

# IMPLEMENTACIÓN DE UN SERVIDOR WEB PARA GESTIONAR UN CORE TIME-TO-DIGITAL-CONVERTER MULTICANAL EMPLEADO EN SISTEMAS MÉDICOS PET

*Adrián Suárez Zapata  
José Torres País  
Pedro A. Martínez Delgado  
Raimundo García Olcina  
Julio Martos Torres  
Jesús Soret Medel*



**IMPLEMENTACIÓN DE UN  
SERVIDOR WEB PARA GESTIONAR  
UN CORE *TIME-TO-DIGITAL-  
CONVERTER* MULTICANAL  
EMPLEADO EN SISTEMAS  
MÉDICOS PET**

*Adrián Suárez Zapata  
José Torres País  
Pedro A. Martínez Delgado  
Raimundo García Olcina  
Julio Martos Torres  
Jesús Soret Medel*



**Editorial Área de Innovación y Desarrollo,S.L.**

Quedan todos los derechos reservados. Esta publicación no puede ser reproducida, distribuida, comunicada públicamente o utilizada, total o parcialmente, sin previa autorización.

© del texto: **los autores**

ÁREA DE INNOVACIÓN Y DESARROLLO, S.L.

C/ Els Alzamora, 17 - 03802 - ALCOY (ALICANTE) [info@3ciencias.com](mailto:info@3ciencias.com)

Primera edición: **agosto 2018**

ISBN: **978-84-949151-3-0**

DOI: <http://dx.doi.org/10.17993/IngyTec.2018.44>



## RESUMEN

Esta contribución describe el desarrollo e implementación de un sistema de alta resolución basado en un Core Time-to-Digital Converter (TDC) para la mejora de imágenes médicas. Dicho sistema es capaz de obtener diferencias temporales para múltiples canales simultáneos, con el objetivo de mejorar una de las tecnologías que más ha prosperado en los últimos años en el ámbito de la medicina nuclear. Esta área médica ha avanzado significativamente en los últimos años debido al aporte que la tecnología ha proporcionado en términos de mejora de resolución en los sistemas empleados. Una de las técnicas que más ha avanzado en este ámbito ha sido la Tomografía por Emisión de Positrones (PET), basado en una técnica médica mediante la cual se obtienen imágenes de la distribución espacial y temporal de los procesos metabólicos que se generan en el interior del organismo.

El TDC que se presenta es capaz de obtener una resolución inferior a 100 ps, que es la necesaria para obtener una mejora apreciable en la calidad de imagen de los PET actuales. Para ello, se ha empleado la estructura interna de las últimas familias de los dispositivos electrónicos Field-Programmable Gate Array (FPGA) basadas en lógica de acarreo y un estricto proceso de calibrado. Asimismo, se demuestra la idoneidad del uso del TDC en sistemas médicos PET, ofreciendo la posibilidad de determinar el Time of Flight (TOF). Una técnica que incorporan pocos PET en el mercado debido a su complejidad, y que conlleva múltiples ventajas como el aumento de la precisión en la detección de neoplasias, mejor reconstrucción de la imagen y la reducción del tiempo de exposición del paciente a los radiofármacos que son introducidos en su cuerpo.

En atención a la problemática expuesta, el sistema a implementar será capaz de detectar eventos con una resolución temporal muy inferior a la frecuencia del reloj del sistema, realizar la gestión de los datos obtenidos mediante un microprocesador software embebido en la FPGA y de proveer una representación gráfica de los mismos. Esta representación gráfica podrá realizarse a nivel local, empleando una comunicación serie de baja tasa de datos, y a nivel remoto a través de una aplicación web que se comunique con el sistema mediante una comunicación de alta velocidad de gestión remota.



## ABSTRACT

*This contribution describes the development and implementation of a high resolution system based on a Time-to-Digital Converter (TDC) core for the improvement of medical images. This system is able to obtain temporary differences for multiple simultaneous channels, with the aim of improving one of the technologies that has most prospered in recent years in the field of nuclear medicine. This medical area has advanced significantly in recent years due to the contribution that technology has provided in terms of improving resolution in the systems used. One of the techniques that has advanced the most in this field has been Positron Emission Tomography (PET), based on a medical technique through images of the spatial and temporal distribution of metabolic processes that are generated inside the organism are obtained.*

*The TDC that is integrated is able to obtain a resolution lower than 100 ps, which is the necessary to obtain an appreciable improvement in the image quality of the current PET. For this purpose, the internal structure of the last families of the electronic Field-Programmable Gate Array (FPGA) devices based on carry logic and a strict calibration process has been used. Likewise, the suitability of the use of the TDC in PET medical systems is demonstrated, offering the possibility of determining the Time of Flight (TOF). A technique that incorporates few PETs in the market due to its complexity, and that brings multiple advantages such as increased accuracy in the detection of neoplasms, better reconstruction of the image and reduction of the patient's exposure time to radiopharmaceuticals that are introduced in the body.*

*In attention to the exposed problems, the system to be implemented will be able to detect events with a temporal resolution much lower than the frequency of the system clock, perform the management of the data obtained through a software microprocessor embedded in the FPGA and provide a representation graphic of them. This graphic representation can be done at the local level, using a low-data rate serial communication, and remotely through a web application that communicates with the system through high-speed communication an allow to manage it remotly.*



# ÍNDICE

<b>CAPÍTULO 1: INTRODUCCIÓN.....</b>	<b>13</b>
1.1. Planteamiento de la investigación .....	13
1.2. Objetivos.....	13
<b>CAPÍTULO 2: DESCRIPCIÓN GENERAL DEL SISTEMA PET .....</b>	<b>15</b>
2.1. Tomografía por Emisión de Positrones .....	15
2.1.1. Revisión histórica del PET .....	15
2.1.2. Principio de funcionamiento del PET .....	16
2.1.3. Principales aplicaciones .....	17
2.2. Arquitectura general de un sistema PET .....	17
2.2.1. Anillo de detectores.....	18
2.2.2. Tarjeta de trigger.....	18
2.2.3. Módulo de adquisición .....	19
2.2.4. Procesado y reconstrucción de la imagen.....	19
2.3. Sistemas comerciales PET.....	20
2.3.1. Sistemas PET de Philips .....	20
2.3.2. Sistemas PET de Siemens .....	21
2.4. Sumario.....	22
<b>CAPÍTULO 3: ESTADO DEL ARTE.....</b>	<b>25</b>
3.1. Tiempo de vuelo (TOF) .....	25
3.2. Time-to-digital converter (TDC) .....	28
3.2.1. Primera generación: TDC analógico .....	28
3.2.2. Segunda generación: TDC digital .....	29
3.2.3. Tercera generación: TDC basado en línea de retardo .....	30
3.2.4. Tercera generación: TDC basado en línea de retardo .....	31
2.3. Sumario.....	32
<b>CAPÍTULO 4: DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA .....</b>	<b>35</b>
4.1. Estudio de tecnologías .....	35
4.2. Descripción de la tecnología empleada: FPGA .....	36
4.2.1. Estructura de un dispositivo FPGA .....	36
4.2.2. Descripción del dispositivo empleado: FPGA Spartan 6 .....	36
4.3. IP Cores .....	38
4.4. MicroBlaze .....	38
4.4.1 Arquitectura de MicroBlaze .....	39
4.4.2. Buses y comunicaciones .....	39
4.5. Herramientas para la programación de la FPGA.....	40
4.5.1. Metodología de diseño y programación .....	40
4.5.2. Xilinx ISE Design Suite .....	41
4.6. Sumario.....	44
<b>CAPÍTULO 5: IMPLEMENTACIÓN DEL CORE TDC .....</b>	<b>47</b>
5.1. Arquitectura del sistema TDC.....	47
5.1.1. Core TDC.....	47
5.1.2. Core TDC.....	49
5.1.3. Core TDC.....	50
5.2. Interconexión del Core TDC con MicroBlaze.....	50
5.3. Principio de funcionamiento y elementos del TDC.....	51



5.3.1 Core TDC.....	53
5.3.2 Línea de retardo .....	53
5.4. Sumario.....	56
<b>CAPÍTULO 6: DISEÑO FIRMWARE .....</b>	<b>57</b>
6.1. Gestión de datos del Core TDC.....	57
6.2. Implementación del servidor web .....	61
6.2.1. Línea de retardo .....	66
6.2.2. Programación del servidor web .....	68
6.2.3. Comunicación con el navegador web .....	73
<b>CAPÍTULO 7: DISEÑO SOFTWARE .....</b>	<b>75</b>
7.1. Aplicación software para PC.....	75
7.1.1. Entorno y lenguaje de programación .....	75
7.1.2. Estudio del funcionamiento de la aplicación .....	77
7.2. Programación del sitio web.....	83
7.2.1. Entorno y lenguaje de programación .....	83
7.2.2. Estudio del funcionamiento de la aplicación .....	87
7.3. Sumario.....	89
<b>CAPÍTULO 8: PRUEBAS Y RESULTADOS .....</b>	<b>91</b>
8.1. Validación de la resolución del TDC gestionado por MicroBlaze.....	91
8.2. Estudio de la resolución del TDC con múltiples canales.....	93
8.3. Análisis del funcionamiento del TDC en un sistema médico PET .....	96
<b>CAPÍTULO 9: CONCLUSIONES .....</b>	<b>101</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>103</b>

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Comparativa entre FPGA y ASIC. ....	35
<b>Tabla 2.</b> Comparación de las diferentes variantes de Ethernet. ....	67

## ÍNDICE DE FÓRMULAS

<b>Fórmula 1.</b> Tiempo de retardo de una célula. ....	31
<b>Fórmula 2.</b> Diferencia de tiempo de retardo entre líneas. ....	31

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Comparativa de resultados de resolución entre técnica SPECT y PET. ....	15
<b>Figura 2.</b> Representación de la generación y aniquilación de un positrón. ....	16
<b>Figura 3.</b> Arquitectura global de un sistema médico PET. ....	17
<b>Figura 4.</b> Tipos de eventos de coincidencia para PET. ....	18
<b>Figura 5.</b> Características de diseño de diversos equipos PET.....	19
<b>Figura 6.</b> Sistema PET comercial GEMINI LXL de Philips. ....	20
<b>Figura 7.</b> Sistema PET comercial TruePoint PET-CT de Siemens. ....	21
<b>Figura 8.</b> Tipos de detección de aniquilaciones en un sistema PET en anillo. ....	25
<b>Figura 9.</b> Tipos de detección de aniquilaciones en un sistema PET en anillo. ....	26
<b>Figura 10.</b> Comparativa de resultados entre PET convencional (inf.) y PET con TOF (sup.) respecto al número de detectores. ....	26
<b>Figura 11.</b> Arquitectura de un circuito TDC analógico. ....	28
<b>Figura 12.</b> Línea de retardo compuesta por puertas lógicas y biestables. ....	29

<b>Figura 12.</b> Estructura interna de un dispositivo FPGA. ....	35
<b>Figura 13.</b> Kit de desarrollo Atlys de Digilent. ....	36
<b>Figura 14.</b> Diagrama de flujo del codiseño seguido. ....	40
<b>Figura 15.</b> Entorno de ISE Project Navigator. ....	41
<b>Figura 16.</b> Simulación del TDC con la herramienta ISim. ....	41
<b>Figura 17.</b> Entorno de programación del hardware de la FPGA con XPS. ....	42
<b>Figura 18.</b> Estructura interna del hardware implementado en la FPGA mediante PlanAhead. ....	43
<b>Figura 19.</b> Entorno de programación del software SDK. ....	43
<b>Figura 20.</b> Esquema conceptual de la implementación del TDC. ....	46
<b>Figura 21.</b> Arquitectura del Core TDC. ....	48
<b>Figura 22.</b> Comunicación entre MicroBlaze y un periférico o Core de usuario. ....	49
<b>Figura 23.</b> Diagrama de flujo del funcionamiento del Core TDC. ....	51
<b>Figura 24.</b> Histograma de calibración (número de muestras vs número de bin). ....	52
<b>Figura 25.</b> Estructura interna de los CLBs y de la primitiva CARRY4 de la FPGA. ....	53
<b>Figura 26.</b> Esquema de la línea de retardo. ....	53
<b>Figura 27.</b> Esquema interno de una línea de retardo integrada en la FPGA. ....	54
<b>Figura 28.</b> Distribución de los canales integrados en la Spartan-6 empleada. ....	55
<b>Figura 29.</b> Fragmento de código de asignación de registros extraído de user_logic.vhd. ....	56
<b>Figura 30.</b> Esquema de comunicación entre MicroBlaze y Core TDC. ....	57
<b>Figura 31.</b> Opción para generar el driver del Core TDC. ....	58
<b>Figura 32.</b> Funciones de lectura y escritura en los registros del Core TDC. ....	58
<b>Figura 33.</b> Ejemplo de comunicación serie simplificado entre MicroBlaze y PC. ....	60
<b>Figura 34.</b> Configuración de red local en Windows. ....	61
<b>Figura 35.</b> Configuración de la placa de desarrollo Atlys. ....	62
<b>Figura 36.</b> Configuración de preprogramación del Servidor Web. ....	63
<b>Figura 37.</b> Configuración del modo Debug del servidor web. ....	63
<b>Figura 38.</b> Reporte emitido por el servidor web indicando que está funcionando correctamente. ....	64
<b>Figura 39.</b> Sitio web demo de Digilent. ....	64
<b>Figura 40.</b> Configuración del periférico Ethernet Lite. ....	67
<b>Figura 41.</b> Rutina de control de callbacks. ....	68
<b>Figura 42.</b> Rutina de control de recv_callback. ....	68
<b>Figura 43.</b> Rutina de sent_callback(). ....	69
<b>Figura 44.</b> Configuración del tamaño del descriptor y de la velocidad de enlace. ....	70
<b>Figura 45.</b> Modificación de parámetros de opciones de memoria del protocolo lwIP. ....	71
<b>Figura 46.</b> Modificación del parámetro de tamaño de paquete del buffer de acarreo. ....	71
<b>Figura 47.</b> Modificación de los parámetros de la librería xilmsf. ....	72
<b>Figura 48.</b> Escritura y formateo de los datos del Core TDC en el archivo data.xml. ....	72
<b>Figura 49.</b> Formato del archivo data.xml generado. ....	73
<b>Figura 50.</b> Ejemplo de panel frontal de LabVIEW. ....	75
<b>Figura 50.</b> Ejemplo de diagrama de bloques de LabVIEW. ....	75
<b>Figura 51.</b> Diagrama de flujo de la aplicación programada con LabVIEW. ....	78
<b>Figura 52.</b> Entorno de visualización programado en LabVIEW. ....	79
<b>Figura 53.</b> Bloque de representación del vector global de muestras en un histograma. ....	80
<b>Figura 54.</b> Bloque de control del panel frontal. ....	80
<b>Figura 55.</b> Bloque de visualización de los subhistogramas y sus estadísticos. ....	81
<b>Figura 56.</b> Diagrama de bloques de la aplicación para PC programada en LabVIEW. ....	81
<b>Figura 57.</b> Ejemplo de programación de JavaScript mediante etiqueta <script>. ....	83

<b>Figura 58.</b> Ejemplo de programación de JavaScript mediante archivo externo .js. ....	84
<b>Figura 59.</b> Entorno de programación web Dreamweaver. ....	85
<b>Figura 60.</b> Resultado tras generar la imagen MFS. ....	86
<b>Figura 61.</b> Diagrama de flujo del sitio web. ....	87
<b>Figura 62.</b> Sitio web de visualización de medidas temporales del Core TDC. ....	87
<b>Figura 63.</b> Resultados previos del TDC gestionado por MicroBlaze. ....	90
<b>Figura 64.</b> Primeros resultados de resolución temporal del TDC con MicroBlaze. ....	91
<b>Figura 65.</b> Resultados del TDC gestionado por MicroBlaze en la primera versión de la aplicación para PC. ....	92
<b>Figura 66.</b> Esquema de configuración para verificar la resolución real del TDC de 4 canales. ....	93
<b>Figura 67.</b> Interconexión de la señal de entrada con la primera línea de retardo. ....	93
<b>Figura 68.</b> Histograma de diferencias temporales entre canal 1 y 4 con Origin. ....	94
<b>Figura 69.</b> Visualización de las diferencias temporales entre los canales 1 y del 2 al 4 en el entorno de LabVIEW. ....	94
<b>Figura 70.</b> Sistema de test del Core TDC. ....	95
<b>Figura 71.</b> Sistema PET comercial "MAMMI breast PET".....	96
<b>Figura 72.</b> Arquitectura del anillo empleado para realizar pruebas en un sistema PET real. ....	96
<b>Figura 73.</b> Esquema de configuración de la prueba con el anillo del sistema PET. ....	97
<b>Figura 74.</b> Anillo de sistema PET empleado para las pruebas reales. ....	98
<b>Figura 75.</b> Pruebas con un anillo de sistema PET real. ....	98

# CAPÍTULO 1: INTRODUCCIÓN

El presente trabajo de investigación nace por el creciente interés comercial que existe en el mercado de los sistemas médicos de detección de enfermedades debido a que en la última década han avanzado notablemente en términos de precisión, permitiendo a los facultativos realizar cada vez diagnósticos más precoces y más fiables.

Esta evolución se debe en gran medida, al evidente desarrollo tecnológico que ha existido en este ámbito y a las técnicas de procesado y reconstrucción de imagen. Habitualmente para el desarrollo de este tipo de dispositivos en sistemas PET se emplean Circuitos Integrados para Aplicaciones Específicas (ASIC, del inglés Application-Specific Integrated Circuit) debido a su optimización. No obstante, estos dispositivos presentan un coste muy elevado y no son reconfigurables, por lo que con este trabajo se pretende ofrecer una alternativa de bajo coste y reconfigurable que posea características similares a las que ofrece un ASIC.

## 1.1. Planteamiento de la investigación

La investigación realizada se centra en la implementación de un sistema electrónico cuya finalidad es la medición de eventos con una resolución inferior a 100 pico segundos (ps) de modo que pueda aplicarse en sistemas médicos de detección de anomalías mediante la evaluación del paciente y la reconstrucción de imágenes que muestran el interior de su organismo tal y como realizan los sistemas basados en Tomografía por Emisión de Positrones (PET, del inglés Positron Emission Tomography). La posibilidad de poder obtener una resolución temporal del orden de pico segundos permite, entre otras mejoras, realizar una reconstrucción de la imagen del interior del organismo mucho más precisa y de mayor calidad, por lo que los facultativos pueden diagnosticar la existencia de anomalías de forma más prematura.

Para tal efecto, el sistema a implementar será capaz de detectar eventos con una resolución temporal muy inferior a la frecuencia del reloj del sistema, realizar la gestión de los datos obtenidos y de proveer una representación gráfica de los mismos. Esta representación gráfica podrá realizarse a nivel local, empleando una comunicación serie de baja tasa de datos, y a nivel remoto a través de una aplicación web que se comuniquen con el sistema mediante una comunicación de alta velocidad.

## 1.2. Objetivos

El objetivo principal del trabajo es implementar y desarrollar un sistema de etiquetado temporal de alta resolución. Este sistema electrónico deberá ser completamente reconfigurable, deberá comunicarse con otros sistemas empleando estándares de comunicación y podrá ser implementado en sistemas médicos PET. Los datos temporales adquiridos por el sistema implementado se representarán y se registrarán en un fichero histórico o DataLogger, el cual ofrece la posibilidad de análisis de los valores adquiridos en cualquier otro momento. De esta forma, se pretende conseguir un sistema electrónico capaz de:

- Implementar el sistema en un dispositivo reconfigurable de bajo coste capaz de gestionar los datos y ofrecer la posibilidad de comunicarse con otros sistemas con una tasa de datos elevada.
- Desarrollar un sistema multicanal de alta resolución temporal capaz de ser reconfigurado en función de los canales que disponga el sistema PET en el que se desee implantar.
- Crear un entorno gráfico para la visualización de la resolución temporal de los datos obtenidos.
- Diseñar un entorno web mediante el cual se monitoricen los datos adquiridos de forma remota.

Como objetivos preliminares para poder implementar el sistema electrónico descrito anteriormente se plantea: de:

- Realizar un estudio preliminar del estado del arte y una revisión histórica de sistemas de medida de alta resolución temporal similares.
- Analizar los principios de funcionamiento de los sistemas médicos PET, para adaptar de la forma más óptima posible el sistema que se va a desarrollar e implementar en esta contribución.
- Revisar las tecnologías reconfigurables que existen actualmente en el mercado que mejor se adapten a las características del presente trabajo.



## **CAPÍTULO 2: DESCRIPCIÓN GENERAL DEL SISTEMA PET**

En el capítulo actual se van a presentar los sistemas médicos de Tomografía por Emisión de Positrones (PET), exponiendo sus orígenes, el principio de funcionamiento en el que se basan y las principales aplicaciones en las que se emplea para detectar enfermedades. Asimismo, se describirá la arquitectura general del PET y los módulos que lo conforman y se presentarán algunos ejemplos de estos sistemas médicos que se encuentran actualmente en el mercado.

### **2.1. Tomografía por Emisión de Positrones**

Los sistemas médicos basados en la Tomografía por Emisión de Positrones emplean una técnica médica mediante la cual se obtienen imágenes de la distribución espacial y temporal de los procesos metabólicos que se generan en el interior del organismo vivo (Herraiz, 2008).

El desarrollo de la técnica PET se fundamenta en el estudio in vivo de la radiación generada por el radiofármaco que se inyecta al paciente y ha favorecido la utilización de la medicina nuclear en la evaluación de las enfermedades puesto que es un método no-invasivo y muy útil para los investigadores médicos.

A continuación, se va a exponer una breve revisión histórica de estos sistemas médicos, la descripción de su principio de funcionamiento, se van a identificar y definir los módulos que lo conforman y, por último, se van a presentar algunos de los sistemas comerciales más destacados.

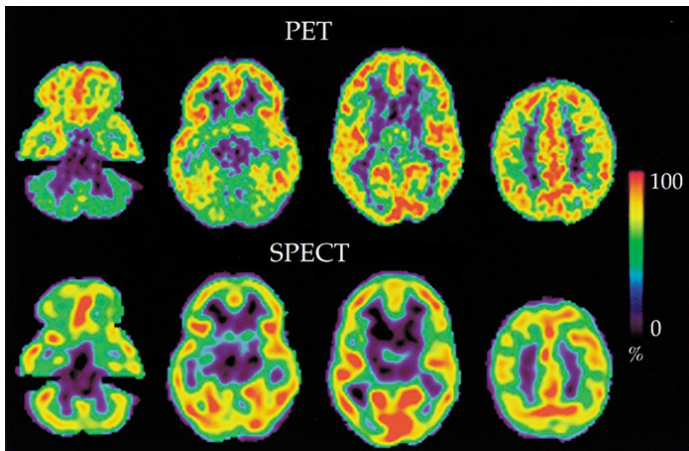
#### ***2.1.1. Revisión histórica del PET***

El nacimiento de la medicina nuclear tuvo sus comienzos durante la segunda década del siglo XX con el uso de técnicas basadas en el empleo de radioisótopos biológicamente no activos para el estudio de disfunciones orgánicas, por lo que no resultaban ser adecuados para el estudio de procesos químicos como el metabolismo. No obstante, el reconocimiento oficial de la medicina nuclear como nueva especialidad médica no se produjo hasta la década de los setenta tras el empleo de nuevas técnicas de imagen que hacían uso de isótopos biológicamente activos como el oxígeno, carbono o flúor (Guerra, 2007).

En las últimas décadas la medicina nuclear ha avanzado considerablemente, debido entre otros factores al enorme progreso y las innovaciones de las técnicas diagnósticas, permitiendo evaluar de forma fiable y mucho más precisa el estado del paciente. En este sentido cobra gran importancia las aportaciones realizadas desde el Consejo Europeo para la Investigación Nuclear (del francés, Conseil Européen pour la Recherche Nucléaire o CERN) basadas en el desarrollo de detectores de alta precisión capaces de detectar fotones procedentes de desintegraciones de partículas. Un ejemplo de la importancia del CERN en los orígenes de la medicina nuclear moderna fue el prototipo de scanner PRT (del inglés, Partial Ring Tomograph) desarrollado en la década de los ochenta, ya que fue uno de los primeros sistemas basados en la detección de tejidos malignos y el precursor de la tecnología de imágenes con fines médicos.

Históricamente, las técnicas de medicina nuclear proporcionaban tan solo imágenes planares y estaban basadas en la Tomografía Computerizada por Emisión de un Fotón Único (SPECT). Esta técnica, a pesar de ser muy popular y exitosa debido a su relativa simplicidad, se ha visto limitada por una reducida resolución espacio-temporal y por una pobre corrección de la atenuación (Millet, , *et al.*, 2000).

En la Figura 1 se muestran imágenes reconstruidas mediante la técnica SPECT (parte inferior) y su sucesora, la técnica PET (parte superior), pudiéndose apreciar la clara evolución en la calidad de la reconstrucción que ha supuesto esta última en la medicina nuclear.



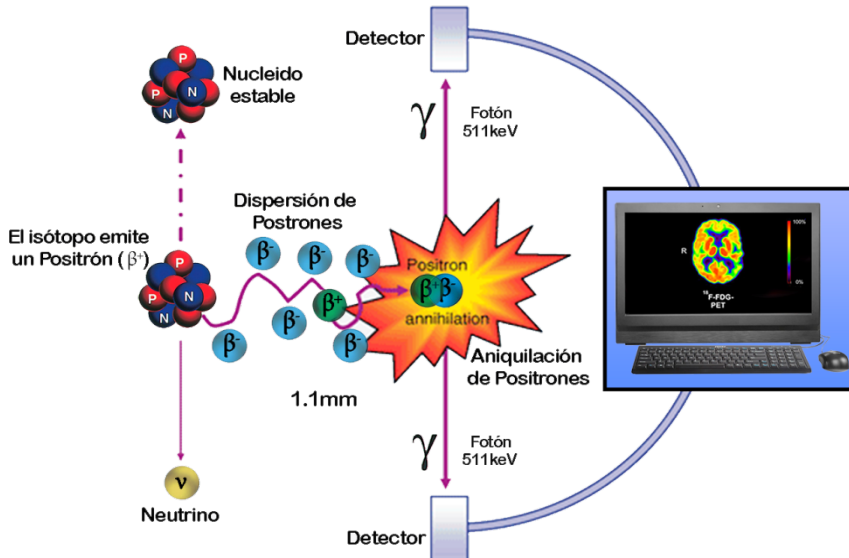
**Figura 1.** Comparativa de resultados de resolución entre técnica SPECT y PET.

**Fuente:** [3].

### 2.1.2. Principio de funcionamiento del PET

En cuanto al plano técnico, el sistema PET proporciona imágenes del organismo que se relacionan con el comportamiento celular mediante la detección de la radiación generada por el fármaco emisor de positrones. Para ello, realiza la detección del par de fotones de 511 keV, resultantes de la aniquilación positrón-electrón, que se propagan a lo largo de una línea, denominada Línea de Respuesta (del inglés, Line of Response o LOR). De este modo, los sistemas PET están formados por un conjunto de detectores de rayos gamma, colocados habitualmente en anillo, de forma que cada uno de ellos proporciona información acerca de los eventos que se han producido en su interior. La presente contribución muestra la parte encargada de detectar las parejas de fotones que se desplacen por la LOR y, en consecuencia, que impactan en detectores enfrentados dentro de una ventana temporal del orden de nano segundos (ns).

En la Figura 2 puede observarse la aniquilación de un positrón con su partícula opuesta y las dos partículas gamma generadas con un desplazamiento en dirección antiparalelo por la LOR e identificadas por dos detectores opuestos.



**Figura 2.** Representación de la generación y aniquilación de un positrón.

**Fuente:** elaboración propia.

### 2.1.3. Principales aplicaciones

Aunque en sus comienzos la principal función del PET radicaba en el estudio funcional del cerebro y del corazón, los avances en esta técnica han posibilitado su empleo en ámbitos como la oncología para la detección de cáncer, neurología para el estudio de enfermedades como la epilepsia y en investigaciones biomédicas.

Especialmente, la técnica PET cobra gran importancia en el campo de la oncología debido a que es capaz de detectar la actividad de masas muy pequeñas de células cancerosas que reflejan fielmente la actividad tumoral y, además permite discernir entre tumores benignos o malignos y el grado de malignidad de estos últimos.

Otra área en la que ha supuesto grandes avances ha sido en la enfermedad de Alzheimer, ya que la técnica PET cerebral permite medir la cantidad y distribución de la glucosa en el metabolismo y, de este modo, deducir la capacidad funcional de las zonas cerebrales.

## 2.2. Arquitectura general de un sistema PET

En este punto se va a realizar una breve descripción de los módulos que forman un proceso PET completo y a ubicar, dentro del mismo, el sistema de etiquetado temporal que se ha implementado y se presenta en esta contribución. De este modo, en la Figura 3 se diferencian en cuatro módulos todos los procesos que realiza un sistema médico PET:

1. Anillo de detectores
2. Tarjeta de *trigger*
3. Módulo de adquisición
4. Procesado y reconstrucción de imagen

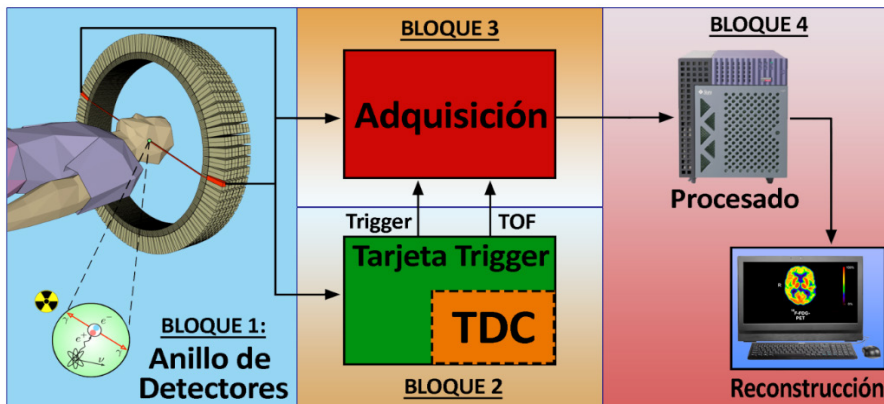


Figura 3. Arquitectura global de un sistema médico PET.

