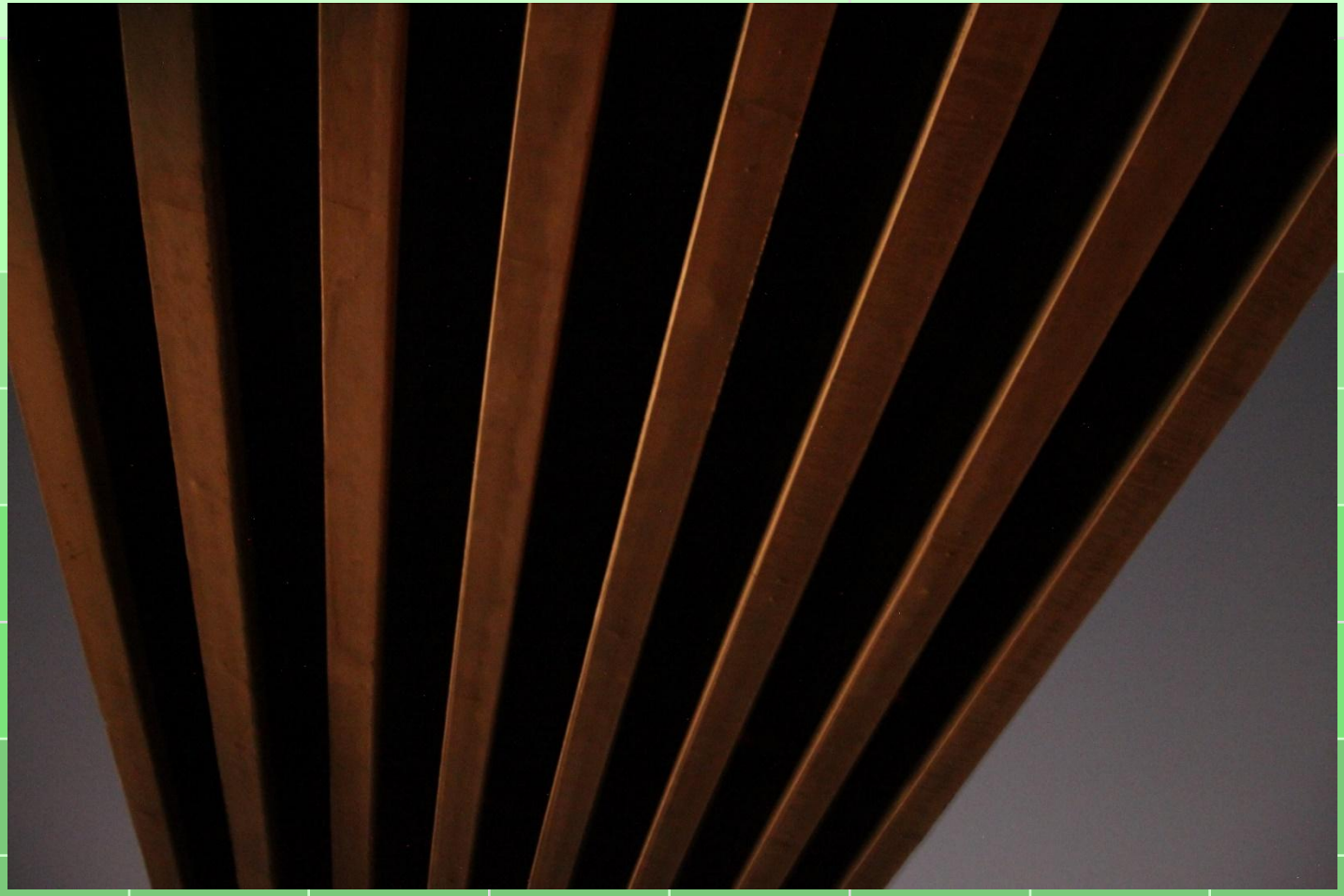


Las cosas no se dicen, se hacen, porque al hacerlas se dicen solas.

Woody Allen

Producto Integrador de Aprendizaje (PIA) 40 puntos

1. Sube los archivos solicitados como entregables a la plataforma de Google Classroom antes de la fecha límite.
2. Envía un mensaje al profesor para notificarle que has subido los archivos.
3. Espera a que el profesor apruebe tu entrega (VoBo).
4. Una vez aprobada, acuerda con el profesor la fecha y hora para llevar a cabo la entrevista.

Durante la entrevista, asegúrate de traer tu computadora con las evidencias del PIA con la que presentaras los resultados obtenidos en forma Oral y escrita.

Actividades y proyectos en proceso

	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	
PF7	Diseño Combinacional con HDL	F	
PF8	Multiplexor	5	
AF3	Decodificador con Display	10	
PF9	Flip Flops	F	
PF10	Pulsos de sincronía	5	
AF4	Diseño Secuencial	10	
AF5	PIA	40	Día del examen

1.- Manual: (Biestable)

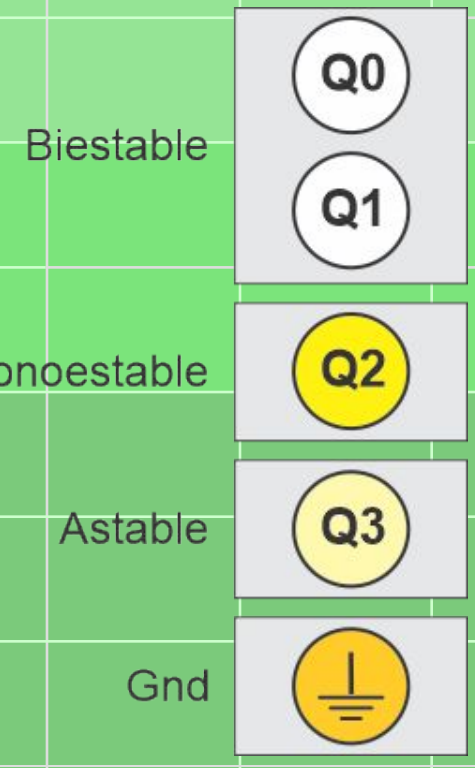
Flip Flop SC (SN74LS00), Eliminador de rebotes

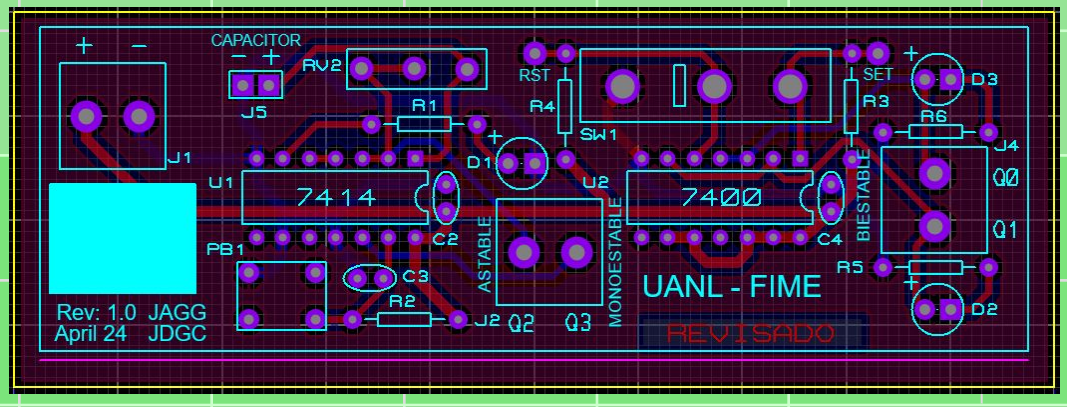
2.- Manual: (Monoestable)

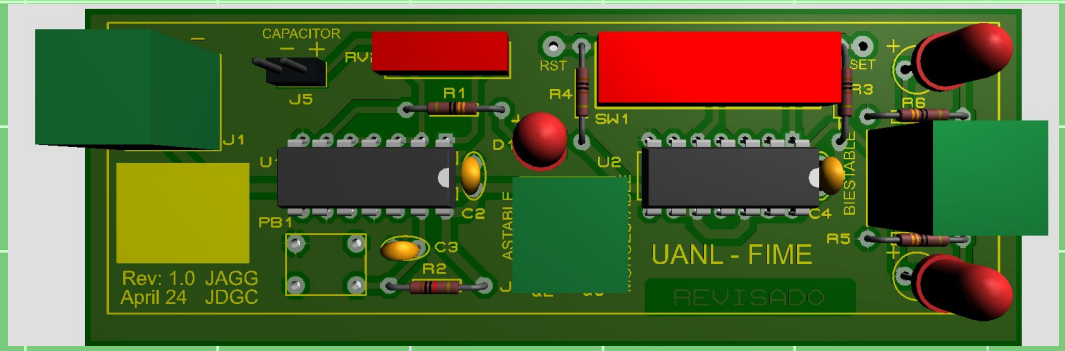
Compuerta Not, Schmitt Trigger (SN74LS14).

3.- Automático: (Astable).

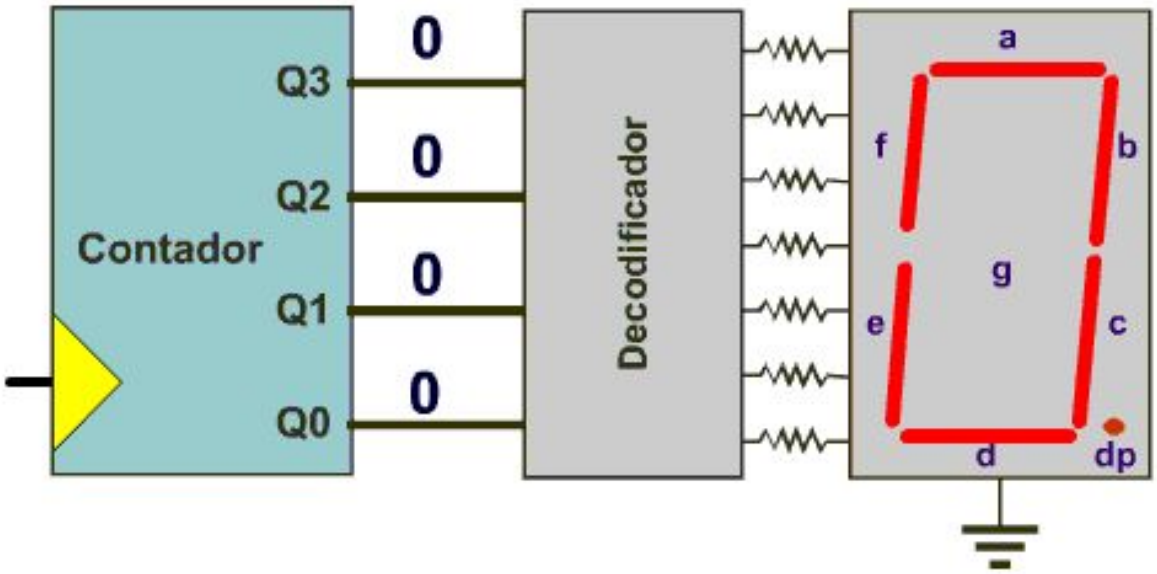
Compuerta Not, Schmitt Trigger retroalimentada (SN7414).







Contadores



Contador (*counter* en inglés)

Es un circuito digital secuencial construido a partir de Flip Flops y compuertas lógicas capaz de:

- Realizar el cómputo (conteo) de los pulsos que recibe en la entrada de sincronía (Clk).
- Actuar como divisor de frecuencia.

Habitualmente, el conteo se realiza en un código binario, que con frecuencia será el binario natural o el BCD natural (contador de décadas).

Los contadores se pueden clasificar en:

- **Binarios (n bit´s)**
- **Décadas (0 a 9)**
- **Especiales**

Pueden ser ascendentes (UP) y/o descendentes (DOWN)

Contadores Binarios

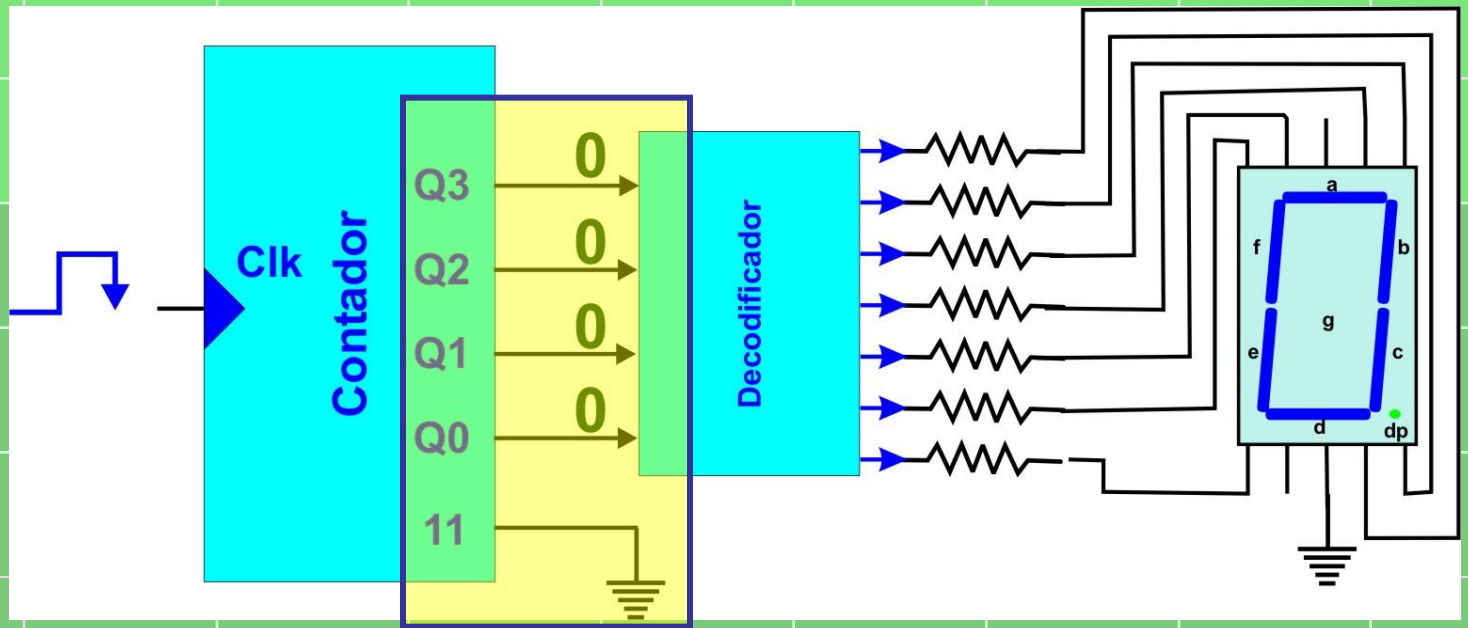
- **Los contadores binarios pueden ser clasificados por:**
  - **1.- El número de bit's.**
  - **2.- El número máximo de estados (Módulo del contador)**

Bit's	Conteo Ascendente	Conteo Descendente
1	0 a 1	1 a 0
2	0 a 3	3 a 0
3	0 a7	7 a 0
4	0 a 15	15 a 0
5	0 a 31	31 a 0
6	0 a 63	63 a 0
7	0 a 127	127 a 0
8	0 a 255	255 a 0
9	0 a 511	511 a 0
10	0 a 1023	1023 a 0

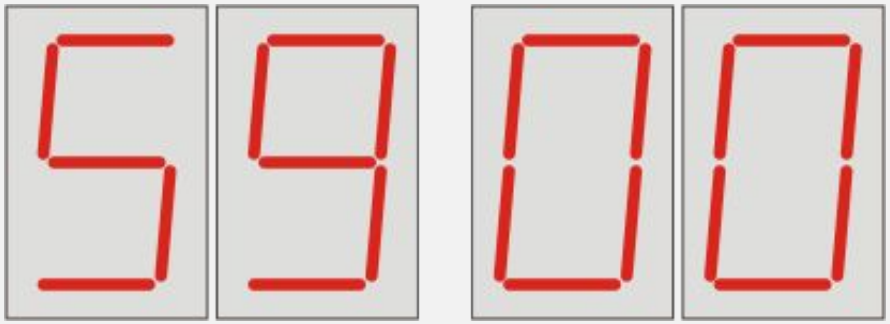
Contadores Decimales

Los contadores decimales o también llamados de décadas o en BCD son de cuatro bit's, necesarios para contar de:

- **0 a 9 ascendente**
- **9 a 0 descendente**

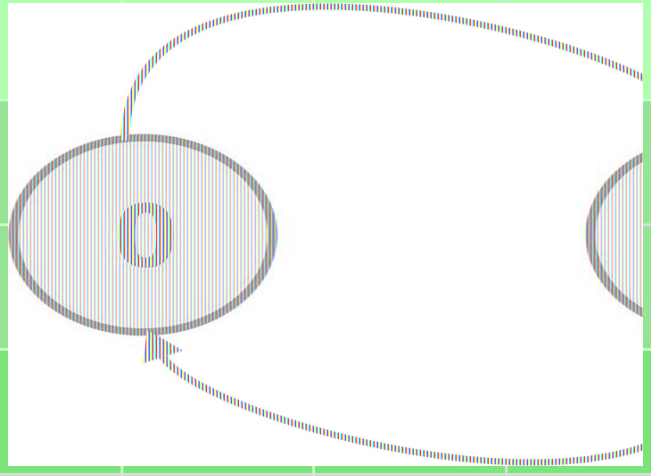


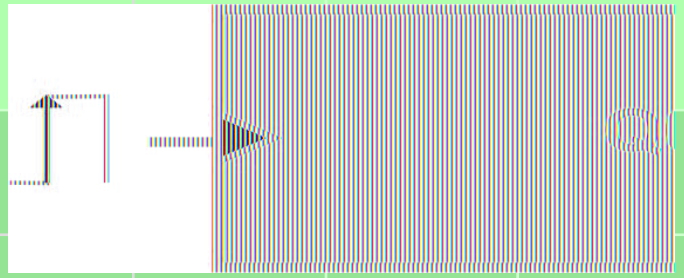
Contadores Especiales



- **Un contador especial puede ser un contador de un minutero o segundero de un reloj digital o marcador deportivo de 0 a 59 en forma ascendente y/o de 59 a 0 en forma descendente.**

Contador Binario de un Bit





Tres opciones de diseño por medio de ABEL-HDL

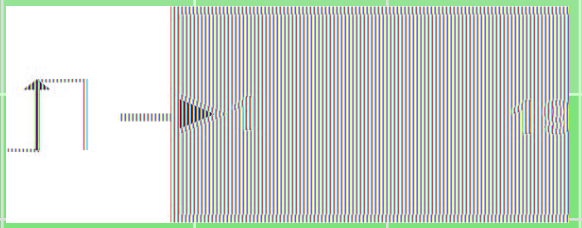
1.- State_Diagram

2.- Truth_table (Q0:>Q0)

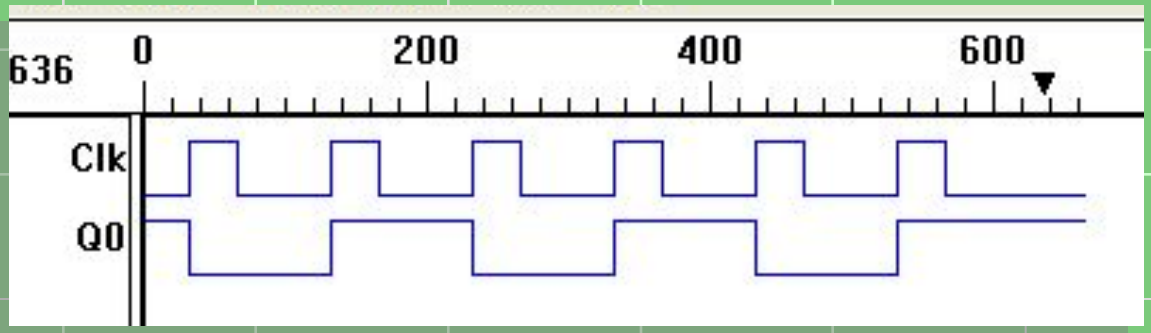
3.- Equations S:=(S+1)

Contador Binario de un Bit
state_diagram

```
MODULE cbub
"Entrada
Clk pin 1;
"salida Registrada
Q0 pin 19 istype 'reg';
"sincronización
S=Q0;
equations
S.Clk=Clk;
Declarations
E0=0;
E1=1;
State_diagram S
State E0:
goto E1;
State E1:
Goto E0;
```



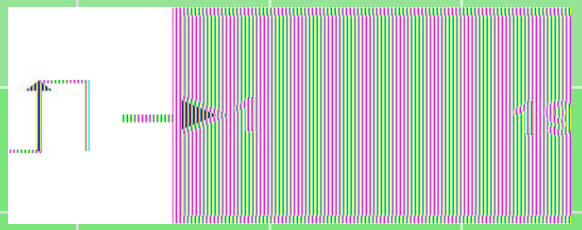
```
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
END
```



```
MODULE cbub
"Entrada
Clk pin 1;
"salida Registrada
Q0 pin 19 istype 'reg';
"sincronización
S=Q0;
equations
S.Clk=Clk;
Truth_table
(Q0:>Q0)
0:>1;
1:>0;
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
```

Contador Binario de un Bit

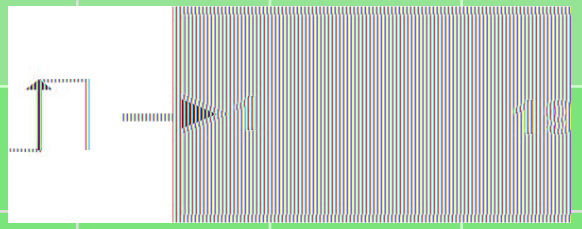
Truth_table (Q0:>Q0)



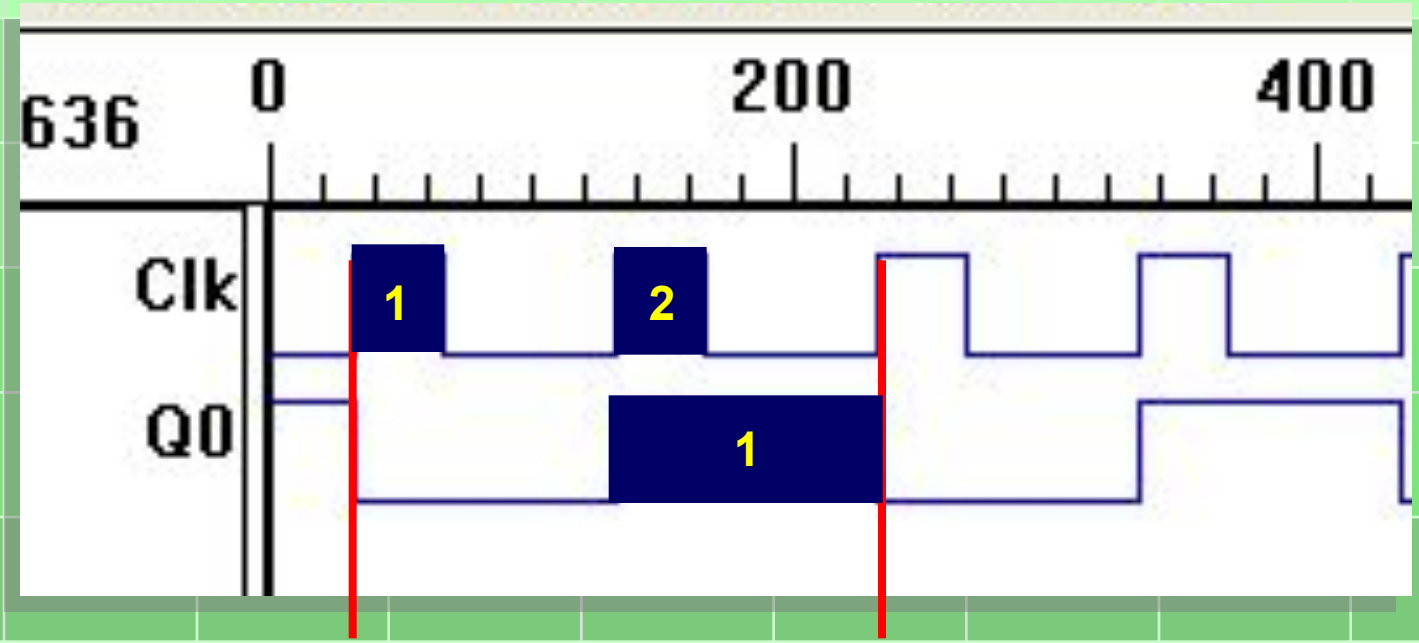
Contador Binario de un Bit

Equations S:=(S+1)

```
MODULE cbub
"Entrada
Clk pin 1;
"salida Registrada
Q0 pin 19 istype 'reg';
"sincronización
S=Q0;
equations
S.Clk=Clk;
S:=(S+1)
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
```

