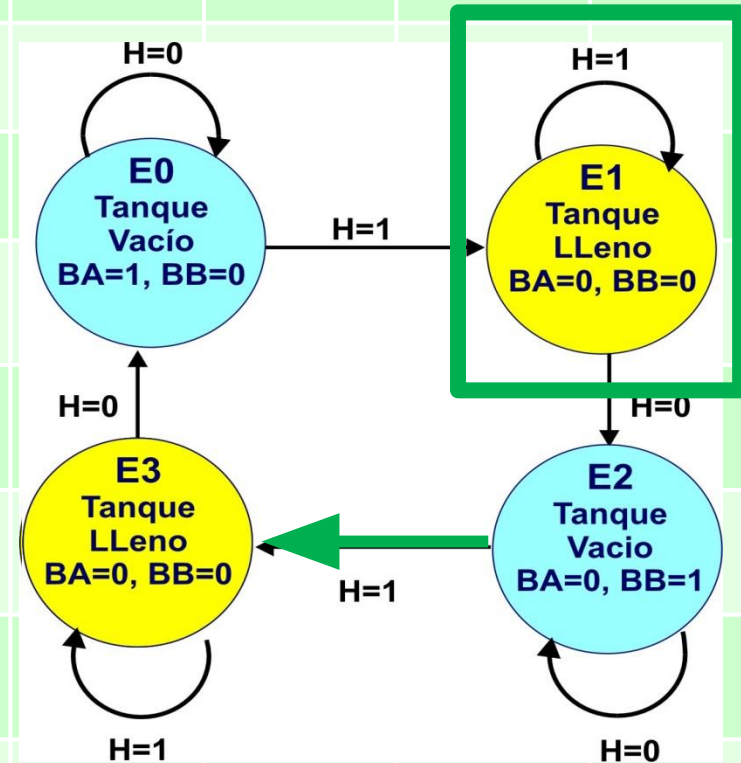


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2		
E3		

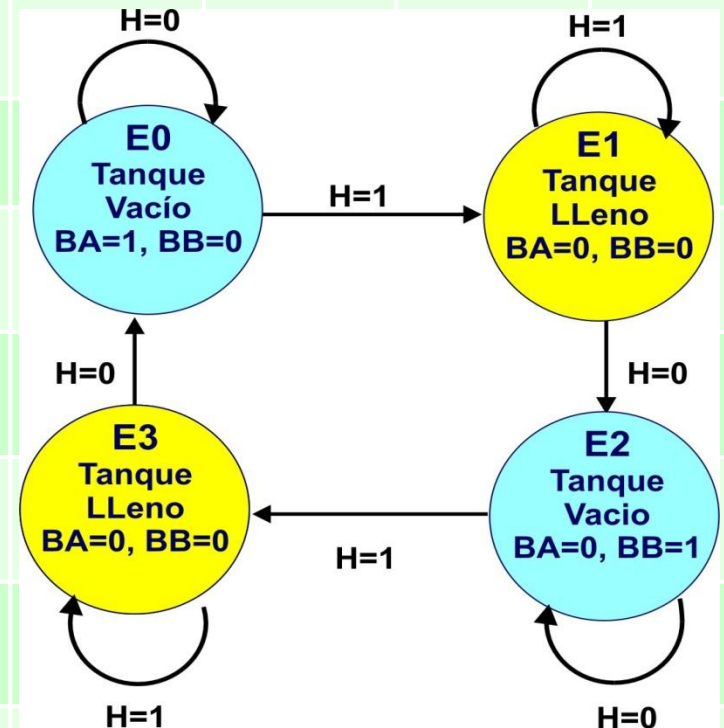


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2		
E3		

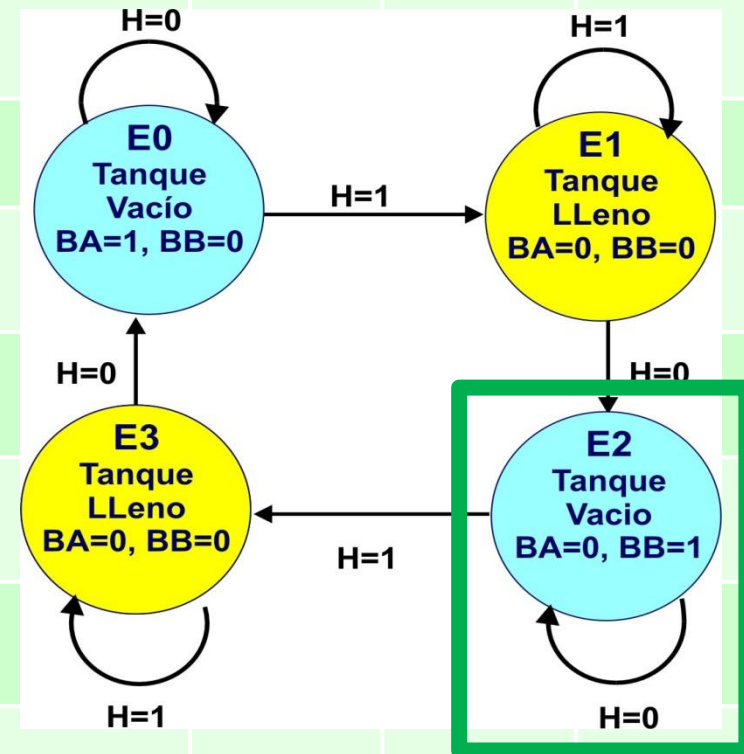


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	
E3		

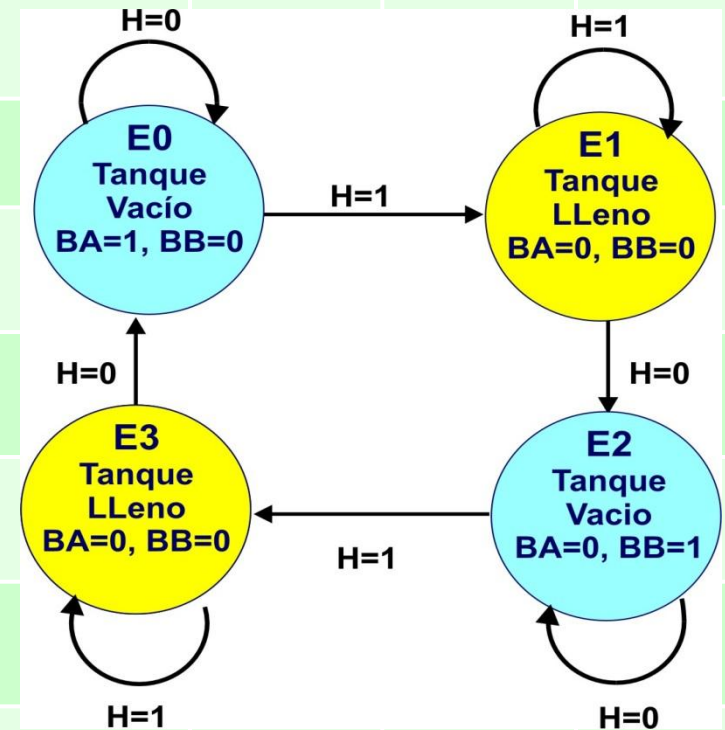


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	
E3		

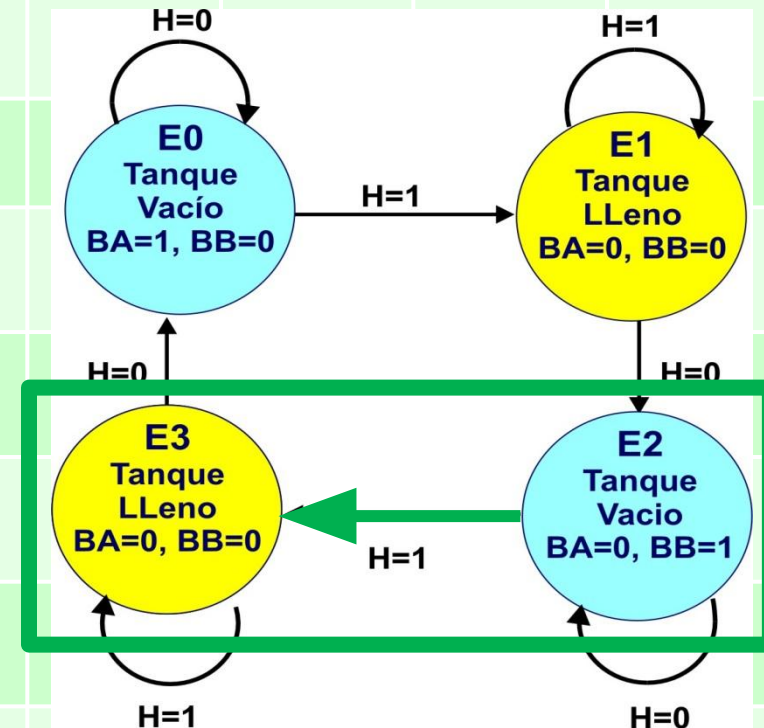


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3		

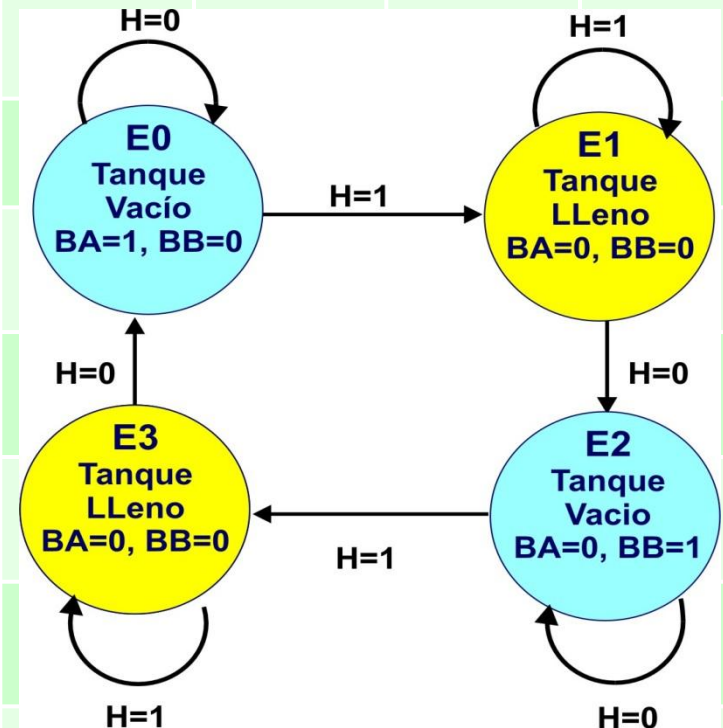


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3		

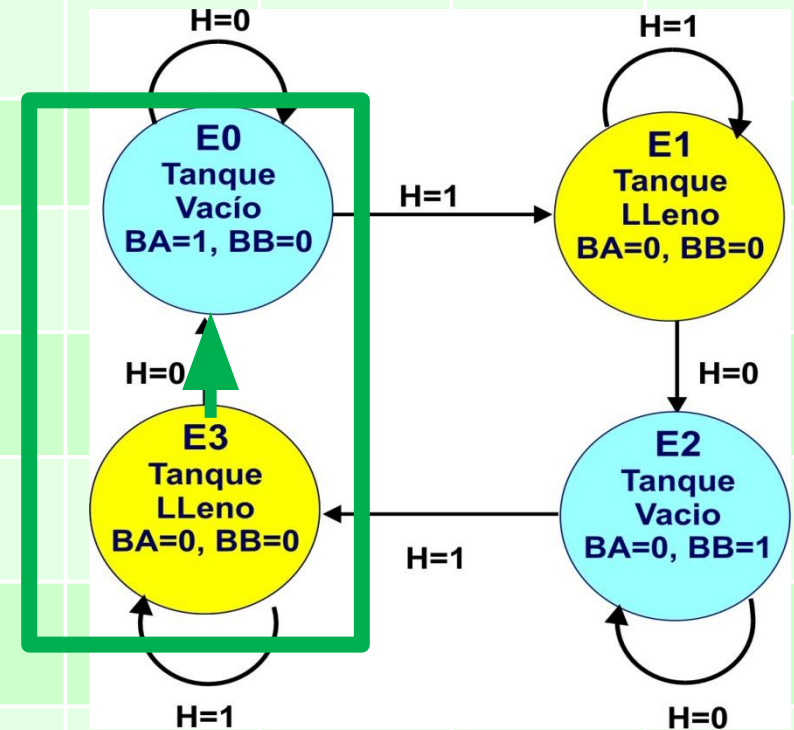


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3	E0	

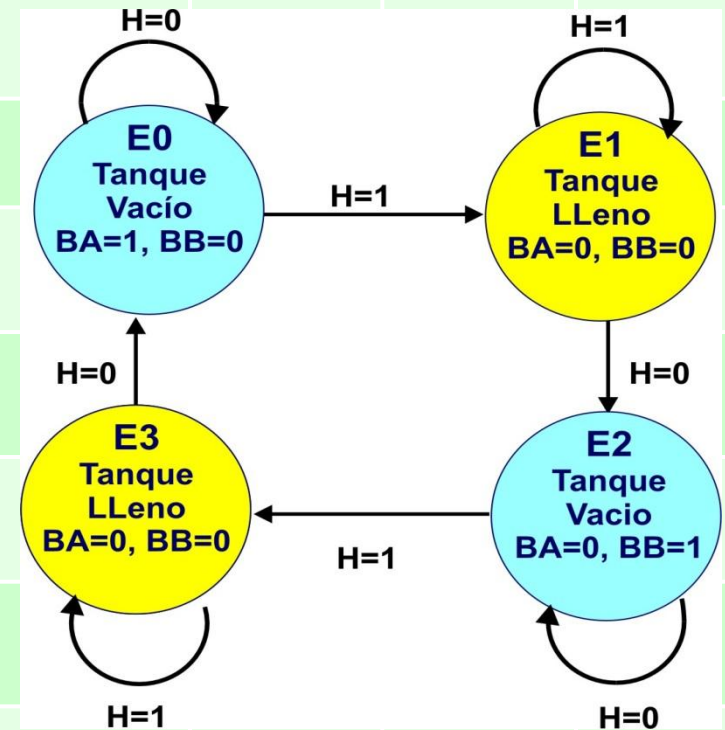


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3	E0	

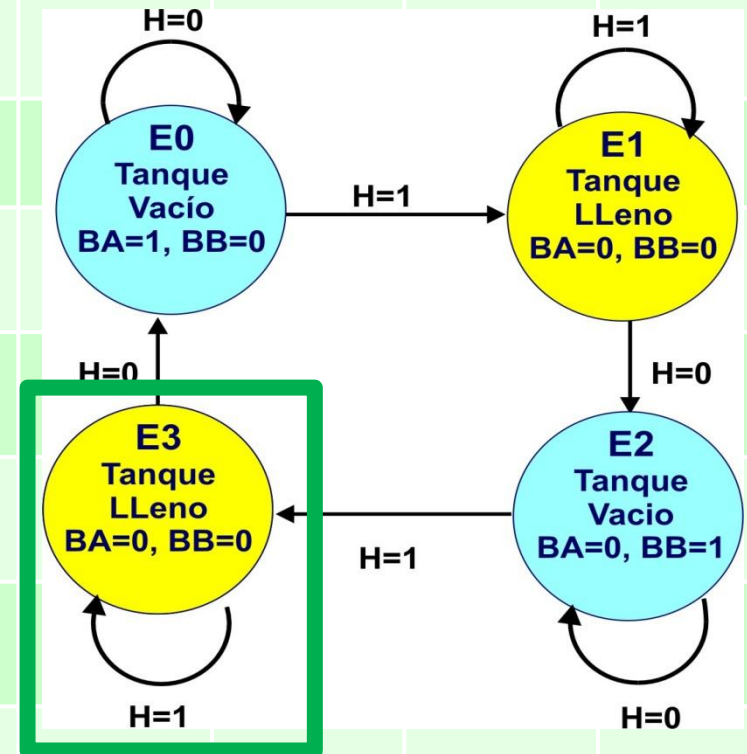


Tabla de estado siguiente

Estado Presente	Estado siguiente	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3	E0	E3

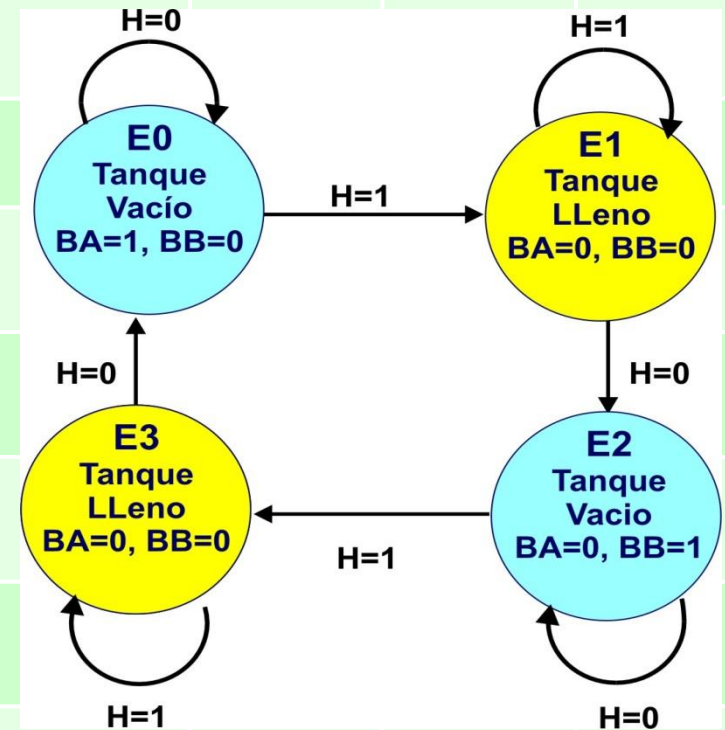


Tabla de estado siguiente

Estado Presente	Estado siguiente		Salidas	
	H=0	H=1	BA	BB
E0	E0	E1	1	0
E1	E2	E1	0	0
E2	E2	E3	0	1
E3	E0	E3	0	0

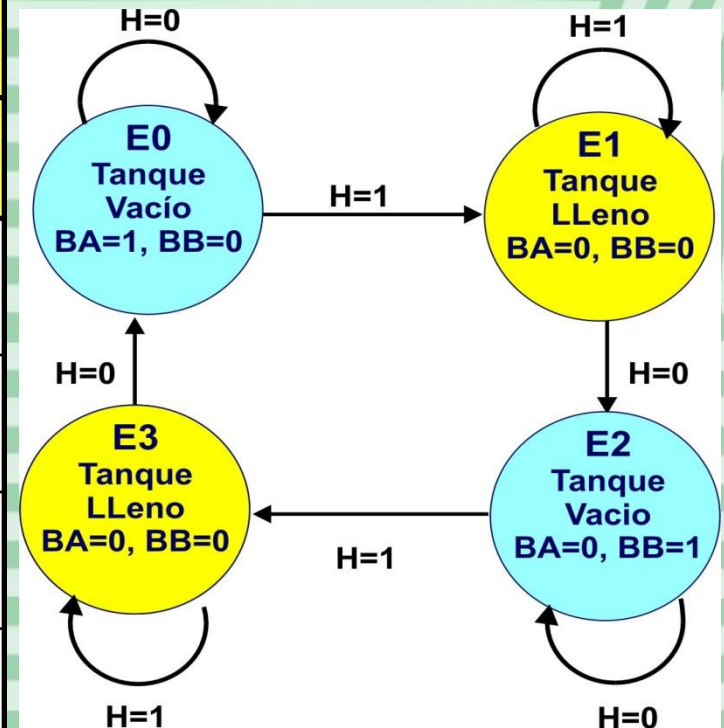
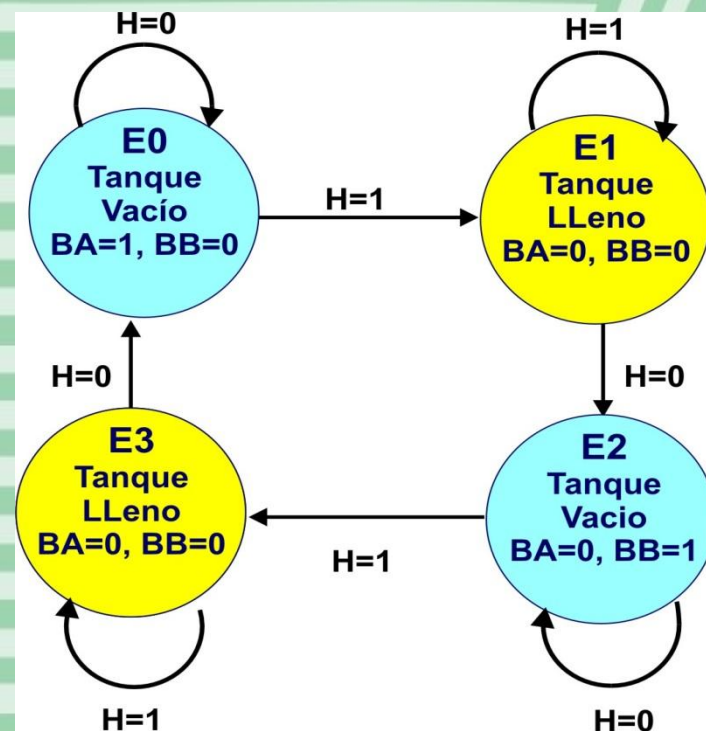
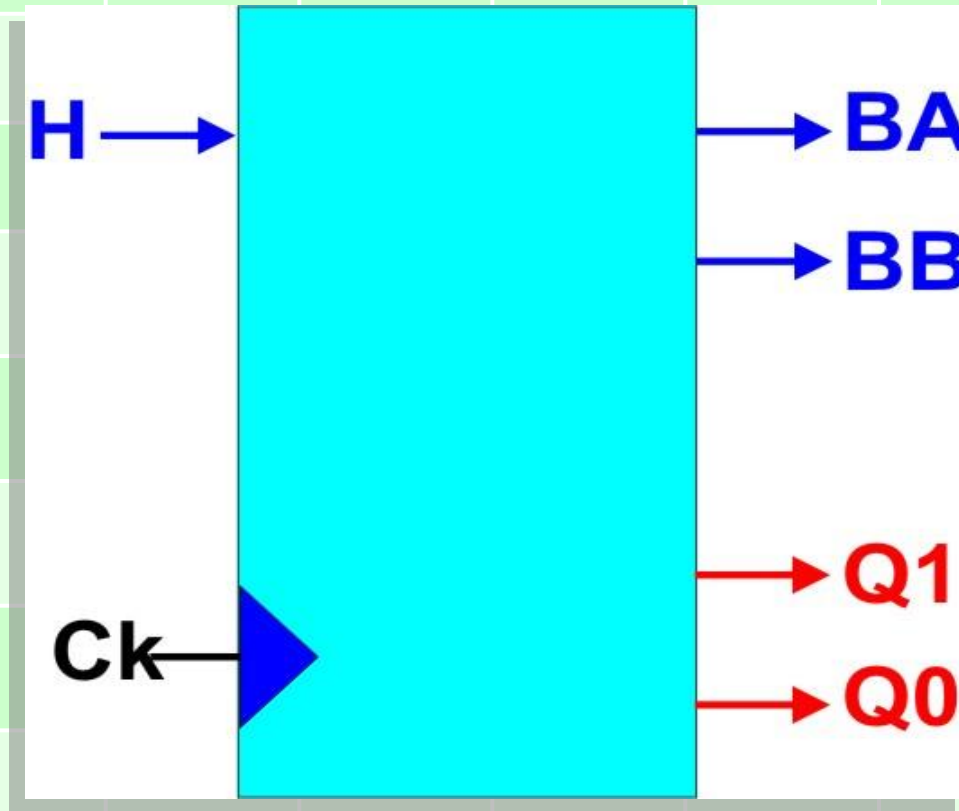


