

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE  
TECNOLOGÍAS Y SERVICIOS DE  
TELECOMUNICACIÓN**

**TRABAJO FIN DE GRADO**

***DISEÑO Y PROTOTIPADO DE PADS  
DE PERCUSIÓN CON  
RECONOCIMIENTO DE PRESIÓN***

**LUCAS DE MIGUEL ANGUITA**

**2021**



# GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACION

## TRABAJO FIN DE GRADO

**Título:** Diseño y prototipado de Pads de percusión con reconocimiento de presión  
**Autor:** Lucas de Miguel Anguita  
**Tutor:** Pablo Ituero Herrero  
**Ponente:** D. ....  
**Departamento:** Departamento de Ingeniería Electrónica

## MIEMBROS DEL TRIBUNAL

**Presidente:** D. ....

**Vocal:** D. ....

**Secretario:** D. ....

**Suplente:** D. ....

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:  
.....

Madrid, a                      de                      de 20...

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y  
SERVICIOS DE TELECOMUNICACIÓN**

**TRABAJO FIN DE GRADO**

***DISEÑO Y PROTOTIPADO DE PADS DE  
PERCUSIÓN CON RECONOCIMIENTO DE  
PRESIÓN***

**LUCAS DE MIGUEL ANGUITA**

**2021**



## RESUMEN

El presente proyecto tiene como principal objetivo obtener un prototipo de Pads de percusión, es decir, una batería musical electrónica. Este dispositivo pretende simular una batería acústica utilizando pads contruidos ad-hoc y empleando una FPGA para el procesamiento digital del audio.

Los pads tendrán incorporados sensores piezoeléctricos, estos sensores proporcionan una señal que depende de la presión que se ha ejercido sobre ellos. De esta forma obtenemos el control del volumen asociado al golpe recibido. Se diseñará e implementará la circuitería analógica necesaria para acondicionar la señal proveniente de estos sensores. El objetivo de este proceso es acercarse lo máximo posible al funcionamiento de una batería acústica real.

Se diseñará y realizará la circuitería analógica encargada de muestrear la señal de los sensores para su posterior conversión ADC para tener una señal adecuada entrante a la FPGA.

Seguidamente se estudiará y desarrollará una interfaz de comunicación con una memoria DDR que tendrá almacenados bancos de sonidos. Tras esto se llevará a cabo un controlador general del sistema con los correspondientes protocolos para controlar los accesos a la memoria y su correcta reproducción culminando con la obtención de un prototipo funcional implementado en VHDL.

Gracias a las características de este proyecto, al lenguaje de programación utilizado y al entorno (Vivado), se podrán añadir funcionalidades, por ejemplo, efectos, de manera muy sencilla dependiendo de la aplicación que se le quiera dar.

## SUMMARY

The main objective of this project is to obtain a prototype of percussion pads, i.e., an electronic musical drum kit. This device aims to simulate an acoustic drum kit using ad-hoc pads and using an FPGA for digital audio processing.

The pads will have piezoelectric sensors incorporated; these sensors provide a signal that depends on the pressure that has been exerted on them. In this way we obtain the control of the volume associated with the blow received. The analogue circuitry necessary to condition the signal coming from these sensors will be designed and implemented. The aim of this process is to get as close as possible to the operation of a real acoustic battery.

The analogue circuitry in charge of sampling the signal from the sensors will be designed and implemented for its subsequent ADC conversion to have a suitable signal coming into the FPGA.

Next, a communication interface will be studied and developed with a DDR memory that will have stored banks of sounds. After this, a general system controller will be developed with the corresponding protocols to control accesses to the memory and its correct reproduction, culminating in a functional prototype implemented in VHDL.

Thanks to the characteristics of this project, the programming language used and the environment (Vivado), it will be possible to add functionalities, for example, effects, in a very simple way depending on the application that you want to give it.

## **PALABRAS CLAVE**

Sensor piezoeléctrico, FPGA, VHDL, Nexys A7, Máquina de estados finita con ruta de datos, pad, DDR2.

## **KEYWORDS**

Piezoelectric sensor, FPGA, VHDL, Nexys A7, Finite State Machine with data path, pad, DDR2.



## AGRADECIMIENTOS

*A mi familia.*

*A mis amigos y compañeros por hacer el paso por la universidad más ameno y divertido.*

*A mi tutor, Pablo Ituero, por haberme ayudado todas las veces que lo he necesitado.*

*Al profesor Alfredo Sanz por su ayuda y consejo.*

# ÍNDICE DEL CONTENIDO

<b>1. INTRODUCCIÓN Y OBJETIVOS.....</b>	<b>1</b>
1.1. INTRODUCCIÓN .....	1
1.2. OBJETIVOS .....	2
1.3. RESULTADOS.....	3
1.4. METODOLOGÍA .....	4
1.5. DESARROLLO TEMPORAL.....	5
1.6. ESTRUCTURA DEL DOCUMENTO .....	5
<b>2. DISEÑO Y DESARROLLO DE LA INTERFAZ ANALÓGICA .....</b>	<b>6</b>
2.1. ESTUDIO BIBLIOGRAFICO.....	6
2.2. PROTOTIPADO DE LOS PADS.....	7
2.2.1. ELECCIÓN DE MATERIALES .....	7
2.2.2. ELECCIÓN DE LOS SENSORES .....	9
2.3. ACONDICIONAMIENTO DE LA SEÑAL .....	11
<b>3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLADORES .....</b>	<b>15</b>
3.1. CONVERSIÓN ADC .....	15
3.2. CONTROLADOR DE AUDIO DEL SISTEMA .....	19
3.2.1. INTERFAZ DE COMUNICACIÓN CON LA DDR2 .....	20
3.2.2. CONTROLADOR DE REPRODUCCIÓN .....	24
3.2.3. CONTROLADOR DE ACCESOS A MEMORIA .....	25
3.2.4. REGISTRO DE DATOS .....	27
3.2.5. GENERADOR DE ENABLES.....	28
3.2.6. GENERACIÓN DE LA SEÑAL DE SALIDA .....	29
<b>4. PRUEBAS DE LOS ALGORITMOS DESARROLLADOS .....</b>	<b>32</b>
4.1. PRUEBAS DE LA CONVERSIÓN ADC.....	32
4.2. PRUEBAS DEL DETECTOR DE PICOS .....	33
4.3. PRUEBAS DE LA INTERFAZ CON LA MEMORIA.....	33
4.4. PRUEBAS DEL CONTROLADOR DE REPRODUCCIÓN .....	34
4.5. PRUEBAS DEL CONTROLADOR DE ACCESOS A MEMORIA .....	35
4.6. PRUEBAS DEL REGISTRO DE DATOS.....	35
4.7. PRUEBAS DEL GENERADOR DE ENABLES .....	36
<b>5. RESULTADOS .....</b>	<b>37</b>
<b>6. CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>38</b>
6.1. CONCLUSIONES .....	39

6.2. LÍNEAS FUTURAS .....	40
<b>7. BIBLIOGRAFÍA .....</b>	<b>41</b>
<b>ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES.....</b>	<b>43</b>
A.1 INTRODUCCIÓN .....	43
A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO .....	43
A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS .....	43
A.4 CONCLUSIONES .....	44
<b>ANEXO B: PRESUPUESTO ECONÓMICO.....</b>	<b>45</b>

# INDICE DE FIGURAS

Figura 1.1: <i>Pads de Percusión comerciales</i> [2] .....	1
Figura 1.2: <i>Puertos PMOD; vista frontal, tal y como están en la PCB</i> [3] .....	1
Figura 1.3: <i>Esquema prototipo final</i> .....	2
Figura 1.4: <i>Montaje completo del prototipo, visto desde arriba</i> .....	3
Figura 1.5: <i>Entorno de desarrollo PSpice</i> [4].....	4
Figura 1.6: <i>Entorno de desarrollo Vivado</i> [5].....	4
Figura 1.7: <i>Entorno de desarrollo Matlab</i> [6] .....	4
Figura 2.1: <i>Sensores piezoeléctricos</i> [8] .....	6
Figura 2.2: <i>Ejemplo modulación PCM</i> [11].....	7
Figura 2.3: <i>Diseño de la estructura de los Pads</i> .....	7
Figura 2.4: <i>Pad de práctica de percusión</i> [12] .....	8
Figura 2.5: <i>Esquema final del pad</i> .....	8
Figura 2.6: <i>Diafragma de un sensor piezoeléctrico cerámico</i> [22] .....	9
Figura 2.7: <i>Sensores piezoeléctricos usados en el prototipo</i> .....	9
Figura 2.8: <i>Montaje del pad, vista isométrica</i> .....	10
Figura 2.9: <i>Montaje del pad, vista inferior</i> .....	10
Figura 2.10: <i>Montaje de las capas del pad</i> .....	10
Figura 2.11: <i>Circuito acondicionador de señal</i> .....	11
Figura 2.12: <i>Simulación del circuito acondicionador de señal</i> .....	12
Figura 2.13: <i>Medidas circuito de acondicionamiento</i> .....	12
Figura 2.14: <i>Divisor de tensión</i> .....	13
Figura 2.15: <i>Comparación señal del sensor original y acondicionada</i> .....	13
Figura 2.16: <i>Montaje del circuito de acondicionamiento</i> .....	14
Figura 3.1: <i>Diagrama ASMD para la lectura de registros XADC</i> .....	16
Figura 3.2: <i>Diagrama ASMD para la detección del valor de pico</i> .....	17
Figura 3.3: <i>Gráfica de pulsos</i> .....	18
Figura 3.4: <i>Esquema del sistema ADC</i> .....	18
Figura 3.5: <i>Esquema simplificado del controlador de audio</i> .....	19
Figura 3.6: <i>Protocolo de lectura DDR2</i> .....	20
Figura 3.7: <i>Protocolo de escritura DDR2</i> .....	20
Figura 3.8: <i>Diagrama ASMD del controlador del MIG</i> .....	21
Figura 3.9: <i>Simulación de lectura y escritura en memoria</i> .....	22
Figura 3.10: <i>Diagrama ASMD del controlador de lectura de datos de la memoria</i> .....	22
Figura 3.11: <i>Diagrama ASMD del controlador de reproducción</i> .....	24
Figura 3.12: <i>Diagrama ASMD del controlador de accesos a memoria</i> .....	25
Figura 3.13: <i>Circuito del registro de datos</i> .....	27
Figura 3.14: <i>Circuito generador de enables</i> .....	28
Figura 3.15: <i>Onda representada como PWM</i> [3].....	29
Figura 3.16: <i>Filtro Sallen-Key Butterworth paso bajo de 4º orden</i> [3] .....	29
Figura 3.17: <i>Respuesta en frecuencia del filtro de paso bajo Sallen-Key Butterworth</i> [3].....	30
Figura 3.18: <i>Módulo PWM</i> .....	30
Figura 3.19: <i>Esquema del controlador de audio</i> .....	31
Figura 3.20: <i>Esquema parte digital del sistema</i> .....	31

Figura 4.1: <i>Test-bench para comprobar el funcionamiento del conversor ADC</i> .....	32
Figura 4.2: <i>Test-bench para comprobar el funcionamiento del detector de picos</i> .....	33
Figura 4.3: <i>Test-bench para comprobar el funcionamiento del Controlador de reproducción</i> .....	34
Figura 4.4: <i>Test-bench para comprobar el funcionamiento del Controlador de accesos a memoria</i> ..	35
Figura 4.5: <i>Test-bench para comprobar el funcionamiento del Registro de datos</i> .....	35
Figura 4.6: <i>Test-bench para comprobar el funcionamiento del Generador de enables</i> .....	36

# 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1. INTRODUCCIÓN

Una batería electrónica es un instrumento electrónico moderno, diseñado principalmente para servir como alternativa a los instrumentos acústicos de percusión [1]. Una batería electrónica consiste en un módulo de sonido electrónico o digital que produce sonidos sintetizados o muestreados y uno o más pads equipados con sensores para activar los sonidos.



Figura 1.1: Pads de Percusión comerciales [2]

Al igual que los tambores normales, los pads se golpean con baquetas o con las manos y se tocan de manera similar a una batería acústica.

El funcionamiento de estos dispositivos reside en los sensores piezoeléctricos que llevan equipados los pads. El golpe de la baqueta provoca una diferencia de potencial que es captada por estos sensores. Generalmente, en las baterías electrónicas del mercado esta señal es llevada a un sintetizador que produce el sonido de batería deseado, dependiendo del pad golpeado, de la fuerza aplicada sobre éste y otros datos.

En el presente proyecto el encargado de sintetizar los sonidos es la placa Nexys A7 de Digilent [3] que incorpora una FPGA Artix-A7. En concreto se emplea el modelo XC7A100T-1CSG324C, siendo esta referencia la que se utiliza en el software de Xilinx (*la sección previa al guion hace referencia al tipo de arquitectura que se monta sobre la plataforma y la sección posterior al guion hace referencia a la plataforma que sustenta todo el hardware necesario para poder emplear dicha arquitectura*).

La selección de esta placa se basa en dos razones:

- Está disponible dentro del Departamento de Ingeniería Electrónica.
- Encaja en las necesidades del proyecto.

A cada pad se le puede asignar cualquier sonido, de tal forma que el percusionista que use este tipo de instrumento tiene total libertad para personalizar el kit de sonidos. La mayoría de los módulos modernos tienen asignado a los pads dos o más platillos, bombos, de dos a tres toms y un hi-hat. Pero cabe la posibilidad de asignar cualquier tipo de sonido, no solo de batería. Dando una gran ventaja respecto a las baterías acústicas, por lo que muchos músicos de la industria recurren a esta posibilidad.

Otro motivo por el cual se selecciona esta placa es por la conversión ADC de las señales de los sensores piezoeléctricos. Esta familia de FPGAs cuenta con conversores ADC integrados, conocidos como XADC. La placa tiene un módulo periférico (pmod) que permite conectar señales analógicas externas con los pines de la FPGA reservados para el XADC, permitiendo así la comunicación entre los pads y la FPGA.



Figura 1.2: Puertos PMOD; vista frontal, tal y como están en la PCB [3]

## 1.2.OBJETIVOS

El objetivo esencial de este Trabajo de Fin de Grado consiste en desarrollar un prototipo de pad de percusión. Pudiendo consolidar los conocimientos impartidos en el Grado de Ingeniería en Tecnologías y Servicios de la Telecomunicación en los temas relacionados con el tratamiento y procesado de la señal de audio, lenguajes de descripción de hardware para procesado en tiempo real e integración de distintos submódulos que engloban un prototipo completamente funcional.

Esta son las tareas desarrolladas para llevar a cabo este proyecto:

- Estudio bibliográfico sobre las técnicas a utilizar, ejemplos de prototipos y circuitería analógica necesaria.
- Diseño, evaluación y construcción de los Pads. Pasando por la elección de materiales aptos, formas y selección de los sensores necesarios.
- Diseño e implementación de la circuitería analógica encargada de acondicionar la señal del sensor piezoeléctrico.
- Diseño y realización de la circuitería analógica encargada de muestrear la señal de los sensores.
- Conversión ADC de la señal del sensor piezoeléctrico entrante a la FPGA.
- Estudio y desarrollo de una interfaz de comunicación con una memoria DDR que tendrá almacenados bancos de sonidos.
- Desarrollo del controlador general del sistema.
- Estudio, elección y desarrollo de la etapa de salida.
- Desarrollo y verificación en VHDL de la parte digital del sistema empleando la herramienta VIVADO de Xilinx.
- Finalizando con el montaje de un prototipo empleando la placa Nexys A7.

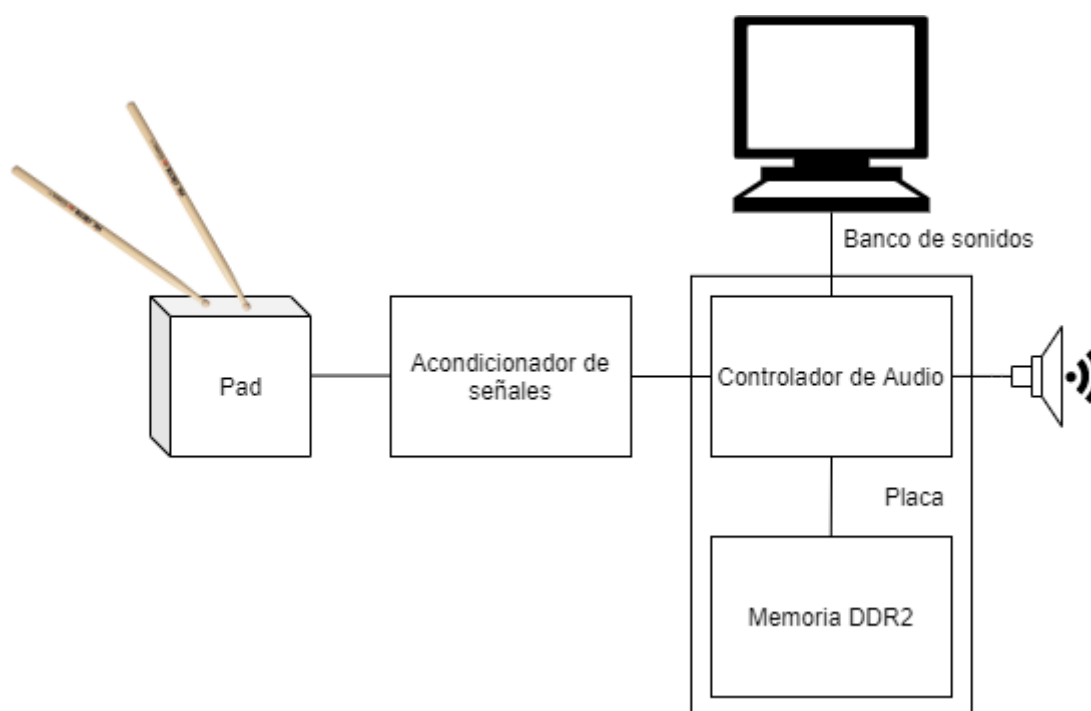


Figura 1.3: Esquema prototipo final

### 1.3.RESULTADOS

El resultado de este Trabajo de Fin de Grado consiste en la obtención de un prototipo funcional de Pads de percusión.

Como se ha explicado anteriormente el proyecto tiene dos partes diferenciadas:

- **Parte analógica.** Encargada de generar, acondicionar y muestrear las señales procedentes de los pads.
- **Parte digital.** Encargada del procesamiento digital del audio empleando una FPGA.