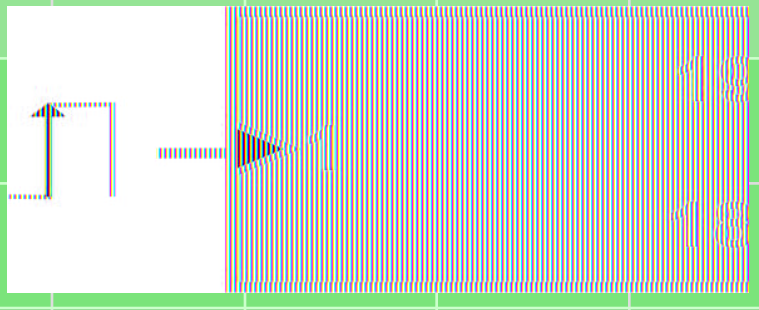


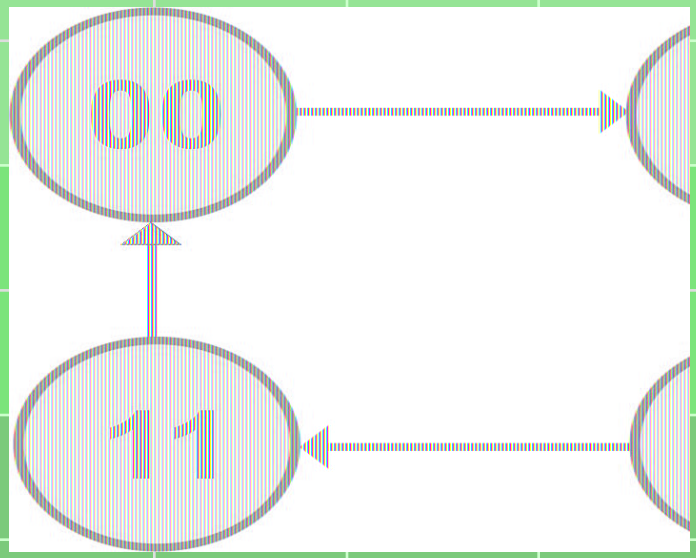
Contador Binario de un Bit



Divisor de frecuencia

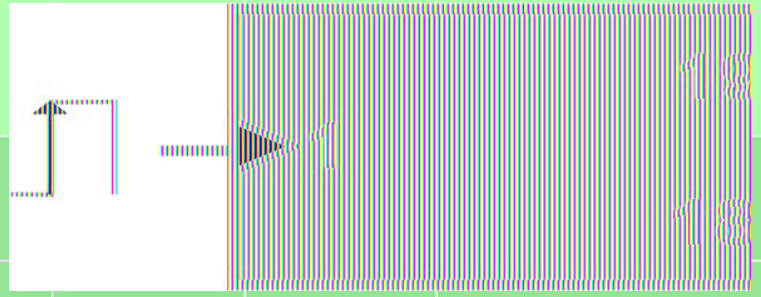
Contador Binario de dos Bit's ascendente





Contador Binario de dos Bit's State_diagram

```
MODULE cbub
"Entrada
Clk pin 1;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
Declarations
E0=[0,0];
E1=[0,1];
E2=[1,0];
E3=[1,1];
```



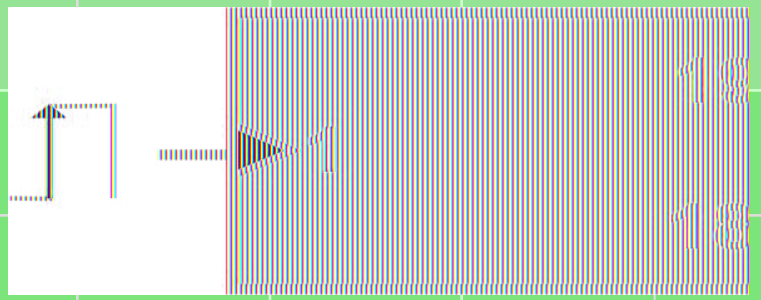
```
State_diagram S
State E0:
goto E1;
State E1:
Goto E2;
State E2:
Goto E3;
State E3:
Goto E0;
```

```
MODULE cbubas
"Entrada
Clk pin 1;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
```

```
Truth_Table
([Q1,Q0] :>[Q1,Q0])
[0,0]:>[0,1];
[0,1]:>[1,0];
[1,0]:>[1,1];
[1,1]:>[0,0];
```

Contador Binario de dos Bit's ascendente

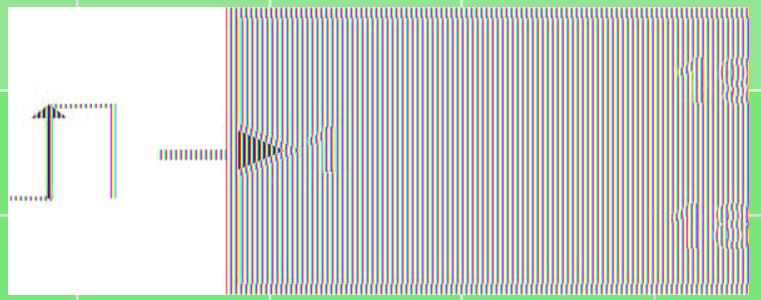
Truth_Table
([Q1,Q0] :>[Q1,Q0])



```
MODULE cbubas
"Entrada
Clk pin 1;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
S:=(S+1);
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
END
```

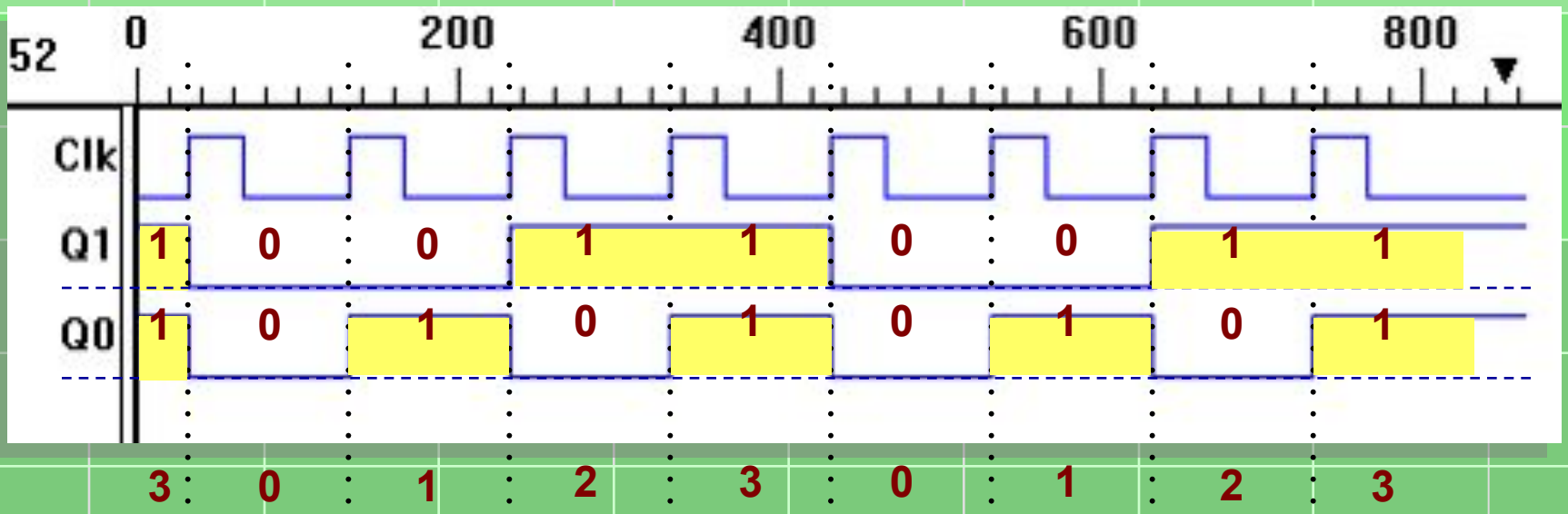
Contador Binario de dos Bit's ascendente

equations S:=(S+1);

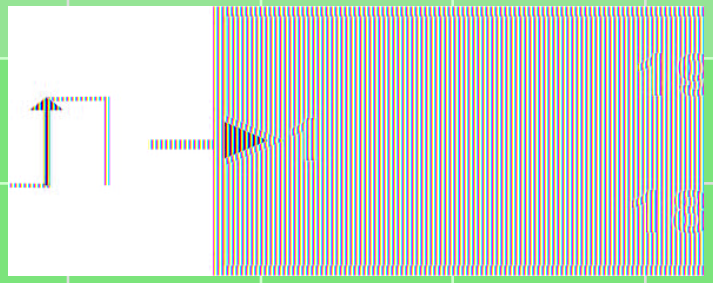


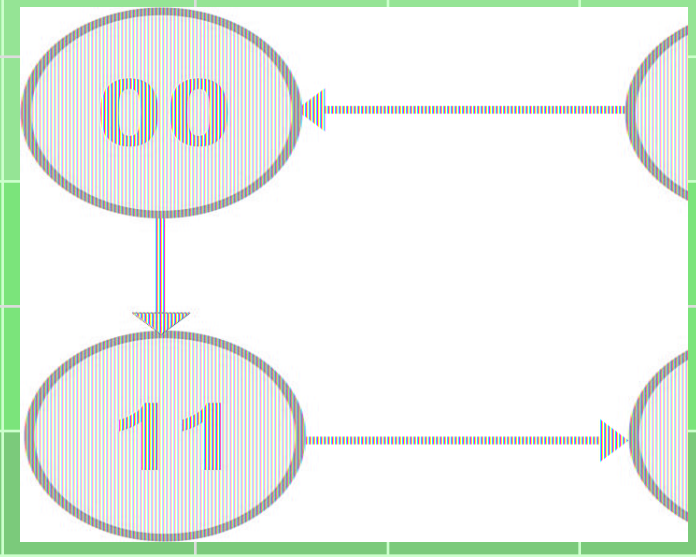
```
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
END
```

Contador Binario de dos Bit's ascendente



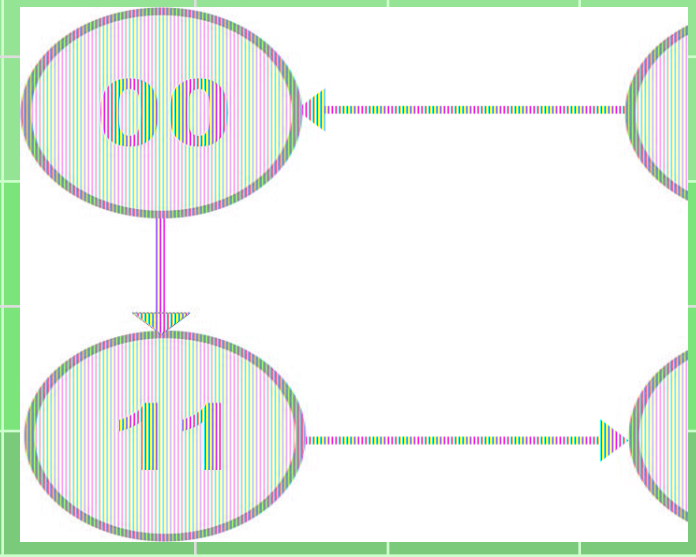
Contador Binario de dos Bit's descendente





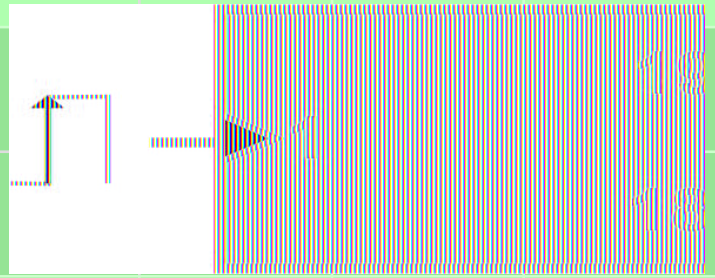
Contador Binario de dos Bit's descendente

```
State_diagram S
State E0:
goto E3;
State E1:
Goto E0;
State E2:
Goto E1;
State E3:
Goto E2;
```



Contador Binario de dos Bit's descendente

```
MODULE cbubas
"Entrada
Clk pin 1;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
```



Truth_Table

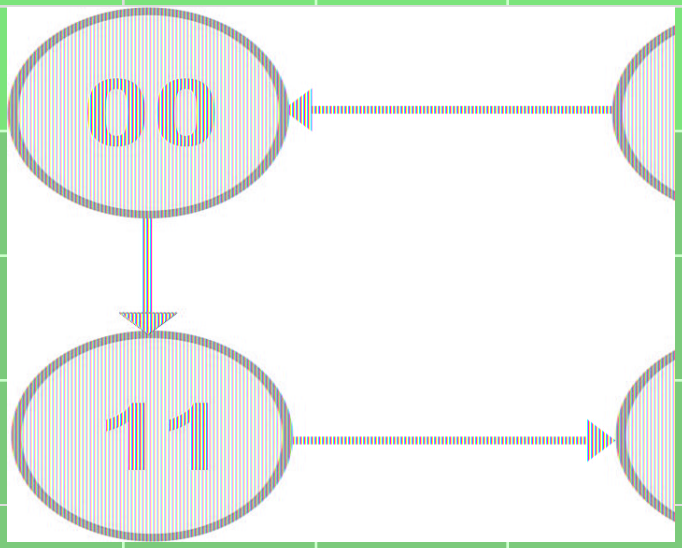
([Q1,Q0] :>[Q1,Q0])

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

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

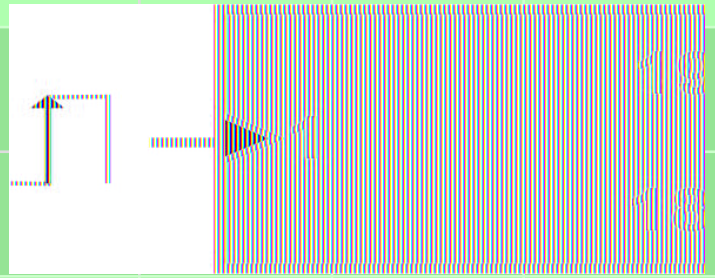
[1,0]:>[0,1];

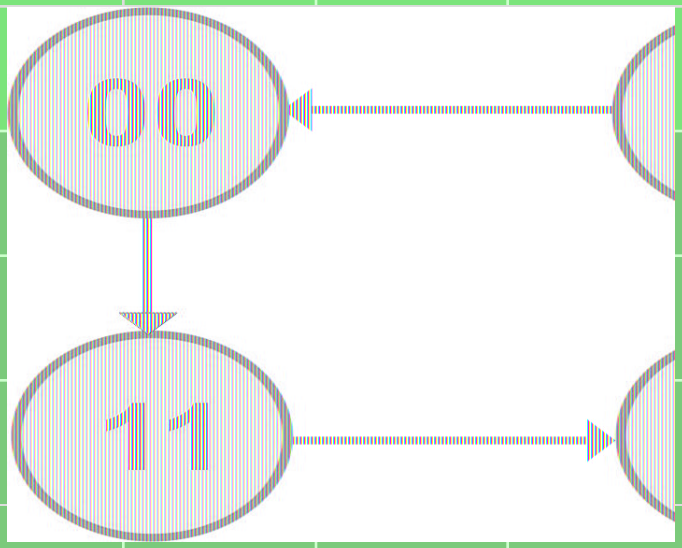
[1,1]:>[1,0];



Contador Binario de dos Bit's descendente

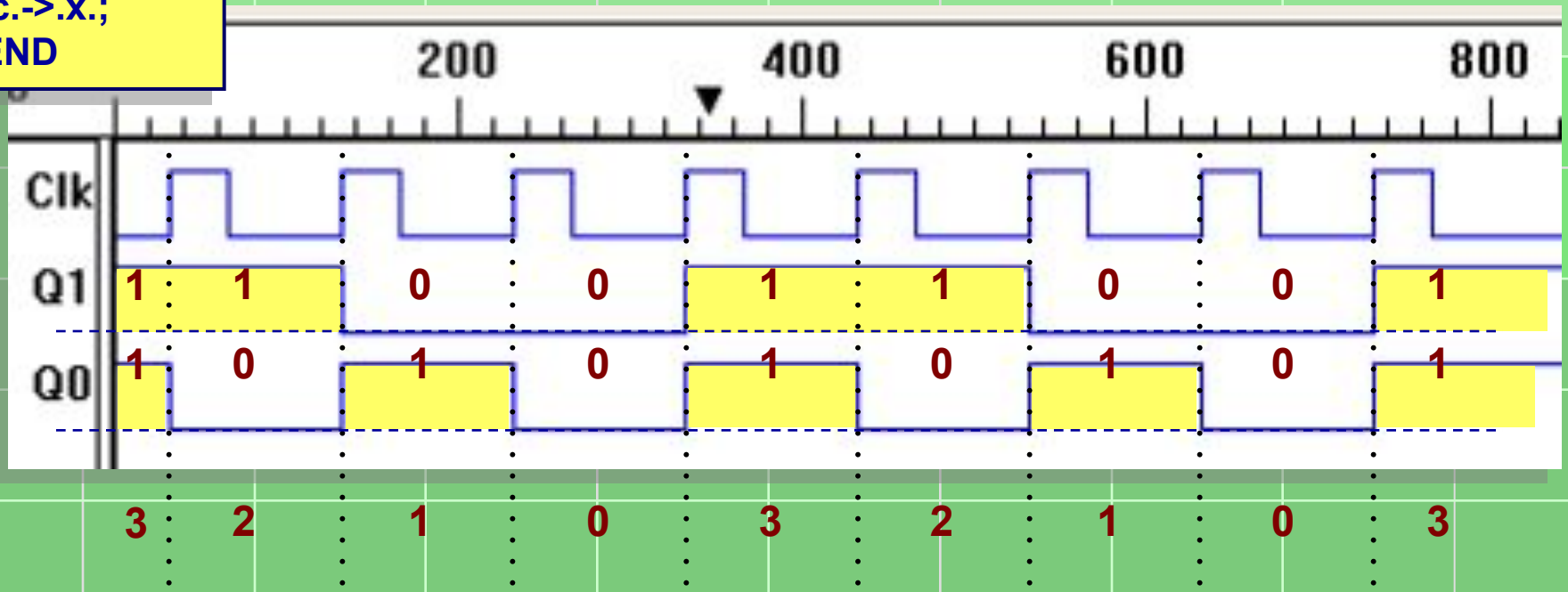
```
MODULE cbubas
"Entrada
Clk pin 1;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
S:=(S-1);
```





```
test_vectors
(Clk->Q0)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
END
```

Contador Binario de dos Bit's descendente

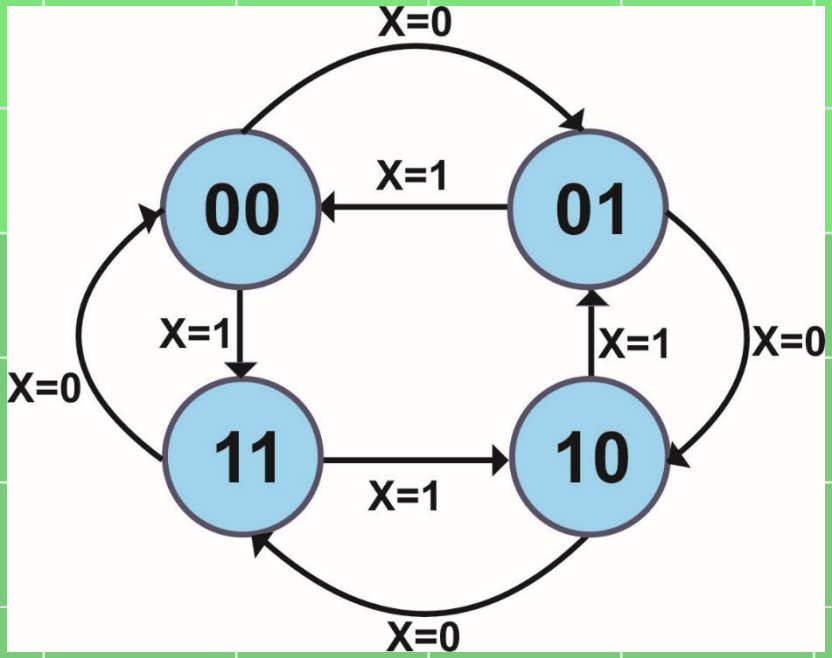


Contador binario de dos bit's

Que contenga una entrada X de modo que:

a) Si X=0 se ascendente

b) Si X=1 sea descendente



State_diagram S

State E0:

If X Then E3 else E1;

State E1:

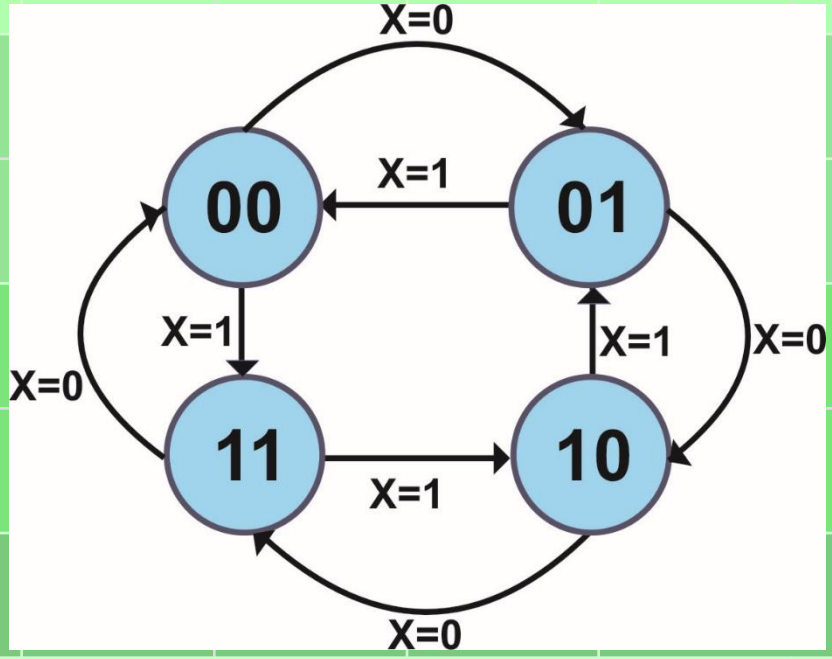
if X then E0 else E2;

State E2:

if X then E1 else E3;

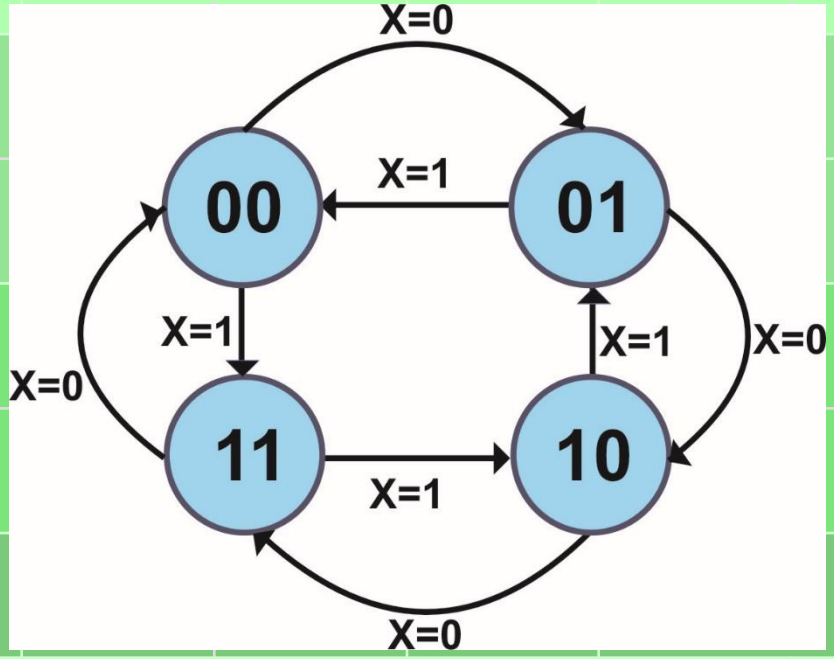
State E3:

if X then E2 else E0;



Truth_table
([X,Q1,Q0] :>[Q1,Q0])

