Índice

1. Introducción

En esta memoria vamos a tratar un proyecto realizado para la asignatura *Diseño de Circuitos y Sistemas Electrónicos* impartida en la ETSI de Telecomunicación de la Universidad Politécnica de Madrid.

El proyecto se propone como trabajo adicional a la asignatura, para tratar los temas de diseño de circuitos digitales de una forma más práctica y obtener conocimientos mucho más amplios sobre tecnologías relacionadas con estos campos.

En concreto, la propuesta consiste en el diseño y simulación de un sistema utilizando VHDL. El sistema propuesto consiste en un ecualizador de audio, del cual se implementarán algunos subsistemas de procesado digital en VHDL. Esta práctica se apoya sobre el trabajo realizado en el año anterior en el *Laboratorio de Sistemas Electrónicos Digitales*, donde se realizó este sistema utilizando un microcontrolador.

El esquema de dicho sistema puede verse en la figura 1. El proyecto que nos ocupa se centrará en la realización del subsistema de procesado digital de audio, cuyo esquema puede verse en la figura 2.

Para la realización del proyecto se ha utilizado la herramienta ModelSim. De forma auxiliar, se utilizará la herramienta MATLAB para el tratamiento de las señales obtenidas de forma sencilla y eficiente, para poder así evaluar el funcionamiento del sistema.

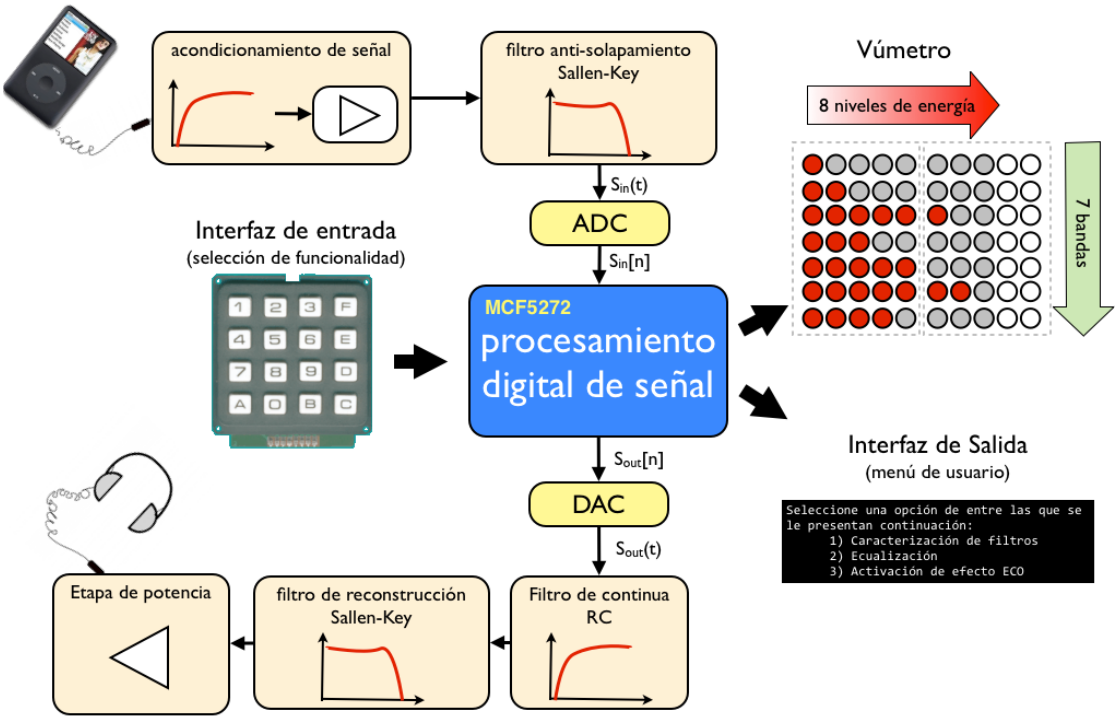

Figura 1: Descripcción del sistema completo

1.1. Descripción de los módulos VHDL

El subsistema digital representado en la figura 2 se va a dividir en los siguientes módulos principales VHDL:

Banco de filtros Este módulo incluye los 7 filtros digitales que separarán la señal de entrada en sus diferentes bandas de frecuencia. Incluye 7 submódulos correspondientes a los 7 filtros digitales. Además, este módulo ofrece la funcionalidad de selección de ganancia, como se explicará posteriormente.