La duración del proyecto ha sido de aproximadamente 9 meses. La idea surge durante el mes de mayo de 2020, pero no se comenzó a llevar a cabo hasta finales septiembre de ese mismo año. Durante este tiempo se ha trabajado en el desarrollo de este proyecto, los primeros meses se dedicaron exclusivamente a la parte analógica del prototipo: estudio bibliográfico, búsqueda de materiales, diseño de los pads, etc. A finales del mes de enero comienza a desarrollarse paralelamente la parte digital del sistema.

De modo que a mediados de junio de 2021 se ha alcanzado la primera versión del prototipo. Por tanto, podemos afirmar que, pese a haber recortado expectativas y funcionalidades dejándolas pendientes para trabajo futuro, se han alcanzado las diferentes metas acordadas al inicio del proyecto: disponer de un prototipo de pad de percusión.

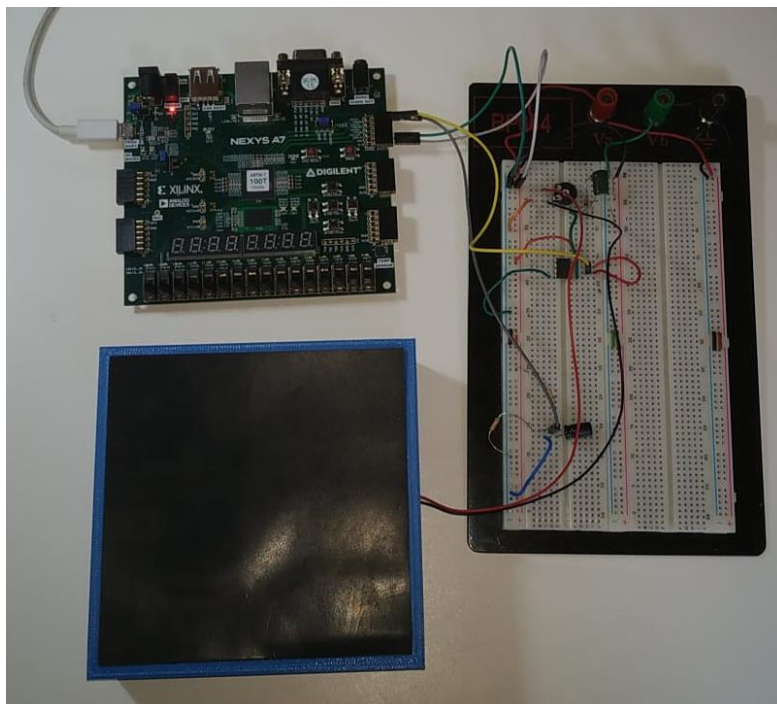


Figura 1.4: Montaje completo del prototipo, visto desde arriba

Todo el código desarrollado se encuentra disponible en el siguiente repositorio público:

<https://github.com/lucasdemiguel/Drumpad.git>



## 1.4.METODOLOGÍA

Por un lado, la metodología utilizada para desarrollar la interfaz analógica del sistema utiliza el entorno de simulación de PSpice.



Figura 1.5: Entorno de desarrollo PSpice [4]

Este entorno nos permite realizar un estudio previo del diseño de la circuitería de acondicionamiento y muestreo de la señal. De esta forma podemos saber si el circuito tiene el comportamiento esperado antes de realizar un montaje. La información que extraemos de este análisis también ayuda al estudio del comportamiento una vez que ya está montado el circuito, pues de esta forma es más sencillo identificar si las muestras tomadas se alejan mucho del comportamiento ideal.

Para el desarrollo y verificación de la parte digital del sistema se emplea la placa de desarrollo para estudiantes Nexys A7, que cuenta con una FPGA Artix-7. Usando el entorno de desarrollo Vivado 2020.2.



Figura 1.6: Entorno de desarrollo Vivado [5]

Vivado es un entorno que permite el desarrollo y verificación de sistemas electrónicos mediante un lenguaje de descripción hardware, en este caso: VHDL. Todos los módulos que componen el sistema final se implementan de este modo. Una vez finalizado el sistema completo es posible introducirlo en la FPGA y comprobar su funcionamiento en un sistema real, más allá de las simulaciones llevadas a cabo en Vivado. Esto es gracias a que Vivado es un entorno muy completo que permite realizar todo el ciclo de diseño de un sistema, pasando la creación del fichero (.vhd), por la síntesis y la implementación, terminando con la generación de un fichero (.bit) que volcar en la FPGA.

Finalmente, para la generación de los ficheros del banco de sonidos que estarán guardados en la memoria DDR se utiliza el entorno de MATLAB R2019b.



Figura 1.7: Entorno de desarrollo Matlab [6]

De esta forma es posible llevar a cabo el montaje del prototipo final.

## 1.5.DESARROLLO TEMPORAL

Una vez vistos los pasos y metodología desarrollados en los puntos anteriores es interesante hacer un desglose del tiempo dedicado a cada paso.

- **Estudio Bibliográfico:** estudio de otros prototipos, materiales necesarios, sensores, formas y circuitería analógica. **Entre 20 y 30 horas.**
- **Diseño, evaluación y construcción de los Pads:** incluyendo el desarrollo de la circuitería de acondicionamiento de la señal, uno de los puntos más laboriosos el proyecto. **Aproximadamente 140 horas.**
- **Conversión ADC de la señal del sensor piezoeléctrico:** etapa encargada de llevar la señal de los sensores hasta la entrada de la FPGA. **Entre 40 y 50 horas.**
- **Interfaz DDR2:** estudio y desarrollo de un controlador encargado de la comunicación con la memoria. **Entre 70 y 80 horas.**
- **Controlador general del sistema:** diseño, desarrollo e implementación. **Entre 30 y 40 horas.**
- **Verificación y montaje del prototipo:** el montaje y las pruebas de funcionamiento abarcaron en torno a **20 horas.**
- **Escritura del Trabajo de Fin de Grado:** en torno a **60 horas.**

Por tanto, se estima que el tiempo dedicado al proyecto ha sido de aproximadamente **420 horas.**

## 1.6.ESTRUCTURA DEL DOCUMENTO

El trabajo desarrollado está organizado en diferentes capítulos:

- El primero de ellos dedicado al bloque analógico del proyecto. Se comentan los objetivos principales, junto con los resultados obtenidos.
- En el segundo capítulo, se realiza una descripción muy detallada de los diferentes bloques digitales que conforman la arquitectura del sistema completo, los cuales son de gran relevancia para el correcto funcionamiento del prototipo.
- El tercer capítulo está dedicado a la verificación de todos los bloques del anterior punto.
- Finalmente, en los últimos capítulos se comentarán las conclusiones obtenidas y las líneas de trabajo futuras.

## 2. DISEÑO Y DESARROLLO DE LA INTERFAZ ANALÓGICA

Como se ha mencionado anteriormente, en este capítulo se desarrolla todo lo referido a la interfaz del hardware externo que permite el correcto funcionamiento de los pads de percusión, incluyendo la construcción de un prototipo impreso en 3D de un pad.

### 2.1. ESTUDIO BIBLIOGRAFICO

Este estudio incluye la búsqueda de ejemplos y diseños comerciales para conseguir una base de conocimientos con el fin de poder realizar un mejor diseño de los pads.

El funcionamiento de los pads de las baterías electrónicas está basado en sensores piezoeléctricos. Estos están situados en el interior o bajo el pad en la posición que le permita detectar mejor las vibraciones [7]. Cuando el tambor es golpeado, el sensor detecta no solo que ha recibido un impacto, sino con la fuerza que ha sido este golpe en función de la vibración.

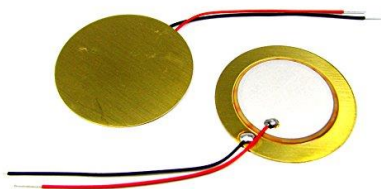


Figura 2.1: *Sensores piezoeléctricos* [8]

Actualmente las baterías musicales electrónicas son mucho más sofisticadas que la explicación del anterior párrafo. Aunque su funcionamiento básico es el mismo, suelen tener una serie de sensores colocados por todo el pad [9]. De esta forma se obtiene un mayor control de la zona y fuerza con la que se ha dado el golpe, permitiendo reproducir diferentes sonidos en función de estos datos.

Para que un pad o batería electrónica reproduzca sonido se requiere un módulo que se encargue de sintetizar los sonidos y otras posibles funciones del kit, como manejar diferentes efectos, por ejemplo. Las baterías electrónicas, incluyendo las más básicas, ofrecen una variedad de bancos de sonido. Los kits más avanzados ofrecen la posibilidad de personalizar los sonidos asignados a cada pad al gusto del músico [10]. La mayoría de los sintetizadores funciona mediante “*PCM sample sound technology*” [9]. Es decir, el módulo reproduce sonidos pregrabados cuando se golpean los pads. La modulación PCM (*Pulse Code Modulation*) [11] es un procedimiento utilizado para transformar una señal analógica en una secuencia de bits, la amplitud de la señal es muestreada a una frecuencia determinada y cuantificada al valor más cercano dentro de un rango de pasos digitales.

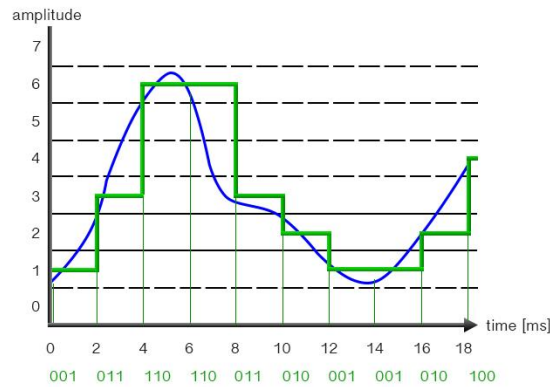


Figura 2.2: Ejemplo modulación PCM [11]

Con todos estos datos tenemos establecidas las especificaciones mínimas que tiene cumplir el dispositivo final.

## 2.2.PROTOTIPADO DE LOS PADS

En esta sección desarrollamos los pasos seguidos para el diseño y montaje de un primer prototipo de la estructura de los pads.

### 2.2.1. ELECCIÓN DE MATERIALES

Los pads requieren de un armazón resistente capaz de recibir golpes de las baquetas sin romperse. Cada pad está formado por dos piezas: la primera de ellas está formada por el esqueleto que sostiene la segunda pieza, que recibe los golpes. Esta última pieza tiene que imitar lo mejor posible el funcionamiento de los parches de un tambor.

Para el esqueleto se han barajado opciones como madera, aluminio o plástico. Generalmente los modelos de pads de percusión en el mercado son de aluminio o plástico ya que ofrecen un modelo más compacto y resistente. Debido a que el aluminio es un material más complicado de manejar para montar un prototipo la elección fue el plástico.

Por tanto, para el diseño de esta carcasa se emplea impresión 3D. Se desarrolla un diseño modular para poder tener mayor versatilidad a la hora de añadir pads, utilizando el programa informático de diseño CATIA, un programa desarrollado para proporcionar apoyo desde la concepción del diseño hasta la producción y el análisis de productos.

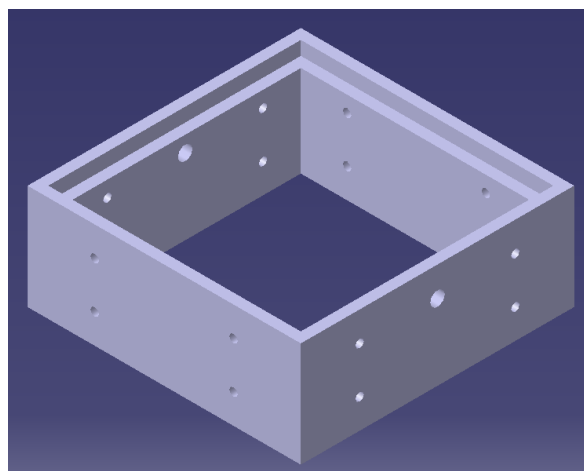


Figura 2.3: Diseño de la estructura de los Pads

Sobre estas piezas van los materiales encargados de imitar a los parches. Para diseñar esta parte se pretende reproducir la estructura de un pad de practica de percusión.



Figura 2.4: Pad de práctica de percusión [12]

Estos pads se diseñan para aproximarse a la tensión y respuesta de un parche cuando se golpea, o para proporcionar menos rebote permitiendo entrenar los músculos del percusionista [13]. Los materiales que más se utilizan son:

- Discos de Mylar sobre un sustrato de espuma. Mylar es el nombre comercial del BoPET (*biaxially-oriented polyethylene terephthalate*) [14], una película de poliéster hecha de tereftalato de polietileno.
- Discos de malla estirada sobre un marco.
- Finas capas de elastómero [15]. Un polímero natural, generalmente caucho.

Por sencillez, se opta por intentar simular esta última estructura utilizando dos finas capas: una de silicona y otra de foam.

El foam es una espuma elaborada fundamentalmente a base de polietileno [16]. Se trata de un material flexible y ligero fácil de manipular. Se fabrica en gran cantidad de formatos y se utiliza en muchos sectores diferentes, por ejemplo: la construcción o embalaje de productos. Es ideal para la protección contra impactos, ralladuras de productos y superficies delicadas, además, garantiza protección ante las posibles vibraciones y golpes que puedan producirse.

Estas dos capas reposan sobre una fina capa de madera bajo la cual se encuentra el sensor piezoeléctrico. Quedando de la siguiente forma: silicona-foam-madera, sobre la pieza en 3D.



Figura 2.5: Esquema final del pad

### 2.2.2. ELECCIÓN DE LOS SENSORES

Para el desarrollo y construcción de los pads se emplean sensores piezoeléctricos, pero hay que seleccionar el tipo que se adecúe mejor para el uso que se le quiere dar en el presente proyecto. Existen diferentes tipos clasificados por factores como el material del cristal piezoeléctrico [17]. Algunas de las limitaciones que establecen la naturaleza del cristal para la aplicación son: el ángulo en el que se corta la oblea de cristal, dimensión y espesor de la placa y el método de montaje [18].

De entre estos materiales podemos diferenciar los sensores cerámicos y de sal de Rochelle [19], que es un cristal sintético artificial. La sal de Rochelle se usa principalmente en tocadiscos, micrófonos y audífonos, con mayor sensibilidad que los cerámicos, pero estos ofrecen una gran cantidad de ventajas: son fáciles de obtener y su precio es muy económico, son química y mecánicamente más robustos y son menos sensibles ante el ruido [20]. Por lo que la decisión final se decanta por sensores cerámicos.

Básicamente la estructura de un sensor cerámico consiste en una placa de cerámica piezoeléctrica que tiene electrodos en ambos lados y una placa metálica (de latón, acero inoxidable, etc.). La placa cerámica piezoeléctrica se une a la placa metálica con adhesivos, esta estructura se denomina diafragma. Funciona de la siguiente manera: cuando el diafragma se somete a una deformación mecánica aparece una diferencia de potencial entre sus caras [21].

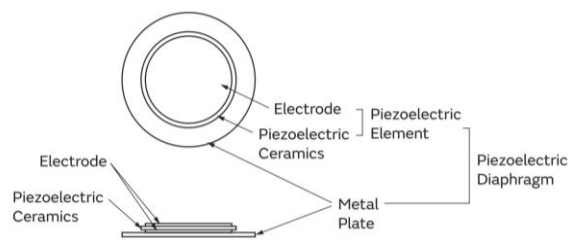


Figura 2.6: Diafragma de un sensor piezoeléctrico cerámico [22]

Los sensores comprados son del fabricante *Timesetl* [23]. Son 15 sensores de 27mm de diámetro con un precio de 8,99€. Son sensores de cerámicos con una placa metálica de latón.



Figura 2.7: Sensores piezoeléctricos usados en el prototipo

Finalmente, el montaje del pad es el siguiente:

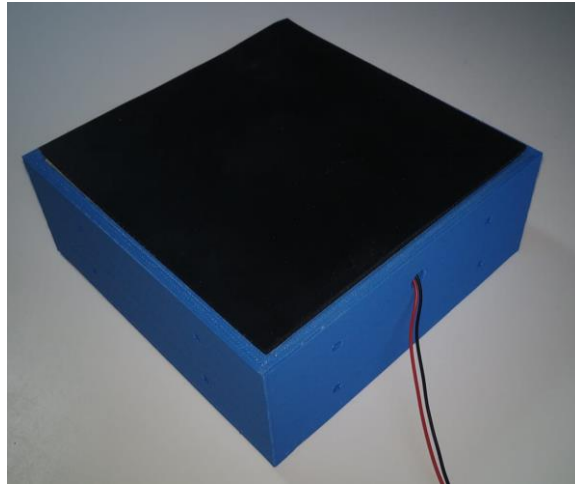


Figura 2.8: *Montaje del pad, vista isométrica*

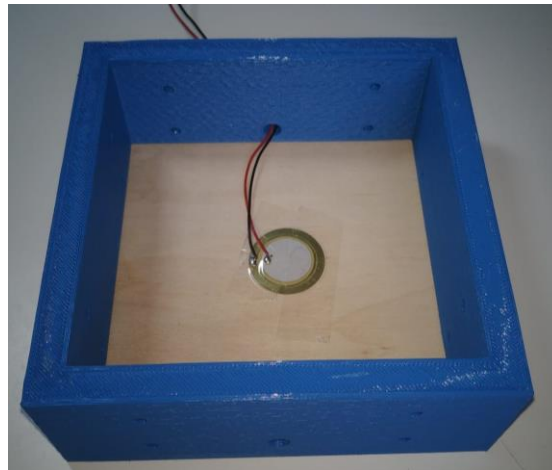


Figura 2.9: *Montaje del pad, vista inferior*

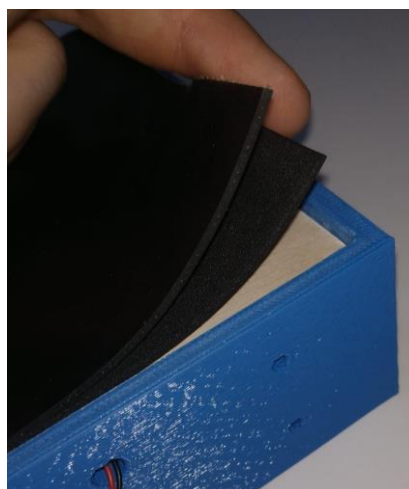


Figura 2.10: *Montaje de las capas del pad*

### 2.3. ACONDICIONAMIENTO DE LA SEÑAL

Las señales que proporciona un sensor requieren ser acondicionadas para su posterior conversión ADC. Para realizar este proceso se va a diseñar la circuitería de acondicionamiento que convierta la señal de los piezoeléctricos en un pulso limpio y fácil de leer permitiendo obtener medidas precisas, condición esencial para la exactitud de la adquisición de datos [24]. Este tipo de circuitos son capaces de efectuar otras funciones adicionales como amplificar o linealizar [25]. En nuestro caso el circuito se encarga de controlar la sensibilidad y filtrar la señal de los sensores.