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



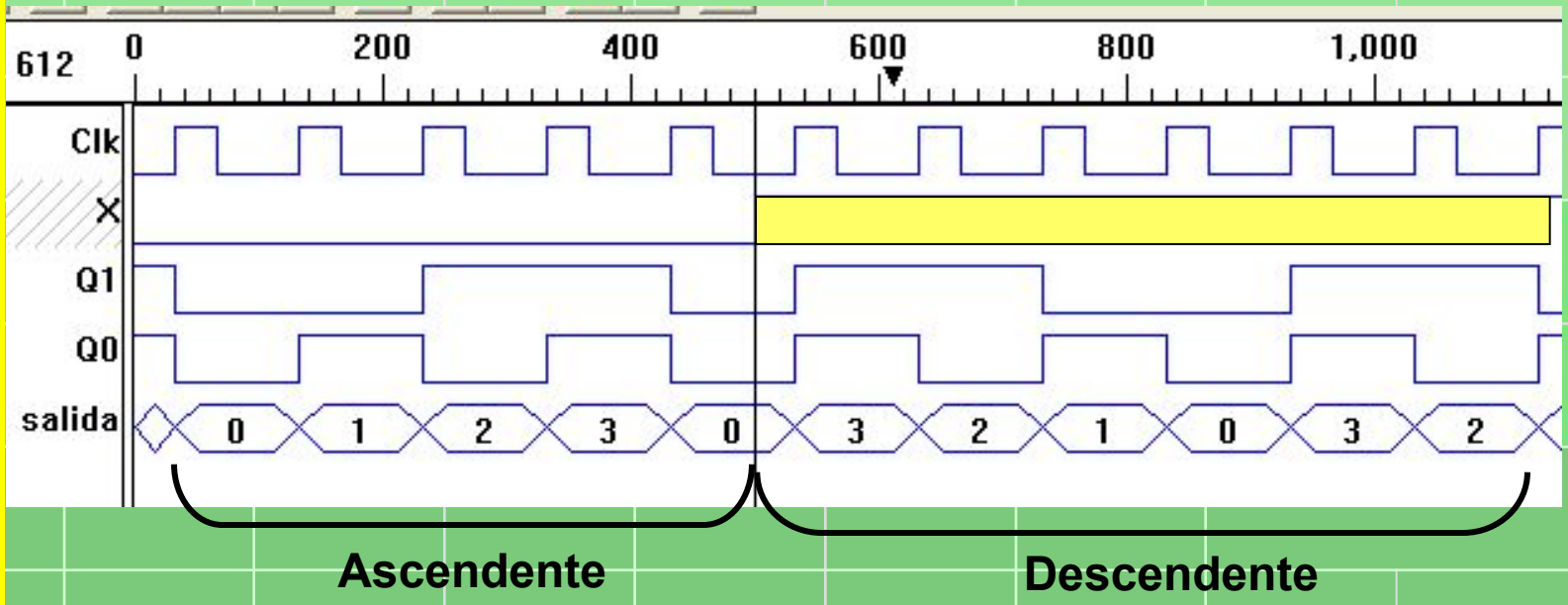
Contador de 2 Bits A/D

```
MODULE contbad
"Entrada
Clk,X pin 1,2;
"salida Registrada
Q1,Q0 pin 19,18 istype 'reg';
"sincronización
S=[Q1,Q0];
equations
S.Clk=Clk;
when !X then S:=(S+1);
when X then S:=(S-1);
```

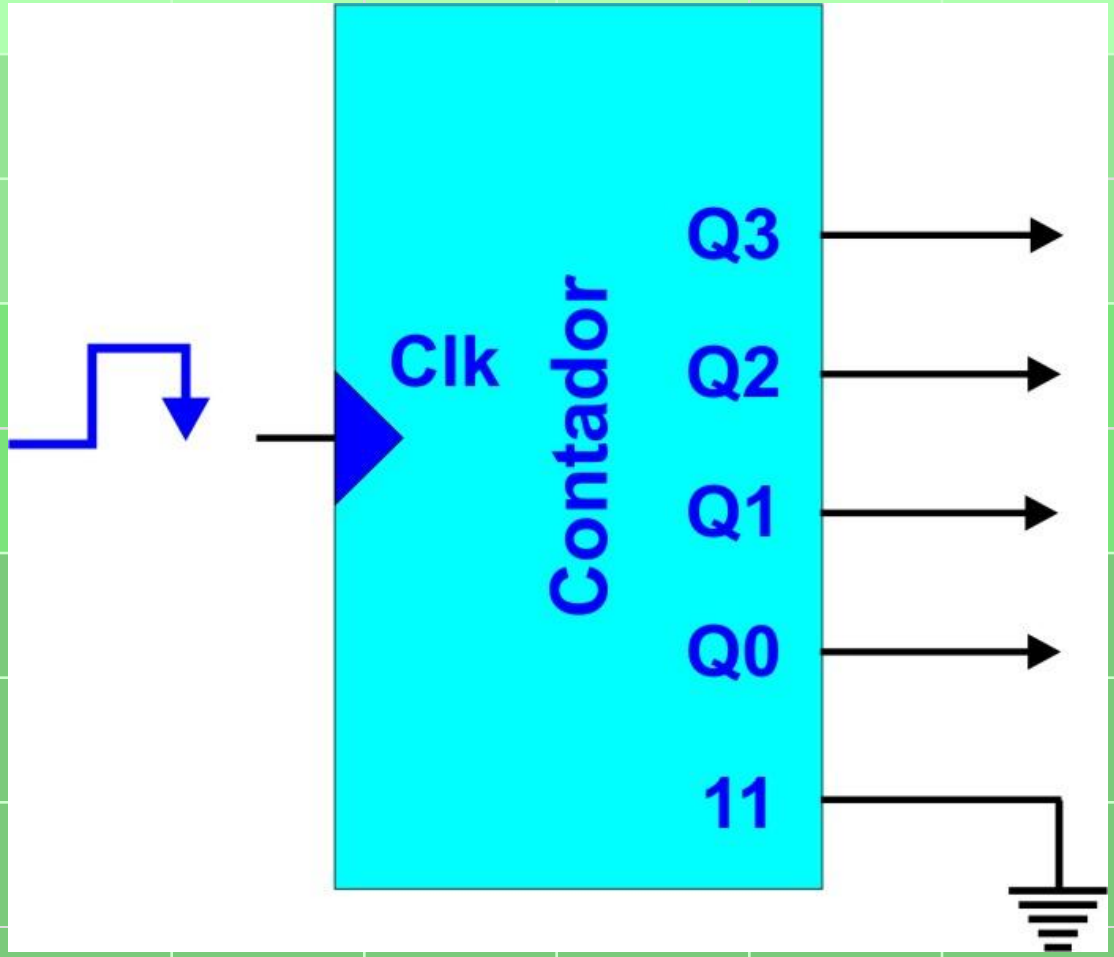
```
test_vectors
([Clk,X]->Q1)
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
END
```

Simulación

```
test_vectors
([Clk,X]->Q0)
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,0]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,1]->.x.;
[.c.,0]->.x.;
```

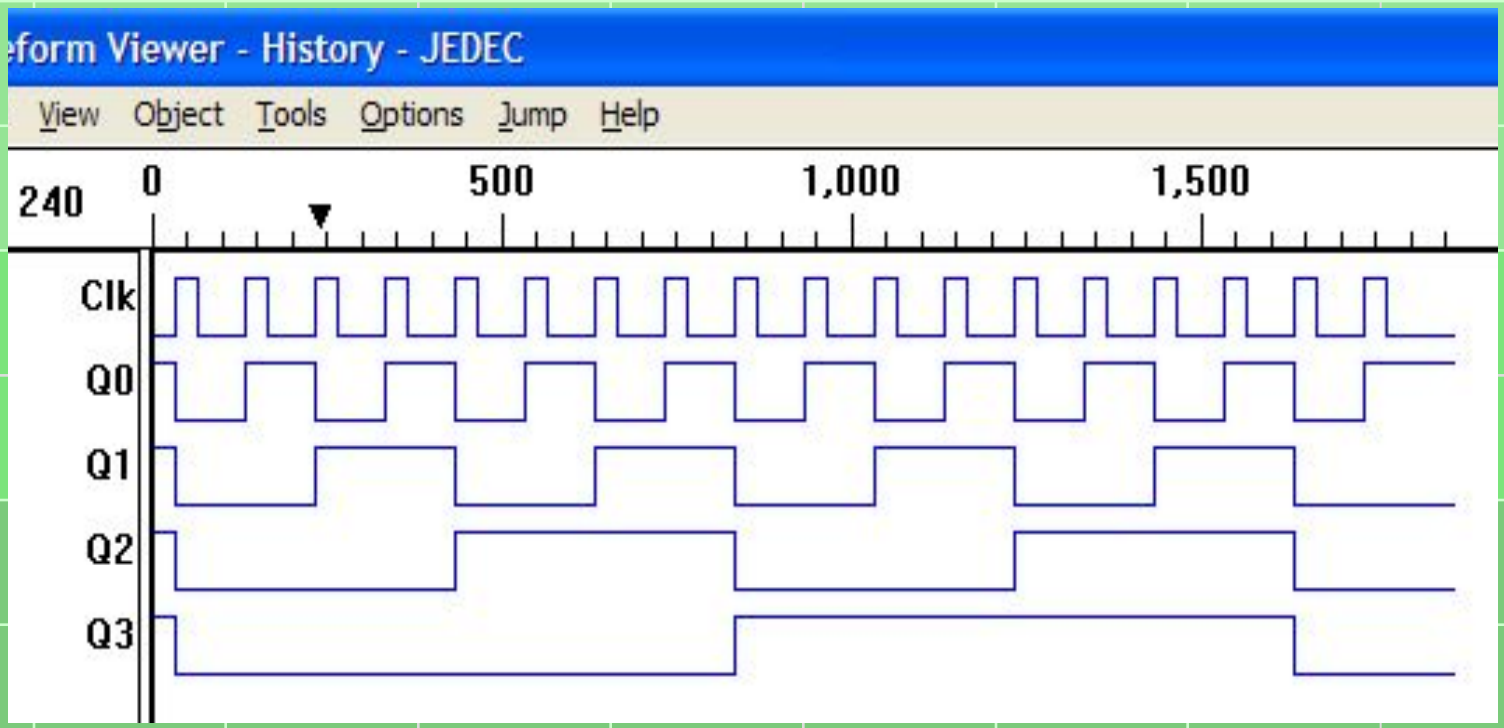


Contador Binario ascendente de 0 a 15

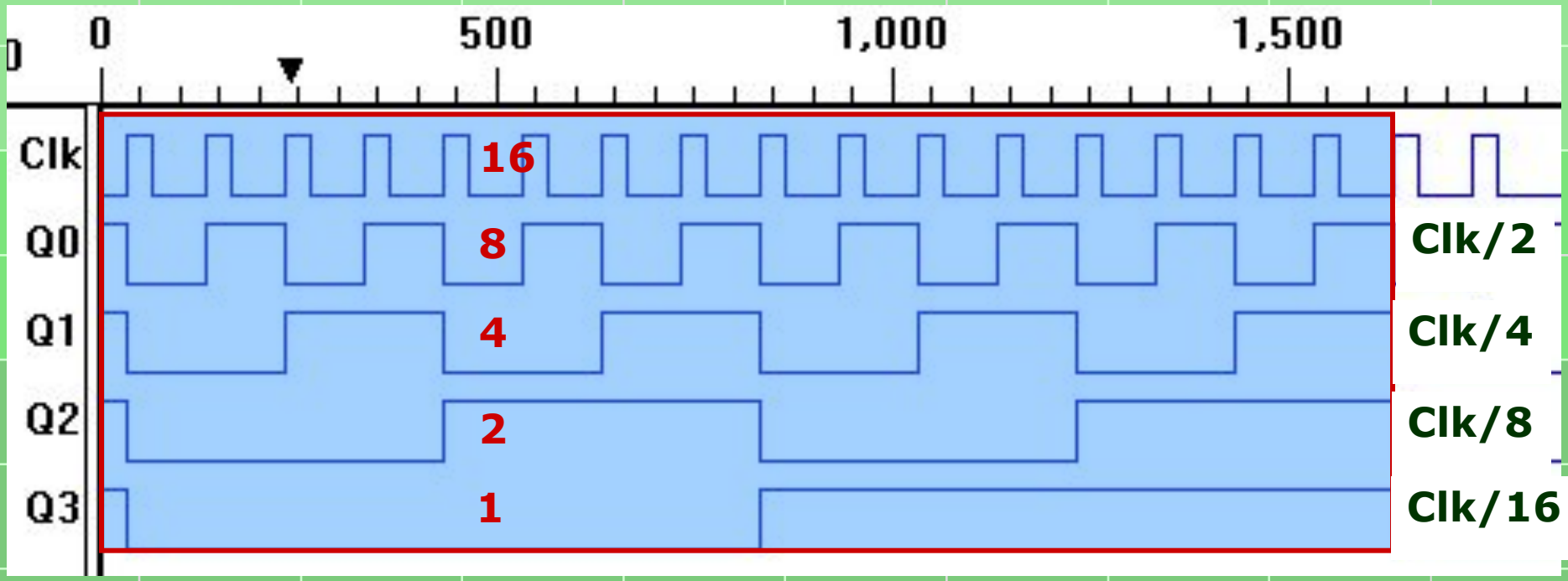


```
MODULE conta
"Constantes la señal de reloj en la simulación .c. se sustituye por C
" La salida .x. se sustituye por solo x
        C,x = .c.,.x.;
"Entrada de reloj
        Clk pin 1;
"Salidas de los Flip Flops
        Q3..Q0 pin 19..16 istype 'reg,buffer';
Declarations

        Cuenta = [Q3..Q0];

Equations

Cuenta.Clk=Clk;

Cuenta:= (Cuenta + 1);
```

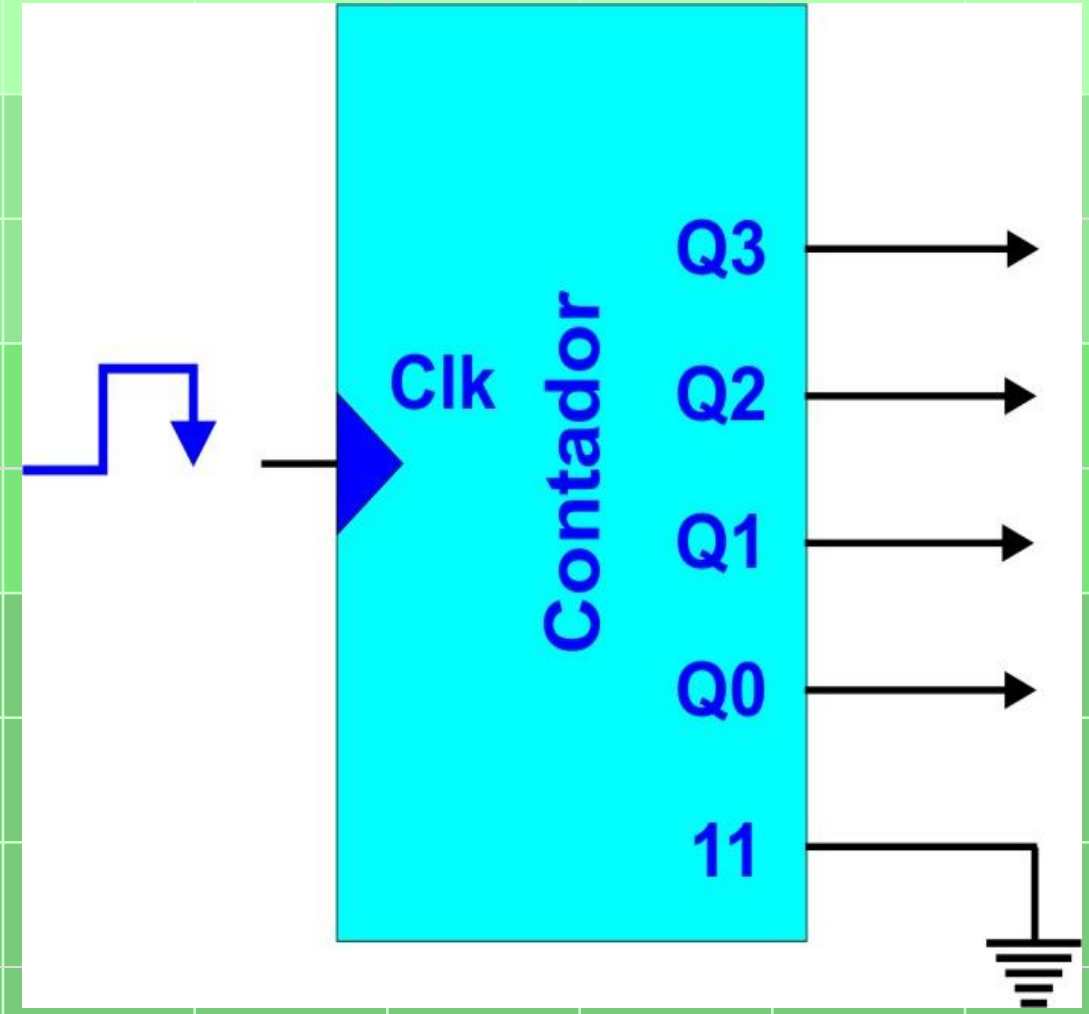
- Test_Vectors
- ([Clk] -> Cuenta)
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- END



Divisor de frecuencias

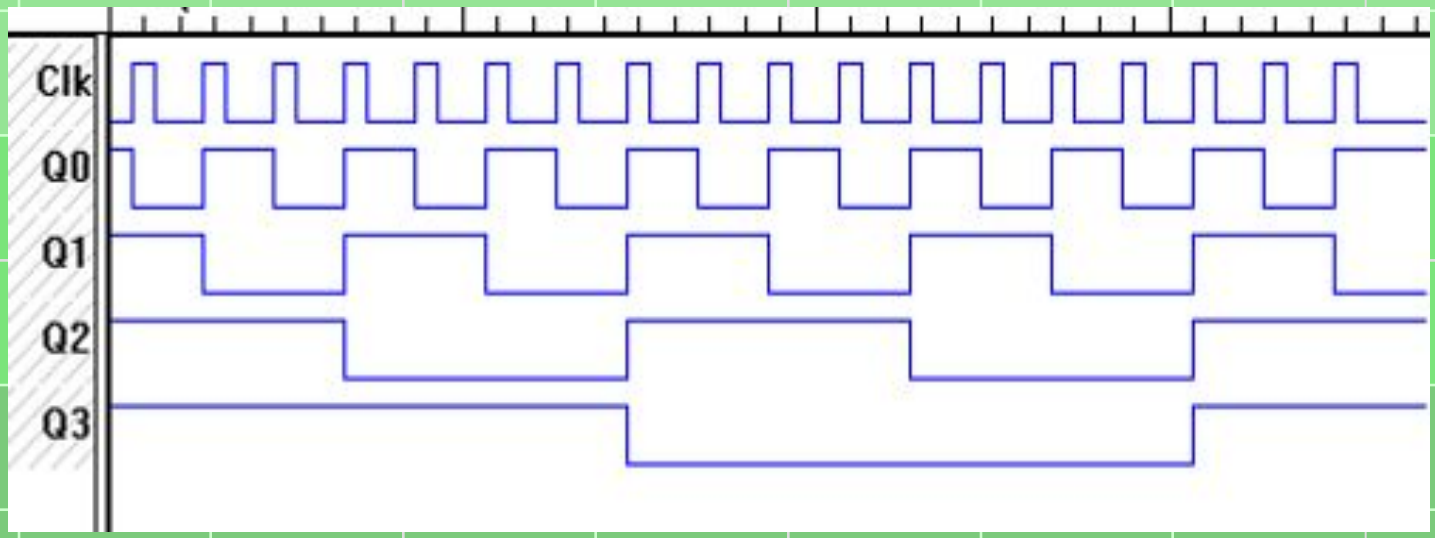


Contador Binario descendente de 0 a 15



```
MODULE conta
"Constantes la señal de reloj en la simulación .c. se sustituye por C
" La salida .x. se sustituye por solo x
        C,x = .c.,.x.;
"Entrada de reloj
        Clk pin 1;
"Salidas de los Flip Flops
        Q3..Q0 pin 19..16 istype 'reg';
Declarations
Cuenta = [Q3..Q0];
Equations
Cuenta.Clk=Clk;
Cuenta:= (Cuenta - 1);
```

- Test_Vectors
- ([Clk] -> Cuenta)
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- [C] -> x ;
- END



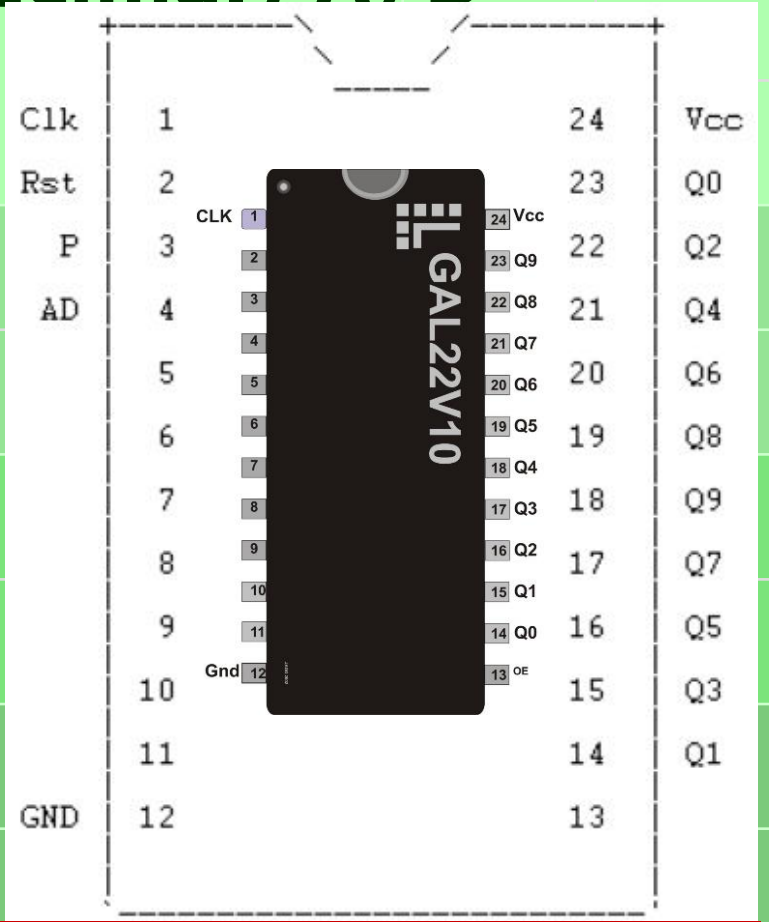
Contador Binario

Diseñe un contador binario de diez bits que contenga cuatro entradas de control:

1) Clk que sean los pulsos de sincronia para la velocidad de conteo
2) UD de modo que con el valor de UD=0 el conteo sea en forma ascendente y con UD=1 el conteo sea en forma descendente.
3) P (Paro) de modo que al oprimirlo el conteo se detenga sin importar la entrada UD y al soltarlo continúe la cuenta partiendo del valor actual.
4) Rst (Reset asíncrono) de modo que al activarla el contador se ira a ceros sin importar el que este presente el pulso de reloj o los valores de las otras entradas.

