

INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

**MT 4001
Electrónica Digital**

Tarea de diseño 2

David Rodríguez Camacho – 2019024546
Andrew Eliécer Vargas Puffenberger – 2020425926
Emmanuel Naranjo Blanco – 2019053605

Profesor: Ing. Carlos Salazar García, MSc.

Semestre II - 2021

I. Desarrollo del diseño

Para la solución del problema propuesto, se dividió en distintas etapas más simples para posteriormente realizar un ensamble que cumpla con las especificaciones de diseño. Se presenta a continuación el procedimiento para llegar al circuito diseñado.

i. Requerimientos y restricciones del problema.

Se requiere un circuito que cuente 99 milisegundos una vez que haya recibido una señal de activación, el cual tiene tres entradas de 1 bit y una salida de 8 bits como se muestra en la Figura 1.

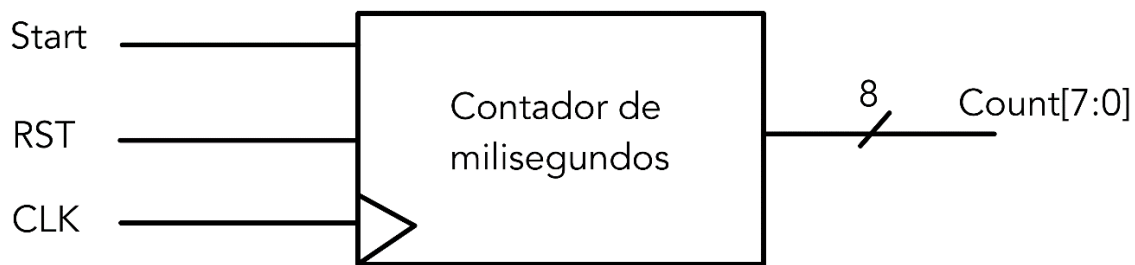


Figura 1. Diagrama de entradas y salidas del sistema a diseñar.

Para esto se tiene un reloj que opera a una frecuencia de 100 MHz, una señal Start que consta de un pulso con duración de un periodo del reloj y un reset asíncrono que se activa en alto. De este modo, cuando ocurra un flanco positivo del CLK y Start esté en alto, el sistema contará los milisegundos.

Como resultado se tiene una salida Count en BCD que permitirá llevar el conteo desde 0 hasta 99.

ii. Pulso de inicio Start.

Como el pulso de Start solo se recibe una vez, se decidió guardar esta pulsación en un Flip Flop D para que el contador se ejecute y comience el funcionamiento del circuito diseñado. Para esto se conectó la salida Q de este FF-D al enable del primer contador de 17 bits y la señal Start al enable del FF-D, cuyo valor de D es 1. De este modo, cuando el Flip Flop reciba el pulso de start, se va a activar, y el valor de D va a dejar que el contador empiece a funcionar (enable = 1). Este se muestra en la Figura 2.

