Implementación de un filtro pasabajos - 2019
Práctica de laboratorio $N°3$
Electrónica Digital

Michel Gartner
michelgartner8@gmail.com

22 de octubre de 2019

Índice

1. Especificación de requerimientos del sistema

Diseño e implementación de un filtro pasa bajos para obtener la frecuencia de voz utilizando una placa Nexys3 Spartan6 XC6LX16-CS324. Los datos de la voz se toman de un micrófono conectado a la placa de desarrollo a través de un conector de expansión pMod. La voz filtrada debe enviarse mediante el dispositivo UART a la PC para su análisis.

La frecuencia de muestra es de 20kHz. Se implementa un filtro FIR de orden 20 con una frecuencia de corte de 3kHz.

Este informe presenta el núcleo del trabajo en tres secciones distintas, correspondientes a las tres etapas del desarrollo del filtro. En primer lugar se hizo la implementación y verificación del módulo UART. En segundo lugar se hizo la integración del módulo pMod con el módulo UART. Por último se realizó la integración completa del filtro con todos los módulos necesarios.

2. UART

El módulo UART se utiliza para enviar datos de la placa hacia la pc.

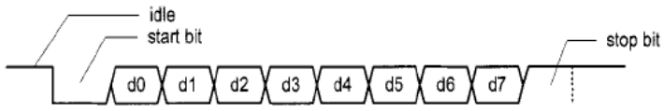

Figura 1: Esquema de la señal de salida del módulo UART

Realizamos una implementación para enviar 8 bits de datos, con 1 start bit en bajo y un stop bit en alto, como se muestra en la figura 1, con una frecuencia de 20 kHz.

Para que no se superpongan dos muestras consecutivas el baud rate debe ser mayor a 200.000 baudios. El menor valor estándar de baud rate que cumple esta condición es 460.000 baudios, por lo que se busca ese valor.

En la implementación provista por la cátedra el baud rate se indica de forma indirecta a partir del valor del baud rate divisor (DVSR) usando la fórmula.

$$DVSR = \frac{100M}{16 * baudrate}\,. \tag{1}$$

Se fijo un valor de DVSR=14 y, por lo tanto, el valor de baud rate implementado es de 446429 baudios o 2.24 microsegundos por bit.

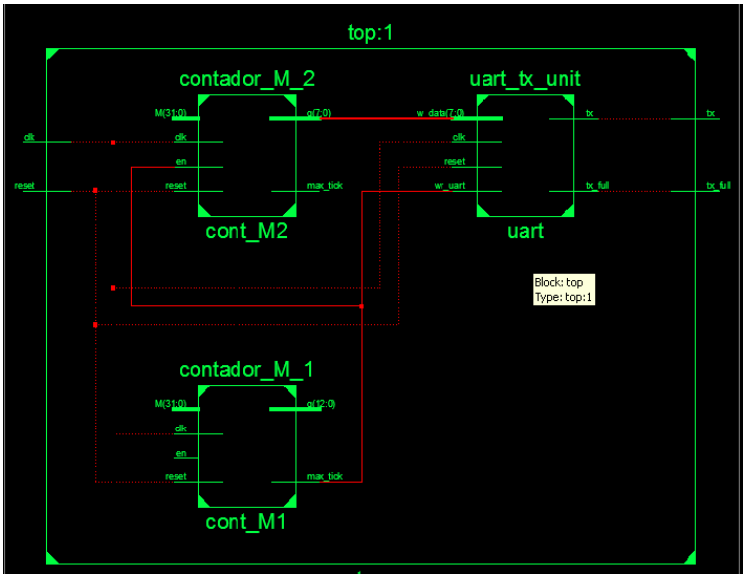

Figura 2: Diagrama de bloques de la lógica del módulo 'top' para la implementación del módulo UART aislado

En la figura 2 se muestra el diagrama del top para la integración del módulo UART. El submódulo 'contador M 1' genera una pulso con una frecuencia de 20 kHz y es enviado a la entrada 'w uart' para indicar que se inicie la transmisión de bits. El submódulo 'contador M 2' se utiliza para generar una señal de entrada que va a ser verificada posteriormente. En este caso es un contador de 0 hasta 1023, que es el rango que se puede representar con 4 números en hexadecimal.

2.1. Testeo

Para verificar el correcto funcionamiento del módulo UART se realizó un test-bench para analizar y comprobar que las señales son las esperadas y luego se probó con la placa y se verificó con un software en la pc que llegaban los datos esperados.