Fuente: elaboración propia.

### 2.2.1. Anillo de detectores

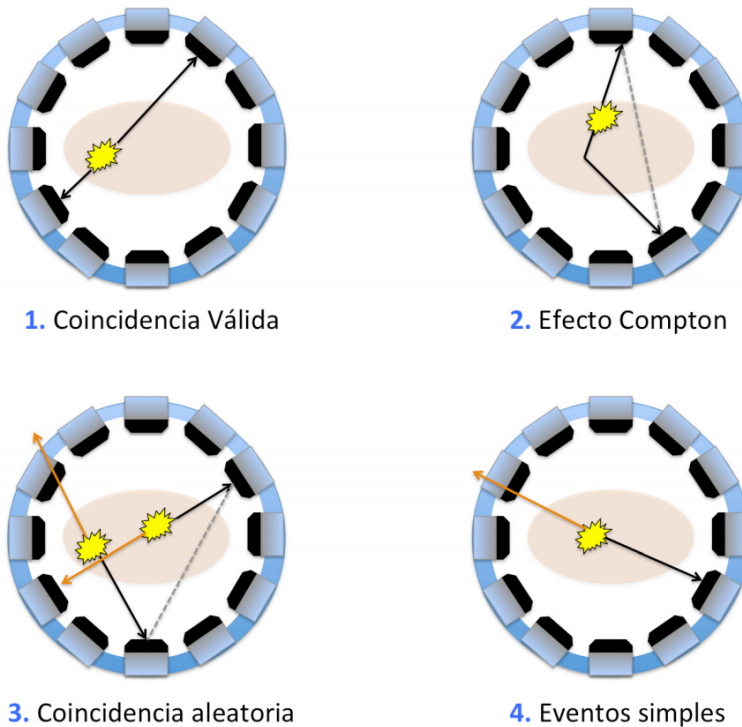
Los sistemas PET cuentan con un módulo de detectores, habitualmente dispuestos en anillo completo o parcial, encargado de identificar los rayos gamma que se producen en la aniquilación de un positrón por un electrón y de proporcionar información acerca de los eventos que se han producido en el interior del propio anillo.

La dirección en la que se dispersan los rayos o partículas gamma es antiparalela a través de la LOR, por lo que el sistema implementado debe monitorizar los eventos que se producen en los detectores cercanos a la LOR por la que se desplazan las partículas gamma procedentes de la aniquilación. Es muy importante conocer qué detectores han identificado las partículas gamma tras producirse la aniquilación para que los módulos que se explicarán a continuación puedan determinar si se ha detectado una zona afectada por algún tipo de anomalía.

### 2.2.2. Tarjeta de trigger

El módulo en el que se ubica la tarjeta de trigger es el encargado de determinar qué eventos procedentes de los detectores son correctos y cuáles deben descartarse, es decir, discierne si se ha producido lo que se conoce como coincidencia. Si se produce una coincidencia, la tarjeta de trigger envía al módulo de adquisición una señal de disparo empleando una ventana temporal mediante la cual se espera a que se produzca una aniquilación.

Se considera que una coincidencia es verdadera cuando los detectores reciben dos partículas gamma de la misma aniquilación en sentidos opuestos. No obstante, en esta ventana también es posible detectar coincidencias no verdaderas, como pueden ser las aleatorias, en las que se producen dos aniquilaciones seguidas y únicamente se detecta una; o las coincidencias dispersas en las que la LOR se asigna incorrectamente, es decir, las partículas gamma se detectan en dos detectores no opuestos. El Tiempo de Vuelo (del inglés, Time of Flight o TOF) toma gran importancia en este sentido ya que permite disminuir la ventana temporal para reducir el número de coincidencias no verdaderas. En la Figura 4 se muestran los tipos de coincidencias mencionados.



**Figura 4.** Tipos de eventos de coincidencia para PET.

**Fuente:** elaboración propia.

Esta tarjeta de trigger cobra gran importancia debido a que el sistema de etiquetado temporal de alta resolución o Time-to-Digital Converter que se presenta en este trabajo se ubica dentro de este módulo de disparo, y, además de proporcionar la señal disparo al módulo de adquisición, determina el valor del Tiempo de Vuelo que se detallará en el próximo capítulo.

### 2.2.3. Módulo de adquisición

La principal función de éste módulo es adquirir y procesar las señales procedentes de los detectores del anillo cuando reciba la señal disparo de la tarjeta de trigger, indicando que se ha producido una coincidencia.

Uno de los parámetros que puede emplearse para la reconstrucción de imágenes en sistemas PET es el valor de la energía. Una vez ha sido adquirido, digitalizado y almacenado este valor, se envía al bloque de procesado.

### 2.2.4. Procesado y reconstrucción de la imagen

En este bloque se procesará toda la información obtenida de los bloques anteriores para la reconstrucción de la imagen. Las principales funciones de este bloque de procesado están relacionadas con la corrección y descarte de las coincidencias aleatorias que hayan podido tener lugar en el proceso de adquisición y con la corrección en tiempo real de la pérdida de cuentas relacionada con el tiempo muerto. El tiempo muerto corresponde con el tiempo requerido para que un sistema



de conteo registre y procese completamente un evento, durante el cual no podrá ser registrado un evento adicional. Otras correcciones que suelen realizarse en el bloque de procesado son la normalización debido a la respuesta no uniforme de todos los elementos detectores, corrección de eventos producidos por la radiación dispersa dependientes del ancho de ventana seleccionado y corrección de la atenuación corporal que produce el paciente en el interior del anillo de detectores (Borrajó-Sánchez y Cabrero-Fraile, 2010).

En este bloque cobra especial importancia el valor del TOF, ya que, si además de procesar el valor de energía adquirido, en el algoritmo se introduce la variable TOF los resultados que se obtienen son mucho más precisos y permiten poder realizar una reconstrucción de imagen del organismo del paciente de mayor calidad.

### 2.3. Sistemas comerciales PET

La tecnología PET ha avanzado significativamente en los últimos años, con implicaciones en los equipos PET/CT, destacando el uso de nuevos cristales centelladores con la incorporación de la técnica del tiempo de vuelo (que se detallará en el próximo capítulo), la mejora en la resolución espacial y en la sensibilidad de los equipos, y los algoritmos de reconstrucción de imagen tomográfica. En la Figura 5 se detallan las características de diseño del tomógrafo PET de diversos equipos, relacionadas con estos avances (Martí-Climent, Prieto y García-Velloso, 2009).

**Tabla 2.** Características de diseño del tomógrafo PET de diversos equipos PET/CT.

Marca	Siemens		GE Healthcare			Philips	
	HI-REZ LSO	Biograph LSO	RX LYSO	STE/VCT BGO	ST BGO	Gemini GSO	TF LYSO
<b>Cristal</b>							
<b>Tamaño Cristal (mm<sup>3</sup>)</b>	4x4x20	6,45x6,45x25	4,2x6,3x30	4,7x6,3x30	6,3x6,3x30	4x6x20	4x4x22
<b>Nº cristales</b>	24336	9216	15120	13440	10080	17864	28336
<b>Campo Axial (cm)</b>	16,2	16,2	15,7	15,7	15,7	18	18
<b>Campo Transaxial (cm)</b>	58,5	58,5	70	70	70	56	57,6
<b>Modo de adquisición</b>	3D	3D	2D/3D	2D/3D	2D/3D	3D	3D
<b>Ventana de coincidencia</b>	4,5 ns	6 ns 4,5 ns*	6 ns	2D: 11,7 ns 3D: 9,3 ns	11,7 ns	7,5 ns	6 ns
<b>Ventana de energía (keV)</b>	425-650	350-650 400-650*	425-650	2D: 375-650 3D: 425-650	375-650	410-665	440-665
<b>Referencia</b>	Brambilla et al. <sup>22</sup>	Martinez et al. <sup>16</sup>	Kemp et al. <sup>60</sup>	Terás et al. <sup>61</sup>	Mawlawi et al. <sup>62</sup>	Lamare et al. <sup>63</sup>	Surti et al. <sup>17</sup>

\* Con la electrónica Pico-3D.

**Figura 5.** Características de diseño de diversos equipos PET.

**Fuente:** [5]

A continuación, se presentan algunas características de los sistemas que comercializan dos de las marcas más importantes en el campo de los sistemas PET: Philips y Siemens.

#### 2.3.1. Sistemas PET de Philips

Los últimos sistemas PET de Phillips que poseen la tecnología TOF diseñados en comparación con los sistemas no-TOF, permiten hasta un 30 por ciento de mejora en la resolución del contraste, manteniendo la precisión cuantitativa. Estos sistemas permiten obtener una resolución de tiempo de 495 pico segundos (ps) proporcionando la resolución más rápida para todos los sistemas PET disponibles en la actualidad con TOF. Philips ofrece ahora la tecnología TOF para prácticamente todos los presupuestos y necesidades clínicas y comercializa los siguientes modelos

de sistemas PET asegurando en todos ellos valores de resolución temporal inferiores a 650 ps (Philips Healthcare, 2010; Surti, *et al.*, 2007):

- Ingenuity TF
- TruFlight Select
- Gama GEMINI

Uno de los modelos más avanzados que ha desarrollado Philips es el sistema GEMINI LXL (Figura 6) que ofrece las tecnologías 4D PET/CT y destaca por la reconstrucción y procesamiento de imágenes PET de alta definición, una longitud de escaneo de 190 cm que permite obtener imágenes de cuerpo entero en una sola sesión y el empleo de nuevas tecnologías cristal que permiten escaneos más rápidos. De forma más específica, este sistema posee las siguientes características (Philips Healthcare, 2018):

- PET crystal material: LYSO
- PET crystal dimensions: 4 x 4 x 22 mm
- PET sensitivity (center): 7000 cps/MBq (NEMA)
- PET peak NECR: 94 kcps (NEMA)
- ET transverse spatial resolution (1 cm): 4.7 mm (NEMA)



**Figura 6.** Sistema PET comercial GEMINI LXL de Philips.

**Fuente:** [8].

### **2.3.2. Sistemas PET de Siemens**

La compañía Siemens es junto a Philips, otra de las grandes empresas de comercialización de sistemas médicos exploratorios de alta tecnología. Siemens posee actualmente cuatro sistemas PET divididos en dos familias (Casey, 2008):

- Gama Mct
- TruePoint PET-CT

De las familias anteriores, cabe destacar la TruePoint PET-CT (Figura 7) ya que es la que actualmente más evolucionada se encuentra de Siemens ya que ofrece imágenes

de alta definición, exploraciones de cuerpo completo en un tiempo de 10 minutos y consigue resoluciones temporales de hasta 500 ps en los sistemas que cuentan con TOF. En cuanto al plano técnico, el sistema TruePoint PET·CT de Siemens posee las siguientes características (Siemens Medical Solutions, 2018):

- 4x4 mm cristales LSO.
- 21,6 cm de campo axial PET
- 165 kcps NECR
- 185 cm de rango de escaneo



**Figura 7.** Sistema PET comercial TruePoint PET·CT de Siemens.

**Fuente:** [11].

## 2.4. Sumario

A partir de lo expuesto en el presente capítulo, se concluye que los sistemas médicos PET son el resultado de los últimos avances en el ámbito de la medicina nuclear y grandes empresas como Philips y Siemens han apostado por este tipo de tecnología para desarrollar equipos sanitarios, lo que significa que es una técnica muy depurada y que posee gran proyección.

Esto es debido a que los sistemas actuales alcanzan resoluciones temporales del orden de 500 ps, es decir, pueden detectar anomalías en el organismo de forma muy precisa. Cabe destacar que esta resolución que indican los fabricantes de los sistemas presentados es de carácter global, y por este motivo, el sistema Time-to-Digital Converter (TDC) que se presenta en el presente trabajo debe de cumplir como característica fundamental que posea una resolución temporal inferior a 100 ps. Este rango se considera óptimo debido a que existe una relación entre la resolución temporal y la precisión espacial del sistema dado que tomando como referencia la velocidad de la luz (aprox.  $3 \cdot 10^8$  m/s), a la cual llegan las partículas gamma a los detectores del PET, se tiene un error de 100 ps que se corresponde con una precisión de 3 cm, por lo que con esa resolución temporal podrían detectarse partículas cancerosas con un error de  $\pm 1,5$  cm.

Asimismo, los sistemas comerciales expuestos poseen una gran proyección debido la capacidad de determinar el TOF y la posibilidad de adquirir información empleando el método 4D de lectura de detectores.

*Implementación de un servidor web para gestionar un core time-to-digital-converter multicanal empleado en sistemas médicos PET*



## CAPÍTULO 3: ESTADO DEL ARTE

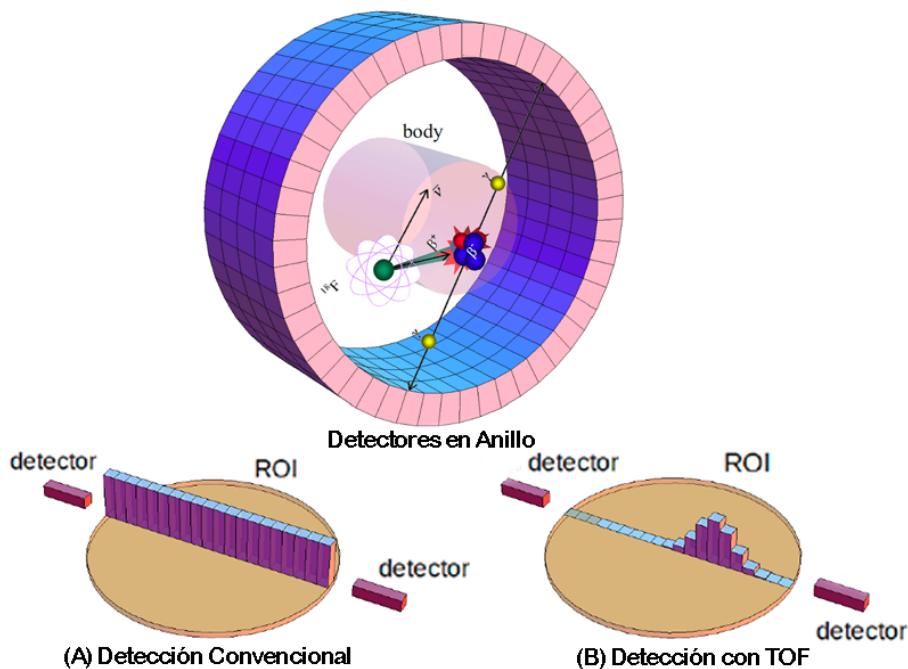
En los próximos puntos se van a describir las características principales del TDC que se emplea para realizar el etiquetado temporal de eventos. El TDC a implementar en este trabajo debe ser capaz de obtener una resolución inferior a 100 ps, que es la necesaria para obtener una mejora apreciable en la calidad de imagen de los PET actuales.

Otra de las características que debe cumplir el TDC es ofrecer la posibilidad de determinar el TOF. Una técnica que incorporan pocos PET en el mercado debido a su complejidad, y que conlleva múltiples ventajas como el aumento de la precisión en la detección de neoplasias, mejor reconstrucción de la imagen y la reducción del tiempo de exposición del paciente a los radiofármacos que son introducidos en su cuerpo.

### 3.1. Tiempo de vuelo (TOF)

Entre los avances más significativos en la tecnología de los sistemas PET se encuentra una que es de carácter puramente electrónico como es la capacidad del sistema de determinar el TOF (Surti , *et al.*, 2007), es decir, el tiempo que transcurre desde la creación de cada una de las partículas gamma hasta su interacción con el detector. Esto permite determinar con mucha mayor precisión la posición del radionúclido que ha generado dichas partículas y, por lo tanto, proporciona una importante mejora en la calidad de la imagen final, facilitando el diagnóstico al especialista y evitando, por ejemplo, en el caso de detección de cáncer, falsos negativos que serían críticos para el paciente (Torres , *et al.*, 2013).

En la Figura 8 se muestra cómo el TOF mejora la precisión de la posición en la que se ha detectado la aniquilación de positrones de modo que es posible discernir dicha posición en un punto de la LOR.

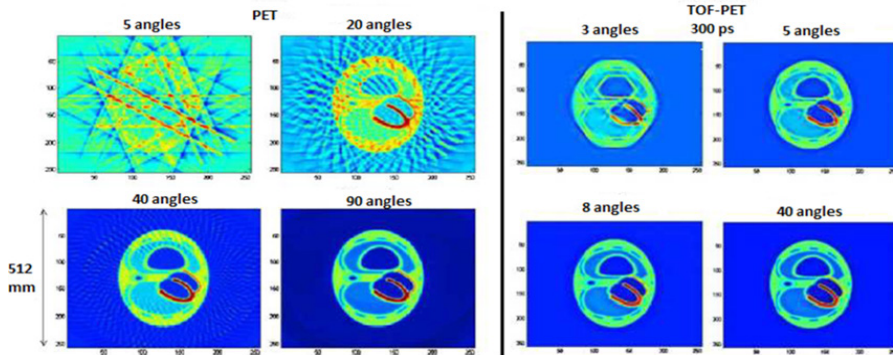


**Figura 8.** Tipos de detección de aniquilaciones en un sistema PET en anillo.

**Fuente:** elaboración propia.

Gracias a la medida del TOF también se consigue reducir el número de anillos y sensores obteniendo la misma calidad de resultados, esta técnica se vuelve muy interesante a la hora de implementar sistemas PET dado que permite mejorar sustancialmente el diseño de los mismos.

Asimismo, notar que la aplicación del TOF permite que la calidad de los resultados no dependa sólo de la cantidad de sensores y de la precisión de éstos sino también de la eficacia del cálculo del TOF, por lo que se podrían abaratar costes en la fabricación de los sistemas PET, ya que sería posible disminuir la cantidad y precio de los sensores obteniendo resultados idénticos (Karp, Surti, Daube-Witherspoon y Muehllehner, 2008). Ésta es una de las características más importantes del TOF debido a que, actualmente, los costes de un sistema PET son relativamente elevados y pocos centros médicos pueden permitirse adquirirlos. Como puede observarse en la Figura 9, con solo la medida de 5 ángulos y la aplicación del TOF se consiguen resultados idénticos o incluso mejores que la aplicación de 40 ángulos sin la utilización del TOF.

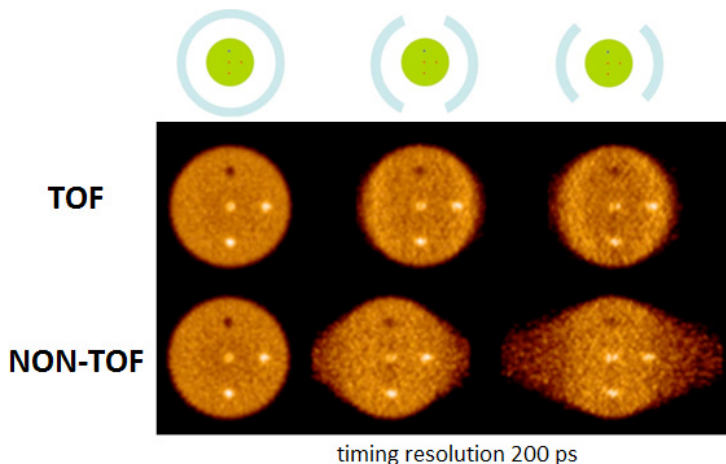


**Figura 9.** Tipos de detección de aniquilaciones en un sistema PET en anillo.

**Fuente:** elaboración propia.

Como se ha comentado anteriormente, uno de los beneficios que ofrece el TOF en sistemas PET es la reducción del tiempo de exposición del paciente y una menor dosis de radiofármaco en el cuerpo, dado que mediante el TOF se pueden obtener resultados más óptimos en un tiempo menor.

Otra de las ventajas del empleo del TOF en sistemas PET es la posibilidad de disminuir el número de detectores manteniendo los mismos resultados en la reconstrucción de la imagen que en sistemas con una cantidad de sensores mayor y, por tanto, de coste muy superior. Esta característica se muestra en la Figura 10 mediante una comparativa de tres tipos de arquitectura: en anillo completo, en anillo eliminando algunos detectores de la parte inferior y superior y, por último, empleando únicamente la mitad de detectores al colocarlos únicamente en los laterales del anillo.



**Figura 10.** Comparativa de resultados entre PET convencional (inf.) y PET con TOF (sup.) respecto al número de detectores.

**Fuente:** (Surti y Karp, 2007).

Existen diversas maneras de obtener electrónicamente el TOF. En una primera aproximación, los sistemas que obtienen el tiempo de vuelo se basan en circuitos analógicos, que utilizan fuentes de corriente extremadamente precisas

y que transforman la carga que acumula un condensador en valores de tensión proporcionales al tiempo de carga que luego son digitalizados (Napolitano, Alimenti y Carbone, 2010). Esto presenta diversas desventajas, principalmente problemas de escalabilidad, de dificultad en el diseño, y de disipación de potencia estática. En la actualidad se están empleando circuitos digitales junto con líneas de retardo en diversas configuraciones que emplean el retardo en la propagación de elementos lógicos individuales para realizar la medida del TOF en sistemas PET (Bogdan , *et al.*, 2005; Straayer y Perrott, 2009). Estos sistemas superan los problemas que aparecen con el enfoque analógico, y pueden incluso compensar los efectos de las fluctuaciones tanto en el voltaje de alimentación como en la temperatura.

Por consiguiente, el sistema de etiquetado temporal de alta resolución que se presenta en esta contribución debe implementar la medida del TOF, además de detectar los eventos producidos en el anillo de detectores. No obstante, la determinación del TOF puede ocasionar problemas debidos a los eventos, ya que se producen a una frecuencia indeterminada y deben ser detectados en tiempos del orden de pico segundos lo que significa que el sistema debe trabajar a frecuencias de decenas de giga hercios. El empleo de una frecuencia tan elevada ocasiona diversos problemas en el plano hardware por lo que, para solucionar esta problemática, se va a implementar un sistema de etiquetado temporal o TDC cuya estructura interna permite obtener resoluciones temporales de pico segundos sin necesidad de emplear relojes de sistema tan elevados.

## **2.2. Time-to-digital converter (TDC)**

En este punto se presentan las bases teóricas que permiten el desarrollo del sistema de etiquetado temporal y se realiza un recorrido por los principios de los Convertidores de Tiempo a Digital comenzando por su definición y revisando sus distintas arquitecturas y generaciones.

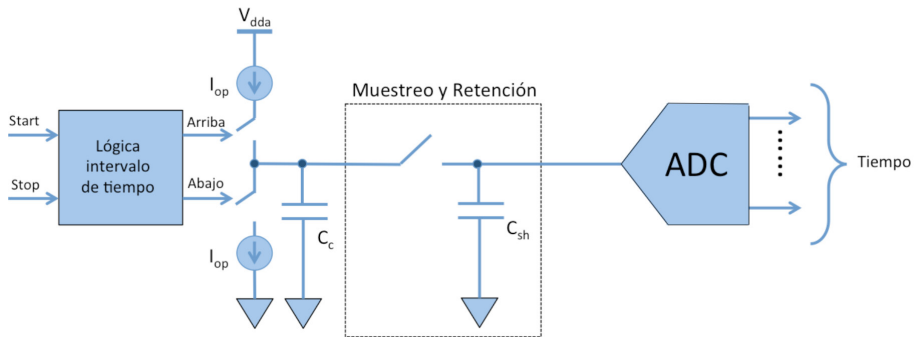
Un convertidor de tiempo a digital es un sistema que permite medir un dato de tiempo, discretizarlo y representarlo en un formato digital. De este modo, los convertidores representan el bloque fundamental entre la información codificada del dominio del tiempo continuo y el ámbito digital (Henzler, 2010). Los TDCs se emplean habitualmente en ámbitos como la física de alta potencia, el análisis de comportamiento de partículas y el análisis del TOF (Torres , *et al.*, 2013; Wu, 2009).

### **3.2.1. Primera generación: TDC analógico**

La primera generación de TDCs fue implementada al principio de la década de los noventa y se basaba en circuitos puramente analógicos formados generalmente por un Convertidor de Tiempo a Amplitud (del inglés, Time Amplitude Converter o TAC) y un sistema de adquisición ADC, de alta resolución y velocidad, encargado de digitalizar la señal convertida.

En cuanto a su funcionamiento, cuando se recibe una señal de Start un condensador comienza a cargarse a través de un circuito bomba de corriente hasta que el sistema recibe una señal de Stop, indicando ha finalizado el pulso temporal a medir, de forma que el ADC adquiere y convierte al dominio digital el valor de la tensión que se ha

almacenado en el condensador y, finalmente, se produce la descarga del mismo (Gao, Gao, Hu-Guo y Hu, 2011). En la Figura 11 se muestra el diagrama de bloques genérico de este tipo de circuito analógico.



**Figura 11.** Arquitectura de un circuito TDC analógico.

**Fuente:** elaboración propia.

Esta solución analógica posee muy poca linealidad debido a que la resistencia de salida de la fuente de corriente del integrador es finita y no ideal por lo que en estado de saturación perdería la linealidad. Para solventar este problema, se podría utilizar un integrador RC pero, en ese caso, ocasionaría limitaciones en el ancho de banda.

### 3.2.2. Segunda generación: TDC digital

A pesar de la precisión temporal que ofrecían los TDC analógicos, la necesidad de implementar un sistema que trabaje en el ámbito de la lógica digital generó la segunda generación de TDCs. En estos nuevos sistemas, cobró gran importancia la tecnología CMOS dado que permitió realizar implementaciones más económicas y compactas, un procesamiento más flexible y complejo de las señales y un bajo consumo que su predecesor analógico no podía ofrecer.

Otro factor muy importante por el que los diseños de TDCs se inclinan hacia los sistemas digitales, es la estabilidad ante distorsiones como ruido y acoples y la modularidad que ofrecen. De este modo, los diseños digitales son muy óptimos, siempre que no exista una etapa de conversión analógica dentro del sistema, que convierta el valor de tiempo a digital.

La técnica en la que se basan habitualmente los TDCs para cuantificar intervalos de tiempo es la cuenta de los ciclos de reloj que se producen en el correspondiente intervalo de medición. Una de las posibles arquitecturas de un TDC de segunda es la basada en dos señales de entrada (Start y Stop) que son procesadas por un biestable RS que genera una señal de habilitación. Esta señal controla el ancho de la señal de reloj que acciona el contador, de modo que, cuando se sitúa a nivel lógico alto a la salida del sistema, se obtiene el último valor de cuenta en formato digital que indica la diferencia temporal que se ha medido entre la señal Start y Stop.

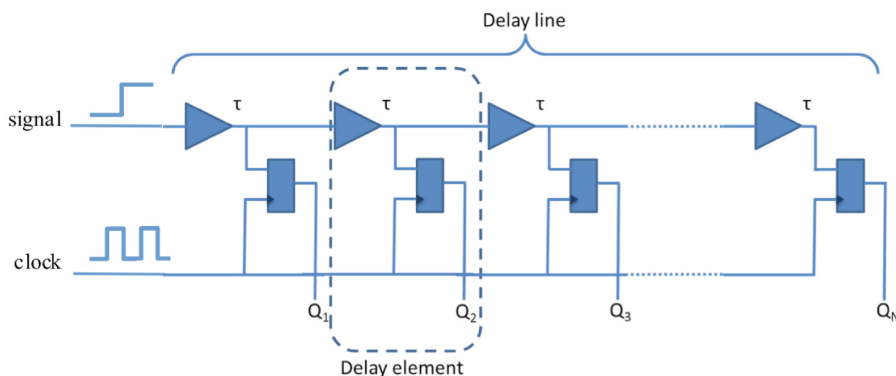
Para mejorar la exactitud de la medición de tiempos en este tipo de sistemas se puede aumentar la frecuencia del reloj, sin embargo, como consecuencia de esto el

consumo se vería incrementado y la cuenta de pulsos estaría siempre limitada por la frecuencia del reloj de referencia del sistema [20]. Otro inconveniente que debe tenerse en cuenta en este tipo de arquitectura es el ruido que puede introducir el biestable RS junto con la puerta lógica, ya que puede afectar a la precisión de la conversión, particularmente, en sistemas que requieran alta velocidad.

### 3.2.3. Tercera generación: TDC basado en línea de retardo

La tercera generación de TDCs son los formados por líneas de retardo (del inglés, Delay Lines o DLL) implementadas mediante puertas digitales de forma que pueden proporcionar una resolución temporal muy inferior a la que presentaban las dos generaciones anteriores. Esto permite determinar resoluciones temporales inferiores a la frecuencia máxima del reloj limitada únicamente por la tecnología de fabricación de los componentes lógicos del propio TDC. Estas líneas de retardo están formadas por circuitos lógicos o células analógicas que representan las etapas de la misma y determinan la resolución temporal máxima del sistema.

La arquitectura más sencilla de este tipo de TDC emplea una técnica basada en generar el retardo requerido mediante el paso de la señal a través de puertas lógicas inversoras colocadas en cascada de modo que se genera un retardo asociado y conocido en cada puerta inversora. El paso por estas puertas queda almacenado en un registro que indicará la cantidad de puertas lógicas que se han atravesado durante el intervalo comprendido entre las señales de Start y Stop. Dado que el retardo de las puertas está caracterizado y es conocido, es posible determinar, a partir del número de etapas que se han atravesado, el valor de la diferencia temporal.



**Figura 12.** Línea de retardo compuesta por puertas lógicas y biestables.

**Fuente:** elaboración propia.

La resolución de este tipo de sistemas está ligada al retardo asociado a las puertas que se empleen y, cabe destacar, que esta arquitectura está limitada por el número de puertas que pueden conectarse en cascada, ya que este tipo de conexiones genera desajustes en el retardo de cada etapa.

La expresión matemática que permite obtener el retardo de cada etapa o resolución máxima del sistema viene dada por:

**Fórmula 1.** Tiempo de retardo de una célula.

$$T_{etapa} = \frac{T_{clk}}{N}$$

**Fuente:** elaboración propia.

donde,  $T_{etapa}$  determina el tiempo de retardo de una célula,  $T_{clk}$  es el periodo de reloj del sistema y  $N$  es el número de etapas o células de la línea de retardo.

