

UAH

Arquitectura Eficiente para el Procesamiento de un Array Ultrasónico basado en Técnicas de Codificación

**Máster Universitario en Sistemas Electrónicos Avanzados.
Sistemas Inteligentes
Departamento de Electrónica**

Presentado por:

D. Rodrigo García Ortiz

Dirigido por:

Dra. M^ª Carmen Pérez Rubio

Dr. Álvaro Hernández Alonso

Alcalá de Henares, a 29 de Mayo de 2014

Agradecimientos

En primer lugar me gustaría dedicarle este trabajo a mi madre y a mi hermano, por multitud de razones, pero especialmente por haber estado siempre para escucharme, aconsejarme y motivarme.

También me gustaría dedicárselo a Álvaro y M^a Carmen, no sólo por haber sido grandes tutores, sino por ser aún mejores personas y haberme brindado la oportunidad de dar pasos importantes en mi vida.

A Javi, Fran y Elena, que sin ellos la universidad no habría sido igual, y han conseguido llegar a ser personas muy importantes para mí, además de unos auténticos BONOBOS.

A mis compañeros de laboratorio Joaquín, Cesar y Chema. Gracias a ellos el año de beca paso volado. Gente con la que a uno le gusta compartir su tiempo.

¡A Balconete y a toda su gente! Un lugar donde siempre ir a olvidarse de todo en la mejor compañía.

A mis amigos de la capital. Por todos esos buenos ratos juntos y porque sin ellos ir al local no tendría sentido.

Y como no, a Cintia, por ser un gran apoyo siempre, por aguantarme incluso cuando no me aguanto ni yo, y en definitiva, por ser alguien tan importante para mí.

A todos, GRACIAS.

ÍNDICE

1	RESUMEN.....	9
2	ABSTRACT.....	10
3	INTRODUCCIÓN.....	11
3.1	Objetivo.....	11
3.2	Contexto.....	12
3.3	Estructura.....	12
4	Estado de la Cuestión.....	13
4.1	Técnicas <i>Phased Array</i>	13
4.1.1	Material EMFi.....	14
4.2	Técnicas de Acceso al Medio.....	16
4.2.1	FDMA.....	16
4.2.2	TDMA.....	16
4.2.3	CDMA.....	16
4.3	Arquitectura Hardware.....	19
4.3.1	DSPs.....	20
4.3.2	ASIC.....	21
4.3.3	FPGA.....	21
4.4	System on Chip (SoC).....	23
5	Arquitectura global del sistema propuesto.....	25
5.1	Diagrama de bloques.....	25
5.2	Bloque de emisión.....	26
5.2.1	Estrategia de codificación.....	26
5.2.2	Modulación BPSK.....	29
5.2.3	Características del array de ultrasonidos (US).....	30
5.3	Bloque de Recepción.....	32
5.3.1	Demodulación BPSK.....	32
5.3.2	Algoritmo de correlación eficiente de CSS.....	34
5.3.3	Representación del entorno explorado (formación de imágenes B-Scan).....	35
5.4	Especificaciones del sistema.....	36
6	Implementación del bloque emisor.....	37
6.1	Plataforma Hardware.....	37

6.1.1	Procesado Digital	37
6.1.2	Procesado Analógico.....	40
6.1.2.1	Digital to Analog Converter (DAC).....	40
6.1.2.2	Filtrado y Pre-amplificación	42
6.1.2.3	Amplificador de potencia.....	44
6.2	Implementación hardware	44
6.2.1	Esquema	44
6.2.2	Bloques Lógicos.....	45
6.2.2.1	Módulo Control_CSS.....	45
6.2.2.2	Módulo Prescaler	46
6.2.2.3	Módulo BPSK.....	47
6.2.2.4	Módulo FIFO	49
6.2.2.5	Módulo SUM32	50
6.2.2.6	Módulo DAC ctrl.	50
6.3	Simulaciones	53
6.4	Recursos FPGA.....	55
7	Implementación del bloque receptor	56
7.1	Plataforma Hardware.....	56
7.1.1	Procesado Digital	57
7.1.1.1	System On Chip (Microblaze)	58
7.1.2	Acceso a Memoria.....	61
7.1.2.1	Protocolo NPI.....	62
7.1.3	Convertor Analógico Digital.....	63
7.2	Implementación Hardware	64
7.2.1	Esquema	64
7.2.2	Bloques lógicos	64
7.2.2.1	Módulo ADC ctrl.	65
7.2.2.2	Módulo Demod. BPSK	66
7.2.2.3	Módulo Retardos.....	67
7.2.2.4	Módulo Correlador CSS.....	68
7.2.2.5	Módulo DMA_NPI	70
7.2.3	Microblaze.....	71
7.3	Simulaciones	73
7.3.1	Simulación adquisición	73

7.3.2	Simulación demodulación	73
7.3.2.1	Simulación de retardos	74
7.3.3	Simulaciones correlación	74
7.3.4	Simulación NPI	78
7.4	Recursos FPGA	81
8	Resultados experimentales	82
8.1	Planteamiento	82
8.2	Escenario de pruebas	82
8.3	Aplicación de comunicaciones	83
8.4	Resolución Angular	84
8.5	Profundidad de exploración	88
8.6	Resolución Axial	89
8.7	Escenarios compuestos	91
9	Conclusiones y trabajos futuros	94
9.1	Conclusiones	94
9.2	Trabajos Futuros	94
9.3	Publicaciones derivadas	95
10	Bibliografía	96

ÍNDICE DE FIGURAS

Fig. 1. Diagrama de generación de frente de ondas con un <i>array</i> de 6 elementos.....	13
Fig. 2. Lámina de EMFi observada al microscopio [Emf03].....	15
Fig. 3. Esquema genérico de la arquitectura interna de una FPGA.....	22
Fig. 4. Estructura típica de un CLB.....	22
Fig. 5. Arquitectura típica de un IOB.....	23
Fig. 6 Diagrama de bloques del sistema propuesto.....	25
Fig. 7. Proceso de generación de las macro-secuencias Ms_n	27
Fig. 8. Funciones de Auto-Correlación y Correlación-Cruzada de macro-secuencias Ms_n , obtenidas a partir de $N=4$ CSS de longitud $L=4$ y $w_0=32$	29
Fig. 9. Esquema de demodulación.	32
Fig. 10. (a) Señal modulada BPSK con amplitud 1, $f_c=1$ Hz y $f_c =20$ Hz. (b) Señal demodulada.	34
Fig. 11 Esquema del correlador N-ECSS usado en este trabajo [PSUJ12].....	34
Fig. 12 (a) Imágen médica. (b) Imagen entorno industrial.....	36
Fig. 13. Diagrama de conexión de elementos hardware del emisor.....	37
Fig. 14. Plataforma de desarrollo Genesys Digilent Inc.	38
Fig. 15. Módulo de conversión digital analógico Digilent PmodDA2 [Dig06].....	40
Fig. 16 Señal sinusoidal a 80 kHz obtenida a la salida del DAC121S101 con sobremuestreo de 20.....	41
Fig. 17. Ganancia y Fase del filtro.	43
Fig. 18. Retardo de grupo del filtro.....	43
Fig. 19. (a) Esquema de filtro y amplificador. (b) Ruteado PCB.....	43
Fig. 20. Resultado PCB real.....	44
Fig. 21. Bloques HD implementados sobre plataforma hardware.	44
Fig. 22. Diagrama de bloques del módulo Control_CSS.	46
Fig. 23. Diagrama de bloques del módulo Prescaler.....	47
Fig. 24. Diagrama de bloques del módulo BPSK.	48
Fig. 25. Diagrama de bloques del módulo FIFO.....	49
Fig. 26 Diagrama del módulo SUM32.....	50
Fig. 27. Cronograma de comunicación SPI entre el DAC y el dispositivo de control [TX13].	51
Fig. 28. Máquina de estados de funcionamiento del módulo DAC_ctrl [Dig06].	52
Fig. 29. Diagrama de bloques del módulo DAC_ctrl.....	53

Fig. 30. Cronograma de comunicación SPI.	53
Fig. 31. Cronograma de simulación de comunicación SPI entre emisor y DAC.	54
Fig. 32. Simulación de la emisión de un único código con un ángulo de 64°	54
Fig. 33. Flujo de ejecución del módulo receptor.	56
Fig. 34. Creación del SoC. Especificaciones del sistema.	59
Fig. 35. Bus interface entre <i>Microblaze</i> y los periféricos que forman el SoC.	60
Fig. 36. Asignación de puertos del SoC.	60
Fig. 37. Asignación de direcciones a los periféricos del SoC conectados al bus PLB.	61
Fig. 38. Interface del controlador de memoria multipuerto <i>MPMC</i>	61
Fig. 39. Conexión física del Data path del MPMC [Xil11].	62
Fig. 40. Cronograma protocolo NPI en modo ráfaga de 16 datos de 64 bits [Xil11].	63
Fig. 41. (a) PmodAD1. (b) Diagrama de bloques de PmodAD1.	64
Fig. 42. Diagrama de bloques HW del módulo receptor.	64
Fig. 43. Cronograma del protocolo de comunicación SPI con AD7476A [AD14].	65
Fig. 44. Máquina de estados del funcionamiento del módulo ADC_ctrl.	66
Fig. 45. Diagrama de bloques del proceso de demodulación.	67
Fig. 46. Diagrama de bloques del módulo Retardos.	68
Fig. 47. Representación del proceso de sincronización y habilitación de escritura de datos de la función de correlación obtenida del correlador CSS para una macro-secuencia de $N=8$ y $w_0=16$; $f_{sr}=80$ kHz y $O_f=20$	69
Fig. 48. Máquina de estados del funcionamiento del bloque DMA_NPI.	71
Fig. 49. Flujograma de la aplicación desarrollada sobre <i>Microblaze</i>	72
Fig. 50. Simulación de la comunicación con el ADC AD7476A.	73
Fig. 51. Simulación de la señal resultante tras realizar la demodulación de una secuencia de 4 elementos con modulación BPSK y portadora a 80 kHz.	73
Fig. 52. Simulación del proceso de demodulación.	74
Fig. 53. Simulación del bloque de Retardos.	74
Fig. 54. Simulación de entradas al correlador CSS y las funciones CCF y ACF a su salida para la recepción de la macro-secuencia Ms_4 formada por 4 secuencias de 4 bits y una w_0 de 16 bits.	75
Fig. 55. Funciones CCF y ACF para la recepción de la macro-secuencia Ms_4 formada por 4 secuencias de 4 bits y una w_0 de 16 bits.	76
Fig. 56. Escenario real de simulación.	76
Fig. 57. segmento de ACF de la macro-secuencia Ms_1 formada por 32 secuencias de 32 bits, w_0 de 354 bits, portadora $f_c=40$ kHz y frecuencia de muestreo $f_{sr}=400$ kHz obtenida con <i>ChipScope</i>	77

Fig. 58. segmento de ACF de la macro-secuencia Ms_1 formada por 32 secuencias de 32 bits, w_0 de 354 bits, portadora $f_c=40\text{ kHz}$ y frecuencia de muestreo $f_{sr}=400\text{ kHz}$ obtenida con <i>Matlab</i>	77
Fig. 59. (a) Pico de auto-correlación distorsionado con y flanco de subida de la señal de <i>enable</i> obtenido con <i>ChipScope</i> . (b) Correlación cruzada con distorsión y flanco de subida de la señal de <i>enable</i> obtenido con <i>ChipScope</i>	78
Fig. 60. Cronograma de protocolo NPI para escritura de 8 <i>word</i> en modo ráfaga [Xil11]. ...	78
Fig. 61. Simulación del proceso completo de escritura en memoria externa.	79
Fig. 62. Captura de <i>ChipScope</i> de proceso completo de escritura en memoria externa.	80
Fig. 63. Simulación del proceso de escritura de una ráfaga de 8 <i>word</i>	80
Fig. 64. Captura de <i>ChipScope</i> del proceso de escritura de una ráfaga de 8 <i>word</i>	81
Fig. 65. Escenario base donde se han desarrollado las pruebas reales.	83
Fig. 66. Interfaz gráfica de la aplicación de comunicaciones entre el PC y la FPGA de recepción.	84
Fig. 67. Líneas A-scan con: Reflector 1 a 40 cm y -85° ; Reflector 2 a 40 cm y 85°	85
Fig. 68. Líneas A-scan con: Reflector 1 a 40 cm y -75° ; Reflector 2 a 40 cm y 75°	86
Fig. 69. Líneas A-scan con: Reflector 1 a 40 cm y -25° ; Reflector 2 a 40 cm y 25°	87
Fig. 70. Líneas A-scan con: Reflector 1 a 40 cm y -15° ; Reflector 2 a 40 cm y 15°	88
Fig. 71. Imagen B-scan con Reflector a 50 cm y 0° , contraste de -3dB (a). Imagen B-scan con Reflector a 80 cm y 0° , contraste de -3dB (b). Imagen B-scan con Reflector a 110 cm y 0° , contraste de -3dB (c). Imagen B-scan con Reflector a 120 cm y 0° , contraste de -3dB (d). ...	89
Fig. 72. Líneas A-scan correspondientes al código 5 y código 4 representadas en dB tomando como 0 dB el máximo del pico de auto-correlación de menor amplitud. Reflector 1: 50 cm y -4° ; Reflector 2: 51.5 cm y 12°	90
Fig. 73. Líneas A-scan correspondientes al código 5 y código 4 representadas en dB tomando como 0 dB el máximo del pico de auto-correlación de menor amplitud. Reflector 1: 50 cm y -4° ; Reflector 2: 50.5 cm y 12°	91
Fig. 74. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 85 cm, 0° , de madera y ancho de 10 cm; Reflector 2 a 35 cm, 40° , de madera y ancho de 10 cm.	92
Fig. 75. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 50 cm, -40° , de metal y ancho de 6 cm; Reflector 2 a 30 cm, 20° , de metal y ancho de 6 cm.	92
Fig. 76. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 30 cm, -40° , de metal y ancho de 6 cm; Reflector 2 a 50 cm, 20° , de metal y ancho de 6 cm.	93

1 RESUMEN

En este trabajo se presenta una implementación basada en SoC (*System on Chip*), para un sistema de generación de imágenes de ultrasonidos en aire mediante el uso de macro-secuencias, derivadas de conjuntos de secuencias complementarias (CSS), con el fin de realizar la exploración de varios sectores simultáneamente.