Para poder diseñar un circuito adecuado a las necesidades del proyecto se ha consultado al profesor Alfredo Sanz. Su recomendación fue montar un circuito como el siguiente:

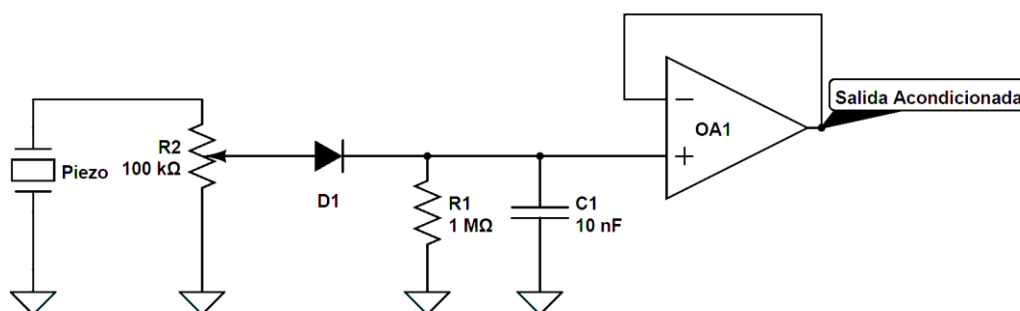


Figura 2.11: Circuito acondicionador de señal

El circuito tiene este diseño por las siguientes razones:

- El potenciómetro permite controlar la sensibilidad, pues un golpe con la misma fuerza en dos sensores distintos tendrá una respuesta ligeramente diferente. De esta forma, podemos tener el volumen balanceado en todos los pads que compongan el prototipo final.
- Con el diodo, resistencia y el condensador detectamos la envolvente de la señal y la filtramos mediante un FPB con frecuencia de corte entorno a los 16Hz, eliminando ruido y los componentes de alta frecuencia de la señal que provoca el sensor.
- Finalmente, se coloca un buffer con el fin de aislar el circuito del resto del sistema.

El circuito va alimentado con las salidas proporcionadas por la placa Nexys A7 entre 0 y 3,3V. Se realiza la siguiente simulación usando la herramienta de diseño PSpice con el fin de comprobar comportamiento del sistema.



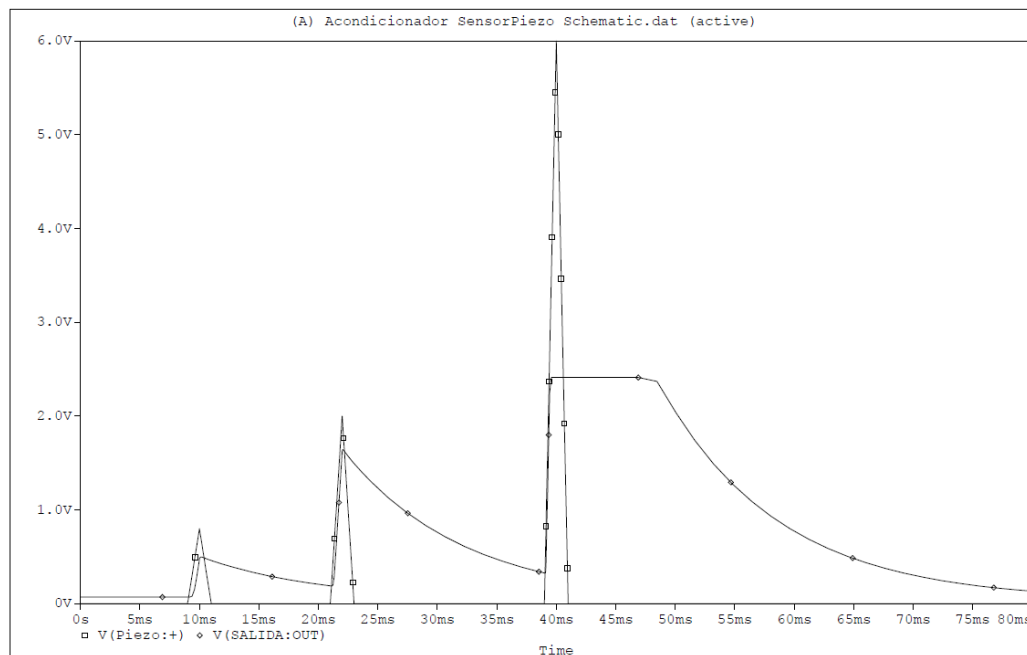


Figura 2.12: Simulación del circuito acondicionador de señal

Con esta simulación se verifica el comportamiento del circuito para su posterior montaje.

Una vez puesto en funcionamiento el circuito se realizan las siguientes medidas:

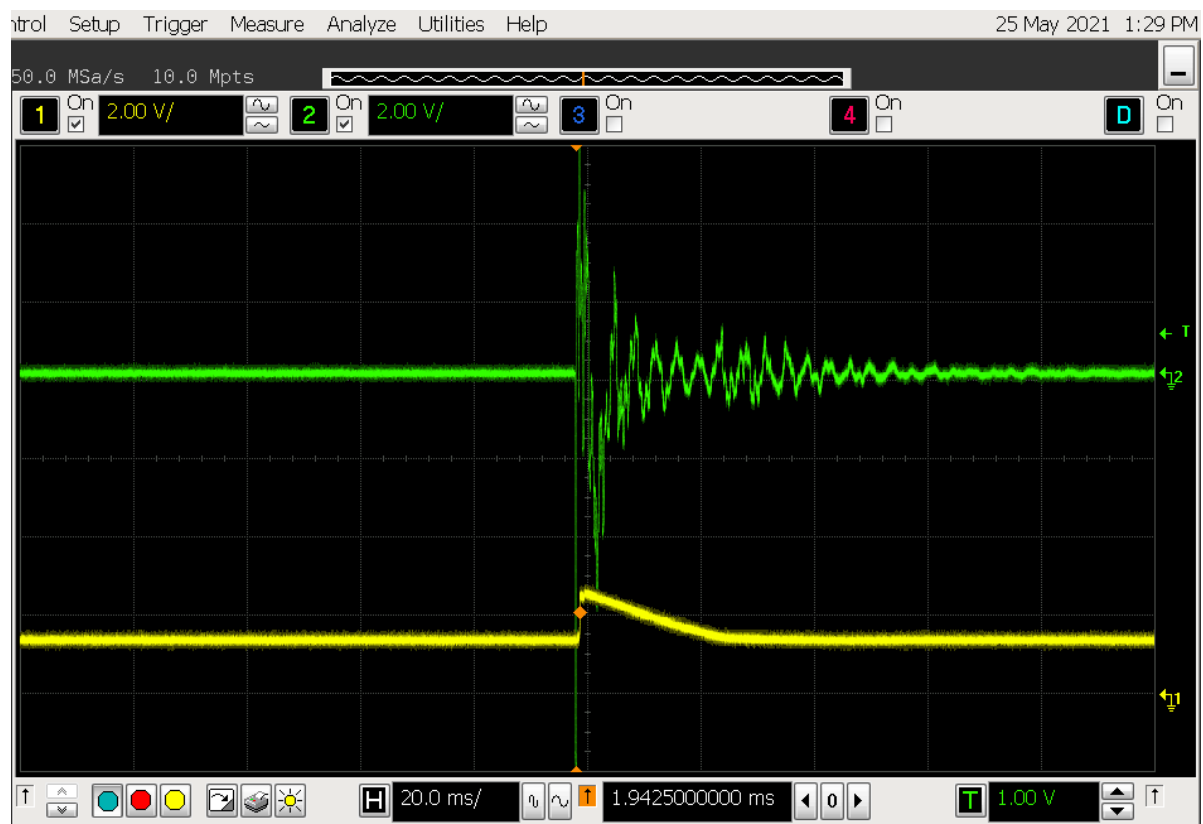


Figura 2.13: Medidas circuito de acondicionamiento

Podemos ver que existe mucho offset a la salida, esto se produce por alimentar el amplificador asimétricamente. Se soluciona midiendo los pulsos entre la salida del circuito y el siguiente divisor de tensión:

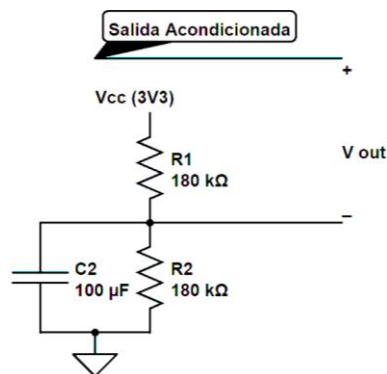


Figura 2.14: *Divisor de tensión*

Mediante este divisor junto con el condensador para eliminar el ruido obtenemos la siguiente salida:



Figura 2.15: *Comparación señal del sensor original y acondicionada*

Como se puede comprobar el pulso obtenido es apto para su posterior su conversión a digital.

A continuación, se puede ver el resultado del montaje del circuito:

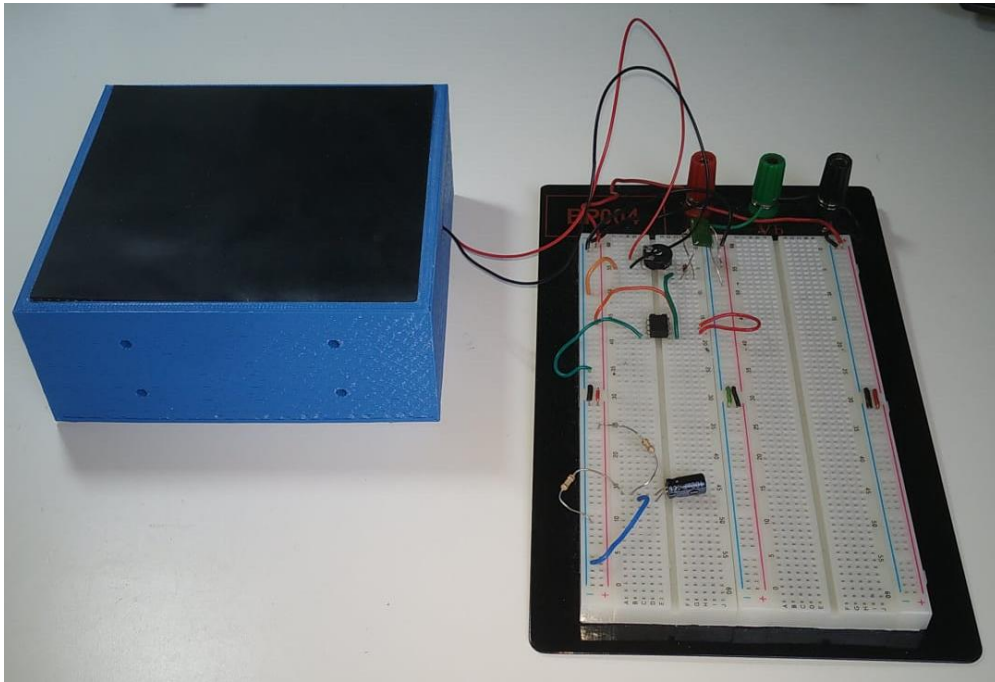


Figura 2.16: *Montaje del circuito de acondicionamiento*

## 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLADORES

En este capítulo se explican los diferentes bloques de la etapa digital del sistema para el procesamiento de audio, desde la conversión a digital de la entrada analógica hasta la salida de la señal de audio.

### 3.1. CONVERSIÓN ADC

La conversión ADC se realiza utilizando los conversores integrados en la FPGA, como se ha mencionado anteriormente. Las FPGAs de la familia Artix 7 incorporan varios conversores ADC a los que denominan con el nombre comercial XADC, en la placa Nexys A7 hay cuatro de estos XADCs accesibles. Para poder acceder a estos módulos utilizaremos el *XADC Wizard v3.3* [26] generado con la interfaz IP de Vivado.

La funcionalidad del XADC se configura a través de los registros de control (véase la sección *Register File Interface* en la Guía del Usuario del XADC [26]). Entre estos registros se encuentran múltiples datos, los que nos interesan son las lecturas de los puertos externos, cuyas direcciones se corresponden con los siguientes números hexadecimales: 12, 13, 1A y 1B.

Para poder acceder a estos registros se ha utilizado el código de la demostración proporcionada por *Digilent: Nexys A7-100T XADC Demo* [27]. Este proyecto es una demostración que utiliza la circuitería del convertidor analógico-digital de la placa Nexys A7-100T, los *switches*, los LEDs y el *display* de siete segmentos, escrita en Verilog. Cuando se programa en la placa, los niveles de voltaje entre 0 y 1 voltios se leen en el pmod JXADC. Mediante dos *switches* se selecciona el canal que se va a convertir. Los 16 LEDs de la placa se incrementan de derecha a izquierda a medida que aumenta la diferencia de tensión entre los pines del canal seleccionado. El *display* de siete segmentos muestra la diferencia de voltaje entre los pines del canal seleccionado en voltios.

Para poder utilizar el código de la demo en el presente proyecto, es necesario modificarlo de la siguiente manera. Los módulos encargados de controlar el *display* se eliminan, la funcionalidad de los *switches* se utiliza como puerto de entrada (SW) para seleccionar el canal y se emplean como salidas las siguientes señales:

- *do\_drp*: El bus de datos con la señal digital muestreada
- *drdy*: Una señal de control que avisa de la validez de los datos en *do\_drp*.

La funcionalidad de los LEDs se mantiene para facilitar el test en la placa. Estos cambios se realizan con el objetivo de multiplexar en el tiempo la lectura de datos de las cuatro entradas analógicas mediante la siguiente máquina de estados:

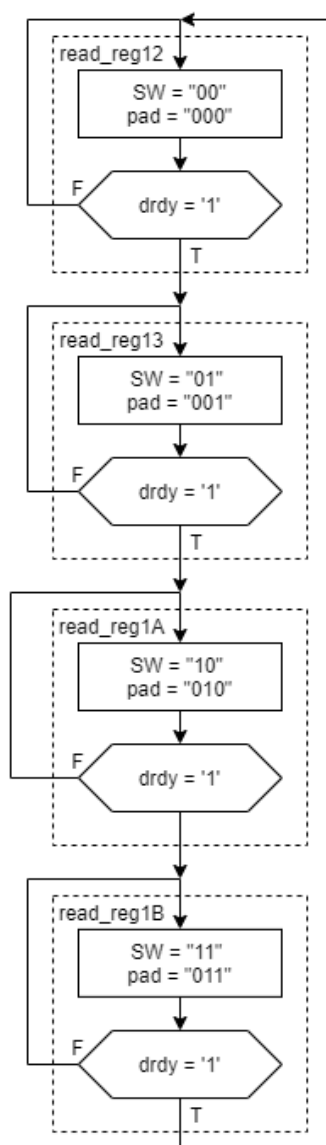


Figura 3.1: Diagrama ASMD para la lectura de registros XADC

Funciona con cuatro estados *read\_reg12*, *read\_reg13*, *read\_reg1A* y *read\_reg1B* que recorren secuencialmente los cuatro canales para actualizar el dato convertido de su volumen. En concreto, se encargan de solicitar la lectura de cada registro (modificando el valor de *SW*) y con la señal *pad* indica al resto del sistema qué canal se está leyendo. Una vez la señal *drdy* toma el valor '1', el valor del bus de datos, *do\_drp*, se traslada a la señal *Volumen\_in* utilizada por el resto del sistema.

A pesar de contar con 4 XADCs, se espera poder utilizar un número más elevado de canales empleando un esquema de multiplexación en el tiempo, asignando varios pads a un solo canal. Esta opción se contempla dentro del trabajo futuro y no se ha desarrollado en el proyecto.

Uno de los objetivos de este bloque es la obtención del valor de pico de los pulsos entrantes. Como se explica en apartados anteriores este valor de pico está directamente relacionado con la fuerza del golpe que reciben los pads. Este valor de pico sirve para la modulación del volumen con el que se reproduce el audio. Para obtener el valor de pico del pulso entrante se utiliza la siguiente máquina de estados:

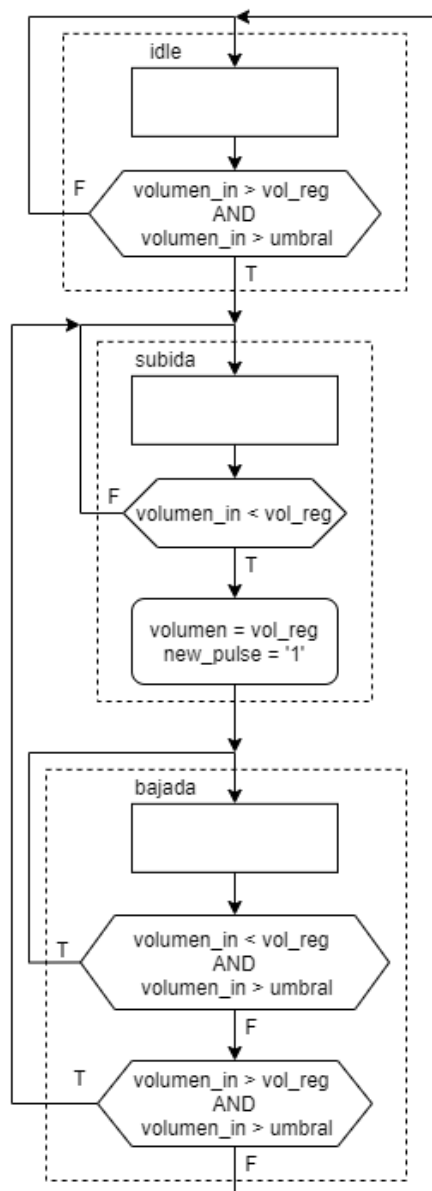


Figura 3.2: Diagrama ASMD para la detección del valor de pico

El diagrama se diseña en función de la forma de las señales analógicas entrantes. En estado de reposo la máquina se encuentra en el estado *idle*, en el momento que *volumen\_in* es mayor que el volumen registrado el anterior ciclo de reloj, *vol\_reg*, y mayor que un umbral para filtrar el ruido significa que el pulso se encuentra en un flanco de subida y se produce un cambio al estado *subida*. En este estado se comprueba cuando *volumen\_in* es menor que el volumen registrado, esta situación se cumple cuando comienza un flanco de bajada. En este momento, *vol\_reg* tiene el valor de pico del pulso, por tanto, este valor se asigna a la salida *volumen* y se indica al resto del sistema que ha llegado un nuevo pulso con la señal *new\_pulse*. En el estado de *bajada* si *volumen\_in* tiene un valor mayor al valor registrado significa que hay otro flanco de subida. Si *volumen\_in* tiene un valor inferior al umbral la señal entrante se encuentra en reposo.

