

# E4

## Filtros compensadores

Esta práctica se divide en dos partes, la primera es obligatoria mientras que la segunda es voluntaria.

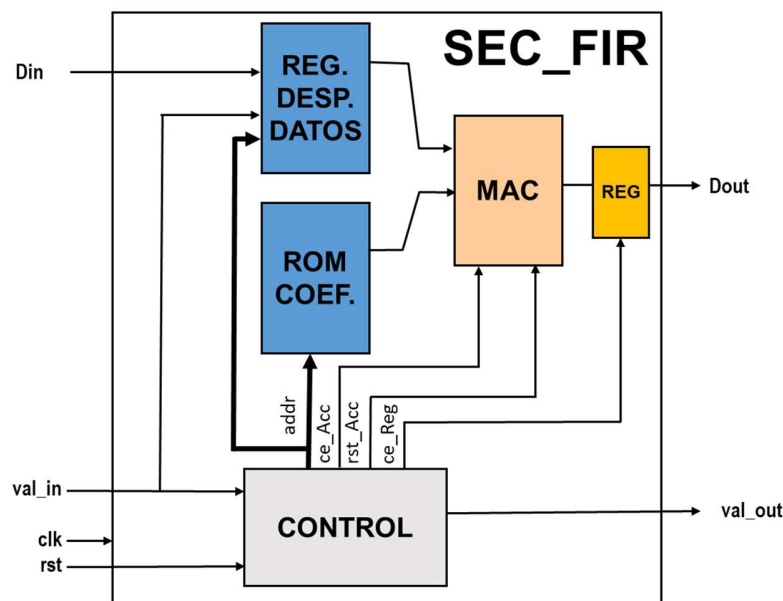
E4.1 Implementación de un filtro FIR secuencial que compensará la respuesta del CIC

E4.2 Implementación de un filtro FIR paralelo que compensará la respuesta del convertidor digital analógico (DAC).

### E4.1 Filtro FIR compensador del CIC

El **objetivo** de esta parte es el modelar con el lenguaje Verilog un filtro **FIR secuencial** que compensará la respuesta en frecuencia del CIC. La frecuencia de muestreo de los datos de entrada será de 50kHz, mientras que la frecuencia de reloj del filtro será de 100MHz.

A continuación, se presenta el **esquema de implementación del filtro SEC\_FIR**.



*Filtro FIR secuencial compensador del CIC*

El módulo SEC\_FILTER dispone de los siguientes puertos:

- Din: entrada del dato cuantificado con formato signed [Win, Win-1], siendo Win=16 bits
- val\_in: entrada binaria que informa de que existe una muestra válida en Din
- rst: reset del sistema

- clk: entrada de reloj
- Dout: salida del dato filtrado, con formato signed [Wout, Wout-3] siendo Wout=19 bits
- val\_out: salida binaria que informa de que existe una muestra válida en Dout

Las especificaciones del filtro son:

- Arquitectura secuencial
- Número de etapas: N=17
- Coeficientes h\_comp simétricos (proporcionados en el fichero Param\_comp\_and\_CIC.m)
- Tamaño de la entrada: Win = 16 bits
- Tamaño de la salida Wout = 19 bits
- El modelo del filtro deberá ser sintetizable y podrá ser implementado en un dispositivo FPGA Cyclone IV EP4CE115F29C7 funcionando a una frecuencia de reloj de 125 MHz.
- El ancho de banda de la señal de entrada será de 15kHz.

**Ficheros** necesarios para realizar las prácticas:

- Filtro\_compensaCIC.mdl: Modelo Simulink del filtro compensador
- Param\_compensaCIC.m: función Matlab para configurar el modelo anterior
- REG\_MUX.v: Fichero con el interfaz Verilog del registro de desplazamiento y del multiplexor.
- \*TB\_REG\_MUX.v: Test-bench del modelo REG\_MUX
- MULT\_ACC.v: Fichero con el interfaz Verilog del multiplicador y del acumulador
- \*TB\_MULT\_ACC.v: Test-bench del multiplicador más acumulador
- ROM.v: Fichero con el interfaz Verilog de la memoria ROM
- \*TB\_ROM.v: Test-bench de la memoria ROM.
- CONTROL.v: Fichero con el interfaz Verilog de la máquina de estados necesaria para implementar el control del circuito
- \*TB\_CONTROL.v: Test-bench de la máquina de estados CONTROL
- SEC\_FILTER.v: Fichero con el interfaz Verilog del filtro completo
- \* SEC\_FILTER.sdc: Fichero con las restricciones temporales del filtro
- \*TB\_SEC\_FILTER.v: banco de pruebas del módulo SEC\_FILTER
- \*TB\_SEC\_FILTER.do: configuración de las formas de onda del banco de pruebas del módulo TB\_SEC\_FILTER en el visor "waves"
- \*Wrap\_SEC\_FILTER.v: fichero para medir la velocidad del circuito

Los ficheros marcados con \* no se proporcionan, el resto se proporcionan incompletos.

