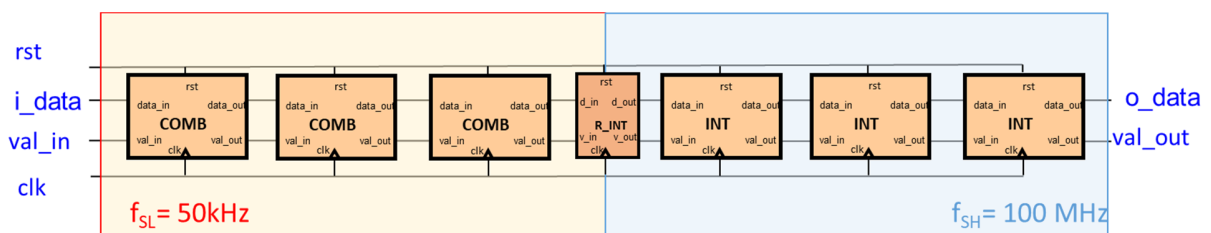


E3

Filtro Interpolador: CIC

El **objetivo** de esta práctica es el modelar con el lenguaje Verilog un filtro CIC interpolador por 2000 y con entradas y salidas de ancho parametrizable, capaz de cambiar la tasa de muestro de la señal de 50 KHz a 100 MHz.

A continuación se presenta el **esquema de implementación del CIC**.



Filtro interpolador CIC

El módulo CIC dispone de los siguientes puertos:

- i_data: 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 i_data
- rst: reset del sistema activo a nivel alto
- clk: entrada de reloj
- o_data: salida del dato interpolado, con formato signed [Wout, Wout-1] siendo Wout=16 bits
- val_out: salida binaria que informa de que existe una muestra válida en o_data

Las especificaciones del filtro CIC son:

- Número de etapas: $N=3$
- Retardo de las etapas peine (comb) $M=1$
- Factor de interpolación: $R=2000$
- Tamaño de la entrada: Win = 16 bits
- Crecimiento de los datos Wg (a calcular)
- El modelo del CIC 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:

- CIC.mdl : Modelo Simulink de referencia del filtro interpolador CIC de 3 etapas. Duplicar el bloque CIC Interpolador Modelo pc para implementar el modelo con precisión recortada con el que hacer el estudio de la cuantificación
- Test_CIC.m: función Matlab para testear el circuito y comparar la simulación Simulink con la de Verilog.
- COMB.v: Fichero con el interfaz Verilog del filtro peine
- *TB_COMB.v: Test-bench del filtro peine
- INT.v: Fichero con el interfaz Verilog del filtro Integrador
- *TB_INT.v: Test-bench del filtro integrador
- R_INT.v: Fichero con el interfaz Verilog del módulo expansor y generador de la señal de validación
- *TB_R_INT.v: Test-bench del modelo R_INT.
- CIC_pc.v: Fichero con el interfaz Verilog del filtro CIC con salida con precisión completa (Win+Wg)
- *TB_CIC_pc.v: banco de pruebas del módulo CIC con precisión completa
- *TB_CIC_pc.do: configuración de las formas de onda del banco de pruebas del módulo TB_CIC en el visor "waves"
- *CIC.v: Fichero con el interfaz Verilog del filtro CIC con salida con precisión recortada a 16 bits
- *CIC.sdc: Fichero con las restricciones temporales del filtro CIC
- *TB_CIC.v: banco de pruebas del módulo CIC
- *TB_CIC.do: configuración de las formas de onda del banco de pruebas del módulo TB_CIC en el visor "waves"

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

Tareas a realizar

- 1) Estudie el material proporcionado para entender el funcionamiento del filtro interpolador CIC. Responda en la memoria a las siguientes preguntas:
 - ¿Qué es un filtro CIC?
 - ¿Cuál es la ganancia de un filtro CIC interpolador?
 - ¿Para qué lo estamos utilizando en nuestro proyecto?
 - ¿Qué ventajas introduce frente a otras posibles soluciones?
- 2) Utilice el modelo Simulink del filtro (CIC.mdl) para realizar un análisis del crecimiento de la ruta de datos. En este fichero se proporciona el modelo con precisión completa, duplíquelo para implementar el modelo cuantificado.

