Máster Universitario en Ingeniería de Sistemas Electrónicos

**Imagen que contiene Texto

Descripción generada automáticamente**

**Práctica E4.1: Filtro FIR Compensador del CIC**

Jose Luis, Rocabado Rocha

Gianmarco Leopoldo, Sangoi Da Roza

04/28/2022

# **ÍNDICE**

[**ÍNDICE** 2](#_Toc102588045)

[**ÍNDICE DE FIGURAS** 3](#_Toc102588046)

[Descripción del módulo 4](#_Toc102588047)

[Interfaz 6](#_Toc102588048)

[Recursos Hardware 8](#_Toc102588049)

[Frecuencia Máxima 11](#_Toc102588050)

[Camino Critico 11](#_Toc102588051)

[Verificación 12](#_Toc102588052)

[Simulaciones con MATLAB y SIMULINK 12](#_Toc102588053)

[Testbenchs 14](#_Toc102588054)

[Resolución problemas encontrados 18](#_Toc102588055)

# **ÍNDICE DE FIGURAS**

[Ilustración 1. Respuesta en frecuencia del filtro CIC de tres etapas implementado en la práctica anterior. 4](#_Toc102588056)

[Ilustración 2. Respuesta en frecuencia del filtro compensador de 17 coeficientes. 5](#_Toc102588057)

[Ilustración 3. Recursos del filtro compensador (TOP) 8](#_Toc102588058)

[Ilustración 4. Recursos del módulo de control 9](#_Toc102588059)

[Ilustración 5. Recursos del módulo de memoria ROM 9](#_Toc102588060)

[Ilustración 6. Recursos del módulo de celda básica (multiplicador + acumulador) 10](#_Toc102588061)

[Ilustración 7. Recursos del módulo de desplazamiento y multiplexor. 10](#_Toc102588062)

[Ilustración 8. Frecuencia máxima de operación del sistema (Fmax). 11](#_Toc102588063)

[Ilustración 9. Camino crítico del sistema generado por el Technology Map Viewer. 11](#_Toc102588064)

[Ilustración 10. Zoom de la simulación de Simulink que compara el modelo ideal con el modelo cuantificado con precisión completa. 12](#_Toc102588065)

[Ilustración 11. Arriba: Plot de los coeficientes del filtro cuantificados. Abajo: Resultado de la respuesta ante el impulso en 12](#_Toc102588066)

[Ilustración 12. Resultados de la señal filtrada para una señal senoidal de frecuencia 1kHz, 10kHz y 15 kHz. 13](#_Toc102588067)

[Ilustración 13. Verificación de la memoria ROM. 14](#_Toc102588068)

[Ilustración 14. Verificación del módulo de control 15](#_Toc102588069)

[Ilustración 15. Verificación del multiplicador + acumulador 15](#_Toc102588070)

[Ilustración 16. Verificación del registro de desplazamiento y el multiplexor de salida 16](#_Toc102588071)

[Ilustración 17. Verificación de nuestro filtro compensador (entrada: sinusoidal 10KHz) 17](#_Toc102588072)

[Ilustración 18. Respuesta al impulso de nuestro filtro (TOP) 17](#_Toc102588073)

# Descripción del módulo

Cuando implementamos un filtro CIC interpolador, como el realizado en la práctica 3 para nuestro sistema de modulador AM/FM normalmente queremos un paso de banda plano y estrecho en la región de transición, estas son cualidades deseadas que sin embargo no son propias de un filtro CIC debido a la caída prolongada en la banda de paso, es por esto por lo que se implementa un filtro compensador que en conjunto con el filtro CIC nos permita obtener una banda de paso con ganancia constante.

Si observamos la respuesta en frecuencia del filtro CIC y del filtro compensador (figura 1 y figura 2 respectivamente) podemos comprobar que obtenemos el efecto deseado.

