

UNIVERSIDAD DE TALCA.

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA CIVIL MECATRÓNICA

Diseño e implementación de un sintetizador de audio
en una plataforma Zynq-7000 SoC

Informe Memoria de Título.

Profesor Guía:

Dr. Roberto Ramírez Alegría.

Profesor del Ramo:

Daniel Díaz Besoain.

LUIS ALBERTO GONZALEZ HARVEY.

CURICÓ-CHILE.

2021

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular stamps and signatures. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' with a signature over it. The right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO' with a signature over it.

Curicó, 2023

Diseño e implementación de un sintetizador de audio en una plataforma Zynq-7000 SoC

Luis Alberto Gonzalez Harvey

Agosto de 2021.

Luis Alberto Gonzalez Harvey, 2021.

Resumen

En el presente informe se dará a conocer el control y la implementación de un Sintetizador de Sonido en una plataforma Zynq 7000. En donde se dará a conocer la información necesaria para abordar y construir un sintetizador de sonido. Bajo la consideración de tomar en cuenta los sintetizadores en el mercado, sus funciones, la simulación de los efectos y su implementación. Todo el presente desarrollo estará en el formato planteado de la asignatura de memoria de título.

Dentro del informe se abarcará la información sobre diferentes tipos de sintetizadores de sonido y los efectos que este puede realizar, a la vez del equipo disponible para la implementación de la memoria. Por otro lado, se buscará dar a conocer diferentes tópicos dentro del informe, tales como los objetivos, los alcances y limitaciones, la selección de sonidos para el sintetizador, junto con los efectos de sonido de este, considerando además mediante diagramas su funcionamiento y una breve explicación sobre la estructura de su programación. Es mediante el enfoque de la información mostrada durante la memoria, que se demostrará su diferencia frente a otros modelos del mercado y la metodología que se llevará a cabo, para llegar a concretar la memoria.

Cabe destacar que dentro del informe por su extenso trabajo en código se optó por el desarrollo de código mediante diagramas y pseudocódigos, para un mayor entendimiento de este.

Agradecimientos

En este tiempo cuando mi enfermedad me carcome y lo único que busco es poder descansar y morir tranquilo, agradezco a las siguientes personas por mantenerme con vida, frente a esta profunda angustia de mi vida.

Primeramente, agradezco a HM y BRS por darme las fuerzas cada día para seguir luchando en la carrera, puesto que sin esa meta casi imposible por cumplir no tendría motivación para seguir cada día en mi vida.

Continuando agradezco a mi familia, en especial a mi madre por costear el tratamiento que llevo, para mantenerme estable con mi enfermedad, puesto que si no estuviese medicado no estaría estable para poder seguir estudiando y trabajando.

Agradezco a la gran mayoría de los compañeros de mi generación y parte de mis compañeros en la universidad, que sabiendo de mi enfermedad me cuidaron y apoyaron. Esto fue de un gran goce en mi vida de poder compartir con ellos y poder disfrutar de la mejor etapa de mi vida, en comparación de la enseñanza básica y media, que solo me brindo grandes heridas en mí y constantes pesadillas, que me mantienen agotado hasta el día de hoy.

Es importante para mí destacar toda la ayuda y cariño que me brindo bienestar estudiantil, de ello la señorita Cristina Reyes, la señorita Giovanna Rojas, la señorita Cecilia y a la señorita Paulina Gonzalez. Las cuales me prestaron su cariño y atención en mis peores momentos, se preocuparon de mí y me apoyaron, de no ser por ellas no estaría aquí en este momento.

Continuando le agradezco a mis amigos de mi infancia de al frente de mi casa, en especial a Hernán Gómez, puesto que es un hermano para mí y aunque no sepa todas las enfermedades que padezco, le deseo lo mejor del mundo. A mis amigos que conocí durante mi periodo de la universidad, Esteban Marciel, Richard Duarte, Felipe Herrera, Neftalí Villar, Alexander Ruiz, Esteban Bravo y los miembros del laboratorio LAZiR., le agradezco de todo corazón todos los bellos momentos que compartí con ustedes.

Finalmente les doy las gracias a mis compañeros por el apodo que me dieron de “Padrino” es un apodo que me gusto más que mi nombre. El cual atesorare hasta el día de mi muerte.

Lo único que pido es poder seguir estable y poder lograr mis metas, antes que mi enfermedad me quite los pocos vestigios de salud y estabilidad que me quedan, por lo cual únicamente quiero descansar en paz.

TABLA DE CONTENIDOS.

1	Introducción.....	13
1.1	Introducción general.....	13
1.2	Estado del Arte	15
1.2.1	Sintetizadores de sonido portátiles.....	15
1.2.2	Sintetizadores base para el desarrollo de la memoria	21
1.2.3	Efectos de sonidos	22
1.2.4	Hardware para la implementación.....	25
1.2.5	Discusión.....	28
1.3	Objetivos	29
1.3.1	Objetivo general	29
1.3.2	Objetivos específicos.....	29
1.3.3	Alcances y limitaciones.....	29
1.4	Metodología	31
1.4.1	Estudio y programación en la FPGA Zynq7000	31
1.4.2	Realizar simulación, pruebas y modelaje de las ondas de sonido en Matlab	31
1.4.3	Traspasar los códigos de Matlab a la FPGA Zynq7000.....	32
1.4.4	Conexión de elementos periféricos a la FPGA	32
1.4.5	Realización de pruebas en la Zybo Z7	32
1.4.6	Organización de la memoria	33
1.5	Estudio comparativo de mercado	33
2	Especificaciones de la FPGA Zybo Z7	35
2.1	Introducción general.....	35
2.2	Especificaciones	35
2.3	Zynq APSoC Arquitectura	38
2.4	Reloj/Oscilador.....	39
2.5	Audio.....	40
2.6	I/O Básicas	42
2.7	Pmod Puertos.....	43
2.7.1	Pmod Estándar.....	44
2.7.2	Pmod de alta velocidad	44
3	Funciones de sonido del sintetizador de audio en una plataforma Zynq-7000	46

3.1	Introducción	46
3.2	Implementación del modo compositor.....	53
3.2.1	Selección de audio.....	56
3.3	Utilización de los efectos de sonido.....	58
3.3.1	Selección de audio de los efectos de sonido	61
3.3.2	Efecto 1 Repetir por 2	63
3.3.3	Efecto 2 Repetir por 4	65
3.3.4	Efecto 3 Inversión	67
3.3.5	Efecto 4 Fill in.....	68
3.3.6	Efecto 5 Bouncing ball.....	71
3.3.7	Efecto 6 Glich.....	74
3.3.8	Efecto 7 Distorsión.....	77
3.3.9	Efecto 8 Half rate	79
3.3.10	Efecto 9 Feedback.....	81
3.3.11	Efecto 10 Filtro pasa alto	85
3.3.12	Efecto 11 Filtro pasa bajo.....	89
3.3.13	Efecto 12 Filtro Distorsión.....	92
3.3.14	Efecto 13 Fm	95
3.3.15	Efecto 14 Barrido de baja velocidad	98
3.3.16	Efecto 15 Barrido de alta velocidad	102
3.3.17	Efecto 16 Barrido Distorsión.....	105
3.4	Modificación de parámetros del modo patrón y de los efectos de sonido	108
3.4.1	Control perilla A.....	110
3.4.2	Control perilla B.....	111
3.5	Visualización de los datos de las variables de los efectos de sonido y modo patrón	114
4	Implementación del sintetizador de sonido en una plataforma Zynq-7000	116
4.1	Introducción	116
4.2	Implementación del Hardware del sintetizador.....	116
4.2.1	Detector Táctil de 16/8 teclas detección IC TTP229-BSF.....	117
4.2.2	Encoder rotativo KY-040.....	120
4.2.3	Pantalla LCD.....	121
4.2.4	Uso del Hardware Mixto.....	123
4.3	Implementación del software para efectos de filtrados y barridos.....	127

4.4	Análisis de costos del sintetizador de sonido	128
5	Resultados experimentales.....	129
5.1	Introducción	129
5.2	Implementación de los efectos de sonido del sintetizador	129
5.2.1	Adquisición y post tratamiento de la señal.....	129
5.2.2	Implementación del audio original.....	131
5.2.3	Implementación del efecto 1 Repetir por 2	133
5.2.4	Implementación del efecto 2 Repetir por 4	134
5.2.5	Implementación del efecto 3 Inversión	135
5.2.6	Implementación del efecto 4 Fill in.....	136
5.2.7	Implementación del efecto 5 Bouncing ball.....	137
5.2.8	Implementación del efecto 6 Glich	139
5.2.9	Implementación del efecto 7 Distorsión.....	141
5.2.10	Implementación del efecto 8 Half rate	141
5.2.11	Implementación del efecto 9 Feedback.....	142
5.2.12	Implementación del efecto 10 Filtro pasa alto	144
5.2.13	Implementación del efecto 11 Filtro pasa bajo	145
5.2.14	Implementación del efecto 12 Filtro distorsión.....	147
6	Conclusiones.....	150
6.1	Introducción	150
6.2	Sumario	150
6.3	Conclusión.....	152
6.4	Trabajos futuros.....	153
7	Referencias.	155
8	Enlaces videos de la implementación.....	159
8.1	Modo patrón simple	159
8.2	Modo patrón compuesto.....	159
8.3	Modo patrón activo sus 16 tiempos	159
8.4	Efecto de sonidos	159

Lista de Figuras

Fig.1.1.Sintetizador JUNO DS61[9].	16
Fig.1.2.Sintetizador Yamaha MX61 v2[11].	17
Fig.1.3.Sintetizador Korg Kross 2-61[13].	18
Fig.1.4.Diagrama del Sintetizador PO-32 KO[14].	20
Fig.1.5.Definición musical de la distorsión armónica[17].	24
Fig.1.6.Diagrama de bloque de los filtros digitales FIR y IIR: (a) FIR y (b) IIR.	25
Fig.2.1.Enumeración de los elementos de la Zybo Z7-10[33].	37
Fig.2.2.Arquitectura general de la Zybo Z7-10[33].	39
Fig.2.3.Esquema del reloj de la Zybo Z7-10[33].	40
Fig.2.4.Esquemático de conexión de entradas y salidas básicas de la Zybo Z7-10[33].	43
Fig.3.1.Diagrama del setup del sintetizador de sonido.	48
Fig.3.2.Diagrama de Flujo modo compositor del sintetizador de audio.	49
Fig.3.3.Diagrama de Flujo modo de efectos de sonido.	50
Fig.3.4.Diagrama de Flujo modo modificación de parámetros del modo compositor.	51
Fig.3.5.Diagrama de Flujo modo modificación de parámetros del modo efecto de sonido.	52
Fig.3.6.Gráfica de cantidad de modo patrón de los modelos investigados.	53
Fig.3.7.Diagrama de Flujo del modo compositor del sintetizador de audio.	55
Fig.3.8.Acondicionamiento del audio para simulación e implementación.	58
Fig.3.9.Diagrama de Flujo del modo efecto de sonido del sintetizador de audio.	60
Fig.3.10.Gráfica vectorial original del sonido 1.	63
Fig.3.11.Diagrama de flujo del efecto 1.	64
Fig.3.12.Gráficas vectoriales del sonido 1 y la aplicación del efecto 1.	65
Fig.3.13.Diagrama de flujo del efecto 2.	66
Fig.3.14.Gráficas vectoriales del sonido 1 y aplicación del efecto 2.	66
Fig.3.15.Diagrama de flujo del efecto 3.	67
Fig.3.16.Gráficas vectoriales del sonido 1 y la aplicación del efecto 3.	68
Fig.3.17.Gráficas vectoriales del sonido 1 y la aplicación del efecto 4.	70
Fig.3.18.Gráfica vectorial del sonido 1 y sus respectivos rebotes, con la aplicación del efecto 5.	74
Fig.3.19.Gráficas vectoriales del sonido 1 y los vectores del Glich, con la aplicación del efecto 6.	77
Fig.3.20.Diagrama de flujo del efecto 7.	78
Fig.3.21.Gráfica vectorial del sonido 1 con la aplicación del efecto 7.	79
Fig.3.22.Diagrama de flujo del efecto 8.	80
Fig.3.23.Gráfica vectorial del sonido 1 con la aplicación del efecto 8.	81
Fig.3.24.Diagrama de bloque de la estrategia de Feedback negativo[50].	82
Fig.3.25.Diagrama polo zero del sistema del Feedback negativo.	83
Fig.3.26.Diagrama de flujo del efecto 9.	84
Fig.3.27.Gráfica vectorial del sonido 1 con la aplicación del efecto 9.	85
Fig.3.28.Respuesta de filtrador de un filtro pasa bajo, de una frecuencia de corte de 3000 Hz, en función del número de coeficientes ocupados[52].	86
Fig.3.29.Diagrama de flujo del efecto 10.	87
Fig.3.30.Gráfica vectorial del sonido 1 con la aplicación del efecto 10.	88

Fig.3.31.Transformada de Fourier de la Fig.3.30.....	89
Fig.3.32.Diagrama de flujo del efecto 11.....	90
Fig.3.33.Gráfica vectorial del sonido 1 con la aplicación del efecto 11.....	91
Fig.3.34.Transformada de Fourier de la Fig.3.33.....	92
Fig.3.35.Diagrama de flujo del efecto 12.....	93
Fig.3.36.Gráfica vectorial del sonido 1 con la aplicación del efecto 12.....	94
Fig.3.37.Transformada de Fourier de la Fig.3.36.....	95
Fig.3.38.Diagrama de flujo del efecto 13.....	97
Fig.3.39.Gráficas vectoriales de la señal portadora, moduladora y la aplicación del efecto 12 sobre el sonido 1 (Señal Fm).....	98
Fig.3.40.Diagrama de flujo del efecto 14.....	100
Fig.3.41.Gráfica vectorial del sonido 1 con la aplicación del efecto 14.....	101
Fig.3.42.Transformada de Fourier de la Fig.3.41.....	101
Fig.3.43.Diagrama de flujo del efecto 15.....	103
Fig.3.44.Gráfica vectorial del sonido 1 con la aplicación del efecto 15.....	104
Fig.3.45.Transformada de Fourier de la Fig.3.44.....	105
Fig.3.46.Diagrama de flujo del efecto 15.....	106
Fig.3.47.Gráfica vectorial del sonido 1 con la aplicación del efecto 16.....	107
Fig.3.48.Transformada de Fourier de la Fig.3.46.....	108
Fig.3.49.Diagrama de flujo del funcionamiento de la perilla A.....	111
Fig.3.50.Diagrama de flujo del funcionamiento de la perilla B.....	112
Fig.3.51.Distribución de bloques de información según línea de la pantalla LCD.....	115
Fig.4.1.Diagrama interno de conexiones del integrado TTP229-BSF.....	118
Fig.4.2.Tarjeta física teclado táctil 8/16 teclas TTP229-BSF.....	118
Fig.4.3.Diagrama del tren de pulsos de control del teclado táctil 8/16 teclas TTP229-BSF.....	119
Fig.4.4.Pines de conexión encoder rotativo KY-040[57].....	120
Fig.4.5.Señales A y B según su sentido de giro del encoder rotativo KY-040[57].....	121
Fig.4.6.Modelo físico pantalla LCD module 2004 más comunicador I ² C[59].....	122
Fig.4.7.Esquema del hardware mixto del sintetizador de audio ocupado en Vivado.....	123
Fig.4.8.Núcleo Zynq 7 Processing System.....	124
Fig.4.9.Núcleo Zynq 7 Processing System.....	124
Fig.4.10.Núcleo AXI Interconnect.....	125
Fig.4.11.Núcleo AXI GPIO.....	126
Fig.4.12.Núcleo Zybo audio ctrl.....	126
Fig.5.1.Diagrama de grabación y post tratamiento del audio.....	131
Fig. 5.2.Gráfica original vectorial del audio 1 vs su implementación procesada en la FPGA.....	132
Fig.5.3.Transformada de Fourier de la Fig.5.2.....	133
Fig.5.4.Gráfica de la simulación del efecto de repetir por 2 del audio 1 vs su implementación en la FPGA.....	134
Fig.5.5.Gráfica de la simulación del efecto de repetir por 2 del audio 1 vs su implementación en la FPGA.....	135
Fig.5.6.Gráfica de la simulación del efecto de inversión del audio 1 vs su implementación en la FPGA.....	136

Fig.5.7.Gráfica de la simulación del efecto de Fill in del audio 1 vs su implementación en la FPGA..... 137

Fig.5.8.Gráfica ideal y real de la simulación del efecto de Bouncing ball del audio 1 vs su implementación en la FPGA..... 138

Fig.5.9.Gráfica ideal y real de la simulación del efecto de Glich del audio 1 vs su implementación en la FPGA..... 140

Fig.5.10.Gráfica de la simulación del efecto de Distorsión del audio 1 vs su implementación en la FPGA. 141

Fig.5.11.Gráfica de la simulación del efecto de Half rate del audio 1 vs su implementación en la FPGA. 142

Fig.5.12.Gráfica de la simulación del efecto de Feedback del audio 1 vs su implementación en la FPGA. 143

Fig.5.13.Gráfica de la simulación del efecto de Filtro pasa alto del audio 1 vs su implementación en la FPGA..... 144

Fig.5.14.Transformada de Fourier de la Fig.5.13. 145

Fig.5.15. Gráfica de la simulación del efecto de Filtro pasa bajo del audio 1 vs su implementación en la FPGA..... 146

Fig.5.16.Transformada de Fourier de la Fig.5.15. 147

Fig.5.17.Gráfica de la simulación del efecto de Filtro distorsión del audio 1 vs su implementación en la FPGA..... 148

Fig.5.18.Transformada de Fourier de la Fig.5.17..... 149

Lista de Tablas

Tabla 1.1.Efectos de sonido de los sintetizadores PO-20 KO, PO-28 KO y PO-32 KO.	23
Tabla 1.2.Listado de precios de los sintetizadores de música y la FPGA.	34
Tabla 2.1. Listado elementos de la Zybo Z7-10[33].	38
Tabla 2.2. Señales de audio análogas[33].	40
Tabla 2.3. Señales digitales del audio para el códec SSM2603 en modo de esclavo[33].	42
Tabla 2.4. Tipo de Pmod y definición de sus pines de la Zybo Z7-10[33].	44
Tabla 3.1.Sonidos según su canal del modo compositor.	57
Tabla 3.2.Nombre de los efectos de sonido y su posición correspondiente.	61
Tabla 3.3.Sonidos según la posición del teclado táctil del modo de efectos de sonido.	62
Tabla 3.4.Polos y zeros resultante del sistema aplicando Feedback negativo.	82
Tabla 4.1.Ficha técnica básica del teclado táctil de 8/16 teclas.	119
Tabla 4.2.Ficha técnica Encoder rotativo KY-040[57].	120
Tabla 4.3. Ficha técnica Pantalla LCD module 2004[58].	122
Tabla 4.4.Elementos principales ocupados en la implementación del sintetizador de audio.	128

Lista de ecuaciones

(1.1) Ecuación de la distorsión armónica	24
(2.2) Intervalos de vacío del efecto 4	69
(3.3) Ecuación del largo del vector de audio del efecto 5	71
(3.4) Ecuación del largo del vector de audio del efecto 6	75
(3.5) Ecuación del primer sistema del efecto 9	82
(3.6) Ecuación del segundo sistema del efecto 9.....	82
(3.7) Calculo del Feedback negativo del efecto 9	82
(3.8) Ecuación de la moduladora del efecto 13	95
(3.9) Ecuación de la portador del efecto 13.....	95
(3.10) Ecuación de la FM del efecto 13.....	95
(4.11) Ecuación vectorial del efecto de filtrado	127
(4.12) Ecuación vectorial del efecto de barrido.....	128

Abreviaciones

PO KO	Pocket Operator
FPGA	Field Programmable Gate Arrays
IIR	Respuesta infinita al impulso
FIR	Respuesta finita al impulso
MP3	MPEG-2 Audio Layer III
WAV	Waveform Audio File Format
FM	Frecuencia modulada
PCM	Pulse-Code Modulation
MIDI	Musical Instrument Digital Interface
LCD	Pantalla de cristal líquido
DAC	Digital to Analogue Converter
BPM	Beats Per Minute
PLA	Matriz de Lógica Programable
GAL's	Matriz de Lógica Genérica
CPLD	Dispositivos Lógicos Programables Complejos
HDL	Lenguaje de descripción de hardware
SDK	kit de desarrollo de software
m4a	MPEG-4
I²C	Inter Integrated Circuits
PS	Sistema de procesamiento
PL	Lógica programable
APU	Unidad de Procesamiento de aplicaciones
Chx	Canal se sonido X
txt	Plain text
SCL	Reloj serial
SDO	Dato serial de salida
AXI	Advanced eXtensible Interface
GPIO	General Purpose Input/Output

1 Introducción

1.1 Introducción general

Una de las primeras formas de comunicarse entre las personas y dar a entender sus emociones fue a través de la música, tal fenómeno se ha dado a través todas las naciones y culturas, evolucionando hasta el día de hoy [1]. Dentro de esta evolución se tuvo dos ámbitos, la creación de nuevos estilos musicales y la diversidad de tipos de instrumentos, de lo cual no fue hasta llegar a la era de los circuitos electrónicos, que la cantidad de instrumentos y la modalidad de crear música fue creciendo y diversificándose [2].

No fue hasta 1933 en donde se empezaron a crear mezcladores de música [3], para controlar el sonido en las orquestas filarmónicas que dio pie a la evolución de los mezcladores de sonido hasta la actualidad. Tales hechos han permitido la creación de la música en el día de hoy, puesto que mediante a ellos se ha permitido agregar efectos de sonido, lo que dio pie a una variación de nuevos estilos de música y canciones. Sobre todo, en la actualidad donde ya es posible tener mezcladores de sonido de manera portátil [4].

Los sintetizadores de sonido son equipos electrónicos, creados en base al control de voltaje, por medio del aumento o descenso de este, modificando su frecuencia, timbre, intensidad y la forma de onda. Los sintetizadores han ido avanzando, llegando a emular y reproducir diferentes notas musicales, en diferentes instrumentos [4]. Los primeros instrumentos que emulaban los sintetizadores de sonido fue el piano, con el tiempo se han ido expandiendo llegando a instrumentos de cuerdas y percusiones, dichos instrumentos pueden ser controlados por medio del teclado del sintetizador. La principal ventaja de los sintetizadores, además de la capacidad de emular diferentes instrumentos musicales, es la capacidad de poder guardar en la memoria del sintetizador las composiciones creadas y aplicar efectos de sonido. Dentro de los efectos de sonido se encuentra el control de la altura, la intensidad, la frecuencia y el timbre del sonido[2]. De las principales compañías de sintetizadores se encuentran Yamaha, Korg, Roland, Nord y la que ha ganado recientemente popularidad Teenage Engineering[5].

Bajo la premisa anterior en el presente informe se detallará la investigación previa a la puesta en marcha de memoria de título, en la cual se buscará hacer un sintetizador de sonido, portátil que permita la rápida implementación de efectos de sonidos personalizados.

Para esto, se considerará como punto de partida la serie PO KO, de la empresa “Teenage engineering”, la cual es una empresa enfocada a la creación de sintetizadores de música.

Frente a los anteriormente planteado se procederá a emular las funciones básicas del PO KO, bajo la plataforma Zynq-7000, incluyendo mejoras como, el aumento de los canales a utilizar, un aumento en la cantidad de efectos de sonidos, aplicables a todos los sonidos disponibles para estos. En donde en primera instancia se simularán los efectos de sonidos en el software de Matlab[6], permitiendo su análisis y graficación. Posterior a ello los audios y efectos de sonidos serán implementados en el programa SDK de Xilinx[7].

La distribución del contenido de los capítulos son los siguientes:

En el capítulo I de introducción (Estado del arte), se darán a conocer todo el conocimiento previo a la implementación del sintetizador de sonido, abarcando sintetizadores en el mercado, las estrategias de filtrado, que es la FPGA, los efectos principales de los sintetizadores y los precios en el mercado de estos.

En el segundo capítulo se basará en las especificaciones de la FPGA Zybo Z7. En donde se encontrarán con todo el contenido necesario para la implementación del sintetizador de sonido. Tomando en consideración las características principales de la tarjeta y los elementos ocupados durante la memoria.

El tercer capítulo correspondiente a simulación, se tocarán los tópicos de la simulación de efectos de sonidos, los modos del sintetizador, la variación de parámetros y la visualización de estos. En el cual cada tema será explicado a profundidad sobre su funcionamiento y graficación de las señales.

En el cuarto capítulo correspondiente a la implementación, se explicarán los elementos físicos necesarios, externos a la tarjeta Zybo Z7-10 necesarios para la aplicación del sintetizador de sonido. La explicación de los bloques principales del hardware de la tarjeta Zybo Z7-10 en Vivado. Finalizando con un análisis de costos del sintetizador de sonido PO KO del cual se está emulando, frente al coste de los elementos ocupados.

En el quinto capítulo de resultados experimentales: Se realizará la implementación de los efectos de sonidos y su comparación con la simulación de capítulo III, analizando cada resultado obtenido.

Finalmente, en el sexto capítulo de conclusión, se realizará un sumario que abarcará los puntos más importantes de cada capítulo, nombrando su fuerte e importancia de cada uno de estos. Continuando se procederá a realizar las conclusiones del presente informe. Finalmente se dará a conocer los trabajos futuros, los cuales abarcaran los contenidos faltantes por implementar y las mejoras que puede tener el sintetizador de sonido.

1.2 Estado del Arte

En el siguiente capítulo se abarcará toda la información más relevante y comprimida, para el desarrollo de la memoria. Dentro de los tópicos se hablarán del diseño de sintetizadores del serial PO KO de la empresa “Teenage engineering”, su funcionamiento estándar. Prosiguiendo se hablarán los tres principales modelos del cual la memoria toma como ejemplo base para su desarrollo, junto con una breve descripción de cada uno, luego se hablará de sus principales efectos de sonido, donde serán base para el desarrollo de efectos de sonido de la memoria. Prosiguiendo se explicará que es una FPGA, ya que es el corazón de la memoria, en donde se aplicaran los temas hablados anteriormente. Finalmente se mencionarán que son los filtros FIR y IIR, ya que son por medio de estos que es posible la creación de diferentes efectos de sonido.

1.2.1 Sintetizadores de sonido portátiles

1.2.1.1 Sintetizador JUNO DS

Los sintetizadores JUNO DI de la empresa Roland, poseen una gran cantidad de sonidos de reproducción, tales como piano electrónicos y acústicos, órganos, entre otros sonidos, dentro de una gama de más de 1000 sonidos[8][9]. La Fig.1.1 se muestra sintetizador JUNO DS61.



Fig.1.1. Sintetizador JUNO DS61[9].

Posee un total ocho modos patrones el JUNO-DS, donde puede agregar ritmos instantáneos debajo de las partes del teclado o copiar datos de audio (MP3 / WAV) a la memoria USB. Todos los sonidos que incluye el sintetizador pueden ser modificados en la estación de trabajo, por medio el software de Cubase AI y ser cargados por medio de la memoria USB.

El enfoque principal del presente sintetizador de sonido Roland Juno DS, es proveer una solución integral para la composición y grabación de sonidos, utilizando una estación de trabajo de audio digital.

Los modelos del Juno DS se diferencia únicamente de la cantidad de teclas que poseen, de estos existen:

- DS61.
- DS76.
- DS88.

Por parte de los efectos de sonidos, suministrados por los manuales de la empresa Roland del sintetizador Roland Juno DS se encuentran los efectos de micrófono lo cuales son[9]:

- Reverb: Provoca una reverberación de la voz.
- Vocoder: Crea sonido de voz robotizada.
- Auto pitch: Corrige la entonación de la voz, dando un efecto más armónico.

1.2.1.2 Sintetizador Yamaha MX

El sintetizador Yamaha MX posee una amplia biblioteca de sonidos superando los 1000 sonidos del Motif XS. Además de poseer conectividad USB, con lo cual se puede cargar los sonidos creados, por medio de su software Cubase AI, para ordenador y FM essentials, para

control remoto desde el Smartphone[8][10]. La Fig.1.2 se presenta el sintetizador Yamaha MX61 v2.

La principal ocupación de la presente gama de sintetizadores de la gama Yamaha MX, es su compatibilidad con el ordenador, para el uso del Cubase AI, lo que lo convierte en un sintetizador de grabación completo. Además, de poder actuar de manera directa reproduciendo el material desde la estación de trabajo de audio digital.



Fig.1.2. Sintetizador Yamaha MX61 v2[11].

Dentro de las capacidades del sintetizador Yamaha MX61 se tiene la capacidad de funcionar con dos sonidos de manera simultáneas, dividiendo su teclado en dos secciones. Su espacio de memoria le permite guardar un total de 126 espacios de memoria para las composiciones creadas, con un máximo de 16 sonidos.

Los modelos del sintetizador Yamaha MX se diferencian únicamente de la cantidad de teclas que poseen, de estos existen:

- MX49.
- MX61.
- MX88.

Dentro de los efectos de sonido que se encuentran en los manuales de Yamaha, del sintetizador serie MX, solo se nombra la clase de efecto y la cantidad de variedad que posee, de estos efectos se presentan el siguiente listado:

- 9 tipos de reverberación (42 predefinidos).
- 9 tipos de reverberación (42 predefinidos).
- 17 tipos de coros (88 predefinidos).
- 48 tipos y 4 bloques de efectos de inserción (267 predefinidos).
- Ecualizador maestro (5 bandas).

1.2.1.3 Sintetizador Korg Kross2

El sintetizador de la serie Korg Kross2, basa su funcionamiento en una estación de trabajo compacta, la cual incluye una gran variedad de sonidos, con un total de 1280 sonidos. De los 1280 sonidos de la presente serie, posee un total de 120 voces (osciladoras) en modo simple, 60 voces (osciladoras) en modo doble, 1014 muestras de baterías y un secuenciador de 16 pistas[8][11]. La Fig.1.3 se muestra el sintetizador Korg Kross 2-61.



Fig.1.3. Sintetizador Korg Kross 2-61[12].

Dentro de las características técnicas se encuentran:

- La expansión de memoria PCM, la cual permite el almacenamiento de sonidos.
- Una memoria de 128 MB, para guardar las composiciones de sonido.
- Inclusión de un micrófono, para grabación de ritmos, grabación de voz o guitarra.
- Puerto de conexión Audio/MIDI USB.
- Teclado Pad Sampler, que permite crear patrones y muestras.

Los modelos del sintetizador Korg Kross2 se diferencian únicamente de la cantidad de teclas que poseen, de estos existen:

- Korg Kross2 61.

- KROSS 2-88.

Los efectos de sonido que ofrecen los sintetizadores Korg Kross2, se encuentran los siguientes[11]:

a) Enrutamiento de filtro:

- Simple.
- Serial.
- Paralelo.
- 24 dB.

b) Filtro de voz:

- Pasa bajo.
- Pasa alto.
- Pasa banda.
- Rechazo de banda.

1.2.1.4 Sintetizadores serial PO KO

La serie PO KO de la empresa “Teenage engineering”, son un grupo de sintetizadores portátiles de música, cada uno de ellos se enfoca en un área diferente, en donde la mayor diferencia entre ellos, es la actualización de cada modelo, los efectos disponibles y el cambio de sonido. En la Fig.1.4, se muestra el diagrama estándar de estos sintetizadores.

La tecnología que posee los PO KO es un procesador Silicon Labs EFM32 Gecko, dicha tarjeta permite el control de sonido, imagen y variación de parámetros. Dicha tarjeta además se encuentra con los periféricos de un conversor DAC Cirrus Logic, altavoces Knowles de alta calidad y una pantalla LCD, que permite mostrar la distintiva imagen del PO KO.

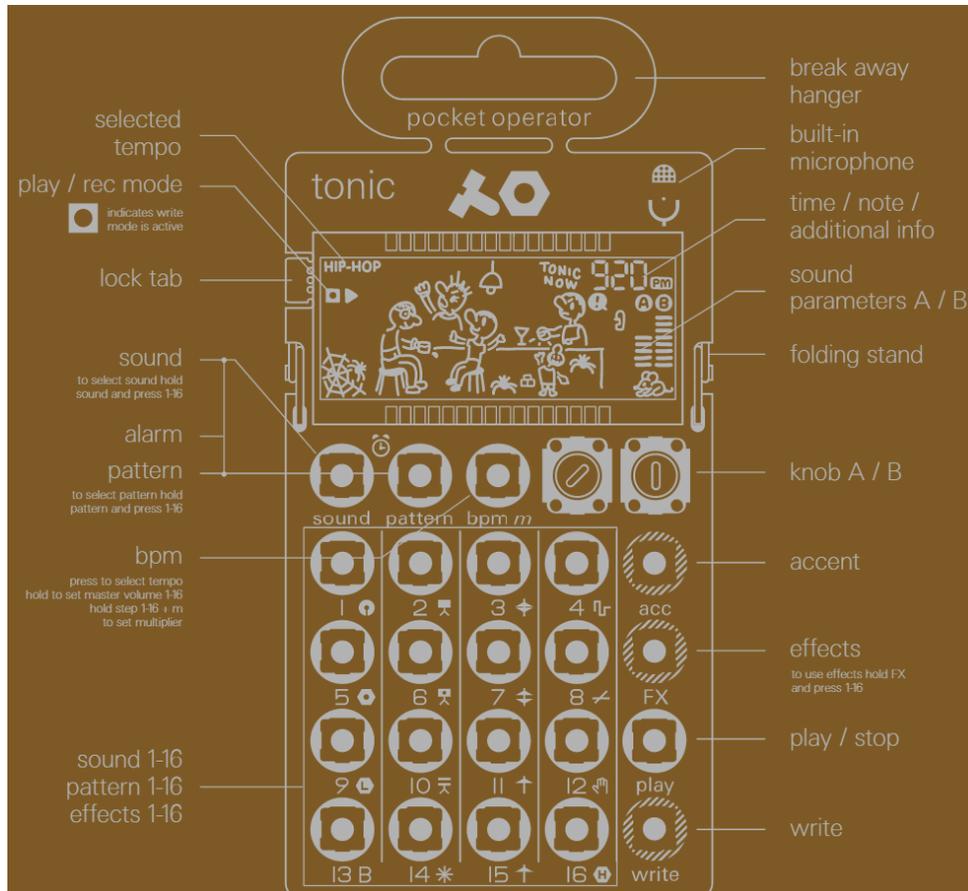


Fig.1.4. Diagrama del Sintetizador PO-32 KO[13].

Mediante lo visto en la Fig. 1.4 es posible destacar lo siguiente:

- Todos los modelos traen un teclado de 16 pulsadores, los cuales representan tres diferentes variantes dependiendo de la modalidad elegida, estas son los 16 sonidos que posee cada dispositivo, los efectos 16 efectos, los cuales pueden variar entre modelo y modelo, y por ultimo las posiciones de los tiempos a la hora de componer una pista. En cada caso se activa estas opciones presionando el botón Sound, FX y Pattern, respectivamente como se nombró al principio.
- Posee un botón de escritura Write, con el que se puede crear una pista de audio, aplicando las diferentes modalidades que posee el PO KO elegido. También con un botón de Play, con el cual se puede escuchar la pista de audio creada o detenerla.

- Posee el botón de BPM (BEATS PER MINUTE), con el cual se controla la velocidad de la pista de audio.
- Dentro de los elementos destacables de encuentran dos perillas, conocidas como A y B, dichas perillas tienen la particularidad de controlar diferentes variables de los sonidos y los patrones del audio, tales como el volumen y el tono.
- Finalmente, una de las características que destaca de los equipos PO KO, es la pantalla LCD, la cual permite la visualización de cada pista de audio, junto a sus variables, cabe destacar que cada modelo viene acompañado con un logotipo único.

1.2.2 Sintetizadores base para el desarrollo de la memoria

Como punto de partida y emulación para la memoria, se ocuparon los siguientes modelos de sintetizadores, los cuales son PO-20 KO, PO-28 KO y PO-32 KO, cabe destacar que cada uno tiene la capacidad de crear sus propias pistas de música. A continuación, se procederá a dar una breve descripción de cada uno de estos:

- a) PO-20 KO (programmable arcade synthesizer)[14]: Es un sintetizador enfocado a sonidos de videojuegos del estilo arcade, este modelo posee la particularidad de poseer un total de 16 acordes, cosa que otros modelos no poseen, además de la distinción de repetición de cuatro ciclos, con diferentes combinación de acordes, hay que recordar que los ciclos del serial PO KO se divide en 16 tiempos.
- b) PO-28 KO (programmable performance synthesizer)[15]: El presente sintetizador se enfoca a sonidos de 8 bits, en los cuales su efectos de sonido son en base a la modulación de onda de estos. Dentro de ellos se encuentra la peculiaridad de que la aplicación de las funciones de efectos son la utilización de filtros, modulaciones y diferentes tipos de barrido.
- c) PO-32 KO (programmable drum and percussion synthesizer)[13]: Es uno de los sintetizadores de sonido más completos, enfocado a efectos de sonidos, el cual combina

las habilidades de las versiones anteriores y además permite la capacidad de hacer grabaciones de sonidos, con la facilidad de modificarlas por medio de los efectos de sonido.

Para la implementación de sintetizador de audio en una plataforma Zynq-7000, se ocuparán el presente modelo de sintetizadores, como base para la realización de la memoria. Esto es debido a su claridad de información suministrada por la empresa “Teenage engineering”. Además del nivel de complejidad que poseen, ya que gran parte de su funcionamiento e implementación, se puede realizar en un microcontrolador o FPGA, que posea puertos de audio. En los cuales se pueda implementar el control de audio, el modo compositor, los modos de efectos de sonido, una indicación simple de pantalla y el control por medio de perillas.

1.2.3 Efectos de sonidos

1.2.3.1 Principales efectos de sonido

La principal función de un sintetizador de sonido es su cualidad de modificar el audio y con ello dar un nuevo eje a la pista de audio a crear. Siendo lo anterior punto clave, puesto que en el mercado se encuentra la disputa entre diferentes modelos y marcas, sobre cuáles son los efectos que más le acomoda al cliente y como se vende solo un modelo con un estilo de efectos específicos, para después vender otro totalmente diferente que sirva de complemento.

En base a los planteado en la sección 1.2.2, se procederá a nombrar los efectos de sonidos de estos tres modelos de sintetizadores anteriormente mencionados. Dicha enumeración de efectos se encuentra en la Tabla 1.1.

Tabla 1.1. Efectos de sonido de los sintetizadores PO-20 KO, PO-28 KO y PO-32 KO.

PO-20 KO	PO-28 KO	PO-32 KO
1. solo chords	1. distorted filter	1. half rate
2. solo chords + lead	2. distorted sweep	2. Distortion
3. solo drums	3. repeat 4	3. squash
4. solo drums + bass	4. repeat 2	4. echo fade
5. lowpass slow sweep	5. lowpass slow sweep	5. pitch lfo
6. lowpass fast sweep	6. lowpass fast sweep	6. ep sweep
7. hipass filter	7. hipass filter	7. mega morph
8. hipass sweep	8. hipass sweep	8. pitch bend up
9. fill-in	9. fill-in	9. punch
10. fill-in (fast)	10. fill-in (fast)	10. 6/8 quantize
11. retrigger	11. retrigger	11. beat-repeat
12. retrigger(fast)	12. retrigger(fast)	12. beat- repeat faster
13. glitch	13. glitch	13. fm
14. glitch (fast)	14. glitch (fast)	14. granular
15. blinds	15. blinds	15. reverse
16. rising arpeggio	16. noise effect	16. bouncing ball

En base a los efectos de sonido de la Tabla 1.1, se procederá a explicar los 7 más importantes, según el punto de vista práctico.

- a) Lowpass slow sweep: Aplicación de un barrido pasa bajo de baja velocidad, a la pista de audio[16].
- b) Lowpass fast sweep: Aplicación de un barrido pasa bajo de alta velocidad, a la pista de audio[16].
- c) Hipass filter: Aplicación de un filtro pasa altos para la señal del audio[16].
- d) Hipass sweep: Aplicación de un barrido pasa altos para la señal de audio[16].
- e) Distorted filter: Aplica un filtro, el cual permite distorsionar el audio, por medio de la aplicación de introducir nuevas frecuencias, estas son no linealidades sin memoria y por medio de ajuste de volumen[16].
- f) Distorted sweep: Aplica un barrido mediante la distorsión del audio, por medio del ajuste del volumen, de manera no lineal[16].
- g) Distortion: Efecto de añadir contenido armónico a una señal sinusoidal. De la cual funciona con las entradas de ondas sinusoidales por debajo del umbral de distorsión, las que permanecen en ondas sinusoidales activadas a la salida. Las entradas de onda sinusoidal con su amplitud, cada vez tiene un mayor desarrollo más severas. La entrada comienza a parecerse a una onda cuadrada recortando severamente una onda sinusoidal,

hasta que esta se asemeje a una onda cuadrada, la cual tiene el efecto de agregar contenido armónico a la señal de onda sinusoidal y de acuerdo a (1.1) con los armónicos presentados en la Tabla de la Fig. 1.5[16].

$$Y(t) = \frac{4A_{peak}}{\pi} \sum_{n=1}^{\infty} \sin[2\pi(2n - 1)ft] \tag{1.1}$$

Partial	Harmonic	Frequency (Hz)	Musical Interval	Error (cents)*
0	1	110		
1	2	220	1 octave exactly	0.00
2	3	330	1 octave + P5	1.96
3	4	440	2 octaves exactly	0.00
4	5	550	2 octaves + MAJ 3	-13.69
5	6	660	2 octaves + P5	1.96
6	7	770	2 octaves + min 7	-31.17
7	8	880	3 octaves exactly	0.00
8	9	990	3 octaves + MAJ 2	3.91
9	10	1,100	3 octaves + MAJ 3	-13.69
10	11	1,210	3 octaves + dim 5	-48.68
11	12	1,320	3 octaves + P5	1.96
12	13	1,430	3 octaves + min 6	40.53
13	14	1,540	3 octaves + min 7	-31.17
14	15	1,650	3 octaves + MAJ 7	-11.73
15	16	1,760	4 octaves exactly	0.00

*The frequency of the harmonic is actually the listed number of cents above or below the musical interval indicated.

Fig.1.5. Definición musical de la distorsión armónica[16].

1.2.3.2 Filtros FIR y IIR

La importancia de la aplicación de los filtros, se encuentra a la hora creación de efectos de sonidos, puesto que, al aplicar algunos efectos, a la onda de sonido, pasa por un proceso de filtrado. Esto último provoca la creación de una onda nueva en base a la aplicación del filtro selección. Dentro de la topología de filtros se encuentran los filtros de FIR y IIR, que dependiendo su uso puede provocar efectos como el eco, la reverberación y el aumento de ruido, entre otras funciones.

La metodología de los filtros FIR y IIR, es que son unos filtros del carácter digital, en donde se diferencia uno del otro a la hora de realizar su retroalimentación, esto es de gran relevancia, puesto que para algunos efectos de sonidos es necesario aplicar esta estrategia de filtros. Dicho proceso se ve reflejada en la Fig. 1.6.

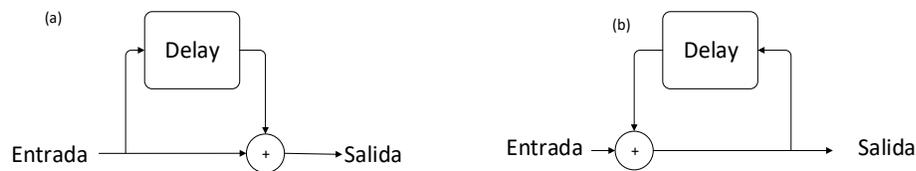


Fig.1.6.Diagrama de bloque de los filtros digitales FIR y IIR: (a) FIR y (b) IIR.

- a) Se genera un retardo de señal de entrada y se le vuelve a inyectar en su salida, esto puede variar en función de los periodos de muestreo a trabajar. Los filtros digitales basados en este funcionamiento se dice que son de respuesta impulsional finita o FIR (Finite Impulse Response).
- b) Se genera un retardo de la señal de salida, dicha señal de salida se vuelve a ingresar en la entrada. Los filtros digitales basados en este funcionamiento poseen el nombre de respuesta impulsional infinita o IIR (Infinite Impulse Response). También se les denomina filtros recursivos o con feedback.

La principal razón, para el desarrollo de la presente memoria, el saber y conocer el efecto de los filtros IIR y FIR, es la aplicación de estos, para todo efecto de sonido necesario, para crear efectos de filtrados y de barrido. Puesto que para el efecto de filtrado se ocupará principalmente la estrategia de filtrado FIR. Mientras que para el barrido se ocuparan la estrategia de filtrado combinada de filtro FIR e IIR. La forma de cómo se aplicará estos filtros será mayormente fundamentada en el capítulo 4.3 de estrategia de control del filtrado y barrido.

1.2.4 Hardware para la implementación

1.2.4.1 Field Programmable Gate Array (FPGA)

Una FPGA es una matriz de puertas programables diseñada para ser programada por los usuarios, en su interior posee básicamente una gran cantidad de bloques lógicos, los cuales se encuentran interconectados por el programador y que pueden ser modificados a disposición mediante un lenguaje dedicado, ya sea VHDL o Verilog [17].

A comienzo de los años 70, la empresa Texas Instruments diseño una máscara programable que luego se le otorgo el termino PLA (Matriz de Lógica Programable). Dicha

marcara estaba conformada principalmente por arreglos de compuertas AND y OR programables, las cuales dichas compuertas se podían conectar entre sí. Permitiendo la implementación suma de productos de ecuaciones lógicas y suma de productos[17].

Los primeros dispositivos de lógica programable de mayor simpleza fueron los PAL (Matriz de Lógica Programable), estos fueron creados en el año 1978 por la empresa MMI. La mayor diferencia con los PLA consiste en un arreglo de compuertas AND programables a continuación de un arreglo de compuertas OR fijo. Permitiendo la suma de ecuaciones lógicas de productos, con un número de variables definido y con el cual se permite una retroalimentación sus salidas[17].

Una distinción innovaría a los primeros dispositivos lógico programables PLA y PAL fueron las GAL's (Matriz de Lógica Genérica), el cual fue diseñado por Lattice Semiconductor en el año 1984. Fundamentalmente, estos dispositivos conservan las mismas propiedades lógicas que los PAL, pero a diferencias de versiones anteriores, estos diseños realizados pueden ser borrados y reprogramados, ya que, poseen una celda CMOS eléctricamente borrable, a diferencia de los PAL que contienen un fusible. Siendo esta una de sus mayores ventajas frente a los otros dispositivos (PLA y PAL), ya que, la presentación de fallos en el diseño puede ser corregidos mediante la reprogramación, permitiendo que se implementen ecuaciones lógicas más complejas[17].

Tiempo después, se diseñaron los CPLD (Dispositivos Lógicos Programables Complejos) ampliando el concepto de los dispositivos PLA, PAL y GAL, a un grado de mayor nivel de integración, esto es debido a que, un CPLD está conformado por una mayor cantidad de compuertas lógicas, a la que se encuentran unidas mediante una interconexión programable, siendo equivalente a gran cantidad de PAL's interconectados en un solo chip. Lo que permite lograr aumentar la fiabilidad y la eficacia de los diseños, a raíz de la disminución de espacio y la reducción del costo. Por último, en el año 1984, los ingenieros Ross Freeman y Bernard Vondershmitt diseñaron la primera FPGA siendo los cofundadores de Xilinx[17].

El desarrollo de la FPGA, a través del tiempo se divide en tres diferentes eras. Desde sus comienzos de la era de la invención (1984-1991), posterior a ello la era de la expansión

(1992-1999), finalizando con la era más reciente, la cual es la era de la acumulación (2000-2007)[18], [19].

En los comienzos del desarrollo de la FPGA, se da la era de la invención (1984-1991). En donde se destaca su primer modelo la Xilinx XC2064, la cual posee un total de 64 bloques lógicos complejos, en una cuadratura de 8x8, una frecuencia de 18 MHz, con un total de 58 pines de entrada y salidas y con una cantidad 1200 puertas. Para su principal propagación de ventas se consideró un buen margen de costos, lo cual fue indispensable para su éxito en ventas[18], [19].

Dentro del desarrollo de la memoria, se ocuparon memorias reprogramables y con una memoria PROM externa, como medida de protección de datos, cuando se apagará la FPGA. Para la estructura interna se ocupó un sistema único de antifusibles, desarrollado por Actel. Lo que logro un ahorro del área de los antifusibles sobre las celdas de la memoria. La FPGA, con mayor capacidad de la presente era fue la Actel 1280. Es mediante la anterior tecnología, fue que se dio un enfoque de eliminar el tamaño del cableado y optar, por conexiones cortas entre bloques adyacentes, las cuales se podían unir mediante las necesidades de la programación, conformando así rutas de enrutamientos largas, generando una mayor velocidad de comunicación[18], [19].

La era que continuo, después de la era de la invención fue la de la expansión (1992-1999). El presente nombre fue otorgado, por el hecho de la aplicación de la ley de Moore, la cual indica que se duplica la cantidad de transistores ocupados cada dos años, lo que aumenta el procesamiento de la FPGA de modelo a modelo. Con el comienzo de la presente era se crea el lenguaje de descripción de hardware (HDL), el cual es ocupado con el propósito de designar lugar y ruta[18], [19].

Con la expansión de la cantidad de transistores, provoco un aumento de dispositivos de la FPGA y a la vez una disminución de costos. En 1992, se crea la FPGA insignia de la presente era la Xilinx XC4010, la cual posee un total de 10000 puertas. Posterior a ello en 1999, llegaría la Virtex XCV1000, con un total de un millón de puertas. Frente al gran incremento de compuertas, solo la tecnología antifusible era viable para la operación de las FPGAs. Para aliviar el proceso de síntesis se ocupó la arquitectura basada en tabla de

búsqueda LUT, lo que permitió una libertad de arquitectura y dio un acceso casi completo a la tecnología de proceso, de la FPGA[18], [19].

La tercera y última era hasta la fecha, es la era de la acumulación (2000-2007). La cual da comienzo con la explotación de páginas de ventas por internet, lo que provoco la necesidad de dar un menor costo. Ante dicha masificación de venta, se tuvo que dar una elección objetiva del tipo de usuario, para ello se crearon FPGA de gama baja y alta. Para la gama baja, se dio prioridad a la eficiencia y producción de familias de bajo costo. Dentro de estas familias se encuentran la serie Spartan de Xilinx, Cyclone de Altera y EC/ECP de Lattice[18], [19].

Para la gama alta se tomó la decisión que el cliente decidiera llenar las funciones de la FPGA, según su necesidad, mediante la adición de bibliotecas de soft logic, para las funciones más importantes. Las funciones lógicas más destacables fueron de los microprocesadores de Xilinx Microblaze y Altera Nios, controladores de memoria y una vasta ocupación de protocolos de comunicación[18], [19].

Ante la presión de disminuir los costos y aumentar la potencia de estos, se modificó la estrategia de arquitectura mediante la aplicación de lógica programable. Po e medio de la agrupación de diferentes bloques lógico específicos, en dichos bloques se contenía una gran memoria, microprocesadores, multiplicadores, entradas y salidas flexibles y transceptores sincrónicos. Esto dio como efecto una reducción de la sobrecarga en el área de programación, rendimiento, potencia y esfuerzo de diseño[18], [19].

1.2.5 Discusión

Dentro de lo analizado en las secciones anteriores, se ha tomado la decisión de tomar como punto de partida el funcionamiento de los sintetizadores PO KO, puesto que son los sintetizadores que entregan un menor precio, el cual se verá en la sección 1.5, posee los funcionamientos básicos de un sintetizador de sonido, tal como se explica de las secciones 1.2.1.4 y la 1.2.2 y son los que más información provee sobre funcionamiento, a diferencia de los sintetizadores de las secciones 1.2.1.1 a la 1.2.1.3. Todas estas funciones del sintetizador de sonido, tales como control de audio, control de periféricos, ya sea pulsadores, perillas y pantalla LCD, pueden ser controladas por medio de la tarjeta Zybo Z7-10, he

implementada en la plataforma de Zynq-7000. Puesto que posee los elementos necesarios para la implementación de un sintetizador de sonido, tales elementos se explicarán en el capítulo 2.

