

Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Electrónica



Reconocimiento en tiempo real de patrones acústicos de  
motosierras y disparos por medio de una implementación  
en FPGA de Modelos Ocultos de Markov

---

Informe de Proyecto de Graduación para optar por el  
título de Ingeniero en Electrónica con el grado académico  
de Licenciatura

---

Erick Edgardo Salas Chaverri

Cartago, 2010

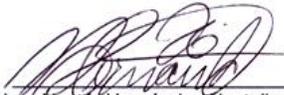
**INSTITUTO TECNOLÓGICO DE COSTA RICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**  
**PROYECTO DE GRADUACIÓN**  
**TRIBUNAL EVALUADOR**

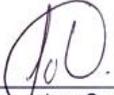
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado de Licenciatura, del Instituto Tecnológico de Costa Rica.

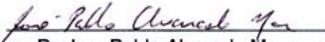
Estudiante: Erick Edgardo Salas Chaverri

Nombre del proyecto:  
Reconocimiento en tiempo real de patrones acústicos de motosierras y disparos por medio de una implementación en FPGA de Modelos Ocultos de Markov

Miembros del Tribunal

  
Ing. Nestor Hernández Hostaller, M.Sc  
Profesor lector

  
Ing. Johan Carvajal Godinez  
Profesor lector

  
Dr.-Ing. Pablo Alvarado Moya  
Profesor asesor

Los miembros de este tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 11 de mayo, 2010

Declaro que el presente proyecto de graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad por el contenido de este anteproyecto de graduación.

Cartago, 11 de mayo de 2010

A handwritten signature in black ink that reads "Erick Salas". The signature is written in a cursive style with a horizontal line underlining the first part of the name.

Erick Edgardo Salas Chaverri.

Céd: 1-1155-0650

## Resumen

Los bosques permiten el mantenimiento de las fuentes de agua para el consumo humano y proporcionan recursos ecológicos. Sin embargo, el entorno natural está siendo dañado por las actividades humanas. Por esta razón se presentan distorsiones en los fenómenos climáticos y desaparecen de especies animales y vegetales.

La detección de la tala y caza puede darse indirectamente por medio de reconocimiento de patrones acústicos. Se pueden capturar señales acústicas mediante una red de sensores, aún bajo la presencia de obstáculos.

Este informe presenta un prototipo de reconocimiento de patrones acústicos basado en Modelos Ocultos de Markov (HMM). Los sonidos analizados corresponden a disparos para el caso de detección de la caza y motosierras para identificar la actividad de tala. Dicho prototipo debe cumplir con los requisitos energéticos para su integración en un nodo en una red de sensores.

La propuesta consiste en una implementación en hardware reconfigurable (FPGA) que incluye etapa de adaptación, adquisición, normalización además de la descomposición de la señal acústica de multi-espectro mediante filtrado digital de respuesta infinita al impulso, extracción de las características y evaluador de los HMM.

**Palabras claves:** HMM, reconocimiento de patrones, FPGA, filtros digitales IIR

## **Abstract**

Forests are necessary for the protection of water sources for human consumption and provide ecological resources. However, the natural environment is being damaged by human activities. For this reason alterations occur in the climate and animal and plant species cease to exist.

The detection of illegal logging and hunting may occur indirectly through recognition of acoustic patterns. Sound signals can be captured through a network of sensors, even in the presence of obstacles.

This report presents a prototype for acoustic pattern recognition based on Hidden Markov Models (HMM). The sounds analyzed correspond to gun shots for the illegal hunting detections and chainsaws for identifications of logging activities. The prototype must meet the energy requirements for integration into a node of a Wireless Sensor Network (WSN)

This work proposes an implementation for the prototype in reconfigurable hardware (FPGA) that includes stage of adaptation, acquisition, and normalization, in addition to the decomposition of the acoustic signal by filtering multi-spectrum digital infinite impulse response, feature extraction and HMM evaluator.

**Key Words:** HMM, pattern recognition, FPGA, IIR Digital Filters.

**A mi padre y madre**

## **Agradecimiento**

La elaboración del presente informe es producto de un gran esfuerzo de investigación y desarrollo pero principalmente fruto de la inspiración divina.

Les agradezco profundamente a todas las personas que hayan mostrado un pelicular aprecio por la presente obra y las implementaciones que le involucra. Al Dr.-Ing Pablo Alvarado Moya por su incalculable guía y ayuda durante todos los procesos del proyecto, al Profesor Ing. Nestor Hernández por su respaldo e iniciativa en los proyectos que coordina. A los demás profesores del Instituto Tecnológico de Costa Rica por su entrega docente que permitió mi formación profesional. A mi familia, parientes, compañeros y mis amigos todos siempre leales, por su incondicional apoyo, confianza y respaldo durante la cristalización de la meta. En especial, a cada ser, que en el futuro lea estas líneas, deberá saber que forma parte de mi.

Para todos, de corazón,

*Gracias.*

# Índice General

|  |           |
|--|-----------|
| <b>Capítulo 1: Introducción .....</b>  | <b>1</b>  |
| 1.1 Detección de actividades de tala y caza en zonas protegidas .....              | 1         |
| 1.2 Trabajos previos.....  | 2         |
| 1.3 Objetivo y estructura del documento.....                                       | 4         |
| <b>Capítulo 2: Marco Teórico .....</b>   | <b>5</b>  |
| 2.1 Adquisición y preproceso de la señal acústica.....                             | 5         |
| 2.1.1 Adquisición de la señal analógica .....                                      | 5         |
| 2.1.2 Normalización de señales acústicas .....                                     | 5         |
| 2.2 Extracción de características de señales acústicas .....                       | 7         |
| 2.2.1 Representación de señales acústicas por su contenido de<br>frecuencias.....  | 7         |
| 2.2.2 Banco de filtros digitales, codificación subbanda.....                       | 7         |
| 2.2.3 Estructura de los filtros IIR .....  | 9         |
| 2.2.4 Implementación en punto fijo .....   | 13        |
| 2.2.5 Codificación CSD.....  | 15        |
| 2.3 Clasificación y los Modelos Ocultos de Markov .....                            | 16        |
| 2.3.1 Evaluación de los modelos .....  | 17        |
| 2.3.2 Antecedentes bibliográficos del reconocimiento de patrones .....             | 19        |
| 2.4 Consumo energético en tecnología FPGA.....                                     | 19        |
| <b>Capítulo 3: Normalización de las señales acústicas .....</b>                    | <b>21</b> |
| 3.1.1 Diseño de la etapa de promediador de señales.....                            | 21        |
| 3.2 Diseño de normalizador .....   | 23        |
| 3.2.1 Implementación del divisor .....   | 25        |
| 3.2.2 Implementación del saturador.....  | 26        |
| 3.3 Resultado de la implementación del normalizador.....                           | 27        |
| 3.3.1 Resultados de la etapa de promediador .....                                  | 27        |
| 3.4 Resultados de la etapa de normalizador.....                                    | 29        |
| <b>Capítulo 4: Etapa de extracción de características de las señales acústicas</b> | <b>32</b> |
| 4.1 Descripción de los filtros digitales .....                                     | 32        |
| 4.1.1 Diseño del banco de filtros digitales.....                                   | 33        |
| 4.2 Aspectos de implementación del filtro digital .....                            | 37        |

|   |  |           |
|---|--|-----------|
| 4.2.1   | Caracterización de las respuestas de los filtros digitales .....                               | 41        |
| 4.2.2   | Detalles de implementación del filtro digital elemental.....                                   | 42        |
| 4.2.3   | Diseño del banco de filtros digitales.....   | 44        |
| 4.3   | Resultados y análisis de la implementación del banco de filtros.....                           | 47        |
| 4.3.1   | Resultados y evaluación de la etapa de banco de filtros .....                                  | 49        |
| 4.3.2   | Consumo de energía entre las tendencias de banco de filtros .....                              | 51        |
| 4.4   | Estimadores de energía por subbanda .....  | 53        |
| 4.5   | Resultados de la implementación de la etapa de extracción de<br>características.....           | 54        |
| 4.5.1   | Funcionamiento de la etapa de extracción de características .....                              | 55        |
| <b>Capítulo 5: Etapa de clasificación .....</b>         |  | <b>57</b> |
| 5.1   | Arquitectura del MAP que implementa el método de Forward basado en<br>HMM                      | 58        |
| 5.1.1   | Descripción de la memoria de acceso y almacenamiento de datos.                                 | 60        |
| 5.1.2   | Descripción de la unidad aritmética de punto flotante.....                                     | 61        |
| 5.2   | Detalles de implementación del MAP y evaluación de los modelos.....                            | 62        |
| 5.2.1   | Descripción del firmware del MAP e implementación del algoritmo<br>hacia adelante.....         | 63        |
| 5.2.2   | Resultados de la etapa de clasificación .....  | 67        |
| 5.2.3   | Uso de recursos en el FPGA en la evaluación de los HMM .....                                   | 67        |
| <b>Capítulo 6: Conclusiones y recomendaciones .....</b> |  | <b>69</b> |
| 6.1   | Conclusiones .....   | 69        |
| 6.2   | Recomendaciones .....  | 71        |
| <b>Capítulo 7: Bibliografía .....</b>                   |  | <b>72</b> |
| <b>Apéndice .....</b>                                   |  | <b>77</b> |
| A. 1  | Diseño de la etapa de filtro analógico .....   | 77        |
| B. 1  | Plataforma utilizada.....  | 79        |
| C. 1  | Diseño del controlador SPI y comunicación entre subsistemas.....                               | 80        |
| D. 1  | Descripción de la etapa de adquisición.....  | 84        |
| E. 1  | Análisis de desborde para distintas estructuras de implementación de<br>filtros digitales..... | 87        |
| F. 1  | Procedimiento hacia adelante en código de alto nivel .....                                     | 88        |
| Glosario .....  |  | 89        |

## Índice de Figuras

|  |    |
|--|----|
| Figura 1-1: Esquema general del prototipo [5]. .....   | 3  |
| Figura 2-1: Bloques internos del AGC [8]. .....  | 6  |
| Figura 2-2: Banco de filtros de $n$ -ésimo nivel para análisis. ....   | 8  |
| Figura 2-3: Estructura de un filtro en forma directa I. ....   | 10 |
| Figura 2-4: Estructura de un filtro en forma directa II. ....  | 11 |
| Figura 2-5: Estructura de un filtro en cascada en secciones de segundo orden. ...  | 12 |
| Figura 2-6: Topología de un HMM de 3 estados. ....   | 17 |
| Figura 2-7: Patrón de cálculo de la variable de adelanto según el estado [26, 25].<br>.....  | 18 |
| Figura 3-1: Algoritmo de estimación del promediador. ....  | 21 |
| Figura 3-2: Esquema de las etapas funcionales del promediador. El acumulador<br>está representado como Acc. ....                                   | 22 |
| Figura 3-3: Comportamiento del umbral para el acumulador (Acc) según el<br>esquema de la figura 3-2. ....  | 23 |
| Figura 3-4: Estructura funcional del normalizador. ....  | 23 |
| Figura 3-5: Esquema las etapas internas del normalizador. ....   | 24 |
| Figura 3-6: Relación en escala logarítmica entre la secuencias $x_{porn}$ y $x_{dn}$ que<br>define el bloque <i>Divisor</i> de la figura 3-5. .... | 25 |
| Figura 3-7: Diagrama funcional del saturador basado en la expresión (3-5). ....  | 26 |
| Figura 3-8: En a) la señal de prueba y en b) la correspondiente respuesta del<br>promediador. ....   | 28 |
| Figura 3-9: En a) la señal de audio y en b) la correspondiente respuesta del<br>promediador. ....  | 28 |
| Figura 3-10: Proceso del promediador y normalizador de la señal de entrada pulso<br>senoidal. ....   | 29 |
| Figura 3-11: En a), un pulso senoidal como una señal de prueba y en b) la<br>correspondiente señal normalizada. ....                               | 31 |
| Figura 3-12: En a), un disparo como una señal de prueba y en b) la<br>correspondiente señal normalizada. ....                                      | 31 |

|  |    |
|--|----|
| Figura 4-1: Distribución de los ceros y polos para el filtro digital pasa alto y b) filtro pasa bajo. ....   | 35 |
| Figura 4-2: Estructura de implementación del bloque elemental del filtro digital factorizado según las etapas del filtro.....  | 37 |
| Figura 4-3: Multiplicador CSD para el coeficiente G_L de la tabla 4-6. ....  | 39 |
| Figura 4-4: Respuesta al impulso de 10000 a) del filtro pasa alto y b) respuesta del filtro pasa bajo. ....  | 41 |
| Figura 4-5: Esquema general del banco de filtros segmentado. ....  | 44 |
| Figura 4-6: Diagrama detallado de los bloques internos del banco de filtros digitales segmentado.....  | 45 |
| Figura 4-7: Adaptación en forma de pila de los elementos de retardo ( $z^{-1}$ ) del banco de filtros digital para el bloque IN mostrado en la figura 4-6. ....  | 46 |
| Figura 4-8: Temporización para cada subbanda según la secuencia determinada por la figura 4-9.....   | 46 |
| Figura 4-9: Secuencia de prioridad para la ejecución de cada subbanda correspondiente al banco de filtros. ....  | 47 |
| Figura 4-10: Respuesta por subbanda del banco de filtros ante una prueba de barrido de frecuencia de 10 Hz a 20 kHz en 5 s. En a) se ilustra la respuesta para la subbanda 0. En b) se ilustra la respuesta para la subbanda 1. Sucesivamente hasta h) para la subbanda 7..... | 50 |
| Figura 4-11: Algoritmo del promediador tipo seguidor.....  | 53 |
| Figura 4-12: Estructura para el estimador de energía para cada subbanda basado en el algoritmo de la figura 4-11. ....   | 54 |
| Figura 4-13: Estimación del máximo de energía por subbanda ante a) una señal de prueba disparo. En b) se ilustra la respuesta para la subbanda 0. En b) se ilustra la respuesta para la subbanda 1. Sucesivamente hasta i) para la subbanda 7.....                             | 55 |
| Figura 5-1: Diagrama de pines del MAP. ....  | 57 |
| Figura 5-2: Diagrama de bloques general del MAP.....   | 59 |
| Figura 5-3: Diagrama de la estructura interna del banco de memoria de datos. ....  | 60 |
| Figura 5-4: Esquema interno de la unidad de aritmética en punto flotante (FPU). ....   | 61 |
| Figura 5-5: Flujo de datos para el cálculo del paso inicialización del algoritmo de adelanto para un HMM. ....   | 63 |

|   |    |
|---|----|
| Figura 5-6: Flujo de datos para el cálculo del paso inducción del algoritmo de adelanto para un HMM. ....   | 64 |
| Figura 5-7: Flujo de datos para la selección de la máxima probabilidad entre los modelos de disparo, motosierra y bosque normal para determinar el estado del bosque. ....  | 64 |
| Figura 5-8: Diagrama de estados para la ejecución del algoritmo hacia adelante. ....  | 65 |
| Figura A-1: Filtro analógico y acople de señal del transductor ( $V_{in}$ ) y la correspondiente señal condicionada ( $V_{acople}$ ). $V_{ref}$ corresponde a $V_{cc}/2$ . ...  | 77 |
| Figura A-2: Respuesta en frecuencia del filtro analógico antialiasing mostrado en la figura A-1 .....   | 79 |
| Figura C-3: Interconexión entre SPARTAN 3E y circuitos integrados de adquisición de señales que componen en SPARTAN 3E [48]. ....   | 81 |
| Figura C-4: Interconexión entre controladores de SPI para los distintos circuitos integrados de adquisición y el prototipo de reconocimiento. ....  | 82 |
| Figura C-5: Formas de onda de funcionamiento de los controladores de SPI y los circuitos integrados de adquisición para la figura C-4. Las especificaciones temporales y funcionales de los circuitos integrados mostrados en la figura C-3 se encuentran en [49,50,51]. .... | 83 |
| Figura D-6: Rango de tensiones permitidas por el bloque de adquisición [52]. ....   | 86 |

## Índice de Tablas

|   |    |
|---|----|
| Tabla 3-1: Detalles de implementación entre los distintos bloques del normalizador según la figura 3-5. ....  | 27 |
| Tabla 4-1: Característica en frecuencia por subbanda, según [6]. ....   | 32 |
| Tabla 4-2: Coeficientes para el filtro propuesto y presentado en [6]. ....  | 33 |
| Tabla 4-3: Resultado de prueba de desborde correspondiente al filtro pasa bajo para distintas estructuras de implementación. ....   | 33 |
| Tabla 4-4: Resultado de prueba de desborde para el filtro pasa alto para distintas estructuras de implementación. ....  | 34 |
| Tabla 4-5: Coeficientes para el filtro digital según el tipo de sección. ....   | 36 |
| Tabla 4-6: Codificación CSD de los coeficientes del filtro digital según la tabla 4-5. ....   | 38 |
| Tabla 4-7: Resumen de implementación en FPGA para el código 4.2-1. ....   | 40 |
| Tabla 4-8: Resumen de implementación en FPGA para el código 4.2-2. ....   | 41 |
| Tabla 4-9: Resumen de los ceros y polos del filtro pasa alto. ....  | 42 |
| Tabla 4-10: Resumen de los ceros y polos del filtro pasa bajo. ....   | 42 |
| Tabla 4-11: Resumen de la utilización de recursos disponibles para la implementación de los multiplicadores CSD en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). ....                           | 43 |
| Tabla 4-12: Resumen comparativo de implementación de los filtros digitales paso bajo y paso alto respecto a sus correspondientes ponderaciones de las subestructuras de multiplicadores CSD. .... | 44 |
| Tabla 4-13: Detalles de implementación entre las estructuras de filtros digitales. ....   | 44 |
| Tabla 4-14: Resumen de la utilización de recursos disponibles para la implementación del banco de filtros segmentado basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). ....               | 47 |
| Tabla 4-15: Detalles de implementación entre distintas tendencias del banco de filtros digital. ....  | 48 |
| Tabla 4-16: Resumen de factores de optimización respecto al área de implementación para distintas tendencias del banco de filtros digital según la tabla 4-15. ....                               | 48 |

|  |    |
|--|----|
| Tabla 4-17: Ancho de banda para cada filtro digital según la figura 4-10 según la relación (4-3).....  | 49 |
| Tabla 4-18: Consumo de potencia entre las distintas tendencias de implementación de banco de filtros digital a la frecuencia de muestreo de 44,1 kHz.....  | 51 |
| Tabla 4-19: Consumo de potencia dinámica en el banco de filtros segmentado para distintas frecuencias de muestreo. ....  | 52 |
| Tabla 4-20: Resumen de la utilización de recursos disponibles para la implementación de la etapa de extracción de características basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). .... | 54 |
| Tabla 5-1: Descripción de los pines del MAP según la figura 5-1. ....  | 58 |
| Tabla 5-2: Resumen de la utilización de recursos disponibles para la implementación del banco de memoria del MAP basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). ....                  | 61 |
| Tabla 5-3: Resumen de la utilización de recursos disponibles para la implementación de la unidad de suma del <i>FPU</i> basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). ....           | 62 |
| Tabla 5-4: Resumen de la utilización de recursos disponibles para la implementación del MAP basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1). ....                                       | 62 |
| Tabla 5-5: Resumen de las funciones de cada estado según la figura 5-8. ....   | 66 |
| Tabla 5-6: Comparación entre implementaciones basadas en software y en hardware para el algoritmo forward según HMM del reconocimiento de patrones para algunas cadenas de observación. ....     | 67 |
| Tabla 5-7: Resumen del número y clase de instrucciones para la evaluación de los HMM. ....   | 68 |
| Tabla 5-8: Resumen de la frecuencia de operación del MAP.....  | 68 |
| Tabla A-1: Lista de resistencias que componen el filtro analógico de la figura A-1. ....   | 78 |
| Tabla A-2: Lista de capacitores que componen el filtro analógico de la figura A-1  | 78 |
| Tabla C-3: Número de ciclos mínimo para el funcionamiento de los sistemas de la figura C-4. ....   | 82 |
| Tabla C-4: Frecuencia de operación del sistema de la figura 1-1 para la frecuencia de muestreo especificada en la tabla D-6 según el modo de operación.....                                      | 83 |

|  |    |
|--|----|
| Tabla C-5: Resumen de la utilización de recursos para el sistema de adquisición y los bloques que le componen de la figura C-4 según la plataforma de implementación mostrada en apéndice B. 1. .... | 84 |
| Tabla D-6: Frecuencia de muestreo del prototipo. ....  | 85 |

## **Capítulo 1: Introducción**

Los bosques permiten el mantenimiento de las fuentes de agua para el consumo humano y proporcionan recursos ecológicos. Sin embargo, el entorno natural está siendo dañado por las actividades humanas. Por esta razón se presentan distorsiones en los fenómenos climáticos y la desaparición de especies animales y vegetales.

El comercio ilegal de maderas exóticas hace de la tala ilícita una actividad lucrativa, que en conjunto con la escasez de guardaparques y la gran extensión de territorio por resguardar propicia el deterioro de las zonas protegidas. Por otro lado el auge de la cacería furtiva debido a la creciente explotación de los productos derivados de los animales, despierta el interés por este comercio ilegítimo que ocasiona la amenaza de la fauna involucrada. Afortunadamente, la creciente visión conservacionista señala que las zonas protegidas consisten en únicos reservorios de recursos hídricos y que podían representar enormes ganancias para el estado por concepto de servicios ambientales [1].

### **1.1 Detección de actividades de tala y caza en zonas protegidas**

La detección en tiempo real de la tala y caza que ocurren en las zonas protegidas del país permitiría una disminución de esos eventos que ocasionan daños al medio ambiente. Por esa razón, aportar una herramienta a las instituciones que velan por la conservación de los bosques para que facilite la monitorización en las densas extensiones de territorio resguardado permitiría ayudar para tal fin.

La detección de la tala y caza puede darse indirectamente por medio del reconocimiento de patrones acústicos. Se pueden capturar señales acústicas mediante redes de sensores como la propuesta en [2], aún bajo la presencia de obstáculos.

## 1.2 Trabajos previos

La presente tesis complementa una secuencia de trabajos previos sobre implementaciones en software del reconocimiento de patrones acústicos, realizados en el contexto del proyecto descrito en [2].

El primer sistema [3,4] es una solución analógica para la detección de eventos acústicos (sonidos de motosierras y disparos). Consiste principalmente en redes neuronales artificiales monocapa entrenadas a través de discriminadores Fisher, un acondicionador de señal y un banco de filtros. El porcentaje de error fue del 12% y 10% respecto al total de los eventos acústicos probados relativos a la motosierras y disparos respectivamente para un radio de cobertura de 30 m que representa 2862 m<sup>2</sup>. El consumo de potencia promedio de dicho sistema fue de 1,8 W.

El segundo sistema consiste en un prototipo de software que utiliza para la extracción de las características un mecanismo basado en *wavelet* y para la clasificación una implementación de Modelos Ocultos de Markov (*HMM*) [5]. Utilizando los paquetes de MatLab para el diseño de wavelets y *HMM* se realizó una implementación fuera de línea.

En [6] se realizó un prototipo algorítmico capaz de reconocer los patrones acústicos correspondientes a los sonidos de disparos, motosierras y los propios del ambiente del bosque. Las muestras de sonidos utilizadas en los trabajos anteriores corresponden a disparos con distintas armas de fuego y motosierras. Dichas muestras fueron adquiridas en el Parque Nacional Braulio Carrillo a distancias de 30m, 90m, 250m y 600m, con orientaciones de 0°, 90° y 180° respecto al punto de adquisición. Las armas empleadas son carabina 22, 2 revolver 32 y 38, una pistola 9mm y una escopeta 12 [5]. Los prototipos anteriores se implementaron en software. El trabajo actual continúa la estrategia general del proyecto con un prototipo en hardware ilustrado en la figura 1-1. Este prototipo cuenta con subsistemas para el acople del transductor hacia el sistema digital que incluye un filtro analógico descrito en la figura A-1 diseñado para evitar el *aliasing*,

el normalizador que varía la amplitud de la señal de audio a un nivel de referencia, el banco de filtros que separa la señal normalizada para su posterior extracción de la energía por cada banda de frecuencia. Posteriormente el codificador (no tratado en este documento), convierte los niveles de energía en códigos o símbolos para que sea interpretado posteriormente en la etapa de clasificación, mediante la evaluación de *HMM* y finalmente se decide sobre el estado del bosque.

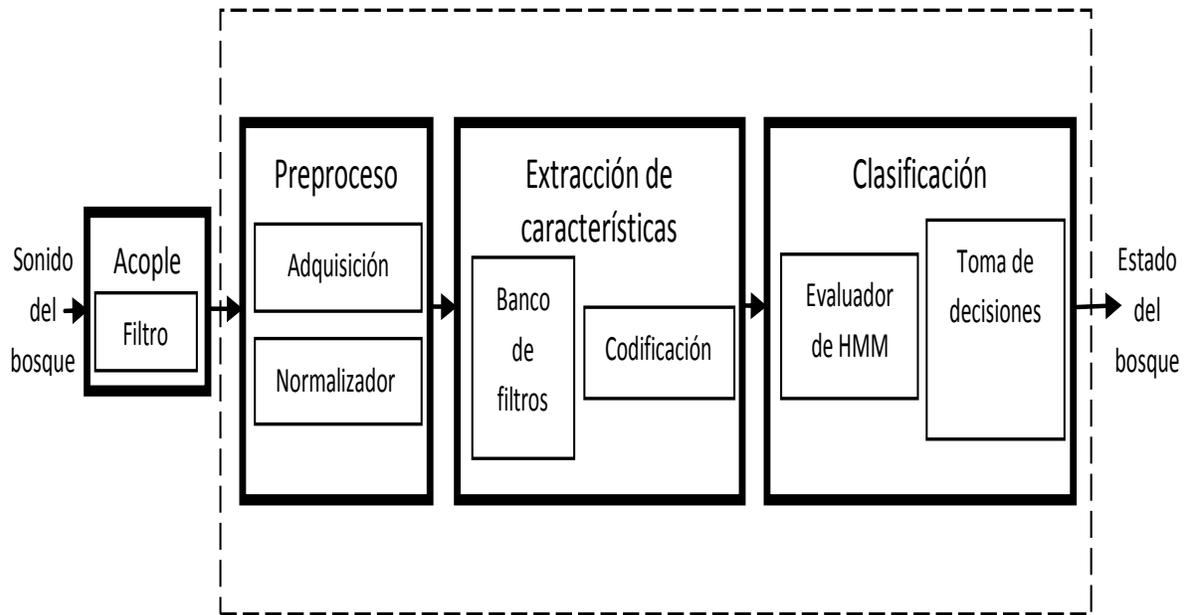


Figura 1-1: Esquema general del prototipo [5].

### **1.3 Objetivo y estructura del documento**

El objetivo de este proyecto es diseñar una arquitectura de un circuito electrónico basado en FPGA integrable a un nodo sensorial que permita analizar digitalmente las señales acústicas en tiempo real para la detección de sonidos característicos de disparos y motosierras.

Dicha arquitectura está implementada en FPGA que incluye los mecanismos de adquisición y preproceso que corresponden al tratamiento previo de las señales de audio. Además incluye el circuito digital que permite la extracción de las características para la clasificación de las señales y finalmente el diseño una estructura electrónica basada en HMM para poder discernir sobre el tipo de evento acústico, según sea característico de motosierra, disparo o del un bosque típico.

Para facilitar la lectura del trabajo, se decidió separar el contenido de un capítulo de resultados y análisis y presentarlo acoplado luego la presentación de cada sistema. Así en el capítulo 2 se presentan los aspectos teóricos de la adquisición y normalización de señales, filtrado digital y los principios matemáticos de los Modelos Ocultos de Markov. En el capítulo 3 se explica la descripción, el diseño, el resumen de implementación y los resultados de las pruebas del normalizador del prototipo. En el capítulo 4 se muestra el diseño, se describen las características de implementación y las respuestas de las distintas pruebas de desempeño del banco de filtro digital. En el capítulo 5 se presenta el algoritmo de adelanto, el diseño, los resultados de implementación y el resumen de las pruebas de ejecución de la estructura que evalúa los HMM.

Finalmente en el capítulo 6 se establecen las conclusiones obtenidas y las recomendaciones para la optimización del prototipo en términos de consumo de potencia.

## Capítulo 2: Marco Teórico

El presente capítulo muestra las bases y fundamentos físicos y matemáticos del proyecto. Se muestran las etapas de adquisición y control de ganancia automática de la señal acústica. Además se incluyen los aspectos matemáticos del filtrado digital y los Modelos Ocultos de Markov.

### 2.1 Adquisición y preproceso de la señal acústica

#### 2.1.1 Adquisición de la señal analógica

Para la adquisición de la señal analógica debe tomarse en cuenta el teorema de muestreo de Nyquist que especifica: dada una señal en banda limitada de  $B$  Hz es representable en forma exacta por muestras equidistantes con periodos de muestreo no mayores que  $1/(2B)$  [7].