Para aclarar el funcionamiento se presenta la siguiente gráfica:

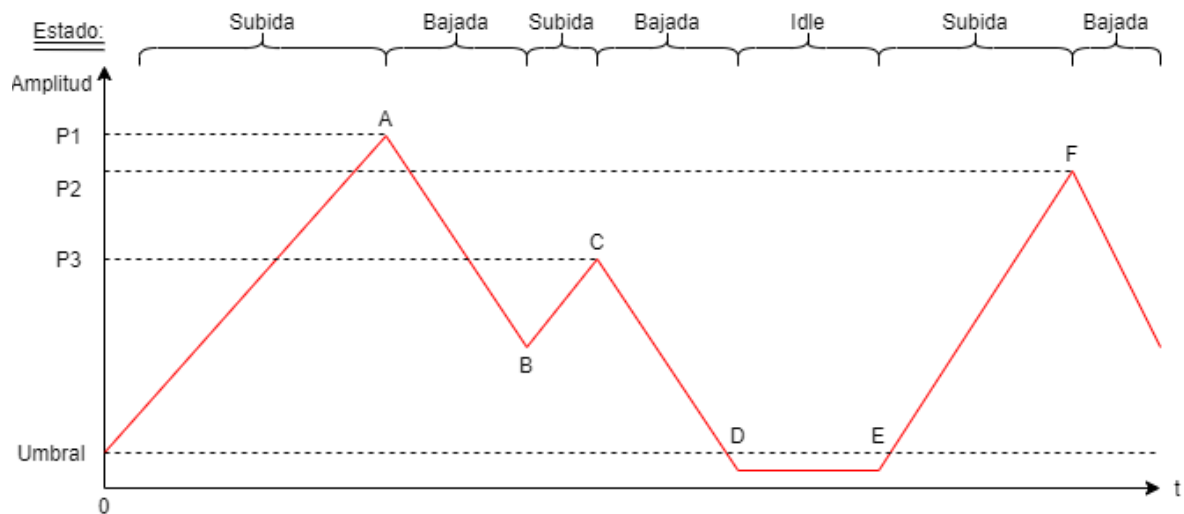


Figura 3.3: Gráfica de pulsos

En la gráfica se puede observar:

- Entre 0 y A el sistema se encuentra en el estado de *Subida*.
- Entre A y B el sistema se encuentra en el estado de *Bajada*.
- Entre B y C el sistema se encuentra en el estado de *Subida*.
- Entre C y D el sistema se encuentra en el estado de *Bajada*.
- Entre D y E el sistema se encuentra en el estado de *Idle*.
- Entre E y F el sistema se encuentra en el estado de *Subida*.

Y los valores de *Volumen* que recibe el resto del sistema son P1, P2 y P3 respectivamente.

Finalmente, el esquema del sistema de conversión ADC queda:

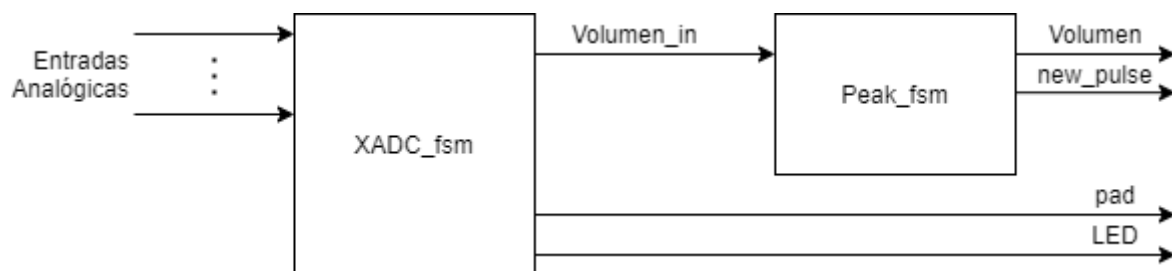


Figura 3.4: Esquema del sistema ADC

Siendo:

- *XADC\_fsm* el encargado de leer los registros.
- *Peak\_fsm* el encargado de la detección del valor de pico de los pulsos.

### 3.2.CONTROLADOR DE AUDIO DEL SISTEMA

En este apartado se van a explicar los diferentes bloques que conforman el controlador del sistema para generar el audio.

El controlador está compuesto por:

- **Controlador de reproducción.** Árbitro que decide como distribuir la información del bloque del sistema ADC. Es decir, decide cómo reproducir los pulsos entrantes.
- **Registro de datos:** Tiene registrados datos importantes para el correcto funcionamiento del sistema.
- **Controlador de accesos a memoria.** El sistema tiene cuatro módulos idénticos de este tipo. Este controlador se encarga de solicitar las lecturas a la memoria y proporcionar los datos al módulo PWM.
- **Interfaz con la memoria DDR2.** Este módulo lee los datos de la memoria siguiendo los pertinentes protocolos y proporciona esta información a los controladores de accesos a memoria.
- **PWM.** Este módulo transforma los datos digitales en un pulso PWM.
- **Generador de enables.** Este módulo genera enables para controlar la multiplexación en el tiempo de los accesos a memoria de los controladores.

Y están interconectados de la siguiente forma:

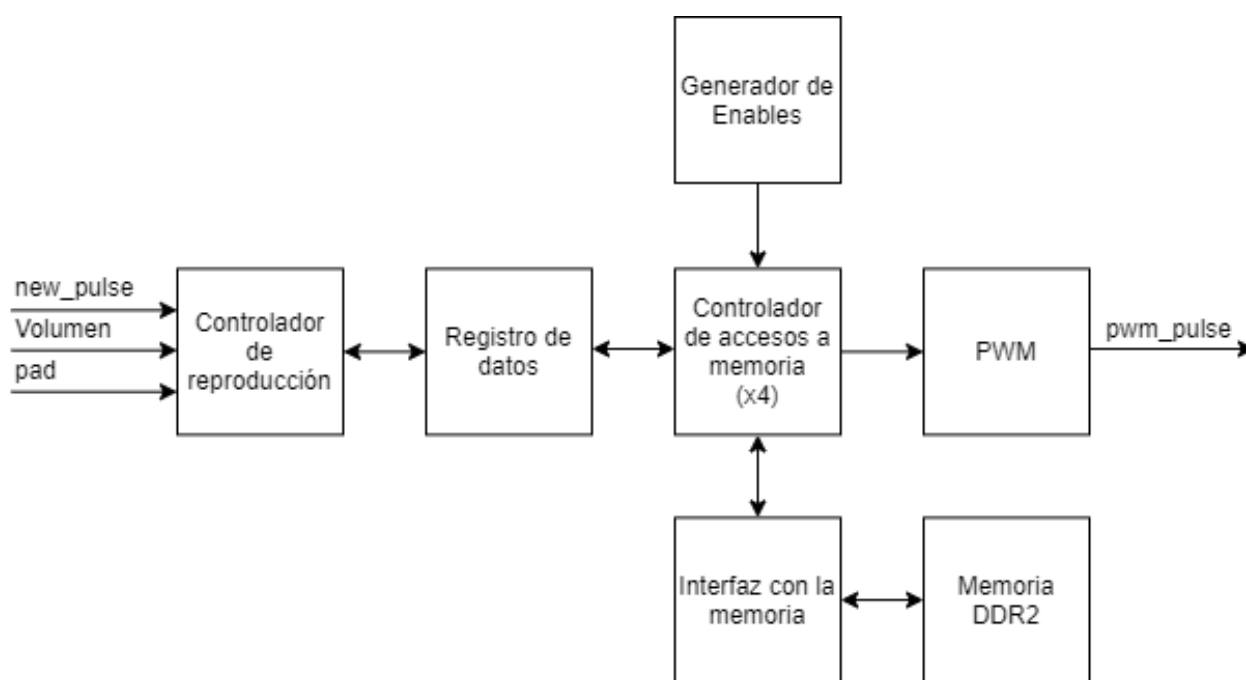


Figura 3.5: Esquema simplificado del controlador de audio



### 3.2.1. INTERFAZ DE COMUNICACIÓN CON LA DDR2

Para diseñar la interfaz de lectura de datos con la memoria DDR2 de la placa se ha hecho un previo estudio de su funcionamiento en las guías de *Xilinx: Zynq-7000 SoC and 7 Series Devices Memory Interface Solutions v4.2* [28] y *7 Series FPGAs Memory Interface Solutions* [29].

Mediante la interfaz IP de Vivado se va a implementar el módulo *Memory Interface Generator (MIG)*. Para su correcta configuración se han seguido los pasos de la sección *Using MIG in the Vivado Design Suite* de la guía *Zynq-7000 SoC and 7 Series Devices Memory Interface Solutions v4.2* [28]. Las señales correspondientes a la interfaz de usuario se especifican en la sección *Core Architecture* de la guía *7 Series FPGAs Memory Interface Solutions* [29].

#### ANÁLISIS DE PROTOCOLOS Y FUNCIONAMIENTO

La memoria requiere de una previa calibración que en simulación tarda entorno a los 50µs en simulación y en la placa es un tiempo variable dependiente de las condiciones actuales de los chips. Una vez completa la calibración, la memoria la confirma con la señal *init\_calib\_complete* que ya se puede leer y escribir en ella.

Los protocolos que sigue el módulo *MIG* para lectura y para escritura son los siguientes:

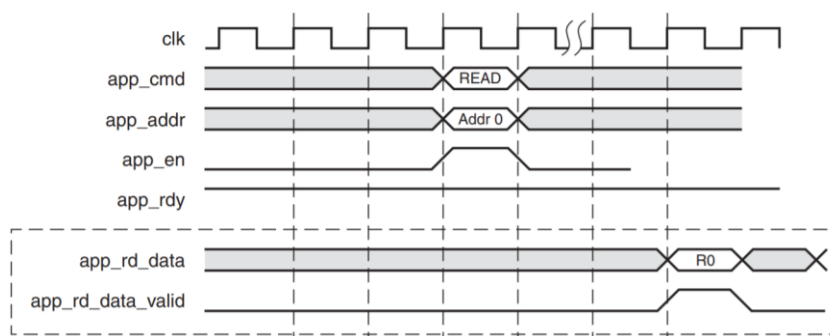


Figura 3.6: Protocolo de lectura DDR2

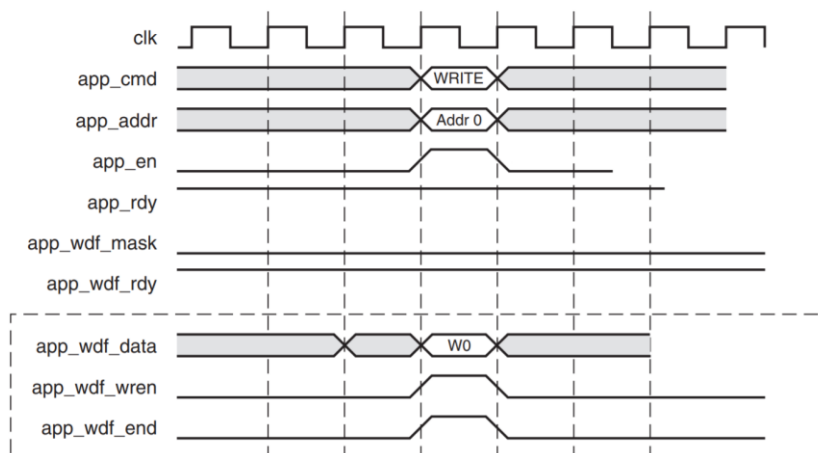


Figura 3.7: Protocolo de escritura DDR2

Con la señal *app\_cmd* se indica el comando que se pretende realizar, este comando se valida mediante la señal *app\_en*. Para el caso de lectura la señal *app\_rd\_data\_valid* confirma que los datos de la señal *app\_rd\_data* son los correspondientes a la dirección requerida. Por otro lado, para el caso de escritura la señal *app\_wdf\_data* representa los datos que se quieren escribir, mientras que *app\_wdf\_end* indica que el ciclo de reloj actual es el último ciclo de datos de entrada y *app\_wdf\_wren* es un enable para permitir la escritura.

La comprensión y el correcto funcionamiento de estas señales es clave para poder diseñar un controlador. La máquina de estados que se utiliza es la que se encuentra a continuación:

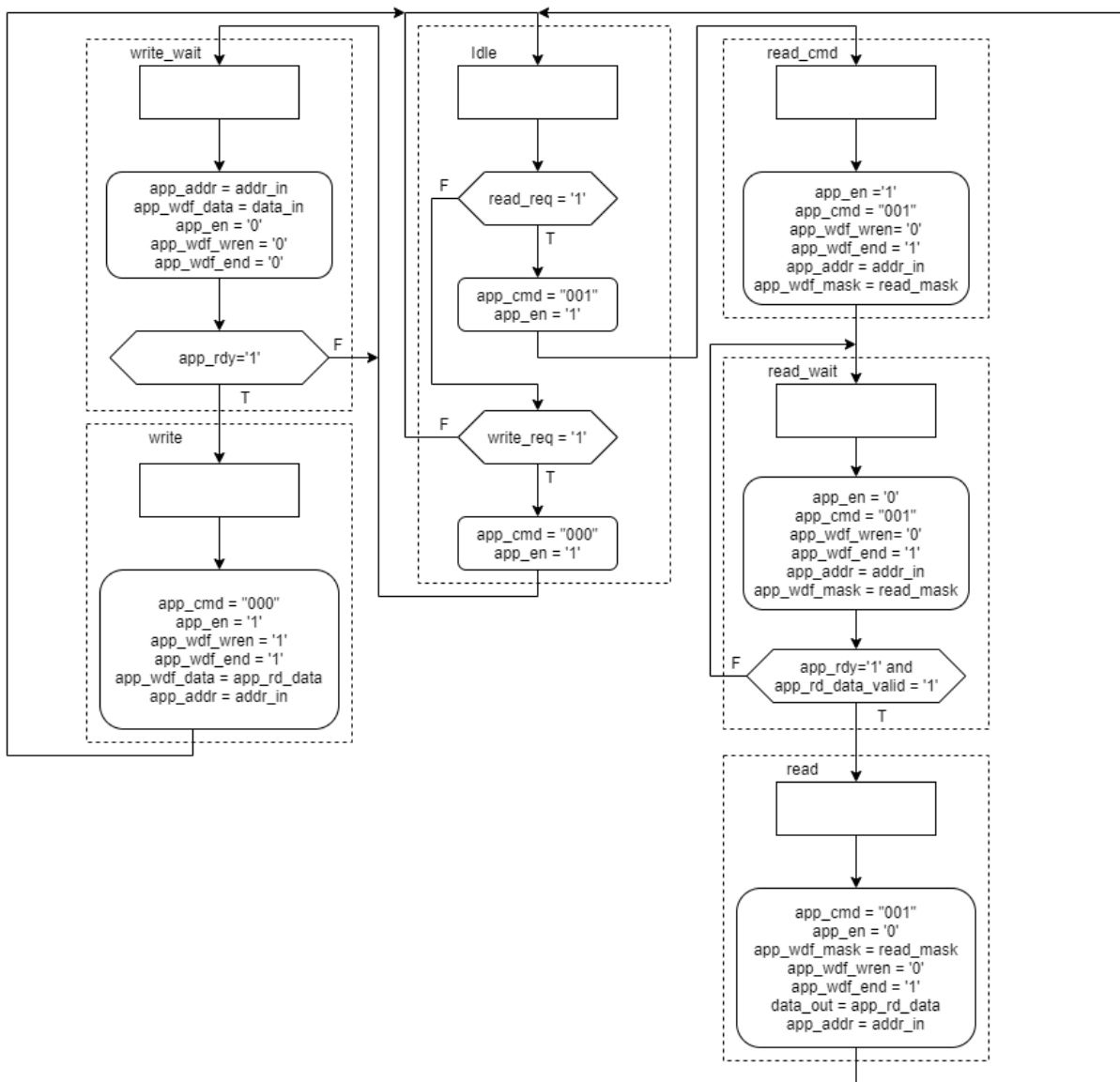


Figura 3.8: Diagrama ASMD del controlador del MIG

El estado de *idle* simplemente se encarga de realizar el comando de lectura o de escritura en función de las señales *read\_req* y *write\_req* respectivamente.

Para la escritura se han diseñado dos estados: *write\_wait* que comprueba que la señal *app\_rdy* tenga valor '1', esta señal indica si la interfaz no está ocupada; y el estado *write* que escribe en memoria. Por otra parte, para la lectura de datos se han diseñado tres estados: el primero, *read\_cmd*, sirve para asegurar que el comando de lectura dure al menos un ciclo de reloj como requiere la memoria, el segundo, *read\_wait*, se encarga de esperar la activación de *app\_rd\_data\_valid*, y finalmente el estado *read* saca los datos leídos a través de *data\_out*.

A continuación, podemos ver una simulación de escritura de datos en una dirección aleatoria y posteriormente lectura de la misma dirección. Se puede observar tanto una correcta transición de los estados como una correcta escritura y lectura de estos. Cabe destacar la importancia de la señal *app\_rd\_data\_valid* pues hasta que su valor no es '1' en el bus de salida de datos hay un valor

totalmente ajeno a la lectura. La activación de esta señal tarda aproximadamente 15 ciclos trabajando a 100MHz, dato que será útil más adelante.