Todo lo concerniente a efectos de sonidos y la composición de patrones de sonidos, pueden ser implementados por medio lenguaje C en el software de SDK de Vivado, por lo cual es posible implementar los filtros vistos en la sección 1.2.3.2, para los efectos de sonido. Además de los algoritmos para controlar las diferentes variables de sonidos, tales como el volumen, los BPM y el control de los canales, por medio de la implementación de un teclado y perillas como periféricos.

1.3 Objetivos

1.3.1 Objetivo general

Implementar un sintetizador de sonido programable, mediante un SoC Zynq7000.

1.3.2 Objetivos específicos

Los objetivos específicos son los siguientes:

1. Implementar algoritmos en la FPGA.
2. Implementar la modulación y efectos de onda, en el programa de Matlab.
3. Realizar la creación de los diferentes tipos de efecto de sonido, en el programa de Matlab.
4. Traspasar las pruebas y códigos realizados en Matlab a la FPGA.
5. Implementar un prototipo de teclado de pulsadores y perillas.
6. Realizar puesta en marcha y pruebas en la FPGA, donde su funcionamiento debe operar de manera similar o mejor a los equipos de la serie PO KO de la empresa “Teenage engineering”.
7. Implementar sonidos y efectos de sonidos, en la plataforma Zynq-7000.

1.3.3 Alcances y limitaciones

Los alcances con los cuales se busca cumplir, en el presente proyecto de memoria son los siguientes:

- a) Los efectos que serán implementados corresponden a los basados en filtros, modulación de onda, estilos de barrido e inclusión de componente armónicos. Tal como se mostró en la sección 1.2.2, en donde se les da prioridad a los efectos del PO-20 KO, PO-28 KO y PO-32 KO.
- b) Dentro de la capacidad de efectos se buscará cumplir con la base de 16 efectos de sonido, tal como se muestra en cada modelo de los sintetizadores PO KO y desde esa cantidad base se buscará agregar más efectos, siempre que se encuentre dentro del tiempo propuesto por la memoria de título.
- c) En base a la operatividad de los sintetizadores portátiles, se trabajará en base a 16 tiempos, puesto que es un compás de tiempo manejable y permitirá la óptima función de efectos de sonidos, sonidos y patrones.
- d) Dentro de la capacidad de sonidos con los que se integrará a FPGA, serán un total de 16 sonidos como punto base, en dos canales, puesto que se buscará aumentar su capacidad de almacenamiento a un número mayor.
- e) En el proyecto de memoria la FPGA será capaz de reemplazar el paquete de 16 sonidos que posee por otro de igual tamaño, dando a diferenciar por el tema que los sintetizadores portátiles carecen de la capacidad de ingresar nuevos sonidos y solo quedan con el paquete prefijado.

Por el caso contrario las limitaciones del presente trabajo son los siguientes:

- a) El proyecto de memoria no contemplara los efectos de modificación de voz, cosa que es una de las peculiaridades de los sintetizadores de hoy en día y está presente desde los modelos del PO-32 KO a la actualidad.
- b) Dentro de la sección 1.2.1.4 se nombró que una de las diferencias de los modelos PO KO es su pantalla LCD, la cual marca una figura distintiva. Dicho lo anterior la presente memoria no contempla como prioridad el diseño de la pantalla LCD, por ende, únicamente servirá para indicar parámetros.
- c) Dentro de los efectos del PO-32 KO hasta los modelos más modernos, de la misma serie, se encuentra la utilización de grabaciones de sonido. Por lo anterior la memoria solo contemplara las grabaciones de sonido, si es que la FPGA a ocupar posea los

equipamientos necesarios para la grabación en vivo, de no ser así solo se contemplara las grabaciones por medio separado, como sonidos.

- d) Por tema de disponibilidad de la pandemia y proceso de tiempo de la memoria, no se logrará realizar la impresión en 3D de la carcasa del sintetizador y tampoco realizar un esquemático, que englobe todos los periféricos conectados a la FPGA.

1.4 Metodología

Para la implementación y puesta en marcha del Sintetizador de Sonido de precio accesible, es necesario seguir una serie de pasos con el objetivo de concretar la memoria y que su resultado sea el esperado. Dentro de los pasos a seguir para el cumplimiento de la memoria es necesario:

1.4.1 Estudio y programación en la FPGA Zynq7000

Dentro de las principales tareas para la realización de la memoria es aprender a programar en la FPGA, puesto que mediante su programación podrá ser posible la interacción para que sintetice el sonido y que cumpla con sus objetivos.

Uno de los aprendizajes a manejar es la elección del lenguaje a ocupar, ya se VHDL o Verilog, mediante ello se deberá aprender como cargar archivos de audio, como modificar la forma de onda de dichos archivos, la habilitación de modificación de parámetros de datos en tiempo real, la asignación de entradas y salida, con lo que se llevará el control de datos entre otros conocimientos necesarios para la puesta en marcha.

1.4.2 Realizar simulación, pruebas y modelaje de las ondas de sonido en Matlab

Para la creación de los efectos de sonido es necesario realizar primero las simulaciones en el software de Matlab, en dicho programa mediante el análisis de punto fijo, además de la inclusión de cada estrategia de control de la onda del audio, se podrá realizar la modificación de la onda. Mediante el programa de Matlab se permitirá simular y comprobar cada efecto, incluido la modificación de parámetros tales como el volumen, el timbre y la velocidad de la onda.

A través de las pruebas de simulación en dicho programa, se podrá tener los resultados de cada efecto ante diferentes paquetes de sonidos, esto servirá como evidencia y respaldo a la hora de traspasar el código a la FPGA Zynq7000.

1.4.3 Traspasar los códigos de Matlab a la FPGA Zynq7000

Una vez terminadas la programación y pruebas en el software de Matlab, se procederá a traspasar el código y las bibliotecas de ser necesario al lenguaje de la FPGA Zynq7000. En dicho punto se aplicarán los conocimientos aprendidos en el punto 1.4.1 y el trabajo de programación del punto 1.4.2, con la finalidad de realizar las pruebas directo de la consola del programa, con el objetivo que realmente se escuche el efecto deseado, con el sonido seleccionado. Además de la programación de los patrones y que todo funcione de manera unísona.

1.4.4 Conexión de elementos periféricos a la FPGA

Estudiar el funcionamiento del teclado a ocupar, además del potenciómetro rotario y la pantalla LCD a ocupar. Para ello se tendrá que estudiar la ficha técnica, además del programa del hardware en el programa de Vivado de Xilinx y su respetiva comunicación en el programa de SDK de Xilinx. Con el fin de adaptar los presentes periféricos al código del modo patrón y efectos de sonidos de la FPGA.

En paralelo de deberá crear un programa la FPGA que permita la interacción del circuito con el paquete de datos y la modificación de este, de tal manera que no haya problema a la hora de muestrear los datos.

1.4.5 Realización de pruebas en la Zybo Z7

Una vez teniendo en óptimas condiciones las conexiones del Hardware externo (teclado y parlantes), programado el Hardware mixto de la FPGA en el software de vivado y terminar de traspasar el código en C de la sección 1.4.3. Se procederá a probar el modo patrón y los efectos de sonidos, grabando el audio de estos modos y pasar los datos de la grabación al software de Matlab. Una vez teniendo las grabaciones de los efectos de sonidos en Matlab, se procederá a graficar y comparar lo obtenido por la simulación de la sección 1.4.2, con lo obtenido por la implementación en la FPGA.

1.4.6 Organización de la memoria

Con los datos de cada efecto del sintetizador obtenida y el funcionamiento de manera óptima del Sintetizador en la FPGA, se procederá a crear el informe final. En dicho informe se dará una explicación de cada efecto de sonido ocupado y su resultante, al modificar el tipo de onda. Se dará a conocer el listado del paquete de sonido elegidos para las pruebas del Sintetizador. También dar a conocer el sonido, para el patrón elegido y como se programó para que funcionase. La realización del circuito de control para el sintetizador, junto al diseño de placa y circuito, con una explicación de cada conjunto de botones y perillas, a toda esta etapa estarán las imágenes de las pruebas, que respaldarán los hechos. Se agregará un comparativo entre los modelos de mercado y el Sintetizador de sonido creado. Se creará una sección donde se encontrará un manual para el usuario, donde se explicarán cada función y combinación, para la operación del Sintetizador. Finalmente, de adjuntara al documento los códigos utilizados para la creación del Sintetizador de Sonido, tanto para la FPGA como su simulación en el software de Matlab.

1.5 Estudio comparativo de mercado

Dentro del mercado existe una gran gama de sintetizadores de música, cada uno con diferentes efectos y particularidades que lo hace único, de estos se destaca el hecho de la existencia de sintetizadores estándares de gran tamaño y dimensionamiento fijo, y los sintetizadores de música del carácter portátil de mucho menos capacidad que su versión industrial, pero con la diferencia de su facilidad de uso y la opción de ocuparlo en cualquier lugar sin necesidad de estar conectado a la red eléctrica.

Dentro de los equipos de sonidos anteriormente nombrados esta la limitante de los efectos de sonido de cada modelo, los sonidos que trae y sobre todo el alto precio que tienen, donde para obtener los efectos deseados o el funcionamiento ideal a la hora de componer, es necesario comprar más de un sintetizador o equipo de sonido. La Tabla 1.2 muestra los precios de los sintetizadores de sonido portátiles PO-20 KO, PO-28 KO, PO-32 KO, el sintetizador de carácter más profesional OP-Z, JUNO-DS61, Yamaha MX61 v2, Korg Kross2 61 y la FPGA Zynq7000, con la finalidad de hacer un comparativo de precios.

Tabla 1.2. Listado de precios de los sintetizadores de música y la FPGA.

Equipo	Precio Promedio(CLP)
PO-20 KO	72.000[20], [21]
PO-28 KO	73.000[22], [23]
PO-32 KO	105.000[24], [25]
OP-Z	520.000[26]
JUNO-DS61	630.424[27]
Yamaha MX61 v2	623.152[28]
Korg Kross2 61	630.497[29]
Chip XC7Z010	54.400[30]
Tarjeta Zybo Z7-10	157.773[31]

Ante los precios presentados en la Tabla 1.2, se da comprender los altos precios que poseen los sintetizadores de alta gama, los cuales superan los valores de \$500.000, debido a su alta complejidad de funcionamiento, explicados en la sección 1.2.1. Dentro de los precios más accesibles se encuentran los sintetizadores del serial PO KO, los cuales son mucho más económicos, pero a costo de un funcionamiento más limitado al modelo del equipo. Por último, se encuentra el Chip XC7Z010, que se encuentra dentro del rango del precio de los sintetizadores PO KO, como se ha detallado en la sección 1.2.4 y próximamente en el capítulo 2, de explicaran el funcionamiento de la Zybo Z7-10, la cual posee la capacidad de tener los mismos elementos de los sintetizadores del serial PO KO y de ser programadas para poder emular su funcionamiento.

2 Especificaciones de la FPGA Zybo Z7

2.1 Introducción general

La SoC FPGA Zybo Z7-10 posee una gran potencia y capacidad, que permite realizar diferentes controles, ya sea electrónica, ondas y en el presente caso control de sonidos. El conocimiento de las especificaciones de la Zybo Z7-10, serán de utilidad para comprender la ubicación de cada elemento, durante el desarrollo de los siguientes capítulos. Las especificaciones de la FPGA, también servirán para conocer los alcances y limitaciones, de la tarjeta Zybo Z7-10. Lo que permitirá una mejor implementación a la hora de la creación de la sintetización de audio.

2.2 Especificaciones

La Zybo Z7 es un chip que posee un procesador y una parte lógica programable, que pueden ser operados mediante una placa de desarrollo de circuitos digitales, la cual contiene un gran número de funciones integradas. La Zybo Z7 se encuentra su construcción alrededor de la familia Xilinx Zynq-7000, la cual se basa en arquitectura de “Xilinx All Programmable System-on Chip (AP Soc)”. Dicha arquitectura posee una integración de un procesador ARM Cortex-A9 de doble núcleo, con Xilinx 7 series “Field Programmable Logic Gate Array” (FPGA). Dentro de las funciones integradas por la placa de circuitos digitales, se tiene un rico conjunto de periféricos multimedia y de conectividad, que la provee de una gran flexibilidad y potencia agregada por la FPGA. La capacidad de la Zybo Z7 permite funciones que incluye conector Pcam compatible con MIPI CSI-2, entrada HDMI y conexiones tipo Pmod, entre otras funciones. Los conectores Pmod de la Zybo Z7, permite el acceso al catálogo de Digilent de más de 70 placas periféricas Pmod, las cuales incluyen controladores de motor, sensores, pantallas y más[32].

Dentro de las características principales, de la FPGA Zybo Z7-10, ocupada se encuentra los siguientes listados.

Zynq Procesador[32]:

- Procesador de doble núcleo Cortex-A9 de 667 MHz

- Controlador de memoria DDR3L, con 8 canales y 4 puertos esclavos AXI3 de alto rendimiento.
- Controladores periféricos de ancho de banda alta: Ethernet 1G, USB 2.0 y SDIO.
- Controladores periféricos de ancho de banda baja: SPI, UART, CAN e I²C.
- Programable JTAG, flash Quad-SPI y tarjeta microSD.
- Programación lógica equivalente a Artix-7 FPGA.
- Look-up Tables (LUTs) de 17600.
- Cantidad de Mosaicos de administración de reloj de 2.

Memoria[32]:

- DDR3L de 1 GB con un bus de 32 bits @ 1066 MHz.
- Memoria Flash Quad-SPI de 16 MB con programación de fábrica.
- Bloque de RAM de 270 KB.
- Un total de 35200 Flip-Flops internos.

Alimentación[32]:

- Energía suministrada desde un USB o cualquier alimentación externa de 5V.

USB and Ethernet[32]:

- Gigabit Ethernet PHY.
- Circuito de programación USB-JTAG.
- Puente USB-UART.
- USB 2.0 PHY con host y soporte de dispositivo.

Audio y video[32]:

- Conector de cámara Pcam con MIPI Soporte CSI-2.
- Puerto de fuente HDMI (salida) con CCA.
- Códec de audio con estéreo auriculares, entrada de línea estéreo y tomas de micrófono.

Interruptores, pulsadores y Leds internos[32]:

- 6 pulsadores (2 procesadores conectado).
- 4 interruptores deslizantes.
- 5 Leds (1 procesador conectado).
- 1 RGB Led.

Expansión de conectores[32]:

- 5 Puertos Pmod:
 - Un total de 8 procesadores I/O.
 - Un total de 32 FPGA I/O.
 - Una capacidad de 4 entradas análogas 0-1V diferencial pares a XADC.

Para la clasificación de elementos de la FPGA Zybo Z7, se encuentra en la Fig.2.1 y descrita en la tabla 2.1. Hay que recalcar en la Fig.2.1, se encuentran los elementos de la tarjeta Zybo Z7-10, la cual solo tiene leves diferencias que su versión posterior, de la Zybo Z7-20, denotadas en la tabla 2.1.

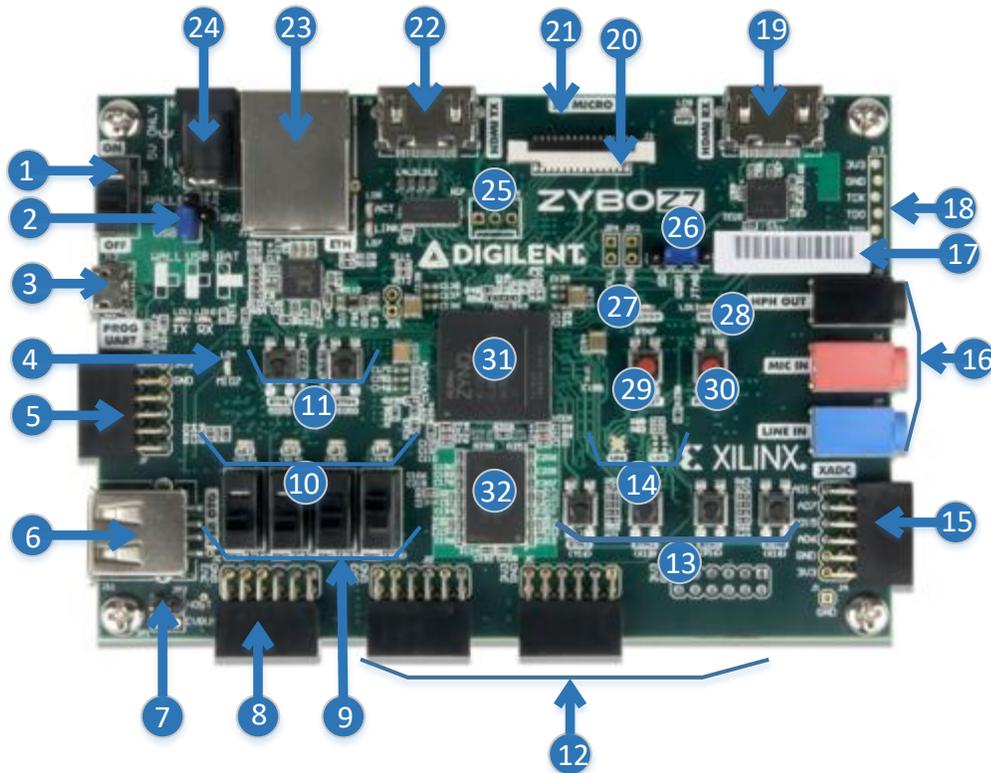


Fig.2.1. Enumeración de los elementos de la Zybo Z7-10[32].

Tabla 2.1. Listado elementos de la Zybo Z7-10[32].

N°	Descripción	N°	Descripción
1	Interruptor de alimentación	17	Etiqueta de dirección MAC única
2	Puente de selección de potencia	18	Puerto externo JTAG
3	Puerto USB JTAG/UART	19	Puerto de entrada HDMI
4	LED de usuario MIO	20	Puerto MIPI CSI-2
5	Puerto MIO Pmod	21	Conector micro SD (en la otra cara)
6	Puerto USB 2.0 Host/OTG	22	Puerto de salida HDMI
7	Puente de activación de alimentación del host USB	23	Puerto Ethernet
8	Puerto estándar Pmod	24	Conector de fuente de alimentación externa
9	Interruptores internos	25	Conector ventilador(solo Zybo Z7-20)
10	Leds internos	26	Puente de selección de modo de programación
11	Botones MIO internos	27	Led indicador de fuente de alimentación
12	Puertos de alta velocidad Pmod (JB disponible Zybo Z7-20)	28	Led de programación realizada de la FPGA
13	Botones internos	29	Botón de reinicio del procesador
14	Led RGB(solo uno para Zybo Z7-10)	30	Botón de limpiado de la configuración de la FPGA
15	Puerto Pmod XADC	31	Zynq-7000
16	Puerto códec Audio	32	DDR3L Memoria

2.3 Zynq APSoC Arquitectura

La Zynq APSoC se encuentra dividida en dos subsistemas:(I) el sistema de procesamiento (PS) y (II) lógica programable (PL). En la Fig.2.2 da una descripción general de la arquitectura de la Zynq, en donde la sección en color verde claro corresponde al PS y el color amarillo al PL.

La arquitectura PL, es similar a la FPGA Artix de la serie 7 de Xilinx, con la única diferencia que posee varios puertos y buses dedicados, los cuales acoplan firmemente a la PS[33]. El PL tampoco trae el mismo hardware de configuración típico de la FPGA serie 7, donde esta debe ser configurada directamente por el procesador o mediante el puerto JTAG.

La arquitectura de la PS posee un gran número de componentes, de la cuales que incluyen una Unidad de Procesamiento de aplicaciones (APU, que incluye 2 Cortex-A9). Otros componentes son su interconexión de Arquitectura de Bus de Microcontrolador Avanzado (AMBA), controlador de memoria DDR3 y una gran cantidad de controladores de periféricos con sus I/O multiplexadas a 54 pines dedicados. Los controladores periféricos que no poseen sus entradas y salidas conectadas a los pines MIO pueden en su lugar, enrutar sus I/O a través del PL[32].

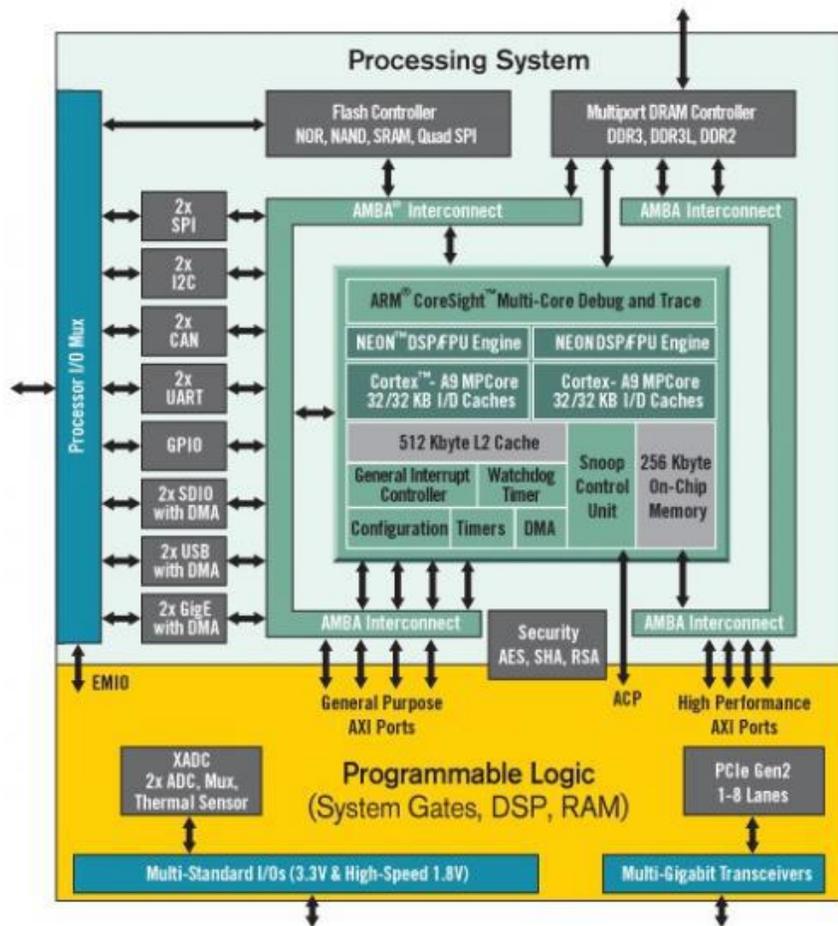


Fig.2.2. Arquitectura general de la Zybo Z7-10[32].

2.4 Reloj/Oscilador

La Zybo Z7 proporciona un reloj de 33.3333 MHz, para la entrada Zynq PS_CLK, la cual se utiliza para generar los relojes para cada uno de los subsistemas. La entrada de 33.3333 MHz permite que el procesador funcione a una frecuencia máxima de 667MHz y junto con el controlador de memoria DDR3, que puede operar con una frecuencia máxima de reloj 533 MHz.

El PS posee un PLL dedicado, el cual es capaz de generar hasta cuatro relojes de referencia, cada uno con frecuencias configurables. Estos cuatro relojes se pueden utilizar para sincronizar la implementación de lógica personalizada en el PL. Además, la Zybo Z7 proporciona un reloj de referencia de 125 MHz directamente al pin K17 del PL. El reloj de referencia externo permite utilizar el PL completamente independiente de la PS. El PL del

Zynq Z7-10 también incluye dos MMCM y dos PLL que se pueden usar para generar relojes con frecuencias precisas y relaciones de fase.

En la Fig.2.3 se describe el esquema del reloj que se utiliza en la Zybo Z7. El presente esquema cuenta con la salida del reloj de referencia del Ethernet PHY, el cual utiliza como reloj de referencia de 125 MHz para el PL[32].

Para el desarrollo de la implementación del sintetizador de audio, se ocupará un reloj de 44100 Hz, puesto que la presente frecuencia es la unidad típica y estándar para muestrear señales de sonido.

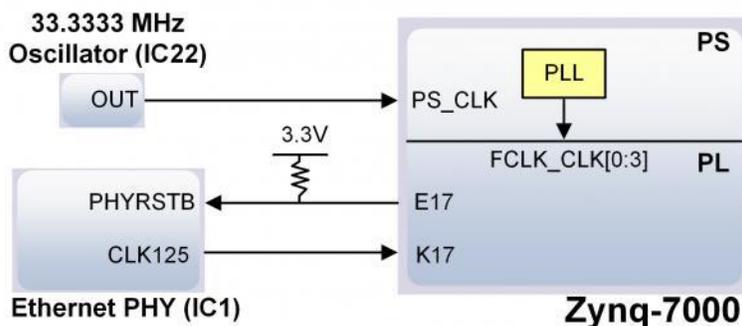


Fig.2.3. Esquema del reloj de la Zybo Z7-10[32].

2.5 Audio

El dispositivo análogo del códec de audio SSM2603 proporciona el procesamiento de audio digital integrada al Soc Ap Zynq-7000. El presente códec de audio permite la grabación y reproducción estéreo a frecuencias de muestreo entre 8kHz a 96kHz. Dentro de su hardware posee tres conectores de audio de 3.5 mm, los cuales se encuentran descritos en la tabla 2.2. La alimentación analógica es proporcionada por una fuente lineal dedicada (IC5) [32].

Tabla 2.2. Señales de audio análogas[32].

Jack de audio	Descripción	Canal	Color
J5	Salida estéreo de audífonos	Estéreo	Negro
J6	Entrada de micrófono	Mono	Rosa
J7	Entrada línea estéreo	Estéreo	Azul claro

La interfaz digital del SSM2603 se encuentra conectada al lado del PL de la Zynq. Los datos del audio se transfieren por medio del protocolo I²S. La configuración de la interfaz digital se realiza a través de un bus I²C y la dirección del presente códec es 0011010b. Para todas la I/O digitales se tiene un nivel de 3.3 V y se conectan a un banco FPGA, de igual valor de alimentación.

Para el funcionamiento del códec de audio, este debe sincronizarse desde la Zynq en el pin MCLK. El códec de audio utilizara el reloj maestro, para establecer la frecuencia de muestreo de audio. Para el óptimo funcionamiento del reloj maestro debe ser un múltiplo entero de la frecuencia de muestreo deseado. La configuración predeterminada requiere un reloj maestro de 12.288 MHz, lo que da como resultante una frecuencia de muestreo de 48kHz.

El códec de audio posee dos modos: maestro y esclavo, siendo este último el modo predeterminado. En el modo esclavo las direcciones de las señales se dan a conocer en la tabla 2.3. Para cuando se configura en modo maestro, la dirección de RECLRC, BCLK y PBLRC se invierten. En el modo maestro, el códec genera las frecuencias adecuadas para esos relojes. Indiferente de la selección de los relojes, PBDAT necesitara ser desactivado y RECDAT muestreado en sincronía con ellos. Finalmente, el reloj maestro siempre provendrá de la Zynq [32].

Para el capítulo 5 se ocuparán principalmente las configuraciones de audio y como puerto de salida se ocupará el J5, de la tabla 3.2 que corresponde a la salida de audio estéreo. Este audio estéreo será conectado a un amplificador, con la finalidad de escuchar las modificaciones del audio realizadas.

Tabla 2.3. Señales digitales del audio para el códec SSM2603 en modo de esclavo[32].

Pin SSM2603	Protocolo	Dirección desde la Zynq	Pin Zynq
BCLK	I ² S(Reloj serial)	Salida	R19
PBDAT	I ² S(Dato de reproducción)	Salida	R18
PBLRC	I ² S(Canal de reproducción del reloj)	Salida	T19
RECDAT	I ² S(Registro de dato)	Entrada	R16
RECLRC	I ² S(Canal de registro del reloj)	Salida	Y18
SDIN	I ² C(Dato)	Entrada/ Salida	N17
SCLK	I ² C(Reloj)	Salida	N18
MUTE	Activación digital(Activación baja)	Salida	P18
MCLK	Reloj maestro	Salida	R17

2.6 I/O Básicas

La placa Zybo Z7 incluye un total cuatro interruptores deslizantes, cuatro botones de tipo pulsador, cuatro leds individuales y un led tricolor, los que se encuentran conectado al PL de la Zynq, tal como se representa en la Fig.3.4. Además, la Zybo Z7 posee dos botones y un led conectados directamente a la PS, por medio de los pines del MIO, tal como lo indica la Fig.2.4. Es preciso mencionar que todo interruptor y semiconductor se encuentran conectadas a una resistencia en serie, con el fin de evitar cortocircuitos involuntarios. La lógica de funcionamiento de los interruptores de la Zybo Z7 es de servir como entradas, con un estado bajo en reposo y estado alto cuando se presionen. Por parte de los interruptores deslizantes, estos generan entradas constantemente en estado bajo o alto, dependiendo su posición [32].

Para el control de los leds e interruptores del PL y del PS pueden ser realizados por medio de los GPIO de la Zynq. De estos se ocuparán los leds del PL para muestrear la secuencia de los patrones de tiempo e indicar la activación de efectos de la implementación, los cuales serán detallados en el capítulo 3. Mientras que, el bloque PL se ocuparan sus botones, los cuales se encuentran a la entrada de este. Los botones del bloque PL servirán para activar y controlar, los canales y los efectos de sonidos. Estos modos, canales y efectos, serán explicados y simulados en el capítulo 3, implementados en el capítulo 4 y los resultados experimentales en el capítulo 5.

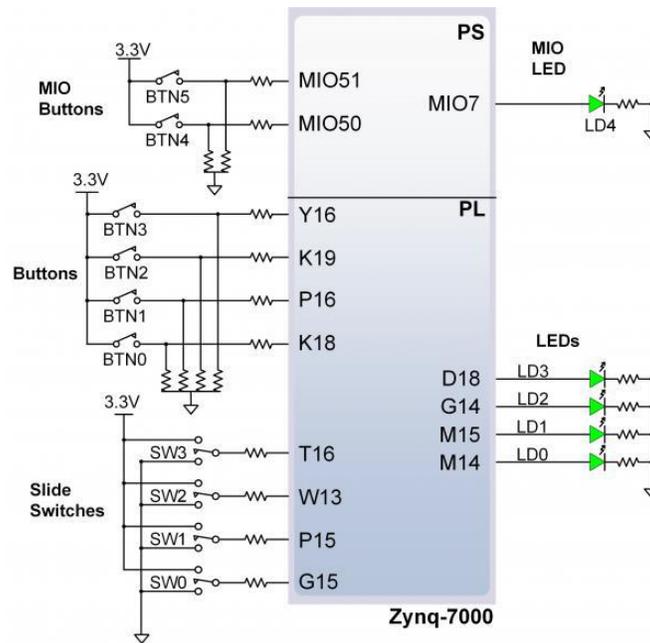


Fig.2.4. Esquemático de conexión de entradas y salidas básicas de la Zybo Z7-10[32].

2.7 Pmod Puertos

Los puertos Pmod son conectores hembra de una dimensión de 2 x 6, de forma rectangular, con un espaciado 100 mm, los cuales pueden ser acoplados por conectores estándar de la misma dimensión de 2 x 6 pines. Cada puerto Pmod posee un total de 12 pines, con dos pines destinados a Vcc de 3.3V (pines 6 y 12) y dos pines de tierra Gnd (pines 5 y 11). Los otros 8 pines del Pmod, están libres para ser ocupados como I/O, y poseen una capacidad de corriente máxima de salida de hasta 1 [A]. La distribución de los pines del Pmod se muestra en la Fig.2.5.

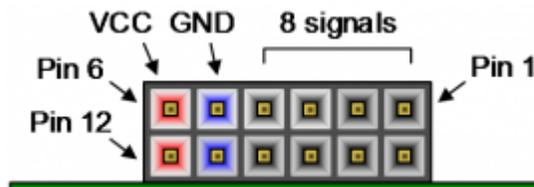


Fig.2.5. Puertos Pmod de la Zybo Z7-10[32].

Tabla 2.4. Tipo de Pmod y definición de sus pines de la Zybo Z7-10[32].

	Pmod JA	Pmod JB	Pmod JC	Pmod JD	Pmod JE	Pmod JF
Tipo de Pmod	XADC	Alta velocidad	Alta velocidad	Alta velocidad	Estándar	MIO
Pin 1	N15	V8	V15	T14	V12	MIO-13
Pin 2	L14	W8	W15	T15	W16	MIO-10
Pin 3	K16	U7	T11	P14	J15	MIO-11
Pin 4	K14	V7	T10	R14	H15	MIO-12
Pin 7	N16	Y7	W14	U14	V13	MIO-0
Pin 8	L15	Y6	Y14	U15	U17	MIO-9

La Zybo Z7 posee seis puertos Pmod, los cuales se distribuyen según su función. Dentro de la clasificación de los tipos de puerto Pmod se encuentran: Estándar, conexión MIO, XADC y de alta velocidad. En la tabla 2.4, se especifica que puerto Pmod corresponde a una de estas cuatro categorías, además de la clasificación de cada uno de estos pines.

Por tema de desarrollo solo se explicarán a profundidad los Pmod tipo estándar y de alta velocidad, puesto que estos son los tipos de Pmod que se ocuparon para la implementación y su funcionamiento cumple con los requisitos de los elementos de la memoria. Dichos elementos serán explicados en el capítulo 4.2.

2.7.1 Pmod Estándar

Los puertos Pmod estándar se encuentran conectado al PL Zynq, por medio de unas resistencias en serie de 200 Ω . Estas resistencias cumplen la finalidad de servir contra los cortocircuitos, en caso que el usuario envíe una señal errónea, cuando se supone que debe ser usada como entrada. La desventaja de esta estrategia de protección es que limita la velocidad de conmutación de las señales de datos[32].

2.7.2 Pmod de alta velocidad

Los puertos Pmod de alta velocidad tiene el mismo sistema de conexión que los Pmod estándar, con la diferencia que los datos de las señales se enrutan como pares diferenciales emparejadas por impedancia de 80 Ω , para conmutaciones de alta velocidades. Los Pmod de

alta velocidad poseen almohadillas, para cargas resistencias para una mayor protección. Las resistencias en serie derivadas de los Pmod no ofrecen protección contra cortocircuitos, pero permiten velocidades de conmutación mucha más rápidas [32].

Estos puertos solo deben usarse cuando se requiera señales de alta velocidad (16 Mbps) o cuando otros puertos Pmod se encuentre ocupados, sirviendo como un Pmod normal. Se recomienda que en aplicaciones de gran carga solo se ocupen los Pmod del tipo estándar.

Los puertos Pmod de alta velocidad serán los ocupados durante la implementación, con el fin de controlar los periféricos, tales como el teclado táctil, el encoder rotativo y la pantalla LCD, por medio de los GPIO de la Zybo. Tales periféricos serán detallados en el capítulo 4.2.

3 Funciones de sonido del sintetizador de audio en una plataforma Zynq-7000

3.1 Introducción

En el presente capítulo se abarcará la funcionalidad y los comportamientos, de los diferentes modos que posee el sintetizador de audio implementado en una plataforma Zynq-7000. En dicha distinción de los diferentes modos, se procederá a explicar su funcionamiento, junto a su diagrama correspondiente, para lograr entender su operación. Para cada diagrama es necesario comprender que cada modalidad (modo de efecto de sonidos, modo compositor y modo variación de parámetros) posee su propio funcionamiento dentro del sintetizador de sonido. Los cuales servirán para emular y representar las funciones de los sintetizadores de sonidos investigados y detallados en el capítulo 1. De estos sintetizadores de sonido se dará un mayor enfoque al funcionamiento del serial PO KO.

Dentro de los modos a tratar se encuentra en la sección 3.2 el modo compositor, el cual se encuentra en funcionamiento de sus 16 tiempos, puesto que el sintetizador posee un total de 16 pulsadores, en donde cada uno producirá un sonido en su tiempo correspondiente de activación. Dentro de la modificación y a la vez mejora se encuentra la adición de dos canales, lo cual no se encuentra en otros modelos del serial PO KO. Posterior a ello en la sección 3.3 se tocará los efectos de sonidos, los cuales son un total de 16 efectos de sonidos, basados en efectos básicos y la aplicación de filtrado FIR e IIR, nombrados en los capítulos 1.2.3.2 y posterior 4.3. Es de mencionar que ya se han explicado los primeros 8 efectos de sonidos en el capítulo 1.2.3 y se volverán a explicar con mayor profundidad, estos y los otros 8 efectos desde la sección 3.3.2 a la 3.3.17.

Para el cambio de efecto de sonido se realizará por medio de la combinación de dos botones, el btn 1 integrado en la FPGA y una tecla del teclado táctil de 16 posiciones, como se muestra en la Fig.3.1 del Setup. Es mediante esta combinación que se podrá activar los efectos de sonidos, pasando desde el efecto 1 al 16. Para oír los sonidos originales del modo de efecto de sonido, se deberá entrar únicamente entrar al canal 16 (de modo de efectos de sonidos), en caso que ya se ha aplicado algún efecto, solo hay que apretar únicamente el btn 1. Dentro de las capacidades de implementar un sintetizador de audio en la FPGA se

encuentra la ventaja de cambiar los efectos de sonidos y los sonidos que se le van a aplicar los efectos, únicamente modificando la programación del efecto y reemplazando el archivo de audio correspondientemente.

Puesto que ya se habló de la activación de los efectos de sonido y del modo compositor, queda otra variable bastante importante a tratar la cual es la modificación de los parámetros del sintetizador de sonido, los cuales se tratarán en mayor profundidad en la sección 3.4. En dicho capítulo se tocará los temas de modificación de parámetros necesarios, ya sea velocidad, volumen, efecto especial de cada canal y efecto, entre otras variables y como está afectará a los otros modos.

Las Fig.3.2 a la Fig.3.5 representan los diagramas de flujos asociados a cada modo: (I) modo compositor, (II) modo de efectos de sonido (Fx) y (III-IV) modo modificación de parámetros para (i) y (ii). En dichos diagramas se muestran el funcionamiento del sintetizador de audio. Es de mencionar que en cada sección (3.2, 3.3 y 3.4) poseerá su propio diagrama de flujo desarrollado a profundidad y los presentados en las Fig. 3.2 a Fig. 3.5 darán a conocer de manera generalizada el funcionamiento de los modos interactuando entre sí, de manera comprimida.

En la Fig. 3.2 se presente el funcionamiento del modo patrón, también conocido como modo compositor, que permite crear ritmos por medio de la activación de sus 16 posiciones de intervalo de tiempo, por canal. En dicho diagrama de flujo se hace la consideración de cuando una posición es activada, el sonido de dicha posición del modo patrón sonara, de manera periódica, en su posición dentro de los 16 intervalos de tiempo. En caso contrario que este ya se encuentre activo, al volverlo a presionar se desactivará el sonido de la posición.

En la Fig. 3.3 se muestra el diagrama de flujo asociado a el funcionamiento del modo de efectos de sonido, el cual se encuentra abreviado como “Fx”. El modo de funcionamiento es el siguiente, al entrar al modo de efectos de sonidos, si no se aprieta una combinación del teclado y el botón Fx (btn 1 según la Zybo Z7), no se aplicará el efecto de sonido y solo sonará el sonido original, de los audios para efectos de sonido. En caso contrario que, si se dé la combinación del teclado táctil más el botón Fx, se aplicara el efecto correspondiente al número del teclado táctil, modificando el sonido original. Todos estos elementos nombrados representan en la Fig.3.1 correspondiente al setup del sintetizador.

En el diagrama de la Fig. 3.4 se da a conocer el funcionamiento del modo de modificación de parámetros del modo compositor. Su funcionamiento se basa en la modificación de dos parámetros, la intensidad del volumen de audio y la velocidad del BPM (beats per minute). Cada canal del modo patrón, poseerá su modificación independiente, por medio del giro del encoder rotario implementado. Dicho tema se tocará en el capítulo 3.4, de modificación de parámetros.

Para el diagrama presentado en la Fig. 3.5, posee los mismos principios de el de la Fig. 3.3, puesto que ocupa la misma estrategia de control de variables, mediante encoder rotatorio, pero para el presente caso, solo se mantiene la modificación del volumen del sonido del efecto elegido, con la perilla A. Para la perilla B, se le encarga la tarea de modificar las variables correspondientes a cada efecto de sonido, tales como el tiempo de inserción de vacío (efecto 4), el comienzo de los rebotes (efecto 5) y tiempo de inserción del Glich (efecto 6).

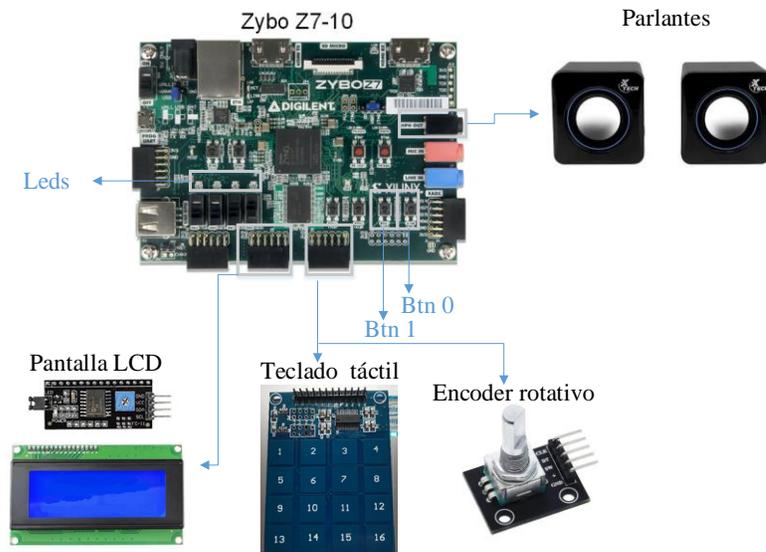


Fig.3.1. Diagrama del setup del sintetizador de sonido.

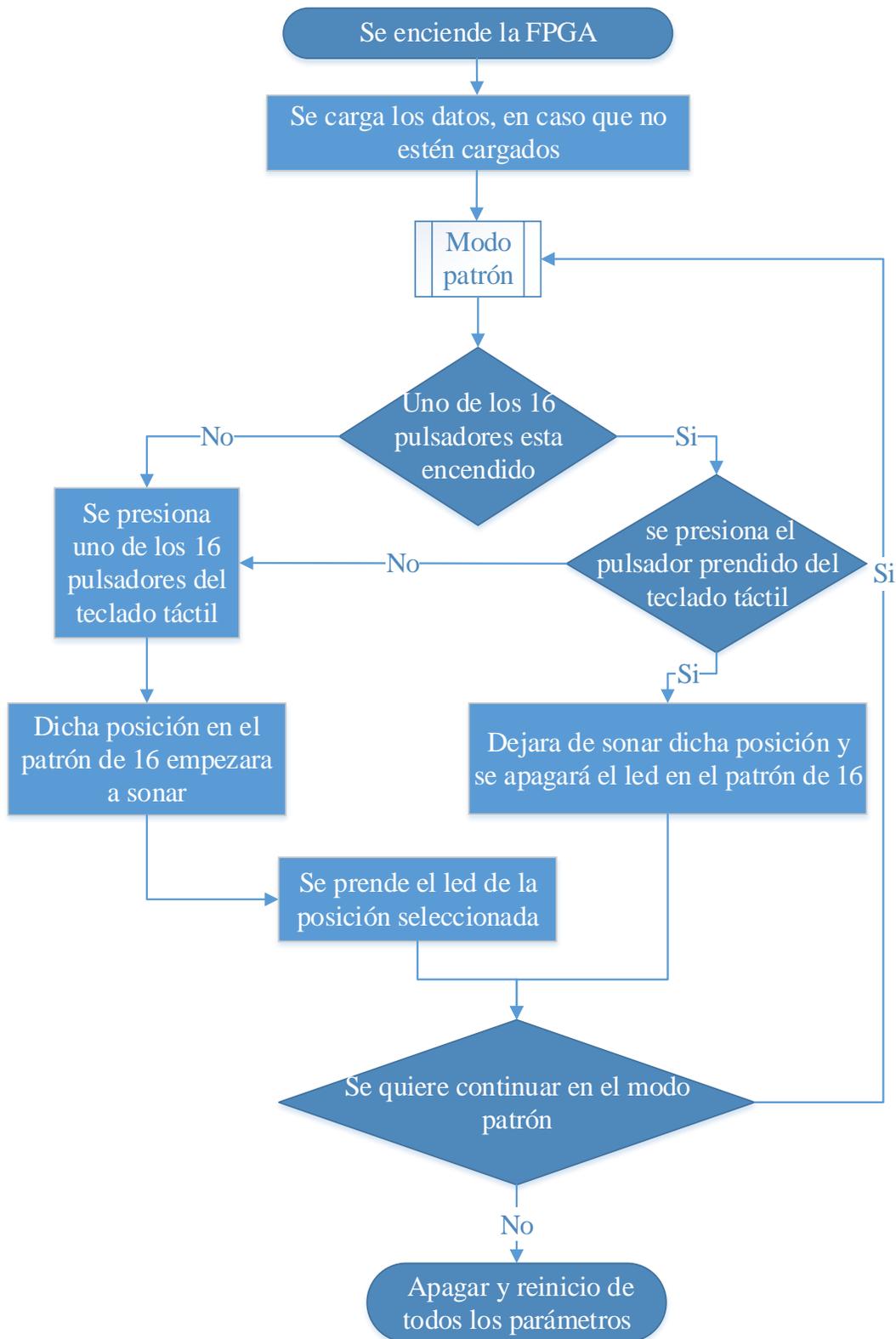


Fig.3.2. Diagrama de Flujo modo compositor del sintetizador de audio.

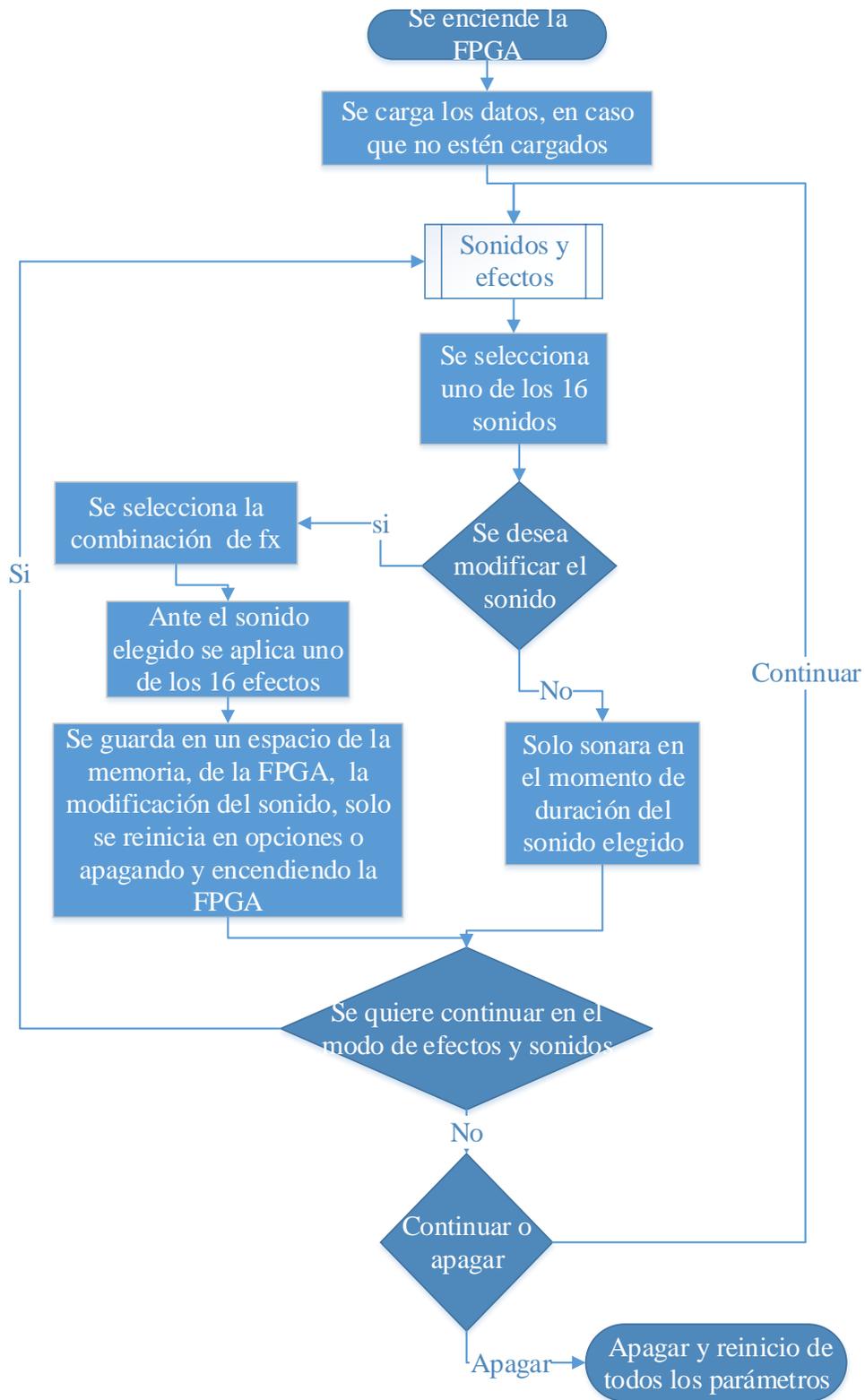


Fig.3.3. Diagrama de Flujo modo de efectos de sonido.

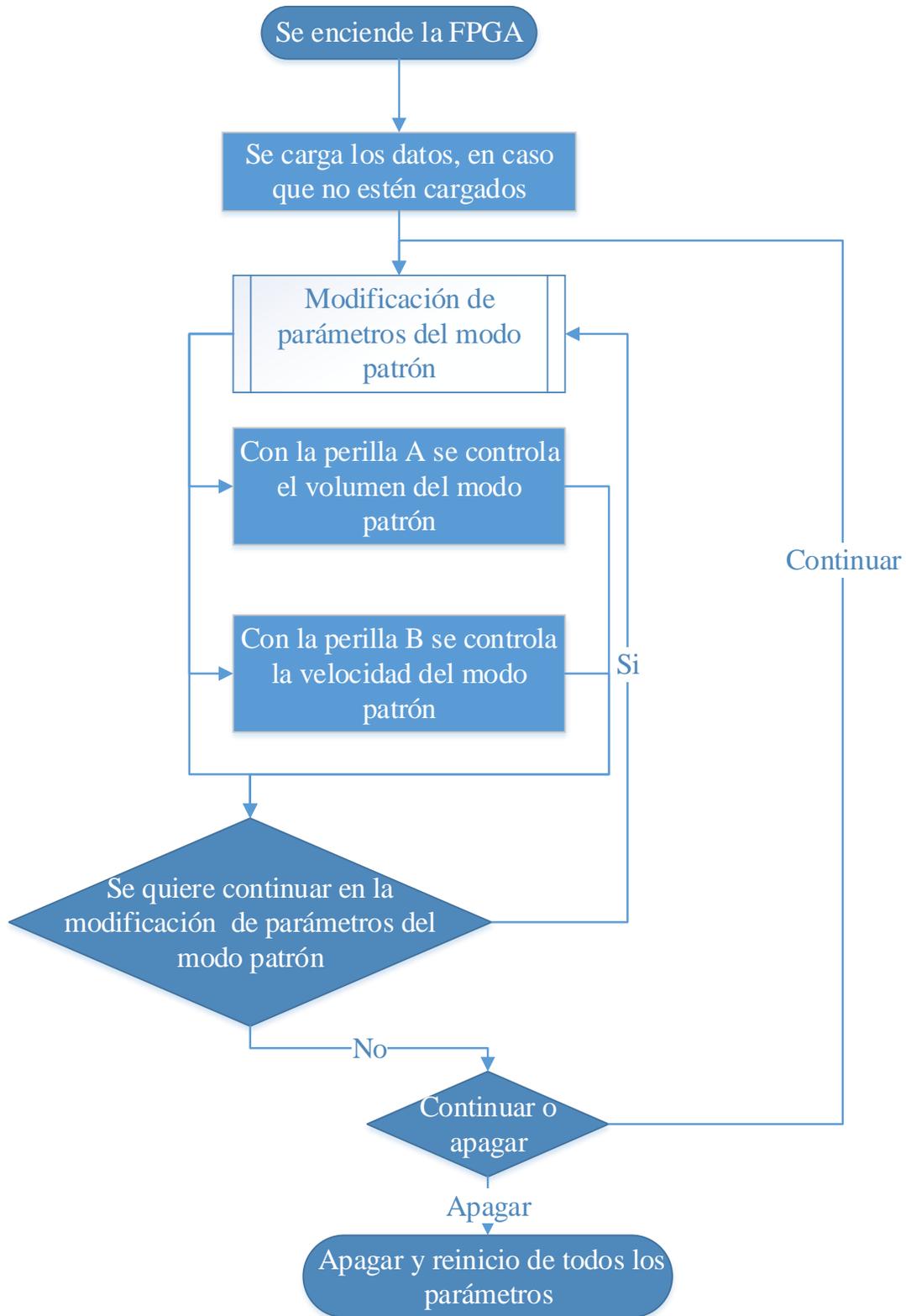


Fig.3.4. Diagrama de Flujo modo modificación de parámetros del modo compositor.

