



Desarrollo e implementación de un filtro FIR en FPGA

Autor:

Ing. Juan Manuel Guariste

Materia:

Circuitos Lógicos Programables

Índice

Descripción del proyecto realizado	4
Estructura del filtro FIR	5
Integración de los módulos del sistema	5
Validación mediante Testbench	6
4.1 Frecuencia generada por el NCO	6
4.2 Generación de habilitación	7
4.3 Filtro FIR	9
Implementación en FPGA	10
5.1 Diagrama en bloques del circuito implementado	10
5.2 Simulación en Vivado	11
5.3 Implementación del diseño en la FPGA	11
5.4 Uso de recursos	13

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	18 de abril de 2025

Descripción del proyecto realizado

El presente trabajo aborda el desarrollo e implementación de un filtro FIR (Finite Impulse Response) sobre una FPGA Arty-Z7-10.

Los filtros digitales son sistemas diseñados para procesar señales discretas, es decir, señales que han sido previamente muestreadas y digitalizadas. Su principal función es modificar o mejorar ciertas características de la señal, ya sea atenuando el ruido, eliminando componentes indeseadas o resaltando información relevante. A diferencia de los filtros analógicos, los filtros digitales realizan cálculos aritméticos sobre las muestras, lo que les otorga una gran flexibilidad en cuanto a diseño y adaptación a diferentes requisitos.

En particular, este proyecto se centró en la implementación de un filtro FIR. Estos se caracterizan por tener una respuesta finita al impulso, donde la salida en cada instante depende exclusivamente de las últimas N muestras de entrada y de un conjunto fijo de coeficientes. Los filtros FIR son ampliamente valorados en sistemas de procesamiento digital de señales debido a su estabilidad inherente y a su capacidad para mantener una fase lineal, una propiedad fundamental en muchas aplicaciones.

El desarrollo consistió en la implementación, en lenguaje VHDL, de un módulo que replica la estructura típica de un filtro FIR. Dicho módulo fue integrado con diversos bloques auxiliares, entre los que se destacan: un generador de señal NCO (Numerically Controlled Oscillator), un bloque de habilitación de muestreo, un VIO (Virtual Input/Output) para el ajuste dinámico de parámetros en tiempo real y un ILA (Integrated Logic Analyzer) para la observación y captura de señales durante la ejecución. La interacción entre estos módulos se puede observar en la figura 1.

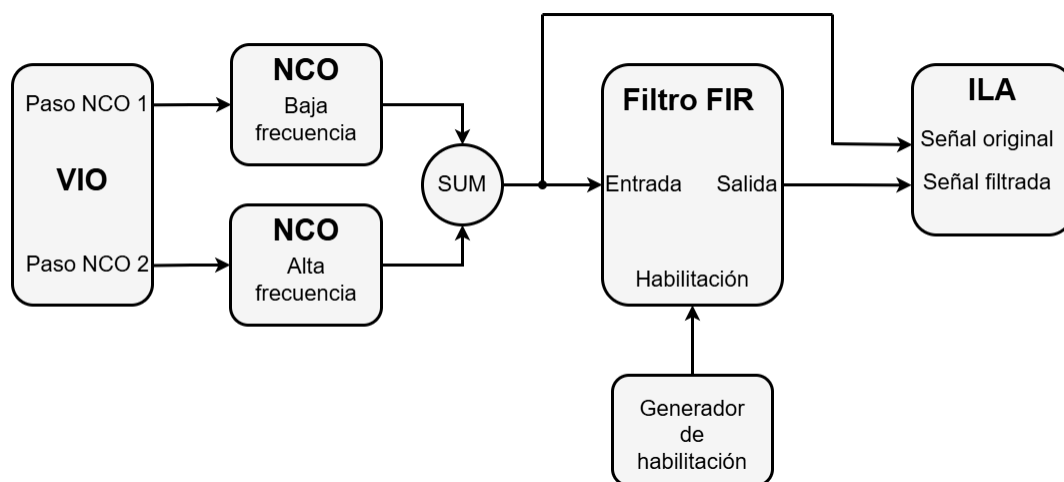


Figura 1. Arquitectura general del sistema.