Tabla de estado siguiente

Estado Presente	Estado siguiente		Salidas		FFs	
	H=0	H=1	BA	BB	Q1	Q0
E0	E0	E1	1	0	0	0
E1	E2	E1	0	0	0	1
E2	E2	E3	0	1	1	0
E3	E0	E3	0	0	1	1



Determinar las entradas y salidas



Pasos en la programación en ABEL-HDL

- 1.- Entradas y salidas**
- 2.- Sincronización**
- 3.- asignar valores**
- 4.- State_diagram o Truth_table**

Código en ABEL-HDL

Entradas, salidas y asignación de terminales

MODULE DBOMBAS

"Entradas

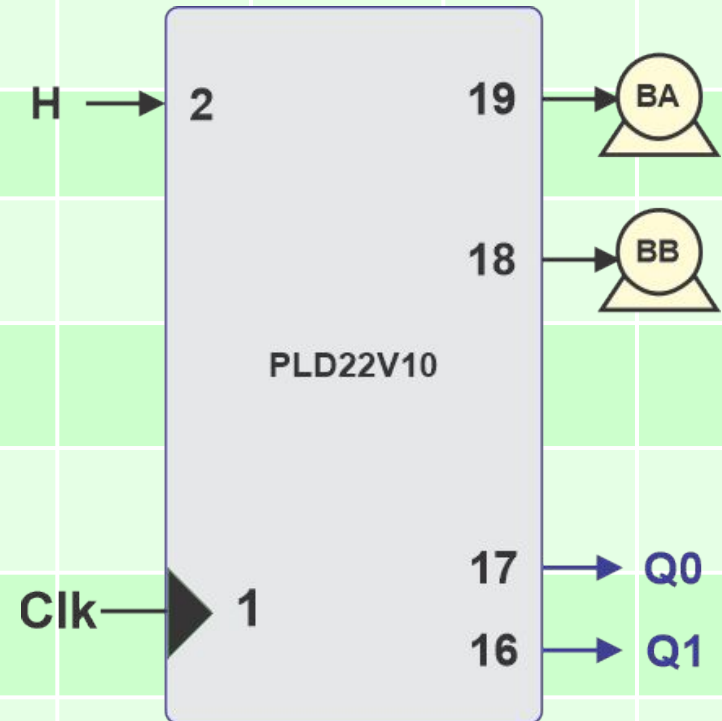
Clk, H Pin 1,2;

" Salidas Combinacionales

BA,BB Pin 19,18 istype'com';

" Salidas Registradas

Q1,Q0 pin 17,16 istype'reg';



JAGG

Archivo en ABEL-HDL

Sincronización

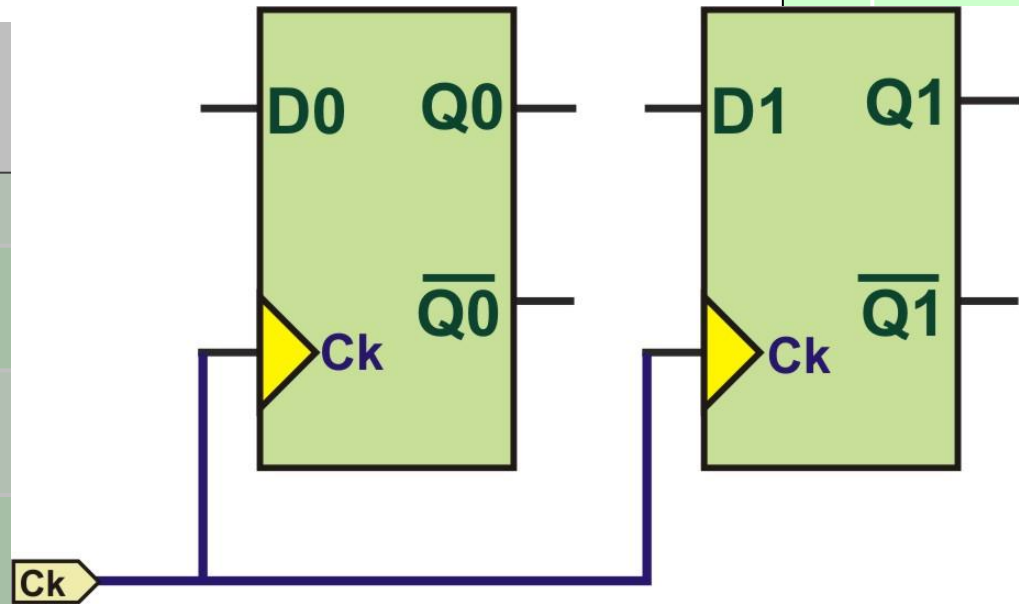
" Conectar el Clk a los dos Flip Flops sincronizar

DECLARATIONS

Sreg=[Q0,Q1];

EQUATIONS

Sreg.clk=CLK;



JAGG

Archivo en ABEL-HDL

Asignar Valores a los estados

" Asignar Valores a los estados

DECLARATIONS

E0=[0, 0];

E1=[0, 1];

E2=[1, 1];

E3=[1, 0];

Estados	Asignación de valores a los estados	
	Q1	Q0
E0	0	0
E1	0	1
E2	1	1
E3	1	0

Archivo en ABEL-HDL

Definir la secuencia

state_diagram Sreg;

State E3:

BA=0; BB=0;

if H==0 then E0;

if H==1 then E3;

State E0:

BA=1; BB=0;

if H then E1 else E0;

State E2:

BA=0; BB=1;

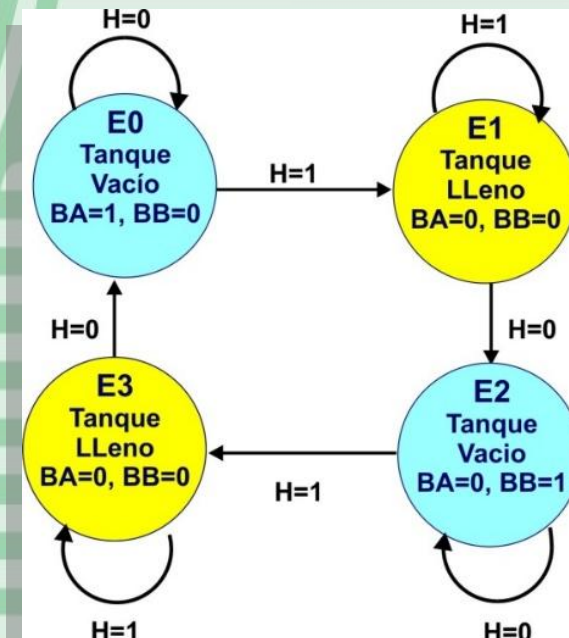
if !H then E2 else E3;

State E1:

BA=0; BB=0;

if H then E1;

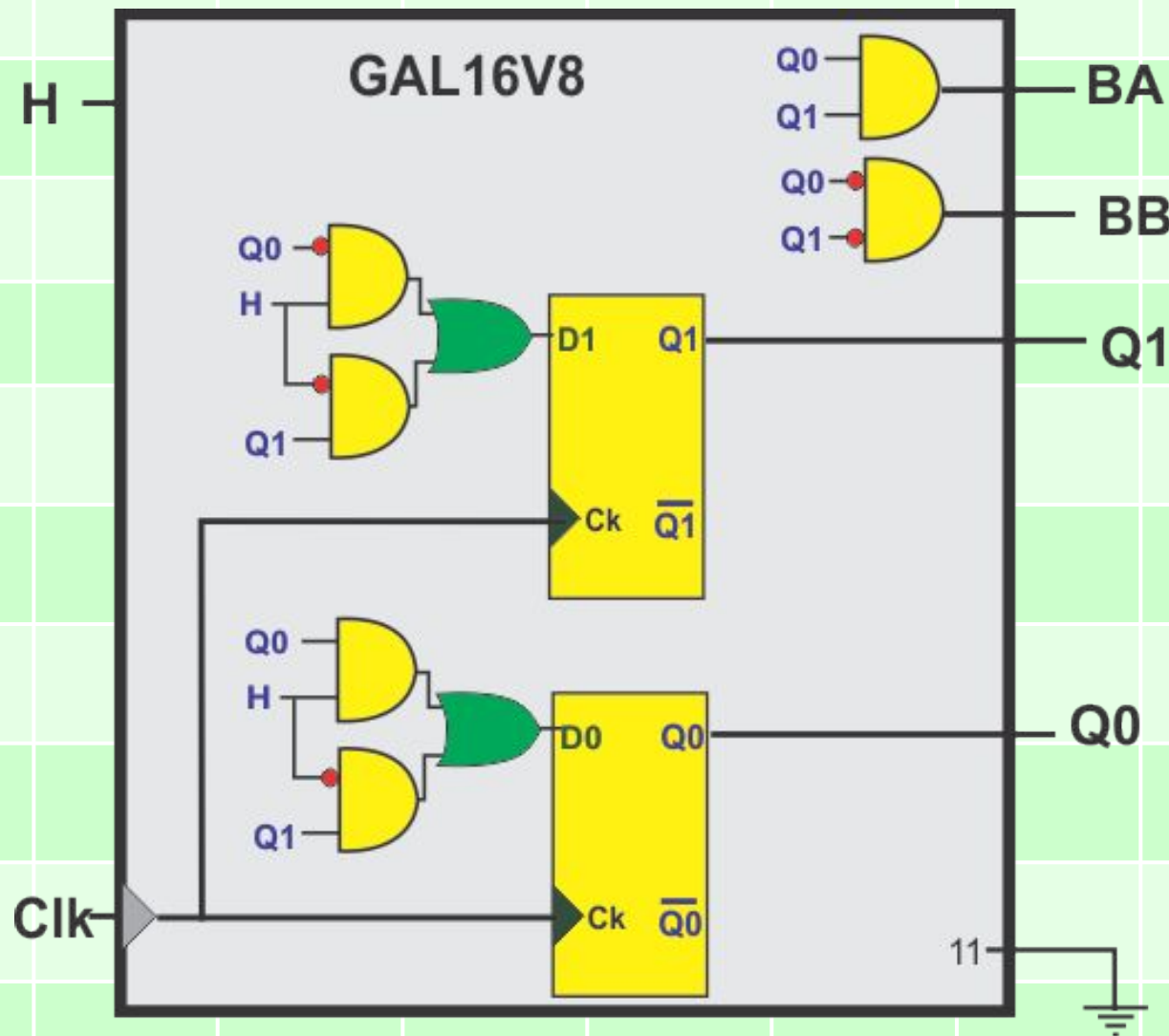
if !H then E2;



Estado Presente	Estado Próximo	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3	E0	E3

JAGG

Estructura interna en el PLD



Programación con Truth_table en modo secuencial :->

Truth_Table

([Entrada, Estado presente] :-> [Estado siguiente])

MODULE dbn

Clk,H pin 1,2;

BA,BB pin 19,18 istance 'com';

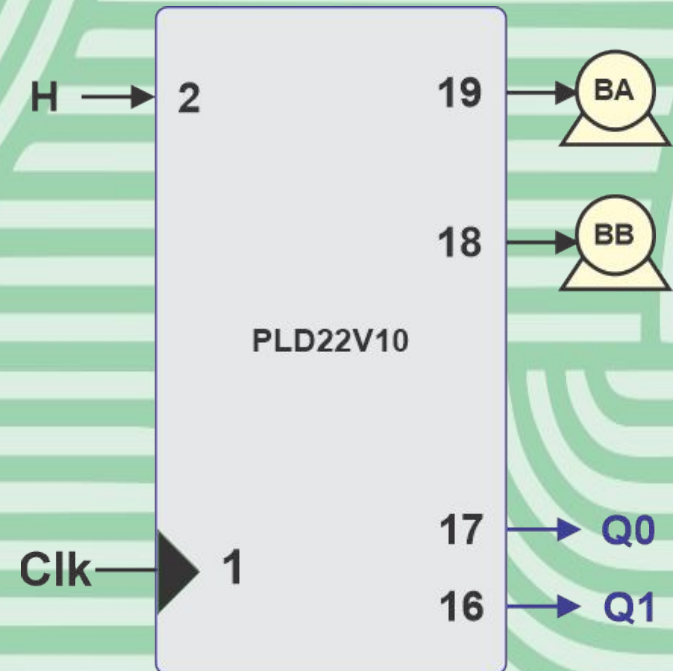
Q1,Q0 PIN 17,16 ISTANCE 'REG';

“Sincronización

UANL=[Q1,Q0];

equations

UANL.Clk=Clk;



Truth_table

$([H, Q1, Q0] \rightarrow [Q1, Q0])$

$[0, 0, 0] \rightarrow [0, 0];$

$[1, 0, 0] \rightarrow [0, 1];$

$[0, 0, 1] \rightarrow [1, 0];$

$[1, 0, 1] \rightarrow [0, 1];$

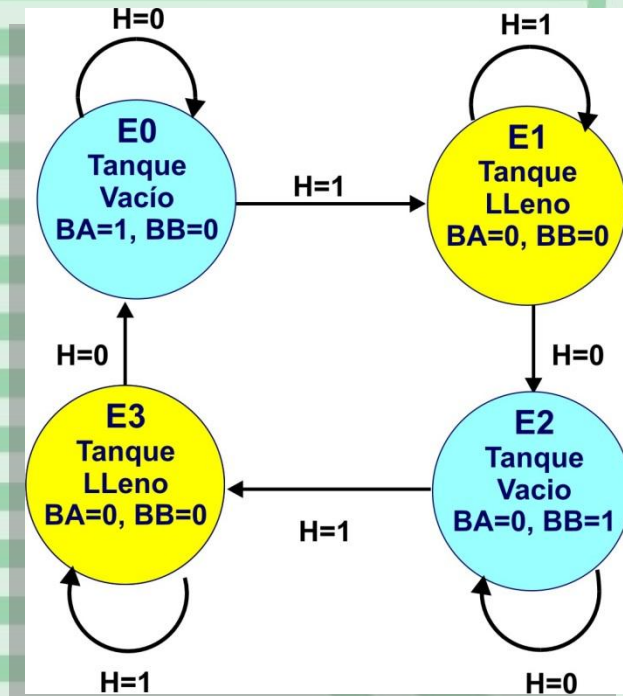
$[1, 1, 0] \rightarrow [1, 1];$

$[0, 1, 0] \rightarrow [1, 0];$

$[0, 1, 1] \rightarrow [0, 0];$

$[1, 1, 1] \rightarrow [1, 1];$

Estados	Q1	Q0
E0	0	0
E1	0	1
E2	1	0
E3	1	1



JAGG

Truth_table

([**H**, Q1,Q0]:>[Q1,Q0])

[**0**, 0,0]:>[**0**,**0**];

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

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

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

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

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

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

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

Estado	H	Q1	Q0	Q1+1	Q0+1	BA	BB
E0	0	0	0	0	0	1	0
E1	0	0	1	1	0	0	0
E2	0	1	0	1	0	0	1
E3	0	1	1	0	0	0	0
E0	1	0	0	0	1	1	0
E1	1	0	1	0	1	0	0
E2	1	1	0	1	1	0	1
E3	1	1	1	1	1	0	0

Truth_table

([Q1,Q0]->[BA,BB])

[0,0]->[1,0];

[0,1]->[0,0];

[1,0]->[0,1];

[1,1]->[0,0];

Truth_table

$([H, Q1, Q0] \Rightarrow [Q1, Q0] \Rightarrow [BA, BB])$

$[0, 0, 0] \Rightarrow [0, 0] \Rightarrow [1, 0];$

$[0, 0, 1] \Rightarrow [1, 0] \Rightarrow [0, 0];$

$[0, 1, 0] \Rightarrow [1, 0] \Rightarrow [0, 1];$

$[0, 1, 1] \Rightarrow [0, 0] \Rightarrow [0, 0];$

$[1, 0, 0] \Rightarrow [0, 1] \Rightarrow [1, 0];$

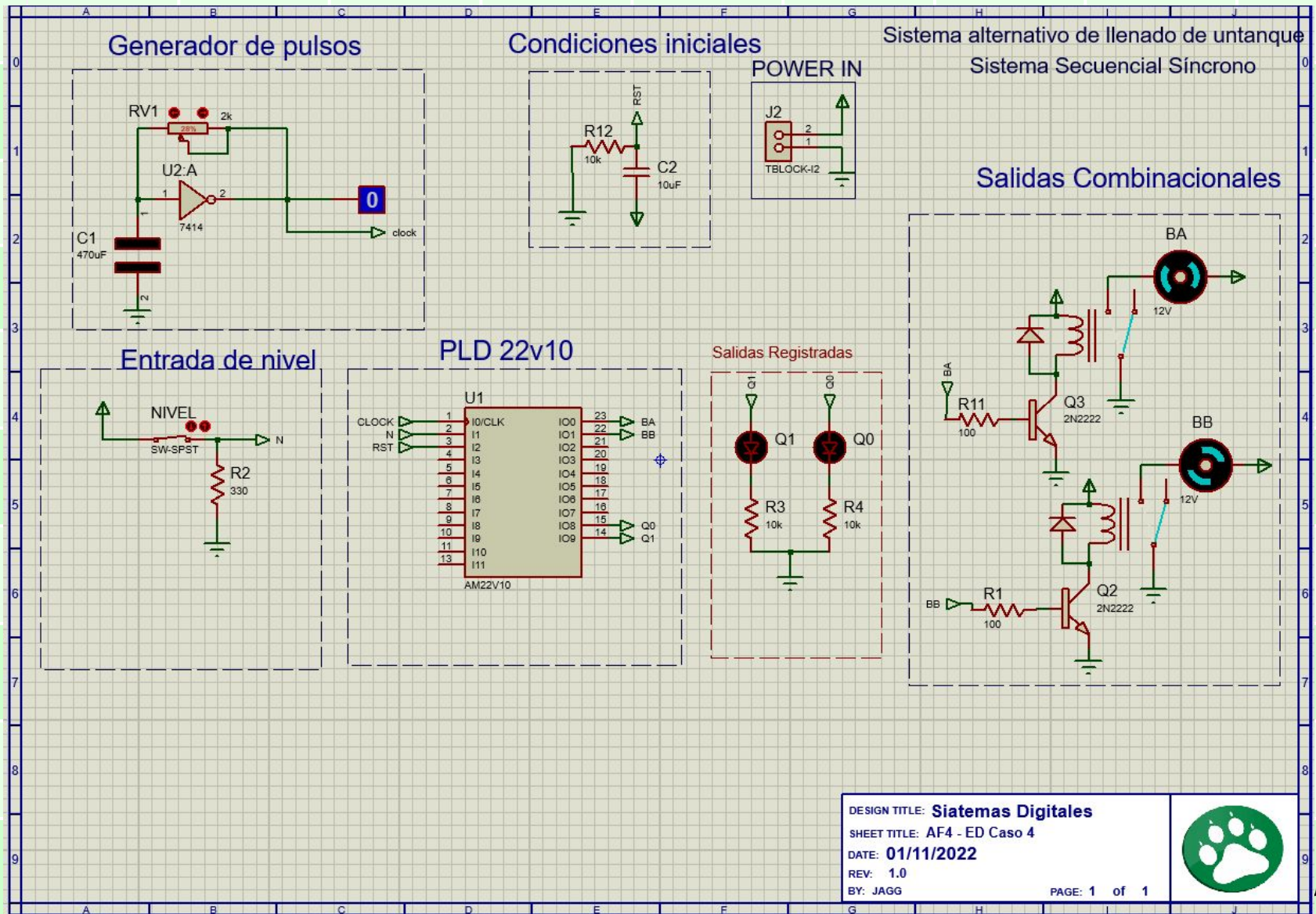
$[1, 0, 1] \Rightarrow [0, 1] \Rightarrow [0, 0];$