Mediante la concatenación de secuencias CSS interpoladas con ceros se obtienen macro-secuencias cuyas funciones de auto-correlación y correlación cruzada presentan una zona de correlación cero (ZCZ), la cual puede ser ajustada para lograr una ventana libre de interferencias en el área a analizar.

Los resultados de correlación obtenidos a partir de los ecos de las macro-secuencias recibidas proporcionan las líneas A-scan de la imagen B-scan final. En comparación con otros esquemas de codificación, que también proporcionan ZCZ, esta solución permite reducir la carga computacional y por lo tanto el número de recursos hardware necesarios, lo que facilita un procesado en tiempo real.

Se presenta la implementación hardware sobre una plataforma FPGA de Xilinx de los bloques de emisión y recepción del sistema de generación de imágenes de ultrasonidos, para un *array* US de 8 elementos y la exploración simultánea de hasta 32 sectores diferentes, con una profundidad de 1.5 m y una apertura de $\pm 64^\circ$.

2 ABSTRACT

This work presents a SoC-based implementation for ultrasonic imaging systems in airborne transmission which uses macro-sequences derived from complementary set of sequences (CSS) to achieve simultaneous beam steering in several directions. By means of CSS concatenation and zero padding a new macro-sequence can be obtained which achieves zero auto- and cross-correlation zone (ZCZ) characteristics that can be adjusted to provide interference-free CDMA within the scanning area. The correlation results with the emitted macro-sequences provide the A-scan lines of the final B-scan image. In contrast to other encoding schemes with similar ZCZ size, the one proposed here allows a hardware design that further minimizes the computational load to make real-time processing more feasible. The hardware implementation in a Xilinx FPGA of all the stages forming the emission and reception modules of the imaging ultrasonic system is presented. The array has 8 elements and permits the simultaneous analysis of 32 different sectors, with a 1.5m depth and $\pm 64^\circ$ aperture.

3 INTRODUCCIÓN

3.1 Objetivo

Los sistemas ultrasónicos *Phased Array* (PA) consisten en un conjunto de elementos que son activados con distintos retardos temporales, consiguiendo así conformar un haz electrónico que puede orientarse en la dirección adecuada [RS83]. Se logra de este modo realizar un barrido del medio sin necesidad de mover el transductor.

Para aumentar la tasa de generación de imágenes, recientemente se han propuesto trabajos que aplican técnicas de codificación a las señales emitidas por el *array*, de modo que se pueden superponer las informaciones correspondientes a cada línea de la imagen a representar [DHJ⁺11]. La prestación del sistema final depende en gran medida de las propiedades de correlación de los códigos empleados. Por otro lado, el empleo de estas técnicas implica el desarrollo de algoritmos de procesamiento más complejos y el empleo de códigos de longitud elevada. Esto conlleva una elevada carga computacional, cuya implementación práctica puede llegar a superar los límites impuestos por la necesidad de trabajar en tiempo real o demandar el uso de plataformas de elevado coste.

Para dar solución a dichos inconvenientes se suelen plantear dos líneas de actuación. Por un lado, la utilización de nuevos esquemas de codificación, como los propuestos en [PSU⁺12] [ZYH05] [CY06], con zonas en las que la correlación es nula. La particularidad de estos códigos, derivados en su mayoría de conjuntos de secuencias complementarias (CSS) [TL72], reside en que sus funciones de correlación aperiódicas presentan una ventana libre de interferencias (IFW, *Interference Free Window*) alrededor del origen. De este modo, es posible mitigar en gran medida las interferencias inter-símbolo (ISI) y por acceso múltiple (MAI), siempre y cuando los retardos relativos entre las distintas recepciones estén dentro de esa IFW. Por otro lado, la segunda línea de trabajo consiste en la definición y posterior implementación hardware de arquitecturas eficientes para el detector asociado a la codificación elegida, típicamente a partir de dispositivos FPGA (*Field-Programmable Gate Array*).

Este trabajo presenta una arquitectura eficiente basada en SoC para el procesamiento de señales de ultrasonidos en PA codificadas con macro-secuencias formadas por CSS, que permiten la emisión y procesamiento en paralelo de entornos de larga distancia y transmisión por aire. El esquema de codificación utiliza técnicas CDMA con el fin de disminuir el tiempo de adquisición con respecto a un PA convencional e incrementar el SNR. La arquitectura se basa en hardware reconfigurable de bajo coste FPGA (*Field Programmable Gate Array*) debido a la flexibilidad que otorga a la hora de modificar los parámetros del sistema PA. La electrónica necesaria para el escaneo de 32 sectores diferentes con un *array* de 8 elementos se ha desarrollado sobre dos placas basadas en FPGA, una emisora y otra receptora. El procesamiento de alto nivel se ha llevado a cabo sobre un PC. Para la transmisión de los datos desde la FPGA al PC se ha implementado una comunicación Ethernet con protocolo TCP/IP controlada por el procesador *Microblaze*.

Las aplicaciones de estos tipos de sistemas son variadas, desde la generación de imágenes ultrasónicas en medicina [SPR91] [CK08] o la realización de ensayos no destructivos hasta la construcción de mapas del entorno en robótica móvil [PNI⁺08].

3.2 Contexto

El presente trabajo ha sido llevado a cabo dentro del grupo de investigación GEINTRA, perteneciente al Departamento de Electrónica de la Universidad de Alcalá, que goza de amplia experiencia en el desarrollo de sistemas basados en entornos ultrasónicos. Concretamente, este trabajo ha sido posible gracias a la financiación de la Universidad de Alcalá (proyecto SIMULTANEOUS, ref. UAH2011/EXP-003); del Ministerio de Ciencia e Innovación (proyecto LEMUR ref. TIN2009-14114-C04-01); y del Ministerio de Economía y Competitividad (proyecto LORIS, ref. TIN2012-38080-C04-01, y proyecto DISSECT-SOC, ref. TEC2012-38058-C03-03).

3.3 Estructura

Este trabajo presenta el desarrollo de una arquitectura eficiente en FPGA para la implementación en tiempo real de los algoritmos de procesamiento de señales ultrasónicas codificadas, a partir de secuencias derivadas de CSS, para el barrido simultáneo en todas las direcciones con sistemas *Phased Arrays* (PA). El documento se encuentra organizado de la siguiente forma: en el apartado 1 se describe el estado de la cuestión; en el apartado 5 se muestra el esquema del sistema que ha sido desarrollado junto con una descripción de los elementos que intervienen en el diseño; a continuación, en el apartado 6 y 7 se describen en detalle los bloques de emisión y recepción implementados sobre la plataforma hardware; seguidamente en el apartado 8 se muestran los resultados experimentales obtenidos; y por último, en el apartado 9 se mencionan las conclusiones y los trabajos futuros.

4 Estado de la Cuestión.

4.1 Técnicas *Phased Array*

En comparación con transductores ordinarios de un único elemento, los *Phased Arrays* (PA) permiten generar una el haz ultrasónico en la dirección deseada sin necesidad de desplazar, manual o mecánicamente, el transductor. Los sistemas PA además permiten cambiar el patrón de emisión cambiando los parámetros geométricos del *array*. Por otro lado, los transductores basados en un único elemento tan solo pueden cambiar el patrón de emisión modificando la excitación del transductor, ya que a mayor frecuencia en la señal de excitación se obtienen patrones de emisión más estrechos. Las técnicas *Phased Array* representan un método atractivo para obtener imágenes de ultrasonidos, muy frecuentemente utilizadas en imagen médica. Además, los equipos de test basados en ultrasonidos son muy utilizados en la industria para la realización de ensayos no destructivos NDT (*Non-Destructive Testing*).

Para deflectar el haz en el ángulo deseado se lleva a cabo la estimulación de cada uno de los elementos que forman el *array* aplicándoles un retardo determinado y de esta manera generar un frente de onda en el sector deseado tal y como se muestra en la siguiente imagen (véase la Fig. 1):

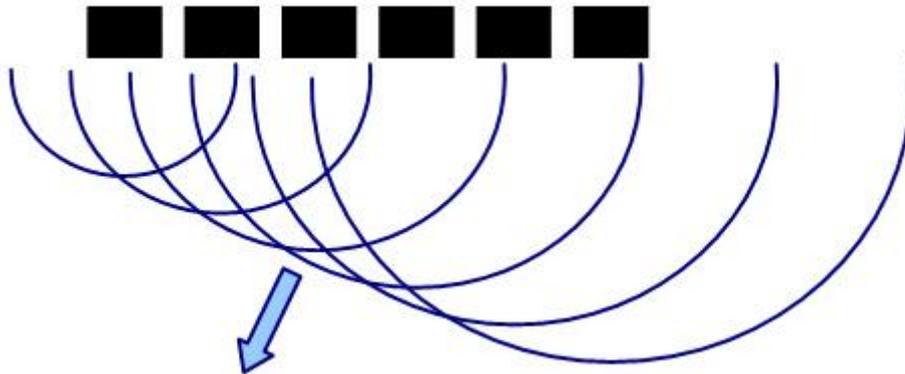


Fig. 1. Diagrama de generación de frente de ondas con un *array* de 6 elementos¹.

Las características principales son:

- Se usa tanto bajo el agua como en contacto con medios de alta densidad tales como tejidos humanos o sobre metales, donde el sonido se propaga a mayor velocidad y menor atenuación en comparación con las transmisiones por aire.
- La señal de excitación de los elementos del *array* es un pulso, lo que implica que la relación señal ruido SNR (*Signal to Noise Ratio*) debe ser alta. De esta manera estas técnicas se usan más frecuentemente en aplicaciones donde el rango de exploración es reducido (cm) que en exploraciones de larga distancia.

¹ Imagen obtenida de: <http://www.the-ndt-guy.com/?p=102>

- La desventaja más importante que presentan los sistemas PA es su baja tasa de imágenes ya que es necesario realizar una emisión para la obtención de cada línea de las que conforman la imagen de ultrasonidos. Esto varía en función del número de sectores angulares explorados.

La solución que se adopta para aumentar la tasa de generación de imágenes consiste en combinar las técnicas PA con CDMA para de esta manera, realizar la exploración de la zona deseada llevando a cabo una única emisión.

4.1.1 Material EMFi

Existen gran cantidad de transductores de ultrasonidos disponibles en el mercado, donde los transductores piezocerámicos son los más comunes, así como los más desarrollados tecnológicamente en comparación con el resto. No obstante, existen importantes líneas de investigación y desarrollo para crear materiales que consigan mejorar las carencias que éstos presentan. Las principales características que se buscan en materiales que vayan a ser usados en transductores ultrasónicos son las siguientes: ancho de banda elevado, bajo coste y sensibilidad alta.