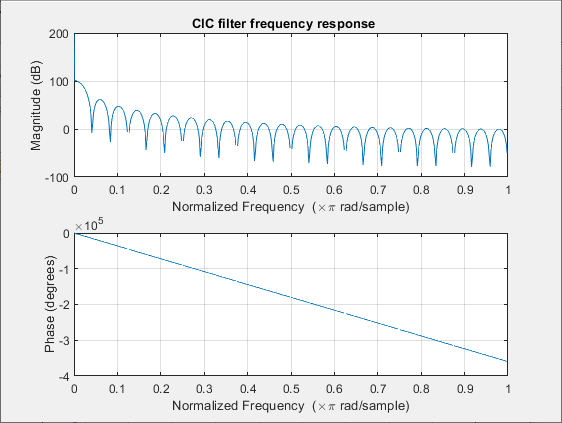


Ilustración 1. Respuesta en frecuencia del filtro CIC de tres etapas implementado en la práctica anterior.

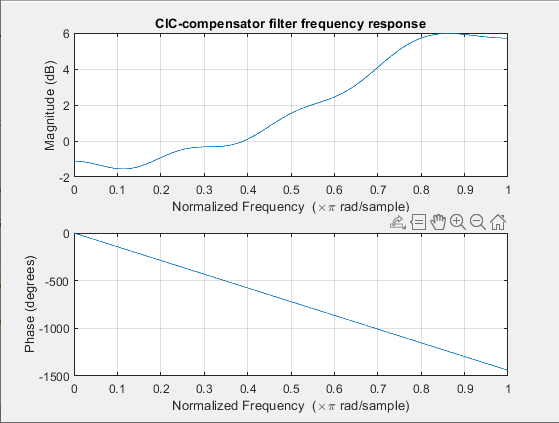


Ilustración 2. Respuesta en frecuencia del filtro compensador de 17 coeficientes.

Es necesario remarcar que Matlab tiene una normalización [0,1] donde 0.5 correspondería con la mitad de la frecuencia de muestreo. Por otro lado, se ha de tener en cuenta que el filtro CIC introduce un *upsampling* cambiando la frecuencia de muestreo. Por esta razón, si analizamos las dos respuestas en frecuencia respecto a frecuencias analógicas podremos comprobar como el filtro compensador afectará únicamente a la banda pasante del filtro CIC permitiéndonos obtener el resultado deseado.

En esta práctica se realizará un módulo de filtro FIR secuencial compensador para el filtro CIC realizado con anterioridad. La frecuencia de muestreo de los datos de entrada será de 50kHz, mientras que la frecuencia de reloj del filtro será de 100MHz.

El filtro se constituirá de 4 bloques:

1. Un registro de desplazamiento y multiplexor para la salida (REG\_MUX).
2. Un multiplicador más acumulador (celda básica en filtros secuenciales) (MULT\_ACC).
3. Una ROM (ROM).
4. Una máquina de estados para direccionar la memoria ROM, la celda del multiplicador acumulador y el registro de salida (CONTROL).

# Interfaz

* Módulo SEC\_FILTER (top):

|  |  |  |  |
| --- | --- | --- | --- |
| **PARÁMETROS** | | | |
| **Nombre** | **Descripción** | | |
| Win | Cuantificación de la entrada y salida | | |
| Wc | Cuantificación de los coeficientes | | |
| Num\_coef | Número de coeficientes | | |
| **INTERFAZ** | | | |
| **Nombre** | **Tipo** | **Formato** | **Descripción** |
| clk | in | bit | Entrada de reloj |
| rst | in | bit | Reset síncrono |
| val\_in | in | bit | Entrada de validación de datos |
| din | in | S[Win, Win-1] | Entrada de datos filtro CIC |
| dout | out | S[19, 16] | Salida de datos truncada a 19 bits. |
| val\_out | out | bit | Salida de la señal de validación |

* Módulo CONTROL:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PARÁMETROS** | | | | |
| **Nombre** | | **Descripción** | | |
| Num\_coef | | Número de coeficientes | | |
| **INTERFAZ** | | | | |
| **Nombre** | **Tipo** | | **Formato** | **Descripción** |
| clk | in | | bit | Entrada de reloj |
| rst | in | | bit | Reset síncrono |
| val\_in | in | | bit | Entrada de validación de datos |
| addr | out | | U[log2(Num\_coef), 0] | Salida de direccionamiento |
| rst\_Acc | out | | bit | Reset del acumulador |
| ce\_Acc | out | | bit | Salida de la señal de habilitación del acumulador |
| val\_out | out | | bit | Salida de la señal de validación |