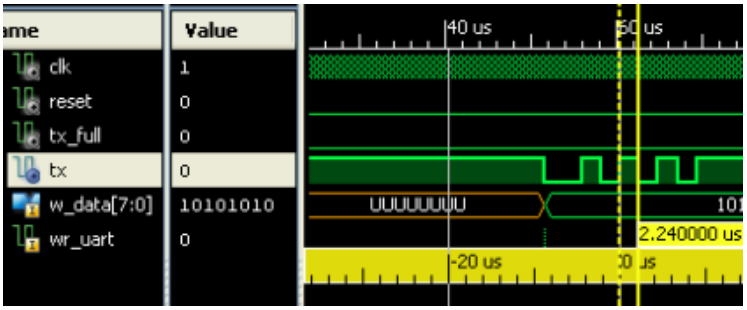

Figura 3: Test bench del módulo UART. Se oberva que la duración de un bit de salida es de 2.24 μs

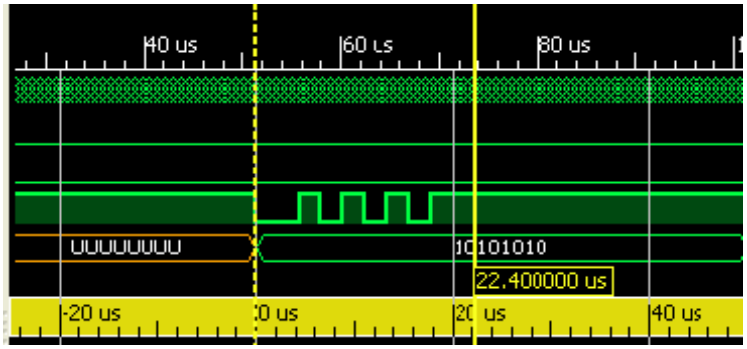

Figura 4: Test bench del módulo UART. Se oberva que la duración de los 10 bits de salida es de 22.4 µs. Además se observa el start bit en bajo y el stop bit en alto.

En las figuras 3 y 4 se observa el test bench para la implemetación del módulo UART. Se observa que se envian 10 bits de 2,24μs , donde el start bit es un 0, luego están los 8 bits de 'w data' y luego el stop bit que es un 1. Los bits de 'w data' se envían desde el menos significativos hacia el más significativo.

Luego se realizó la prueba en la PC utilizando el software CuteCom y se verificó la recepción initerrumpida de valores entre 0 y 1023.

3. Integración PmodMic - UART

El siguiente paso es integrar el PmodMic con el módulo UART. En este paso se busca leer una señal con el módulo PmodMic y enviar los datos hacia una pc con el módulo UART.

