

**Comentarios y correcciones a la soluciones de la pregunta de desarrollo de la
Prueba Presencial de Enero 2013.**

1. El diseño del Comparador es correcto.
2. El resultado del contador está bien, pero no responde a la pregunta porque no hace el diseño que se pide en el enunciado.

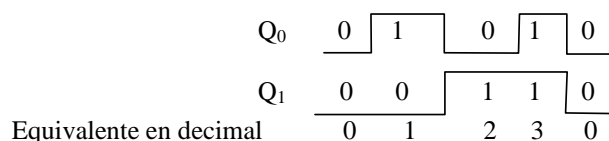
Pone directamente el circuito del contador asíncrono (se supone que lo sabe de memoria) y lo intenta analizar, construyendo “a posteriori” una tabla de verdad que, además, no la explica y hay que descubrir que cuando dice “*siempre cambia en las bajadas de reloj*” se está refiriendo a los cambios de Ck_0 que se producen en el paso de la fila 2ª a 3ª, 4ª a 5ª, etc... y en Ck_1 en las filas 4ª a 5ª, 8ª a 9ª, ...

En este tipo de circuito la forma más sencilla de realizar el diseño es mediante el *razonamiento lógico* a partir de la función que se desea diseñar.

Como queremos diseñar un contador binario sabemos que el circuito deberá cumplir la siguiente tabla de verdad:

Estado INICIAL $Q_1(t) \quad Q_0(t)$	Estado FINAL $Q_1(t+\Delta t) \quad Q_0(t+\Delta t)$
0 0	0 1
0 1	1 0
1 0	1 1
1 1	0 0

Si dibujamos el cronograma correspondiente a estas transiciones entre estados, tenemos:



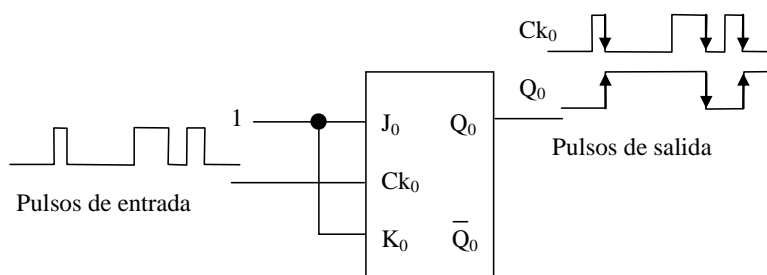
Para el diseño vamos a usar biestables J-K que se disparan en las bajadas del reloj.

Partimos de la base de que en un circuito asíncrono el reloj no es común a todos los biestables, sino que se usa como entrada general de los pulsos de entrada al circuito y, además, sabemos que los biestables T se caracterizan porque, si $T=1$, el biestable cambian de estado cada vez que, por ejemplo, el reloj pasa de alta a baja (podría ser al revés, cambiar al pasar de baja a alta y habría que modificar el diseño) Es decir su tabla de verdad es

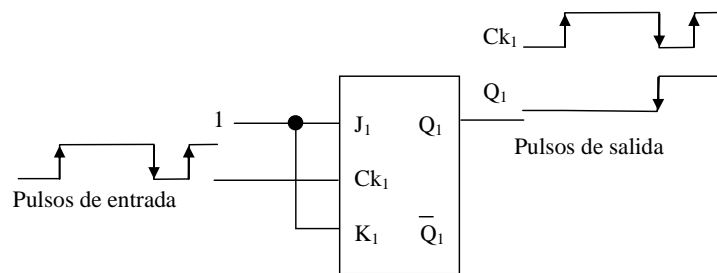
$T=J=K$	Ck	Q_n	Q_{n+1}
1	↓	0	1
1	↓	1	0

Por tanto, a la entrada Ck del primer biestable del contador (el correspondiente al bit menos significativo, Q_0) le entrarán los pulsos procedentes del comparador, que son los pulsos que tiene que contar y, si observamos el cronograma anterior vemos que con las bajadas de los pulsos del primer biestable se producen los cambios en el segundo biestable, por lo que la salida Q_0 del primer biestable deberá entrar en la entrada Ck del segundo biestable.