$[1, 1, 0] \Rightarrow [1, 1] \Rightarrow [0, 1];$

$[1, 1, 1] \Rightarrow [1, 1] \Rightarrow [0, 0];$

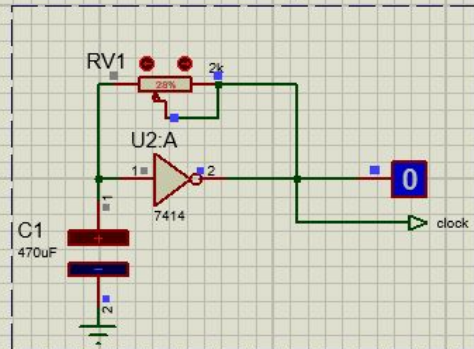
Estado	H	Q1	Q0	Q1+1	Q0+1	BA	BB
E0	0	0	0	0	0	1	0
E1	0	0	1	1	0	0	0
E2	0	1	0	1	0	0	1
E3	0	1	1	0	0	0	0
E0	1	0	0	0	1	1	0
E1	1	0	1	0	1	0	0
E2	1	1	0	1	1	0	1
E3	1	1	1	1	1	0	0

Simulación en Proteus

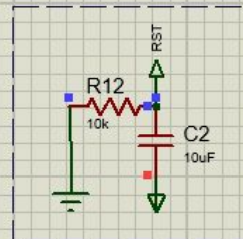


Simulación en Proteus

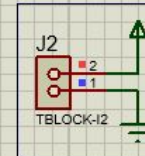
Generador de pulsos



Condiciones iniciales



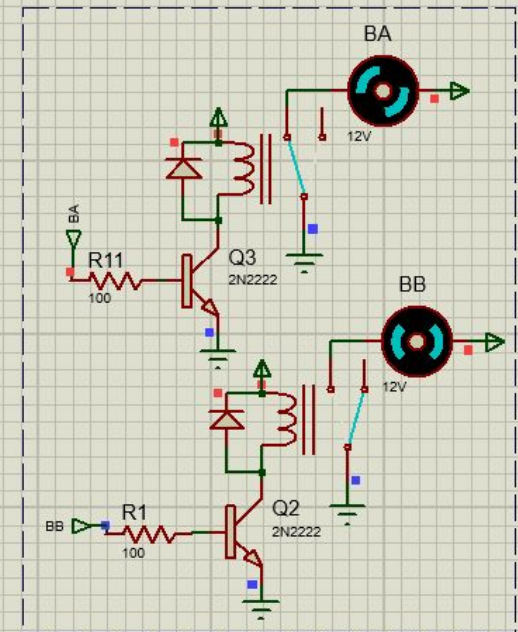
POWER IN



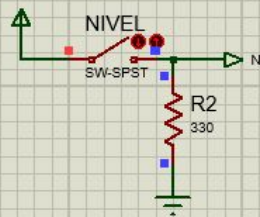
Sistema alternativo de llenado de un tanque

Sistema Secuencial Síncrono

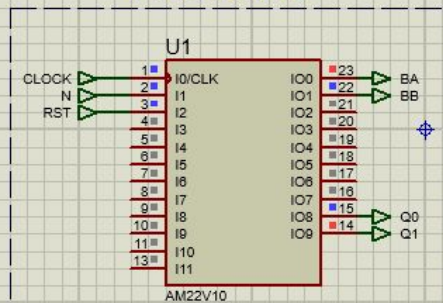
Salidas Combinacionales



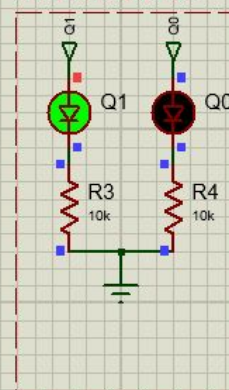
Entrada de nivel



PLD 22v10



Salidas Registradas



DESIGN TITLE: **Sistemas Digitales**

SHEET TITLE: **AF4 - ED Caso 4**

DATE: **01/11/2022**

REV: **1.0**

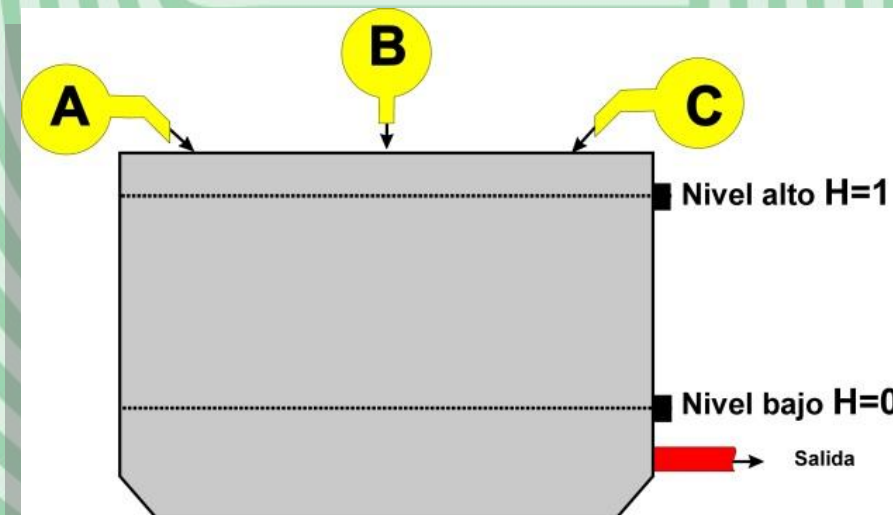
BY: **JAGG**

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

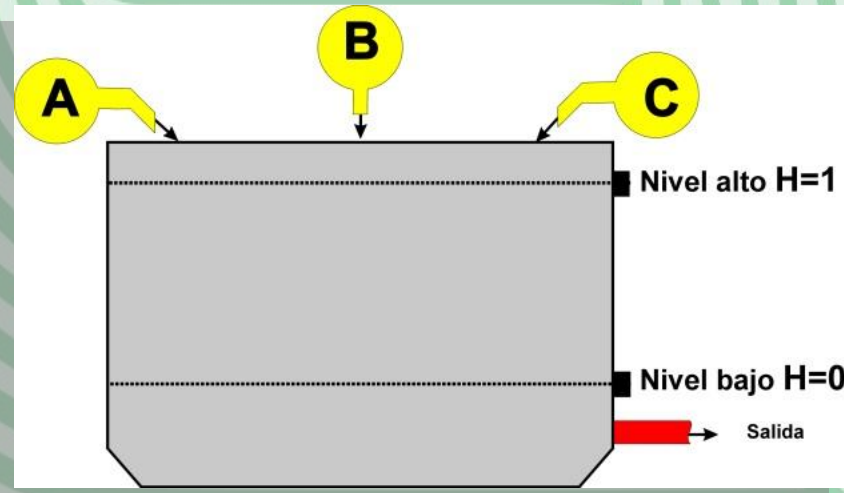


Diseñe un Sistema Secuencial usando la Máquina de Moore que controle el llenado de un Tanque con las siguientes especificaciones:

El sistema consta de tres bombas llamadas "**A**", "**B**" y "**C**" y un sensor de nivel "**H**" que indica con **H=1** Tanque lleno y con **H=0** Tanque vacío, el sistema deberá de trabajar bajo la siguiente secuencia:

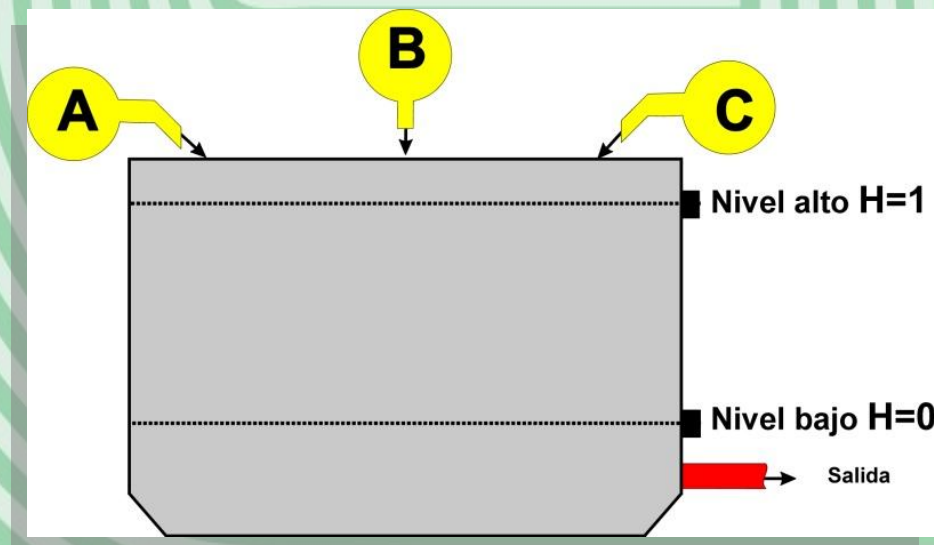


- a) El llenado debe iniciarse con la bomba "A" cuando el tanque se encuentre vacío ($H=0$), y debe detenerse cuando el tanque esté lleno ($H=1$).
- b) Si el tanque se vacía de nuevo ($H=0$), el llenado debe realizarse con la bomba "B" hasta que el tanque esté lleno ($H=1$), y luego debe detenerse.

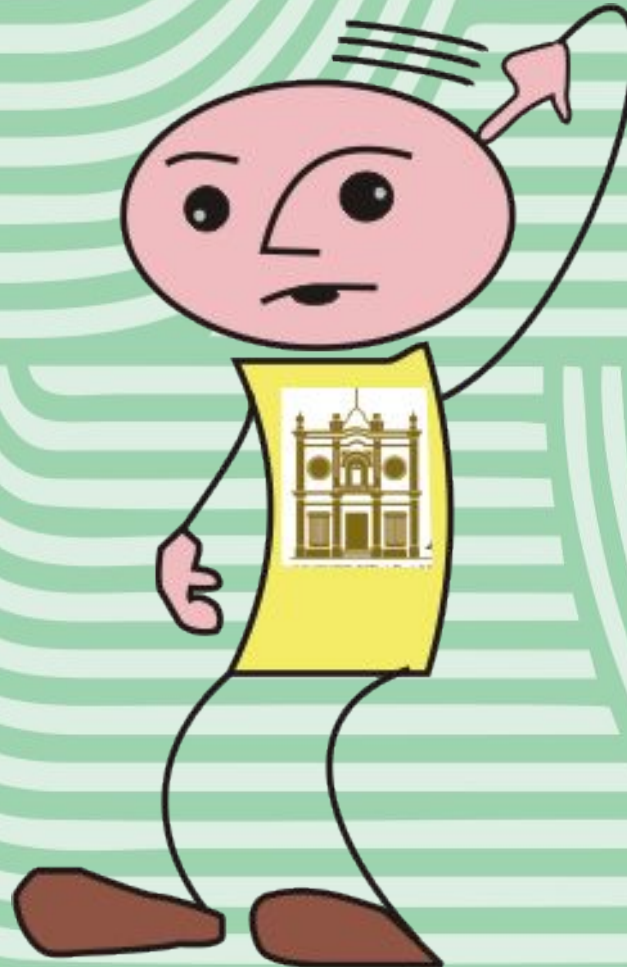


- c) Si de nuevo se vacía el Tanque ($H=0$), el llenado deberá hacerse con la bomba "C" hasta llenar el Tanque y entonces desconectarla.
- d) Si de nuevo se vacía el Tanque ($H=0$), el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que las tres bombas se alternen en su funcionamiento.

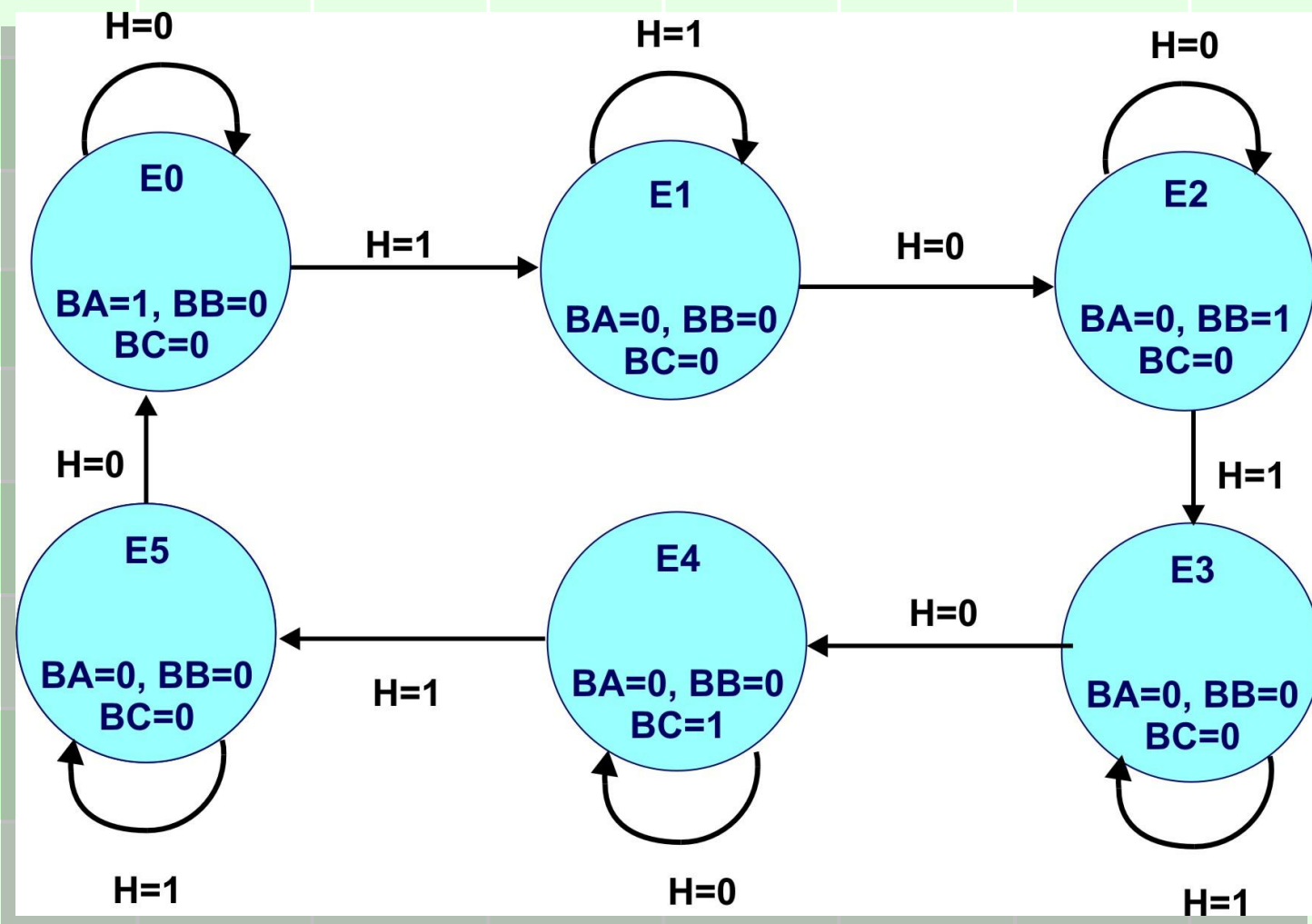
De esta manera, se asegura que las tres bombas se utilicen de manera equitativa y que el tanque siempre esté lleno.



Como seria el diagrama de transición ?

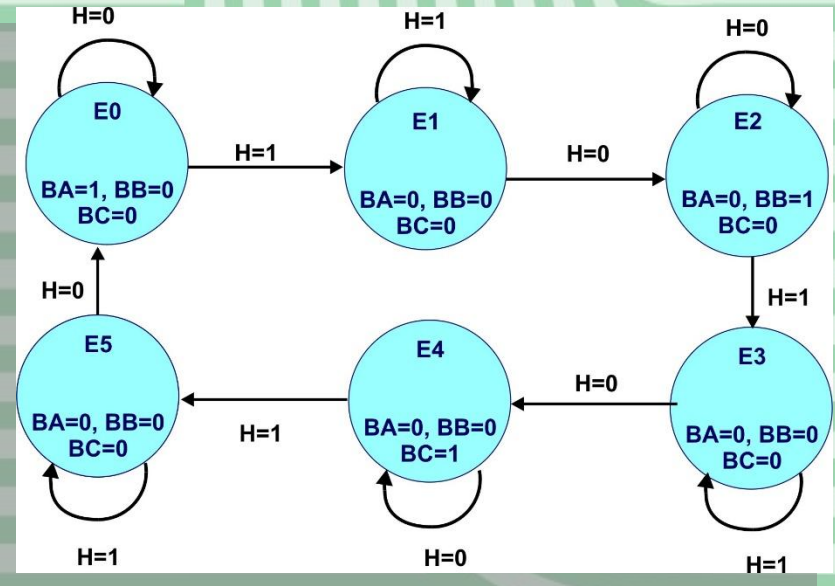


JAGG

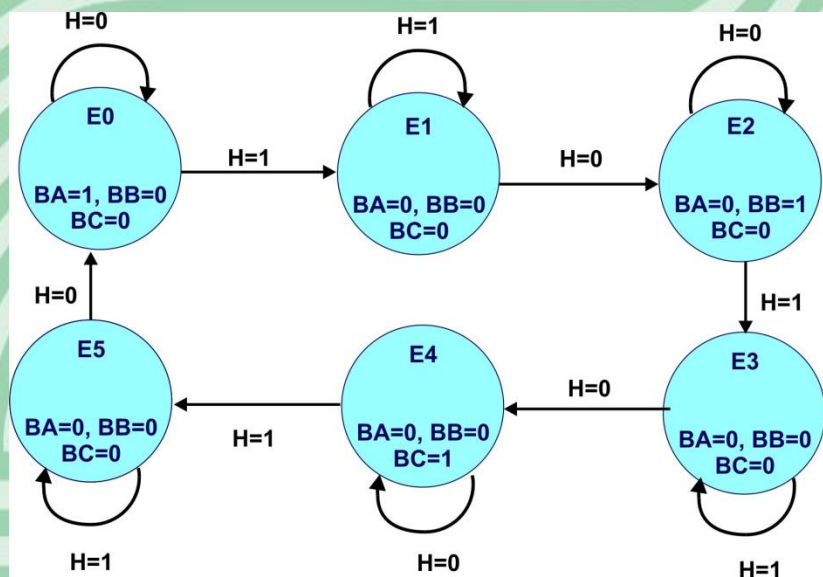


3 Flip Flops

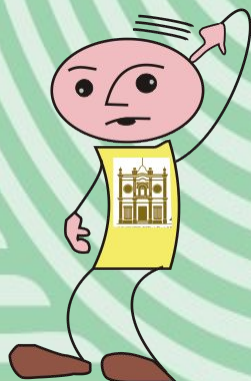
	Q2	Q1	Q0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1



3 Flip Flops

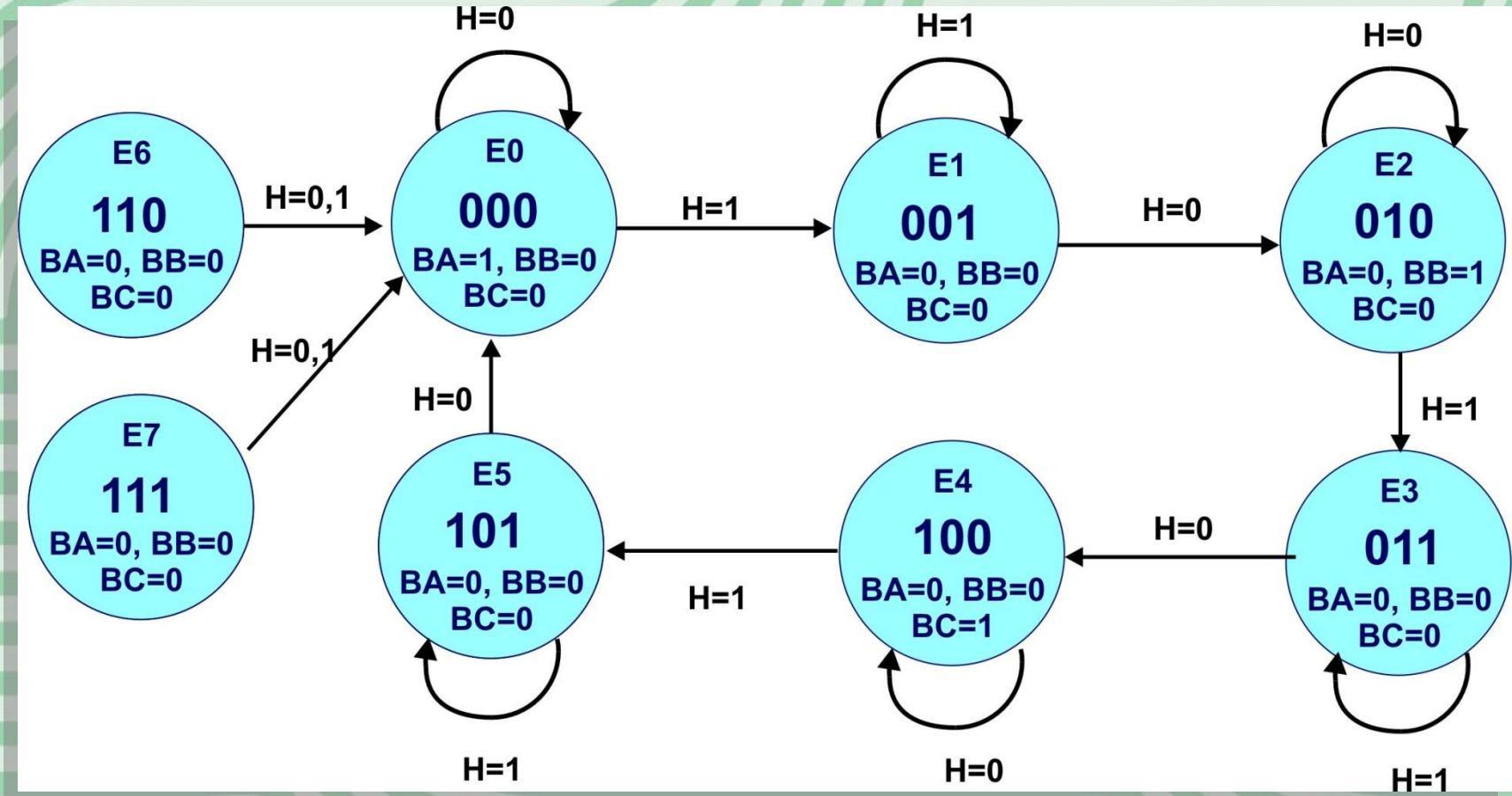


Que Hacer con E6 y E7 ?