Cuatro bits (Hexadecimal) A/D

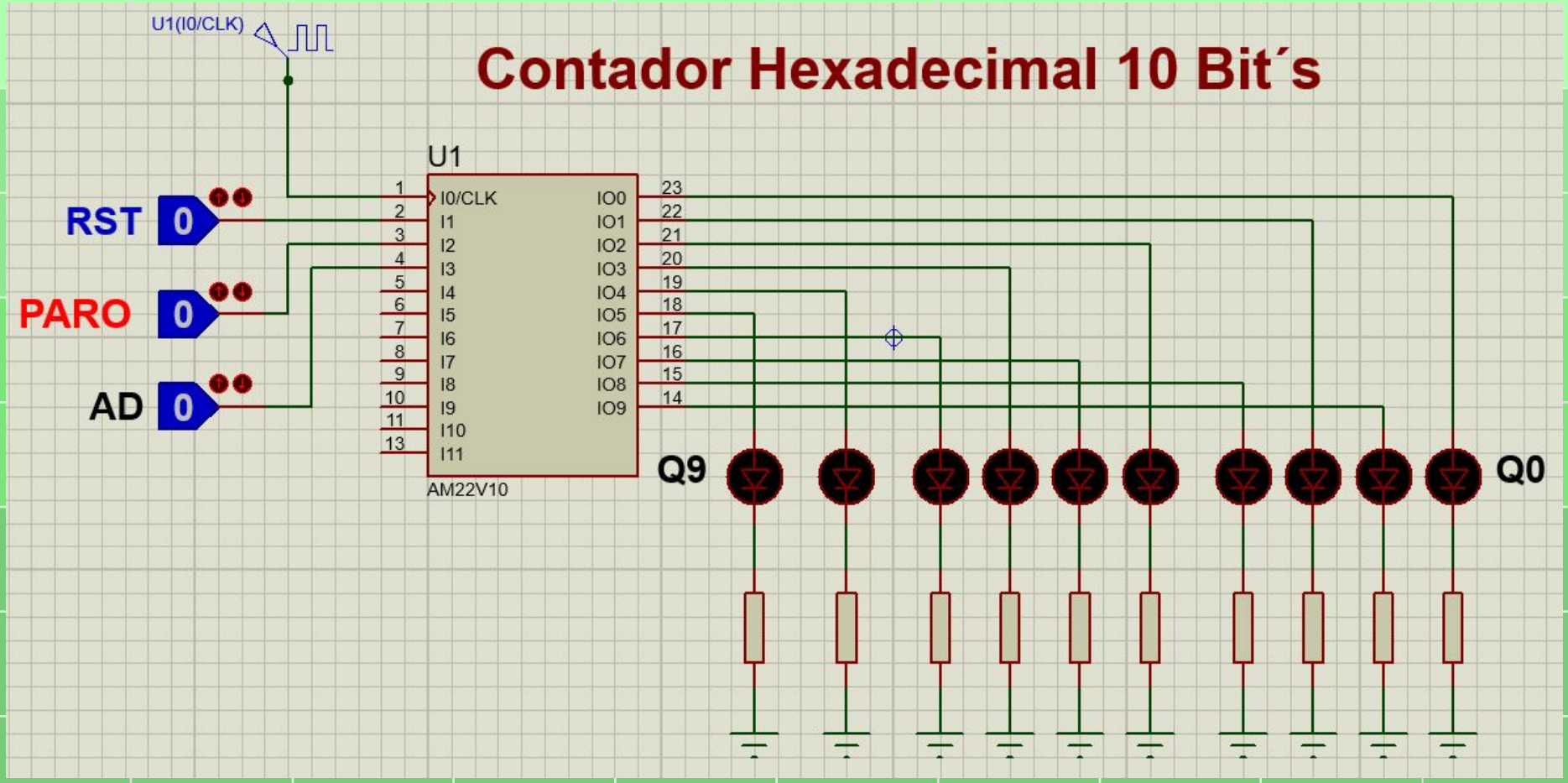
```
MODULE bin
"Entradas
Clk,AD,P,Rst pin 1..4;
"salidas Registradas
Q9..Q0 pin 23..14 istype 'reg';
"sincronización
S=[Q9..Q0];
equations
S.Clk=Clk;
S.ar=Rst;
when !P&!AD then S:=(S+1);
when !P&AD then S:=(S-1);
when P then S:=S;
END
```



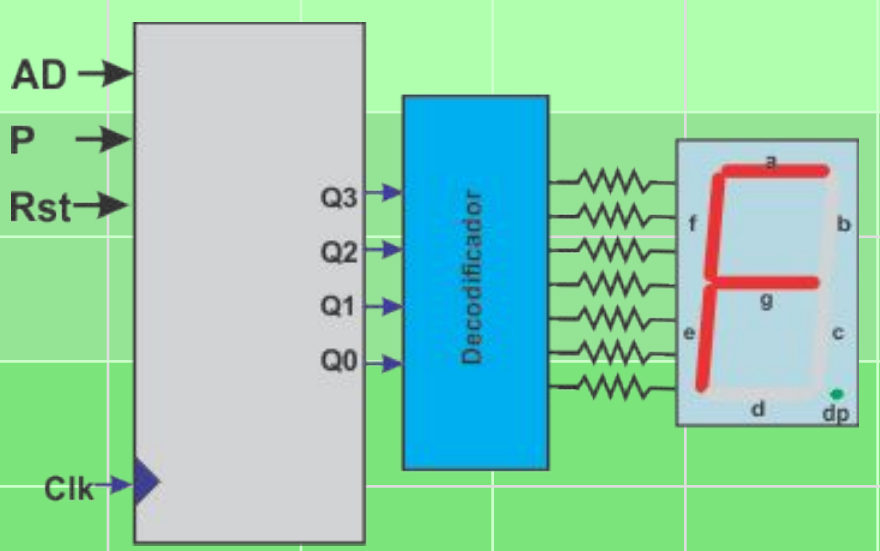
!

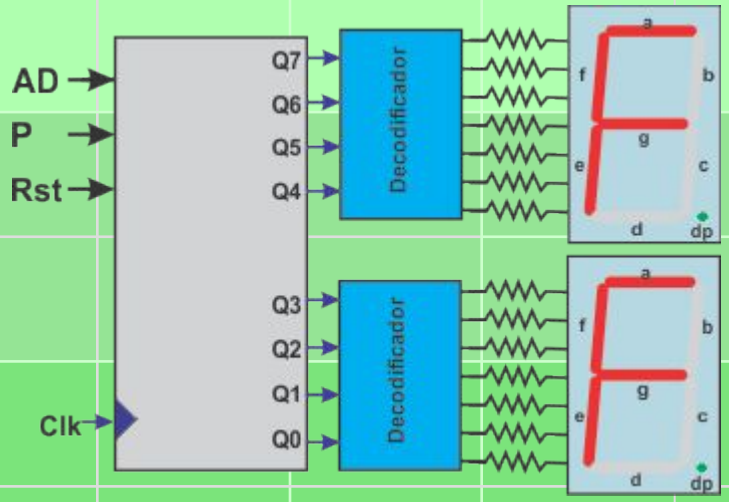
La asignación de terminales no corresponde con la propuesta

Simulación en PROTEUS



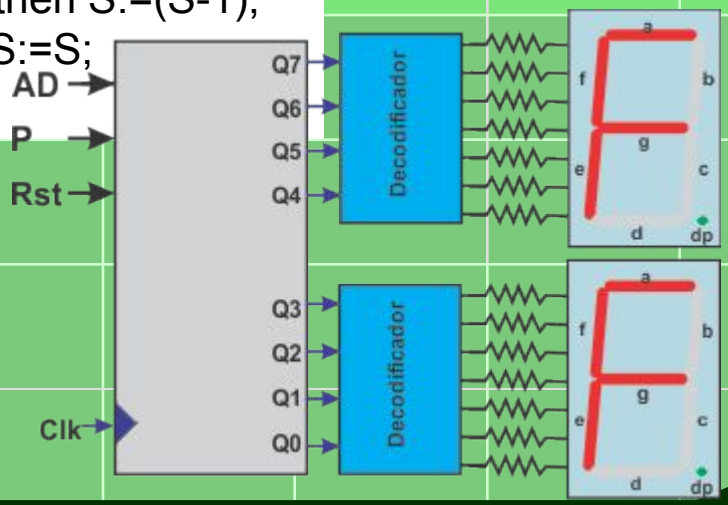
Contador Hexadecimal con Display





Contador

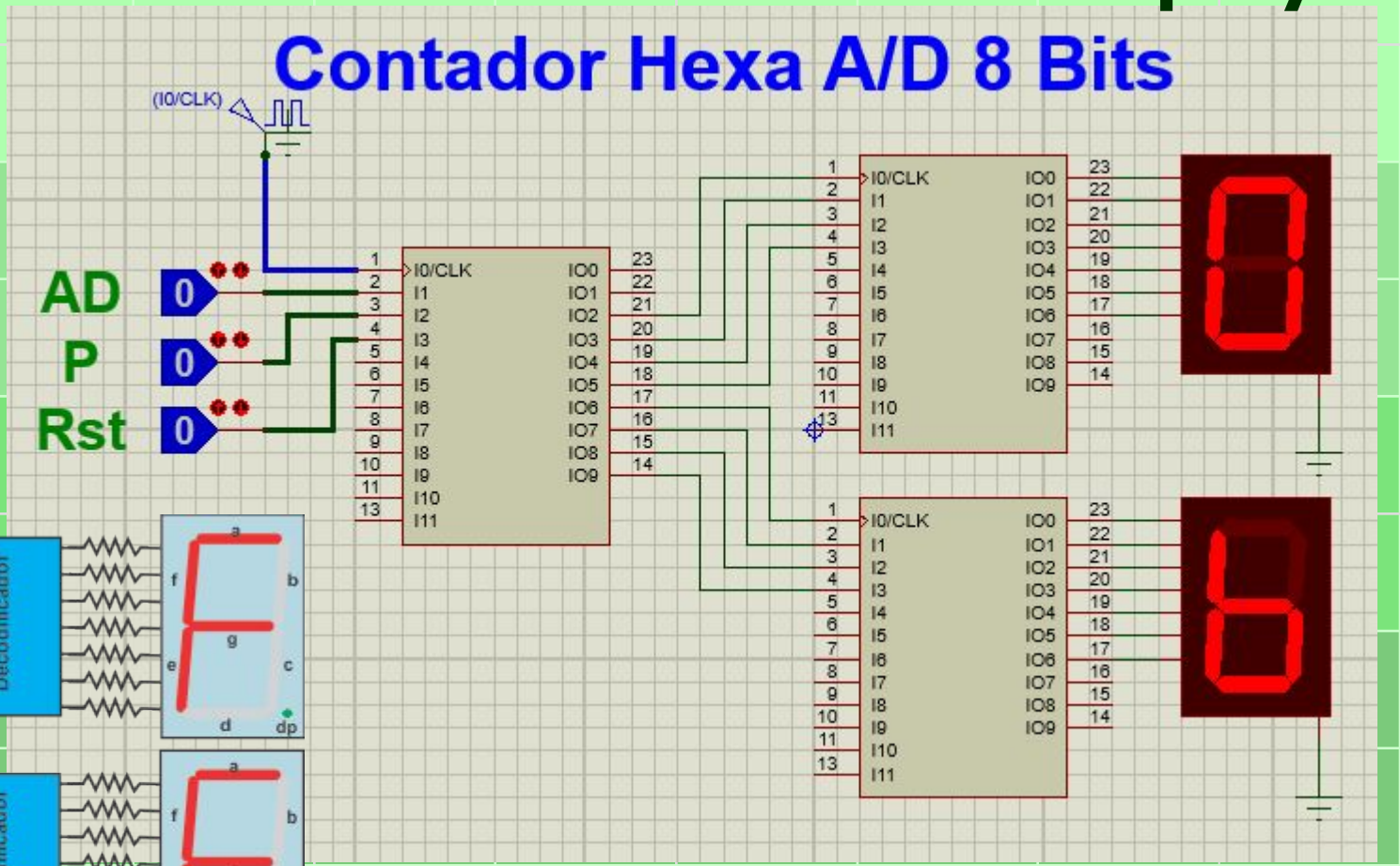
```
MODULE hexaocho
"Entrada
Clk,AD,P,Rst pin 1..4;
"salidas Registradas
Q7..Q0 pin 21..14 istype 'reg';
"sincronización
S=[Q7..Q0];
equations
S.Clk=Clk;
S.ar=Rst;
when !P&!AD then S:=(S+1);
when !P&AD then S:=(S-1);
when P then S:=S;
END
```



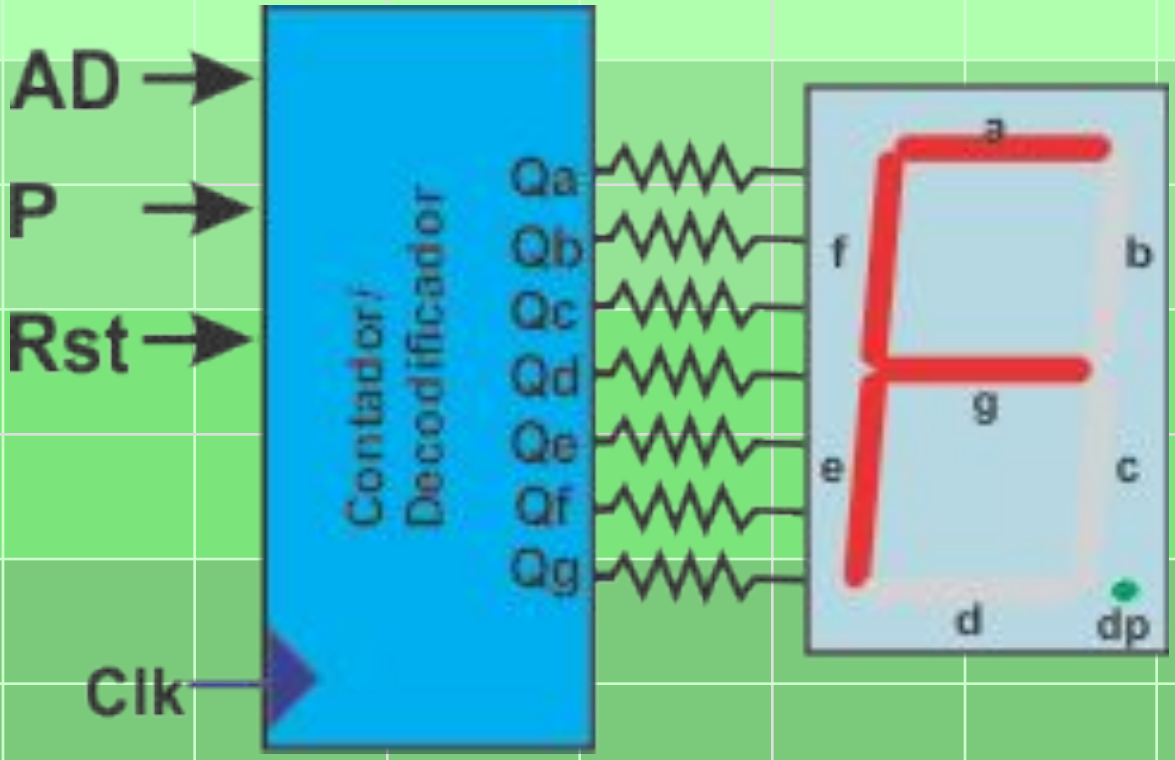
Decodificador Hexa a 7 segmentos

```
MODULE decohex
A,B,C,D pin 1..4;
a,b,c,d,e,f,g pin 23..17 istype 'com';
E=[A,B,C,D];

truth_table
(E->[a,b,c,d,e,f,g])
0->[1,1,1,1,1,1,0];
1->[0,1,1,0,0,0,0];
2->[1,1,0,1,1,0,1];
3->[1,1,1,1,0,0,1];
4->[0,1,1,0,0,1,1];
5->[1,0,1,1,0,1,1];
6->[1,0,1,1,1,1,1];
7->[1,1,1,0,0,0,0];
8->[1,1,1,1,1,1,1];
9->[1,1,1,0,0,1,1];
10->[1,1,1,0,1,1,1];
11->[0,0,1,1,1,1,1];
12->[1,0,0,1,1,1,0];
13->[0,1,1,1,1,0,1];
14->[1,0,0,1,1,1,1];
15->[1,0,0,0,1,1,1];
END
```

Contador Hexadecimal 8 bits con Display



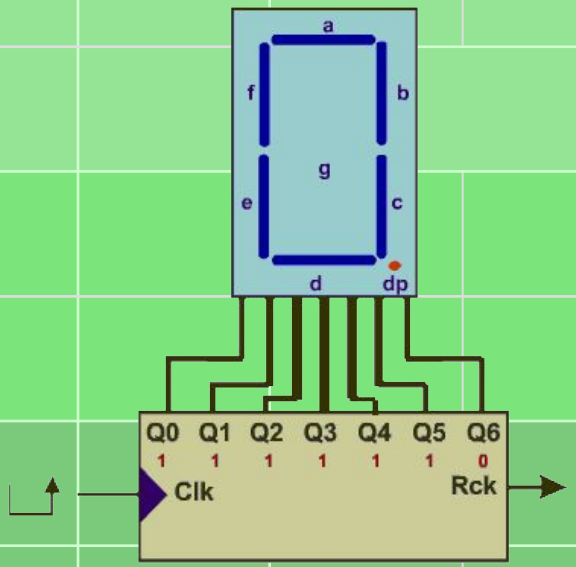
C:\Users\USER\Documents\E2016\ABL\Contadores\Hexa\Hexa8B\sim



Rst síncrono

Contadores que incluyen el decodificador (Hexadecimal)

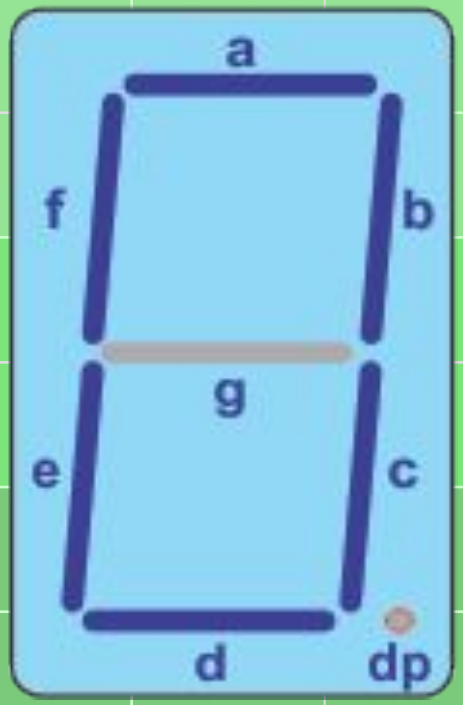
	Q0	Q1	Q2	Q3	Q4	Q5	Q6
	a	b	c	d	e	f	g
E0	1	1	1	1	1	1	0
E1	0	1	1	0	0	0	0
E2	1	1	0	1	1	0	1
E3	1	1	1	1	0	0	1
E4	0	1	1	0	0	1	1
E5	1	0	1	1	0	1	1
E6	1	0	1	1	1	1	1
E7	1	1	1	0	0	0	0
E8	1	1	1	1	1	1	1
E9	1	1	1	0	0	1	1
E10	1	1	1	0	1	1	1
E11	0	0	1	1	1	1	1
E12	1	0	0	1	1	1	0
E13	0	0	1	1	1	1	1
E14	1	0	0	1	1	1	0
E15	1	0	0	0	1	1	1



```
MODULE hexadecI
"Entrada
Clk,AD,P,Rst pin 1..4;
"salidas Registradas
Qa,Qb,Qc,Qd,Qe,Qf,Qg    pin
    21..15 istype 'dc,reg';
"sincronización
S=[Qa,Qb,Qc,Qd,Qe,Qf,Qg];
equations
S.Clk=Clk;
```

```
declarations
E0=[1,1,1,1,1,1,0];
E1=[0,1,1,0,0,0,0];
E2=[1,1,0,1,1,0,1];
E3=[1,1,1,1,0,0,1];
E4=[0,1,1,0,0,1,1];
E5=[1,0,1,1,0,1,1];
E6=[1,0,1,1,1,1,1];
E7=[1,1,1,0,0,0,0];
E8=[1,1,1,1,1,1,1];
E9=[1,1,1,0,0,1,1];
EA=[1,1,1,0,1,1,1];
EB=[0,0,1,1,1,1,1];
EC=[1,0,0,1,1,1,0];
ED=[0,1,1,1,1,0,1];
EE=[1,0,0,1,1,1,1];
EF=[1,0,0,0,1,1,1];
```

E0=[1,1,1,1,1,1,0];



STATE_DIAGRAM S
STATE E0:
IF !Rst&!P&!AD then E1;
IF !Rst&!P&AD then EF;
if !Rst&P then E0;
IF Rst then E0:
STATE E1:
IF !Rst&!P&!AD then E2;
IF !Rst&!P&AD then E0;
if !Rst&P then E1;
if Rst then E0:
STATE E2:
IF !Rst&!P&!AD then E3;
IF !Rst&!P&AD then E1;
if !Rst&P then E2;
if Rst then E0:
if Rst then E0:
STATE E3:
IF !Rst&!P&!AD then E4;
IF !Rst&!P&AD then E2;
if !Rst&P then E3;
if Rst then E0:

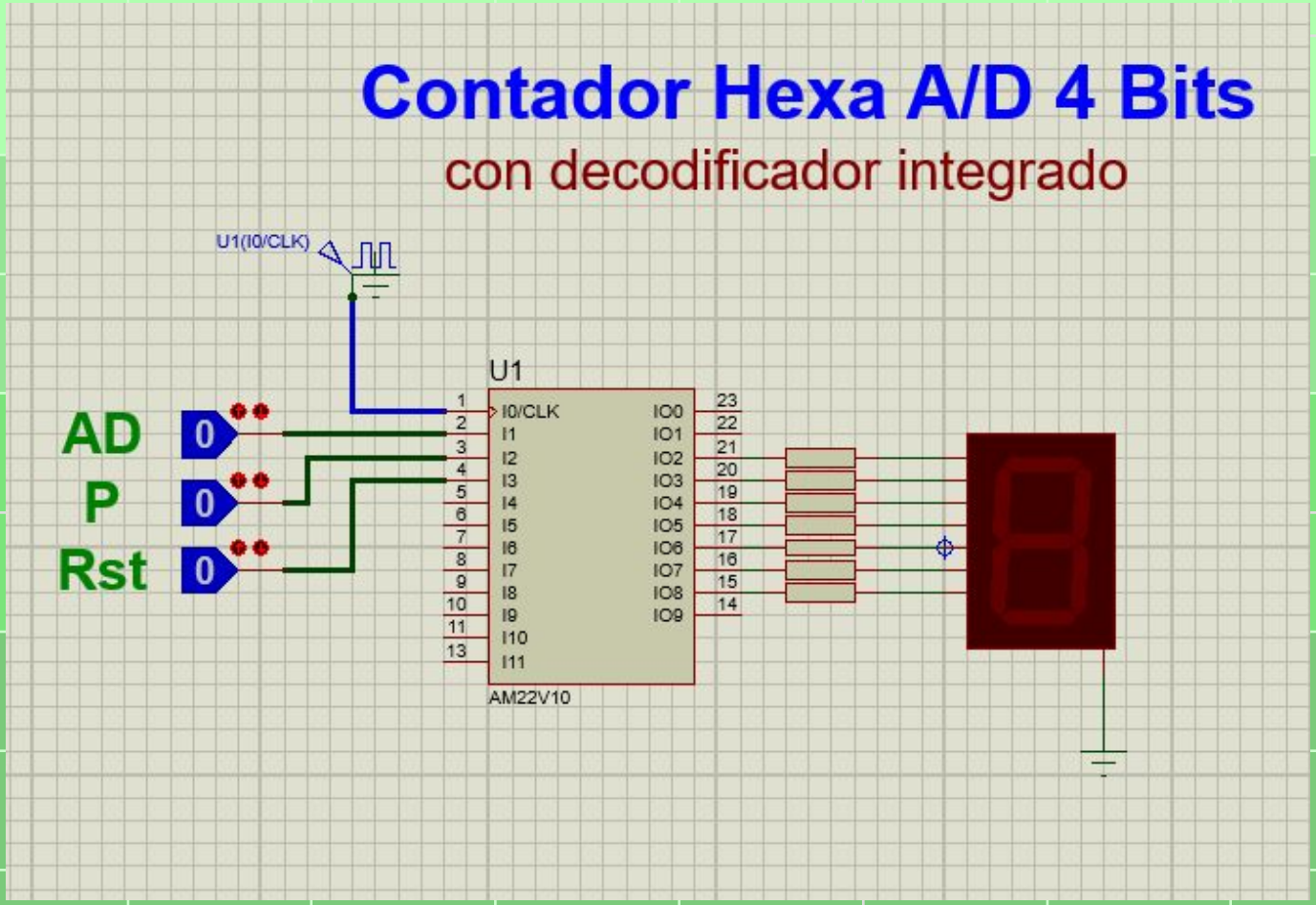
STATE E4:
IF !Rst&!P&!AD then E5;
IF !Rst&!P&AD then E3;
if !Rst&P then E4;
if Rst then E0:
STATE E5:
IF !Rst&!P&!AD then E6;
IF !Rst&!P&AD then E4;
if !Rst&P then E5;
if Rst then E0:
STATE E6:
IF !Rst&!P&!AD then E7;
IF !Rst&!P&AD then E5;
if !Rst&P then E6;
if Rst then E0:
STATE E7:
IF !Rst&!P&!AD then E8;
IF !Rst&!P&AD then E6;
if !Rst&P then E7;
if Rst then E0:

STATE E8:
IF !Rst&!P&!AD then E9;
IF !Rst&!P&AD then E7;
if !Rst&P then E8;
if Rst then E0:
STATE E9:
IF !Rst&!P&!AD then EA;
IF !Rst&!P&AD then E8;
if !Rst&P then E9;
if Rst then E0:
STATE EA:
IF !Rst&!P&!AD then EB;
IF !Rst&!P&AD then E9;
if !Rst&P then EA;
if Rst then E0:
STATE EB:
IF !Rst&!P&!AD then EC;
IF !Rst&!P&AD then EA;
if !Rst&P then EB;
if Rst then E0:

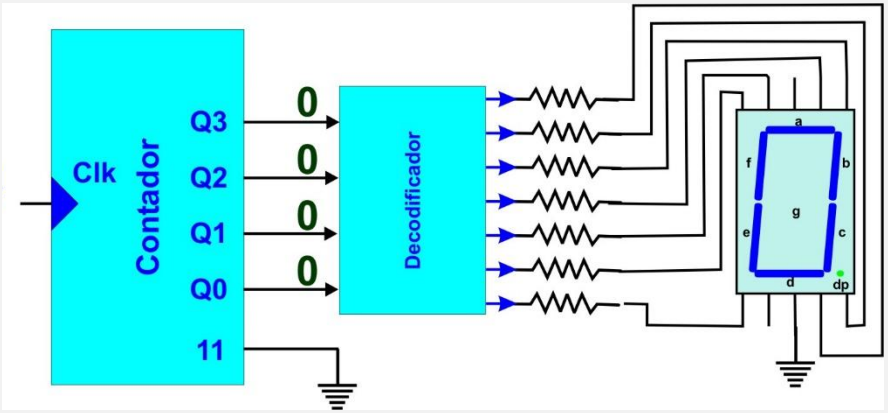
STATE EC:
IF !Rst&!P&!AD then ED;
IF !Rst&!P&AD then EB;
if !Rst&P then EC;
if Rst then E0:
STATE ED:
IF !Rst&!P&!AD then EE;
IF !Rst&!P&AD then EC;
if !Rst&P then ED;
if Rst then E0:
STATE EE:
IF !Rst&!P&!AD then EF;
IF !Rst&!P&AD then ED;
if !Rst&P then EE;
if Rst then E0:
STATE EF:
IF !Rst&!P&!AD then E0;
IF !Rst&!P&AD then EE;
if !Rst&P then EF;
if Rst then E0:
END

Contador Hexa A/D 4 Bits

con decodificador integrado



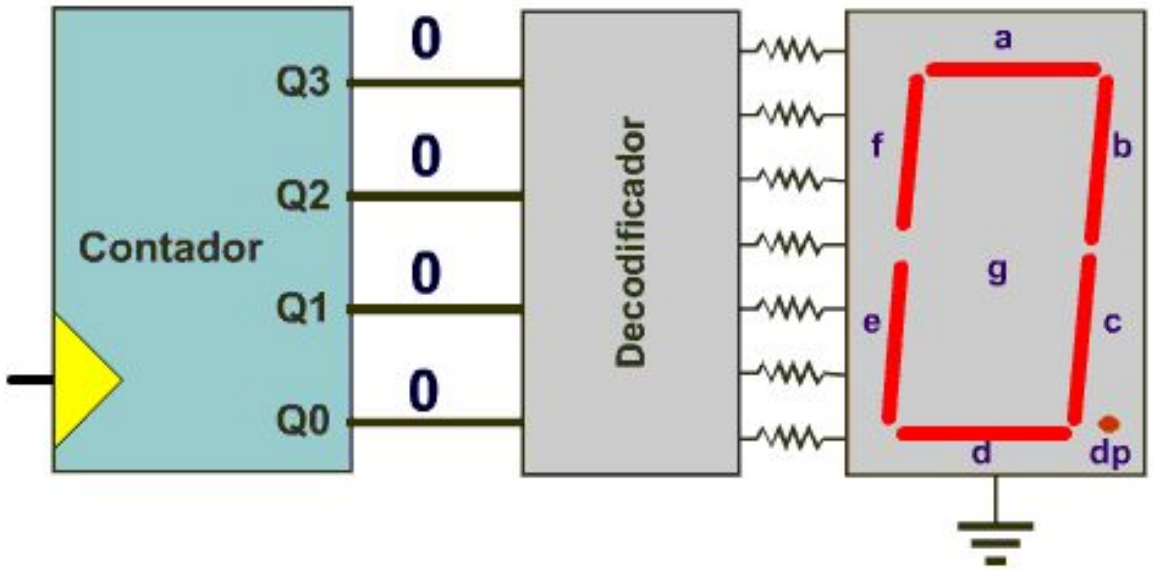
Contadores Decimales



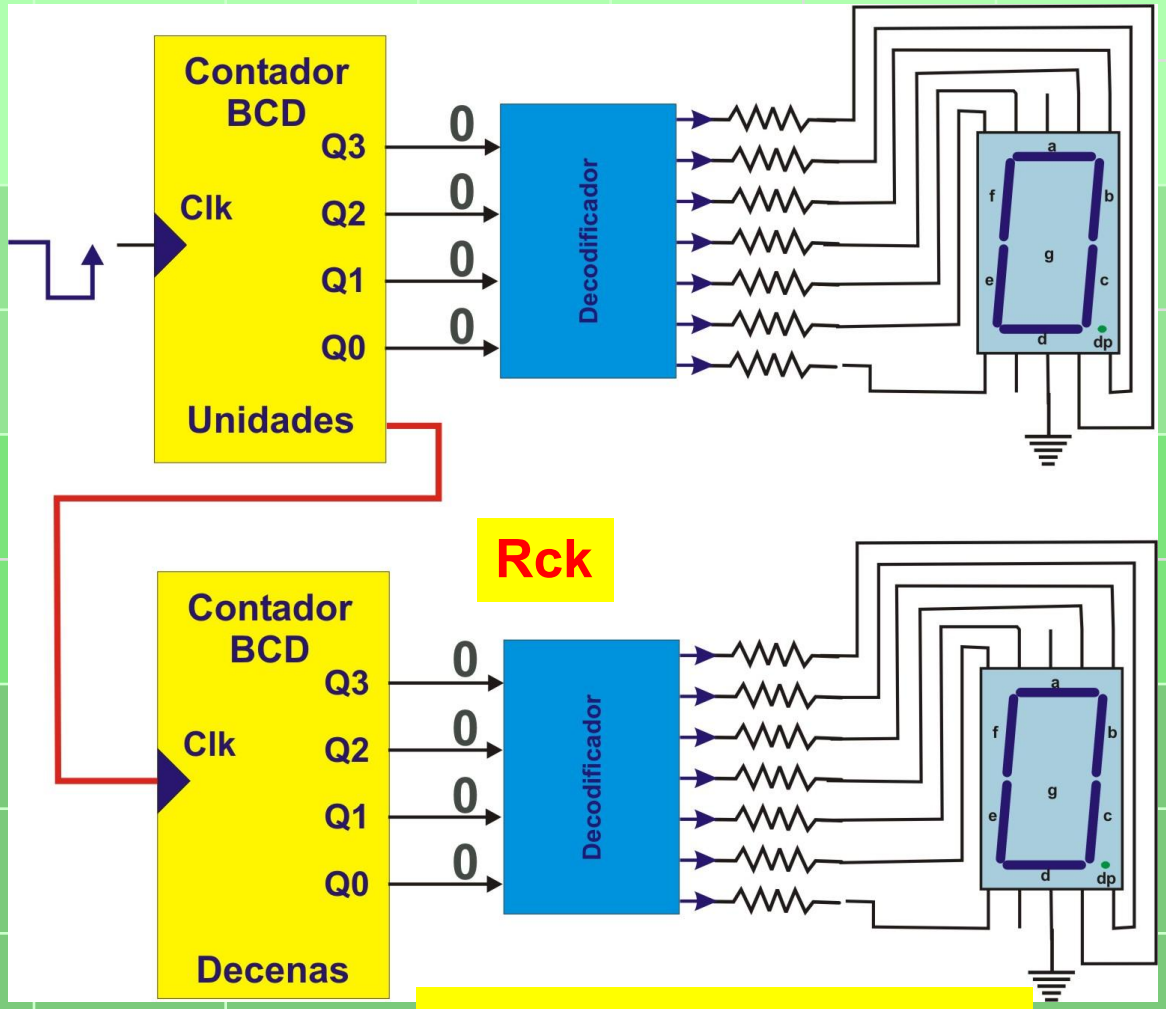
Los contadores decimales o también llamados de décadas o en BCD son de cuatro bit's, porque solo cuentan de:

- **0 a 9 ascendente**
- **9 a 0 descendente**

Contador decimal (BCD o décadas)

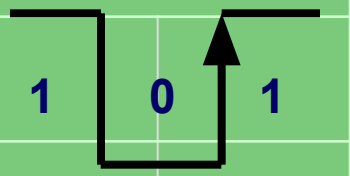


Contador de 0 a 99

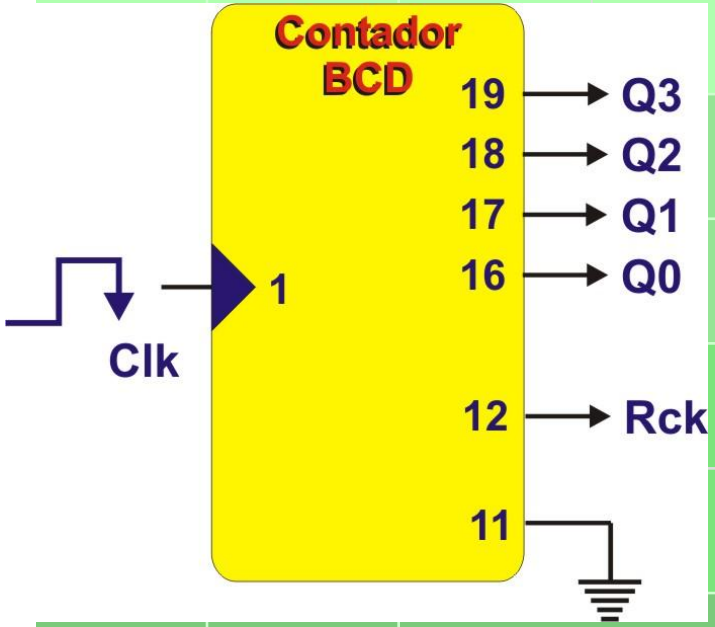


Rck = Ripple Clock

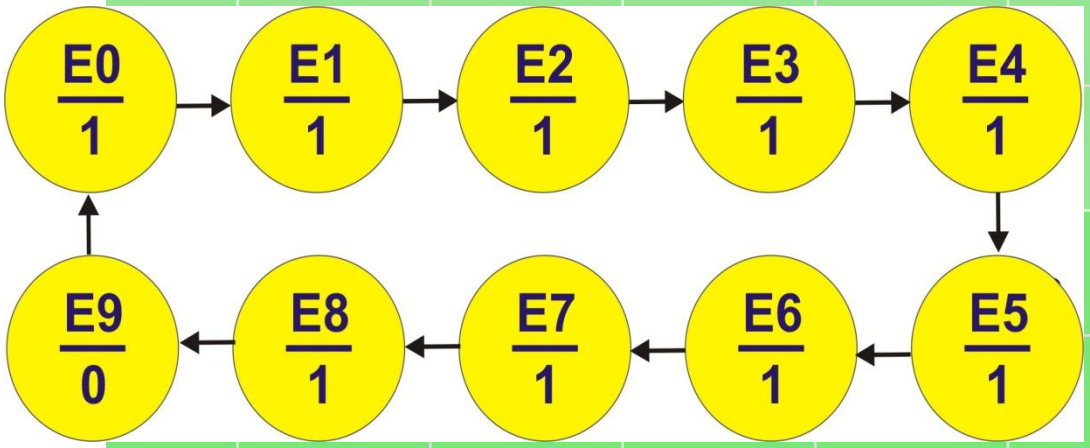
	Ck	Q3	Q2	Q1	Q0	Rck
0	↑	0	0	0	0	1
1	↑	0	0	0	1	1
2	↑	0	0	1	0	1
3	↑	0	0	1	1	1
4	↑	0	1	0	0	1
5	↑	0	1	0	1	1
6	↑	0	1	1	0	1
7	↑	0	1	1	1	1
8	↑	1	0	0	0	1
9	↑	1	0	0	1	0
0	↑	0	0	0	0	1



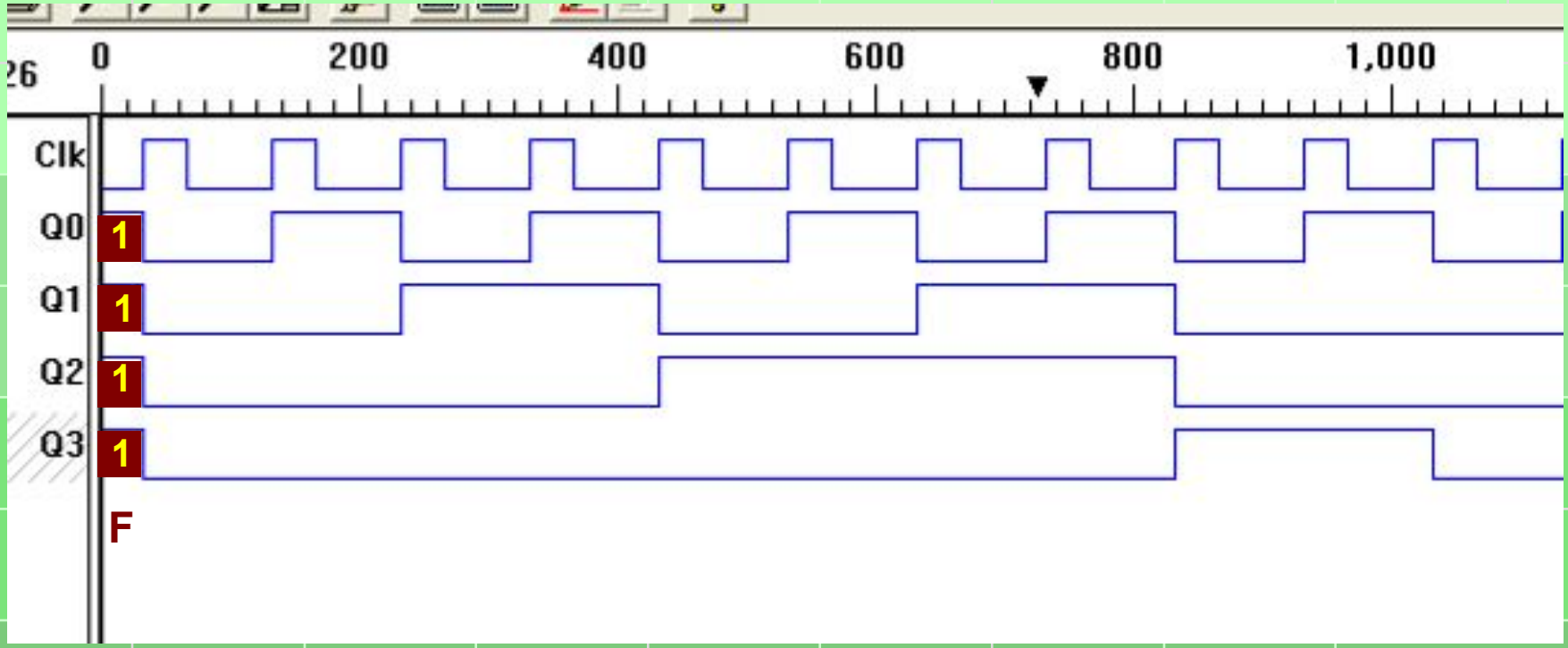
```
MODULE deca
"entrada
Clk pin 1;
"Salida combinacional
Rck pin 12 istype 'com';
"Salidas registradas
Q3..Q0 pin 19..16 istype 'reg';
D=[Q3..Q0];
equations
D.clk=Clk;
```



Truth_table
(D:>D)
0:>1;
1:>2;
2:>3;
3:>4;
4:>5;
5:>6;
6:>7;
7:>8;
8:>9;
9:>0;

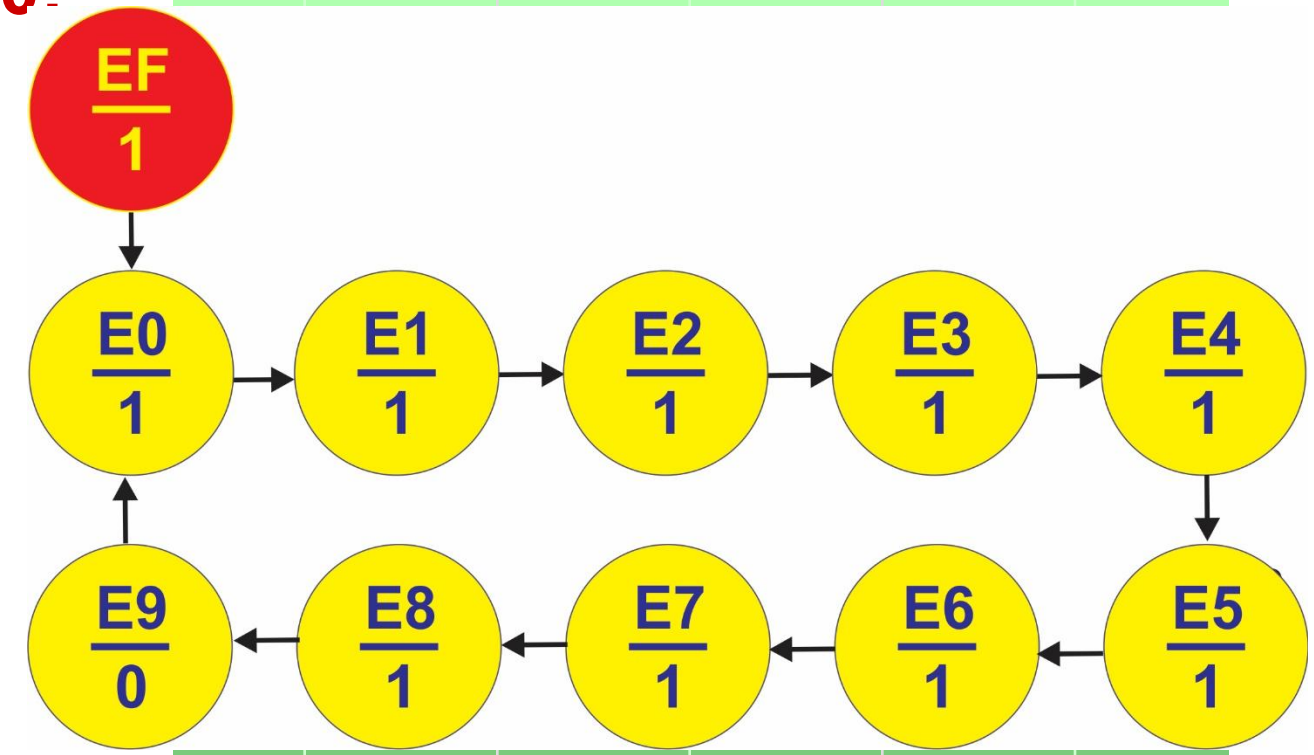


Truth_table
(D->Rck)
0->1;
1->1;
2->1;
3->1;
4->1;
5->1;
6->1;
7->1;
8->1;
9->0;


26
0
200
400
600
800
1,000
Clk
Q0
1
Q1
1
Q2
1
Q3
1
F

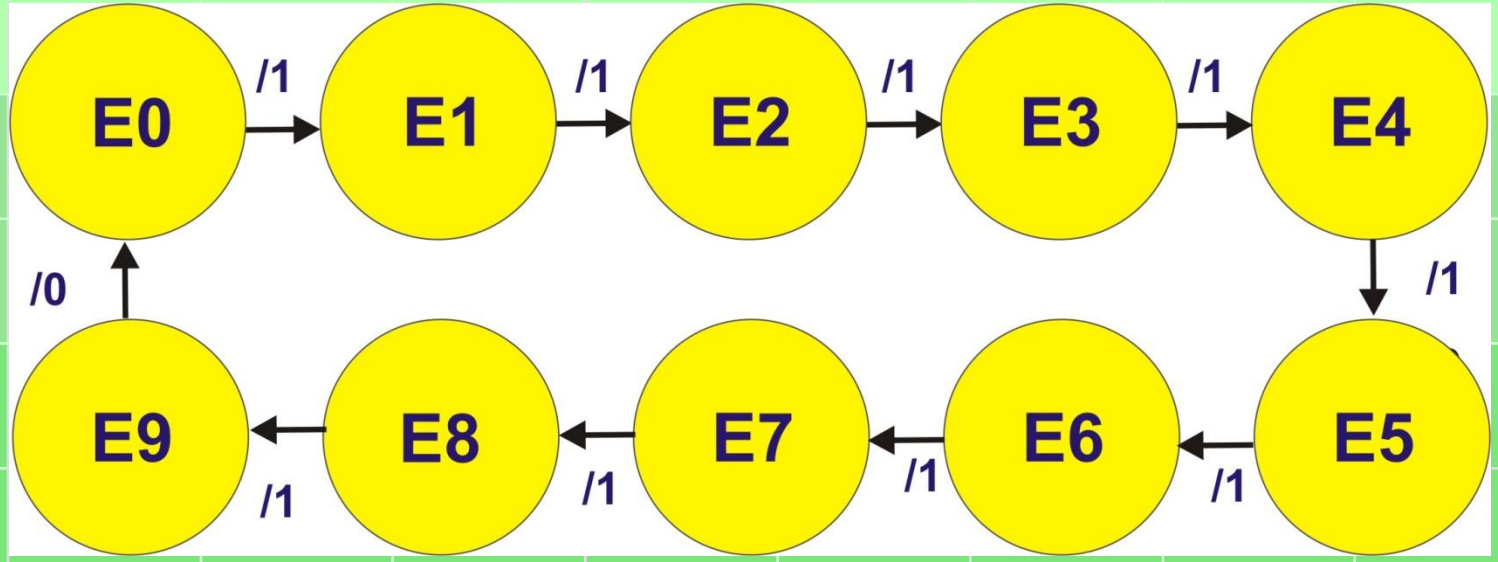
Truth_table
(D:>D)
^HF:>0:
0:>1;
1:>2;
2:>3;
3:>4;
4:>5;
5:>6;
6:>7;
7:>8;
8:>9;
9:>0;

^HF = Hexadecimal F (1111)



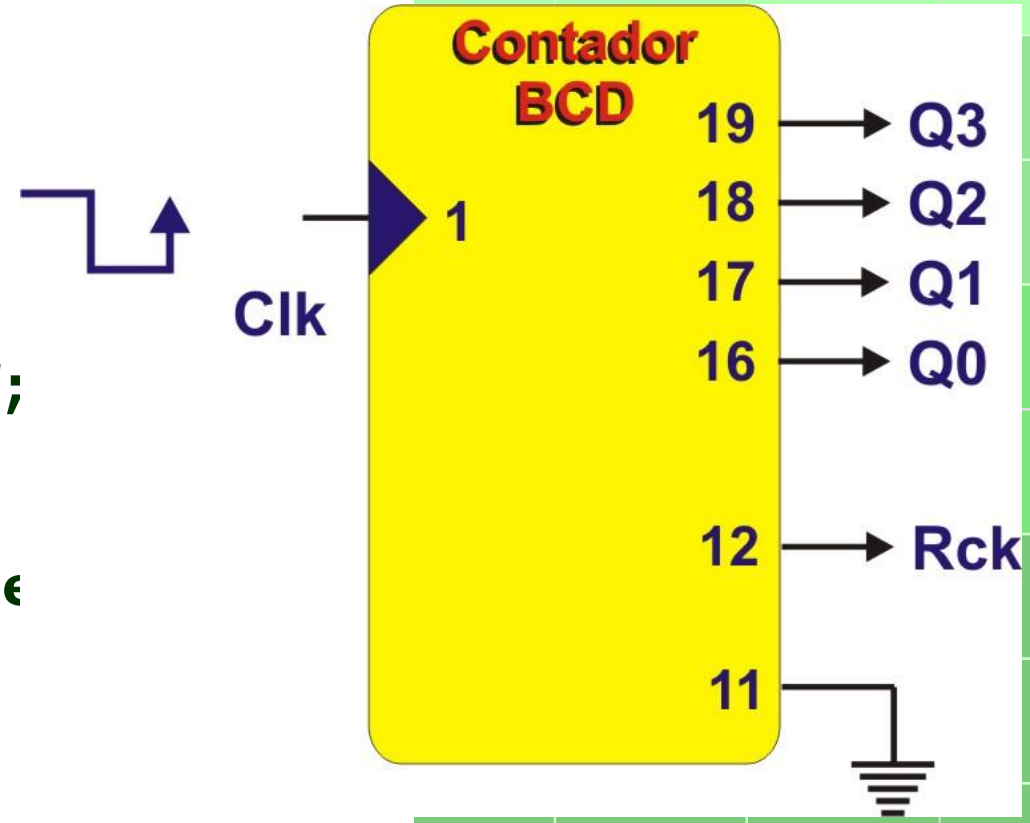
Truth_table
(D->Rck)
0->1;
1->1;
2->1;
3->1;
4->1;
5->1;
6->1;
7->1;
8->1;
9->0;

Diagrama de transición



	Ck	Q3	Q2	Q1	Q0	Rck	
8	↑	1	0	0	0	1	
9	↑	1	0	0	1	0	
0	↑	0	0	0	0	1	

```
MODULE deca

"entrada

Clk pin 1;

"Salida combinacional

Rck pin 12 istype 'com';

"Salidas registradas

Q3..Q0 pin 19..16 istype

D=[Q3..Q0];

equations
D.clk=Clk;
```