De esta manera, existen materiales como los polímeros con propiedades piezoeléctricas, esta propiedad consiste en que el material genera una diferencia de potencial cuando se ejerce una fuerza sobre él y viceversa. Dentro de este tipo de material, destacan el PVDF (polifluoruro devinidileno) así como los piezoeléctricos de ingeniería. A este último es al que pertenece el material EMFi [Emf03], el cual está formado por una serie de láminas cuya estructura celular es de espuma de polipropileno.

El comportamiento piezoeléctrico del EMFi es debido a su estructura interna, ya que está formado por cavidades con carga permanente, las cuales se comportan como dipolos. En la siguiente imagen (véase la Fig. 2) se puede apreciar al microscopio una sección de una lámina EMFi [Emf03]. Si se aplica una fuerza sobre el material el momento dipolar cambia, lo que provoca que se vuelva sensible a las fuerzas normales a su superficie.

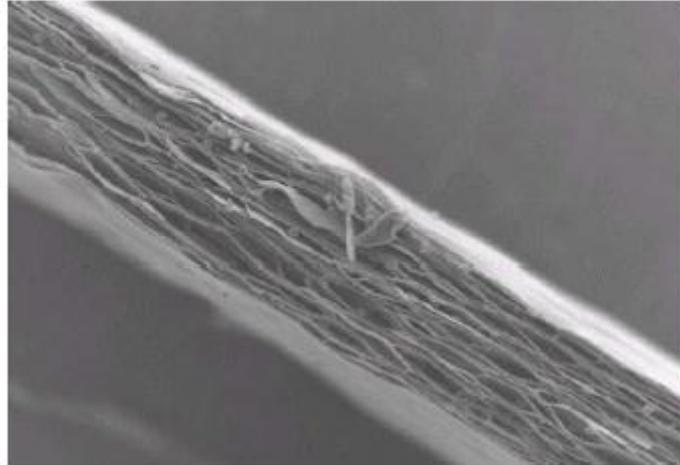


Fig. 2. Lámina de EMFi observada al microscopio [Emf03].

El proceso que se lleva a cabo para conseguir que las cavidades que forman el material se comporten como dipolos es sometiendo al EMFi a descargas de más de 10 kV, provocando la ruptura dieléctrica del material. A partir de ese momento las cavidades quedan cargadas de forma permanente. Todos los dipolos de su estructura se encuentran orientados en la misma dirección al finalizar el proceso de fabricación. De esta manera no es necesario aplicar un campo eléctrico externo para polarizar los dipolos. El coeficiente piezoeléctrico d_{33} del material EMFi ronda los 25-250 pC/N, el cual es superior al de los PVDF. Este coeficiente da idea de la sensibilidad del material ya que refleja la variación de la carga en función de la fuerza.

Por otro lado la impedancia acústica del EMFi es reducida si se compara con otros materiales. Esto se traduce en que su comportamiento es bueno al utilizarse como transductor de ultrasonidos en aire. Asimismo el ancho de banda de este material es elevado (hasta 300 kHz) y presenta una respuesta prácticamente lineal y uniforme hasta los 200 kHz. Otra de las ventajas de este material es que el patrón de radiación se determina por la frecuencia de trabajo, así como por la geometría del transductor, y ambas son configurables por el usuario.

El principal inconveniente que presenta el material EMFi es la alta tensión de excitación necesaria para lograr niveles aceptables de presión sonora. Las tensiones recomendadas de uso son entre 300 y 600 Vpp, aunque en este trabajo se ha comprobado que con tensiones menores el funcionamiento es aceptable. Este inconveniente no es único del EMFi ya que muchos de los transductores comerciales también requieren de altas tensiones de excitación.

4.2 Técnicas de Acceso al Medio.

Uno de los aspectos que tienen más importancia en los sistemas de comunicaciones es la forma en la que se accede al medio de comunicación. Es preciso el uso de técnicas que permitan acceder al mayor número de usuarios posible a un mismo medio, sin que éstos se interfieran mutuamente. Es por esto que surgen técnicas de acceso múltiple al medio.

Las principales técnicas de acceso múltiple son:

- TDMA: Acceso múltiple por división de tiempos.
- FDMA: Acceso múltiple por división de frecuencia.
- CDMA: Acceso múltiple por división de códigos.

4.2.1 FDMA

Con esta técnica el ancho de banda de comunicaciones queda dividido en bandas de frecuencias menores, manteniendo una determinada distancia entre ellas y evitando de esta manera interferencias entre los canales de comunicación situados en bandas adyacentes. A cada usuario se le asigna una banda de frecuencia, la cual puede ser utilizada por éste de forma permanente. No obstante, esa banda de frecuencia no se aprovecha convenientemente siempre y cuando el canal no esté utilizándola. Por otro lado, esta técnica no precisa de la sincronización entre usuarios para la utilización de un canal.

4.2.2 TDMA

En TDMA diferentes usuarios comparten una misma banda de frecuencia. No obstante, esta es accedida por cada uno de ellos en un instante de tiempo determinado *time slot*. Cada usuario sólo podrá transmitir durante el intervalo de tiempo que le corresponde. Es necesario mantener tiempos de guarda para evitar solapamiento de información entre las emisiones de los diferentes usuarios, así como sincronizar de forma precisa cada uno de estos usuarios para evitar interferencias entre ellos. Esto provoca que esta técnica sea compleja de implementar y lenta.

4.2.3 CDMA

Esta técnica permite el acceso al medio de múltiples usuarios mediante la asignación de códigos pseudo-aleatorios. A cada usuario se le asigna un código que presenta propiedades de ortogonalidad con el resto de códigos asignados a los demás usuarios. Esto permite que todos ellos puedan acceder al medio de forma simultánea sobre el mismo canal.

Dichos códigos son usados para transformar la señal de un usuario en una señal de espectro ensanchado mediante la técnica conocida como *Spread Spectrum* (Espectro Extendido). Para lograr que la información correspondiente a cada usuario pueda ser separada y decodificada es necesario que tanto el emisor como el receptor conozcan el correspondiente código.

Este tipo de codificación presenta varias ventajas entre las que cabe destacar:

- **Una mayor inmunidad al ruido.** Al realizar una codificación con CDMA, el espectro de la señal queda distribuido en un ancho de banda mayor que la señal en

banda base, a esto se le conoce como *spread spectrum*. Esto provoca que interferencias producidas a frecuencias determinadas afecten de manera reducida y constante. Si no se utilizara esta técnica, cualquier interferencia producida en banda base podría provocar la imposibilidad de recuperación de la información. Asimismo, es posible el aumento de la potencia de emisión de la información codificada emitida por un usuario sin que esto afecte al resto.

- **Mayor capacidad.** La tecnología CDMA permite que un mayor número de usuarios compartan las mismas frecuencias.
- **Mayor seguridad.** Resulta complicado descifrar la información si no se conoce el código utilizado.

Uno de los métodos utilizados para la decodificación de la información es la correlación con el código correspondiente [P09]. La operación de correlación entre dos señales se lleva a cabo con el objetivo de medir su grado de similitud.

Dada una familia A con M códigos binarios de longitud $L\{A = a_m[l] \in \{-1,1\}; 0 \leq m \leq M - 1; 0 \leq l \leq L - 1\}$, la función de auto-correlación (ACF) discreta periódica es (1)

$$R_{a_m, a_s}[\tau] = \sum_{l=0}^{L-1} a_m[l] a_s[l + \tau] \quad (1)$$

La expresión anterior representa la ACF cuando $m=s$ y la función de correlación cruzada (CCF) periódica discreta cuando $m \neq s$. Las secuencias a_m y a_s son periódicas, es decir, $a_m = (\dots, a_m[0], a_m[1], \dots, a_m[L-1], a_m[0], a_m[1], \dots, a_m[L-1], \dots)$. La función de correlación aperiódica viene dada por (2):

$$C_{a_m, a_s}[\tau] = \begin{cases} \sum_{l=0}^{L-1-\tau} a_m[l] a_s[l + \tau], & 0 \leq \tau \leq L - 1 \\ \sum_{l=0}^{L-1+\tau} a_m[l - \tau] a_s[l], & 1 - L \leq \tau \leq 0 \\ 0, & |\tau| \geq L \end{cases} \quad (2)$$

Cuando $m=s$ se obtiene la ACF aperiódica, y cuando $m \neq s$ la CCF aperiódica. Normalmente, sólo se considera la función de correlación aperiódica en el rango $0 \leq \tau \leq L - 1$.

Un criterio usado de forma común para evaluar el comportamiento de una familia de códigos es el cálculo de su cota máxima de correlación (3):

$$\theta = \max\{\theta_{AC}, \theta_{CC}\} \quad (3)$$

Donde θ_{AC} representa el valor máximo de los lóbulos laterales obtenido de las auto-correlaciones de los M códigos de la familia (4), y θ_{CC} es el pico máximo de las funciones de correlación cruzada (5).

$$\theta_{AC} = \max \left\{ \frac{|R_{a_m, a_m}[\tau]|}{R_{a_m, a_m}[0]}; \forall m \in [0, \dots, M-1]; \forall \tau \neq 0 \right\} \quad (4)$$

$$\theta_{AC} = \max \left\{ \frac{|R_{a_m, a_s}[\tau]|}{R_{a_m, a_m}[0]}; \forall m, s \in [0, \dots, M-1]; m \neq s; \forall \tau \right\} \quad (5)$$

Si se lleva a cabo la sustitución en las expresiones anteriores de R_{a_m, a_s} , R_{a_m, a_m} por C_{a_m, a_s} y C_{a_m, a_m} , respectivamente, se obtienen las cotas para el caso de las funciones de correlación aperiódica.

Un grupo de códigos se comporta de forma ideal cuando en su ACF para $\tau=0$ aparece un pico con una amplitud elevada y es 0 en $\tau \neq 0$. Además estos códigos deberían estar incorrelados entre sí, lo que implica su CCF sería 0 en todo momento. No obstante, no existen códigos con estas características.

Se le llama interferencias inter-símbolo (ISI) a los picos secundarios que aparecen en la ACF; a las interferencias que aparecen cuando dos o más códigos no incorrelados son transmitidos por un mismo canal se las conoce como interferencias por acceso múltiple (MAI). Se debe tratar usar códigos que tengan cotas de correlación lo más bajas posible con el fin de minimizar estas interferencias.

Se dice que dos códigos son ortogonales si (6):

$$R_{a_m, a_s} = \begin{cases} L, & \tau = 0 & m = s \\ 0, & \tau \neq 0 & m \neq s \end{cases} \quad (6)$$

Es preciso notar que la expresión anterior es nula únicamente para desplazamiento cero. Esto es complejo de conseguir ya que para ello emisor y receptor deben estar perfectamente sincronizados. Asimismo, existen efectos de reflexión que producirían que los códigos se recibieran en distintos instantes de tiempo acabando con la ortogonalidad entre códigos. Esta situación, en la que la señal se puede propagar por distintos caminos (debido a la reflexión de objetos o del suelo), recibe el nombre de *propagación multicamino*.

Para entornos que no requieren de un sincronismo estricto surgen dos nuevos conceptos:

1. Ortogonalidad generalizada: Dos códigos son ortogonales generalizados cuando presentan una zona w_0 que representa una zona alrededor del origen donde las funciones de correlación tienen valor cero. Cuanto mayor sea w_0 mejores características presentarán los códigos. El grado de ortogonalidad generalizada de los códigos viene dada por w_0 y es conocida como *zona de correlación cero* (ZCZ, *Zero Correlation Zones*) o *ventana libre de interferencias* (IFW, *Interference Free Window*). Concretamente, la IFW hace referencia a la zona alrededor del origen $2w_0+1$.
2. Cuasi-ortogonalidad generalizada: En este caso, las interferencias alrededor del origen no son nulas, sino que se encuentran por debajo de un umbral tolerado por el sistema. A la *zona de reducidas interferencias* (LCZ, *Low Correlation Zone*) se le denomina w_q .