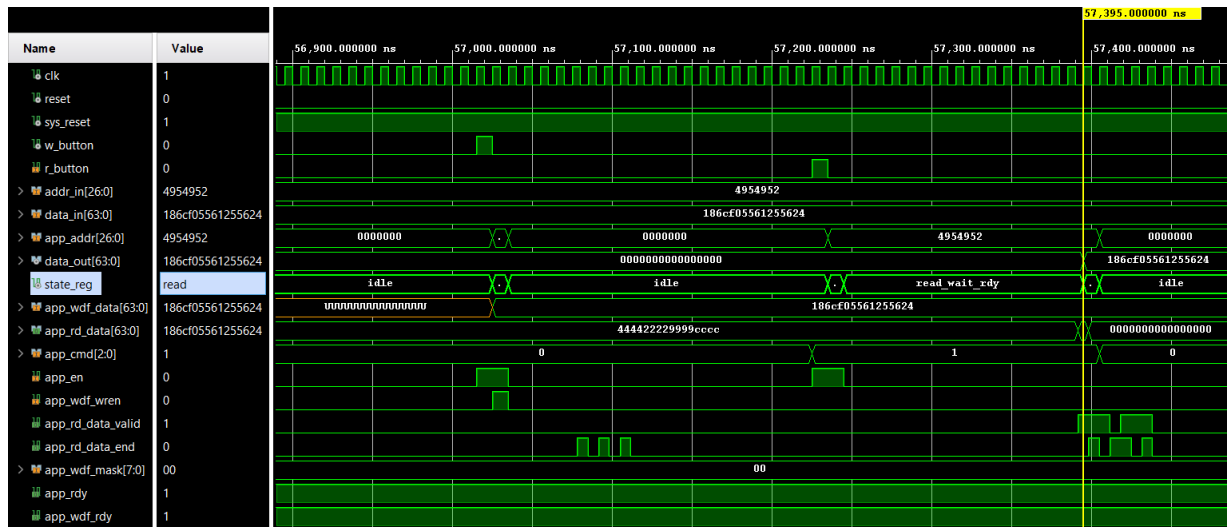


Figura 3.9: Simulación de lectura y escritura en memoria

### CONTROLADOR DE LA INTERFAZ

Con el objetivo de simplificar el modelo se ha reducido el diseño del anterior apartado para controlar la lectura de datos exclusivamente, ya que debido a las especificaciones del proyecto la operación de escritura no se va a realizar. Quedando el siguiente diagrama:

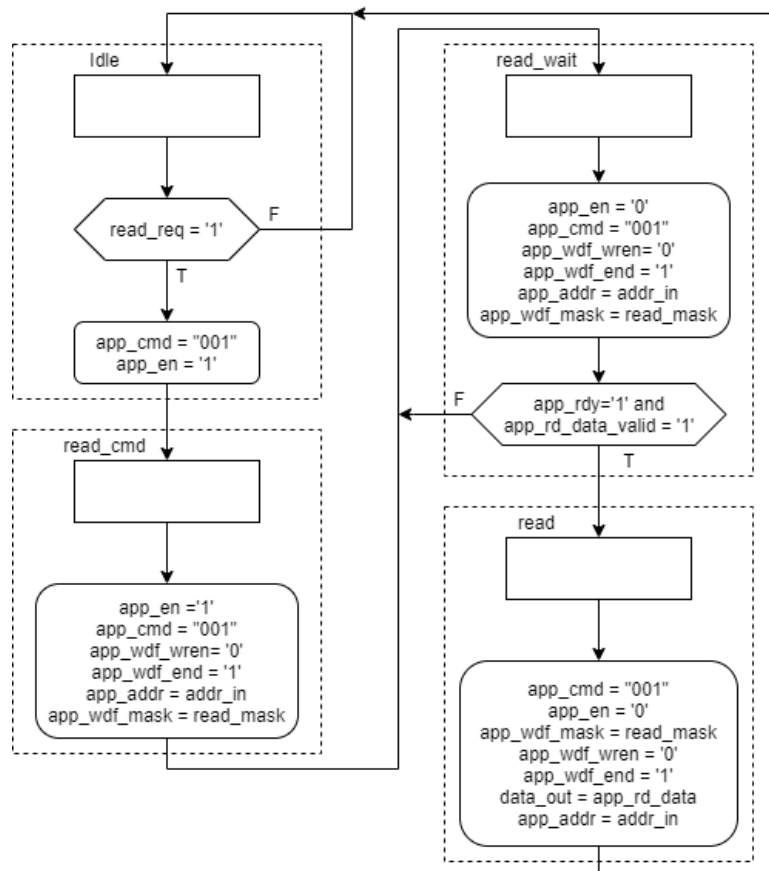


Figura 3.10: Diagrama ASMD del controlador de lectura de datos de la memoria

## ESCRITURA DE BANCO DE SONIDOS EN LA MEMORIA DDR2

Este punto ha sido problemático durante el desarrollo del proyecto. En el apartado anterior se plantea que el dispositivo final solo realiza lecturas de datos de la memoria, pero para que esto sea posible se tiene que escribir en la memoria previamente los datos de los sonidos a reproducir. Es decir, los archivos, .wav en este caso, del banco de sonidos almacenados en el ordenador tienen que ser escritos en la memoria.

Debido al tiempo disponible y la dificultad que conlleva este paso no se ha logrado alcanzar una solución que incluya la utilización de un protocolo de comunicación con el PC y el acceso rápido a la memoria DDR con los datos recibidos.

Con respecto al último punto, durante el desarrollo de la interfaz con la memoria se han realizado muchas pruebas y diseños de diferentes controladores para realizar lecturas y escrituras de direcciones individuales así como de ráfagas de datos. Estas pruebas se comentan en detalle en el siguiente capítulo del documento.

En lo relativo al protocolo de comunicaciones, se han planteado algunas soluciones, pero no se ha llegado a desarrollar ninguna.

La resolución de esta parte del proyecto ha supuesto una gran inversión de tiempo. En la sección de trabajo futuro se contemplan distintas vías para abordar este problema.

### 3.2.2. CONTROLADOR DE REPRODUCCIÓN

El sistema debe ser capaz de reproducir varios sonidos a la vez, mientras se esté reproduciendo uno de los pads seguirán llegando pulsos. Esto implica que el sistema requiere un control de los sonidos que está reproduciendo, este bloque se ha diseñado para manejar y distribuir la información entrante de los pads.

El prototipo va a ser capaz de reproducir hasta cuatro sonidos a la vez. Cada vez que llegue un pulso nuevo se registra el volumen y el pad del que proviene. Esta información se distribuye a cuatro módulos idénticos que se encargan de solicitar los datos del pad correspondiente a la memoria.

La decisión de a cuál de estos módulos se asigna cada nuevo pulso se toma en base a dos datos: *uso* y *orden*.

- *Uso* indica si el módulo se encuentra en uso o no.
- *Orden* indica cuándo ha empezado a usarse el módulo siendo 1 el más nuevo y 4 el más antiguo.

Desde el estado de *idle* cuando se reciba un pulso nuevo, *New\_pulse*, se comprueba cuál de los módulos está en uso. Si hay alguno libre, el pulso se asigna al primer módulo detectado como libre. En caso contrario, es decir, que estén todos en uso, el pulso se asigna al módulo que lleve más tiempo en uso. Para ello se comprueba en *Search\_older* cuál de los módulos tiene orden 4, desalojando los datos viejos y dando paso a los datos del pulso nuevo. Cada una de estas operaciones conlleva la actualización de los datos de *orden*.

Por tanto, el diagrama que tendrá el controlador será el siguiente:

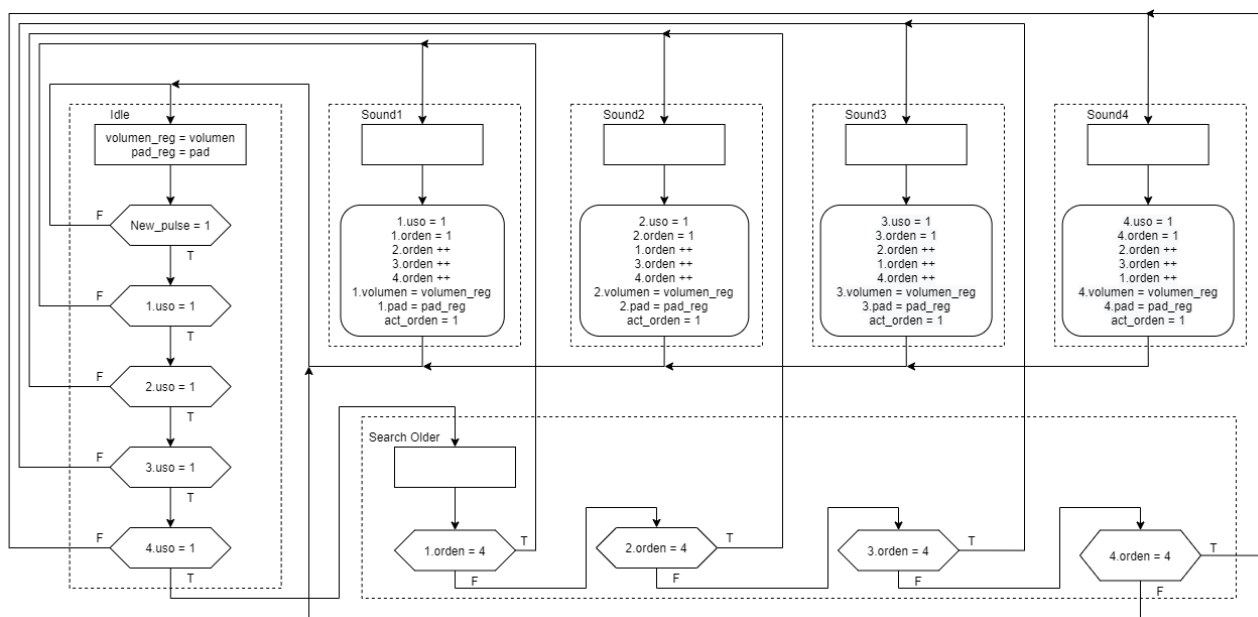


Figura 3.11: Diagrama ASMD del controlador de reproducción

El apartado siguiente describe los cuatro módulos idénticos encargados de los accesos a memoria.

### 3.2.3. CONTROLADOR DE ACCESOS A MEMORIA

El acceso a memoria lo realizan cuatro bloques en un esquema de multiplexación en el tiempo. Un generador de enables dedicado es quien arbitra mediante estas señales de enable el acceso de cada uno de estos módulos a los buses de la memoria, datos y direcciones. En función de si se asigna un pulso o no a cada controlador, éste realiza peticiones a memoria o no.

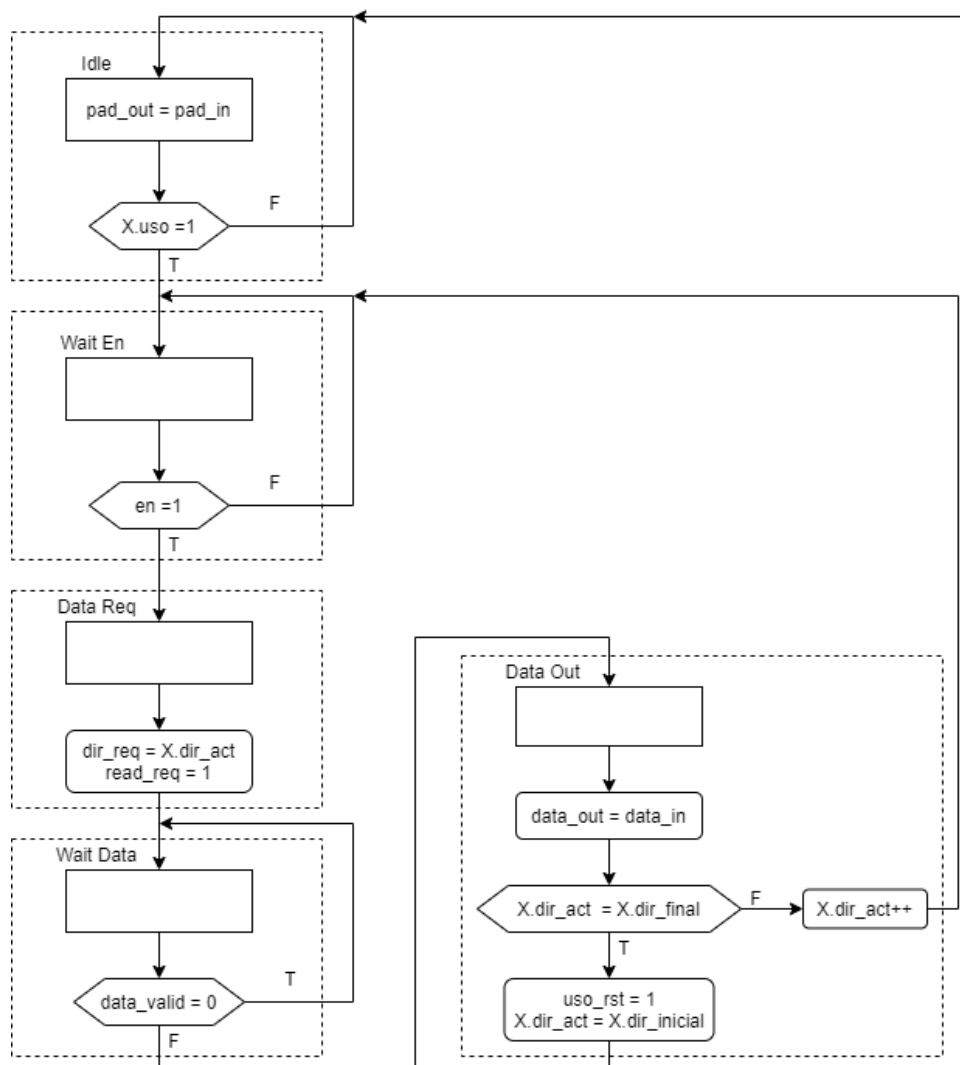


Figura 3.12: Diagrama ASMD del controlador de accesos a memoria

La figura 3.10 representa el diagrama ASMD que describe el funcionamiento del controlador. Como puede observarse el sistema espera a que esté en uso ( $X.uso = 1$ ) en el estado de *idle*. Una vez en uso (estado *wait\_en*) el sistema tiene que esperar a que le llegue el turno del esquema de multiplexación en el tiempo ( $en = 1$ ). El valor del bus de direcciones, compartido por los cuatro módulos, tendrá valor 'Z' si el valor de la señal enable es nulo. De esta forma se evita que dos módulos escriban datos a la vez en el bus.

Una vez activado el *enable* ejecuta una solicitud de lectura de la dirección actual que necesita ser leída en el estado *Data\_Req*. A continuación, desde el estado *Wait\_Data* se espera a que la señal *data\_valid* tenga valor '1', señal correspondiente a *app\_rd\_data\_valid* del controlador de la memoria que indica la validez de los datos leídos como se ha explicado en apartados anteriores. Tras haberse activado se hace la transición al estado *Data\_out* donde se sacan los datos leídos y se incrementa la dirección actual.

Este proceso se repite hasta que las direcciones actual y final coincidan, lo que significará que se han leído todos los datos, entonces la señal de *uso\_rst* pasará a tener valor '1' y se habrá finalizado la lectura de datos dejando el módulo libre para reproducir un nuevo sonido.

La información sobre la dirección inicial y final que debe leer se obtiene gracias al dato del *pad* que debe reproducir, se explica con más detalle en el siguiente apartado.

### 3.2.4. REGISTRO DE DATOS

Este módulo almacena datos relevantes, necesarios para los otros bloques, permitiendo así el correcto funcionamiento del sistema. Hay un módulo de registro por cada controlador de accesos a memoria.

A continuación se explican con detalle los datos guardados en el módulo:

- *Orden*: indica al sistema el tiempo que lleva en uso el sistema, siendo 1 el más nuevo y 4 el más antiguo. Este dato se actualiza con cada activación de la señal *act\_orden*. Este dato lo escribe y lee el controlador de reproducción.
- *Uso*: indica si el módulo controlador de accesos a memoria se encuentra en uso o no. El controlador de reproducción activa esta señal y lo lee el controlador de accesos a memoria. El valor de esta señal pasa a ser '0' si el controlador de accesos a memoria activa la señal *uso\_rst*.
- *Volumen*: esta señal tiene guardado el valor de pico de los pulsos analógicos entrantes a la placa. Es escrito por el controlador de reproducción y leído para modular la intensidad de la señal de salida.
- *Dir\_inicial* y *Dir\_final*: estos datos están guardados estáticamente en *dir\_ini1*, *dir\_ini2*, *dir\_ini3* y *dir\_ini4* para *Dir\_inicial*, y en *dir\_fin1*, *dir\_fin2*, *dir\_fin3* y *dir\_fin4* para *Dir\_final*. Se proporcionan unas direcciones inicial y final u otras en función del *pad* que se solicite.