* Módulo REG\_MUX (top):

|  |  |  |  |
| --- | --- | --- | --- |
| **PARÁMETROS** | | | |
| **Nombre** | **Descripción** | | |
| Win | Cuantificación de la entrada y salida | | |
| Num\_coef | Número de coeficientes | | |
| **INTERFAZ** | | | |
| **Nombre** | **Tipo** | **Formato** | **Descripción** |
| clk | in | bit | Entrada de reloj |
| ce | in | bit | Entrada de habilitación de datos |
| sel | in | U[log2(Num\_coef), 0] | Selector del mux |
| din | in | S[Win, Win-1] | Entrada |
| dout | out | S[Win, Win-1] | Salida |

* Módulo MULT\_ACC:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PARÁMETROS** | | | | |
| **Nombre** | | **Descripción** | | |
| Win | | Cuantificación de la entrada y salida | | |
| Wc | | Cuantificación de los coeficientes | | |
| **INTERFAZ** | | | | |
| **Nombre** | **Tipo** | | **Formato** | **Descripción** |
| clk | in | | bit | Entrada de reloj |
| rst | in | | bit | Reset síncrono del acumulador |
| ce | in | | bit | Habilitador del acumulador |
| din | in | | S[Win, Win-1] | Entrada del acumulador |
| coef | in | | S[Wc, Wc-1] | Coeficiente del acumulador |
| dout | out | | S[Win+Wc, Win+Wc-3] | Salida filtrada con 3 bits enteros para evitar desbordamiento |

* Módulo ROM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PARÁMETROS** | | | | |
| **Nombre** | | **Descripción** | | |
| Wc | | Cuantificación de los coeficientes | | |
| Num\_coef | | Número de coeficientes | | |
| **INTERFAZ** | | | | |
| **Nombre** | **Tipo** | | **Formato** | **Descripción** |
| clk | in | | bit | Entrada de reloj |
| addr | in | | U[log2(Num\_coef), 0] | Direccionamiento de la memoria |
| data | out | | S[Wc, Wc-1] | Salida de la ROM |

# Recursos Hardware

* Recursos de nuestro sistema SEC\_FILTER:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Ilustración 3. Recursos del filtro compensador (TOP)

Para estimar los recursos se tiene en cuenta los bits utilizados en cada modulo del sistema y sus registros:

* En el modulo de control la estimación es de 10 LEs debido a las 4 salidas y el registro de de estado de señal.
* La memoria ROM se implementará dentro de una memoria M9K para reducir la lógica del sistema.
* El registro de desplazamiento se tienen 16 bits de entrada registrados 17 veces = 272 Les.
* En el módulo de multiplicación y acumulador la entrada es de 16 bits y 4 salidas de 34 bits = 152 LEs
* Modulo SEC\_FILTER 16 bits de entrada y 34 de salida truncado = 65 LEs.
* Recursos del módulo de CONTROL:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Ilustración 4. Recursos del módulo de control

* Recursos del módulo ROM:

Tabla

Descripción generada automáticamente

Ilustración 5. Recursos del módulo de memoria ROM

Este módulo se implementó con lógica por lo que está bien que nos salga como recursos 0 memorias M9K.

* Recursos del módulo MULT\_ACC:

Tabla

Descripción generada automáticamente

Ilustración 6. Recursos del módulo de celda básica (multiplicador + acumulador)

* Recursos del módulo REG\_MUX:

Tabla

Descripción generada automáticamente

Ilustración 7. Recursos del módulo de desplazamiento y multiplexor.

# Frecuencia Máxima

La placa utilizada en el laboratorio de prácticas consiste en una Cyclone IV DE-115 cuya frecuencia máxima es de 250MHz.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 8. Frecuencia máxima de operación del sistema (Fmax).

Mediante la herramienta *TimeQuest Timing Analyzer*, podremos calcular la frecuencia máxima a la que trabajará el modelo diseñado. Utilizando el *wrapper* (instanciación con registros en las entradas y salidas) creado, hemos sido capaces de obtener una frecuencia máxima de trabajo de ~163.19 MHz tal y como se observa en la siguiente figura.

# Camino Critico

El camino critico es aquel en que la señal tarda más desde su ingreso hasta su destino. Mediante el *Technology Map Viewer* podremos encontrar en que parte del circuito se encuentra.