4.3 Arquitectura Hardware

Existen varias alternativas tecnológicas de posible aplicación a este trabajo. A continuación se describen las principales características de las tecnologías más destacadas:

- Los microcontroladores son dispositivos cuyo objetivo es el gobierno de uno o varios procesos físicos. Estos suelen incorporar una CPU (no muy rápida), memorias tipo ROM/PROM/EPROM/EEPROM/FLASH para contener el programa de control y RAM para almacenar los datos (muy rápidas y de baja capacidad de almacenamiento), así como periféricos tales como temporizadores, canales serie y paralelo, convertidores A/D y D/A, etc. Esta plataforma es de bajo coste y permite el desarrollo de casi cualquier funcionalidad. No obstante, debido a su carácter genérico, no optimiza recursos ni tiempos de operación, por lo que carece de la eficiencia de otros dispositivos.
- Otra alternativa son los procesadores digitales de señal, DSPs, los cuales presentan una circuitería interna optimizada para la implementación de funciones tales como el filtrado, la correlación, el análisis espectral... Su estructura interna básica responde fundamentalmente a una arquitectura Harvard, pensada para la óptima ejecución de instrucciones y realización de operaciones aritméticas. La zona de manejo de datos de la CPU suele estar especialmente diseñada, disponiendo de múltiples ALU y multiplicadores, capaces de realizar varias operaciones aritméticas en un único ciclo máquina del sistema. Presentan la ventaja de poder ser programados en lenguaje de alto nivel. Sin embargo, el número de operaciones por ciclo de reloj que pueden llegar a realizar es mucho menor que el que se consigue en una plataforma de hardware reconfigurable FPGA, siendo a veces complicado la ejecución en tiempo real.
- Los sistemas específicos ASIC son circuitos digitales que se diseñan para realizar una función determinada. Estos dispositivos permiten una optimización máxima de los recursos y de procesado. Pero tiene varios inconvenientes, ya que el dispositivo no puede emplearse en una tarea diferente de para la que se desarrolló debido a lo estricto del diseño. De la misma manera, la etapa de pruebas y testado es costosa y compleja.
- Los sistemas de hardware reconfigurable FPGA (*Field Programming Gate Array*) son una alternativa que permite una solución intermedia entre la flexibilidad de los

procesadores de uso general y las plataformas de hardware específico. Éstos permiten abarcar una gran cantidad de procesos simultáneos dedicados al igual que con los sistemas ASIC, a costa de una menor eficiencia en el aprovechamiento de recursos así como un mayor consumo de energía.

4.3.1 DSPs

Un DSP (*Digital Signal Processor*) es un sistema basado en un microprocesador que posee un conjunto de instrucciones, un hardware y un software optimizados para el procesamiento de señales digitales. Para poder realizar estas operaciones de procesamiento como son el filtrado o la FFT es preciso realizar operaciones aritméticas complejas y repetitivas, trabajando en tiempo real. Por esta razón, la arquitectura de los DSPs está pensada para realizar de forma eficiente el mayor número de operaciones posible.

Estos dispositivos suelen estar basados en la arquitectura Harvard. Esta arquitectura consiste en separar la memoria interna de programa de la de datos. De esta manera es posible acceder de forma simultánea a través de dos buses diferentes, tanto a los datos como a las instrucciones consiguiendo una mayor velocidad en la ejecución del programa.

La CPU está formada por varios registros de uso general así como una serie de ALUs y multiplicadores, que permiten llevar a cabo operaciones aritméticas y lógicas en paralelo. Para realizar estas operaciones la CPU cuenta con una serie de unidades funcionales. Otra de las características de la arquitectura de los DSPs es el uso de VLIW (*Very Long Instruction Word*). Basándose en que la CPU consta de varias ALUs y multiplicadores, se realiza una lectura de varias instrucciones cada vez que se lleva a cabo un acceso a memoria de programa. Permitiendo la ejecución de varias instrucciones de forma simultánea. El tamaño de esta palabra es variable y la ejecución, de las instrucciones en paralelo depende de los recursos que se estén utilizando en cada momento.

Otro elemento que suele ser común son los bloques DMA (*Direct Memory Access*), que permiten el acceso a memoria externa y a los periféricos de forma eficiente. Puesto que la mayor parte de las operaciones relacionadas con el procesamiento de la señal en tiempo discreto, como la convolución o la transformada de Fourier, se realizan mediante la implementación de suma de productos, estos dispositivos vienen preparados para realizar estas operaciones específicas MACs y el fabricante da información acerca de cuál es el ratio con el que el DSP es capaz de realizar esta operación.

Cuando las frecuencias de muestreo crecen por encima de unos pocos MHz, un DSP tiene que trabajar mucho para transferir los datos sin ninguna pérdida. Esto es porque el procesador debe utilizar los recursos compartidos como buses de memoria, o incluso el núcleo del procesador para otras tareas como interrupciones y pueden llegar a colapsarse por algún tiempo. El DSP puede tomar un programa estándar de C y ejecutarlo. Este código C puede tener un alto nivel de ramificación y de toma de decisiones, como por ejemplo, las pilas de protocolos de sistemas de comunicaciones. Esto es difícil de implementar dentro de una FPGA.

Un DSP está diseñado para ofrecer el re-uso de las unidades de procesamiento, por ejemplo, un multiplicador que se utiliza para el cálculo de una FIR puede ser re-utilizado por otra rutina

que calcula la FFT. Es por esto que si se requiere un cambio de contexto importante, el DSP puede implementarlo por bifurcación a una parte nueva del programa. Por el contrario, una FPGA necesita construir recursos dedicados para cada configuración.

4.3.2 ASIC

Los circuitos integrados ASIC (*Application-Specific Integrated Circuit*) son circuitos diseñados para una aplicación específica, es decir, los ASIC son fabricados con toda la funcionalidad requerida para un determinado diseño. Estos dispositivos pueden contener funciones analógicas, digitales y combinaciones de ambas. Son dispositivos con altas prestaciones pero sólo resultan rentables para fabricaciones masivas ya que sus costes de diseño son muy elevados.

Según el método seguido en su diseño se tienen:

Full-Custom: El diseño se lleva a cabo totalmente a medida. Se consiguen áreas reducidas, con el consiguiente abaratamiento de costes y un funcionamiento optimizado. No obstante, presenta como inconvenientes un elevado coste y tiempo de desarrollo, mayor complejidad en las herramientas de desarrollo así como la necesidad de diseñadores expertos.

Semi-Custom: El dispositivo se diseña a partir del interconexión entre bloques funcionales existentes. De esta manera se logra el abaratamiento de costes y la reducción de los tiempos de desarrollo.

4.3.3 FPGA

Las FPGAs son dispositivos de silicio constituidos por bloques lógicos cuya interconexión y configuración es programable por el usuario con el fin de conseguir la funcionalidad deseada. Los bloques lógicos presentan una funcionalidad reducida por si solos, pero gracias a la gran cantidad de éstos es posible implementar funcionalidades complejas. Asimismo, las FPGAs disponen de células de memoria de configuración, denominados CMC (*Configuration Memory Cell*), distribuidas a lo largo de todo el chip, las cuales almacenan toda la información necesaria para configurar los elementos programables. Estas células de configuración suelen consistir en bloques de memoria RAM y son inicializadas en el proceso de carga del programa de configuración.

Las FPGAs fueron introducidas por Xilinx en 1985 y fueron creadas para proporcionar una solución intermedia entre los ASIC y lo CPLDs (*Complex Programmable Logic Device*). Las principales ventajas de las FPGAs frente a los ASICs son: el tiempo, y por lo tanto el coste, de un llevar a cabo un desarrollo utilizando tecnología basada en FPGAs es mucho menor que si se emplean ASICs; Por otro lado, las FPGAs son programadas por el usuario utilizando lenguajes de descripción hardware, lo cual simplifica de manera considerable la complejidad de los diseños.

Las FPGAs presentan tres tipos de elementos programables:

- 1) Bloques lógicos.

- 2) Bloques de entrada salida.
- 3) Matrices de interconexión.

En la Fig. 3 se muestra el esquema genérico de la arquitectura interna de una FPGA.

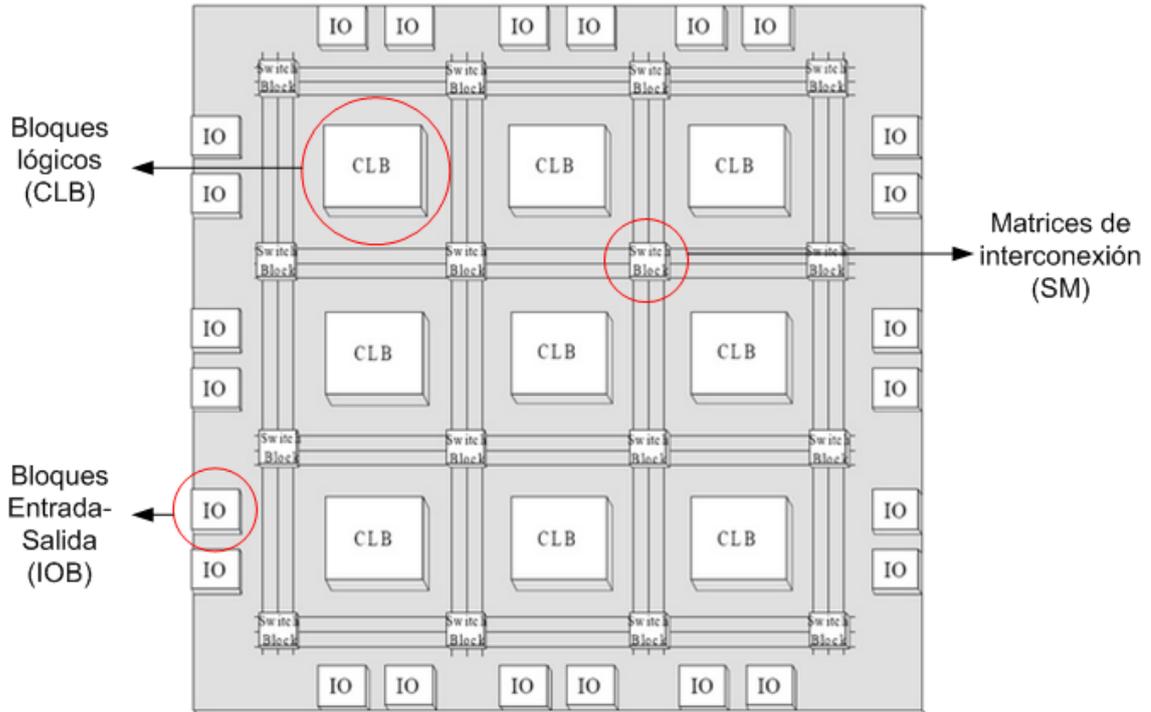


Fig. 3. Esquema genérico de la arquitectura interna de una FPGA².

CLB (Configurable Logic Block): cada uno de estos bloques está formado por elementos de lógica combinacional (L.U.T.) y un registro que puede ser configurado en modo *latch* o *flip-flop*. La lógica combinacional permite implementar funciones booleanas a partir de las entradas. La salida de las LUT va a un registro cuya salida va a un multiplexor, o bien directamente al multiplexor. De esta manera es posible configurar las salidas del CLB como secuenciales o combinacionales. En la Fig. 4 se muestra la estructura típica de un CLB.

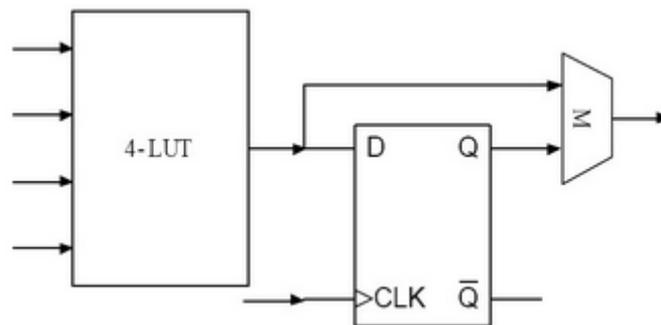


Fig. 4. Estructura típica de un CLB³

² Imagen obtenida de: <http://www.edafun.com/component/k2/item/77-whats-fpga?.html>

IOB (*Input-Output Block*): estos bloques se sitúan en los extremos de la FPGA y como su nombre indica son bloques de entrada/salida. Cada bloque puede ser configurado de forma independiente para funcionar como entrada, salida o bidireccional, pudiendo configurarse, además, como tri-estado. Los IOBs pueden programarse para trabajar con diferentes niveles de tensión y cada uno de los IOBs incluyen flip-flops que pueden utilizarse para registrar tanto las entradas como las salidas. En la Fig. 5 se muestra la arquitectura típica de un IOB.