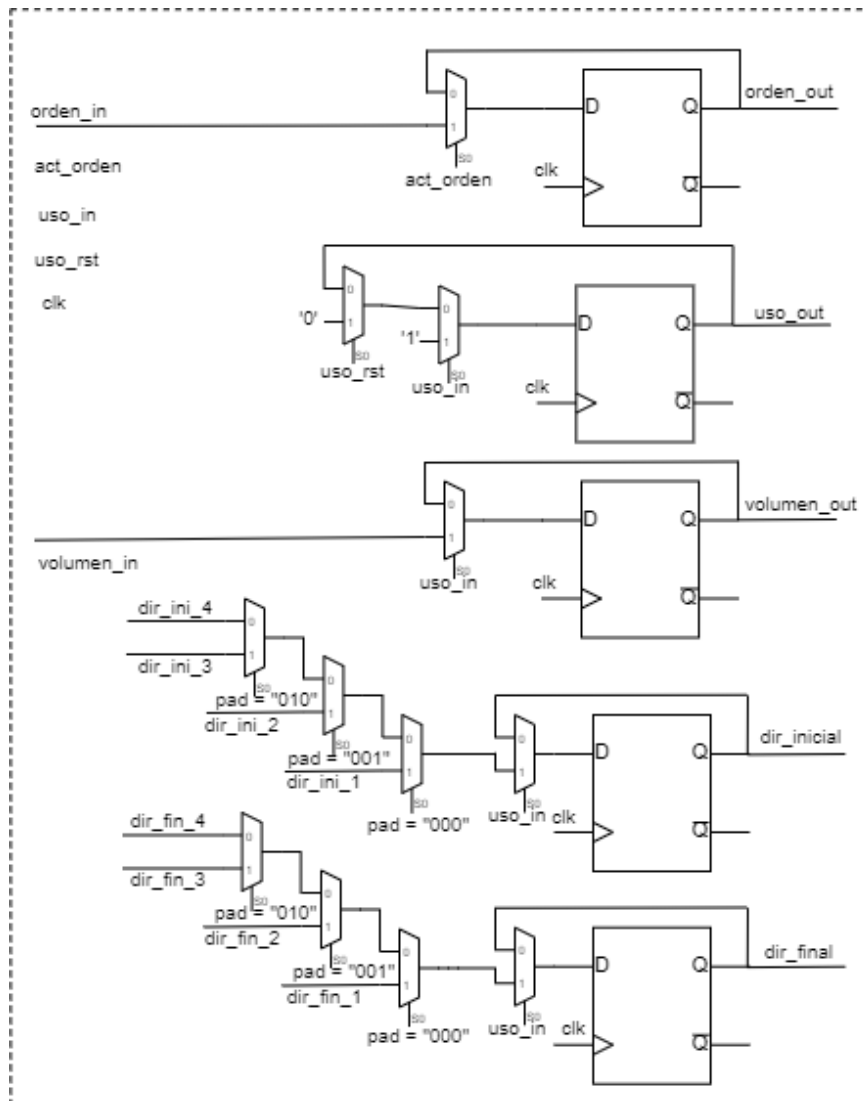


Figura 3.13: Circuito del registro de datos



### 3.2.5. GENERADOR DE ENABLES

El bloque encargado de generar enables es el siguiente:

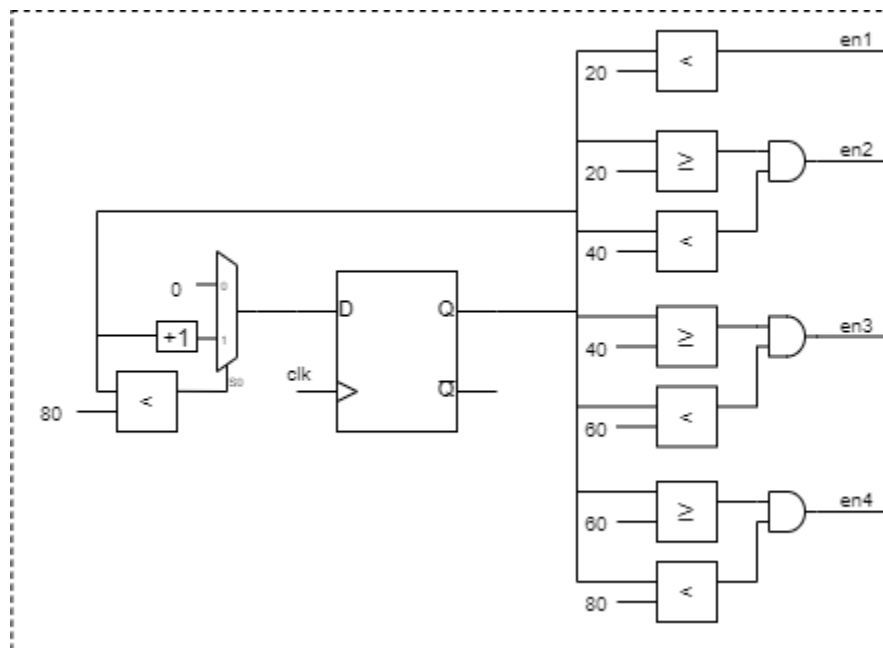


Figura 3.14: Circuito generador de enables

Las señales *en1*, *en2*, *en3* y *en4* son los enables de los módulos de accesos a memoria. Cada una de estas señales estará activa durante 20 ciclos de reloj. Haber elegido este número se debe a que la señal de *app\_rd\_data\_valid*, que indica la validez de los datos leídos de la memoria DDR tarda aproximadamente 15 ciclos en valer '1', como se explicó anteriormente.

### 3.2.6. GENERACIÓN DE LA SEÑAL DE SALIDA

La señal de salida se genera multiplicando por los datos de *Volumen* las salidas de los controladores de accesos a memoria, es decir, los datos leídos, los sonidos a reproducir. Posteriormente estos datos se suman y genera una señal PWM.

Una señal PWM (*pulse-width modulated*) es una cadena de pulsos a una frecuencia fija, con cada pulso de una anchura diferente. Esta señal digital puede pasarse a través de un simple filtro de paso bajo que integra la forma de onda digital para producir una tensión analógica proporcional a la anchura media de los impulsos a lo largo del tiempo [3]. Por ejemplo, si los pulsos tienen valor '1' durante un promedio del 10% del periodo, entonces un integrador producirá un valor analógico que es 10% de la tensión Vdd. La figura 3.15 muestra una forma de onda representada como una señal PWM.

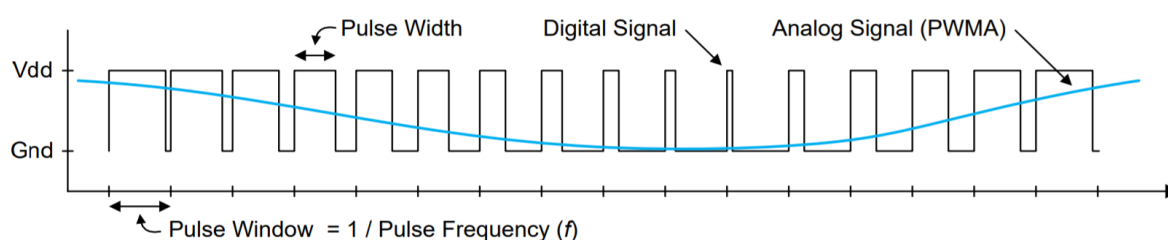


Figura 3.15: Onda representada como PWM [3]

La salida jack de audio de la placa está controlada por un filtro Sallen-Key Butterworth de paso bajo de 4º orden que proporciona una salida de audio mono [3]. El circuito del filtro de paso bajo se muestra en la Figura 3.16. La entrada digital es la señal PWM generada. La señal debe ser conducida a un nivel bajo para el "0" lógico y dejada en alta impedancia para el "1" lógico. Una resistencia de pull-up incorporada a un raíl analógico limpio de 3,3V establecerá la tensión adecuada para el '1' lógico. El filtro de paso bajo en la entrada actuará como un filtro de reconstrucción para convertir la señal PWM en un voltaje analógico en la salida del jack de audio.

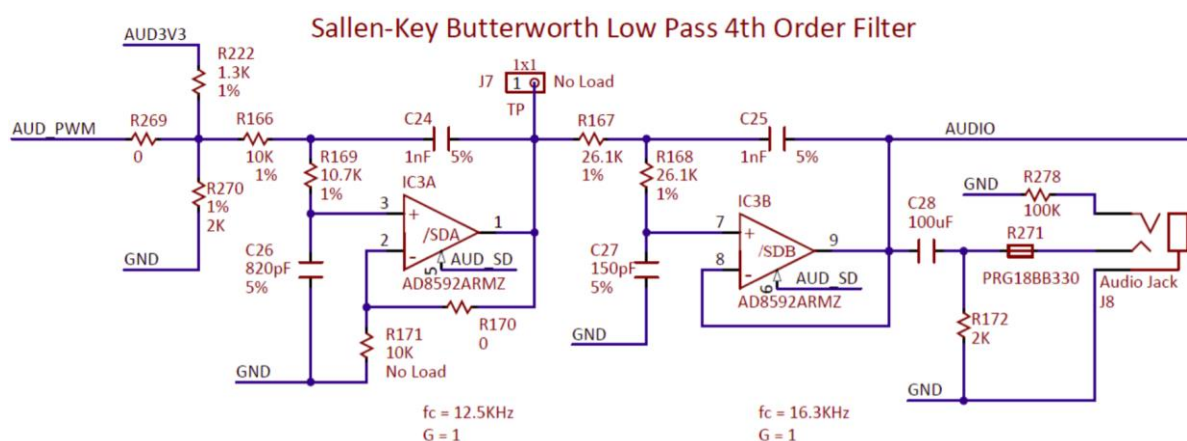


Figura 3.16: Filtro Sallen-Key Butterworth paso bajo de 4º orden [3]

La respuesta en frecuencia del filtro de paso bajo Sallen-Key Butterworth se muestra en la siguiente figura:

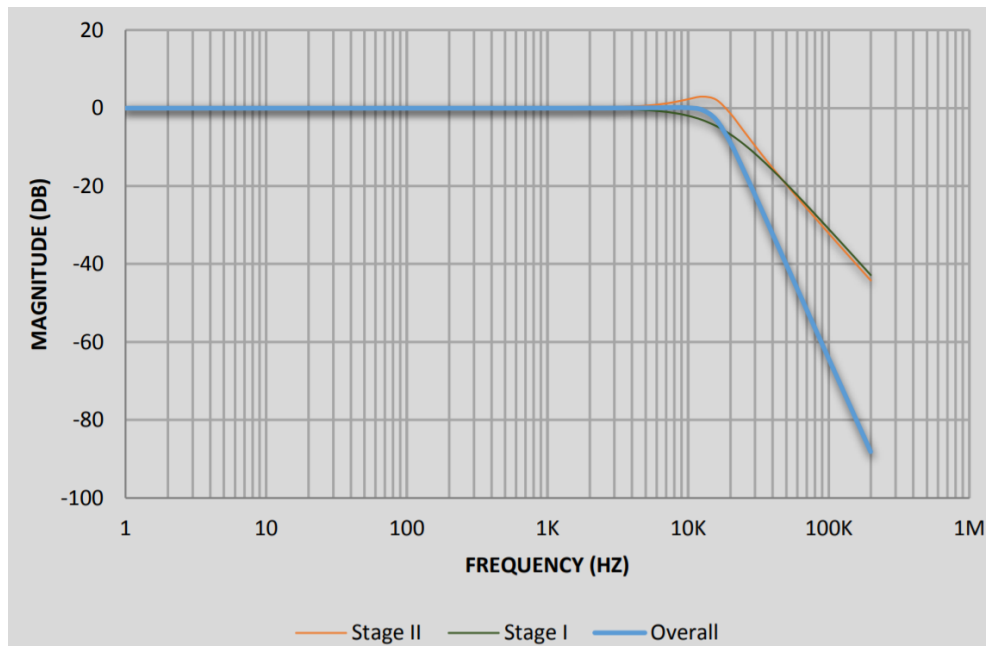


Figura 3.17: Respuesta en frecuencia del filtro de paso bajo Sallen-Key Butterworth [3]

Como los archivos de audio que se reproducirán están muestreados a 44 kHz, la tasa de generación de pulsos tiene que ser la misma: un pulso cada 22.72  $\mu$ s, aproximadamente.

Las muestras digitales de audio del banco de sonidos son mono de 16 bits, es decir, el máximo valor representado es 65536. En un período de muestreo de 22.72  $\mu$ s, un contador que cuente entre 0 y 65536 necesitaría trabajar aproximadamente a 3 GHz, por lo tanto nuestro sistema no va a ser capaz de proporcionar la precisión requerida. Con nuestro reloj de 100 MHz, en 22.72  $\mu$ s somos capaces de producir 2272 pulsos, con lo que sólo podremos tener en cuenta los 11 bits más significativos. De esta forma el valor máximo representado es 2048, por lo tanto todos datos reproducidos van a tener una pequeña disminución de volumen por un factor de  $2048/2272 = 0.901$ . Este recorte de bits supone que la conversión de digital a analógico utilizando señales PWM no es la óptima para el sistema, en trabajo futuro se analizan otras opciones.

El valor de salida del contador se compara con la palabra digital que se quiere convertir, *sample\_in*, si el valor del contador es más pequeño, la salida PWM, *pwm\_pulse*, se pone a '0' y a '1' en el caso contrario.

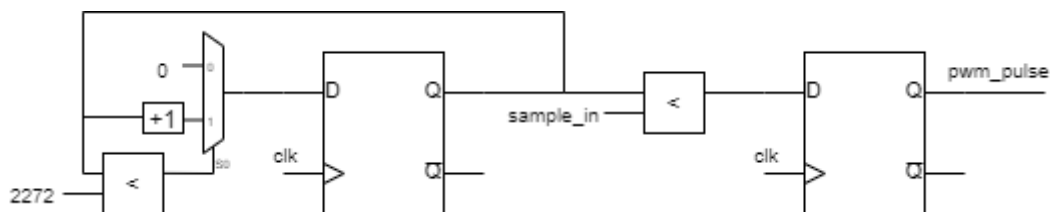


Figura 3.18: Módulo PWM

El esquema final del controlador de audio detallado queda de la siguiente forma:

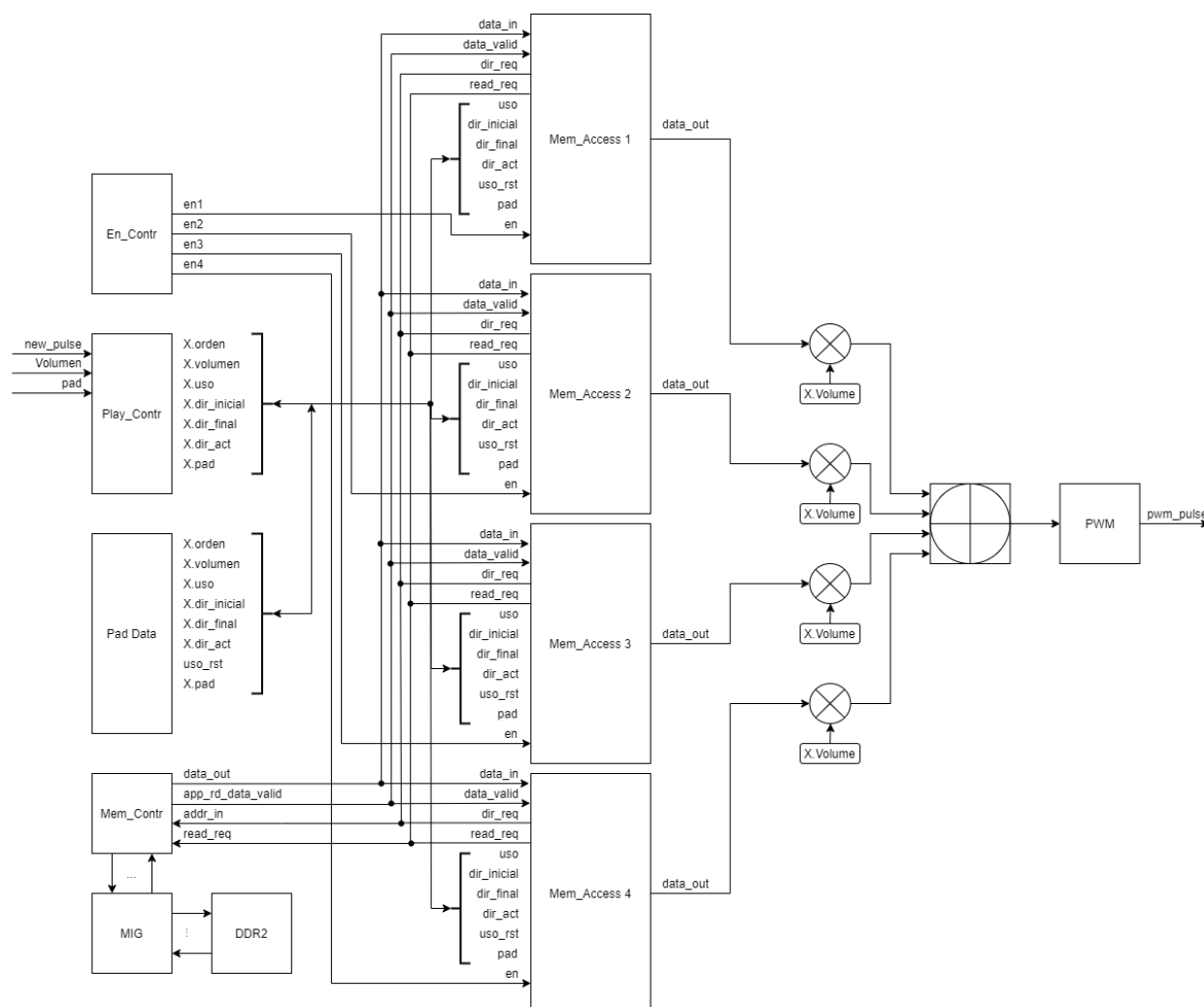


Figura 3.19: Esquema del controlador de audio

Siendo:

- *En\_Contr*: Circuito generador de enables.
- *Play\_Contr*: Controlador de reproducción del sistema.
- *Pad\_Data*: Registro de datos.
- *Mem\_Contr*: Controlador de la interfaz con la memoria.
- *Mem\_Access* (1, 2, 3 y 4): Controlador de accesos a memoria.

Finalmente, el esquema de la parte digital del sistema es el siguiente:

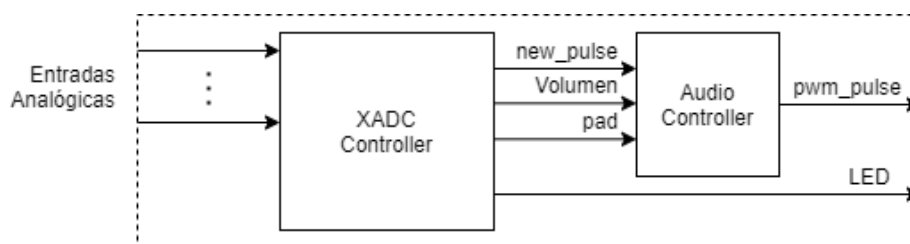


Figura 3.20: Esquema parte digital del sistema

## 4. PRUEBAS DE LOS ALGORITMOS DESARROLLADOS

Este capítulo tiene como objetivo demostrar el correcto funcionamiento de los módulos desarrollados.

Las pruebas realizadas son:

- **Test-Benchs.** Son ficheros en lenguaje VHDL que emplea el entorno de simulación de Vivado para estudiar el funcionamiento de la lógica programable desarrollada a lo largo del tiempo.
- **Test-Manual.** Se carga el programa desarrollado en la Nexys A7 a través de un Bitstream y se comprueba de manera visual su funcionamiento.