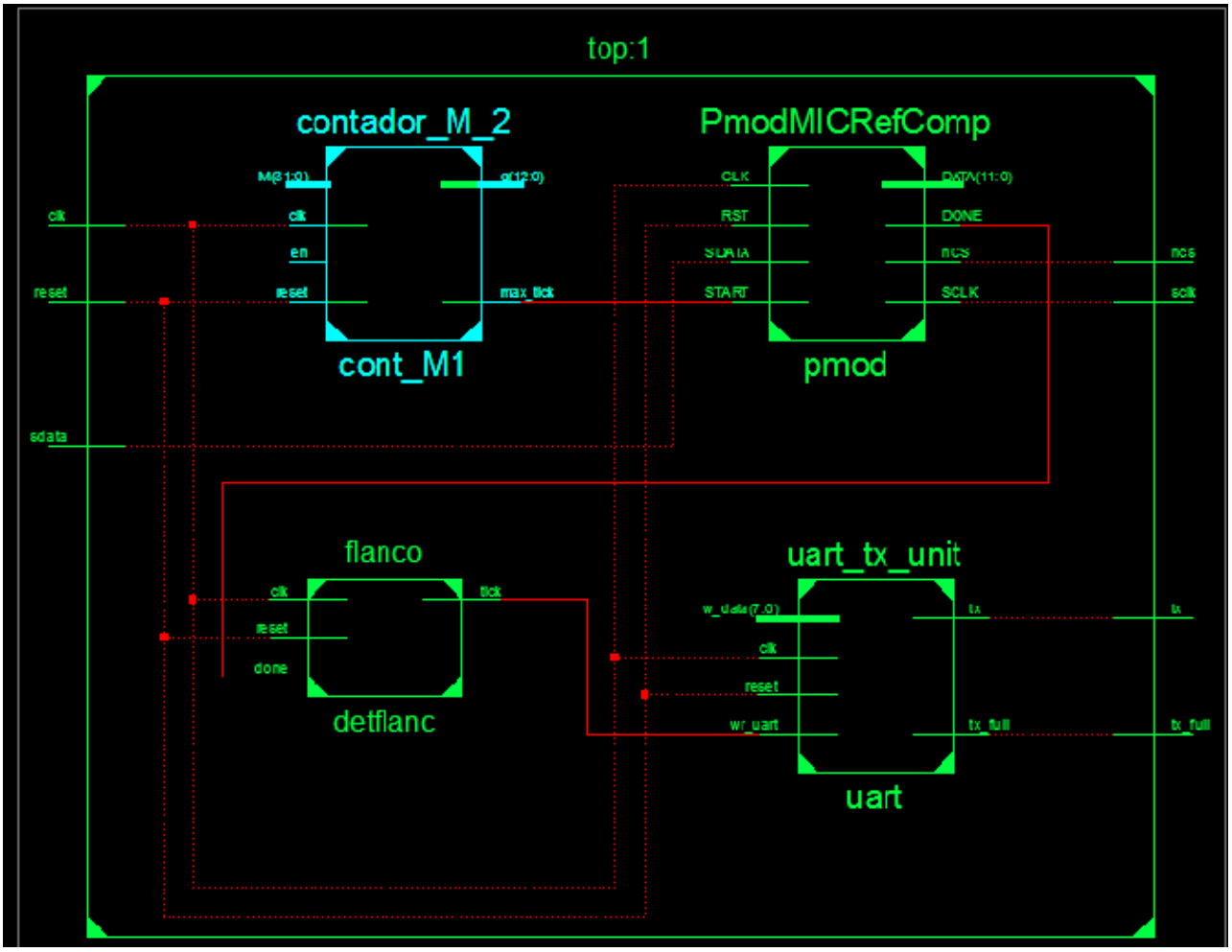

Figura 5: Diagrama de bloques de la lógica del módulo 'top' para la implementación de la integración PmodMic - UART

El diseño del módulo top de esta implementación se muestra en la figura 5. Nuevamente, 'contador M 2' se utiliza para generar un pulso con una frecuencia de 20 kHz. Este pulso se utiliza para dar inicio al sampleo del submódulo 'PmodMICRefComp'. Una vez que se termina de registrar los datos, la señal done pasa de 0 a 1. Para detectar este cambio se utiliza un detector de flanco en el módulo 'detflanc'. Por otro lado, los 8 bits más significativos de la señal 'data' se mandan al submódulo UART para ser transferidos. El pulso generado por el detector de flancos se manda al submódulo uart para indicar que inicie la transferencia.

3.1. Submódulos

3.1.1. PmodMic

El módulo PmodMic transforma la señal del micrófono en serie en una señal en paralelo. Realizamos una implementación para samplear con una frecuencia de 20 kHz. Utilizamos solamente los 8 bits más significativos.

El módulo PmodMic tiene un reloj interno de 12.5MHz. Por ello, para asegurarnos que el pulso de 'start' sea detectado, su duración debe ser de almenos 1/12.5MHz. Dado que el reloj interno de la placa es de 100MHz, el pulso generado por el generador de pulsos se extiende hasta una duración de 8 clocks.

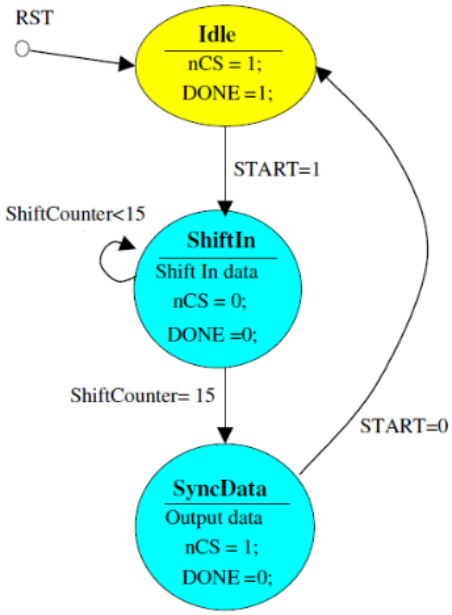

Figura 6: Diagrama de estados del módulo pmod.