Módulo de reverberación Este módulo sirve para retardar y atenuar una señal. Se utilizará para realimentar en el banco de filtros la salida retardada hasta la entrada, para generar un efecto de reverberación. El subsistema de retardo se ha implementado independientemente como un submódulo.

Vúmetro Proporciona información sobre el nivel de señal de cada una de las bandas de audio. Consiste en 7 elementos vúmetro individuales que se agrupan para obtener el vúmetro de 7 bandas.

Todo ello se ha agrupado en un único módulo *ecualizador*, para facilitar la simulación y la ejecución de pruebas. La relación de módulos y señales puede verse

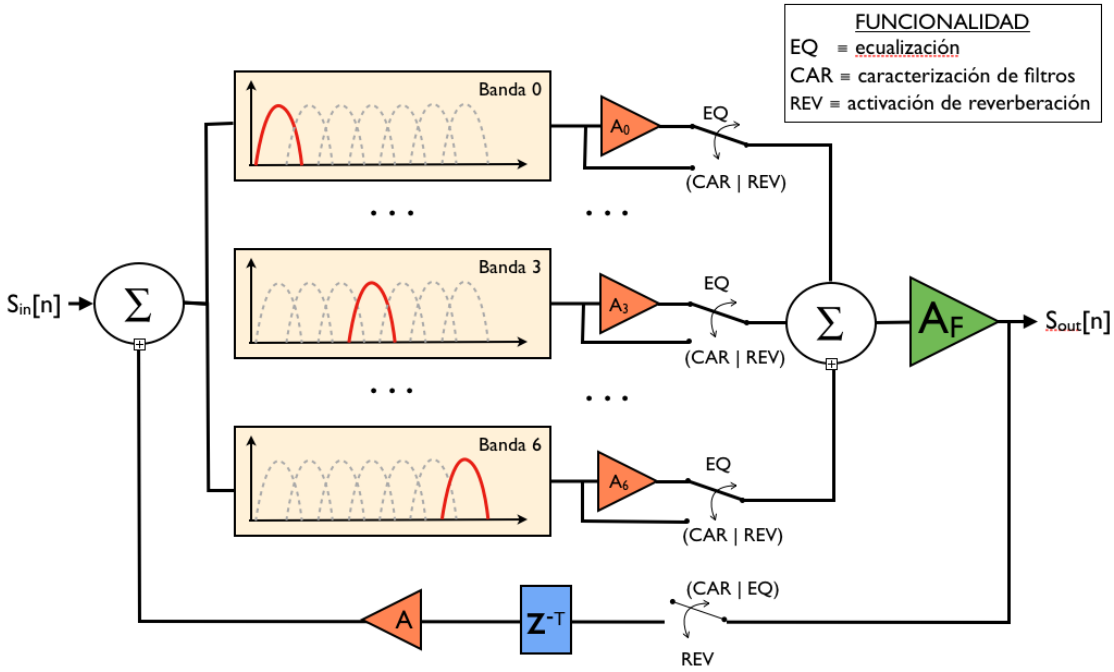

Figura 2: Descripcción del subsistema de procesado digital

en la figura 3.

Por último, se han implementado sumadores y multiplicadores con las estructuras estudiadas en clase. Como mejora, se han sustituido las operaciones de suma y multiplicación de señales en el sistema mediante funciones VHDL (operadores + y *) por sumas y productos obtenidos mediante sumadores y multiplicadores más realistas.

2. Diseño de banco de filtros

El sistema propuesto se trata de un ecualizador de audio, por lo que las señales de entrada que tendremos estarán comprendidas en el rango de frecuencias audibles por el hombre. En concreto, la especificación del sistema propuesto propone señales cuya frecuencia estará comprendida aproximadamente entre los 22 Hz y los 2.8 KHz.

Este rango de frecuencias se dividirá en 7 bandas, para lo cual utilizaremos filtros IIR de segundo orden. Las bandas de frecuencias se detallan en la tabla 2, y los coeficientes para la realización de los filtros IIR en la tabla 2.