	Q2	Q1	Q0
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

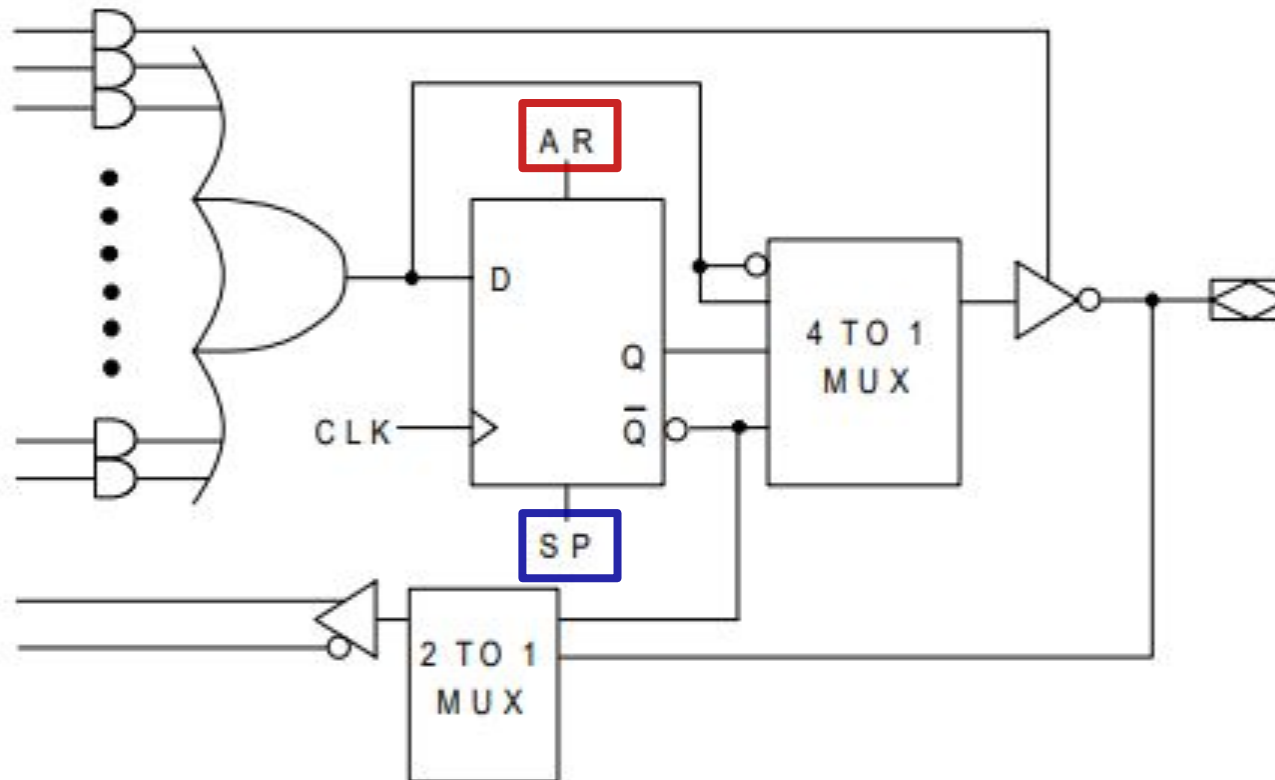
AGG



Estado Presente	Estado siguiente		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
E6	E0	E0	0	0	0
E7	E0	E0	0	0	0

JAGG

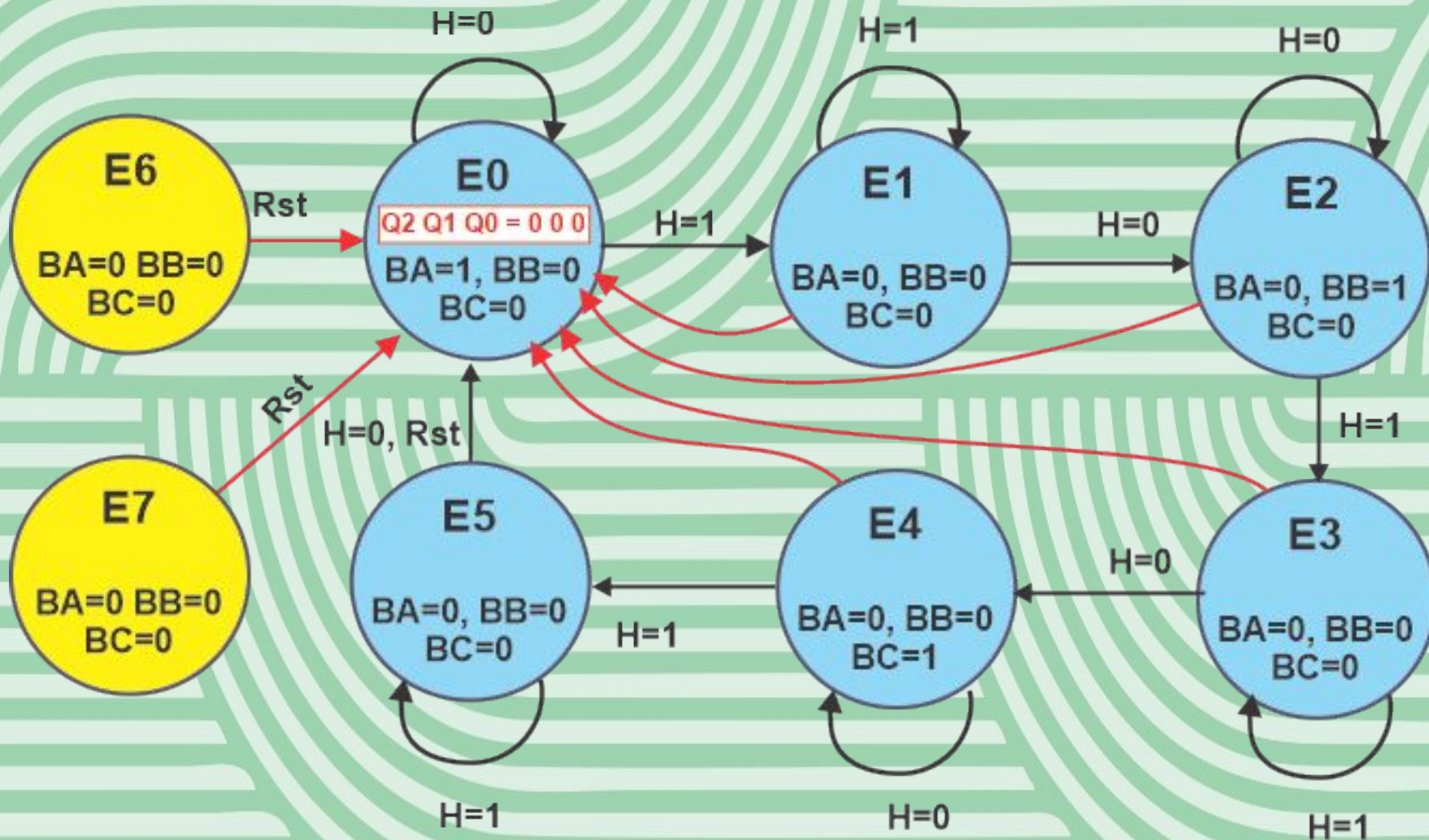
Asynchronous Reset AR



GAL22V10 OUTPUT LOGIC MACROCELL (OLMC)

Synchronous Preset SP

Diagrama de Transición, con Rst



Estado Presente	Estado siguiente		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

Module tresb

“Entrdas

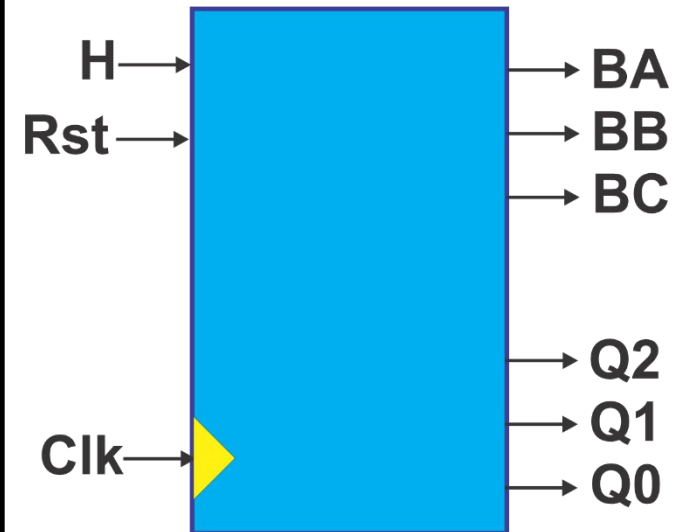
Clk, H, Rst PIN 1,2,3;

1 “Salidas Comb

BA,BB,BC pin 19..17 istype ‘com’;

“Salidas Reg

Q2..Q0 Pin 16..14 istype ‘reg’;



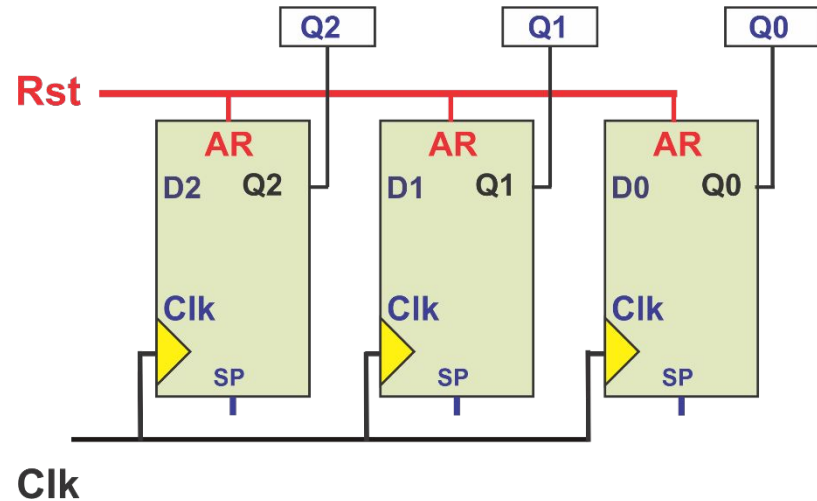
2

$E=[Q2..Q0];$

Equations

$E.clk=Clk;$

$E.Ar = Rst;$



3

Declarations

$E0=[0,0,0];$

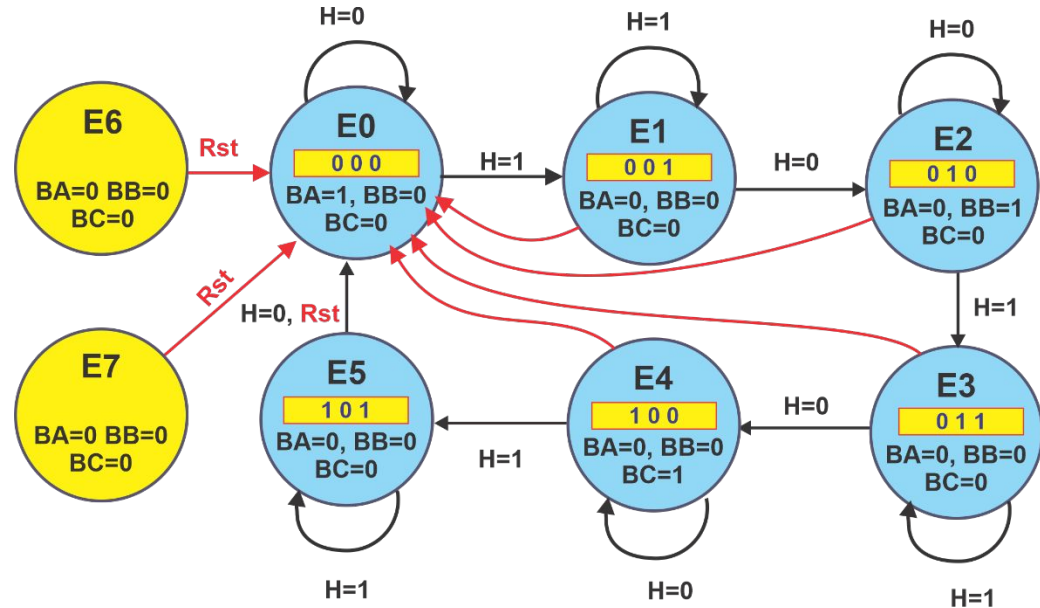
$E1=[0,0,1];$

$E2=[0,1,0];$

$E3=[0,1,1];$

$E4=[1,0,0];$

$E5=[1,0,1];$



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;

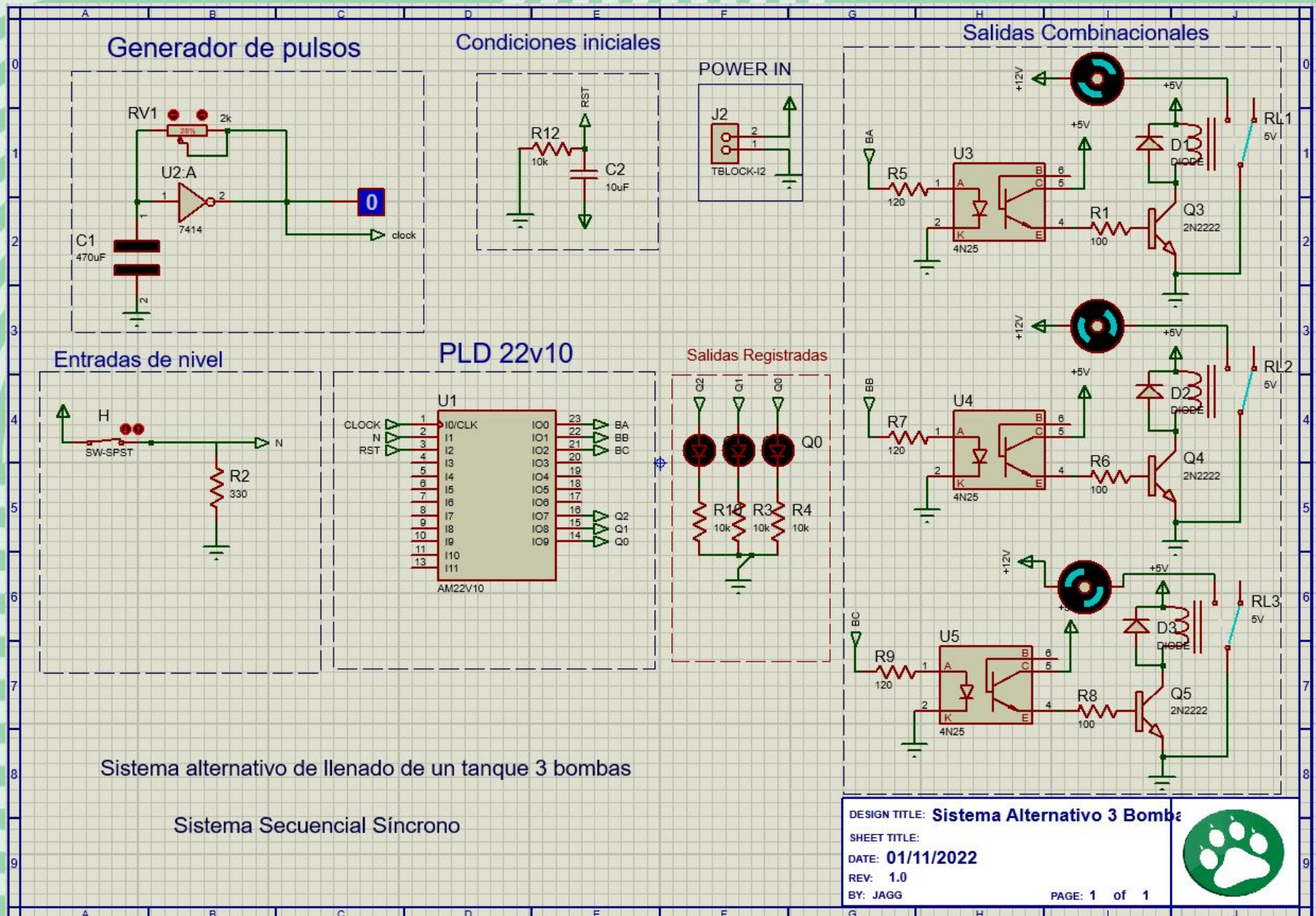
State E4:

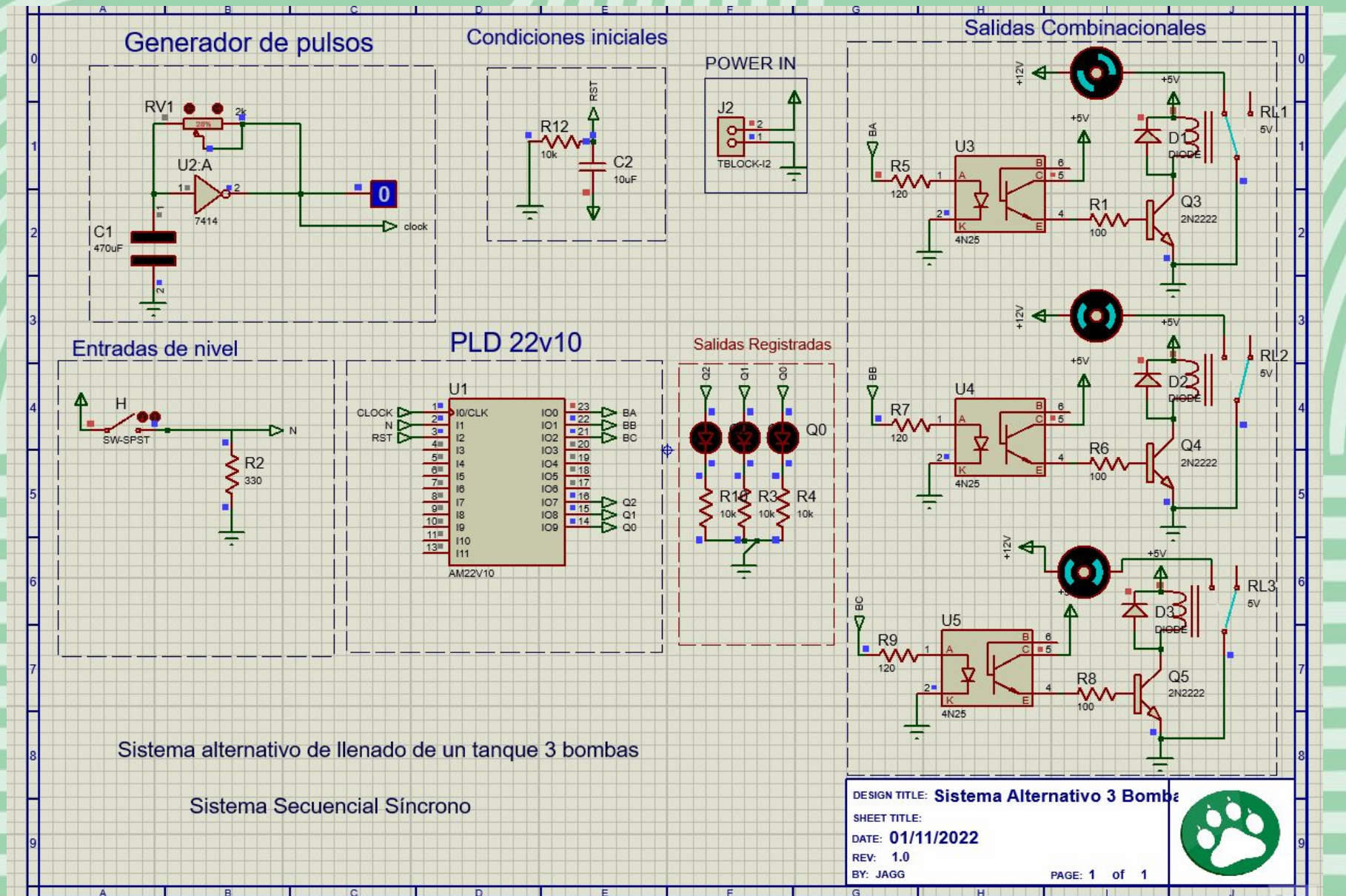
BA=0;BB=0;BC=1;
IF H then E5;
IF !H then E4;

State E5:

BA=0;BB=0;BC=0;
IF H then E5;
IF !H then E0;

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





Diseño de sistemas Secuenciales



Ing. Mario René Montante Pardo
DEP (1933-2019)