Una variante de arquitectura basada en línea de retardo es la denominada matriz de líneas de retardo. Para la construcción de la matriz se utilizan dos líneas de retardo simples de modo que la resolución viene definida por la diferencia de tiempo existente entre las líneas de retardos, empleando en toda la matriz el mismo reloj de referencia. Cabe destacar que el aumento de la resolución es proporcional al aumento del área efectiva de la matriz, es decir, si se eleva el número de filas y columnas de retardo se puede conseguir un aumento de la resolución temporal.

Esta arquitectura mejora la resolución temporal si se compara con la arquitectura simple, sin embargo, emplea el doble de recursos en cuanto a circuitería, y la resolución está limitada por el *jitter* y el *offset* de las líneas de retardo de la matriz.

Otra variante de la arquitectura de líneas de retardo es el método de medida basado en el principio de Vernier. Para implementar este método se realiza la resta entre los tiempos que tarda una señal en atravesar dos líneas de retardo independientes de manera que, al emplear el método Vernier, es posible hallar la diferencia de tiempo que existe entre ambas.

El principal fundamento de este método radica en que la diferencia de tiempo de la etapa de retardo en dos líneas de retardo debe ser exactamente equivalente al periodo de reloj dividido por el número de etapas de retardo. De forma matemática:

**Fórmula 2.** Diferencia de tiempo de retardo entre líneas.

$$T_{etapa} = T1 - T2 = \frac{T_{clk}}{2^n}$$

**Fuente:** elaboración propia.

donde  $T1$  y  $T2$  son el tiempo de retardo de cada etapa en dos líneas de retardo respectivamente, y  $m$  es el número de bits de la salida digital. Para implementar un TDC basado en la línea de retardo de Vernier deberían emplearse dos líneas de retardo cuyas señales de reloj de referencia estuvieran totalmente sincronizadas, ya que un desfase entre sus señales de reloj desencadenaría un error en el dato de tiempo digital de la salida [21].

### 3.2.4. Tercera generación: TDC basado en línea de retardo

En este punto se van a referenciar y a describir brevemente algunas publicaciones sobre TDCs implementados en sistemas médicos PET:

- B.K. Swann , *et al.*, “A 100-ps time-resolution CMOS time-to-digital converter for positron emission tomography imaging applications” (2004). La arquitectura de este TDC combina un contador digital y un circuito de interpolación temporal para realizar la medida del intervalo de tiempo. El rango dinámico de este sistema puede ser reprogramado sin degradarse a penas la resolución temporal y para la conversión emplea un conversor ADC de 5 bits, obteniendo una resolución de 312.5 ps.
- O. Bourrion y L. Gallin-Martel, “An integrated CMOS Time-to-Digital Converter for coincidence detection in a Liquid Xenon PET prototype” (2006). Este TDC estaba basado en una arquitectura de dos contadores y una línea de retardo con 128 etapas de retardo. Este sistema fue implementado en tecnología CMOS de 0,35  $\mu\text{m}$  y obtenía una resolución inferior a 250 ps.
- A.S. Yousif , *et al.*, “A Fine Resolution TDC Architecture for Next Generation PET Imaging” (2007). La arquitectura de este TDC se basa en una jerarquía de estructura de retardo que logra una latencia de un ciclo y alta velocidad de operación, consiguiendo resoluciones temporales de 31 ps y un consumo inferior a 1 mW.
- C. Hervé , *et al.*, “High resolution time-to-digital converter (TDC) implemented in field programmable gate array (FPGA) with compensated process voltage and temperatura PVT variations” (2012). En esta contribución se presentan dos posibles arquitecturas para la implementación de un TDC en un dispositivo *Field Programmable Gate Array* o FPGA: la primera está basada en un interpolador de línea de retardo consiguiendo resoluciones entre el rango de 40-100 ps y la segunda toma como base un interpolador con reloj multifase consiguiendo resoluciones cercanas a 175 ps.

### 2.3. Sumario

En este capítulo se han destacado, por un lado, las características que posee el TOF junto con los beneficios que este parámetro puede aportar en un sistema médico PET ya que gracias a la medida del TOF se consigue reducir el número de anillos, se necesitan menores tiempos de exposición del paciente al tener que emplear una dosis menor de radiofármaco y se obtienen imágenes de mejor calidad para un mismo número de detectores.

Por otro lado, se ha propuesto el sistema TDC como encargado de determinar el TOF junto con las etiquetas temporales de los eventos que se producen en el anillo de detectores del sistema PET y se han presentado las diferentes arquitecturas que poseen los TDC, concluyendo que suelen cumplir las siguientes características:

1. Se emplean contadores para obtener el ancho de pulso.
2. Se realiza interpolación temporal y técnicas de muestreo múltiple para obtener una alta precisión.



3. Se adquiere la señal mediante un ADC, normalmente de alta resolución para digitalizar el valor temporal.

En atención a lo expuesto, se concluye que el tipo de arquitectura que se necesita para implementar el TDC de este trabajo estará basada en los de tercera y cuarta generación debido a se posee como objetivo principal, implementar un TDC de alta resolución capaz de medir diferencias temporales inferiores a 100 ps. Una de las soluciones más empleadas en el ámbito de los TDCs implementados en sistemas médicos PET es la arquitectura de líneas de retardo, dado que ofrecen una buena resolución temporal y pueden ser caracterizadas con el objetivo de realizar una calibración previa para compensar los errores de retardo que pudiera introducir cada etapa a la línea de retardo completa.

Por el contrario, los TDC basados en líneas de retardo deben seguir las siguientes indicaciones para evitar posibles desviaciones en la medida: las líneas de retardo deben de ser lo suficientemente extensas como para cubrir un periodo completo de reloj, las etapas de las líneas deben de colocarse en cascada lo más próximas posibles describiendo una línea recta y todas las etapas que componen las líneas de retardo deben poseer las mismas características y pertenecer a la misma tecnología.

*Implementación de un servidor web para gestionar un core time-to-digital-converter multicanal empleado en sistemas médicos PET*

## CAPÍTULO 4: DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA

En los siguientes puntos se va a realizar una revisión de las posibles tecnologías que podrían emplearse para llevar a cabo la implementación del TDC, certificando que los dispositivos FPGA son los que mayores beneficios aportan para ello. De acuerdo con esta óptica, se van a describir los aspectos más importantes de los dispositivos FPGA así como la metodología y las herramientas de programación que van a emplearse para implementar el TDC en uno de estos dispositivos.

Asimismo, se describirán los elementos que se implementarán dentro de la FPGA para constituir un TDC gestionado por un microprocesador y con capacidad de comunicación con un gran abanico de periféricos.

### 4.1. Estudio de tecnologías

En esta contribución se ha empleado una solución digital basada en un dispositivo FPGA de bajo coste, tanto por la excelente relación precio-prestaciones como por la versatilidad que proporcionan los sistemas lógicos reconfigurables que presentan las FPGAs comparado con otras alternativas basadas en ASICs (del inglés, Application-Specific Integrated Circuit). Otras ventajas de las FPGAs son: la elevada tasa de transferencia, gran velocidad de procesado y que ofrecen la posibilidad de trabajar simultáneamente con un número elevado de señales.

Las principales características por las que se ha decidido emplear un dispositivo FPGA en lugar de un ASIC para implementar un TDC con posibilidad de determinar el TOF se recogen en la Tabla 1.

**Tabla 1.** Comparativa entre FPGA y ASIC.

	<b>Ventajas</b>	<b>Desventajas</b>
<b>FPGA</b>	Bajo coste	Completamente digital
	Time-to-market	Rendimiento menor que un ASIC
	Dispositivo reconfigurable	
	Permite integrar un procesador	
<b>ASIC</b>	Costes recurrentes	Alto coste de desarrollo
	Nº de componentes optimizado	Difícilmente reconfigurable
	Alto rendimiento	Alto volumen para ser rentable

**Fuente:** elaboración propia.

Además de estas características, las FPGAs poseen una estructura interna mediante la que es posible medir tiempos con elevada precisión empleando líneas de retardo, lo que favorece el cálculo del TOF (Haselman, 2010). Sus estructuras internas repetitivas, facilitan la implementación de cadenas de retardo, las cuales se configurarán de manera que puedan medir tiempos del orden de centenares de pico segundos.

## 4.2. Descripción de la tecnología empleada: FPGA

La mayor dificultad de este sistema radica principalmente, en el cálculo del TOF con una resolución temporal dos órdenes de magnitud por debajo del periodo del reloj del dispositivo FPGA. Para comprender correctamente el método empleado para conseguir este objetivo es necesario realizar una visión general de la lógica interna de las FPGAs y, de forma más específica, del modelo empleado en esta contribución.

### 4.2.1. Estructura de un dispositivo FPGA

Las FPGAs son un tipo de circuito digital reconfigurable con arquitectura distribuida que permiten la realización de cualquier sistema combinacional o secuencial. Como se muestra en la Figura 12, este tipo de dispositivos se caracterizan por poseer recursos constituidos por bloques lógicos reconfigurables repartidos por todo el circuito, como son los bloques lógicos internos y los bloques de entrada/salida, y por recursos de interconexión formados por líneas de interconexión y conexiones configurables.

De forma más específica, los bloques lógicos internos realizan las funciones principales de la aplicación, los bloques de E/S permiten conectar el dispositivo con el exterior y los recursos de interconexión conectan interiormente cada bloque lógico dando versatilidad a la FPGA.

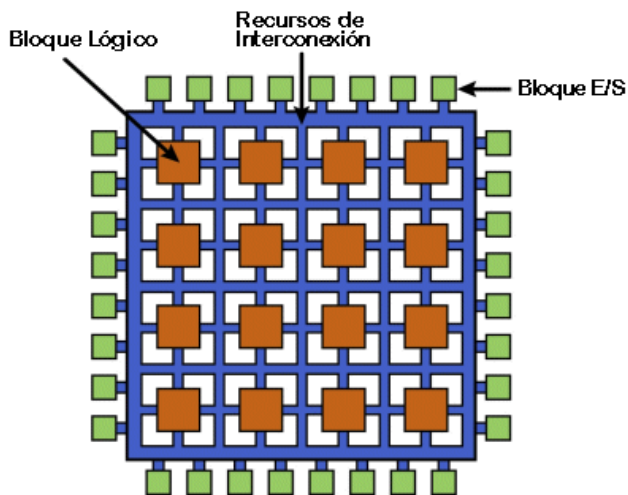


Figura 12. Estructura interna de un dispositivo FPGA.

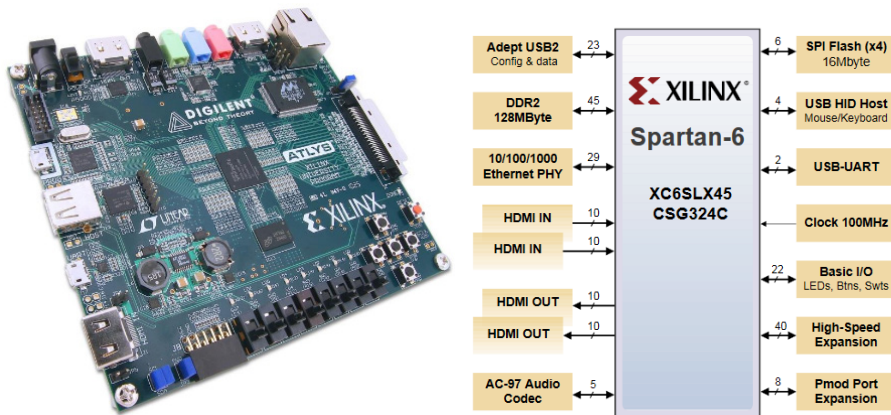
Fuente: elaboración propia.

### 4.2.2. Descripción del dispositivo empleado: FPGA Spartan 6

Se ha decidido emplear la familia Spartan-6 para realizar la implementación del TDC debido a que se trata de una familia de bajo coste y a su vez posee una estructura interna basada en bloques de acarreo que permiten implementar de forma óptima las líneas de retardo descritas en el capítulo anterior (Xilinx, 2015). Las principales características de la familia Spartan-6 son las siguientes (Xilinx, 2011):

- Bajo coste ya que están orientadas a aplicaciones de usuario para grandes volúmenes de producción.
- Pines multi-voltaje y multi-estándares. Hasta 540 pines de entrada salida LVCMOS, LVTTTL, HSTL y SST. 3.3V, 2.5V, 1.8V, 1.5V y 1.2V. Hasta 1.080 Mb/s de velocidad de transferencia para E/S diferenciales con LVDS, RSDS, mini-LVDS, HSTL/SSTL.
- Recursos lógicos: Multiplexores rápidos, multiplicadores dedicados, JTAG.
- DCM (*Digital Clock Manager*): eliminación de “skew”, síntesis de frecuencia.
- 8 líneas de reloj global.
- Soporta el microprocesador Microblaze, PicoBLaze y otros IP *Cores*.

El dispositivo FPGA que se ha empleado para la implementación del presente trabajo ha sido el xc6slx45 de la familia Spartan-6 de Xilinx. Esta FPGA se encuentra integrada en el kit de desarrollo Atlys de Digilent (Figura 13) que permite realizar Comunicaciones Serie y Ethernet, posee un módulo PMOD (del inglés, Peripheral Modules) estándar de doce conexiones, que ofrece la posibilidad de conectar la placa a entradas y salidas para inyectar o leer señales de la FPGA, e incluye algunos periféricos como LEDs, interruptores y pulsadores que permiten interaccionar con el dispositivo. En la siguiente figura se muestra un diagrama de bloques de los periféricos a los que se encuentra conectada la FPGA Spartan-6 en el kit de desarrollo Atlys.



**Figura 13.** Kit de desarrollo Atlys de Digilent.

**Fuente:** (Digilent, 2016).

De forma más específica, la FPGA xc6slx45t posee las siguientes propiedades: contiene 6.822 slices, cada slice está formada por cuatro LUTs de 6 entradas y ocho flip-flop; 2.1Mbits de bloques de memoria RAM; cuatro líneas de reloj (ocho DCMs y cuatro PLL); 58 bloques DSP y soporta hasta 500 Mhz. En la siguiente figura se muestran los diferentes dispositivos.

### **4.3. IP Cores**

Los IP Cores o Núcleos de Propiedad Intelectual son unidades de lógica independiente que realizan una función determinada y que pueden ser integradas en diferentes sistemas de modo que el uso de estos bloques simplifica y agiliza el diseño de un circuito electrónico como una FPGA o un ASIC. Un Core a menudo adopta la forma de un programa de computadora escrito en lenguaje de descripción hardware HDL, tales como Verilog, VHDL o SystemC. Idealmente, un IP Core debe ser totalmente modular, es decir, que fácilmente se pueda adaptar a cualquier tecnología de otros proveedores o diferentes métodos de diseño.

La finalidad de los IP Cores es minimizar la aparición de problemas durante la etapa de diseño, disminuir el time-to-market del producto y reducir los costes que conlleva el desarrollo del mismo. Algunos ejemplos de IP Cores son Receptores/Transmisores Asíncronos Universales (UART), Unidades Aritmético-Lógicas (ALU), Controladores Ethernet, Interfaces PCI...

En el presente trabajo, para poder obtener resoluciones temporales de tal magnitud, generar las etiquetas temporales para las múltiples señales o canales de entrada y realizar la medida del TOF en sistemas PET con una resolución muy inferior a la frecuencia del reloj del sistema, se va a integrar en la FPGA un Core Time-to-Digital Converter (TDC) que ha sido programado en el lenguaje de descripción hardware VHDL (del inglés, Very High Speed Integrated Circuit). Este Core TDC se integrará dentro de la FPGA y proporcionará las etiquetas temporales correspondientes a los eventos producidos en el anillo de detectores. Esta información, será gestionada por un microprocesador software de Xilinx que a su vez se encargará de realizar la comunicación con el entorno de visualización de datos.

### **4.4. MicroBlaze**

Para la realización de este sistema se ha integrado un procesador software MicroBlaze de 32 bits de Xilinx en la propia FPGA con el objetivo de realizar la gestión de los periféricos del sistema, la Comunicación Serie y Ethernet con el PC así como del Core TDC, ya que debe recibir la información del mismo y procesarla. Bajo esta perspectiva, MicroBlaze aporta al sistema mayor versatilidad puesto que se programa con lenguaje de alto nivel y no con VHDL como ocurre con el Core TDC.

El fabricante Xilinx permite integrar en sus FPGAs diversos microprocesadores que proporcionan características y rendimientos muy diferentes. Como se aprecia en la siguiente figura, el microprocesador empotrado más básico es Picoblaze, que es un bloque microprocesador software con arquitectura de 8 bits. A continuación, se sitúa MicroBlaze, que es el microprocesador objeto del estudio, y que también es un bloque microprocesador software de propósito general, pero a diferencia de PicoBlaze, posee una arquitectura de 32 bits y, por último, se encuentra PowerPC, un microprocesador hardware de 32 bits que habitualmente se emplea en ámbitos industriales.

#### 4.4.1 Arquitectura de MicroBlaze

Microblaze es un bloque microprocesador software que ha sido diseñado específicamente para su implementación en FPGAs del fabricante Xilinx. Sus características principales son que, presenta un juego de instrucciones reducido (RISC) y que permite el acceso a instrucciones y datos de forma independiente (Harvard), aunque es habitual que instrucciones y datos estén en la misma memoria física para facilitar de este modo la depuración del software. Las características principales de MicroBlaze son:

- Presenta un núcleo de 32-bits.
- Los buses de direcciones y datos son de 32 bits.
- Tiene 32 registros de propósito general (R0-R31).
  - Estos registros se ponen a cero cuando se configura la FPGA.
  - Además, no se modifican después de un Reset.
- Presenta instrucciones de 3 operadores y 2 modos de direccionamiento (directo e inmediato).
- Posee un juego de instrucciones reducido (RISC) que permite el acceso a instrucciones y datos de forma independiente (Harvard).
- Presenta una *pipeline* de 5 etapas: la primera instrucción tarde en ejecutarse 5 ciclos de reloj para evitar errores de sincronización. Las demás instrucciones se ejecutan en un ciclo de reloj.
- La Unidad Aritmético-Lógica (ALU) realiza las siguientes operaciones hardware: multiplicación y división entera y desplazamiento de bits con solo una instrucción.
- Formato *Big-Endian* (b0...b31).
- Interfaz de depuración hardware (MDM).
- Multiplicación y división entera hardware.

MicroBlaze posee acceso directo tanto a la memoria local como al contador de la memoria externa y a los diferentes coprocesadores que se pueden integrar. Por otro lado, mediante el empleo de un bus compartido tiene acceso al resto de bloques de la FPGA (control de interrupciones, timers, Ethernet, diferentes puertos PCI, USB...) así como a diversos periféricos configurables de entrada/salida.

#### 4.4.2. Buses y comunicaciones

La comunicación de MicroBlaze con los periféricos y Cores se establece mediante el bus principal del sistema denominado PLB (del inglés, Processor Local Bus) que sigue el estándar de arquitectura de bus CoreConnect de IBM y está formado por maestros (hasta 16), esclavos (sin límite, aunque Xilinx recomienda hasta 16), árbitro de bus encargado de gestionar las peticiones y prioridades del bus e interconexiones realizadas mediante bridges.

Xilinx ofrece otro bus denominado AXI4 (Interfaz Avanzada Extensible 4) correspondiente a la cuarta generación de la especificación de interfaz de AMBA. Se trata del nuevo estándar recientemente adaptado por Xilinx como interfaz mejorada a utilizar en actuales y futuras versiones. Consolida una amplia gama de interfaces en una única (AXI4), por lo que los diseñadores sólo necesitan saber una familia de interfaces. Hace que la integración de IP Cores desde diferentes ámbitos, así como el desarrollo de IP Cores propios sea más fácil. A su vez, cabe destacar que AXI4 está optimizado para conseguir el máximo rendimiento, mayor flexibilidad, máximas prestaciones y menor latencia.

De este modo, para la implementación del sistema TDC se han empleado ambos buses. En primer lugar, se ha implementado un sistema en el que la velocidad de funcionamiento no es la máxima prioridad, sino que lo es el correcto funcionamiento del TDC, por lo que se ha empleado bus PLB al poseer más conocimientos y disponer de mayor experiencia en el ámbito de sistemas integrados empleando dicho bus. En segundo lugar, una vez comprobado el correcto funcionamiento del TDC, se ha buscado obtener las máximas prestaciones posibles en cuanto a frecuencia de funcionamiento del sistema y velocidad de comunicación tanto entre periférico como con el entorno de visualización, por lo que se ha empleado el bus AXI4.

Por otro lado, se encuentran buses de comunicaciones jerárquicamente inferiores a PLB que se listan y se describen brevemente a continuación:

- **Local Memory Bus (LMB)** que permite un acceso rápido en un solo ciclo de reloj a la memoria dedicada (BRAM) dentro de la FPGA, realiza transferencias de información muy eficientes debido al mínimo número de señales de control y protocolo simple y es más rápido que el Bus PLB (125 MHz).
- **Fast Simplex Links (FSL)** son interfaces unidireccionales punto a punto dedicados y sin arbitraje basados en memoria FIFO, permiten acceso rápido en 2 ciclos de reloj, posee un máximo de 8 entradas y 8 salidas, posee un tamaño de FIFO configurable y reloj de FIFO síncrono o asíncrono.
- **Xilinx Cache Link (XCL)** permite la conexión directa de memorias externas a MicroBlaze mediante buffers de tipo FSL, necesita el uso de caché, puede controlar hasta 8 memorias externas DDR y emplea un controlador de memorias diseñado por Xilinx.

## **4.5. Herramientas para la programación de la FPGA**

### **4.5.1. Metodología de diseño y programación**

Para poder desarrollar e implementar el Core TDC comandado por el microprocesador MicroBlaze es necesario realizar un codiseño, es decir, un diseño concurrente por parte del mismo equipo de diseñadores de los componentes hardware y software de un sistema empotrado o integrado, usando para ello metodologías y herramientas que consideren la interacción hardware/software.



Con el codiseño se intenta conseguir un compromiso óptimo entre las soluciones hardware que ofrecen buenas prestaciones en velocidad, aunque el coste puede ser elevado; y las soluciones software que son más flexibles y de menor coste, aunque más lentas.

El flujo básico que se ha desarrollado para realizar el codiseño ha sido el siguiente:

- Especificación: Definir la funcionalidad del sistema a desarrollar.
- Particionado hardware/software: Decidir que funciones se realizarán mediante rutinas software en el microprocesador empustrado y cuáles mediante circuitos hardware específicos.
- Descripción hardware: Elección del microprocesador y de la FPGA adecuados para la aplicación y diseño de *Cores* mediante lenguaje HDL.
- Desarrollo Software: Desarrollo de la aplicación en lenguaje de alto nivel, como puede ser C, y compilación.

En la Figura 14 se muestra el flujo básico que se ha seguido para realizar el codiseño.



**Figura 14.** Diagrama de flujo del codiseño seguido.

**Fuente:** elaboración propia.

#### 4.5.2. Xilinx ISE Design Suite

El conjunto de herramientas de programación que proporciona Xilinx ISE Design Suite permite realizar el codiseño del sistema TDC a implementar. En el presente trabajo, los programas ISE, ISim, XPS y PlanAhead han permitido desarrollar la parte hardware del sistema mientras que para la programación software se ha empleado la herramienta SDK. A continuación, se describen brevemente cada una de estas herramientas de programación.

Implementación de un servidor web para gestionar un core time-to-digital-converter multicanal empleado en sistemas médicos PET

## ISE Project Navigator

La creación, compilación, síntesis y depuración del Core TDC programado en VHDL se ha realizado con la herramienta ISE Project Navigator (Figure 15). Este software permite otorgarle la funcionalidad necesaria que debe realizar el TDC y verificar su correcto funcionamiento al ser integrado en la FPGA.

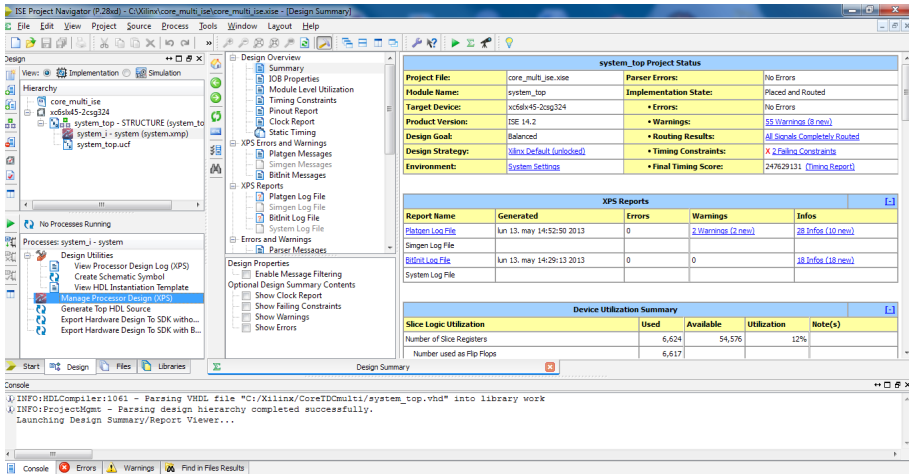


Figura 15. Entorno de ISE Project Navigator.

Fuente: elaboración propia.

## ISim

Antes de integrar el Core TDC en la FPGA se han realizado las pertinentes simulaciones empleando la herramienta ISim para la depuración y la prueba de señales de entrada. En la Figura 16 se muestra una captura del programa ISim empleado para la simulación.

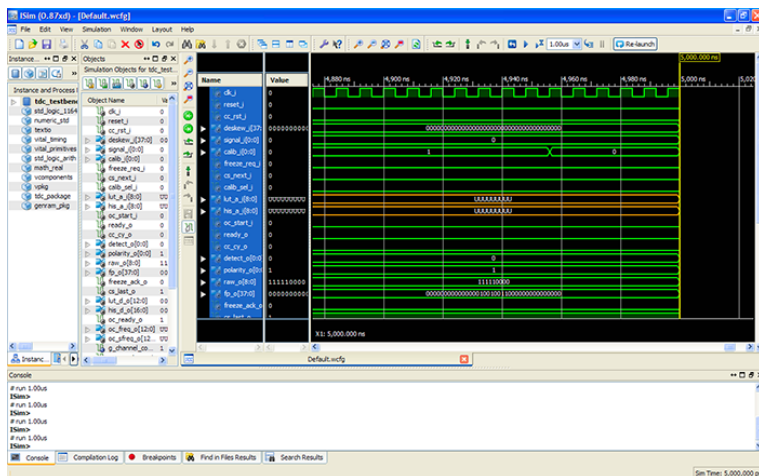
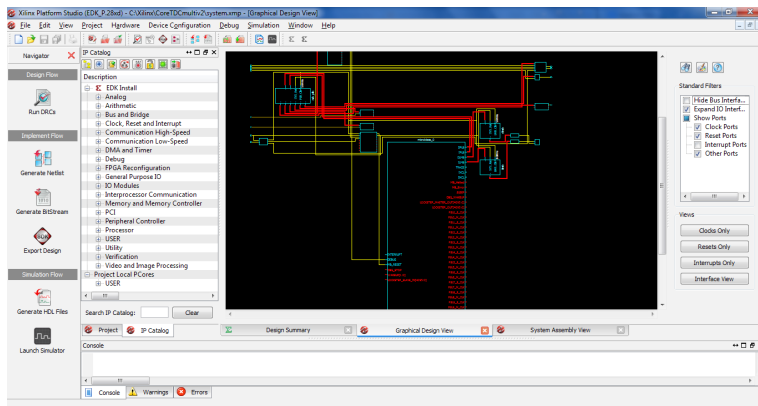


Figura 16. Simulación del TDC con la herramienta ISim.

Fuente: elaboración propia.

## XPS

Para crear el microprocesador software MicroBlaze de los periféricos (Comunicación Serie UART, Ethernet, interruptores...) integrar el Core TDC diseñado en Project Design y generar la configuración hardware completa de la FPGA se ha empleado la herramienta Xilinx Platform Studio (XPS) perteneciente también a la suite Xilinx ISE. De esta forma, es posible integrar el Core TDC como un periférico más del sistema para que MicroBlaze lo gestione y se comunique con él mediante una FIFO o mediante la lectura/escritura de registros. En la Figura 17 se muestra una captura del programa XPS empleado para la generación de MicroBlaze y el resto de periféricos.



**Figura 17.** Entorno de programación del hardware de la FPGA con XPS.

**Fuente:** elaboración propia.

## PlanAhead

Esta herramienta se ha empleado para analizar la distribución interna del hardware implementado en la FPGA, así como el reparto de recursos. A partir de este análisis es posible determinar la posición de las líneas de retardo para verificar que se implementan correctamente siguiendo las directrices que se han descrito en el capítulo anterior. Asimismo, esta herramienta permite realizar reconfiguración parcial, modificando los bloques que se deseen sin necesidad de volver a generar el hardware completamente. En la Figura 18 se muestra el diseño de las líneas de retardo implementadas en la FPGA mediante PlanAhead.



Una vez expuestas las herramientas de programación y las características tanto de la FPGA a emplear como del microprocesador MicroBlaze así como de las herramientas de programación de la Xilinx ISE Design Suite, se puede sintetizar todo el proceso de la siguiente forma: el Core TDC genera en tiempo real una etiqueta temporal con la información del TOF para cada señal recibida y lo almacena en unos registros. MicroBlaze accede a esta información y la procesa adecuadamente para obtener imágenes biomédicas de mayor calidad. Esta estructura permite que el Core TDC y el microprocesador trabajen de manera asíncrona, dado que las señales cuyo TOF hay que determinar pueden ser recibidas en intervalos de tiempo menores que el periodo del reloj de la FPGA. Además, este planteamiento permite que todo el procesamiento de la información lo realice MicroBlaze, que se programa en lenguaje C utilizando el entorno SDK, lo que facilita enormemente la tarea de programación y depuración software. Esto permite valorar qué fragmentos de código se programan en lenguaje de alto nivel y cuáles en VHDL según el grado de complejidad que requiera el algoritmo a implementar. Un ejemplo de esta versatilidad en la elección del tipo de lenguaje de programación se produce en la generación del mapa de coincidencias que se emplea para conocer si se ha producido un evento de interés, ya que dicho algoritmo es menos complejo si se realiza en lenguaje C que si se programa en VHDL.