Banda	f_0	f_{c1}	f_{c2}
0	31.25	22.10	44.19
1	62.5	44.19	88.39
2	125	88.39	176.78
3	250	176.78	353.55
4	500	323.55	704.11
5	1000	707.11	1414.21
6	2000	1414.21	2828.43

Cuadro 1: Descripción de las distintas bandas y sus filtros asociados

Filtro	Ganancia	a_0	a_1	a_2	b_0	b_1	b_2
0	8	1024	-2029	1006	1024	0	-1024
1	17	1024	-2011	988	1024	0	-1024
2	34	1024	-1970	955	1024	0	-1024
3	66	1024	-1878	890	1024	0	-1024
4	125	1024	-1660	772	1024	0	-1024
5	227	1024	-1115	569	1024	0	-1024
6	392	1024	141	239	1024	0	-1024

Cuadro 2: Descripción de los coeficientes de los filtros IIR

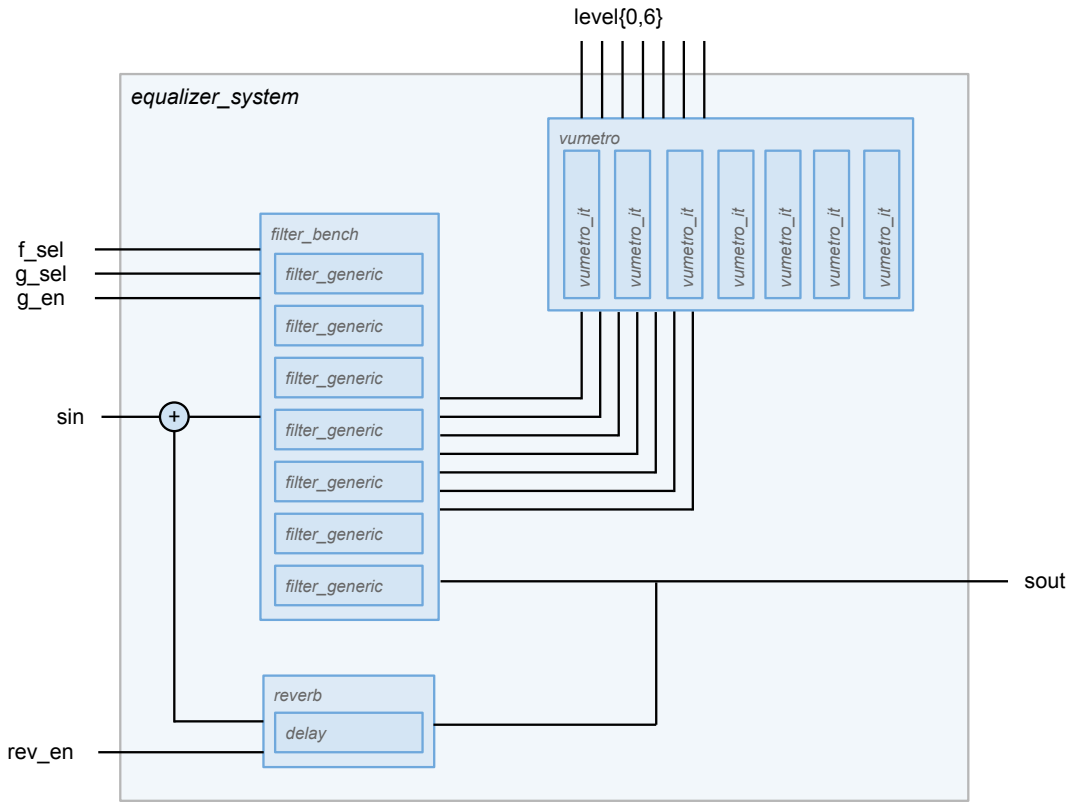

Figura 3: Relación de módulos VHDL

2.1. Implementación de los filtros en VHDL

Para una implementación más sencilla de los filtros, utilizaremos la *Forma directa II*. El detalle de esta implementación se puede ver en la figura 4.

A la hora de implementar estos filtros digitales, tenemos que tener en cuenta una limitación muy importante. Debemos definir un ancho de palabra fijo para la representación de las señales de entrada y de salida. En nuestro caso, hemos escogido un ancho de palabra de *16 bits*, correspondientes a *6 bits enteros* y *10 bits fraccionarios*. Esto es especialmente conveniente a la hora de representar los coeficientes de los filtros, ya que para todos ellos tenemos que $a_0 = 1024$. Normalizar los coeficientes realizando una división por 1024 equivale a desplazar los bits 10 lugares a la derecha (o lo que es lo mismo, la coma decimal 10 lugares a la izquierda). De esta forma, cuando representemos los coeficientes enteros como palabras binarias de 16 bits, bastará como tomar los 10 bits menos significativos como fraccionarios para tener el coeficiente normalizado a 1024.