En la figura 6 se muestra el diagrama de estados del módulo pmod. La implementación de este módulo fue provista por la cátedra

3.1.2. Detector de flanco

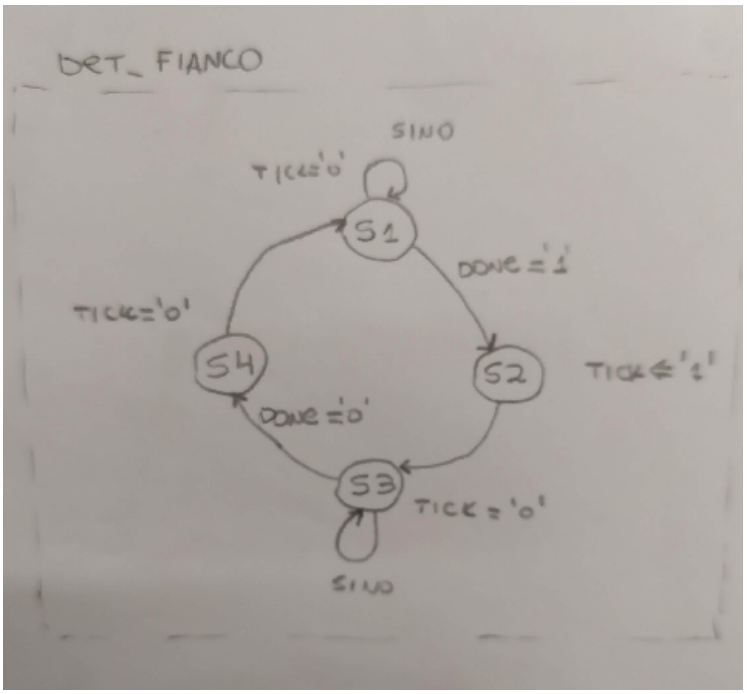

Figura 7: Diagrama de estados del detector de flancos.

En la figura 7 se muestra el diagrama de estados del detector de flanco implementado. Notar que el estado S4 es redundante dado que en esta aplicación no se necesitan detectar flancos descendentes.

3.2. Testeo

Para verificar el correcto funcionamiento de la integración PmodMic-Uart se realizó un test-bench para analizar y comprobar que las señales son las esperadas y luego se probó con la placa y se verificó con un software en la pc que llegaban los datos esperados.

3.2.1. Análisis de señales relevantes

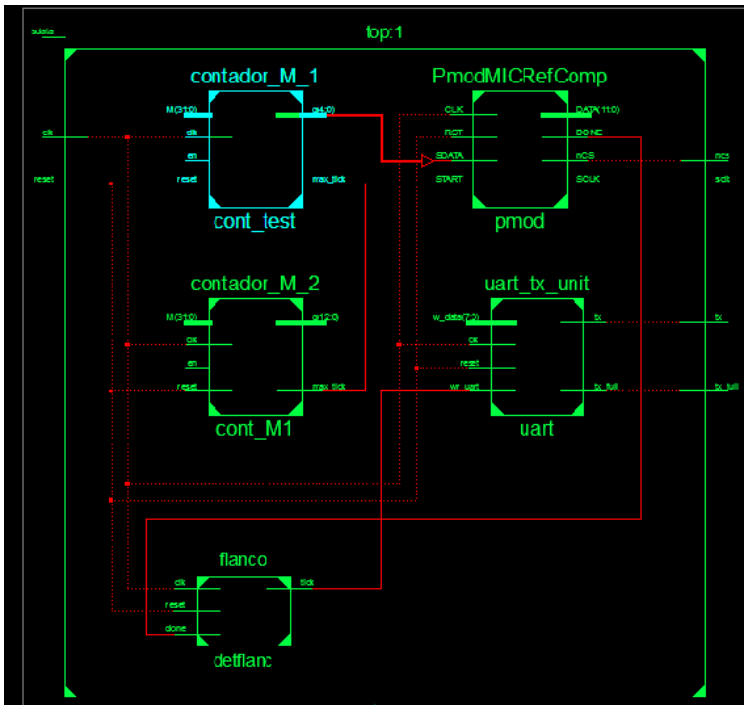

Figura 8: Diagrama de bloques de la lógica del módulo 'top' para la implementación de la integración PmodMic - UART con fin de realizar el test-bench

En la figura 8 se muestra el diagrama del módulo top para el test bench. En este se utiliza como señal de entrada 'sdata' el bit más significativo de un contador de 3 bits.

