

Máster Universitario en Ingeniería de Sistemas Electrónicos



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Práctica E3: Filtro Interpolador CIC

Jose Luis, Rocabado Rocha

Gianmarco Leopoldo, Sangoi Da Roza

04/08/2022

ÍNDICE

ÍNDICE	2
ÍNDICE DE FIGURAS.....	3
Descripción del módulo.....	4
Interfaz.....	5
Recursos Hardware	6
Frecuencia máxima.....	7
Camino critico	7
Verificación	8
Resolución de problemas encontrados	11

ÍNDICE DE FIGURAS

<i>Ilustración 1. Etapa COMB.</i>	4
<i>Ilustración 2. Etapa INT.</i>	4
<i>Ilustración 3. Tabla de recursos del sistema.</i>	6
<i>Ilustración 4. Frecuencia máxima de operación del sistema (F_{max}).</i>	7
<i>Ilustración 5. Camino crítico del sistema generado por el Technology Map Viewer.</i>	8
<i>Ilustración 6. Verificación etapa COMB.</i>	8
<i>Ilustración 7. Verificación Upsample.</i>	9
<i>Ilustración 8. Verificación etapa INT.</i>	9
<i>Ilustración 9. Verificación del sistema completo (TOP).</i>	10
<i>Ilustración 10. Verificación del TOP con la salida truncada a 16bits.</i>	10
<i>Ilustración 11. Problema encontrado en nuestro módulo INT.</i>	11

Descripción del módulo

El módulo diseñado consiste un filtro interpolador CIC, este módulo será implementado para la elaboración del modulador AM/FM, con el podremos cambiar la frecuencia de muestreo de 48KHz provenientes del módulo DDS (E1) a 96MHz que irán al módulo DP_MOD (E2).

La estructura del filtro interpolador CIC consta de 3 etapas:

1. Etapa de filtros peines (COMB): En esta etapa se implementa un filtro FIR denominado peine, el cual resta la entrada con la misma entrada retrasada, como se muestra en la siguiente formula:

$$H_c(z) = 1 - z^{-M}$$

Ilustración 1. Etapa COMB.

Cabe mencionar que en esta etapa se implementa un arreglo en cascada de estos filtros para obtener una mayor respuesta en frecuencia, en nuestro caso nuestra etapa COMB tiene 3 filtros.

2. Etapa de muestreo ascendente: En esta etapa se expande la salida (introduciendo ceros entre muestras) y se genera la señal de habilitación de los filtros integradores.
3. Etapa de filtros integradores: En esta última etapa se implementa el mismo número de filtros integradores que los filtros peines usados, la función de estos filtros IIR es la de sumar la entrada con la salida retrasada, como se muestra en la siguiente formula:

$$H_i(z) = \frac{1}{1 - z^{-1}}$$

Ilustración 2. Etapa INT.

Interfaz

- Módulo CIC (top):

PARÁMETROS			
Nombre	Descripción		
Win	Tamaño de la entrada		
Wg	Guard Bits		
N	Número para determinar las etapas del filtro (tamaño de los wire)		
INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Reset síncrono del filtro Interpolador CIC
val_in	in	bit	Entrada de validación de datos
i_data	in	S[M, M-1]	Entrada de datos filtro CIC
o_data	out	S[M, M-1]	Salida de datos.
val_out	out	bit	Salida de la señal de validación

- Módulos COMB, INT y R_INT:

PARÁMETROS			
Nombre	Descripción		
Win	Tamaño de la entrada		
Wg	Guard Bits		
INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Reset síncrono del uilo COMB
val_in	in	bit	Entrada de validación de datos
i_data	in	S[M, M-1]	Entrada de datos filtro CIC
o_data	out	S[M, M-1]	Salida de datos.
val_out	out	bit	Salida de la señal de validación

Recursos Hardware

Tabla de recursos del módulo DP_MOD del sintetizador	
Hardware	Cantidad
Elementos Lógicos	254
Multiplicadores	0
Memorias MK9	0