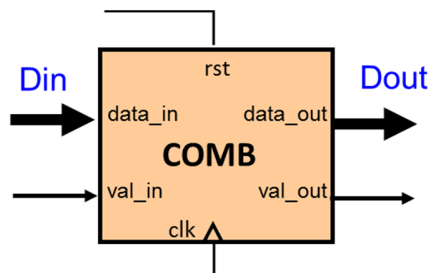
Estime la cuantificación adecuada de la ruta de datos para evitar que se produzca desbordamiento o que haya pérdidas de precisión a la salida de las etapas peine y a la salida del filtro.

Complete los operadores del modelo cuantificado con la cuantificación correcta. Tras ello, simule el funcionamiento comparando las salidas del bloque de referencia con el bloque cuantificado. Para ello pruebe a introducir una señal sinusoidal a la entrada del filtro, varíe la frecuencia f_0 de dicha señal en el rango de 1kHz a 15kHz, y observe la señal a la salida de la última etapa peine y a la salida del filtro.

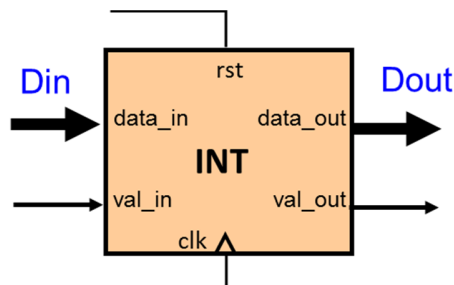
Testee el sistema con otros tipos de señales (cuadrada, triangular etc.) para comprobar que funciona correctamente. Explique en la sección 1 de la memoria las simulaciones realizadas y la conclusión a la que ha llegado tras este análisis.

- 3) Implemente en Verilog y testee los siguientes módulos. Aplique en ellos la cuantificación de salida obtenida en el apartado anterior. 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.

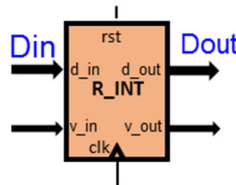
- a. COMB: celda básica de un filtro peine con $M=1$ y parametrizado con $Win=16$ bits. Realizar el test_bench TB_COMB.v para comprobar el comportamiento del circuito.



- b. INT: celda básica de un filtro interpolador parametrizado (Win y Wg) Realizar el test_bench TB_INT.v para comprobar el comportamiento del circuito.



- c. R_INT bloque que expande la salida (introduciendo R-1 ceros entre muestras) y genera la señal de habilitación de los filtros integradores para conseguir un factor de interpolación R. El circuito debe ser parametrizable con Win=16 bits.



- 4) Implemente en Verilog el filtro interpolador completo mostrado en la figura 1 (CIC_pc.v) utilizando los módulos implementados en el apartado anterior y la instrucción generate de Verilog.
Realice el testeo completo del circuito implementado con precisión completa (TB_CIC_pc.v) comparándolo con las señales obtenidas a partir del modelo simulink cuantificado del apartado 2. Utilice para ello la función Matlab Test_CIC.m excitando la entrada del filtro con un impulso, con una señal sinusoidal y con una señal cuadrada.
- 5) En el modelo Simulink: escale la salida del CIC para realizar un truncado a 16 bits aprovechando todo el rango dinámico de amplitud entre [-1, 1]. (Introducir un escalado por $2^{(-Wg)}$ y un bloque convert para transformar la salida en 16.15). Justifique por qué estamos realizando este escalado.
- 6) En el modelo Verilog: realice el módulo CIC.v a partir del CIC_pc.v, escalando la salida para realizar un truncado a 16 bits aprovechando todo el rango dinámico de amplitud entre [-1, 1]. Compare la salida con la del paso anterior realizando un testeo completo.
- 7) Estime el número de recursos hardware (LE) que se requieren para la implementación del módulo CIC. Implemente el módulo CIC en el dispositivo Cyclone IV EP4CE115F29C7 e indique los recursos hardware que se requiere en su implementación. Razone si los recursos obtenidos son coherentes con los estimados.
- 8) Escriba un código Verilog con un “wrap” para el módulo CIC y obtenga la frecuencia máxima de funcionamiento en el dispositivo indicado.
- 9) No olvide poner las conclusiones de la práctica en la que indique claramente si se han cumplido los objetivos, y en caso contrario especificar por qué no se han llegado a cumplir diciendo hasta qué punto se han conseguido.

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