Dependiendo del tipo de módulo se realiza una prueba o ambas en función de la relevancia de la información que se pueda obtener.

### 4.1. PRUEBAS DE LA CONVERSIÓN ADC

Para comprobar el funcionamiento del conversor ADC se debe crear un archivo que simule las entradas analógicas pues estas no se introducen en la simulación utilizando el Test-Bench. El atributo SIM\_MONITOR\_FILE utilizado en la instanciación del XADC apunta la ubicación de este archivo conocido como *Analog Stimulus file*. Este archivo tendrá una forma similar a las siguientes líneas:

```
TIME VAUXP[0] VAUXN[0] VAUXP[1] VAUXN[1] VAUXP[2] VAUXN[2] VAUXP[3] VAUXN[3]
00000 0.005 0.0 0.200 0.0 0.50 0.0 0.1 0.0
34000 0.020 0.0 0.400 0.0 0.49 0.0 0.2 0.0
67000 0.049 0.0 0.600 0.0 0.51 0.0 0.5 0.0
99999 0.034 0.0 0.900 0.0 0.53 0.0 0.0 0.0
```

El formato del archivo de estímulos analógicos se basa en datos delimitados por espacios o por tabulaciones y puede ser creado en una hoja de cálculo, por ejemplo. Toda la información de la marca de tiempo debe aparecer en la primera columna. Otras columnas enumeran los valores analógicos para los sensores en el chip y las entradas analógicas externas. El orden de las columnas no es importante, el único requisito es que la información de la marca de tiempo aparezca en la primera columna. Por cada marca de tiempo añadida a la primera columna, se añade un valor correspondiente a las otras columnas. Sólo es necesario listar las columnas de los canales de entrada analógica necesarios. En este ejemplo, sólo los canales auxiliares del 0 al 3 están listados en el archivo.

Para realizar estas pruebas se ha hecho un archivo en una hoja de cálculo con una señal senoidal para las cuatro entradas analógicas. El resultado esperado es la misma señal senoidal sin cortes pues se leen todas las entradas cíclicamente:

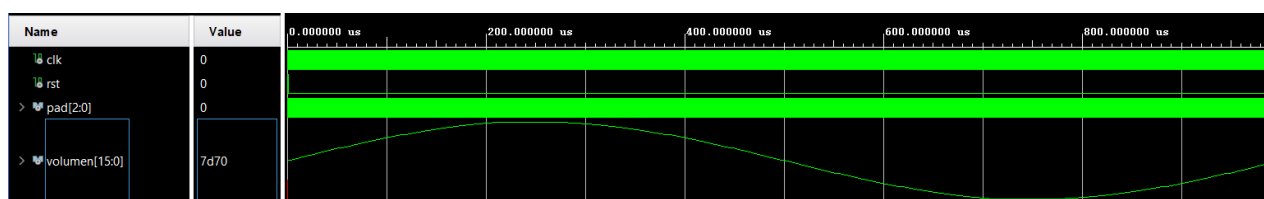


Figura 4.1: Test-bench para comprobar el funcionamiento del conversor ADC

Una vez comprobado que el bloque funciona correctamente, se carga dentro de la Nexys A7 el archivo bitstream y con la ayuda de los LEDs de la placa se comprueba que se ha implementado correctamente.

Para comprobar en placa se usa un generador de señales. Generando una señal senoidal de 1Hz, gracias a la funcionalidad implementada en los LEDs, se aprecia como la señal se convierte correctamente. Como se explica anteriormente en el documento los 16 LEDs de la placa se incrementan de derecha a izquierda a medida que aumenta la diferencia de tensión entre los pines del canal seleccionado. Por tanto, en la placa se ve como se encienden y se apagan los LEDs en función de la amplitud de la señal entrante.

Tras comprobar el correcto funcionamiento con una señal controlada se hacen pruebas con el circuito de acondicionamiento y los sensores. El funcionamiento entra dentro de lo esperado.

## 4.2.PRUEBAS DEL DETECTOR DE PICOS

Con el mismo archivo de estímulos del apartado anterior comprobamos el funcionamiento del detector de picos.

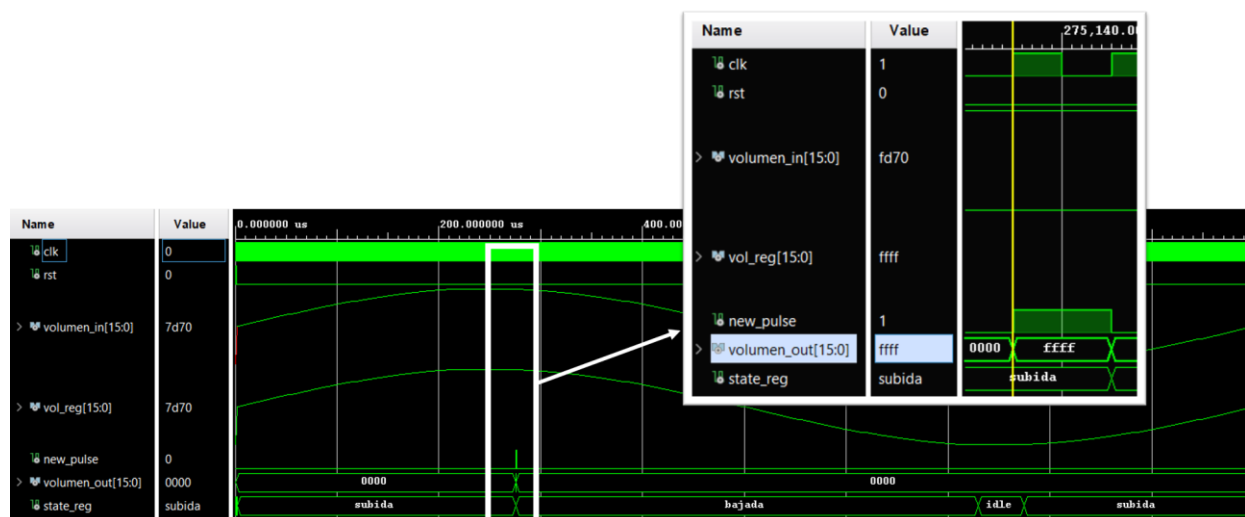


Figura 4.2: Test-bench para comprobar el funcionamiento del detector de picos

Se puede comprobar como en el momento que el volumen registrado, *vol\_reg*, tiene un valor más grande que el volumen entrante, *volumen\_in*, se generan las señales de salida, *new\_pulse* y *volumen\_out*, correctamente. Cabe destacar las correctas transiciones entre los diferentes estados.

## 4.3.PRUEBAS DE LA INTERFAZ CON LA MEMORIA

Las primeras pruebas realizadas consisten en escribir en las 100 primeras posiciones de memoria los números del 1 al 100 y posteriormente leer las mismas direcciones usando un diagrama similar al de la Figura 3.8 (que únicamente escribe de un dato en un dato, no 100 consecutivamente). Tanto para leer como para escribir sigue los mismos protocolos que ese controlador.

Para escribir los datos se activa la señal *write\_req* comienza el proceso de escritura escribiendo cada número en las respectivas direcciones ordenadamente. Una vez finalizado este proceso de escritura se activa *read\_req* y comienza el proceso de lectura, donde se leen todas las direcciones ordenadamente esperando la señal de validación de los datos leídos, *app\_rd\_data\_valid*, para cada una de las direcciones.

La realidad es que este proceso no llega a funcionar debidamente. Aparentemente, la escritura de datos funciona como es debido, pero en el momento de leer las direcciones los datos no se guardan de forma correcta. Tal y como se propone el controlador, en la dirección 'X' debe estar guardado el número 'X', por ejemplo, en la dirección número 53 el dato a leer debería ser también el número 53. Pero, al leer las direcciones, los datos escritos no se corresponden con el número de dirección, se saltan posiciones.

Al no conseguir resolver este problema, se propone pasar de escribir 100 datos consecutivos a un esquema que trabaja con datos de uno en uno, obteniendo el controlador de la Figura 3.8 simulado en la Figura 3.9.

El problema que presenta esta solución es la incapacidad de escribir un banco entero de sonidos, guardados en el ordenador, en la memoria DDR. Lo que implica que la memoria estará sin escribir. Por tanto, este aspecto se intenta resolver mediante otro controlador que por falta de tiempo no se ha podido implementar y queda pendiente para trabajo futuro.

#### 4.4. PRUEBAS DEL CONTROLADOR DE REPRODUCCIÓN

Para probar el funcionamiento de este controlador se han ido simulando diferentes pulsos de entrada para comprobar la correcta actualización de las señales en diferentes situaciones. El Test-Bench diseñado se muestra en la Figura 4.3. Se estudian las siguientes situaciones que se pueden dar en el circuito:

1. En la llegada de los primeros cuatro pulsos, éstos se van asignando a los diferentes módulos correctamente.
2. A la llegada de los dos siguientes se desalojan correctamente los módulos que llevaban más tiempo en uso.
3. Finalmente, se comprueba la correcta actualización de la señal *orden* en la situación en la que los módulos vayan finalizando sus lecturas de memoria y, por tanto, su uso.

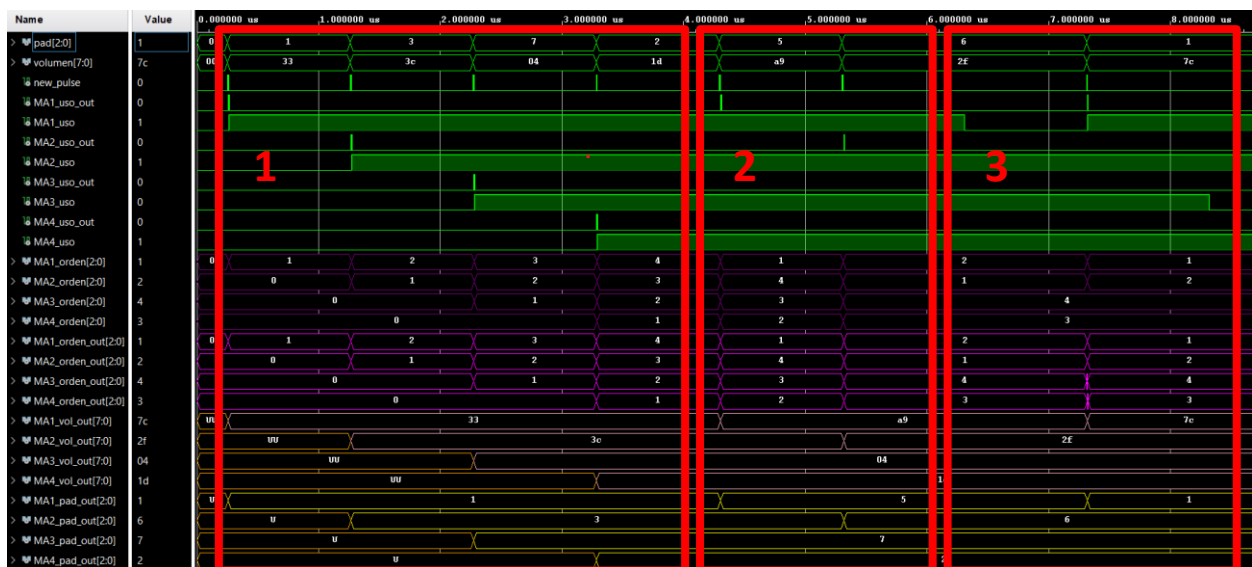


Figura 4.3: Test-bench para comprobar el funcionamiento del Controlador de reproducción

## 4.5.PRUEBAS DEL CONTROLADOR DE ACCESOS A MEMORIA

En esta prueba podemos comprobar que los módulos que solicitan los datos a la memoria funcionan como es debido. En la siguiente simulación vemos cómo se leen cuatro direcciones de memoria. En el momento que la dirección actual, *dir\_act\_out*, coincide con la dirección final, *dir\_final*, y ha sido leída la señal de *uso* pasa a tener valor '0', por lo que el funcionamiento es el esperado.

Cabe destacar que *dir\_act\_out* tiene valor 'Z' cuando el enable no está activo, pues en este bus de datos estarán escribiendo cuatro módulos, de esta forma evitamos que más de un módulo escriba datos en el bus.

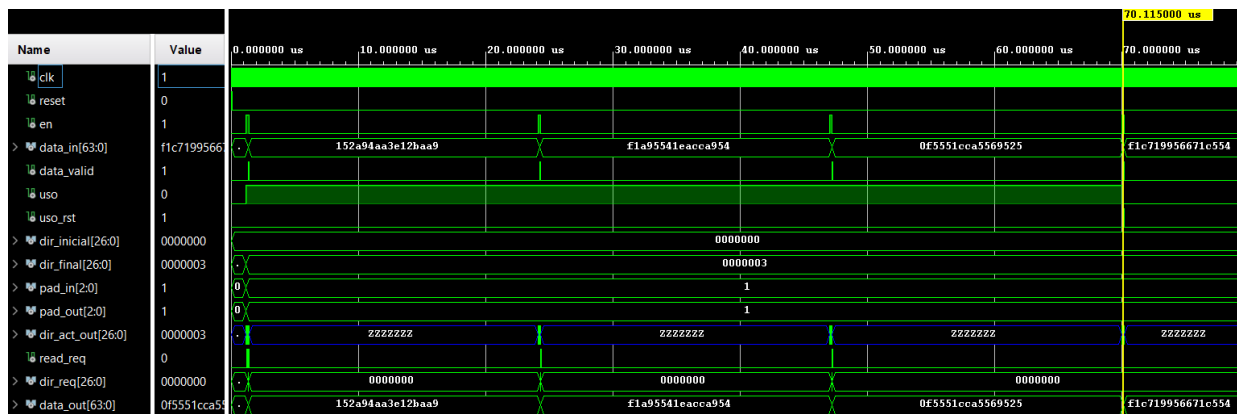


Figura 4.4: Test-bench para comprobar el funcionamiento del Controlador de accesos a memoria

## 4.6.PRUEBAS DEL REGISTRO DE DATOS

En esta prueba entran tres solicitudes de datos de diferentes pads, lo que significa la reproducción de tres sonidos diferentes. Se observa cómo las direcciones inicial y final cambian correctamente. El *uso* y *orden* del módulo tienen el comportamiento esperado. Los datos se actualizan en el momento necesario y los datos se quedan guardados de forma correcta.

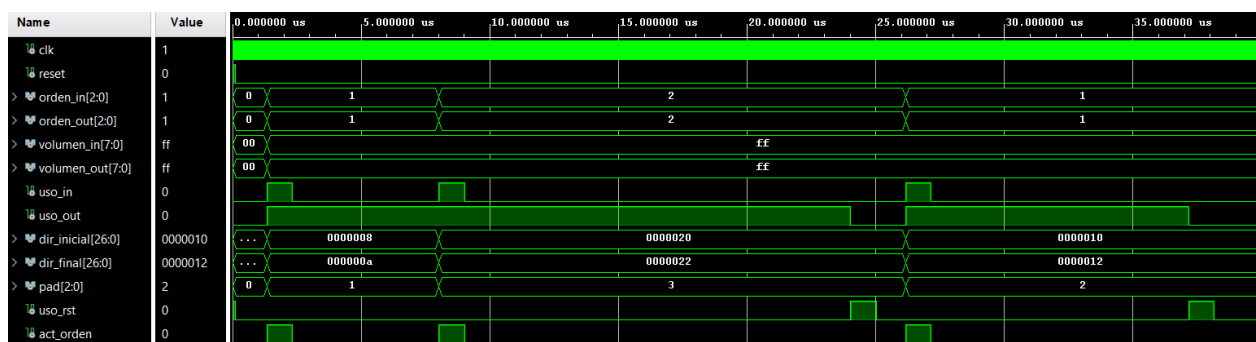


Figura 4.5: Test-bench para comprobar el funcionamiento del Registro de datos



## 4.7.PRUEBAS DEL GENERADOR DE ENABLES

En esta prueba comprobamos que las señales de enable de los controladores de accesos a memoria duran 20 ciclos y se activan correctamente.

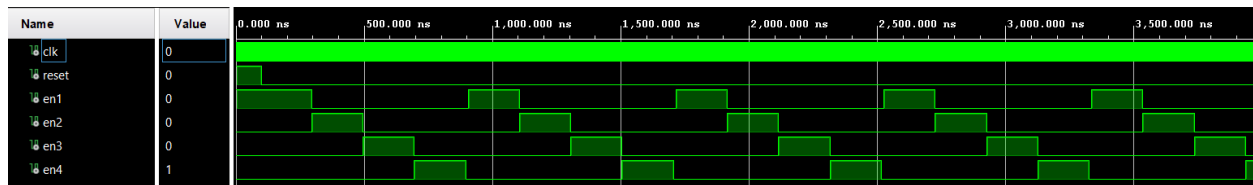


Figura 4.6: Test-bench para comprobar el funcionamiento del Generador de enables

En la figura se observa una correcta activación de los enables, lo que permitirá una correcta multiplexación en el tiempo de los controladores de accesos a memoria.

## 5. RESULTADOS

La siguiente tabla muestra un resumen del trabajo realizado con los distintos componentes que forman el sistema completo y el grado de consecución obtenido en cada uno de ellos:

Bloque	Descripción VHDL	Verificación SPICE	Verificación VHDL	Test	Prototipado
<b>Pad + Sensor Piezo Eléctrico</b>		✓		✓	✓
<b>Acondicionador de Señales</b>		✓		✓	✓
<b>Conversor ADC</b>	✓		✓	✓	✓
<b>Detector de picos</b>	✓		✓	✓	
<b>Interfaz Artix7 – DDR2</b>	✓		✓		
<b>Controlador de reproducción</b>	✓		✓		
<b>Controlador accesos a memoria</b>	✓		✓		
<b>Registro de datos</b>	✓		✓		
<b>Generador de enables</b>	✓		✓		
<b>Módulo PWM</b>	✓		✓	✓	
<b>Protocolo PC-Placa</b>					
<b>Sistema Completo</b>					

Como se observa, la parte de adquisición de señal está concluida hasta el nivel de prototipo. La parte de procesamiento digital ha estado completamente condicionada por las comunicaciones con la memoria DDR2 de la placa, tanto desde el controlador desarrollado como desde el PC para cargar los bancos de sonido. El desarrollo de un protocolo de comunicaciones entre la placa y el PC suponía un esfuerzo en tiempo fuera de lo esperable en el presente TFG. Por lo tanto, no se ha podido llegar al nivel de test en placa para la mayoría de los módulos digitales que dependen de los datos almacenados en memoria. Por último, la parte de generación de señal, correspondiente al módulo PWM, sí se ha testado en placa.