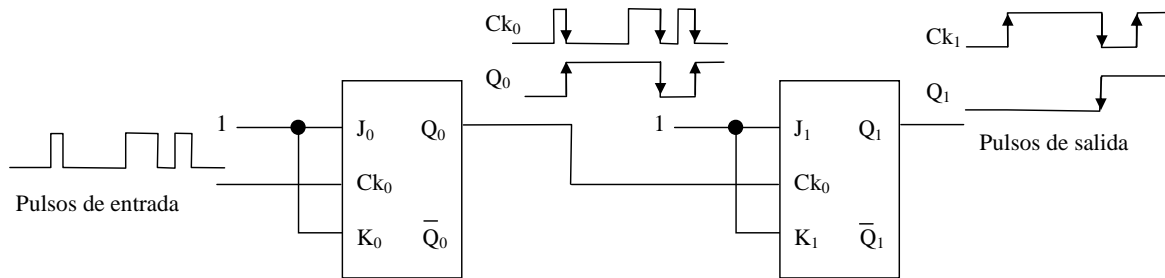
Así, para el primer biestables y ante un tren de pulsos de entrada como el de la figura, resulta:



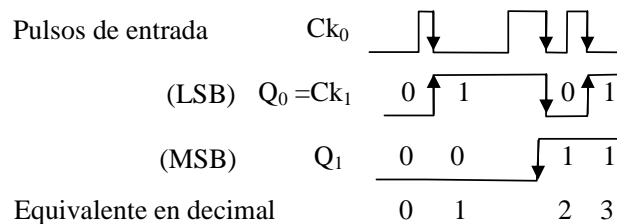
De forma análoga, si ahora usamos los pulsos de salida de este biestable para disparar al biestable correspondiente al bit más significativo, resulta:



Conectando ambos circuitos en serie obtenemos el circuito del Contador Asíncrono de 2 bits.



Como podemos ver en el cronograma siguiente, el circuito diseñado cuenta los pulsos que le entran.



Ambos contadores, el que cuenta las palabras $A > B$ y el que cuentan las palabras $A < B$, son iguales.

- El diseño del Restador es correcto.
- Esta pregunta no la responde. El resultado del restador que ha diseñado está representado en C-2.

En efecto, este restador opera sobre números positivos, pero da lugar a números positivos y negativos ya que cuando el número palabras $A > B$ es mayor que el nº de palabras $A < B$ el resultado de la resta es positivo (los números positivos se representan de igual forma en S-M, C-1 y C-2) pero, cuando el número de palabras $A < B$ es mayor que el nº de palabras $A > B$ el resultado de la resta es negativo (los números negativos tienen distinta representación en S-M, C-1 y C-2).

Por tanto, deberemos ver cuál es el resultado del restador diseñado ante las palabras de entrada, M y m , que generan números negativos. Es decir, si tenemos en cuenta la tabla de verdad del restador completo usado en el diseño, resultan números negativos en la salida en los siguientes casos:

$M (M_1 M_0)$	$m (m_1 m_0)$	Resultado de la resta en decimal	Resultado del restador $C_2 R_1 R_0$	Equivalente en decimal de la salida del restador en la representación en C-2
0 0 (0, en decimal)	0 1 (1)	-1	1 1 1	-1
0 0 (0)	1 0 (2)	-2	1 1 0	-2
0 0 (0)	1 1 (3)	-3	1 0 1	-3
0 1 (1)	1 0 (2)	-1	1 1 1	-1
0 1 (1)	1 1 (3)	-2	1 1 1	-2
1 0 (2)	1 1 (3)	-1	1 1 1	-1

El resultado en decimal (3ª columna) y el equivalente en decimal del resultado del restador (última columna) coinciden. Por lo tanto, el resultado del restador diseñado es en C-2.

Puesto que no se especifica en el enunciado el tipo de restador a diseñar, se puede diseñar otro tipo de restador, por ejemplo, un restador en C-1 o en S-M.

5. El decodificador que ha diseñado sólo es válido cuando el resultado de la resta es un número positivo, pero no lo es cuando el resultado es un número negativo.

Lo sería si el restador lo hubiera diseñado para restar en S-M, ya que en este caso los números positivos y negativos sólo se diferencian en el bit de signo.

En el caso de la representación en C-2 las magnitudes de los números positivos y negativos en binario no coinciden. Por tanto, la tabla de verdad del decodificador a diseñar es:

Número en decimal	C ₂ R ₁ R ₀			Signo								Magnitud							
				a ₁	b ₁	c ₁	d ₁	e ₁	f ₁	g ₁	a ₀	b ₀	c ₀	d ₀	e ₀	f ₀	g ₀		
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0		
+1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0		
+2	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1		
+3	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1		
-1	1	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0		
-2	1	1	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1		
-3	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1		

A partir de aquí obtenemos las expresiones mínimas de los segmentos (g₁, a₀, b₀, ..., g₀) en función de C₂, R₁ y R₀.

Como la configuración C₂ R₁ R₀= 100 no se va a dar, la podemos considerar como indiferente y usarla para obtener las expresiones mínimas de los segmentos.