Esta arquitectura permite procesar señales generadas internamente en hardware y validar, en tiempo real, tanto la señal original como la salida filtrada, facilitando la comparación entre los resultados experimentales y las simulaciones teóricas de manera eficiente.

Estructura del filtro FIR

El filtro FIR implementado sigue la estructura clásica de convolución discreta, cuya ecuación general se expresa como:

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k] \quad (1)$$

Donde:

- $y[n]$: salida filtrada en el instante n
- $x[n]$: muestra de entrada en el instante n
- $h[k]$: coeficientes que definen la respuesta en frecuencia del filtro
- N : número de coeficientes o *taps*

El filtro fue diseñado de manera que la cantidad de *taps* sea configurable, lo cual permite ajustarlo según los requerimientos específicos de atenuación y selectividad de cada aplicación. En la implementación final, los coeficientes fueron previamente calculados y escalados para utilizar aritmética en punto fijo bajo el formato $Q1,11$. Esto permite realizar multiplicaciones y acumulaciones de forma eficiente, optimizando el uso de recursos lógicos en la FPGA sin sacrificar precisión.

Integración de los módulos del sistema

Para validar el correcto funcionamiento del filtro FIR en la FPGA, se diseñó una arquitectura modular que permite generar señales, aplicar el filtrado y observar los resultados en tiempo real.

Los componentes principales del sistema son:

- NCO: módulo basado en un Oscilador Controlado Numéricamente, encargado de generar señales senoidales digitales. Este componente fue provisto por la cátedra. En el diseño se implementaron dos instancias del NCO, configuradas para producir componentes de baja y alta frecuencia. Ambas señales se combinan para conformar una señal compuesta, que actúa como entrada del filtro FIR.
- Filtro FIR: bloque encargado de realizar la operación de convolución sobre las muestras de entrada, aplicando los coeficientes definidos y generando la señal filtrada.
- Generador de habilitación: módulo responsable de generar la señal de habilitación que sincroniza la adquisición de muestras por parte del filtro. Permite controlar con precisión la frecuencia de muestreo.
- VIO: permite la modificación en tiempo real de parámetros como la frecuencia de las señales generadas y la activación de reset de módulos, a través del entorno de depuración Vivado.
- ILA: permite la observación y captura de señales internas durante la ejecución, facilitando la validación del comportamiento del sistema y la comparación con resultados teóricos.

Validación mediante Testbench

Tras completar la implementación del filtro FIR y los módulos auxiliares, se llevó a cabo una validación inicial mediante simulación. Para ello se utilizó un *testbench* escrito en VHDL, ejecutado en el entorno ModelSim.

El principal objetivo de esta etapa fue validar el correcto funcionamiento individual de cada uno de los módulos que conforman el sistema, así como también comprobar su integración y la coherencia en su interacción. Para ello, se verificaron los siguientes aspectos:

- Que el módulo NCO generara señales senoidales con la frecuencia esperada, confirmando que la salida coincidiera con el valor teórico calculado en función del parámetro *paso*.
- Que el módulo generador de habilitación activara correctamente su señal de control, de acuerdo al valor establecido en su contador interno, garantizando de esta manera la sincronización entre la adquisición de datos y el procesamiento en el filtro.
- Que el filtro FIR procesara las muestras de entrada de manera correcta, verificando que la salida se ajustara a lo esperado, en base a los coeficientes definidos durante el diseño y al comportamiento teórico del filtro.

A continuación, se presenta un análisis detallado de cada uno de los bloques mencionados, junto con sus respectivas simulaciones.

4.1. Frecuencia generada por el NCO

La frecuencia de salida generada por el NCO depende directamente del valor de *paso* configurado, de acuerdo con la siguiente expresión:

$$f_{\text{out}} = \frac{\text{paso} \times f_{\text{clk}}}{2^{\text{ADDR_W}}}$$

Donde:

- f_{out} : frecuencia de salida de la señal generada.
- *paso*: es el incremento en cada ciclo de reloj.
- f_{clk} : frecuencia del reloj de sistema, en este caso 125 MHz.
- *ADDR_W*: cantidad de bits de las direcciones de la LUT, configurado en 12 bits.