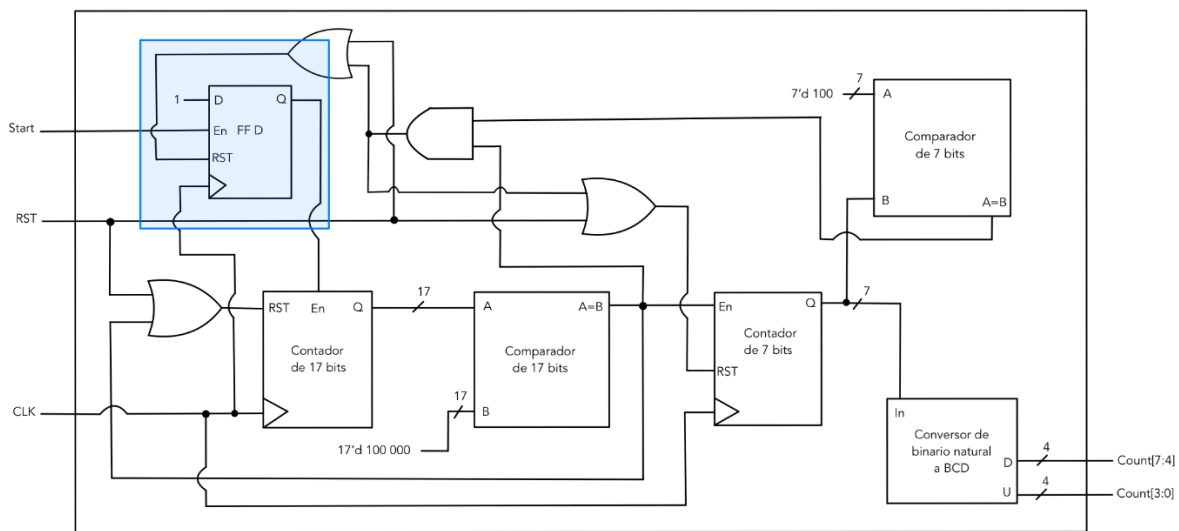


Figura 2. FF-D utilizado para la pulsación de inicio.

iii. Bloques Contadores y Comparadores.

Dado que se requiere contar milisegundos, y el periodo de la señal de reloj es de 100 MHz, es necesario aplicar la estructura de un contador junto con un comparador de N bits ambos, para dividir la frecuencia del CLK y aplicarla en nuestro diseño contador de ms.

Para esto se aplicó la relación

$$10ns \cdot X = 1ms$$

$$X = 100\,000$$

Donde el periodo del CLK es 10 ns y se requiere pulsaciones de 1 ms, lo cual indica que se debe utilizar un contador y comparador de 17 bits para que el contador pueda contar hasta $131\,071_{10}$. Esta salida corresponde a una entrada del comparador, la otra entrada es el número binario de 100 000. Una vez ambas entradas ($A=B$) sean iguales, se generará un pulso a la salida que activará el enable de un segundo comparador y reiniciará el primer contador para que cuente nuevamente desde cero.

El segundo contador se trata de un contador de 7 bits, que se apoyará de otro comparador de 7 bits para realizar el conteo desde 0 hasta 99, el cual es el objetivo del proyecto.

Ambas etapas se muestran en la Figura 3.

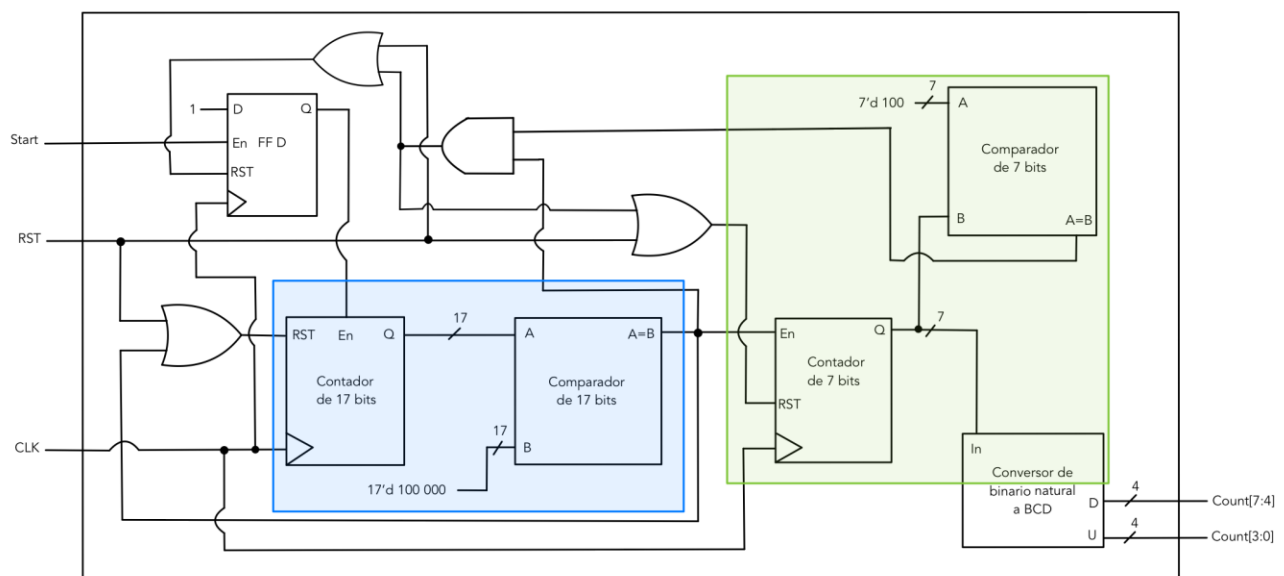


Figura 3. Uso de contadores y comparadores.

iv. Uso del RST.