Las señales con frecuencias superiores al muestreo ( $f_s$ ) producen solapamiento de espectros y en consecuencia se recuperan frecuencias distintas a las presentes en la señal analógica.

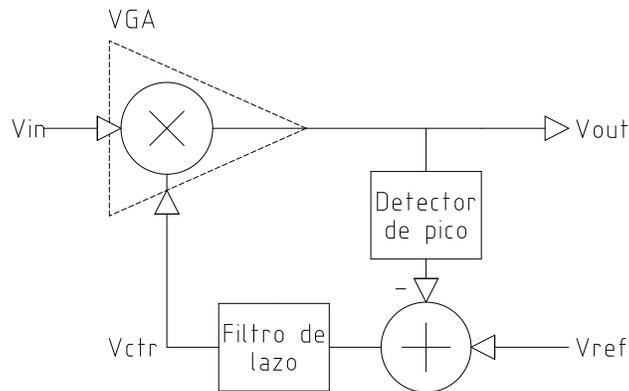
Para prevenir el solapamiento se debe acoplar un filtro paso bajo. Este filtro debe preceder al conversor analógico digital o *ADC*. El filtro debe suprimir las frecuencias de solapamiento tal que la frecuencia de corte ( $f_c$ ) esté bajo el criterio,

$$f_c \leq \frac{f_s}{2} \quad (2-1)$$

#### 2.1.2 Normalización de señales acústicas

La normalización en general es un procedimiento que ajusta las amplitudes de las señales a un nivel constante. Normalmente la normalización tiene como referencia un nivel ajustable, que depende de la técnica de normalización. El normalizador está basado en el control de ganancia automática AGC (Automatic Gain Control). Un AGC típico consiste de los bloques mostrados en la figura 2-1, donde VGA (Variable Gain Amplifier) es el amplificador de ganancia variable, Vin

representa la señal de entrada al AGC,  $V_{ctr}$  corresponde a la señal de control del VGA,  $V_{ref}$  es la tensión de referencia para el funcionamiento del filtro de lazo, un detector de picos y  $V_{out}$  es la salida del AGC.



**Figura 2-1: Bloques internos del AGC [8].**

El método de normalización por valor máximo o pico ajusta todos los niveles máximos relativos hacia aproximadamente el valor pico máximo absoluto. Dicho procedimiento limita la dinámica de la señal.

La entrada de la señal es amplificada por el VGA, luego el detector de picos extrae el valor máximo de la salida, finalmente el filtro de lazo genera la magnitud dc con referencia a  $V_{ref}$  para variar la ganancia del VGA [8].

La normalización debe ajustarse con el pico de la señal en lugar de su energía promedio en el tiempo, para forzar la extracción de características notorias de los eventos acústicos. Este método normaliza el sonido según los picos de señal de corta duración.

## 2.2 Extracción de características de señales acústicas

### 2.2.1 Representación de señales acústicas por su contenido de frecuencias

Las señales de audio se caracterizan por concentrar la energía en las frecuencias bajas [9,10], por esa razón se prefiere la codificación subbanda para extraer las características de las señales acústicas. Dicha codificación divide el espectro en octavas enfatizando las subbandas correspondientes a las frecuencias bajas, duplicando repetidamente el ancho de banda hasta bisecar el espectro de audio.

Los anchos de banda están relacionados con la duración de los pulsos propios de la señal acústica bajo el principio de incertidumbre descrito como

$$\Delta\omega \cdot \Delta t \sim 1 \quad (2-2)$$

que expresa la proporcionalidad inversa de la incertidumbre de la frecuencia ( $\Delta\omega$ ) respecto al intervalo de tiempo  $\Delta t$  [10]. Dicho principio argumenta que una señal concentrada en un intervalo reducido de tiempo le corresponde un amplio intervalo de frecuencias [11], razón que justifica la duplicación de las subbandas para los distintos anchos de pulso por analizar.

### 2.2.2 Banco de filtros digitales, codificación subbanda

El banco de filtros digitales en codificación subbanda divide el espectro de la señal en la mitad por cada subsección, al tiempo que la frecuencia de muestreo se reduce por un factor de diezmo, en este caso factor de 2 [12]. En esta configuración, mostrada en figura 2-2 se conoce como banco de filtros de análisis diádico asimétrico de  $n$  – niveles o (*n-Level Asymmetric Dyadic Analysis Filter Bank*) [13], sólo la subbanda de baja frecuencia se descompone por sección por medio del bloque LPF (Low Pass Filter, filtro pasa bajo) y del bloque HPF (High Pass Filter, filtro pasa alto). Nótese que debido a que todos los bloques LPF- $n$  y HPF- $n$  son idénticos y en conjunto con el submuestreo permite la reutilización sistematizada de estructuras [14].

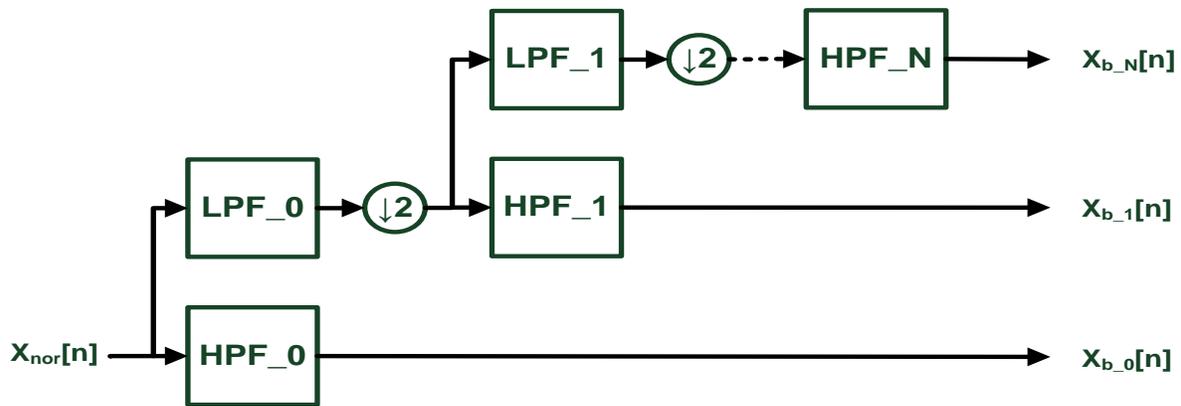


Figura 2-2: Banco de filtros de  $n$ -ésimo nivel para análisis.

El funcionamiento adecuado del banco de filtros digitales en subbanda exige que el *aliasing* resultante sea insignificante para que el diezmado y los filtros *QMF* (*quadrature mirror filter*) tengan una respuesta en frecuencia complementaria adecuada [6].

### Filtro IIR

El filtro digital elegido anteriormente en [6] para el banco de filtros a utilizar en el proyecto es IIR tipo elíptico o de Cauer. Los filtros IIR (Respuesta Infinita al Impulso) son filtros recursivos en el cual cada muestra de salida es la suma ponderada de la entrada (en algunos casos las entradas anteriores) y las salidas anteriores [15].

Los filtros IIR tienen las siguientes ventajas respecto a los filtros tipo FIR [16]:

- Requiere menor cantidad de coeficientes.
- Necesita menor cantidad de elementos de memoria.
- Presenta menor latencia.
- Respuesta es más cercana al modelo analógico.

### 2.2.3 Estructura de los filtros IIR

El filtro IIR se define en el dominio de la transformada z como

$$H(z) = \frac{\sum_{i=0}^M b_i z_i^{-i}}{\sum_{i=0}^N a_i z_i^{-i}} \quad (2-3)$$

Consiste en una sección tipo FIR hacia adelante, también conocido como filtro sólo ceros en referencia a los coeficientes  $b_i$  como los coeficientes de los ceros y una sección FIR de retroalimentación con los coeficientes  $a_i$  como los coeficientes de los polos [16]. Los parámetros  $M$  y  $N$  representan el número de coeficientes del numerador y denominador respectivamente.

En el dominio del tiempo, el filtro IIR puede expresarse como

$$y(n) = \sum_{i=0}^M b_i x(n-i) + \sum_{i=0}^N a_i y(n-i) \quad (2-4)$$

Los filtros IIR pueden implementarse utilizando distintas estructuras como la forma directa I, forma directa II, esquema bicuadrático (combinación de forma directa I y II, bicuadrático en cascada) [16,15].

#### Forma directa

La forma directa se representa directamente como en (2-4), mediante un cociente de dos polinomios que resultan en implementaciones de deducción directa [17]. Existen dos tipos de formas directas (I y II). El primer tipo de formas o estructura mostrada en la figura 2-3 necesita  $M+N+1$  multiplicaciones,  $m+n$  sumas y  $m+n+1$  posiciones de memoria [15]. La estructura directa I cuya ecuación de diferencias está dada por

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \dots + b_M x(n-M) - a_1 y(n-1) - a_2 y(n-2) - \dots - a_N y(n-N) \quad (2-5)$$

tiene mayor número de elementos de retraso, por lo que su uso no es recomendado para implementaciones prácticas [18].

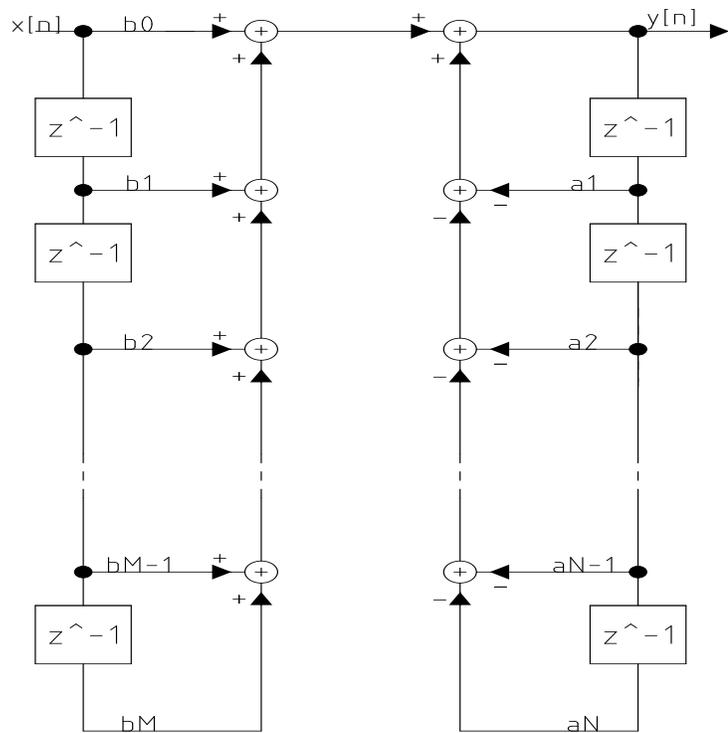


Figura 2-3: Estructura de un filtro en forma directa I.

El segundo tipo de estructura de filtro digital mostrada en la figura 2-4 se conoce como estructura directa II. Este tipo de realización o estructura necesita  $M+N+1$  multiplicaciones,  $m+n$  sumas y el máximo de  $m, n$  posiciones de memoria [15]. La ecuación de diferencias correspondiente está dada por

$$\begin{aligned}
 w(n) &= -a_1w(n-1) - a_2w(n-2) + \dots + a_nx(n-N) + x(n) \\
 y(n) &= b_0w(n) + b_1w(n-1) + \dots + b_Mx(n-M)
 \end{aligned}
 \tag{2-6}$$

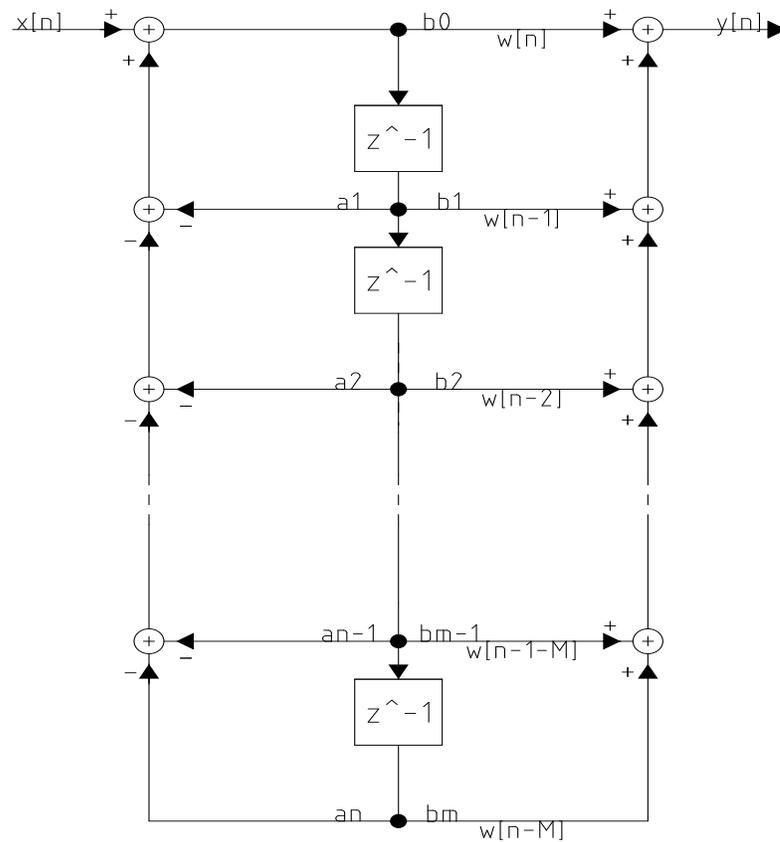


Figura 2-4: Estructura de un filtro en forma directa II.

Las anteriores estructuras son sensibles a la cuantificación de coeficientes [17].

### Estructura en cascada

Un desarrollo de la función de transferencia (2-4) con coeficientes  $a_i$  y  $b_i$  reales, permite expresar de manera equivalente la función de transferencia del filtro digital como productos de polinomios de segundo grado (y de primer grado, en el caso de filtros con un grado impar) [19].

$$H(z) = \prod_{k=1}^K H_k(z) \quad (2-7)$$

donde

$$H_k = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}, k = 1, 2, \dots, K \quad (2-8)$$

Los sistemas en cascada organizados en bloques de segundo orden como se muestra en la figura 2-5 admiten obtener modularidad de la implementación del filtro digital [15].

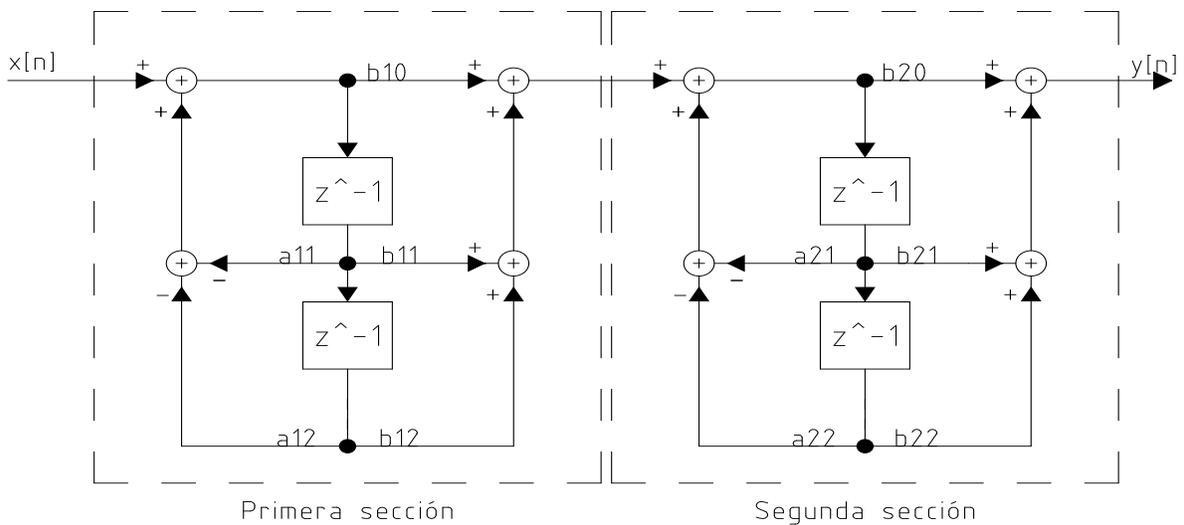


Figura 2-5: Estructura de un filtro en cascada en secciones de segundo orden.

En estructuras en cascada cada emparejamiento de ceros y polos de  $H(z)$  produce una configuración particular [17], pero siempre deben agruparse las parejas de raíces conjugadas para obtener coeficientes reales [15].

#### **2.2.4 Implementación en punto fijo**

La escogencia de una estructura puede influir en la complejidad del hardware [17] y al mismo tiempo puede influir en el comportamiento del filtro digital con un número limitado de bits destinado para la representación de los coeficientes y los elementos de memoria.

De la estructura de implementación puede depender la sensibilidad o desvío respecto al diseño. Las operaciones aritméticas que se realizan con una longitud de la palabra fija y limitada pueden alcanzar los rangos establecidos aún con truncamiento o redondeo [17].

Las implementaciones de filtros digitales basadas en operaciones en punto fijo requieren una adecuada caracterización en cuantificación de coeficientes, cuantificación interna, desborde y estabilidad para asegurar el cumplimiento de las especificaciones de diseño [16]. A continuación se describen las distintas caracterizaciones de implementación en punto fijo.

##### **Cuantificación de coeficientes y estabilidad**

La respuesta en frecuencia de un filtro digital puede ser afectada por la cuantificación de coeficientes. Dicho efecto puede provocar que el filtro sea inestable dependiendo de la posición respecto al círculo unitario [16].

Según [15], una relación de la perturbación del polo y la sensibilidad inherente ante las variaciones de los coeficientes derivados de la cuantificación es expresada por

$$\Delta p_i = - \sum_{k=1}^N \frac{p_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^N (p_i - p_l)} \Delta a_k \quad (2-9)$$

donde  $p_i$  es el polo  $i$ -ésimo,  $N$  es el orden del denominador de la función de transferencia del filtro y  $\Delta a_k$  representa el error de cuantificación del  $k$ -ésimo coeficiente del denominador de (2-8).

En los sistemas de segundo orden, la agrupación apropiada de los pares de polos conjugados pueden minimizar la perturbación o el desvío  $\Delta p_i$  típico en los filtros digitales [15] debido que la perturbación de los polos  $\Delta p_i$  es inversamente proporcional al incremento de la distancia entre los polos  $(p_i - p_l)$ .

Si existen ceros de la función de transferencia ubicados sobre el círculo unitario como en el caso de los filtros IIR elípticos [15], la sensibilidad a la cuantificación de los coeficientes puede controlarse directamente mediante la asignación de la longitud de la palabra.

La estructura directa es sensible a los efectos de la cuantificación. Un cambio en los coeficientes del filtro debido a la cuantificación probablemente produzca un cambio en la posición de los polos [15], por lo que se prefieren implementaciones en cascada de filtros de segundo orden por tener mayores prestaciones implementaciones en punto fijo [15].

### **Cuantificación interna y desborde**

Las implementaciones en punto fijo tienen el rango de representación limitado respecto a las arquitecturas basadas en punto flotante. Además las operaciones aritméticas realizadas pueden alcanzar el máximo de representación provocando desborde. Ambas situaciones ocasionan una distorsión indeseable en la señal [16].

Las estrategias más comunes para eliminar el desborde son la saturación y el escalamiento. El escalamiento consiste en modificar la longitud de la palabra

respecto al ancho de bus de datos disponible. El escalamiento es la técnica más adecuada para minimizar el desborde debido que no se agregan módulos comparadores. Existen dos tipos de escalamiento: por incremento o decremento según se amplía el ancho del bus respecto la longitud de la palabra de referencia o se reduce la longitud de la palabra para un ancho del bus constante. El escalamiento por incremento tiene implicaciones en el área de implementación pero el escalamiento por decremento tiene implicaciones en la relación señal a ruido [16]. La utilización de la técnica de saturación implica el acople de módulos adicionales y ocasiona una distorsión por recorte de la señal.

### 2.2.5 Codificación CSD

En [20] describe la codificación *CSD* (canonical-signed-digit), como la representación de dígitos binarios con un número mínimo de elementos distintos de cero. El sistema *CSD* está basado en el *SD* que es muy útil para implementar con reducida complejidad multiplicadores o sumadores sin acarreo.

La representación emplea una secuencia de símbolos (0, + o -) según sea suma o resta en una posición que denota una potencia de 2. Para el caso de 3.1416, el equivalente *CSD* retorna +0-.00+00+000+0000- que se expande a

$$3.1416 = 2^2 - 2^0 + 2^{-3} + 2^{-6} + 2^{-10} - 2^{-15} \approx 3.14157104492$$

El desvío relativo en la codificación anterior es del 0.0009212%

En [20] se planea la utilización de *CSD* para la implementación de filtros digitales, en [22] se propone algoritmos y estructuras para la utilización de *CSD* y en [23] se describe el análisis y la utilización de *CSD* para representación de fracciones.

Para construir un multiplicador (multiplicación por una constante) basado en *CSD*, se usan las sumas y restas de argumentos (el argumento como la entrada del multiplicador) desplazados tantos bits (hacia la derecha o izquierda) como la codificación *CSD* lo indique, descrito como una combinación de desplazamientos del argumento.

## 2.3 Clasificación y los Modelos Ocultos de Markov

En [6] se justifica la elección de HMM como clasificadores para el proyecto. Es necesaria la síntesis en un circuito digital que implemente los Modelos Ocultos de Markov (*HMM*) para que analice las muestras correspondientes a las señales acústicas que son de naturaleza aleatoria.

Un *HMM* está integrado por [24, 25,26]:

1. Un conjunto de estados denotado como  $S = \{S_1, S_2, \dots, S_N\}$ ,  $N$  representa el número de estados.
2. El conjunto de símbolos del alfabeto. Los símbolos son parte de la cadena de observación del sistema modelado. Se define como  $V = \{V_1, V_2, \dots, V_M\}$ ,  $M$  representa el tamaño del alfabeto discreto y cada  $V_k$  hace referencia a un símbolo distinto.
3. La distribución de probabilidad de transición entre estados definida como  $A = \{a_{ij}\}$  y  $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ , es decir,  $a_{ij}$  es la probabilidad de estar en estado  $j$  en el instante  $t$  si en el instante anterior  $t - 1$  estaba en el estado  $i$ . Los índices  $i$  y  $j$  están definidos como  $1 \leq i, j \leq N$ .
4. La distribución de probabilidad por símbolo observado en el estado  $j$ . Definida como  $B = \{B_j(k)\}$  donde el elemento  $B_j$  está definido como  $B_j = P(V_k \text{ para } t | q_t = S_j)$ , bajo los índices  $1 \leq j \leq N, 1 \leq k \leq M$ .
5. La distribución de estado inicial  $\pi = P(q_1 = S_i)$ ,  $1 \leq i \leq N$ . donde  $\pi_i$  es la probabilidad de que el primer estado sea el estado  $S_i$ .

Una característica de los *HMM* radica en la dependencia de transición, en la cual la probabilidad de cambiar de estado depende únicamente del estado presente. Lo anterior se conoce como propiedad de Markov o suposición de transición de primer orden de Markov [26,27].

En la figura 2-6 se muestra la topología de un HMM de 3 estados junto con los elementos  $a_{ij}$  correspondientes a cada transición de estados.

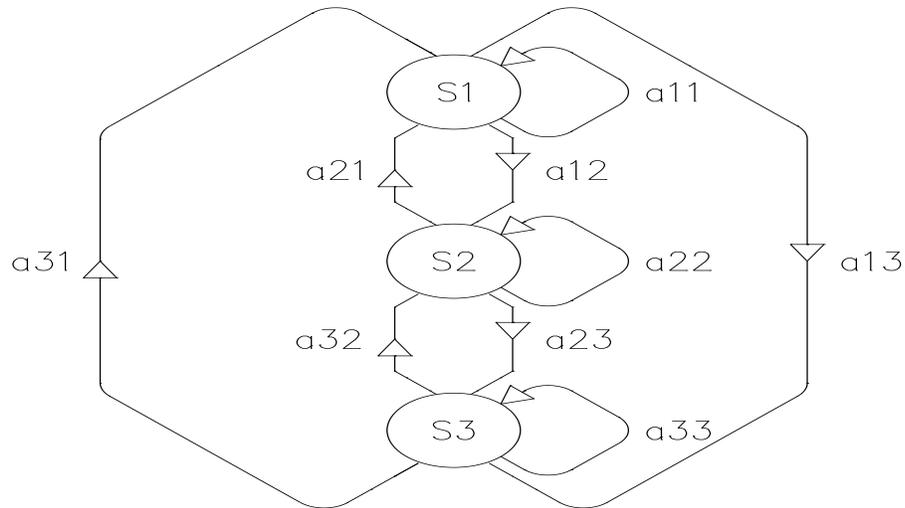


Figura 2-6: Topología de un HMM de 3 estados.

### 2.3.1 Evaluación de los modelos

El algoritmo hacia adelante es un procedimiento eficiente para la evaluación de los HMM que calcula la probabilidad de la secuencia en el tiempo  $t$  según ocurre cada observación, mediante la variable de adelanto  $\alpha_t(i)$ . Según la siguiente definición:

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \quad (2-10)$$

donde existe  $\alpha_t(i)$  por cada modelo de HMM denotado como  $\lambda = (A, B, \pi)$ .

El procedimiento del cálculo de  $\alpha_t(i)$  se efectúa en tres partes [6,26]:

#### 1. Inicialización

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2-11)$$

#### 2. Inducción

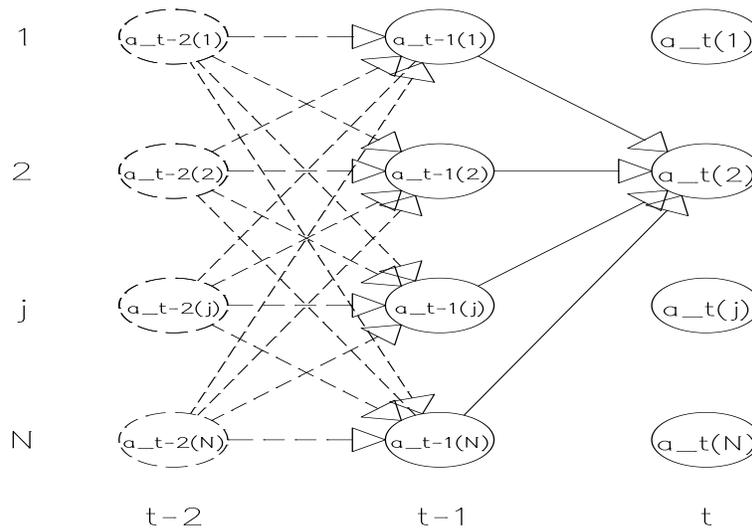
$$\alpha_t(j) = \left[ \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t), \quad \begin{matrix} 1 \leq t \leq T - 1 \\ 1 \leq j \leq N \end{matrix} \quad (2-12)$$

En la figura 2-7 se muestra la secuencia del cálculo de la variable de adelanto por estado del HMM que se acumulan en el tiempo.

### 3. Terminación

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2-13)$$

En el paso final o terminación, la probabilidad para cada modelo es la suma de las correspondientes variables de adelanto en el tiempo terminal o  $T$ .



**Figura 2-7: Patrón de cálculo de la variable de adelanto según el estado [26, 25].**

Finalmente, las probabilidades de los tres modelos para una secuencia de observación dada, se deben comparar para reconocer el estado del bosque que represente más probablemente el evento acústico.

### 2.3.2 Antecedentes bibliográficos del reconocimiento de patrones

Los métodos de reconocimiento de patrones son usados en la solución de problemas de clasificación y detección. En [28, 29] se describe una arquitectura capaz de reconocer patrones basado en HMM para una conversación en tiempo real implementado en un sistema FPGA.

En artículos relacionados con la implementación de HMM en FPGA predomina la implementación del algoritmo de Viterbi, utilizado en la solución del problema de encontrar cuál es la secuencia de estados ocultos más probable. En el contexto de este proyecto sólo se requiere el cálculo de la probabilidad, que se realiza con el algoritmo hacia adelante, que al requerir menos pasos consume a su vez menos potencia [30, 31, 32].

## 2.4 Consumo energético en tecnología FPGA

La disipación de potencia en circuitos CMOS que componen el arreglo estructural del FPGA se representa mediante la siguiente ecuación [33, 34].

$$P_{total} = p_t(C_L V_{dd}^2 f_{clk}) + I_{sc} V_{dd} + I_{fuga} V_{dd} \quad (2-14)$$

La primera componente representa la potencia por conmutación, la cual incluye la carga y descarga del capacitor parásito  $C_L$ , la frecuencia de operación  $f_{clk}$ , la tensión  $V_{dd}$  y  $p_t$  como el factor de actividad que establece la probabilidad que ocurra consumo de potencia por transición. El segundo término denota el consumo de  $I_{sc}$  que es la corriente que fluye cuando los transistores PMOS y NMOS están mutuamente encendidos y el tercer término contiene la corriente de fuga  $I_{fuga}$ . Los dos primeros términos contribuyen a la disipación de potencia *dinámica* y el tercer a la disipación de potencia *estática* [35].