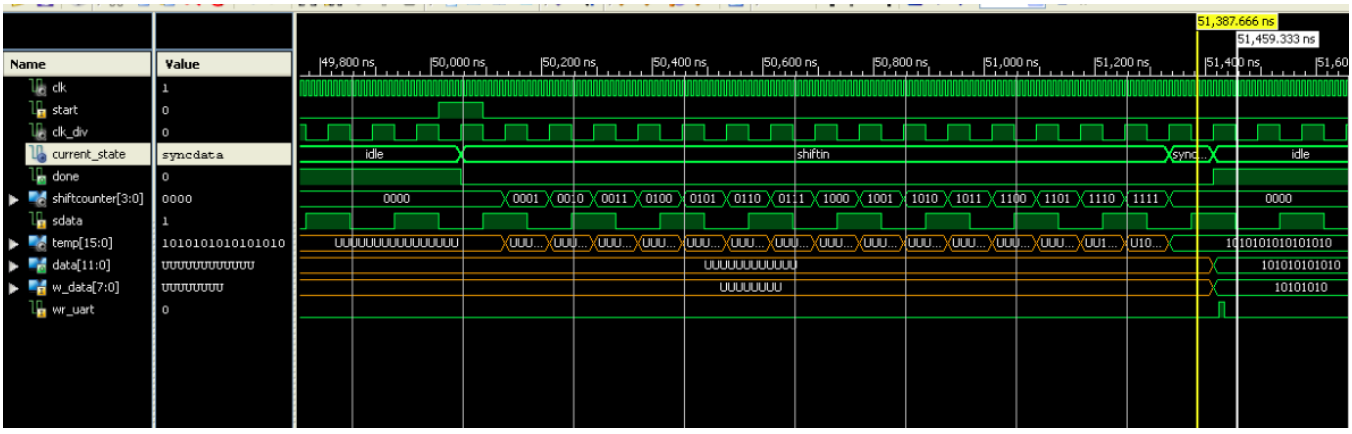

Figura 9: Test bench de la integración PmodMic - UART. Se incluyen señales intermedias relevantes.

En la figura 9 se muestran las señales intermedias principales del test bench. En primer lugar se ve que la señal 'start' es un pulso de longitud 8 clocks. Con esta señal en 1, el próximo flanco ascendente del clock interno 'clk div' se cambia la estado 'shiftin' y done es 0. En este estado, en cada flanco ascendente se copia el valor de 'sdata' al bit menos significativo de 'temp' y luego se

shiftea 'temp' Esto se hace durante 16 periodos del clk interno para luego pasar al estado 'shiftsync'. Luego de un periodo, se transiciona al estado 'idle', donde se genera el vector 'data' que son los 12 bits mas significativos de 'temp' y la señal 'done' es un 1. De estas salidas se generan las entradas para el UART. La señal 'w data' son los 8 bits mas significativos de 'data', mientras que la señal 'wr uart' es un pulso que se produce al detectar el flanco ascendente de done desde 0 a 1.

3.2.2. Testeo en audacity

Una vez verificada la integración PmodMic-Uart en el test-bench se procede a leer los datos adquiridos por el micrófono con un programa de audio para pc (Audacity). Los parámetros para leer los datos crudos son: - encoding: Unsigned. 1 Channel. Frecuencia: 20 kHz.

Se verificó el correcto funcionamiento del módulo.

4. Integración completa del filtro

El módulo top consiste de un contador para generar pulsos de 20kHz, el módulo pMod para tomar la señal del micrófono, un detector de flanco, un shift register para almacenar 20 muestras consecutivas, el filtro FIR para generar la señal filtrada y, el módulo UART para enviar la señal filtrada a la PC.