	Resource	Usage
1	▼ Total logic elements	254 / 114,480 (< 1 %)
1	-- Combinational with no register	7
2	-- Register only	36
3	-- Combinational with a register	211
2		
3	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	0
2	-- 3 input functions	181
3	-- <=2 input functions	37
4	-- Register only	36
4		
5	▼ Logic elements by mode	
1	-- normal mode	56
2	-- arithmetic mode	162
6		
7	▼ Total registers*	247 / 117,053 (< 1 %)
1	-- Dedicated logic registers	247 / 114,480 (< 1 %)
2	-- I/O registers	0 / 2,573 (0 %)
8		
9	Total LABs: partially or completely used	20 / 7,155 (< 1 %)
10	Virtual pins	0
11	> I/O pins	36 / 529 (7 %)
12		
13	M9Ks	0 / 432 (0 %)
14	Total block memory bits	0 / 3,981,312 (0 %)
15	Total block memory implementation bits	0 / 3,981,312 (0 %)

Ilustración 3. Tabla de recursos del sistema.

Los elementos lógicos del CIC serían (parámetros: Win= 16 y Wg = 22):

$$(16+17+1) + (17+18+1) + (18+19+1) + (38) + (38+1) + (38+1) + (38+1) = 263.$$

Nos sale este valor un poco más elevado porque hemos utilizado el crecimiento máximo para poder realizar la instanciación mediante el comando generate, y es posible que *Quartus* realizara una optimización.

Los logic elements en modo aritmético que serían las sumas, al calcularlas tenemos:

$$(19 \times 3) + (38 \times 3) = 171.$$

Este número si se acerca más al generado por el sintetizador

Frecuencia máxima

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

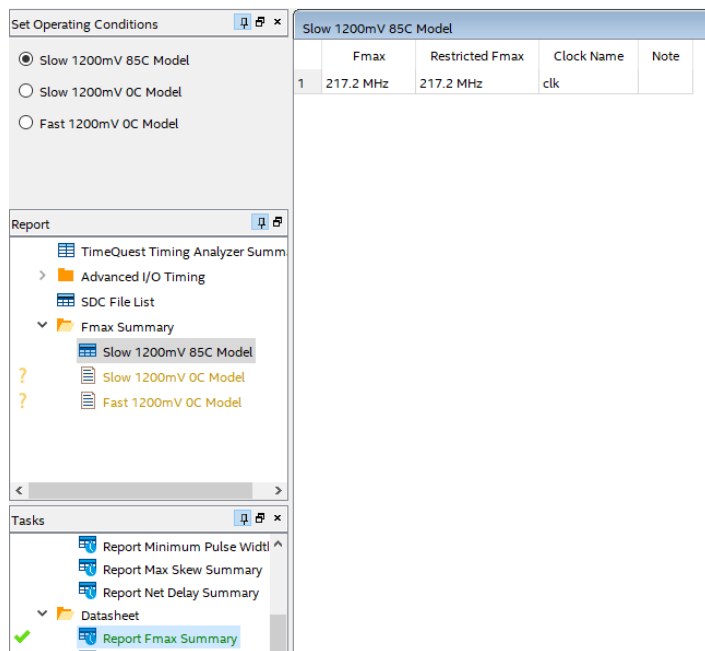


Ilustración 4. Frecuencia máxima de operación del sistema (Fmax).

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 ~217.2MHz 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.

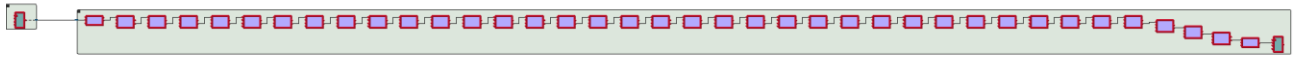


Ilustración 5. Camino crítico del sistema generado por el Technology Map Viewer.

El camino crítico de nuestro módulo CIC se encuentra en la etapa de los filtros integradores, en esta etapa hay sumas de 38 bits que producen un acarreo que tiene que recorrer 38 posiciones, esto hace mucho más largo el recorrido de la señal en la etapa INT que en la etapa de filtros peine (COMB).