La señal de reset es asíncrona al sistema, es decir, es la señal de mayor prioridad del circuito secuencial. Esta se activa en alto, que significa que cuando su estado es 0, el circuito opera normalmente, y cuando su estado es 1, automáticamente el sistema se reinicia a un estado de cero en la salida, sin importar si había o no un proceso de conteo en camino.

Además, una vez que el conteo de milisegundos llegue a 99, se activará el RST para comenzar en el estado cero y esperar a que se active nuevamente la señal Start según el pulso del CLK. Para esto se utilizan compuertas lógicas conectadas a las entradas RST de cada módulo y las respectivas salidas de los comparadores que indican mediante un 1 lógico cuando $A=B$ en cada caso.

v. Conversor de binario a BCD.

Dado que la salida del circuito se debe mostrar las unidades y decenas en BCD e internamente se está trabajando en binario natural, se requiere de un conversor de binario natural a BCD. Para esto se apoya del bloque implementado en la Tarea de Diseño 1 (Figura 4) cuya estructura interna se calculó bajo la teoría expuesta por Education @ B. Tech (2016), el cual desplaza de 1 bit en 1 bit a la derecha y comparando grupos de bits en las secciones unidad o decena. Donde si estos superan o igualan a 5_{10} se suma 3_{10} a estos bits, de lo contrario se mantiene todo igual y se sigue realizando el desplazamiento. (*Tarea de diseño 1, 2021*)

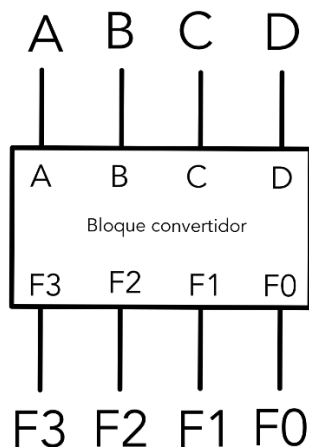


Figura 4. Bloque base para realizar la conversión de binario a BCD.

Tomado de *Tarea de diseño 1* (p.9).

Tomando las ecuaciones de la Tarea 1:

$$F3 = A + BD + BC$$

$$F2 = AD + BC'D'$$

$$F1 = B'C + CD + AD'$$

$$F0 = AD' + A'B'D + BCD'$$

Para este módulo se usan las mismas compuertas lógicas que están contenidas en el módulo de conversión de la tarea 1 (presentadas en la figura 3). Para el caso de 8 bits, se recomienda realizar la conexión similar a la presentada en la Figura 6, lo que varía es que los bits de las centenas en este caso serían 0 porque no se va a trabajar con números entre 0 y 98. En la figura 7, en la entrada se tienen ocho bits y las salidas se dividen en dos, unidades (Y [3:0]) y decenas (Y [7:4]), todos estos números en sistema BCD. Las cuales, para el caso específico del presente proyecto se refieren a los bits de la salida Count. Es decir, Count [7:0] = {Y [7:4], Y [3:0]}.