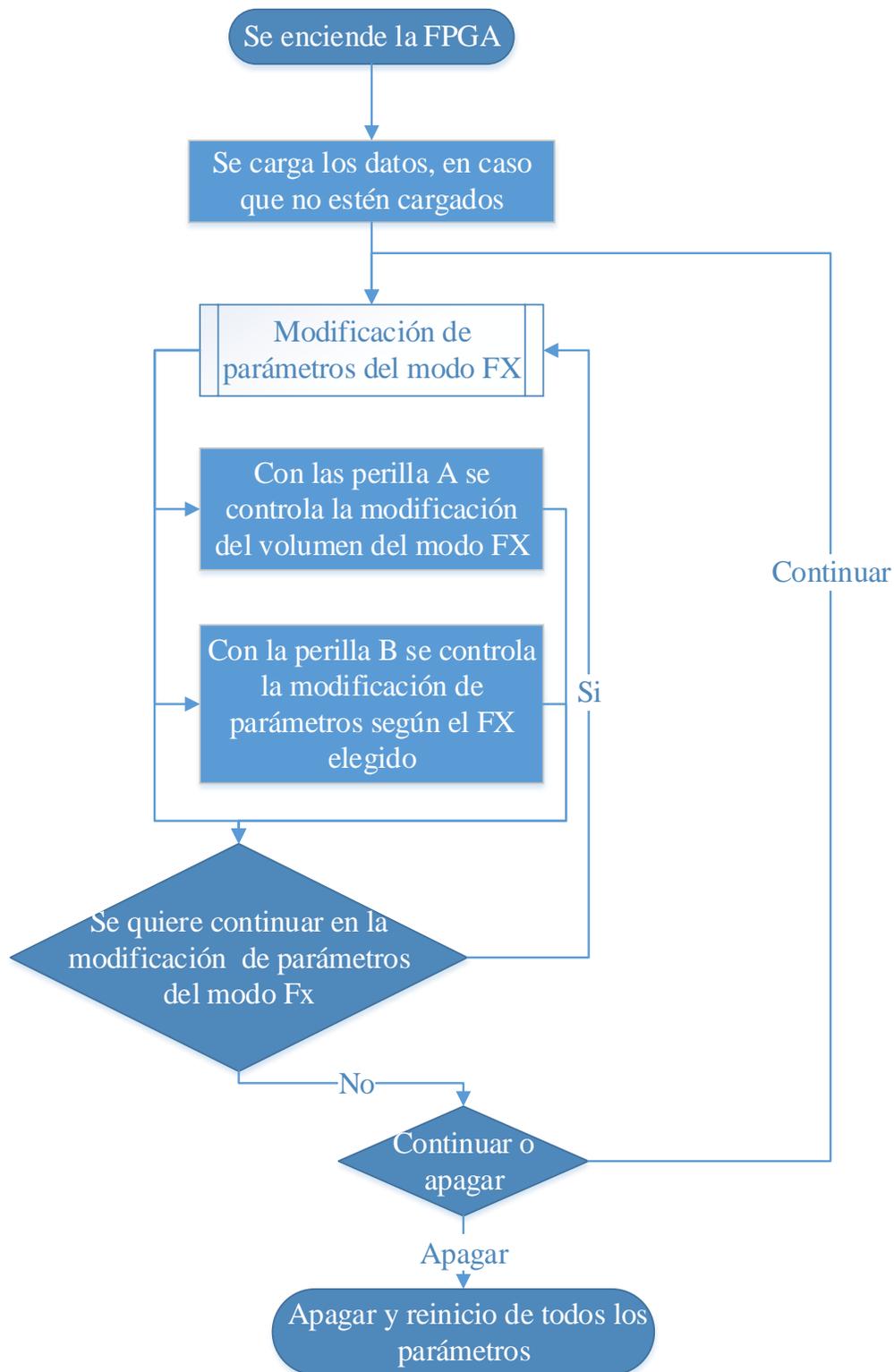


Fig.3.5. Diagrama de Flujo modo modificación de parámetros del modo efecto de sonido.

3.2 Implementación del modo compositor

El modo compositor se ha nombrado y explicado en las secciones anteriores. El objetivo de este modo es la creación de patrones, por medio del guardado de los estados de las posiciones de sonido. Esto es logable por medio de las posiciones de sonidos, distribuida en tiempos de 16 intervalos. Dichas posiciones son activadas o desactivadas por medio del teclado táctil de 16 teclas, en donde cada tecla se le asigna un intervalo de tiempo.

Según la modificación e innovación se ha logrado aumentar el número de canales del modo patrón, pasando de 2 canales existentes de manera comercial (PO KO operation) a un total de 15 canales ofrecidos en la presente implementación. Esto es posible mediante la inclusión de un pulsador extra (btn 0) a los 16 pulsadores anteriormente mencionados, se podrá cambiar de canal, permitiendo trabajar con dos diferentes sonidos, que funcionará de manera paralela y siendo reproducidas por el sintetizador al mismo tiempo. La presente mejora no se ha implementado en los sintetizadores portátiles, en especial al serial PO KO, anteriormente investigada, siendo únicamente la presente aplicación mostrada en software de sintetización de audio tales como el FL Studio[34], con un máximo de 125 canales de modo patrón, por el hecho que es una implementación únicamente en software. Las gráficas de los modelos nombrados versus la cantidad de modo patrones que poseen se exponen en la Fig.3.6.

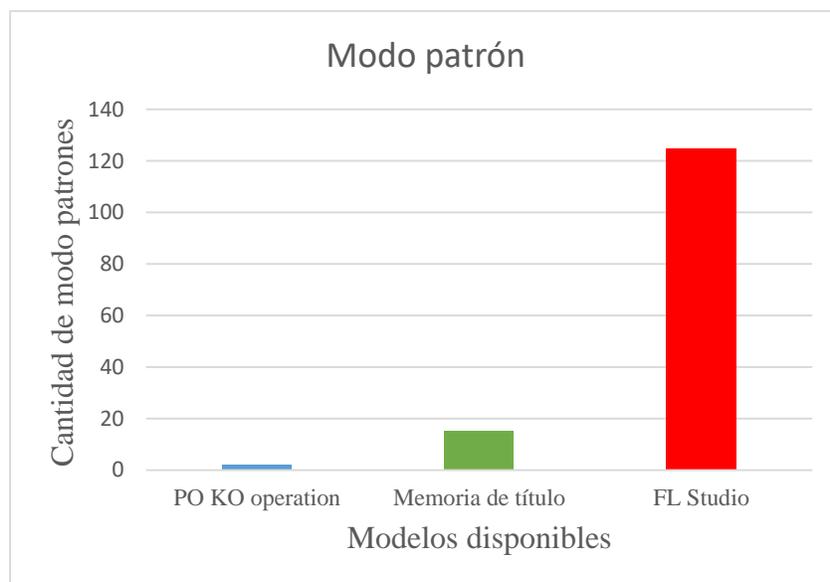


Fig.3.6. Gráfica de cantidad de modo patrón de los modelos investigados.

En la Fig. 3.7 se expone el funcionamiento del modo compositor, en la cual se ve como el sintetizador de audio en una plataforma Zynq-7000, va a recibir las diferentes variables necesarias para su funcionamiento, en donde se analizará todo el proceso desde que se entra al modo compositor hasta el momento que se apaga o se sale de dicho modo. En donde es necesario entender que el presente diagrama de flujo solo está contemplado para un canal, siendo el comportamiento de ambos canales exactamente iguales.

Dentro del diagrama de flujo mostrada en la Fig. 3.7, se considera el guardado de datos de cada una de las 16 posiciones, siendo estas guardadas con un dato en alto, si se encuentra anteriormente activa y con un estado en bajo si estas no se encuentran activas. Dentro del modo patrón únicamente emitirá sonido las variables en alto, en su tiempo correspondiente del intervalo de 16 tiempos. Posterior a ello si se presiona una de las 16 teclas la variable “Var1” pasará a estar en alto. La función de crear la variable “Var1” es para hacer reconocimiento que se presionó alguna tecla, para luego hacer la discriminación alguna de las 16 posiciones se encuentra ya con un estado alto, ya sea por una activación anterior o que en caso contrario se encuentre desactivada. En caso que, sí se encuentre en alto, proseguirá a discriminar si el pulsador sigue presionado, por lo cual, si se deja de presionar la variable “Var2”, pasara a estar en alto, este método de tener las variables “Var1” y “Var2” sirve para simular una activar por flanco de bajada, puesto que no existe una función que emule un flanco de bajada de la señal del pulsador, se tiene que emular por medio de las presentes dos variables, una para la activación del pulsador y otra para la desactivación del presente. Continuando con el diagrama de flujo al cumplirse las condiciones que la variable de la posición de 1-16, “Var1” y “Var2”, se procederá a apagar el led de la posición correspondiente y el sonido correspondiente a la posición procederá a dejar de sonar. Lo que provocara que la variable asignada una de las 16 posiciones pase a estar en estado bajo.

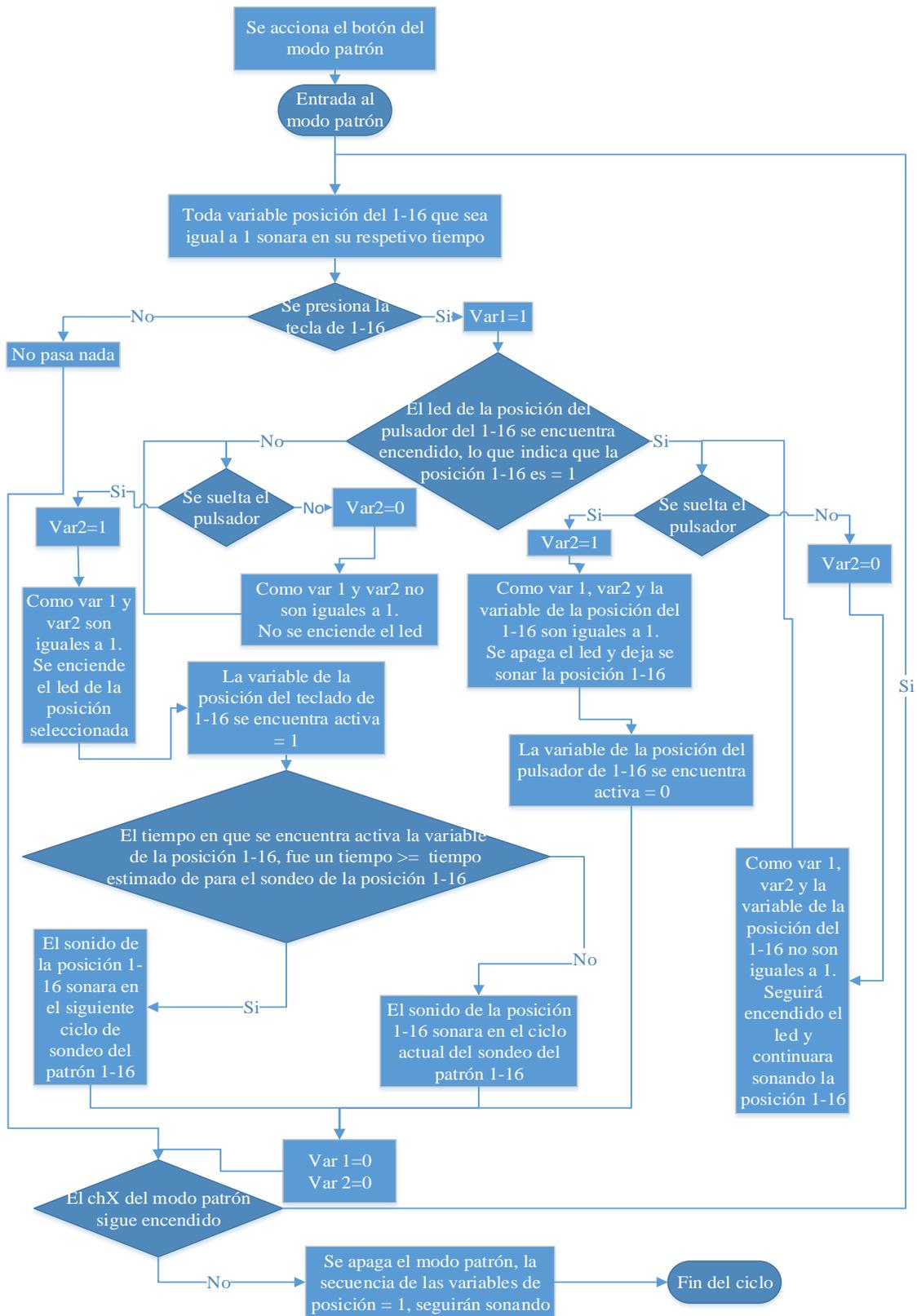


Fig.3.7. Diagrama de Flujo del modo compositor del sintetizador de audio.

En caso de que la variable de estado de la posición de 1-16, se encuentre en estado bajo, procederá a repetirse la emulación de activación por flanco de bajada con las variables “Var1” y “Var2”. Con lo cual el led asignado a una de las 16 posiciones se mantendrá encendido, indicando que la variable asignada se encuentra activa, estos leds se encuentran indicados por las Fig.2.1 número 11 y en la Fig.3.1. Junto a la activación de una de las 16 posiciones se dará que se emitirá el sonido correspondiente al tiempo de activación, de la posición, el cual puede sonar en el presente ciclo de intervalos, siempre y cuando el flanco de bajada se haya realizado antes del tiempo correspondiente para dicha posición. En caso de que la activación se haya realizado tiempo después del periodo de tiempo correspondiente para dicha posición de 1-16, comenzará a sonar en el siguiente ciclo. Lo que es equivalente a que ya haya pasado un ciclo, de 16 intervalos de tiempo, desde el punto de partida de la variable a reproducirse. Como resultado de lo anterior explicado la variable correspondiente a la posición de 1-16, se encontrará en estado en alto.

Si se llega a cerrar el modo compositor, toda variable de posición de 1-16 que se encuentren en alto seguirá sonando en su respectivo tiempo asignado, hasta que se desactiven volviendo a activar los pulsadores, de nuevo en el modo patrón o que se apague el sintetizador de sonido.

3.2.1 Selección de audio

Dentro de la selección de los sonidos del modo patrón, se realizaron una serie de consideraciones, con el propósito de hacer los sonidos de este modo, lo más parecido a los sonidos de los sintetizadores de la serie PO KO vistos en el capítulo 1.2.1.

Los requerimientos corresponden a:

1. Su duración el cual debe ser entre 0.4 y 1 segundo, puesto que es una sola posición de las 16 posiciones de intervalos de tiempo. En donde estos intervalos de tiempo del modo patrón se encontrarán disponibles para los 15 canales. Ya que en el modo patrón, el ciclo de intervalos de 16 tiempos poseen una velocidad original de 3.75 BPM (beats per minute).

2. Debe poseer una buena calidad de sonido, para no tener problemas a la hora de reproducirse.
3. Los sonidos deben ser de efectos o instrumentos. Por el tema que los sonidos ofrecidos por las páginas de descarga de sonidos de efectos, no poseían buena calidad o pedían requisitos de pago, se prefirió descargar canciones y recortas las pistas de audio, en donde se destacaban los solos de los instrumentos.

Es mediante lo anterior mencionado que la lista de sonidos del modo patrón de los canales 1 al 15 se muestran en la siguiente Tabla 3.1.

Tabla 3.1. Sonidos según su canal del modo compositor.

Canal	Nombre del sonido	Canal	Nombre del sonido
Ch1	Pikachu[35]	Ch9	Shamisen3[36]
Ch2	Punch[37]	Ch10	Kodo1[38]
Ch3	Piano1[39]	Ch11	Guitarra electrica1[38]
Ch4	Bajo1[40]	Ch12	Platillo [41]
Ch5	Redoble de tambor[40]	Ch13	Pulso1[38]
Ch6	Trompeta1[39]	Ch14	Punteo Guitarra eléctrica[38]
Ch7	Shamisen1[36]	Ch15	Kodo2[38]
Ch8	Shamisen2[36]		

El procedimiento para acondicionar el archivo de audio descargado, para destinarlo y ser utilizado en el sintetizador de sonido de la FPGA es a través de: (I) Realizar la descarga del archivo de audio en el formato .wav (Waveform audio file format) o .mp3(MPEG-2 Audio Layer III) los cuales son los formatos de audio estándar. (II)Se debe pasar por una etapa de acondicionado a través del software FL Studio 20(versión gratuita)[34]. En el FL Studio se analiza la pista de audio y se corta los segmentos que poseen las condiciones explicadas en el párrafo anterior, además de suavizar el término de la pista de audio, para que suene de forma más natural. (III) Exportar el archivo de audio al software de Matlab, en donde por medio de la función “audioread”[42], convierte el archivo de audio en un vector, bajo la frecuencia de 44100 Hz, la cual es la frecuencia de muestreo de todas pistas de audio en general. (IV) Ya teniendo el audio vectorizado ya es posible que se le pueda aplicar todo tipo efectos y modificaciones, los cuales serán presentados en el capítulo 3.3 y posterior a ello hay que nuevamente hacer tratamiento al vector de audio para que pueda ser leído por el

programa SDK de Xilinx, dentro del sintetizador de la FPGA, en donde hay que pasar todos los datos del vector a positivo, sumándole al vector de audio un 1, posterior a ello multiplicarla por 1000 y expórtalo haciendo que el programa en el software de Matlab escriba un archivo txt. (plain text). Dicho archivo txt. debe ser cambiado a .c y reubicado a la carpeta sdk, del proyecto en que se está trabajando con el sintetizador de audio en Vivado de la FPGA Zybo Z7. El procedimiento de acondicionamiento del audio se representa en la Fig. 3.8.

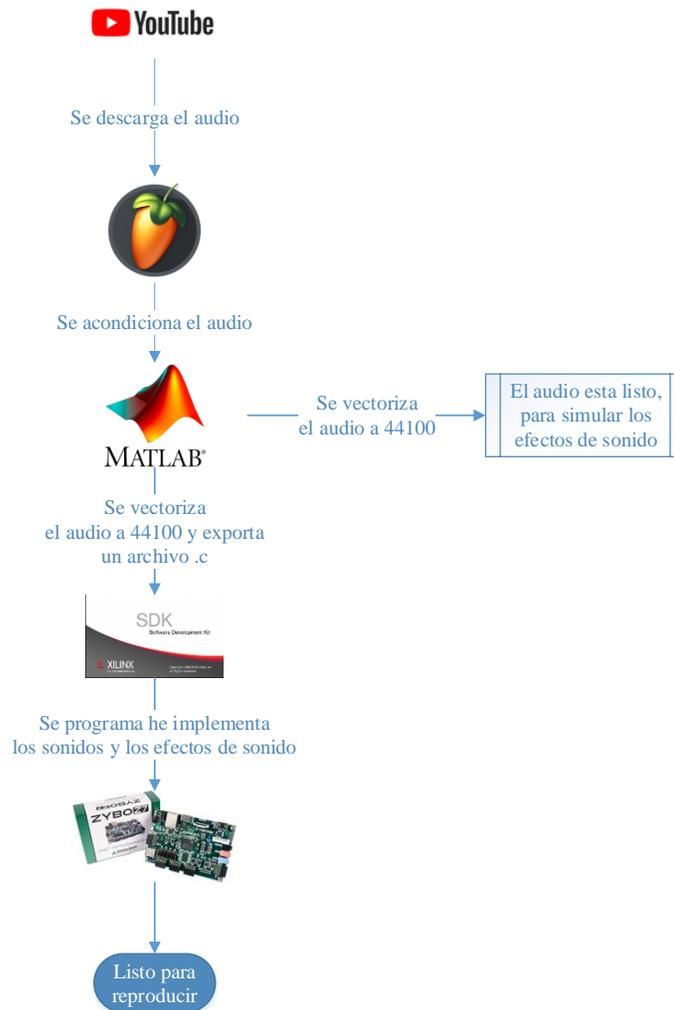


Fig.3.8. Acondicionamiento del audio para simulación e implementación.

3.3 Utilización de los efectos de sonido

En la presente sección del capítulo 3.3 se dará a conocer los efectos de sonidos elegidos para el diseño e implementación de un sintetizador de audio en una plataforma Zynq-7000. Estos corresponden a un total de 16 efectos de sonido, los cuales se pueden clasificar de tres

maneras: (I) Variación del número de vectores o multiplicados por una ganancia. (II) Son por medio de un cálculo y modificación del valor de sus vectores. (III) Mediante la aplicación de filtros FIR y IIR, los cuales se explicaron su utilización en capítulo 1.2.3.2 y su aplicación en el capítulo 4.3. Los efectos de sonido escogidos están basados en los efectos de sonidos vistos en capítulo 1.2.3, siendo estos, parte de los efectos de sonidos tocados en dicho capítulo.

Para el desarrollo de los efectos se desarrollarán en el software de Matlab y posteriormente a ello serán implementados en el sintetizador de audio cuya explicación se tocará en el capítulo 5.2. Además, solo se mostrará cómo afecta los efectos de sonido al primer sonido personalizado para los efectos de sonido, puesto que para temas de simpleza solo se realizará la variación de un sonido personalizado a través de los 16 efectos de sonidos, en vez de repetir el mismo procedimiento para todos los sonidos seleccionados, para los efectos de sonido. El sonido número uno a graficar muestra en la Tabla 3.3, del capítulo 3.3.1.

Mediante el diagrama de flujo de la Fig. 3.9 se pueden dar a conocer lo siguiente. Que la entrada al modo de efecto de sonido es mediante la activación del canal 16, la cual se logra con la combinación del primer botón de la Zybo Z7, llamado btn 0 más el botón número 16 del teclado táctil. Una vez dentro del modo de efecto de sonido, se encuentra por defecto la versión original de los 16 sonidos especiales para los efectos de sonido sin que se aplique estos. En caso de que no se aprete una combinación del botón 2 de la tarjeta Zybo Z7, llamado btn 1, volverá a la estadía de sonidos en su forma original, de los sonidos de los efectos de sonido, donde solo apretando la tecla correspondiente de 1 a 16, del teclado táctil sonará el sonido correspondiente. En caso de que no se apreté el teclado no emitirá ningún sonido. Si es que se continua en el canal 16 y se apreté una combinación de teclas, del botón btn 1 más una tecla del 1 al 16 del teclado táctil, se activara el efecto de sonido correspondiente a dicha posición. Una vez dentro del efecto de sonido y haber soltado el botón btn 1 más la tecla correspondiente al canal elegido, se puede volver a apretar únicamente una de las 16 teclas del teclado táctil y sonará los mismos sonidos de los audios especiales de los efectos de sonidos, anteriormente mencionados, pero en este caso aplicando el efecto de sonido correspondiente elegido (en caso que no se apreté el teclado táctil no se emitirá un nuevo sonido) tal como se explica en la Fig.3.9. Para cambiar el efecto siempre que se encuentre en el canal 16, hay que volver a apretar nuevamente la combinación del botón 2(Fx) más el

número del teclado táctil del 1 al 16 y en caso de querer volver a escuchar los audios originales de los sonidos de efectos de sonido, solamente hay que apretar botón 2(Fx).

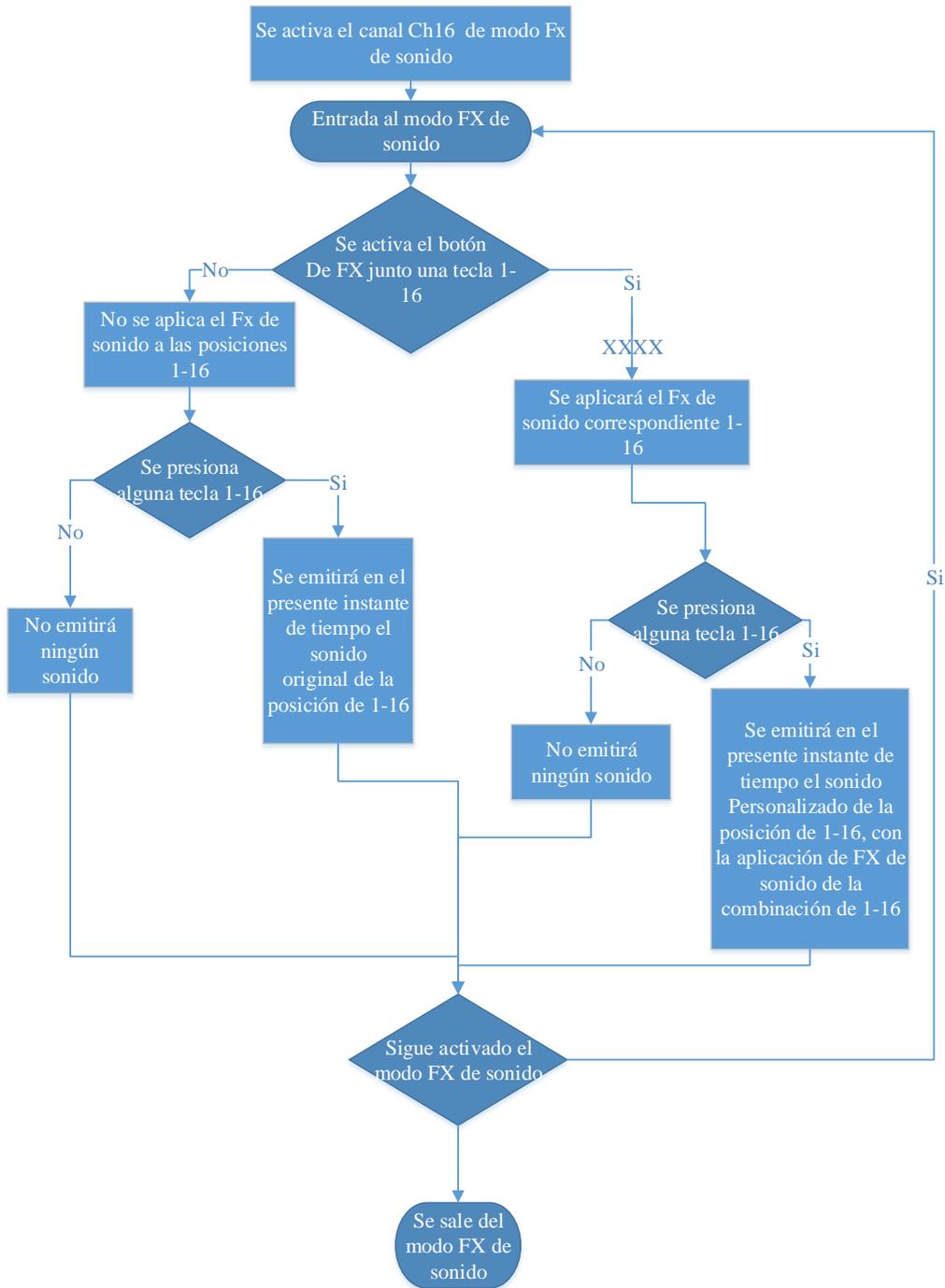


Fig.3.9. Diagrama de Flujo del modo efecto de sonido del sintetizador de audio.

Es importante nombrar que para el funcionamiento del canal 16, cuando se accione cualquiera de los sonidos de los efectos, el modo compositor dejara de sonar, hasta que termine el sonido del efecto de sonido. De igual manera para los sonidos originales de los efectos de sonido, los cuales únicamente se reproducirán si se mantiene presionado su posición del teclado táctil del 1 al 16. Los sonidos originales son aquellos sonidos que no se le ha aplicado los efectos de sonidos, y aparecen por defecto al entrar al canal 16 o ya estando dentro de este canal apretando el btn 1.

En la Tabla 3.2, se muestra los efectos de sonidos aplicados, junto a la posición correspondiente de la combinación explicada en el parrado anterior de botón 2(Fx) más el número del teclado táctil de la posición del 1 al 16. Estos efectos fueron definidos en la Tabla 2.1.

Tabla 3.2.Nombre de los efectos de sonido y su posición correspondiente.

Número de efecto	Nombre del efecto
1	Repetir por 2
2	Repetir por 4
3	Invertido
4	Fill in
5	Bouncing ball
6	Glich
7	Distorsión
8	Harlf rate
9	Feedback
10	Filtro pasa alto
11	Filtro pasa bajo
12	Filtro distorsión
13	Fm
14	Barrido baja velocidad
15	Barrido alta velocidad
16	Barrido distorsión

3.3.1 Selección de audio de los efectos de sonido

Para la selección de los audios, para los sonidos de efectos de sonido, se prefirió aplicar 16 nuevos sonidos, los cuales son diferente a los sonidos del modo patrón. La anterior decisión fue con la finalidad de apreciar de mejor manera la aplicación de los efectos de sonido en el audio, puesto que, para la selección de estos, se procuró elegir sonido más

compuestos, es decir con la inclusión de más instrumentos y diálogos. Los cuales poseen una duración de más de 0.5 segundos hasta los 5 segundos.

Para el acondicionamiento del audio antes de ser procesado y recibido por el sintetizador de audio en una plataforma Zynq-7000, se realizó el mismo trabajo que el explicado en el segundo y tercer párrafo, del capítulo 3.3.1, y vistos en la Fig. 3.8. Los audios seleccionados para ser ocupados en el modo de efectos de sonidos, se presentan en la Tabla 3.3.

Tabla 3.3. Sonidos según la posición del teclado táctil del modo de efectos de sonido.

Tecla	Nombre del sonido	Tecla	Nombre del sonido
1	Alfabeto D JoJo[43]	9	Alfabeto I JoJo´s[43]
2	Explosión [44]	10	Alfabeto K JoJo´s[43]
3	Aishite[45]	11	Alfabeto N JoJo´s[43]
4	Ora ora[46]	12	Alfabeto O JoJo´s[43]
5	Alfabeto A JoJo´s[43]	13	Alfabeto P JoJo´s[43]
6	Alfabeto F JoJo´s[43]	14	Alfabeto Q JoJo´s[43]
7	Alfabeto G JoJo´s[43]	15	Alfabeto Y JoJo´s[43]
8	Alfabeto H JoJo´s[43]	16	Kamimashita[47]

Para la realización de las gráficas de los efectos de sonido se trabajará sobre el sonido 1 de los sonidos de efecto especial. En donde para apreciar el cambio provocado por los efectos de sonido, es necesario conocer la forma de onda, de los vectores del presente audio. La cual se presenta en la Fig. 3.10.

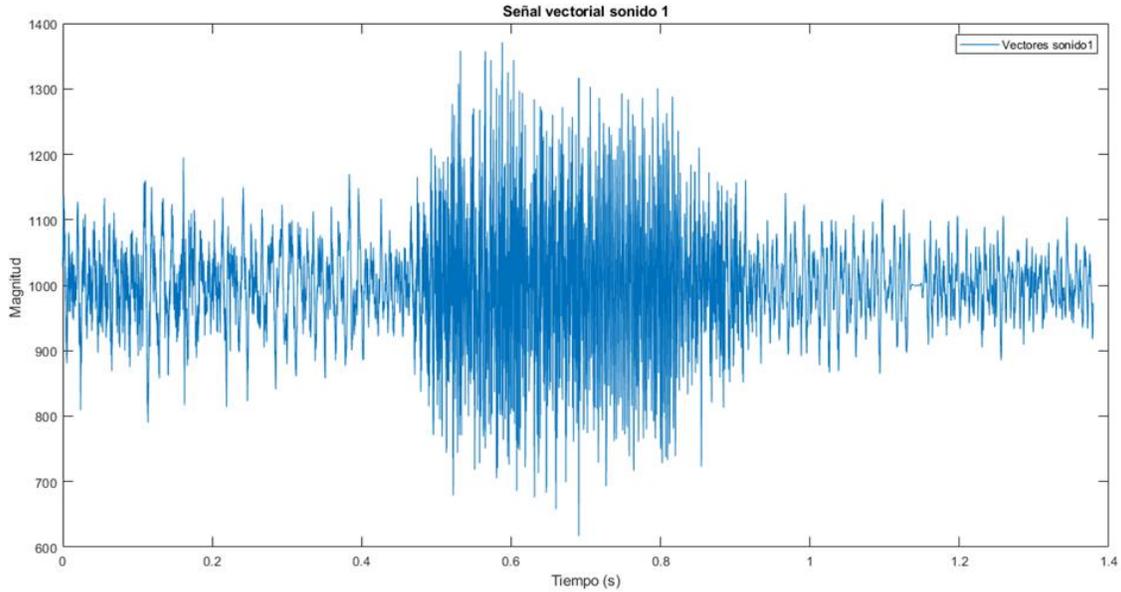


Fig.3.10. Gráfica vectorial original del sonido 1.

3.3.2 Efecto 1 Repetir por 2

El efecto 1 consiste en repetir por 2, el cual al ser activado hace que se repita dos veces más el audio. Este efecto entra en la clasificación (I), dentro de las tres clasificaciones de la sección 3.3. Esto es debido que únicamente se repite todo el vector de audio nuevamente. El diagrama de flujo del efecto 1 se muestra en la Fig. 3.11.

El diagrama de flujo del efecto 1, mostrado en la Fig. 3.11, da a conocer que antes de comenzar el efecto, pasa por el acondicionamiento del audio de la Fig.3.8. Una vez seleccionado el efecto número 1, se calcula el largo del vector, se inicia un contador igual a 0. Posterior a la primera reproducción del audio, se hace la decisión si el contador es igual a 2, en caso de que no sea igual a 2 aumenta el contador en uno y se vuelve a reproducir la pista de audio. En caso que, si sea el contador igual a 2, se procederá a terminar de reproducirse el efecto. El efecto 1 una vez concluido su reproducción, se encontrará en condiciones de reproducirse con cualquier sonido de la Tabla 3.3, siempre que se encuentre dentro del mismo efecto. En la Fig.3.12, se representa el audio original del sonido 1 versus la aplicación del efecto de repetir por dos sobre este sonido 1.

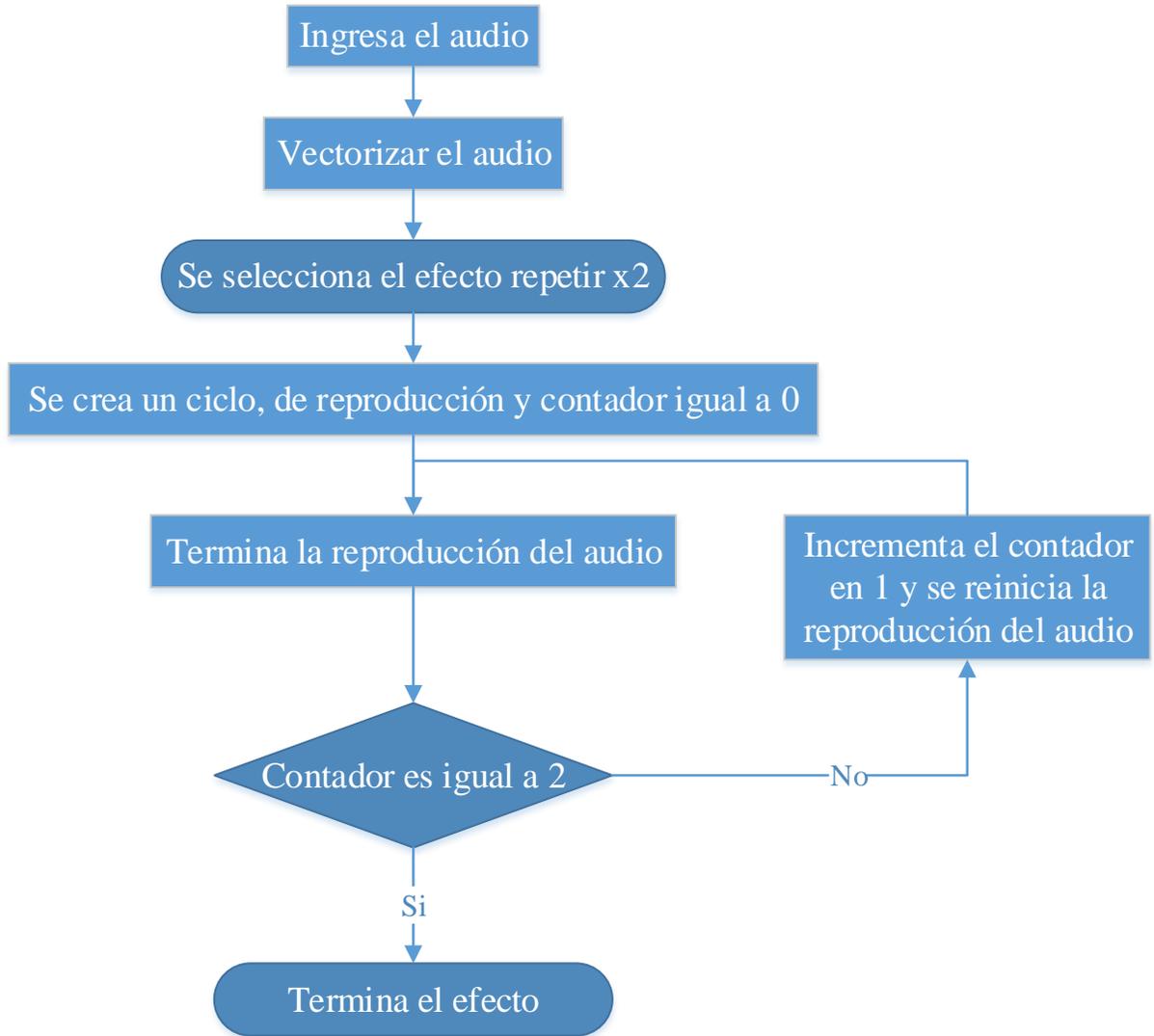


Fig.3.11. Diagrama de flujo del efecto 1.

Como se puede apreciar en la Fig. 3.12 de la aplicación del efecto vs la gráfica original. Hay un aumento de dos repeticiones más del audio original. Esto indica que la duración del audio bajo el efecto de repetir por 2 aumenta tres veces. En el cual una vez terminado la segunda repetición el efecto concluye.

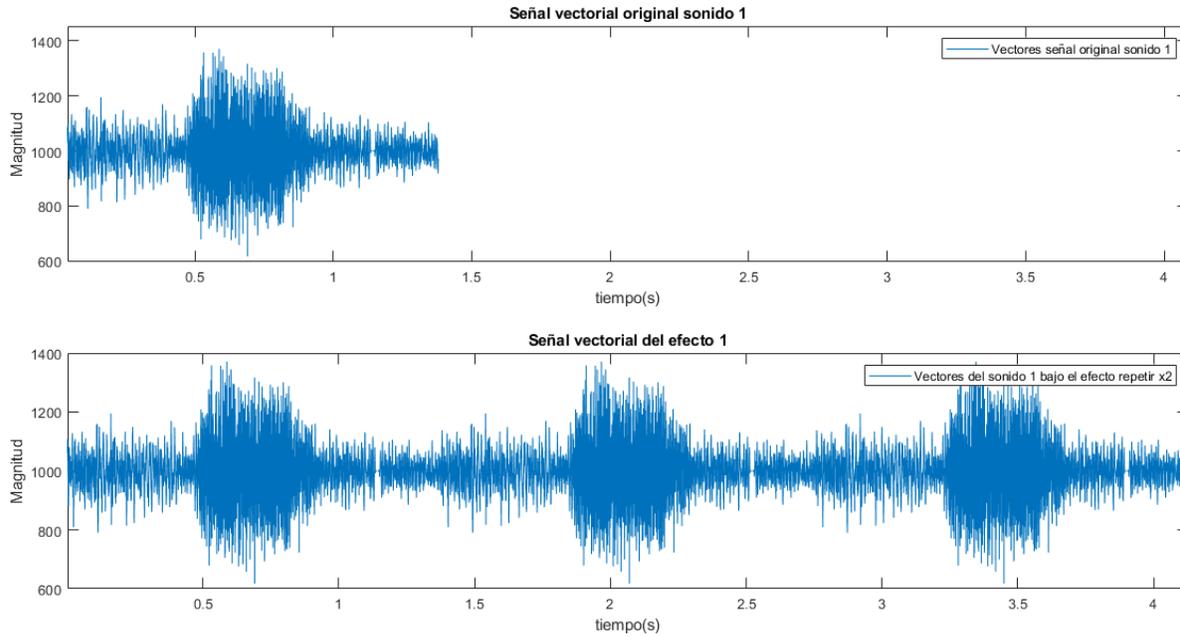


Fig.3.12. Gráficas vectoriales del sonido 1 y la aplicación del efecto 1.

3.3.3 Efecto 2 Repetir por 4

Para el efecto de sonido dos, corresponde al efecto de repetir por 4, cuyo funcionamiento es el mismo que su efecto predecesor (efecto 1), lo único que para este caso no se repite por dos, si no se repite cuatro veces. Esto indica la duración del presente efecto es 5 veces más, en comparación del sonido 1 de los efectos especiales.

En la Fig. 3.13, su lógica de funcionamiento es similar al efecto de sonido 1. Posee el mismo concepto de partida, el cual es el acondicionamiento de la señal visto en la Fig.3.8 hasta la etapa de simulación. En donde al entrar al efecto de sonido 2, se calculará el largo del vector de audio, se crean dos variables, una que lleve el largo del vector de audio y otra que sirva de contador. La variable del contador se encargará de contar las repeticiones, para realizar dicho encargo se le asigna el valor de 0. Una vez terminada la primera reproducción se realiza una condición de causa “If” la cual pedirá si contador es igual a 4, en caso que no aumentara el contador en uno y volverá a repetirse el audio, hasta que el contador sea igual a 4. En caso de que, si sea el valor del contador igual a 4, procederá a terminar el efecto de sonido. Una vez terminado y siempre que se encuentre dentro del efecto 2, se podrá volver a activar el efecto con otro audio, seleccionando una de las 16 teclas del teclado táctil.

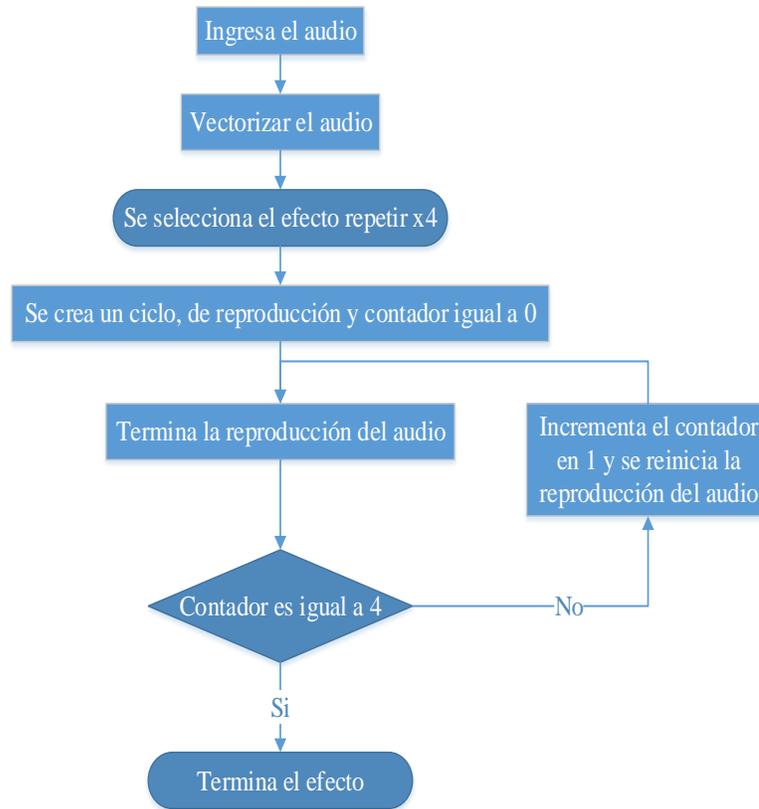


Fig.3.13.Diagrama de flujo del efecto 2.

La Fig. 3.14 corresponde a la gráfica del efecto 2, actuando sobre el sonido 1 de los sonidos de efectos especiales versus su sonido original.

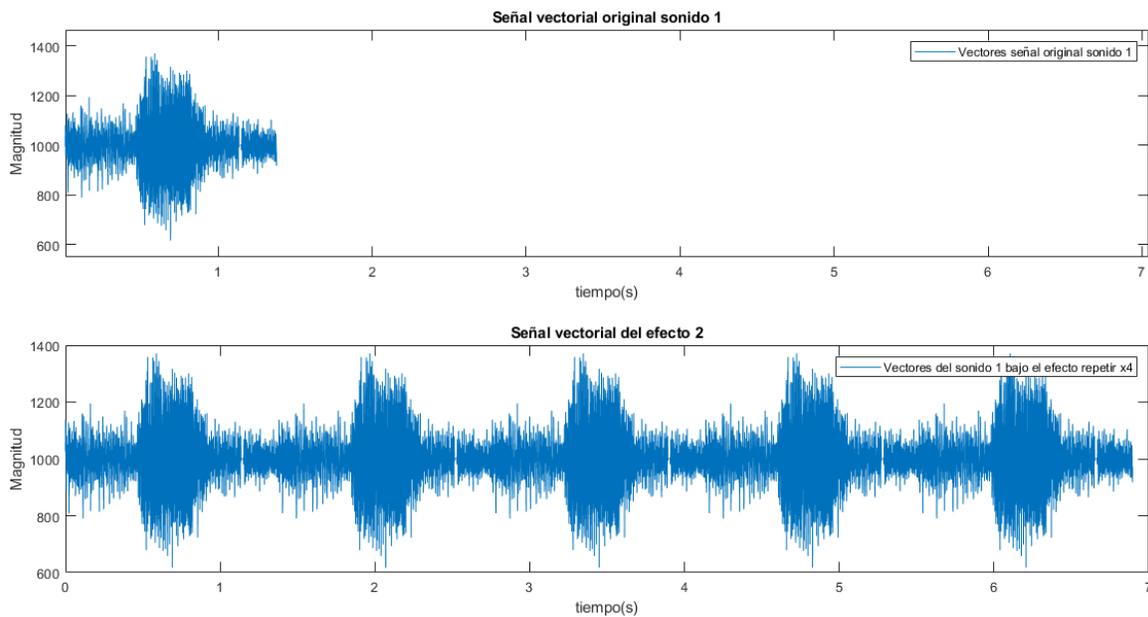


Fig.3.14. Gráficas vectoriales del sonido 1 y aplicación del efecto 2.

En comparación de la gráfica de la Fig. 3.14 de la aplicación del efecto, con la gráfica del sonido 1 original, se ve un aumento de su duración y la señal vectorizada del audio del sonido 1 se repite un total de cuatro veces más, tal como indica el efecto de repetir por 4.

3.3.4 Efecto 3 Inversión

El efecto número 3 correspondiente al efecto de inversión, posee de la primera clasificación de los efectos de la sección 3.3 párrafo uno. La razón de esta clasificación es debido que, únicamente se modifica la posición de reproducción del audio, reproduciendo el audio desde el ultimo vector hasta el primero. Lo que provoca que se perciba el audio de manera inversa, es de destacar que la duración del presente efecto es igual a su sonido original. La lógica del efecto se representa en el diagrama de flujo de la Fig. 3.15.



Fig.3.15. Diagrama de flujo del efecto 3.

En la Fig. 3.15 comienza con el acondicionamiento de la señal de audio, tal como se demuestra en la Fig. 3.8 hasta la etapa de simulación. Una vez seleccionado el efecto 3 de inversión se calcula el largo del vector de audio y posterior a ello se crea un ciclo “for” con el mismo largo del vector de audio. Dentro de este ciclo “for” se procede a reproducir el audio, desde el ultimo vector de audio hasta el primero. Una vez terminado de reproducirse terminara la aplicación del efecto 3 y se encontrara en condiciones de ser aplicado el efecto

a otro audio de los 16 disponibles (Tabla 3.3), siempre que se encuentre en el efecto 3 y que se presione alguna de las 16 teclas del teclado táctil.

El análisis de la Fig. 3.16 de la aplicación del efecto 3, frente a la señal original, se ve que mantiene el mismo tiempo de duración, pero la señal es diferente. Esto es debido a como se reprodujo el sonido del vector del sonido 1, puesto que es la misma señal, lo único que, reproducida en orden inverso, empezando por el final de la original hasta llegar al principio de esta.

En la presente Fig. 3.16, se representa la gráfica del sonido1 original versus la aplicación del efecto de inversión sobre el mismo audio.

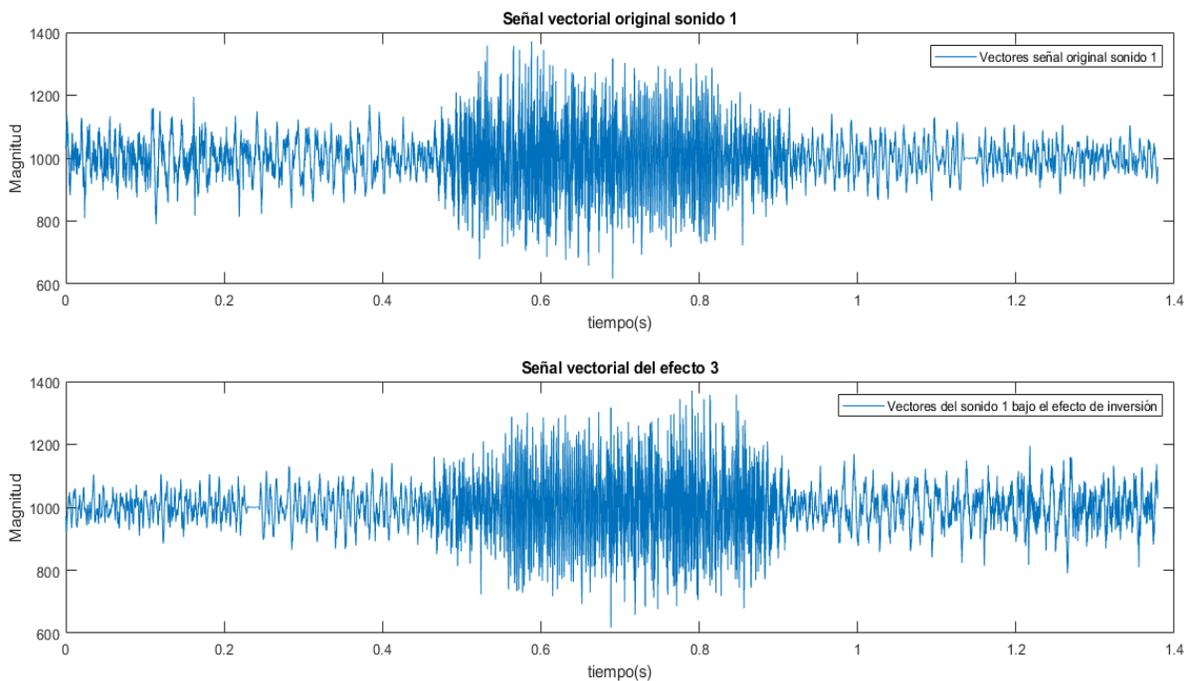


Fig.3.16. Gráficas vectoriales del sonido 1 y la aplicación del efecto 3.

3.3.5 Efecto 4 Fill in

El efecto 4 Fill in, entra en la primera categoría de efectos de la presente sección 3.3, puesto que su conformación es en base a las modificaciones del vector de audio sin la necesidad de proceder a calcular cada vector. El funcionamiento del efecto 4 consiste en agregar tiempos sin emitir audio, en estos tiempos el valor del audio será cercano a cero. Para las pruebas se consideró realizar un total de 4 interrupciones en la pista de audio elegida y la duración de estos tiempos de vacío es equivalente a un 5% en función del largo del vector de audio.