Esta relación permite establecer de forma precisa la frecuencia de la señal generada, lo cual es fundamental para las pruebas y validaciones del sistema.

La figura 2 muestra la señal senoidal generada por el NCO con un *paso* de 2, obteniendo una frecuencia de salida de aproximadamente 61,03 kHz.

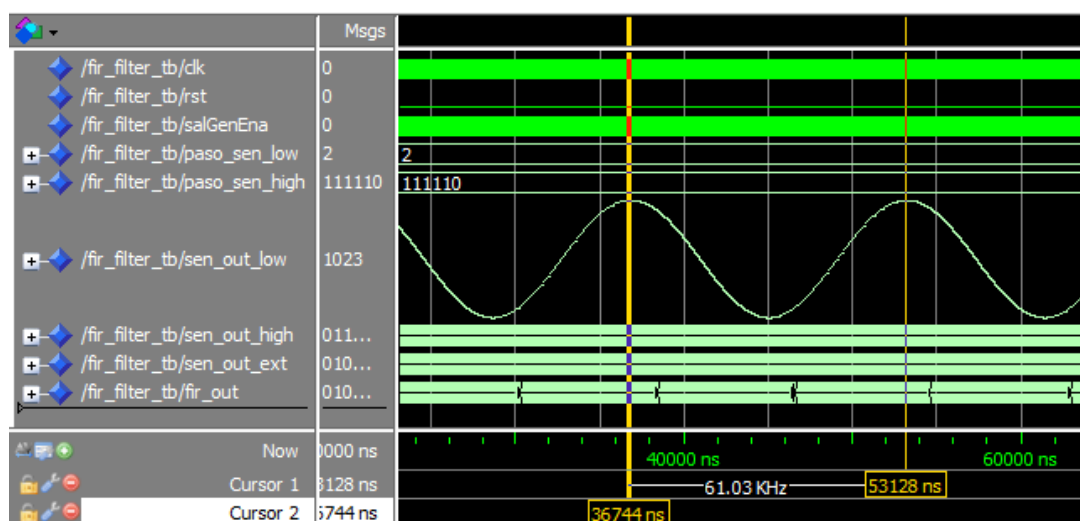


Figura 2. Señal senoidal generada por el NCO con paso = 2.

En la figura 3 se presenta la simulación de una señal de mayor frecuencia, obtenida configurando el valor de *paso* en 62. En este caso, la frecuencia de salida alcanzada fue aproximadamente 1,89 MHz.

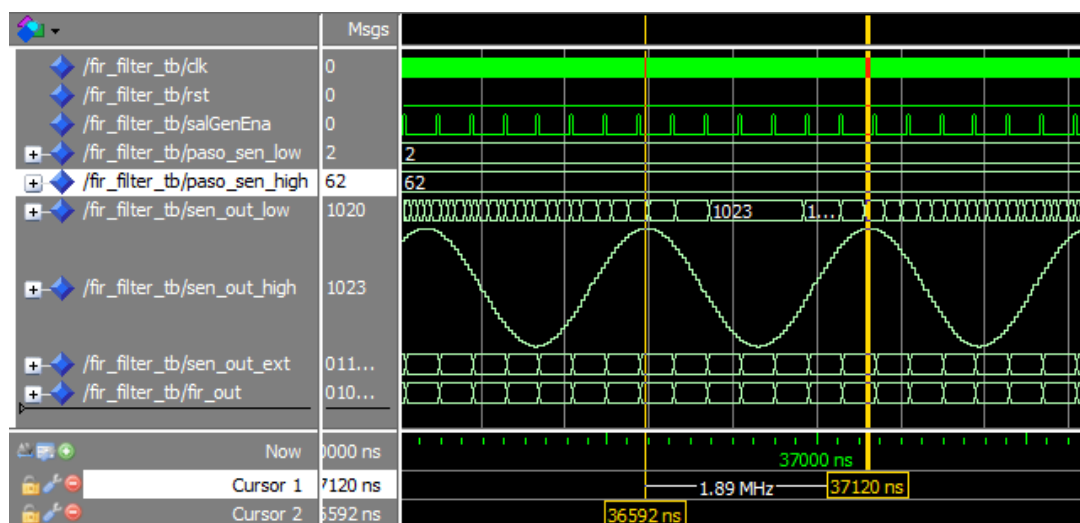


Figura 3. Señal senoidal generada por el NCO con paso = 62.