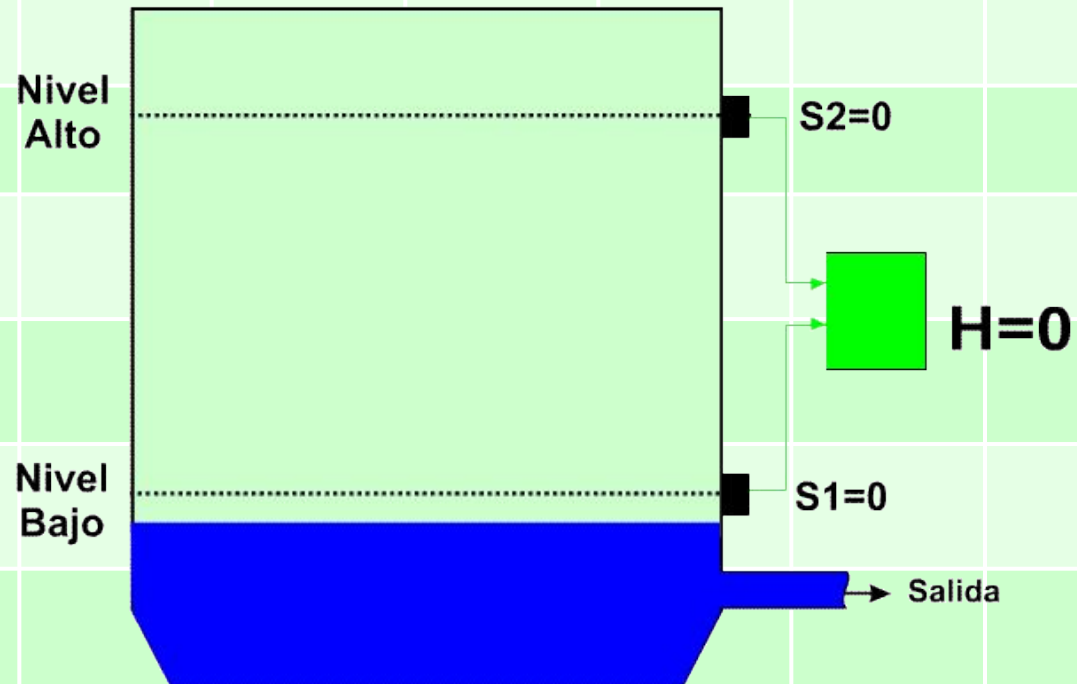
**No nos atrevemos a muchas cosas porque son difíciles,
pero son difíciles porque no nos atrevemos a hacerlas.**

Lucio Anneo Séneca

*dejó un legado invaluable en el campo de las matemáticas al crear el método Montante,
una herramienta revolucionaria que simplificó la resolución de determinantes mediante
números enteros y facilitó el trabajo de generaciones de estudiantes y profesionales*

JAGG

Teorema Fundamental



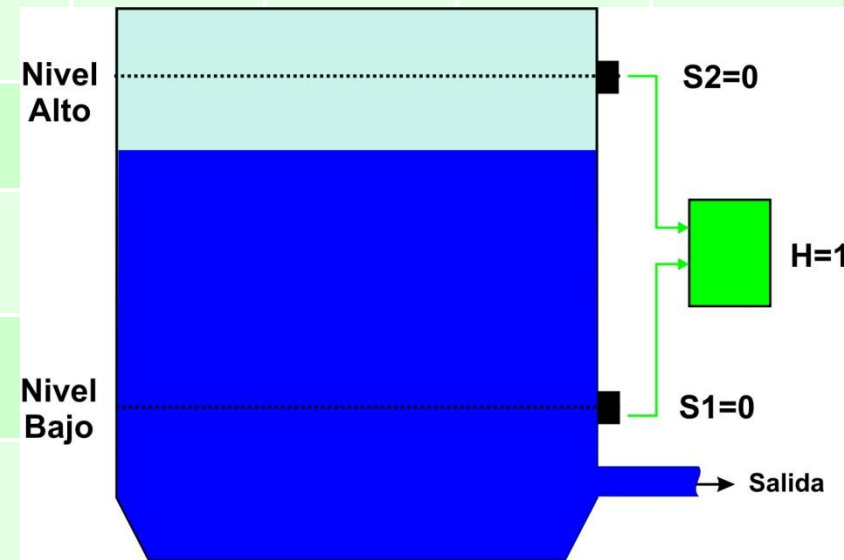
Detector de nivel

Diseñe un sistema secuencial **asíncrono** para detectar nivel de un tanque que cuenta con dos sensores llamados

S1 (Nivel Bajo)

S2 (Nivel Alto)

que contenga una salida H de modo que:



$H=0$ cuando el nivel va de S1 hacia S2 (subida) hasta que llegue a S2.

$H=1$ cuando el nivel va de S2 hacia S1. (bajada) hasta que llegue a S1 como lo indica el siguiente diagrama de tiempos

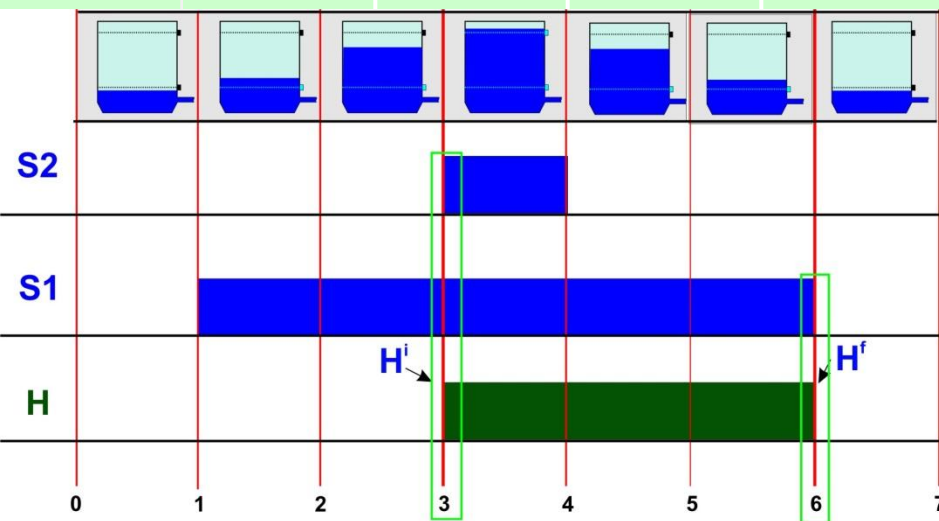
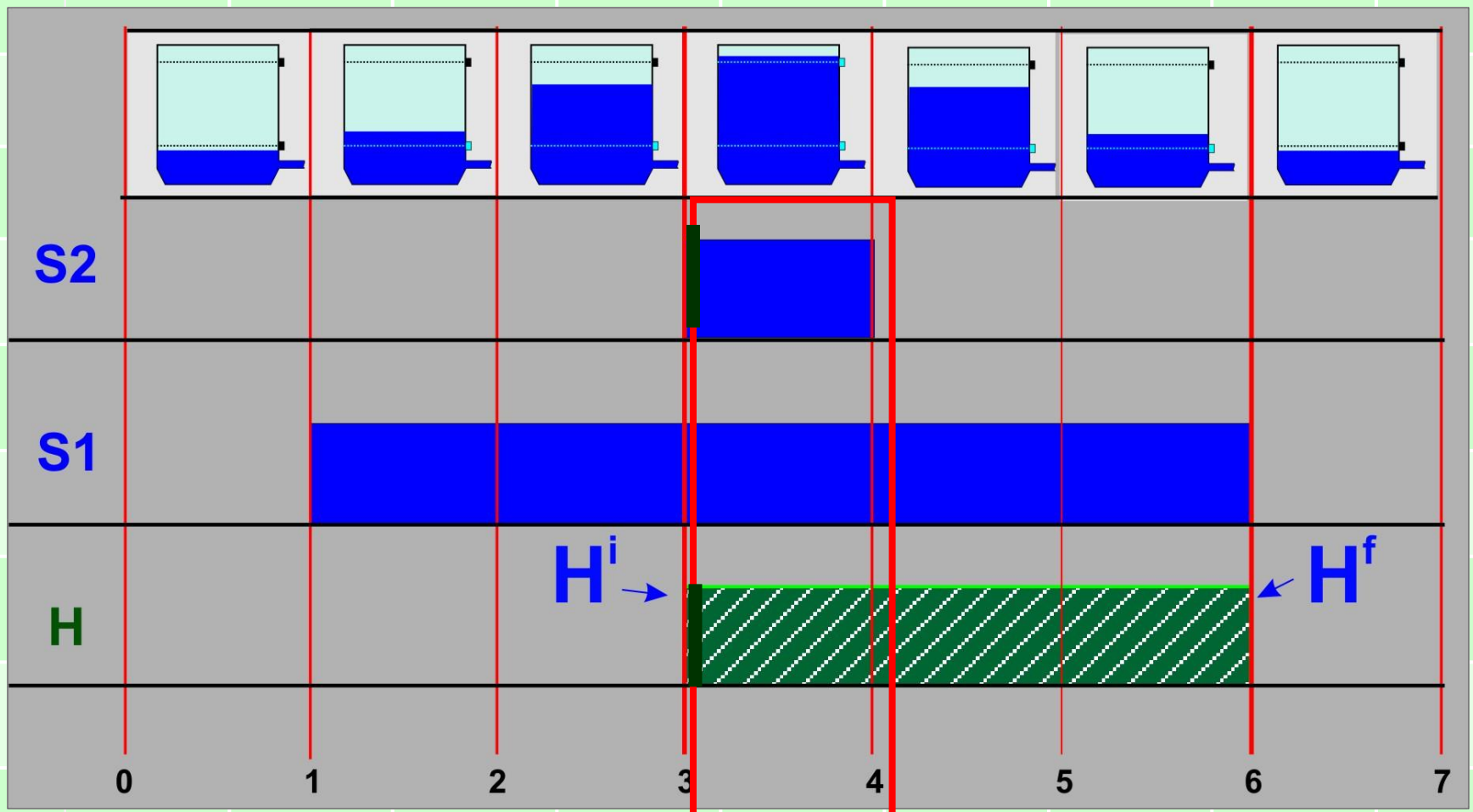


Diagrama de Tiempos

De que depende que $H=1$ o H^i

de $S2=1$

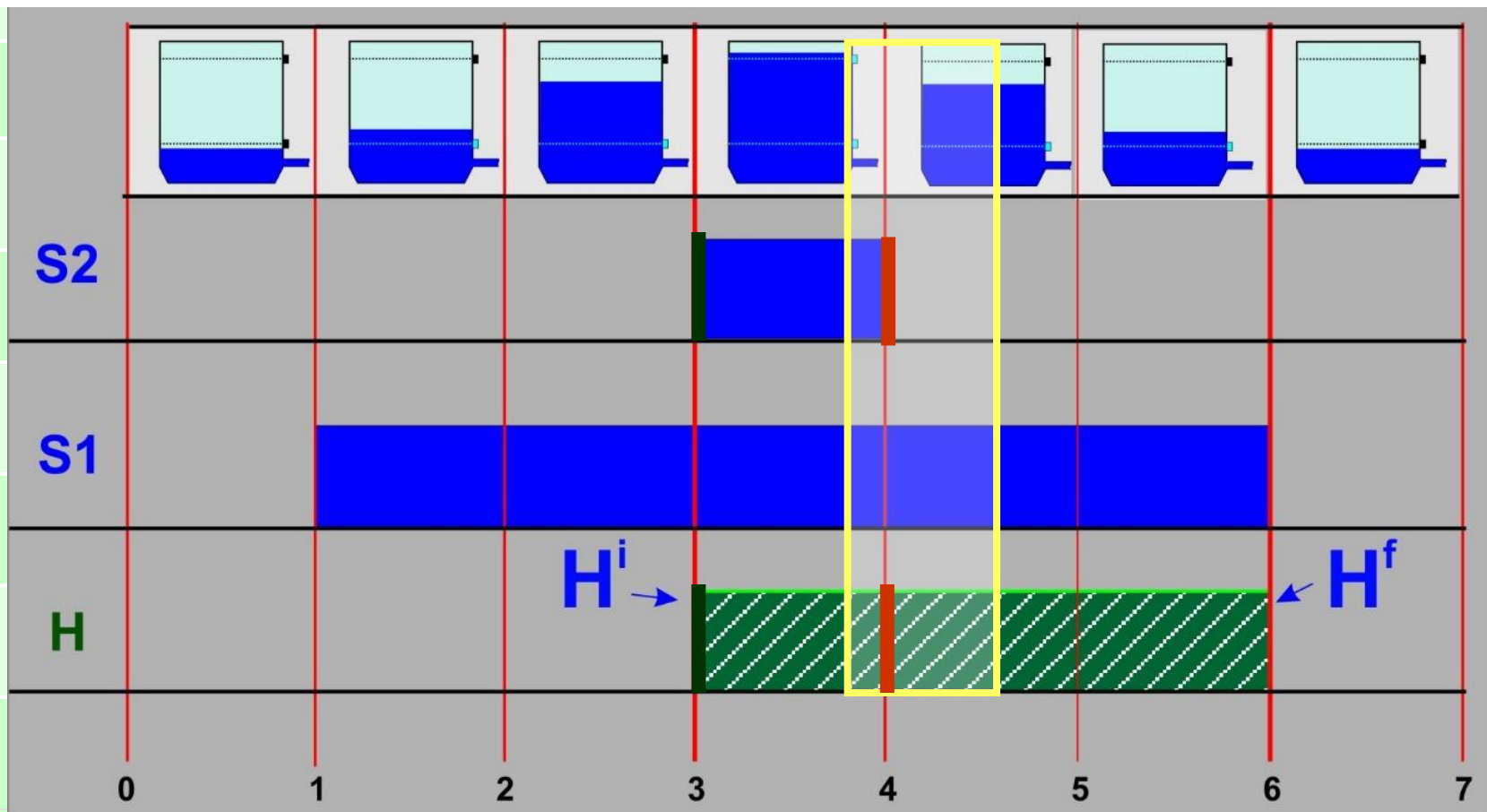


JAGG

Obtención de las Ecuaciones

$$H^i = S2$$

Pero cuando $S2=0$, $H=1$, como lo mantenemos

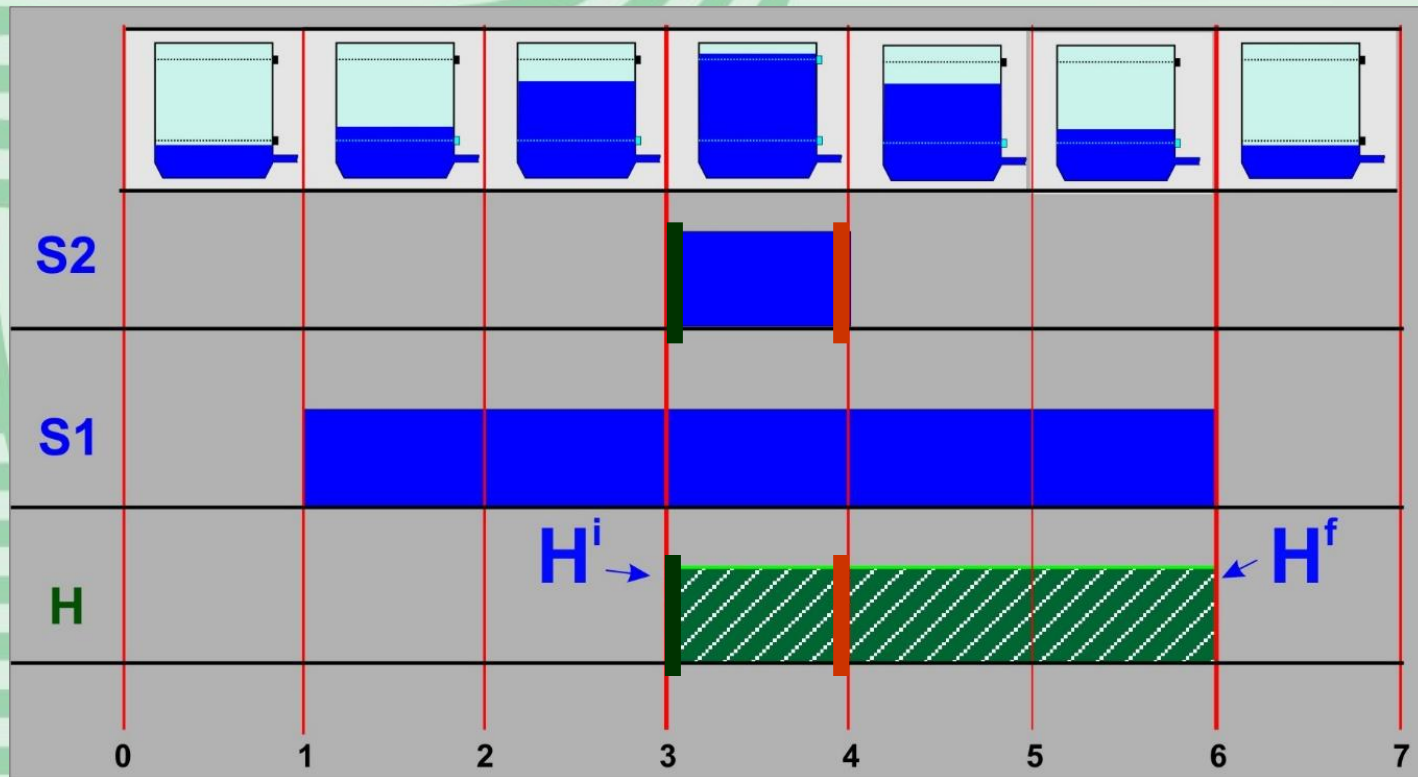
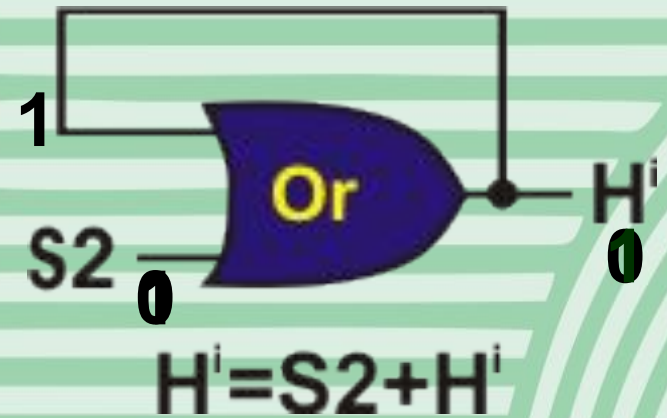


Obtención de las Ecuaciones

$$H^i = S2$$

Pero cuando $S2=0$, $H=1$

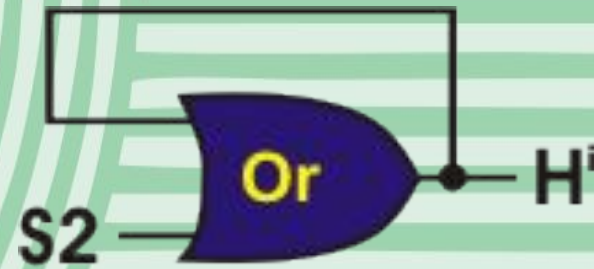
Para sostener el valor de $H=1$ usaremos una operación OR con retroalimentación