Verificación

Para la verificación de las celdas básicas se generan los *TestBench* correspondientes que de forma sencilla nos permiten validar la ejecución correcta de las operaciones.

Para la celda “peine”, se comprobará que la resta de la entrada con la entrada anterior se realiza cada vez que “val_in” se encuentra a nivel alto. Se consideran suficientes 4 datos de entrada.

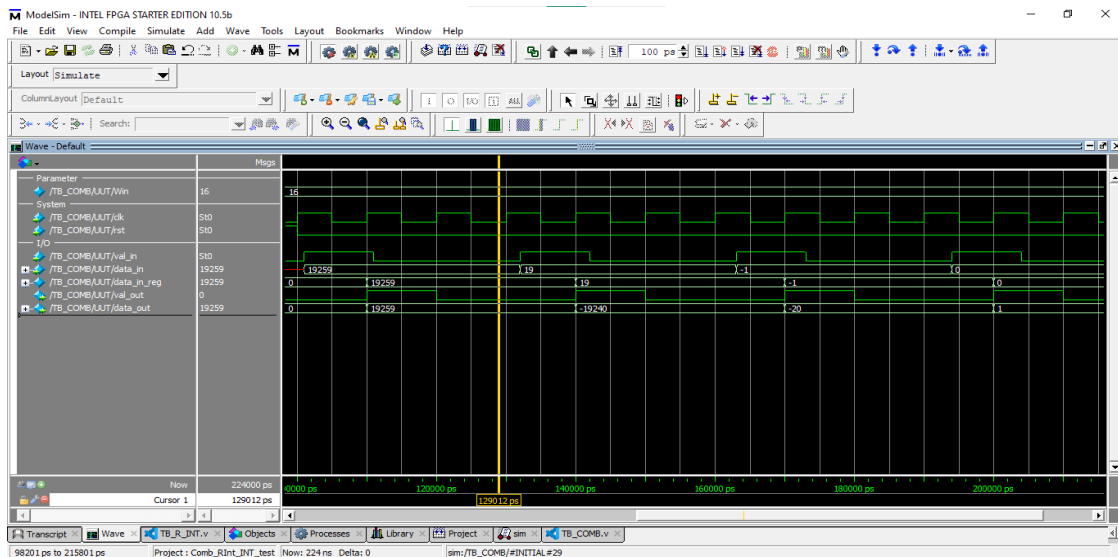


Ilustración 6. Verificación etapa COMB.

Para la etapa de *upsampling* el TB generará un dato cada “R_Value” ciclos de reloj obteniendo así un $f_{SL} = f_{CLK} / R_value$. Posteriormente, con un periodo de muestreo de $f_{SH} = f_{CLK}$ se rellenará con $R_value - 1$ ceros. Para agilizar la simulación de la validación del módulo, se utilizará un factor de *upsampling* de 5 tal y como se observa en la siguiente figura.