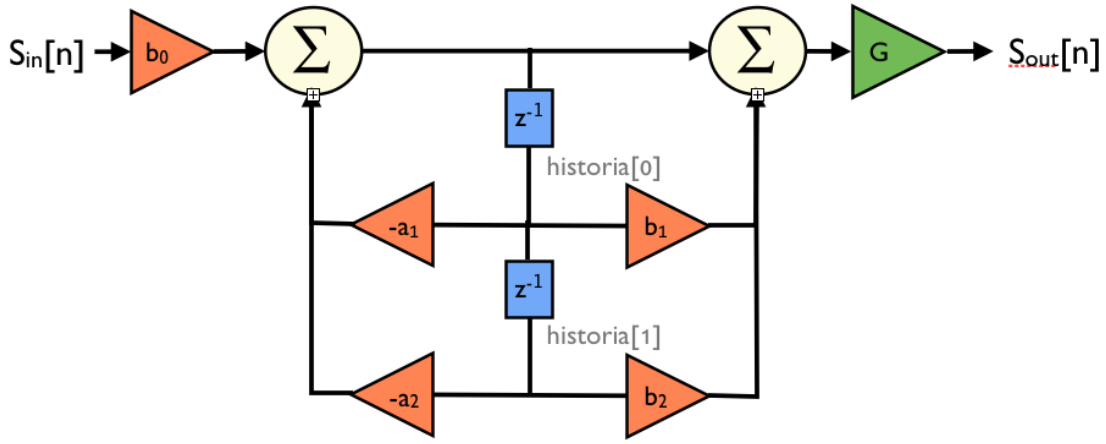

Figura 4: Diagrama de bloques de un filtro IIR de segundo orden

2.2. Ruido de cuantificación

El haber elegido un ancho de palabra de 16 bits con 6 bits enteros y 10 fraccionarios nos impone una limitación considerable en cuanto a la resolución de nuestro sistema a la hora de representar señales. Sobretodo en señales de amplitud baja, tendremos un ruido de cuantificación que puede llegar a influir en el correcto funcionamiento de nuestro sistema.

Para intentar minimizar el ruido de cuantificación, se puede intentar ajustar más el numero de bits enteros, ya que las señales con las que trabajaremos estan normalizadas a 1, y a priori debería bastarnos con un bit entero y 15 fraccionarios. Este enfoque se ha intentado realizar sin éxito en la implementación, ya que aunque las señales de entrada esten normalizadas a 1, en puntos intermedios de los filtros algunas señales pueden tomar valores más altos, produciendo desbordamiento. Por ello, se ha decidido tomar una postura más conservadora y mantener la representación con 10 bits fraccionarios, que aunque no es óptima en sentido del ruido de cuantificación, nos ofrece mayor seguridad frente a desbordamiento.

Otra forma de abordar este problema sería utilizar palabras de 32 bits en lugar de 16. Esto nos ofrecería una resolución enormemente mayor, mejorando sensiblemente la calidad del sistema. No obstante, trabajar con palabras de 32 bits supone una complejidad mucho mayor a la hora de implementar los sumadores y multiplicadores vistos en clase. Debido a que el objetivo del proyecto es mayormente didáctico y no obtener una calidad de audio excepcional, se ha decidido mantener las palabras de 16 bits y asumir los efectos del ruido de cuantificación.

2.3. Ajuste de ganancias

Además de separar la señal en 7 bandas de frecuencias, el banco de filtros de nuestro sistema debe aplicar a cada uno una ganancia ajustable y disponer a la salida de la señal global sumada.

La selección de ganancia para cada banda se realiza mediante 3 entradas adicionales:

Entrada f_sel Entrada de 3 bits mediante la cual seleccionamos el filtro (de 0 a 6) cuya ganancia queremos ajustar.

Entrada g_sel Entrada de 4 bits mediante la cual seleccionamos la ganancia que queremos aplicar al filtro seleccionado. Estas ganancias se encuentran definidas en el propio sistema en forma de tabla, de forma que a cada uno de los valores 0 a 15 le corresponde un valor de ganancia predefinido.

Entrada g_en Señal de *enable* del sistema de selección de ganancia. Cuando se detecta un flanco de subida se aplica la ganancia seleccionada al filtro seleccionado.

Para evitar desbordamiento en la señal al aplicar la ganancia, trabajaremos con posibles valores de ganancia menores que 1 (atenuaremos la señal). De esta forma, la ganancia numero 0 corresponde a una ganancia de 0dB, y de ahí se irá bajando a medida que aumente el índice de la ganancia seleccionada.

3. Diseño de módulo de reverberación

El subsistema de reverberación consiste en un módulo que ofrece a la salida la señal que obtiene a la entrada con un retraso de N posiciones y aplicandole una atenuación determinada.