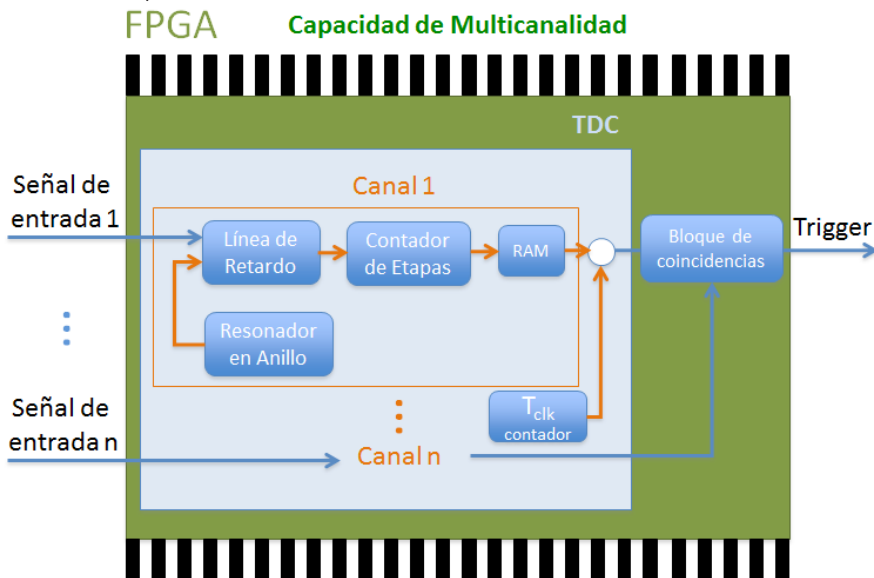
*Implementación de un servidor web para gestionar un core time-to-digital-converter multicanal empleado en sistemas médicos PET*

## CAPÍTULO 5: IMPLEMENTACIÓN DEL CORE TDC

En este capítulo se va a describir el funcionamiento y arquitectura interna del Core TDC que será interconectado al microprocesador MicroBlaze para que éste gestione la información. En los siguientes puntos se va a exponer el flujo de datos que existe desde que se reciben los eventos del anillo de detectores hasta que se envían al entorno de visualización, identificando los elementos que forman parte de la arquitectura del sistema TDC y los procesos que realiza el Core internamente.

### 5.1. Arquitectura del sistema TDC

En los siguientes subapartados se van a describir los tres procesos por los que se somete la información desde que se reciben los eventos hasta que se envían a la aplicación de visualización, así como de los componentes que integran el Core TDC. En Figura 20 se muestra un esquema básico de los bloques de la tarjeta de trigger y del Core TDC implementado en la misma.



**Figura 20.** Esquema conceptual de la implementación del TDC.

**Fuente:** elaboración propia.

#### 5.1.1. Core TDC

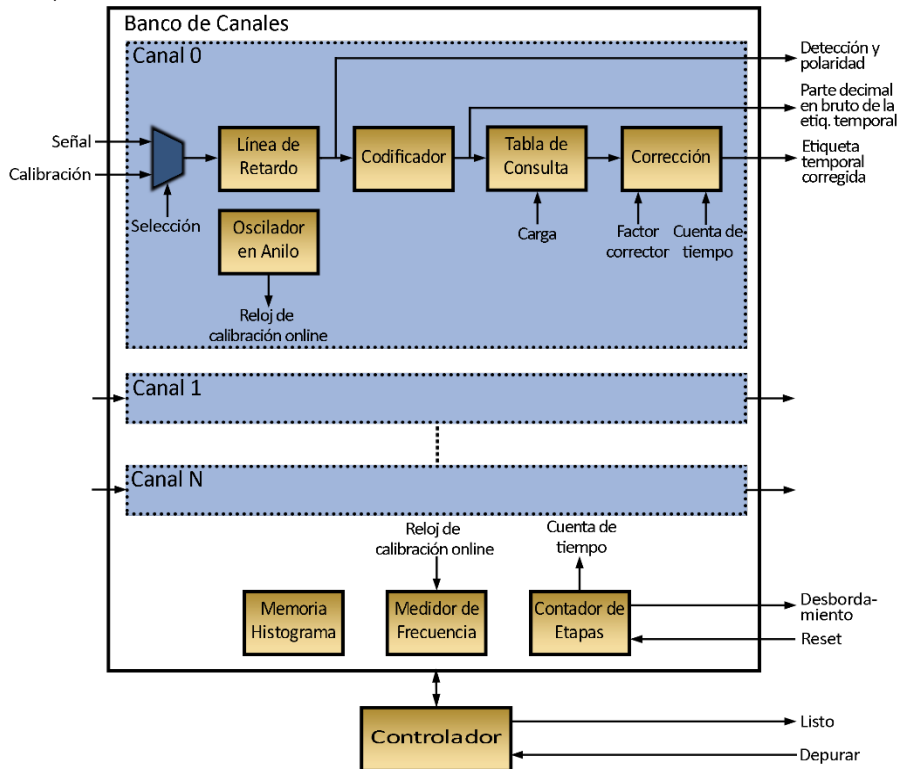
En cuanto a la arquitectura interna del Core TDC, cabe prestar especial atención a los dos bloques que son fundamentales para conseguir resoluciones temporales tan precisas: línea de retardo y proceso de calibrado. No obstante, existen otros bloques internos que se describen a continuación:

- Controlador: Se trata de una máquina de estados que controla todos los procesos para la detección y etiquetado temporal de los eventos. En esta máquina de estados se realizan los siguientes procesos: calibrado, detección de eventos, cálculo de la etiqueta temporal, activación del modo de depuración y selección del canal de entrada.
- Multiplexor de entrada: Se encarga de conmutar su entrada para proporcionar al Core la señal para realizar el calibrado o para realizar el procesado de la misma cuando sea necesaria.
- Línea de Retardo: Se basa en la inyección de la señal en la línea de retardo formada por un conjunto de biestables correctamente dispuestos en cascada de forma alineada. Mediante el flanco de subida de la señal de disparo se indica que la señal se inyecta al comienzo de la línea. Cuando se produce el flanco de bajada de la señal de disparo se determina el número de biestables que han cambiado de estado, es decir, por los que ha transcurrido la señal. De este modo, es posible conocer el retardo total y marcar la señal con una resolución temporal del orden de pico segundos.
- Oscilador en Anillo: Se emplea para compensar las variaciones de la tensión de alimentación y/o temperatura (del inglés, *Process-Voltage-Temperature* o PVT) que pueden producir error en la medida de tiempos. Este elemento genera una señal de frecuencia determinada por la señal de entrada, que permite obtener un factor corrector con el objetivo de perder la influencia de estas dos variaciones en el etiquetado temporal.
- Medidor de frecuencia: Este componente se encarga de calcular la frecuencia del Oscilador en Anillo con el objetivo de realizar la compensación en tiempo real que permite eliminar la dependencia de variaciones de tensión de alimentación y temperatura.
- Codificador: Se encarga de contar el número de estados altos y bajos acumulados en la línea de retardo por lo que se puede conocer por cuántas etapas de retardo ha transcurrido la señal de entrada dentro del intervalo de la señal de disparo. Mediante la calibración, es posible calcular el valor de cada etapa de retardo de modo que se puede determinar la parte decimal del valor de la etiqueta temporal.
- Tabla de consulta (LUT): En este componente se almacenan los valores de los retardos asociados a cada etapa o *bin* de la línea de retardo, por lo que, gracias a esta tabla y los valores que proporciona el Codificador, es posible obtener la parte decimal comentada en el componente anterior.
- Contador de etapas: Este componente se encarga de contar los flancos de subida y bajada de la señal a medir de modo que su valor proporciona la parte entera de la etiqueta temporal.



- Corrección: Su función es sumar la parte entera calculada por el Contador de Etapas o valor bruto, la parte decimal calculada gracias al Codificador y la Tabla de Consulta más un factor corrector, de forma que es posible determinar la etiqueta temporal del evento.
- Memoria de histograma: Se trata de una memoria que guarda el valor de los histogramas de calibración generados con el que se puede calcular el retardo asociado de cada etapa de retardo y obtener la parte decimal.

En la Figura 21 se muestra la arquitectura interna del Core TDC y pueden localizarse los componentes mencionados anteriormente



**Figura 21.** Arquitectura del Core TDC.

**Fuente:** elaboración propia.

### 5.1.2. Core TDC

Dada la gran cantidad de información y las velocidades a las cuales se trabaja, se necesita un bloque de lectura/escritura dentro del periférico TDC para que realice la comunicación del mismo con el microprocesador MicroBlaze. Esto se puede implementar mediante el empleo de un bloque FIFO (del inglés, First in, first out) o mediante registros. En el sistema actual se ha empleado el método de lectura/escritura por registros debido a que cada uno de los canales de entrada al Core TDC proporciona una salida de datos de etiquetas temporales o timestamps por lo que

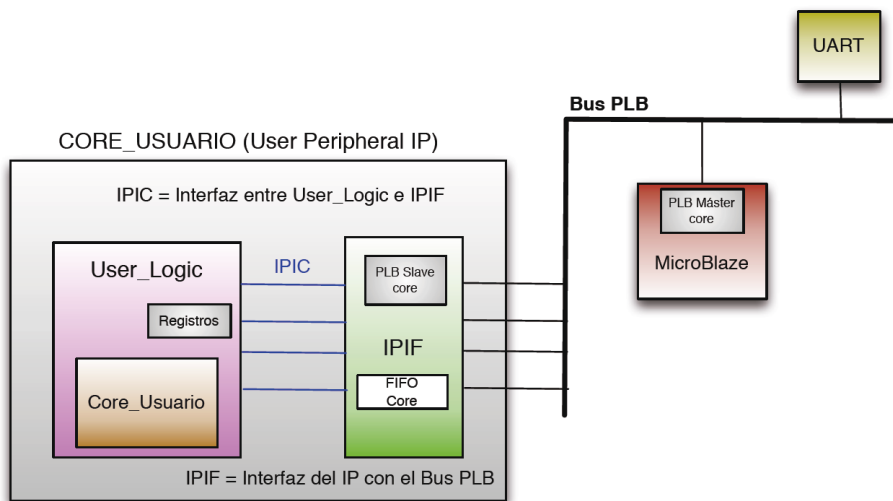
se puede asignar a cada una de estas salidas un registro independiente agilizando y facilitando la lectura de datos desde MicroBlaze. Este método permite que MicroBlaze pueda acceder en todo momento a los datos generados de forma asíncrona por el Core TDC y pueda enviarlos al entorno de visualización.

### 5.1.3. Core TDC

MicroBlaze aporta al sistema mayor versatilidad puesto que se programa con lenguaje de alto nivel y no con lenguaje de descripción hardware VHDL como ocurre con el Core TDC. Esto permite que MicroBlaze, además de realizar las tareas de gestión de datos recibidos del Core TDC y el envío de los mismos a través de sus periféricos de comunicación, pueda aportarle otras funcionalidades al sistema como la implementación de un Servidor Web mediante el empleo del sistema operativo Xilkernel embebido en la nueva librería de comunicaciones lwIP (del inglés, Lightweight TCP/IP) internamente.

## 5.2. Interconexión del Core TDC con MicroBlaze

El Core TDC integrado como periférico de MicroBlaze en la FPGA, se comunicará a través del bus PLB, pero para poder realizarlo correctamente emplea el protocolo simplificado IPIC (del inglés, IP Interconnect) que permite acceder al bus PLB de forma más sencilla empleando el interfaz IPIF (del inglés, Intellectual Property Interface). Es importante mencionar que el programador dispone de dos interfaces para comunicar MicroBlaze con un periférico: mediante la lectura/escritura de registros y/o mediante el empleo de una FIFO. No obstante, en el presente trabajo, como se ha mencionado anteriormente, se va a emplear la comunicación mediante registros. En la Figura 22 se ilustra el diagrama de bloques de la conexión de un Core genérico con el microprocesador MicroBlaze.



**Figura 22.** Comunicación entre MicroBlaze y un periférico o Core de usuario.

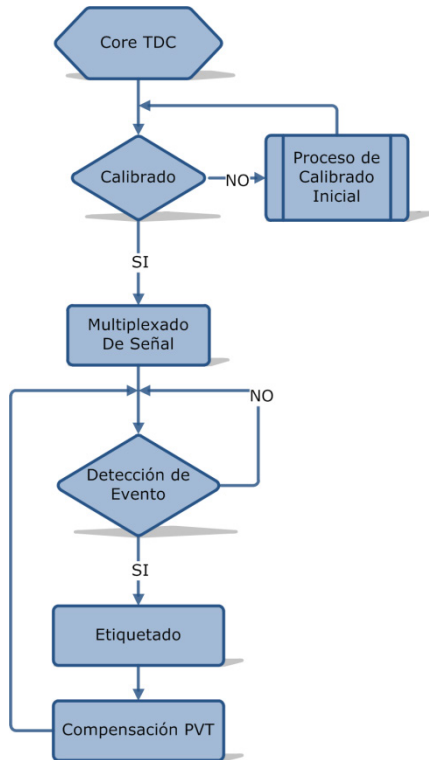
**Fuente:** elaboración propia.

### 5.3. Principio de funcionamiento y elementos del TDC

El TDC se integrará como Core o periférico en la FPGA realizando las funciones ya comentadas y, del mismo modo, el dispositivo FPGA pasará a formar parte de la tarjeta de trigger dentro del sistema electrónico del PET. Por ende, el TDC se encarga de detectar coincidencias y de enviarlas a MicroBlaze, el cual recupera la etiqueta temporal y el número de canal de los registros. A continuación, se aplica el algoritmo adecuado con el fin de determinar si se ha producido una coincidencia, referenciando la diferencia de tiempo con la marca de tiempo anterior y aplicando el mapa coincidencia detector. En ese caso se activa la señal de disparo y también se proporciona información adicional sobre la coincidencia.

Dentro de este orden de ideas, se expone el proceso de desarrollo e implementación del TDC de alta resolución temporal que realiza las siguientes funciones:

- La resolución temporal requerida para obtener una mejora significativa de la calidad de imagen debe ser inferior a 100 ps.
- Debe de detectar aniquilaciones a partir de ventanas temporales y de un mapeado de detectores configurable.
- Generar información temporal de los eventos (TOF) que pueda ser incorporada a los algoritmos de reconstrucción para mejorar la calidad de imagen de los sistemas PET.
- Debe de ser capaz de adquirir valores de varios canales o detectores para poder ser aplicado a un sistema PET real.
- Para desarrollar estas funciones, el sistema TDC realiza una serie de procesos en un determinado orden representados en el siguiente diagrama de flujo (Figura 23).



**Figura 23.** Diagrama de flujo del funcionamiento del Core TDC.

**Fuente:** elaboración propia.

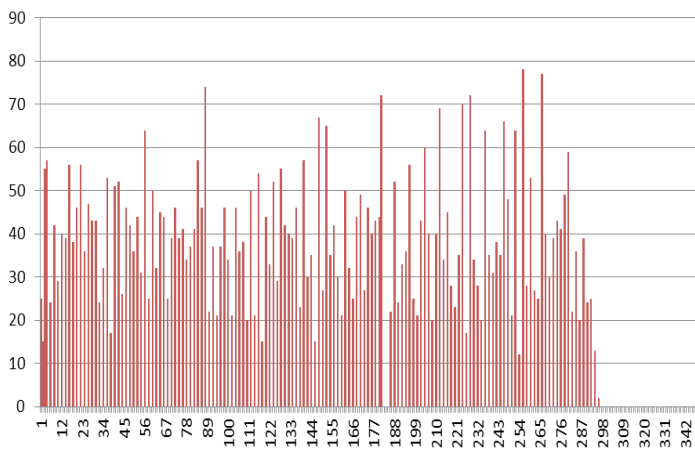
A partir de este diagrama, se puede realizar una breve descripción de los distintos procesos que realiza el Core TDC:

1. Primeramente, el sistema detecta si se ha concluido correctamente el proceso de calibración inicial. En ese caso pasa a la siguiente etapa, si el sistema no está calibrado, comienza el proceso de calibración.
2. Una vez calibrado el sistema, se realiza el multiplexado en la entrada y comienza la medición de la señal de entrada.
3. A continuación, el sistema permanece en espera hasta la recepción de un nuevo evento transmitido desde el anillo de detectores.
4. Si se recibe un evento se indica y se procede a la generación y escritura en el registro correspondiente del *timestamp*.
5. Finalmente, se realiza el proceso de compensación de variaciones de tensión y temperatura y el sistema vuelve a la situación de espera hasta que se reciba un nuevo evento. entorno de visualización.

### 5.3.1 Core TDC

Una vez implementado este sistema en la Spartan-6 de Xilinx, se aprecia que el retardo de los bins no es constante, dado que depende no sólo del retardo propio de cada bin o etapa de retardo sino también de la interconexión entre ellos. Debido a que el retardo de los elementos de la línea no es homogéneo, es necesario realizar un proceso de calibrado inicial. Para realizar la calibración, se procede a la inyección al sistema de señales en instantes de tiempo aleatorio, y a la realización de un histograma que recoja el número de eventos detectado a la salida de cada uno de los bins. La altura de las columnas de este histograma proporciona el valor del retraso temporal que introduce cada uno de los bins, incluyendo las interconexiones entre ellos. Estos valores se almacenan en una LUT, de modo que para realizar la medida del TOF de una señal, además de contar el número de ciclos de reloj que han tenido lugar desde un origen de tiempos arbitrario, basta con leer las salidas de cada uno de los bins y buscar en la LUT el valor temporal equivalente.

En la Figura 24 se muestra el histograma de calibración. Dado que el periodo de la señal de reloj es de 5 ns (correspondiente a 200 MHz) se aprecia como el retardo total acumulado hasta el bin 296 corresponde con ese valor de 5 ns. A partir de esta información, se puede calcular el retardo de cada bin dada su altura en el histograma.



**Figura 24.** Histograma de calibración (número de muestras vs número de bin).

**Fuente:** elaboración propia.

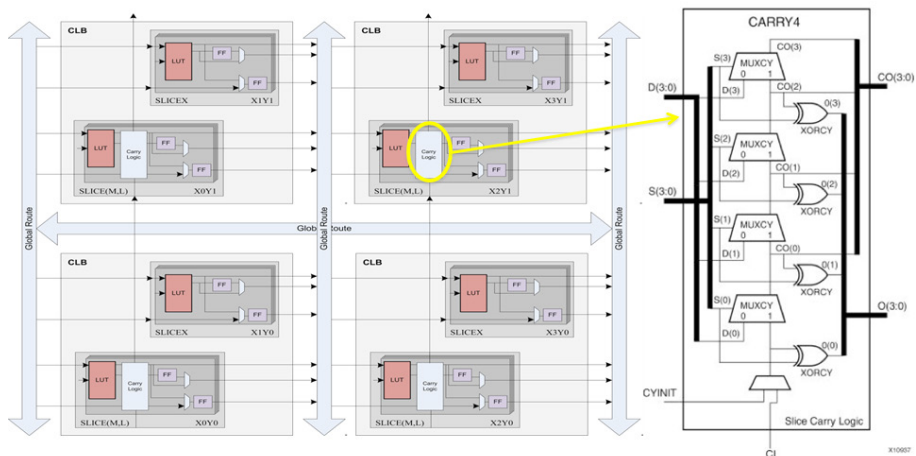
### 5.3.2 Línea de retardo

Uno de los motivos por los que se ha decidido emplear una FPGA de la familia Spartan-6 es porque su estructura interna está basada en primitivas CARRY4, es decir, los bloques de lógica de acarreo que se van a utilizar para poder cuantificar pequeños retardos temporales. Asimismo, permiten integrar el TDC como Core en su interior y emplear los bloques CARRY4 para programar las líneas de retardo mediante las cuales es posible obtener la resolución temporal requerida.

En la Figura 25 se ilustra un esquema de la lógica interna de la primitiva CARRY4. Para su correcto funcionamiento como línea de retardo, las entradas "S" están fijadas a

nivel alto, por lo que la señal inyectada en CYINIT al principio de la línea de retardo atraviesa los elementos MUXCY sin ningún cambio. Como cada primitiva CARRY4 contiene 4 elementos MUXCY, la línea de retardo tiene 4 bloques de retardo (bins) por cada primitiva CARRY4 que se utilice.

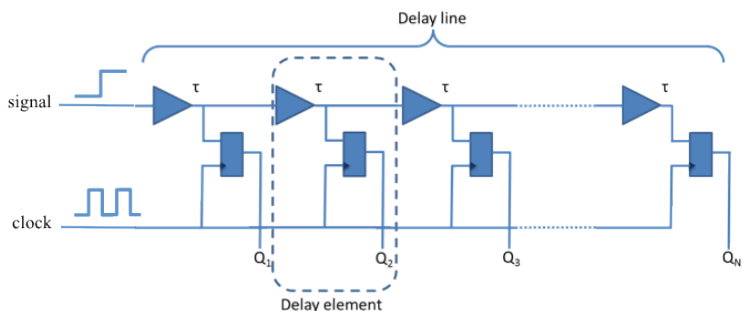
Adicionalmente, en la Figura 25 se muestra también, la estructura de los CLBs (del inglés, Configurable Logic Blocks) que contienen los diferentes tipos de slices que existen en esta FPGA. Puede apreciarse como no todos los tipos de slices contienen las unidades lógicas de acarreo, y también se observa como estas unidades se encuentran interconectadas de manera directa entre ellas para poder realizar su función de manera eficiente.



**Figura 25.** Estructura interna de los CLBs y de la primitiva CARRY4 de la FPGA.

**Fuente:** elaboración propia.

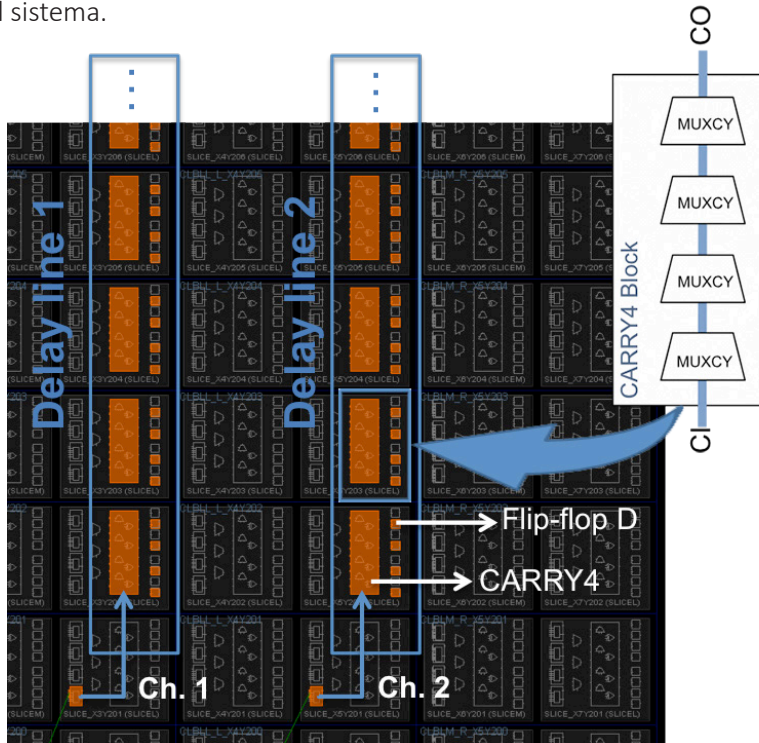
La línea de retardo formada por estos elementos de retardo se muestra en la Figura 26. Esta línea debe tener un retardo total superior al periodo del reloj. De este modo, la señal de reloj se inyecta en la línea y cuando se recibe una señal de la cual se quiere determinar su TOF, el sistema lee el valor de todas las salidas CO de la línea de retardo. Esto permite determinar el tiempo de llegada de la señal con una resolución menor que la del periodo del reloj, y que depende del retardo temporal introducido por los elementos retardadores.



**Figura 26.** Esquema de la línea de retardo.

**Fuente:** elaboración propia.

En la Figura 27 se puede observar la estructura de la línea de retardo que se ha empleado para integrar el TDC de la presente contribución, basada en la línea de Vernier modificada, para optimizar y medir tiempos inferiores a la frecuencia del reloj del sistema. Para aplicar la estructura de Vernier se emplean dos líneas para realizar la resta entre los retardos de cada una de ellas, de modo que, se pueden medir diferencias de tiempo muy pequeñas. El punto más importante de esta topología es que el retardo total de cada una de las líneas debe ser mayor o igual al periodo de reloj del sistema.

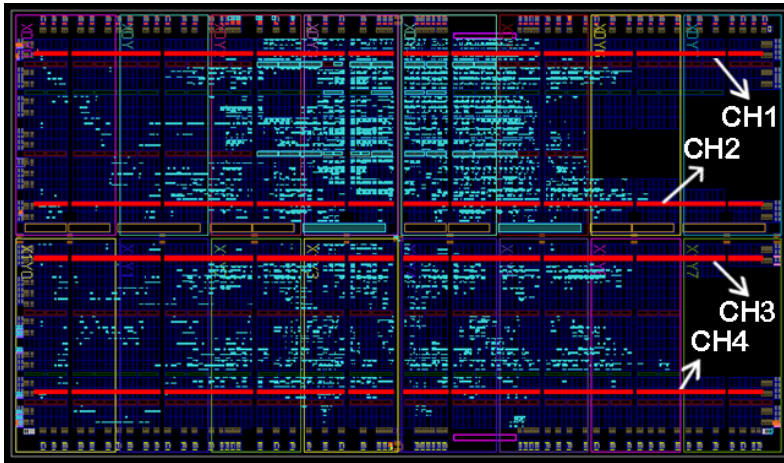


**Figura 27.** Esquema interno de una línea de retardo integrada en la FPGA.

**Fuente:** elaboración propia.

En este trabajo, el software PlanAhead ha resultado especialmente importante en el proceso de implementación de las líneas de retardo en el interior de la FPGA, ya que como se expuso anteriormente, dichas líneas estarán formadas por bloques CARRY4 conectados en cascada. Para que el error introducido sea mínimo se hace necesario que todas las etapas de retardo CARRY4 de una misma línea de retardo estén situados en la misma vertical. Esta tarea no es trivial, puesto que la FPGA Spartan-6 no posee una estructura interna uniforme por lo que se debe realizar un análisis para decidir la posición óptima para cada línea de retardo de cada canal. Una vez se ha decidido en qué parte se sitúan las líneas de retardo, se obtiene la posición exacta de las slices de la FPGA que se van a emplear para incluirlas en el archivo de restricciones UCF (del inglés, User Constraints File) del proyecto generado con el programa XPS comentado anteriormente, para que, al generar el hardware de la FPGA, se tengan en cuenta y se incluyan las líneas de retardo en las posiciones deseadas. En la Figura 28 se muestra la implementación de cuatro líneas de retardo (en color rojo) en la

FPGA de forma que se verifica que se han cumplido las restricciones indicadas en el archivo UCF.



**Figura 28.** Distribución de los canales integrados en la Spartan-6 empleada.

**Fuente:** elaboración propia.

## 5.4. Sumario

Como se desprende de todos los puntos comentados anteriormente, la integración de un Core de estas características en una FPGA es una tarea compleja que requiere del empleo de varias herramientas de desarrollo y programación para conseguir la funcionalidad esperada. De forma más específica, el TDC requiere procesados de un nivel de complejidad muy superior debido a que obtener medidas de tiempos dos órdenes de magnitud inferior a la frecuencia máxima de reloj del sistema no es una tarea trivial.

En este capítulo, se han descrito los componentes y elementos que conforman el Core TDC con el objetivo de dividir el sistema en bloques inferiores y poder facilitar la comprensión del funcionamiento del sistema. A su vez, se ha plasmado un diagrama de flujo de las tareas más importantes que realiza el TDC y se han descrito de forma extendida dos de los elementos más importantes que posee: calibración y línea de retardo.

Por último, es conveniente anotar que el proceso de calibración junto con el empleo de líneas de retardo, permite que el sistema TDC realice medidas temporales inferiores a 100 ps, significando esto, la obtención de imágenes de mayor calidad en sistemas PET. En el capítulo de Pruebas y Resultados se verificará el correcto funcionamiento del TDC, debiéndose obtener, para múltiples canales, resoluciones temporales inferiores a 100 ps, tal y como indicaban las especificaciones iniciales.



## CAPÍTULO 6: DISEÑO FIRMWARE

En los siguientes puntos se va a realizar un estudio del firmware programado a través de la revisión de los diagramas de flujo y, remarcando algunos aspectos importantes del código. No obstante, cabe destacar que todos los códigos de este proyecto se encuentran íntegramente en el DVD adjunto a la memoria.

Dentro de esta perspectiva, se va a prestar especial atención a la lectura y envío de los datos que proporciona el Core TDC y a la integración de un Servidor Web que permite validar el correcto funcionamiento de la comunicación mediante Ethernet.

### 6.1. Gestión de datos del Core TDC

Como se ha comentado en los capítulos anteriores, el TDC se integra como periférico o Core de usuario de MicroBlaze de forma que la comunicación entre ambos sistemas se realiza mediante registros de lectura/escritura.

La comunicación entre el Core TDC y MicroBlaze se configura mediante un archivo programado en VHDL denominado "user\_logic.vhd" que actúa como interlocutor. Este archivo es uno de los más importantes en el proceso de integración del Core TDC, ya que actúa de interfaz entre el microprocesador y el periférico en tanto que posee la función de indicar los puertos de conexión con los archivos de entidades inferiores. En la arquitectura del user\_logic se desarrolla la conexión del periférico con MicroBlaze, donde se declara el componente mediante la palabra COMPONENT, se declaran las entradas y salidas del periférico empleando un proceso PORTMAP con el que se define la conexión a los puertos y, además, se realiza la conexión del periférico con los registros esclavos necesarios para la comunicación con el microprocesador.

