# Laboratorio 07: Transmisor serial asíncrono

Por:

Ian Gabriel Cañas Fernández, 1092228

.

.

# Transmisor serial asíncrono

Transfiriendo bytes desde la FPGA hasta una terminal.

# Objetivos:

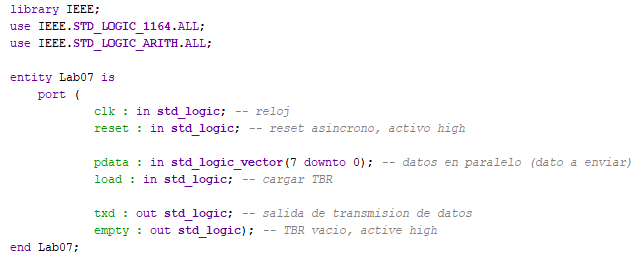
* Diseñar un sistema de transmisión asíncrona de datos seriales.

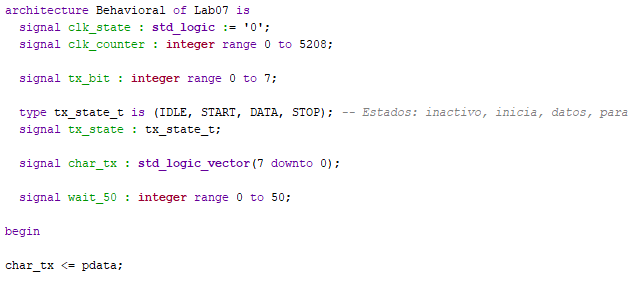
# Procedimiento:

En el presente laboratorio estaremos utilizando el puerto UART de la tarjeta de desarrollo para la transmisión de datos mediante una forma serial en el momento solicitado. Se estará utilizando una máquina de estados para ello y reconociendo la introducción de los datos. Estaremos trabajando con la codificación ASCII de 8 bits para los caracteres introducidos.

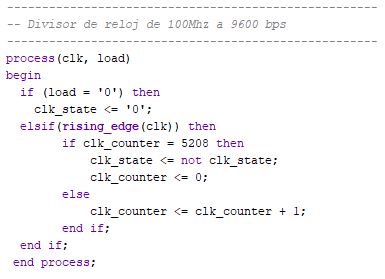
## El transmisor serial asíncrono

Empezamos generando los puertos, funciones y señales necesarias para el laboratorio:

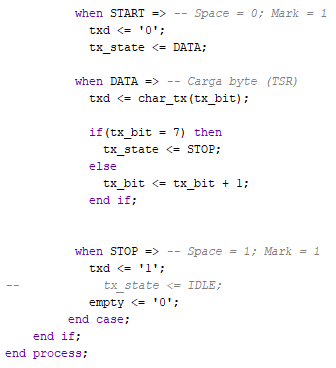
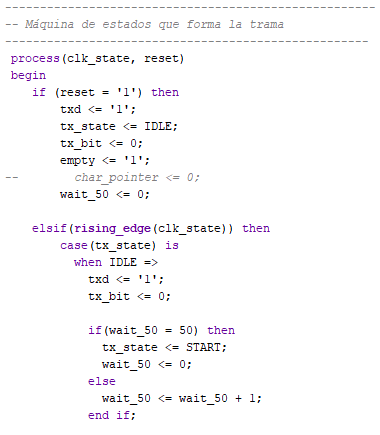




Generamos, entonces, una función que genera la frecuencia deseada que va a representar la velocidad de transmisión de datos, mediante el uso de la siguiente ecuación:



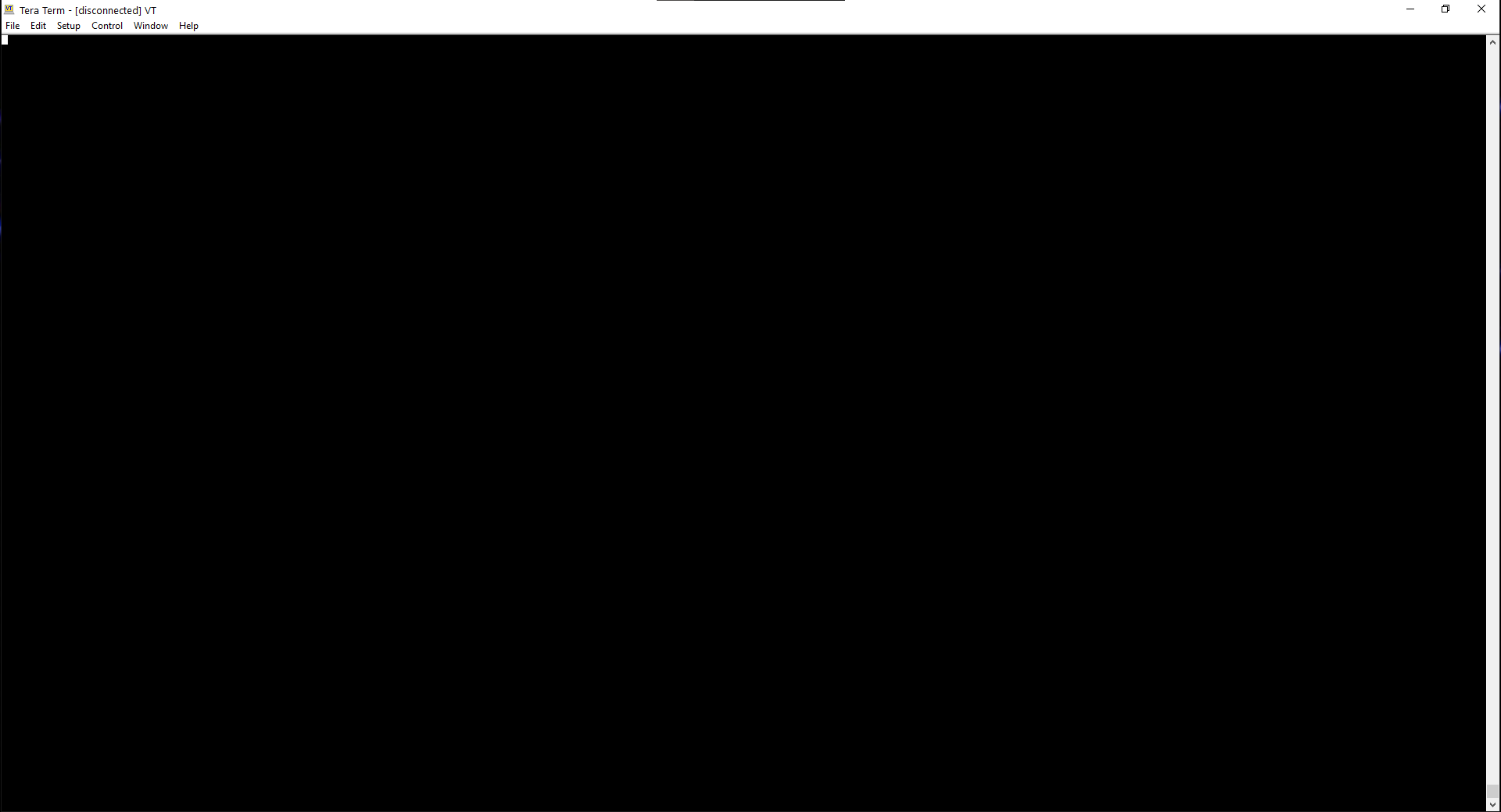
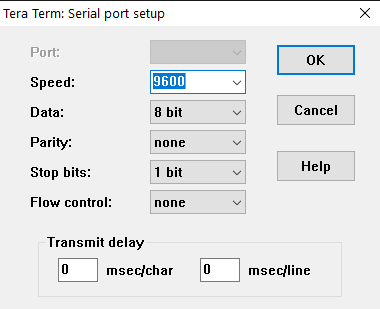
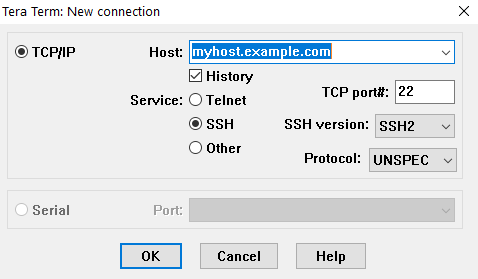
Obsérvese que no siempre se estará generando dicha frecuencia, sino que estamos dependiendo de que se esté pulsando el botón que active la señal de carga (load), pues en dicho caso se activa el siguiente proceso.