Diagrama

Descripción generada automáticamente

Ilustración 9. Camino crítico del sistema generado por el Technology Map Viewer.

El camino critico de nuestro módulo SEC\_FILTER se encuentra en la etapa de los registros de desplazamientos y multiplexores.

# Verificación

### Simulaciones con MATLAB y SIMULINK

Para el desarrollo y verificación del filtro, lo primero que se hizo fue entender las señales de entrada y salida de nuestro sistema. Esto se realizó simulando con un fichero en *MATLAB* y *Simulink* las entradas y salidas que debería tener nuestro filtro. Desde comparar el modelo ideal con el modelo cuantificado con precisión completa, a obtener la respuesta ante el impulso y realizar un barrido en frecuencia.

Pantalla de computadora con fondo negro

Descripción generada automáticamente con confianza baja

Ilustración 10. Zoom de la simulación de Simulink que compara el modelo ideal con el modelo cuantificado con precisión completa.

Cuando comparamos ambos modelos (ilustración 10), se puede observar que se obtiene un error prácticamente negligible debido a la cuantificación. En cuanto a la respuesta al impulso, se espera obtener por definición los coeficientes del filtro tal y como se observa en la siguiente figura.

Imagen que contiene Gráfico

Descripción generada automáticamente

Ilustración 11. Arriba: Plot de los coeficientes del filtro cuantificados. Abajo: Resultado de la respuesta ante el impulso en

Por último, al observar la salida ante una señal senoidal de 1V de amplitud y diferentes frecuencias se pretende comprobar cómo, para frecuencias mayores de señal, realmente se amplifican para compensar el filtro CIC permitiéndonos obtener una banda pasante con una ganancia constante (figura 5).

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 12. Resultados de la señal filtrada para una señal senoidal de frecuencia 1kHz, 10kHz y 15 kHz.

### Testbenchs

Para la verificación de los módulos se generaron los *TestBench* correspondientes que de forma sencilla nos permiten validar la ejecución correcta de las operaciones. Desde el fichero en MATLAB elaborado podemos realizar las simulaciones con distintas frecuencias y ondas (sinusoidal, Chirp e Impulse). Principalmente para nuestras simulaciones empleos una onda sinusoidal que iremos variando en frecuencias de 1KHz, 10KHz y 15KHz para ver el comportamiento de nuestro filtro al aumentar la frecuencia de señal.

* Testbench módulo de la memoria ROM:

Se comprobó que el direccionamiento de los coeficientes en nuestra memoria estaba bien realizado.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 13. Verificación de la memoria ROM.

* Testbench módulo de Control:

En este módulo se comprobó el funcionamiento de nuestra máquina de estados.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración 14. Verificación del módulo de control

* Testbench módulo de multiplicador más acumulador:

Esta es la celda básica de un filtro secuencial. Se verifico la respuesta de este módulo simulando las dos señales de control que se le inyectan. Una es la señal para realizar el reset del acumulador y la otra es para habilitar el acumulador y este aumente al recibir una muestra.

Imagen que contiene Diagrama

Descripción generada automáticamente

Ilustración 15. Verificación del multiplicador + acumulador

* Testbench módulo de desplazamiento y multiplexor:

En este módulo se verifico la carga del nuevo dato y que se pudiera direccionar el bus del registro de desplazamiento correctamente.

Escala de tiempo

Descripción generada automáticamente con confianza media

Ilustración 16. Verificación del registro de desplazamiento y el multiplexor de salida

* Testbench modulo filtro de compensación (TOP):

Para la verificación del top se compró como a distintas frecuencias este realmente compensara al filtro CIC para obtener una banda pasante con una ganancia constante

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 17. Verificación de nuestro filtro compensador (entrada: sinusoidal 10KHz)

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 18. Respuesta al impulso de nuestro filtro (TOP)

# Resolución problemas encontrados

En el desarrollo de nuestro modulo se tubo como principal problema el ajuste de las señales de control, estas ocasionaban que el módulo MULT\_ACC no funcionase correctamente. Para solucionar esto simplemente se ajustaron las señales con la simulación.

Como nota adicional cabe mencionar que se intento diseñar el módulo de la memoria ROM con memorias M9K sin embargo no se pudo lograr esto y se diseño con lógica.