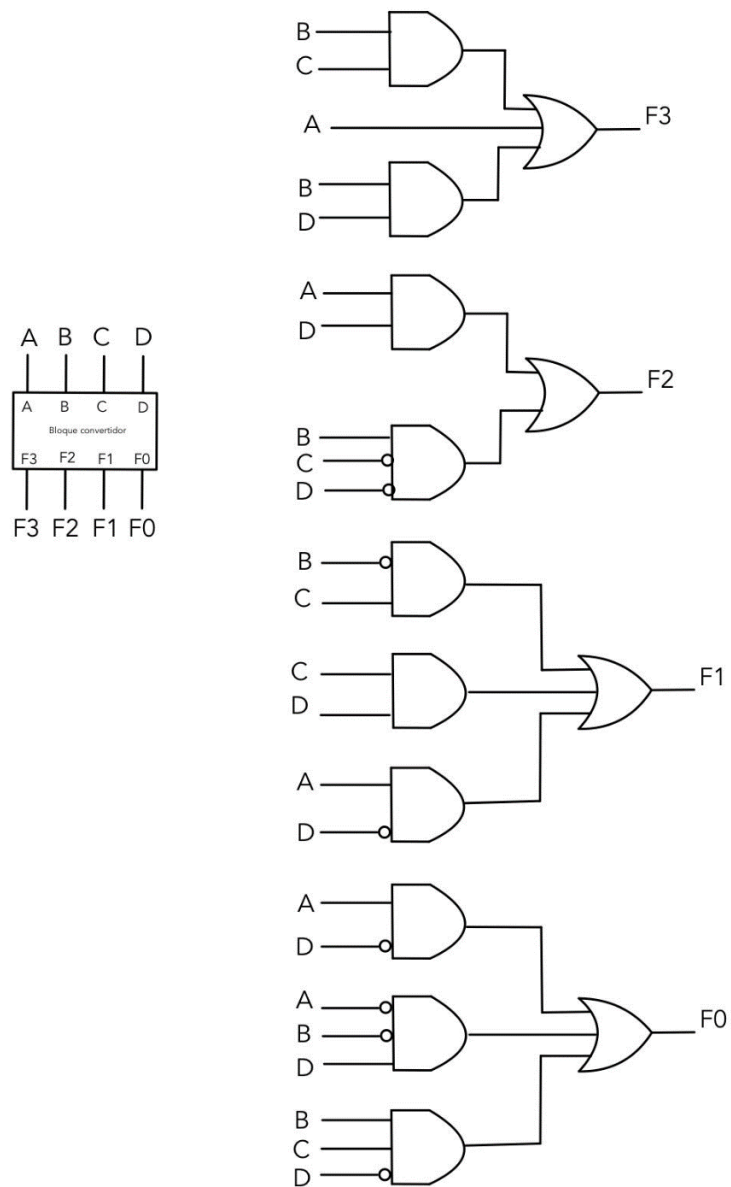


Figura 5. Diseño para el módulo de conversión de la figura 2. Tomado de *Tarea de diseño 1* (p.11).