La estrategia para la reducción de la corriente puede enfocarse en los parámetros que son generalmente manipulables, como la capacitancia y la frecuencia. La capacitancia es relativa al número de compuertas que conmutan en el tiempo y la longitud de los conductos de la señal [34].

El consumo por corriente *estática* depende de los componentes en términos de área (número de registros, componentes aritméticos y la conectividad mediante los multiplexores) [36].

La disipación de potencia por corriente *dinámica* se puede estimar basada en la relación [20],

$$P_{din} \approx I_{Din} \cdot V_{cc} = K \cdot f_{max} \cdot N \cdot \tau_{LE} \cdot V_{cc} \quad (2-15)$$

donde  $K$  es una constante, la frecuencia del sistema  $f_{max}$  en MHz, el promedio de celdas que conmutan para cada ciclo de reloj y  $N$  denota el número de celdas en la implementación.

El consumo energético por concepto de la corriente *dinámica* ocurre solamente en proporción a la transición de la señal de reloj. Existen diversas maneras que reducción de consumo de energía por corriente dinámica, como la reducción de ciclos ociosos, decremento de la frecuencia de trabajo o la deshabilitación de circuitos digitales en desuso [37].

## Capítulo 3: Normalización de las señales acústicas

En el presente capítulo se presentan los bloques internos que corresponden a la etapa de normalización de las señales acústicas. Se introduce con la descripción del promediador, luego explica el diseño del normalizador y finalmente los bloques divisor y saturador. Finalmente se muestran los detalles de implementación y resultados de funcionamiento.

### 3.1.1 Diseño de la etapa de promediador de señales.

El promediador es una estructura encargada de calcular el valor medio de la señal. Como se muestra en la figura 3-1 los parámetros principales del promediador son  $\Delta\_carga$  y  $\Delta\_descarga$ .

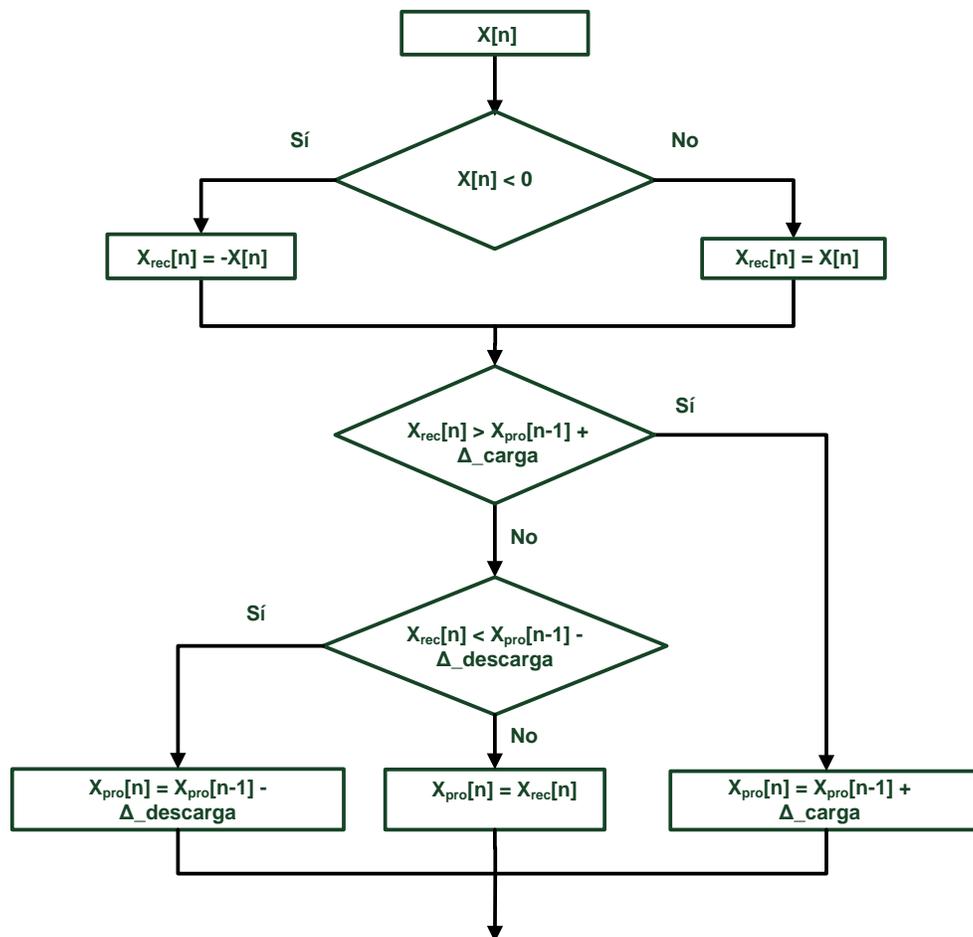


Figura 3-1: Algoritmo de estimación del promediador.

La estructura del promediador se muestra en figura 3-2. La señal  $x[n]$  es rectificadora debido a que interesa la magnitud de la señal, resultando en  $x_{rec}[n]$ . Luego, dicha señal es evaluada mediante la función

$$X_{pro}[n] = \begin{cases} X_{pro}[n-1] + \Delta_{carga} & \text{para } X_{rec}[n] > X_{pro}[n-1] + \Delta_{carga} \\ X_{pro}[n-1] - \Delta_{descarga} & \text{para } X_{rec}[n] < X_{pro}[n-1] - \Delta_{descarga} \\ X_{rec}[n] & \text{el resto} \end{cases} \quad (3-1)$$

El funcionamiento del promediador está basado en la limitación de la tasa máxima o mínima de cambio.

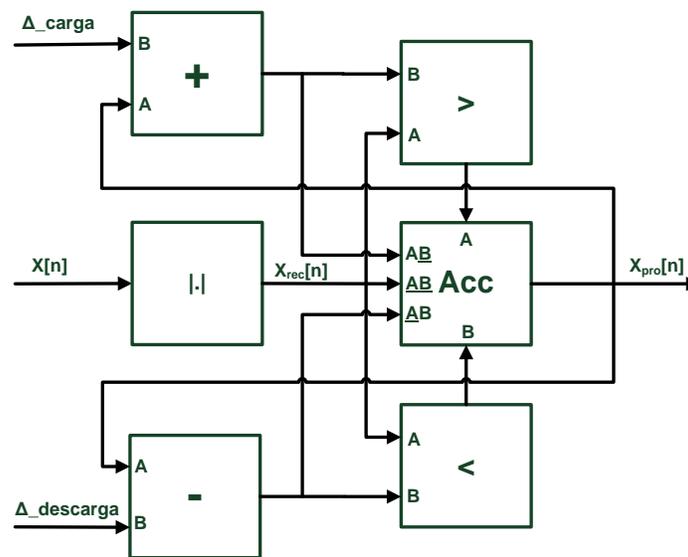


Figura 3-2: Esquema de las etapas funcionales del promediador. El acumulador está representado como Acc.

El crecimiento o decrecimiento del acumulador es limitado cuando el valor de  $X_{rec}[n]$  está fuera del intervalo definido (figura 3-2). En caso contrario  $X_{cap}[n]$  sigue al valor de  $X_{rec}[n]$ .

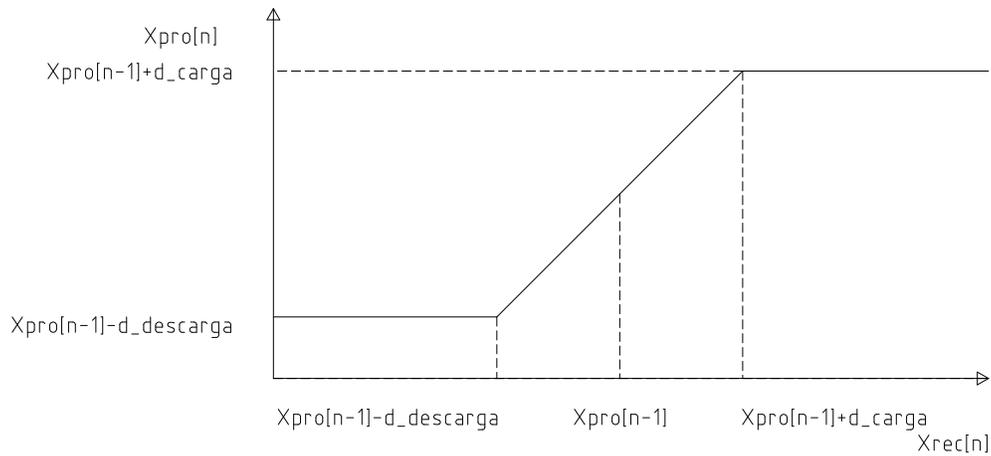


Figura 3-3: Comportamiento del umbral para el acumulador (Acc) según el esquema de la figura 3-2.

### 3.2 Diseño de normalizador

El bloque de preproceso introducido en la figura 1-1 incluye una etapa de adquisición y normalización de la señal del audio. El normalizador se describe mediante

$$x_{nor}[n] = \frac{x[n]}{x_{pro}[n] + k_{nor}} \quad (3-2)$$

El valor de señal normalizada  $x_{nor}[n]$  se define como la relación de la muestra de la señal  $x[n]$  entre el valor promedio  $x_{pro}[n]$  y la constante de normalización  $k_{nor}$  (utilizada para evitar la división entre cero). La estructura funcional se muestra en la figura 3-4.

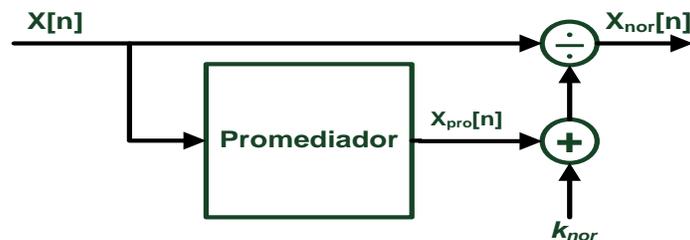


Figura 3-4: Estructura funcional del normalizador.

La expresión anterior requiere punto flotante puesto que la señal rondará alrededor de 1. Para corregir esto debe normalizarse la señal de la siguiente forma:

$$x_{nor}[n] = x[n] \frac{2^{nbits-3}}{x_{pro}[n] + k_{nor}} \quad (3-3)$$

La expresión (3-3) define el proceso de normalización en aritmética entera en términos del número de bits (en el caso del prototipo,  $nbits = 16$ ). La amplitud de señal en condición promedio de la entrada es la norma de  $2^{nbits-orden\_norma} = 8192$ . La norma depende del ancho del bus de datos ( $nbits = 16$ ) y del número de orden de la norma ( $orden\_norma = 3$ ). Dicha condición ocurre en un tiempo definido por la constante de descarga, según el algoritmo mostrado en la figura 3-1.

El promediador tiene un patrón aditivo como se define en (3-1) en función del parámetro de carga o descarga y en conjunto con el sistema de normalización varía la ganancia del AGC (bloque multiplicador de la figura 3-5).

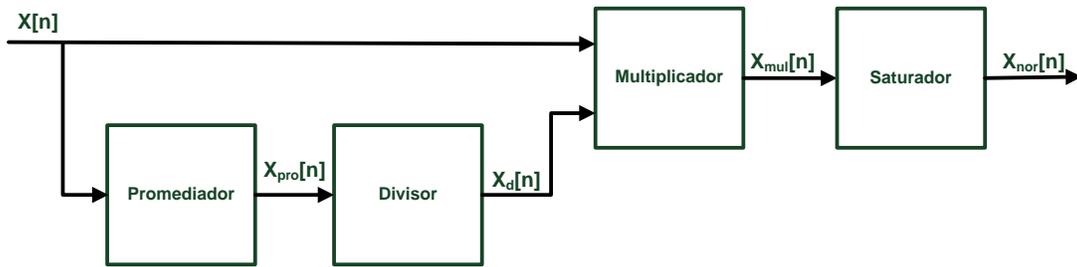


Figura 3-5: Esquema las etapas internas del normalizador.

La caracterización entre los parámetros de carga y descarga tiene distintas funciones en el normalizador. Inicialmente, el AGC tiene una ganancia máxima que amplifica la señal incluyendo el ruido. Dependiendo de las magnitudes de entrada, el promediador obtiene una lectura de la energía promedio de cualquier excitación en una ventana de tiempo de duración finita.

### 3.2.1 Implementación del divisor

Según el funcionamiento del normalizador establecido en la figura 3-5, se requiere una división, como se define en (3-3). Basado en un desarrollo logarítmico de la expresión (3-3) se obtiene

$$\log x_d[n] = \log(2^{nbits-3}) - \log(x_{por}[n] + k_{nor}) \quad (3-4)$$

que describe el comportamiento del bloque *Divisor* del normalizador (figura 3-5) expresado en la gráfica de la figura 3-6.

La implementación del *Divisor* basado en memoria ROM no requiere un circuito digital secuencial o procesamiento de algún algoritmo de división que minimiza el consumo de potencia [20,38,39], cuya penalidad resulta de innecesarios ciclos de reloj que directamente va en detrimento de la eficiencia energética, ni de una estructura compleja combinatorial de múltiples niveles que involucran un excesivo número de transiciones de señales [40].

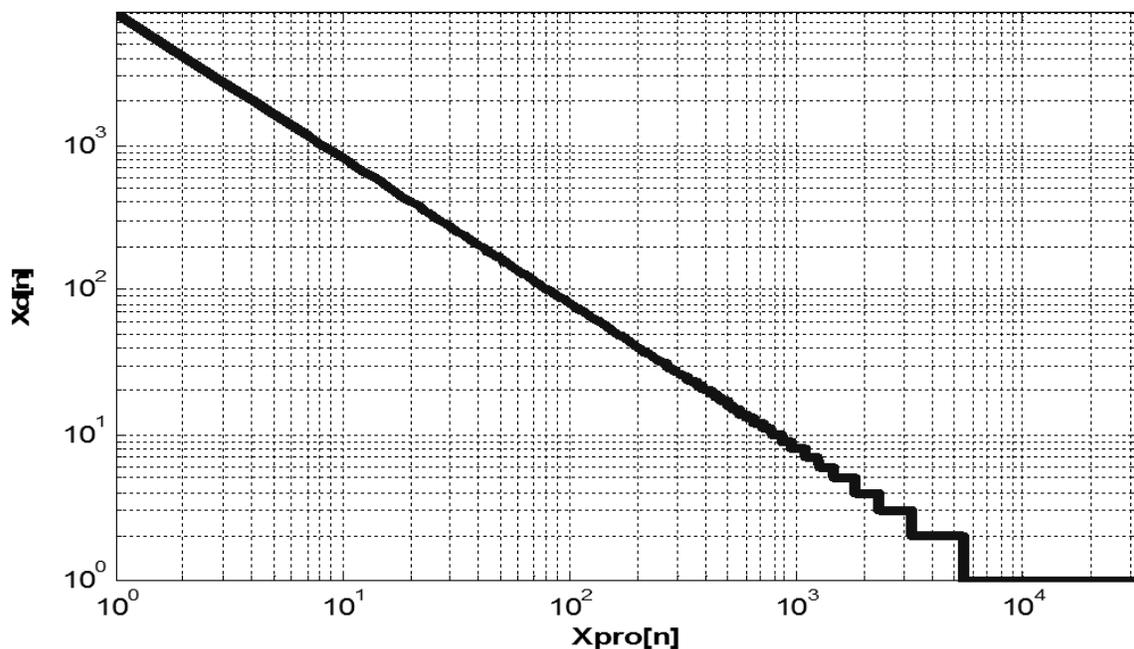


Figura 3-6: Relación en escala logarítmica entre la secuencias  $x_{por}[n]$  y  $x_d[n]$  que define el bloque *Divisor* de la figura 3-5.

### 3.2.2 Implementación del saturador

El saturador limita la amplitud de la señal según

$$X_{nor}[n] = \begin{cases} 2^{n_{bits}-1} - 1 & \text{para } X_{mul}[n] > 2^{n_{bits}-1} - 1 \\ -2^{n_{bits}-1} & \text{para } X_{mul}[n] < -2^{n_{bits}-1} \\ X_{mul}[n] & \text{resto} \end{cases} \quad (3-5)$$

El saturador tiene la estructura funcional mostrada en la figura 3-7 y contiene un selector que elige las señales que se conducen hacia  $X_{nor}[n]$  basado en las condiciones de los comparadores. Con  $n_{bits} = 16$ , los niveles máximos y mínimos corresponden a 32767 y -32768 respectivamente.

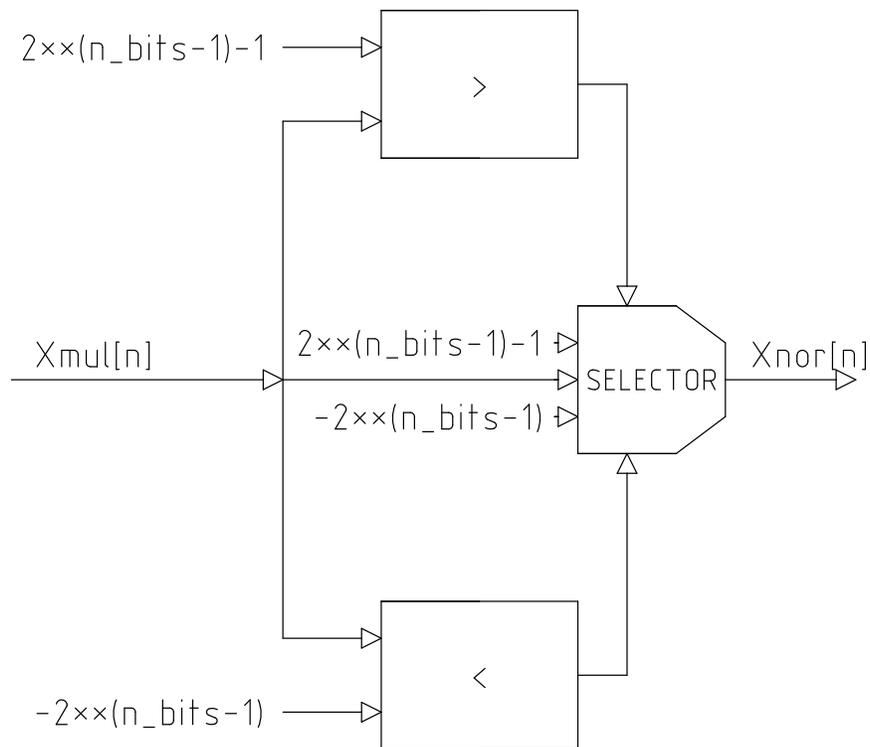


Figura 3-7: Diagrama funcional del saturador basado en la expresión (3-5).

### 3.3 Resultado de la implementación del normalizador

En la tabla 3-1 se resume la utilización de componentes para la implementación del normalizador utilizables para una FPGA Spartan 3E de Xilinx (ver anexo B. 1).

Tabla 3-1: Detalles de implementación entre los distintos bloques del normalizador según la figura 3-5.

| Estructura            | Slice | Flip Flop | LUT | MULTI18x18 |
|-----------------------|-------|-----------|-----|------------|
| Promediador           | 72    | 16        | 134 | 0          |
| LUT(divisor)          | 124   | 0         | 218 | 0          |
| Multiplicador         | 0     | 0         | 0   | 1          |
| Saturador             | 15    | 0         | 26  | 0          |
| Normalizador(Total)   | 212   | 19        | 283 | 1          |
| Utilización total (%) | 4     | 0,204     | 4   | 5          |

El módulo de normalización opera con una frecuencia de muestreo de 43,4 kHz. La frecuencia de operación de los módulos de control de adquisición es 860 kHz utilizando la FPGA Spartan 3E de Xilinx que funciona con un oscilador de 50 MHz (ver anexo B. 1).

En la siguiente sección se describe la respuesta del promediador ante señales en pleno funcionamiento.

#### 3.3.1 Resultados de la etapa de promediador

El promediador se prueba mediante una señal pulso como la señal de prueba mostrada en la figura 3-8.a. Dicha señal tiene una duración aproximadamente de 25 ms con una amplitud con valor máximo de 0,4 V y produce un resultado en el promediador como se muestra en la figura 3-8.b, cuya respuesta corresponde a una señal con crecimiento y decrecimiento lineales. El prototipo presentado en esta tesis se proponen los valores de  $\Delta_{carga}$  y  $\Delta_{descarga}$  como 16 y 1 respectivamente. Además el promediador mantiene la tendencia de incremento hasta encontrar el máximo relativo de la señal de entrada, luego el patrón de decremento se mantiene hasta cambiar de condición según (3-1).

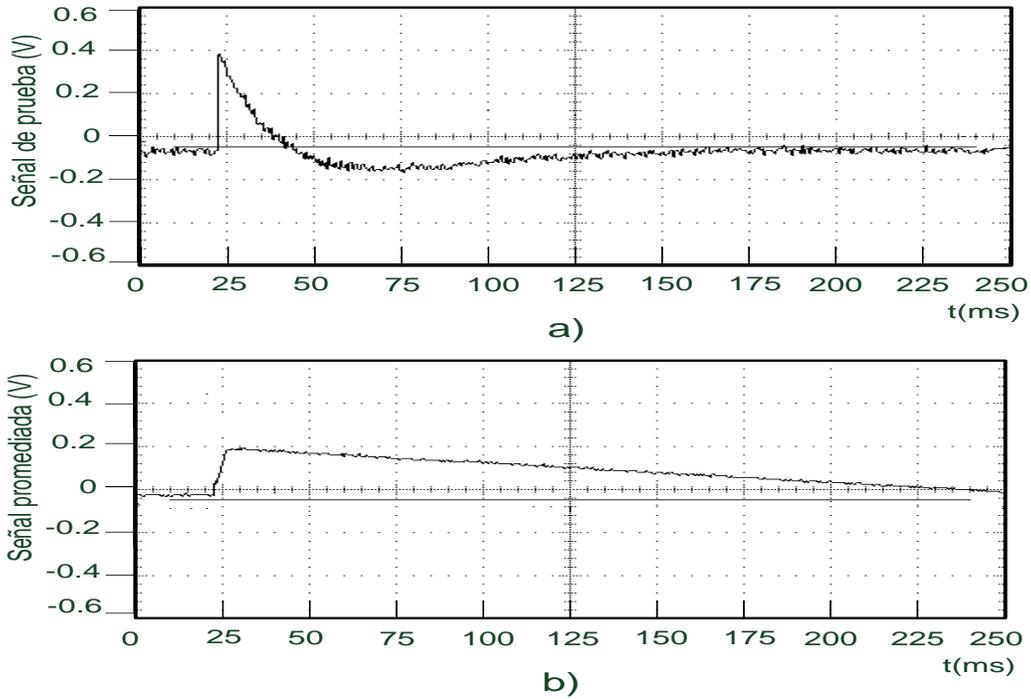


Figura 3-8: En a) la señal de prueba y en b) la correspondiente respuesta del promediador.

En la figura 3-9.b, se muestra la respuesta del promediador ante una excitación de una señal de audio representada en la figura 3-9.a. Dicha respuesta denota la cantidad de energía promedio de la señal de entrada.

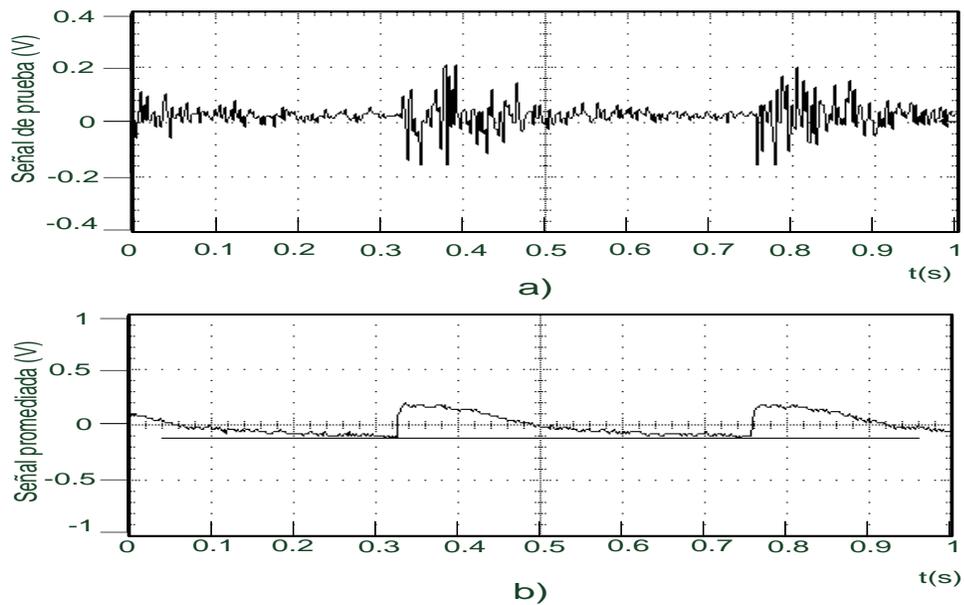


Figura 3-9: En a) la señal de audio y en b) la correspondiente respuesta del promediador.

### 3.4 Resultados de la etapa de normalizador

El algoritmo del normalizador propuesto se implementó en Matlab y se obtuvo la respuesta del normalizador ante una señal de entrada pulso senoidal con una frecuencia de 500 Hz y 7500 amplitud. La duración del pulso es 15 ms. Dicha señal que contiene un ruido, como se muestra en la figura 3-10. La frecuencia de muestreo es de 44,1 kHz.

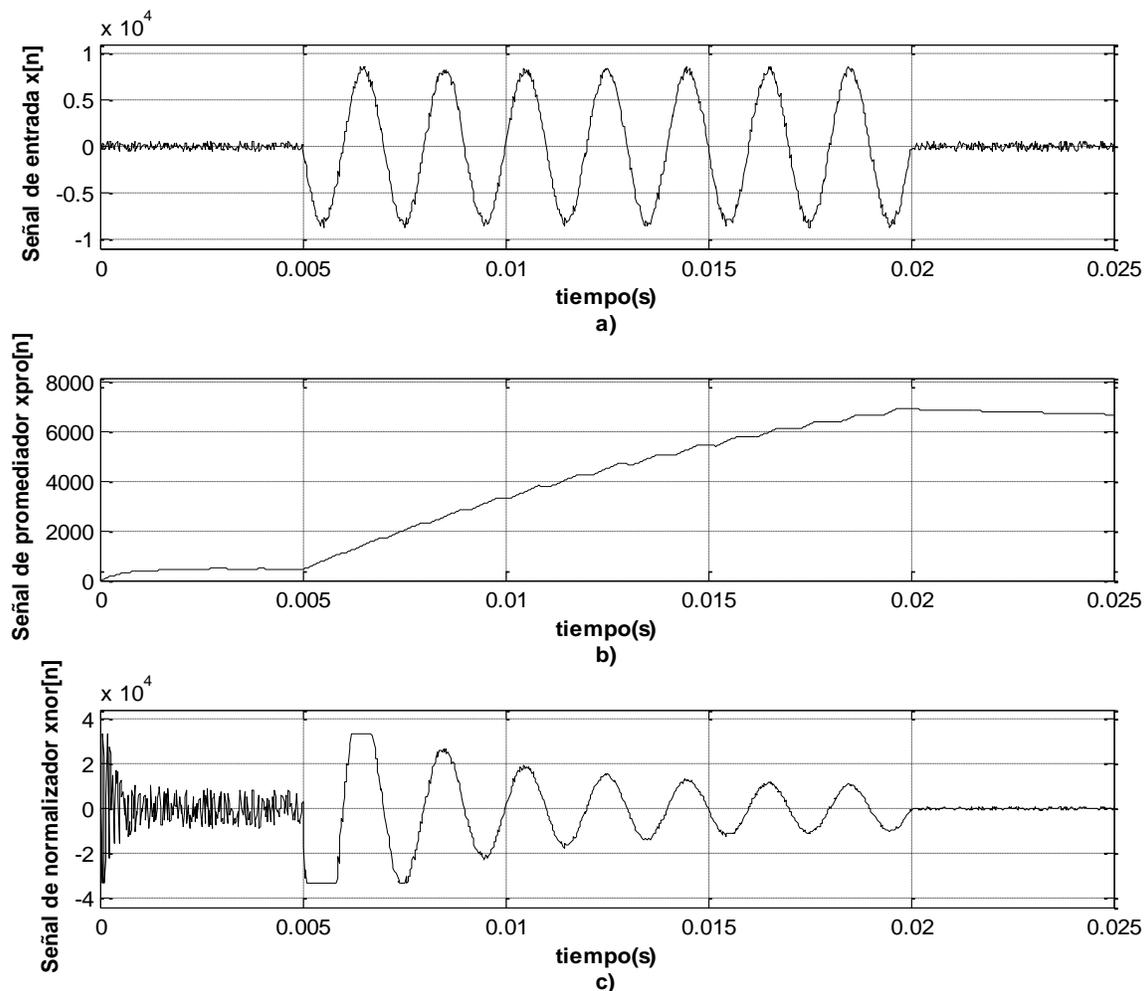


Figura 3-10: Proceso del promediador y normalizador de la señal de entrada pulso senoidal.