Asignación de valores a los estados

```
declarations
E0=[0,0,0,0];
E1=[0,0,0,1];
E2=[0,0,1,0];
E3=[0,0,1,1];
E4=[0,1,0,0];
E5=[0,1,0,1];
E6=[0,1,1,0];
E7=[0,1,1,1];
E8=[1,0,0,0];
E9=[1,0,0,1];
```

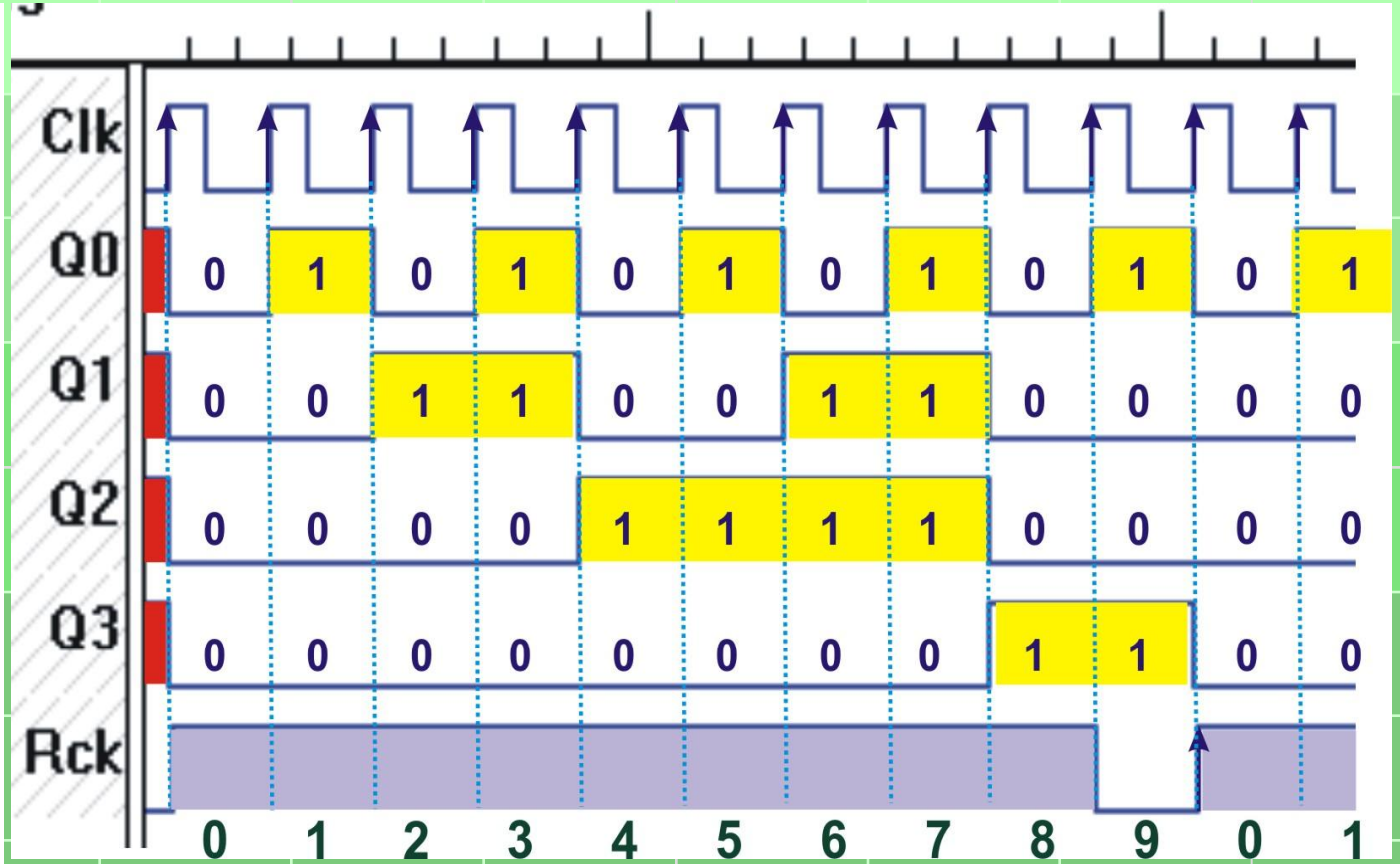
Máquina de Mealy

```
state_diagram D
State E0:
goto E1 with Rck=1;
state E1:
goto E2 with Rck=1;
state E2:
goto E3 with Rck=1;
state E3:
goto E4 with Rck=1;
state E4:
goto E5 with Rck=1;
```

```
state E5:
goto E6 with Rck=1;
state E6:
goto E7 with Rck=1;
state E7:
goto E8 with Rck=1;
state E8:
goto E9 with Rck=1;
state E9:
goto E0 with Rck=0;
```

test_vectors (Clk->D)

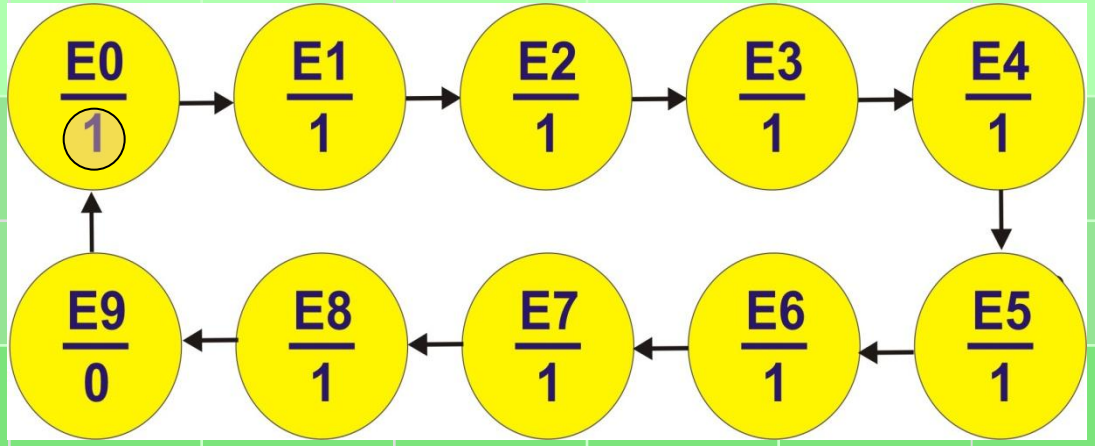
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;



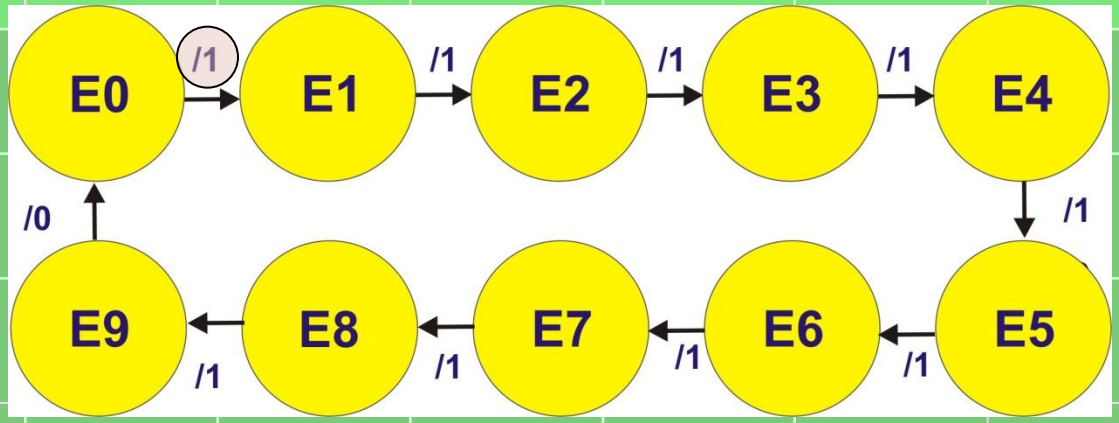
Truth_table (D->Rck)

0->1;
1->1;
2->1;
3->1;
4->1;
5->1;
6->1;
7->1;
8->1;
9->0;

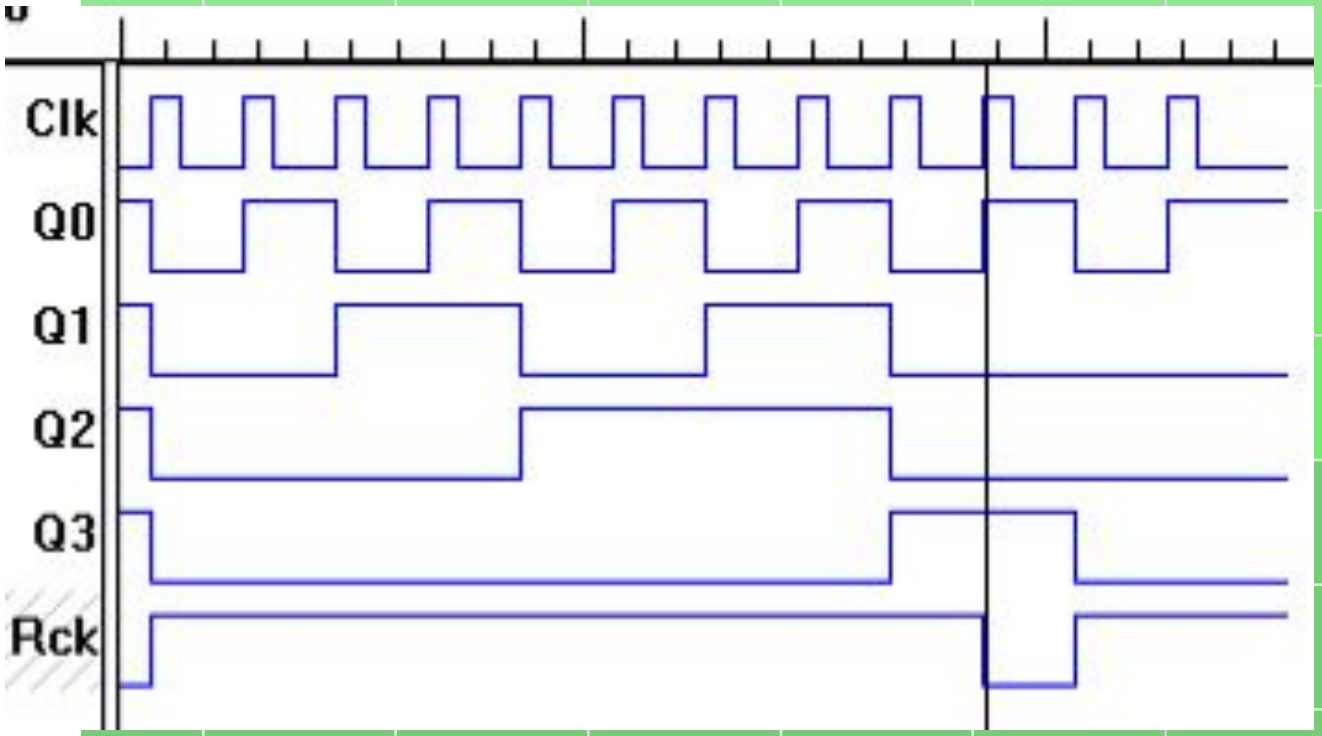
Máquina de Moore

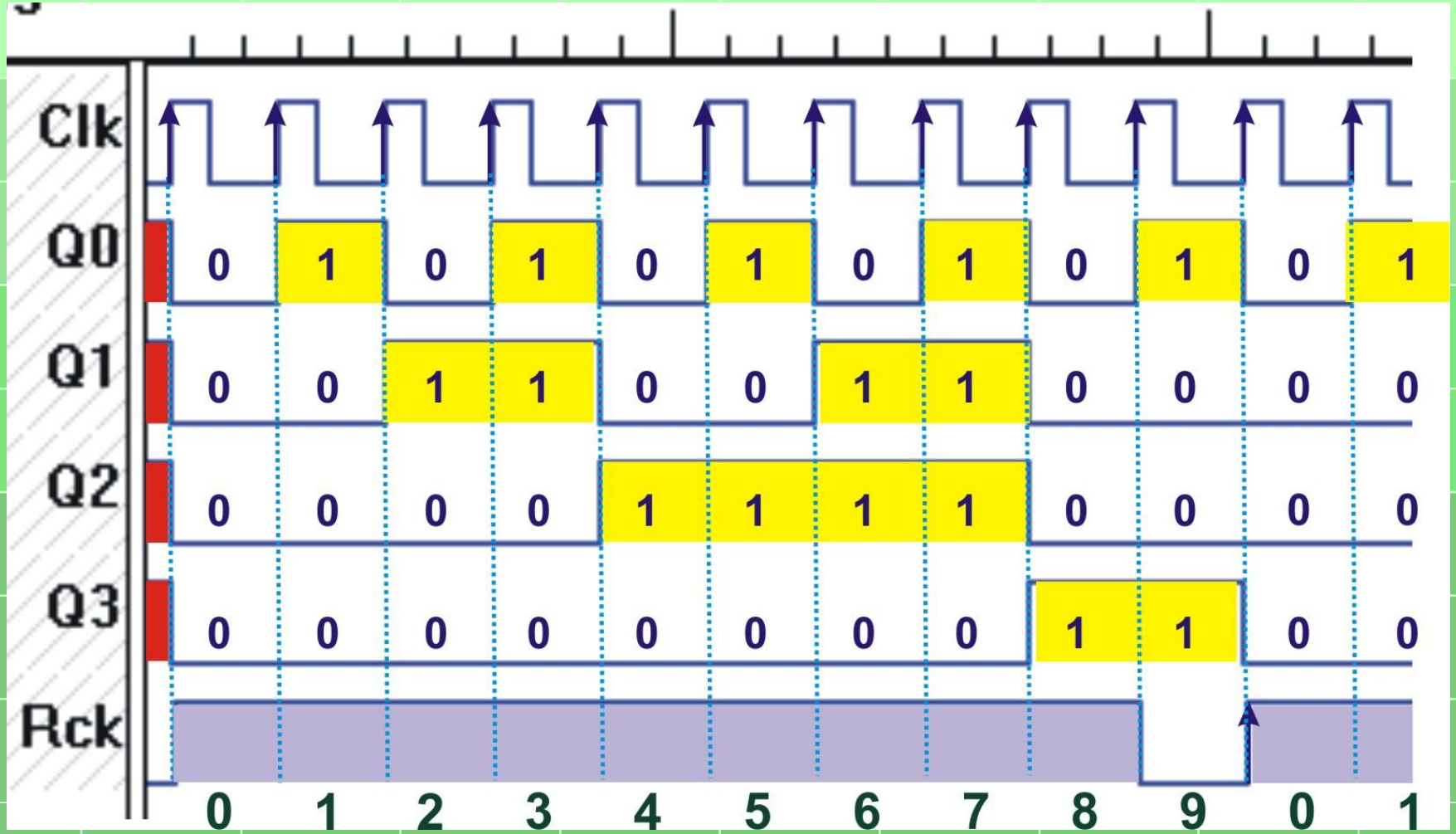


Máquina de Mealy

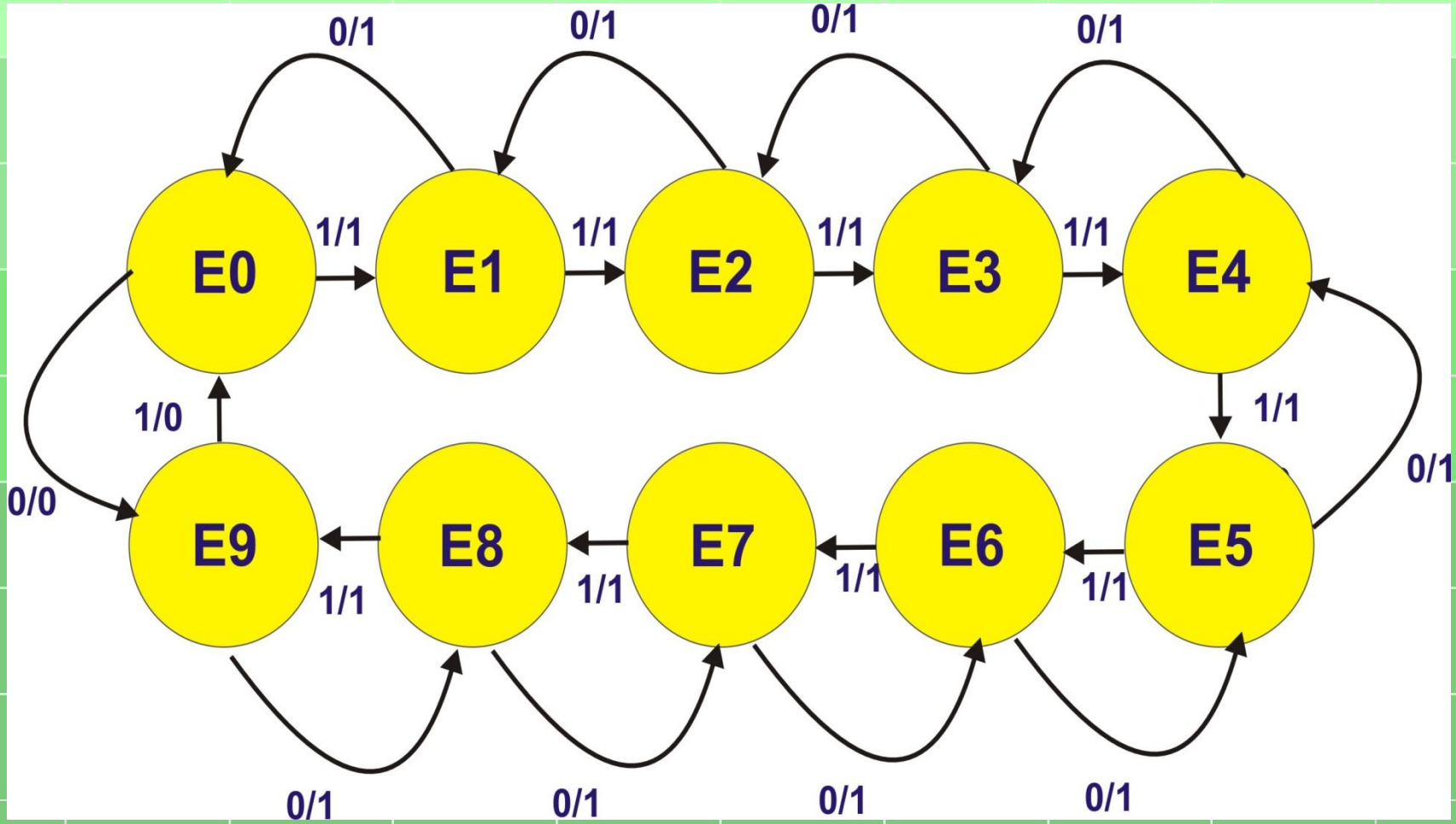


```
test_vectors
(Clk->D)
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
.c.->.x.;
END
```

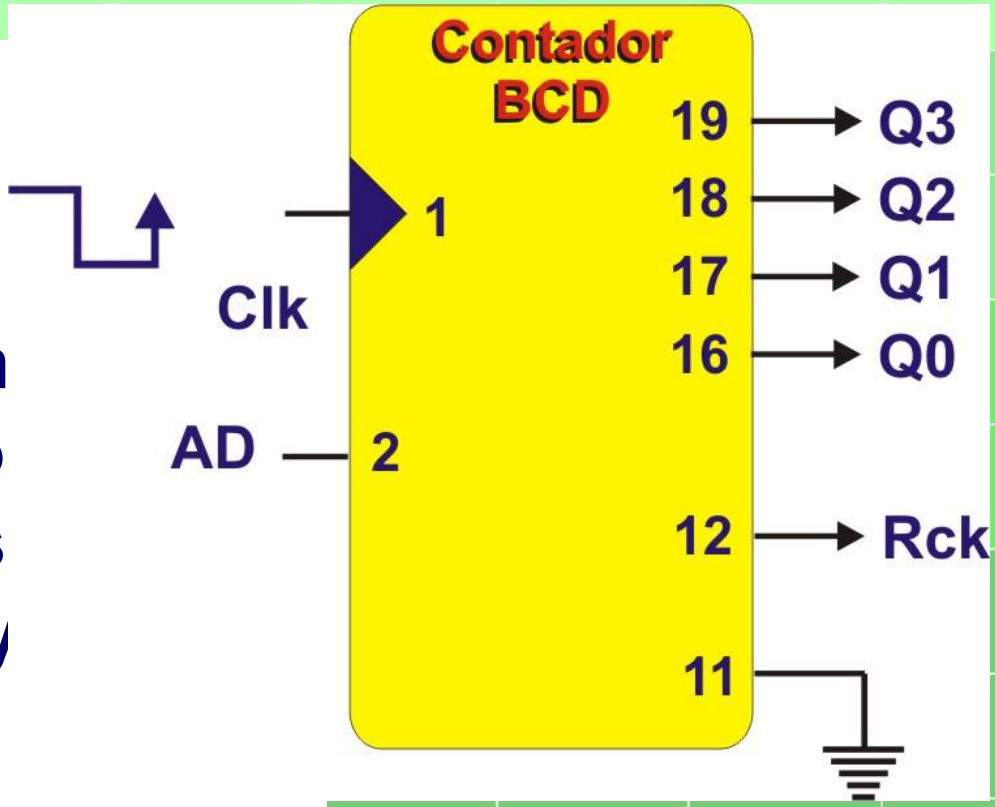



Clk
Q0
0 1 0 1 0 1 0 1 0 1 0 1
Q1
0 0 1 1 0 0 1 1 0 0 0 0
Q2
0 0 0 0 1 1 1 1 0 0 0 0
Q3
0 0 0 0 0 0 0 0 1 1 0 0
Rck
0 1 2 3 4 5 6 7 8 9 0 1

Contador asendente/descendente



```
MODULE adbcd
"entradas
Clk, AD pin 1,2;
"salida Combinacion
Rck pin 12 istype 'co
"salidas Registradas
Q3..Q0 pin 19..16 isty
D=[Q3..Q0];
equations
D.clk=Clk;
```



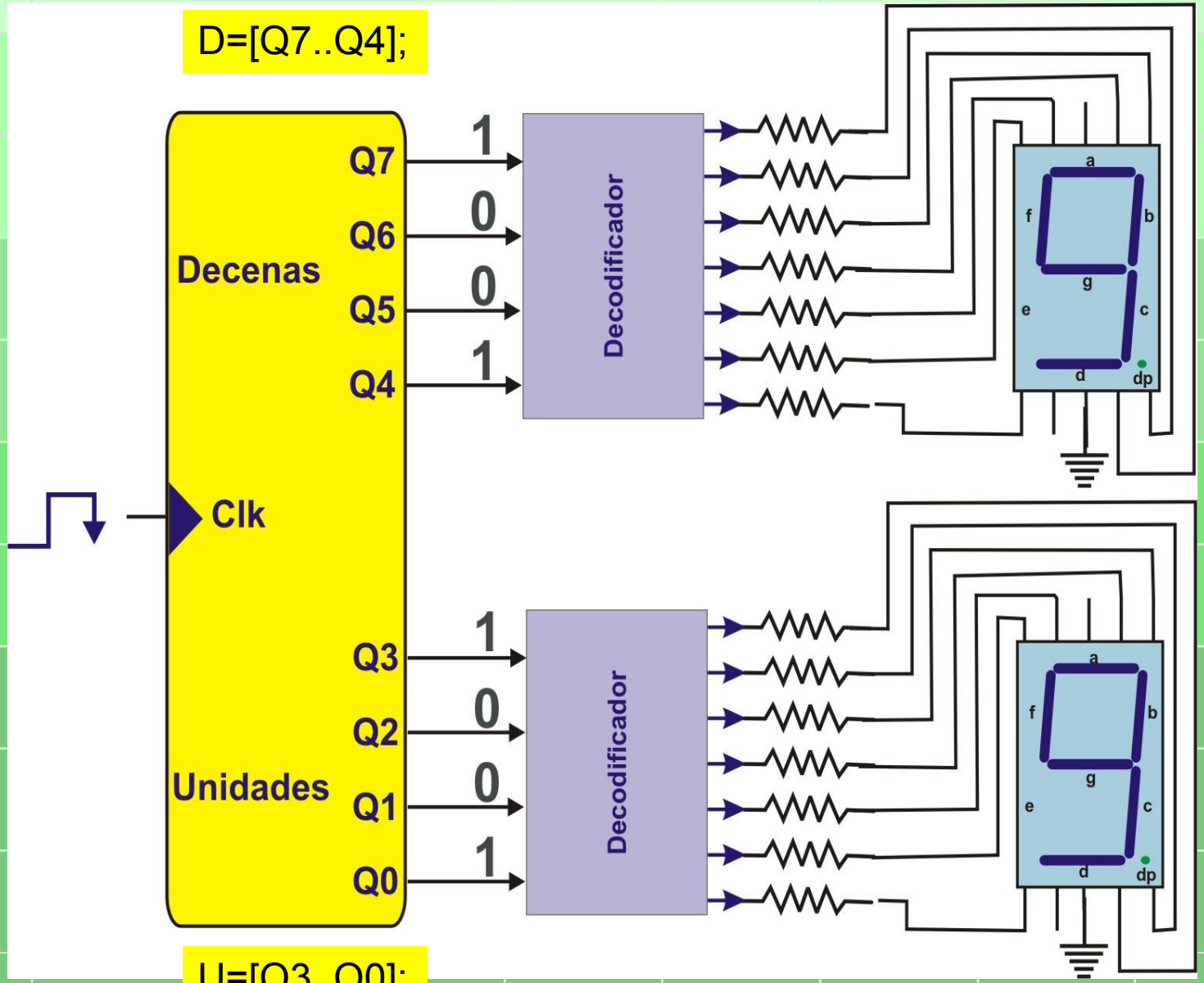
```
"asignación de valores a los estados
declarations
E0=[0,0,0,0];
E1=[0,0,0,1];
E2=[0,0,1,0];
E3=[0,0,1,1];
E4=[0,1,0,0];
E5=[0,1,0,1];
E6=[0,1,1,0];
E7=[0,1,1,1];
E8=[1,0,0,0];
E9=[1,0,0,1];
```