Los resultados observados coinciden con los valores teóricos calculados mediante la fórmula, validando el correcto funcionamiento del módulo NCO.

4.2. Generación de habilitación

El sistema implementado en FPGA opera con una frecuencia de 125 MHz, correspondiente a la señal de *clock* principal. Sin embargo, para el correcto funcionamiento del filtro FIR es necesario reducir la frecuencia de muestreo de las señales, ajustándola a las necesidades de la aplicación y a la banda de interés.

Para lograr dicha reducción, se implementó un generador de habilitación, cuyo propósito es actuar como divisor de frecuencia. Este módulo genera una señal de habilitación que se activa cada N ciclos de reloj, siendo N un valor configurable mediante un parámetro genérico.

El funcionamiento se basa en un contador que opera sincronizado con la señal de *clock* a 125 MHz. Este contador incrementa su valor en cada flanco activo del reloj y, al alcanzar el valor configurado N , genera una habilitación que permite la adquisición y posterior procesamiento de una nueva muestra de entrada. Este mecanismo determina la frecuencia efectiva de muestreo, la cual se calcula mediante la siguiente expresión:

$$f_{\text{muestreo}} = \frac{f_{\text{clk}}}{N}$$

Donde:

- f_{muestreo} : frecuencia de muestreo efectiva aplicada al filtro FIR.
- f_{clk} : frecuencia del *clock* principal del sistema (125 MHz).
- N : valor del divisor, configurable en la instancia del módulo.

En este proyecto se configuró $N = 10$. De esta manera, por cada 10 ciclos de reloj, se habilita el procesamiento de un nuevo dato, lo que implica una frecuencia de muestreo reducida a:

$$f_{\text{muestreo}} = \frac{125 \text{ MHz}}{10} = 12,5 \text{ MHz}$$

Esta estrategia permite adaptar la frecuencia de muestreo al requerimiento de la aplicación, sin necesidad de modificar el *clock* global, y garantiza que el filtro FIR procese datos de manera sincronizada y controlada.

La figura 4 ilustra el comportamiento del generador de habilitación, mostrando cómo se produce una señal de habilitación cada 10 ciclos de *clock*.

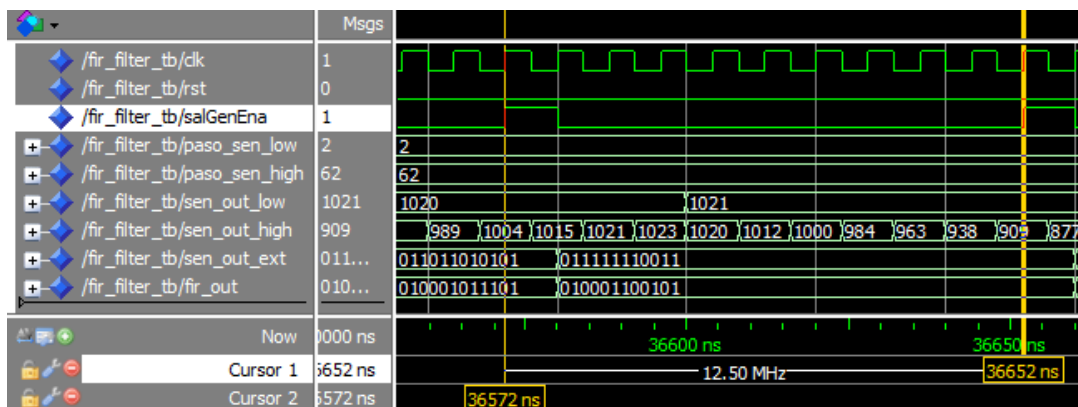


Figura 4. Funcionamiento del generador de habilitación con $N=10$.