En la figura 3-10.b contiene la respuesta del promediador y en la figura 3-10.c la salida del normalizador. El efecto del saturador (figura 3-5) limita el nivel de la señal resultante de la multiplicación según (3-5).

La figura 3-10.c ilustra para este caso particular que en el proceso de normalización ocurren varias fases:

1. Cuando el promediador está descargado o levemente cargado, caso típico de sonidos leves en el ambiente, el normalizador aumenta la ganancia logrando una notable amplificación de la señal incluyendo al ruido.
2. Cuando la señal de entrada contiene un pulso aun con el promediador descargado, el normalizador modifica la ganancia desde nivel alto hacia abajo a una tasa de cambio según el promediador. La salida del normalizador experimenta saturación.
3. La amplificación o atenuación del normalizador depende de si el valor promedio es menor o mayor a la norma, y es independiente si la transición es menor a la norma. Si el promediador tiene un valor alto debido a eventos pasados y la entrada relativamente pequeña, la entrada será atenuada. Por otro lado, si el promediador tiene un valor pequeño, la señal es amplificada, independientemente de sus valores.

Al módulo implementado se alimenta con un pulso senoidal con una frecuencia de 500 Hz.. La duración del pulso es 15 ms con amplitud de 0.4 V y se captura con un osciloscopio como se muestra la figura 3-11. Nótese la amplificación del pulso. Inicialmente el AGC tiene la ganancia máxima que gradualmente disminuye en el tiempo de acuerdo con la función de entrada según (3-5). La forma típica de la respuesta del normalizador ante un pulso senoidal corresponde a una envolvente lineal cuya pendiente está definida por  $\Delta_{descarga}$ . Dicha respuesta es similar a la obtenida en la implementación en software como lo muestra la figura 3-10.c. En el apéndice D. 1 se describen las relaciones entre las tensiones eléctricas y el valor binario equivalente en los procesos de adquisición y la conversión digital analógico.

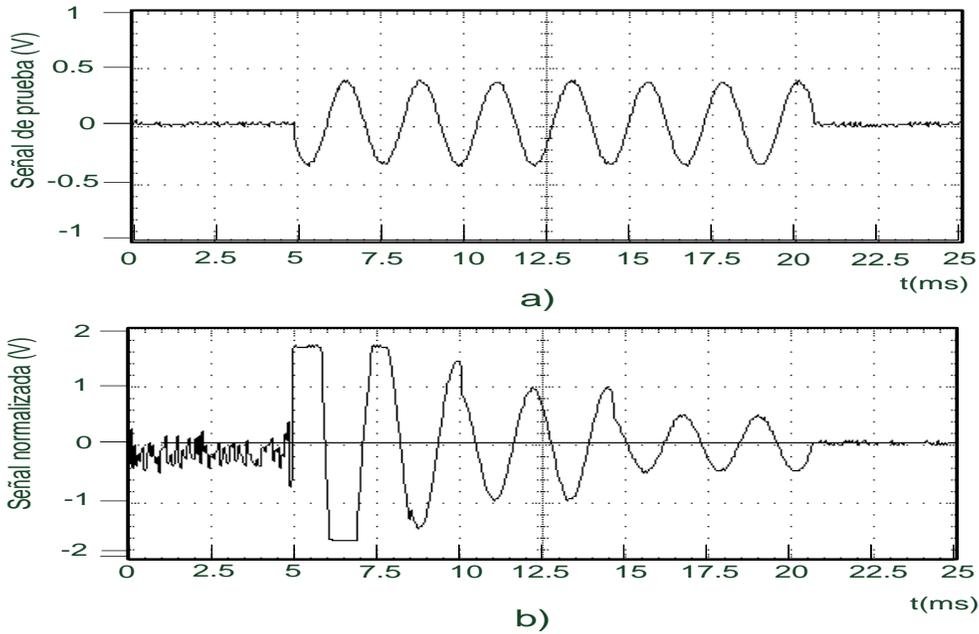


Figura 3-11: En a), un pulso senoidal como una señal de prueba y en b) la correspondiente señal normalizada.

En la figura 3-12 se muestra la normalización para una señal obtenida de un disparo de aproximadamente 42 ms de duración. Nótese el cambio de escala en la figura 3-12 como resultado de la normalización, en el cual existe ganancia en los primeros picos de la señal.

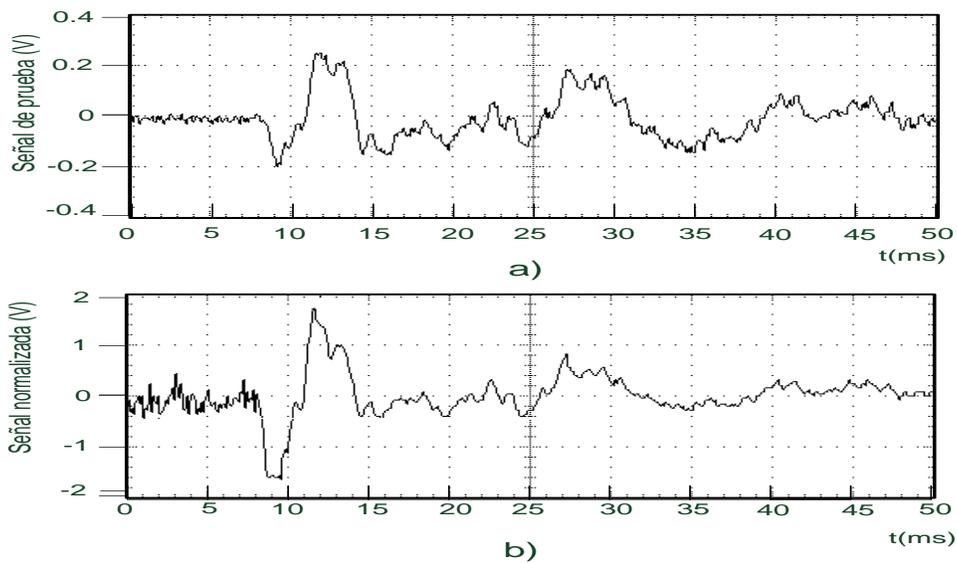


Figura 3-12: En a), un disparo como una señal de prueba y en b) la correspondiente señal normalizada.

## Capítulo 4: Etapa de extracción de características de las señales acústicas

En el presente capítulo se presentan los bloques funcionales correspondientes a la etapa de extracción de características como el banco de filtros digitales y los estimadores de energía por subbanda. Se incluye la descripción, implementación, consumo de energía y desempeño del filtro digital, además de los detalles de diseño del estimador de energía. En final instancia se muestran diversos resultados de funcionamiento ante distintas pruebas.

### 4.1 Descripción de los filtros digitales

Según las consideraciones presentadas en [6], las bandas utilizadas por el banco de filtros apto para esta tesis se presentan en la tabla 4-1. Nótese la reducción gradual del ancho de banda para cada subbanda.

Tabla 4-1: Característica en frecuencia por subbanda, según [6].

| Subbanda | Frecuencia alta (Hz) | Frecuencia baja (Hz) | Ancho de banda (Hz) |
|----------|----------------------|----------------------|---------------------|
| 0        | 20.000,00            | 11.025,00            | 8.975,00            |
| 1        | 11.025,00            | 5.512,50             | 5.512,50            |
| 2        | 5.512,50             | 2.756,25             | 2.756,25            |
| 3        | 2.756,25             | 1.378,13             | 1.378,13            |
| 4        | 1.378,13             | 689,06               | 689,06              |
| 5        | 689,06               | 344,53               | 344,53              |
| 6        | 344,53               | 172,27               | 172,27              |
| 7        | 172,27               | 86,13                | 86,13               |

El banco de filtros consiste en un total de 8 filtros digitales para 8 subbandas de frecuencia, cuyas características son,

- Filtros QMF elíptico o de Cauer de orden 3.
- IIR.
- Rizado de la banda pasante menor a 1 dB.
- Rizado del a banda no pasante menor a 30 dB.
- Banda de transición entre 0,4 y 0,7 de la frecuencia normalizada.

En el dominio Z, el filtro digital de tercer orden puede ser expresando como

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}} \quad (4-1)$$

Los coeficientes para la función de transferencia (4-1) se resumen en la tabla 4-2. Nótese la simetría que existe en los coeficientes del numerador del filtro paso bajo y la antisimetría del numerador del filtro paso alto.

Tabla 4-2: Coeficientes para el filtro propuesto y presentado en [6].

| Número de coeficiente | Denominador $a_i$ | Numerador del filtro paso bajo $b_i$ | Numerador del filtro paso alto $b_i$ |
|-----------------------|-------------------|--------------------------------------|--------------------------------------|
| 0                     | 1                 | 0,211933103                          | 0,210320841                          |
| 1                     | 0,006150841       | 0,50261334                           | -0,497393977                         |
| 2                     | 0,422261262       | 0,50261334                           | 0,497393977                          |
| 3                     | 0,000680784       | 0,211933103                          | -0,210320841                         |

#### 4.1.1 Diseño del banco de filtros digitales

El banco de filtros requerido es la estructura digital que considera las bandas de frecuencia descritas en la tabla 4-1. El banco de filtros tiene como núcleo operativo un filtro digital dual que contempla el filtro pasa alto y pasa bajo. La selección de la estructura adecuada para la implementación del filtro digital elemental está basada en el análisis del desborde para las estructuras en forma directa I, forma directa II, sistema en cascada de secciones de segundo orden organizado en forma directa I y II tal como se describieron en la sección 2.2.3 del marco teórico.

Tabla 4-3: Resultado de prueba de desborde correspondiente al filtro pasa bajo para distintas estructuras de implementación.

|                        | Directa I | Directa II | SOS DI | SOS DII |
|------------------------|-----------|------------|--------|---------|
| <b>Salida</b>          | 9/1000    | 0/1000     | 0/1000 | 0/1000  |
| <b>Prod Num</b>        | 0/4000    | 0/4000     | 0/6000 | 0/6000  |
| <b>Prod Den</b>        | 0/3000    | 0/3000     | 0/4000 | 0/4000  |
| <b>Acc Num</b>         | 0/3000    | 0/3000     | 0/4000 | 0/4000  |
| <b>Acc Den</b>         | 0/3000    | 0/3000     | 0/4000 | 0/4000  |
| <b>Estados</b>         | -         | 127/3000   | -      | 0/4000  |
| <b>Estados Num</b>     | -         | -          | 0/1000 | -       |
| <b>Estados Den</b>     | -         | -          | 9/4000 | -       |
| <b>Sección Entrada</b> | -         | -          | -      | 0/1000  |
| <b>Sección Salida</b>  | -         | -          | -      | 0/1000  |

**Tabla 4-4: Resultado de prueba de desborde para el filtro pasa alto para distintas estructuras de implementación.**

|                        | <b>Directa I</b> | <b>Directa II</b> | <b>SOS DI</b> | <b>SOS DII</b> |
|------------------------|------------------|-------------------|---------------|----------------|
| <b>Salida</b>          | 2/1000           | 0/1000            | 0/1000        | 0/1000         |
| <b>Prod Num</b>        | 0/4000           | 0/4000            | 0/6000        | 0/6000         |
| <b>Prod Den</b>        | 0/3000           | 0/3000            | 0/4000        | 0/4000         |
| <b>Acc Num</b>         | 0/3000           | 0/3000            | 0/4000        | 0/4000         |
| <b>Acc Den</b>         | 0/3000           | 0/3000            | 0/4000        | 0/4000         |
| <b>Estados</b>         | -                | 127/3000          | -             | 0/4000         |
| <b>Estados Num</b>     | -                | -                 | 0/1000        | -              |
| <b>Estados Den</b>     | -                | -                 | 2/4000        | -              |
| <b>Sección Entrada</b> | -                | -                 | -             | 0/1000         |
| <b>Sección Salida</b>  | -                | -                 | -             | 0/1000         |

El resumen del análisis de desbordes según la estructura de implementación se muestra en la tabla 4-3 y la tabla 4-4 correspondiente al filtro paso bajo y paso alto respectivamente. La prueba consiste en ingresar un conjunto de datos aleatorio en un rango definido en la entrada del filtro digital y contabilizar la cantidad de desbordes por cada sección de cada estructura analizada. Las secciones incluyen: la salida final del filtro (salida), los multiplicadores (Prod Num y Prod Den para el numerador y denominador respectivamente), los sumadores (Acc Num y Acc Den para el numerador y denominador respectivamente), los estados y las secciones de entrada y salida. Las implementaciones analizadas son: forma directa I (Directa I), forma directa II (Directa II), forma directa I organizada en sistema de segundo orden (SOS DI) y forma directa II organizada en sistema de segundo orden (SOS DII) . El código para realizar esta prueba se desglosa en el apéndice E. 1.

En la tabla 4-3 se muestra que la implementación del filtro pasa bajo en estructura directa I, directa II o en la forma de segundo orden tipo directa I ocurrieron desbordes para un vector de prueba de 1000 valores aleatorios. En dicha prueba sólo la estructura de segundo orden en cascada en forma directa II estuvo exenta de desbordes.

De manera similar que en la tabla 4-3, en la tabla 4-4 se muestra que sólo la estructura de segundo orden en cascada organizada en forma directa II estuvo

exenta de desbordes. El anterior resultado determina que las sistemas de segundo orden también conocidas como biquadráticas son aptas cuando se usa aritmética de punto fijo en contraste con estructuras implementadas en forma directa I y II [17,15]. Reorganizando el filtro digital de tercer orden como implementación cascada de sistema primer orden (FOS, First order system) y sistema de segundo orden (SOS, Second order system) se obtiene

$$H(z) = G \left( \frac{b_{01} + b_{11}z^{-1}}{a_{01} + a_{11}z^{-1}} \right) \left( \frac{b_{02} + b_{12}z^{-1} + b_{22}z^{-2}}{a_{02} + a_{12}z^{-1} + a_{22}z^{-2}} \right) \quad (4-2)$$

En la figura 4-1 se muestran los planos en el dominio z de polos y ceros para el filtro pasa alto y pasa bajo con los coeficientes dados de la tabla 4-2. Nótese la existencia del par de polos y ceros conjugados complejos para ambos tipos de filtro y su correspondiente simetría. Cada uno de los SOS se organizo agrupando los polos y ceros complejos conjugados y por separado el correspondiente polo y cero de primer orden. Tanto el filtro digital pasa alto y pasa bajo son sistemas estables debido que los polos correspondientes se encuentran dentro del círculo unitario.

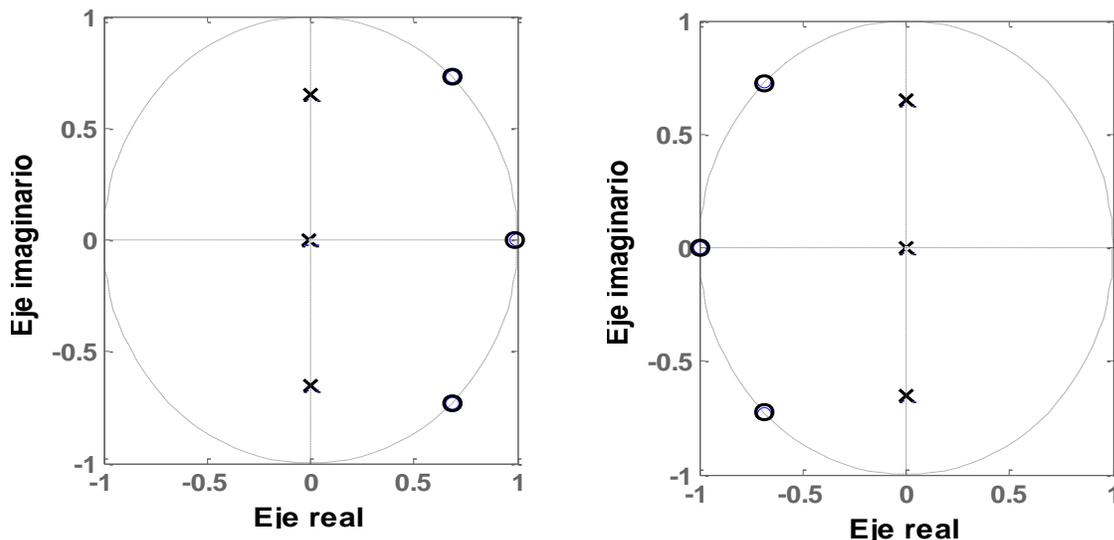


Figura 4-1: Distribución de los ceros y polos para el filtro digital a) filtro pasa alto y b) filtro pasa bajo.

Los coeficientes de (4-2) se resumen en la tabla 4-5. El denominador de ambos tipos de filtros es idéntico y en el caso del numerador, de 5 coeficientes, 3

son iguales entre el filtro paso alto y paso bajo, lo que permite la simplificación de la estructura del filtro dual mediante la utilización de un bloque común.

Tabla 4-5: Coeficientes para el filtro digital según el tipo de sección.

| Sección primer orden   |                   | Sección segundo orden |                   |                  |
|--|-------------------|-----------------------|-------------------|------------------|
| <b>Coeficientes numerador del filtro pasa bajo, <math>G_L = 0,211933103133258</math></b> |                   |                       |                   |                  |
| b01  | b11               | b02                   | b12               | b22              |
| 1  | 1                 | 1                     | 1,37156599344564  | 1                |
| <b>Coeficientes numerador del filtro pasa alto, <math>G_H = 0,210320841107964</math></b> |                   |                       |                   |                  |
| b01  | b11               | b02                   | b12               | b22              |
| 1  | -1                | 1                     | -1,36492957370444 | 1                |
| <b>Coeficientes denominador</b>  |                   |                       |                   |                  |
| a01  | a11               | a02                   | a12               | a22              |
| 1  | 0,001612226202529 | 1                     | 0,00453857917431  | 0,42225394424122 |

El bloque común contempla el flujo de datos y estructuras que son comunes en el filtro digital. Como se muestra en la figura 4-2, el bloque SOS es común a los bloques FOS\_L (sistema de primer orden para el filtro pasa bajo) y FOS\_H (sistema de primer orden para el filtro pasa alto) debido que el SOS tiene 2 coeficientes de 3 que son idénticos en contraste al coeficiente que tienen en común FOS\_L y FOS\_H según la tabla 4-5. Dicho ordenamiento permite reducir el número de componentes necesarios para la implementación del filtro, como las unidades de memoria y la lógica digital para el uso de los coeficientes.

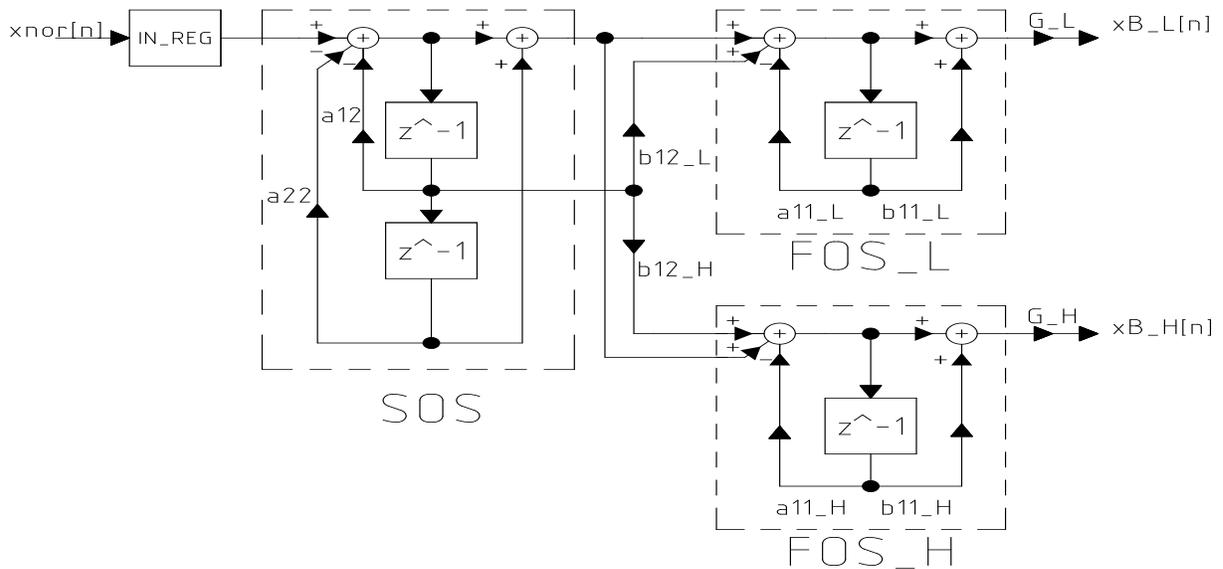


Figura 4-2: Estructura de implementación del bloque elemental del filtro digital factorizado según las etapas del filtro.

## 4.2 Aspectos de implementación del filtro digital

Para implementar una estructura del filtro digital elemental establecida en la figura 4-2 se necesitan definir como primer paso el diseño de los multiplicadores y finalmente el ancho del bus de datos.

Para el diseño de los multiplicadores empleados en el filtro digital se utilizó la codificación CSD (Canonic Signed Digit). Dichos multiplicadores efectúan el cálculo de la señal por una constante o el correspondiente coeficiente. En [41,23] se demuestra la ventajas del método CSD frente a la multiplicación binaria estándar.

**Tabla 4-6: Codificación CSD de los coeficientes del filtro digital según la tabla 4-5.**

|       | Valor decimal | CSD                  | Potencias                    | Codificación decimal | Desvio relativo (%) |
|-------|---------------|----------------------|------------------------------|----------------------|---------------------|
| a01_L | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| b01_L | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| b11_L | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| a11_L | 0,001612      | 000.000000000+0-0+0  | -9;(-)-11;-13                | 0,001587             | 1,572199            |
| b12_L | 1,371566      | 0+0.-0-0000-00+000-  | +1;(-)-1;(-)-3;(-)-8;-11     | 1,371582             | -0,001169           |
| G_L   | 0,211933      | 000.0+00-0-00+0000   | -2;(-)-5;(-)-7;-10           | 0,211914             | 0,008984            |
|       |               |                      |                              |                      |                     |
| a01_H | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| a11_H | 0,001612      | 000.000000000+0-0+0  | -9;(-)-11;-13                | 0,001587             | 1,572199            |
| b01_H | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| b11_H | -1            | 00-.0000000000000000 | (-)-0                        | -1                   | 0                   |
| b12_H | -1,364930     | 0-0.+0+000+0+00+0+   | (-)+1;-1;-3;-7;-9;-12;-14    | -1,364929            | 0,000027            |
| G_H   | 0,210321      | 000.0+00-0-000-0-00  | -2;(-)-5;(-)-7;(-)-11;(-)-13 | 0,210327             | -0,002999           |
|       |               |                      |                              |                      |                     |
| b02   | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| b22   | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| a01   | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
|       |               |                      |                              |                      |                     |
| a02   | 1             | 00+.0000000000000000 | 0                            | 1                    | 0                   |
| a12   | 0,004539      | 000.00000000+00+0+0  | -8;-11;-13                   | 0,004517             | 0,484240            |
| a22   | 0,422254      | 000.+00-0-0000+0-00  | -1;(-)-4;(-)-6;-11;(-)-13    | 0,422241             | 0,003016            |

Los multiplicadores empleados en el filtro digital no deben incrementar los efectos de la cuantificación de coeficientes, por lo que se requiere que operen con todo el rango de representación.

En la tabla 4-6 se resume la codificación CSD para los coeficientes dados en la tabla 4-5. Se decidió ampliar el ancho del bus de datos interno del filtro digital, de 16 bits utilizado en la etapa del normalizador (sección 3.2) hacia los 19 bits para enfrentar los desbordes en los cálculos internos de los filtros debido a que se considera que los coeficientes CSD empleados no alcanzan duplicar y que los acumuladores internos no producen desbordes en la estructura de la figura 4-2.

En contraste con los multiplicadores binarios convencionales, N bits produce resultados de 2N bits, la codificación CSD permite utilizar un ancho de bus de datos adaptable a una implementación específica.

Nótese que los desvíos relativos de todas las codificaciones CSD no superaron el 2% por lo que se considera la codificación CSD como apta el uso de multiplicador.

La anterior codificación CSD de la tabla 4-6 está limitada a la resolución de 15 bits (resguardando el bit de signo),  $\pm 2^{-15} \approx 3.051 \times 10^{-5}$  debido a que el ancho de bus externo utilizado en el banco de filtros es de 16 bits complemento a dos, en contraste a los 19 bits del cálculo interno de los filtros. Nótese que los coeficientes que tienen un número superior de 4 potencias tienen una representación decimal con un desvió menor al 0,01%.

### Caracterización de multiplicadores

En la figura 4-3 se muestra una estructura de un multiplicador CSD para el coeficiente G\_L, combinando desplazamientos en 2, 5 7 y 10 veces la entrada G\_L\_in.

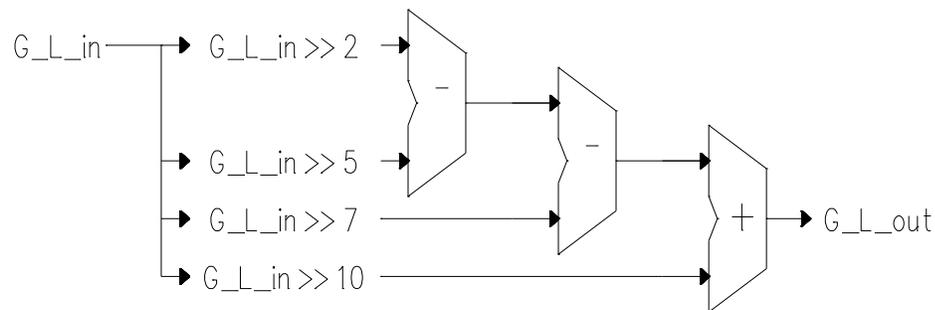


Figura 4-3: Multiplicador CSD para el coeficiente G\_L de la tabla 4-6.

El paradigma de programación correspondiente en VHDL influye en el rendimiento de la utilización del área sintetizada. Para ilustrar el concepto, se presenta el siguiente código en VHDL, siendo `coeficiente_a12_in` como una señal tipo entero de 16 bits.

---

**Código 4.2-1: Corrimiento y suma para la implementación de la multiplicación.**

---

```
coeficiente_a12_v_in <= CONV_STD_LOGIC_VECTOR(coeficiente_a12_in, n_bits);  
  
a_v <= shr(coeficiente_a12_v_in, CONV_STD_LOGIC_VECTOR(8, n_bits));  
b_v <= shr(coeficiente_a12_v_in, CONV_STD_LOGIC_VECTOR(11, n_bits));  
c_v <= shr(coeficiente_a12_v_in, CONV_STD_LOGIC_VECTOR(13, n_bits));  
partial <= (a_v(15) & a_v) + b_v + c_v;
```

---

Como se nota en el código 4.2-1, se realizan 3 diferentes corrimientos hacia la derecha (mediante la función shr, Shift right) para inducir división entre potencias de 2, para finalmente sumar las señales parciales. Dicha forma tiene el desempeño mostrado en la tabla 4-7.

**Tabla 4-7: Resumen de implementación en FPGA para el código 4.2-1.**

| Componente        | Valor |
|-------------------|-------|
| Slices            | 8     |
| LUT de 4 entradas | 13    |

Mediante otro tipo de sintaxis como la mostrada en el código 4.2-2, que utiliza la operación división entre potencias de 2 de manera explícita, tiene el desempeño mostrado en la tabla 4-8.

---

**Código 4.2-2: Corrimiento y suma para la implementación de la multiplicación utilizando señales y operadores tipo entero.**

---

```
a <= coeficiente_a12_v_in / 2**pot1;  
b <= coeficiente_a12_v_in / 2**pot2;  
c <= coeficiente_a12_v_in / 2**pot3;  
partial <= a + b + c;
```

---

La herramienta de síntesis interpreta distintamente la función división asignándole distintas estructuras. En la tabla 4-8 se muestra que el número de componentes en la implementación como LUT de 4 entradas utilizando el código 4.2-2 se incrementa aproximadamente 6 veces en comparación del descrito en el código 4.2-1 que es la adecuada para reducir el consumo de potencia por requerir menor área de implementación.

Tabla 4-8: Resumen de implementación en FPGA para el código 4.2-2.

| Componente        | Valor |
|-------------------|-------|
| Slices            | 41    |
| LUT de 4 entradas | 75    |

#### 4.2.1 Caracterización de las respuestas de los filtros digitales

Los filtros digitales descritos mediante la función de transferencia (4-2) con los coeficientes resumidos en la tabla 4-5 tienen las respuestas al impulso de 1000 unidades mostrada en la figura 4-4 para el filtro pasa alto y pasa bajo individualmente. Dicha respuesta fue captada a partir de la implementación en punto fijo para 20 muestras. En [42] describe el método de Prony para obtener la función de transferencia de un filtro digital tipo IIR a partir de la respuesta del impulso experimental, para obtener los polos y ceros que le corresponde al filtro pasa alto y pasa bajo.