En la Figura 29 se muestra el código del archivo user\_logic se muestran los datos que el Core TDC mapea a los registros que le permiten comunicarse con MicroBlaze.

```

545  -- implement slave model software accessible register(s) read mux
546  SLAVE_REG_READ_PROC : process( slv_reg_read_sel, slv_reg0, slv_reg1,
547  begin
548
549      case slv_reg_read_sel is
550          when "1000000000" => slv_ip2bus_data <= resta_32;
551          when "0100000000" => slv_ip2bus_data <= resta2_32;
552          when "0010000000" => slv_ip2bus_data <= resta3_32;
553          when "0001000000" => slv_ip2bus_data <= ready_o_32;
554          when "0000100000" => slv_ip2bus_data <= reset_i_32;
555          when "0000010000" => slv_ip2bus_data <= slv_reg4;
556          when "0000001000" => slv_ip2bus_data <= his_d_o_32;
557          when "0000000100" => slv_ip2bus_data <= lut_a_i_32;
558          when "0000000010" => slv_ip2bus_data <= lut_d_o_32;
559          when "0000000001" => slv_ip2bus_data <= raw_o_32;
560          when others => slv_ip2bus_data <= (others => '0');
561      end case;
562
563  end process SLAVE_REG_READ_PROC;
564

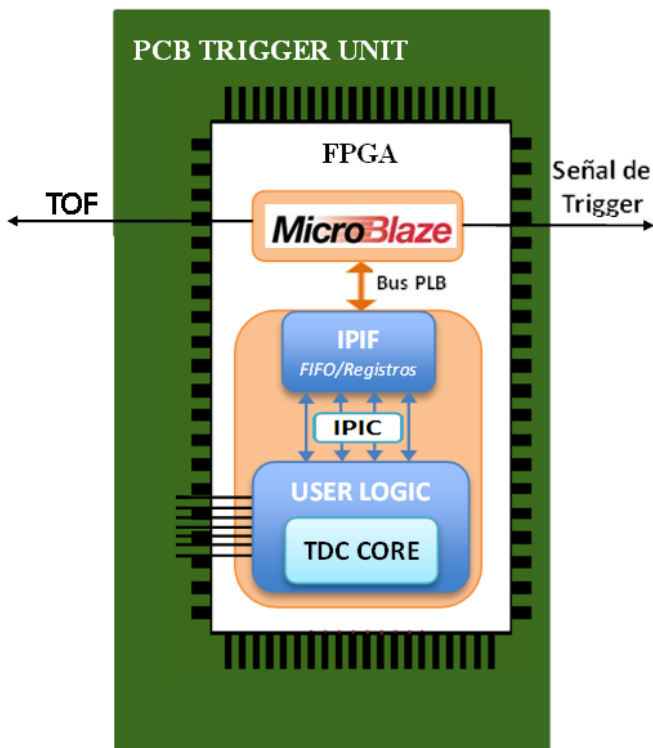
```

**Figura 29.** Fragmento de código de asignación de registros extraído de user\_logic.vhd.

**Fuente:** elaboración propia.

En la versión actual, se emplean 10 registros de comunicación de los que cabe prestar especial atención a `ready_o_32`, ya que informa si el Core ha sido calibrado correctamente y puede comenzar a detectar eventos y realizar timestamps; `reset_i_32`, que permite realizar un reseteo del Core TDC desde MicroBlaze; y los tres primeros registros `restan_32` correspondientes a tres valores de restas temporales realizadas en cuatro canales que se encuentran actualmente implementados en la FPGA. Estos valores de resta son los que MicroBlaze lee y gestiona para que sean enviados al entorno de visualización de datos, ya que proporcionan los valores temporales medidos.

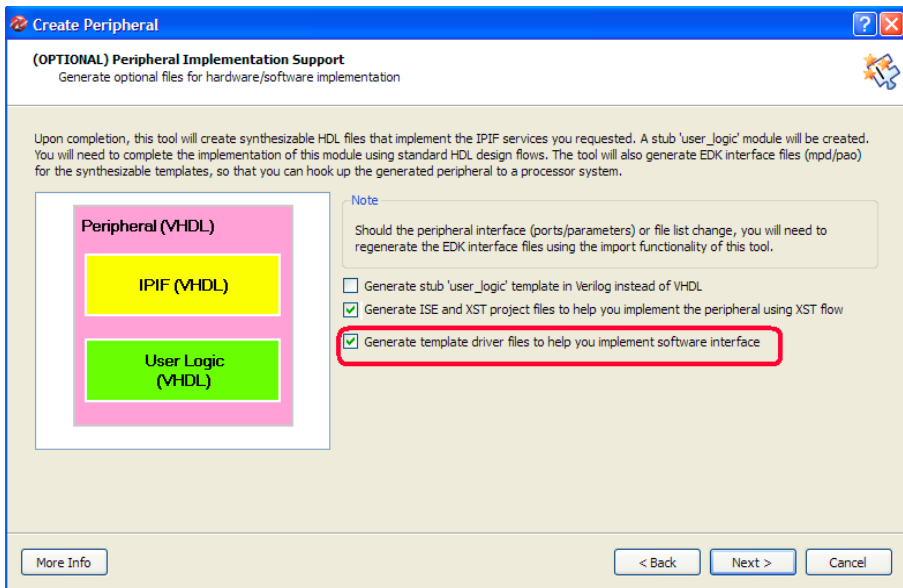
En la Figura 30 se muestra un esquema de la comunicación que se realiza entre MicroBlaze y el Core TDC para compartir los datos. Siguiendo una estructura jerárquica desde la entidad superior que es MicroBlaze hasta el Core TDC, la comunicación principal entre ambos se realiza a través del bus PLB, como la mayoría de periféricos (UART, GPIO, Ethernet...) que se encuentran interconectados con MicroBlaze. El periférico TDC posee una estructura interna basada en la comunicación del Core TDC, que es el que incluye la funcionalidad del sistema de etiquetado temporal, con su entidad superior `user_logic` descrita anteriormente. Para que desde `user_logic` se transmitan los datos del periférico TDC a MicroBlaze, se emplea el interfaz IPIF y el protocolo IPIC para realizar la lectura/escritura de los registros. De este modo, se consigue un canal de comunicación directo entre MicroBlaze con el Core TDC.



**Figura 30.** Esquema de comunicación entre MicroBlaze y Core TDC.

**Fuente:** elaboración propia.

Una vez revisados los datos que el Core TDC escribe en los registros, se va a proceder a la descripción de cómo MicroBlaze accede a dichos datos mediante los registros empleando lenguaje de alto nivel. Para facilitar la programación del driver del periférico TDC, la plataforma XPS crea un test de funcionalidad para cada periférico implementado en MicroBlaze. Dicho test realiza la función de driver o librería, permitiendo controlar el funcionamiento del periférico para dar origen a una aplicación. XPS genera para este propósito funciones que admitan la escritura y la lectura de cada registro del periférico, seleccionando la opción Generate template driver files en el proceso de importación del periférico. En la Figura 31 se muestra esta opción.



**Figura 31.** Opción para generar el driver del Core TDC.

**Fuente:** elaboración propia.

El driver generado toma el nombre del periférico, en este caso CORE\_TDC, y posee una extensión .h, de manera que, es posible incluirla en la compilación del proyecto de firmware y emplear las funciones propias de acceso a los registros del Core TDC como la captura del código de MicroBlaze (Figura 32).

```

/**
 *
 * Write/Read 32 bit value to/from CORE_TDC user logic slave registers.
 *
 * @param BaseAddress is the base address of the CORE_TDC device.
 * @param RegOffset is the offset from the slave register to write to or read from.
 * @param Value is the data written to the register.
 *
 * @return Data is the data from the user logic slave register.
 *
 * @note
 C-style signature:
 */ void CORE_TDC_mWriteSlaveRegn(Xuint32 BaseAddress, unsigned RegOffset, Xuint32 Value)
   Xuint32 CORE_TDC_mReadSlaveRegn(Xuint32 BaseAddress, unsigned RegOffset)

```

**Figura 32.** Funciones de lectura y escritura en los registros del Core TDC.

**Fuente:** elaboración propia.

Las funciones de la captura anterior se describen a continuación:

- ***CORE\_TDC\_mWriteSlaveRegn()*** se emplea para escribir desde MicroBlaze datos en el Core, de modo que es posible realizar un reseteo únicamente del periférico TDC, si se desea, desde el microprocesador. Esto también permite modificar algunos parámetros del TDC como puede ser el ancho de la ventana de coincidencias. Para escribir en un registro determinado, se debe sustituir del prototipo de función expuesta, el carácter “n” por el número del registro al que se quiera acceder. En cuanto a los parámetros que se deben indicar en la función, se debe incluir la dirección base del **Core** TDC, el desplazamiento del registro, que en este caso es nulo, y el valor que se quiere escribir en el mismo. Cabe mencionar, que este valor puede ser como máximo de 32 bits ya que la arquitectura de MicroBlaze no permite una profundidad mayor de palabra de datos. Por último, destacar que esta función de escritura no devuelve ningún valor al ser de tipo **void**.
- ***CORE\_TDC\_mReadSlaveRegn()*** se emplea para realizar la lectura de los registros de modo que es posible recibir los datos de medidas temporales que el **Core** TDC ha adquirido. Esto permite realizar la lectura secuencial de las tres restas temporales que se han mencionado anteriormente, realizar la gestión de estos datos y, a continuación, enviarlos al entorno de visualización. Para leer un registro determinado, se debe sustituir del prototipo de función expuesta, el carácter “n” por el número del registro al que se quiera acceder. En cuanto a los parámetros que se deben indicar en la función, se debe incluir, como ocurre con la función de escritura, la dirección base del **Core** TDC y el desplazamiento del registro. El valor que ha sido leído del registro se devuelve en una variable de 32 bits a la que ha sido igualada la función.

La comunicación con el entorno de visualización se va a realizar de dos formas: mediante comunicación serie y empleando comunicación Ethernet. Para poder enviar los datos a la aplicación de visualización del PC se va a emplear comunicación serie punto a punto entre la FPGA y el PC. Este tipo de comunicación, a pesar de poseer una tasa de datos muy inferior a la de Ethernet y no poder realizar enlaces de más de 5 metros, es muy sencilla de configurar y resulta muy útil para la realización de pruebas del sistema TDC en tiempo real.

En la Figura 33 se muestra una captura de código de un ejemplo simplificado de la metodología que se emplea para realizar el envío de datos mediante comunicación serie. Dentro de un superbucle, se accede primeramente a los datos del Core TDC, y, a continuación, se realiza el envío serie de los mismo, correspondientes a los resultados de las tres restas temporales realizadas. Como puede observarse, para el envío serie se ha empleado la función `xil_printf()` mediante la cual es posible escribir datos en el buffer de salida del periférico de una forma sencilla y más eficaz que empleando la función de lenguaje C `printf()`. Esta función permite enviar el valor en decimal en forma de cadena de texto seguida de un salto de línea, facilitando la tarea

de identificación de paquetes a la aplicación software para PC que se describirá en el siguiente capítulo.

```

while(1)
{
    //Salida Resta
    resta = CORE_TDC_mReadSlaveReg0(XPAR_CORE_TDC_BASEADDR, 0x00000000);
    resta2 = CORE_TDC_mReadSlaveReg1(XPAR_CORE_TDC_BASEADDR, 0x00000000);
    resta3 = CORE_TDC_mReadSlaveReg2(XPAR_CORE_TDC_BASEADDR, 0x00000000);

    //Envío por COM a PC
    xil_printf("%d\n\r", resta);
    xil_printf("%d\n\r", resta2);
    xil_printf("%d\n\r", resta3);
}

```

**Figura 33.** Ejemplo de comunicación serie simplificado entre MicroBlaze y PC.

**Fuente:** elaboración propia.

## 6.2. Implementación del servidor web

Con el objetivo de validar el correcto funcionamiento del Core TDC con una comunicación de alta velocidad, se ha implementado en la FPGA un servidor web que permite enviar datos desde este dispositivo a la Red. De esta forma, se puede tener acceso a los datos y a su visualización de forma remota sin necesidad de realizar una conexión punto a punto con la FPGA como ocurre con la comunicación serie expuesta en el apartado anterior.

Para realizar la implementación del Servidor Web en la FPGA, se ha empleado como base un ejemplo que proporciona el fabricante Digilent adaptado a la placa de desarrollo Atlys que se emplea en el presente proyecto (Digilent, 2013). A partir del análisis de este ejemplo, se consideró migrar el sistema al bus AXI con la finalidad de facilitar la labor de comunicación Ethernet, ya que empleando el bus PLB se producían errores de comunicación.

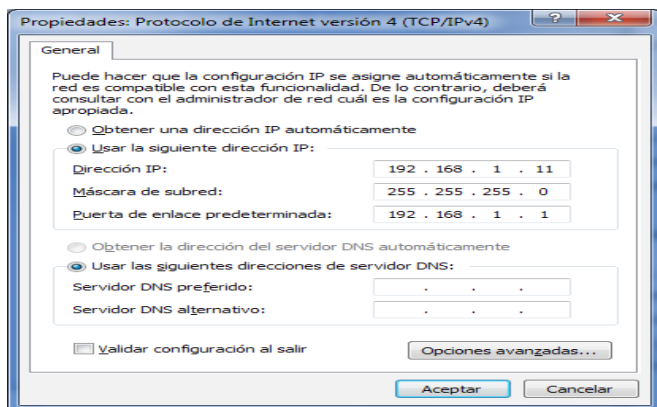
Por consiguiente, se verificó el correcto funcionamiento del Core TDC y MicroBlaze empleando el bus de comunicaciones AXI y, tras ello, se comenzó a realizar la adaptación del Servidor Web al sistema TDC empleando el bus AXI. Con esto, se consigue a su vez migrar el sistema a un bus más moderno y poder aumentar la frecuencia de funcionamiento de MicroBlaze ya que estaba limitada por el bus PLB.

El proyecto de demostración que ofrece como soporte Digilent describe cómo crear un sistema integrado empleando la pila de red TCP/IP Lightweight IP (lwIP) y la librería de gestión de archivos de memoria Xilinx Memory File System (XilMfs), ambos optimizados para ser empleados en los sistemas embebidos de Xilinx PowerPC y MicroBlaze. El archivo demo de servidor web descargado del sitio Web de Digilent está formado por el siguiente sistema de directorios:

- **doc:** contiene la documentación descriptiva del servidor web.
- **memfs:** en ella se encuentra el contenido para generar la imagen MFS (del inglés, *Memory File System*) del sitio web de ejemplo. Esta imagen puede generarse mediante el comando `mfsngen` en la consola de Windows.

- ready for download: en esta carpeta se encuentran los archivos necesarios para realizar una precompilación del proyecto, es decir, el archivo software ejecutable (Atlys\_Webserver\_Demo.elf) y la imagen MFS ya generada con el contenido web de ejemplo (comentado en el ítem anterior) para ser cargada en la FPGA.
- sw: incluye los archivos necesarios para implementar el firmware con la herramienta Xilinx SDK.
- hw: incluye el proyecto con el diseño hardware creado con la herramienta XPS.

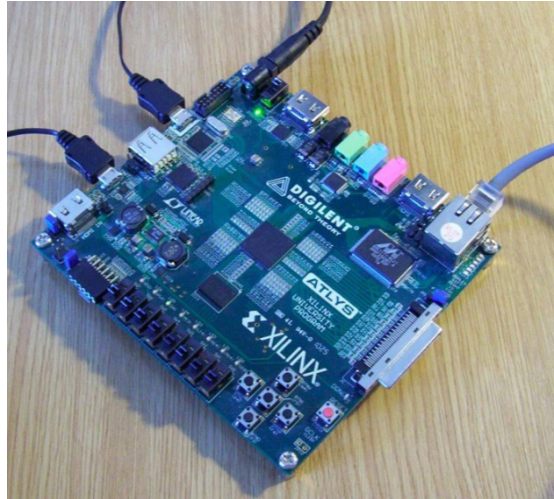
La depuración del servidor web se ha realizado en red local, tal y como aconseja la documentación de Digilent. Para ello, se ha configurado el puerto Ethernet de Windows con una IP fija que pertenezca a la misma subred que el servidor web, de modo que si el servidor por defecto se configura con la IP 192.168.1.10, se debe configurar el puerto del PC con la IP fija para que pertenezca a la misma subred como, por ejemplo, la IP 192.168.1.11. Además de esta configuración, se debe verificar que el adaptador Ethernet del PC se encuentra configurado en modo de negociación automática de velocidad. Esta opción está seleccionada de forma predeterminada en Windows, sin embargo, debe tenerse en cuenta dado que puede ser causa de errores de comunicación con el servidor web. En la Figura 34 se muestra la configuración de la red que se ha realizado en Windows.



**Figura 34.** Configuración de red local en Windows.

**Fuente:** elaboración propia.

Por otro lado, debe verificarse que la placa de desarrollo Atlys se encuentra correctamente configurada y, para ello, se deben conectar al PC tanto el programador para realizar la programación de la FPGA como el puerto UART de comunicación serie para realizar la depuración del código, ya que si se produce un error el servidor web está configurado de forma que lo reporta empleando la UART. Del mismo modo, debe de conectarse el cable Ethernet para establecer con el PC (o con el router) la comunicación de alta velocidad. Por último, se debe verificar que el jumper J11 se encuentra configurado en la posición que proporcione una tensión de alimentación de 5V. En la Figura 35 se muestra el conexionado de la placa de desarrollo Atlys.



**Figura 35.** Configuración de la placa de desarrollo Atlys.

**Fuente:** elaboración propia.

Llegados a este punto, existen dos vías para integrar el servidor web en la FPGA: cargar de forma directa el archivo ELF que alberga la funcionalidad software junto con el archivo bit en el que se encuentra la configuración del hardware; o realizar un análisis del proyecto empezando por el proyecto hardware de XPS prestando especial atención a la configuración del periférico Ethernet y revisar el proyecto software programado en SDK analizando en profundidad las funciones que realizan la comunicación con el sitio web. Puesto que se desea desarrollar un Servidor Web adaptado a las necesidades del presente proyecto se ha procedido a realizar la segunda tarea, es decir, un análisis en profundidad del Servidor Web que proporciona Digilent, para poder comprenderlo correctamente y realizar las modificaciones necesarias con el fin de adaptarlo a las características del sistema TDC.

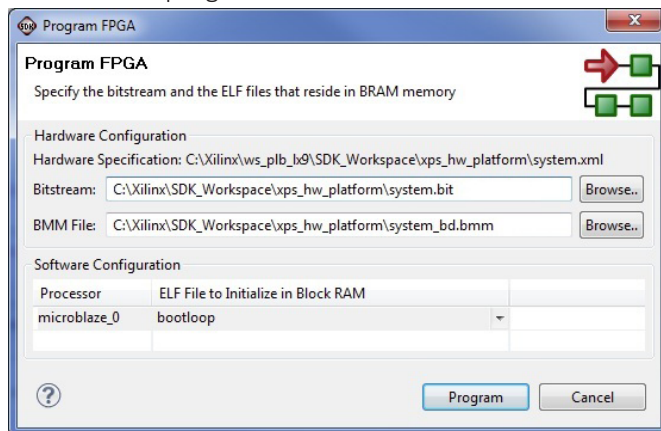
Atendiendo a esta consideración, en primer lugar, se ha estudiado el proyecto hardware de XPS y, además de comprender la configuración del periférico Ethernet que se expondrá en el siguiente subapartado, se han identificado algunas configuraciones que si no se tienen en cuenta para la implementación del hardware pueden generar errores en la comunicación.

De estas configuraciones cabe destacar: el tamaño de memoria caché de datos e instrucciones de 16 KB; la necesidad de activar la interrupción de los periféricos UART y Ethernet; la inclusión en el proyecto hardware de un timer con interrupción habilitada; y realizar algunas modificaciones en la configuración de MicroBlaze para optimizar la velocidad del Servidor Web, como aumentar el tamaño de las líneas de datos e instrucciones.

Una vez analizado el plano hardware, se procede a generar el archivo Bitstream y la exportación del mismo a SDK, para poder modificar algunos parámetros del plano software relacionados con las librerías lwIP y XilMfs. No obstante, debido a la importancia de dicha configuración, se expondrá en los siguientes puntos de manera más detallada.



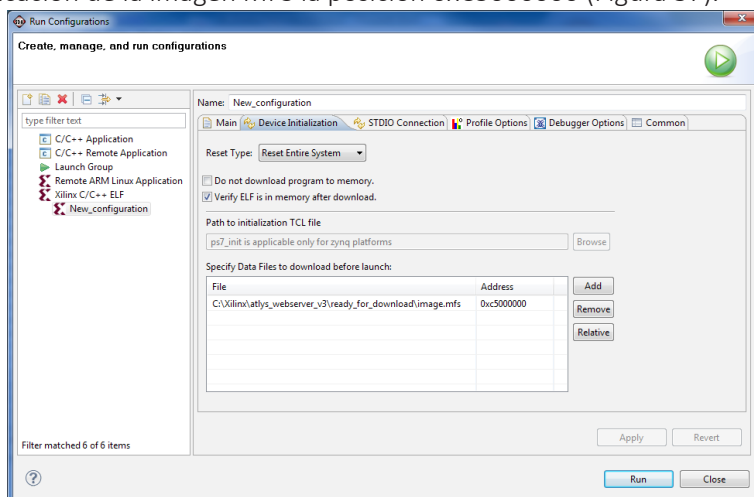
De esta forma, el último paso a realizar es programar la FPGA, y, para ello, se realiza una programación de inicialización del hardware incluyendo los archivos hardware Bitstream y BMM pero dejando en la casilla del archivo ELF la opción por defecto bootloop. En la Figura 36 se muestra una captura de la configuración que se debe seleccionar para realizar la programación.



**Figura 36.** Configuración de preprogramación del Servidor Web.

**Fuente:** elaboración propia.

Una vez preprogramada, se procede a la programación de la aplicación que contiene el Servidor Web mediante la opción Run Configuration. Dentro de esta opción, se crea una nueva configuración de Debug mediante la opción Xilinx C/C++ ELF y en la pestaña Device Initialization se selecciona la ruta en la que se encuentra la imagen MFS que contiene el sitio Web que se desea publicar en el servidor Web y se indica la dirección de memoria en la que se desea ubicar dicha imagen. Esta ubicación suele elegirse a partir de la dirección base de la memoria DDR del sistema añadiéndole un offset, de forma que, si la dirección base de la DDR es 0xC4000000, se selecciona como ubicación de la imagen MFS la posición 0xC5000000 (Figura 37).

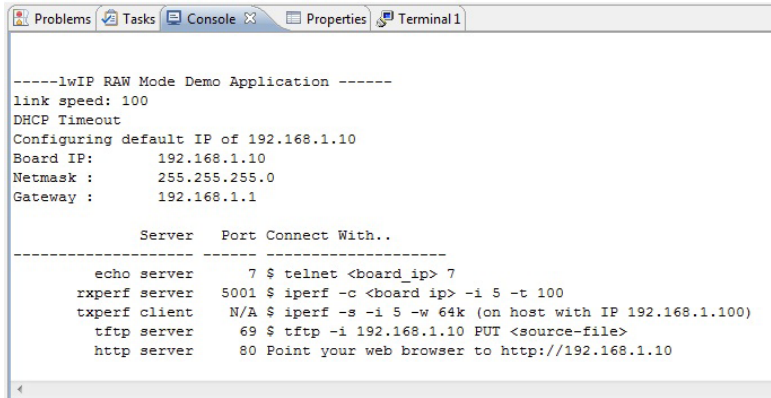


**Figura 37.** Configuración del modo Debug del servidor web.

**Fuente:** elaboración propia.



Tras realizar sendas configuraciones, en la consola de la herramienta de programación SDK se puede observar el proceso de integración del proyecto en la FPGA, y una vez finalizado, en la pestaña del terminal que refleja todos los datos recibidos por comunicación serie, debe de aparecer el siguiente mensaje emitido por el servidor web indicando que se ha implementado correctamente, como muestra la Figura 38.



```

-----lwIP RAW Mode Demo Application -----
link speed: 100
DHCP Timeout
Configuring default IP of 192.168.1.10
Board IP:      192.168.1.10
Netmask :     255.255.255.0
Gateway :     192.168.1.1

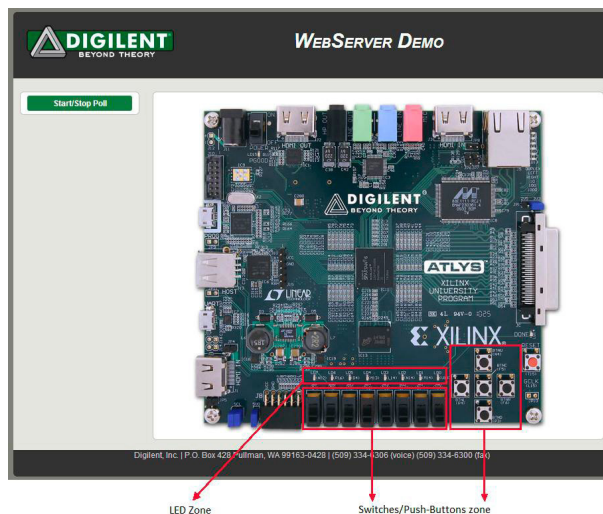
      Server  Port Connect With..
-----
echo server  7 $ telnet <board_ip> 7
rxperf server 5001 $ iperf -c <board_ip> -i 5 -t 100
txperf client N/A $ iperf -s -i 5 -w 64k (on host with IP 192.168.1.100)
tftp server  69 $ tftp -i 192.168.1.10 PUT <source-file>
http server  80 Point your web browser to http://192.168.1.10

```

**Figura 38.** Reporte emitido por el servidor web indicando que está funcionando correctamente.

**Fuente:** elaboración propia.

El reporte emitido mediante comunicación serie por el servidor web indica que se encuentra funcionando correctamente por lo que, si se accede a su IP local desde un navegador, se puede acceder al sitio web demo que está precargado en la imagen MFS que proporciona Digilent. En la Figura 39 se muestra la apariencia de este sitio.



**Figura 39.** Sitio web demo de Digilent.

**Fuente:** elaboración propia.

Este sitio web permite interactuar de forma dinámica con la placa de desarrollo en tanto que si se hace clic en cualquiera de los LEDs, automáticamente cambia el estado del mismo tanto en el sitio Web como en la propia placa de desarrollo. Por otro lado, si se modifica el estado de los interruptores o pulsadores de la placa, dicho cambio se plasma en el sitio Web.

Por ende, este ejemplo proporcionado por Digilent va a sentar las bases del Servidor Web que se va a embeber en la FPGA, ya que incorpora todas las funcionalidades necesarias de forma que, modificando algunos aspectos de la comunicación, integrando el sistema TDC, y modificando el aspecto del sitio Web, es posible obtener un entorno de visualización remoto de los datos temporales que proporciona el TDC.

### **6.2.1. Línea de retardo**

Ethernet es un estándar de redes de área local para computadores que realiza el acceso al medio mediante CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones), un protocolo de acceso múltiple basado en la monitorización de la portadora, empleado en este tipo de redes para mejorar sus prestaciones. Ethernet se basa en el estándar internacional IEEE 802.3 y define las características tanto de cableado como de señalización de nivel físico, así como de los formatos de tramas de datos del nivel de enlace del modelo de capas OSI.

Ethernet mejora en gran medida la comunicación del sistema TDC con el PC, ya que le aporta inteligencia en la red en cuanto a que es capaz de detectar colisiones y de realizar una gestión muy eficiente en términos de conexión de múltiples equipos a la red, es decir, que quieran acceder al entorno de visualización que se ha integrado en la FPGA.

En este sentido, con este protocolo cualquier equipo está autorizado a transmitir a través de la línea en cualquier momento y sin ninguna prioridad entre ellos. Esta comunicación se realiza de manera simple como se resume a continuación:

- Cada equipo verifica que no exista ninguna comunicación en la línea antes de transmitir.
- Si dos equipos transmiten simultáneamente se produce una colisión, es decir, varias tramas de datos se ubican en la línea al mismo tiempo.
- Cuando esto sucede, los dos equipos interrumpen su comunicación y esperan un período de tiempo aleatorio. Una vez que el primero ha excedido el período de tiempo, puede reintentar la transmisión.

En la Tabla 2 se muestran las diferentes variantes que ofrece la tecnología Ethernet según el tipo de cable y las características de velocidad y distancia que se pueden conseguir para cada una de ellas.

Como se desprende de la tabla anterior, existen diferentes velocidades y distancias en función de la variante y tipo de cableado que se emplee. En el caso del transceiver que posee la placa Atlys, es posible realizar comunicaciones de hasta 1Gb/s, sin embargo, para conseguir esa velocidad, es necesario emplear el periférico Trimode Ethernet

Interface que requiere una configuración muy compleja y una programación de las restricciones de ruteado interno de la FPGA muy minuciosa que podría perjudicar a las líneas de retardo implementadas en el hardware interno, por lo que se va a emplear el periférico Ethernet Lite mediante bus AXI, ya que es la configuración que se aconseja en la documentación del servidor web proporcionado por Digilent. Con este periférico, la máxima velocidad que puede conseguir la comunicación es de 100 Mb/s, por lo que podrían alcanzarse distancias de hasta 100 m desde la posición en la que se encuentre la FPGA hasta el punto de conexión a la red.

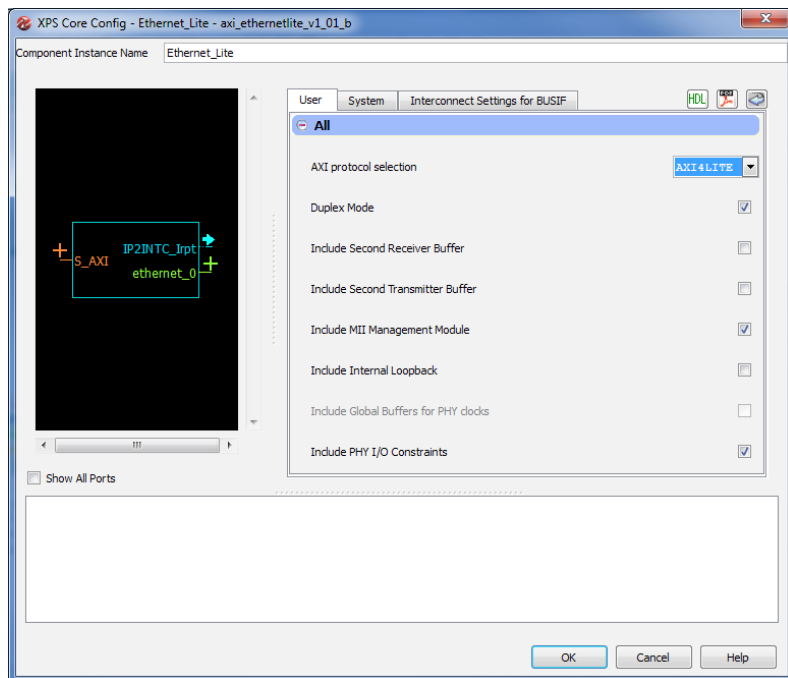
**Tabla 2.** Comparación de las diferentes variantes de Ethernet.