4.3. Filtro FIR

El filtro implementado corresponde a un pasabajos de orden 32, diseñado con una frecuencia de corte de 300 kHz y una frecuencia de muestreo de 12,5 MHz. Los coeficientes fueron calculados aplicando el método de diseño mediante ventana de Hamming.

Para validar su funcionamiento, se utilizó como señal de entrada una composición de dos componentes senoidales: una de 1,89 MHz y otra de 61,03 kHz.

La figura 5 muestra la señal de entrada obtenida a partir de la suma de ambas senoidales.

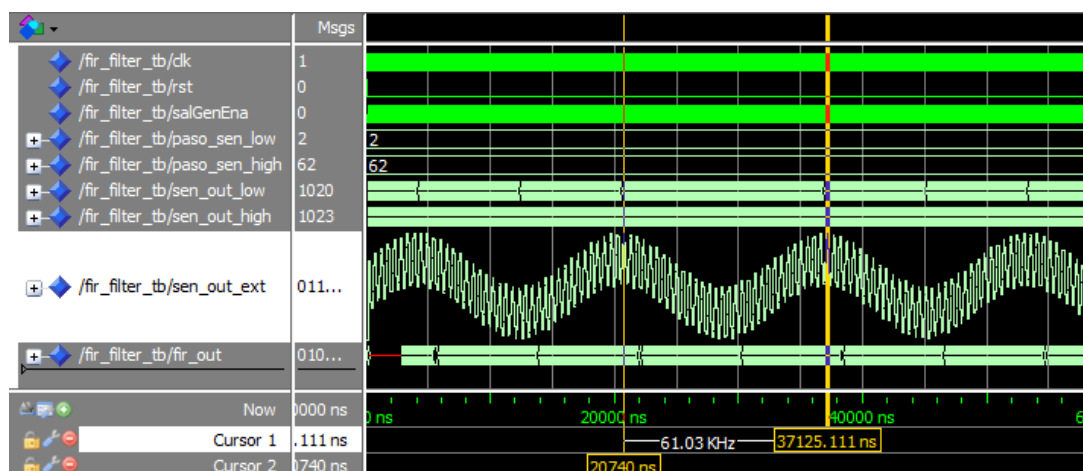


Figura 5. Señal compuesta por la suma de dos senoidales: 1,89 MHz y 61,03 kHz.

En la figura 6 se presenta la señal correspondiente a la salida del filtro. Tal como se esperaba, la componente de alta frecuencia fue atenuada, mientras que la de baja frecuencia se mantiene, ya que se encuentra por debajo de la frecuencia de corte del filtro.

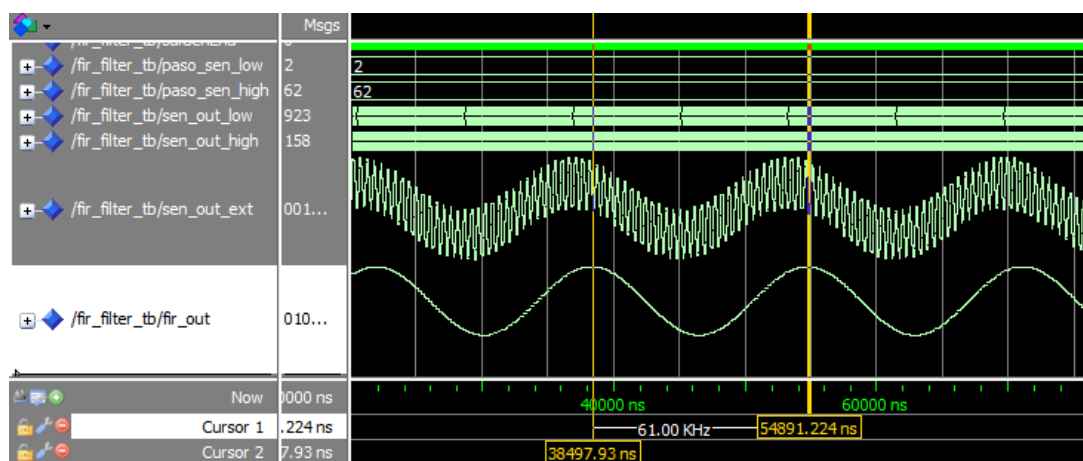


Figura 6. Salida del filtro FIR, con atenuación de la componente de alta frecuencia.