Para la implementación de este módulo se ha utilizado como base una cola FIFO, en la cual se va introduciendo la señal que posteriormente se va obteniendo por la salida tras N ciclos de reloj. Para ello es importante inicializar la cola para que esté llena de palabras a cero, ya que si no la primera muestra de la señal de entrada ocuparía la primera posición y sería la que se ofrecería a la salida en el siguiente ciclo de reloj.

En nuestro sistema, el numero de palabras en la cola será siempre fijo e igual al número de muestras que queramos retrasar la señal. La implementación de una cola FIFO tiene cierta complejidad derivada del hecho de que el numero de palabras en la cola es variable y puede ser impredecible, conllevando posibles problemas de desbordamiento o cola vacía. Sin embargo, puesto que el funcionamiento de nuestro retardador es mucho más predecible (llegará una muestra por cada ciclo

de reloj y sacaremos una muestra por cada ciclo de reloj, resultando en un tamaño constante) podemos simplificar considerablemente este diseño.

Se ha implementado para ello el subsistema *delay*, que consiste en una memoria de N posiciones (tamaño configurable mediante genéricos) con un puntero señalando a una de esas posiciones. En cada ciclo de reloj, se sacará la señal de la posición indicada por el puntero y se añadirá en su lugar la señal que llegue a la entrada. El valor del puntero se va aumentando de forma cíclica, de forma que la señal que acabamos de introducir no se sacará hasta que el puntero vuelva a apuntar a esa misma posición, es decir, tras N ciclos de reloj.

Posteriormente, la señal se atenúa mediante un valor de ganancia también configurable mediante genéricos, y se realimenta a la entrada del banco de filtros.

4. Diseño de vúmetro

El vúmetro ofrece información sobre el nivel de señal disponible en cada una de las bandas de señal. Para abordar la implementación de este módulo, se ha implementado un subsistema vúmetro con úna única entrada que ofrece el nivel de la señal que tenga a la entrada. Estos vúmetros individuales se agrupan en un módulo vúmetro de 7 entradas, que conectaremos a la salida de cada uno de los filtros del banco de filtros.

Cada uno de los vúmetros individuales generan a la salida una señal de 8 bits, simulando los leds que tradicionalmente se utilizan en estos sistemas. El número de bits que se pongan a 1 en la señal (leds que se iluminan) indicará el nivel de la señal en esa banda.

Para detectar el nivel de la señal, observaremos la palabra a nivel binario. Una palabra de la señal está representada de la siguiente forma:

$$X_{15}X_{14}X_{13}X_{12}X_{11}X_{10}X_9X_8X_7X_6X_5X_4X_3X_2X_1X_0$$

Definiremos cada uno de los 8 niveles de señal dividiendo los 16 bits en conjuntos de dos. Así, el nivel 0 corresponderá a todos los bits a 0, el nivel 1 se alcanzará cuando sólo los bits $X_1 o X_0$ estén a nivel alto, el nivel 2 si los bits $X_4 o X_3$ están a 1, y así sucesivamente. cada uno de estos niveles *encenderá* a la salida el led correspondiente a su nivel además de todos los de los niveles anteriores.

Para tener en cuenta las posibles palabras negativas en complemento a dos, es importante tener en cuenta que debemos observar siempre el valor absoluto de las señales, ya que si no siempre obtendríamos el nivel más alto para cualquier entrada negativa.

4.1. Mantenimiento del nivel

Debido a las posibles variaciones rápidas de la señal, es importante que los picos a la salida del vúmetro se mantengan un mínimo tiempo para su correcta visualización.

Para implementar este comportamiento, se ha definido el siguiente procedimiento:

1. Se toma una muestra a la entrada y se mantiene en memoria
2. Se inicia un contador (configurable mediante genéricos) que indica el tiempo mínimo que se han de mantener los niveles a la salida
3. Si a la entrada llega alguna muestra mayor a la que tenemos en memoria, se sustituye y se reinicia el contador, comenzando el proceso desde el principio.
4. Si el contador llega a cero, se elimina la señal de memoria, permitiendo que otras señales más bajas tomen el lugar de la muestra. En este punto el pico anterior deja de mantenerse y el nivel del vúmetro puede bajar
5. La siguiente muestra entra en memoria y se reinicia el proceso

Esto nos permite mantener de forma sencilla los niveles un mínimo de tiempo en pantalla sin perder información de posibles picos que puedan aparecer mientras estamos manteniendo la señal.

5. Mejoras