El tiempo de duración del intervalo de tiempo de la interrupción se representa en la ecuación (2.2). En dicha ecuación “Y” corresponde al tiempo de interrupción, “L” representa el largo del vector, “X” el cual indica el número de interrupción. La distribución de las cuatro interrupciones está en factor del 20% del largo del vector.

$$Y = [L \cdot X \cdot 0.2, L \cdot X \cdot 0.2 + 0.05] \quad (2.2)$$

La lógica del funcionamiento del presente efecto 3, se representa en el pseudocódigo 1

Algoritmo 1 Seudocódigo Fill in

```

1: audio[So1] = vectores de audio
2: i = 0
3: ins = ]0, 0.25]
4: x = 1
5: inicio ciclo
6: hasta i ≤ So1 − 1
7:     para i ≤ So1 · x · ins hacer
8:         reproducir audio[i]
9:     fin
10:    para i ≥ So1 · x · ins y i ≤ So1 · x · ins + 0.05 hacer
11:        No reproducir audio[i]
12:    fin
13:    para i == So1 · x · ins + 0.05 hacer
14:        x = x + 1
15:    fin
16:    i = i + 1
17: fin del ciclo
18: fin del efecto

```

En el pseudocódigo 1 se muestra la lógica del efecto 4, en el comienzo de este se selecciona uno de los 16 sonidos disponible de la Tabla 3.3, procediendo a vectorizar el audio elegido. Continuado se crea una variable X, que indica el número de interrupciones, con un valor original de 1. Posterior a ello se crea un ciclo “for”, con una duración igual al largo del vector de audio. Dentro del ciclo “for”, se realiza la emisión del audio hasta llegar al momento de la interrupción, causada por la condición “if”, mediante la ecuación (2). En caso que valor del ciclo “for” hasta el momento sea menor a la condición, seguirá reproduciendo el audio, en caso contrario se introducirá una interrupción, donde no se emitirá sonido. Terminando

esta condición “if” el valor de X aumentará en uno y se volverá a repetir el ciclo, durante las cuatro interrupciones previstas. Luego de la última interrupción, se procederá a emitir el resto de los vectores de audio faltantes. Finalizando el efecto 4 quedara en condiciones de volver a repetirse con cualquier audio seleccionado de la Tabla 3.3, siempre que se encuentre dentro de este efecto.

En la gráfica de la Fig. 3.17, se muestra la aplicación del efecto 4 Fill in sobre el audio 1 de los sonidos para los efectos de sonido.

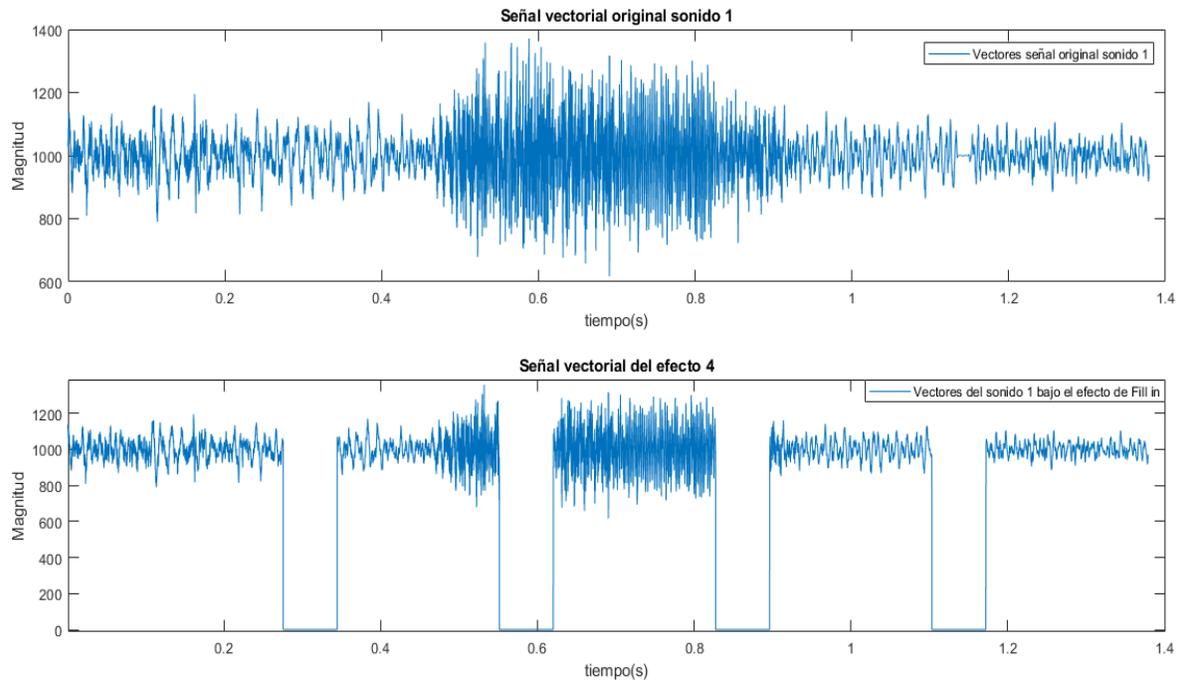


Fig.3.17. Gráficas vectoriales del sonido 1 y la aplicación del efecto 4.

Analizando la Fig. 3.17 de la aplicación del efecto 4, frente a la señal original. Se ve que mantiene la duración de su tiempo. También se logra notar las cuatro interrupciones, donde la gráfica vectorial del sonido 1 cae a cero, esto indica que no se reproduce ningún sonido. Para temas prácticos en la simulación en el software de Matlab los espacios en de las interrupciones, son remplazadas con un valor muy cercano a cero, pero en el caso de la implementación en el software de SDK de Xilinx, solo se le indicara a dijo espacio de tiempo, de las interrupciones, que no se reproduzca el vector de audio, respetando las interrupciones vistas en la Fig. 3.17

Dentro de las modificaciones de parámetros del capítulo 3.4, se encuentra la consideración del comienzo de los intervalos de vacío, los que estarán dentro del rango de > 0 y ≤ 0.25 .

3.3.6 Efecto 5 Bouncing ball

El efecto número 5 Bouncing ball posee su significado en el rebote de una pelota, por el hecho que emula la reacción de esta, en donde cada vez va disminuyendo cada uno de sus rebotes hasta dejar se saltar. Llevando la explicación del efecto 5 a su implementación indica que, poco antes de terminar de reproducirse el vector de audio este se vuelve a reproducirse nuevamente, con una magnitud de sonido inferior, pero sobre el mismo audio original. Esta repetición se puede hacer un número de n veces, hasta que el volumen del audio sea casi cero. Según la clasificación del párrafo 1 de la sección 3.3 entra en la primera categoría, puesto que solamente se reproduce la suma de un vector de audio, más el mismo vector de audio desfasado en un tiempo a elección, pero con una amplitud de magnitud cada vez menor y disminuyendo con el tiempo.

Para la aplicación del presente efecto de sonido se tomó un número de 3 rebotes, uno seguido a otro cuando se alcance un 80% del audio reproducido. Es decir, cuando el primer audio llegue al 80% sonará el primer rebote y así sucesivamente hasta acabar. En la ecuación (3) para la implementación del efecto 5 se calcula el largo total del vector de audio, desde el comienzo con el audio original hasta el término del ultimo rebote. En el cual Kr corresponde al largo total del vector del efecto 5, Lr al largo del vector de la pista de audio original y nr el número de rebotes, para un efecto de ingreso en el 80% de la reproducción del audio.

$$Kr = Lr \cdot nr + Lr \cdot (1 - 0.2 \cdot nr) \quad (3.3)$$

En el siguiente pseudocódigo 2 se representa el funcionamiento del efecto de sonido número 5.

Algoritmo 2 Seudocódigo Bouncing ball

```

1: audio[So1] = vectores de audio
2: i = 0
2: ib1 = 0, ib2 = 0, ib3 = 0
4: Porcentaje de la instancia del comienzo del rebote
3: ins = 0.8 intervalo posible ]0, 1]
4: Activación de los rebotes
4: Bo1 = 0, Bo2 = 0, Bo3 = 0
4: Ganancia de los rebotes
4: G1 = 0.8, G2 = 0.6, G3 = 0.4 decrecientes en el tiempo
7: inicio ciclo
8: hasta i ≤ So1 − 1
10:     reproducir audio[i]
9:     para i ≤ (So1 − 1) · ins hacer
10:         Bo1 = 1
11:     fin
25: i = i + 1
26: fin del ciclo
7: inicio ciclo
8: hasta ib1 ≤ So1 − 1 y Bo1 == 1 hacer
10:     reproducir audio[ib1] · G1
9:     para ib1 ≤ (So1 − 1) · ins hacer
10:         Bo2 = 1
11:     fin
25: ib1 = ib1 + 1
26: fin del ciclo
7: inicio ciclo
8: hasta ib2 ≤ So1 − 1 y Bo2 == 1 hacer
10:     reproducir audio[ib2] · G2
9:     para ib2 ≤ (So1 − 1) · ins hacer
10:         Bo3 = 1
11:     fin
25: ib2 = ib2 + 1
26: fin del ciclo
7: inicio ciclo
8: hasta ib3 ≤ So1 − 1 y Bo3 == 1 hacer
10:     reproducir audio[ib3] · G3
25: ib3 = ib3 + 1
26: fin del ciclo
27: fin del efecto

```

Pasando primero el audio por su proceso de acondicionamiento para la simulación de la Fig.3.8 se procede a aplicar el seudocódigo 2. En el seudocódigo 2 al principio se calcula el largo del vector del efecto, por medio de la ecuación (3.3), y con el cual se crea un ciclo “for” para reproducir dicho audio. El tiempo de reproducción para cada sonido es igual a la duración del sonido original, comenzando su reproducción se preguntará si se llegó al 80% del vector de audio. En caso que no, se continuará reproduciéndose el primer sonido. En caso que si ha llegado al 80% comenzará a reproducirse el primer rebote y en paralelo terminara de sonar el sonido original. Cuando el primer rebote llegue al 80% de su reproducción, comenzará a reproducirse el segundo rebote y en paralelo terminara se sonar el primer rebote. Este proceso se repetirá hasta que llegue el tercer rebote, el cual solo tiene que terminar de reproducirse, para que se finalice el efecto. Al igual que en los demás efectos una vez concluido este quedara en condiciones para aplicar el efecto 5 a otro audio de la Tabla 3.3, siempre que se presioné alguna de las 16 teclas del teclado táctil.

En la gráfica del quinto efecto sobre el sonido 1 de la Tabla 3.3 de la Fig. 3.18 versus a su señal original, se aprecia la simulación de las explicaciones de la presente sección. Se muestra el primer sonido al principio, para luego cercano a su final (a su 80%) comience el primer rebote con una magnitud menor y así sucesivamente hasta terminar con los tres rebotes. En el cual cada disminución de la magnitud del volumen del rebote es un 20% de la señal anterior en función del número de rebotes. Finalmente, tal como se había explicado en el primer párrafo la señal de rebote va disminuyendo progresivamente a través que avanza el tiempo, tal como si fuese un eco.

Dentro de las modificaciones de parámetros, que se explicaran en el capítulo 3.4, está el momento de ingreso del rebote, el cual puede ser un porcentaje de ingreso del audio que variara entre $]0,100]$ %.

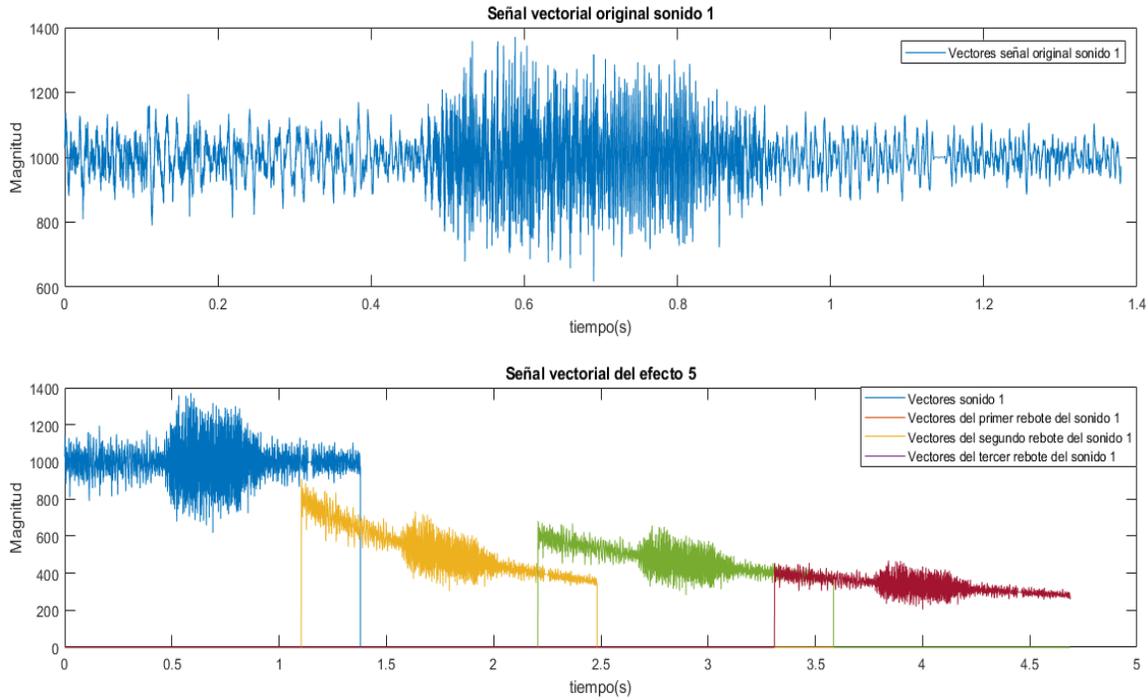


Fig.3.18. Gráfica vectorial del sonido 1 y sus respectivos rebotes, con la aplicación del efecto 5.

3.3.7 Efecto 6 Glich

El efecto número 6 Glich hace referencia a los errores que se generaban en los equipos antiguos de los computadores, en donde a su funcionamiento normal entraba una falla. Para la realización del efecto 6 se elige un nuevo audio, el cual se intercalará entre la reproducción de los audios de la Tabla 3.3. El nuevo audio poseerá una magnitud de volumen un poco menor, una duración menor a 0.2 segundos y llevará el nombre de Glich[39]. Para la realización del efecto el nuevo archivo entrara un total de cinco veces, separados por un 20% de la duración del audio elegido, donde X representa el número que le corresponde a dicha interrupción. Para la duración del presente efecto aumentará en comparación del audio original, puesto que se le suma el tiempo de las repeticiones del archivo Glich, como resultado el vector de audio del efecto 5 estará dictada por la ecuación (3.4). Finalmente, el presente efecto entra en la primera clasificación de los efectos del párrafo 1 de la sección 3.3, por el hecho que solo se modifica el orden de sus vectores y en el preciso caso se da la inclusión de un nuevo sonido.

$$Lgr = N + Ng \cdot 5 \quad (3.4)$$

En la ecuación (3.4), la variable Lgr corresponde al largo del vector del efecto 6, el valor N es el largo vectorial del sonido elegido, de la Tabla 3.3 y Ng es el largo vectorial del archivo de audio Glich, para la implementación esta multiplicado por 5, puesto que se repetirá un total de 5 veces el audio de Glich.

En el pseudocódigo 3 se muestra la lógica del efecto 6, que a diferencia de los demás efectos hay que preparar dos acondicionamientos de audio, uno para la pista del audio de la Tabla 3.3 y otro para el archivo de Glich. Comenzando el efecto hay que calcular el largo del vector de audio elegido y el largo del vector de audio del archivo Glich, posterior a ello se calcula un ciclo “for”, que tendrá un largo igual a lo dictado por la ecuación (3.4). Además, se designa una variable “ins” la cual puede variar entre 0 y 0.2, correspondiente a la introducción del Glich que puede fluctuar entre el 0% al 20% del largo de las pistas de audio de la Tabla 3.3. Por último dentro de las variables se designa un contador X igual a 1, que tendrá la tarea de contar la cantidad de veces que se ha introducido el audio de Glich hasta el momento, con un máximo de simulación e implementación de 5. Dentro del ciclo de reproducción del audio del efecto 5, comenzará el audio seleccionado de la Tabla 3.3 hasta llegar a su 20% en donde se introducirá el primer Glich. Una vez terminado el audio del archivo Glich, la posición de su vector volverá a cero y aumentará el contador X en uno. Frente a esto si X es igual a 5, indicara que ya se han reproducido todos los sonidos Glich. En caso que no se cumpla esta condición, procederá a continuar reproduciendo el audio original desde la posición que quedo anteriormente, para luego volver a repetirse el ciclo, solo que cambiando la instancia de entrada del vector de audio del Glich. En caso que la condición “if”, de X es igual a 5 se cumpla, procederá a terminar de reproducir el audio de sonido original y poniendo fin al efecto de sonido 5.

Algoritmo 3 Seudocódigo Glich

```
1: audio[So1] = vectores de audio
2: i = 0
3: ins = ]0, 0.2]
4: x = 1
5: Glich[G1] = vectores del Glich
6:  $L = So1 + 5 \cdot G1$ 
7: inicio ciclo
8: hasta  $i \leq L - 1$ 
9:   para  $i \leq (So1/5) \cdot ins$  hacer
10:    reproducir audio[i]
11:   fin
12:   para  $i \geq (So1/5) \cdot x \cdot ins + G1 \cdot (x - 1)$ 
13:   y  $i \leq (So1/5) \cdot x \cdot ins + G1 \cdot x$ 
14:   Hacer
15:    reproducir Glich[ $i - ((So1 \cdot x/5) + G1(x - 1))$ ]
16:   fin
17:   para  $i \geq (So1/5) \cdot x \cdot ins + G1 \cdot x$ 
18:   y  $i \leq (So1/5) \cdot (x + 1) \cdot ins + G1 \cdot x$ 
19:   Hacer
20:    reproducir audio[ $i - G1 \cdot x$ ]
21:   fin
22:   para  $i == (So1/5) \cdot (x + 1) \cdot ins + G1 \cdot x$  hacer
23:     $x = x + 1$ 
24:   fin
25:    $i = i + 1$ 
26: fin del ciclo
27: fin del efecto
```

En la gráfica de la Fig. 3.19, se representa la aplicación del efecto 6 sobre el sonido 1 de la Tabla 3.3, en donde en comparación de la gráfica de la Fig. 3.10, se ve un aumento del tiempo de duración, además de la segmentación del audio original, en el cual entre sus separaciones va incluido el sonido de Glich. El audio del Glich se representa con una magnitud de volumen inferior, al del audio original del sonido 1.

Dentro de las modificaciones que se verán en el capítulo 3.4, con respecto al presente efecto, se encuentra el tiempo de inclusión del sonido de Glich, en donde este puede variar entre 0% y 20% de la duración de la pista del audio original, siempre que se tenga como consideración un número de 5 repeticiones del audio Glich.

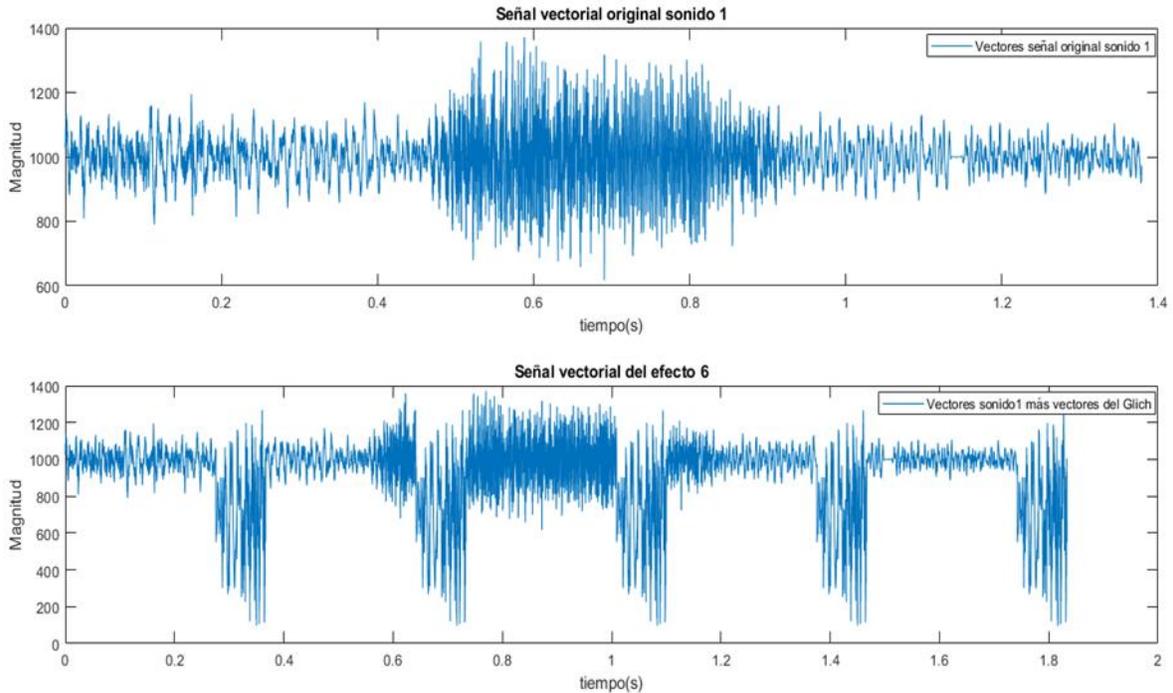


Fig.3.19. Gráficas vectoriales del sonido 1 y los vectores del Glitch, con la aplicación del efecto 6.

3.3.8 Efecto 7 Distorsión

El efecto número 7 corresponde a la aplicación de distorsión en los vectores de audio, por medio de la modificación de cada vector. El efecto 7 se logra con la multiplicación de cada vector por una ganancia aleatoria entre los límites de Y_1 e Y_2 , (0.8 y 1 respectivamente) y la suma de un valor aleatorio entre los límites estimativos de X_1 y X_2 (-20 y 50 respectivamente). La clasificación del presente efecto es de clase 2, del párrafo uno de la sección 3.3, puesto que se modifican todos los vectores de audio, para generar uno nuevo, mediante el cálculo de este.

En el diagrama de flujo de la Fig. 3.20, se representa la lógica del funcionamiento de dicho efecto.

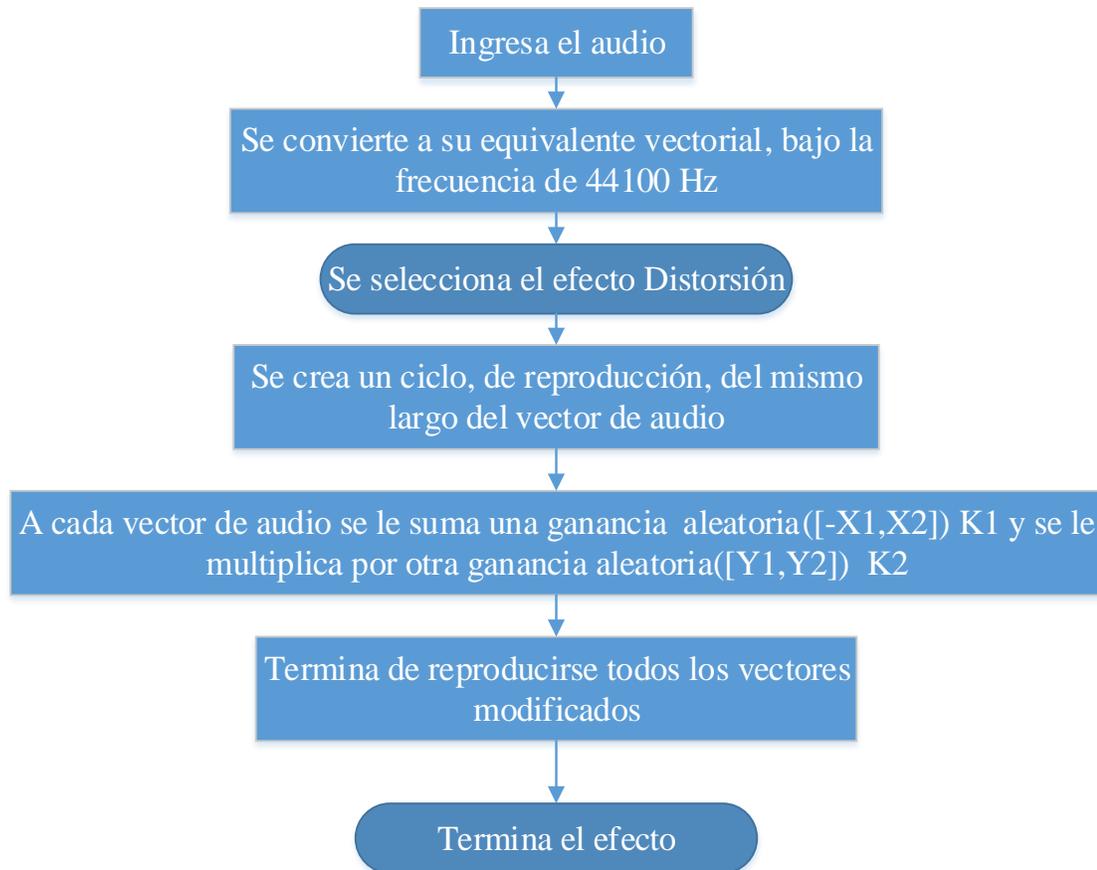


Fig.3.20. Diagrama de flujo del efecto 7.

En la Fig. 3.20 se realiza el proceso de acondicionamiento de la Fig.3.8 hasta la etapa de simulación. Posterior a ello a cada vector del audio, del sonido elegido de la Tabla 3.3, se procede a multiplicarlo por una ganancia Y , para luego sumarle otra ganancia X . Concluyendo el efecto se reproduce el nuevo audio modificado. Quedando listo para aplicar el efecto en otro audio de la Tabla 3.3, siempre que se encuentre dentro del efecto 7.

En la Fig. 3.21, se representa la gráfica vectorial del audio 1 original versus la aplicación del efecto 7 sobre este. En donde se destaca que se conserva el tiempo de reproducción y en parte el patrón de sus vectores, pero la gráfica de la señal se encuentra totalmente distorsionada en comparación de la señal original. Lo cual, permite deslumbrar que, la efectividad del efecto deseado de distorsión, el cual consiste en distorsionar la señal original.

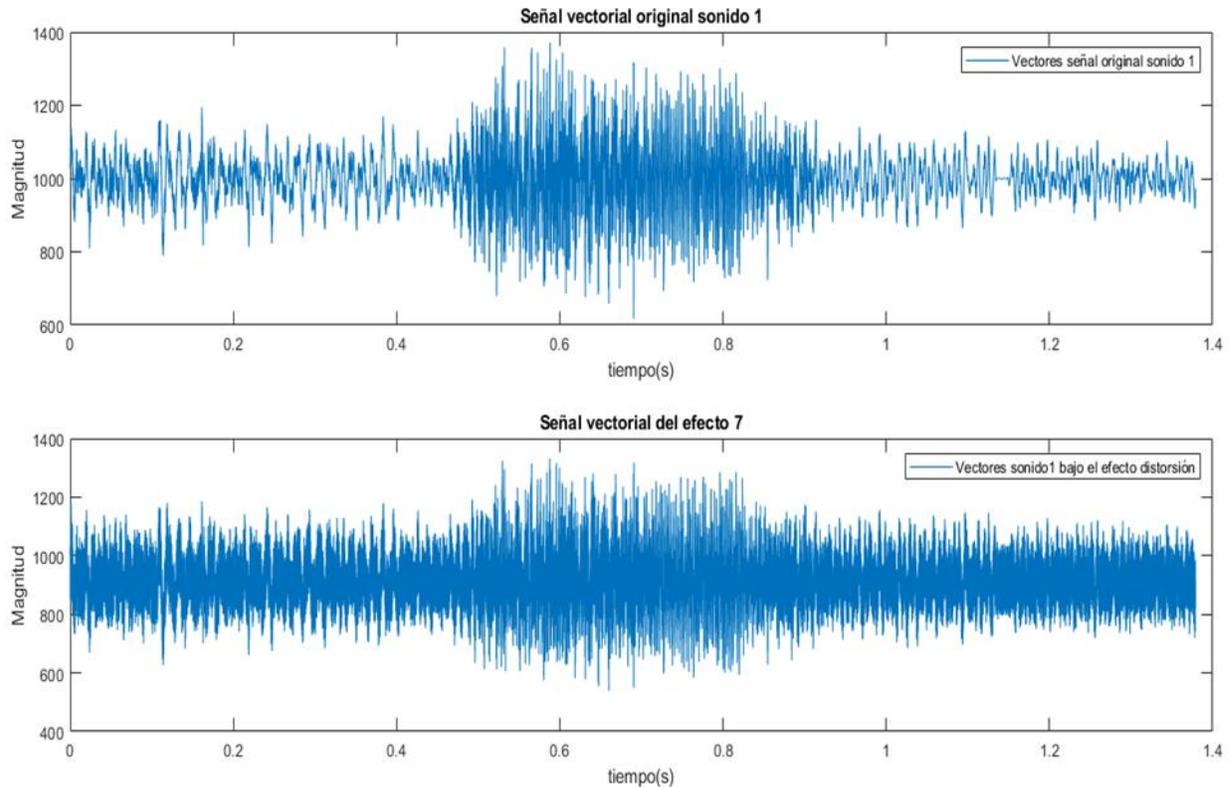


Fig.3.21. Gráfica vectorial del sonido 1 con la aplicación del efecto 7.

3.3.9 Efecto 8 Half rate

El efecto 8 del Half rate, su nombre hace referencia a que a la pista de audio se le divide a la mitad su velocidad de reproducción. Para generar dicho efecto sin tener que cambiar la frecuencia de muestreo, es necesario aumentar al doble el número de vectores, del audio a aplicar el efecto (sonidos de la Tabla 3.3). Esta solución es debido que al disminuir la velocidad a la mitad da como resultado que el tiempo en reproducirse se duplique, lo cual es equivalente a aumentar al doble el número de vectores del audio a muestrear. Por medio de este procedimiento es que el efecto 8 entra en la segunda clasificación de efecto, del apartado uno de la sección 3.3, por el hecho que se tiene que calcular los vectores para poder tener el efecto de reducir la velocidad a la mitad.

La lógica del diagrama de la Fig. 3.22 es primero pasar por la etapa de acondicionamiento de la Fig.3.8, hasta la etapa de simulación. Posterior a ello se activa el efecto 8, se calcula el largo del vector del audio elegidos, de la Tabla 3.3 y se crea dos contadores, el contador B encargado del vector del audio original y A que su valor será el

doble del largo de B. Una vez comenzado el ciclo de reproducción de audio, si A es impar, se reproducirá el vector de audio original correspondiente a la posición B y se suma en uno el contador de A. Cuando A sea un valor par, el audio a reproducir será el promedio de la posición del vector original de B más B+1, terminando ambos contadores aumentaran en uno. Este procedimiento seguirá intercalándose hasta que el contador A llegue al doble de largo que el máximo del contador B. Es por medio de este proceso de intercalación que se aumenta el número de vectores, al doble y por ende se disminuye la velocidad a la mitad.

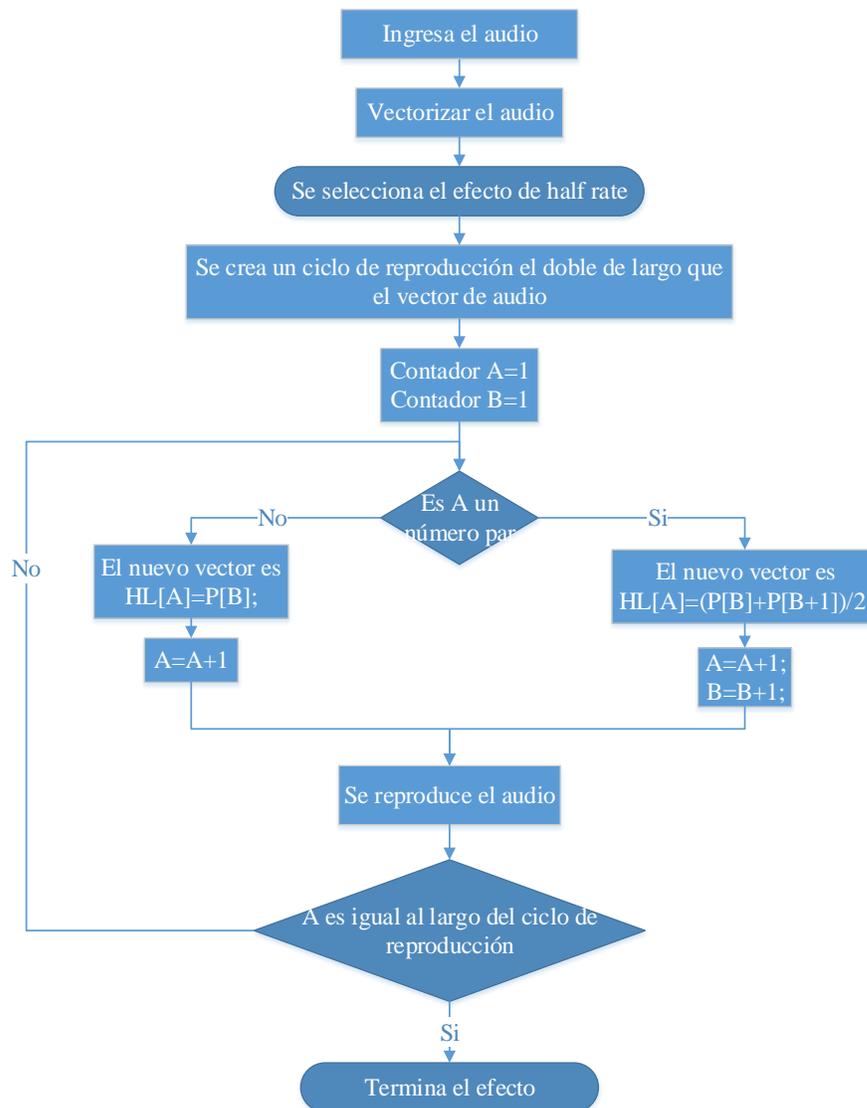


Fig.3.22. Diagrama de flujo del efecto 8.

En la gráfica de la Fig. 3.23 se muestra el efecto del filtro número 8 sobre el sonido 1. Es mediante la presente gráfica que se lograr comprobar que, en comparación a su sonido

original, el tiempo aumento el doble, pero se conserva el mismo patrón de audio, solo que en el presente caso es el doble de grande.

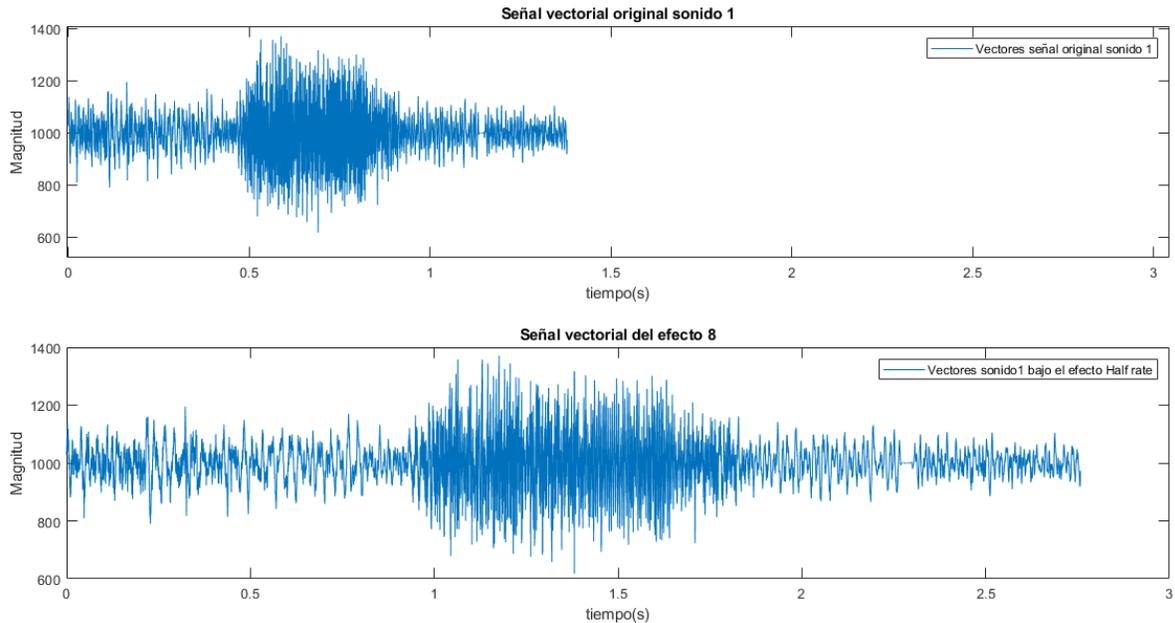


Fig.3.23.Gráfica vectorial del sonido 1 con la aplicación del efecto 8.

3.3.10 Efecto 9 Feedback

El efecto número 9 consiste en la aplicación de Feedback, el cual consiste en crear una retroalimentación en el audio. Para ello en el siguiente efecto se usara el concepto de Feedback negativo[48], tal como se muestra en la Fig.3.24, donde se explica mediante un diagrama en bloque la estrategia de Feedback, la que consiste en calcular la ecuación inversa del sistema 1 por el sistema 2 y volverla a reingresar a la entrada del sistema, pero esta introducción se encontrara desfasada en el tiempo de la reproducción del audio original y se restara a este. Los sistemas ocupados se encuentran en la ecuación (3.5) y (3.6), para el sistema 1 y 2 respectivamente y su respuesta inversa se muestra en la ecuación (3.7). La clasificación del efecto número 9 entra en la segunda categoría, del párrafo uno de la sección 3.3, puesto que hay que calcular todo un nuevo vector de audio en el momento que se reingrese, por medio de restar el vector de audio modificado por el sistema inverso desfasado al vector de audio original.

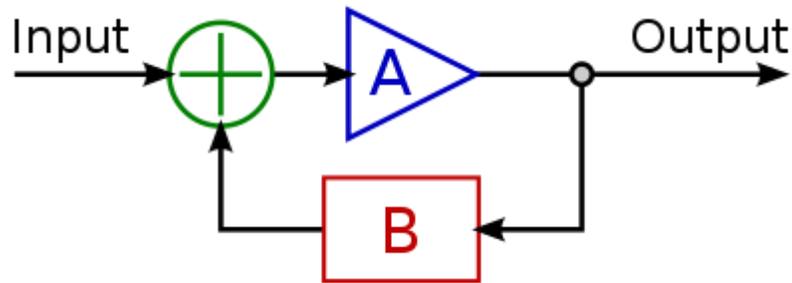


Fig.3.24.Diagrama de bloque de la estrategia de Feedback negativo[49].

$$Sys1 = \frac{2s^3 + 5s + 1}{s^2 + 2s + 3} \quad (3.5)$$

$$Sys2 = \frac{5s + 10}{s + 10} \quad (3.6)$$

$$SysFeedback^{-1} = \frac{2s^3 + 25s^3 + 51s + 10}{11s^3 + 57s^3 + 78s + 40} \quad (3.7)$$

Para comprobar la fiabilidad del sistema inverso elegido, se calculó sus polos y ceros, mediante la función zero y pole de Matlab[50], dichos valores se encuentran en la Tabla 3.4, Posterior a ello se graficó su plano polo y zero en el plano continuo. En donde, se ve que los polos son negativos, por lo cual el sistema es estable, pero posee una parte imaginaria lo que lo hace oscilante, antes que se estabilice. Para el caso de los zeros son negativos, lo que provoca que la señal sea continua. La gráfica del plano continuo de polos y zeros, se muestra en la Fig.3.25.

Tabla 3.4.Polos y zeros resultante del sistema aplicando Feedback negativo.

Polo	Valor	Zero	Valor
1	-3.4189	1	-10
2	-0.8815 + 0.5354i	2	-2.2808
3	-0.8815 - 0.5354i	3	-0.2192

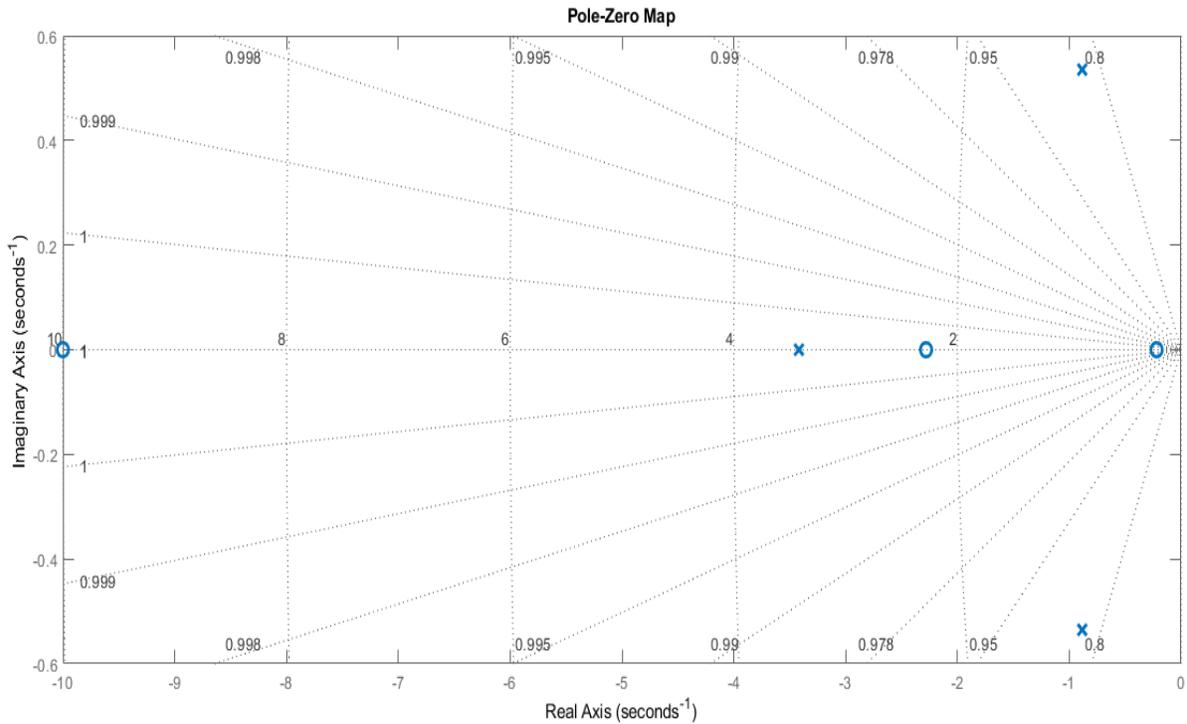


Fig.3.25. Diagrama polo(X) zero(O) del sistema del Feedback negativo.

Para lograr el efecto de sonido de retroalimentación en donde se hay que reingresar el audio modifica, en el audio original con un leve desfase, se procedió al nuevo vector modificado por el Feedback negativo restarlo al audio del vector original. El audio modificado, pasado por el Feedback se reingreso cuando la reproducción del audio original llegue a un 20% de su tiempo reproducido.

En el diagrama de flujo de la Fig. 3.26, comienza con el acondicionamiento de la señal de audio. Partiendo el efecto se calcula el largo vector de audio L y empieza a realizarse dos trabajos en paralelo, el primero es la reproducción del vector de audio original. El segundo es la preparación del nuevo vector de audio, mediante la estrategia de Feedback negativo, del sys 1 y sys 2, el cual modificara el vector de audio original. Finalmente llegando al k% de la reproducción ingresara el nuevo audio modificado restando el valor del vector de audio original, causando el efecto de sonido de Feedback. Una vez terminado quedara listo el efecto para ser ocupado nuevamente, con otro sonido de la Tabla 3.3.

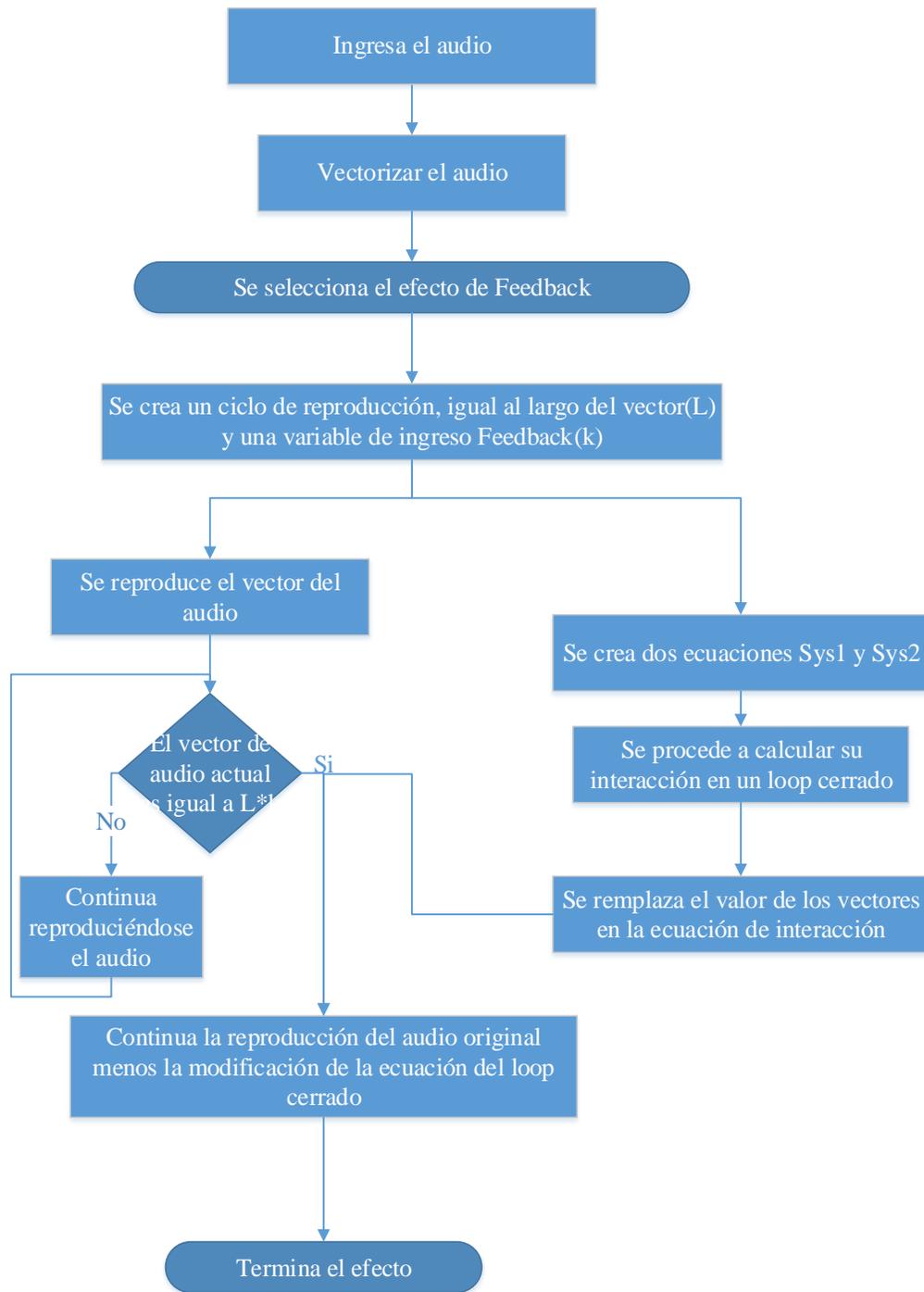


Fig.3.26. Diagrama de flujo del efecto 9.

En la Fig. 3.27, se muestra la aplicación del efecto 9 sobre el audio de sonido 1. En donde es capaz de ver que la silueta y tiempo del audio original se mantiene. Pero se ha modificado

el vector de audio, dando en ocasiones pick de alzas del volumen del audio. Esto provoca que, aunque posean la misma base el sonido ya sea diferente.

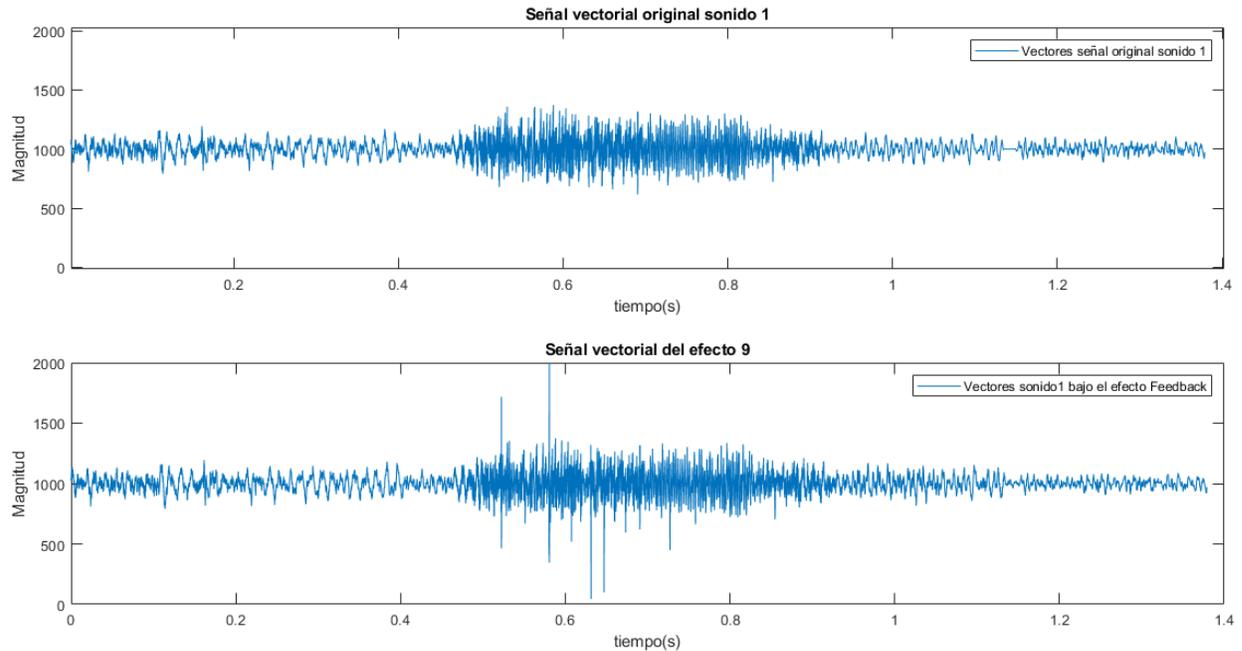


Fig.3.27. Gráfica vectorial del sonido 1 con la aplicación del efecto 9.

3.3.11 Efecto 10 Filtro pasa alto

Para la simulación del filtro 10, el cual consiste en un filtro pasa alto, se ocupó la estrategia de filtrado del filtro FIR, el cual se fundamentará mejor en el capítulo 4.3. Tal como indica el nombre del efecto, permite discriminar las frecuencias de la señal de audio, permitiendo solo pasar las frecuencias altas, en función de la señal de corte elegida. Puesto que el filtrado provocara una nueva serie de vectores de sonido, el efecto 10 entraría en la tercera clasificación, de las clases de efecto del párrafo uno de la sección 3.3, por el uso de filtrado, ocupar un número de n coeficientes y ocupar de manera constante valores anteriores del vector de audio, en función de la cantidad de coeficientes ocupados.

Para el uso del filtrado es importante la consideración del número de coeficientes, puesto que, a mayor número de coeficientes ocupados, es mayor la precisión del filtrado. Tal como se muestra en la Fig. 3.28, la que corresponde a la gráfica de respuesta de filtrado pasa bajo a una frecuencia de 3000 Hz, en función del número de coeficientes ocupados. Para la implementación se ocuparán un número de 20 coeficientes. La cantidad de coeficientes elegidos posee un valor capaz de cumplir el objetivo de filtrado, aunque como todo filtro igual puedes pasar pequeños vestigios de frecuencias bajas.