### Tareas a realizar

- 1) Revise la nota de aplicación de Altera nº 455 "Understanding CIC compensation Filters" y responda a las siguientes preguntas:
  - ¿Por qué necesitamos un filtro compensador de la respuesta del CIC ?

- Represente la respuesta en frecuencia del filtro compensador y del filtro CIC
- 2) Utilice el modelo Simulink proporcionado (Filtro\_compensaCIC.mdl) para realizar un estudio de la respuesta del filtro. Para ello, haga una copia del modelo proporcionado (sin cuantificar) y aplíquese la cuantificación oportuna para obtener la salida con precisión completa. Compare el modelo sin cuantificar con el modelo cuantificado. A partir de ahí, obtenga en primer lugar la respuesta al impulso y después realice un barrido en frecuencia, con una señal sinusoidal de amplitud 1 V y frecuencia en el rango de 1kHz a 15kHz. Observe el crecimiento de la señal a la salida. Explique en la memoria las simulaciones realizadas y la conclusión a la que ha llegado tras este análisis.
- 3) Revise las transparencias del tema y justifique la elección de la arquitectura secuencial para implementar el filtro teniendo en cuenta que la frecuencia de muestreo es 50kHz y la frecuencia de reloj 100MHz. Dibuje esta arquitectura.
- 4) Implemente en Verilog y testee los siguientes módulos. Incluya en la memoria una breve descripción de cada uno, una muestra del test realizado y el número de recursos hardware (LE) que necesitará cada uno. Tenga en cuenta para su diseño que el filtro debe operar a la fclk de 125 MHz.
  - a. REG\_MUX: Incluye el registro de desplazamiento y el multiplexor de salida. Se parametriza el número de bits del dato de entrada (Win). Posee dos entradas de control, ce que habilita la carga de un nuevo dato, y sel que será el bus que direcciona el registro de desplazamiento. Realizar el test\_bench TB\_REG\_MUX.v para comprobar el comportamiento del circuito.
  - b. MULT\_ACC: celda básica de un filtro secuencial (multiplicador más acumulador). Se parametriza el tamaño las dos entradas (Win: cuantificación de los datos, Wc: cuantificación de los coeficientes). Posee dos señales de control, una para hacer el reset del acumulador (síncrono y activo a nivel alto) y otra para habilitar el inicio de la acumulación cada vez que entra una nueva muestra (activo a nivel alto). Realizar el test\_bench TB\_MULT\_ACC.v para comprobar el comportamiento del circuito.
  - c. ROM: modelo de la memoria ROM. Se parametriza la cuantificación de los coeficientes (Wc). Posee una entrada de control dedicada al direccionamiento de los coeficientes (addr). Realizar el test\_bench TB\_ROM.v para comprobar el funcionamiento de este modelo.
  - d. CONTROL: máquina de estados que controla el direccionamiento de la memoria ROM, el funcionamiento de la celda multiplicador acumulador y el registro de salida del filtro. Realizar el testeo completo de la máquina de estados (TB\_CONTROL.v).
  - e. SEC\_FILTER: modelo completo del filtro secuencial. Testearlo comparando con la salida obtenida del modelo Simulink, en primer lugar, con la respuesta al impulso y, tras

comprobar que funciona correctamente, incluir señales sinusoidales de distinta frecuencia hasta 15kHz.