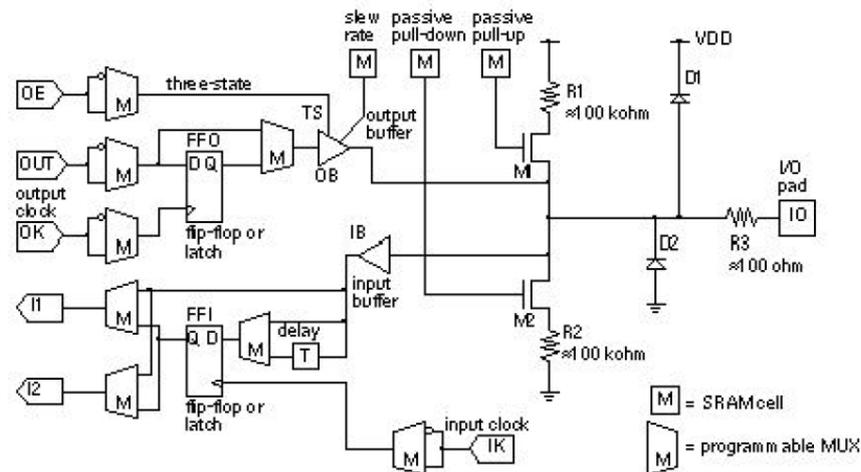


Fig. 5. Arquitectura típica de un IOB⁴

Líneas de interconexión: constituyen los caminos que permiten interconectar las entradas y salidas de los diferentes bloques. Están formadas por líneas metálicas de dos capas que recorren horizontal y verticalmente las filas y columnas existentes entre los CLBs. Dos elementos adicionales participan activamente en el proceso de conexión:

Puntos de Interconexión Programable (PIP, *Programmable Interconnection Point*): permiten la conexión de CLBs e IOBs a líneas metálicas cercanas.

Matrices de interconexión (SW, *Switch Matriz*): son dispositivos de conmutación distribuidos de forma uniforme por la FPGA y permite conectar señales de unas líneas a otras.

4.4 System on Chip (SoC)

Un *System on Chip* o SoC consiste en un dispositivo que integra todos los elementos funcionales de un producto final en un único chip. De esta manera, un diseño basado en SoC suele incorporar, procesadores, memoria, bloques DSP, interfaces con periféricos (GPIO, Ethernet etc.), software embebido, etc.

Considerando las tecnologías mencionadas en el apartado anterior 4.3, a la hora de diseñar un SoC, se debe tener en cuenta que:

³ Imagen obtenida de: <http://www.edafun.com/component/k2/item/77-whats-fpga?.html>

⁴ Imagen obtenida de: <http://iroi.seu.edu.cn/books/asics/Book2/CH06/CH06-21.gif>

- Si se eligen Microcontroladores o DSPs es preciso buscar un dispositivo que ya tenga todos los elementos requeridos en el sistema a diseñar, ya que su arquitectura hardware no es configurable. Actualmente la mayor parte de los microcontroladores podrían considerarse SoCs compuestos por procesadores y un gran número de periféricos. Existen microcontroladores de uso general, pero la mayor parte se fabrican para usos específicos. Éstos representan una muy buena opción si se consigue encontrar uno que se adecúe a las necesidades del sistema ya que su precio unitario es bajo y el coste del proceso de desarrollo es reducido. El principal inconveniente es la dificultad, en ciertos casos, a la hora de encontrar un dispositivo que cumpla todas las características requeridas.
- Un SoC basado en ASIC consiste en crear el dispositivo de silicio totalmente a medida, de acuerdo a las especificaciones del sistema basándose en IPs (Intellectual property), que consisten en bloques funcionales prediseñados, normalmente desarrollados por otras entidades. Técnicamente consiste en la mejor opción ya que presentan las mayores frecuencias de funcionamiento y menor consumo. Además permiten la integración de etapas digitales y analógicas (diseño mixto) reduciendo el número de componentes externos necesarios en un diseño. Sus grandes inconvenientes son: alto NRE (*Non-Recurring Engineering* el cual hace referencia al costo único de investigación, desarrollo, diseño y prueba de un nuevo producto), necesidad de desarrolladores expertos, así como el coste de los derechos intelectuales de los IP utilizados.
- Otra de las alternativas es la creación de un SoC basado en FPGA. El hardware viene preestablecido pero permiten el diseño del Soc mediante el uso de IPs al igual que en los ASIC. Esta alternativa constituye una solución intermedia entre los ASIC y los microcontroladores/DSPs ya que une lo mejor de ambos mundos, módulos hardware preestablecidos y flexibilidad para crear IPs a medida. Tienen un NRE bajo y una alta flexibilidad de diseño. Como desventaja presentan que su frecuencia de funcionamiento es mucho menor que la de los ASIC, así como que el consumo es mayor. Ésta ha sido la alternativa escogida para este diseño.

5 Arquitectura global del sistema propuesto

5.1 Diagrama de bloques

El sistema propuesto (Fig. 6) tiene como objetivo generar imágenes B-scan del entorno explorado, basadas en la emisión simultánea en todas direcciones de señales ultrasónicas codificadas para cada uno de los distintos sectores, A-scan, y la posterior decodificación de los ecos recibidos. Este proceso se divide en dos bloques principales, uno de emisión, donde se almacenan y emiten los códigos con su correspondiente retardo generado convenientemente para lograr la deflexión de cada uno de éstos en el ángulo deseado. Así como un bloque receptor, dividido a su vez en un módulo de procesado a bajo nivel, donde se realiza la adquisición y procesado en tiempo real, y otro de procesado de alto nivel, cuya funcionalidad es la de obtener las imágenes B-scan.

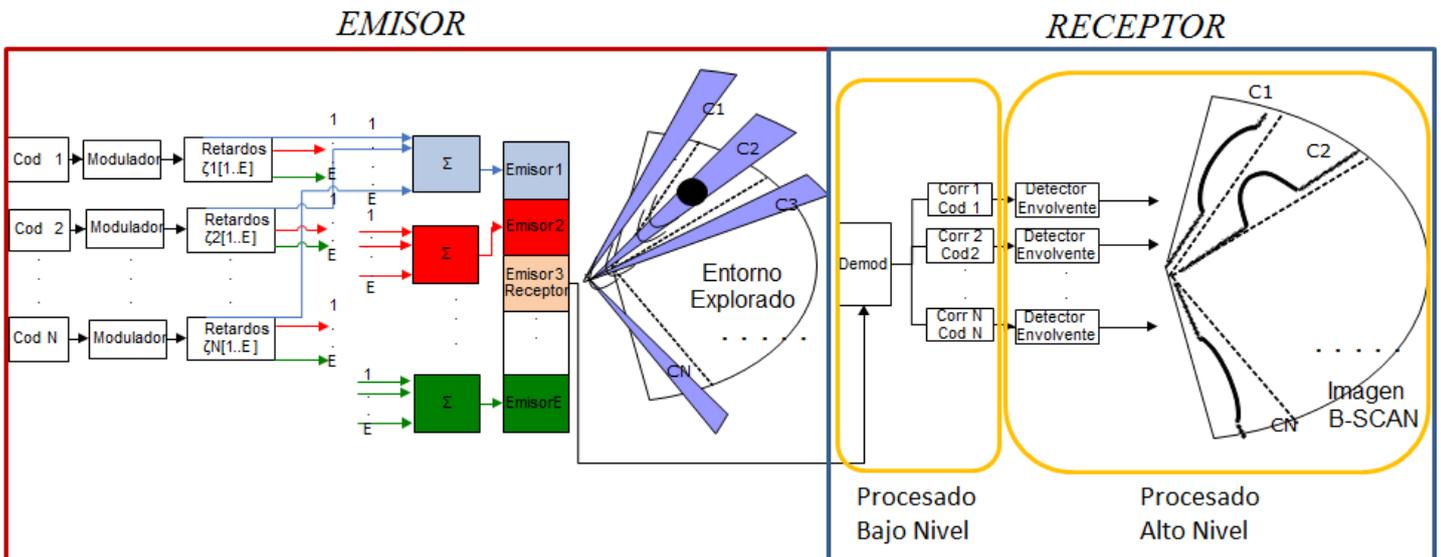


Fig. 6 Diagrama de bloques del sistema propuesto

La arquitectura que se plantea (Fig. 6) cuenta con un bloque emisor, encargado de almacenar y modular los códigos a transmitir, así como de generar los retardos propios de los distintos elementos del *array* necesarios para deflectar cada código en la dirección deseada. Los códigos se suman antes de atacar a cada uno de los transductores para poder emitir simultáneamente en todas las direcciones. Este apartado se ha implementado de forma digital sobre un dispositivo de hardware reconfigurable tipo FPGA (*Field Programmable Gate Array*). Este tipo de plataforma permite generar una lógica fácilmente adaptable tanto al número de códigos que se desean transmitir como a la cantidad de transductores utilizados.

Una vez se tienen los códigos modulados y sumados de forma digital, las señales resultantes correspondientes a cada uno de los distintos transductores son convertidas de digital a analógica mediante una serie de DACs [Dig06], externos al dispositivo FPGA y basados en comunicación serie SPI. Dichas señales analógicas son filtradas y pre-amplificadas con el fin de eliminar la componente continua y ajustar la ganancia para conseguir la potencia deseada en el *array* US.

Las señales pre-amplificadas son llevadas a unos amplificadores de potencia cuya ganancia no es ajustable. Finalmente la salida de los amplificadores de potencia es llevada a su correspondiente transductor (elemento del *array* ultrasónico).

La finalidad del bloque receptor (Fig. 6) consiste en demodular y procesar las señales recibidas procedentes de cada una de las direcciones exploradas (A-scan). Además es el encargado de realizar las tareas de post-procesado y generación de la imagen B-scan final. Como se puede apreciar en la Fig. 6 este proceso está formado por dos etapas:

1. En primer lugar se realiza la adquisición, filtrado y obtención de las líneas A-scan mediante la correlación con los códigos correspondientes a cada uno de los sectores explorados. Para ello se requiere de una etapa analógica formada por un micrófono, una electrónica de acondicionamiento (filtrado y amplificación) y un circuito de adquisición basado en un ADC [Dig11]. La señal adquirida es pre-procesada en una placa basada en lógica programable FPGA igual a la utilizada en emisión.
2. En segundo lugar se lleva a cabo un procesado a más alto nivel en un PC donde se obtienen las imágenes B-scan. Las líneas A-scan adquiridas serán enviadas al PC desde la FPGA mediante comunicación Ethernet.

5.2 Bloque de emisión

5.2.1 Estrategia de codificación

Para poder realizar emisiones simultáneas se ha recurrido a la técnica CDMA (*Code Division Multiple Access*) [Che06]. Esta técnica permite asignar a cada usuario de un mismo canal un código diferente e incorrelado al resto, permitiendo el acceso al medio por cada uno de esos códigos de forma independiente.

Para generar imágenes que mejoren la relación señal ruido, es necesario utilizar códigos que presenten un buen comportamiento de auto-correlación y correlación cruzada en la región del espacio a explorar. De entre las distintas alternativas [PUH09], aquellos códigos que presentan una ventana libre de interferencias IFW (*Interference Free Window*) alrededor del origen de sus funciones de correlación permiten reducir sobremanera las interferencias ISI y MAI. Así, ajustando el tamaño de dicha IFW a la región bajo análisis, se consiguen generar imágenes con un contraste mayor en comparación a las obtenidas con el empleo de códigos tradicionales. Además, la detección de muchos de estos códigos [PSU12] [ZY05] [CY06] puede realizarse de modo eficiente, empleando un menor número de operaciones que con un correlador convencional, y con arquitecturas que pueden ser fácilmente implementadas en hardware configurable [PSU12] [PUH11].

En este trabajo se han utilizado secuencias basadas en CSS (*Complementary Set of Sequences*) para componer códigos de mayor longitud, denotados como macro-secuencias con IFW en su función de correlación. Las secuencias CSS presentan propiedades ideales al realizar la suma de las funciones de auto-correlación (SACF) y la suma de la funciones de correlación-cruzada (SCCF).

Específicamente, se han utilizado macro-secuencias de CSS, construidas según el método descrito en [PSU12]. Los CSS [TL72] utilizados, están formados por N conjuntos incorrelados $\{S_n; 0 \leq n \leq N-1\}$, cada uno de ellos con N secuencias $\{s_{n,m}[l]; 0 \leq n, m \leq N-1; 0 \leq l \leq L-1\}$ de longitud $L = N^P$, siendo N un número potencia de 2 y P un número entero. Las secuencias $s_{n,m}$ de cada conjunto S_n se han concatenado con una separación de w_0 ceros, obteniendo N macro-secuencias $M_{S_n} \{M_{S_n} = [s_{n,0} \text{ wo } s_{n,1} \text{ wo } \dots \text{ wo } s_{n,N-1}]; 0 \leq n \leq N-1\}$ con una IFW en sus funciones de correlación de tamaño $2 \cdot w_0 + 1$.

En (7) se muestran las expresiones discretas de las funciones SACF y SCCF asociados a los CSS originales.

$$\begin{aligned} SACF &= \sum_{m=0}^{N-1} C_{s_{n,m}, s_{n,m}}[\tau] = N \cdot L \cdot \delta[\tau] \\ SCCF &= \sum_{m=0}^{N-1} C_{s_{n,m}, s_{n',m}}[\tau] = 0, n \neq n' \forall \tau(2) \end{aligned} \quad (7)$$

Donde $C_{x,y}$ es la correlación aperiódica entre x e y .