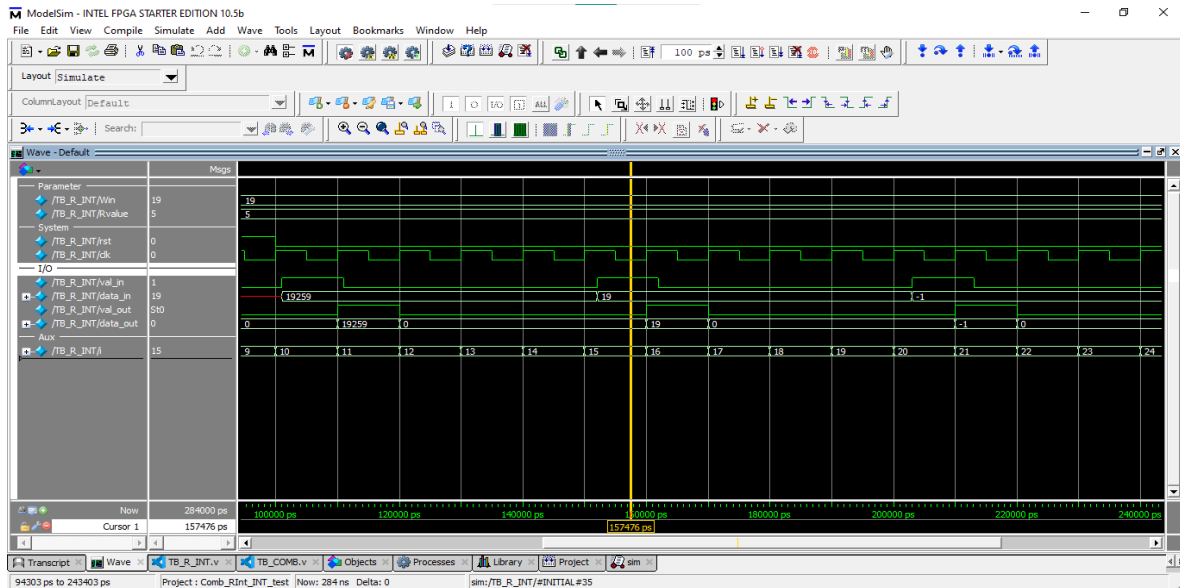


Ilustración 7. Verificación Upsample.

Por último, la celda básica integradora consistirá en un acumulador. Se debe comprobar que la entrada se suma con la anterior salida. Sin embargo, dado que se trata de un sistema síncrono, la anterior salida se corresponde directamente con la salida registrada (al igual que un acumulador).

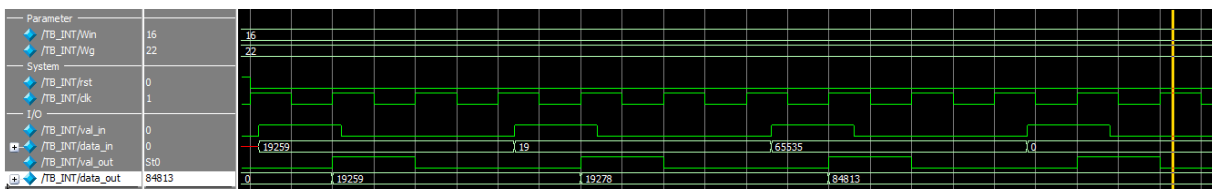


Ilustración 8. Verificación etapa INT.

A continuación, se realiza la validación del módulo *top* “CIC_pc.v” que contiene el filtro completo con las etapas peine/integración y *upsampling* correspondiente. Para validar el diseño, se utilizará el *golden model* con cuantificación generado mediante Matlab y con diferentes señales de entrada. En la siguiente figura podemos observar el resultado para la validación con una señal de entrada de tipo senoidal donde se comprueba además que la frecuencia de muestreo de las entradas es de 50kHz.