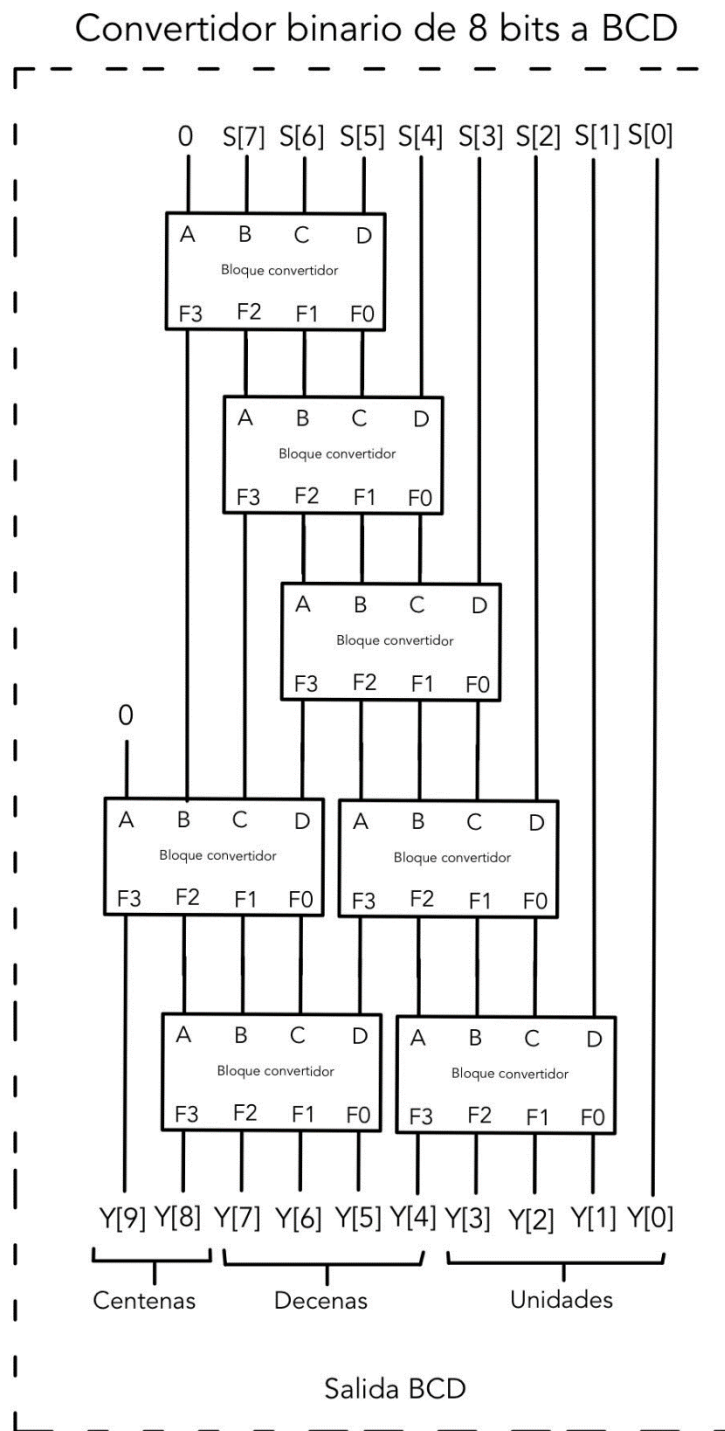


Figura 6. Diseño para el conversor de binario a BCD de 8 bits. Tomado de *Tarea de diseño 1* (p.12).

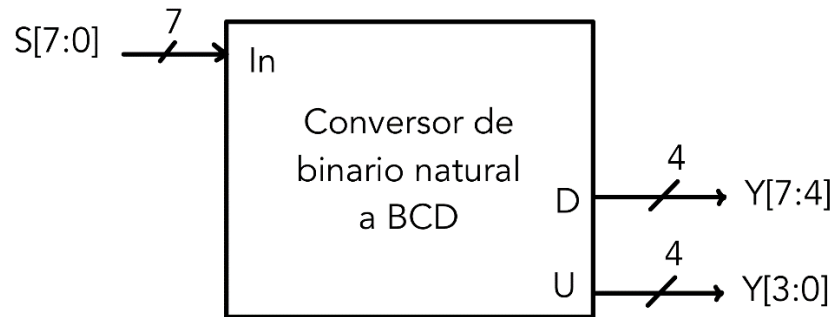


Figura 7. Diseño de bloque para el conversor de binario a BCD de 8 bits.

vi. Implementación.