La Fig. 7 detalla gráficamente el proceso de formación de dichas macro-secuencias M_{S_n} .

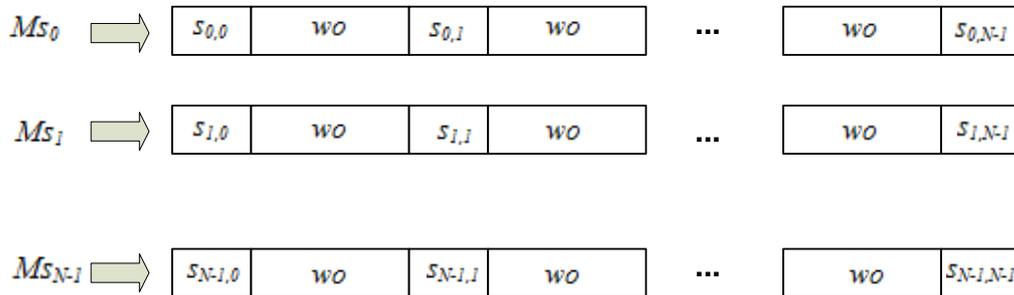


Fig. 7. Proceso de generación de las macro-secuencias M_{S_n} .

Las propiedades de auto-correlación y correlación-cruzada de estas secuencias se muestran en (8).

$$\begin{aligned} ACF &= C_{M_{S_n}, M_{S_n}}[\tau] = \begin{cases} N \cdot L & \tau \neq 0 \\ 0 & -w_0 \leq \tau \leq w_0, \tau \neq 0 \end{cases} \\ CCF &= C_{M_{S_n}, M_{S_n}'}[\tau] = 0, -w_0 \leq \tau \leq w_0 \end{aligned} \quad (8)$$

La ventaja de estos códigos frente a otros que también presentan una IFW, como por ejemplo los LS [ZYH05] o GPC [CY06], viene por el menor número de operaciones a realizar en su detección. Sin embargo, presentan una ganancia de proceso menor, debido a un mayor número

de ceros en el código. Utilizando CSS de longitud $L=N-1$, la longitud de las macro-secuencias Ms_n resultantes es la especificada en (9):

$$L_{Ms_n} = N^2 + (N - 1) \cdot w_o \quad (9)$$

Donde N es el número de secuencias $s_{n,m}$ de cada conjunto S_n ; y w_o es el número de ceros intercalado.

La ganancia de proceso, entendida como la relación entre el pico máximo de auto-correlación y la longitud del código, es (10):

$$Gp = \frac{N^2}{N^2 + (N - 1) \cdot w_o} \quad (10)$$

En la Fig. 8 se muestra un ejemplo de auto-correlación y correlación cruzada de estas macro-secuencias, cuando han sido obtenidas a partir de CSS con $N=L=4$, y $w_o=32$. Como se puede apreciar, aparece una zona libre de interferencias en torno al máximo de correlación, de tamaño $2 \cdot w_o + 1 = 65$ muestras.

Como ya se ha mencionado, es preciso escoger ventanas que se ajusten a las especificaciones de exploración para asegurar que los distintos ecos de los códigos emitidos se reciben dentro de la IFW. Reducir en exceso el tamaño de la ventana degrada la calidad de las imágenes generadas. No obstante, ventanas demasiado extensas acarrearán códigos muy largos que pueden incrementar en exceso la carga computacional del sistema.

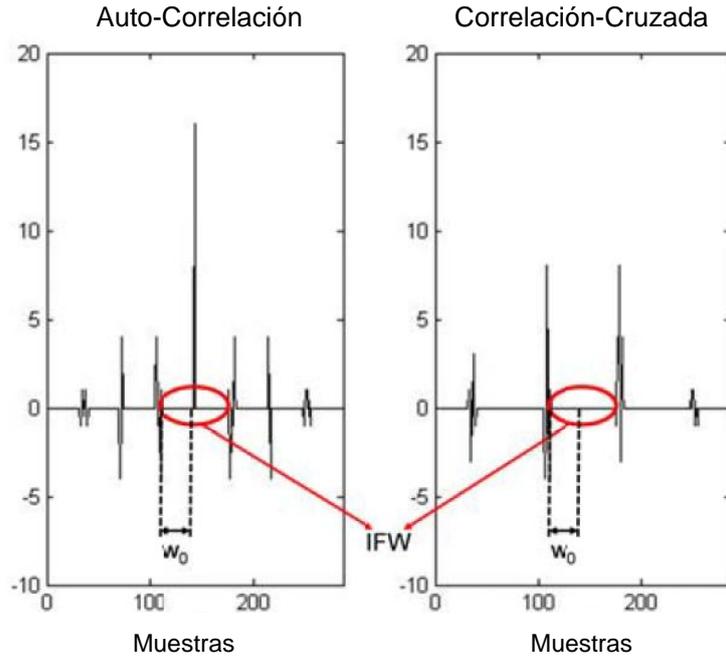


Fig. 8. Funciones de Auto-Correlación y Correlación-Cruzada de macro-secuencias Ms_n , obtenidas a partir de $N=4$ CSS de longitud $L=4$ y $w_0=32$.

5.2.2 Modulación BPSK

La frecuencia de emisión debe encontrarse en la frecuencia ultrasónica de interés. Por esta razón, se ha escogido una modulación de fase BPSK (*Binary Phase Shift Keying*). La principal ventaja de esta modulación digital con respecto a otra modulación basada en saltos de fase como QPSK (*Quadrature Phase Shift Keying*) consiste en que el receptor no necesita tener gran precisión para poder discernir entre símbolos. Además la implementación del modulador y el demodulador resulta relativamente sencilla.

La desventaja principal que presenta esta modulación es que, debido a que los saltos de fase son muy abruptos (π radianes entre distintos símbolos), se introducen componentes en alta frecuencia que provocan que se ocupe un mayor ancho de banda. No obstante, debido a que el sistema planteado no tiene unas limitaciones estrictas en cuanto al ancho de banda, esto no supone un problema en cuanto al correcto funcionamiento.

Esta modulación se consigue multiplicando cada bit del código $c[n]$ que se desea emitir por uno o más ciclos de portadora cuya fase varía entre 0 y π radianes dependiendo del valor del bit. El patrón de modulación $p[n]$ se obtiene según se indica en (11).

$$p[n] = \sum_{i=0}^{L-1} c[i] \cdot s[n - i \cdot O_f] \quad (11)$$

Donde $s[n]$ es el símbolo de modulación; L representa la longitud del código y O_f el número de muestras de este símbolo. El parámetro O_f se obtiene multiplicando el número de ciclos de portadora del símbolo de modulación por la razón entre la frecuencia de muestreo y la de portadora, f_s y f_c respectivamente.

El efecto que produce el introducir más ciclos de portadora por símbolo es el de reducir el ancho de banda, así como el aumento de la ganancia de proceso al realizar la correlación en recepción. Por contra, aparecen lóbulos laterales, que pueden afectar en la nitidez a la hora de formar imágenes. Además, el tiempo de procesado se incrementa en la misma proporción que el número de ciclos que forma el símbolo, lo que supone su principal inconveniente.

En este trabajo se ha utilizado una modulación BPSK con un símbolo de modulación formado por un único ciclo de portadora. Se ha escogido así para no incrementar en exceso la longitud de los códigos, teniendo en cuenta además que la ganancia de proceso resultante permite realizar imágenes con contraste suficiente. Por otro lado, la limitación del ancho de banda apenas afecta a nuestro sistema.

5.2.3 Características del array de ultrasonidos (US)

El sistema propuesto para *Phased Array*, permite deflecar el haz de forma electrónica sin necesidad de desplazar el transductor. Se ha diseñado para poder procesar, tanto en emisión como en recepción, las señales de un *array* ultrasónico desarrollado en un trabajo previo del grupo de investigación GEINTRA. Se detallan a continuación las características principales que intervienen en el diseño del *array*.

Este diseño está pensado para emitir hasta un total de $N=32$ secuencias codificadas a través de un *array* de $E=8$ elementos formados por material piezoeléctrico EMFi, combinando técnicas de *Phased Array* [RS83] y CDMA [Che07]. Esto permite realizar un escaneo simultáneo del entorno en el número de sectores deseado conforme a las dimensiones del *array* de ultrasonidos (US).

El material EMFi [PLK00] del que está formado el *array* presenta una respuesta en frecuencia prácticamente plana hasta 200 kHz. Teniendo en cuenta que las aplicaciones de ultrasonidos suelen estar entre 40 kHz y 100 kHz, el ancho de banda del material no supone un problema para el sistema, a pesar de la alta ocupación del canal que se produce al utilizar secuencias codificadas. Aparte de esta ventaja, el material presenta otras como la baja impedancia acústica, lo cual permite la no utilización de etapas de adaptación al contrario que con el empleo de otras tecnologías. Además, debido a su naturaleza polimérica, se presenta en láminas de espesor de μm . Se tiene una gran flexibilidad a la hora de la construcción del *array*, facilitando la modificación de los patrones emisión-recepción del transductor.

Los parámetros principales del *array* son el número de elementos N y la separación entre elementos d , *pitch*. Estos parámetros definen la frecuencia máxima de funcionamiento, así como las características en la conformación del haz. El aumento en el número de elementos del *array* provoca que se conforme un haz cada vez más estrecho. Es preciso tener en cuenta que la apertura del *array* D depende del número de elementos (12):

$$D = N \cdot w + (N - 1) \cdot k_f \quad (12)$$

Donde w es el ancho de cada elemento; y k_f es la separación entre elementos. Por otro lado, a mayor número de elementos, mayor distancia se necesita a para la formación del haz, es decir, el efecto en campo lejano se logrará a partir de distancias mayores, ya que (13):

$$NF = \frac{D^2}{(4 \cdot \lambda)} \quad (13)$$

Asimismo, la resolución angular $\Delta \theta$ viene dada por (14):

$$\Delta \theta = \frac{\lambda}{(D \cdot \cos(\theta_i))} \quad (14)$$

En Tabla 1, obtenida de [DHJ+11], se muestra la variación de la apertura del *array* D , la resolución angular $\Delta \theta$ y la distancia NF a la que se forma el haz en función del número de elementos, para una frecuencia de emisión $f_0 = 80 \text{ kHz}$.

Tabla 1 Relación entre D (apertura del *array*), $\Delta \theta$ (resolución angular) y NF (formación de campo cercano) en función de E (no de elementos del *array*) para $f_0 = 80 \text{ kHz}$.

E	D (cm)	$\Delta \theta$ (°)	NF (cm)
8	$4 \cdot \lambda = 1.6335$	14.3239	1.5393
16	$8 \cdot \lambda = 3.3670$	7.1620	6.5398
32	$16 \cdot \lambda = 6.8314$	3.5810	26.9418
64	$32 \cdot \lambda = 13.7680$	1.7905	109.3498

La generación de los retos necesarios para la deflexión del haz viene determinada por (15):

$$\Delta t_e = e \cdot \frac{d}{c} \cdot \sin(\theta) + t_0 \quad (15)$$

Donde c es la velocidad del sonido, θ es el ángulo donde se quiere deflectar el haz, Δt_e el retardo en cada uno de los elementos, e es cada uno de los elementos. El parámetro t_0 se introduce para evitar retardos negativos.

5.3 Bloque de Recepción

5.3.1 Demodulación BPSK

La señal recibida modulada en fase BPSK se demodula basándose en el esquema que aparece en la Fig. 9. En él se refleja una señal de referencia sinusoidal que corresponde con el símbolo de modulación, un multiplicador y un filtro paso bajo.

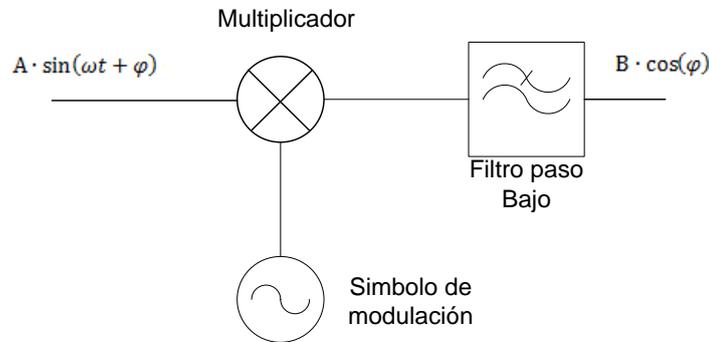


Fig. 9. Esquema de demodulación.