Tipo	Nombre	Cable	Velocidad	Distancia
10Base-T	Ethernet estándar	Par trenzado (cat. 3)	10 Mb/s	100 m
100Base-TX	Ethernet veloz (Fast Ethernet)	Doble par trenzado (UTP cat. 5)	100 Mb/s	100 m
100Base-FX	Ethernet veloz (Fast Ethernet)	Fibra óptica multimodo	100 Mb/s	2 km
1000Base-T	Ethernet Gigabit	Doble par trenzado (UTP cat. 5)	1000 Mb/s	100 m
1000Base-LX	Ethernet Gigabit	Fibra óptica monomodo o multimodo	1000 Mb/s	550 m
10GBase-SR	Ethernet de 10 Gigabits	Fibra óptica multimodo	10 Gb/s	550 m
10GBase-LX4	Ethernet de 10 Gigabits	Fibra óptica multimodo	10 Gb/s	550 m

**Fuente:** elaboración propia.

Para determinar la velocidad del enlace, el servidor web realiza un proceso de autonegociación con el punto de enlace de modo que los adaptadores o conmutadores detectan la velocidad de transmisión automáticamente y utilizan el método más rápido disponible. Esta negociación se realiza utilizando una ráfaga FLP (Pulsos Rápidos de Enlace) que son ignorados por los adaptadores 10BaseT pero respondidos por los adaptadores 100BaseT, pudiendo determinar la velocidad de comunicación máxima.

En cuanto a la configuración del periférico Ethernet Lite de XPS, cabe destacar que se ha conectado al bus AXI4LITE general del sistema y se configura en modo dúplex para que el sistema sea capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea. Otra opción que se ha configurado es la conocida como MII, (del inglés, Media Independent Interface) la interfaz que emplea la capa MAC para comunicarse con la física empleando un enlace de 4 bits a 25 MHz de forma paralela síncrona. Por último, se selecciona la opción Include PHY I/O Constraints con el objetivo de que XPS genere automáticamente las restricciones necesarias para que la comunicación Ethernet funcione de la forma más óptima posible (Figura 40).



**Figura 40.** Configuración del periférico Ethernet Lite.

**Fuente:** elaboración propia.

### 6.2.2. Programación del servidor web

Llegados a este punto, se van a describir los fragmentos de código más relevantes de los archivos que se han empleado para realizar la integración del servidor web en MicroBlaze para comprender su funcionalidad.

El servidor web implementa un subconjunto del protocolo HTTP 1.1. con el objetivo de que posea las funcionalidades más importantes y pueda realizar tareas de control y monitorización a través de un navegador. De este modo, se optimiza el tamaño del proyecto sin perder algunas funcionalidades básicas como:

- Acceso a los archivos ubicados en la memoria DDR interna a través de comandos HTTP GET.
- Control de parámetros de la FPGA a través del navegador empleando el comando HTTP POST.

El control de los datos se realiza mediante la emisión de órdenes de entrada empleando un conjunto de direcciones URL mapeadas a unas determinadas funciones. Cuando el Servidor recibe un comando POST de una URL conocida, éste llama a la función específica correspondiente a la petición realizada. La salida de esta función se retorna al navegador web en formato JavaScript para que dicho navegador pueda interpretar los datos recibidos, actualizarlos y mostrarlos por pantalla para que sean visualizados.

La estructura general del servidor web se realiza mediante un hilo que se encuentra constantemente comprobando si se reciben nuevas entradas por el puerto HTTP 80, de forma que para cada conexión entrante que se recibe por este puerto, se genera un nuevo hilo que procesa las solicitudes.

El hilo HTTP del puerto 80, primero lee las peticiones, después identifica si se trata de una operación GET o POST y seguidamente realiza la operación correspondiente. Cuando se produce una petición GET, el hilo busca un archivo específico en el sistema de archivos de memoria interna, si el archivo existe el contenido del mismo se devuelve al navegador. Si, por el contrario, el archivo no existe o no se encuentra disponible, se genera un código de error HTTP 404 que se reenvía al navegador.

Para atender las peticiones, el servidor web implementa un sistema de funciones callback de modo que, cuando se acepta una nueva conexión la función `accept_callback()` configura las callbacks de envío y recepción. Estas funciones se llaman cuando se recibe una trama ACK (acknowledged) o cuando se recibe un dato. Esta función se muestra en la Figura 41.

```
err_t accept_callback(void *arg, struct tcp_pcb *newpcb, err_t err)
{
    /* keep a count of connection # */
    tcp_arg(newpcb, (void*)palloc_arg());

    tcp_recv(newpcb, recv_callback);
    tcp_sent(newpcb, sent_callback);

    return ERR_OK;
}
```

**Figura 41.** Rutina de control de callbacks.

**Fuente:** elaboración propia.

Cuando un navegador solicita una página web al servidor, éste llama la función `recv_callback` que realiza la decodificación de la petición y el envío de la respuesta apropiada. En la Figura 42 se muestra el fragmento de código de envío de respuesta.

```
/* acknowledge that we have read the payload */
tcp_recved(tpcb, p->len);

/* read and decipher the request */
/* this function takes care of generating a request, sending it,
 * and closing the connection if all data has been sent. If
 * not, then it sets up the appropriate arguments to the sent
 * callback handler.
 */
generate_response(tpcb, p->payload, p->len);

/* free received packet */
pbuf_free(p);
```

**Figura 42.** Rutina de control de `recv_callback`.

**Fuente:** elaboración propia.

Por tanto, la aplicación determina la cantidad de datos que pueden ser enviados de forma que son optimizados y repartidos en varios paquetes de modo que se ocupe todo el buffer en cada trama de envío. En el momento que el receptor o cliente Web realiza la lectura de los datos el buffer se libera. Cuando esto ocurre, lwIP llama a la función `sent_callback()`, indicando qué datos han sido enviados y la existencia de espacio en el buffer para poder rellenarlo con nuevos datos. La función `sent_callback` se estructura como muestra la Figura 43.

```
err_t sent_callback(void *arg, struct tcp_pcb *tpcb, u16_t len)
{
    int BUFSIZE = 1024, sndbuf, n;
    char buf[BUFSIZE];
    http_arg *a = (http_arg*)arg;

    /* if connection is closed, or there is no data to send */
    if (tpcb->state > ESTABLISHED) {
        return ERR_OK;
    }

    /* read more data out of the file and send it */
    sndbuf = tcp_sndbuf(tpcb);
    if (sndbuf < BUFSIZE)
        return ERR_OK;

    n = mfs_file_read(a->fd, buf, BUFSIZE);
    tcp_write(tpcb, buf, n, 1);

    /* update data structure indicating how many bytes
     * are left to be sent
     */
    a->fsize -= n;
    if (a->fsize == 0) {
        mfs_file_close(a->fd);
        a->fd = 0;
    }

    return ERR_OK;
}
```

**Figura 43.** Rutina de `sent_callback()`.

**Fuente:** elaboración propia.

Para realizar la llamada a las callbacks de envío y recepción, se pasa el argumento `tcp_arg` que apunta a una estructura que mantiene la cuenta de cuántos bytes no han sido enviados y cuál es el archivo descriptor que debe emplearse para leer dicho archivo.

### *Librería lwIP TEC/IP stack*

lwIP es una librería de tamaño reducido que se emplea para la implementación del protocolo TCP/IP. Su reducido tamaño hace de esta librería adecuada para sistemas embebidos como el que se está diseñando, ya que emplea alrededor de 10 Kbytes de memoria RAM y de 40 Kbytes de memoria ROM para almacenarla. Los módulos más importantes de la pila lwIP son:

- Administración de *buffer* y memoria: Uno de los factores críticos del sistema es el manejo de memoria debido a que los buffers que se gestionan en este tipo de sistemas puede variar de tamaño dinámicamente. El protocolo TCP puede manejar buffers de algunos cientos de bytes mientras que los otros como el de comunicación serie no ofrecen esta característica.
- Interfaces de red: Guardan toda la información de las diferentes interfaces de red que se estén utilizando en el sistema.
- Protocolos: Cada protocolo se implementa de forma independiente, sin embargo, en este proyecto no es necesario realizar la gestión de los mismos de forma manual, dado que es transparente para el desarrollo del servidor al situarse en un nivel de abstracción más alto que toda la pila de protocolos.

Con el objetivo de conseguir el máximo rendimiento de esta librería, se ha realizado una serie de modificaciones en su configuración. Una de las modificaciones que se realiza posee como finalidad aumentar la capacidad de los descriptors TX y RX y seleccionar la velocidad de 100Mb/s, ya que el periférico no permite velocidades superiores. Esta última modificación no es crítica debido a que en el modo de autonegociación se configura de forma automática la máxima velocidad que permite el enlace. En la Figura 44 pueden observarse las modificaciones realizadas en la librería lwIP.

Name	Value	Default
api_mode	RAW_API	RAW_API
socket_mode_thread_prio	2	2
▲ temac_adapter_options	true	true
n_tx_descriptors	256	64
n_rx_descriptors	256	64
n_tx_coalesce	1	1
n_rx_coalesce	1	1
tcp_rx_checksum_offload	false	false
tcp_tx_checksum_offload	false	false
tcp_ip_rx_checksum_offload	false	false
tcp_ip_tx_checksum_offload	false	false
phy_link_speed	CONFIG_LINKSPEED100	CONFIG_LINKSPEED_AU...
temac_use_jumbo_frames	false	false
emac_number	0	0

**Figura 44.** Configuración del tamaño del descriptor y de la velocidad de enlace.

**Fuente:** elaboración propia.

El sistema de archivos del servidor web se puede considerar como una ROM de gran tamaño, ya que reside en la memoria DDR del sistema de forma que, para maximizar el rendimiento del sistema de archivos, se debe ajustar el tamaño de `memp_n_pbuf` y `memp_tcp_seg` a 1024. Esta modificación se muestra en la Figura 45.

Name	Value	Default
api_mode	RAW_API	RAW_API
socket_mode_thread_prio	2	2
▶ temac_adapter_options	true	true
▲ lwip_memory_options		
mem_size	131072	131072
memp_n_pbuf	1024	16
memp_n_udp_pcb	4	4
memp_n_tcp_pcb	32	32
memp_n_tcp_pcb_listen	8	8
memp_n_tcp_seg	1024	256
memp_n_sys_timeout	8	8
memp_num_netbuf	8	8
memp_num_netconn	16	16
memp_num_api_msg	16	16
memp_num_tcpip_msg	64	64

**Figura 45.** Modificación de parámetros de opciones de memoria del protocolo lwIP.

**Fuente:** elaboración propia.

Por último, otro de los parámetros a modificar para mejorar las características de Ethernet, es el tamaño de paquete del buffer de acarreo mediante el cual se realiza la comunicación entre capas de la pila TCP/IP (Figura 46).

Name	Value	Default
api_mode	RAW_API	RAW_API
socket_mode_thread_prio	2	2
▶ temac_adapter_options	true	true
▶ lwip_memory_options		
▲ pbuf_options	true	true
pbuf_pool_size	1024	256
pbuf_pool_bufsize	1700	1700
pbuf_link_hlen	16	16

**Figura 46.** Modificación del parámetro de tamaño de paquete del buffer de acarreo.

**Fuente:** elaboración propia.

### *Librería Xilinx Memory File System*

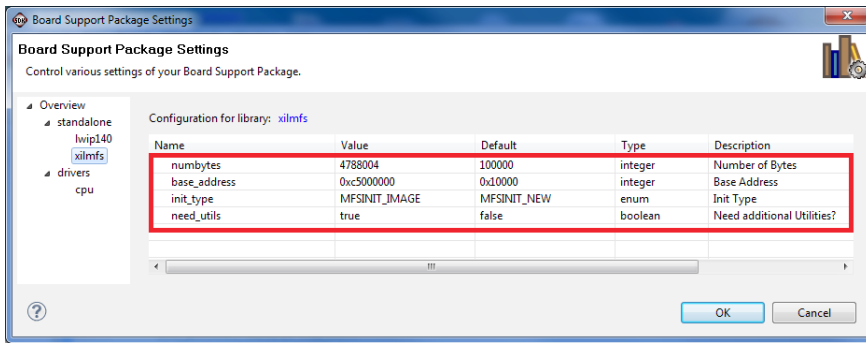
El sistema de archivos de memoria de Xilinx (xilmfs) se utiliza para almacenar un conjunto de archivos en la memoria DDR de la placa de desarrollo. Estos archivos pueden ser accedidos por medio de un comando HTTP GET señalando un navegador web a la dirección IP de la tarjeta de desarrollo y solicitando archivos específicos.

Con el objetivo de optimizar esta librería de gestión de archivos del werveridor web, se van a realizar una serie de modificaciones en alguno de sus parámetros. Los cambios que deben realizarse se muestran en la Figura 47 y corresponden con:

- Aumentar el número de bytes a 4788004 para reservar espacio suficiente para adjuntar la imagen MFS.



- Modificar la dirección *base\_address* a la dirección base del periférico DDR más un offset para evitar sobrescribir el código que alberga la funcionalidad de la aplicación.
- Seleccionar la opción *MSFINIT\_IMAGE* para que al arrancar el sistema se ejecute de forma automática la imagen MFS.
- Seleccionar la opción *true* en el parámetro *need\_utils*.



**Figura 47.** Modificación de los parámetros de la librería xilmsf.

**Fuente:** elaboración propia.

### 6.2.3. Comunicación con el navegador web

Para finalizar el presente capítulo de diseño firmware, se va estudiar el último proceso que ejecuta el servidor web para enviar los datos al navegador. Esto se realiza mediante la creación de un archivo de datos denominado *data.xml* que se anexa en la capeta raíz de la imagen MFS de forma dinámica. En este archivo, se escribe el número de muestra "n", y el valor obtenido del Core TDC para dicha muestra. Cada número de muestra y valor medido se separa mediante un salto de línea para facilitar el tratamiento de datos que se realizará desde la aplicación web que será descrita en el siguiente capítulo. Este proceso de escritura del valor de los datos proporcionados por el Core TDC se realiza 1000 veces con el fin de generar un archivo que contenga mil valores obtenidos de restas temporales medidos en el Core TDC.

En el siguiente fragmento de código (Figura 48) se muestra la rutina que se realiza para escribir el índice de muestra y los mil valores proporcionados por el Core TDC con el formato requerido por la aplicación web.

```

/* Escritura de datos del Core TDC en el archivo data.xml que lee Java Script en
 * la web para la Representación Gráfica*/

strcat(buf, "Muestra, Resta\r\n");
sprintf(buf+j, "%03d,%04d\r\n", 1, CORE_ISEV1_mReadSlaveReg0(XPAR_CORE_TDC_BASEADDR, 0x00000000));

for (i=1; i<1000; i++)
{
    j=j+9; // +9 Incremento de puntero de escritura en archivo data.xml

    core_data = CORE_TDC_mReadSlaveReg0(XPAR_CORE_TDC_BASEADDR, 0x00000000); //Lectura datos registro Core TDC

    sprintf(buf+j, "%03d,%04d\r\n", i, core_data); // Escribir datos Core TDC en archivo data.xml

    xil_printf("\n\r%d\r\n", core_data); // Debug: Muestra por terminal dato leído del Core TDC
}

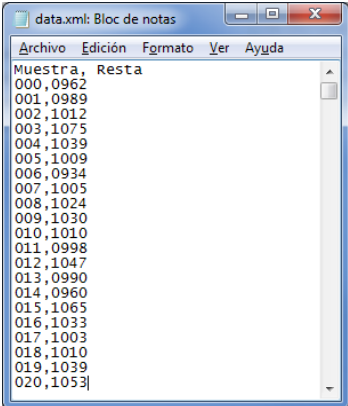
```

**Figura 48.** Escritura y formateo de los datos del Core TDC en el archivo *data.xml*.

**Fuente:** elaboración propia.

Del fragmento de código anterior cabe destacar la función que desempeña la función `strcat()`, empleada para escribir la cabecera “Muestra, Resta” en el archivo `data.xml`, concatenando un bloque de memoria a otro de modo que es posible enlazar dos cadenas de caracteres. Por otro lado, destacar también la función `sprintf()`, dado que es la encargada de escribir el número de muestra y el valor de la resta obtenido del Core TDC para dicho índice.

Mediante estas dos funciones, es posible generar el archivo `data.xml` con el formato deseado, obteniendo el resultado mostrado en la Figura 49.



```
data.xml: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Muestra, Resta
000,0962
001,0989
002,1012
003,1075
004,1039
005,1009
006,0934
007,1005
008,1024
009,1030
010,1010
011,0998
012,1047
013,0990
014,0960
015,1065
016,1033
017,1003
018,1010
019,1039
020,1053
```

**Figura 49.** Formato del archivo `data.xml` generado.

**Fuente:** elaboración propia.

## CAPÍTULO 7: DISEÑO SOFTWARE

En este punto se va a realizar un estudio del software programado, además de la herramienta y lenguaje elegido para la creación de dicho software. Notar que existen dos tipos de entorno de visualización a estudiar: la Aplicación Software para PC que recibirá los datos del sistema TDC mediante una conexión punto a punto de baja tasa de datos y un sitio Web que recibirá los datos mediante comunicación Ethernet de alta velocidad.

La Aplicación para PC se va a programar mediante la herramienta gráfica de programación LabVIEW mientras que el sitio Web que alberga el entorno de visualización remoto se programará empleando lenguaje de HTML y JavaScript.

### 7.1. Aplicación software para PC

La aplicación para PC proporciona una herramienta sencilla de utilizar y fácil de reconfigurar y muy útil para la visualización de los datos proporcionados por el sistema TDC. Esencialmente, esta aplicación programada con LabVIEW permite visualizar los resultados de las restas temporales de los múltiples canales de forma numérica y de forma gráfica mediante un histograma.

A partir de estos datos recogidos, permite realizar una serie de procesados que permiten analizar el comportamiento del TDC y calcular la resolución temporal de las medidas que realiza.

#### 7.1.1. Entorno y lenguaje de programación

En primer lugar, se va a describir el lenguaje de programación elegido para la creación de la aplicación para PC. De las múltiples herramientas de programación de entornos de usuario, se ha elegido LabVIEW, ya que, como se ha comentado, es una herramienta gráfica de programación, lo que significa que los programas no se escriben, sino que se dibujan mediante la conexión de bloques. Existen gran cantidad de bloques pre-diseñados de forma que se facilita la creación del proyecto y el lenguaje que se emplea para ello se denomina lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Los programas en LabVIEW se denominan instrumentos virtuales (VIs), y pueden constar de bloques básicos de programación o de subVIs, es decir, subprogramas que equivaldrían a subrutinas, de forma que se configura un bloque con unas determinadas entradas que son procesadas en su interior y devueltas al programa o VI principal como salidas. Cada VI consta de dos partes diferenciadas (Universidad de Sevilla, 2018):

- **Panel frontal:** es la interfaz con el usuario dado que se emplea para interactuar con él cuando el programa se está ejecutando. En este panel el usuario podrá observar los datos del programa actualizados en tiempo real. En esta interfaz se definen los controles o entradas como pueden ser botones, marcadores, deslizadores... e indicadores como, por ejemplo, gráficas, LEDs virtuales, cuadros de

diálogo... En la Figura 50 se observa un ejemplo del Panel Frontal en el que se genera una señal aleatoria con un control de tiempo de la representación de muestras.

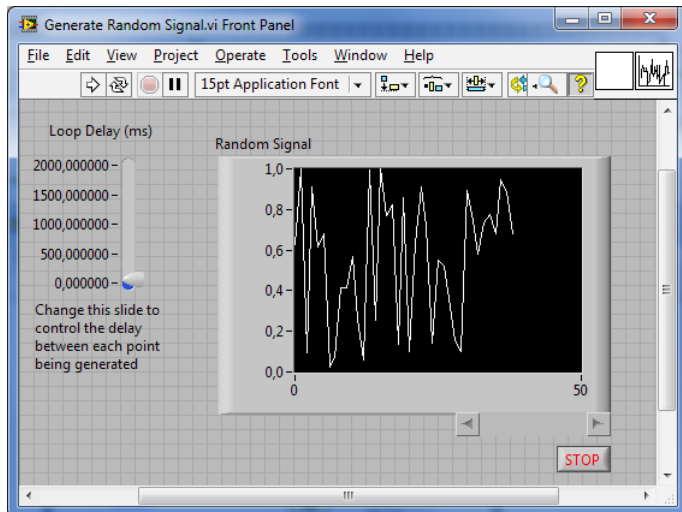


Figura 50. Ejemplo de panel frontal de LabVIEW.

Fuente: elaboración propia.

- Diagrama de bloques: es la parte del VI en la que se realiza la programación por bloques, es decir, donde se define la funcionalidad de la aplicación puesto que es donde se colocan e interconectan los iconos o bloques que realizan una determinada función. En la Figura 51 se muestran los bloques utilizados para implementar el panel frontal de la Figura 50.

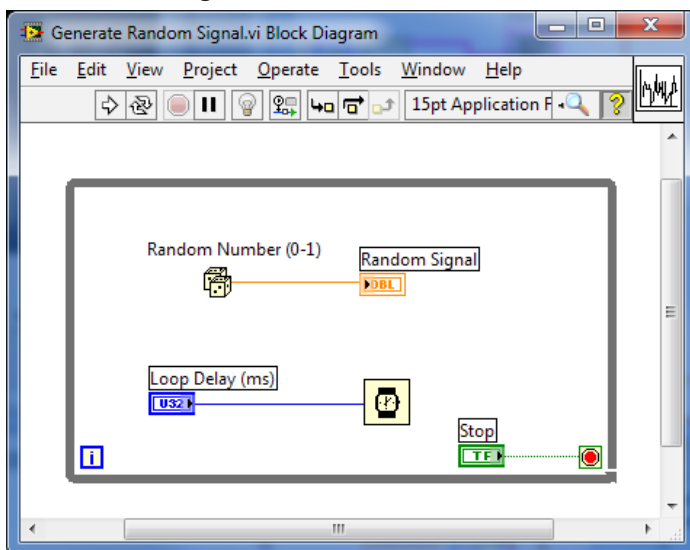


Figura 50. Ejemplo de diagrama de bloques de LabVIEW.

Fuente: elaboración propia.

Por consiguiente, LabVIEW constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Ofrece la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

### ***7.1.2. Estudio del funcionamiento de la aplicación***

El entorno de visualización programado posee la función principal de procesar los datos provenientes del sistema TDC, realizar un cálculo de sus estadísticos más importantes y realizar una representación de los mismos.

De forma más específica, mediante un bloque de procesado que realiza un ajuste Gaussiano de los datos y que permite determinar la desviación estándar, media y FWHM (del inglés, Full Width at Half Maximum). Este último parámetro cobra gran importancia debido a que las medidas realizadas por el TDC al representarlas poseen una distribución Gaussiana de modo que el cálculo de su FWHM permite determinar la resolución temporal de la medida realizada.

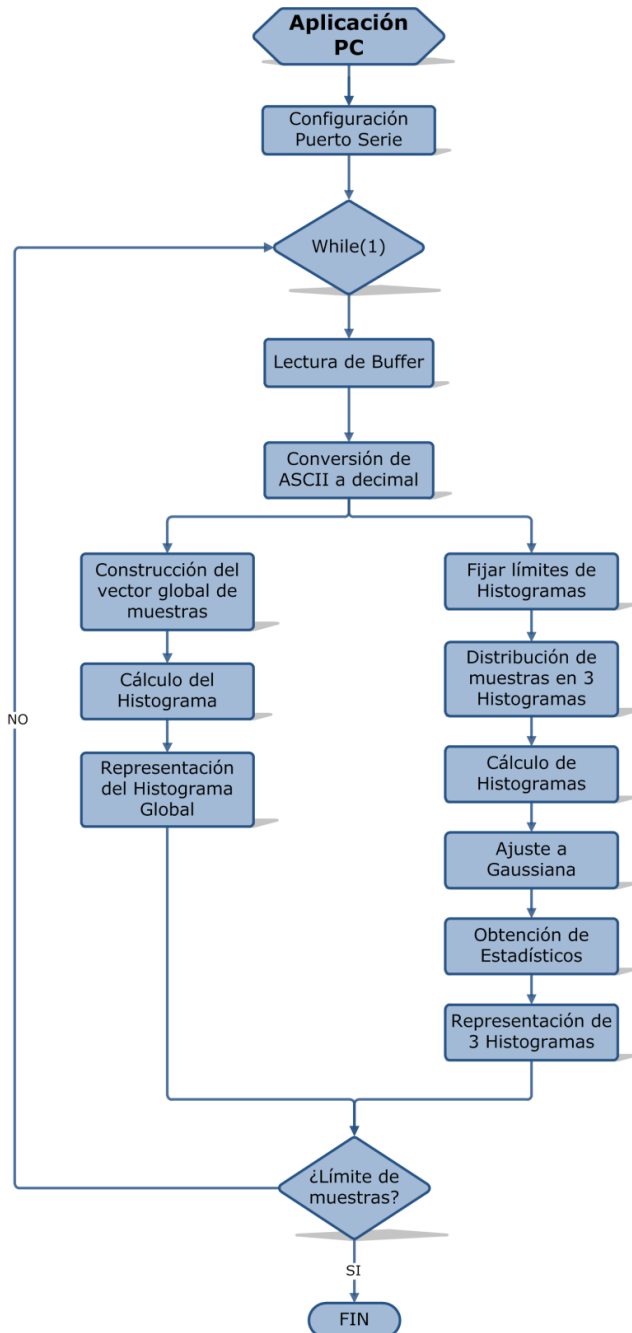
En la última versión de la aplicación para PC se ha programado un entorno de visualización preparado para analizar al mismo tiempo la resolución temporal de 4 canales, lo que equivaldría a analizar los datos obtenidos de 4 detectores dentro de un sistema PET. Con este número de canales, se puede verificar el correcto funcionamiento del sistema TDC multicanal si las resoluciones temporales obtenidas en las tres restas representadas son inferiores a 100 ps. Si esto se verifica, la implementación de un número de canales mayor tanto en la FPGA como en el entorno de visualización sería relativamente sencilla realizando una abstracción de los procesos realizados para el presente sistema de 4 canales.

A continuación, se realiza una descripción más detallada de los procesos que se realizan siguiendo siempre el mismo orden cronológico puesto que se encuentran en el interior de un superbucle:

- Se realiza la lectura de datos del buffer para obtener los valores de medidas temporales enviados desde el TDC.
- Se realiza una conversión de ASCII a número decimal de los valores obtenidos.

- A continuación, se produce una bifurcación en el código ya que se realizan dos tareas de forma paralela.
- Por un lado, se construye el vector global de muestras en el que se almacenan todos los valores recibidos desde el TDC, se calcula el histograma de dicho vector y se representa mediante un histograma.
- Por otro lado, se fijan los umbrales de los tres histogramas a representar correspondientes a los tres valores de restas temporales proporcionados por el TDC, se realiza una distribución de las muestras formando tres subvectores, se realiza el cálculo de los tres histogramas a partir de dichos vectores, se procede a ajustar los datos a una función Gaussiana en cada histograma, se obtienen los estadísticos de los mismos y se realiza la representación de los tres histogramas.
- Por último, se compara el tamaño del vector de muestras global con el número de muestras que se ha fijado desde el entorno para saber si se debe terminar el programa o debe continuar procesando datos.

En el diagrama de flujo de la Figura 51 se recogen las funciones que realiza la aplicación programada con LabVIEW descritas anteriormente.



**Figura 51.** Diagrama de flujo de la aplicación programada con LabVIEW.

**Fuente:** elaboración propia.

### Análisis del panel frontal

En este subapartado, se van a explicar los indicadores y gráficas que se han incluido en el panel frontal para implementar el entorno de usuario y permitir principalmente dos aspectos: observar los histogramas de las medidas temporales realizadas en el TDC y obtener el valor de la resolución temporal de las mismas.

A continuación, en la Figura 52, se adjunta una captura del panel frontal completo en el que se pueden observar las funciones principales del entorno de usuario que se ha programado empleando LabVIEW.

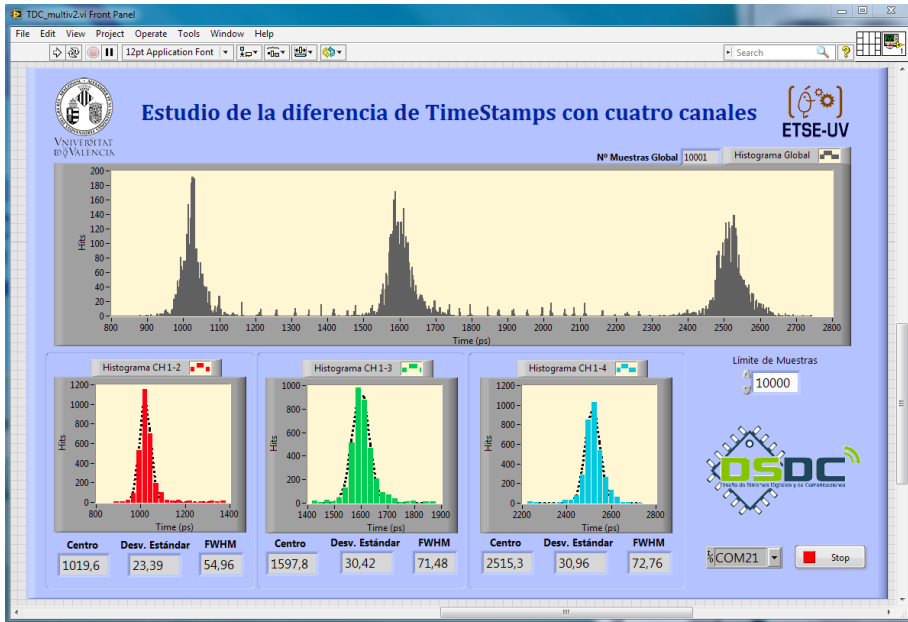


Figura 52. Entorno de visualización programado en LabVIEW.

Fuente: elaboración propia.