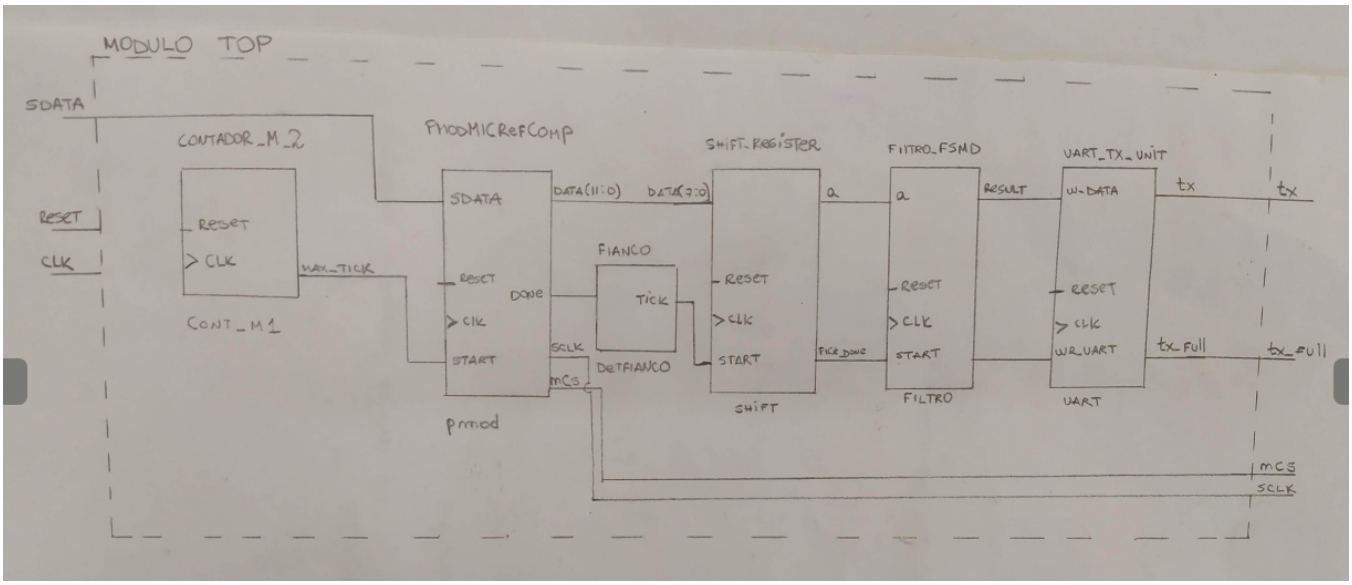

Figura 10: Diagrama de bloques de la lógica del módulo 'top' para la implementación de la integración completa del filtro

En la figura 10 se muestra un esquema del módulo top.

El generador de pulsos genera un tick que da inicio a la recepción del módulo pmod, cuando este termina de recibir la data, la señal 'done' pasa a 1 y el detector de flanco produce un pulso. El pulso es la señal de 'start' del shift register, que almacena el valor de la data. Al finalizar dicha operación

produce un pulso que ingresa como señal de start para el filtro. El filtro toma todas las muestras del shift register para producir la señal filtrada y un pulso al finalizar. Ambas señales ingresan al módulo uart para transferir la data filtrada a la pc a tráves de la señal tx

4.1. Sub-modulos

4.1.1. Shift register

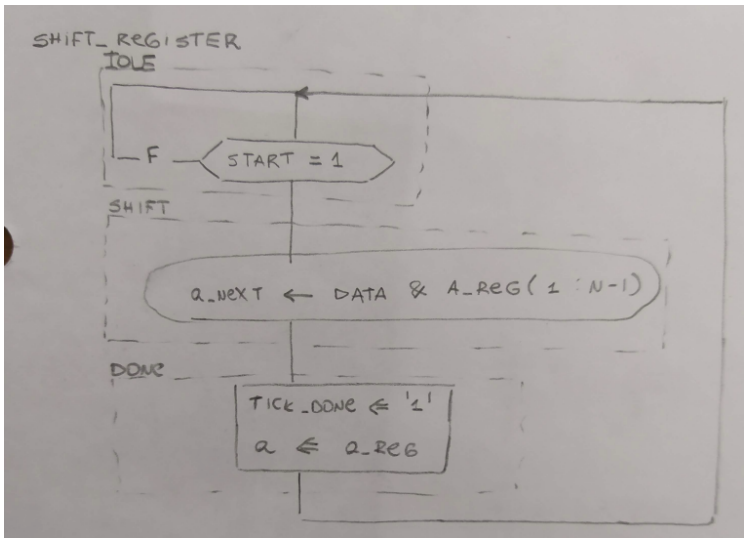

Figura 11: Diagrama de estados del shift register.

En la figura 11 se muestra el diagrama de estados del shift register implementado

4.1.2. Filtro

Se diseña un filtro FIR utilizando el software MATLAB. El filtro es un filtro pasa bajos con una frecuencia de corte de 3 kHz de orden 20, con 14 bits significativos en representación binaria deen punto fijo. La señal generada por un filtro FIR en punto fijo binario es:

$$x_{filtrada} = \sum_{i=1}^{N} b_i x_i \,, \tag{2}$$

donde b_i son las componentes del filtro representadas en punto fijo binario de 14 bits y x_i es una señal de entrada sin filtrar. Luego, la parte entera de la señal sin filtrar es $y = 2^{-14} x_{filtrada}$. Notar que si $x_{filtrada}$ está expresado en su representación binaria, esto es equivalente a descartar los 14 bits menos significativos.