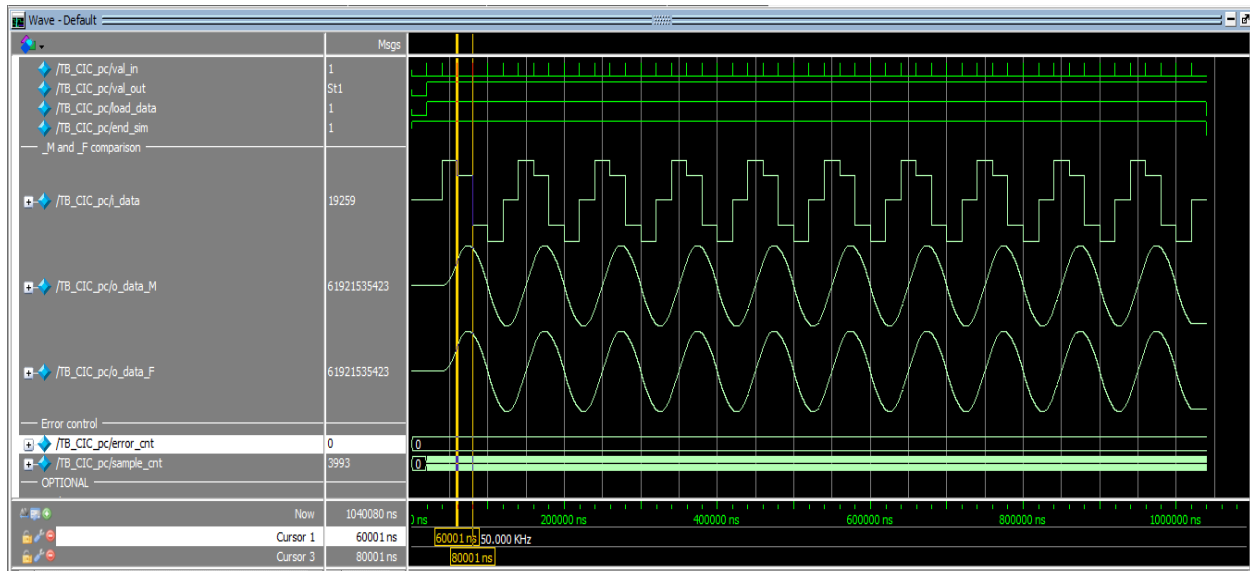


Ilustración 9. Verificación del sistema completo (TOP).

Una vez testeado nuestro modulo top con todas las etapas se procedió a diseñar la salida truncada de 16 bits, esta salida necesita un escalado (realizado en *Simulink*) para que no se produzca overflow ya que necesitamos quedarnos con los bits más significativos. En el fichero MATLAB establecimos una condición para poder probar la salida truncada o la de full precisión según se desee. Una vez obtenemos nuestro *golden model* de la salida truncada se validó el diseño de la misma forma que con el top anterior.