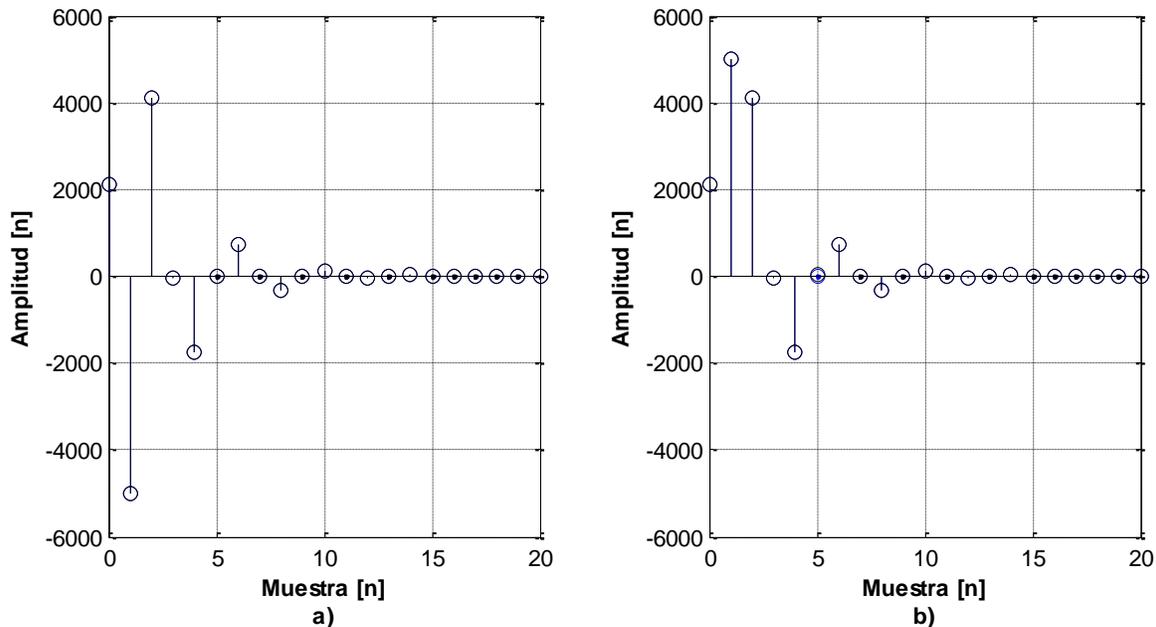


Figura 4-4: Respuesta al impulso de 10000 a) del filtro pasa alto y b) respuesta del filtro pasa bajo.

En la tabla 4-9 se muestran las magnitudes de los polos y ceros que le corresponde al filtro pasa alto. Nótese la similitud de las magnitudes entre la implementación de punto flotante (software) y de punto fijo, cuyo mayor desvío es de 0,0085.

**Tabla 4-9: Resumen de los ceros y polos del filtro pasa alto.**

|       | Punto flotante          | Punto Fijo             | Desvío de la magnitud |
|-------|-------------------------|------------------------|-----------------------|
| Ceros | 1                       | 0,9915                 | 0,0085                |
|       | 1 $\pm 46,9^\circ$      | 0,650 $\pm 46,9^\circ$ | 0,0006                |
| Polos | 0,0016                  | 0,0041                 | 0,0025                |
|       | 0,6498 $\pm 90,2^\circ$ | 0,650 $\pm 90,2^\circ$ | 0,0002                |

**Tabla 4-10: Resumen de los ceros y polos del filtro pasa bajo.**

|       | Punto flotante          | Punto Fijo               | Desvío de la magnitud |
|-------|-------------------------|--------------------------|-----------------------|
| Ceros | 1                       | 0,9999                   | 0,0001                |
|       | 1 $\pm 133,2^\circ$     | 0,9993 $\pm 133,2^\circ$ | 0,0007                |
| Polos | 0,0016                  | 0,0015                   | 0,0001                |
|       | 0,6498 $\pm 90,2^\circ$ | 0,650 $\pm 90,2^\circ$   | 0,0005                |

En la tabla 4-10 se muestra de manera similar que en la tabla 4-9 el desvío de las magnitudes de polos y ceros del filtro paso bajo, cuyo mayor desvío es de 0,0007. Por lo que se demuestra que la implementación en punto fijo es un diseño adecuado para los filtros digitales. Además los nombrados filtros son sistemas estables debido a que los polos se encuentran en la región del círculo unitario.

#### **4.2.2 Detalles de implementación del filtro digital elemental**

En la tabla 4-11 se muestra el resumen de la utilización de componentes para cada uno de los multiplicadores CSD que corresponde a los coeficientes de la tabla 4-6. Nótese que el área de implementación está ligada al número de potencias o corrimientos según la codificación CSD.

**Tabla 4-11: Resumen de la utilización de recursos disponibles para la implementación de los multiplicadores CSD en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| <b>Multiplicador CSD</b> | <b>Potencias</b>             | <b>Componente</b> | <b>Utilizado</b> |
|--------------------------|------------------------------|-------------------|------------------|
| a12                      | -8;-11;-13                   | slices            | 13               |
|                          |                              | 4 input LUT       | 23               |
| a22                      | -1;(-)-4;(-)-6;-11;(-)-13    | slices            | 40               |
|                          |                              | 4 input LUT       | 74               |
| b12_L                    | +1;(-)-1;(-)-3;(-)-8;-11     | slices            | 40               |
|                          |                              | 4 input LUT       | 76               |
| b12_H                    | (-)+1;-1;-3;-7;-9;-12;-14    | slices            | 57               |
|                          |                              | 4 input LUT       | 108              |
| a11_L                    | -9;(-)-11;-13                | slices            | 12               |
|                          |                              | 4 input LUT       | 21               |
| a11_H                    | -9;(-)-11;-13                | slices            | 12               |
|                          |                              | 4 input LUT       | 21               |
| b11_L                    | 0                            | slices            | 0                |
|                          |                              | 4 input LUT       | 0                |
| b11_H                    | (-)0                         | slices            | 10               |
|                          |                              | 4 input LUT       | 19               |
| G_L                      | -2;(-)-5;(-)-7;-10           | slices            | 29               |
|                          |                              | 4 input LUT       | 54               |
| G_H                      | -2;(-)-5;(-)-7;(-)-11;(-)-13 | slices            | 39               |
|                          |                              | 4 input LUT       | 73               |
| Total tipo pasa bajo     | --                           | slices            | 134              |
|                          |                              | 4 input LUT       | 248              |
| Total tipo pasa alto     | --                           | slices            | 171              |
|                          |                              | 4 input LUT       | 318              |

En la tabla 4-12 se muestran la proporción que le corresponde a los multiplicadores CSD por cada tipo de filtro. Nótese que para el caso del filtro pasa bajo, los multiplicadores significan aproximadamente el 66% del total de área de implementación. Por otro lado, para el caso del filtro pasa alto, los multiplicadores significan aproximadamente el 71% del total de área de implementación. Se deduce que una adecuada selección de multiplicadores CSD que sean comunes significa una importante optimización del área de síntesis para establecer la estructura del filtro digital factorizada.

Tabla 4-12: Resumen comparativo de implementación de los filtros digitales paso bajo y paso alto respecto a sus correspondientes ponderaciones de las subestructuras de multiplicadores CSD.

| Componente | Filtro pasa bajo |                     | Filtro pasa alto |                     |
|------------|------------------|---------------------|------------------|---------------------|
|            | Filtro (Total)   | Multiplicadores CSD | Filtro (Total)   | Multiplicadores CSD |
| LUT        | 377              | 248                 | 445              | 318                 |
| Total (%)  | 100              | 65,78               | 100              | 71,46               |

Tabla 4-13: Detalles de implementación entre las estructuras de filtros digitales.

| Filtro      | Slice | Flip Flop | LUT |
|-------------|-------|-----------|-----|
| Pasa alto   | 249   | 73        | 445 |
| Pasa bajo   | 225   | 73        | 377 |
| Dual        | 474   | 146       | 822 |
| Factorizado | 358   | 92        | 646 |

En la tabla 4-13 se detallan los componentes utilizados para sintetizar el filtro pasa alto, pasa bajo y las estructuras dual (unión simple de filtro pasa bajo y alto) y factorizada (versión optimizada de la estructura dual). La estructura factorizada significa una reducción aproximadamente del 21% respecto de todas las *LUT* utilizadas en la versión dual y una optimización del 36% del uso de *Flip-Flops* con referencia del total implementado para la estructura dual.

### 4.3 Diseño del banco de filtros digitales

Según el esquema presentado en la figura 1-1 se requiere la implementación de un banco de filtros que separe la señal previamente normalizada en 8 bandas de frecuencia mostrada en la tabla 4-1.

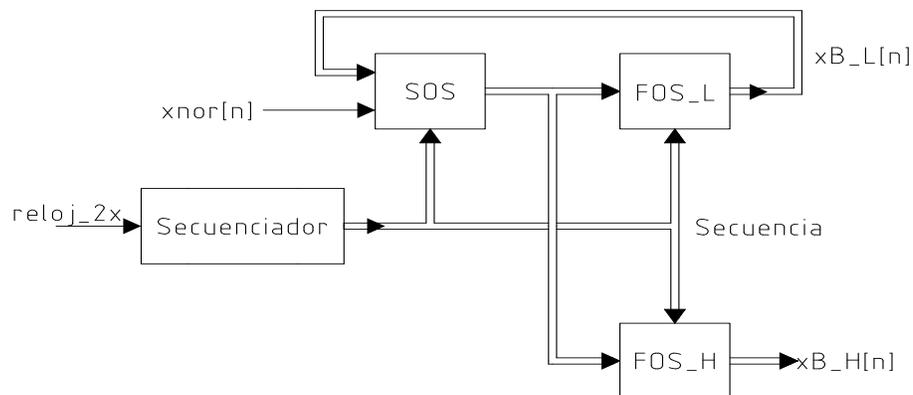
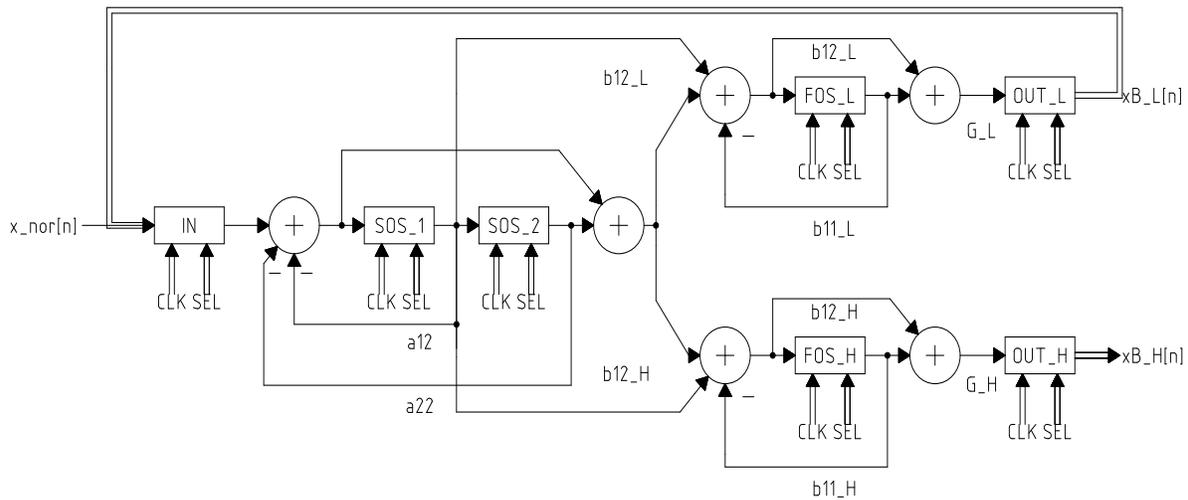


Figura 4-5: Esquema general del banco de filtros segmentado.

El diseño global del banco de filtros corresponde al esquema general según la figura 4-5, catalogado como un sistema de múltiples entradas múltiples salidas, cuya realimentación de la señal filtrada es tomada desde una subbanda inferior hacia una superior. Una característica del presente sistema es la minimización del uso de bloques funcionales, debido que el flujo de datos de todo el banco de filtros utiliza la misma estructura para realizar los cálculos internos de filtrado digital. Dicha propiedad está basada en el modelo de codificación subbanda anteriormente ilustrado en la figura 2-2 en el cual cada filtro digital interno tiene la función de transferencia idéntica para todas las secciones según sea filtro pasa bajo o pasa alto que ligados al submuestreo permiten la bisección consecutiva del espectro.

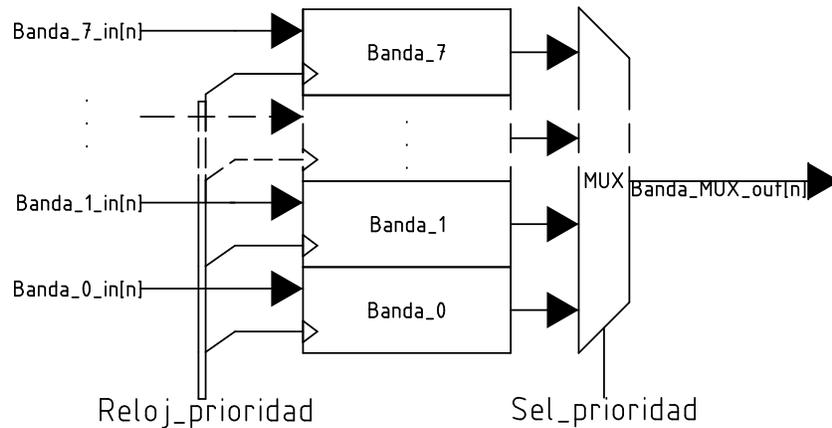


**Figura 4-6: Diagrama detallado de los bloques internos del banco de filtros digitales segmentado.**

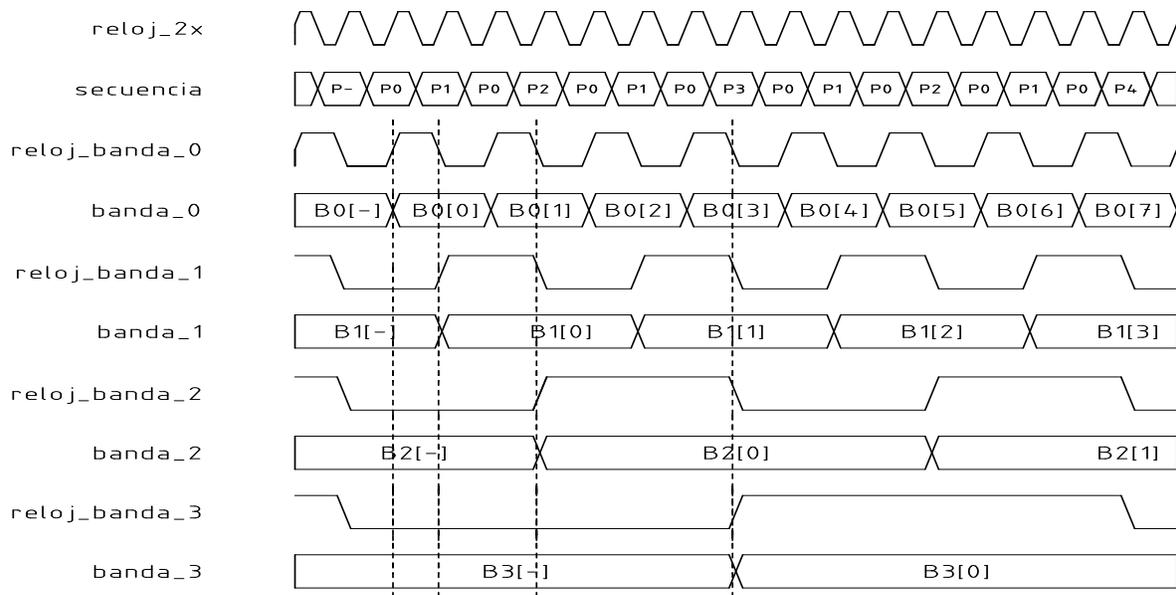
En la figura 4-6 se detalla la distribución de los bloques internos y el flujo de datos en el banco de filtros digitales segmentado. Nótese que el sistema realimenta la salida de la sección pasa bajos (FOS\_L) a través del bloque OUT\_L en conformidad con la estructura establecida en la figura 2-2 que establece el banco de filtros como tipo subbanda asimétrico. La salida del banco de filtros está dada por el bloque OUT\_H con las correspondientes 8 subbandas.

Según la figura 4-6 cada sección funcional del banco de filtros digital opera bajo la misma ruta arquitectónica de cálculo aritmético. Cada resultado por subbanda es almacenado en pilas de registros con selectores (multiplexores) que

almacenan en memoria cada operación efectuada a las muestras de la señal. La estructura de memoria para la entrada de datos (*Bloque\_IN*) se muestra en la figura 4-7, donde las señales *reloj\_prioridad* y *Sel\_prioridad* está regidas por un sistema de selección de prioridad que define la secuencia de trabajo del banco de filtros.



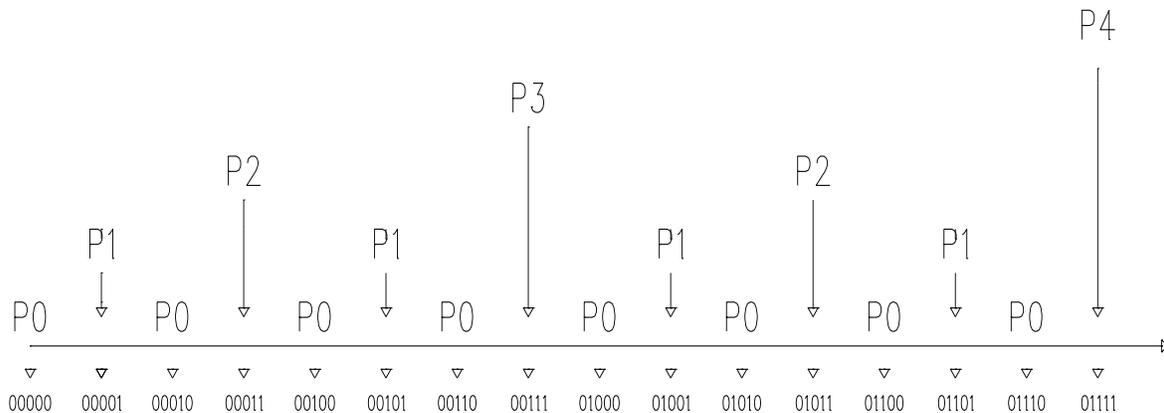
**Figura 4-7: Adaptación en forma de pila de los elementos de retardo ( $z^{-1}$ ) del banco de filtros digital para el bloque IN mostrado en la figura 4-6.**



**Figura 4-8: Temporización para cada subbanda según la secuencia determinada por la figura 4-9.**

En la figura 4-8 se describe el funcionamiento temporal de cada una de las estructuras que calculan las correspondientes subbandas según la secuencia. Dicha secuencia tiene un patrón similar a un codificador de prioridad. Nótese la

multiplexación en el tiempo de los  $P_n$  procesos que se intercalan a través de la línea temporal.



**Figura 4-9: Secuencia de prioridad para la ejecución de cada subbanda correspondiente al banco de filtros.**

En la figura 4-9 se muestra el patrón lógico de la secuencia en función del contador binario que la define. La prioridad está definida como el número del bit menos significativo que contenga el primer cero del contador. Además se nota la relación que existe entre los periodos de los procesos de filtrado como  $P_{n+1} = 2P_n$ .

#### 4.4 Resultados y análisis de la implementación del banco de filtros

El resumen de la implementación del banco de filtros segmentado se muestra en la tabla 4-14. Se detalla la utilización del banco de filtros en términos de los componentes *LUT* y *Flip Flops* implementado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).

**Tabla 4-14: Resumen de la utilización de recursos disponibles para la implementación del banco de filtros segmentado basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| Componente        | Utilizado | Utilización (%) |
|-------------------|-----------|-----------------|
| slices            | 1071      | 23              |
| Flip-Flops        | 1036      | 11              |
| LUT de 4 entradas | 1030      | 11              |

**Tabla 4-15: Detalles de implementación entre distintas tendencias del banco de filtros digital.**

| <b>Banco de filtros</b> | <b>Slice</b> | <b>Flip Flop</b> | <b>LUT</b> |
|-------------------------|--------------|------------------|------------|
| Cascada dual            | 3792         | 1168             | 6576       |
| Cascada factorizada     | 2755         | 724              | 4951       |
| Segmentado              | 1071         | 1036             | 1030       |

En la tabla 4-15 se comparan diversas tendencias de implementación en términos de componentes de FPGA (LUT, Slice, Flip Flop) según sea el tipo de bancos de filtros digital tales el tipo de cascada dual (mostrado en la figura 2-2), cascada factorizada (estructura en cascada basada en el filtro de la figura 4-2) y el banco de filtros segmentado (mostrado en la figura 4-6). Nótese que el tipo segmentado está optimizado en área respecto a las otras tendencias.

**Tabla 4-16: Resumen de factores de optimización respecto al área de implementación para distintas tendencias del banco de filtros digital según la tabla 4-15.**

| <b>Banco de filtros</b> | <b>Slice</b> | <b>Flip Flop</b> | <b>LUT</b> |
|-------------------------|--------------|------------------|------------|
| Cascada dual            | 1            | 1                | 1          |
| Cascada factorizada     | 1,38         | 1,61             | 1,33       |
| Segmentado              | 3,54         | 1,13             | 6,38       |

La relación de optimización entre las tendencias de los bancos de filtros se muestra en la tabla 4-16. Se toma como referencia el tipo de cascada dual como implementación primitiva. Nótese que el tipo de banco segmentado presenta un factor de optimización de 6,38 (elemento LUT) en contraste con la implementación cascada factorizada que presenta un factor de optimización de 1,33. En términos de unidades de almacenamiento (*flip flop*) la optimización es aproximadamente 1,13 para la estructura segmentada en comparación con la implementación cascada factorizada cuya razón de optimización alcanzó el valor de 1,61 debido que la tendencia segmentada requiere almacenamiento adicional de los cálculos por cada proceso.

#### 4.4.1 Resultados y evaluación de la etapa de banco de filtros

En la figura 4-10 se muestra la respuesta del banco de filtros segmentado ante una prueba de un barrido de frecuencia de 10 Hz hasta 20 kHz en 5 s para cada sección subbanda.

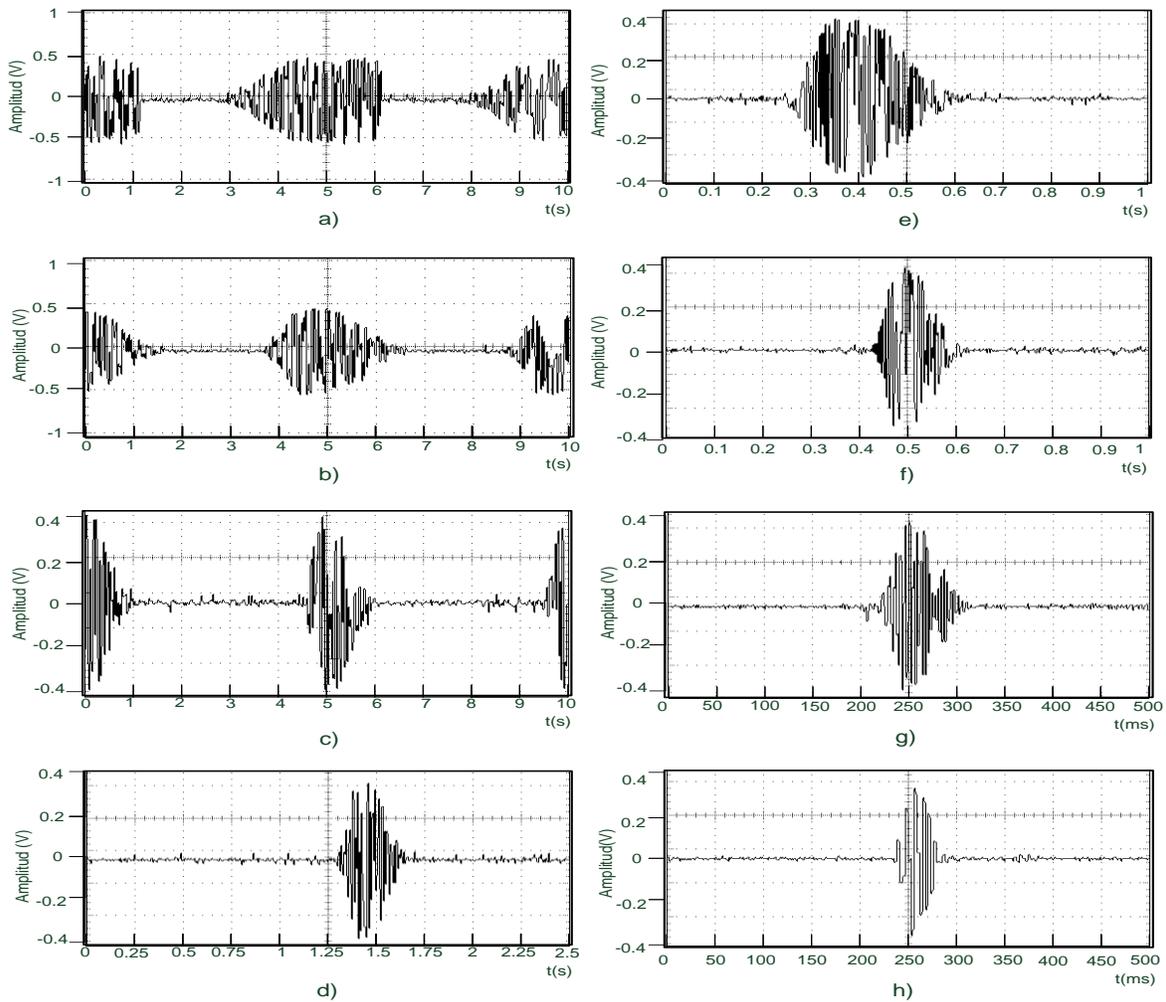
Tabla 4-17: Ancho de banda para cada filtro digital según la figura 4-10 según la relación (4-3).

| Subbanda | $\Delta_t$ (s) | BW experimental(Hz) | BW requerido(Hz) | Desviación (%) |
|----------|----------------|---------------------|------------------|----------------|
| 0        | 2,2            | 8795,6              | 8975             | 1,99           |
| 1        | 1,3            | 5197,4              | 5512,5           | 12,24          |
| 2        | 0,65           | 2598,7              | 2756,25          | 4,26           |
| 3        | 0,3            | 1199,4              | 1378,13          | 4,26           |
| 4        | 0,145          | 579,71              | 689,06           | 7,45           |
| 5        | 0,075          | 299,85              | 344,53           | 4,26           |
| 6        | 0,035          | 139,93              | 172,27           | 10,65          |
| 7        | 0,018          | 71,964              | 86,13            | 8,09           |

En la tabla 4-17 se resume el ancho de banda requerido y obtenido en dicha prueba para cada subbanda según la tabla 4-1. Nótese que todas las respuestas denotan el espectro para cada subbanda y la relación que existe con el ancho de banda correspondiente es

$$\Delta f = \frac{f_{final} - f_{inicial}}{t_b} \Delta t \quad (4-3)$$

ante una señal de prueba en forma de barrido de frecuencias desde  $f_{inicial} = 10Hz$  hasta  $f_{final} = 20kHz$  en un intervalo de  $t_b = 5s$ .



**Figura 4-10: Respuesta por subbanda del banco de filtros ante una prueba de barrido de frecuencia de 10 Hz a 20 kHz en 5 s. En a) se ilustra la respuesta para la subbanda 0. En b) se ilustra la respuesta para la subbanda 1. Sucesivamente hasta h) para la subbanda 7.**

En la tabla 4-17 se resumen los resultados de la prueba según la figura 4-10. El ancho de banda experimental de la subbanda inferior es aproximadamente el doble respecto de la subbanda subsecuente. Nótese que la mayor desviación porcentual del ancho de banda experimental respecto al requerido es de 12,24% correspondiente a la subbanda 1.

#### 4.4.2 Consumo de energía entre las tendencias de banco de filtros

La estimación del consumo de potencia en dispositivos FPGA depende de la cantidad y configuración de la lógica. Para producir estimaciones se requieren valores como la utilización de recursos, frecuencia de reloj y razones de conmutación.