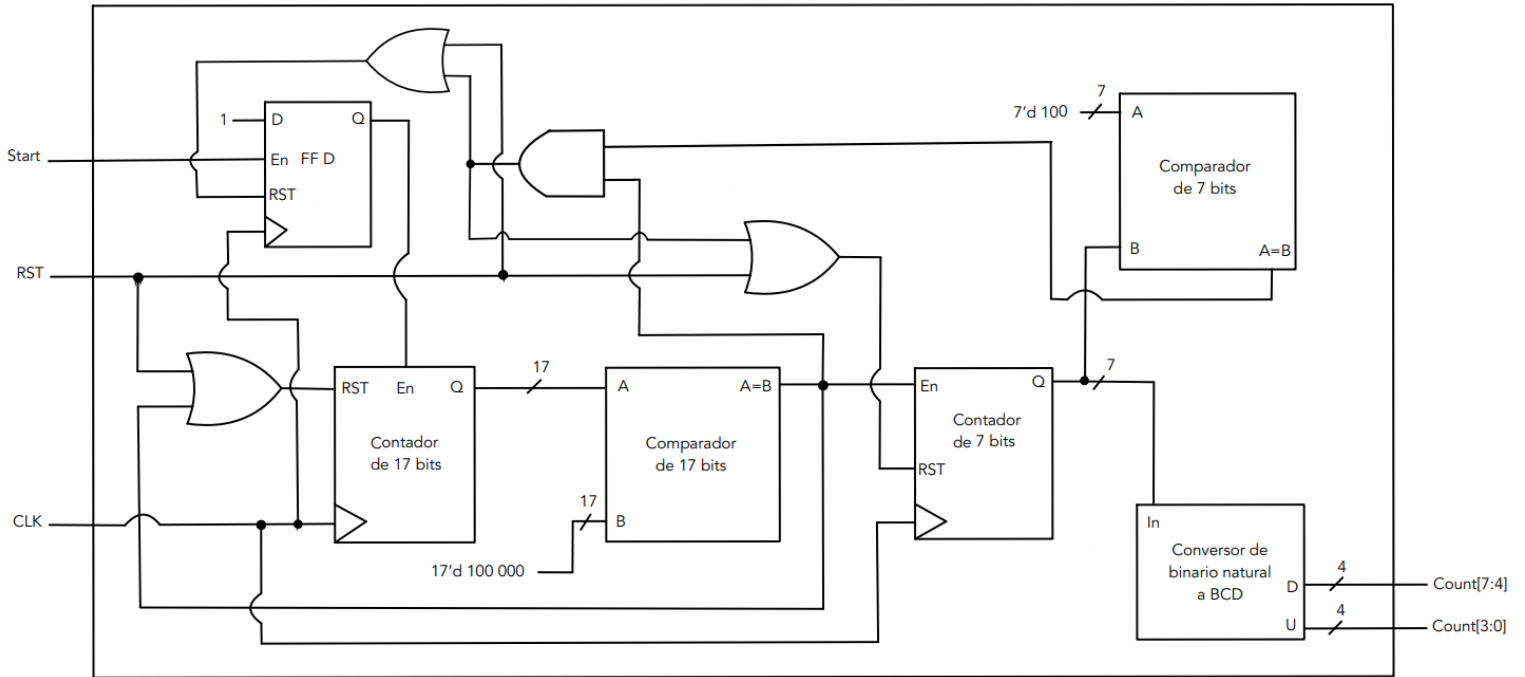
Por último, se implementó cada componente requerido en Verilog, se simularon los componentes para verificar que trabajen de forma esperada y se empleó la simulación del diseño final. Los archivos .v utilizados en la implementación se adjuntan con el presente trabajo, en la carpeta *implementation*.

Cabe destacar que durante esta etapa, se encontró un desfase de dos ciclos de reloj, por lo que se realizó una modificación en el comparador de 17 bits señalado en la Figura. En esta etapa se requiere hacer conteos de cada milisegundo, razón por la cual se hizo una comparación con el número 100 000, como se explicó en la sección anterior. No obstante, se hizo la comparación con 99 998, debido a que el conteo de los flancos de reloj se realiza desde 0 hasta 99 999, además de que una vez que comparador se activa, un flanco de reloj después se reinicia el conteo.

De este modo, al tomar estas consideraciones y hacer la comparación con 99 998 en lugar de 100 000, se logró eliminar el desfase antes presentado para que el sistema funcionase sin ningún problema

II. Diseño Final

Se presenta a continuación el circuito final diseñado.



Referencias

Education @ B.Tech. (2016, 14 de mayo). *Lesson 32 Binary to BCD Converter*. [YouTube].

<https://www.youtube.com/watch?v=sdH2lKiuCjg>

Rodríguez, D. Vargas, A. Naranjo, E. (2021, 5 de octubre). *Tarea de diseño 1*. [Documento PDF en TEC Digital].