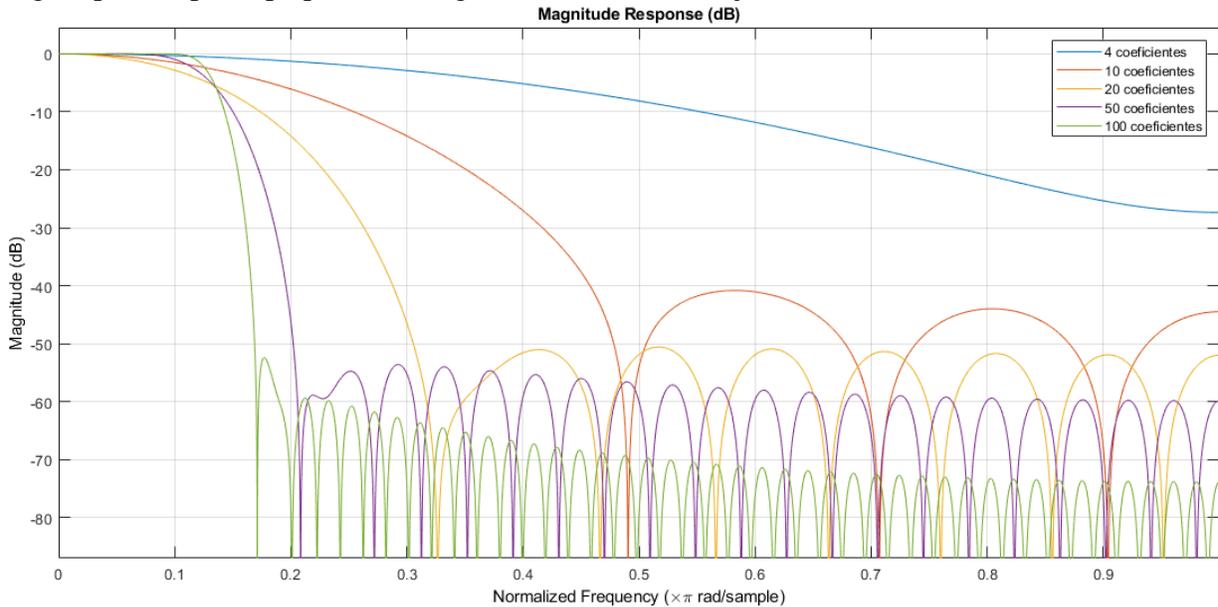


Fig.3.28. Respuesta de filtrador de un filtro pasa bajo, de una frecuencia de corte de 3000 Hz, en función del número de coeficientes ocupados[51].

Mediante el diagrama de flujo de la Fig. 3.29, se logra comprender que primero antes de comenzar el efecto, hay que preparar el vector de audio, al igual que en todos los efectos, pero para todo lo que es filtrado, hay que preparar los n coeficientes a utilizar, en función de la frecuencia de corte a ocupar. Una vez comenzado el efecto se calcula el largo del vector de audio y se procede a crear el nuevo vector de audio a reproducir, el cual se calcula según la ecuación (4.11). En la ecuación (4.11) esta ira creciendo en función del número de coeficientes ocupados, en donde la ecuación se mantendrá de manera constante una vez la posición del vector de audio alcance el número de coeficientes ocupados. Una vez terminado de reproducir el vector de audio pasado por el efecto de filtrado pasa alto, se encontrará listo para ser aplicado a otro audio de la Tabla 3.3, siempre que se encuentre en el modo del efecto 9.

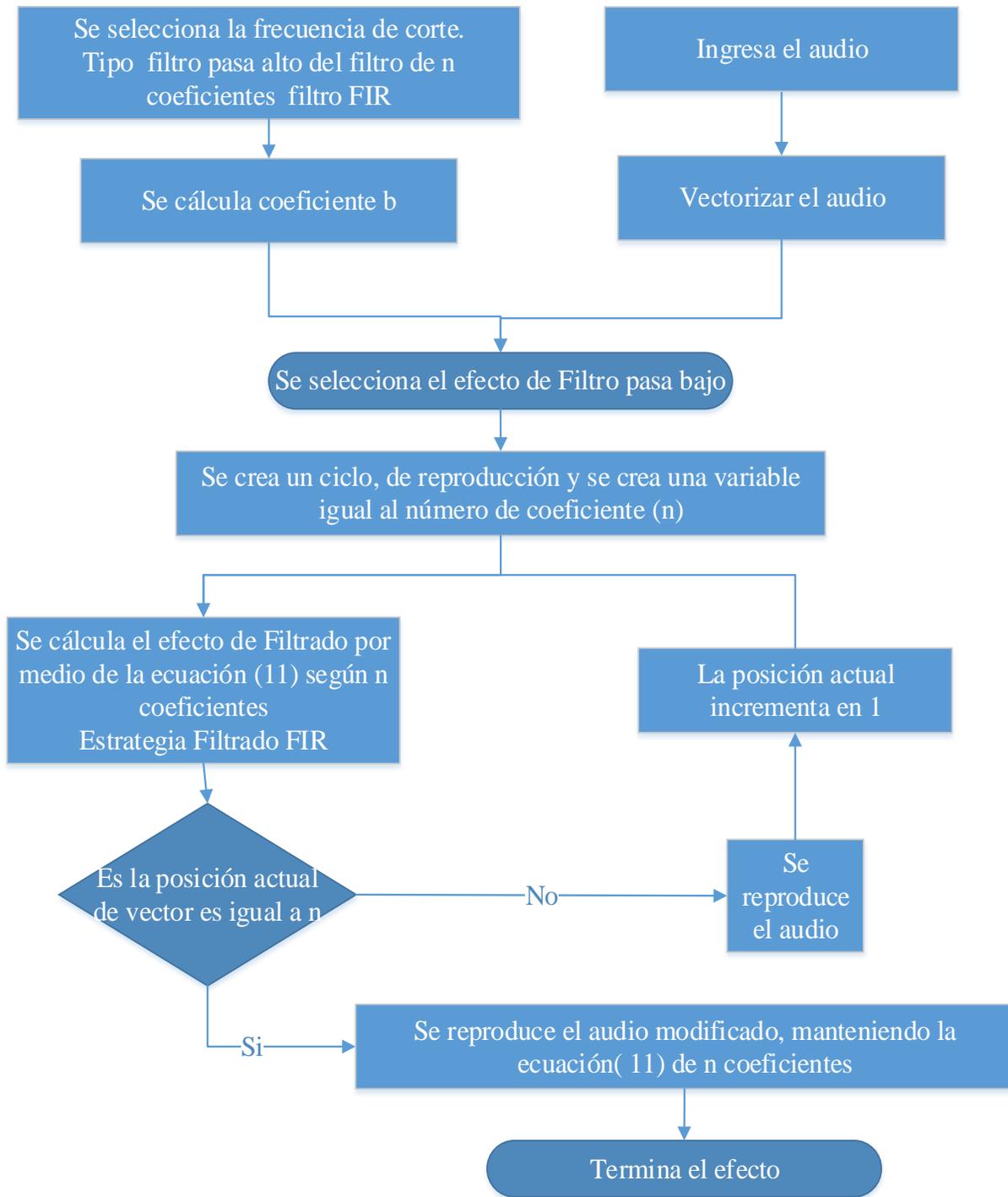


Fig.3.29. Diagrama de flujo del efecto 10.

En la Fig. 3.30, se muestra el efecto de filtrado pasa alto en el audio 1 de la Tabla 3.3. En donde hay una atenuación en su principio y en su final en la magnitud de su volumen. Además de una completa modificación del vector de audio, dando a ver el efecto del filtrado.

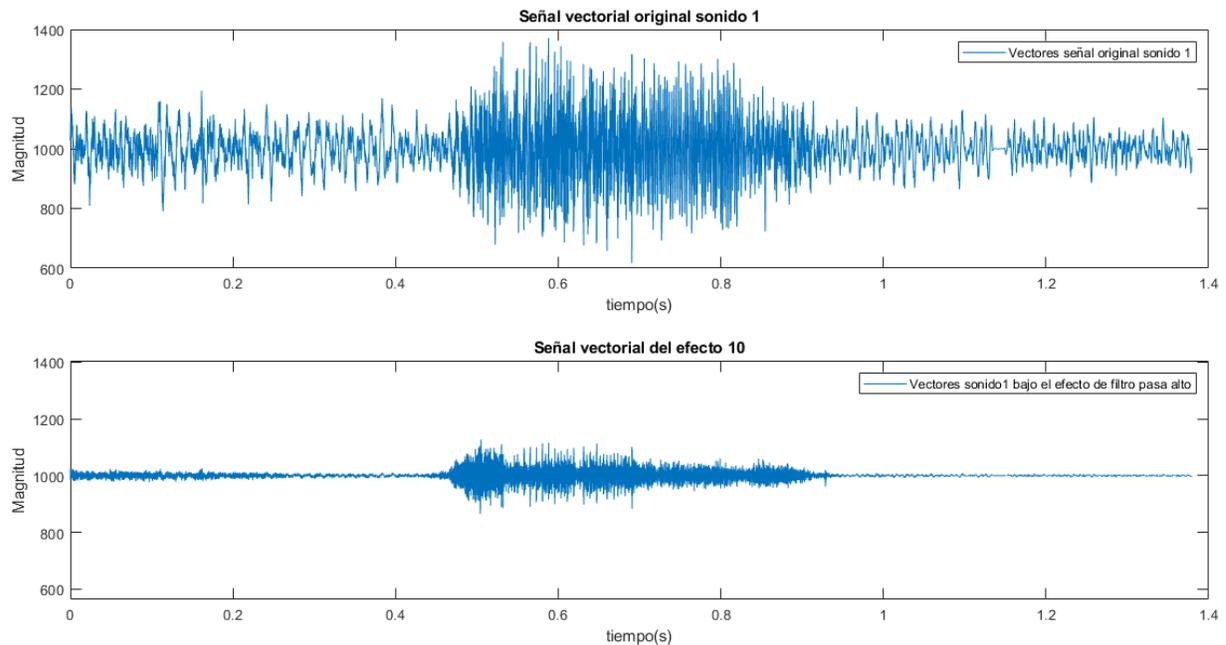


Fig.3.30. Gráfica vectorial del sonido 1 con la aplicación del efecto 10.

En la Fig.3.31, se muestra la eficiencia del filtrado ocupado, frente al espectro de la frecuencia del audio original, por medio de la gráfica de transformada de Fourier. En donde pasan mínimas frecuencias por debajo los 434 Hz y sobre esta frecuencia se un mayor filtrado del espectro de audio, hasta cercano los 3000 Hz. Llegando a los 3000 Hz deja pasar todo el espectro de las frecuencias superiores. En el presente filtrado si se hubiera aumentado el número de coeficientes la calidad del filtrado aumentaría, pero en términos de implementación cumple con su propósito.

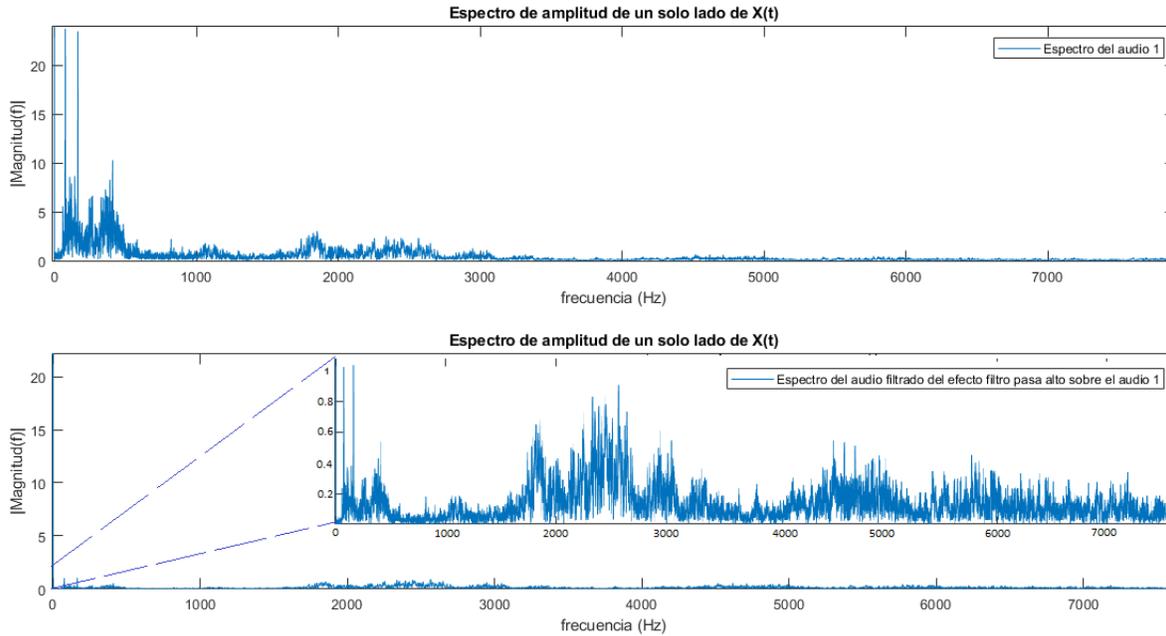


Fig.3.31. Transformada de Fourier de la Fig.3.30.

3.3.12 Efecto 11 Filtro pasa bajo

La simulación del efecto de sonido 11, corresponde al filtro pasa bajo, el cual ocupara la estrategia de filtrado de los filtros FIR. El presente efecto consiste en crear un filtro que solo deja pasar frecuencias bajas, en función de la frecuencia de corte elegida. Por el nivel de complejidad de cálculo el presente efecto entra en la tercera clasificación por las mismas razones, que el filtro anterior (efecto 10). Para las pruebas se utilizó un número de coeficientes de 20 y la frecuencia de corte será de unos 1000 Hz.

En la Fig. 3.32, se muestra el diagrama de flujo del efecto de sonido 11. El cual cumple la misma estrategia, que los diagramas de filtro. La preparación vectorial del audio a reproducir y el cálculo de coeficientes, según la frecuencia de corte elegida ocupando la función “fir1” la cual se fundamentará su explicación en el capítulo 4.3. Comenzando el efecto se empieza a modificar el vector de audio elegido, aplicando la ecuación (4.11), la cual ira incrementando en función de la posición y el número de coeficientes, hasta que llegue a la posición 20, donde se aplicaran los 20 coeficientes y la ecuación se mantendrá de manera constante. Terminando de reproducirse el audio pasado por el filtro pasa bajo, se encontrará listo para ser ocupado por otro audio, de la Tabla 3.3, siempre que se encuentre en el modo del efecto 11.

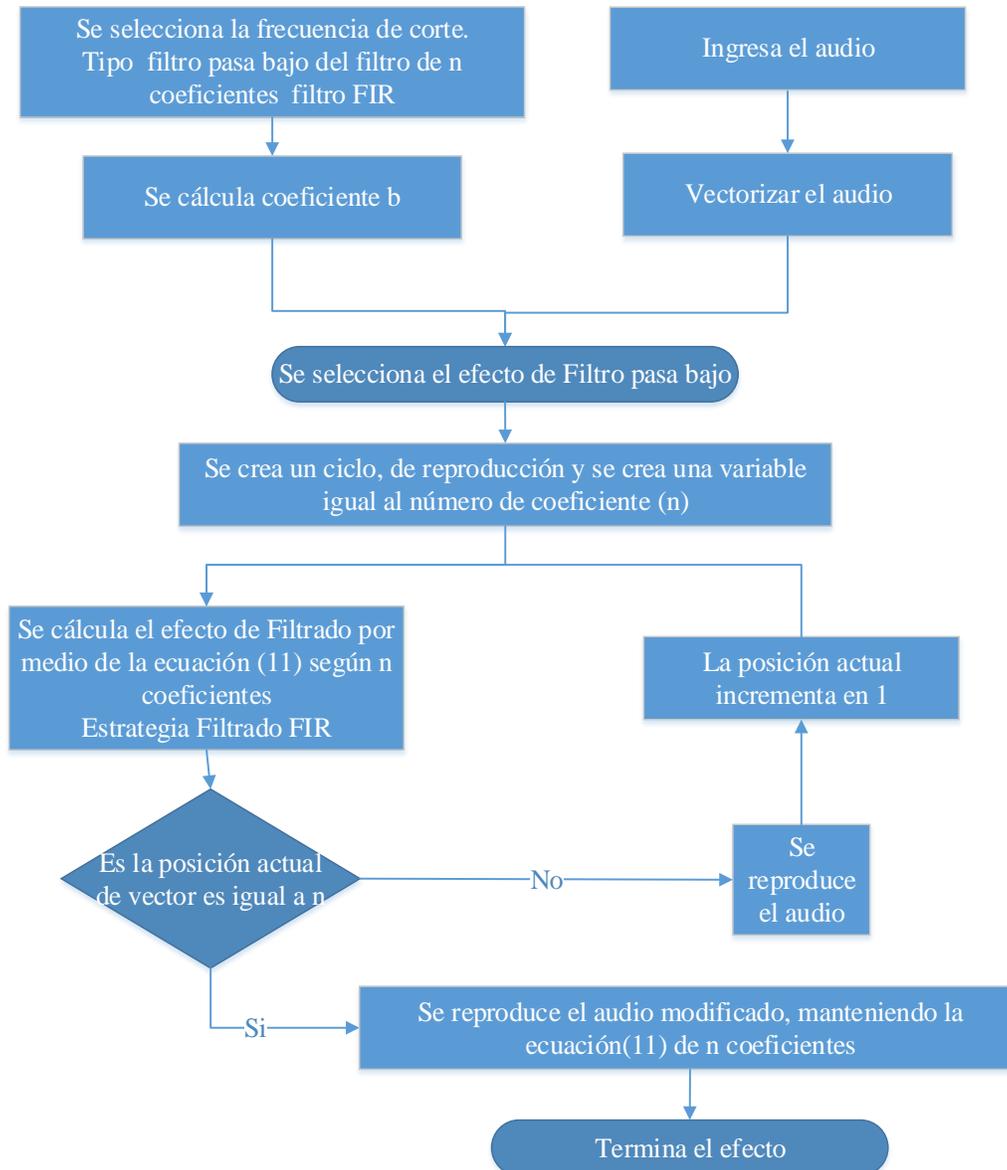


Fig.3.32. Diagrama de flujo del efecto 11.

En la gráfica vectorial de la Fig. 3.33, se muestra la aplicación del efecto 11, sobre el audio original y su señal original. En el presente caso las magnitudes del volumen del audio no son tan comprimidas, es, por el contrario, que se muestran más desorganizado, lo cual

indica que solo está pasando los sonidos pertenecientes a un espectro de audio de baja frecuencia. Pero la gráfica del efecto 11 sigue manteniendo la silueta del audio original.

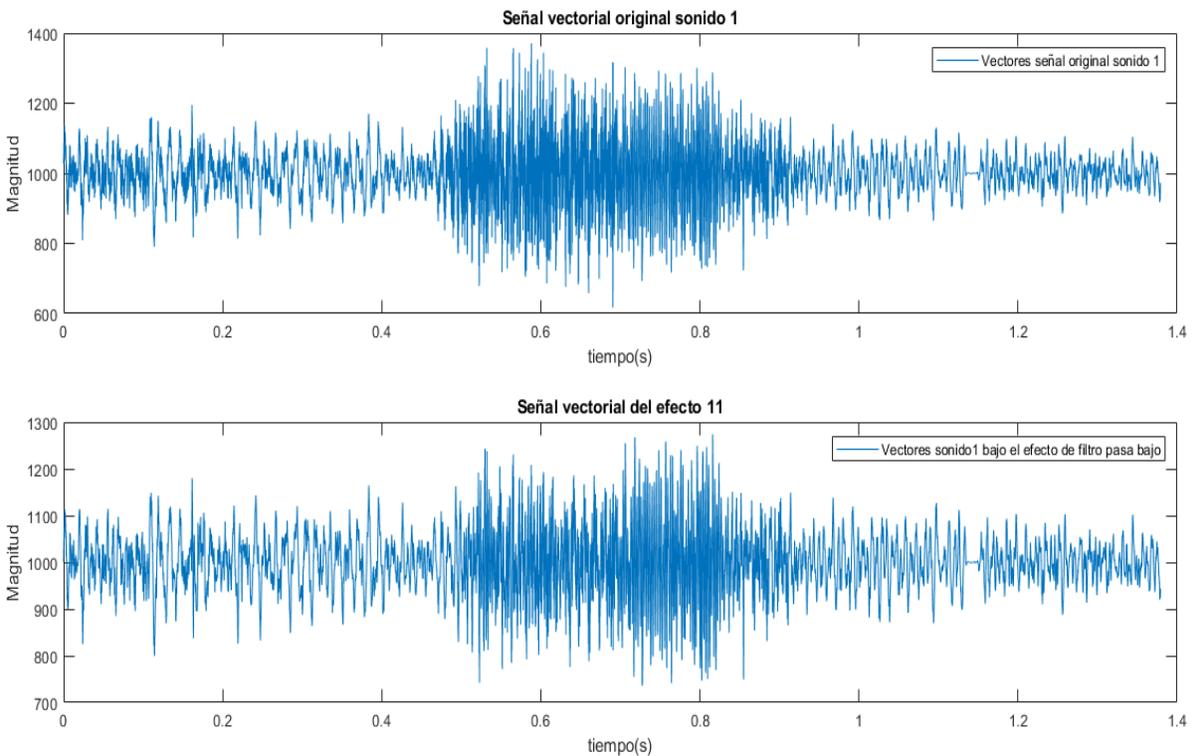


Fig.3.33. Gráfica vectorial del sonido 1 con la aplicación del efecto 11.

Para comprobar la efectividad del filtrado pasa bajo, es necesario ocupar al igual del capítulo anterior, la transformada de Fourier, la cual se representa en la Fig. 3.34. En donde a diferencia de la transformada de Fourier de la gráfica Fig. 3.28, se nota aún más el efecto de filtrado. Puesto que para el filtro pasa bajo, deja pasar casi todo el espectro de las frecuencias inferiores de 1000 Hz. Superior a la frecuencia de 1000 Hz comienza a disminuir la magnitud de las frecuencias hasta casi desaparecer, las frecuencias mayores a la frecuencia de corte elegida.

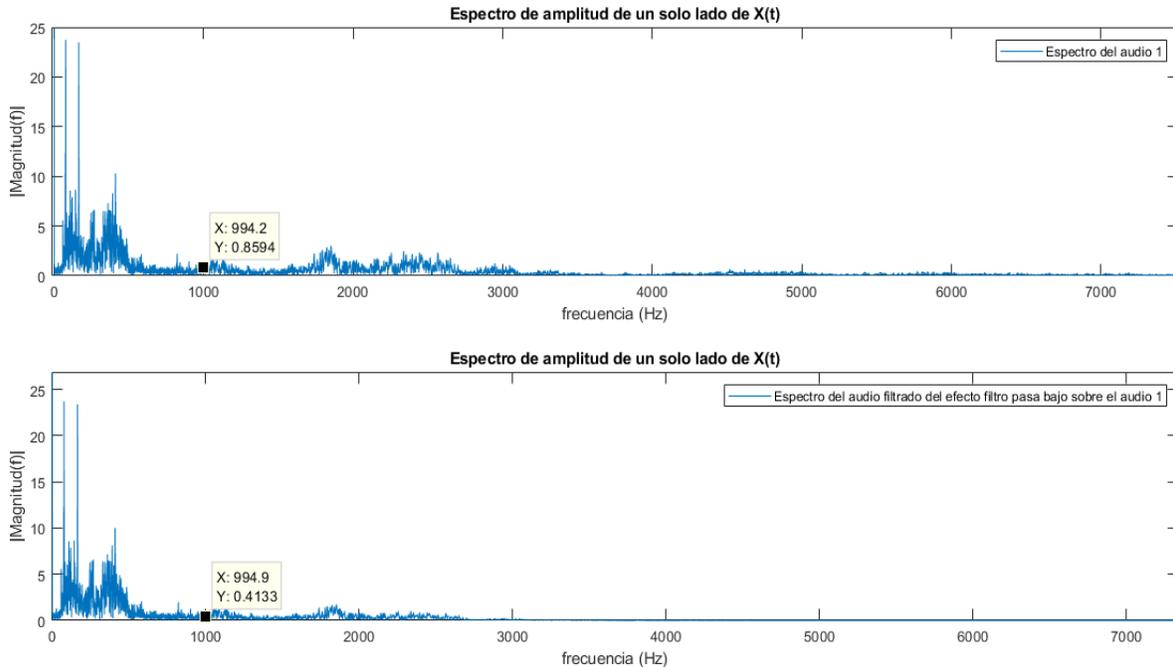


Fig.3.34. Transformada de Fourier de la Fig.3.33.

3.3.13 Efecto 12 Filtro Distorsión

El efecto número 12 correspondiente al filtro distorsión, el cual es un filtro común, al igual que el efecto 7 de distorsión, se le suma una ganancia de pequeña magnitud, con el fin de hacer una variación en el vector de audio y a la vez se le aplica el efecto de filtrado. El filtro a ocupar es la estrategia de filtro FIR, y será un filtro pasa banda de 20 coeficientes, con una frecuencia de corte entre 1900 Hz hasta los 9000 Hz y la ganancia a sumar será un 1% del valor del vector correspondiente. Dentro de los riesgos de ocupar el filtro distorsión es que no se verá reflejado en su estado puro el efecto del filtrado, por ende, la transformada de Fourier no cortará a la perfección el espectro de la frecuencia esperada.

En el diagrama de flujo anterior de la Fig. 3.35, se muestra el mismo procedimiento que en los diagramas de filtrado anterior, antes de comenzar el efecto es necesario preparar el vector de audio y calcular los coeficientes, en función de este caso a sus dos frecuencias de corte. Comenzando el efecto se calcula el largo del vector original y se crea un ciclo en donde se realizará el filtrado, según la ecuación (4.11). En dicha ecuación ira creciendo en medida que aumenta la posición del vector, hasta llegar a su máximo de coeficientes, desde ese punto se mantiene la ecuación. Ya obtenido el valor del vector del cálculo de filtrado, se

le agregara una ganancia aleatoria, igual al 1% el valor del vector de audio. Finalmente se reproduce el efecto de filtrado junto a su ruido incluido. Terminando la reproducción de sonido del vector de audio modificado, se encuentra listo para ser ocupado nuevamente el efecto 12 con otro sonido de la Tabla 3.3.

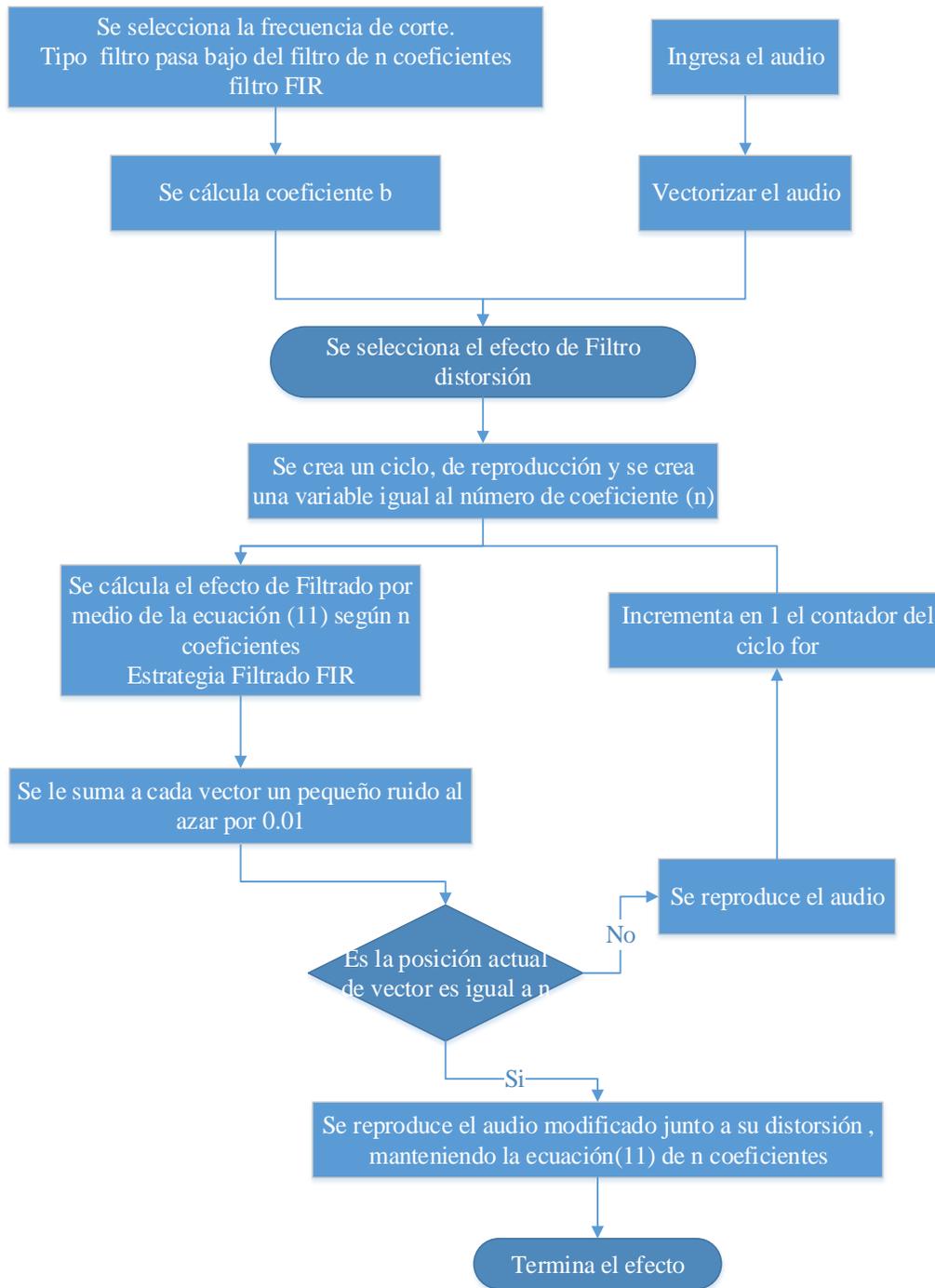


Fig.3.35. Diagrama de flujo del efecto 12.

En la gráfica vectorial de la Fig. 3.36, se muestra el efecto del filtro pasa banda, más su distorsión agregada. En donde no es tan comprimida como en la gráfica del efecto 10, pero tampoco es tan variantes en su magnitud de vectores, como en el efecto 11. Esto es debido a que como es un filtro pasa banda, mantiene un ancho de banda medio entre los dos efectos de filtrado usados, además que posee su leve variación causada por la inclusión de la distorsión.

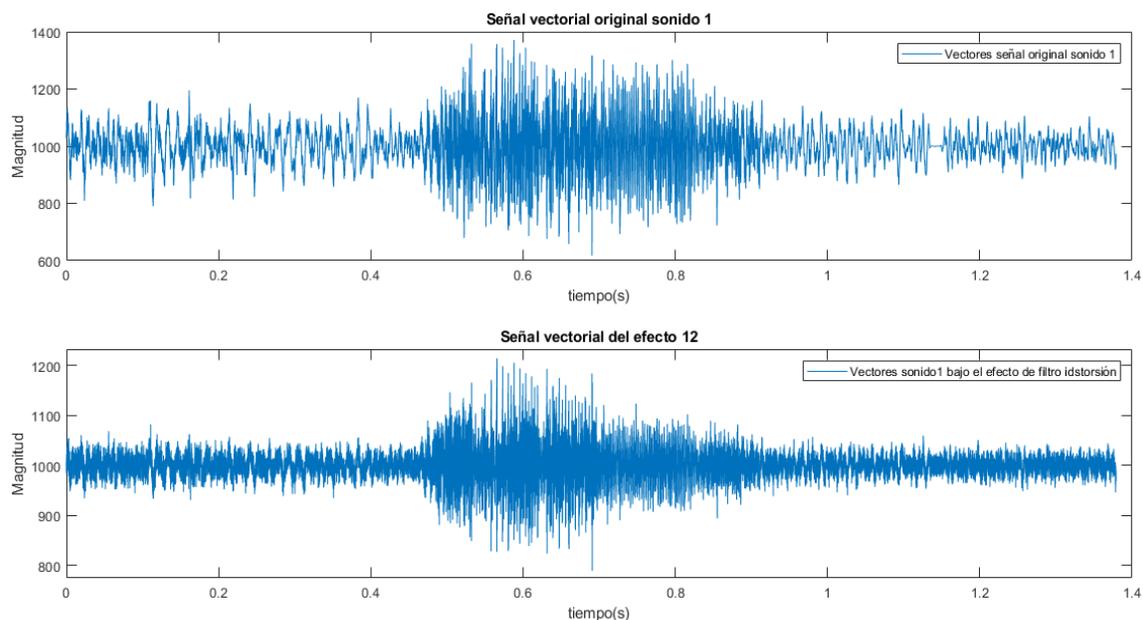


Fig.3.36. Gráfica vectorial del sonido 1 con la aplicación del efecto 12.

En la siguiente Fig. 3.37. se muestra la transformada de Fourier de la Fig. 3.36. La cual posee un comienzo similar a la transformada de Fourier, anteriormente vista del filtro pasa alto. En donde posee un comienzo pasando leves magnitudes de frecuencias, hasta aproximadamente los 500 Hz. En la Fig. 3.37 cercano a la frecuencia de corte de 1900 Hz aumenta la magnitud de frecuencia permitida, hasta llegar a su otro límite de 9000 Hz. Dentro de la variación del efecto 12 frente a las otras estrategias de filtrado, se ve un aumento de la magnitud de frecuencias altas por la aplicación de distorsión.

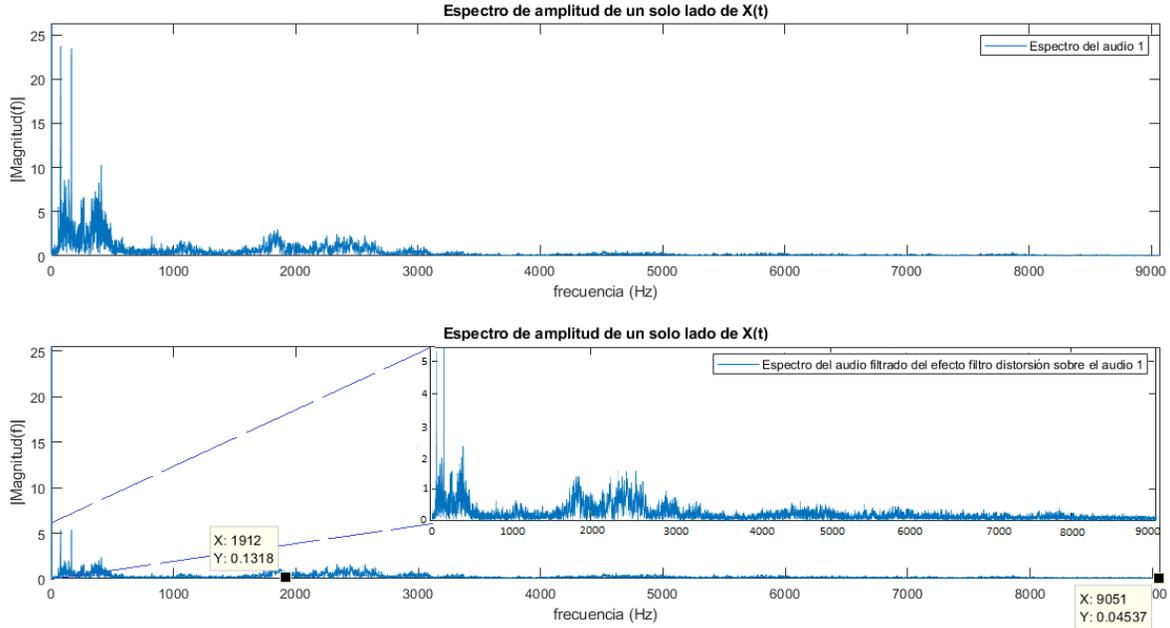


Fig.3.37. Transformada de Fourier de la Fig.3.36.

3.3.14 Efecto 13 Fm

El efecto número 13 corresponde a Fm (frecuencia modulada), el funcionamiento del presente efecto es pasar el vector de audio por el mismo acondicionamiento y tratamiento de una señal al ser transmitida por una señal Fm. Para ello es necesario pasar el vector de audio por medio de una moduladora, esto es mediante la ecuación (3.10). En donde al vector de audio se multiplica por el tratamiento de la frecuencia modulada (3.10), la que incluye la señal moduladora (3.8) y la señal portadora (3.9). Para el respectivo caso del efecto 13 se ocupará un módulo de 8 y una frecuencia moduladora de 40 Hz. Por la cantidad de cálculos necesarios para modular la señal vectorial, el presente efecto entra en la segunda clasificación de los efectos del párrafo 1 de la sección 3.3

$$S_m = \sin(2 \cdot \pi \cdot f_m \cdot t_i) \quad (3.8)$$

$$S_c = y(i) \cdot \cos(2 \cdot \pi \cdot f_c \cdot t_i) \quad (3.9)$$

$$SF_m = y(i) \cdot \cos(2 \cdot \pi \cdot f_c \cdot t_i - (m \cdot \cos(2 \cdot \pi \cdot f_m \cdot t_i))) \quad (3.10)$$

En las ecuaciones (3.8), (3.9) y (3.10) las variables representan a “fm” la frecuencia de modulación de 44100, “ti” es el tiempo que le corresponde a cada vector de audio de la pista de audio, “fc” la frecuencia de corte, m el valor de la moduladora e “y(i)” corresponde al valor del vector de audio.

En el diagrama de flujo de la Fig. 3.38 se muestra el funcionamiento del efecto 13. En donde es necesario definir todas las variables, nombrada en el párrafo anterior, junto con vectorizar el audio elegido. Comenzando el efecto se calcula un ciclo “for” con el mismo largo del vector de audio y en cada instancia ir calculando el nuevo vector de audio, con la aplicación del efecto de Fm, ocupando la ecuación (3.10). Finalizando el efecto, ya habiendo reproducido todo el vector de audio modificado, el efecto se encontrará listo para ser reutilizado en otro sonido de la Tabla 3.3, siempre que se encuentre dentro del modo efecto 11.

En la Fig. 3.39, se muestra las gráficas de la señal portadora, la cual es la modificación del vector de audio, ante la ecuación (3.9). La señal moduladora, la cual es la frecuencia que dicta la forma que tendrá la señal modulada, esta es calculada por la ecuación (3.8). Finalmente, la señal de la frecuencia modulada, que es posible obtener por la ecuación (3.10). Esta última señal es la que se desea obtener al aplicar el efecto de Fm, modificando los vectores de audio original y que para el actual caso se aplicó para el sonido1. En dicha gráfica se puede ver que posee el contenido de la señal portadora, pero se encuentra moldeada por la forma de onda, de la señal moduladora.

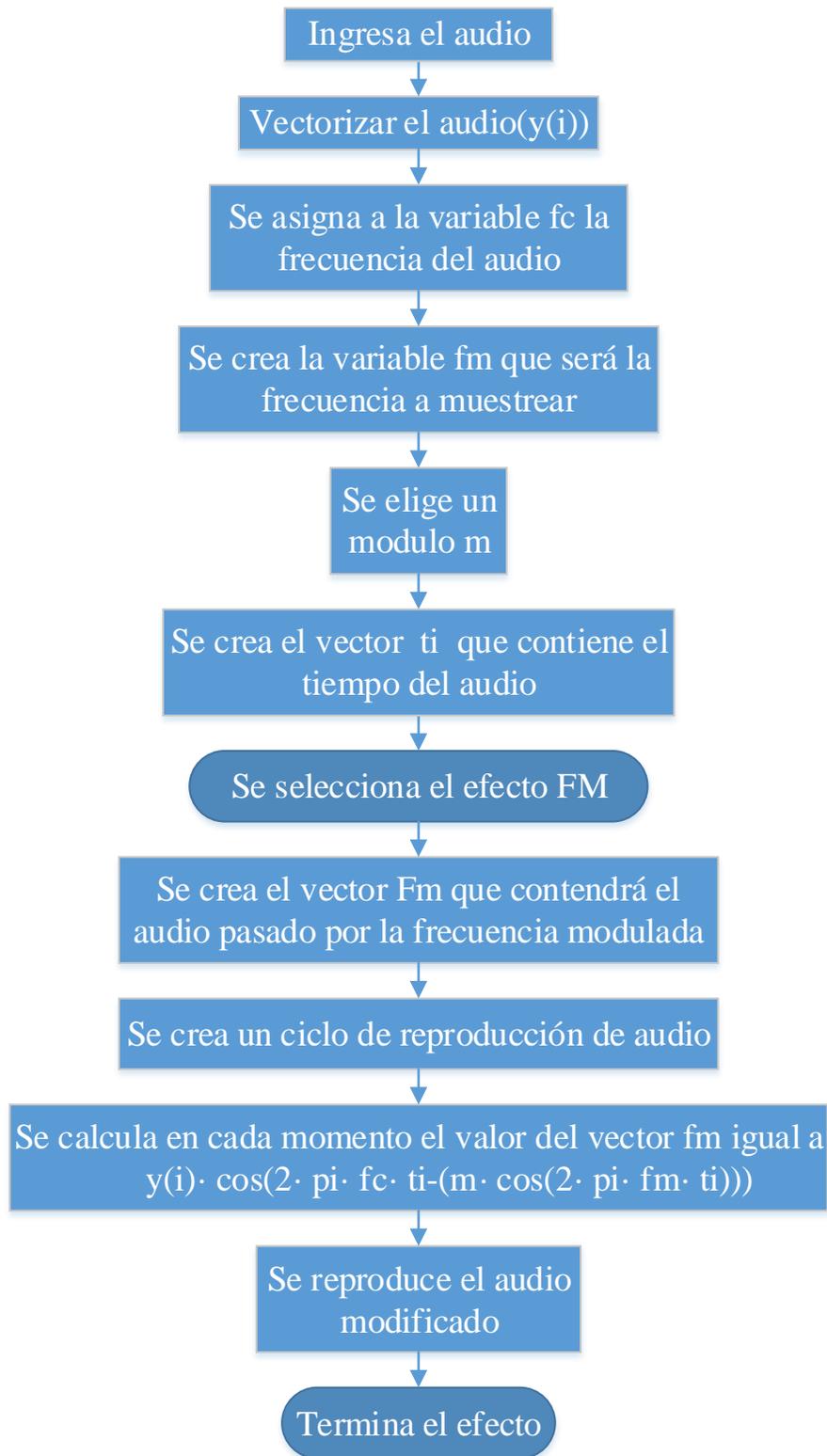


Fig.3.38. Diagrama de flujo del efecto 13.

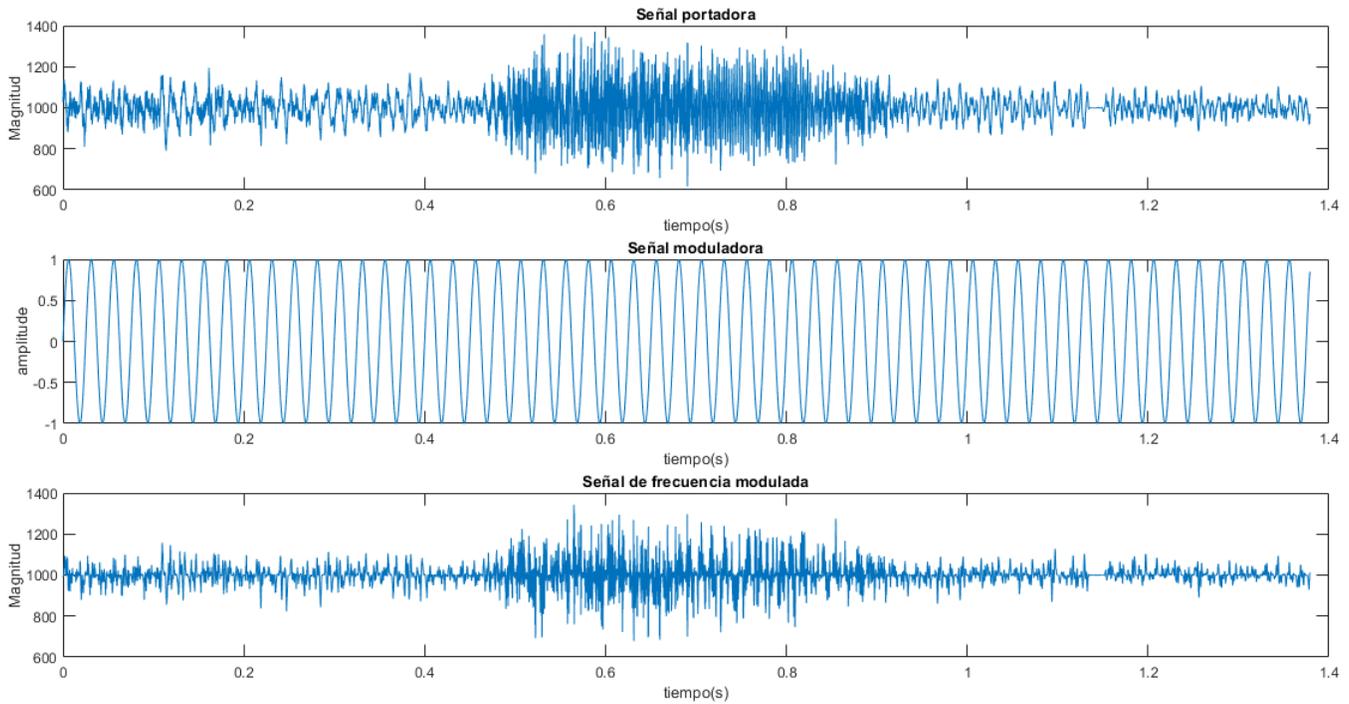


Fig.3.39. Gráficas vectoriales de la señal portadora, moduladora y la aplicación del efecto 12 sobre el sonido 1 (Señal Fm).

3.3.15 Efecto 14 Barrido de baja velocidad

El efecto 14 correspondiente al barrido de baja velocidad, el cual es utilizado por medio de la aplicación de la combinación del filtrado FIR y IIR, al vector de audio. Esta estrategia de filtrado provoca que se logre un barrido a toda la señal de audio. En el efecto 14 es necesario un barrido de baja velocidad, esto indica que solo permitirá frecuencias bajas, puesto que una baja velocidad, indica una baja frecuencia. El caculo es llevado por medio de la ecuación (4.12), la explicación de la presente ecuación, junto el cálculo de sus coeficientes del filtro FIR y IIR, se realizarán en el capítulo 4.3. Por su nivel de complejidad entra en la tercera categoría, del párrafo uno de la sección 3.3, lo que provoca que quede en la misma categoría que los efectos de filtrado.

En la implementación del efecto 14 se realizó un barrido de baja de velocidad, con una frecuencia de corte hasta los 1000 Hz, con un total de un máximo de 10 coeficientes, tanto para el segmento FIR y IIR del filtrado, para el sonido 1.

En la Fig. 3.40. se representa el diagrama de flujo, del funcionamiento del barrido de baja velocidad. Comenzando con el procedimiento de filtrado, pero para el presente caso se calcula dos tipos de coeficientes y se aplica una nueva ecuación, para calcular el efecto de filtrado. Es mediante lo anterior mencionado que es necesario calcular los coeficientes a y b , donde a son los coeficientes de la parte de FIR del filtro y b son los coeficientes de la parte IIR del filtro. En paralelo es necesario calcular el vector del audio a reproducir. Una vez comenzado el efecto, se procede a calcular un ciclo “for”, del mismo largo del vector de audio, en donde se procederá a calcular el efecto de barrido de baja velocidad, mediante la ecuación (4.12). Dicha ecuación (4.12) va ir incrementando en función de la posición del vector del audio y llegando a su máximo cuando la posición del vector de audio sea igual al número de coeficientes. Una vez calculado la modificación del vector de audio original, se procederá a reproducirlo. Para cuando haya terminado de reproducirse el vector de audio modificado, el efecto se acabará, quedando listo para ser implementado el efecto por otro sonido a elección, dentro de los sonidos de la Tabla 3.3 y siempre que se encuentre en el modo de efecto de sonido 14.

En la gráfica de la Fig. 3.41, se presentará los vectores de audio modificados mediante el efecto de sonido 14. La gráfica de la Fig.3.41 guardan un parecido a la forma gráfica del filtro pasa bajos(Fig.3.33), solo que en el presente caso posee una leve variación por el efecto del barrido de baja velocidad.

En la gráfica de la Fig. 3.39 se muestra la transformada de Fourier de la Fig. 3.38. La presente gráfica de la Fig.3.39 es posible notar, que es más limpio y completo el filtrado, en comparación del filtro pasa bajo de la Fig. 3.34. Puesto que se permite todo el espectro de las frecuencias baja a 1000 Hz, superando la frecuencia de corte, empieza a disminuir hasta ser casi nula la actividad de frecuencias altas.

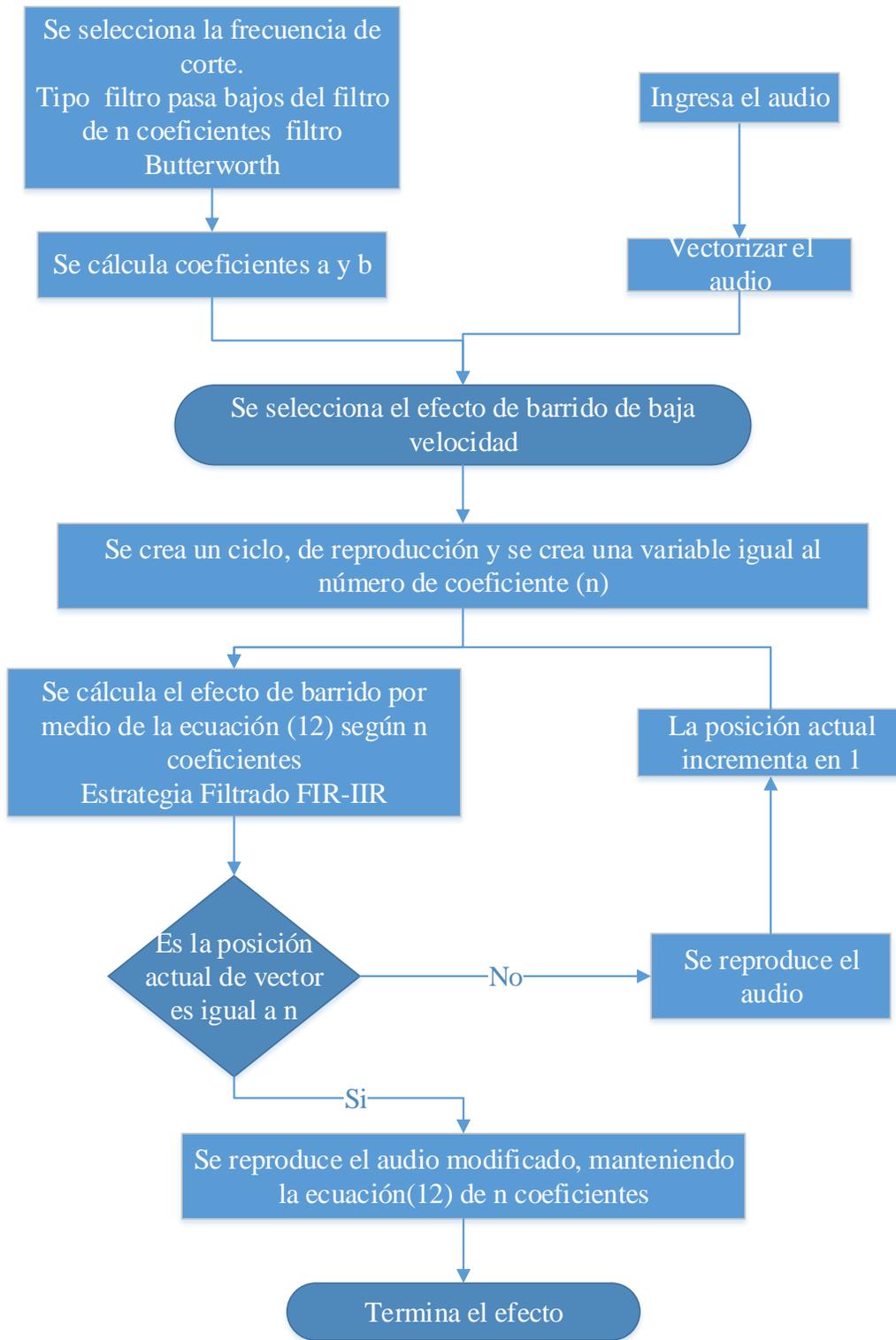


Fig.3.40. Diagrama de flujo del efecto 14.

