

①
a) Al resetear, el contador se pone a 0000, y luego, con cada pulso de reloj, irá contando para arriba, haciendo un barrido secuencial de la memoria. Los datos que salen de la memoria se cargan en el registro un período de reloj más tarde.

b) Cuando O_7 se pone a nivel alto, activa el \overline{LOAD} del contador, que en el siguiente pulso de reloj se pone a \emptyset .

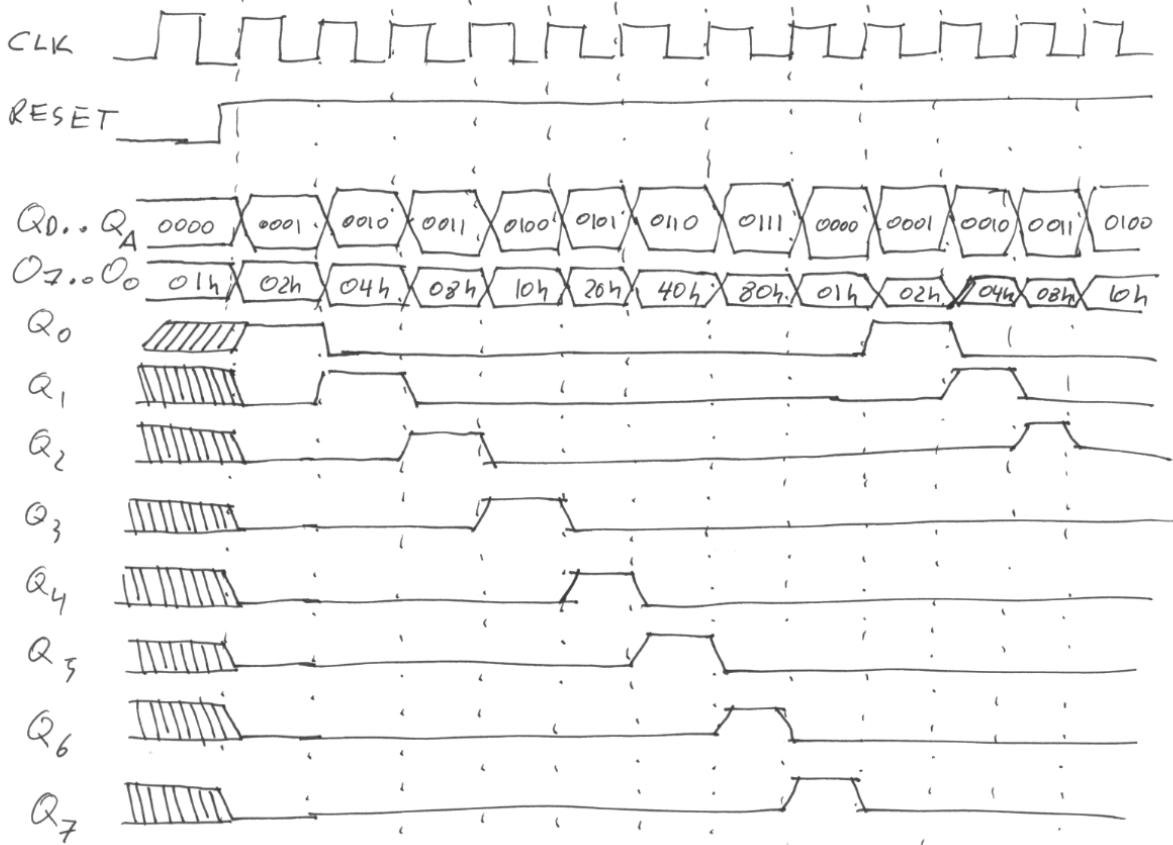
c) Contador: $ENT = EMP = 1$, para habilitar la cuenta.