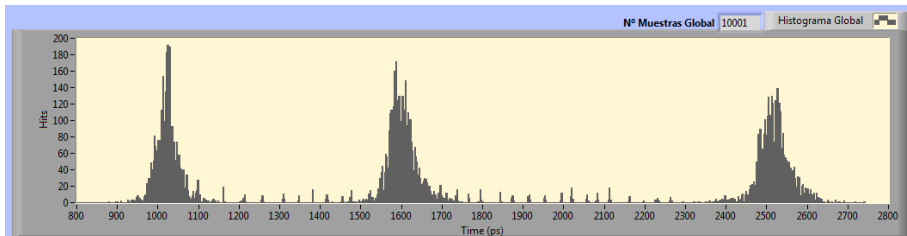
En la captura anterior se observa la disposición general de los bloques, de ellos cabe destacar los dos principales, como son la representación de los histogramas y el cálculo del parámetro FWHM que indica la resolución temporal del sistema en pico segundos.

Una característica muy importante de los gráficos empleados es que permiten la exportación de los datos adquiridos a archivos de texto plano como puede ser un .txt o a una hoja de cálculo. Este aspecto es muy relevante debido a que pueden emplearse los vectores de muestras adquiridos de forma inalámbrica para realizar un cálculo de estadísticos mucho más completo y preciso empleando programas con un mayor poder de procesado como puede ser Matlab.

De forma más precisa, si se realiza un estudio dividiendo el panel frontal del entorno de visualización programado en bloques siguiendo el orden del diagrama de flujo anterior, se tendría en primer lugar, la representación del histograma que representa el vector global de muestras (Figura 53). En este histograma se pueden



observar representados los tres histogramas correspondientes al resultado de las tres restas temporales que proporciona el TDC a partir de los 4 canales que posee implementados. Otro factor a destacar en este bloque es que es posible medir el retardo que existe entre cada histograma o resta en pico segundos. Por último, en cuanto a este bloque, notar que en la parte superior derecha se indica el número de muestras que se están representando.



**Figura 53.** Bloque de representación del vector global de muestras en un histograma.

**Fuente:** elaboración propia.

En segundo lugar, se va a describir el cuadro de controles del panel frontal en el que se puede controlar el valor del límite de muestras a almacenar en el vector global y, por lo tanto, a representar en el histograma global (Figura 54). También es posible seleccionar el puerto COM en el que se encuentra conectada la FPGA y mediante el cual se van a obtener los datos del TDC. Otro control que puede encontrarse en el Panel Frontal en la parte inferior derecha, es el correspondiente a la parada de la ejecución del programa mediante el botón Stop, de forma que, el programa automáticamente deja de ejecutar el superbucle de igual forma que cuando se llega al valor de límite de muestras fijado.



**Figura 54.** Bloque de control del panel frontal.

**Fuente:** elaboración propia.

El tercer bloque funcional que puede encontrarse en el entorno de visualización, corresponde a los subhistogramas o histogramas individuales correspondientes a los valores de las tres restas temporales que proporciona el TDC. En estos histogramas se muestran los valores de cada resta temporal individualizada junto con una Gaussiana ideal (línea de puntos negra) para poder realizar la comparativa de forma visual. En la parte inferior de la representación del histograma se muestran los tres estadísticos más representativos de los valores adquiridos: el centro o bin más significativo del

histograma, el valor de la desviación estándar y el valor de la FWHM equivalente a la resolución temporal de las medidas realizadas. Este tercer bloque puede visualizarse con mayor precisión en la Figura 55.

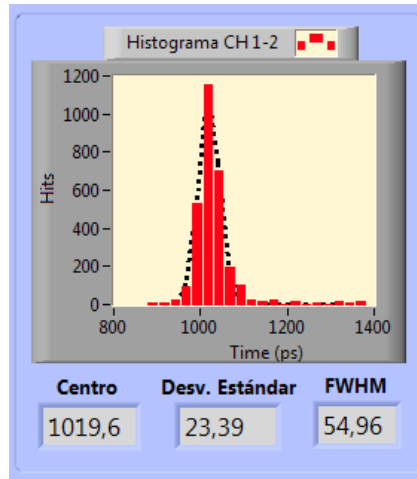


Figura 55. Bloque de visualización de los subhistogramas y sus estadísticos.

Fuente: elaboración propia.

### Análisis del diagrama de bloques

En este subapartado se va a identificar las partes más importantes del Diagrama de Bloques programado para realizar las funciones reflejadas en el diagrama de flujo de la Figura 51. En la Figura 56 se muestra el diagrama de bloques completo correspondiente a la aplicación programada para visualizar los datos del TDC en el PC.

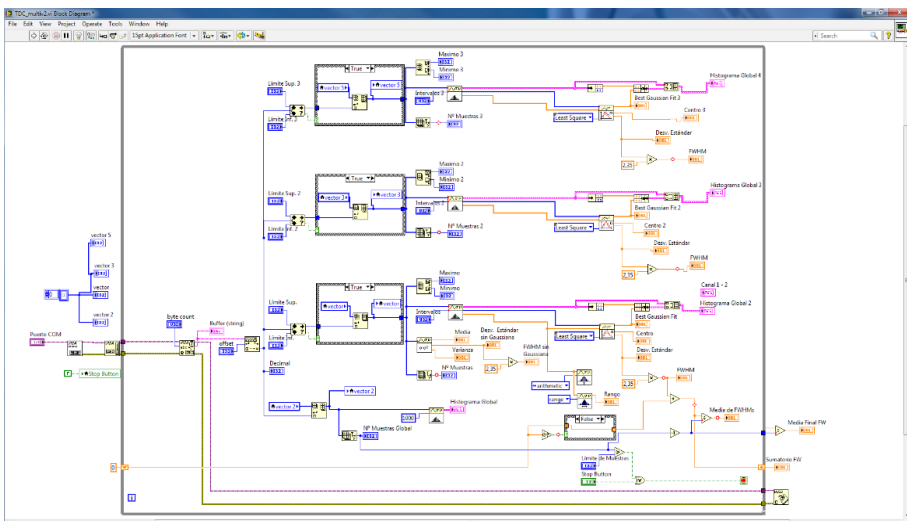


Figura 56. Diagrama de bloques de la aplicación para PC programada en LabVIEW.

Fuente: elaboración propia.

En el diagrama de bloques anterior puede observarse en la parte izquierda, fuera del superbucle, la configuración de la comunicación serie y la inicialización de vectores y variables. Una vez comienza a ejecutarse el bucle infinito, la primera tarea a realizar es la lectura del buffer y la conversión de la cadena de caracteres leída a tipo decimal numérico. Una vez se realiza esta operación se produce la bifurcación comentada en el diagrama de flujo del programa, y se ejecuta el cálculo y representación del histograma global y de los tres subhistogramas hasta que el número de muestras adquiridas es igual al límite fijado en la variable que lo controla.

## 7.2. Programación del sitio web

El sitio web a programar posee como función, validar la viabilidad de poder enviar los datos obtenidos en el sistema TDC mediante una comunicación de alta velocidad. Para verificar que este propósito es posible, se ha programado un sitio web que realice las funciones de entorno de visualización de los datos adquiridos por el TDC de forma remota, mostrándolos tanto numéricamente como de forma gráfica y, verificando que son coherentes con los resultados obtenidos en la aplicación para PC expuesta en los puntos anteriores.

### 7.2.1. Entorno y lenguaje de programación

La programación del sitio web de visualización de datos del sistema TDC se ha realizado empleando como base lenguaje HTML (del inglés, HyperText Markup Language) y funciones específicas de JavaScript que permiten desempeñar tareas más complejas como la representación y actualización de los datos.

El lenguaje HTML se emplea en Internet para definir las páginas de WWW (del inglés, WORLD WIDE WEB). Los ficheros HTML son ficheros puramente ASCII, que pueden ser escritos con cualquier editor básico, tal como Notepad en Windows. También se pueden utilizar procesadores de texto más avanzados como Front Page o Adobe Dreamweaver. En el fichero HTML se introducen unas marcas o caracteres de control llamadas TAGs, que son interpretadas por el navegador Web de modo que, cuando éste lee un fichero ASCII con extensión .htm o .html, interpreta estas TAGs y formatea el texto de acuerdo con ellas.

Por otro lado, JavaScript proporciona la funcionalidad necesaria al sitio Web para ser capaz de implementar gráficas y otras funciones de valor añadido. La integración de JavaScript en archivos HTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web. En la programación del sitio web del presente trabajo se han empleado únicamente las dos siguientes (Flanagan, 2006):

- El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del archivo HTML. Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript dentro de la cabecera del archivo (dentro de la etiqueta `<head>`). Para que la página HTML resultante sea válida, es necesario añadir el atributo *type* a la etiqueta `<script>` y, siguiendo el estándar, para JavaScript el valor correcto de este atributo es `text/javascript`. Este método se

emplea cuando se define un bloque pequeño de código o cuando se quieren incluir instrucciones específicas en un determinado documento HTML que completen las instrucciones y funciones que se incluyen por defecto en todos los documentos del sitio web. El principal inconveniente es que, si se quiere hacer una modificación en el bloque de código, es necesario modificar todas las páginas que incluyen ese mismo bloque de código JavaScript. En la Figura 57 se muestra un ejemplo de programación en JavaScript mediante la etiqueta <script>.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de código JavaScript en el propio documento</title>
<script type="text/javascript">
  alert("Un mensaje de prueba");
</script>
</head>

<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

**Figura 57.** Ejemplo de programación de JavaScript mediante etiqueta <script>.

**Fuente:** elaboración propia.

- Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos HTML enlazan mediante la etiqueta <script>. Se pueden crear todos los archivos JavaScript que sean necesarios y cada archivo HTML puede enlazar tantos archivos JavaScript como necesite. Además del atributo *type*, este método requiere definir el atributo *src*, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar. Cada etiqueta <script> solamente puede enlazar un único archivo, pero en una misma página se pueden incluir tantas etiquetas <script> como sean necesarias. Las principales ventajas de enlazar un archivo JavaScript externo es que se simplifica el código HTML de la página, se puede reutilizar el mismo código JavaScript en todas las páginas del sitio web y cualquier modificación realizada en el archivo JavaScript se ve reflejada inmediatamente en todas las páginas HTML que lo enlazan. En la Figura 58 se muestra un ejemplo de esta metodología.

Archivo `codigo.js`

```
alert("Un mensaje de prueba");
```

Documento XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3
.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de código JavaScript en el propio documento</title>
<script type="text/javascript" src="/js/codigo.js"></script>
</head>

<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

**Figura 58.** Ejemplo de programación de JavaScript mediante archivo externo .js.

**Fuente:** elaboración propia.

De forma específica, la programación que se ha realizado para generar el sitio Web se ha basado en el enlace a archivos JavaScript externos en los que se ubica la funcionalidad de algunos bloques complejos como los de representación gráfica. Los bloques de menor complejidad programados manualmente, como la representación de la tabla de valores adquiridos, también se ha programado empleando JavaScript, pero al tratarse de funciones menos complejas se han desarrollado en el mismo archivo HTML debido a su corta extensión.

La herramienta que se ha empleado para la programación del sitio web ha sido Adobe Dreamweaver CS6, se trata de una aplicación en forma de estudio que está destinada a la construcción, diseño y edición de sitios, videos y aplicaciones Web basados en estándares. La gran ventaja de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que, en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están programadas en JavaScript, lo que ofrece una gran flexibilidad. Otra de las ventajas que posee, es la posibilidad de programar en el modo ventana dividida mediante el cual es posible visualizar en tiempo real el aspecto del sitio Web que se está diseñando. En la Figura 59 se muestra una captura de esta herramienta de programación web.

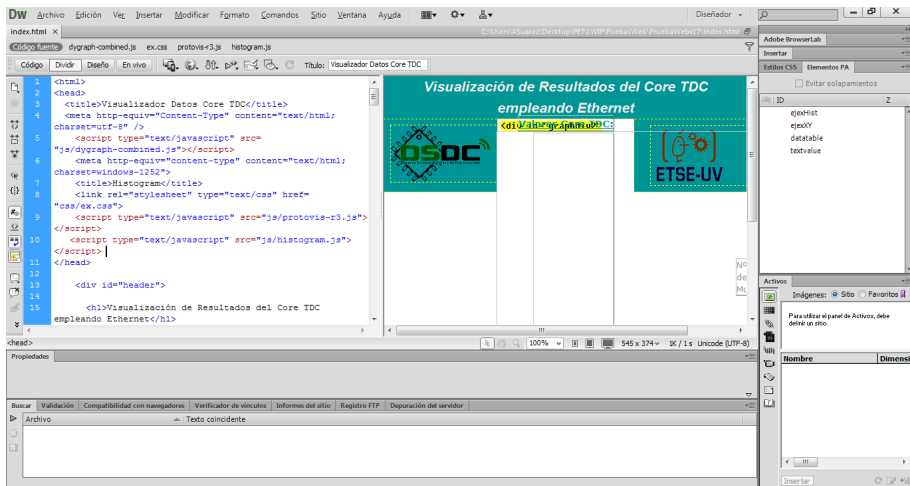


Figura 59. Entorno de programación web Dreamweaver.

Fuente: elaboración propia.

Asimismo, es importante describir brevemente el proceso de generación de la imagen MFS que se carga en la DDR de la FPGA para implementar el Servidor Web con el sitio web que se está describiendo en el presente capítulo. Este archivo MFS se genera con la consola de Windows a partir de los archivos fuente del sitio web que se quiere integrar en el servidor web.

Para facilitar esta labor, se ha programado un archivo ejecutable con extensión BAT en el que se han incluido los parámetros que hay que emplear para generar la imagen MFS. Para que la imagen se genere correctamente se ha generado una carpeta en la que se añaden además del archivo HTML, denominado index.html, las siguientes subcarpetas:

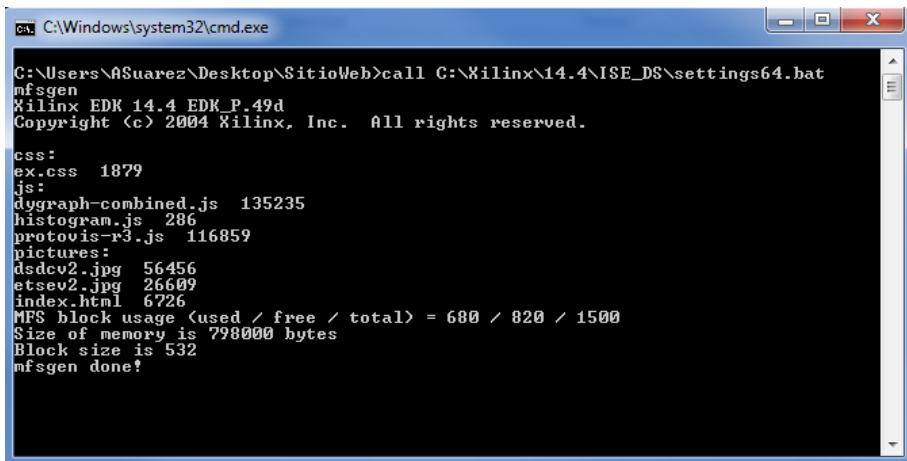
- **css:** en esta subcarpeta se incluirán todos los archivos de estilos necesarios para visualizar correctamente el sitio web.
- **js:** en esta subcarpeta se adjuntarán los archivos JavaScript externos necesarios para realizar las funciones más complejas y específicas del código.
- **pictures:** en esta subcarpeta se añadirán todas las fuentes gráficas del sitio web.

El contenido del archivo BAT necesario para generar correctamente está compuesto por la llamada al motor de ISE para generar la imagen y por el comando mfsngen al que se le indica la ruta de destino, el nombre de la imagen MFS a generar, el tamaño de dicho archivo, las subcarpetas que deben cargarse en la imagen y, por último, el fichero index.

```
call C:\Xilinx\14.4\ISE_DS\settings64.bat
```

```
mfsngen-cvbf ../image.mfs 1500 css js pictures index.html
```

La Figura 60 muestra una captura tomada de la consola de Windows se muestra el resultado tras realizar el proceso de generación de imagen MFS:



```

C:\Windows\system32\cmd.exe

C:\Users\ASuarez\Desktop\SitioWeb>call C:\Xilinx\14.4\ISE_DS\settings64.bat
mfsngen
Xilinx EDK 14.4 EDK_P_49d
Copyright (c) 2004 Xilinx, Inc. All rights reserved.

css:
ex.css 1879
js:
dygraph-combined.js 135235
histogram.js 286
protovis-r3.js 116859
pictures:
dsdcv2.jpg 56456
etsev2.jpg 26609
index.html 6726
MFS block usage (used / free / total) = 680 / 820 / 1500
Size of memory is 798000 bytes
Block size is 532
mfsngen done!
  
```

**Figura 60.** Resultado tras generar la imagen MFS.

**Fuente:** elaboración propia.

### 7.2.2. Estudio del funcionamiento de la aplicación

El entorno de visualización programado posee la función principal de procesar y representar en un sitio web los datos provenientes del sistema TDC. Para realizar estas tareas, el sitio Web se ha programado de forma que, en primer lugar, realiza la lectura de los datos del TDC que han sido escritos en el archivo data.xml. Una vez se han guardado dichos datos en un vector mediante JavaScript, se realiza un formateo de los datos para ser introducidos en la función de representación de la gráfica XY en la que se muestran los valores de la resta temporal de un canal respecto al índice de muestra, para ser incluidos en la tabla de valores y para ser reordenados y representados en un histograma.

En este caso, el entorno de visualización web se ha programado para que procese y represente valores provenientes de la resta de dos canales ya que no es necesario procesar los datos correspondientes a cuatro canales, como se ha descrito en la aplicación para PC programada con LabVIEW, debido a que con dos canales se poseen suficientes datos para realizar la validación de la comunicación Ethernet y el correcto funcionamiento del servidor web integrado en la FPGA.

A continuación, en la Figura 61, se muestra el diagrama de flujo correspondiente a los procesos que se realizan en el sitio web programado con Dreamweaver.

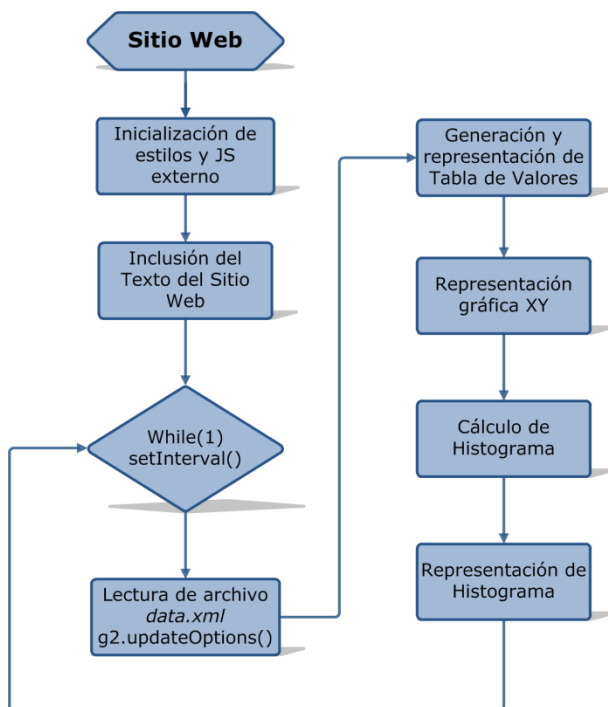


Figura 61. Diagrama de flujo del sitio web.

Fuente: elaboración propia.

Cabe destacar que las funciones que se emplean para realizar la lectura de los datos del archivo data.xml y la representación gráfica XY están basadas en la librería “Dygraphs JavaScript Visualization” (Dygraphs, 2018) mientras que para el cálculo y representación del histograma se ha empleado la librería “Protovis” (Bostock y Heer, 2009). A continuación, en la Figura 62, se muestra una captura del aspecto final del sitio web procesando datos reales.

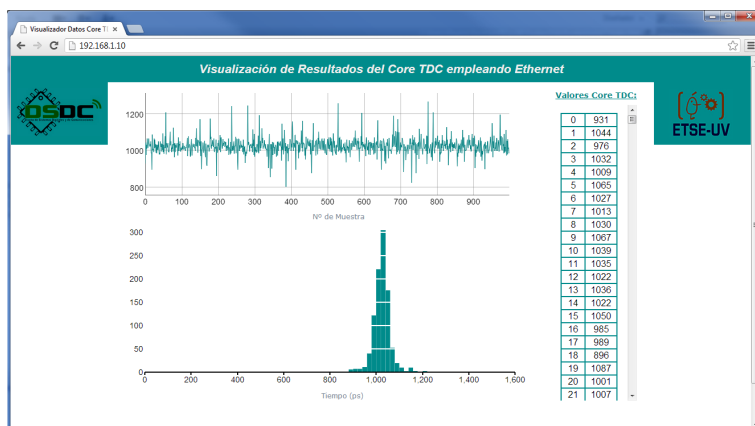


Figura 62. Sitio web de visualización de medidas temporales del Core TDC.

Fuente: elaboración propia.



### 7.3. Sumario

Una vez presentados, descritos y analizados los dos entornos de visualización que se han programado, se debe verificar su validez mediante la realización de diferentes pruebas de funcionamiento que certifiquen que el sistema TDC puede integrarse en un sistema PET real en tanto que posee una resolución inferior a 100 ps y es capaz de mantener este valor cuando se implementan múltiples canales.

La aplicación para PC programada en LabVIEW permitirá validar la resolución temporal que es capaz de obtener el sistema TDC para múltiples canales y también se puede emplear en un sistema real en el que se le conecten la salida de cuatro canales. No obstante, se ha programado de forma estructurada y modular, por lo que no conllevaría demasiado trabajo realizar una ampliación del programa diseñado para que fuese capaz de representar más de 3 histogramas.

Por lo que respecta al sitio Web programado, cabe destacar que su principal función es validar que se reciben datos correctamente y que estos son coherentes mediante el empleo de una comunicación de alta velocidad como es Ethernet. Esto sirve para sentar las bases de una comunicación con una tasa de transmisión de hasta 1 Gb/s (si se emplea el periférico Trimode Ethernet), ya que se debe emplear una comunicación capaz de gestionar sistemas con un número de detectores muy elevado en los que la comunicación serie queda totalmente descartada por su baja velocidad de transmisión.

Por último, es conveniente destacar que en el próximo capítulo se testeará el funcionamiento de todos los módulos de los dos entornos descritos para verificar que su programación se ha realizado correctamente.

*Implementación de un servidor web para gestionar un core time-to-digital-converter multicanal empleado en sistemas médicos PET*

## CAPÍTULO 8: PRUEBAS Y RESULTADOS

Una vez expuestos los principios físicos, médicos y electrónicos del sistema, se van a realizar diferentes pruebas para verificar que se ha implementado un Core TDC gestionado por MicroBlaze con capacidad de adquirir valores de varios canales, manteniendo una resolución temporal inferior a la centena de pico segundos y, empleando para ello, un dispositivo FPGA de bajo coste.

Por otro lado, se va a integrar el TDC en un sistema PET para comprobar su correcto funcionamiento en una aplicación médica PET real.

### 8.1. Validación de la resolución del TDC gestionado por MicroBlaze

La primera tarea a realizar poseía como objetivo principal integrar el Core TDC desarrollado y testeado con la herramienta ISE. Esta no es una tarea trivial dado que el Core TDC es un sistema muy complejo que debe obtener resoluciones temporales ínfimas por lo que, a pesar de haber sido probado y verificado su correcto funcionamiento con ISE, al importarlo a un sistema comandado por MicroBlaze existían una serie de procesos que no funcionaban correctamente.

Con esto, pese a no tener un funcionamiento totalmente correcto en tanto que la señal `ready_0` no adoptaba el comportamiento esperado, era posible obtener el valor de la parte decimal (raw) proporcionado por el Encoder. Sin embargo, el valor de la siguiente etapa, es decir, a la salida de la LUT en la que se almacenan los valores de retardo de cada bin, era siempre nulo. En la Figura 63 se muestra la diferencia del valor del raw entre dos canales antes de pasar por la LUT. Notar que existen cuatro histogramas debido a que en las pruebas realizadas el algoritmo de detección de flancos no estaba en funcionamiento, por lo que al inyectar una señal cuadrada existen cuatro momentos en los que se produce un flanco de subida.

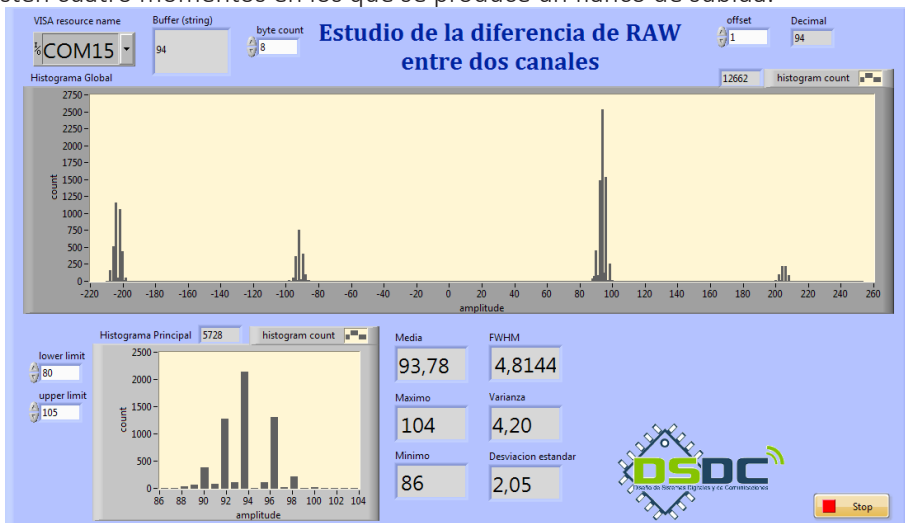


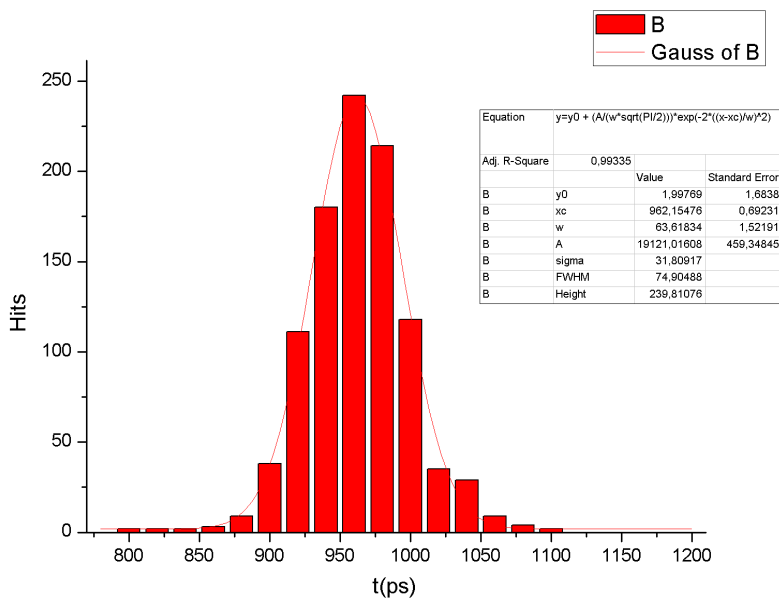
Figura 63. Resultados previos del TDC gestionado por MicroBlaze.

Fuente: elaboración propia.

La distribución Gaussiana y los valores que se muestran en la figura anterior verificaban que el Core TDC funcionaba correctamente ya que de forma teórica se obtenían valores cercanos a los 100 ps al emplear como base empírica de los cálculos el valor de raw.

Llegados a este punto, se realizaron las pertinentes modificaciones en el código del Core TDC para que realizara los procesos de resta de diferencias temporales en el archivo user\_logic.vhd, se modificaron algunos parámetros de implementación de las líneas de retardo como el número de etapas para que se adapten perfectamente a la estructura interna de la FPGA que se está empleando y se eliminaron fragmentos de código relacionados con la base de tiempo que para realizar las pruebas de resolución y obtener el resultado de la resta temporal en MicroBlaze no son necesarios.

Con la realización de dichos cambios se consiguió que MicroBlaze pudiera realizar correctamente la lectura de los datos proporcionados por el Core TDC y que los mostrara por pantalla. A partir de estos datos, se realizó un análisis preliminar de sus estadísticos empleando el programa de cálculo y representación gráfica Origin. En la Figura 64 se muestran los resultados obtenidos, prestando especial atención al valor de FWHM dado que es inferior a 100 ps, como se propone en los objetivos del presente trabajo.

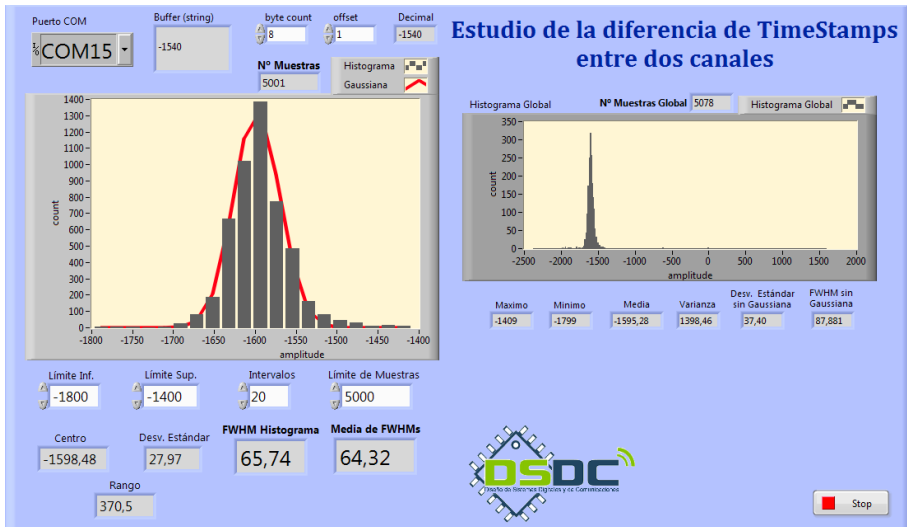


**Figura 64.** Primeros resultados de resolución temporal del TDC con MicroBlaze.

**Fuente:** elaboración propia.

Una vez verificado el correcto funcionamiento del TDC, se programó la primera versión de la aplicación para PC mediante la que se procesaban los datos provenientes de las restas temporales entre los canales 1 y 2, se realizaban los estadísticos de los mismos y se procedía a representarlos gráficamente empleando un histograma. En la Figura

65 se muestran los resultados obtenidos y la apariencia del entorno programado que sentó las bases del que se emplea actualmente y que ha sido presentado en el capítulo anterior.