La señal obtenida a la salida del multiplicador se representa con la siguiente expresión (16):

$$A \cdot \sin(\omega t + \varphi) \cdot \sin(\omega t) = -\frac{A}{2} \cdot [\cos(2\omega t + \varphi) - \cos(\varphi)] \quad (16)$$

Donde A es la amplitud de la señal modulada, ω es la pulsación y φ es la fase de la señal modulada.

Es preciso eliminar el término que es función del doble de la frecuencia para quedarnos únicamente con el término dependiente de la fase, por lo que se introduce un filtro paso bajo. Para integrar ambas etapas en una única, y con el fin de buscar una fácil implementación sobre una plataforma digital, se aplica sobre la señal un filtro de correlación con el símbolo de modulación. De esta manera se obtiene (17):

$$\begin{aligned}
 S_{dem} &= \int_0^T A \cdot \text{sen}(\omega t + \varphi) \cdot \text{sen}(\omega t) dt \\
 &= A \cdot \left[\frac{1}{2} \cdot T \cdot \cos(\varphi) - \frac{\text{sen}(\varphi + 2\omega T)}{4\omega} \right] + \frac{A \cdot \text{sen}(\varphi)}{4\omega} \\
 &= \frac{A}{2} \cdot T \cdot \cos(\varphi) + \frac{A}{2\omega} \left[\cos\left(\frac{2\omega T + 2\varphi}{2}\right) \cdot \text{sen}(-\omega T) \right] \\
 &= \frac{A}{2} \cdot T \cdot \cos(\varphi) + \frac{A}{2\omega} \left[\cos\left(\frac{2\omega T + 2\varphi}{2}\right) \cdot 0 \right] = \frac{A}{2} \cdot T \cdot \cos(\varphi)
 \end{aligned} \tag{17}$$

Donde A es la amplitud de la señal modulada, ω es la pulsación, φ es la fase de la señal modulada y T es el periodo. Por simplicidad de diseño la demodulación de la señal se realiza de forma digital tras su adquisición. La expresión digitalizada de la ecuación (17) se muestra a continuación en (18):

$$Sd_{dem} = \sum_{k=0}^{O_f-1} A \cdot \text{sen}\left(\frac{k}{O_f} \cdot 2\pi + \theta\right) \cdot \text{sen}\left(\frac{k}{O_f} \cdot 2\pi\right) = \frac{A}{2} \cdot O_f \cdot \cos(\theta) \tag{18}$$

Donde A es la amplitud de la señal modulada y O_f es el número de muestras del periodo de la señal modulada. Puesto que el proceso de demodulación se realiza de forma aperiódica, la señal resultante se comportara de forma sinusoidal y el valor de fase de los símbolos adquiridos se obtiene cada O_f muestras.

En la siguiente gráfica se muestra una señal modulada con BPSK con un símbolo de modulación $f_c=1$ Hz, amplitud $A=1$ y $O_f=20$ muestras por símbolo, y dicha señal tras su demodulación.

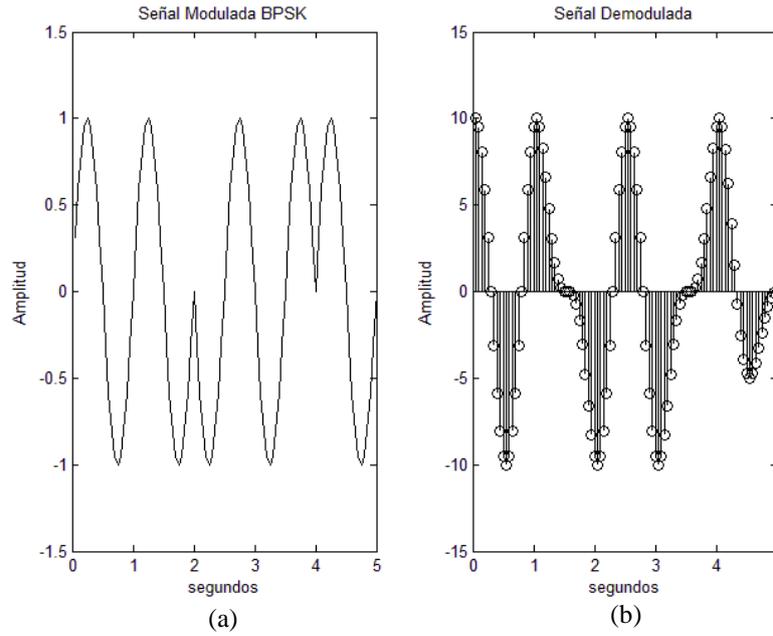


Fig. 10. (a) Señal modulada BPSK con amplitud 1, $f_c=1$ Hz y $f_c=20$ Hz. (b) Señal demodulada.

5.3.2 Algoritmo de correlación eficiente de CSS

Los CSS, y por tanto las macro-secuencias derivadas, pueden detectarse mediante correladores eficientes que reducen el número de operaciones frente a correladores directos tradicionales.

En este trabajo se ha implementado un algoritmo de correlación eficiente M-ESSC (*Efficient Set of Sequences Correlator*) desarrollado por [PSUJ12] el cual modifica la arquitectura propuesta en [PUH+10]. Este correlador permite obtener a su salida, de forma simultánea, las funciones de correlación aperiódica SCCF y SCAF de las N entradas $s_{n,n}$ con los n conjuntos CSS, S_n . De esta manera, si en cada una de las entradas se introducen todas las secuencias pertenecientes a un conjunto S_n , de forma simultánea, a la salida correspondiente al conjunto n , se obtendrá la función SCAF. Mientras que en el resto de salidas, correspondientes a los demás conjuntos CSS, se obtendrá la función SCCF.

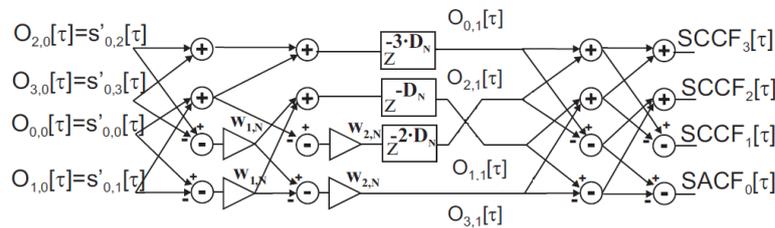


Fig. 11 Esquema del correlador N-ECSS usado en este trabajo [PSUJ12].

La principal ventaja de este algoritmo frente a una implementación directa tradicional consiste en la reducción del número de operaciones llevadas a cabo. La comparativa entre la versión clásica y el correlador eficiente empleado en este trabajo, se refleja en Tabla 2 y Tabla 3.

Tabla 2 Número de operaciones para la correlación simultánea de N CSS ortogonales con distintos métodos en función de N [PSUJ12].

Arquitectura	Sumas	Multiplicadores
Método clásico	$N^2 - N$	N^3
N ESSC utilizado	$(2 \cdot N) \cdot \log_2(N)$	0

Tabla 3 Requisitos de memoria para la correlación simultánea de OCSS con distintos métodos en función del número N de OCSS, la longitud del código $L=N^N$ y el tamaño de los datos de entrada DW .

Arquitectura	Memoria
Método clásico	$DW \cdot N^2 + N^3$
N ESSC utilizado	$\frac{N^2 - N}{2} \cdot (DW + \log_2(N))$

Asimismo, esta arquitectura, al ser modular, es altamente configurable, lo que permite implementar correladores en función de los parámetros de diseño DW (número de bits necesarios para representar los valores de entrada), N_{SN} (número de periodos por símbolo de modulación) y O_f .

5.3.3 Representación del entorno explorado (formación de imágenes B-Scan)

La obtención de imágenes generadas con ultrasonidos se basa en el fenómeno de reflexión de la onda acústica cuando se encuentra con obstáculos en su propagación. La onda es reflejada retornando hasta su fuente generadora, siempre y cuando el obstáculo se encuentre en una posición perpendicular en relación al haz incidente.

Como formas básicas de representación tenemos el A-scan, que contiene la información de profundidad de un determinado ángulo de deflexión, y el B-scan, que ofrece la información de un plano de corte y que se construye por la unión de sucesivas A-scan adquiridas a diferentes ángulos. Las exploraciones basadas en imágenes de ultrasonidos B-scan son frecuentemente usadas tanto en industria, para la realización de ensayos y exploraciones no destructivos, entre otros, o en medicina, para pruebas tales como las ecografías. En la siguiente figura se puede apreciar una imagen de ultrasonidos obtenida con un escáner médico (Fig. 12.a) y otra realizada para obtener mapeos de corrosión en tuberías (Fig. 12.b).

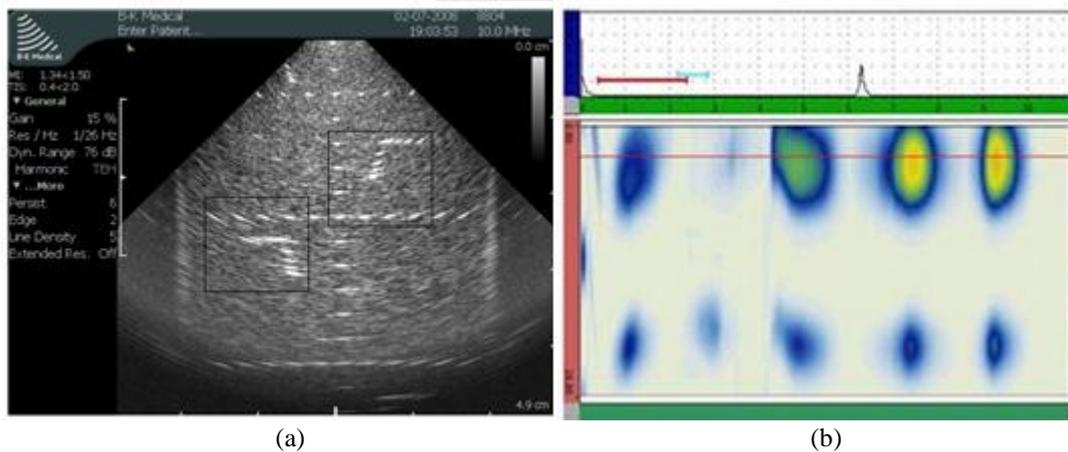


Fig. 12 (a) Imágen médica. (b) Imágen entorno industrial.

En este trabajo se han obtenido mapas del entorno basados en las líneas A-scan formadas con la energía de la función de correlación aperiódica de los ecos recibidos de cada uno de los códigos emitidos para los diferentes sectores explorados.

5.4 Especificaciones del sistema

De cara a particularizar la arquitectura e implementación del procesamiento para un *Phased Array* concreto [DHJ⁺11], se han determinado los siguientes parámetros generales:

- Barrido simultáneo de hasta $N=32$ sectores diferentes, desde -64° a 64° , empleando una codificación basada en 32 códigos de longitud $L_{Msn}=11998$ bits.
- Emisión a través de un *array* de ultrasonidos de $E=8$ elementos.
- Profundidad de exploración de $d_{mx}=1.5\text{m}$ a partir de la conformación del haz a $d_{mn}=0.30\text{m}$. Esto implica una semi-ventana libre de interferencias de tamaño $w_o=354$.
- Modulación BPSK con frecuencia de portadora $f_c=80$ kHz, para tratar la emisión según la respuesta en frecuencia del material EMFi empleado en la fabricación del *array* [14].

Por motivo de la alta carga computacional que conlleva la implementación del sistema, se ha optado por el desarrollo de dos bloques (emisor y receptor) en plataformas hardware diferenciadas.

6 Implementación del bloque emisor

El bloque de emisión está formado por dos etapas (Fig. 13):

- Procesamiento digital de la señal: que incluye el almacenamiento de los códigos, la modulación y por último la generación de los retardos de emisión. Todo ello se lleva a cabo sobre una plataforma de hardware reconfigurable FPGA.
- Procesado analógico de la señal: encargado de la conversión digital-analógica, así como de la adecuación de los niveles de tensión necesarios para la emisión a través del *array* ultrasónico.