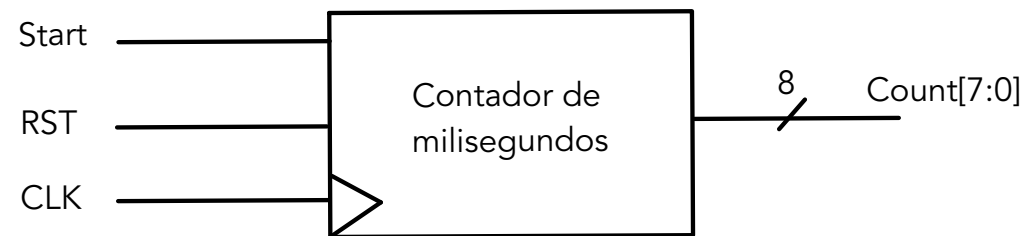


Diagrama de 1 bloque

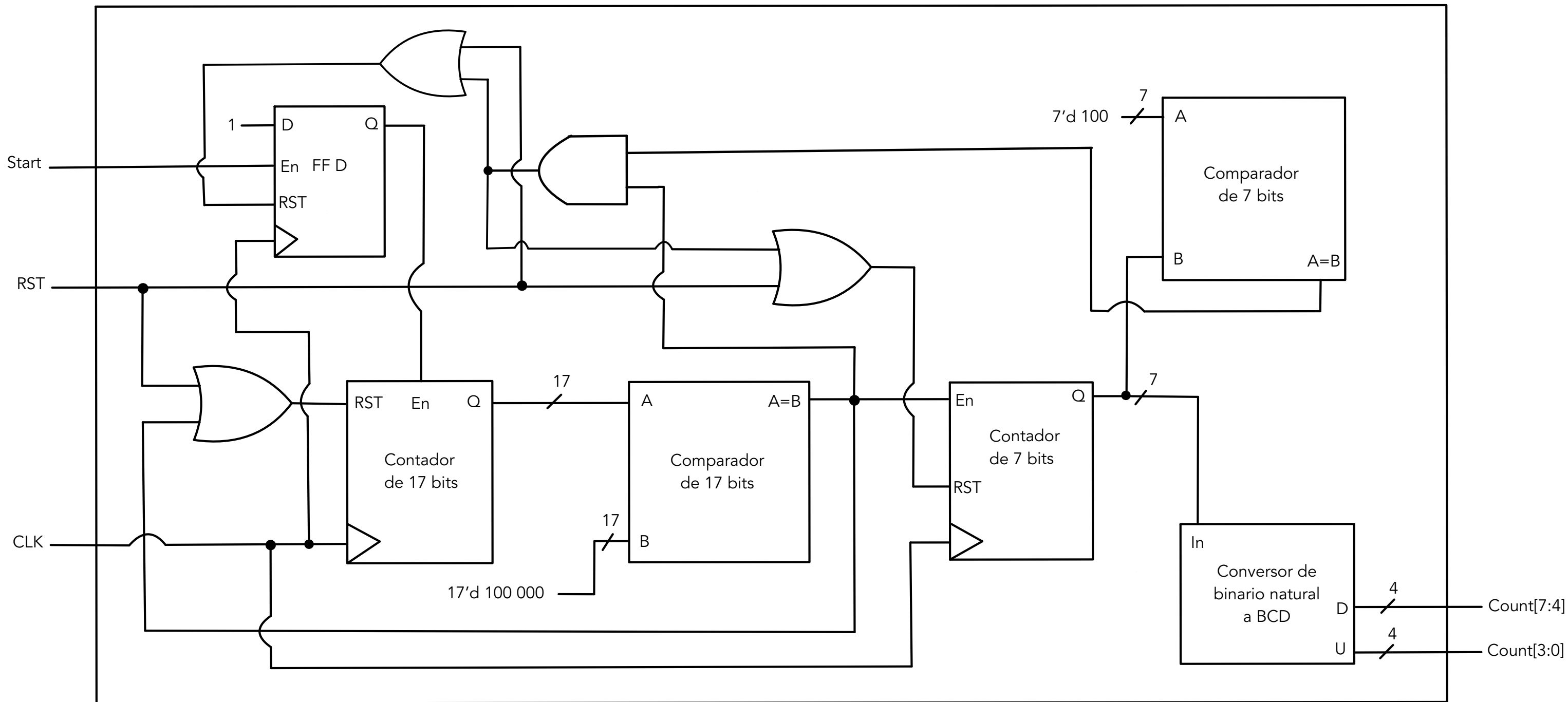


Diagrama de 2 bloque