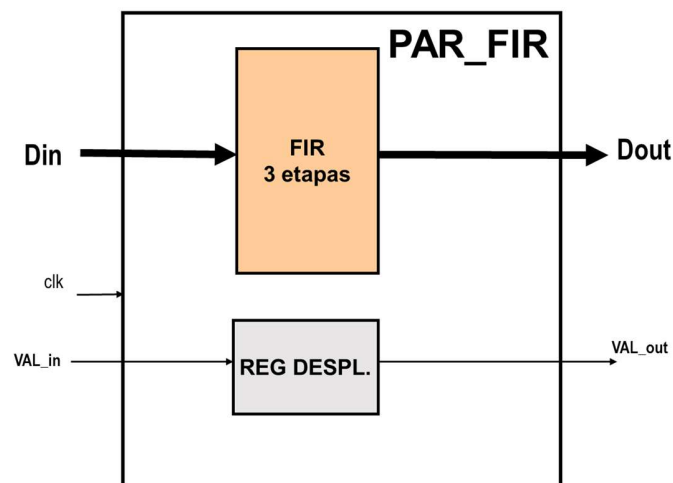
- 5) Escriba un código Verilog con un “wrap” para el módulo SEC\_FILTER y obtenga la frecuencia máxima de funcionamiento en el dispositivo indicado.
- 6) Realice la simulación tras el emplazado y rutado del circuito (simulación a nivel de puertas) para comprobar el correcto funcionamiento del circuito una vez implementado dentro de la FPGA.
- 7) Escribir el documento final teniendo en cuenta las **Consideraciones generales aplicables a todas las prácticas**. Puede incluir en la Sección 1 del documento toda la información que considere oportuna en relación a las tareas 1 y 2.

Entregue la memoria en la fecha prevista, aunque no haya podido terminar todos los apartados.

## E4.2 Filtro FIR compensador del DAC

El **objetivo** de esta parte es el modelar con el lenguaje Verilog un filtro **FIR paralelo** que compensará la respuesta en frecuencia del DAC. La tasa de muestreo de la señal de entrada coincide con la de reloj y debe llegar a 125MHz.

A continuación, se presenta el **esquema de implementación del filtro FIR**.



El módulo PAR\_FILTER dispone de los siguientes puertos:

- Din: entrada del dato cuantificado con formato signed [Win, Win-1], siendo Win=16 bits
- VAL\_in: entrada binaria que informa de que existe una muestra válida en Din
- rst: reset del sistema
- clk: entrada de reloj
- Dout: salida del dato filtrado, con formato signed [Wout, Wout-3] siendo Wout=19 bits

- VAL\_out: salida binaria que informa de que existe una muestra válida en Dout

Las especificaciones del filtro son:

- Arquitectura paralela
- Número de etapas:  $N=3$
- Coeficientes  $h_{dac}$  simétricos (proporcionados en el fichero Param\_compensaDAC.m)
- Tamaño de la entrada:  $W_{in} = 16$  bits
- Tamaño de la salida  $W_{out} = 19$  bits
- El modelo del filtro deberá ser sintetizable y podrá ser implementado en un dispositivo FPGA Cyclone IV EP4CE115F29C7 funcionando a una frecuencia de reloj de 125 MHz.
- El ancho de banda de la señal de entrada será de 15kHz.

**Ficheros** necesarios para realizar las prácticas:

- Filtro\_compensaDAC.mdl : Modelo Simulink del filtro compensador
- Param\_compensaDAC.m: función Matlab para configurar el modelo anterior (hay que hallar la cuantificación de salida  $W_{out}$  con precisión completa)
- \*PAR\_FILTER.v: Modelo Verilog del filtro compensador
- \*TB\_PAR\_FILTER.v: banco de pruebas del módulo PAR\_FILTER
- \*WRAP\_PAR\_FILTER.v: modelo para medir la máxima frecuencia de reloj.

Los ficheros marcados con \* no se proporcionan, el resto se proporcionan incompletos.

### Tareas a realizar

- 1) Utilice el modelo Simulink proporcionado (Filtro\_compensaDAC.mdl) para realizar un estudio de la respuesta del filtro y la cuantificación a aplicar en el mismo. Para ello obtenga en primer lugar la respuesta al impulso y después realice un barrido en frecuencia, en el rango de 1kHz a 15kHz, y observe la señal a la salida. Explique en la memoria las simulaciones realizadas y la conclusión a la que ha llegado tras este análisis. Explique cómo ha decidido la cuantificación de salida del filtro.
- 2) Revise las transparencias del tema y busque la arquitectura adecuada para implementar el filtro paralelo. Dibuje esta arquitectura indicando claramente la segmentación del circuito. Tenga en cuenta para su diseño que el filtro debe operar a la  $f_{clk}$  de 125 MHz.
- 3) Implemente en Verilog y testee el módulo PAR\_FILTER. Incluya en la memoria una breve descripción, una muestra del test realizado y el número de recursos hardware (LE) que necesitará.
- 4) Escriba un código Verilog con un “wrap” para el módulo PAR\_FILTER y obtenga la frecuencia máxima de funcionamiento en el dispositivo indicado.

- 5) Escribir el documento final teniendo en cuenta las **Consideraciones generales aplicables a todas las prácticas**. Puede incluir en la Sección 1 del documento toda la información que considere oportuna en relación a las tareas