**Tabla 4-18: Consumo de potencia entre las distintas tendencias de implementación de banco de filtros digital a la frecuencia de muestreo de 44,1 kHz.**

| <b>Tendencia de banco de filtros</b> | <b>Potencia estática (mW)</b> | <b>Potencia dinámica (μW)</b> | <b>Potencia total (mW)</b> |
|--------------------------------------|-------------------------------|-------------------------------|----------------------------|
| Cascada dual                         | 80,99                         | 950                           | 81,95                      |
| Cascada factorizada                  | 80,99                         | 940                           | 81,94                      |
| Segmentada                           | 80,99                         | 500                           | 81,49                      |

En la tabla 4-18 se resumen los consumos de potencia entre las tendencias de implementación de los bancos de filtros digitales (tabla 4-16). Nótese en la optimización en términos de potencia dinámica que existe entre el diseño del banco de filtros segmentado que alcanza una reducción del 47,37% respecto al consumo del banco de filtros en dual (mostrado en la figura 2-2). De manera similar, el banco de filtros segmentado tiene una optimización en potencia dinámica del 46.8% respecto a la disipación en el banco de filtros en configuración cascada factorizada. Para estimar el consumo de potencia se aplicó a los distintos bancos de filtros una prueba de esfuerzo mediante una señal cuadrada de 129 Hz con una amplitud a todo el ancho de la representación desde  $-2^{n_{bits}-1}$  hasta  $2^{n_{bits}-1} - 1$ . Dicha señal pretende excitar todas las secciones debido a su contenido armónico y su propagación a través de toda la estructura en cascada del banco de filtros y la topología subbanda.

**Tabla 4-19: Consumo de potencia dinámica en el banco de filtros segmentado para distintas frecuencias de muestreo.**

| <b>Frecuencia de muestreo (kHz)</b> | <b>Potencia dinámica (<math>\mu</math>W)</b> |
|-------------------------------------|--|
| 44,1                                | 500  |
| 22,05                               | 340  |
| 11,025                              | 220  |

La potencia estática mostrada en la tabla 4-18 representa la potencia eléctrica que consume el FPGA por concepto de alimentación y programación sin influencia de algún diseño. La potencia estática incluye la disipación de energía por efecto de las corrientes de fuga de los transistores, el consumo interno y la potencia disipada por las resistencias de terminación del dispositivo. Por esa razón, la potencia estática mostrada en la tabla 4-18 es constante sin importar la implementación. En cambio la potencia dinámica corresponde al consumo de energía producto de las transiciones correspondientes al tratamiento de la señal en las distintas arquitecturas del banco de filtros.

En la tabla 4-19 se muestra el consumo de potencia dinámica para el banco de filtros tipo segmentado en función de la frecuencia de muestreo utilizada. Nótese que disminuir a la mitad la frecuencia de muestreo representa una reducción aproximadamente de 32% del consumo de potencia dinámica.

## 4.5 Estimadores de energía por subbanda

La estimación de energía se realiza por medio del cálculo del promedio de la magnitud de la señal. Dicho cálculo se realiza por medio del promediador seguidor. En la figura 4-11 se muestra el algoritmo de promediador tipo seguidor.

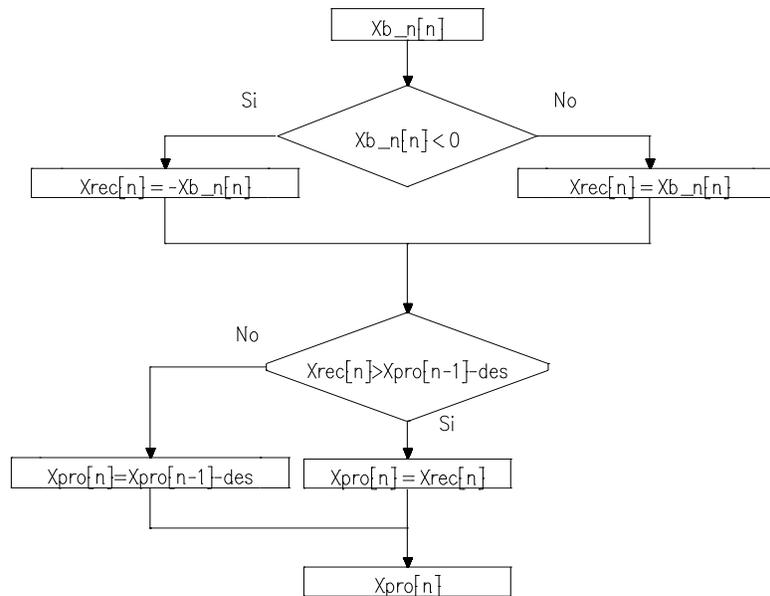


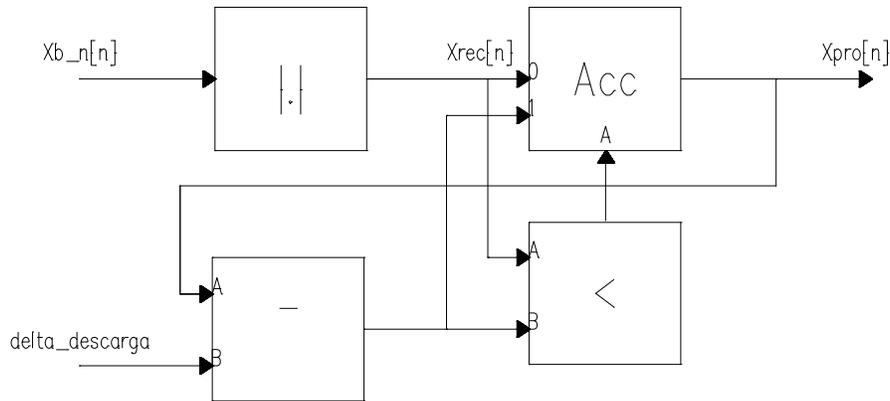
Figura 4-11: Algoritmo del promediador tipo seguidor.

El promediador seguidor es la estructura mostrada en la figura 4-12 cuyo funcionamiento está descrito por

$$X_{pro}[n] = \begin{cases} X_{pro}[n-1] - \Delta_{descarga} & \text{para } X_{rec}[n] < X_{pro}[n-1] - \Delta_{descarga} \\ X_{rec}[n] & \text{el resto} \end{cases} \quad (4-4)$$

$$\Delta_{descarga} = 2^n, n = 0, 1, \dots, k-1 \quad (4-5)$$

El bloque Acc actúa como acumulador que almacena el valor de  $X_{rec}[n]$  y la tendencia de descarga.



**Figura 4-12: Estructura para el estimador de energía para cada subbanda basado en el algoritmo de la figura 4-11.**

La tendencia de descarga  $\Delta_{descarga}$  determina la velocidad de descarga del acumulador. Para cada subbanda del banco de filtros le corresponde un promediador seguidor con un valor de  $\Delta_{descarga}$  distinto para calcular la energía promedio. El valor de  $\Delta_{descarga}$  depende del nivel de la subbanda y esta descrita por (4-5) donde  $k$  representa el nivel de la subbanda.

#### 4.6 Resultados de la implementación de la etapa de extracción de características

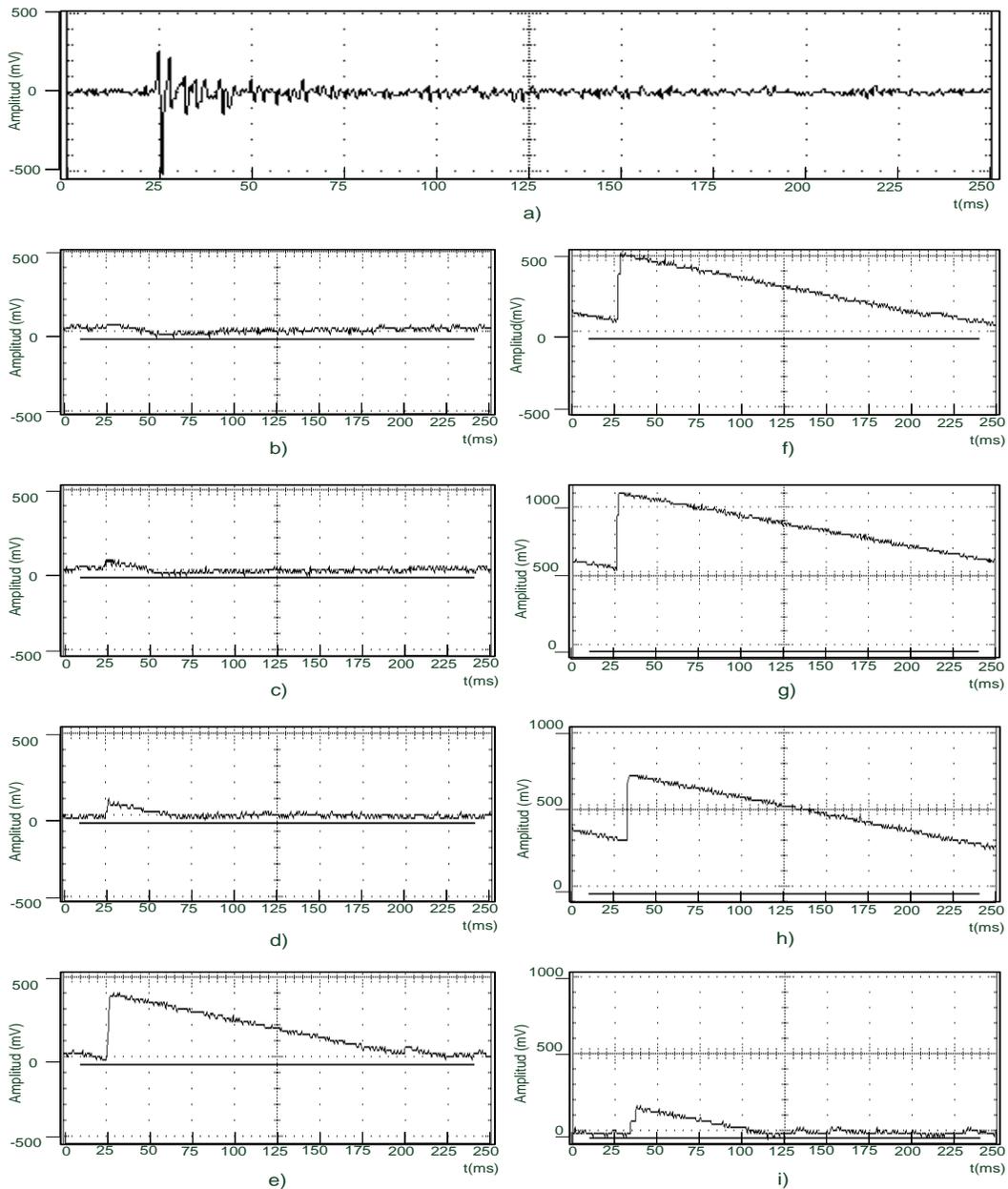
En la tabla 4-20 se muestra el resumen de la utilización de componentes en la implementación de la etapa de extracción de características de las señales acústicas. Se incluyen los bloques de adquisición, normalización, el banco de filtros tipo segmentado y los estimadores de energía por subbanda. Nótese la utilización de sólo un multiplicador dedicado y sólo 23% de LUT disponibles para el dispositivo FPGA Spartan 3E de Xilinx (ver anexo B. 1).

**Tabla 4-20: Resumen de la utilización de recursos disponibles para la implementación de la etapa de extracción de características basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| Componente          | Utilizado | Utilización (%) |
|---------------------|-----------|-----------------|
| slices              | 1580      | 33              |
| Flip-Flops          | 1262      | 13              |
| LUT de 4 entradas   | 2173      | 23              |
| Multiplicador 18x18 | 1         | 5               |

#### 4.6.1 Funcionamiento de la etapa de extracción de características

La etapa de extracción de características estima la energía promedio por subbanda de las señales acústicas.



**Figura 4-13: Estimación del máximo de energía por subbanda ante a) una señal de prueba disparo. En b) se ilustra la respuesta para la subbanda 0. En b) se ilustra la respuesta para la subbanda 1. Sucesivamente hasta i) para la subbanda 7.**

En la figura 4-13 se muestra la estimación de energía por subbanda para una señal proveniente de un disparo normalizado. Nótese las diferencias entre los

distintos resultados de los estimadores de energía para cada subbanda en particular.

Se presentan 2 aspectos: se extrae la información del contenido en frecuencia de la señal en la que se concluye que existe mayor información entre la subbanda 3 (figura 4-13.e) hasta la subbanda 6 (figura 4-13.h) y finalmente se demuestran las diferencias entre los tiempos de aparición para cada contenido de frecuencia por subbanda. En las subbanda 6 (figura 4-13.h) y subbanda 7 (figura 4-13.i) dicho efecto es notable.

## Capítulo 5: Etapa de clasificación

En este capítulo se describe la estructura del circuito digital requerido para implementación de los Modelos Ocultos de Markov, el cual analiza las cadenas de observación que provienen de la etapa de codificación mediante el algoritmo hacia adelante (*forward*). La estructura debe clasificar los eventos en actividad normal del bosque, disparo o sonido de motosierra, aún con ruido, distorsión o algún efecto que altera el sonido. El algoritmo hacia adelante es un conjunto de procedimientos de cálculo matricial que requiere aritmética en punto flotante debido al tratamiento de probabilidades.

Para la etapa de clasificación y toma de decisiones que integra el prototipo mostrado en la figura 1-1 se diseñó el *MAP (Matrix and Arrays Processor)*, que es una estructura digital cuya arquitectura está optimizada para ejecutar operaciones en punto flotante de manera combinacional con entradas matriciales o arreglos. Esta arquitectura soporta el estándar de punto flotante de 32 bits (IEEE – 754) [43,39,38]. En la figura 5-1 se muestra el diagrama de puertos del sistema MAP y en la tabla 5-1 se resume la descripción de pines del *MAP*.

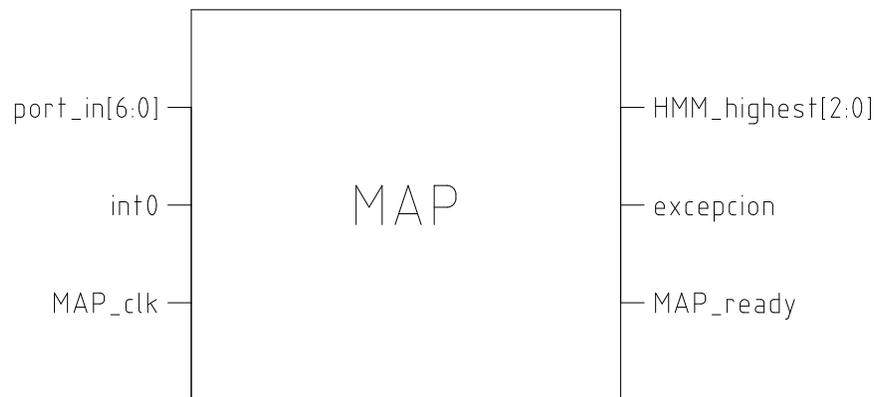


Figura 5-1: Diagrama de pines del MAP.

La entrada de reloj (*MAP\_clk*) del sistema tiene la frecuencia mínima de 800 Hz que es relativa al periodo de muestreo de la cadena de observación, según [6] corresponde a 0,1 s. La entrada *Port\_in* corresponde al puerto de ingreso de datos al sistema que representa la entrada de símbolos del alfabeto de la cadena de observación para que sea procesado por el *MAP*. La entrada *int0* carga cada

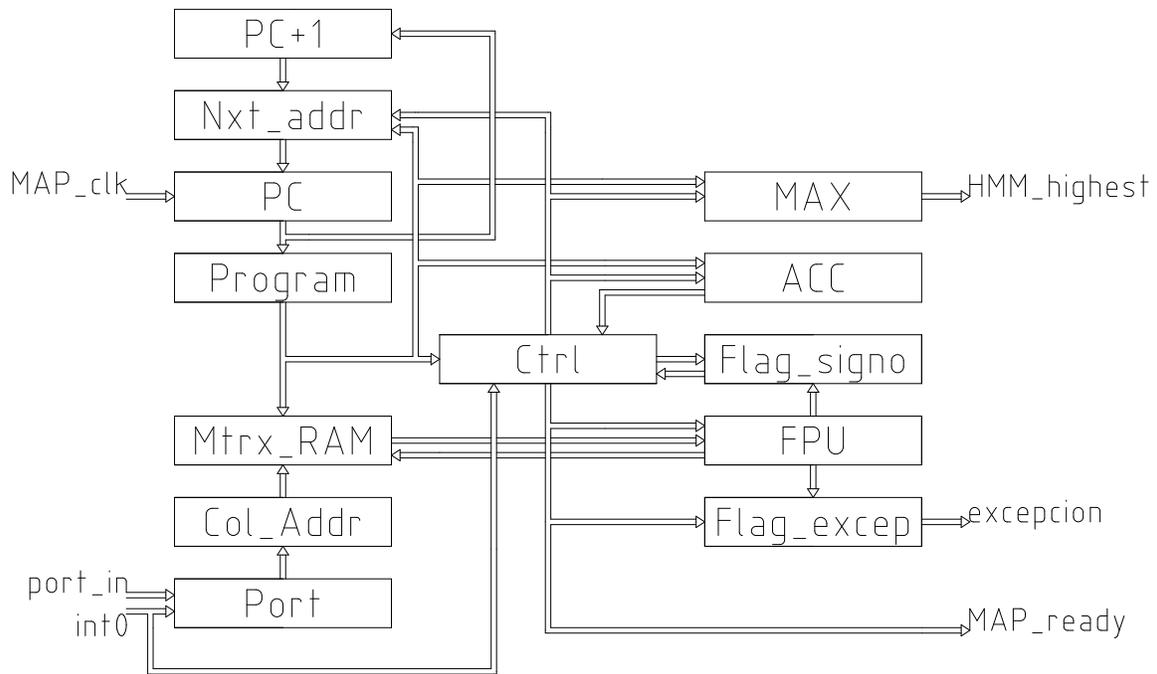
símbolo ingresado en el puerto. Respecto a las salidas del sistema del *MAP*, las señales de los pines de *HMM\_highest* representan el modelo *HMM* de mayor probabilidad que denota el estado del bosque. El estado de *MAP\_ready* indica la condición de espera por un nuevo símbolo. La condición de excepción muestra que los cálculos aritméticos internos no son válidos por causa de la ejecución de división entre cero.

**Tabla 5-1: Descripción de los pines del MAP según la figura 5-1.**

| <b>Puerto</b> | <b>Descripción</b>   |
|---------------|--|
| MAP_clk       | Entrada de reloj   |
| int0          | Entrada de interrupción externa para el ingreso de datos al puerto Port_in   |
| Port_in       | Ingreso de datos que corresponde a cada carácter de la cadena de observación |
| HMM_highest   | Puerto de salida codificada que indica cual HMM tiene mayor probabilidad     |
| Excepcion     | Señal de salida que muestra una condición interna de excepción numérica      |
| MAP_ready     | Señal de salida que indica que el sistema está listo para recibir datos      |

## **5.1 Arquitectura del MAP que implementa el método de Forward basado en HMM**

El circuito mostrado en la figura 5-2 detalla los bloques funcionales del *MAP*. La arquitectura incluye un controlador de datos de entrada (*Port*), un banco de memoria (acceso y almacenamiento temporal de cálculos aritméticos, *Mtrx\_RAM*), la memoria de programa (*Program*), un contador de programa (*PC*), el control general de la arquitectura (*Ctrl*), el acumulador de los *HMM* de probabilidad máxima (*MAX*), el acumulador de número de símbolo (*ACC*), la unidad de aritmética en punto flotante (*FPU*) y el bloque de la bandera de excepción.



**Figura 5-2: Diagrama de bloques general del MAP.**

El *MAP* es un microprocesador dedicado de funcionamiento monociclo cuya arquitectura es tipo Harvard al poseer los bloques de memoria separados (memoria de instrucciones y datos).

La arquitectura del *MAP* soporta operaciones de punto flotante (suma, comparación, multiplicación y división), operaciones en punto fijo como el decremento de un registro índice, instrucciones de bifurcación como el salto tipo con o sin condición.

El cálculo de direcciones para el acceso de las instrucciones es distinto para el almacenamiento de los datos. A continuación se describen los bloques de banco de memoria y la unidad de punto flotante.

### 5.1.1 Descripción de la memoria de acceso y almacenamiento de datos

La estructura *MAP* tiene un banco de memoria para el acceso de las matrices. Esta unidad prevé bloques de cálculo de direcciones para el alineamiento por filas y columnas.

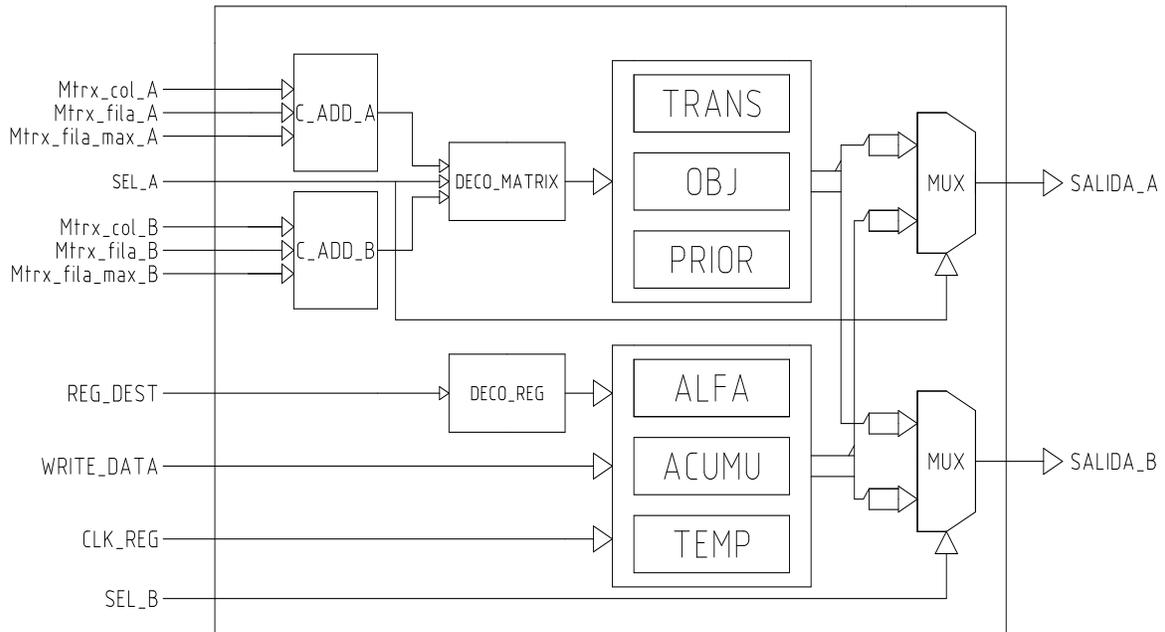


Figura 5-3: Diagrama de la estructura interna del banco de memoria de datos.

Para el almacenamiento de datos se tiene un arreglo de registros con la habilidad de lectura independiente de la escritura. El banco de memoria direcciona 2 elementos simultáneamente en forma separada de cualquier lugar del mapa de direcciones disponible para la obtención de los operandos.

En la tabla 5-2 se resume la utilización de componentes usados para la implementación del banco de memoria. Nótese el uso de 2 multiplicadores dedicados para el cálculo del acceso de las filas y columnas correspondientes a las matrices. El espacio de registros requiere sólo el 7% del área disponible para *Flip flops* del FPGA Spartan 3E de Xilinx (ver anexo B. 1).

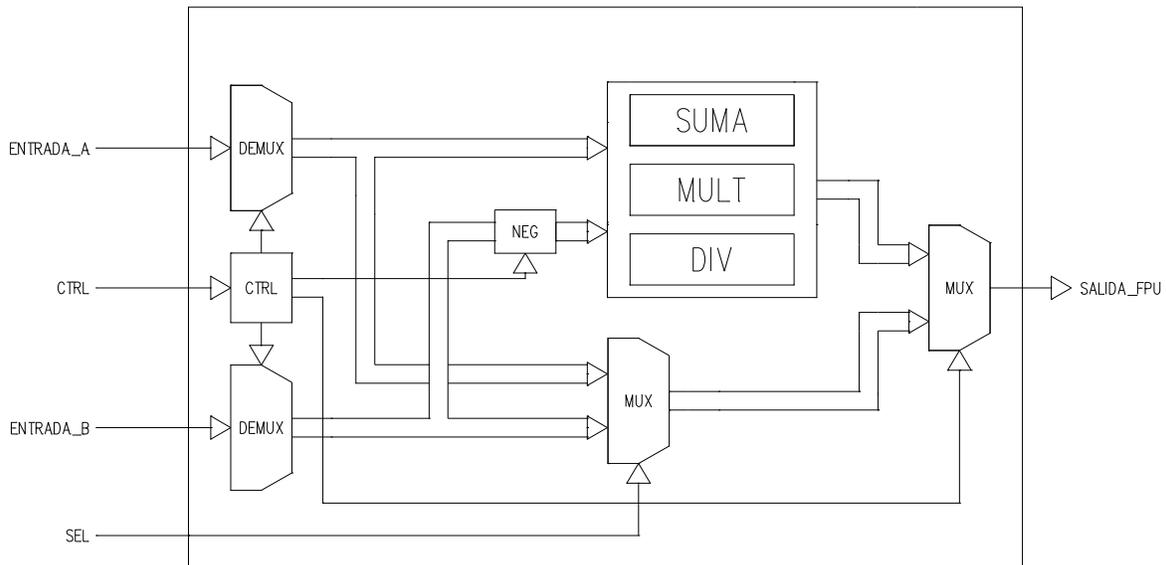
**Tabla 5-2: Resumen de la utilización de recursos disponibles para la implementación del banco de memoria del MAP basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| Componente          | Utilizado | Utilización (%) |
|---------------------|-----------|-----------------|
| slices              | 1175      | 25              |
| Flip-Flops          | 704       | 7               |
| LUT de 4 entradas   | 2227      | 23              |
| Multiplicador 18x18 | 2         | 10              |

### 5.1.2 Descripción de la unidad aritmética de punto flotante

El funcionamiento del MAP está especializado en las operaciones aritméticas de punto flotante con precisión de 32 bits (IEEE – 754). Dichas operaciones son implementadas en estructuras digitales combinatoriales separadas (suma, multiplicación y división). La unidad de punto flotante es una adaptación de la biblioteca para VHDL desarrollada para FPGA bajo licencia GPL [45].

En la figura 5-4 se muestran los bloques internos de FPU. Para el caso de la resta, se utiliza un bloque negador del argumento. Se incluye una estructura para soportar la selección de las operaciones según la señal de control (*ctrl*).



**Figura 5-4: Esquema interno de la unidad de aritmética en punto flotante (FPU).**

En la tabla 5-3 se muestran los componentes utilizados para la implementación de las operaciones en punto flotante (8% y 13% de componentes

LUT para la suma y división respectivamente). Como criterio de optimización de consumo de potencia, los bloques dedicados para las operaciones en punto flotantes son deshabilitados en condición de desuso por medio de demultiplexores.

**Tabla 5-3: Resumen de la utilización de recursos disponibles para la implementación de la unidad de suma del FPU basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| Operación de punto flotante | Componente          | Utilizado | Utilización (%) |
|-----------------------------|---------------------|-----------|-----------------|
| Suma                        | slices              | 429       | 9               |
|                             | LUT de 4 entradas   | 767       | 8               |
| Multiplicación              | slices              | 134       | 2               |
|                             | LUT de 4 entradas   | 245       | 2               |
|                             | Multiplicador 18x18 | 4         | 20              |
| División                    | slices              | 633       | 13              |
|                             | LUT de 4 entradas   | 1217      | 13              |

## 5.2 Detalles de implementación del MAP y evaluación de los modelos

La implementación del MAP requiere el 57% del área disponible de componentes LUT en el FPGA debido al uso de memorias especializadas para el acceso de las matrices para cada HMM. Según la tabla 5-4 dicha estructura utiliza el 30% para la multiplicación en punto flotante de los 20 multiplicadores dedicados disponibles y el 7% de los Flip-Flops aprovechables en la plataforma utilizada para el almacenamiento de datos de 32 bits de ancho.

**Tabla 5-4: Resumen de la utilización de recursos disponibles para la implementación del MAP basado en una FPGA Spartan 3E de Xilinx (ver anexo B. 1).**

| Componente          | Utilizado | Utilizado (%) |
|---------------------|-----------|---------------|
| slices              | 2861      | 61            |
| Flip-Flops          | 720       | 7             |
| LUT de 4 entradas   | 5308      | 57            |
| BRAM                | 2         | 10            |
| Multiplicador 18x18 | 6         | 30            |

### 5.2.1 Descripción del firmware del MAP e implementación del algoritmo hacia adelante

Para el cálculo de la probabilidad de que una cadena de observaciones pertenezca a un modelo dado (motosierra, disparo o bosque en condiciones normales) se implementó una secuencia de instrucciones del MAP basado en el algoritmo hacia adelante descrito el código f-1 (ver anexo F. 1) que corresponde a una adaptación en programación en alto nivel de la evaluación de modelos (ver sección 2.3.1).