Los resultados obtenidos confirman que el filtro FIR opera correctamente, cumpliendo con la atenuación deseada en las frecuencias superiores a la de corte.

Implementación en FPGA

Una vez validados mediante simulación los componentes principales que conforman el sistema, se procedió a realizar su implementación en una FPGA, utilizando la herramienta de desarrollo *Vivado*.

Dado que no se disponía de acceso físico a una FPGA propia, se empleó una unidad provista por la cátedra, la cual fue accedida de manera remota. Esta modalidad permitió cargar el diseño, realizar pruebas funcionales y observar las señales en tiempo real.

Con el objetivo de facilitar la interacción durante las pruebas, se incorporó un módulo *VIO* (Virtual Input/Output), que permitió simular señales de entrada directamente desde el entorno de *Vivado*. Asimismo, se utilizó un *ILA* (Integrated Logic Analyzer), que facilitó la observación de las formas de onda correspondientes a las señales de entrada y salida del filtro FIR.

5.1. Diagrama en bloques del circuito implementado

En las figuras 7 y 8 se presenta el esquemático correspondiente a la implementación del sistema en la FPGA. En dichos diagramas se observa la interconexión de los distintos módulos que componen la arquitectura, así como su sincronización con la señal de *clock* general.

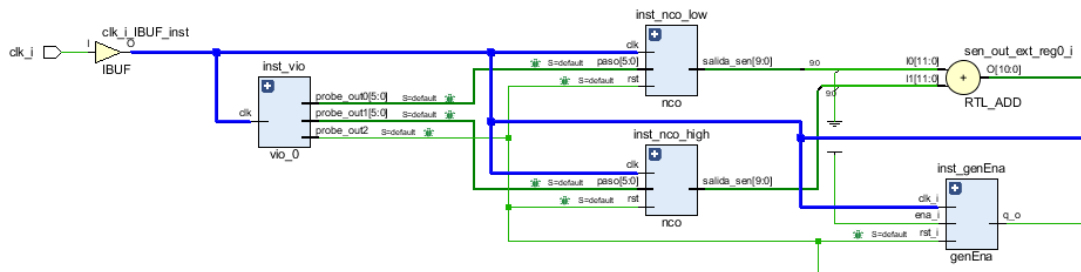


Figura 7. Esquemático de la implementación — Parte A.

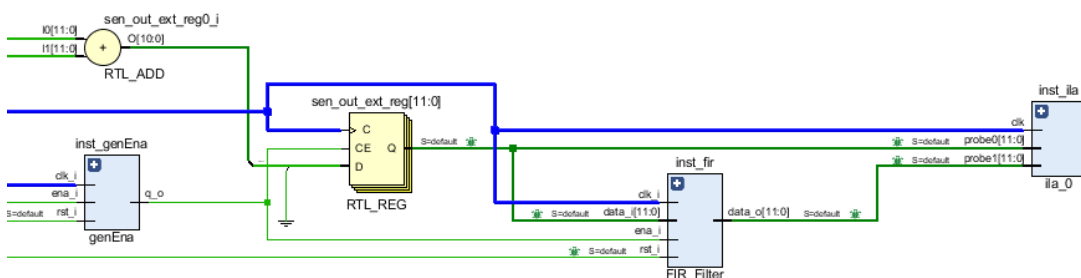


Figura 8. Esquemático de la implementación — Parte B.