Memoria: $\overline{OE} = \overline{CE} = 0$, para habilitar el dispositivo y sus salidas

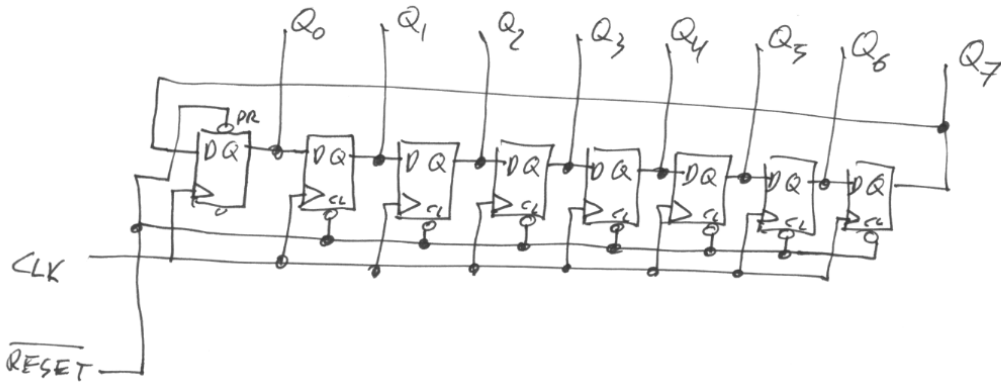
$V_{DD} = 5V$, porque no lo vamos a programar.

Registro: $\overline{OC} = 0$, para habilitar la salida.

d)



e) Si es un contador en anillo. Puede hacerse así:



(2)

a)

$$S = \overline{Q_1} \overline{Q_0} + E$$

⇒ es un autómata de MEALY.

b)

$$D_0 = \overline{Q_0} \oplus E = Q_0 E + \overline{Q_0} \overline{E}$$

$$D_1 = \overline{Q_1} Q_0 \overline{E} + Q_1 E + Q_1 \overline{Q_0}$$

Configurado: Pin 1 = entrada CLK

Pin 2 = entrada E

Pin 19 = salida S (combinacional).

Pin 18 = registro Q₀ (registrado).

pin 17 = registro Q₁ (registrado).

Las conexiones se especifican a la hoja siguiente

c)

$S = \overline{Q_1} \overline{Q_0} + E$
un cero en 110

$D_0 = Q_0 E + \overline{Q_0} \overline{E}$
↓ ↓
unos: x 11 x 00

$D_1 = \overline{Q_1} Q_0 \overline{E} + Q_1 E + Q_1 \overline{Q_0}$
↓ ↓ ↓
unos: 010 1x1 10x

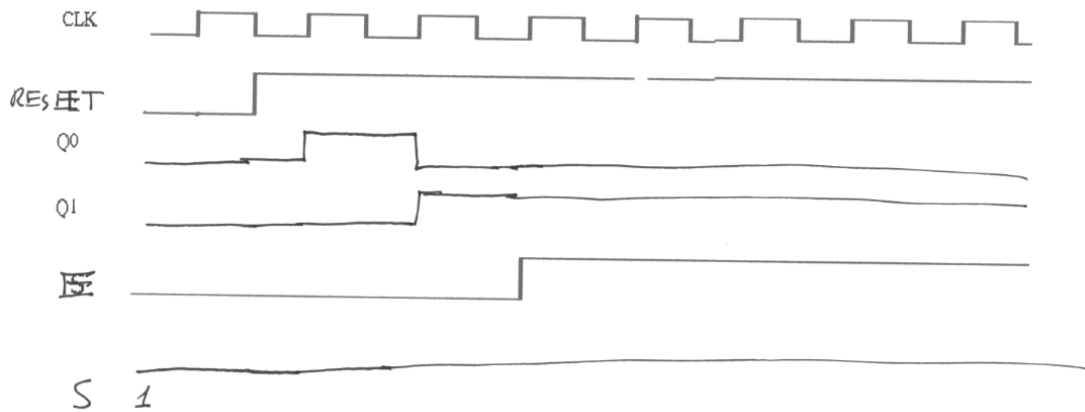
Estado actual $Q_1 Q_0$	Entrada E	Estado siguiente	Salida S
00	0	01	1
00	1	00	1
01	0	10	1
01	1	01	1
10	0	11	1
10	1	10	1
11	0	00	0
11	1	11	1