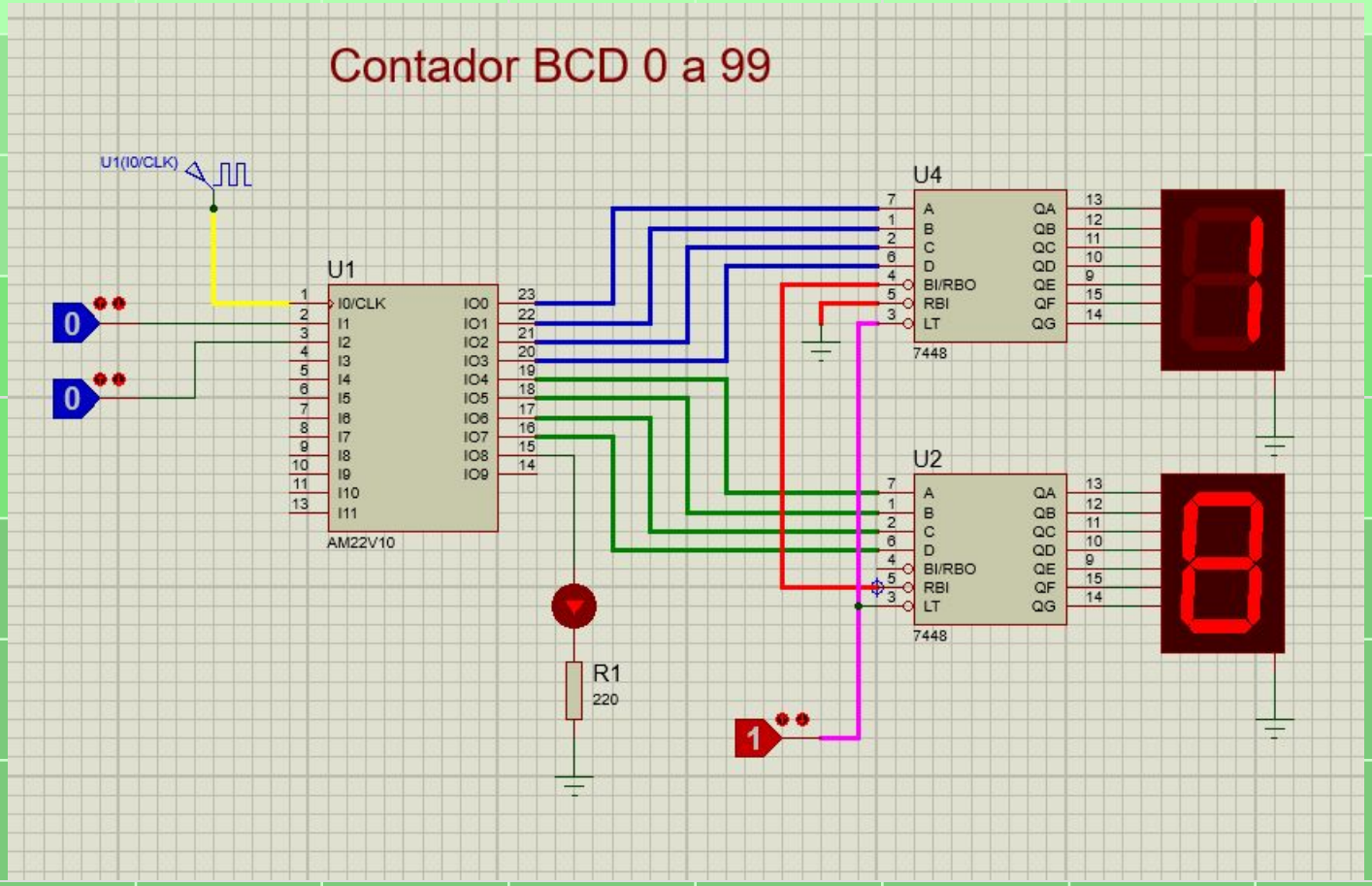
Máquina de Mealy

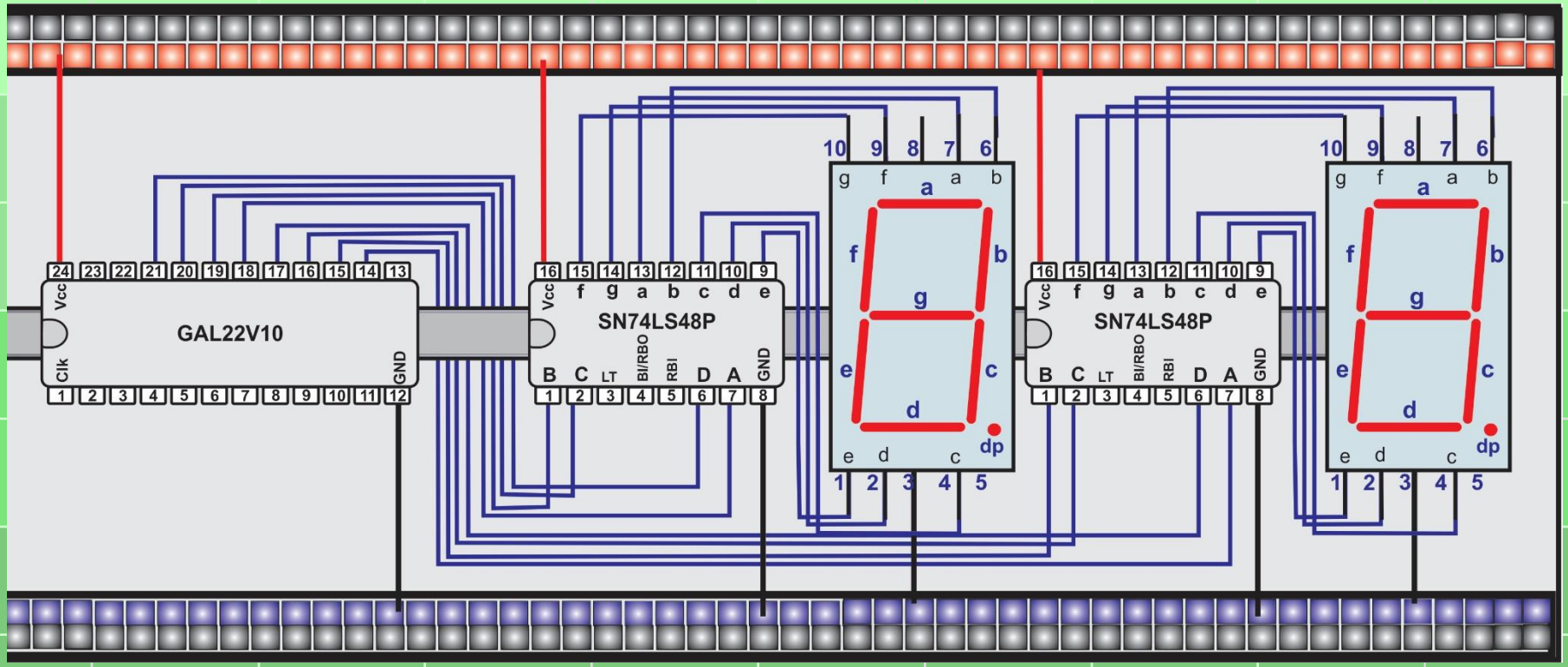
```
state_diagram D
State E0:
If AD then E1;
If !AD Then E9 with Rck=0;
state E1:
If AD then E2 else E0 with Rck=1;
state E2:
If AD then E3 else E1 with Rck=1;
state E3:
If AD then E4 else E2 with Rck=1;
state E4:
If AD then E5 else E3 with Rck=1;
state E5:
If AD then E6 else E4 with Rck=1;
state E6:
If AD then E7 else E5 with Rck=1;
state E7:
If AD then E8 else E6 with Rck=1;
state E8:
If AD then E9 else E7 with Rck=1;
state E9:
If AD then E0 with Rck=0;
if !AD Then E8 with Rck=1;
```

Máquina de Moore

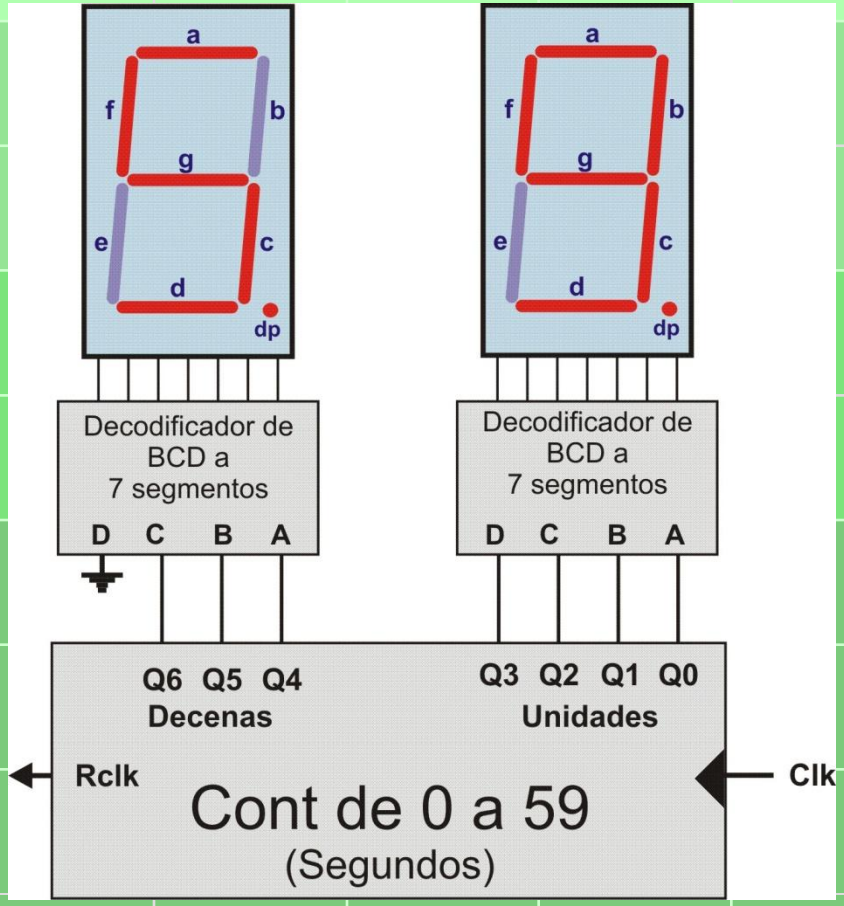
```
state_diagram D
State E0:
Rck=1;
If AD then E1 else E9;
state E1:
Rck=1;
If AD then E2 else E0;
state E2:
Rck=1;
If AD then E3 else E1;
state E3:
Rck=1;
If AD then E4 else E2;
state E4:
Rck=1;
If AD then E5 else E3;
state E5:
Rck=1;
If AD then E6 else E4;
state E6:
Rck=1;
If AD then E7 else E5;
state E7:
Rck=1;
If AD then E8 else E6;
state E8:
Rck=1;
If AD then E9 else E7 ;
state E9:
Rck=0;
If AD then E0 else E8;
```


D=[Q7..Q4];
Decenas
Q7
Q6
Q5
Q4
1
0
0
1
Clk
Decodificador
a
b
c
d
e
f
g
dp
Unidades
Q3
Q2
Q1
Q0
1
0
0
1
Decodificador
U=[Q3..Q0];


Contador BCD 0 a 99
U1(I0/CLK)
U1
I0/CLK
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
AM22V10
U4
U2
A
B
C
D
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
7448
R1
220
0
0
1

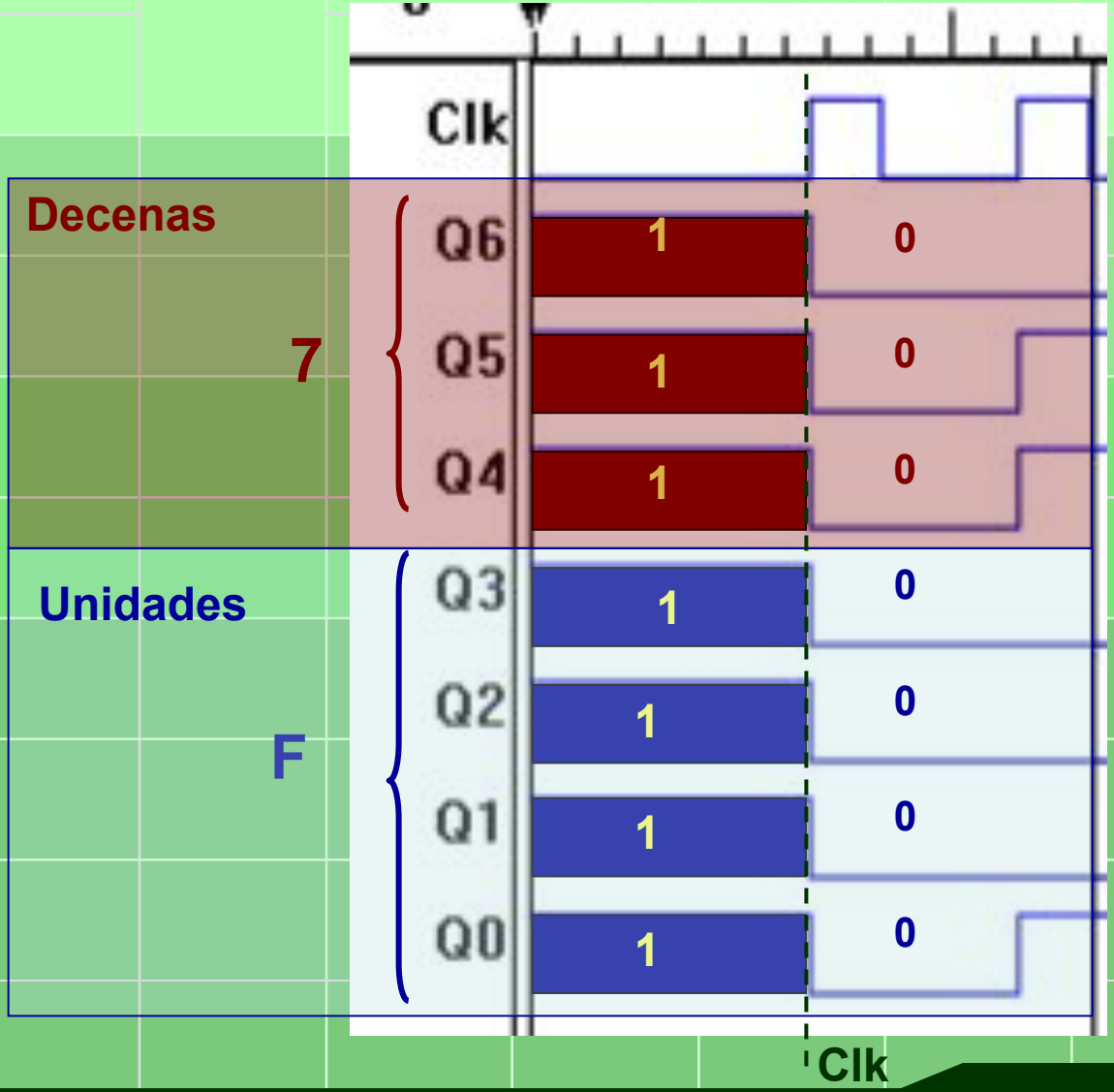

GAL22V10
Vcc
Clk
GND
SN74LS48P
B C LT BI/RBO RBI D A
f g a b c d e
dp
SN74LS48P

Contador de 0 a 59 ascendente



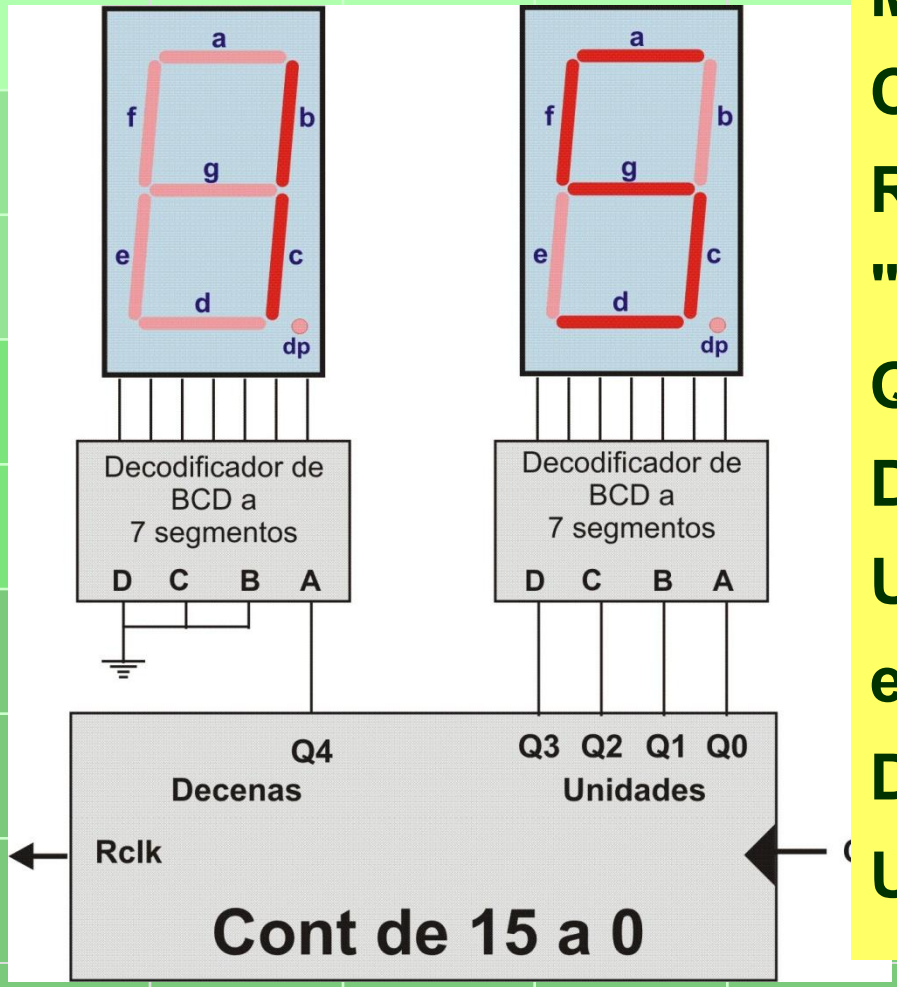

```
MODULE cnn
"entrada
Clk pin 1;
"Salida
Rck pin 12 istype 'com';
"Salidas registradas
Q6..Q0 pin 19..13 istype 'reg';
D=[Q6..Q4];
U=[Q3..Q0];
equations
D.clk=Clk;
U.Clk=Clk;
```

[7, ^HF]:>[0,0];




```
truth_table
([D,U]:>[D,U])
[7, ^HF]:>[0,0];
[0,0]:>[0,1];
[0,1]:>[0,2];
[0,2]:>[0,3];
[0,3]:>[0,4];
[0,4]:>[0,5];
[0,5]:>[0,6];
[0,6]:>[0,7];
[0,7]:>[0,8];
[0,8]:>[0,9];
[0,9]:>[1,0];
[1,0]:>[1,1];
[1,1]:>[1,2];
[1,2]:>[1,3];
.....................
[5,8]:>[5,9];
[5,9]:>[0,0];
```

Contador de 15 a 0 descendente

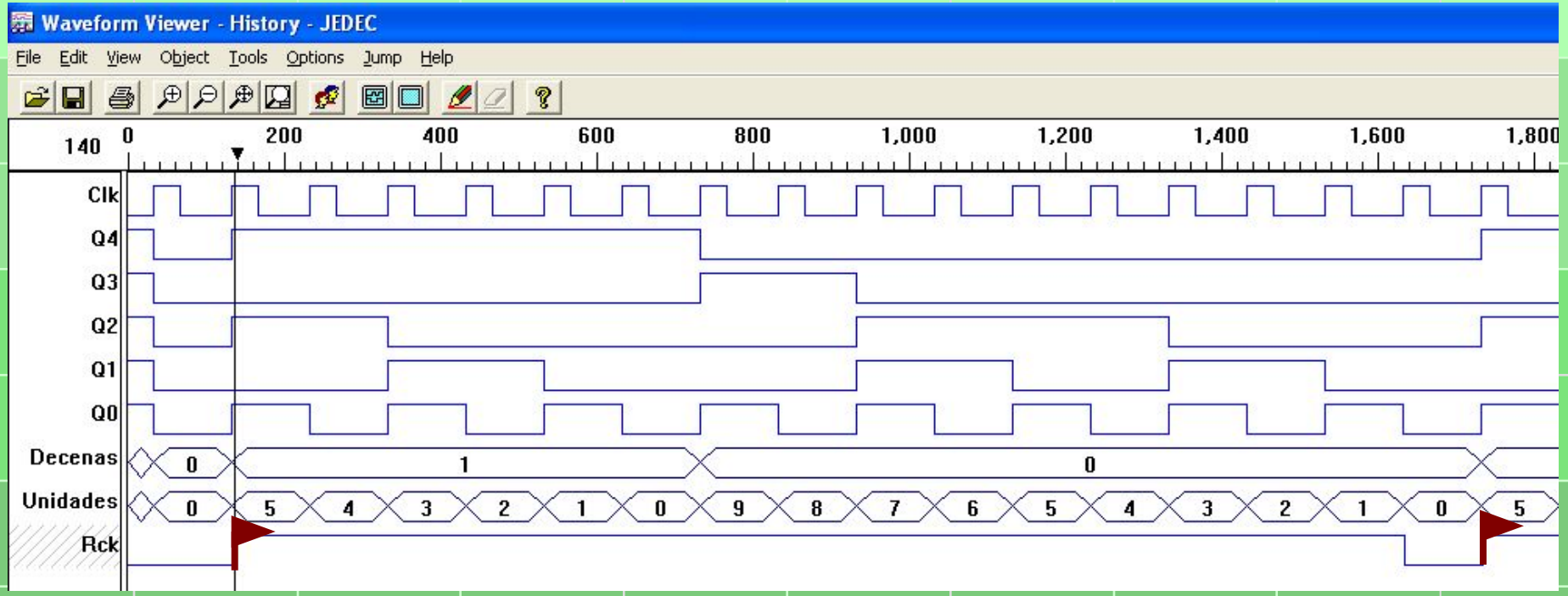


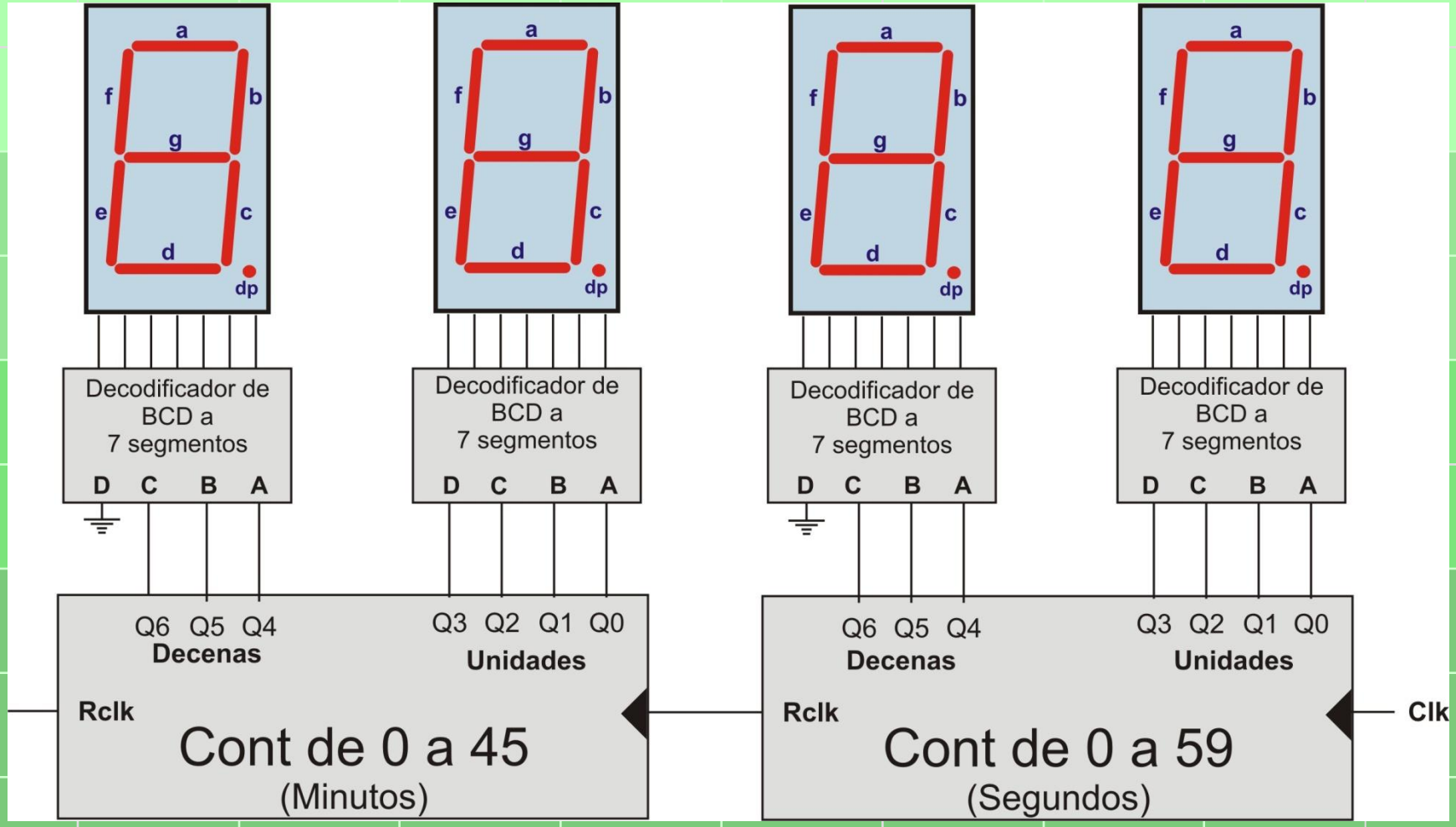

```
MODULE cont
Clk pin 1;
Rck pin 12 istype 'com';
"Salidas registradas
Q4..Q0 pin 19..15 istype 'reg';
D=[Q4];
U=[Q3..Q0];
equations
D.clk=Clk;
U.Clk=Clk;
```

```
truth_table
([D,U]:>[D,U])
[1,^HF]:>[0,0];
[0,0]:>[1,5];
[0,1]:>[0,0];
[0,2]:>[0,1];
[0,3]:>[0,2];
[0,4]:>[0,3];
[0,5]:>[0,4];
[0,6]:>[0,5];
[0,7]:>[0,6];
[0,8]:>[0,7];
[0,9]:>[0,8];
[1,0]:>[0,9];
[1,1]:>[1,0];
[1,2]:>[1,1];
[1,3]:>[1,2];
[1,4]:>[1,3];
[1,5]:>[1,4];
```