Respecto a la caracterización del sistema completo se aportan los siguientes datos:

- **Características físicas de los pads.** Las dimensiones del pad impreso en 3D son 13 cm de ancho, 13 cm de largo y 5cm de alto, las paredes son de 1 cm de anchura. El material con el que se ha impreso ha sido filamento PLA.
- **Alimentación de la parte analógica.** El bloque analógico del sistema (Pad y acondicionador de señal) es alimentado con la propia placa Nexys A7 entre 0 y 3.3 V.
- **Frecuencia de muestreo de los golpes.** El circuito acondicionador de señal incorpora un filtro paso bajo con frecuencia de corte entorno a los 16 Hz como se comenta anteriormente. Es decir, dos golpes seguidos al pad separados por menos de 62.5 ms se detectarían como un solo golpe, pero esta velocidad para tocar esta fuera del alcance del percusionista medio por tanto no supone ningún problema.
- **Cuantificación ADC.** Las muestras de las señales analógicas son de 16 bits.
- **Frecuencia parte digital.** La parte digital del sistema funciona a 100 MHz. Por otra parte, la memoria DDR2 requiere de una frecuencia de reloj de 200 MHz para funcionar.
- **Cuantificación y frecuencia de muestreo de las muestras digitales.** Las muestras del banco de sonidos son de 16 bits y han sido muestreadas a 44 kHz. Tienen formato .wav mono. A pesar de no haber sido guardadas en la memoria, se han tenido en cuenta estos valores para los cálculos generales.
- **Latencia del sistema digital.** La latencia máxima que experimenta el sistema digital es de aproximadamente 30  $\mu$ s. Para hacer este cálculo se han considerado el máximo número de estados que tiene que atravesar la señal hasta llegar al módulo PWM y, además, el peor caso en el que el pulso llega justo cuando se acaba de pasar un dato nuevo para reproducción. En cualquier caso se puede considerar un retardo extremadamente pequeño para la tarea que se lleva a cabo. Se puede concluir que el sistema es apto para ser usado en directo.

Los resultados de la utilización de recursos hardware en la FPGA Artix7 necesarios para la implementación son los que se muestran en la siguiente tabla:

Recurso	Cantidad	Disponible	Utilización (%)
LUT	800	63400	1.26
FF	262	126800	0.21
DSP	8	240	3.33
IO	28	210	13.33
BUFG	2	32	6.25

Como se observa en la tabla se utiliza una pequeña cantidad de los recursos disponibles dentro de la FPGA. Esto abre la posibilidad de interesantes continuaciones y mejoras futuras una vez se tenga el prototipo funcionando.

## 6. CONCLUSIONES Y LÍNEAS FUTURAS

### 6.1. CONCLUSIONES

La realización de este proyecto ha permitido afianzar los conocimientos impartidos a lo largo del Grado en los temas relacionados con el tratamiento de audio, descripción de circuitos en lenguaje VHDL, comprensión del funcionamiento de una FPGA, tratamiento de señales analógicas y la comprensión del funcionamiento de memorias.

Se han logrado muchos de los objetivos que estaban previstos al inicio del proyecto. A pesar de no haber logrado cubrir todos los objetivos iniciales se han conseguido numerosos resultados:

- Se ha diseñado y construido un pad de percusión funcional.
- Se ha diseñado e implementado la circuitería analógica necesaria para acondicionar la señal del sensor piezoeléctrico.
- Se ha logrado una correcta conversión ADC de los pulsos entrantes de los sensores a la FPGA.
- Se ha desarrollado un sistema de procesamiento de audio digital capaz de manejar hasta cuatro entradas analógicas.
- Se ha implementado una interfaz de lectura de datos de una memoria DDR2.

Cabe destacar que las mayores dificultades encontradas han sido:

- **El estudio, diseño e implementación de la circuitería analógica de acondicionamiento de señal.** Ha supuesto una gran inversión de tiempo debido a que se ha necesitado un periodo de familiarización con los materiales del laboratorio y la falta de experiencia a la hora de solventar errores de los circuitos analógicos.
- **El estudio del funcionamiento de los protocolos de la memoria DDR junto con el diseño e implementación de un controlador para una correcta lectura y escritura de datos.** Esta dificultad junto con la falta de tiempo ha provocado el recorte de objetivos entorno a este apartado. Es decir, aunque el sistema funcione la memoria estará vacía como se ha explicado anteriormente.

A pesar de las dificultades encontradas en este Trabajo de Fin de Grado, haber sido capaz de alcanzar una primera versión del prototipo ha sido muy satisfactorio a nivel personal ya que me ha servido para aumentar mis conocimientos, mi capacidad de investigación y mi autosuficiencia.

## 6.2.LÍNEAS FUTURAS

El presente proyecto tiene la posibilidad de realizar mucho trabajo futuro. En concreto se van a contemplar las siguientes líneas de trabajo:

- **Mejora del banco de sonidos.** Como se ha explicado la interfaz con la memoria DDR2 ha consumido gran parte del tiempo disponible. Por lo tanto, este es un punto clave para la continuación del proyecto. Existe la posibilidad de terminar el desarrollo del controlador para que sea capaz de escribir correctamente los datos en la memoria. Otra alternativa podría ser optar por una opción diferente en lugar de utilizar la memoria DDR2 como por ejemplo usar una tarjeta SD.
- **Mejora del esquema de ADC.** En este momento la conversión está limitada por los cuatro conversores ADC incorporados en la placa Nexys A7. Se propone la utilización de un esquema de multiplexación en el tiempo analógica dando paso a que haya más de cuatro pads funcionando, pudiendo tener más sonidos y acercarse más al funcionamiento de una batería acústica real.
- **Terminar y mejorar la construcción de más pads.** Actualmente solo se ha construido un pad funcional junto con su circuito de acondicionamiento en una placa de inserción. Construir más pads junto con el diseño de una placa PCB para todos los pads que se quieran desarrollar proporciona un diseño más compacto y elegante.
- **Mejora del controlador de audio.** El controlador de audio está limitado a la reproducción de hasta cuatro sonidos al mismo tiempo, lo que puede provocar una peor experiencia, musicalmente hablando, para el usuario. Es decir, mientras se toque el instrumento podrían interrumpirse reproducciones de sonidos previos. Por lo que sería interesante mejorar este aspecto del controlador para que sea capaz de reproducir más sonidos a la vez, suficientes para evitar la situación anterior.
- **Cambiar el módulo PWM.** Como se explica en el apartado de la generación de la señal de salida la conversión DAC mediante un módulo PWM no es la que mejor encaja con el sistema, pues las muestras de 16 bits tienen que ser recortadas a 11 bits para que el módulo funcione correctamente. Por tanto, hay que estudiar otras opciones más adecuadas para realizar esta conversión.
- **Añadir efectos musicales al sistema.** Realizar esta mejora añadiendo un surtido de efectos hará que el prototipo sea mucho más completo e interesante para hacer música con él.

## 7. BIBLIOGRAFÍA

- [1] Contributors to Wikimedia projects. "Electronic drum". Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Electronic\\_drum](https://en.wikipedia.org/wiki/Electronic_drum) (accedido el 26 de octubre de 2020).
- [2] Musicalecer. Creación, edición y producción musical | Cómo hacer música desde el principio... <https://musicalecer.files.wordpress.com/2020/12/t4-8-4-roland-spd-one.png> (accedido el 10 de mayo de 2021).
- [3] Digilent, "Nexys4 DDR™ FPGA board reference manual", abril de 2016.
- [4] Electronic Circuit Optimization & Simulation | Cadence PSpice | PSpice. <https://www.pspice.com/sites/default/files/pspice-for-ti-logo.png> (accedido el 11 de mayo de 2021).
- [5] Farnell. [https://es.farnell.com/productimages/large/en\\_GB/2780319-40.jpg](https://es.farnell.com/productimages/large/en_GB/2780319-40.jpg). (accedido el 11 de mayo de 2021).
- [6] THE PRODUCT KEYS. <https://theproductkeys.com/wp-content/uploads/2019/07/Matlab-R2019b-Crack-Activation-Key-2020-Torrent.png>. (accedido el 11 de mayo de 2021).
- [7] Colaboradores de los proyectos Wikimedia. "Batería electrónica". Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Batería\\_electrónica](https://es.wikipedia.org/wiki/Batería_electrónica) (accedido el 26 de octubre de 2020).
- [8] "Piezoelectric Sensor Market By Type (Piezoelectric Force Sensors, Piezoelectric Accelerometers, Piezoelectric Pressure Sensor), By Application (Industrial & Manufacturing, Automotive, Healthcare, Aerospace), Industry Trends, Estimation and Forecast, 2017-2025 | Market Research Reports & Consulting Services | ERC", Market Research Reports & Consulting Services | ERC. <https://www.esticastresearch.com/report/piezoelectric-sensor-market/> (accedido el 13 de mayo de 2021).
- [9] J. Dempsey. "How do electronic drums work?" Guitars, Pianos & Musical Instruments | Dawsons Music & Sound. <https://www.dawsons.co.uk/blog/how-do-electronic-drums-work> (accedido el 27 de octubre de 2020).
- [10] "How do electronic drum pads work?" thenationalparksmusic.com. <https://thenationalparksmusic.com/how-do-electronic-drum-pads-work/> (accedido el 27 de octubre de 2020).
- [11] Colaboradores de los proyectos Wikimedia. "Modulación por impulsos codificados". Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Modulación\\_por\\_impulsos\\_codificados](https://es.wikipedia.org/wiki/Modulación_por_impulsos_codificados) (accedido el 19 de octubre de 2020).
- [12] Auvisa.com. [https://auvisa.com/14602-thickbox\\_default/remo-rt08-pad-practica-caja-sorda-bateria-8.jpg](https://auvisa.com/14602-thickbox_default/remo-rt08-pad-practica-caja-sorda-bateria-8.jpg). (accedido el 13 de mayo de 2021).
- [13] Contributors to Wikimedia projects. "Practice pad". Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Practice\\_pad](https://en.wikipedia.org/wiki/Practice_pad) (accedido el 20 de noviembre de 2020).
- [14] Contributors to Wikimedia projects. "BoPET". Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/BoPET> (accedido el 20 de noviembre de 2020).
- [15] Contributors to Wikimedia projects. "Elastomer". Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/Elastomer> (accedido el 20 de noviembre de 2020).

- [16] "FOAM: Propiedades, usos y aplicaciones". RA-PACK. <https://www.ra-pack.com/blog/noticias/foam-propiedades-usos-y-aplicaciones> (accedido el 20 de noviembre de 2020).
- [17] Contributors to Wikimedia projects. "Piezoelectricity". Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/Piezoelectricity#Materials> (accedido el 17 de noviembre de 2020).
- [18] "Piezoelectric transducer: Types, advantages, diagram | linquip". Industrial Manufacturing Blog | linquip. <https://www.linquip.com/blog/piezoelectric-transducer/> (accedido el 17 de noviembre de 2020).
- [19] Contributors to Wikimedia projects. "Potassium sodium tartrate". Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Potassium\\_sodium\\_tartrate](https://en.wikipedia.org/wiki/Potassium_sodium_tartrate) (accedido el 20 de noviembre de 2020).
- [20] L. Leija, Métodos de procesamiento avanzado e inteligencia artificial en sistemas sensores y biosensores. Barcelona: Reverte, 2009.
- [21] B. Jiménez, "Materiales piezoeléctricos: Formas de presentación, ventajas y desventajas en las aplicaciones", Boletín de la Sociedad Española de Cerámica y Vidrio, vol. 34, diciembre de 1995.
- [22] MURATA, "Piezoelectric sound components", P37E, diciembre de 2015.
- [23] "Timesetl 15pcs Accessories Pickup Bass Accessories 27mm Discos Piezoelectricos con Conductores Microfono Tambo Noiseless Pickups Dimarzio Humbucker Pickups Latón: Amazon.es: Instrumentos musicales". Amazon.es. [https://www.amazon.es/TIMESETL-Accessories-Piezoelectricos-Conductores-Microfono/dp/B077YDMF1T/ref=pd\\_vtp\\_21\\_1/260-9627142-0498805?\\_encoding=UTF8&pd\\_rd\\_i=B077YDMF1T&pd\\_rd\\_r=1ade1179-e71f-4b64-b85e-4c793e4cee10&pd\\_rd\\_w=jS2AN&pd\\_rd\\_wg=L2Oux&pf\\_rd\\_p=4a3fef14-56e7-4d23-9b3f-f6dee5d5bb8e&pf\\_rd\\_r=VPH4GSZK4T386QS1DS50&psc=1&refRID=VPH4GSZK4T386QS1DS50](https://www.amazon.es/TIMESETL-Accessories-Piezoelectricos-Conductores-Microfono/dp/B077YDMF1T/ref=pd_vtp_21_1/260-9627142-0498805?_encoding=UTF8&pd_rd_i=B077YDMF1T&pd_rd_r=1ade1179-e71f-4b64-b85e-4c793e4cee10&pd_rd_w=jS2AN&pd_rd_wg=L2Oux&pf_rd_p=4a3fef14-56e7-4d23-9b3f-f6dee5d5bb8e&pf_rd_r=VPH4GSZK4T386QS1DS50&psc=1&refRID=VPH4GSZK4T386QS1DS50)
- [24] "¿Qué es el acondicionamiento de señal o acondicionador de señal? | Dewesoft". Data Acquisition Systems (DAQ) and Solutions | Dewesoft. <https://dewesoft.com/es/daq/que-es-acondicionamiento-de-seal> (accedido el 7 de diciembre de 2020).
- [25] "¿Qué es un acondicionador de señal? | Funciones". HBM. <https://www.hbm.com/es/7339/que-es-un-acondicionador-de-senal-funciones/> (accedido el 7 de diciembre de 2020).
- [26] Xilinx, "XADC wizard v3.3", Vivado Design Suite, PG091, LogiCORE IP Product Guide, septiembre de 2016.
- [27] "Digilent/Nexys-A7-100T-XADC". GitHub. <https://github.com/Digilent/Nexys-A7-100T-XADC> (accedido el 17 de marzo de 2021).
- [28] Xilinx, "Zynq-7000 soc and 7 series devices memory interface solutions v4.2", User Guide, UG586, diciembre de 2018.
- [29] Xilinx, "7 series fpgas memory interface solutions", User Guide, UG586, enero de 2012.

## ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

### A.1 INTRODUCCIÓN

Este proyecto tiene como finalidad el interés personal ya que en los últimos años me han llamado la atención los instrumentos electrónicos, en particular las baterías musicales electrónicas. La idea surge a raíz de cursar la asignatura de Ingeniería de la música en la cual los instrumentos electrónicos están cubiertos en el temario.

### A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO

Uno de los puntos fuertes de este proyecto es la versatilidad que puede ofrecer a la hora de personalizar el kit. Actualmente no ofrece ninguna ventaja respecto a otros fabricantes respecto a personalizar los pads, en general todos los fabricantes ofrecen grandes posibilidades de personalización de los sonidos a reproducir. Sin embargo, si se plantea la opción de añadir efectos sí que ofrece una gran ventaja. El particular podría personalizar y solicitar los efectos que necesite y cabe la posibilidad de que estos sean cambiados en un futuro, mientras que los modelos actuales de mercado generalmente ofrecen un grupo cerrado de efectos.

### A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS

El **impacto económico** es el impacto más importante. La realización de un sistema de estas características supone muchas horas de trabajo y especialmente de programación. La programación con la que se desarrolló el prototipo requiere un nivel medio en el uso de VHDL y por tanto puede presentar una dificultad adicional.

El **impacto social** es de poca importancia. El prototipo diseñado se plantea para estar destinado al mundo de la música. Este mundo suele ser parte de las actividades de ocio de muchas personas. El impacto que puede llegar a tener es la inquietud que puede provocar en algunos melómanos.

Respecto al **impacto ético** del proyecto, el prototipo puede ser utilizado como instrumento principal o auxiliar para percusionistas y músicos. Podría ser útil en la iniciación de aquellas personas que quieran aprender percusión o incluso podría ser útil en el ámbito de la producción musical.

El **impacto ambiental** es destacable. Gracias al uso de la FPGA se pueden reutilizar recursos hardware. Por lo que si se quieren cambiar características del prototipo se podrá reutilizar la plataforma.



## A.4 CONCLUSIONES

Dado que el prototipo diseñado está pensado para uso personal no tiene gran impacto en la sociedad ni en la ética. Desde el punto de vista económico, el costo de realización de puede considerarse elevado. A pesar de este coste, dado que es un prototipo creado por motivos personales y que no pretende ser un producto comercializable, no supone un problema para la viabilidad del proyecto. Por último, entre el conjunto de la electrónica disponible, la elección de una FPGA hace que el proyecto asuma un aspecto ambiental positivo ya que con la misma plataforma se pueden realizar varias versiones sin tener que cambiar de electrónica.

## ANEXO B: PRESUPUESTO ECONÓMICO

### COSTE DE MANO DE OBRA (coste directo)

horas	Precio/hora	TOTAL
420	25 €	<b>10,500 €</b>

### COSTE DE RECURSOS MATERIALES (coste directo)

	Precio de compra	Uso en meses	Amortiz. en años	TOTAL
Ordenador personal (Software incluido)	950.00 €	9	5	142.50 €
Impresora 3D	199.00 €	2	5	6.63 €
Nexys A7	238.63 €	9	5	35.79 €
Osciloscopio	301.00 €	6	5	30.10 €
Generador de señales	296.95 €	6	5	29.70 €
Otros (Filamento PLA, Componentes electrónicos...)	50.00 €	9	5	7.50 €

<b>TOTAL</b>	<b>252.22 €</b>
--------------	-----------------

<b>GASTOS GENERALES (costes indirectos)</b>	15%	sobre CD	<b>1,612.83 €</b>
<b>BENEFICIO INDUSTRIAL</b>	6%	sobre CD+CI	<b>741.90 €</b>

SUBTOTAL PRESUPUESTO		13,106.96 €
IVA APLICABLE	21%	2,752.46 €

<b>TOTAL PRESUPUESTO</b>	<b>15,859.42 €</b>
--------------------------	--------------------