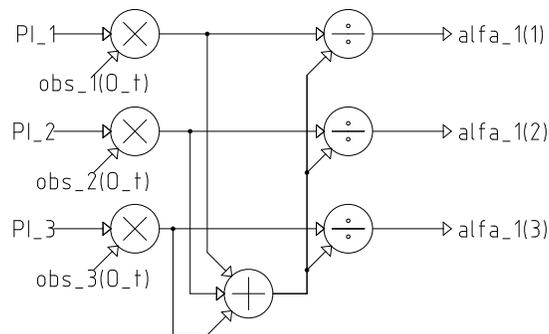
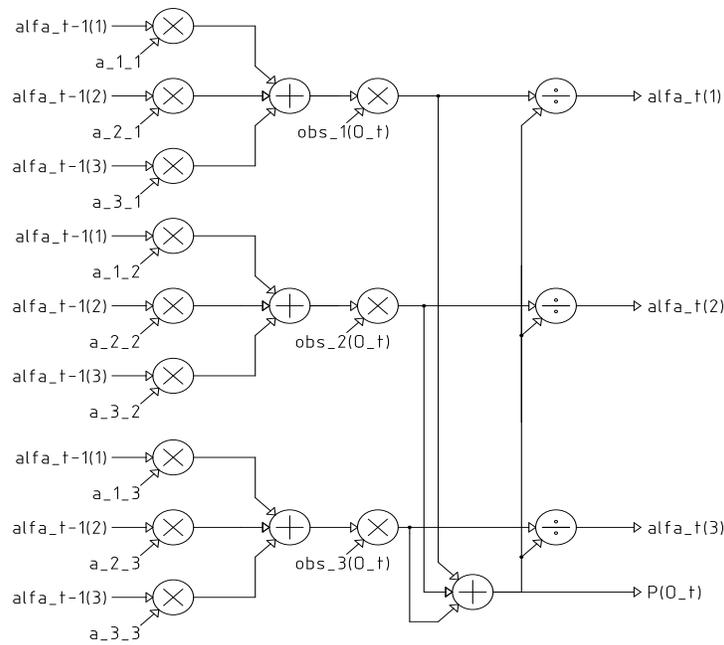


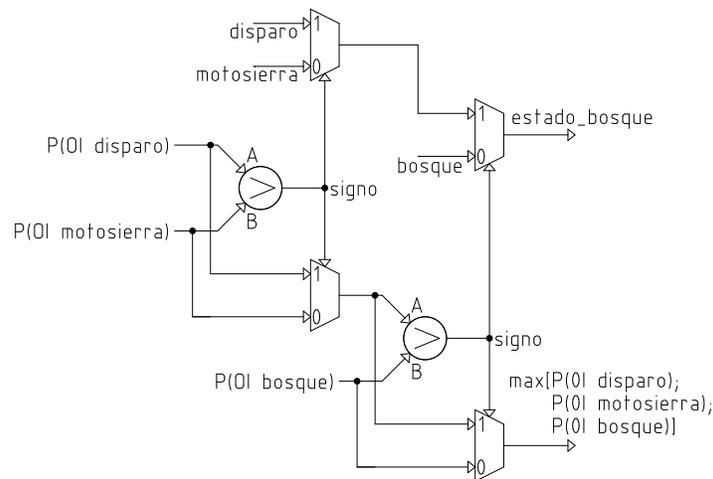
Figura 5-5: Flujo de datos para el cálculo del paso inicialización del algoritmo de adelante para un HMM.

La secuencia de instrucciones para la evaluación de los *HMM* se puede describir con el flujo de datos mostrado en la figura 5-6; dicho flujo detalla la fase de inicialización para cada *HMM* por separado. Se incluye la normalización de las probabilidades.



**Figura 5-6: Flujo de datos para el cálculo del paso inducción del algoritmo de adelanto para un HMM.**

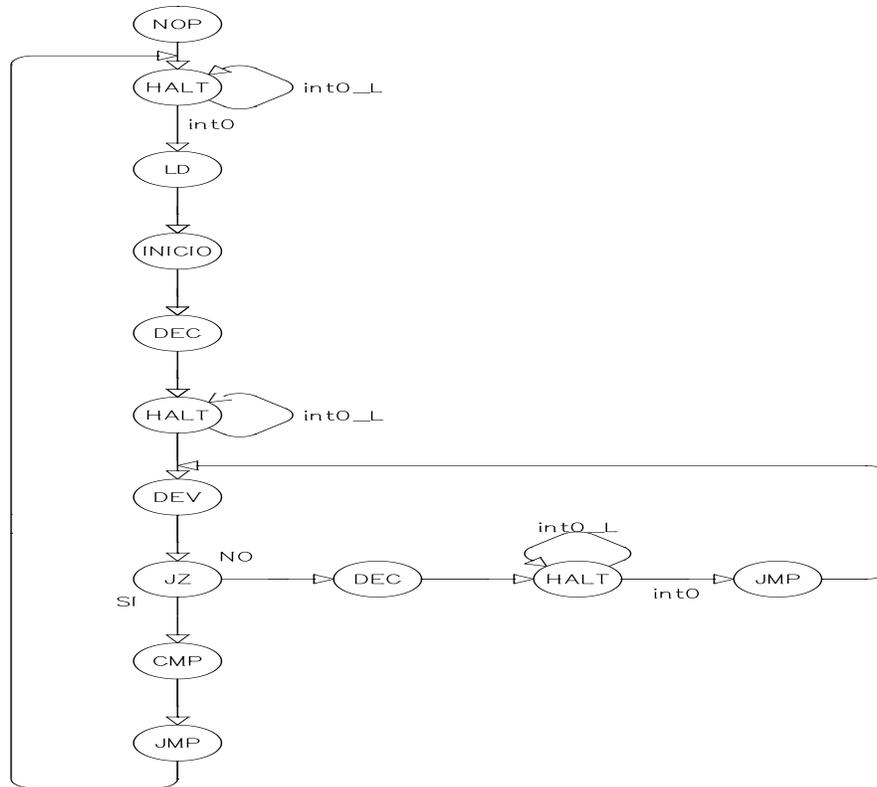
En la figura 5-6 se especifican las operaciones aritméticas y el flujo de datos para la fase de inducción necesarios para el cálculo de la variable de adelanto para cada *HMM* (ver sección 2.3.1).



**Figura 5-7: Flujo de datos para la selección de la máxima probabilidad entre los modelos de disparo, motosierra y bosque normal para determinar el estado del bosque.**

Para la determinación del estado del bosque debe seleccionarse el *HMM* que tenga la mayor probabilidad para una cadena de observación arbitraria. En la figura 5-7 se muestra el flujo de datos para la fase comparativa entre los *HMM* correspondientes al disparo, motosierra y bosque normal. La señal *estado\_bosque*

(codificada como: disparo = 001, motosierra = 010 y bosque = 100) establece el valor en el puerto *HMM\_highest* del MAP.



**Figura 5-8: Diagrama de estados para la ejecución del algoritmo hacia adelante.**

El diagrama de estados mostrado en la figura 5-8 define el procedimiento para la carga de los símbolos de la cadena de observación seguido con la evaluación de los tres *HMM*. El diagrama de estados está compuesta de dos ciclos: el ciclo principal establece el retorno al estado *HALT* y el ciclo interno procesa cada símbolo. El estado *HALT* define la espera por el siguiente símbolo de la cadena.

La secuencia inicia con el ingreso del primer símbolo de la cadena de observación. La carga de dicho símbolo ocurre con la condición de *intO* asertiva e inicia la ejecución del algoritmo hacia adelante.

**Tabla 5-5: Resumen de las funciones de cada estado según la figura 5-8.**

| <b>Estado</b> | <b>Función</b>   |
|---------------|--|
| NOP           | No realiza ninguna operación   |
| Halt          | Detención en espera de datos en el puerto de entrada                           |
| LD            | Carga el valor de la longitud de la cadena de observación                      |
| Inicio        | Ejecuta el paso de inicialización por cada HMM                                 |
| DEC           | Decrementa en uno el contador de símbolos de cadena de observación             |
| DEV           | Ejecuta el paso de inducción por cada HMM según el símbolo entrante            |
| JZ            | Salto si el contador de símbolos es cero                                       |
| JMP           | salto sin condición  |
| CMP           | Procedimiento de comparación para el cálculo de la máxima probabilidad por HMM |

En la tabla 5-5 se muestra un resumen de las funciones de cada estado que compone la secuencia descrita en la figura 5-8. Cada estado está compuesto de instrucciones del *MAP* con funciones especializadas. El estado *INICIO* corresponde a la etapa de inicialización presentado en el código f-1.

Adicionalmente en el estado *DEV* se ejecuta la sección de inducción. Finalmente en el estado *CMP* se ejecutan las sentencias terminales de comparación entre las probabilidades calculadas para cada uno de los modelos.

En la sección 5.2.3 se presenta el resumen comparativo entre las instrucciones ejecutadas en la implementación del algoritmo hacia adelante.

### 5.2.2 Resultados de la etapa de clasificación

En la tabla 5-6 se muestran los resultados de la evaluación de los modelos (disparo, motosierra y bosque normal) en el MAP para distintas cadenas de observación.

**Tabla 5-6: Comparación entre implementaciones basadas en software y en hardware para el algoritmo forward según HMM del reconocimiento de patrones para algunas cadenas de observación.**

| Cadenas de observación  |            | 29;14;30;15            | 29;30;14;15            | 22;9;11;15             | 22;28;18;27            |
|-------------------------|------------|------------------------|------------------------|------------------------|------------------------|
| Implementación software | Disparo    | $1,115 \cdot 10^{-5}$  | $3,963 \cdot 10^{-2}$  | $1,115 \cdot 10^{-5}$  | $2,495 \cdot 10^{-5}$  |
|                         | Motosierra | $1,107 \cdot 10^{-6}$  | $1,107 \cdot 10^{-6}$  | $1,107 \cdot 10^{-6}$  | $1,987 \cdot 10^{-3}$  |
|                         | Bosque     | $8,646 \cdot 10^{-6}$  | $6,813 \cdot 10^{-2}$  | $8,891 \cdot 10^{-6}$  | $5,955 \cdot 10^{-5}$  |
| Implementación hardware | Disparo    | $1,115 \cdot 10^{-5}$  | $3,963 \cdot 10^{-2}$  | $1,115 \cdot 10^{-5}$  | $2,495 \cdot 10^{-5}$  |
|                         | Motosierra | $1,107 \cdot 10^{-6}$  | $1,107 \cdot 10^{-6}$  | $1,107 \cdot 10^{-6}$  | $1,987 \cdot 10^{-3}$  |
|                         | Bosque     | $8,646 \cdot 10^{-6}$  | $6,813 \cdot 10^{-2}$  | $8,891 \cdot 10^{-6}$  | $5,955 \cdot 10^{-5}$  |
| Modelo reconocido       |            | Disparo                | Bosque                 | Disparo                | Motosierra             |
| Desvío máximo           |            | $3,282 \cdot 10^{-13}$ | $1,312 \cdot 10^{-10}$ | $1,345 \cdot 10^{-12}$ | $2,379 \cdot 10^{-10}$ |

La prueba consiste en el ingreso consecutivo de 4 cadenas diferentes de 4 símbolos de longitud. Las probabilidades resultantes entre los HMM de disparo, motosierra y bosque se comparan con las obtenidas en la implementación en software (código f-1, ver apéndice F. 1).

Los resultados de la implementación del MAP están en formato de punto flotante de 32 bits comparados con la implementación en software que utiliza el estándar de 64 bits, se obtiene la desviación en el orden  $10^{-10}$  según la tabla 5-6.

### 5.2.3 Uso de recursos en el FPGA en la evaluación de los HMM

En la tabla 5-7 se muestra la cuenta de instrucciones para un ciclo de ejecución del algoritmo hacia adelante (forward). Nótese que la instrucción de multiplicación es ejecutada un 50% y la suma un 25.64% ambas para un total de 234 instrucciones ejecutadas.

**Tabla 5-7: Resumen del número y clase de instrucciones para la evaluación de los HMM.**

| <b>Tipo de instrucción</b> | <b>Clase</b>        | <b>Cuenta de instrucciones</b> | <b>Procentaje por clase (%)</b> |
|----------------------------|---------------------|--------------------------------|---------------------------------|
| Punto Flotante             | Suma                | 60                             | 25,64                           |
|                            | Multiplicación      | 117                            | 50,00                           |
|                            | División            | 36                             | 15,38                           |
|                            | Comparación         | 2                              | 0,85                            |
| Aritmética entera          | Carga               | 1                              | 0,43                            |
|                            | Decrementar         | 4                              | 1,71                            |
| Salto                      | Salto incondicional | 4                              | 1,71                            |
|                            | Salto si cero       | 3                              | 1,28                            |
| Misceláneas                | Detención           | 4                              | 1,71                            |
|                            | Máximo              | 2                              | 0,85                            |
|                            | Mayor               | 1                              | 0,43                            |
| Total de instrucciones     |                     | 234                            | 100,00                          |

El ciclo de ejecución del algoritmo forward para el prototipo propuesto corresponde al cálculo de la probabilidad de 3 modelos HMM descritos en [6]. Cada HMM tiene 3 estados y una longitud de 4 símbolos como cadena de observación.

**Tabla 5-8: Resumen de la frecuencia de operación del MAP.**

| <b>Ciclo más extenso (instrucciones)</b> | <b>Periodo de muestreo de la cadena (s)</b> | <b>Frecuencia mínima de operación del MAP (Hz)</b> |
|--|---|--|
| 80                                       | 0,1   | 800  |

En la tabla 5-8 se detalla para una frecuencia de muestreo de 10 Hz por símbolo observado y un tamaño de ciclo de 80 instrucciones para el cálculo de la probabilidad, que la frecuencia mínima de operación del *MAP* debe de 800 Hz.

## Capítulo 6: Conclusiones y recomendaciones

### 6.1 Conclusiones

En éste trabajo se ha propuesto una implementación en hardware reconfigurable para un sistema de reconocimiento de patrones acústicos. Como punto de partida se utilizó el prototipo de software expuesto en un trabajo anterior [6]. El trabajo se ha dividido en tres grandes bloques: una etapa de adquisición y preproceso, una etapa de extracción de características y un último módulo de clasificación de dichas características.

El preproceso incluye una estructura de normalización en un circuito digital de un control automático de ganancia cuyo funcionamiento está regido por un sistema de promediación. Además, tiene un divisor binario implementado en un arreglo de memoria sólo lectura de 16 bits de ancho de bus y de  $2^{13}$  localizaciones de memoria y un multiplicador de 32 bits para la cual el saturador disminuye el ancho del bus hasta 16 bits con el valor está en rango de 32767 hasta -32768. Para la implementación del normalizador se requiere el 4% de 9312 LUT disponibles, un multiplicador de 18 bits por 18 bits y 19 flip-flops.

La etapa de extracción de características se puede separar en 2 bloques principales: el banco de filtros y los estimadores de energía. El banco de filtros está basado en una arquitectura segmentada en la cual el flujo de datos está multiplexado en el tiempo y en las estructuras aritméticas.

El banco de filtros de 8 subbandas de frecuencia tiene forma de una cascada de sistemas de segundo orden. La codificación interna tiene un ancho de bus de datos de 19 bits en aritmética en punto fijo con multiplicadores en CSD de 16 bits, que producen una desviación del 0,01% máxima respecto a los multiplicadores binarios estándar con una resolución en el orden  $\pm 2^{-15} \approx 3.051 \times 10^{-5}$ . Dicha implementación de filtros permite un desplazamiento de la ubicación de los polos menor de 0,0007 en el plano z. La estrategia de diseño del banco de filtros digitales segmentado permite una optimización hasta de aproximadamente 6,38

veces en términos del área de implementación respecto de la tendencia en cascada común. Además, el banco de filtros digitales segmentado con una frecuencia de muestreo de 44100Hz reduce el consumo de potencia dinámica en un 47,37% respecto al consumo de 950  $\mu$ W de la estructura en cascada a la misma frecuencia. Además una reducción de la frecuencia de muestreo a la mitad produce un decremento del 32% del consumo de potencia de índole dinámico en el banco de filtros digitales segmentado. Las respuesta del banco de filtros digitales segmentado definen anchos de banda con un error o desvío máximo del 12.24%.

La implementación de la etapa de extracción de características requiere del 23% de los 9312 LUT disponibles, el 13 % de los 9312 flips-flops disponibles.

Para la etapa de clasificación y toma de decisiones se diseñó una estructura para determinar el estado del evento del bosque clasificado como bosque normal, motosierra o disparo mediante la selección del HMM de mayor probabilidad para cadenas de observación dadas. Para dicha estructura se ha diseñado en este trabajo el procesador especializado (*MAP*) capaz de realizar operaciones aritméticas bajo el estándar de punto flotante de 32 bits con argumentos matriciales, tal como lo requiere el algoritmo hacia adelante.

El procesador *MAP* tiene una arquitectura tipo Harvard de ejecución monociclo con optimización de la unidad de cálculo en punto flotante de forma combinacional. El acceso de datos es realizado en un solo ciclo mediante direccionamiento directo o por registro. Los registros de almacenamiento de los cálculos temporales cuyo ancho de bus es de 32 bits requieren el 7 % del 9312 flips fops disponibles y el área dedicada para la implementación de los datos matriciales corresponde al 23% de los 9312 LUT utilizables en el FPGA empleado en el presente prototipo.

Respecto al diseño de la unidad de cálculo en punto flotante sólo se requiere el 12% de los 9312 LUT disponibles para el multiplicador y sólo el 8% de los LUT aprovechables para la suma en punto flotante de 32 bits. La desviación en la

ejecución del cálculo de la probabilidad para cada uno de los HMM está en el orden  $10^{-10}$  para la implementación de hardware (MAP) en contraste con la implementación por software.

La ejecución del algoritmo hacia adelante para evaluar tres HMM de tres estados cada uno, con 4 símbolos por cadena de observación requieren 234 instrucciones de las cuales el 50% corresponde a multiplicaciones en punto flotante. Además, la frecuencia mínima de operación del MAP debe ser al menos 800 Hz debido que se requieren 80 instrucciones máximo para el cálculo de la probabilidad por cada símbolo de la cadena, cuya observación ocurre cada 0,1s.

## **6.2 Recomendaciones**

Dentro de las recomendaciones para la optimización de las estructuras digitales que se involucran en el reconocimiento de patrones y conjuntamente a los factores de implementación que encuadran en esta tesis se encuentran la evaluación del efecto de la variación del ancho del bus de datos en términos de consumo de estructuras (área en FPGA) y potencia para las etapas de adquisición, normalización y filtrado digital.

Se recomienda evaluar el efecto de la reducción del número de potencias de la codificación CSD usada en la implementación de los multiplicadores de los filtros.

Finalmente se sugiere evaluar la utilización de estándares de punto flotante de menor ancho de bus que el IEE-754 (32 bits) con el fin de reducir el área de implementación en FPGA sin perjudicar notablemente el rendimiento en los procesos de reconocimiento de patrones.

## Capítulo 7: Bibliografía

[1] Rodriguez, Juan José. Ampliar reservas naturales protegidas sería negocio muy rentable. [En línea] La Nación, 14 de junio de 2009. [Citado el: 14 de octubre de 2009.] [http://www.nacion.com/ln\\_ee/2009/junio/14/aldea1996433.html](http://www.nacion.com/ln_ee/2009/junio/14/aldea1996433.html).

[2] Hernández, Néstor. Diseño de una red inalámbrica de telecomunicaciones para la protección ambiental en el bosque. Cartago : Propuesta de proyecto de investigación, 2004.

[3] Leiva, Arnoldo. Diseño e implementación de la etapa de detección de disparo de armas de la red inalámbrica de telecomunicaciones para la protección de las zonas protegidas. Cartago: Instituto Tecnológico de Costa Rica: Proyecto de graduación, 2005.

[4] Salas, Walter. Diseño e implementación de la etapa de detección de motosierras de la red inalámbrica de telecomunicaciones para la protección de las zonas protegidas. Cartago: Instituto Tecnológico de Costa Rica: Proyecto de graduación, 2005.

[5] Sáez, Maria Gabriela. Reconocimiento de patrones acústicos para la protección del ambiente utilizando wavelets y Modelos Ocultos de Markov. Cartago: Instituto Tecnológico de Costa Rica: Proyecto de graduación, 2006.

[6] Smith, Esteban. "Reconocimiento digital en línea de patrones acústicos para la protección del ambiente por medio de HMM." Proyecto de graduación. Cartago. 2008.

[7] Strembler, Ferrel G. Introducción a los sistemas de comunicación. México : Addison Wesley Iberoamericana, 1993.

[8] Hwang-Cherng Chow; I-Hsin Wang; , "High performance automatic gain control circuit using a S/H peak-detector for ASK receiver," Electronics, Circuits and Systems, 2002. 9th International Conference on , vol.2, no., pp. 429- 432 vol.2, 2002

[9] Ballow, Glen. Handbook for sound engineers. Indiana : Howard W. Sams & CO, 1987.

[10] Overley, J. "Energy distribution in music," Audio, IRE Transactions on , vol.4, no.5, pp. 120- 123, Sep 1956

[11] Alvarado Moya, Dr. -Ing. Pablo. Señales y Sistemas. Fundamentos Matemáticos. Cartago : Centro de desarrollo de material bibliográfico. 2008. ISBN 978-9968-514-06-4.

[12] Hamalainen, M.; , "Optimization of multirate crossover filters," Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on , vol., no., pp.63-66, 1999

[13] Abut, H.; Ergezinger, S.; , "Low-rate speech encoding using vector quantization and subband coding," Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. , vol.11, no., pp. 449- 452, Apr 1986

[14] P. P. Vaidyanathan, Multirate Systems and Filterbanks. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[15] Proakis, John G; Manolakis, Dimitris G. Tratamiento digital de la señal. Cuarta edición. Madrid : Prentice-Hall, 2007.

[16] Francis, Michael. Infinite Impulse Response Filter Structures in Xilinx FPGAs. [En línea] Xilinx, agosto de 2009. [Citado el: 25 de noviembre de 2009.] [http://www.xilinx.com/support/documentation/white\\_papers/wp330.pdf](http://www.xilinx.com/support/documentation/white_papers/wp330.pdf).

[17] Burrus, C. Sidney; McClellan, James H; Oppenheim, Alan V; Parks, Thomas W; Schafer, Ronald W; Schuessler, Hans W;. Ejercicios de tratamiento de la señal. Utilizando Matlab v.4. Madrid : Prentice-Hall, 1998.

[18] Ogata, Katsuhiko. Sistemas de Control en Tiempo Discreto. Segunda edición. México : Prentice Hall, 1996.

[19] Sheno, B.A. Introduction to digital signal processing and filter design. New Jersey : Wiley-Interscience, 2006.

[20] Meyer-Baese, Uwe. Digital Signal Processing with Field Programmable Gate Arrays. Berlin : Springer, 2007.

[21] Rainer, Alois. Circuit for CSD coding of a binary number represented in two's complement. [En línea] US patent N° 4623872, 18 de noviembre de 1986. [Citado el: 7 de diciembre de 2009.] <http://www.freepatentsonline.com/4623872.pdf>.

[22] J. O. Coleman and A. Yurdakul, "Fractions in the Canonical-Signed-Digit Number System," in Proc. 2001 Conf. on Information Sciences and Systems,(Johns Hopkins University), Mar. 2001.

- [23] Faúndez Zanuy, Marcos. Tratamiento digital de voz e imagen. Barcelona : Alfaomega, 2001.
- [24] Rabiner, Lawrence R. A tutorial on hidden Markov Models and Select applications in speech Recognition. . : Proceedings of the IEEE, vol. 77, NO. 2, February 1989.
- [25] Sung-Jung, Cho. Introduction to Hidden Markov Model and its application. Samsung Advanced Institute of Technology, 2005.
- [26] Poole, David. Álgebra lineal. Una introducción moderna. Segunda edición. México : Thomson Learning, 2007.
- [27] Schuster, Jeffrey; Gupta, Kshitij; Hoare, Raymond; Jones, Alex K;. Speech Silicon: An FPGA Architecture for Real-Time Hidden Markov-Model-Based Speech Recognition. EURASIP Journal on Embedded Systems Volume 2006. Issue 1 (January 2006)
- [28] S.J.Melnikoff, S.F.Quigley, and M.J.Russell. [En línea] "Implementing a Simple Continuous Speech Recognition System on an FPGA", presented at International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2002..
- [29] Vaidhyathan, A.; Lakshmiprabha, V.; , "Speech recognition of monosyllables using hidden Markov model in VHDL," TENCON 2004. 2004 IEEE Region 10 Conference , vol.A, no., pp. 76- 79 Vol. 1, 21-24 Nov. 2004
- [30] Derrien, S. y Quinton, P. "Parallelizing HMMER for Hardware Acceleration on FPGAs," Application -specific Systems, Architectures and Processors, 2007. ASAP. IEEE International Conf. on , vol. no. pp.10-17, 9-11 July 2007.doi: 10.1109/ASAP.2007.4429951.
- [31] S.J.Melnikoff, S.F.Quigley, and M.J.Russell, . "Implementing a Simple Continuous Speech Recognition System on an FPGA", presented at International Symposium on Field-Programmable custom Computing Machines (FCCM), 2002.
- [32] Xilinx. Spartan - 3E FPGA Family. Complete data sheet. [En línea] 18 de Abril de 2008. [Citado el: 6 de diciembre de 2009.] [http://www.xilinx.com/support/documentation/data\\_sheets/ds312.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf).
- [33] Kilts, Steve. Advance FPGA Design. Architecture, Implementation and optimization. New Jersey : John Wiley and Sons, 2007.
- [34] Kailasam, Umadevi. "High Level VHDL Modeling of a Low-Power ASIC for a Tour Guide." University of South Florida. Department of Computer Science and Engineering. March 29, 2004

- [35] John Lach, Vinu Vijay Kumar, "Application-Specific Product Generics," Computer Society. IEEE, pp. 64-74, August, 2009
- [36] Mano, Morris; Kime, Charles R. Logic and computer design fundamentals. New Jersey : Prentice Hall, 1997.
- [37] Patterson, David A; Hennessy, John L. Computer Organization and Design. The Hardware / Software interface. Third edition. San Francisco, CA. 2005.
- [38] Kolouch, J. "Combinational Divider in FPGA," Radioelektronika, 2007. 17th International Conference , vol., no., pp.1-4, 24-25 April 2007
- [39] Saini, Vijender, Singh, Balwinder y Devi, Rekha. "Area Optimization of FIR Filter and its Implementation on FPGA." International Journal of Recent Trends in Engineering, Vol 1, No. 4, May 2009
- [40] T. W. Parks, C. S. Burrus. Digital Filter Design. . : John Wiley & Sons, 1987. 1987.
- [41] Goldberg, David. What Every Computer Scientist Should Know About Floating-point Arithmetic. . : Association for Computing Machinery, 1991.
- [42] Detrey, Jérémie; De Dinechin, Florent. "FPLibrary." A VHDL Library of Parametrisable Floating-Point and LNS Operators for FPGA. [En línea] 0.94, 31 de Julio de 2006. [Citado el: 14 de abril de 2010.] <http://www.ens-lyon.fr/LIP/Arenaire/Ware/FPLibrary/>.
- [43] Franco, Sergio. Diseño con amplificadores operacionales y circuitos integrados analógicos. México : McGraw-Hill, 2002. ISBN 970-10-4595-5.
- [44] Coughlin, Robert y Driscoll, Frederick. Amplicadores operacionales y circuitos integrados lineales. Quinta edición. México : Prentice Hall, 1999.
- [45] Digilent. Spartan 3E Starter Board. [En línea] Digilent. [Citado el: 9 de diciembre de 2009.] <http://www.digilentinc.com/Products/Detail.cfm?Prod=S3EBOARD>.
- [46] Linear Technology, Corporation. Serial 12-Bit/14-Bit, 3Msps. Simultaneous Sampling ADCs with Shutdown. Datasheet. [En línea] 2004. [Citado el: 26 de diciembre de 2009.] <http://cdlinear.com/docs/Datasheet/14071fa.pdf>.
- [47] Technology, Linear. Dual Programmable Gain Amplifiers with Serial Digital Interface. Datasheet. [En línea] 2004. [Citado el: 26 de diciembre de 2009.] <http://www.datasheetcatalog.org/datasheet2/b/0dp8shwthzqay1eks5yx21777d7y.pdf>.

[48] Technology, Linear. Quad 16-Bit Rail-to-Rail DACs. Datasheet. [En línea] 2004. [Citado el: 26 de diciembre de 2009.]  
<http://cds.linear.com/docs/Datasheet/2604fd.pdf>.

[49] Xilinx. Spartan-3E Starter Kit. Board User Guide. [En línea] [Citado el: 21 de noviembre de 2009.]  
[http://www.digilentinc.com/Data/Products/S3EBOARD/S3EStarter\\_ug230.pdf](http://www.digilentinc.com/Data/Products/S3EBOARD/S3EStarter_ug230.pdf).

## Apéndice

### A. 1 Diseño de la etapa de filtro analógico

Se propone el circuito del filtro analógico mostrado en la figura A-1 basado en amplificadores operacionales. Dicha configuración proporciona el acondicionamiento de la señal proveniente del transductor o la fuente de sonido [46, 47]. Dicho circuito admite la posibilidad de cambiar la ganancia estática independientemente de las frecuencias de corte bajo o alto. Al final utiliza un condensador de acople que permite la conexión apropiada con la etapa de adquisición descrita en el apéndice D. 1.