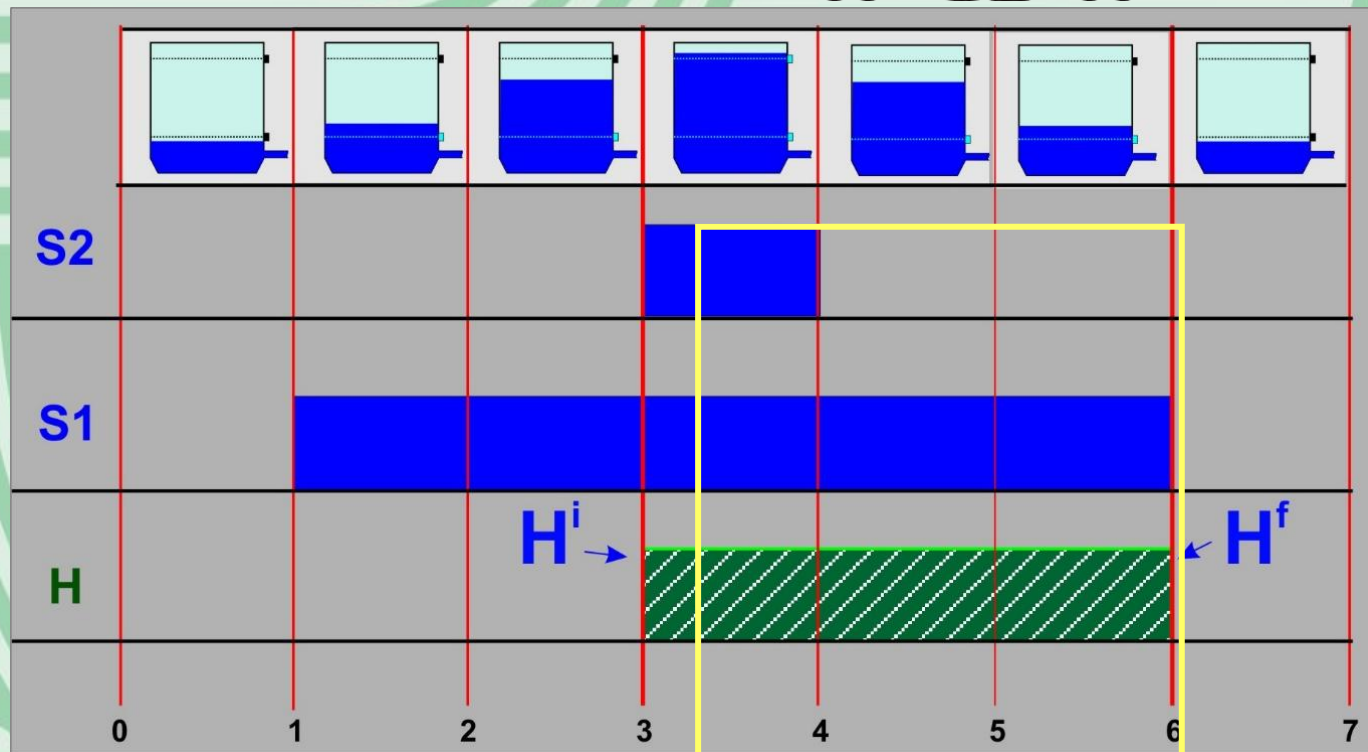
JAGG

Obtención de las Ecuaciones

$$H^i = S2 + H^i$$



$$H^i = S2 + H^i$$

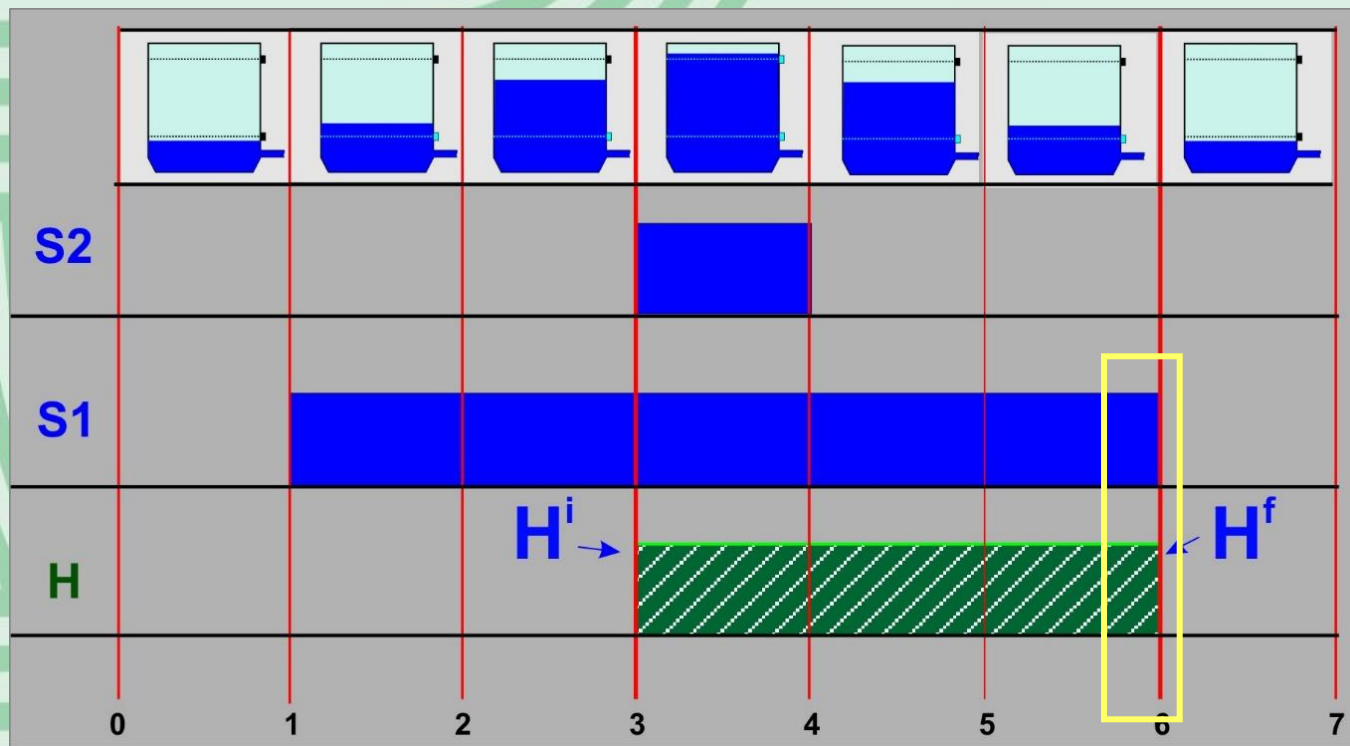


Obtención de las Ecuaciones

De que depende que $H=0$ o H^f

de $S1=0$

$$H^f = (S1)'$$



Aplicando el teorema fundamental

$$H = H^i (H^f)'$$



1997



2007

Dr. Cesar Elizondo Gonzalez

JAGG

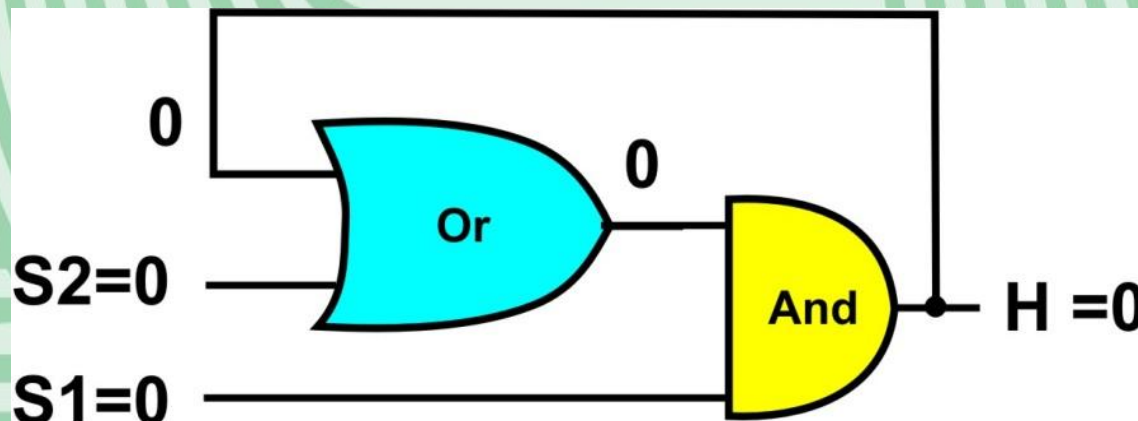
Aplicando el teorema fundamental

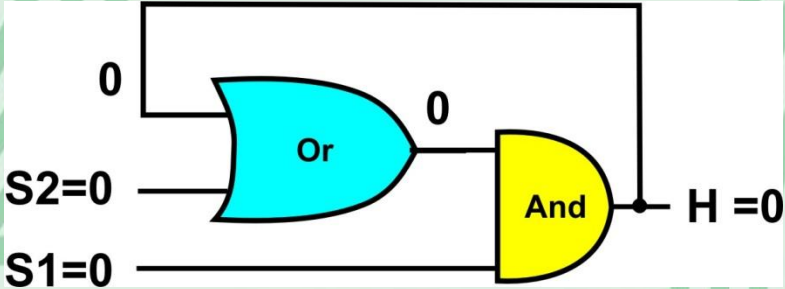
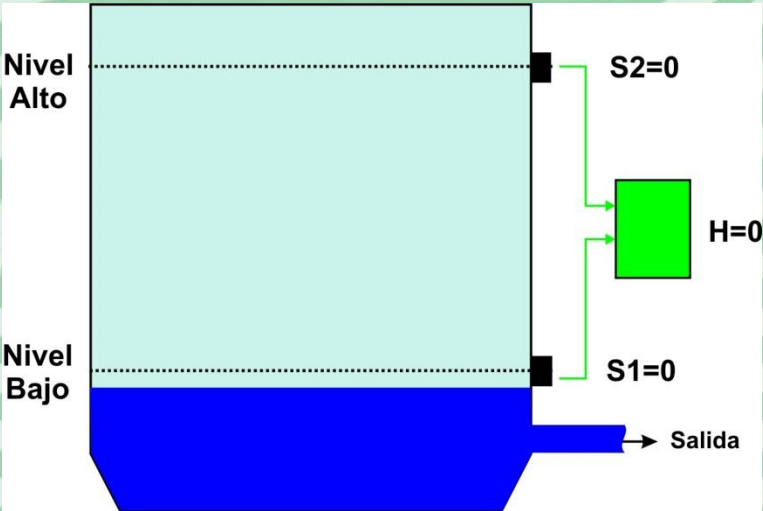
$$H = H^i (H^f)'$$

$$H^i = S2 + H, \quad H^f = S1'$$

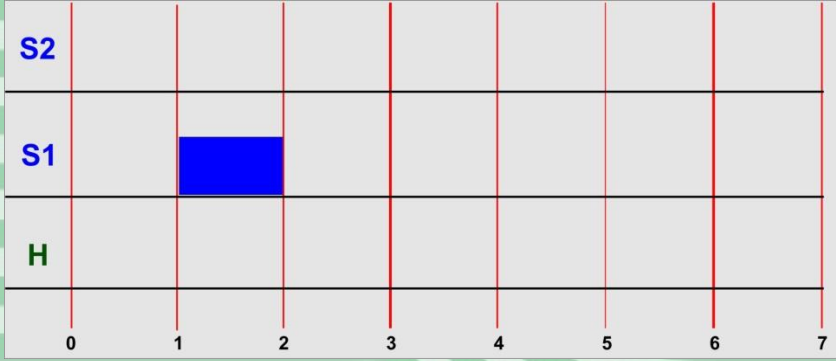
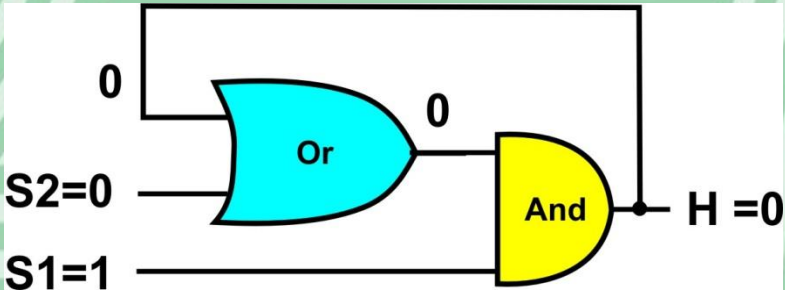
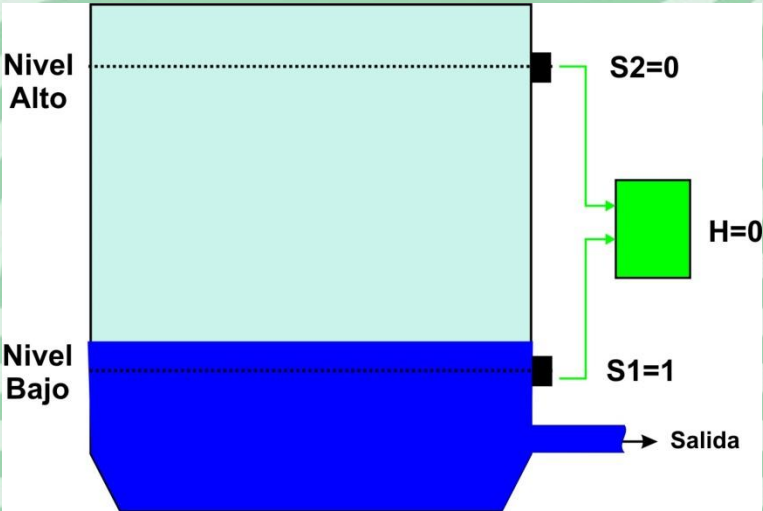
Obtenemos la ecuación para H

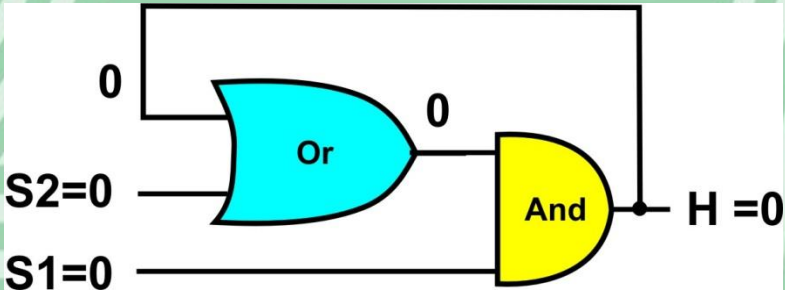
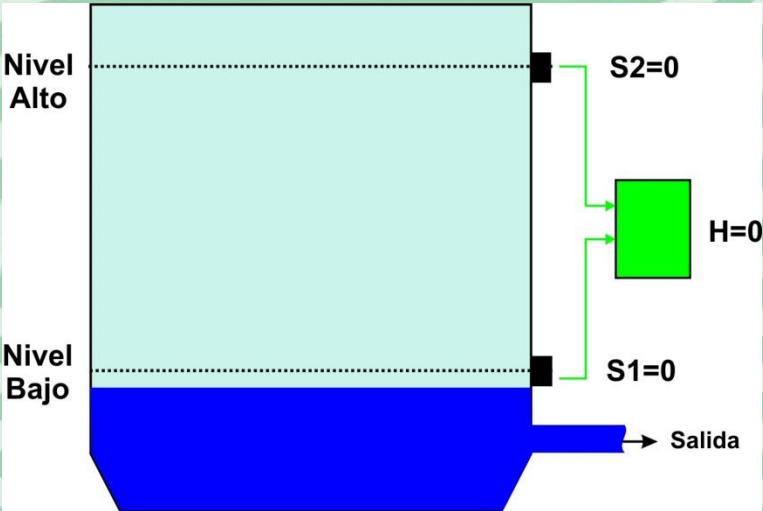
$$H = (S2 + H) S1$$



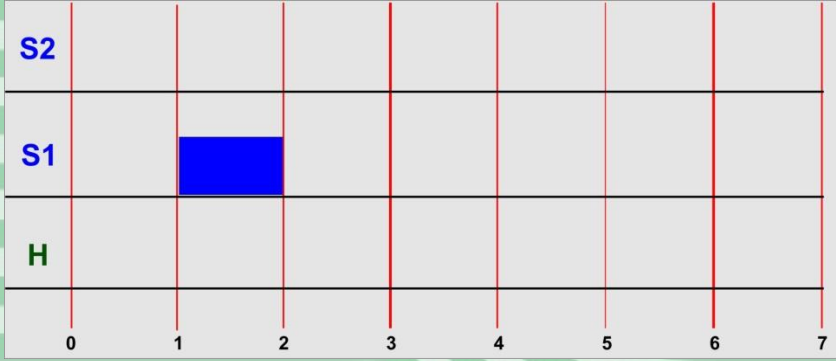
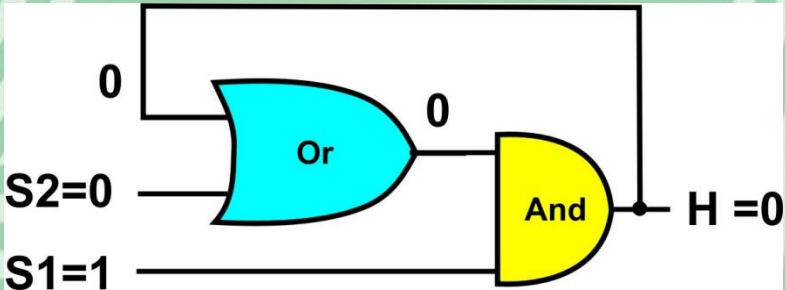
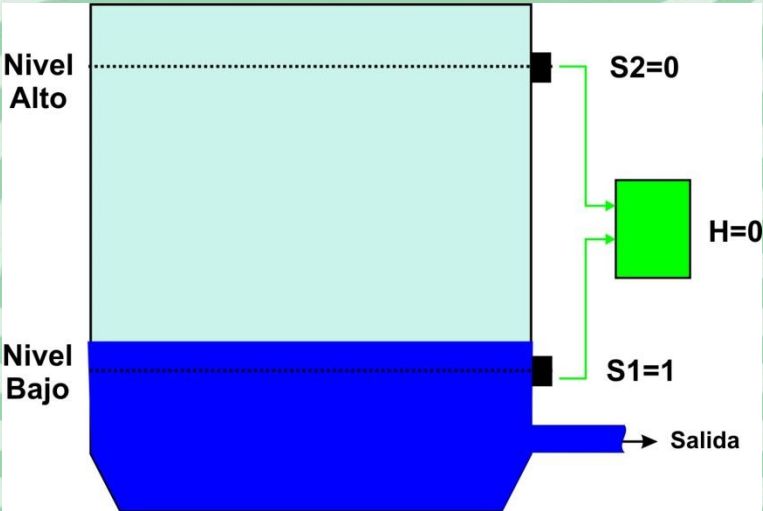


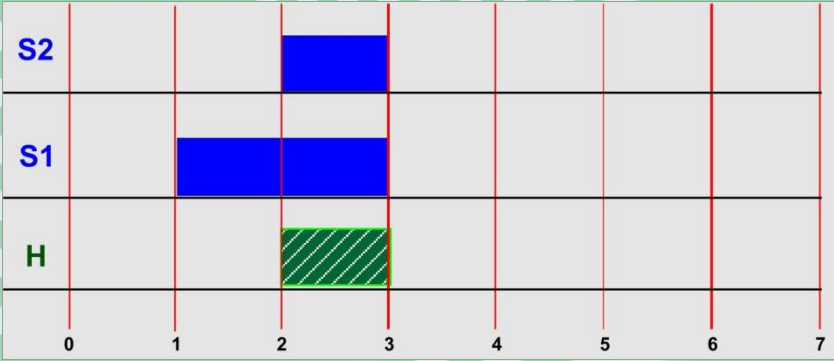
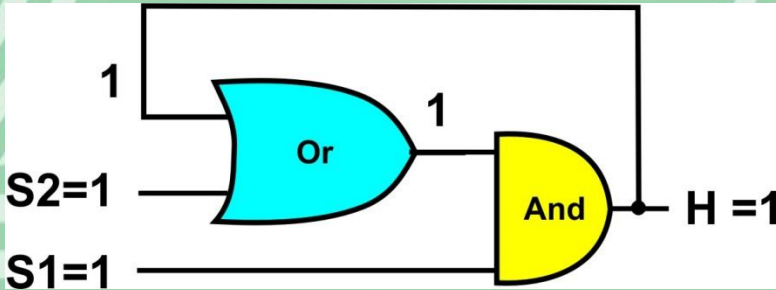
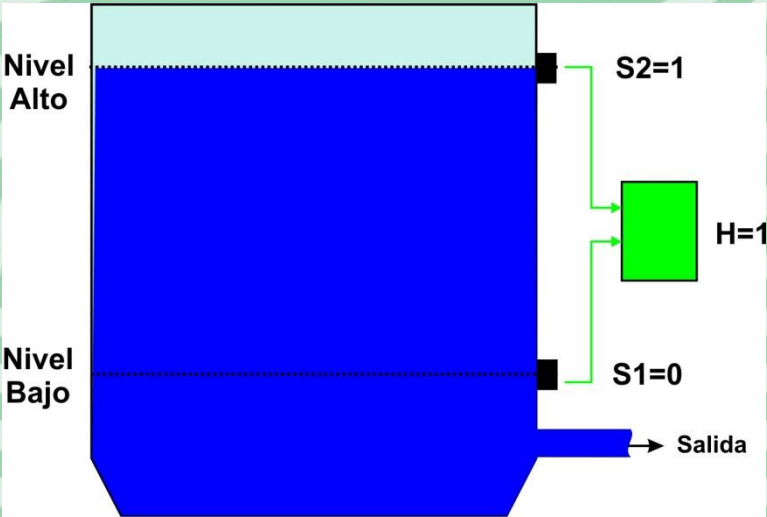
S2									
S1									
H									
	0	1	2	3	4	5	6	7	

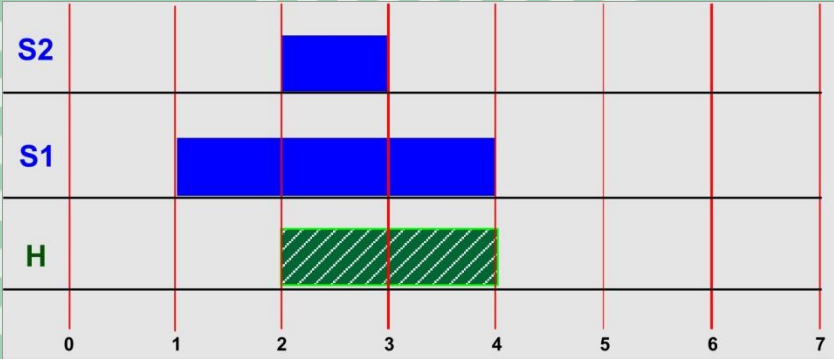
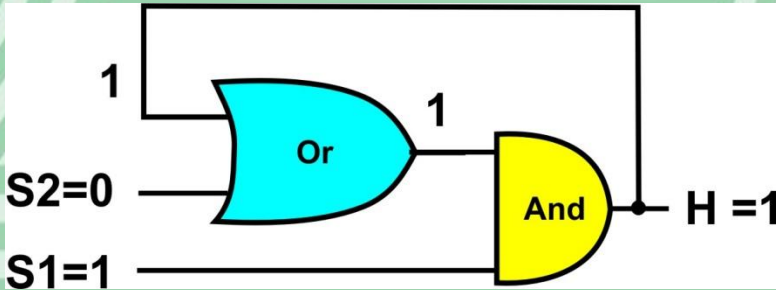
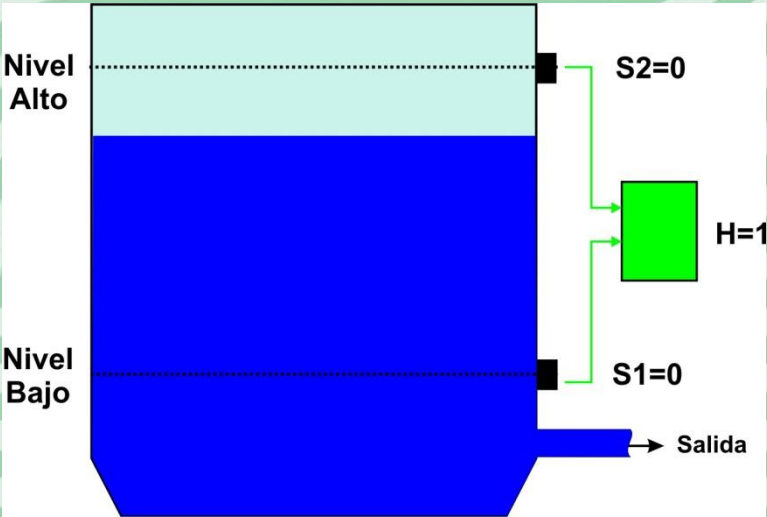