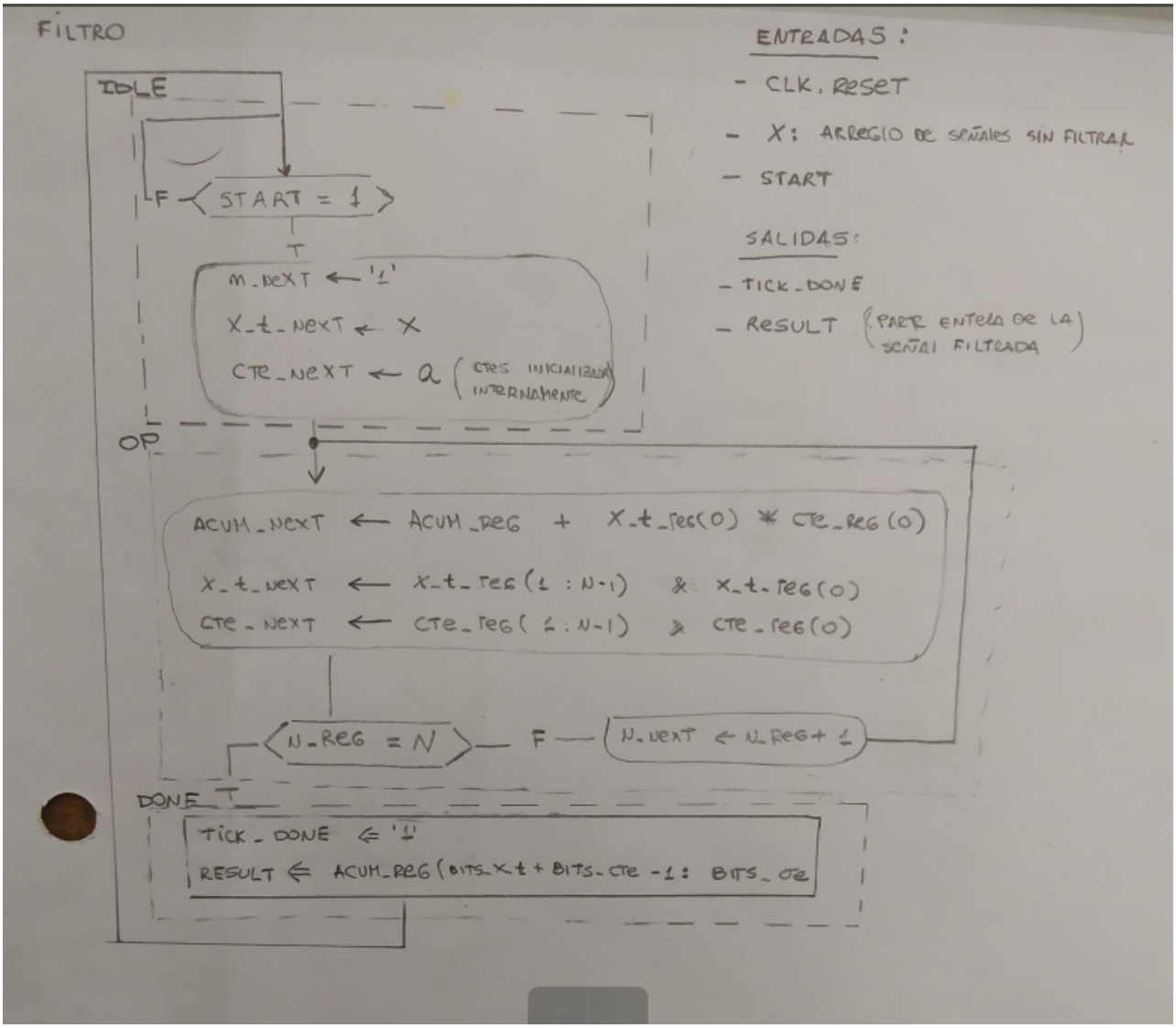

Figura 12: Diagrama de estados del filtro FIR.

En la figura 12 se muestra el diagrama de estados del filtro FIR implementado

4.2. Testeo

Para verificar el funcionamiento del módulo del filtro, se ingresan 20 señales iguales a 1. De esta manera, la salida del filtro es la suma de todos los coeficientes, que por diseño del filtro debe ser aproximadamente igual a 1. Se realizó tal verificacion obteniendo un valor de $x_{filtrada} = 1{,}00024$.