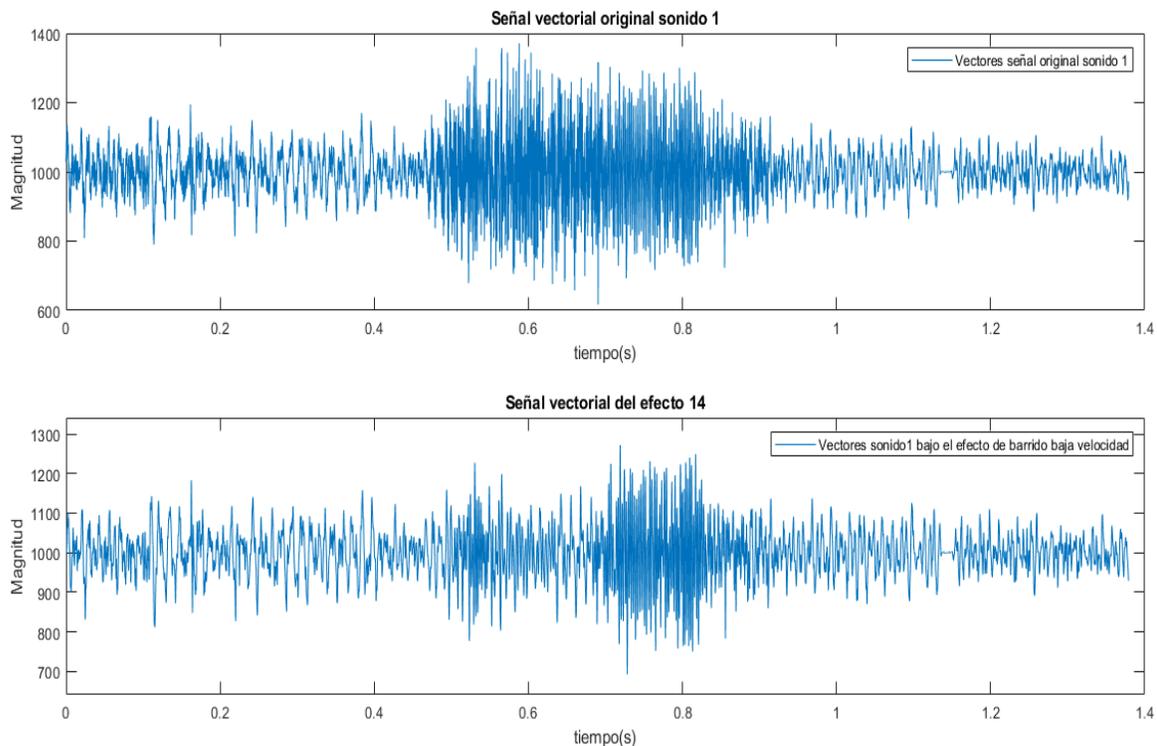


Fig.3.41. Gráfica vectorial del sonido 1 con la aplicación del efecto 14.

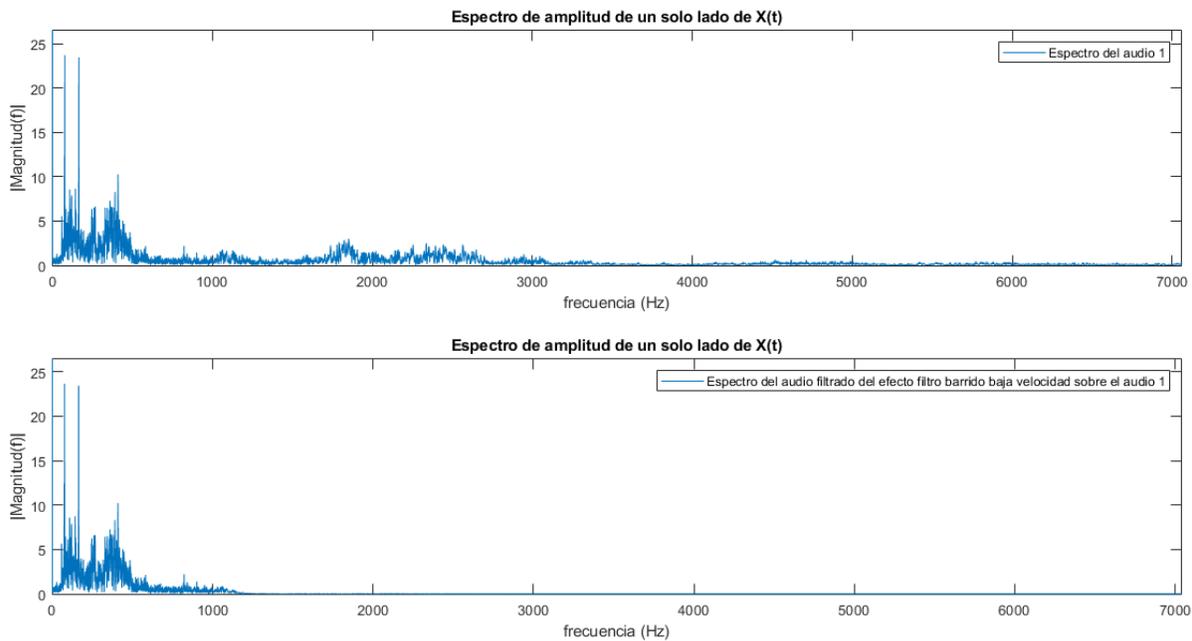


Fig.3.42. Transformada de Fourier de la Fig.3.41.

3.3.16 Efecto 15 Barrido de alta velocidad

El efecto 15 correspondiente al barrido de alta velocidad, como su nombre indica es un barrido que se le realiza al vector de audio, pasando solo las altas velocidades, es decir solo frecuencias altas. La estrategia de filtrado es igual al barrido del efecto 14 (ocupando Filtros IIR y FIR) y es mediante lo cual que entra en la tercera clase de clasificación de efectos, del párrafo uno de la sección 3.3.

Para realizar el cálculo es necesario aplicar un filtro pasa alto, con una frecuencia de corte de 3000 Hz y con un máximo alcanzado de coeficientes de 10.

La Fig. 3.43 se muestra el diagrama de flujo del efecto de 15. En comparación del diagrama de flujo de la Fig. 3.40 con los diagramas Fig.3.29, Fig.3.32 y Fig.3.35, cambia la selección de coeficientes, puesto que el barrido necesita una menor cantidad de estos. Comenzando el efecto se calcula el largo del vector del audio original, para crear un ciclo “for”, en donde se modificará el vector de audio según la ecuación (4.12). Al igual que en el efecto anterior esta ira incrementando en función de la posición actual del vector de sonido, hasta llegar a la posición de n número de coeficientes. Terminando de calcular el efecto de barrido de alta velocidad y de reproducir el vector de audio modificado el efecto quedara listo para ser aplicado a otro sonido dentro de la selección (de la Tabla 3.3), siempre que se encuentre dentro del modo del efecto 15.

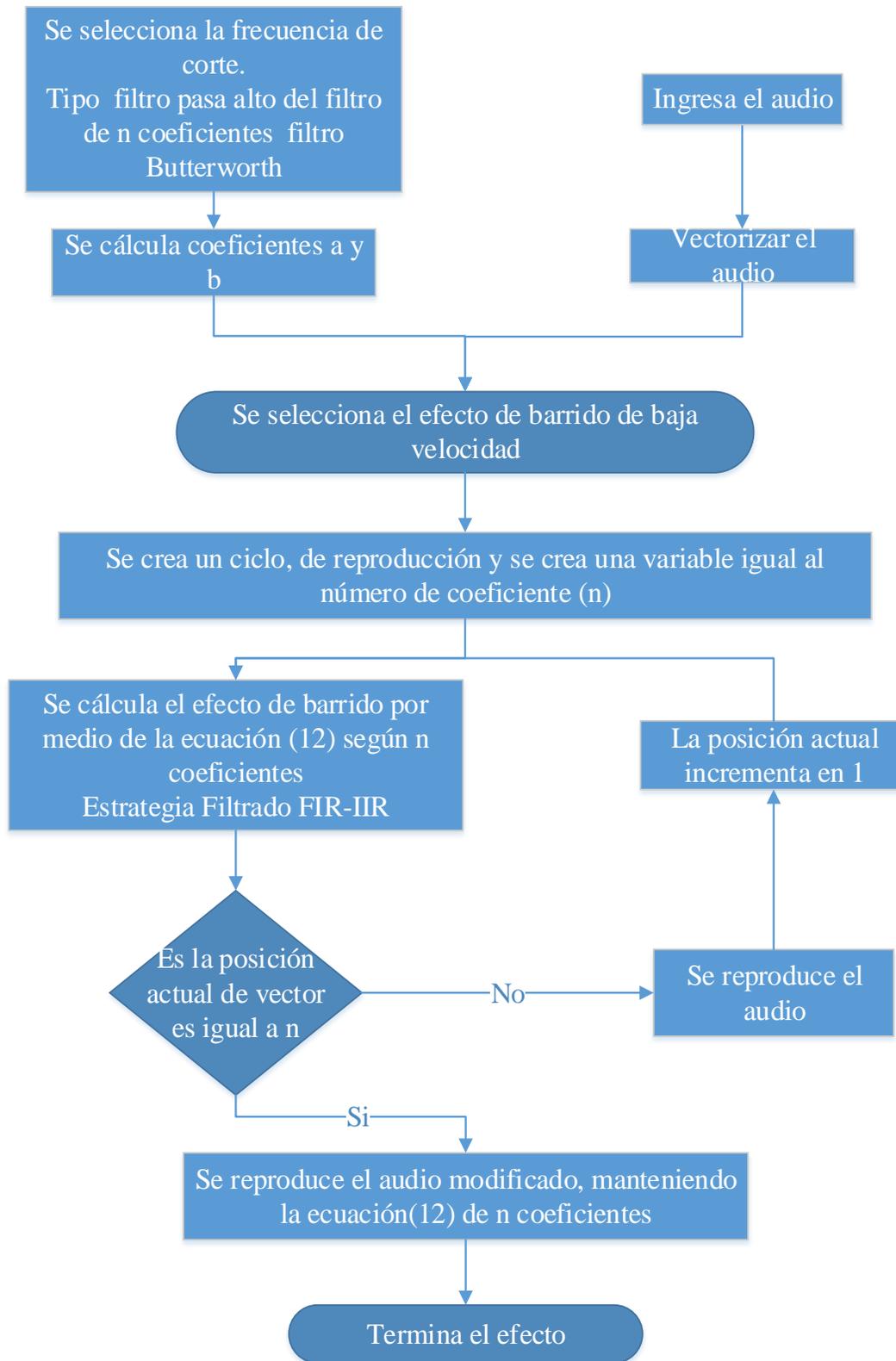


Fig.3.43. Diagrama de flujo del efecto 15.

En la Fig. 3.44 se muestra la gráfica vectorial de la aplicación del barrido de alta velocidad, sobre el sonido 1. En comparación de la aplicación del filtro pasa alto de la Fig. 3.30, se ve en este caso una gráfica mucho más compacta, con una variación de la magnitud del volumen del audio, más limitada. Esto es debido que la representación de una señal de alta velocidad su periodo es más corto, en este preciso caso hace que sea más compacta la variación de los vectores.

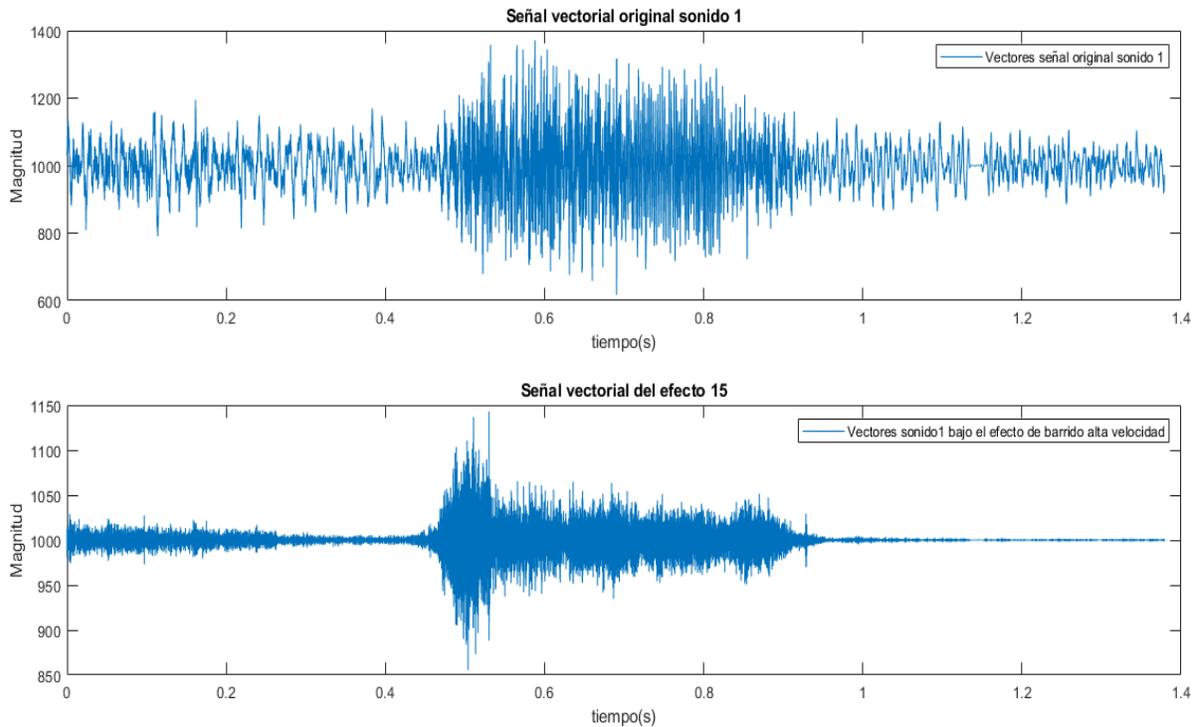


Fig.3.44 .Gráfica vectorial del sonido 1 con la aplicación del efecto 15.

La respuesta de la transformada de Fourier, de la gráfica de la Fig. 3.44, se presente en la siguiente Fig. 3.45. En la cual se ve un filtrado más eficaz a comparación del filtrado del efecto 10. Es de notar que no hay ingreso del espectro de frecuencias bajas, hasta llegar a cercano la mitad de la frecuencia de corte de 3000 Hz, en donde la magnitud absoluta de la frecuencia va aumentando, hasta llegar a la frecuencia de corte. Una vez llegado a la anterior frecuencia, empieza pasar todo el espectro de las frecuencias altas.

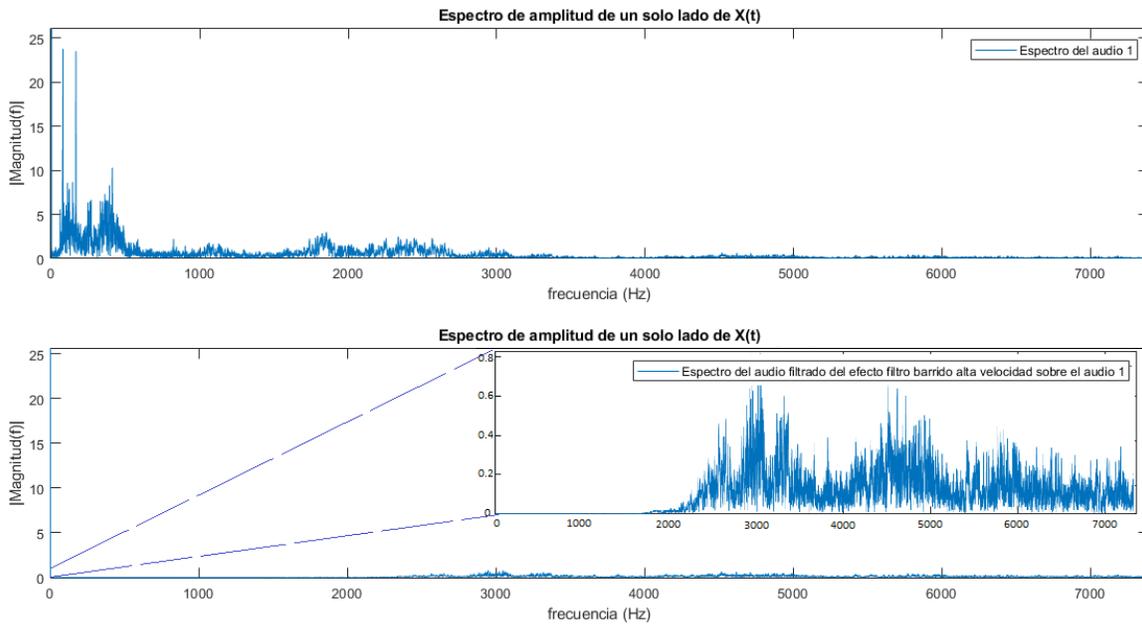


Fig.3.45. Transformada de Fourier de la Fig.3.44.

3.3.17 Efecto 16 Barrido Distorsión

El ultimo efecto de los 16 efectos de sonidos, del sintetizador de sonido, es el efecto de barrido distorsión. Este efecto toma en consideración el efecto de distorsión y filtro de distorsión, en donde se realiza la modificación del vector de sonido, agregando un valor aleatorio a la salida del vector modificado. El barrido distorsión posee el mismo funcionamiento que los otros tipos de barridos, solo que se le agrega la modificación anteriormente mencionada. Por su complejidad entra en la tercera categoría de efectos de sonido.

Para la presente prueba se utilizará un barrido de velocidades medias, lo que es un filtro pasa banda aplicando la estrategia de barrido. La frecuencia de corte será de 500 Hz hasta los 17000 Hz, con un máximo de coeficientes permitidos de 7 y la suma de una ganancia de 1% el valor del vector modificado por el barrido.

La lógica del diagrama de flujo de la Fig. 3.46, responde a las ya vistas en los demás efectos de barrido. Antes del efecto se estima las frecuencias corte, se calcula los coeficientes de a y b, y en paralelo se calcula la representación vectorial del audio elegido. Comenzando el efecto se crea un ciclo “for” de igual de largo que en el vector de audio original. Se procede a calcular el efecto de barrido, mediante la ecuación (4.12) y terminando el cálculo del barrido

se agrega un ruido, equivalente al 1% del valor de este vector modificado. Finalizando se procede a reproducir el vector de audio modificado por el efecto de barrido distorsión. Quedando listo para ser reutilizado, en otro sonido a elección.

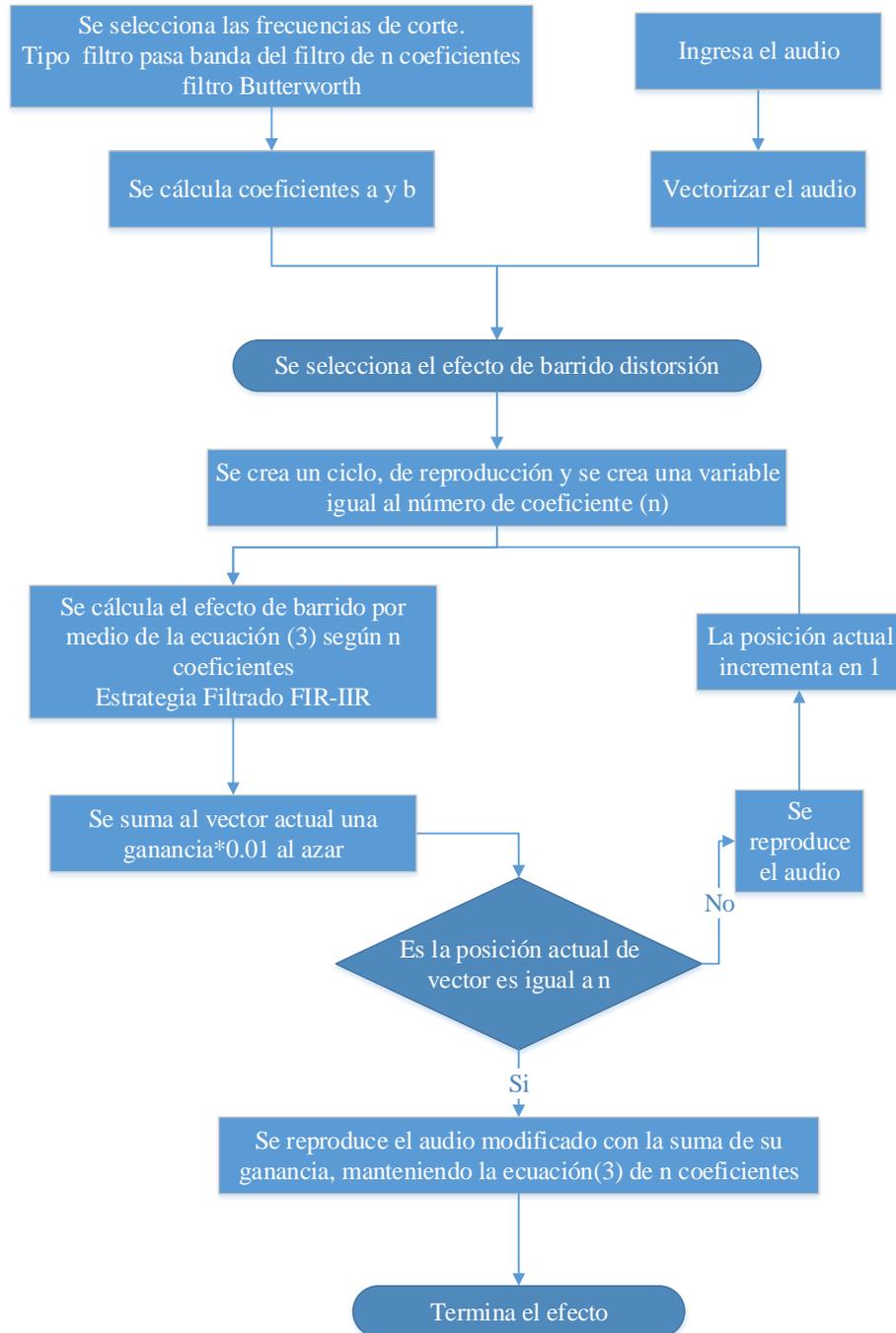


Fig.3.46. Diagrama de flujo del efecto 15.

La representación de la gráfica vectorial, se presenta en la Fig. 3.47, la cual presenta una forma onda, promediada entre las ya vistas en los efectos 14 y 15. Esta forma de onda de la Fig. 3.47 es debido a que engloba casi todo el espectro de audio del vector de audio original, eliminando solo las frecuencias más bajas. Es de considerar que la presente gráfica se encuentra bajo los efectos de distorsión, mediante la alteración del vector de barrido, agregando una pequeña ganancia.

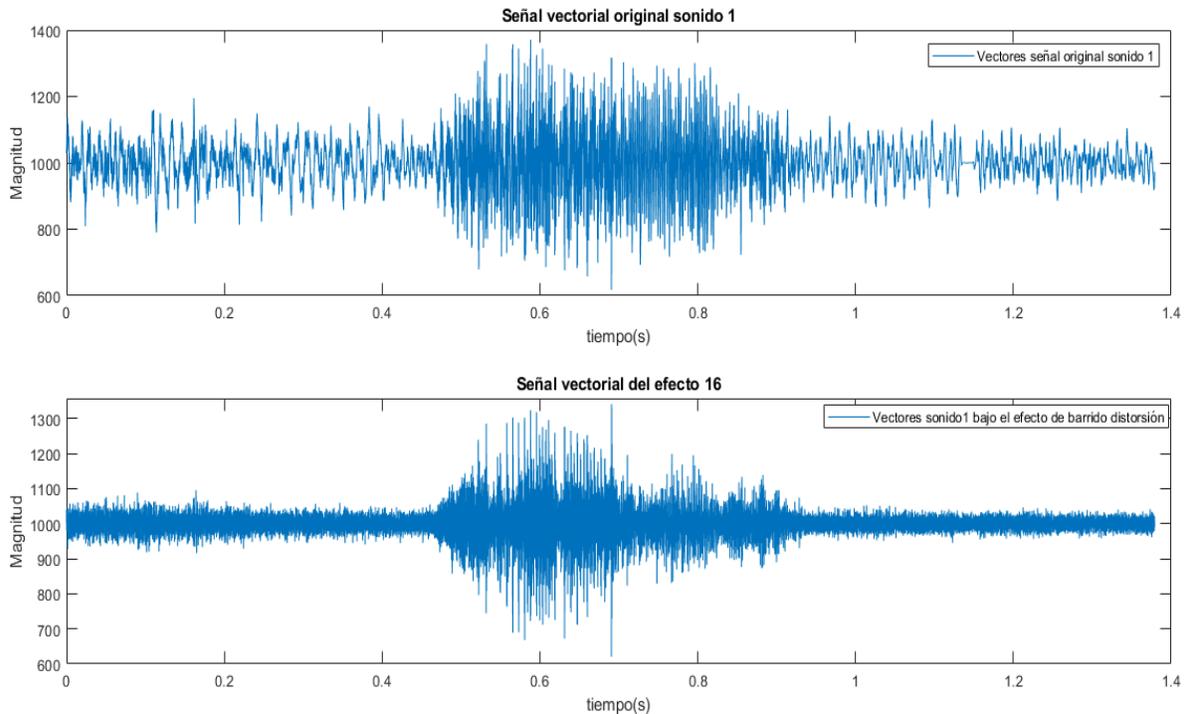


Fig.3.47. Gráfica vectorial del sonido 1 con la aplicación del efecto 16.

La repuesta de la transformada de Fourier, de la anterior gráfica Fig. 3.47, se muestra en la Fig. 3.48. En donde se visualiza que el espectro de la frecuencia de la anterior graficada ejerce bien el filtrado de las frecuencias bajo a 500 Hz. Esto permite el paso todo el espectro de las frecuencias superior a la tope inferior de la frecuencia de corte, pero es debido mencionar que no se hace el filtrado esperado llegando a los 17000 Hz, debido a que incluyo efecto de distorsión. La inclusión de la distorsión indica que haya frecuencias no deseadas, dentro del barrido de la señal.

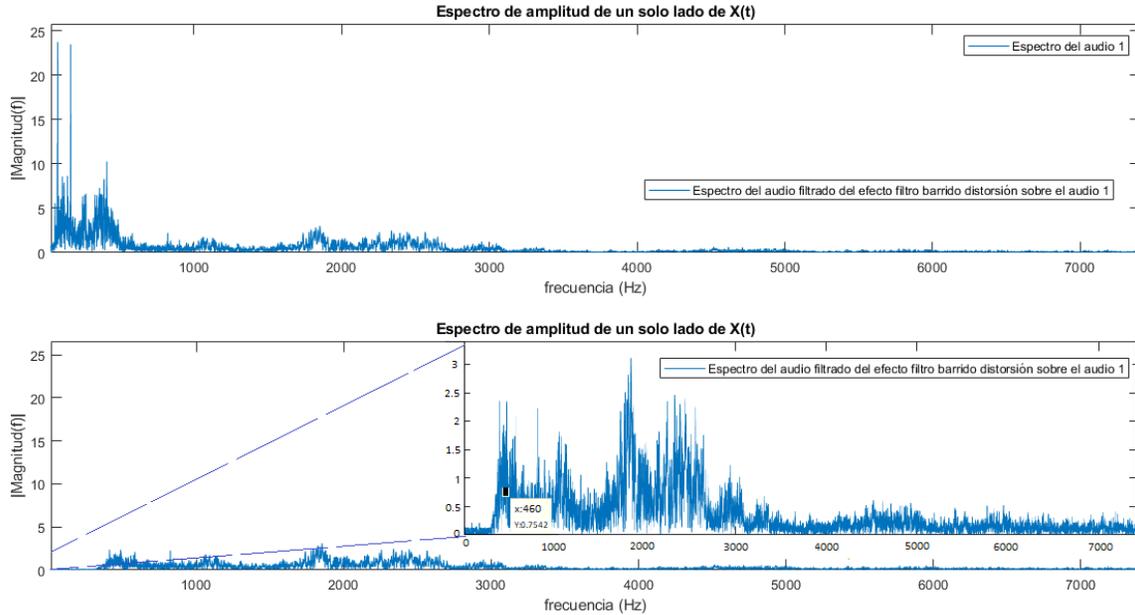


Fig.3.48. Transformada de Fourier de la Fig.3.46.

3.4 Modificación de parámetros del modo patrón y de los efectos de sonido

Para la modificación de parámetros del sintetizador de sonido, implementado en una plataforma Zynq-7000, se realizó la elección de ocupar un encoder rotativo, para dichas funciones. Estas funciones abarcan tres variables principales a modificar, primero los BPM (Pulsaciones por minuto), segundo el control del volumen del audio y tercero la modificación de alguna variable específica, que se encuentre incluida dentro de las funciones de los efectos de sonido.

Dentro del funcionamiento del encoder rotativo, se encuentra el hecho que, al tener varios canales a controlar es posible controlar cada parámetro de estos por separado, por el simple hecho que dicho dispositivo rotativo puede girar de manera indefinida y ser analizado simplemente por el código a utilizar. Esta medida de control es bastante conveniente por el simple hecho que el sintetizador de sonido posee un gran número de canales y efectos a controlar, lo que provoca que se pueda controlar cada una de manera separada y no como un potenciómetro. En el caso que se ocupara un potenciómetro este estaría limitado, por su funcionamiento al solo poder dar un valor estándar para todas las variables por igual. El encoder rotativo a ocupar posee una precisión de 20 pulsos por cada giro, lo que es equivalente a por cada 18° del giro este envía un pulso, para sus características técnicas se

presentarán en el capítulo 4.2.2. Para el funcionamiento del encoder rotativo, se encuentra en el siguiente pseudocódigo 4.

Algoritmo 4 Pseudocódigo Perillas

```
1: clk = 0
2: DT = 0
3: Pausa = 0
4: resta = 0
5: suma = 0
6: variable = valor original de la variable
7: inicio ciclo modificacion de variable
8:   para clk == 1 y DT == 1 hacer
9:     suma = 0
10:    resta = 0
11:    pausa = 1
12:   fin
13:   para pausa == 1 y clk == 0 y DT == 1
14:   Hacer
15:     suma = 1
16:     resta = 0
17:     pausa = 0
18:   fin
19:   para pausa == 1 y clk == 1 y DT == 0
20:   Hacer
21:     suma = 0
22:     resta = 1
23:     pausa = 0
24:   fin
25: variable = variable + suma – resta
26: fin del ciclo
```

En el pseudocódigo 4, se muestra las variables a ocupar, en donde “clk”, corresponde al pin A del encoder y “DT” a su pin B. Mientras que las demás variables son variables de estado, que indicaran cuando se encuentra en reposo, las señales de A y B, además de si se va incrementar o disminuir el valor de la variable a controlar. En el comienzo con ambas señales de A y B, en estado alto, lo que indica que están en reposo, en consecuencia, la variable pausa pase a 1. Posterior a ello si se quiere aumentar el valor de la variable, es necesario que salga de reposo las señales y se gire en este caso en sentido horario. Lo que provocara que la señal A pase a cero y que la señal B siga en 1, este hecho hará que suma sea uno (dependiendo la escala de la variable a modificar) y pausa sea cero, esto da hincapié a

que se tenga que pasar nuevamente por el estado de reposo, donde ambas señales de A y B sean 1. En caso contrario que se desee disminuir el valor de la variable, es necesario girar en sentido anti horario, lo que provocara que las señales salgan de su estado de reposo, y que pase B a valer cero y A a seguir valiendo 1. El resultado de esta condición hace que pausa sea cero y resta sea uno (dependiendo la escala de la variable a modificar). Finalmente, el valor de suma y resta se le agregara al valor de la variable a modificar, en caso de incrementar, suma será uno y resta cera cero, en caso contrario suma cera cero y resta cera uno.

La importancia del sistema de protección del seudocódigo de la variable pausa, es asegurar que las señales pasen a su estado de reposo, luego de girar en cualquiera de los dos sentidos de horario. Si no se tuviera la variable pausa, en cada ciclo de reloj de la FPGA estaría incrementando y disminuyendo sin control, las variables de suma y resta.

3.4.1 Control perilla A

El sintetizador de audio posee una variable que siempre estará presente en todos los modos la cual es el volumen, tanto para los canales del modo patrón como para los efectos de sonido. Es por ello que se eligió que la perilla A, correspondiera a la modificación del volumen de los canales.

En el diagrama de flujo de la Fig. 3.49, se muestra el funcionamiento de la perilla A, Para todo canal, de manera independiente del valor del volumen, si se gira en sentido horario aumentara el volumen y si es en sentido anti horario disminuirá. Siempre que se respete el rango del volumen entre 0% a 100%.

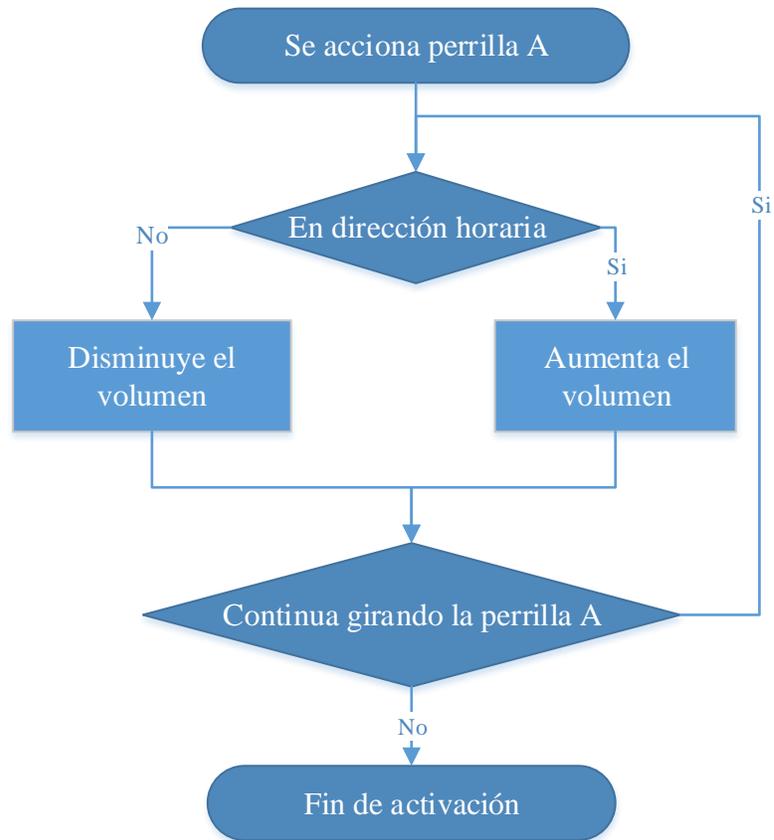


Fig.3.49. Diagrama de flujo del funcionamiento de la perilla A.

3.4.2 Control perilla B

La segunda perilla (B) a ocupar, se encarga del control de variables que no estarán siempre presentes, tal es el caso como el control de los BPM (pulsaciones por minuto). Los BPM solo estarán disponibles desde el canal 1 al 15. El otro caso se encuentra en el canal 16, de efectos de sonidos, en donde se encargará de controlar dependiendo el efecto de sonido, algunas de sus variables disponibles, variando según el efecto. Los efectos de sonido en donde se aplica el funcionamiento del encoder rotario es del efecto 4 al 6. En el siguiente diagrama de flujo de la Fig. 3.50, se da a conocer el principal funcionamiento de la perilla B.

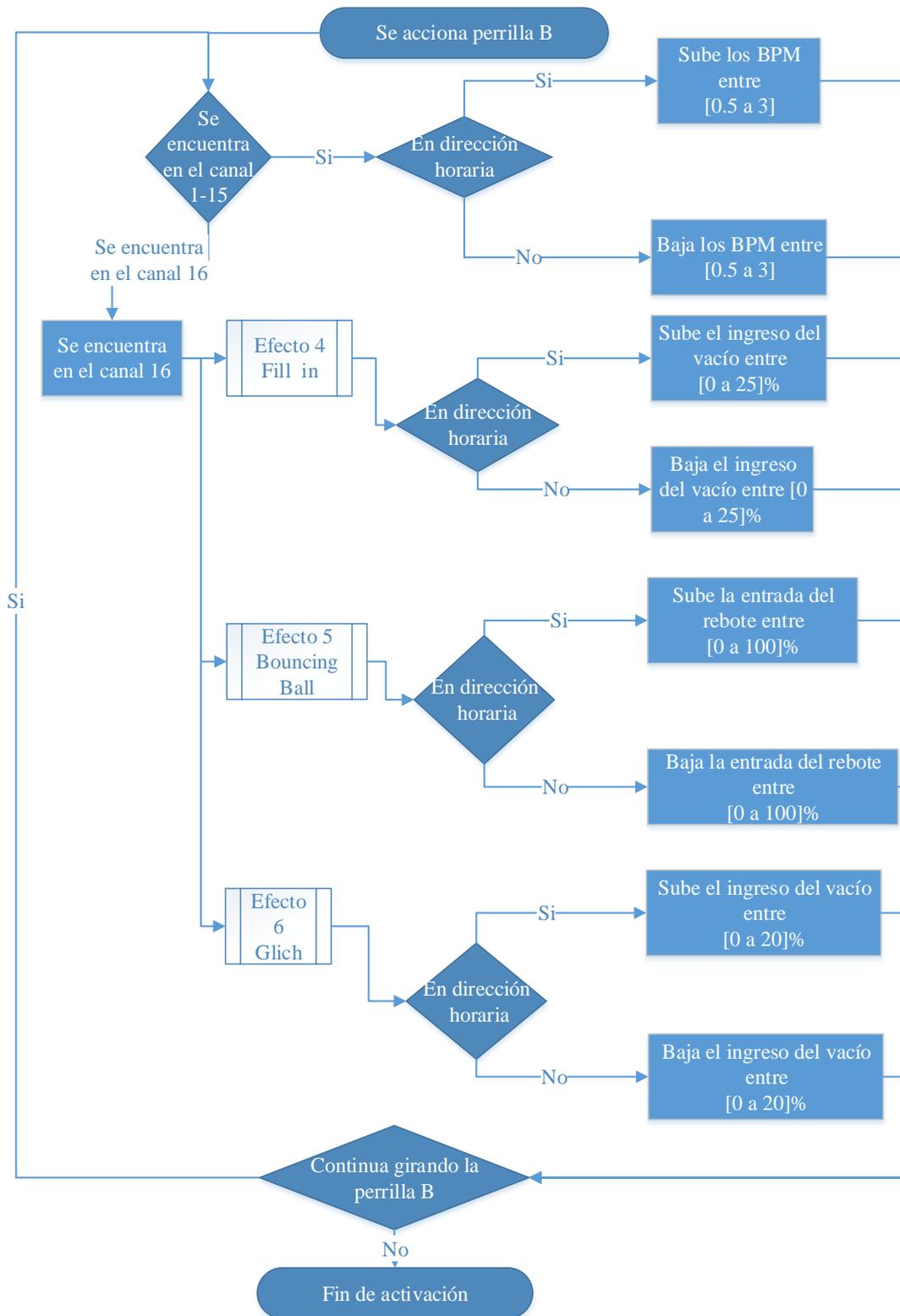


Fig.3.50. Diagrama de flujo del funcionamiento de la perrilla B.

Para el funcionamiento de la perilla B, se encuentra en primer lugar, un funcionamiento similar a lo ocupado por la perilla A, para este caso del canal del 1 al 15, se hará control de los BPM de cada canal. En donde esta ira en un rango entre 0.5 a 3, dando a entender que la velocidad del modo patrón, puede disminuir a la mitad de su velocidad original o puede llegar hasta triplicarla. Dado el funcionamiento de la perilla A, también se asignó que el giro del sentido horario se ocupara para incrementar el valor de las variables a controlar y el sentido anti horario, para disminuirlo.

Para la modificación de parámetros del canal 16 efecto 4 correspondiendo a Fill in. En donde la variable a controlar será el ingreso del tiempo de vacío, el cual ira entre 0% al 25% del sonido, teniendo en cuenta como se explicó en el capítulo 3.3.5, son un total de 4 instancias de vacío. Lo que indica que el primer ingreso de vacío, puede ser entre el comienzo del audio o hasta cuando el audio llegue a un cuarto de su reproducción. En donde el resto los ingresos del vacío, poseerán la misma separación de manera uniforme uno del otro.

La modificación de parámetros del canal 16 efecto número 5, correspondiente al efecto Bouncing ball. Se controlará el inicio del primer rebote, el cual comandará el resto de los rebotes del efecto. Dicho comienzo de rebote puede ser desde 0% al 100% de la reproducción del audio. Lo que indica que puede sonar todos los rebotes al comienzo del audio o en caso contrario, hasta llegando a terminar de reproducirse el audio, continuaría el primer rebote, terminado este continuara el segundo rebote y así sucesivamente. Puesto como se explicó en el capítulo 3.3.6, el efecto de rebote seguirá funcionando de igual manera, lo único que variará en él, es en qué punto del audio original comenzará a escucharse el efecto de fondo.

La última modificación de parámetros del canal 16 efecto número 6 correspondiente al efecto de Glich, el cual posee propiedades similares al efecto 4, lo único que en vez de agregar tiempo sin que se reproduzca el audio, se agregan un sonido externo de una corta duración de tiempo (≤ 0.2 seg). La cantidad de ingresos del audio del Glich son un total de 5, por lo cual el control de ingreso será entre 0% al 20% de la reproducción del audio original. Esto da a indicar al igual que en los casos anteriores, que el primer Glich puede sonar, al comienzo del audio o hasta cuando este llegue a un quinto, de su reproducción. Dando a

entender que el resto de los “glitches” se encontrara separados de manera uniforme, por un quinto de distancia de reproducción del sonido original.

3.5 Visualización de los datos de las variables de los efectos de sonido y modo patrón

La visualización de las variables, se encuentra a cargo de una pantalla LCD, dicha pantalla posee un total de 4 líneas de pantalla, con un total de 20 caracteres, por cada línea, además poseer el tipo de comunicación I²C, las características técnicas se mostrarán de una manera más detalla, en el capítulo 4.2.3.

La importancia de la pantalla LCD es mediante su visualización de las variables de estados, es posible saber de manera más fácil, en que canal se encuentra el sintetizador, el volumen de cada canal, el nombre del canal y el nombre del efecto de sonido entre otras variables. Es mediante la siguiente distribución de bloques, según la línea de la pantalla de la pantalla LCD, de la Fig. 3.51, que se da a conocer toda la información que dará la pantalla.

Para la primera línea de la pantalla, se tiene contemplado, que indique el número del canal “ChX” en el cual se encuentre, además del nombre de dicho canal. En caso que se encuentre en el canal 16, este dará el nombre del sonido de la Tabla 3.3.

En la segunda línea se mostrará, el valor del volumen del audio, tal como se informó en la sección 3.4, este se encontrará en el rango de 0 a 100. Por otro lado, también se mostrará el valor del BPM, del respectivo canal de 1 a 15, tomando en cuenta como se dio a conocer en la sección 3.4 que dicho valor estará entre 0.5 a 3.

La tercera línea, está destinada únicamente si se encuentra en el canal 16. Puesto que en dicha línea se dará a conocer el nombre del efecto de sonido, que se está aplicando.

Finalmente, para la cuarta línea, se encuentra las variables de los efectos de sonido a modificar, tal como se explicó e informo en la sección 3.4, esta solo se activará con los efectos del 4 al 6. Donde se dará a conocer el nombre de la variable a modificar y el valor de este.

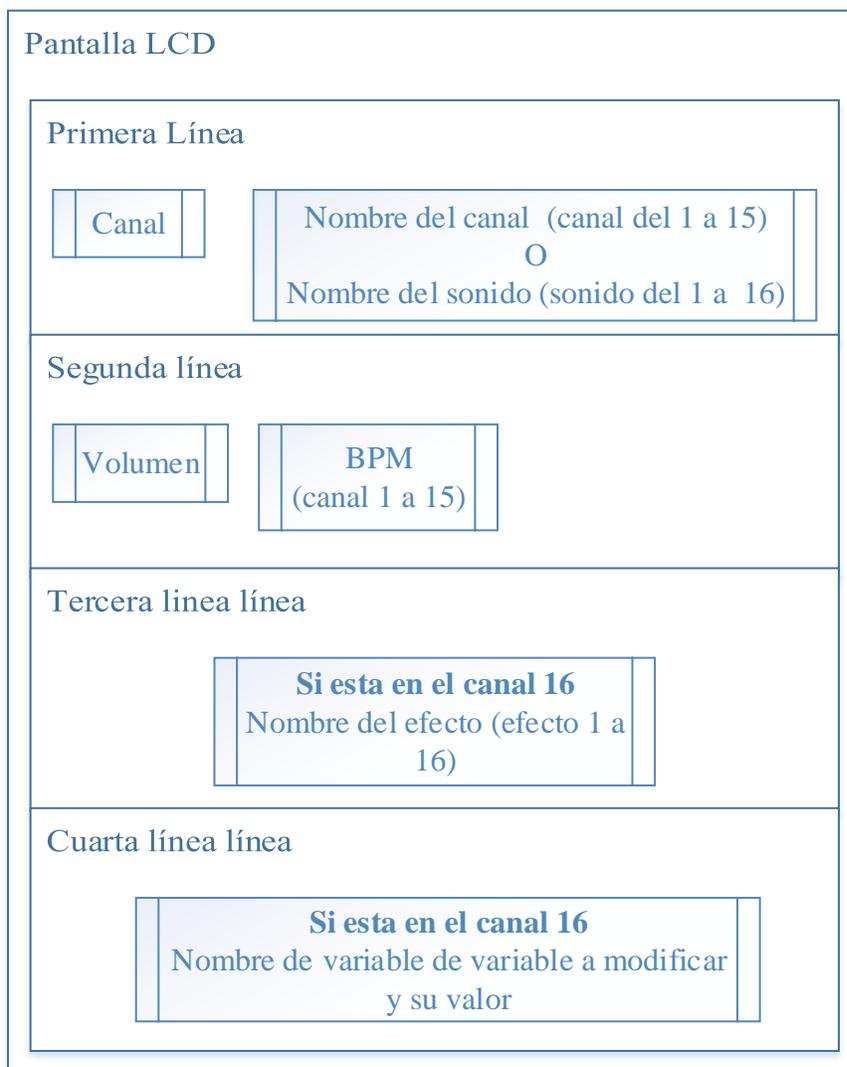


Fig.3.51. Distribución de bloques de información según línea de la pantalla LCD.

4 Implementación del sintetizador de sonido en una plataforma Zynq-7000

4.1 Introducción

Para la realización práctica e implementación en la Zynq-7000, es necesario hacer constatación de los elementos disponibles, empezando por el análisis del equipo necesario para la conformación del sintetizador de sonido.

Prosiguiendo se explicarán los bloques IP principales del Hardware mixto de Vivado, con lo cuales fue posible la conformación de un sintetizador de sonido.

Continuando se explicará la implementación del software concerniente a la aplicación de la estrategia de filtrado y barrido de los efectos de sonidos. Dentro de esta estrategia de control se desarrollará a mayor profundidad los filtros FIR e IIR explicados en el capítulo 1.2.3.2, dando a conocer en este caso las ecuaciones necesarias para llevar a cabo los efectos de filtrado y barrido.

Finalmente, dado que la presente memoria es una adaptación y emulación, de los sintetizadores portátiles de la serie PO KO de Teenage engineering. Se realizará un análisis de costos de los elementos ocupados en la memoria frente a los sintetizadores, de la presente marca en la cual se está basando la actual memoria.

4.2 Implementación del Hardware del sintetizador

En la presente sección 4.2, se dará a explicar el funcionamiento y control del Hardware externo ocupado en la presente memoria. Pasando por los diferentes elementos empezando por el teclado elegido, que permitirá controlar y activar los diferentes modos, como se explicó en el capítulo 3. Posterior a ello se mostrará el control de las perillas para sonido, las que se encargaran de controlar diferentes variables, tales como el volumen, los BPM (pulsaciones por minuto) y algunas variables disponibles, en función de los efectos de sonido. Finalmente, se dará a conocer el tipo de pantalla elegida, sus características principales y su modo de comunicación.

Por último, se explicará el funcionamiento del Hardware mixto, aplicado en el software de Vivado. El presente Hardware mixto permite el control de las funciones y periféricos de

la Zybo Z7, en donde cada bloque cumple una función específica. Las funciones de los bloques específicos permiten utilizar las funciones de la FPGA explicadas en el capítulo 2 y del Hardware externo, que se explicara en la presente sección.

4.2.1 Detector Táctil de 16/8 teclas detección IC TTP229-BSF

Dentro del variedad de teclados a elegir, para el desarrollo del sintetizador de sonido en la FPGA Zynq7000, es necesario la consideración de su funcionalidad y simpleza, a la hora de implementarlo. Esta decisión es en base a los mostrado en los sintetizadores de la serie PO KO vistos en los capítulos 1.2.1 y 1.2.2, donde es necesario poseer un teclado con un total de 16 teclas, distribuidas de forma de matriz de 4x4.

El detector táctil IC TTP229-BSF es un teclado de múltiple comunicación, puesto que puede funcionar de 8 a 16 teclas, además de actuar con detección capacitiva, para el funcionamiento de su panel táctil. Posee un modo simple en el cual se puede operar con ochos teclas y conectar directamente sus salidas al respectivo microcontrolador a ocupar. En donde cada pin de salida corresponde a la activación de una tecla de la posición 1 a 8. También se puede controlar mediante la comunicación “2-wire”, la cual solo cuenta con 2 pines de conexión “SCL” correspondiente a la señal de reloj que recibirá desde el microcontrolador y el “SDO” correspondiente a la señal de salida, que permitirá notificar al microcontrolador que se ha activado cierta tecla[52].

Llevando lo anterior mencionado a la utilización de las 16 teclas para el sintetizador de audio implementado en la FPGA, el teclado 16 posiciones se deberá activar mediante las conexiones especiales internas de la tarjeta, también conocida como TP, las cuales se encuentran desde la combinación 0 a 7, dentro de los puertos de la tarjeta P1 y P2. Dichas conexiones especiales se encuentran en estado alto de manera permanentes, siempre que este alimentada la tarjeta y mediante la conexión de estos terminales, permite que la señal vaya a cero. Lo que permite diferentes modificaciones, dentro de estas se encuentra trabajar con trenes de pulso de una señal en alto o bajo voltaje, deshabilitar las teclas y controlar el modo de 8 o 16 teclas. Para los requisitos necesarios para el sintetizador de sonido, es necesario activar el TP2, cual permite trabajar con el modo de 16 teclas, este debe ser controlado mediante la técnica de comunicación anteriormente mencionada de “2-wire”[52]. La Fig. 4.1,

se muestra el diagrama de conexión interno del integrado TTP229-BSF, el cual permite la detección de teclas y comunicación del teclado táctil, hay que destacar que los pines de conexión internos en la Fig. 4.1, poseen el mismo nombre que las conexiones especiales TP. Además, en la Fig. 4.2, se muestra el modelo físico del teclado.

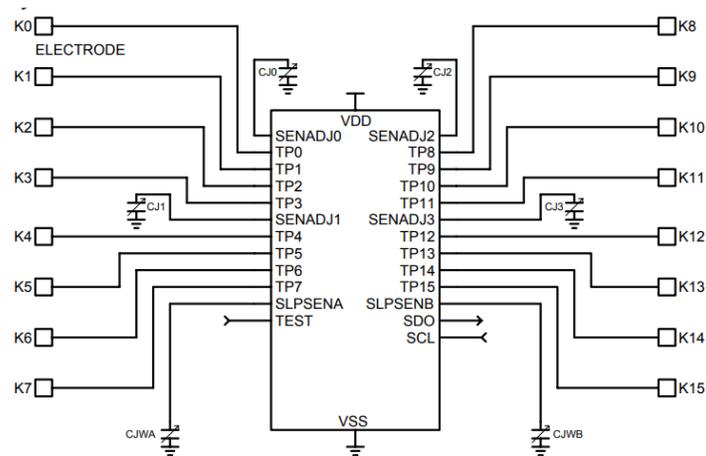


Fig.4.1. Diagrama interno de conexiones del integrado TTP229-BSF.



Fig.4.2. Tarjeta física teclado táctil 8/16 teclas TTP229-BSF.

Dentro de sus especificaciones técnicas se encuentra, se encuentra: (I)El tipo de comunicación del integrado TTP229-BSF, (II)La lógica del voltaje que posee, la cual es CMOS, (III)El rango de frecuencia en la cual puede trabajar, entre otras variables. Los principales datos técnicos, necesarios para poder trabajar en con el teclado se encuentran en la Tabla 4.1.

Tabla 4.1. Ficha técnica básica del teclado táctil de 8/16 teclas.