Como se puede observar, mientras no se active la señal de RESET, no se va a limpiar el TBR, por lo que no se va a poder cargar otro byte a la terminal. Incluso, si no se activa LOAD, tampoco va a generarse la frecuencia de 9600 baudios, por lo que el proceso estaría haciendo nada. Sin embargo, al pulsarse dicho botón, sucedería la siguiente frecuencia:

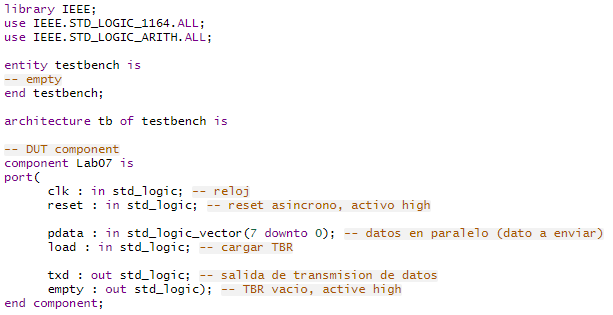
1. Se carga un ‘1’ que pueda garantizar que la terminal detecte un *falling\_edge* en el siguiente estado.
2. En el estado START se carga un ‘0’ para generar el *falling\_edge* recién mencionado.
3. En DATA, se carga el primer bit de la secuencia de datos, y se lleva un conteo. Se prosigue con el segundo bit, así hasta llegar al último.
4. En STOP, se carga el bit ‘1’ que indica que la secuencia ha sido cargada.

Mediante este simple código, podemos fácilmente cargar caracteres a la terminal. En este caso específico estaremos trabajando con la terminal Tera Term.

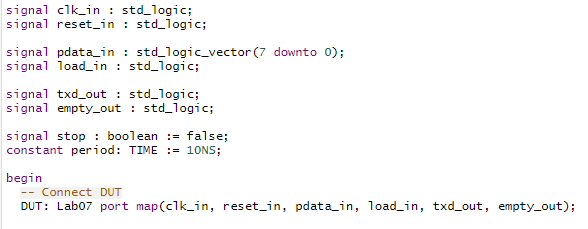


Además, se ha generado un testbench para confirmar la evolución de las señales a lo largo del ciclo, precisamente para saber si se está logrando la secuencia esperada en el orden solicitado.

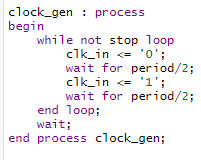
Empezamos introduciendo los puertos y librerías del banco de pruebas:



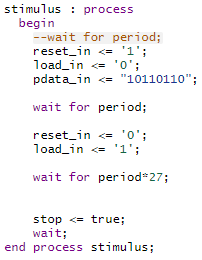
Generando, a continuación, las señales a simular y asignándolas en orden a sus respectivos puertos.



Luego, generamos un proceso que simule el comportamiento cíclico del cristal que funciona como reloj interno de la tarjeta de desarrollo.

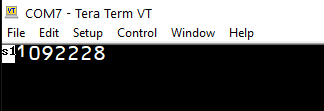


Finalmente, generamos un proceso que empiece reiniciando las señales para preparar la carga de información y luego cargar los ocho bits que serían representados mediante switches, para luego cargar la información mediante la señal de load.