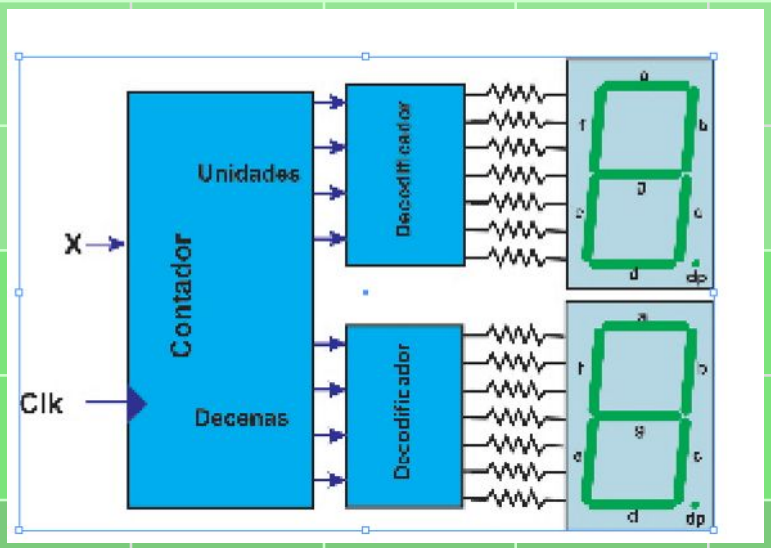
```
“tabla para Rck
truth_table
([D,U]->[Rck])
[0,0]->[0];
[0,1]->[1];
[0,2]->[1];
[0,3]->[1];
[0,4]->[1];
[0,5]->[1];
[0,6]->[1];
[0,7]->[1];
[0,8]->[1];
[0,9]->[1];
[1,0]->[1];
[1,1]->[1];
[1,2]->[1];
[1,3]->[1];
[1,4]->[1];
[1,5]->[1];
```

```
TEST_VECTORS
([Clk]->[D,U,Rck])
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
[.c.]->[.x.,.x.,.x.];
END
```


Waveform Viewer - History - JEDEC
File Edit View Object Tools Options Jump Help
140
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
Clk
Q4
Q3
Q2
Q1
Q0
Decenas
0
1
0
Unidades
0 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 5
Rck

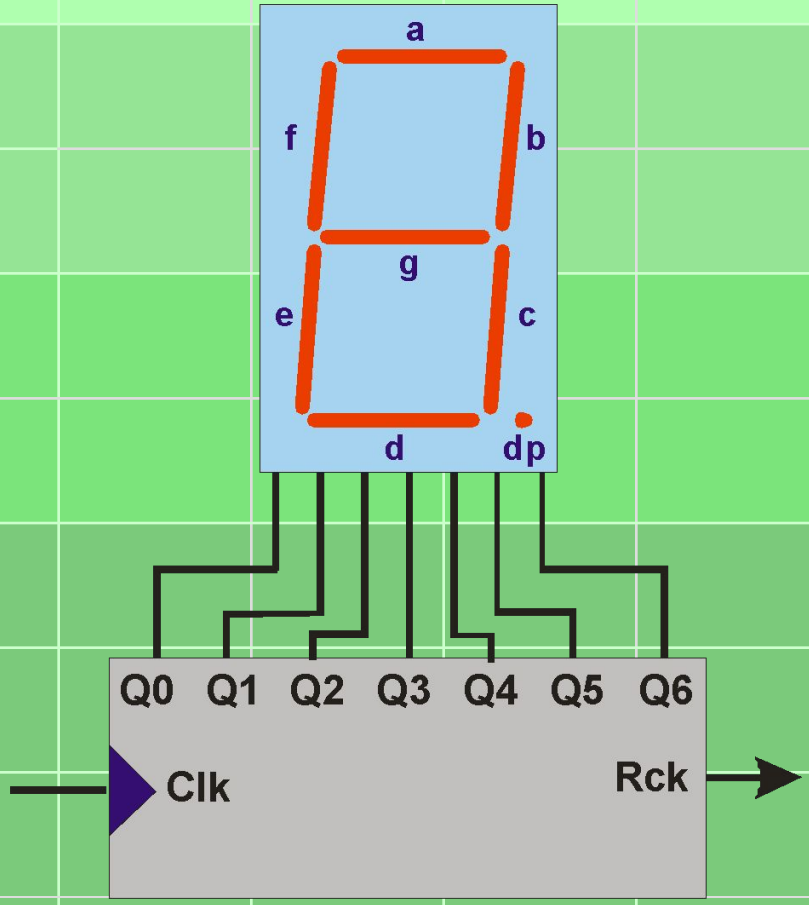

a
f
b
g
e
c
d
dp
Decodificador de BCD a 7 segmentos
D C B A
Q6 Q5 Q4
Decenas
Q3 Q2 Q1 Q0
Unidades
Rclk
Cont de 0 a 45
(Minutos)
Cont de 0 a 59
(Segundos)
Clk

Contador de 0 a 78 ascendente/descendente



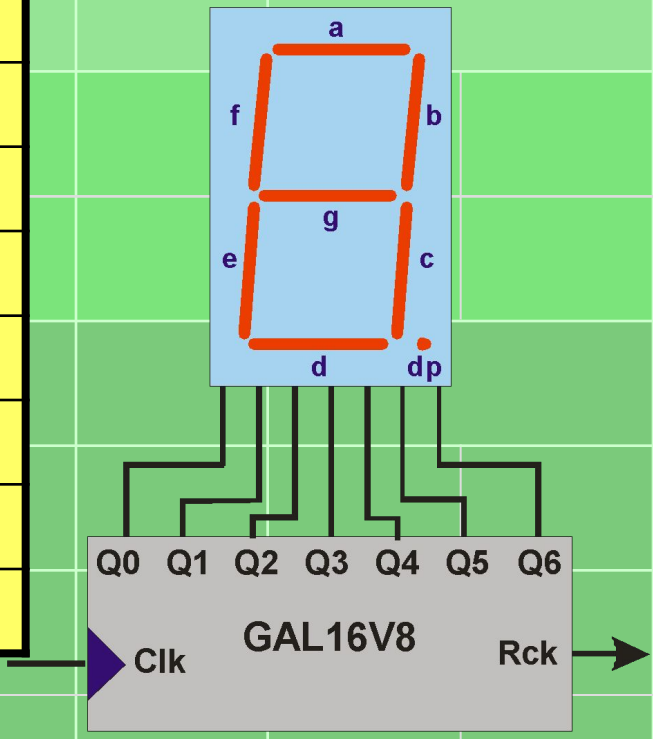

```
Module cont
"Entradas
Ck,X pin 1,2;
Q7..Q0 pin 19..12 istype 'dc,reg';
Sd=[Q7..Q0];
Equations
Sd.clk=Ck;
Declarations
D=[Q7..Q4];
U=[Q3..Q0];
Truth_table
([X,D,U]:>[D,U])
[0,0,0]:>[0,1];
[0,0,1]:>[0,2];
[0,0,2]:>[0,3];
[1,0,2]:>[0,1];
.....;
[1,1,2]:>[1,1];
.....;
[0,7,8]:>[0,0];
[1,0,0]:>[7,8];
[0,^HF,^HF]:>[0,0];
[1,^HF,^HF]:>[0,0];
```

Contadores que incluyen el decodificador



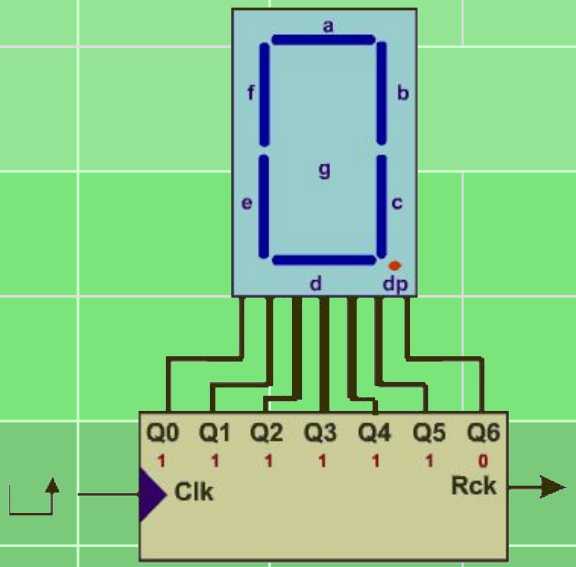
Contadores que incluyen el decodificador (BCD)

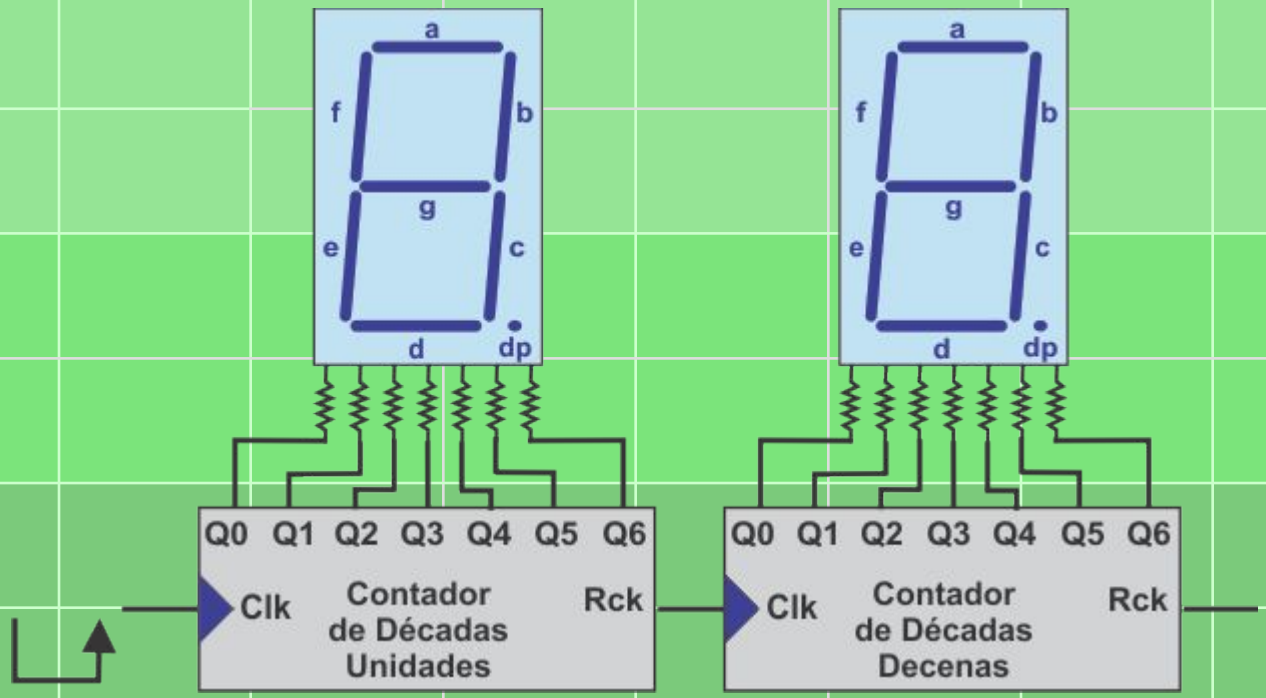
	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
	a	b	c	d	e	f	g	Rck
E0	1	1	1	1	1	1	0	1
E1	0	1	1	0	0	0	0	1
E2	1	1	0	1	1	0	1	1
E3	1	1	1	1	0	0	1	1
E4	0	1	1	0	0	1	1	1
E5	1	0	1	1	0	1	1	1
E6	X	0	1	1	1	1	1	1
E7	1	1	1	0	0	0	0	1
E8	1	1	1	1	1	1	1	1
E9	1	1	1	X	0	1	1	0



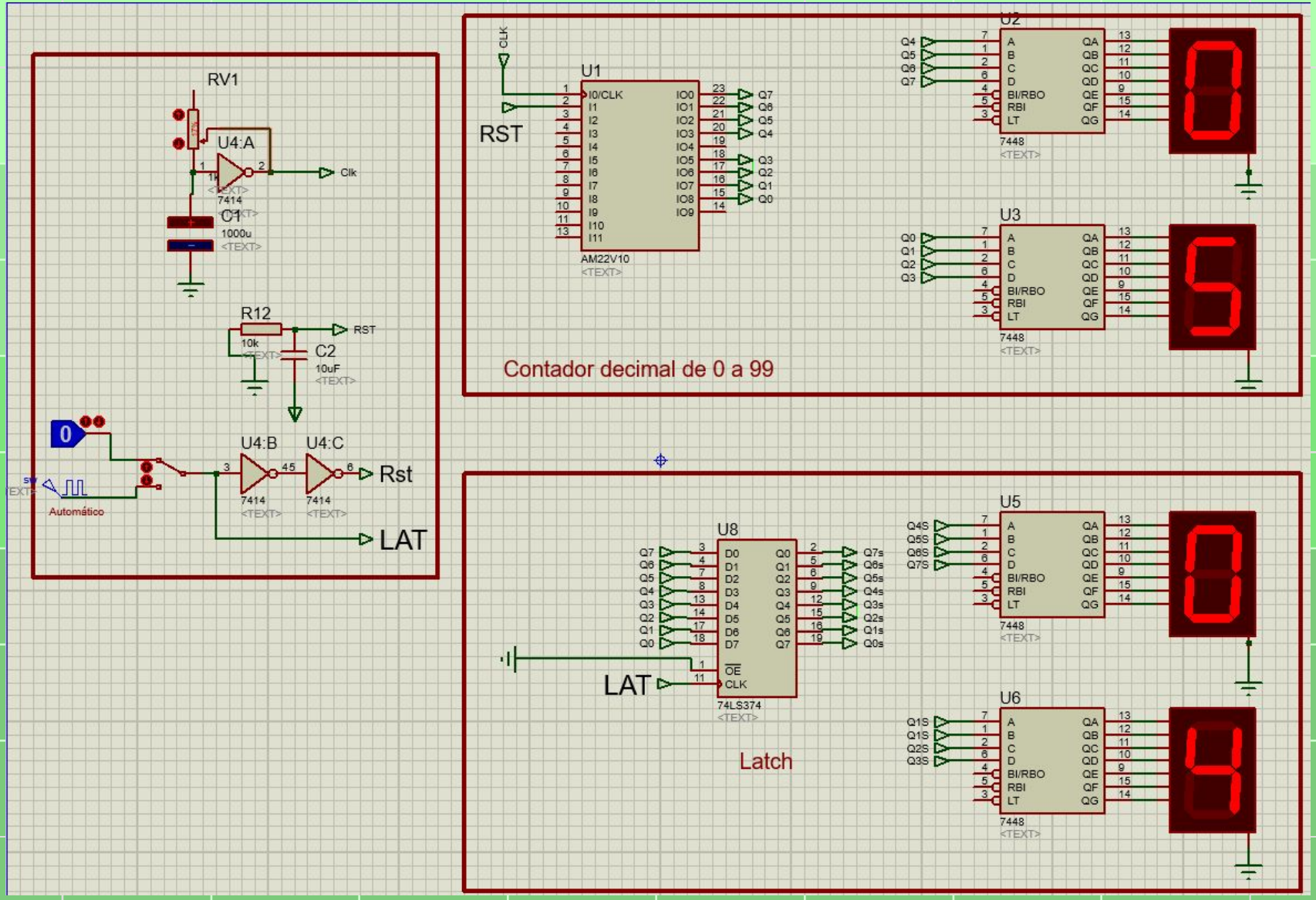
	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
	a	b	c	d	e	f	g	Rck
E0	1	1	1	1	1	1	0	1
E1	0	1	1	0	0	0	0	1
E2	1	1	0	1	1	0	1	1
E3	1	1	1	1	0	0	1	1
E4	0	1	1	0	0	1	1	1
E5	1	0	1	1	0	1	1	1
E6	1	0	1	1	1	1	1	1
E7	1	1	1	0	0	0	0	1
E8	1	1	1	1	1	1	1	1
E9	1	1	1	X	0	1	1	1
E10	1	1	1	0	1	1	1	1
E11	0	0	1	1	1	1	1	1
E12	1	0	0	1	1	1	0	1
E13	0	0	1	1	1	1	1	1
E14	1	0	0	1	1	1	0	1
E15	1	0	0	0	1	1	1	0

Contadores que incluyen el decodificador (Hexadecimal)

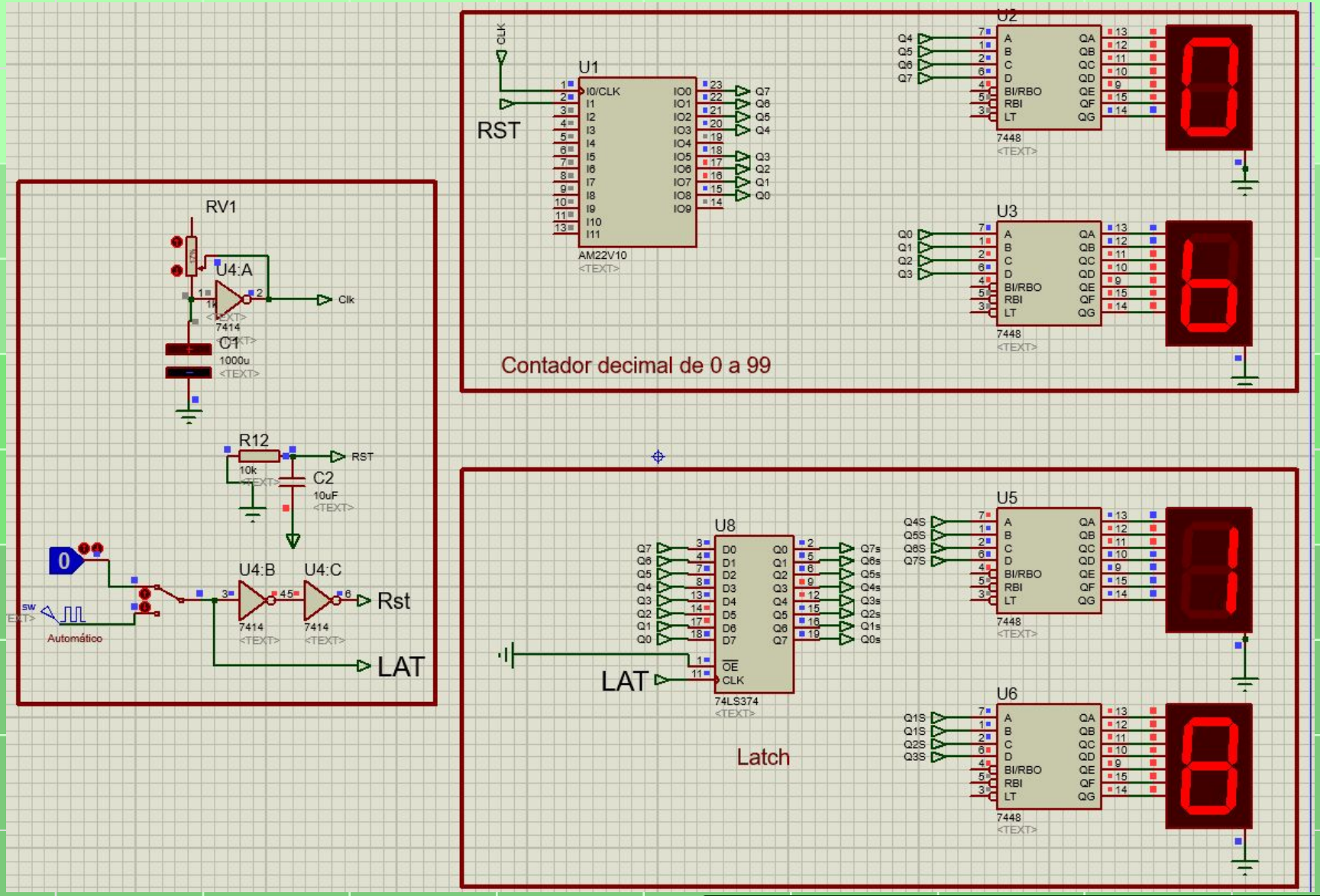



a
f
b
g
e
c
d
dp
Q0 Q1 Q2 Q3 Q4 Q5 Q6
Clk
Contador de Décadas Unidades
Rck
a
f
b
g
e
c
d
dp
Q0 Q1 Q2 Q3 Q4 Q5 Q6
Clk
Contador de Décadas Decenas
Rck

Aplicación


RV1
U4:A
7414
Clk
C1
1000u
R12
10k
RST
C2
10uF
0
U4:B
U4:C
7414
7414
Rst
LAT
Automático
U1
AM22V10
CLK
RST
U2
7448
U3
7448
Contador decimal de 0 a 99
U8
74LS374
LAT
U5
7448
U6
7448
Latch

Aplicación


RV1
U4:A
Clk
7414
C1
1000u
R12
10k
RST
C2
10uF
U4:B
U4:C
Rst
Automático
LAT
U1
AM22V10
CLK
RST
U2
7448
U3
7448
Contador decimal de 0 a 99
U8
74LS374
LAT
U5
7448
U6
7448
Latch

Actividad para el Lunes

- Realizar una reflexión sobre
- Con que se quedan de este curso
- En que ha contribuido en su formación como ingenieros
- Elaborar un escrito