Desarrollo de Productos Electrónicos
Lógica Digital y Microprogramable

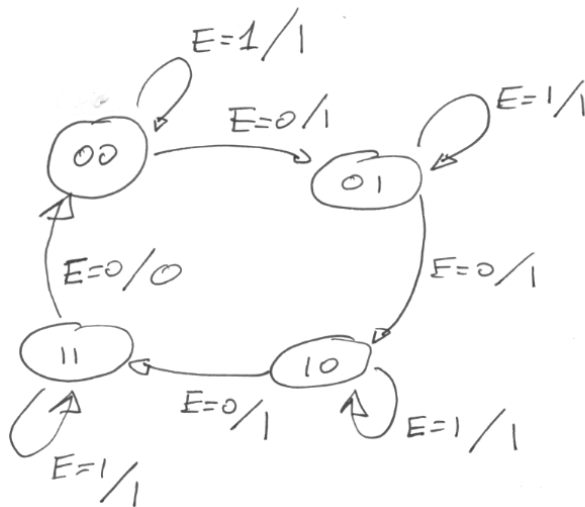
Examen de problemas 2ª evaluación

NOMBRE: _____ APELLIDOS: _____

Problema 2



e) Diagrama de estados



Es un contador de dos bits. La entrada E es una habilitación de cuenta (activa a nivel bajo). La salida S se activa a fin de cuenta (11) si la entrada de habilitación está activada ($E=0$), por lo tanto es una señal de acarreo de cuenta (Ripple clock).

f)

```

MODULE contador
E, S, CLK pin;
Q1, Q0 pin istype 'reg';

equations
[Q1, Q0].clk = CLK;

STATE_DIAGRAM [Q1, Q0];

state [0, 0]:
if E==1 then [0, 0] with S=1
else [0, 1] with S=1

state [0, 1]:
if E==1 then [0, 1] with S=1
else [1, 0] with S=1

state [1, 0]:
if E==1 then [1, 0] with S=1
else [1, 1] with S=1

state [1, 1]:
if E==1 then [1, 1] with S=1
else [0, 0] with S=0

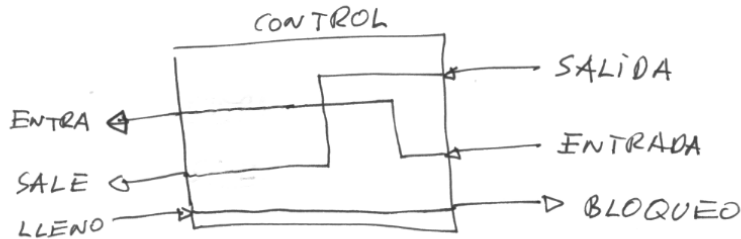
END
  
```

3) * se supone que todas las entradas y salidas son activas a nivel alto *

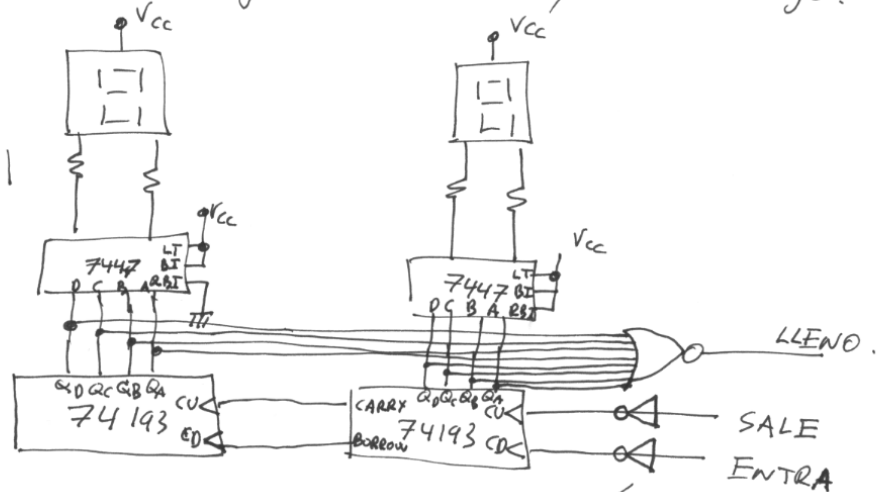
b) Si el aparcamiento no se permite que entren más coches, por lo tanto se bloquea la entrada \Rightarrow LLENO = BLOQUEO.

Cuando hay plazas libres, si entra un coche (pulso en ENTRADA) se hará disminuir el número de plazas libres (pulso en ENTRA) por lo tanto ENTRA = ENTRADA.

Lo mismo ocurre al salir un coche: SALE = SALIDA.



a) Habrá que usar un contador reversible. El 74193 tiene dos entradas de reloj para contar arriba y contar abajo.