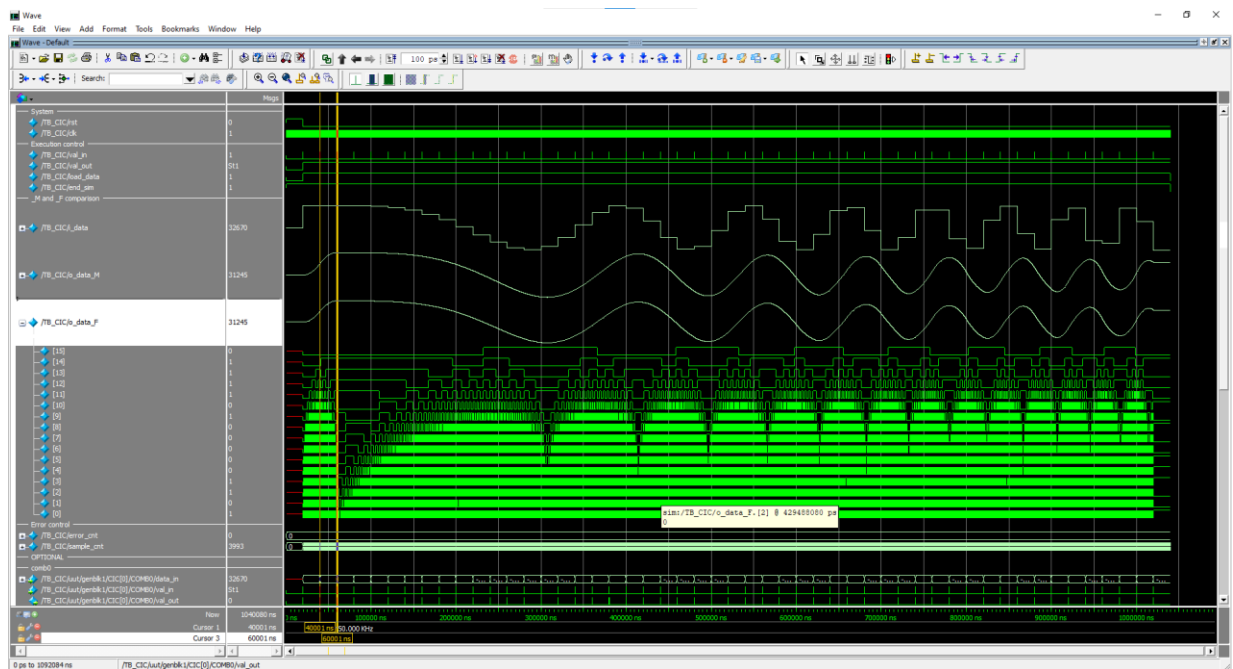
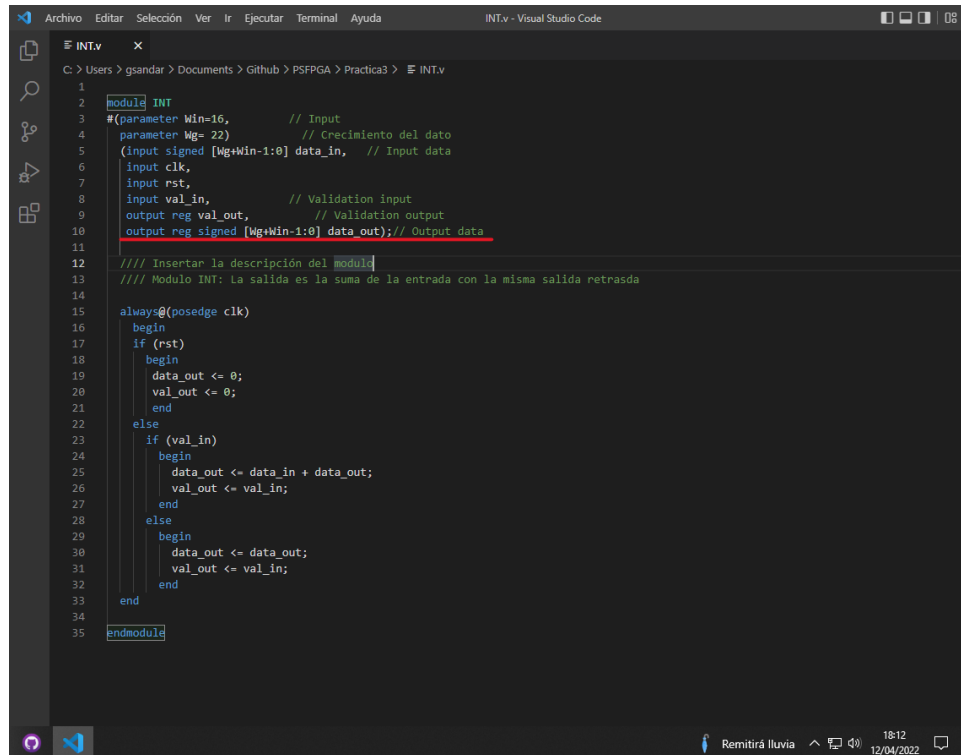


Ilustración 10. Verificación del TOP con la salida truncada a 16bits.

Resolución de problemas encontrados

El único problema que tuvimos fue en el módulo de integración, al verificarlo con el testbench elaborado notamos que teníamos un registro adicional y esto nos generaba una oscilación en la señal de salida porque aumentaba la latencia en 1. Para solucionarlo simplemente eliminamos este registro ya que estábamos registrando la salida de la suma dos veces.



```
1 module INT
2
3   #(parameter Win=16,           // Input
4     parameter Wg= 22)          // Crecimiento del dato
5   (input signed [Wg+Win-1:0] data_in, // Input data
6
7     input clk,
8     input rst,
9     input val_in,              // Validation input
10    output reg val_out,         // Validation output
11    output reg signed [Wg+Win-1:0] data_out); // Output data
12
13    //// Insertar la descripción del módulo
14    //// Módulo INT: La salida es la suma de la entrada con la misma salida retrasada
15
16    always@(posedge clk)
17    begin
18      if (rst)
19      begin
20        data_out <= 0;
21        val_out <= 0;
22      end
23    else
24      if (val_in)
25      begin
26        data_out <= data_in + data_out;
27        val_out <= val_in;
28      end
29    else
30      begin
31        data_out <= data_out;
32        val_out <= val_in;
33      end
34    end
35  endmodule
```

Ilustración 11. Problema encontrado en nuestro módulo INT.