Variable	Valor	
1.Voltaje de operación	2.4V-5.5V	
2.Interfaz de comunicación modo 16/8 teclas	2-wire	
3.Sensibilidad de capacitancia	1-50pF	
4. Corriente estándar en las teclas	8 modo	2 μA
	16 modo	2.5 μA
5. Tiempo de respuesta 8 teclas	16 ms	
6. Tiempo de respuesta 16 teclas	32 ms	
7. Frecuencia SCL	1K-512K Hz	
8. DV	93 μs	
9.Tw	10 μs	
10.Tout	2 ms	

El sistema de lectura para el teclado táctil en modo de 16 teclas, es mediante un tren de pulsos entregado por el pin SDO, en función de la frecuencia de entrada del pin SCL. Para la implementación se mantuvo el modo por defecto entregado por los TP, de las conexiones internas especiales de la tarjeta. La cual hace que trabaje con lógica de activación baja, es decir que cuando se active una tecla el periodo de tiempo correspondiente para dicha tecla permanecerá en estado bajo. Es de destacar que el tren de pulsos provoca que, en cada periodo de tiempo del sondeo, correspondiente a cada una de las 16 teclas se encuentre en una variación lógica entre 1V(Vcc) a 0V y únicamente el periodo de la tecla correspondiente se mantendrá en 0 cuando sea pulsada. La Fig.4.3 se muestra el esquema del tren de pulsos de la comunicación “2-wire”, de los pines “SCL” y “SDO”.

Set 16-keys active-low

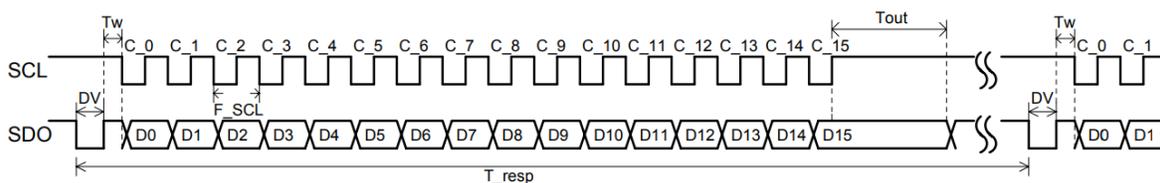


Fig.4.3. Diagrama del tren de pulsos de control del teclado táctil 8/16 teclas TTP229-BSF.

El teclado táctil IC TTP229-BSF, es el elegido para la implementación del sintetizador de audio. Por el ahorro de pines a ocupar, también por la simpleza del diseño, el cual permite tener las 16 teclas dentro de la misma tarjeta. Además de la velocidad de comunicación, la cual se ajusta a la frecuencia ocupada por el sintetizador de audio.

Finalmente, por el diseño de esta siendo mucho más sensitiva al momento de presionar el teclado y su tamaño compacto.

4.2.2 Encoder rotativo KY-040

El encoder rotativo, es un dispositivo de medición que puede entregar dos señales, desfasadas en función del sentido de giro de su perrilla. Es ocupado comúnmente para el control de velocidad de motores o control de voltaje de una salida. Su principal diferencia a otros tipos de control de variable de perilla como potenciómetros, es el hecho que los encoder poseer un giro sin fin.

Su funcionamiento está basado en la diferencia de sus dos salidas, la cual dependiendo el sentido del giro y de su precisión, en donde una de las dos señales de reloj pasara primero a un estado bajo (0 Volt)[53]. El Encoder rotativo KY-040, posee una variedad de parámetros, informados en la Tabla 4.2.

Tabla 4.2. Ficha técnica Encoder rotativo KY-040[53].

Variable	Valor
1. Voltaje de operación	5 V
2. Pulsos 360°/Rotación	20
3. Angulo mecánico	360° Continuos
4. Dimensión	(30 x 18 x 30)mm

El encoder rotativo KY-040, posee un total de 5 pines de conexión, los fundamentales Vcc (+) y GND, respectivos a la alimentación del encoder. Los pines CLK(A) y DT(B), correspondiente a las salidas de la señal de reloj del encoder. Finalmente se encuentra el pin SW, el cual es un interruptor para la generación de presión del encoder. En la Fig.4.4 se muestra el diagrama de pines del encoder rotativo KY-040. Además, en la figura Fig. 4.5 se encuentra el muestreo de la señal de salida del encoder anteriormente mencionada, de la señal A y B, según su sentido de giro[53].

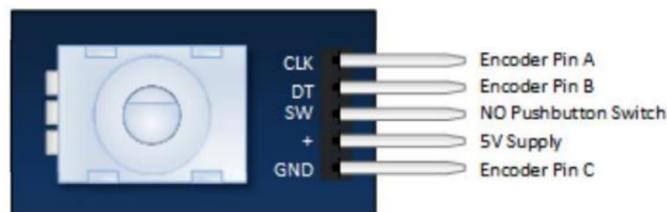


Fig.4.4. Pines de conexión encoder rotativo KY-040[53].

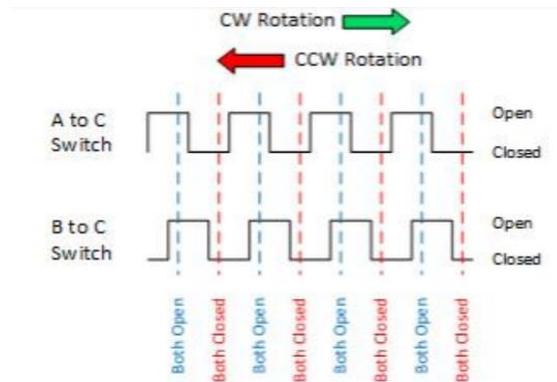


Fig.4.5. Señales A y B según su sentido de giro del encoder rotativo KY-040[53].

La elección del encoder, como elemento de control de parámetros para el sintetizador es por su ventaja sobre el potenciómetro, puesto que los potenciómetros solo llegan a una posición máxima, tanto como una resistencia mínima permitida o máxima. La presente posición del potenciómetro hace que mantenga un valor fijo, para todas las variables que requisen de este elemento. Es por ello que frente al sintetizador de audio en la FPGA, no es un elemento prescindible. Ya que para la memoria se va a hacer una serie de control de volumen, frecuencia y variables de efectos de sonido, las cuales un potenciómetro no es capaz de solventar todas estas variables, ya que solo mantiene un valor fijo, para los diferentes canales. Por el caso contrario un encoder es capaz de tener un giro sin fin y ser analizado por código, en tema de comparación de pulso, lo que le permite cubrir todas las variables anteriormente nombradas.

4.2.3 Pantalla LCD

La pantalla LCD elegida como monitor de variables y funciones del sintetizador de audio en la FPGA es una pantalla LCD module 2004, la cual incluye comunicación I²C (Inter Integrated Circuit) incorporada. La presente pantalla LCD, posee un alto contraste, con 4 líneas de escrituras y un total de 20 caracteres por cada línea[54]. En la Tabla 4.3, se presentan las especificaciones técnicas de la pantalla LCD module 2004.

Tabla 4.3. Ficha técnica Pantalla LCD module 2004[54].

Variable	Valor
1. Tipo de luz de fondo	Azul
2. Cantidad de muestreo	4 líneas x 20 caracteres
3. Voltaje de operación	5 V DC
4. Consumo máximo de corriente	25 mA

El tipo de comunicación de la presente pantalla es por medio del protocolo de comunicación I^2C , el cual posee solo dos pines, además de los pines de comunicación. El pin SCL, el cual como se había nombrado en las secciones 4.2.1 y 4.2.2, es el encargado de en este presente caso recibir la señal de reloj de la FPGA, con la cual ayuda a sincronizar los datos de envío y el funcionamiento de la presente pantalla. Los datos de envío es por medio del pin SDA, el cual se encargará en primera medida avisar que se encuentra lista la pantalla para recibir datos, y posterior a ello recibir los paquetes de datos de la FPGA, que se deseen escribir[54]. En la Fig. 4.6. se muestra el modelo físico de la pantalla LCD a trabajar.

Fig.4.6. Modelo físico pantalla LCD module 2004 más comunicador I^2C [55].

Mediante la información de los párrafos anteriores se eligió la pantalla LCD module 2004 con comunicación I^2C , por el hecho de poseer un amplio espacio de escritura, para poder muestrear todas las variables necesarias, en consideración de los modelos PO KO, mencionados en el capítulo 1.2.2 y su desarrollo de esquema en el capítulo 3.5. Además de lo anteriormente mencionado, está la inclusión del protocolo de comunicación el cual permite ahorrar el número de pines a ocupar a una cantidad de 2 pines, sin contabilizar los pines de alimentación (Vcc y GND).

- Configuración de memoria DDR
- Lógica de interconexión para Vivado IP – Interfaz PS
- Relojes e interrupciones del PL
- Reloj interno del PS



Fig.4.8. Núcleo Zynq 7 Processing System.

4.2.4.2 Processor System Reset

El bloque Processor System Reset encontrado en el software de Vivado, permite la configuración de las condiciones de reinicio en la entrada y generar las condiciones de adecuando para la salida. El presente bloque de la Fig.4.9 posee un total de 5 señales de entrada y cinco de salida[57].

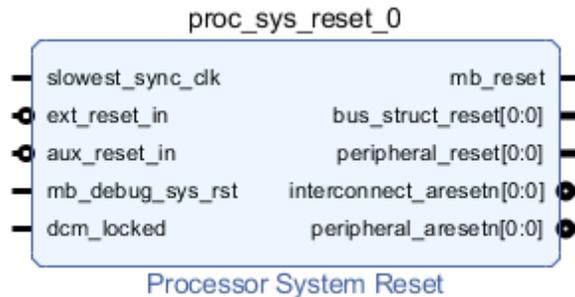


Fig.4.9. Núcleo Zynq 7 Processing System.

4.2.4.3 Advanced eXtensible Interface (AXI)

Por medio de las interfaces AXI (Advanced eXtensible Interface), es posible controlar las conexiones del PS y el PL, mostrados en el capítulo 2, permitiendo controlar las entradas y salidas básicas, las salidas de audio y los periféricos de los Pmod, entre otras funciones. Las cuales, proporcionan enlaces de alto ancho de banda y baja latencia[58]. Desde la

creación de las interfaces AXI en el año 2003, con su primera versión, hasta el año 2010 con su última versión con el AXI4, los AXI consta de tres tipos:

- AXI4: Es considerada una interfaz de alto rendimiento, que permite la comunicación mapeada en la memoria, logando una transferencia de datos de hasta 256 ciclos.
- AXI4-Lite: Es ocupada para realizar transacciones mapeadas en memoria en memoria individuales, lo que proporciona una única transferencia por transacción.
- AXI4-Stream: El presente AXI4 no ocupa mapeo en memoria y permite un tamaño de ráfaga de datos ilimitada.

Dentro del desarrollo del Hardware mixto ocupado a la hora de conformar un sintetizador de sonido, se ocuparon varios bloques AXIs, tales como AXI Interconnect, AXI GPIO y Zybo control audio. Este último, aunque no posea el nombre de AXI si cumple la misma función de estos, permitiendo controlar las salidas de audio. Estos núcleos son descritos a continuación.

4.2.4.3.1 AXI Interconnect

Dentro de las IP AXI de Vivado, se encuentra el núcleo AXI Interconnect, que posee la función de hacer transferencias mapeadas en memoria, permitiendo conectar uno o más dispositivos maestros/esclavos. En la Fig.4.10 se encuentra el núcleo AXI Interconnect.

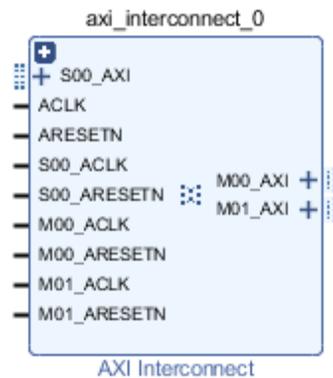


Fig.4.10. Núcleo AXI Interconnect.

4.2.4.3.2 AXI GPIO

El integrador IP de Vivado, correspondiente al núcleo AXI GPIO (General Purpose Input/Output), está encargado de controlar todos los periféricos tanto internos como

externos, de entrada/salida de la Zybo Z7, mediante el sistema de procesamientos, tales elementos fueron nombrados y explicados los capítulos 2.6 y 2.7. El núcleo que ocupa el AXI GPIO es por medio de una interfaz AXI4-Lite, los anchos de los canales pueden configurarse de manera independiente, mediante la habilitación de sus puertos como entrada o salida. Dentro del ancho de canal para pines GPIO se encuentra en el rango de 1 a 32 bits, junto a la programación de cada bit como entrada o salida[59]. La Fig.4.11 se presenta el núcleo AXI GPIO.

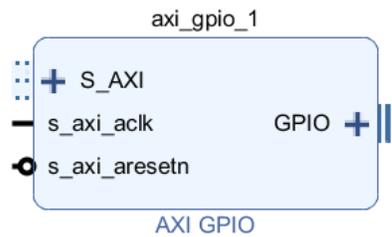


Fig.4.11. Núcleo AXI GPIO.

4.2.4.3.3 Zybo control audio

El integrador IP de Vivado, correspondiente al núcleo Zybo control audio, permite la comunicación y control del códec de audio SSM2603, explicados en el capítulo 2.5. Para el control de dicho núcleo es necesario el núcleo del AXI Interconnect, el cual permite la interacción del control del audio con la Zynq 7000. Del núcleo de la Zybo control audio, es posible la activación y control de los puertos de audio, por medio de las señales de reloj de su salida, correspondiente a los Jack de audio y al dato que se quiera enviar. La Fig.4.12 se presenta el núcleo de la Zybo audio ctrl.

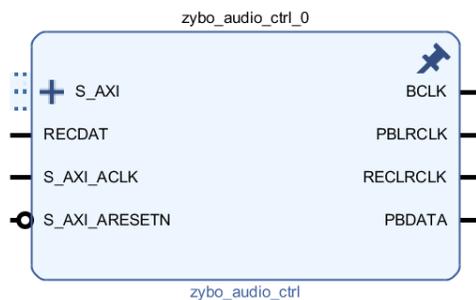


Fig.4.12. Núcleo Zybo audio ctrl.

4.3 Implementación del software para efectos de filtrados y barridos

Dentro de la principal utilidad se encuentra a la realización del filtrado mediante los filtros FIR y IIR, esta toda el área enfocada al discernimiento de la frecuencia de la señal de audio es fundamental a la hora de realizar los efectos de sonidos. Dichos efectos de sonidos poseen su desarrollo en los capítulos 1.2.3 de [a-f] y 3.3, en donde se simularon, sacando como resultado su gráfica y efectividad de filtrado, que se eligieron a implementar.

La obtención de los coeficientes para los efectos de filtrado es por medio del cálculo realizado por la función “fir1[60]” del software de Matlab[6], la cual su fundamento es la utilización de un filtro FIR basado en ventanas. Cuya finalidad es entregar los coeficientes, para realizar la modificación de los vectores de audio, causando dicho efecto a realizar, ya sea filtro pasa bajo, pasa banda o pasa alto, según las especificaciones del efecto a aplicar.

La aplicación de los coeficientes entregados por el filtro FIR se logra apreciar en la siguiente ecuación (4.11). En donde la variable “y” es el vector modificado, “b(n)” el valor del coeficiente a aplicar y “x(n)” el valor de los vectores de audio, de la pista de sonido original. Es de considerar que la efectividad del filtrado del audio depende del número de coeficientes utilizados y la frecuencia de corte que se desea aplicar.

$$Y(n) = b(1) * x(n) + b(2) * x(n - 1) + \dots + b(nb + 1) * x(n - nb) \quad (4.11)$$

Continuado con el fundamento de la utilidad del filtrado, enfocado en el presente caso por los dos tipos de filtrado FIR y IIR, se encuentran los efectos de barrido, que para su aplicación es necesario combinar ambas definiciones, mediante la obtención de sus coeficientes dados por la función “butter[61]” del software de Matlab[6]. Dicha función posee similitud con la función “fir1[60]”, a la hora de seleccionar que tipo de filtro se desea ocupar si pasa alto, bajo o pasa banda, pero con la principal diferencia que esta función entrega dos coeficientes, una para ser ocupada en la sección FIR y otra para la sección IIR.

En la siguiente ecuación (4.12), se muestra la combinación de las estrategias de filtrado. En donde se agrega una nueva variable la cual es “a(n)”, correspondiente al coeficiente del filtro IIR ocupado. Al igual que en el caso del filtro anterior la calidad del

efecto de barrido se verá influenciada por la cantidad de coeficientes ocupados y la frecuencia de corte a elección.

$$a(1)Y(n) = b(1) * x(n) + b(2) * x(n - 1) + \dots + b(nb + 1) * x(n - nb) - a(2) * y(n - 1) \dots + a(na + 1) * y(n - na) \quad (4.12)$$

4.4 Análisis de costos del sintetizador de sonido

Frente a los precios presentados en la Tabla 1.5, se realizará el comparativo de costos ocupados en la presente memoria, con la finalidad de dar a conocer si la memoria es más económica que el resto de sintetizadores de la serie PO KO o si se mantiene dentro de los precios del mercado de dichos sintetizadores. Hay que considerar también la diferencia de precio de los elementos disponibles, por el simple hecho que un producto generado en masa es mucho más barato, que la obtención de cada elemento necesario por separado.

Para el desarrollo de la memoria del diseño e implementación de un sintetizador de audio en una plataforma Zynq-7000, se realizaron las compras de manera online por tema de la pandemia y a proveedores locales o de entrega en un tiempo moderado, con la finalidad de no perder mucho tiempo por problemas de insumos. En la presente Tabla 4.4, se muestra los principales elementos ocupados y sus respectivas cotizaciones, ante proveedor o precio mercado disponible.

Mediante el cálculo generado como producto de la Tabla 4.4, se logra un total de 74.899 (CLP), lo cual es pasado por 2899(CLP) en comparación del sintetizador PO-20 KO y considerando la aplicación de la tarjeta Zybo Z7 da un valor de 178.272 (CLP). El presente resultado no es un precio muy alejado a lo ofrecido en el mercado y considerando el tema de producción en masa, costos de disponibilidad de insumos y ventajas ofrecidas por lo expuesto en la memoria, lo que da que el sintetizador creado es una buena opción.

Tabla 4.4.Elementos principales ocupados en la implementación del sintetizador de audio.

Equipo	Precio Promedio(CLP)
Mini parlantes Usb	4.999[62]
Matriz táctil 4x4 teclera capacitiva	5.900[63]
2x Encoder rotatorio brick KY-040 rotary 360	4.200[64]
Pantalla LCD2004 con I2C incorporado	5.400[65]
Chip XC7Z010	54.400[30]
Tarjeta Zybo Z7-10	157.773[31]

5 Resultados experimentales

5.1 Introducción

En el capítulo 5 se mostrarán los resultados experimentales obtenidos del sintetizador implementado en la tarjeta Zybo Z7, mediante la plataforma de la Zynq-7000. Estas grabaciones no fueron realizadas en un ambiente adecuado, por lo cual algunos resultados pueden presentar distorsión con respecto a los resultados en vivo.

5.2 Implementación de los efectos de sonido del sintetizador

En la sección 5.2 se tocarán los tópicos sobre el trabajo experimental e implementación del sintetizador de audio, en una plataforma Zynq 7000. Comenzando con la sección 5.2.1, en donde se dará a explicar sobre cómo se obtuvo las muestras de audio, su tratamiento post implementación, y las condiciones que presentan dichas tomas de prueba de audio.

En las secciones 5.2.2 a la 5.2.14, se tocará a fondo todo lo concerniente a toma de pruebas de audio, empezando desde la señal original, hasta el efecto 12 de los efectos de sonido. Puesto que, por motivos de pandemia y falta de condiciones de implementación, no se logró aplicar todos los efectos de sonido. Los contenidos de cada capítulo de implementación se harán un comparativo de la simulación realizada y lo obtenido de manera experimental. Dando a conocer sus diferencias y explicando sus gráficas comparativas, de simulación versus implementación.

5.2.1 Adquisición y post tratamiento de la señal

Para adquirir los datos procesados por el sintetizador se utilizó como medio de grabación un celular, por lo cual, las mediciones no están libres de ruido ambiente. Es por ello que también hay que hacer la consideración, de que la grabadora del celular, da prioridad al mensaje del audio. Esto puede llegar a deformar la señal de audio esperado, aunque el contenido del sonido sea prácticamente igual al original.

Posterior a la grabación de audio se hace un tratamiento similar a la obtención de los sonidos del modo patrón y del modo efecto de sonido. En donde para el presente caso se

obtiene un archivo “.m4a” (MPEG-4), el cual es un archivo de audio de igual o superior nivel de calidad que el “.mp3” (MPEG-2 Audio Layer III)[66]. Pero para ser tratado y modificado, dejando solo el audio, emitido por la FPGA, es necesario hacer la conversión a un archivo “.wav” (Waveform Audio File Format). Por la condicionante, que este formato de archivo(.wav) es posible leerlo en el FL Studio[34] y hacer el tratamiento de audio, eliminando todos los espacios de tiempos en donde no haya sonido. Dejando de esta manera únicamente el audio del efecto a graficar.

Después del tratamiento de recorte de la grabación, se obtiene únicamente el audio necesario, con el contenido deseado, en el cual, se procede a enviar los datos al software de Matlab[6]. En este software se repite el mismo procedimiento de control de datos de audio, que se explicó en el capítulo 3.1.1. Mediante la lectura del archivo de audio, se es capaz de transformar dicha señal a una vectorizada, dentro de los rangos de los gráficos vistos en el capítulo 3.3, con lo cual es posible analizar los audios de los efectos de sonido, es importante destacar que no todas las señales se verán iguales a la simulación por los procesos anteriormente mencionados, también que las señales de audio, se mantendrán entorno a un único eje, lo que provocara que las señales de baja magnitud, sean representadas con una amplitud menor, en vez de estar desplazadas a una magnitud de menor valor.

Todo el procedimiento de adquisición de datos y post tratamiento de la señal se presente el diagrama de la Fig.5.1.



Fig.5.1. Diagrama de grabación y post tratamiento del audio.

5.2.2 Implementación del audio original

Para la presente prueba del audio original versus su implementación en la FPGA. Se trabajará con el primer audio, de la Tabla 3.3, tal como se realizó para todas las pruebas de efectos de sonido, del capítulo tres. La representación gráfica de lo anteriormente planteado, se muestra en la Fig. 5.2.

La primera gráfica de la Fig.5.2 muestra la representación vectorial del sonido 1 de la Tabla 3.3, la cual se le hizo el mismo tratamiento que en el capítulo 3, por medio del

procesamiento de la Fig.3.8. La segunda gráfica de la Fig.5.2 es el resultado obtenido por la reproducción del sonido 1 de la Tabla 3.3 en la FPGA Zybo Z7, dicha señal para graficarla ha pasado por el procesamiento explicado en la sección 5.2.1 y ejemplificado en la Fig.5.1.

De la Fig.5.2 se realiza el comparativo de las dos gráficas con la finalidad de ver la diferencia entre la simulación y la aplicación experimental del sonido. De este proceso se logra dilucidar que ambas gráficas, aunque contiene el mismo sonido, son representadas de manera diferente en sus gráficas. En donde la señal de la implementación del sonido 1 posee una silueta similar y un tiempo de reproducción, dentro del mismo rango de la señal original. Los puntos en que difieren las gráficas de la Fig.5.2 es la aparición de una mayor cantidad de ruido en la señal en la segunda gráfica, por el hecho del procedimiento de grabación. Tal caso se logra confirmar, puesto que en la señal original cercano a los 1.2 segundos, se encuentra un valle donde la señal no presenta cambio, mientras que en su implementación existe ruido en ella.

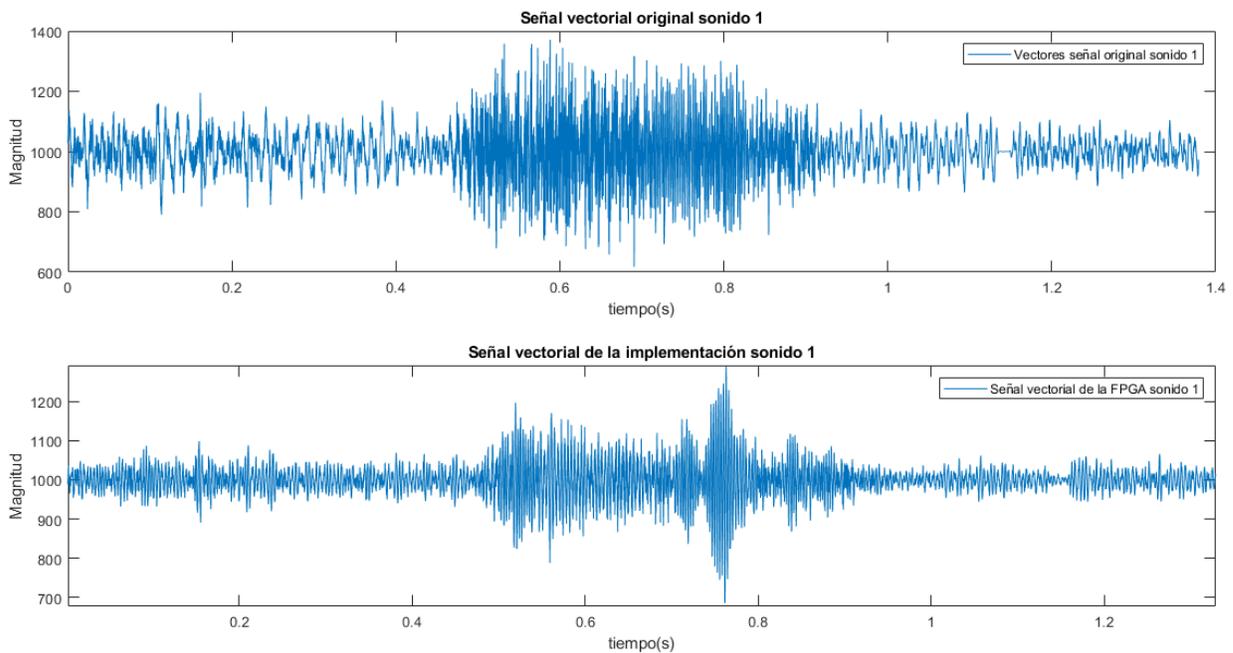


Fig. 5.2. Gráfica original vectorial del audio 1 vs su implementación procesada en la FPGA.

Con la finalidad de comprobar la representatividad de la señal, de la implementación del sonido 1, se procuró aplicar la transformada de Fourier a ambas gráficas. Dichas gráficas se presentan en la Fig.5.3.

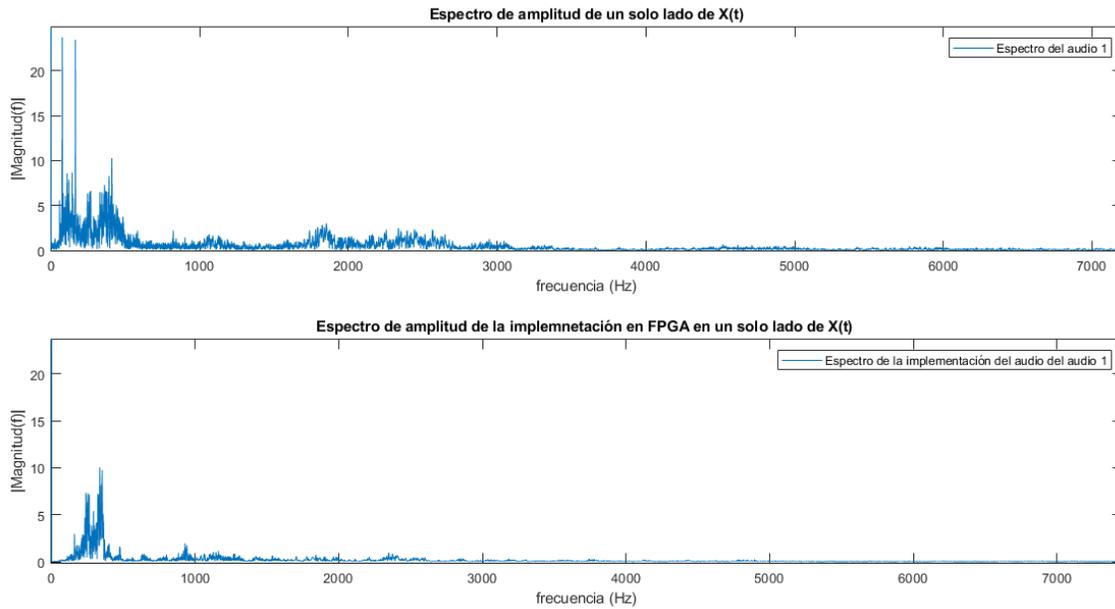


Fig.5.3. Transformada de Fourier de la Fig.5.2.

Ante lo presentado en las gráficas de la Fig. 5.3, se logra ver que por motivos de grabación se atenúan las frecuencias de la implementación, además de una pérdida en los primeros 100 Hz. Pero sigue poseyendo el mismo espectro que la señal original, después de los 100 Hz, solo que un poco más atenuada, por el proceso de grabación y reproducción de audio de la FPGA.

5.2.3 Implementación del efecto 1 Repetir por 2

Para la implementación del efecto 1, correspondiente a repetir por 2 la señal de audio posee el mismo funcionamiento, que el explicado en el capítulo 3.3.2. En el presente caso, al igual que en todos los siguientes efectos de sonido, se hará el comparativo de la simulación versus la implementación. Dichas gráficas se muestran en la Fig.5.4.

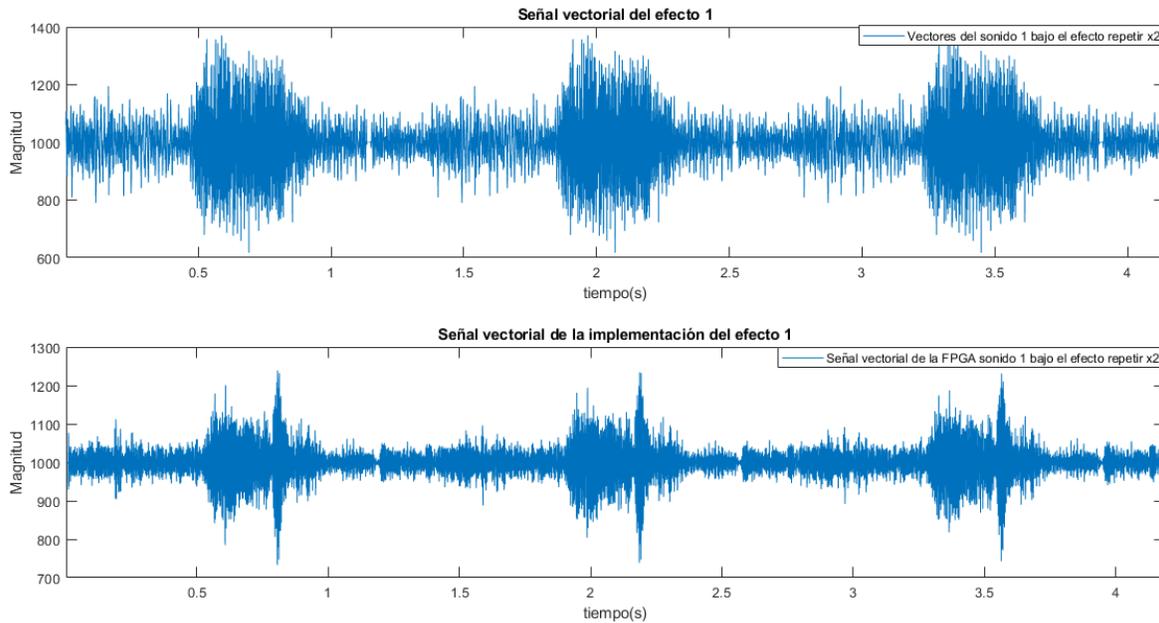


Fig.5.4. Gráfica de la simulación del efecto de repetir por 2 del audio 1 vs su implementación en la FPGA.

La gráfica de la Fig. 5.4, aunque las señales de la implementación y lo simulado, no sean igual como se explicó en la sección 5.1.2, se cumplen los objetivos del efecto de sonido, puesto que en ambas se repite dos veces más, el sonido del audio 1. Por lo cual se triplica la duración total del audio, en donde en ambos casos la duración del tiempo es similar.

5.2.4 Implementación del efecto 2 Repetir por 4

El efecto número 2 correspondiente a repetir por 4. el objetivo a cumplir es que la señal obtenida se repita un total de 4 veces. La gráfica de la simulación versus la implementación se muestra en la Fig. 5.5.

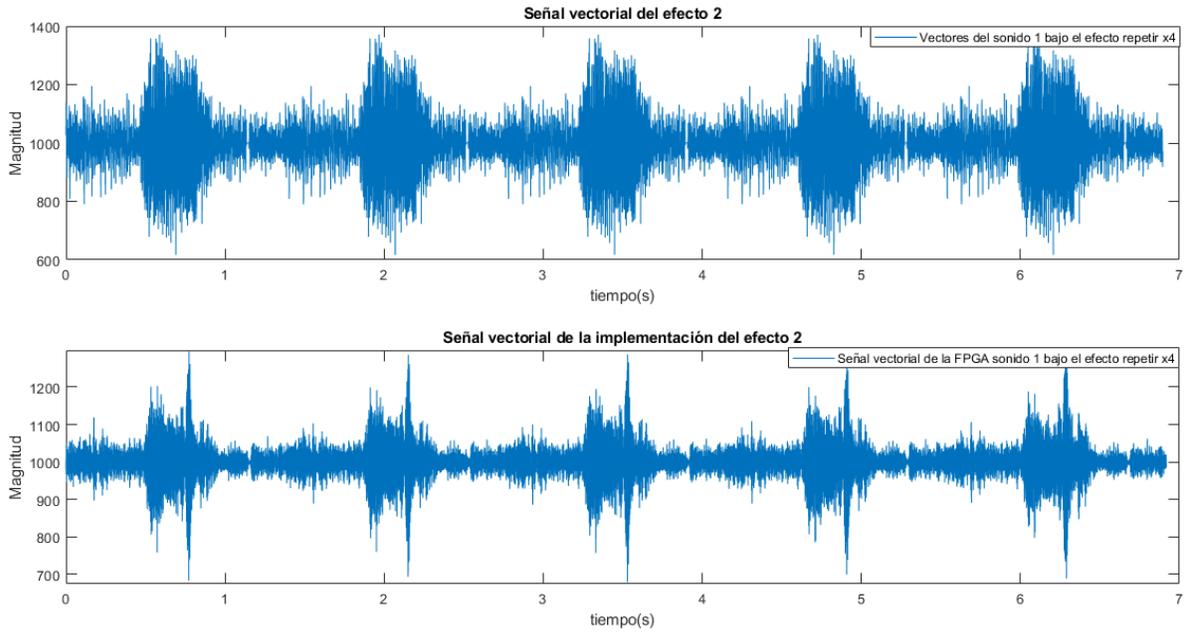


Fig.5.5. Gráfica de la simulación del efecto de repetir por 2 del audio 1 vs su implementación en la FPGA.

Frente a la implementación se cumple el objetivo buscado. Por el hecho que la señal obtenida se repite un total de cuatro veces más y su duración entra en el mismo rango de tiempo que el de la señal simulada. Aunque no posea la misma forma de onda, la señal simulada con la señal de la implementación, sí posee similitud la forma de onda de la implementación con el audio 1 obtenido en la segunda gráfica de la Fig. 5.2.

5.2.5 Implementación del efecto 3 Inversión

Para la señal deseada de la implementación del efecto 3 de Inversión, se busca obtener la señal inversa a la vista en la Fig.5.2. En esta Fig.5.2 se espera que posea el mismo periodo de tiempo del sonido 1, pero en orden inverso. El funcionamiento de dicho efecto se explica en el capítulo 3.3.4. Las gráficas de la simulación versus la implementación, se muestran en la Fig. 5.6.

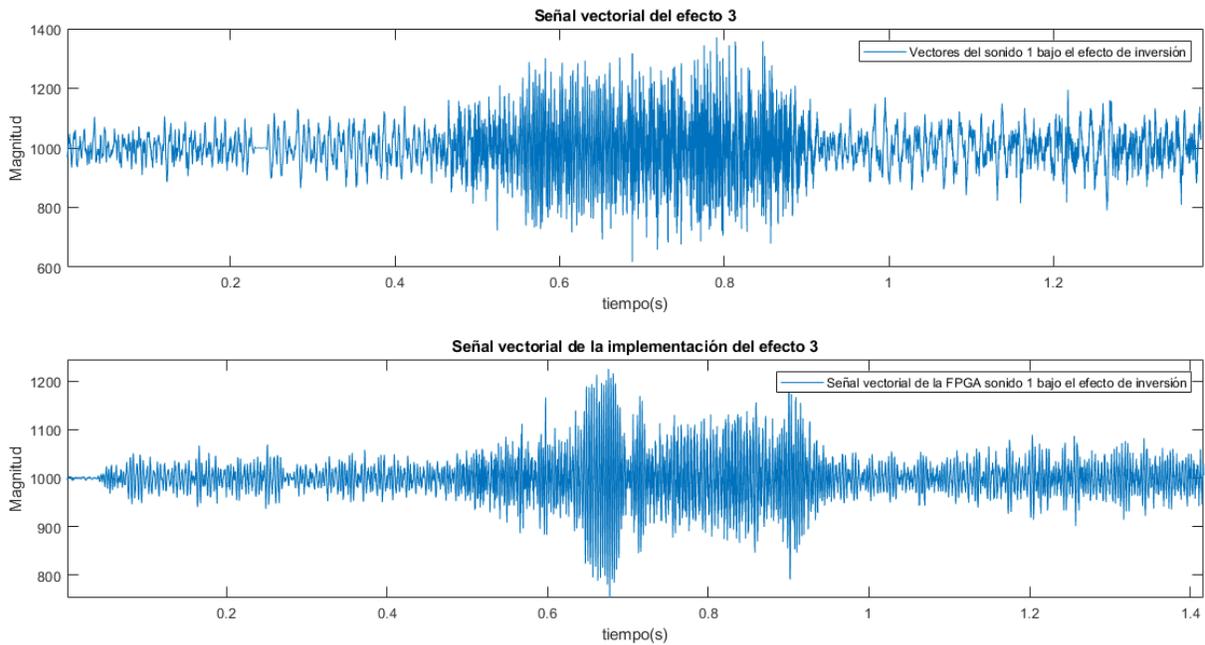


Fig.5.6. Gráfica de la simulación del efecto de inversión del audio 1 vs su implementación en la FPGA.

En la Fig.5.6, se logra dar a conocer que las gráficas de la simulación y de la implementación, son las señales inversas a las vistas en la Fig. 5.2. Cumpliendo la misma forma de onda, pero invertida y el mismo periodo de tiempo, de la señal de original, de la simulación e implementación.

5.2.6 Implementación del efecto 4 Fill in

El efecto 4 correspondiente a Fill in, su funcionamiento se basa en introducir pequeños espacios de tiempo, donde no se emitirá sonido. El presente efecto posee una explicación más completa en el capítulo 3.3.5. Lo buscado en la señal de la implementación, es que se cumpla los tiempos de vacío de la señal, sabiendo que puede tener una mínima amplitud de ruido, por el proceso de grabación. También que se encuentre dentro de los tiempos de la simulación. Dichas gráficas de la simulación del efecto versus la implementación, se muestra en la Fig. 5.7.

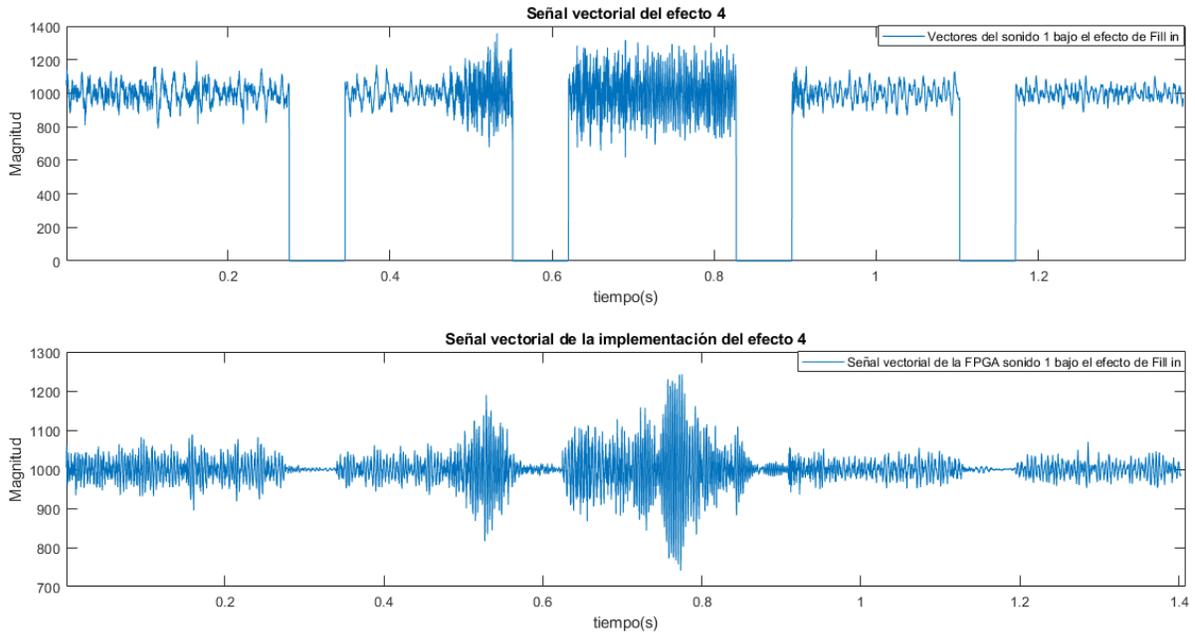


Fig.5.7. Gráfica de la simulación del efecto de Fill in del audio 1 vs su implementación en la FPGA.

En la gráfica de la señal de implementación de la Fig.5.7, se logra ver que se cumple el objetivo de la aplicación del efecto 4, puesto que, se posee los tiempos de espacios de vacío como se muestra en la gráfica de simulación. En el caso de la implementación se muestra de manera continua, a través del eje de los 1000, esto es debido que como se había dicho en la sección 5.2.1 por el tratamiento del audio provoca que se muestre de manera continua. En el presente caso los tiempos de vacío de la implementación, se muestran como un valor casi estable de la magnitud del valor de 1000, con una amplitud casi nula. Por lo cual, se logra concluir que, si es efectivo el efecto de sonido, por los tiempos de vacío de la simulación en donde no hay ningún cambio en la amplitud, también se cumple que ambos están aproximadamente dentro del mismo periodo de tiempo y que respetan sus versiones originales, de la Fig. 5.2.

5.2.7 Implementación del efecto 5 Bouncing ball

Para la implementación del efecto 5 Bouncing ball, se debe cumplir que la señal de la implementación se aplique los tres rebotes programados. Cada rebote debe ser de una magnitud menor, tal como se explica en el capítulo 3.3.6. En el capítulo de simulación se muestra el caso ideal, separando cada rebote de la señal original, logrando apreciar de una mejor manera, cada rebote y su amplitud. Para el presente caso de la Fig. 5.8, se obtiene tres

gráficas, las primeras gráficas corresponde a la señal simulada y la tercera a la señal de la implementación. Para la primera gráfica se tiene el caso ideal, con cada rebote por separado al audio 1. En la segunda gráfica, se obtiene el caso real de la simulación, obteniendo el audio 1 y los rebotes de este, combinadas en una sola gráfica, dando a entender que así serán los datos esperados por la FPGA. Por último, se tiene la tercera gráfica de la implementación, la cual tiene que seguir el mismo patrón de la segunda gráfica, para cumplir con su efectividad del efecto 5.

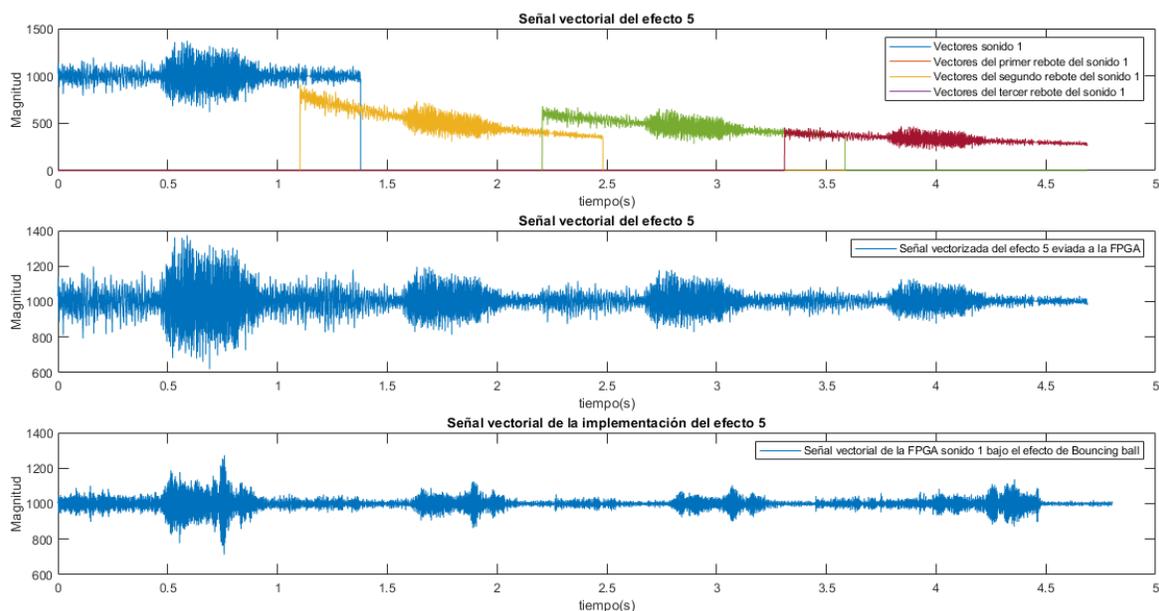


Fig.5.8. Gráfica ideal y real de la simulación del efecto de Bouncing ball del audio 1 vs su implementación en la FPGA.

Para la primera gráfica de la Fig.5.8 se obtiene el caso ideal del efecto 5 sobre el sonido 1, identificando cada rebote por separado. Los cuales irán disminuyendo con el tiempo su magnitud. La importancia de apreciar la primera gráfica, es lograr identificar la señal original y cada rebote por separado, además de observar como estos van disminuyendo con el tiempo.

En la segunda gráfica de la Fig.5.8, es una versión más realista. Puesto que muestra la simulación de como realmente sería la implementación. Esto es debido a que toda la señal de sonido se encuentra en el mismo eje en el tiempo que transcurre, combinando la señal original con los rebotes incluidos. Es de destacar que cada rebote en la primera gráfica se muestra con una menor magnitud, disminuyendo en el tiempo, mientras que en la segunda

gráfica al tener que esta todas las señales combinadas, cada rebote es representado con una amplitud cada vez menor. Esto es debido que una menor magnitud, al ser grabada y unida a toda la señal en un mismo eje, esta es representada por una señal de menor amplitud. Por ende, cada rebote tendrá una menor amplitud, en el eje de magnitud.

En el tercer grafico de la Fig.5.8 se da a conocer la implementación del efecto 5 sobre el sonido 1, en la FPGA. Para dicho caso se logra dar a cuenta, que posee el mismo patrón que lo visto en la segunda gráfica, de dicha Fig.5.8. En la cual, se destaca que la señal vectorial del audio es constante y que, en cada rebote va disminuyendo la amplitud del audio. En las tres gráficas, se encuentran dentro del mismo periodo de tiempo, esto da a entender que la aplicación del efecto de Bouncing ball si es efectiva.

5.2.8 Implementación del efecto 6 Glich

En el anterior capítulo 3.3.7 se fundamentó el funcionamiento del efecto 6 de Glich. En dicho capítulo se dio a comprender, que el funcionamiento del efecto 6 es similar que al del efecto 4 de Fill in, solo que, en vez de crear tiempos sin reproducir el audio, se procederá a agregar en esos tiempos un sonido externo. Haciendo referencia a los ruidos de fallo, que había en los computadores antiguos. La Fig.5.9 al igual que en la sección anterior 5.2.7, se procederá a mostrar tres gráficos, uno de simulación, uno de la simulación del caso real y otro de la implementación.

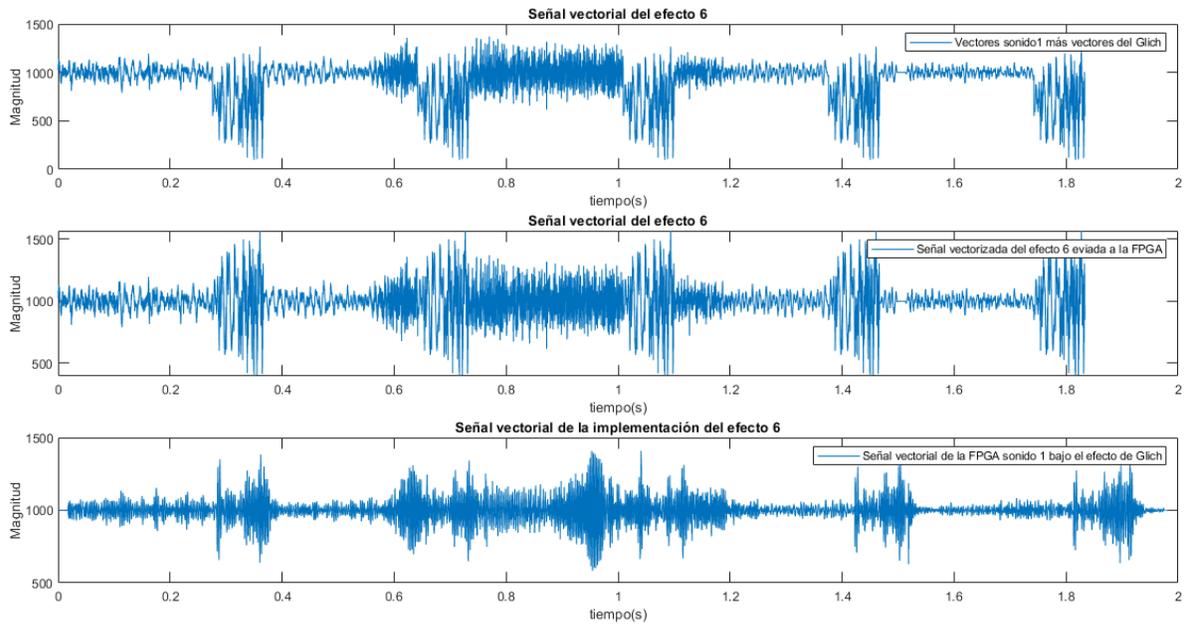


Fig.5.9. Gráfica ideal y real de la simulación del efecto de Glich del audio 1 vs su implementación en la FPGA.

Al igual que en la sección 5.2.7 se presenta la primera gráfica, la cual representa el caso ideal y se logra diferenciar la inclusión del sonido del Glich. Puesto que, para el caso ideal, el sonido del Glich posee una menor magnitud, que el audio del sonido 1. Es importante recalcar que el tiempo de la señal aumenta, por la inclusión de las repeticiones del sonido Glich original más la del sonido 1.

Para la segunda gráfica, se tiene un caso más realista donde el sonido del Glich y el sonido 1 se encuentra en el mismo eje de la magnitud de 1000, por lo cual ambas señales se encuentran combinadas. La actual gráfica simulada es la esperada por la parte de implementación.

En la tercera gráfica se muestra la modificación del sonido del Glich, al igual que la modificación del sonido 1, por el procedimiento de grabación y tratamiento del audio de la sección 5.2.1. Aunque se esté emitiendo el mismo contenido, la forma de la señal es diferente, pero por lo contrario cumple con el patrón esperado. Es concluible que cumple con lo esperado, se repite los patrones del sonido 1 más las repeticiones del Glich, además de estar dentro de los rangos de tiempo esperado, por las anteriores dos gráficas. Aunque se pase un poco de tiempo, debido a la gran cantidad de cálculos realizados por la FPGA, provocando que la reproducción del audio sea un poco más lenta para el presente caso.