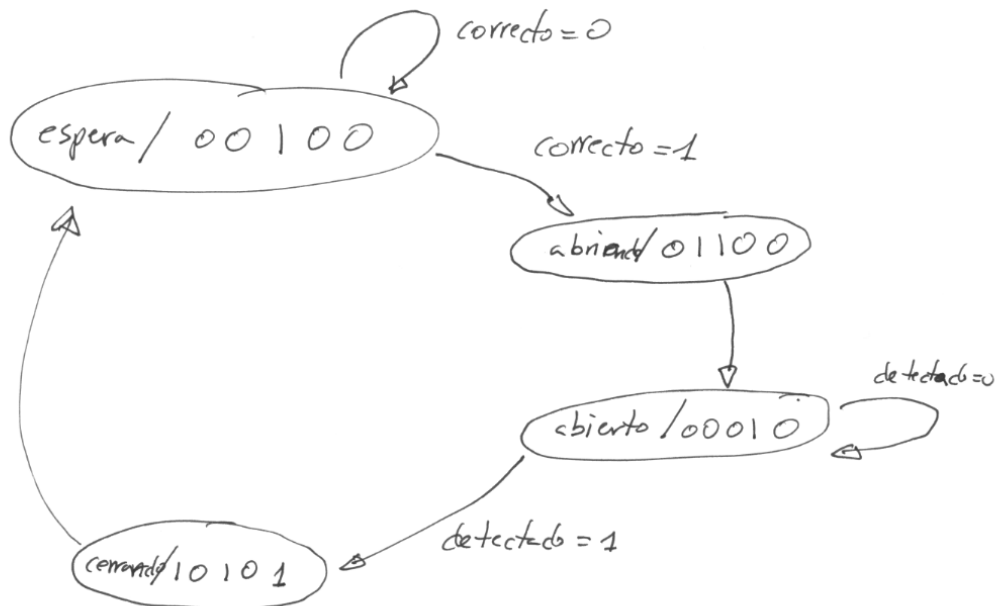
los negadores son porque cuando cuenta para arriba CD debe estar a 1, y viceversa.

④ Se puede realizar con un autómata:

Entradas: DETECTADO CORRECTO

Salidas: cerrar abrir Rs Vs salida

Por MOORE:



Que se puede hacer en una PAL16V8 en ABEL, así:

```
MODULE control
```

```
detectado, correcto, cerrar, abrir, Rs, Vs, salida, CLK pin;
Q1, Q0 pin istype 'reg';
```

```
//definición de los estados
```

```
espera = [0,0];
abriendo = [0,1];
abierto = [1,0];
cerrando = [1,1];
```

```
//array de salidas
```

```
salidas = [cerrar, abrir, Rs, Vs, salida];
```

```
equations
```

```
[Q1, Q0].clk = CLK;
```

```
STATE_DIAGRAM [Q1, Q0];
```

```
state espera:
salidas = [0,0,1,0,0];
if correcto=1 then abriendo;
else espera;
```

```
state abriendo:
```

```
salidas = [0,1,1,0,0];
goto abierto;
```

```
state abierto:
```

```
salidas = [0,0,0,1,0];
```

```
if detectado=1 then cerrando;
else abierto;
```

```
state cerrando:
```

```
salidas = [1,0,1,0,1];
goto espera;
```

```
END
```

2

CLK	1	20	Vcc
correcto	2	19	
detectado	3	18	!Q0
	4	17	!Q1
	5	16	!salida
	6	15	!Vs
	7	14	!Rs
	8	13	!abrir
	9	12	!cerrar
GND	10	11	

SIGNATURE: N/A

```

cerrar    = ( !Q1.Q & !Q0.Q );
abrir     = ( Q1.Q & !Q0.Q );
Rs        = !( !Q1.Q & Q0.Q );
Vs        = ( !Q1.Q & Q0.Q );

salida    = ( !Q1.Q & !Q0.Q );
Q1.D     = ( Q1.Q & Q0.Q
#         !Q1.Q & !Q0.Q ); " ISTYPE 'INVERT'
Q1.C     = ( CLK );
Q0.D     = ( Q1.Q & Q0.Q & correcto
#         !Q1.Q & Q0.Q & detectado ); " ISTYPE 'INVERT'
Q0.C     = ( CLK );

```