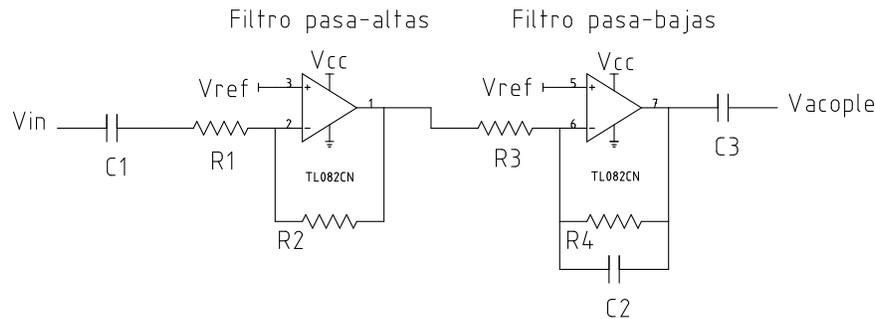


Figura A-1: Filtro analógico y acople de señal del transductor ( $V_{in}$ ) y la correspondiente señal condicionada ( $V_{acople}$ ).  $V_{ref}$  corresponde a  $V_{cc}/2$ .

La función de transferencia del filtro pasa banda correspondiente a la figura A-1 está dada por la relación

$$H_{BP}(s) = H_{HP}(s) \cdot H_{LP}(s) \quad (\text{A-1})$$

Dicho filtro se puede descomponer en dos secciones como el filtro pasa alto empleado para eliminar la componente CD, cuya función de transferencia está determinada por

$$H_{HP}(s) = \frac{sC_1R_2}{sR_1C_1 + 1} \quad (\text{A-2})$$

La frecuencia de corte bajo se puede calcular mediante

$$f_{LP} = \frac{1}{2\pi R_1 C_1} \quad (\text{A-3})$$

El filtro pasa bajo tiene la función de prevenir el aliasing. Dicho filtro tiene la función de transferencia

$$H_{LP}(s) = \left(\frac{R_4}{R_3}\right) \frac{1}{sR_4 C_2 + 1} \quad (\text{A-4})$$

La frecuencia de corte alto se puede calcular mediante

$$f_{HP} = \frac{1}{2\pi R_4 C_2} \quad (\text{A-5})$$

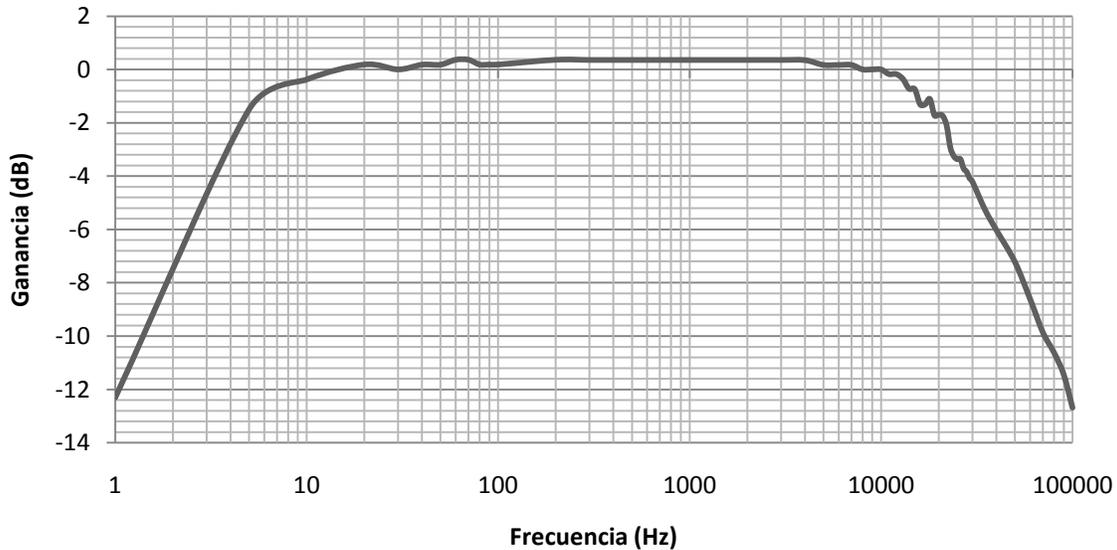
Basado en (A-3) y (A-5), se calcula las resistencias (tablas A-1) y capacitores (tablas A-2) del filtro analógico para las frecuencias de corte de 5 Hz y 22 kHz.

**Tabla A-1: Lista de resistencias que componen el filtro analógico de la figura A-1.**

| Componente | Valor ( $\Omega$ ) |
|------------|--------------------|
| R1         | 3300               |
| R2         | 3300               |
| R3         | 750                |
| R4         | 750                |

**Tabla A-2: Lista de capacitores que componen el filtro analógico de la figura A-1**

| Componente | Valor ( $\mu\text{F}$ ) |
|------------|-------------------------|
| C1         | 10                      |
| C2         | 0,001                   |
| C3         | 10                      |



**Figura A-2: Respuesta en frecuencia del filtro analógico antialiasing mostrado en la figura A-1**

En la figura A-2 se muestra la medición de la respuesta en frecuencia del filtro analógico. Nótese que la frecuencia de corte bajo y alto para el filtro analógico es aproximadamente 5 Hz y 22 kHz respectivamente.

## **B. 1 Plataforma utilizada**

La plataforma utilizada es la Spartan 3E Starter Board de Digilent [48] que contiene una FPGA de Xilinx XC3S500E [33] de capacidad de 500 mil compuertas para un equivalente de 10476 celdas lógicas. Está bajo el encapsulado FG320AGQ0545. Además, la plataforma Spartan 3E contiene el circuito integrado LTC1407, para la adquisición y codificación entre las señales analógicas y digitales [49].

Para el acople con las señales analógicas externas, se integró un amplificador de ganancia programable LTC6912 mostrado en [50]. Adicionalmente, para los procesos de pruebas, la unidad DAC utilizada es el circuito integrado LTC2614 documentado en [51]. Los anteriores módulos utilizan el protocolo SPI para la comunicación con la FPGA.

El arreglo CLB de la FPGA (CLB equivale a 4 Slices) es de 46 filas y 34 columnas, para un total de 1164 CLB o 4656 Slices. En memoria RAM distribuida alcanza 73kb y 360kb en bloques de RAM. Contiene 20 multiplicadores binarios dedicados [33].

La plataforma soporta configuraciones de 3,3V LVTTTL, 3,3V, 2,5V, 1,8V, 1,5V y 1,2V en LVCMOS.

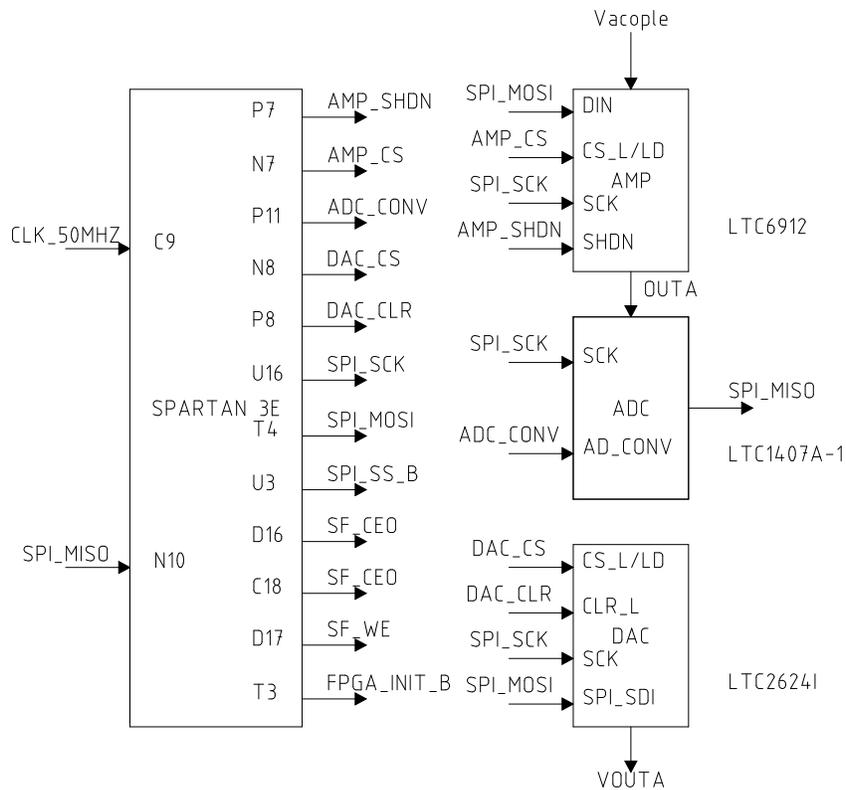
### **C. 1 Diseño del controlador SPI y comunicación entre subsistemas**

Los sistemas que comprenden la plataforma Spartan 3E dedicados para la adquisición tales como el amplificador (LTC6912 documentado en [50]), el ADC (LTC1407 descrito en [1]) y el DAC (LTC2614 documentado en [51]) empleado exclusivamente para la extracción de las señales en los procesos de prueba, utilizan el estándar de comunicaciones SPI como sistema de intercambio de información.

En la figura C-3 se muestra la interconexión entre los circuitos integrados y la FPGA. La misma utiliza una estructura maestro y 3 esclavos independientes. Las conexiones de pines del FPGA se detallan, además del flujo de las señales analógicas Vacople, OUTA y VOUTA (señal de entrada del módulo SPARTAN, la señal de entrada al ADC y la salida del DAC respectivamente). La señal de selección de dispositivo es única para cada componente,

- AMP: AMP\_CS.
- ADC: ADC\_CONV.
- DAC: DAC\_CS.

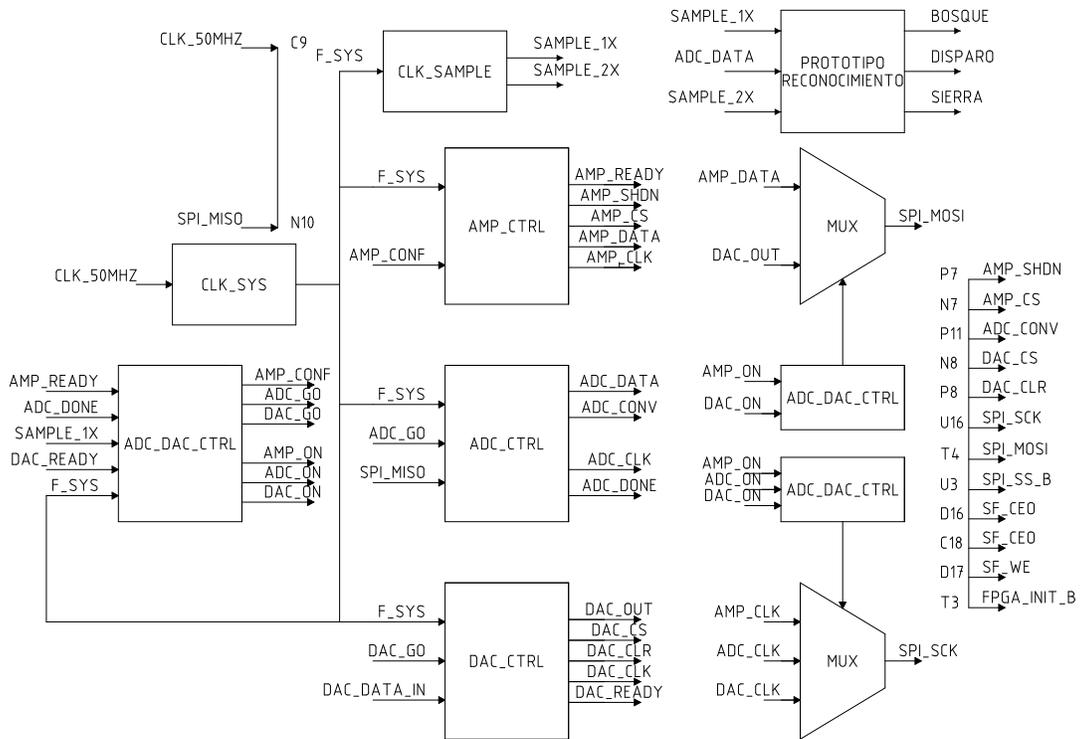
Las señales de transferencia de datos son compartidas, como SPI\_MOSI, SPI\_MISO y SPI\_SCK. Dado que las anteriores señales son la base para el intercambio de información, en este caso, maestro (FPGA) – esclavo y debida que cada esclavo funciona uno al tiempo, es necesaria la implementación de un controlador de las señales SPI. Dicho módulo está mostrado en la figura C-4.



**Figura C-3: Interconexión entre SPARTAN 3E y circuitos integrados de adquisición de señales que componen en SPARTAN 3E [48].**

Cada módulo de la figura C-4 está sincronizado por medio de la señal F\_SYS, sin embargo la plataforma SPARTAN sólo tiene tres conectores externos para las señales SPI (SPI\_MOSI, SPI\_MISO, SPI\_SCK) para la comunicación con los circuitos integrados de adquisición, por lo que requiere la multiplexación de dichas señales para la transferencia de información entre los subsistemas.

En la figura C-5 se detallan los diagramas temporales del funcionamiento de los controladores de los subsistemas. Se nota que las señales SCK\_SPI (AMP, ADC y DAC) y los controladores internos (AMP\_STATE, ADC\_STATE y DAC\_STATE) son activos dependiendo del estado del controlador principal o ADC\_DAC\_CTRL.

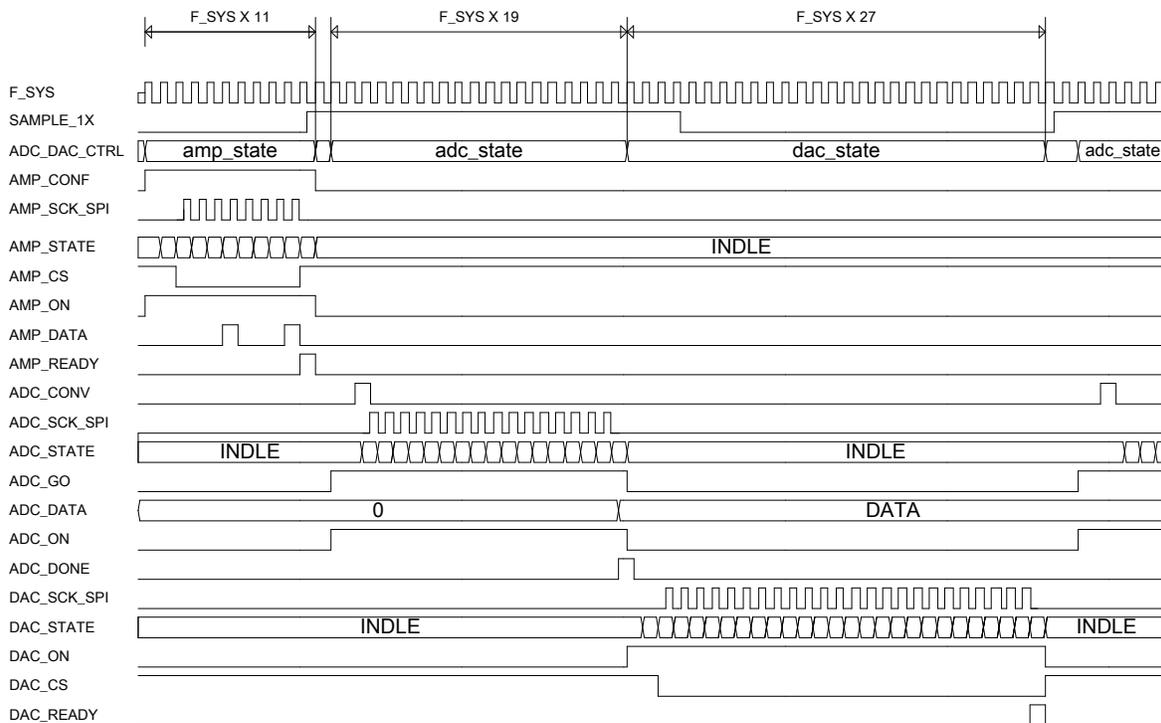


**Figura C-4: Interconexión entre controladores de SPI para los distintos circuitos integrados de adquisición y el prototipo de reconocimiento.**

En la tabla C-3 se resume la cantidad de periodos de la señal F\_SYS para el funcionamiento del sistema de adquisición mostrado en la figura 1-1. Es importante señalar que el subsistema DAC es implementado sólo para procedimiento de prueba y por lo tanto el prototipo general no lo incluye.

**Tabla C-3: Número de ciclos mínimo para el funcionamiento de los sistemas de la figura C-4.**

| Subsistema | Ciclos |
|------------|--------|
| AMP        | 11     |
| ADC        | 19     |
| DAC        | 27     |



**Figura C-5: Formas de onda de funcionamiento de los controladores de SPI y los circuitos integrados de adquisición para la figura C-4. Las especificaciones temporales y funcionales de los circuitos integrados mostrados en la figura C-3 se encuentran en [49,50,51].**

**Tabla C-4: Frecuencia de operación del sistema de la figura 1-1 para la frecuencia de muestreo especificada en la tabla D-6 según el modo de operación.**

| Modo de operación   | Número de Ciclos | Frecuencia (MHz) |
|---------------------|------------------|------------------|
| Adquisición sin DAC | 19               | 0,86             |
| Adquisición con DAC | 46               | 2,083            |

**Tabla C-5: Resumen de la utilización de recursos para el sistema de adquisición y los bloques que le componen de la figura C-4 según la plataforma de implementación mostrada en apéndice B. 1.**

| Bloque          | Componente | Utilización |
|-----------------|------------|-------------|
| CLK_SYS         | slices     | 3           |
|                 | Flip_flop  | 5           |
|                 | LUT        | 6           |
| CLK_SAMPLE      | slices     | 4           |
|                 | Flip_flop  | 6           |
|                 | LUT        | 8           |
| AMP_CTRL        | slices     | 5           |
|                 | Flip_flop  | 7           |
|                 | LUT        | 9           |
| ADC_CTRL        | slices     | 24          |
|                 | Flip_flop  | 38          |
|                 | LUT        | 12          |
| DAC_CTRL        | slices     | 19          |
|                 | Flip_flop  | 21          |
|                 | LUT        | 25          |
| ACQUISITION SYS | slices     | 65          |
|                 | Flip_flop  | 78          |
|                 | LUT        | 65          |

El sistema de adquisición propuesto en la figura C-4 y cuyo diagrama de señales en tiempo se muestra en la figura C-5, está diseñado para una frecuencia de muestreo arbitraria. Sin embargo para el sistema de la figura 1-1 y sus características resumidas en la tabla D-6, la frecuencia de la señal F\_SYS requerida es aproximadamente la especificada en la tabla C-5.

### **D. 1 Descripción de la etapa de adquisición**

La etapa de adquisición está basada en la operación de los circuitos integrados descritos en el apéndice C. 1 y mostrados en la figura C-3. Los mismos tienen un sistema de conversiones ADC/DAC gobernados por las expresiones:

$$ADC_{out} [ADC_{res} - 1:0] = \frac{G_{amp} (AMP_{in} - AMP_{ref})}{ADC_{rango}} \times 2^{ADC_{res}-1} \quad (D-6)$$

$$DAC_{out} (V) = \frac{DAC_{in} [DAC_{res} - 1: 0]}{2^{DAC_{res}}} \times (DAC_{ref} \pm 5\%) \quad (D-7)$$

que definen los rangos de tensión eléctrica y el equivalente binario analógico-digital y digital-analógico respectivamente.

**Tabla D-6: Frecuencia de muestreo del prototipo.**

| Característica                           | Valor |
|--|-------|
| Frecuencia de muestreo (Hz)              | 44100 |
| Frecuencia de muestreo obtenida, Fs (Hz) | 43400 |
| Desvío (%)                               | 1,59  |

En [52] se establecen los niveles de tensión permitidos por el bloque de adquisición, y se resumen en la figura D-6 Los parámetros son los siguientes: la tensión de referencia es  $AMP_{ref} = 1,65V$ , el rango de conversión es  $ADC_{rango} = 1,25V$ , la resolución de ADC es  $ADC_{res} = 1,65 V$  en complemento a 2, la magnitud de la ganancia del amplificador analógico  $G_{AMP} = 1$ , por lo que es equivalente según

$$DAC_{out} (V) = \frac{DAC_{in} [DAC_{res} - 1: 0]}{2^{DAC_{res}}} \times (DAC_{ref} \pm 5\%) \quad (D-7)$$

a los valores extremos de  $ADC_{out}$  como  $2^{ADC_{res}-1} - 1$  y  $-2^{ADC_{res}-1}$  entre los niveles de tensión  $AMP_{in}$  en el intervalo 2,9 V y 0,4 V respectivamente.

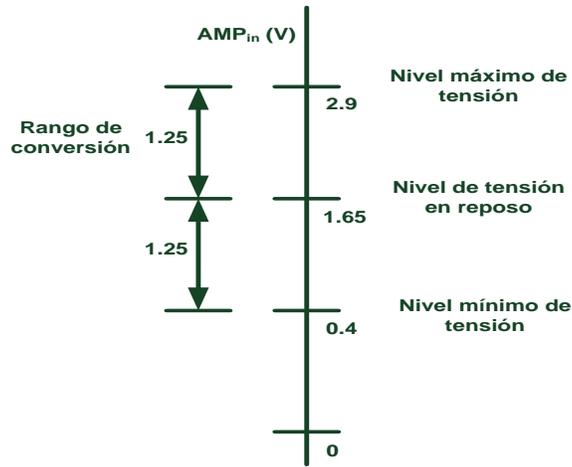


Figura D-6: Rango de tensiones permitidas por el bloque de adquisición [52].

El bloque de conversión digital – analógico está descrito por

$$DAC_{out} (V) = \left( \frac{DAC_{in} [n_{bits} - 1: 0]}{2^{n_{bits}} - 1} \right) \times (DAC_{ref} \pm 5\%) \quad (D-8)$$

bajo los siguientes parámetros: la resolución del DAC como  $DAC_{res} = 12 \text{ bits}$ , tensión de referencia  $DAC_{ref} = 3,3 \text{ V}$ . La sensibilidad de representación es de 0,805 mV/bit.

Debido que el DAC opera con representación binaria absoluta es necesario adecuar los datos para representar señales. Para ello se ubica el valor medio del rango  $2^{DAC_{res}-1}$  en el equivalente analógico  $AMP_{ref} = 1,65 \text{ V}$ . Se reacomoda (D-7) para obtener (D-8) que define la relación entre el valor binario de  $n_{bits} = 16$  y la tensión obtenida en los oscilogramas documentados en esta tesis. El valor de  $DAC_{in}$  corresponde a la señal DAC\_DATA\_IN del diagrama de la figura c-4.

Basado en el sistema presentado en el apéndice C. 1 y en [6] la frecuencia de muestreo teórica y experimental se resumen en la tabla D-6, con desvío entre las frecuencias cercano al 2%.

## E. 1 Análisis de desborde para distintas estructuras de implementación de filtros digitales

**Código E-1: Programa en MATLAB para el análisis de desborde para distintas estructuras de implementación del filtro digital.**

---

```
%Carga de los coeficientes del banco de filtros
load 'coeficientes_filtros'

poli_num_bajo = num_filtro_paso_bajo;
poli_num_alto = num_filtro_paso_alto;
poli_deno = den_filtros;

alto = 2; %filtro pasa alto columna 2
bajo = 1; %filtro pasa bajo columna 1

%número de estructuras
num_estructuras = 4;

DFI = 1;          %Forma directa I
DFII = 2;         %Forma directa II
SOS_DFI = 3;      %Forma directa I en sistema de segundo orden
SOS_DFII = 4;     %Forma directa II en sistema de segundo orden

b = [ poli_num_bajo ;poli_num_alto];

a = poli_deno;

[sos_bajo,g_bajo] = tf2sos(b(bajo,:),a);
[sos_alto,g_alto] = tf2sos(b(alto,:),a);

%FDI bajo          FDI alto, forma directa I
%FDII bajo         FDII alto forma directa II
%SOS FDI bajo      SOS FDI alto, Sistema de segundo orden DF I
%SOS FDII bajo     SOS FDII alto Sistema de segundo orden DF II

Hdf1 = [dfilt.df1(b(bajo,:),a), dfilt.df1(b(alto,:),a);...
        dfilt.df2(b(bajo,:),a), dfilt.df2(b(alto,:),a);...
        dfilt.df1sos(sos_bajo,g_bajo), dfilt.df1sos(sos_alto,g_alto);...
        dfilt.df2sos(sos_bajo,g_bajo), dfilt.df2sos(sos_alto,g_alto)];

for indice = 1:num_estructuras
    for tipo = 1:alto

        %Aritmética en punto fijo

        Hdf1(indice,tipo).Arithmetic = 'fixed'
        Hdf1(indice,tipo).AccumWordLength =
Hdf1(indice,tipo).ProductWordLength;
        Hdf1(indice,tipo).AccumMode = 'FullPrecision';
        Hdf1(indice,tipo).CoeffAutoScale = 0;

        %Tamaño parte fraccionaria
```

```

Hdf1(indice,tipo).DenFracLength = 15;

rand('state',5);
q = quantizer([10,9], 'RoundMode', 'round');
xq = randquant(q,1000,1);
x = fi(xq,true,10,9);

fipref('LoggingMode', 'on', 'DataTypeOverride', 'ScaledDoubles');
y = filter(Hdf1(indice,tipo),x);
fipref('LoggingMode', 'off', 'DataTypeOverride', 'ForceOff');

%reporte final

R(indice,tipo) = qreport(Hdf1(indice,tipo))
end
end

```

---

## F. 1 Procedimiento hacia adelante en código de alto nivel

**Código F-1: Programa en MATLAB para la ejecución del algoritmo hacia adelante para una cadena de observación arbitraria. Código adaptado de [6].**

---

```

function [P, alpha] = forwardprocedure(trans,obs,prior,data)
%FUNCION DE ADELANTO PARA EL ALGORITMO ADELANTE-ATRAS
%
%           [P, alpha] = forwardprocedure(trans,obs,prior,data)
%
%Calcula los valores de alpha para el algoritmo de adelanto y la
%probabilidad de salida del modelo de Markov luego de cada dato de
%entrada.
%Las entradas trans, obs, prior corresponden a la matriz de transición,
%observación y condición inicial respectivamente. La entrada data
%representa la cadena de símbolos de observación.

T = length(data);
N = size(trans,1);

alpha = zeros(N,T);
P = zeros(1,T);

%/ INICIALIZACIÓN
for i = 1:N
    alpha(i,1) = prior(i)*obs(i,data(1));
end

%Normalización
c = sum(alpha(:,1)); %Constante de normalización,
for i = 1:N
    alpha(i,1) = alpha(i,1)/c;
end

```

```

P(1)= c; %Toma la señal de probabilidad.

%/ INDUCCIÓN
for t = 2:T
    temp = zeros(1,N);
    %Cálculo de la variable alpha
    for j = 1:N
        for i = 1:N
            temp(i) = alpha(i,t-1)*trans(i,j);
        end
        alpha(j,t) = sum(temp)*obs(j,data(t));
    end
    %Normalización
    c = sum(alpha(:,t)); %Constante de normalización,
    for j = 1:N
        alpha(j,t) = alpha(j,t)/c;
    end

    P(t)= c; %Toma la señal de probabilidad.
end

```

---

## Glosario

|   |     |
|---|-----|
| <i>ADC</i> : Del inglés, <i>Analog Digital Converter</i> . Convertidor Analógico Digital .....                  | 5   |
| <i>AGC</i> : Del inglés, <i>Automatic Gain Control</i> . Control de ganancia automática .....                   | 5   |
| <i>CSD</i> : Del inglés, <i>Canonic Signed Digit</i> . Sistema Canónico de dígitos con signo..                  | 15  |
| <i>FPGA</i> : Del inglés, <i>Field Programmable Gate Array</i> .....  | iii |
| <i>HMM</i> : Del inglés, <i>Hidden Markov Models</i> . Modelos Ocultos de Markov .....                          | iii |
| <i>IIR</i> : Del inglés, <i>Infinite Impulse Response</i> . Respuesta Infinita al Impulso. ....                 | 8   |
| <i>MAP</i> : Del acrónimo inglés, <i>Matrix and Array Processor</i> . Procesador para arreglos y matrices. .... | 57  |
| <i>overflow</i> : Del inglés, desborde .....  | 14  |
| <i>QMF</i> : Del inglés, <i>quadrature mirror filter</i> . Filtro espejo en cuadratura.....                     | 8   |
| <i>ROM</i> : Del inglés <i>Read Only Memory</i> . Memoria de sólo lectura. ....                                 | 25  |
| <i>SPI</i> : Del inglés <i>Serial Peripheral Interface</i> . ....   | 80  |
| <i>wavelet</i> : ent. Pequeña onda .....  | 2   |