**Figura 65.** Resultados del TDC gestionado por MicroBlaze en la primera versión de la aplicación para PC.

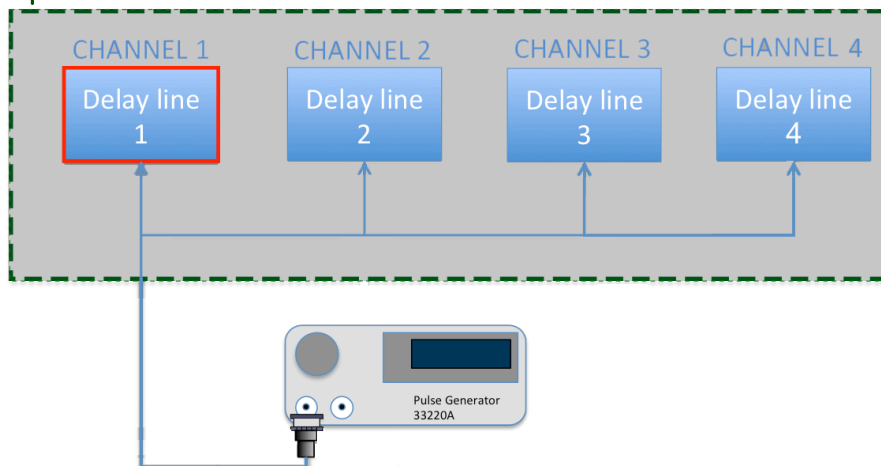
**Fuente:** elaboración propia.

## 8.2. Estudio de la resolución del TDC con múltiples canales

A partir de la configuración anterior, se va a describir el proceso que se ha seguido para verificar la resolución del TDC con 4 canales, en el que se ha empleado un generador de funciones para simular la llegada de eventos a los diferentes canales. Dentro de la FPGA, esta señal se distribuye progresivamente en cada una de las 4 líneas de retardo.

Esta configuración permite medir los diferentes retardos con respecto al canal 1 teniendo en cuenta que, conforme aumenta el número de canal, el rutado interno es mayor y, por tanto, los canales superiores tendrán mayor retardo que los más cercanos al canal 1. En la Figura 66 se muestra la configuración que se ha implementado para simular el comportamiento de un sistema PET de 4 canales y testear la resolución temporal del TDC.

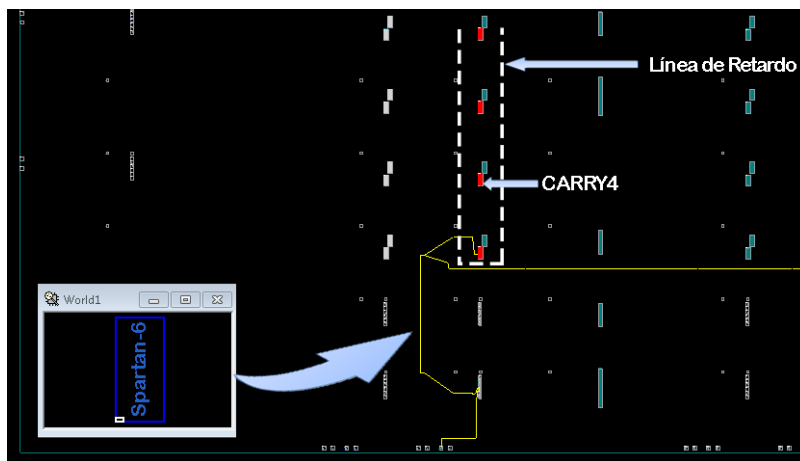
### Spartan-6



**Figura 66.** Esquema de configuración para verificar la resolución real del TDC de 4 canales.

**Fuente:** elaboración propia.

Análogamente, la Figura 67 muestra cómo se obtiene la señal de entrada desde el pin de la FPGA, se bifurca para obtener la señal de entrada del canal 1 y la señal que se conectará a las 3 líneas de retardo restantes (línea amarilla). A su vez, se observa la interconexión de la señal de entrada con el primer bloque de acarreo CARRY4 de la línea de retardo (bloques rojos) correspondiente al canal 1.

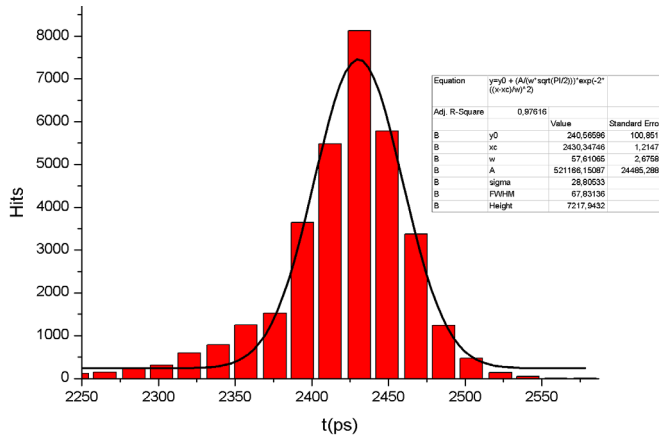


**Figura 67.** Interconexión de la señal de entrada con la primera línea de retardo.

**Fuente:** elaboración propia.

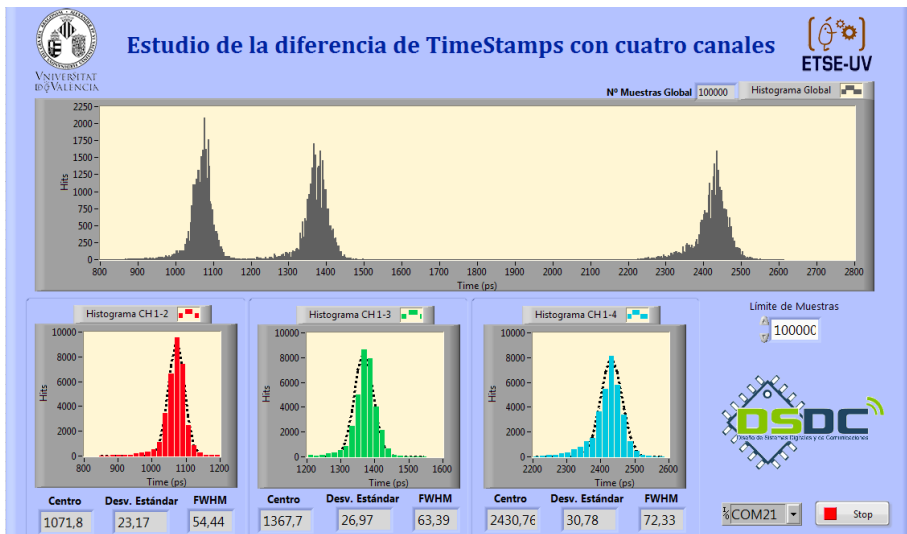
La visualización de los resultados se realiza mediante la aplicación para PC descrita en el capítulo anterior. MicroBlaze realiza continuamente la lectura de los registros del Core TDC y los carga en el periférico UART para que sean enviados mediante comunicación serie al entorno de usuario.

En la Figura 68 se ilustra un histograma obtenido con el programa de cálculo estadístico Origin mediante el que se ha determinado el valor de FWHM entre el canal 1 y 4 (peor caso) con el objetivo de verificar que los datos obtenidos en el entorno de usuario programado en LabVIEW son coherentes.



**Figura 68.** Histograma de diferencias temporales entre canal 1 y 4 con Origin.  
**Fuente:** elaboración propia.

Una vez analizados los datos obtenidos con el programa de cálculo de valores estadísticos, se proceden a analizar los valores estadísticos que el entorno de usuario proporciona. En la Figura 69 se muestran los 3 histogramas correspondientes a la diferencia temporal entre el primer canal y los tres restantes dentro del entorno mencionado.

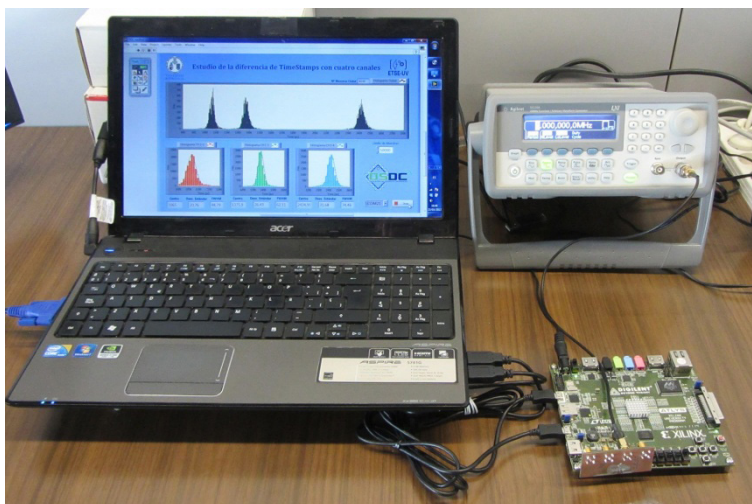


**Figura 69.** Visualización de las diferencias temporales entre los canales 1 y del 2 al 4 en el entorno de LabVIEW.  
**Fuente:** elaboración propia.

Estudiando el parámetro FWMH para todos los canales, se obtiene una resolución inferior a 100 ps, por tanto, valores muy inferiores a los que se pretendía mantener ya que tras realizar diferentes pruebas empleando para cada una de ellas 100.000 muestras se ha obtenido una media de 50-75 ps de resolución temporal.

Cabe destacar que la separación que existe entre cada histograma se debe al retardo del rutado interno que se ha realizado en la FPGA, dado que al bifurcar la señal y al no estar juntas las líneas de retardo sino distribuidas por toda la FPGA, se produce una separación temporal. Esto es apreciable en la distribución de las Gaussianas en tanto que la diferencia temporal entre los dos primeros histogramas es mucho menor que la diferencia entre el segundo y tercer histograma debido a que corresponden a las líneas de retardo 2 y 3 que se encuentran prácticamente juntas en el centro de la estructura interna de la FPGA, mientras que el tercer histograma posee una separación temporal mayor debido a que corresponde a la línea de retardo del canal 4, situada en el extremo opuesto de la línea del canal 1.

De este modo, al realizar la resta entre los canales 2, 3 y 4 con el canal 1 se obtienen resoluciones similares, pero en la representación temporal de los tres histogramas se aprecia una separación mayor entre el tercer histograma y los dos primeros. En la Figura 70 se muestra el sistema formado por un generador de funciones, el kit de FPGA Atlys y el PC con LabVIEW, que se ha empleado para realizar el test del Core TDC.



**Figura 70.** Sistema de test del Core TDC.

**Fuente:** elaboración propia.

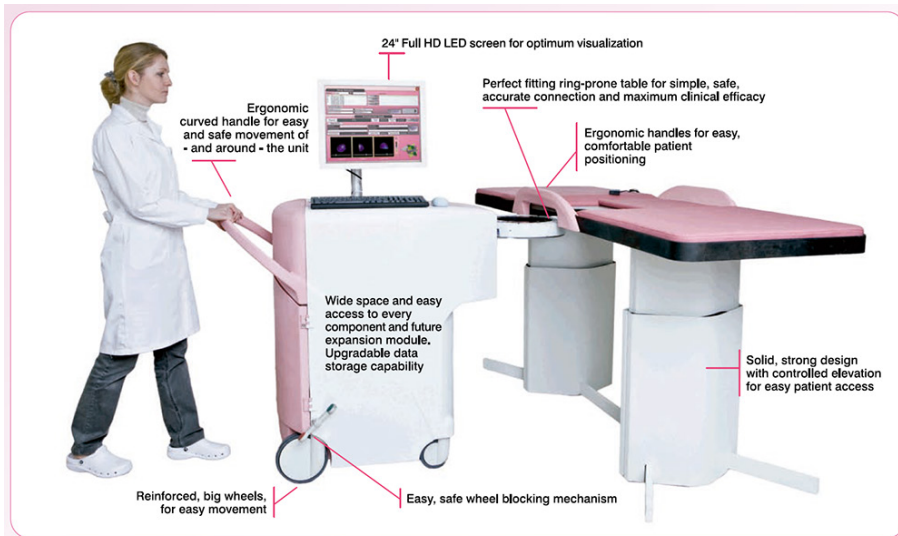
### **8.3. Análisis del funcionamiento del TDC en un sistema médico PET**

Una vez se ha comprobado el correcto funcionamiento del TDC integrado en la FPGA, verificando que permite realizar la lectura de varios canales manteniendo para todos ellos resoluciones inferiores a 100 ps, se han realizado pruebas en un sistema real PET. El sistema que se ha empleado para verificar el correcto funcionamiento



del TDC es el anillo de detectores pertenecientes al prototipo “MAMMI breast PET” (OncoVision, 2018) mostrado en la Figura 71 y desarrollado por la empresa ONCOVISION (GEM-IMAGING S.A.) y el Instituto de Instrumentación para Imagen Molecular (I3M) perteneciente al CSIC (Gonzalez , *et al.*, 2012; Moliner , *et al.*, 2012).

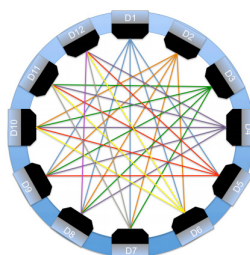
El equipo está destinado a la detección de cáncer de mama en mujeres y ha sido cedido al Grupo de Diseño de Sistemas Digitales y de Comunicaciones (DSDC) del Departamento de Ingeniería Electrónica de la Universidad de Valencia para poder realizar tareas de investigación a partir del mismo.



**Figura 71.** Sistema PET comercial “MAMMI breast PET”.

**Fuente:** [30].

Por lo que respecta a la arquitectura del anillo del sistema “MAMMI breast PET” que se va a emplear para realizar pruebas reales del TDC, cabe destacar que posee doce detectores del tipo Position Sensitive Photo-Multiplier Tubes (PSMTs), un único cristal continuo por cada detector y tiene un diámetro interior de 17 cm.



**Figura 72.** Arquitectura del anillo empleado para realizar pruebas en un sistema PET real.

**Fuente:** elaboración propia.

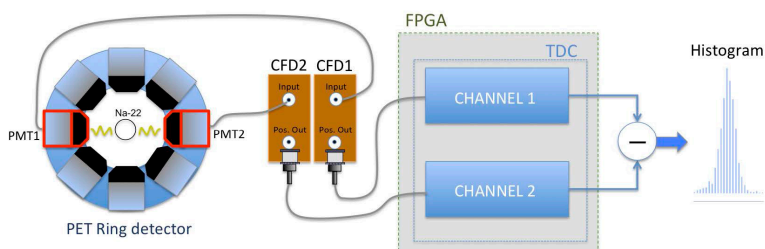
Con el objetivo de simular el radiofármaco que se le suministra al paciente en este tipo de pruebas, se ha empleado una fuente de sodio ( $^{22}\text{Na}$ ) que emula el isótopo radioactivo, es decir, el positrón que se desprende del radionúclido y colisiona con un electrón produciéndose la aniquilación mediante la cual se desprenden dos partículas gamma, que son las que identifican los detectores para determinar que se ha producido una coincidencia. Esta fuente emite de forma isotrópica pares de partículas gamma por lo que emulan el comportamiento de los eventos que se emiten en el interior del organismo del paciente, de forma que es posible detectarlas y visualizarlas mediante el entorno de usuario programado. Cabe destacar que esta fuente es exenta y ha sido cedida por el Instituto de Física Corpuscular (IFIC) habiendo superado los controles pertinentes para poder realizar pruebas con ella.

Para realizar las pruebas reales con el anillo se va a emplear el esquema de configuración que se ilustra en la Figura 73. Este esquema consta de la fuente radiactiva que se sitúa en el interior del propio anillo de 12 detectores PMT, de los cuales se van a emplear dos. Éstos se conectarán al CFD (del inglés, Constant Fraction Discriminator) encargado de digitalizar la señal procedente de los detectores para que puedan ser interpretadas correctamente por el dispositivo FPGA. El CFD se emplea para discriminar aquellas señales procedentes de los detectores del anillo que no superen un nivel umbral fijado. Los pulsos analógicos que superan dicho umbral de discriminación se convierten en los pulsos digitales que se envían a la FPGA, preservando en todo momento la información temporal.

El principio de funcionamiento del CFD es el siguiente: se descompone la señal en dos componentes, una que será retardada unos pocos nano segundos y otra que es atenuada por un determinado factor, tras lo cual se recombinan ambas señales para dar lugar a una nueva señal con un flanco bien definido que marca el instante de ocurrencia del pulso, un comparador dispara un pulso digital cuando esta señal pasa por cero.

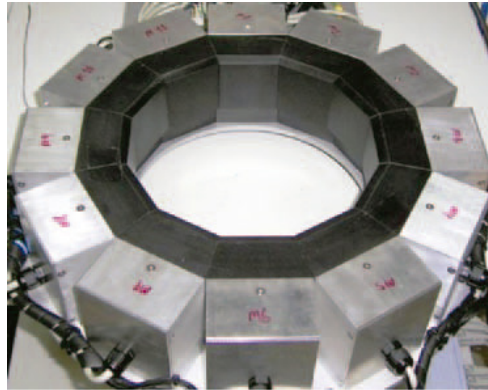
Este pulso digital es leído como señal de entrada de la FPGA, mediante el Core TDC que determina la etiqueta temporal y la almacena en los registros para que sea leída por el microprocesador MicroBlaze.

Una vez se conectan a la FPGA, en su interior se realiza una resta entre los valores de timestamps de cada canal, de forma que se obtiene un histograma que se emplazará en una posición temporal dependiendo la posición en la que se coloque la fuente radiactiva en el interior del anillo.



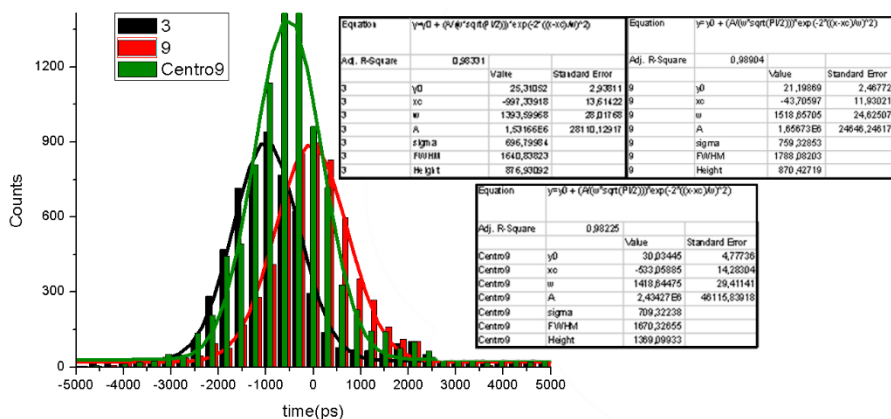
**Figura 73.** Esquema de configuración de la prueba con el anillo del sistema PET.  
**Fuente:** elaboración propia.

Con esta prueba se pretende demostrar que mediante la técnica del TOF, el TDC es capaz de discernir la posición en la que se encuentra la fuente radioactiva con una precisión inferior a 100 ps. Esto sería equivalente a la detección de un cáncer en una determinada posición dentro del anillo en un sistema PET en el que se introdujera la mama de una paciente a la que se le haya suministrado el radiofármaco. En la Figura 74 se muestra una imagen del anillo con el que se han realizado las pruebas.



**Figura 74.** Anillo de sistema PET empleado para las pruebas reales.  
**Fuente:** elaboración propia.

Como se muestra en la Figura 75, al emplazar la fuente en tres posiciones distintas dentro de la línea imaginaria que une los dos detectores que se están adquiriendo (LOR), el TDC determina como resultado de la prueba tres Gaussianas emplazadas en tres posiciones temporales proporcionales a las tres posiciones físicas en las que se ha colocado la fuente radiactiva. Cabe destacar que el FWHM de cada una ellas se corresponden con la resolución. Específicamente, se han estudiado los resultados generados por los detectores 3 y 9 del anillo en tres situaciones: colocando la fuente próxima al detector 3 (histograma de color negro), situándola próxima al detector 9 (histograma de color rojo) y en el centro del anillo (histograma de color verde).



**Figura 75.** Pruebas con un anillo de sistema PET real.  
**Fuente:** elaboración propia.

Como puede observarse, los resultados de FWHM son del orden de 1,5 ns y no de pico segundos. Esto es debido a que las resoluciones temporales que se están obteniendo en esta prueba dependen del jitter o fluctuaciones temporales que introducen al sistema el resto de dispositivos como los propios detectores PMT y el discriminador CFD.

## CAPÍTULO 9: CONCLUSIONES

En este trabajo se han expuesto los principios físicos, médicos y electrónicos que se emplean en los sistemas PET, se ha presentado el método del cálculo del TOF como un parámetro que puede proporcionar sustanciales mejoras en términos de resolución temporal y han sido testadas las capacidades que poseen las FPGAs para realizar medidas precisas de diferencias temporales mediante la implementación de un TDC. Se ha mostrado la capacidad que el sistema implementado posee para medir diferencias temporales inferiores a 100 ps con una configuración multicanal, lo que significa que el TDC posibilita la detección de partículas cancerosas con un error del orden de  $\pm 1,5$  cm. Asimismo, el Core TDC basado en líneas de retardo de Vernier modificadas, funciona correctamente al integrarlo dentro de una FPGA de bajo coste de forma que el sistema es reconfigurable a nivel hardware, es decir, mediante la reconfigurabilidad puede adaptarse a cualquier sistema médico de similares características.

En cuanto a la validación de las pruebas realizadas, se ha verificado el correcto funcionamiento del TDC multicanal de alta resolución temporal al realizar diferencias entre los cuatro canales implementados y obtener valores del orden de decenas de pico segundos a partir de la señal suministrada a la FPGA por el generador de funciones. Análogamente, al realizar la prueba con el anillo perteneciente a un sistema PET real colocando en su interior una fuente radiactiva, se han obtenido resoluciones muy superiores a las que se habían obtenido en las pruebas realizadas con el generador de funciones por lo que el TDC diseñado e implementado sí que cumple los objetivos fijados pero se deben realizar estudios y modificaciones en la electrónica anexa que interviene en el sistema, como el CFD o los detectores, para mejorar la resolución total del sistema PET con el que se han realizado las pruebas.

Con el objetivo de poder realizar una visualización en tiempo real del comportamiento del sistema implementado, se ha programado una aplicación para PC con LabVIEW que muestra el resultado de las diferencias de tiempo entre diferentes canales y realiza un estudio de los parámetros estadísticos que reflejan la calidad de las resoluciones temporales que se están obteniendo. De este modo, es posible reconfigurar diferentes parámetros de la arquitectura interna del TDC y poder observar en tiempo real si dichos cambios se traducen en una mejora del sistema.

Se ha cumplido otro de los objetivos al conseguir implementar un servidor web en la FPGA capaz de convivir con el sistema TDC desarrollado, que permite realizar una comunicación de alta velocidad con un navegador Web mediante el protocolo Ethernet. Esto le aporta al sistema una clara ventaja en el ámbito de las comunicaciones, ya que, hasta el momento, se estaba empleando comunicación serie que no permite altas tasas de transmisión de datos y puede presentar problemas al aumentar el número de canales a procesar.

Finalmente, se van a exponer las modificaciones que se encuentran en proceso de pruebas reales con el anillo con el objetivo de mejorar los resultados de resoluciones temporales del orden de 1,5 ns mostrados en el capítulo de Pruebas y Resultados. Las alternativas que se están considerando son la sustitución de los PMT por detectores

de nueva generación denominados Silicon PM (del inglés, Silicon Photomultipliers) que se caracterizan por ser más estables e introducir un menor jitter a la salida, pero, al ser una tecnología nueva, no están todavía suficientemente caracterizados. No obstante, tras realizar diferentes pruebas con estos nuevos detectores se han verificado mejoras en la resolución en torno a 500 ps. Por otro lado, el sistema CFD para discriminar los pulsos analógicos recibidos del detector que superan cierto umbral y enviarlos como pulsos digitales a la FPGA, también introduce un jitter elevado por lo que se está estudiando emplear otro tipo de sistemas que realizan una función similar como los DLED (del inglés, Double Leading Edge Discriminator).

En síntesis, se ha implementado un sistema TDC que se pretende emplear para el cálculo del TOF, cuyos beneficios son, entre otros: la disminución del tiempo de exposición del paciente al tener que emplear una dosis menor de radiofármaco, la obtención de mejor calidad de imagen reconstruida para un mismo número de detectores y la disminución de diagnósticos de falsos negativos (al paciente se le realiza la prueba y no se detecta el tumor), una de las principales causas de tumores no tratados correctamente.

## REFERENCIAS BIBLIOGRÁFICAS

- Bogdan, M., et al.** (2005). A 96-channel FPGA-based Time-to-Digital Converter (TDC) and fast trigger processor module with multi-hit capability and pipeline. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 554(1-3), pp.444-457.
- Borrajo-Sánchez, J., y Cabrero-Fraile, F. J.** (2010). Tomografía por emisión de positrones (PET): fundamentos y limitaciones tecnológicas. *Archivos de la Sociedad Española de Oftalmología*, 85(4), pp.129-130.
- Bostock, M., y Heer, J.** (2009). *Protovis: A graphical toolkit for visualization*. IEEE transactions on visualization and computer graphics, 15(6).
- Casey, M. E.** (2008). *Improving PET with HD-PET+ time of flight*. Siemens Molecular Imaging.
- Digilent.** (2013). *EDK Web Server*. Recuperado de: [https://reference.digilentinc.com/\\_media/atlys/atlys/atlys\\_axi\\_web\\_server\\_demo\\_v\\_1\\_02.zip](https://reference.digilentinc.com/_media/atlys/atlys/atlys_axi_web_server_demo_v_1_02.zip)
- Digilent.** (2016). *Atlys FPGA Board Reference Manual*. Recuperado de: [https://reference.digilentinc.com/\\_media/atlys/atlys/atlys\\_rm.pdf](https://reference.digilentinc.com/_media/atlys/atlys/atlys_rm.pdf)
- Dygraphs.** (2018). *API Reference*. Recuperado de: <http://dygraphs.com/>
- Flanagan, D.** (2006). *JavaScript: the definitive guide*. California, EE.UU.: O'Reilly Media, Inc.
- Gao, W., Gao, D., Hu-Guo, C., y Hu, Y.** (2011). Integrated High-Resolution Multi-Channel Time-to-Digital Converters (TDCs) for PET Imaging en *Biomedical Engineering, Trends in Electronics, Communications and Software*.
- Gonzalez Martinez, A. J., et al.** (2012). Design and first results of an innovative and dedicated breast PET. *Current Medical Imaging Reviews*, 8(2), pp.144-150.
- Guerra, P.** (2007). *Contribuciones al diseño e implementación de un sistema de alta resolución para tomografía por emisión* (Doctoral dissertation, Telecomunicacion).
- Haselman, M. D.** (2010). *Digital Pulse Timing in FPGAs for Positron Emission Tomography*.
- Henzler, S.** (2010). Time-to-digital converters. Berlín, Alemania: *Springer Science + Business Media*.
- Herraiz, J. L.** (2008). *Técnicas avanzadas de reconstrucción de imagen nuclear PET, X-CT y SPECT* (Trabajo Final de Máster).
- Karp, J. S., Surti, S., Daube-Witherspoon, M. E., y Muehlelehner, G.** (2008). The benefit of time-of-flight in PET imaging: Experimental and clinical results. *Journal of nuclear medicine*, 49(3), pp.462.

**Martí-Climent, J. M., Prieto, E., y García-Velloso, M. J.** (2009). Equipos híbridos en medicina nuclear. *Revista de Física Médica*, 10(1), pp.11-26.

**Millet, P., et al.** (2000). Similarity and robustness of PET and SPECT binding parameters for benzodiazepine receptors. *Journal of Cerebral Blood Flow y Metabolism*, 20(11), pp.1587-1603.

**Moliner, L., et al.** (2012). Design and evaluation of the MAMMI dedicated breast PET. *Medical physics*, 39(9), pp.5393-5404.

**Napolitano, P., Alimenti, F., y Carbone, P.** (2010). A novel sample-and-hold-based time-to-digital converter architecture. *IEEE Transactions on Instrumentation and Measurement*, 59(5), pp.1019-1026.

**OncoVision.** (2018). *MAMMI breast PET*. Recuperado de: <http://oncovision.com/es/mammi/>

**Philips Healthcare.** (2010). *TruFlight PET technology - Clinical implementation of time-of-flight*. Philips NM Clinical Science.

**Philips Healthcare.** (2010). Especificaciones del sistema PET GEMINI LXL. Recuperado de: <http://www.healthcare.philips.com/es/products/nuclearmedicine/products/geminilxl/>

**Siemens Medical Solutions.** (2018). *Inside Biograph TruePoint PET•CT*.

**Straayer, M. Z., y Perrott, M. H.** (2009). A multi-path gated ring oscillator TDC with first-order noise shaping. *IEEE Journal of Solid-State Circuits*, 44(4), pp.1089-1098.

**Surti, S., et al.** (2007). Performance of Philips Gemini TF PET/CT scanner with special consideration for its time-of-flight imaging capabilities. *Journal of Nuclear Medicine*, 48(3), pp.471.

**Surti, S., y Karp, J. S.** (2007). Design considerations for a limited-angle, dedicated breast, TOF PET scanner. *IEEE Nuclear Science Symposium Conference Record*, 5, pp.3667-3672.

**Torres, J., et al.** (2013). High resolution Time of Flight determination based on reconfigurable logic devices for future PET/MR systems. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 702, pp.73-76.

**Universidad de Sevilla** (2018). *Tutorial de LabVIEW*. Recuperado de: [http://www.gte.us.es/ASIGN/IE\\_4T/Tutorial%20de%20Labview.pdf](http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf)

**Wu, J.** (2009). An FPGA wave union TDC for time-of-flight applications. *IEEE Nuclear Science Symposium Conference Record*, pp.299-304.

**Xilinx.** (2011). *Spartan-6 Family Overview*. Recuperado de: [http://www.xilinx.com/support/documentation/data\\_sheets/ds160.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf)



**Xilinx.** (2015). *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*. Recuperado de: [http://www.xilinx.com/support/documentation/data\\_sheets/ds162.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf)





Ingeniería y Tecnología