5.2.9 Implementación del efecto 7 Distorsión

En la sección 5.2.9 se implementará el efecto 7 de distorsión en la FPGA. Tal como se desarrolló en su simulación en el capítulo 3.3.8. Para desarrollar el efecto de distorsión, se provoca multiplicando a cada vector de audio por una ganancia cercana a uno y agregándole otra ganancia de pequeño valor, en función del vector de audio obtenido. Para la gráfica de la Fig.5.10 se representan la simulación versus al resultado de la implantación.

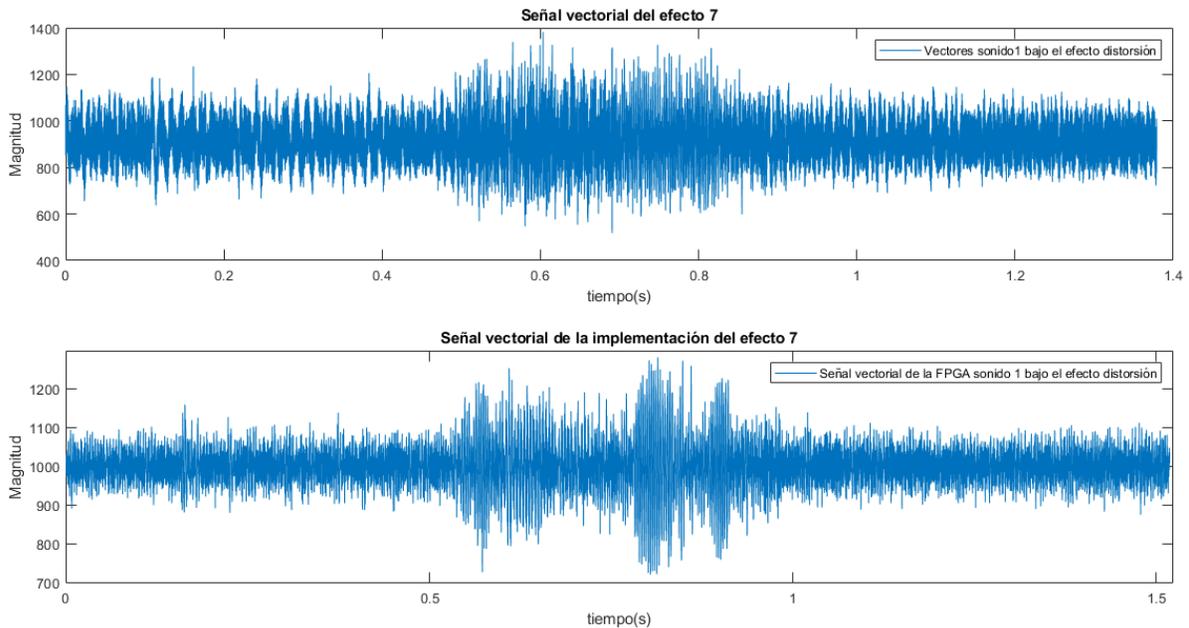


Fig.5.10. Gráfica de la simulación del efecto de Distorsión del audio 1 vs su implementación en la FPGA.

La Fig.5.10 se logra dilucidar que las formas de ondas vectoriales de la simulación e implementación conservan su silueta base, vista en la Fig.5.2. Pero por el procedimiento de distorsión, se ve un aumento del ruido de la señal en ambas gráficas de la Fig.5.10. Ante lo cual, es posible verificar y comprobar, que efectivamente se está realizando el efecto de distorsión de la señal de implementación.

5.2.10 Implementación del efecto 8 Half rate

El efecto 8 correspondiente a Half rate, en donde su objetivo es dividir la velocidad de reproducción del audio a la mitad, tal como se explicó en el capítulo 3.3.9. Esto se logra como se explica en el capítulo citado anteriormente, mediante el aumento de los vectores al doble de su versión original. Esto es posible mediante la técnica de intercalar la señal original,

con el promedio de la posición actual con la siguiente. La Fig.5.11, se muestra la simulación de dicho efecto versus su implementación.

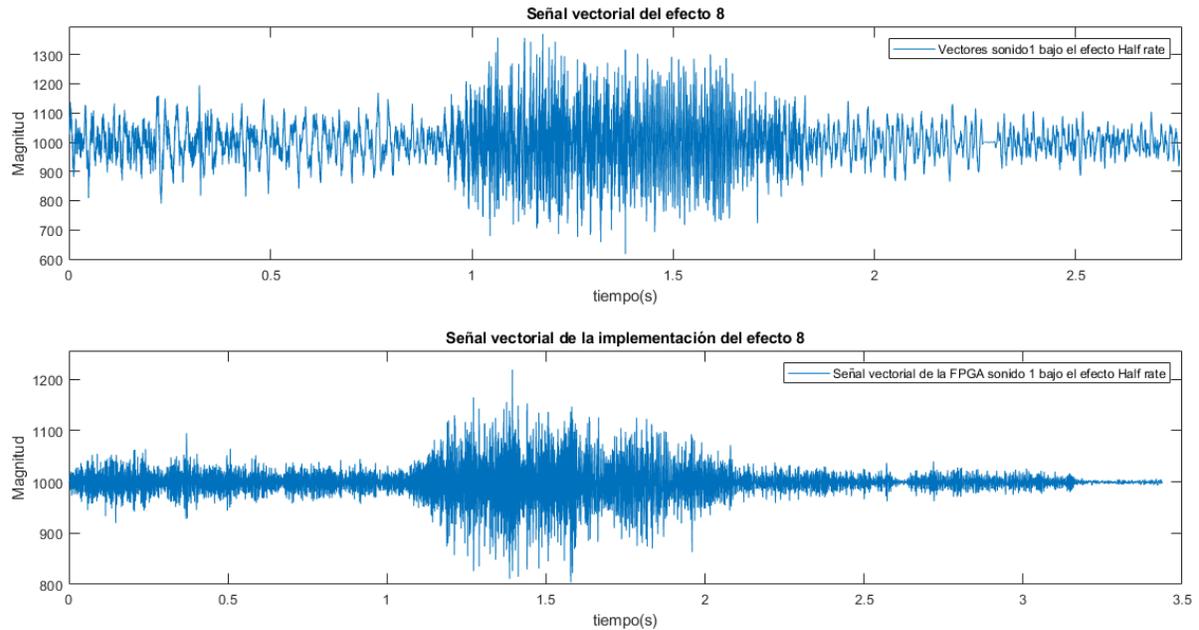


Fig.5.11. Gráfica de la simulación del efecto de Half rate del audio 1 vs su implementación en la FPGA.

La Fig.5.11 aunque se conserven la misma forma de onda base, vista en la Fig.5.2, el tiempo para la simulación se ha duplicado, como era esperado, confirmando que se dividió la velocidad de reproducción a la mitad. Mientras que, en la segunda gráfica correspondiente a la implementación se cumple el objetivo de disminuir la velocidad de reproducción sobre la mitad, pero no se cumple con exactitud la reducción de la velocidad a la mitad, en vez de ello el tiempo de reproducción es un 17% más que la simulación. Esta reducción provoca que, la implementación sea más lenta, que la mitad de velocidad de la señal original, pero no exactamente la mitad, si no un 242% más lenta. Considerando que el 100% sería la velocidad original del sonido 1 de la implementación. Esto es provocado, por la capacidad de reproducción de la FPGA, que, por la cantidad de cálculos necesarios, provoca que se demore un poco más la velocidad de reproducción, sumado al procedimiento de grabación de audio, visto en la sección 5.1.1.

5.2.11 Implementación del efecto 9 Feedback

El efecto número 9 correspondiente a Feedback, posee la característica, de tomar la señal original del audio y reingresarla con un leve desfase de tiempo. En donde al momento

de reingresar la señal, se procede a modificar dicha señal de audio, por medio de una serie de cálculos. Estos cálculos son realizados, mediante la definición de Feedback, la cual hace modificación de los parámetros de entrada, por medio de ecuaciones en base a diferentes sistemas. Para una explicación más profunda, es recomendable revisar el capítulo 3.3.10. La aplicación de dicho efecto en el sonido 1, de la simulación versus la implementación, se representan en la Fig.5.12.

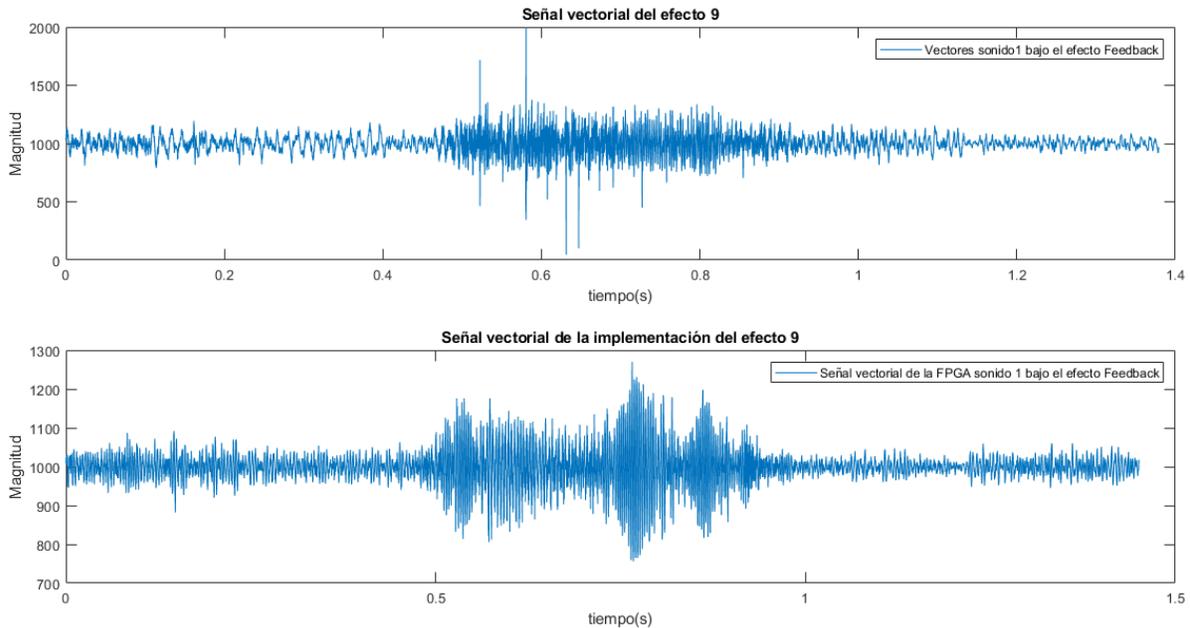


Fig.5.12. Gráfica de la simulación del efecto de Feedback del audio 1 vs su implementación en la FPGA.

Para el presente caso del efecto número 9, solo es posible como estándares de efectividad del efecto, analizar su comportamiento en la implementación. En donde para la simulación se ve una mayor atenuación de la señal, solo teniendo leves pick, en donde pasa a tener el valor de 0 y 2000, en leves periodos de tiempos. Por parte de la implementación, se logra un mayor efecto, puesto que gran parte de la señal del audio es modificada, dando a entender, que la señal del efecto si es afectada, por el proceso de cálculo del efecto 9. La señal de implementación posee un patrón diferente, a la obtenida en la Fig.5.2. Es importante recalcar que, mediante el acondicionamiento de la señal y su respectiva grabación, no se encontraron los grandes pick que si demuestra la simulación.

5.2.12 Implementación del efecto 10 Filtro pasa alto

Para el efecto 10 correspondiente a la aplicación de un filtro pasa alto, el cual consiste en un filtro que solo deja pasar frecuencias sobre la frecuencia de corte elegida. La fundamentación de la frecuencia de corte, su funcionamiento, se ha explicado en el capítulo 3.3.11. Para el cálculo de coeficientes se ocupó la ecuación (4.11), explicada en el capítulo 4.3. Las gráficas de simulación versus implementación, se muestra en la Fig.5.13

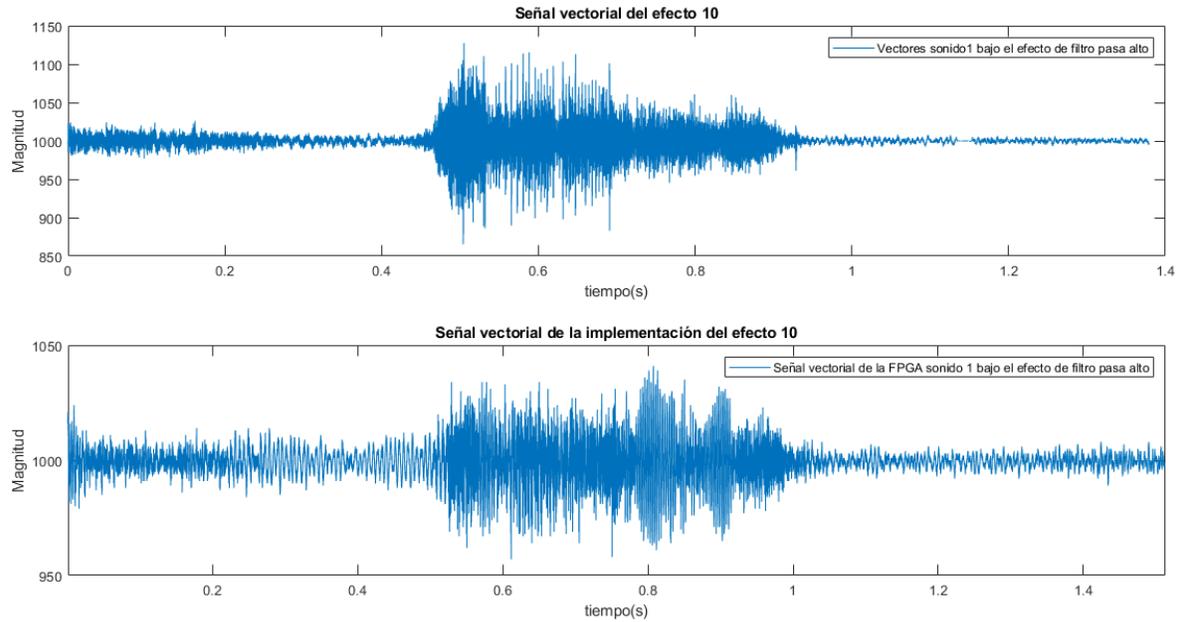


Fig.5.13. Gráfica de la simulación del efecto de Filtro pasa alto del audio 1 vs su implementación en la FPGA.

Tal como se había explicado en el capítulo 3.3.11, ante el procedimiento de filtrado pasa alto, la gráfica de la señal se representa más comprimida y junta. Este efecto es debido a que a una mayor frecuencia menor es la longitud de onda, por ende, la señal filtrada resultante se mostrara más comprimida. Con la finalidad de comprobar su efectividad de filtrado, se realizó su medición por medio de la transformada de Fourier, con el fin de ver el espectro de la frecuencia de ambas señales. Dichas gráficas se muestran en la Fig.5.14.

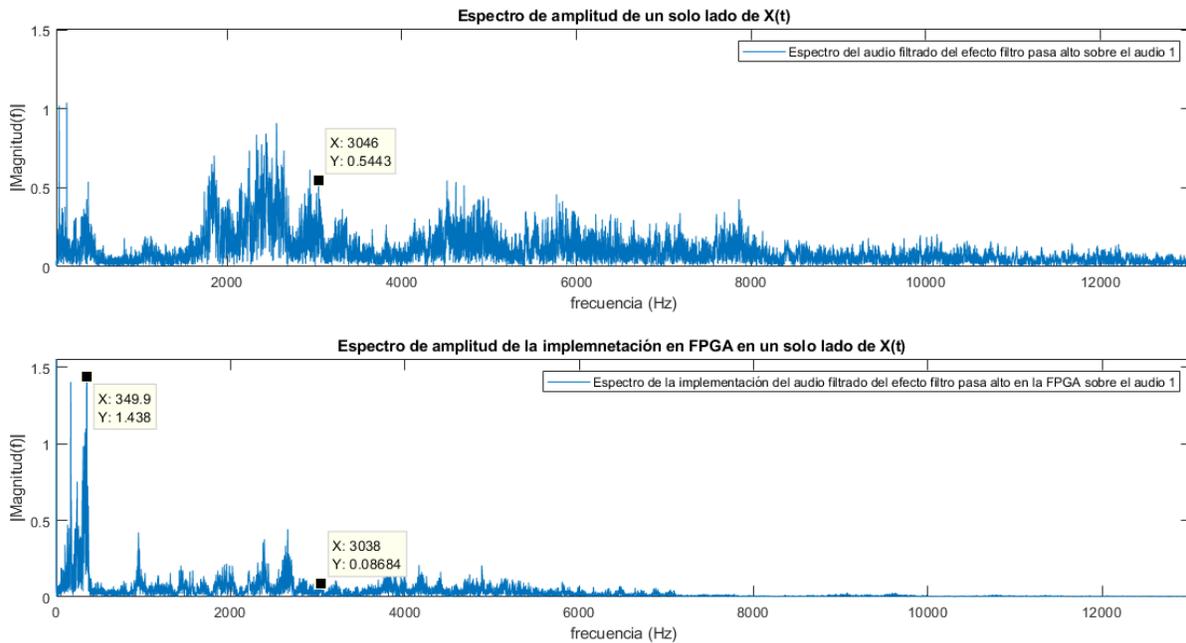


Fig.5.14. Transformada de Fourier de la Fig.5.13.

El análisis que se logra ver de la Fig.5.14, se logra contraponer con la transformada de Fourier de su señal original de la Fig.5.3. En donde para el presente caso del filtrado se ve una gran atenuación de las señales de baja frecuencia, por debajo de la frecuencia de corte de 3000 Hz. En el presente caso es posible ver, que, para el caso de la implementación, es mayor la atenuación de las frecuencias de baja frecuencia, esta variación, es debido al proceso de grabación, la cual deja pasar el espectro de baja frecuencia. Además, es posible observar su efectividad, con solo compararla con el espectro de la señal original, de la Fig.5.3, puesto que se redujo en gran medida las señales de baja frecuencia, mientras que las señales sobre la frecuencia quedan totalmente intactas.

5.2.13 Implementación del efecto 11 Filtro pasa bajo

Para la realización de la implementación del efecto 11 correspondiente al filtro pasa bajo, se realizó la aplicación de la estrategia de filtrado, de los filtros FIR. Además de ocupar una frecuencia de corte de 1000 Hz, lo que indica que solo pasaran en su mayoría frecuencias bajo esta frecuencia de corte. La fundamentación y lógica del efecto, fueron explicadas en el capítulo 3.3.12. Para el actual caso, las gráficas de simulación versus la implementación, del presente efecto, se muestran en la Fig.5.15.

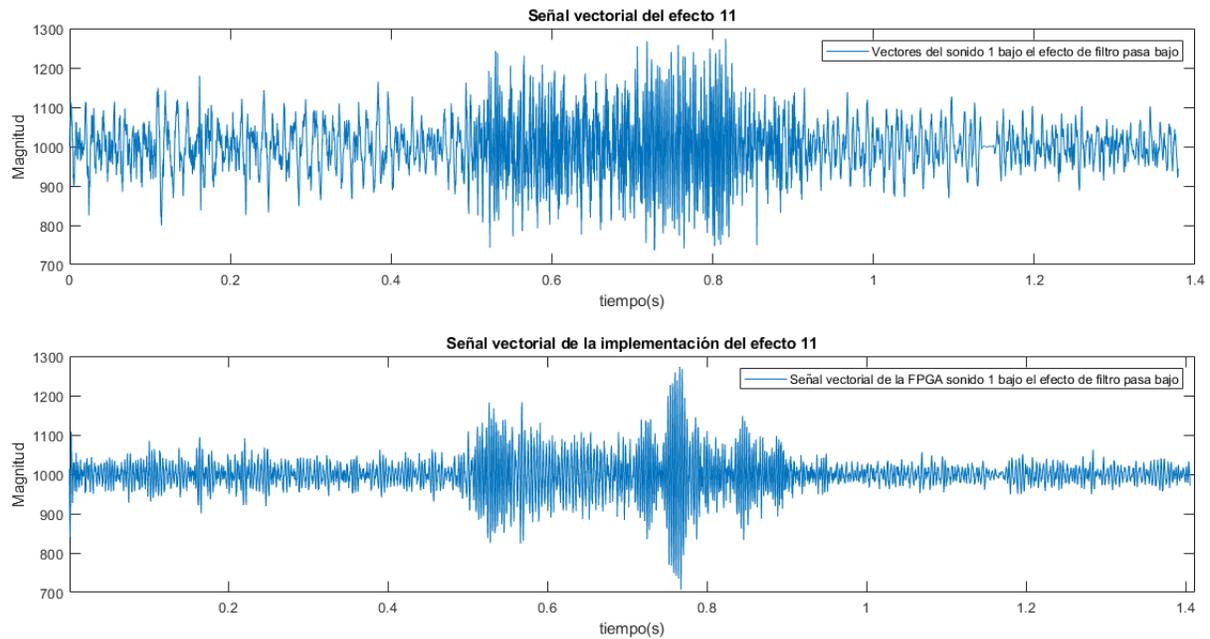


Fig.5.15. Gráfica de la simulación del efecto de Filtro pasa bajo del audio 1 vs su implementación en la FPGA.

En las gráficas de la Fig.4.25, ocurre el proceso inverso a lo visto en la sección 5.1.12, en la Fig.5.13. Puesto que, para el presente caso del efecto 11, las formas de onda son un poco más dispersas, por el hecho al ser bajas frecuencias, implican que la longitud de onda es mayor y por lo cual, la forma de onda es más espaciada, esto se aplica para ambas gráficas, de simulación e implementación. Pero en la implementación no es perfectamente visible por las frecuencias altas, que se pueden filtrar por el proceso de grabación del audio. Para comprobar la efectividad del filtrado, se procede a calcular la transformada de Fourier de las gráficas de la Fig.5.16.

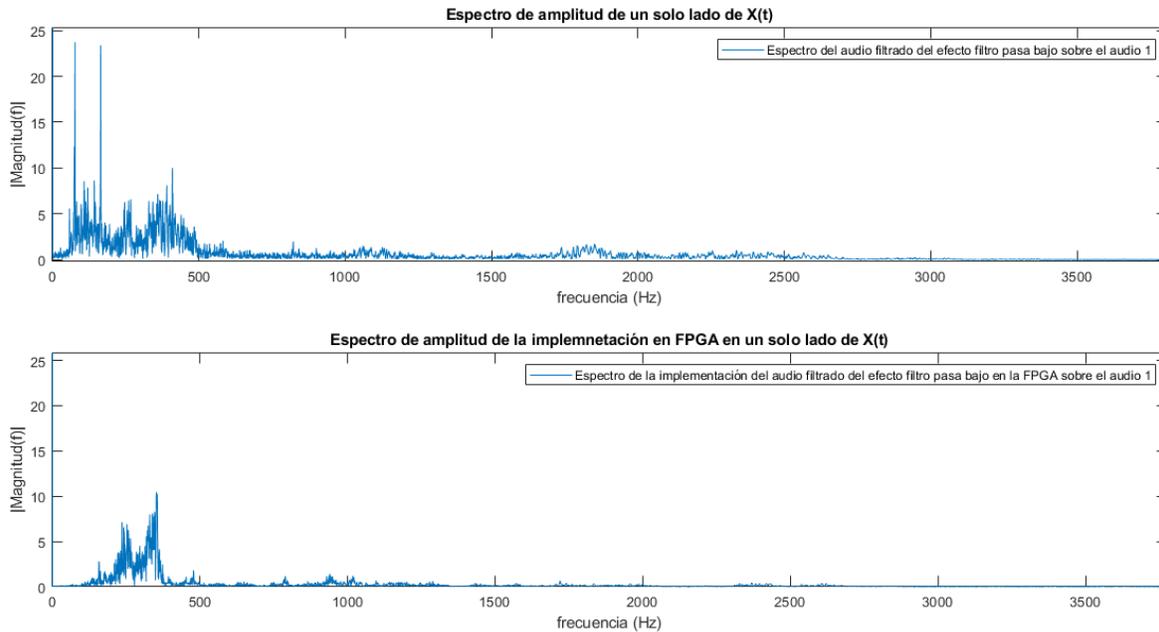


Fig.5.16. Transformada de Fourier de la Fig.5.15.

Para el espectro del audio, del filtro pasa bajo de la Fig.5.16, es posible notar que, en comparación del espectro del audio original, el filtrado deja pasar gran parte de las frecuencias bajas, las cuales son las frecuencias bajo la frecuencia de corte de 1000 Hz. Es por ello que se logra mantener gran parte del espectro del audio, para frecuencias bajas, donde se hizo el comparativo con el espectro del audio original de la Fig.5.3. También es cabo mencionar, que las frecuencias sobre la frecuencia de corte son mínimas, siendo casi despreciables.

5.2.14 Implementación del efecto 12 Filtro distorsión

Para el doceavo efecto, correspondiente al filtro distorsión, se procedió a aplicar la combinación de dos estrategias de efectos: (I) La estrategia de filtrado FIR, con la cual se realizó un filtro pasa banda, con frecuencia de corte entre los 1900 Hz y los 9000 Hz. (II) La estrategia del efecto de distorsión, modificando en pequeña escala los vectores y sumándole una ganancia de 1%, el valor del vector ocupado al azar. Para una mayor profundización del tema, es recomendable revisar el capítulo 3.3.13, en donde se tocaron los tópicos del presente efecto 12. En la siguiente Fig.5.17, se mostrarán los resultados obtenidos, por simulación versus la implementación, del presente efecto.

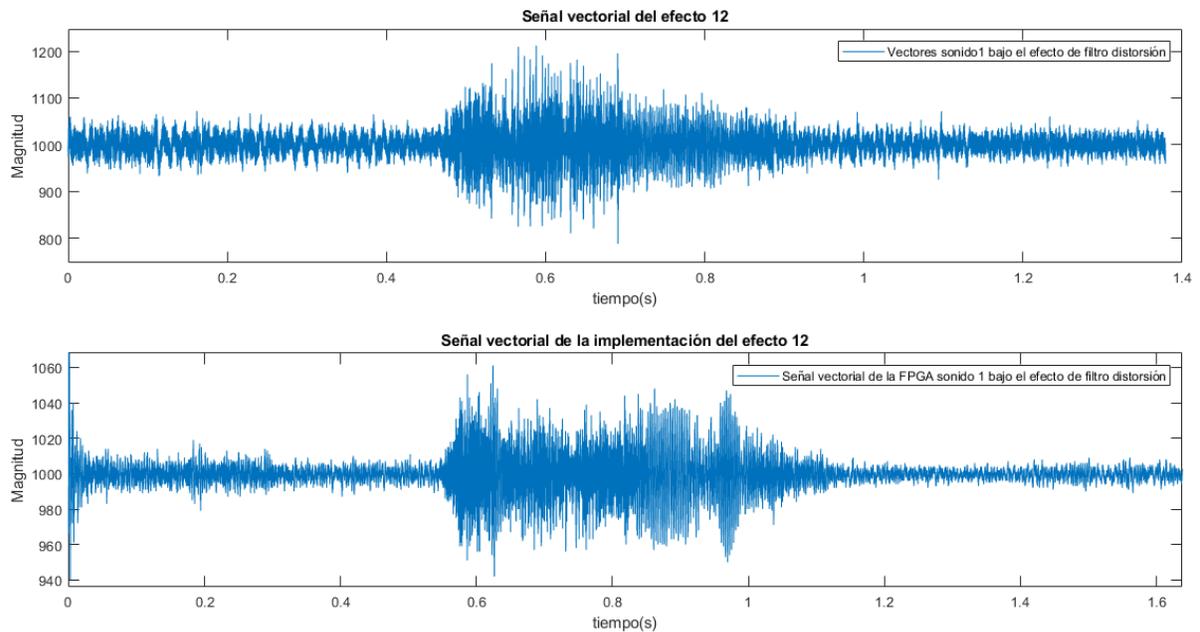


Fig.5.17. Gráfica de la simulación del efecto de Filtro distorsión del audio 1 vs su implementación en la FPGA.

En la Fig.5.17, posee dos características importantes tal como se había nombrado en el párrafo anterior: (I) La frecuencia de corte elegida, al ser una frecuencia mayor, la señal toma una forma más comprimida, por una menor longitud de onda a mayor frecuencia. (II) Aunque posea la misma silueta que sus señales originales de la Fig.5.2, se ven más difusas, por el hecho de la aplicación de distorsión, puesto que, ambas señales de la simulación e implementación cumplen con estas dos características, se ve una aplicación del efecto de filtro de distorsión. Con la finalidad de comprobar su efectividad, se procedió a calcular su transformada de Fourier de ambas gráficas, en la Fig.5.18.

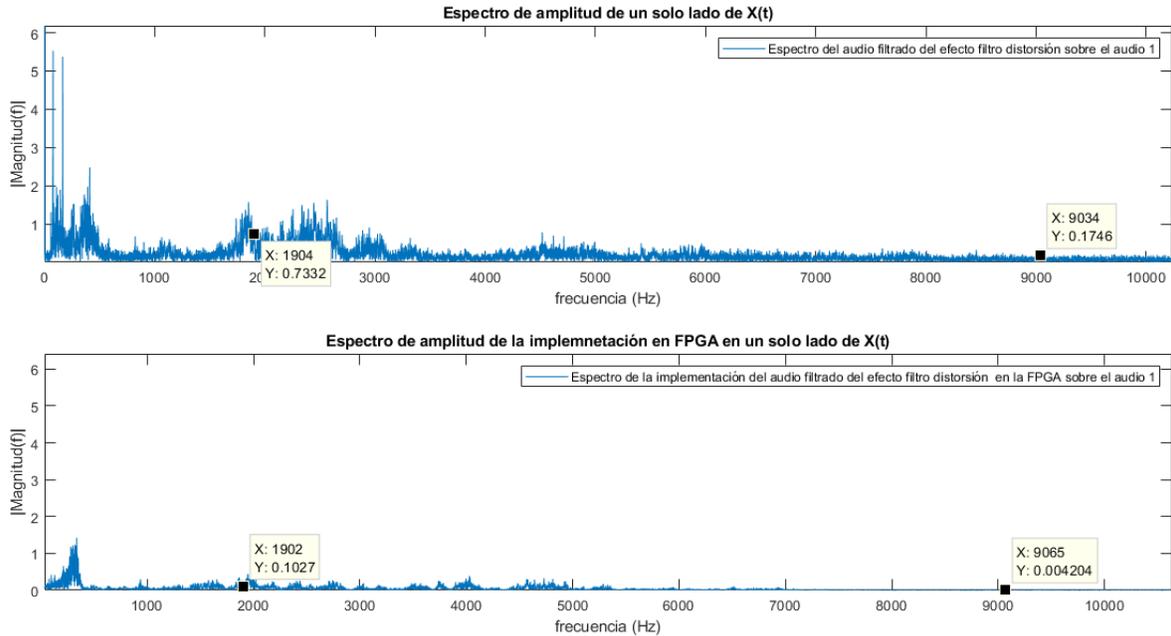


Fig.5.18. Transformada de Fourier de la Fig.5.17.

Dentro de las gráficas mostradas en la Fig.5.18, para la simulación se logra dilucidar un alza al principio de la señal, fuera de la frecuencia de corte, provocada la distorsión de la señal, mientras que, para las frecuencias dentro del rango deseado, se logran rescatar en gran parte. Para la implementación, se muestra una gran atenuación de la señal, puesto que no se logra rescatar todo el espectro deseado, por el proceso de grabación aplicado, además del filtrado aplicado. Pero se logra rescatar casi todo el espectro de frecuencia buscado, teniendo en consideración la gráfica de la implementación de la Fig.5.3. Es por ello que, para el presente caso de la implementación, es efectivo la aplicación del filtro distorsión.

6 Conclusiones

6.1 Introducción

Finalizando la presente memoria de Diseño e implementación de un sintetizador de audio en una plataforma Zynq-7000, se procederá a detallar el análisis de cada capítulo del presente informe, por medio de la sección 6.2 de sumario. Prosiguiendo la sección 6.3 de conclusión se abordará el análisis de todo el presente informe, dando importancia a lo aprendido durante la carrera y la aplicación del sintetizador de sonido, desde el punto de vista mecatrónico. Por último, se darán a conocer las posibles mejoras y complementos que puede obtener el sintetizador de sonido, a través de la sección de 6.4 de trabajos futuros.

6.2 Sumario

En la sección 6.2, se trabajará con el análisis de los 5 capítulos anteriormente planteados, donde se harán un análisis de cada capítulo, explicando sus puntos fuertes y su importancia. De este análisis se tiene los siguientes puntos:

1. El primer capítulo se abordó todos los conocimientos previos a la creación de un sintetizador de sonidos, empezando con un análisis de los diferentes tipos de sintetizadores de sonidos en el mercado, donde se concluyó que la mejor opción dentro de los estándares de la FPGA, fue el sintetizador de la serie PO KO. Prosiguiendo se hizo un análisis de los efectos de sonidos principales, que entrega los sintetizadores de sonido y como parte de ellos se pueden realizar, mediante la estrategia de filtrado de los filtros FII e IIR. También se hizo un análisis de la evolución de la FPGA, pasando por todas sus épocas desde la conformación hasta el periodo de hoy. Finalmente se hizo un análisis de costos de los sintetizadores de sonido estudiados, dando a comprender su precio en el mercado.
2. En el segundo capítulo se analizó la FPGA a trabajar, la cual es la tarjeta Zybo Z7-10 de la empresa Digilent. Donde se abordó las características de la FPGA, sus capacidades y sus alcances, comprobando si se puede implementar un sintetizador de sonido en dicha tarjeta mediante la plataforma Zynq-7000. Después se realizó un análisis de cada

componente interno del Hardware necesario para la conformación del sintetizar, dentro de estos se encuentran:(1) Análisis de la arquitectura de la Zynq. (2) Reloj/Oscilador. (3) Audio. (4) I/O Básicas. (5) Puertos Pmod.

3. El tercer capítulo consto del análisis de cada modo del sintetizador de sonido, desarrollando los cuatro modos principales:(1) Modo compositor: el cuál es el modo donde se puede crear patrones de sonidos, en base a ritmos de 16 tiempos. Este modo consta de 15 canales de diferentes sonidos. (2) Modo efecto de sonido: El cual posee 16 efectos aplicables a unos nuevos 16 sonidos. (3) Modificación de parámetros: El cual se lleva a cabo por el giro de sus perillas (encoder rotativo), que permite modificar el volumen del sonido, la velocidad del modo patrón(BPM) y alterar algunos efectos de sonidos (tiempo de inserción del efecto, para los efectos 4, 5 y 6). (4) Visualización de parámetros: En donde se puede analizar y ver la información entregada por los otros tres modos.

Dentro del presente capítulo también se simulo de manera ideal en el software de Matlab todos los efectos de sonidos, comparándolo con su señal original. En el caso del filtrado y barrido, se graficó su transformada de Fourier, para comprobar la efectividad del efecto de sonido.

4. En el cuarto capítulo correspondiente a la implementación, se dio a conocer todo el Hardware externo a la FPGA Zybo Z7, para la realización del sintetizador de sonido, tales como teclado, encoder rotativo y pantalla LCD. En el cual se analizó cada componente y se explicó su funcionamiento.

Continuando se analizó los principales bloques del Hardware mixto, del software de Vivado, para la realización del sintetizador de sonido. Explicando su funcionamiento y su importancia para el control del Hardware explicado en el capítulo 2 y el Hardware externo del presente capítulo 4.

Prosiguiendo se hizo un mayor análisis a la estrategia de filtrado de los filtros FIR e IIR. En el dónde se explicó cómo implementarlos en el software de SDK de Vivado, por medio de las ecuaciones de cada estrategia de filtrado, para barridos y filtros.

Finalmente se realizó un nuevo análisis de costos al realizado en el capítulo 1, esta vez enfocado a los elementos ocupados durante la memoria, para la creación del sintetizador de sonido versus los sintetizadores de sonidos que se consideraron como punto de partida (serial Pocket Operator).

5. En el quinto capítulo se explica los resultados experimentales de la implementación de cada efecto de sonido. En este se implementó los primeros 12 efectos de sonidos y se comparó con sus versiones de la simulación, del capítulo 3.

6.3 Conclusión

Dentro del desarrollo de la memoria de título, tanto para el proceso de simulación e implementación, del sintetizador de sonido es posible concluir:

La FPGA permite la libertad del usuario tanto de software como de Hardware, crear un sintetizador de sonido adaptado a sus propias necesidades. Esto es debido a que la FPGA, posee una gran variedad de bloques de hardware mixto, que le permite integrar diferentes periféricos según la necesidad del usuario. Estas características, hace que la creación del sintetizador de sonido sea más versátil, en comparación del modelo estándar del sintetizador a comprar, por el hecho que se puede ir agregando elementos en función de la necesidad. Desde el punto de vista del software el usuario puede programar los efectos de sonidos que estime conveniente, además de elegir los sonidos a reproducir y los modos que encuentre más factibles. Lo anterior da una mayor ventaja a la programación en la FPGA frente a los sintetizadores en el mercado, ya que no se debe estar restringido por los sonidos y efectos que entregue el sintetizador de sonido, que solo se puede cambiar comprando un nuevo modelo como en el caso de la serie PO KO o contratando un plan para cargar nuevos sonidos, como el caso de los sintetizadores de sonido profesionales.

Dentro de la programación de la FPGA se debe tomar preocupación al momento de crear las funciones y modos del sintetizador. Esto es debido que, no se le puede exigir que ejecute una gran cantidad de acciones al mismo tiempo, como se ha demostrado en la implementación donde la reproducción no dura el rango de tiempo estimado, en vez de ello toma un 10% más en función del efecto, como se ha demostrado con los resultados

experimentales del capítulo 5. Para corregir el retraso en el tiempo de reproducción es necesario disminuir el número de variables y hacer condiciones más simples, para evitar el sobrecupo de acciones máximas que pueda realizar la FGPA.

Desde punto de vista del análisis económico de la implementación de un sintetizador de sonido en una plataforma Zynq-7000. Este posee un precio igual a los modelos más antiguos del serial PO KO, dícese del PO-20 KO. En consideración de los sintetizadores profesionales de alta gama, estos poseen un precio promedio de 600.000 CLP, que en comparación el sintetizador de sonido realizado durante la memoria es un 87.5% más económico, con 75.000 CLP si solo se utiliza el chip de la Zynq 7000, pero ocupando la tarjeta completa de la Zybo Z7-10 se obtiene un valor de 178.272 CLP lo que da un ahorro de un 70.288%. Esta valoración da a entender que, ya invirtiendo en un sintetizador aplicado a una FPGA, es una opción mucho más económica, la cual cumple con requisitos básicos del usuario y como se ha explicado en las anteriores conclusiones, es una opción mucho más adaptable. Sin embargo, el costo elegir esta opción, es que el usuario necesariamente debe poseer conocimientos sobre el lenguaje de programación a ocupar, además de contar con un manejo en los softwares necesarios para la implementación del sintetizador de sonidos. Estas condiciones pueden desincentivar a posibles usuarios del sistema.

6.4 Trabajos futuros

Dentro de los trabajos futuros se encuentra la implementación de efectos de sonidos, Hardware externo a la Zybo Z7, posibles mejoras que permiten un mayor uso del sintetizador de sonido. Dentro de esta variedad de se encuentra los siguientes trabajos futuros:

- Implementar los últimos cuatro efectos de sonidos (efectos 13-16), presentados en el capítulo 3 de simulación.
- Implementar y programar los encoder rotativos, para la modificación de parámetros, explicado en el capítulo 3 de simulación.
- Implementar y programar la pantalla LCD, para la visualización de parámetros de los diferentes modos y parámetros del sintetizador de sonido.

- Crear paquetes nuevos de sonidos para el modo compositor y para los sonidos de efectos de sonidos, junto a 16 efectos de sonidos nuevos. Con la finalidad de ir cambiando los sonidos y efectos disponible del sintetizador de sonido.
- Externalizar las luces leds, para no ocupar los cuatro leds de la Zybo Z7, en vez de ello ocupar 16 leds, para una mayor precisión al momento de activar alguna posición del teclado táctil. Esto servirá para saber de mejor manera la posición del tiempo correspondiente en el modo compositor. Además, puede ser posible su conformación mediante Multiplexor/Demultiplexor CD74HC4067, utilizando solo cuatro salidas Pmods, para controlar los 16 leds, del sondeo y activación del modo compositor, y ser de apoyo visual para los demás modos.
- Externalizar los cuatro botones de la Zybo Z7. Correspondiente al cambio de canal y selección del efecto de sonido. Permitiendo a los otros dos botones la capacidad de guardar la composición del modo patrón y permitir escuchar solo los efectos de sonidos o los efectos de sonidos con el modo patrón combinados.
- Crear y diseñar una placa PCB donde estarán todas las conexiones del Hardware del sintetizador de sonido.
- Diseñar y crear una carcasa que contenga la placa PCB, el Hardware externo, la FPGA Zybo Z7 y la fuente de alimentación del sintetizador de sonido. Con la finalidad de hacerlo portátil.
- Programar la memoria de la Zybo Z7, para que guarde los datos sin que sea necesario cargar todos los programas en el computador cuando se desee ocupar.

7 Referencias.

- [1] M. del M. Bernabé Villodre, “Importancia de la música como medio de comunicación intercultural en el proceso educativo,” *Teoría la Educ. Rev. Interuniv.*, vol. 24, no. 2, pp. 107–127, 2013, doi: 10.14201/10357.
- [2] L. Magazine, “Instrumentos Electrónicos: Historia y ejemplos - LaCarne Magazine.” [Online]. Available: <https://lacarnemagazine.com/historia-los-instrumentos-electronicos-parte-i/>.
- [3] Wikipedia, “Mesa de mezclas de audio.” .
- [4] J. Merrich, “Historia de los sintetizadores | t.blog.” [Online]. Available: <https://www.thomann.de/blog/es/historia-de-los-sintetizadores/>.
- [5] G. Vidal, “Las cinco mejores marcas de sintetizadores,” 2020. [Online]. Available: <https://soundsmarket.com/blog/las-cinco-mejores-marcas-de-sintetizadores/>.
- [6] C. Moler, “MATLAB.” 1984.
- [7] Xilinx, “SDK.” 1984.
- [8] Musisol, “comparativa juno ds 61 vs yamaha mx61 vs korg kross 2-61.” [Online]. Available: <https://www.musisol.com/blog/guia-de-compra/comparativa-juno-ds-61-vs-yamaha-mx61-vs-korg-kross-2-61/>.
- [9] Roland, “juno ds61.” [Online]. Available: <https://www.roland.com/global/products/juno-ds/juno-ds61/e>.
- [10] Yamaha, “Manual Yamaha mx49-mx61-mx88.” [Online]. Available: https://es.yamaha.com/files/download/other_assets/7/892947/mx49mx61mx88_es_o_m_f0.pdf.
- [11] Korg, “kross2 especificaciones.” [Online]. Available: <https://www.korg.com/es/products/synthesizers/kross2/especificaciones.php>.
- [12] Amazon, “Imagen Korg Kross2 61.” [Online]. Available: https://m.media-amazon.com/images/I/81QrNSbskdL._AC_SL1500_.jpg.
- [13] Teenage.engineering, “PO 32 KO.” [Online]. Available: <https://teenage.engineering/guides/po-32/en>.
- [14] Teenage.engineering, “PO 20 KO.” [Online]. Available: <https://teenage.engineering/guides/po-20/en>.
- [15] Teenage.engineering, “PO 28 KO.” [Online]. Available: <https://teenage.engineering/guides/po-28/en>.
- [16] Akerlof, 濟無 *No Title No Title*, vol. 53, no. 9. 1970.
- [17] E. D. E. Ingeniería, C. Mecatr, I. C. Mecatr, P. Gu, and C. B. Fuentes, “Uso de hardware mixto en la implementación de un convertidor boost,” 2018.
- [18] S. M. Trimberger, “Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology,” *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, 2015, doi: 10.1109/JPROC.2015.2392104.
- [19] D. H. D. Montoya and S. D. A. Monje, “Estudio del mundo de las FPGAs, comparativas e implementación,” p. 109, 2018.
- [20] Paris, “Cotización PO-20 KO.” [Online]. Available: <https://www.paris.cl/sintetizador-teenage-engineering-po-20-arcade-926503999.html>.
- [21] Amazon, “Cotización PO-20 KO.” [Online]. Available:

- <https://www.amazon.com/Teenage-Engineering-TE010AS020A-PO-20-Arcade/dp/B01BKO0HKW>).
- [22] Amazon, “Cotización PO-28 KO.” [Online]. Available: <https://www.amazon.com/Teenage-Engineering-TE010AS028A-Pocket-Operator/dp/B01BKO3GQO>.
- [23] Ripley, “Cotización PO-28 KO.” [Online]. Available: <https://simple.ripley.cl/te-pocket-operator-po-28-robot-mpm00000753057>
- [24] Cutoff, “Cotización PO-32 KO.” [Online]. Available: <https://www.cutoff.es/teenage-sintetizadores/7557-teenage-engineering-po-32-tonic.html>
- [25] Falabella, “Cotización PO-32 KO.” [Online]. Available: <https://www.falabella.com/falabella-cl/product/6780305/Te-Pocket-Operator-Po-32->
- [26] Teenage engineering, “Cotización OP-Z.” [Online]. Available: <https://teenage.engineering/store/op-z>.
- [27] Musisol, “Cotización JUNO-DS61.” [Online]. Available: <https://www.musisol.com/sintetizadores-y-workstations/roland-juno-ds-61.html>.
- [28] Musisol, “Cotización Yamaha MX61 v2.” [Online]. Available: <https://www.musisol.com/sintetizadores-y-workstations/yamaha-mx61-v2-sintetizador.html>.
- [29] Musisol, “Cotización Korg Kross2 61.” [Online]. Available: <https://www.musisol.com/sintetizadores-y-workstations/roland-juno-ds-61.html>.
- [30] Digi-Key, “Cotización FPGA Zynq7000.” [Online]. Available: <https://www.digikey.com/product-detail/es/xilinx-inc/XC7Z010-1CLG400C/122-1854-ND/3925789>.
- [31] Digilent, “Cotización Zybo Z7.” [Online]. Available: <https://digilent.com/shop/zybo-z7-zynq-7000-arm-fpga-soc-development-board/>.
- [32] Digilent, “Zybo Z7 Board Reference Manual,” *Digilent, Inc.*, pp. 1–30, 2017.
- [33] Digilent, “Arty A7 Reference Manual.” [Online]. Available: <https://reference.digilentinc.com/programmable-logic/arty-a7/reference-manual>.
- [34] I.-L. Software, “FL Studio.” 1997.
- [35] alejo g 6, “los sonidos de pikachu,” 2017. [Online]. Available: https://www.youtube.com/watch?v=mvb-3Qg_JQU.
- [36] Shamisen in Tokyo, *Shamisen Girls Kiki - Gentenkaiki*. 2017.
- [37] M. Koenig, *Punch Sounds | Free Sound Effects | Sound Clips | Sound Bites*. Soundbible.com.
- [38] C. F. Media, “Hatsune Miku X Kodo 2018 Live (NHK BS Broadcast, Show only),” 2018. [Online]. Available: <https://www.youtube.com/watch?v=GLrJMxdYnsA>.
- [39] Ado, “ラブカ? feat Ado,” 2020. [Online]. Available: <https://www.youtube.com/watch?v=YTQV48V44Sw>.
- [40] 1st PLACE, “じん / 夜咄ディセイブ 【OFFICIAL MUSIC VIDEO】,” 2013. [Online]. Available: <https://www.youtube.com/watch?v=aPDaFGqMX4I>.
- [41] EHowEspañol, “Variedad de sonidos en los platillos con mazos : Batería: técnica con diferentes palillos,” 2012. [Online]. Available: <https://www.youtube.com/watch?v=LOsLVY1FmF0>.
- [42] Mathworks, “audioread.” [Online]. Available:

- <https://la.mathworks.com/help/matlab/ref/audioread.html>.
- [43] Satô Benkei, “alfabeto de los jojo’s,” 2019. [Online]. Available: <https://www.youtube.com/watch?v=Zsw4LYIHG-o>.
- [44] S. Hikari, “KonoSuba Megumin Explosion!,” 2016. [Online]. Available: <https://www.youtube.com/watch?v=O4tbOvKwZUw>.
- [45] Kikuo, “Kikuo- Aishite Aishite (Ámame, Ámame),” 2015. [Online]. Available: https://www.youtube.com/watch?v=NTrm_idbhUk.
- [46] JRG, “ORA ORA ORA ORA ORA 60FPS,” 2018. [Online]. Available: https://www.youtube.com/watch?v=0fBy_DTGNW8.
- [47] カズヒラ, “All Hachikuji Stutters/Flubs 「物語シリーズ」,” 2019. [Online]. Available: <https://www.youtube.com/watch?v=5zjtfKBX300>.
- [48] Mathworks, “feedback.” [Online]. Available: <https://la.mathworks.com/help/control/ref/lti.feedback.html>.
- [49] Wikiwand, “Negative feedback.” [Online]. Available: https://www.wikiwand.com/en/Negative_feedback.
- [50] Mathworks, “Análisis de polo cero.” [Online]. Available: <https://la.mathworks.com/help/signal/ug/zero-pole-analysis.html>.
- [51] Mathworks, “Diseño de filtro FIR.” [Online]. Available: <https://la.mathworks.com/help/signal/ug/fir-filter-design.html>.
- [52] TonTouch TM, “Ttp229-Lsf 16 Keys or 8 Keys Touch Pad Detector Ic Ttp229-Lsf,” pp. 1–18.
- [53] Rcscomponents.kiev.ua, “ky-040-datasheet.pdf.” .
- [54] adafruit, “Lcm module tc2004a-01,” *Datasheet*, pp. 1–18, 2009.
- [55] Techlookelectronica.com, “Pantalla LCD 2004 con interface I2C – TechLook Electrónica.”
- [56] Xilinx, “Processing System 7 v5.5 LogiCORE IP Product Guide,” *Pg082*, p. 180, 2017.
- [57] Xilinx, “Processor System Reset Module v5.0,” p. 33, 2015.
- [58] I. Xilinx, “AXI Reference Guide, v4.0,” *Xilinx Tech. Doc.*, vol. 1037, no. ug1037, pp. 1–175, 2017.
- [59] Xilinx, “LogiCORE IP Product Guide : AXI GPIO v2.0,” *Pg144*, p. 2004, 2003.
- [60] MATLAB, “Diseño de filtro FIR basado en ventanas - MATLAB fir1 - MathWorks América Latina.” [Online]. Available: <https://la.mathworks.com/help/signal/ref/fir1.html#bulla52-ftype>.
- [61] MATLAB, “Diseño del filtro Butterworth - MATLAB butter - MathWorks América Latina.” [Online]. Available: https://la.mathworks.com/help/signal/ref/butter.html?searchHighlight=butter&s_tid=srchtitle.
- [62] mercadolibre, “Mini Parlantes Usb Para Pc Cable 3,5mm Xtech Xts-110 Negro - \$ 4.990.” .
- [63] MechatronicStore, “Matriz Táctil 4x4 Teclera Capacitiva | Compra Con Envío Gratis.” [Online]. Available: <https://www.mechatronicstore.cl/matriz-de-16-teclas-4x4-tactiles-capacitivas-ttp229/>.
- [64] MechatronicStore, “Encoder Rotatorio Brick KY-040 Rotary 360.”
- [65] MechatronicStore, “Pantalla LCD2004 Con I2C Incorporado,” 2021. [Online].

- Available: <https://www.mechatronicstore.cl/pantalla-lcd2004-con-i2c/>.
- [66] Y. González, “18 tipos de formato de audio y sus características | Grupo Atico34.” [Online]. Available: <https://protecciondatos-lopd.com/empresas/tipos-formato-audio/#:~:text=M4A son las siglas de,una extensión de archivo M4A.>

8 Enlaces videos de la implementación

8.1 Modo patrón simple

<https://drive.google.com/file/d/1Bw7yFkCbkh5PzN7MWAsUveMj5xg12dXQ/view?usp=sharing>

8.2 Modo patrón compuesto

https://drive.google.com/file/d/1Gqx1ES5cqT1nht70hupMcdombgLvrGK_/view?usp=sharing

8.3 Modo patrón activo sus 16 tiempos

<https://drive.google.com/file/d/1O2UXTrGIR5rerkfMcLWgpg6RFkrarLGI/view?usp=sharing>

8.4 Efecto de sonidos

<https://drive.google.com/file/d/1UuzPrZD7gdGBVdtEBo7qn6dl7bAM81tD/view?usp=sharing>