5.2. Simulación en Vivado

La figura 9 presenta las formas de onda obtenidas durante la simulación del sistema en *Vivado*. Con el objetivo de validar su correcto funcionamiento, se replicaron las condiciones utilizadas en las simulaciones previas realizadas con *ModelSim*. Para ello, se aplicó como entrada una señal compuesta por la superposición de dos componentes senoidales: una de 61,03 kHz y otra de 1,89 MHz.

Tal como se observa, la salida del filtro FIR atenúa eficazmente la componente de alta frecuencia, conservando únicamente la señal de baja frecuencia, que se encuentra dentro del rango de paso definido por la frecuencia de corte del filtro.

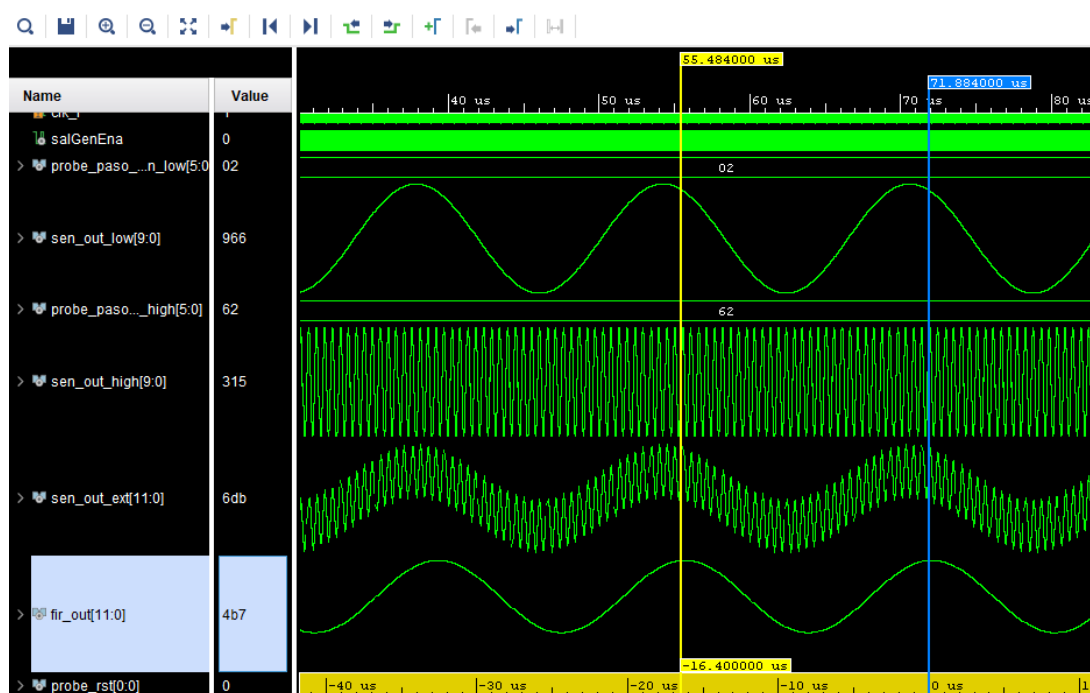


Figura 9. Simulación de las formas de onda en Vivado.

5.3. Implementación del diseño en la FPGA

Una vez verificado el correcto funcionamiento mediante simulaciones en *ModelSim* y *Vivado*, se procedió a configurar la FPGA con el diseño definitivo.

Para generar los estímulos de entrada, se utilizó el módulo *VIO*, el cual permitió ajustar en tiempo real los valores de entrada para los *NCO*. La señal de entrada al filtro estuvo compuesta por la suma de dos senoidales:

- Para la señal de baja frecuencia, se configuró un valor de paso igual a 9, lo que resultó en una senoidal de aproximadamente 274 kHz.
- Para la señal de alta frecuencia, se configuró un valor de paso igual a 63, lo que generó una frecuencia de aproximadamente 1.92 MHz.