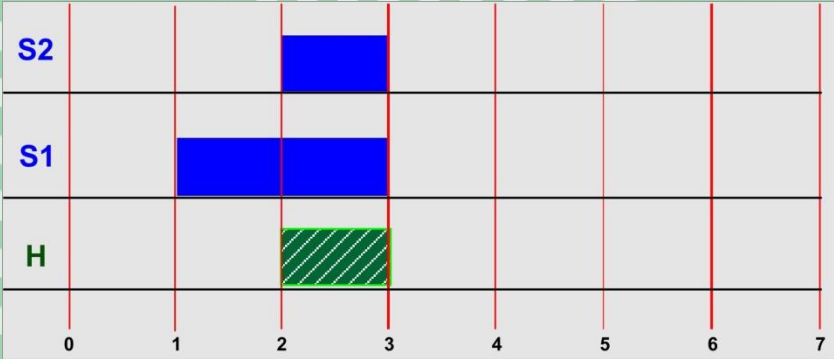
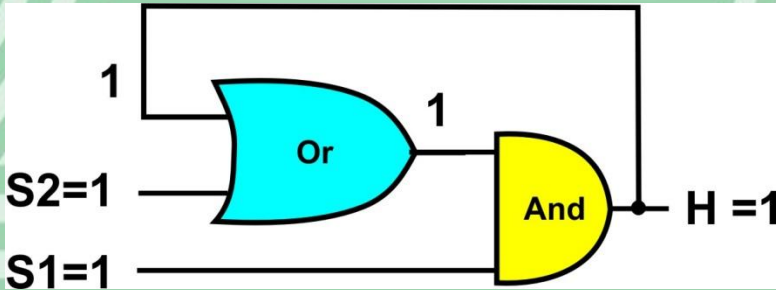
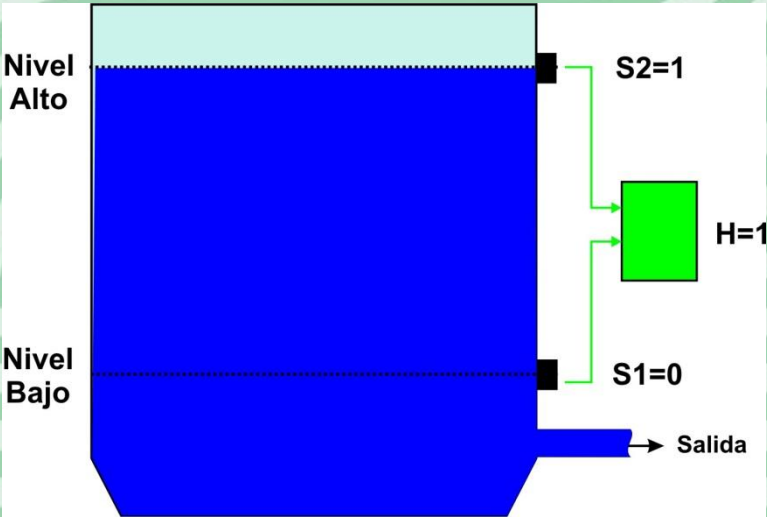


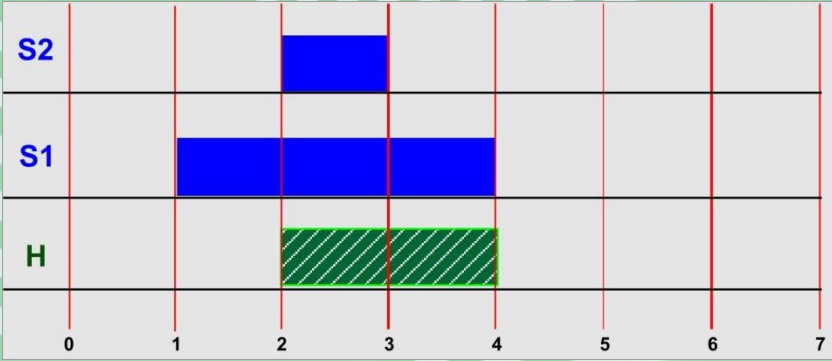
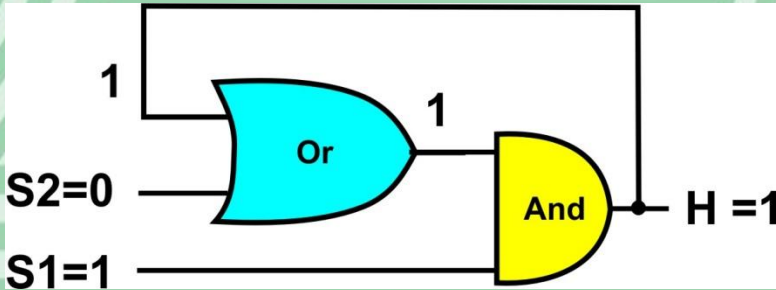
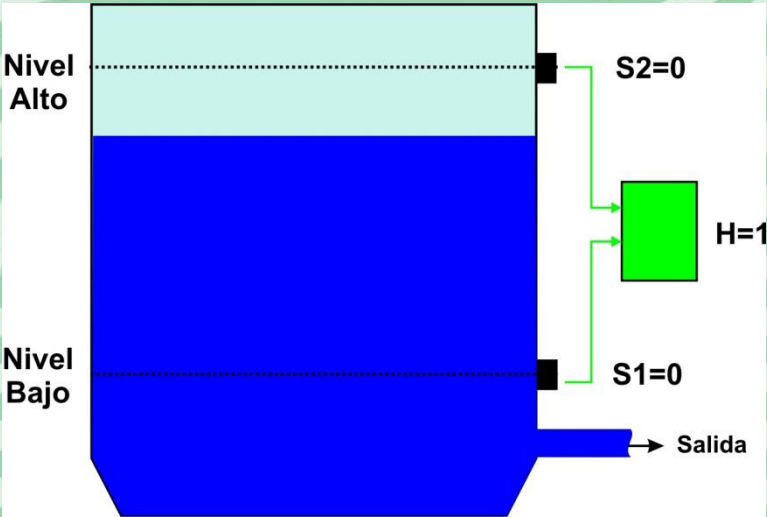
S2									
S1									
H									
	0	1	2	3	4	5	6	7	

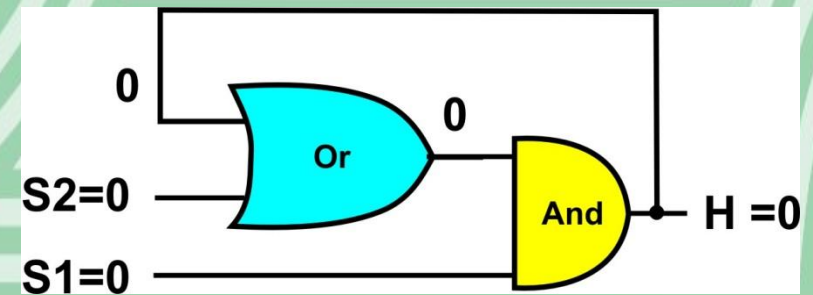
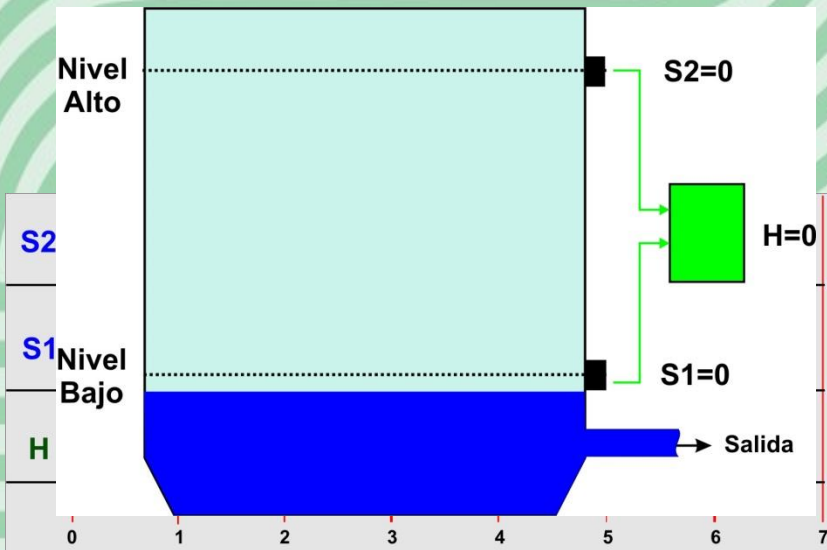




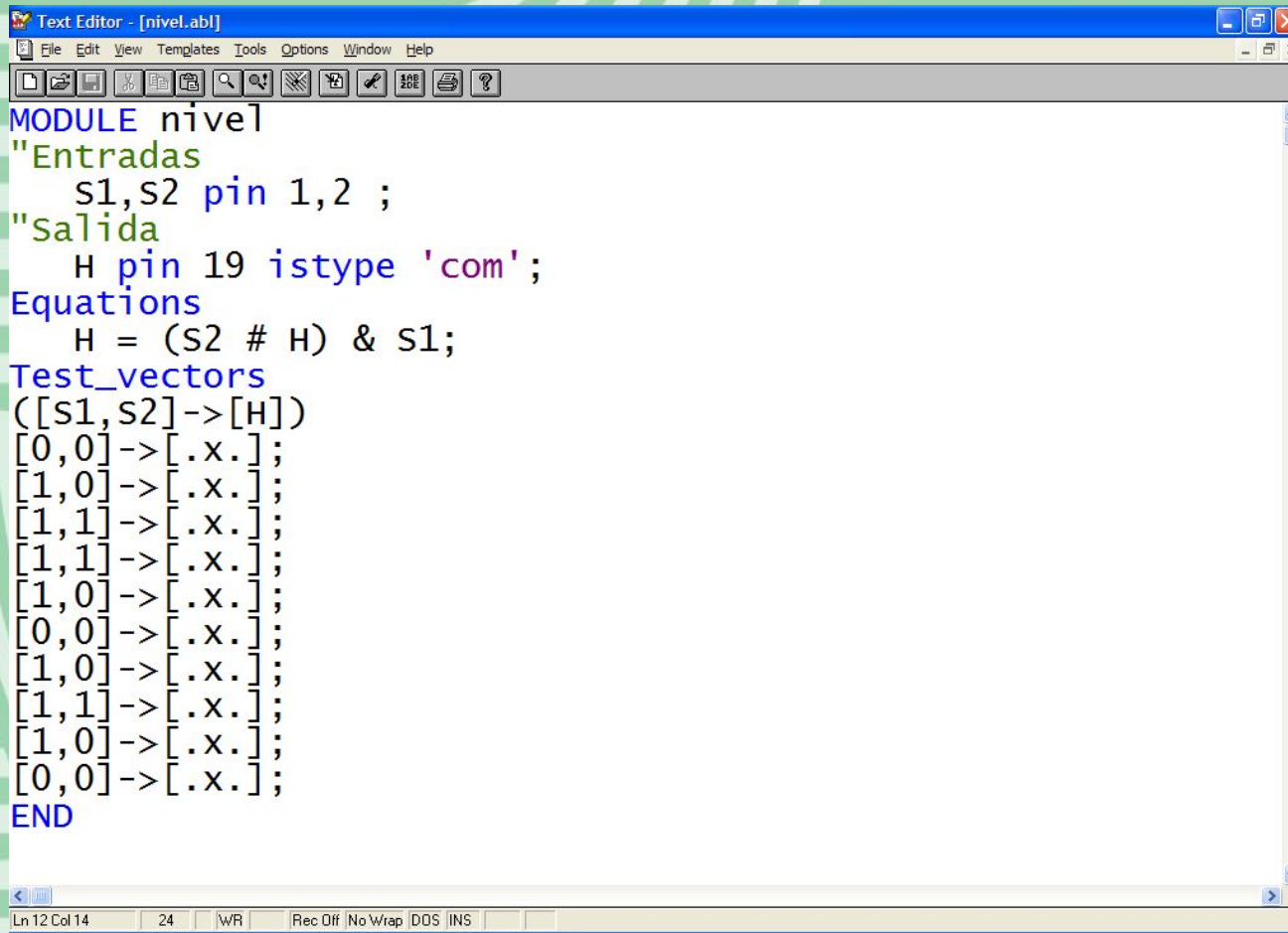






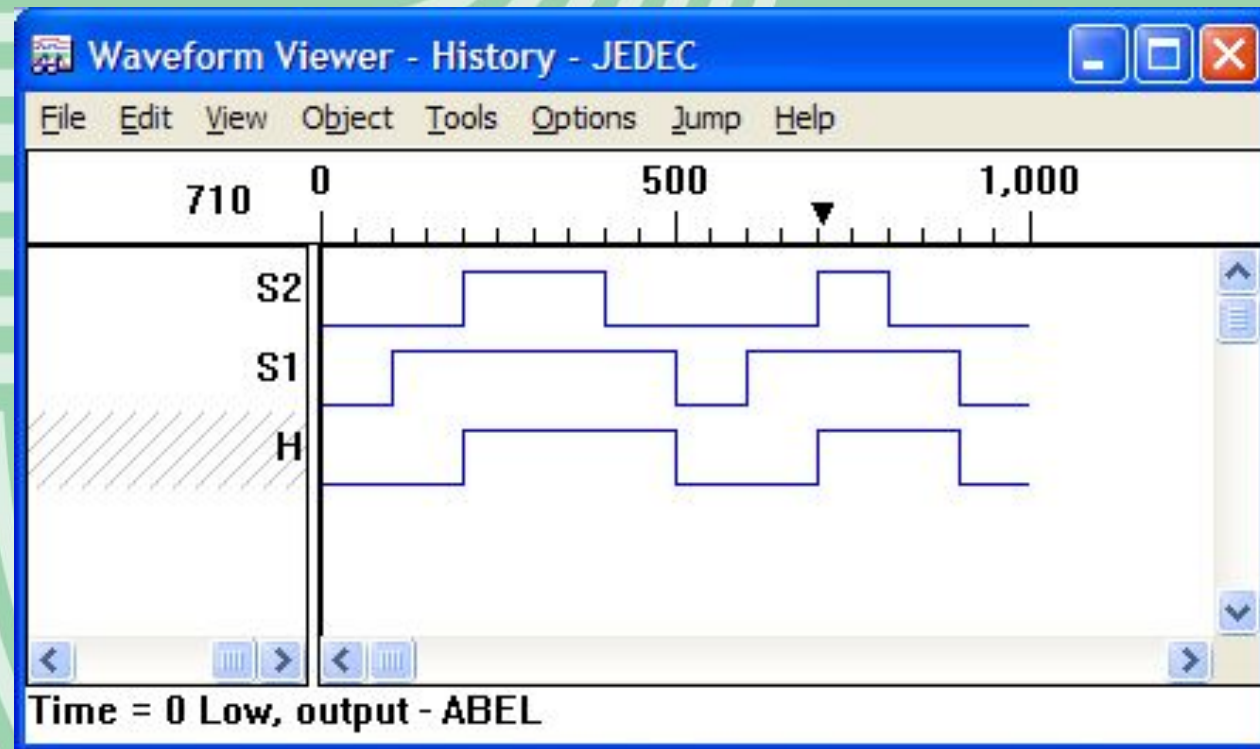


Archivo en ABEL-HDL



```
Text Editor - [nivel.abl]
File Edit View Templates Tools Options Window Help
[Icons]
MODULE nivel
  "Entradas
    s1,s2 pin 1,2 ;
  "Salida
    H pin 19 istype 'com';
  Equations
    H = (S2 # H) & s1;
  Test_vectors
    ([s1,s2]->[H])
    [0,0]->[.x.];
    [1,0]->[.x.];
    [1,1]->[.x.];
    [1,1]->[.x.];
    [1,0]->[.x.];
    [0,0]->[.x.];
    [1,0]->[.x.];
    [1,1]->[.x.];
    [1,0]->[.x.];
    [0,0]->[.x.];
  END
Ln 12 Col 14 24 WR Rec Off No Wrap DOS INS
```

Simulación



Si agregáramos el detector de nivel

MODULE tresbomb

"Entradas

Clk,S1,S2 Pin 1..3;

" Salidas Combinacionales

H,BA,BB,BC Pin 19,14..12 istype'com';

" Salidas Registradas

Q2,Q1,Q0 pin 17..15 istype'reg';