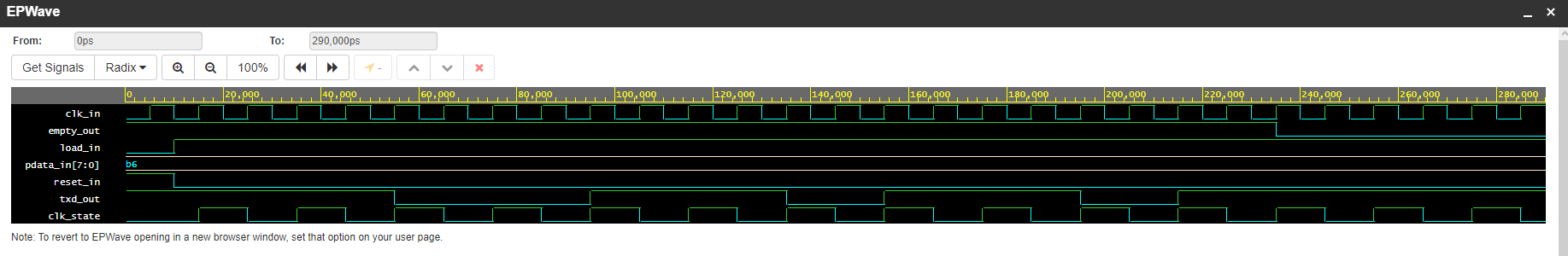


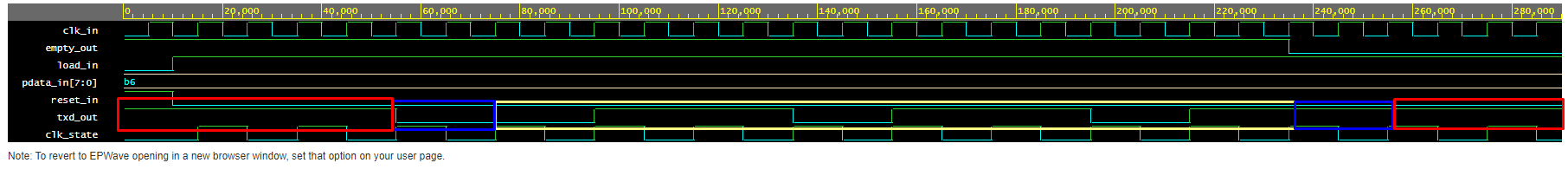
# Resultados:

Hemos visto cómo mediante una simple combinación de ceros y unos en un bus de switches, hemos podido cargar información de caracteres a una terminal sin mucha complicación mediante tan solo un puerto de salida; utilizando, claramente una lógica secuencial y codificada que activaría su detección.



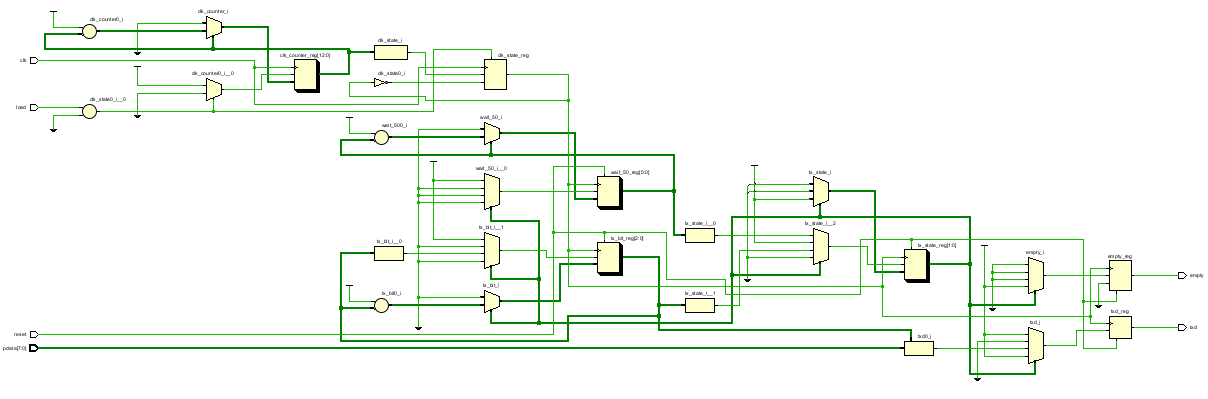
En el testbench adaptado para no trabajar con muchos ciclos de reloj obtenemos el siguiente resultado:

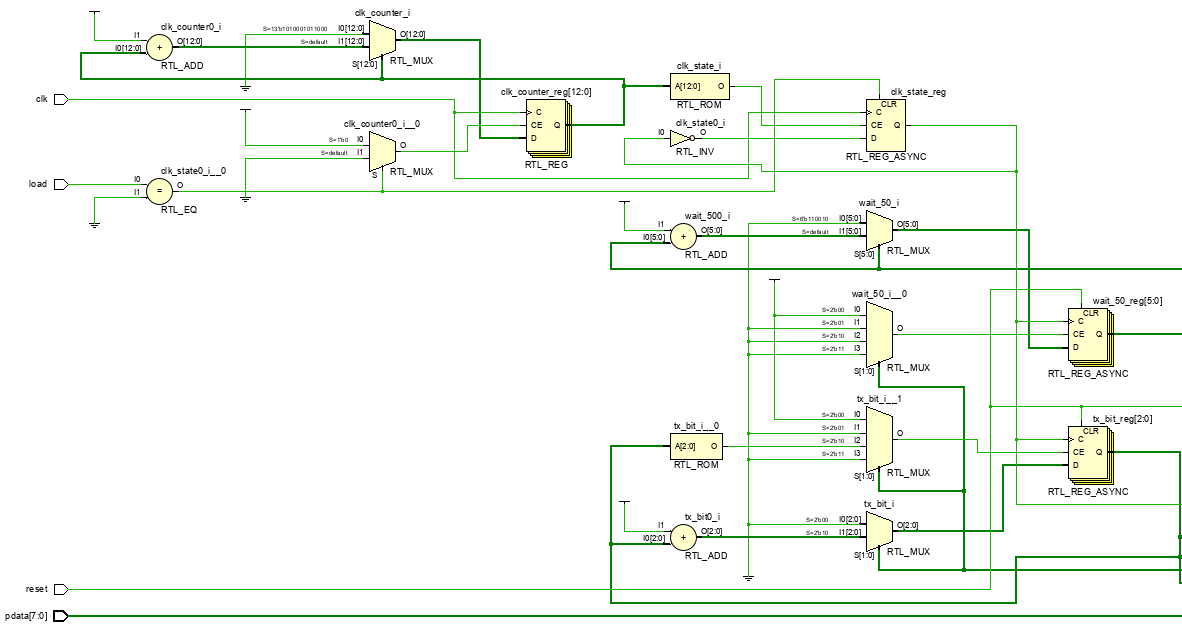


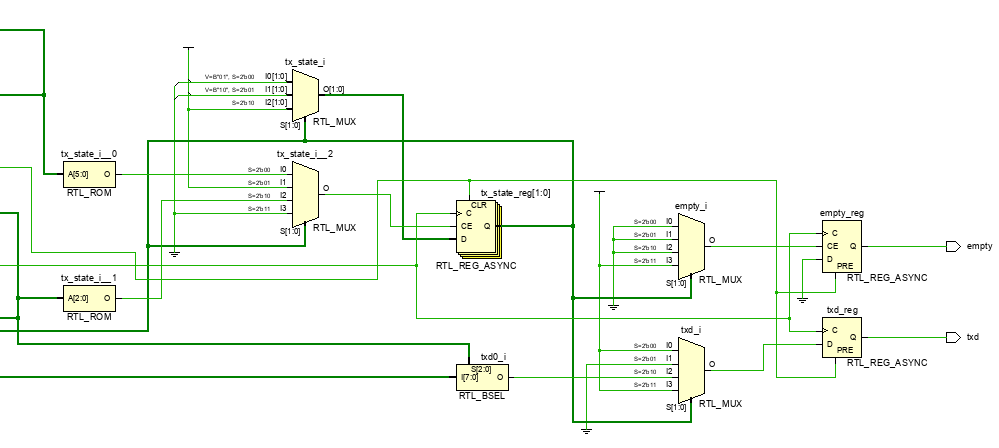


Se puede observar a simple vista las secciones de la señal, empezando por el descanso en las secciones rojas, observando los bits de START y STOP, respectivamente ‘0’ y ‘1’ y, finalmente, la sección marcada de amarillo que va leyendo y asignando cada bit del vector de derecha a izquierda, como corresponde su lectura.

Además, obtenemos el siguiente diagrama del hardware descrito:







# Análisis:

Se pudo comprender el funcionamiento de la trasmisión serial y cómo esta asegura una correcta transmisión mediante bits que anuncian la llegada de un bus de datos. Además, mediante la simulación, se llega a observar muy detalladamente la señal transmitida y cómo esta responde perfectamente a la combinación deseada.

Finalmente, se pudo percibir cómo se transmite a la perfección el carácter enviado sin fallo alguno, a pesar de no estar trabajando exactamente a la frecuencia solicitada, pues dicha exactitud, con el reloj dado no es posible, sino aproximable. Lográndose, incluso con una incertidumbre menor al 1 %.