La figura 10 muestra la configuración de las señales de entrada, incluyendo los valores de paso asignados a los *NCO* y la señal de *reset*.

hw_vio_1					
<div> <div>Q</div> <div>≡</div> <div>⚙</div> <div>+</div> <div>-</div> </div>					
Name	Value	Activity	Direction	VIO	
> paso[5:0]	[U] 9		Output	hw_vio_1	
> paso__0[5:0]	[U] 63		Output	hw_vio_1	
probe_rst	[B] 0		Output	hw_vio_1	

Figura 10. Configuración de entradas mediante VIO.

La figura 11 muestra la señal de entrada y la salida del filtro FIR, observadas mediante el *ILA*. El uso de marcadores permitió validar las frecuencias de cada componente. Se observa que la salida del filtro corresponde únicamente a la señal de baja frecuencia, confirmando que la componente de alta frecuencia fue correctamente atenuada, tal como se esperaba.

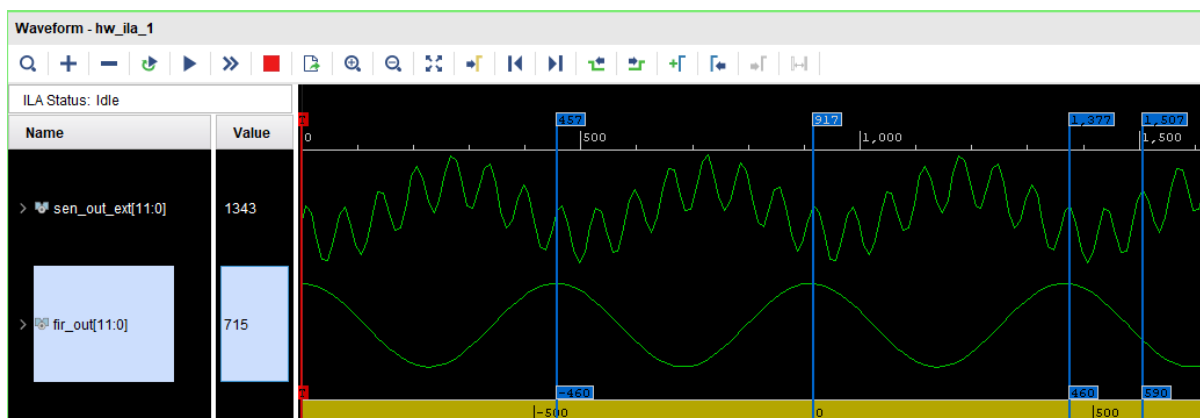


Figura 11. Señales de entrada y salida observadas mediante ILA.

5.4. Uso de recursos

En la figura 12 se presenta la tabla de utilización de recursos de la FPGA una vez implementado el diseño.

Resource	Utilization	Available	Utilization %
LUT	1668	17600	9.48
LUTRAM	88	6000	1.47
FF	1962	35200	5.57
BRAM	1.50	60	2.50
DSP	21	80	26.25
IO	1	100	1.00

Figura 12. Tabla de utilización de recursos.

A su vez, la figura 13 muestra una representación gráfica de los valores expuestos en la tabla anterior, permitiendo una visualización más clara de la distribución de recursos dentro del dispositivo.

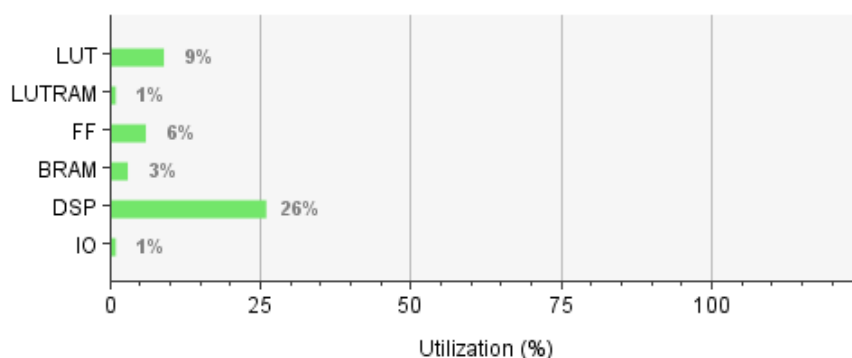


Figura 13. Gráfico de utilización de recursos.