4.2.1. Tiempos caraterísticos

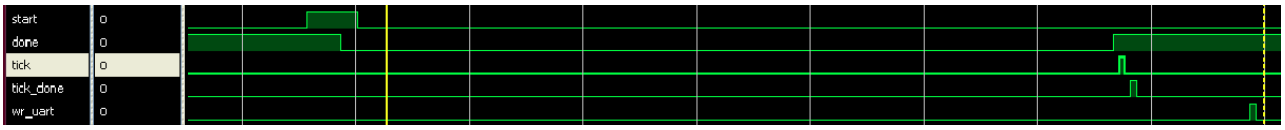

Figura 13: Test-bench de la integración completa del filtro. Se muestran las salidas 'tick' de los submódulos involucrados

El tiempo de todo el proceso debe ser menor a $\frac{1}{20kHz} = 50\mu s$. Los módulos pmod, flanco, sr y filtro tienen una señal de entrada start y una salida done. Los tiempos que requiere cada módulo son:

- pmod: 1420ns
- flanco: 10ns
- shift register: 20 ns
- filtro: 210 ns
- total: 1660 ns

Por otro lado, el módulo uart completa su transimisión luego de transmitir 10 bits, que lo hace en un tiempo de 22,4μs. Luego, el tiempo desde que comienza a ingresar la señal sin filtrar en el módulo pmod con una frecuencia de 20khz hasta que se termina de transmitir a través del módulo uart es de 24,060μs, que es menor a 50µs. Como dato adicional, utilizando el mismo filtro podríamos muestrear con una frecuencia de aproximadamente el doble, es decir, 40kHz.

4.2.2. Testeo en audacity

Se evaluo el desempeño del filtro utilizndo el software Audacity. Para ello se grabo la voz humana superpuesta con una señal externa de 8kHz.

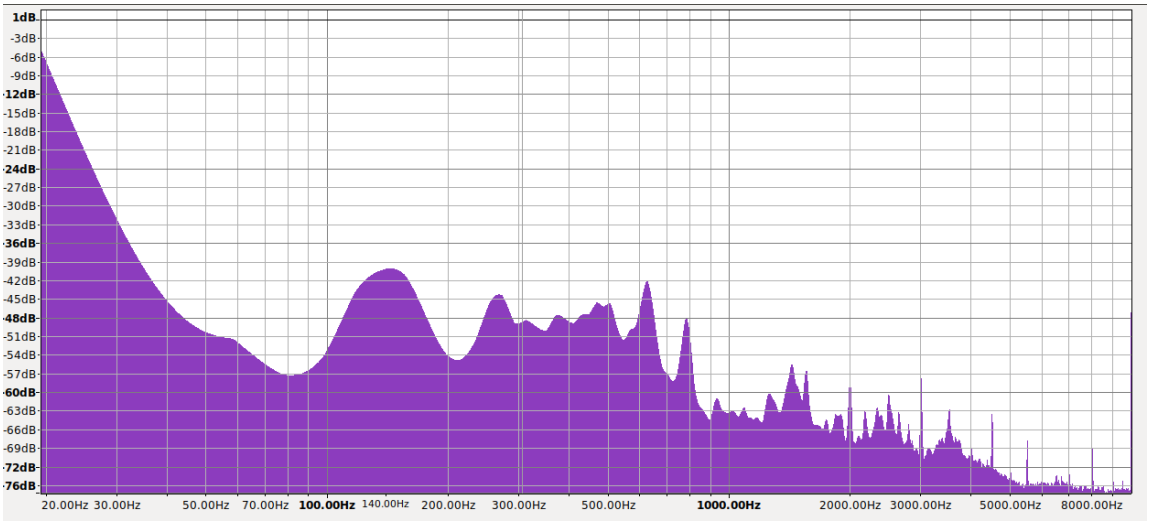

Figura 14: Diagrama de bloques de la lógica del módulo 'top'.

. En la figura 14 se muestra el espectro en frecuencias de la señal filtrada. Se puede observar que el filtro efectivamente atenuó la componente de 8kHz hacia una instensidad menor a -72 dB

5. Conclusiones

Se implementó en una placa Nexys3 un filtro pasa bajos para obtener la frecuencia de voz. Se codificó en lenguaje VHDL. Se realizaron los test-bench correspondientes para los módulos más relevantes y se verificó el correcto funcionamiento utilizando el software Audacity.