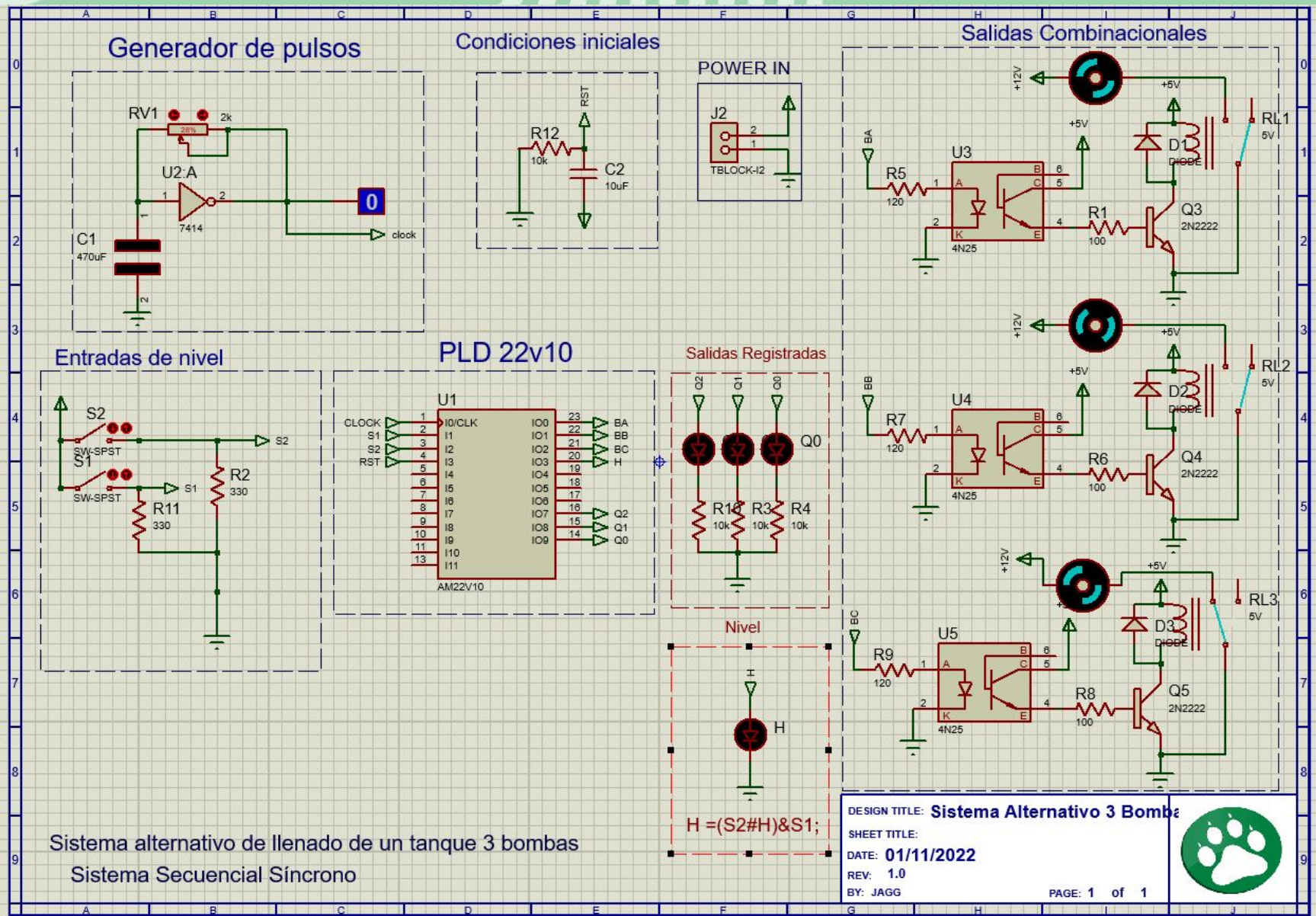
DECLARATIONS

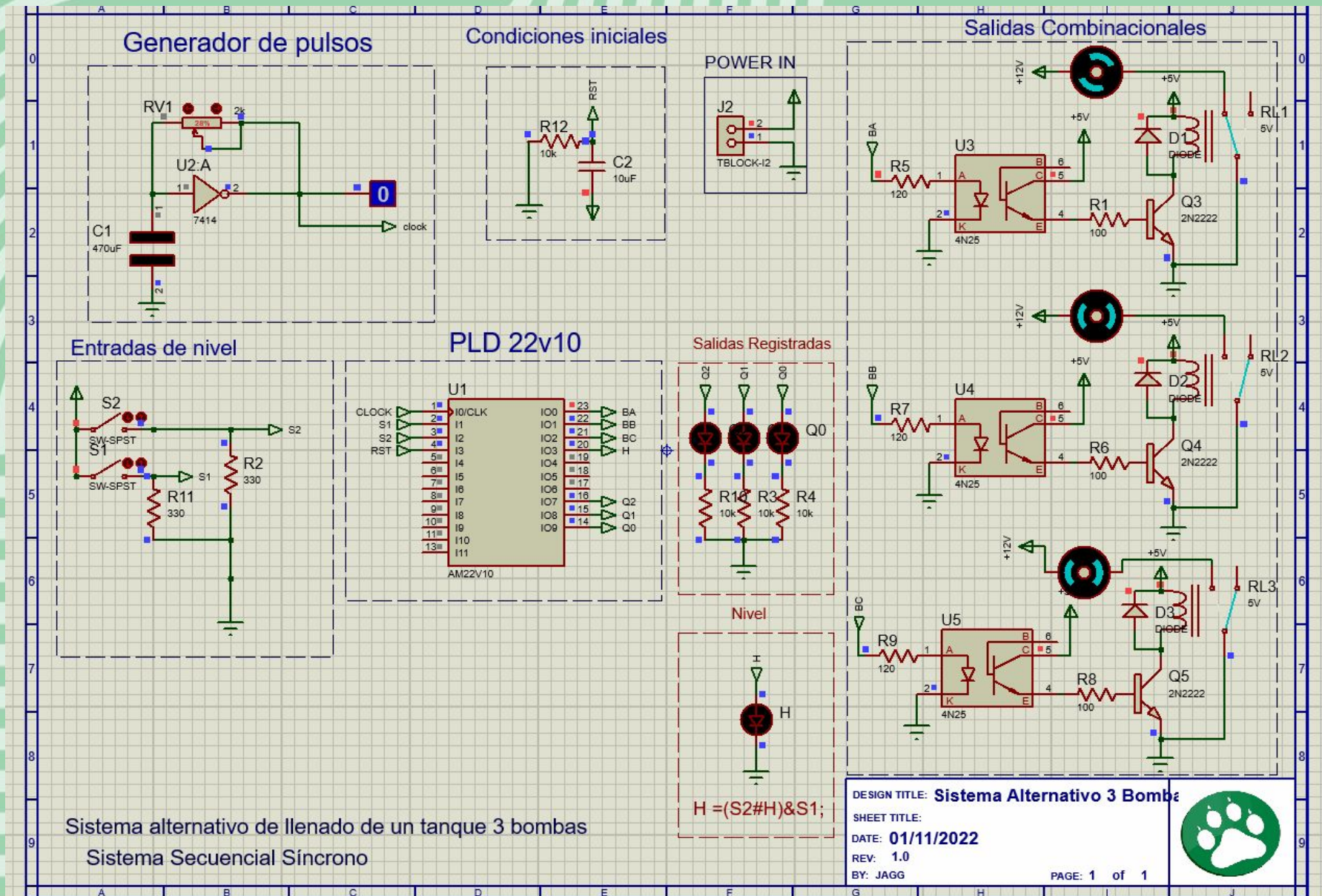
FIME=[Q2,Q1,Q0];

EQUATIONS

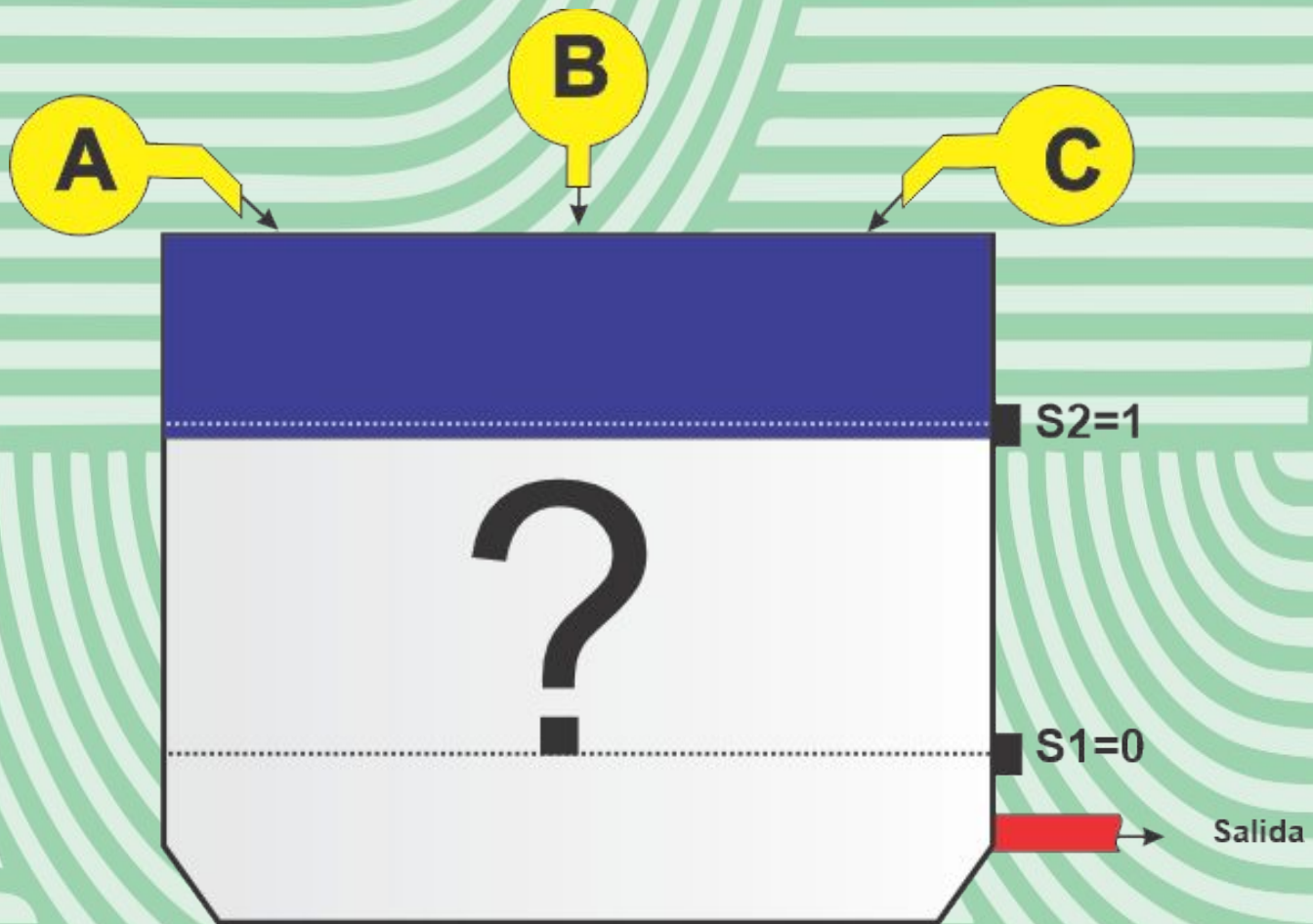
FIME.clk=Clk;

H =(S2#H)&S1;





S2=1, S1=0



Alarma

S2=1, S1=0

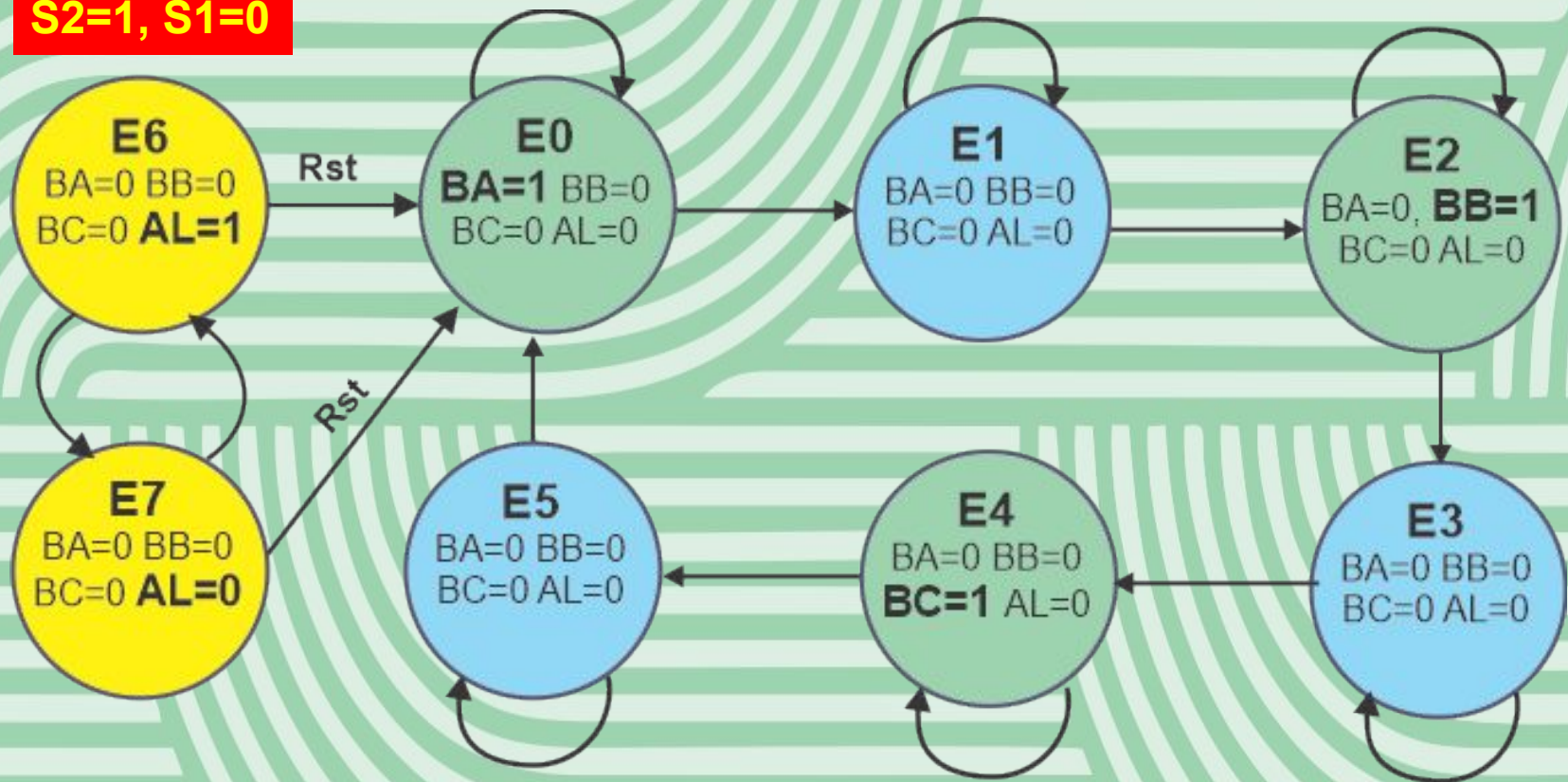


Tabla de Estado Siguiente

		Estado siguiente				Salidas Registradas			Salidas Combinacionales			
Entradas	S2	0	0	1	1							
	S1	0	1	0	1	Q2	Q1	Q0	BA	BB	BC	AL
Estado presente	E0	E0	E0	E6	E1	0	0	0	1	0	0	0
	E1	E2	E1	E6	E1	0	0	1	0	0	0	0
	E2	E2	E2	E6	E3	0	1	0	0	1	0	0
	E3	E4	E3	E6	E3	0	1	1	0	0	0	0
	E4	E4	E4	E6	E5	1	0	0	0	0	1	0
	E5	E0	E5	E6	E5	1	0	1	0	0	0	0
	E6	E7	E7	E7	E7	1	1	0	0	0	0	1
	E7	E6	E6	E6	E6	1	1	1	0	0	0	0

JAGG

MODULE TBAI

"sistema alternativo de 3 bombas

"con alarma

"27 abril 2022 Rev 1.0

" 1 Nov 2022 Rev 1.1

" JAGG

"Entradas

Clk,S1,S2,Rst pin 1..4;

" Salidas Combinacionales

BA,BB,BC,AL Pin 17..19, 23 istype'com';

" Salidas Registradas

Q2,Q1,Q0 pin 14..16 istype'reg';

DECLARATIONS

FIME=[Q2,Q1,Q0];

EQUATIONS

FIME.clk=Clk;

FIME.Ar=Rst;

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];

N=[S2,S1];

STATE_DIAGRAM FIME

STATE E0:

BA=1; BB=0; BC=0; AL=0;
IF N==0 THEN E0;"VACIO"
IF N==1 THEN E0;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E1;"LLENO"

STATE E1:

BA=0; BB=0; BC=0; AL=0;
IF N==0 THEN E2;"VACIO"
IF N==1 THEN E1;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E1;"LLENO"

STATE E2:

BA=0; BB=1; BC=0; AL=0;
IF N==0 THEN E2;"VACIO"
IF N==1 THEN E2;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E3;"LLENO"

STATE E3:

BA=0; BB=0; BC=0; AL=0;
IF N==0 THEN E4;"VACIO"
IF N==1 THEN E3;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E3;"LLENO"

STATE E4:

BA=0; BB=0; BC=1; AL=0;
IF N==0 THEN E4;"VACIO"
IF N==1 THEN E4;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E5;"LLENO"

STATE E5:

BA=0; BB=0; BC=0; AL=0;
IF N==0 THEN E0;"VACIO"
IF N==1 THEN E5;"MEDIO"
IF N==2 THEN E6;"INVALIDO"
IF N==3 THEN E5;"LLENO"

"Alarma

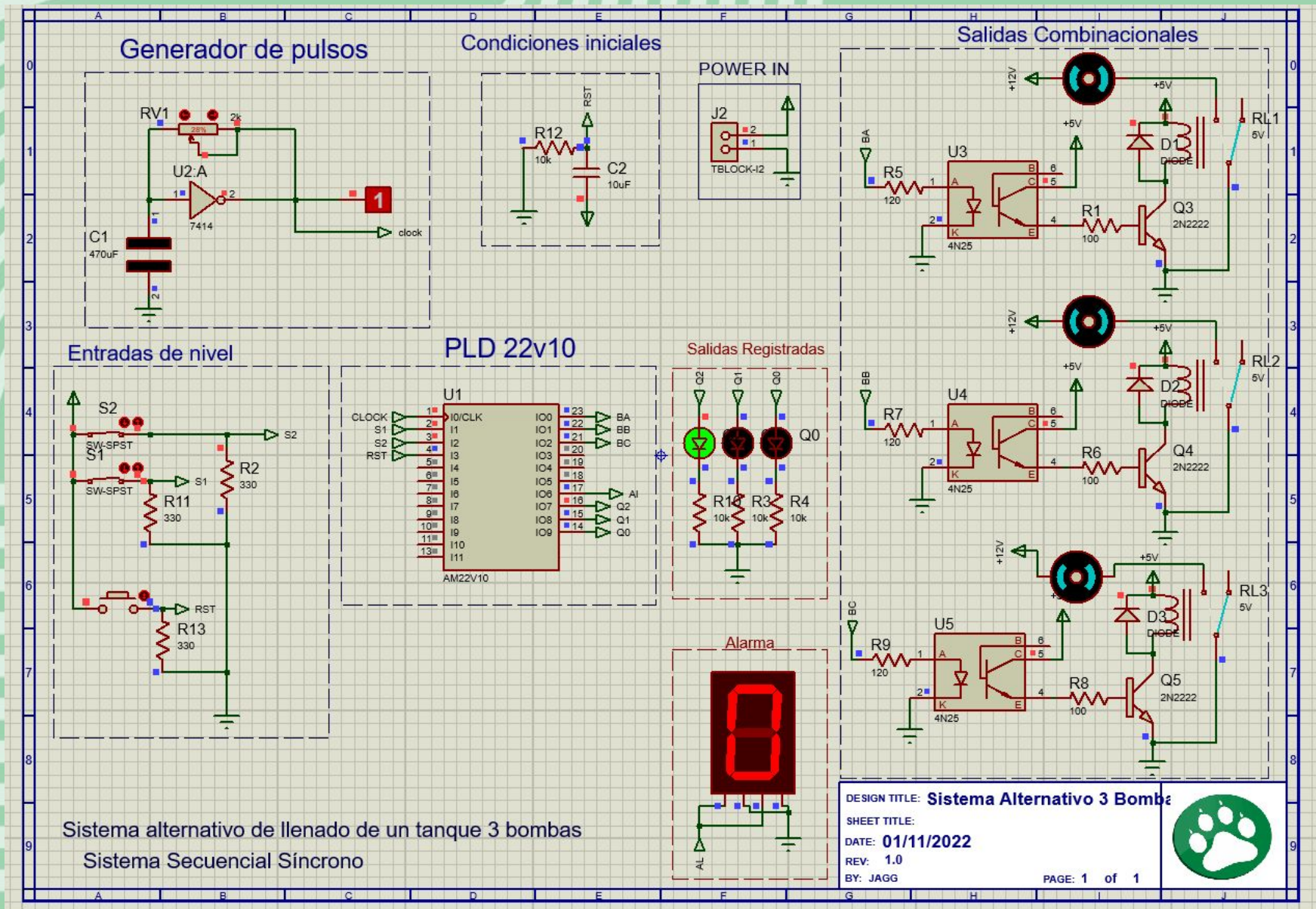
STATE E6:

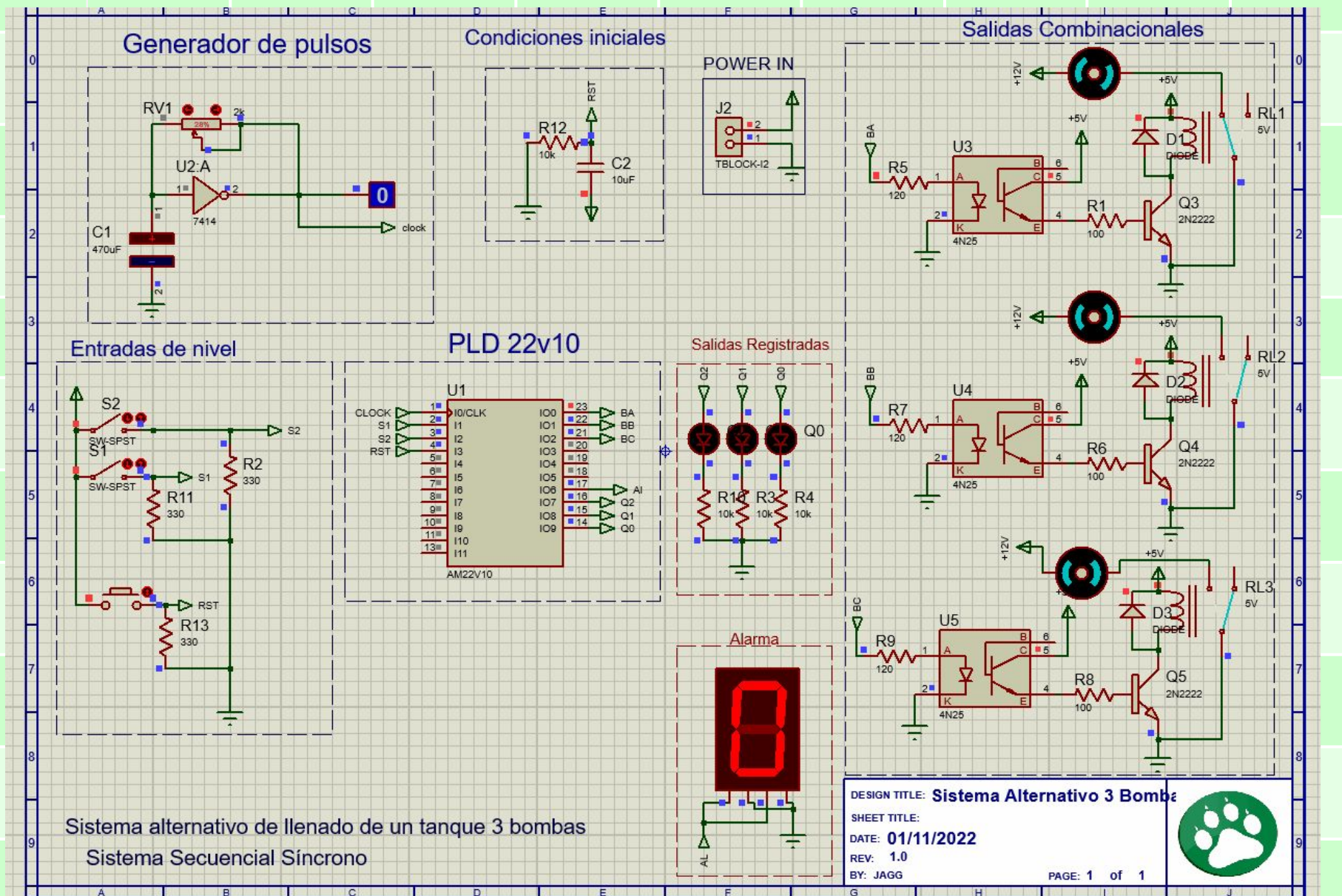
BA=0; BB=0; BC=0; AL=1;
GOTO E7;

STATE E7:

BA=0; BB=0; BC=0; AL=0;
GOTO E6;

END





JAGG

Repaso

- Con que se quedan de esta sesión
- Dos entradas
- State_digram
- Truth_table :>
- Qué hacer con los estados no utilizados (Autoreset)
- Teorema del Dr. Elizondo
- Flip Flop retroalimentado una OR
- Cuantos personajes de FIME citamos hoy