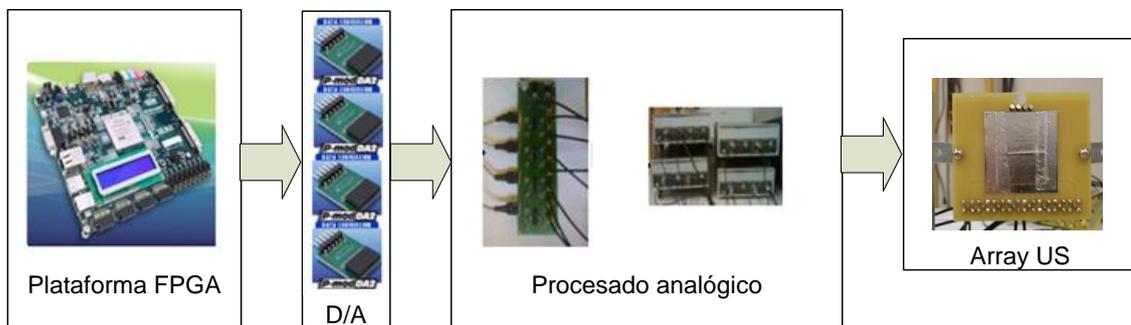


Fig. 13. Diagrama de conexión de elementos hardware del emisor.

6.1 Plataforma Hardware

En este apartado se detallan los elementos hardware que intervienen en el sistema para el bloque de emisión. Asimismo se especifica cuáles han sido los criterios de elección o de diseño del hardware necesario, tenidos en cuenta para el desarrollo de este bloque.

6.1.1 *Procesado Digital*

Este diseño requiere de un dispositivo que disponga de suficiente memoria interna como para poder almacenar todos los códigos que van a ser emitidos. Dado que van a emitirse 32 códigos de longitud 11996 elementos, y puesto que son necesarios al menos dos bits para codificar cada uno de ellos $M_{s_n} \in \{-1,0,1\}$. La memoria RAM requerida para cubrir las necesidades mínimas, debida al almacenamiento de los códigos, que el dispositivo debe tener es de (19):

$$Memoria = L_{M_{s_n}} \cdot 2bit \cdot N = 11996 \cdot 2 \cdot 32 = 768 Kbit \quad (19)$$

Asimismo, por motivo de la necesidad de generar los retardos con gran precisión, el procesamiento debe realizarse en tiempo real y a una frecuencia suficientemente elevada como para

no perder precisión al deflechar el haz. Este aspecto también se ve afectado por el conversor DAC empleado [Dig06].

Por otro lado, es necesaria una plataforma que permita flexibilidad, ya que se pretende realizar un diseño que realice varios procesos en paralelo y al mismo tiempo la adaptación, sin grandes complejidades, a diferentes *arrays* y diferentes códigos. Es decir, debe ser posible variar el número de elementos que conformen dicho *array*, modificar los códigos a emitir, aumentar o reducir el número de éstos, sin tener que modificar radicalmente el diseño.

Teniendo estos factores en cuenta, se ha optado por el empleo de una plataforma de hardware reconfigurable FPGA, que representa una opción intermedia, entre los DSPs y los ASICs. Tras contemplar los requisitos de memoria necesarios, así como la estimación de carga computacional que se va a requerir en este trabajo y la disponibilidad de material, se ha optado por una placa de evaluación, Genesys, desarrollada por Digilent Inc. basada en una FPGA Xilinx Virtex5 LX50T [Xil11] (Fig. 14).



Fig. 14. Plataforma de desarrollo Genesys Digilent Inc.

Para el bloque emisor no será necesario el uso de ninguno de los periféricos que vienen montados sobre la PCB de la placa de evaluación. Tan solo es necesaria la utilización de los pines de entrada salida disponibles, para comunicar la FPGA con los DACs.

Tabla 4 Recursos disponibles Virtex 5 FPGA Xilinx [Xil11].

Device	Configurable Logic Blocks (CLBs)			DSP48E Slices ⁽²⁾	Block RAM Blocks			CMTs ⁽⁴⁾	PowerPC Processor Blocks	Endpoint Blocks for PCI Express	Ethernet MACs ⁽⁵⁾	Max RocketIQ Transceivers ⁽⁶⁾		Total I/O Banks ⁽⁸⁾	Max User I/O ⁽⁷⁾
	Array (Row x Col)	Virtex-5 Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb ⁽³⁾	36 Kb	Max (Kb)					GTP	GTX		
XC5VLX30	80 x 30	4,800	320	32	64	32	1,152	2	N/A	N/A	N/A	N/A	N/A	13	400
XC5VLX50	120 x 30	7,200	480	48	96	48	1,728	6	N/A	N/A	N/A	N/A	N/A	17	560
XC5VLX85	120 x 54	12,960	840	48	192	96	3,456	6	N/A	N/A	N/A	N/A	N/A	17	560
XC5VLX110	160 x 54	17,280	1,120	64	256	128	4,608	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX155	160 x 76	24,320	1,640	128	384	192	6,912	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX220	160 x 108	34,560	2,280	128	384	192	6,912	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX330	240 x 108	51,840	3,420	192	576	288	10,368	6	N/A	N/A	N/A	N/A	N/A	33	1,200
XC5VLX20T	60 x 26	3,120	210	24	52	26	936	1	N/A	1	2	4	N/A	7	172
XC5VLX30T	80 x 30	4,800	320	32	72	36	1,296	2	N/A	1	4	8	N/A	12	360
XC5VLX50T	120 x 30	7,200	480	48	120	60	2,160	6	N/A	1	4	12	N/A	15	480
XC5VLX85T	120 x 54	12,960	840	48	216	108	3,888	6	N/A	1	4	12	N/A	15	480
XC5VLX110T	160 x 54	17,280	1,120	64	296	148	5,328	6	N/A	1	4	16	N/A	20	680
XC5VLX155T	160 x 76	24,320	1,640	128	424	212	7,632	6	N/A	1	4	16	N/A	20	680
XC5VLX220T	160 x 108	34,560	2,280	128	424	212	7,632	6	N/A	1	4	16	N/A	20	680
XC5VLX330T	240 x 108	51,840	3,420	192	648	324	11,664	6	N/A	1	4	24	N/A	27	960
XC5VSX35T	80 x 34	5,440	520	192	168	84	3,024	2	N/A	1	4	8	N/A	12	360
XC5VSX50T	120 x 34	8,160	780	288	264	132	4,752	6	N/A	1	4	12	N/A	15	480
XC5VSX95T	160 x 46	14,720	1,520	640	488	244	8,784	6	N/A	1	4	16	N/A	19	640
XC5VSX240T	240 x 78	37,440	4,200	1,056	1,032	516	18,576	6	N/A	1	4	24	N/A	27	960
XC5VTX150T	200 x 58	23,200	1,500	80	456	228	8,208	6	N/A	1	4	N/A	40	20	680
XC5VTX240T	240 x 78	37,440	2,400	96	648	324	11,664	6	N/A	1	4	N/A	48	20	680
XC5VFX30T	80 x 38	5,120	380	64	136	68	2,448	2	1	1	4	N/A	8	12	360
XC5VFX70T	160 x 38	11,200	820	128	296	148	5,328	6	1	3	4	N/A	16	19	640
XC5VFX100T	160 x 56	16,000	1,240	256	456	228	8,208	6	2	3	4	N/A	16	20	680
XC5VFX130T	200 x 56	20,480	1,580	320	596	298	10,728	6	2	3	6	N/A	20	24	840
XC5VFX200T	240 x 68	30,720	2,280	384	912	456	16,416	6	2	4	8	N/A	24	27	960

En Tabla 4 se pueden observar las principales características del modelo XC5VLX50T. Los dispositivos de la familia Virtex pertenecen a la gama más alta de los fabricados por Xilinx y presentan características tecnológicas mejores en comparación con el resto de familias de Xilinx. De esta manera, esta FPGA es capaz de funcionar con un reloj de 100 MHz y llegar a frecuencias de funcionamiento interno de hasta 200 MHz, permitiendo realizar procesados complejos en tiempo real.

Los parámetros a destacar de la Tabla 4, atendiendo a las especificaciones del diseño en recepción, son: número de bloques RAM, memoria distribuida, número de *slices* y número de DCMs. El resto de elementos no son tan relevantes ya que no van a ser necesarios en el desarrollo. Es importante tratar de utilizar, en la medida de lo posible, sólo bloques RAM, ya que el empleo masivo de memoria distribuida puede provocar el agotamiento de los recursos de interconexión.

Aunque hoy en día casi la totalidad de los dispositivos tienen integrados más de un DCM, es importante tener en cuenta el número de estos elementos de los que se dispone, ya que en este diseño va a ser necesario el uso de dos señales de reloj internas independientes. Por último, conviene mencionar que el número de *slices* es otro de los parámetros más importantes, debido a que indica cuánta lógica se va a poder implementar.

Una gran ventaja que presenta la serie 5 con respecto a su predecesora es el cambio en cuanto a la arquitectura de las pastillas de silicio. Cada *slice* contiene el doble de registros y de L.U.T. (*Look-up Tables*) que los dispositivos de series anteriores. Esto permite una mayor eficiencia en el aprovechamiento del espacio. Así como una mayor capacidad de cómputo por cada elemento, simplificando el conexionado entre *slices*.

6.1.2 Procesado Analógico

Para pasar las señales procesadas digitalmente a los estímulos eléctricos que el *array* de ultrasonidos necesita para su funcionamiento, es necesaria una serie de etapas de electrónica analógica que adecúan la señal a la requerida. Esto se traduce en:

- Conversión D/A.
- Filtrado y pre-amplificación.
- Amplificación.

Los elementos utilizados en cada una de estas etapas se describen en este apartado.

6.1.2.1 Digital to Analog Converter (DAC)

Digilent proporciona una serie de dispositivos analógicos fáciles de conectar con sus placas de desarrollo, a los que llama P-MODs. En este trabajo se ha utilizado uno de estos módulos PmodDA2 véase la Fig. 15, en el que se hallan dos DAC121S101 [TX13] SPI de 12 bits de resolución y salida *rail to rail* [Dig06]. Los principales parámetros que se han tenido en cuenta de este dispositivo son:

- Frecuencia máxima de funcionamiento: $SCLK \leq 30$ MHz.
- Número de bits por trama de datos: 16 bits: 12 bits de datos y 4 bits de control.
- Span: 0 a 4096.
- Alimentación de 2.7 V a 5.5V.
- No linealidad integral: ± 2.6 LSB.
- No linealidad diferencial: $+0.25, -0.15$ LSB.
- Slew Rate: 1 V/ μ s.
- Tiempo de establecimiento: 8 μ s aprox.

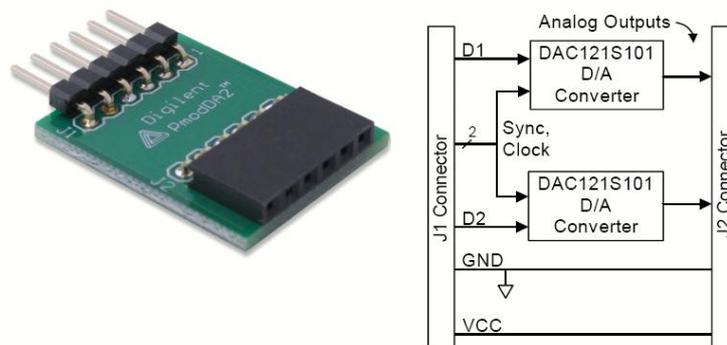


Fig. 15. Módulo de conversión digital analógico Digilent PmodDA2 [Dig06].

Los parámetros más restrictivos para este diseño son la frecuencia máxima del bus SPI, el *Slew Rate* y el tiempo de establecimiento. Para el cálculo del *Slew Rate* se ha considerado que la salida del dispositivo es una senoide, ya que la modulación que se ha utilizado es BPSK (20):

$$V_{out} = \frac{span}{2} \cdot \sin(2 \cdot \pi \cdot f \cdot t) \quad (20)$$

Donde f es la frecuencia de la senoide y $span$ es el valor máximo de tensiones que puede dar el DAC121S101 a su salida. Para obtener el valor máximo de variación se realiza la derivada de V_{out} (21):

$$\left. \frac{dV_{out}}{dt} \right|_{max} = \frac{SPAN}{2} 2 \cdot \pi \cdot f = \frac{3.3V \cdot \pi \cdot 80 \text{ kHz}}{10^6} = 0.83 \text{ V}/\mu\text{s} \quad (21)$$

Se puede observar como la variación máxima roza los límites del dispositivo. No obstante, en el peor caso, al emitir los códigos de forma simultánea, es decir, con todos ellos atacando a los transductores tras sumarse, implica que la variación máxima será difícil de determinar; y en ciertos momentos podrá superar la del *Slew Rate* máximo del dispositivo.

Por otro lado, el tiempo de establecimiento introducirá una distorsión mucho mayor, ya que en este diseño se realiza una conversión cada $0.625\mu\text{s}$ y el tiempo de establecimiento es 12.8 veces superior. El efecto que introduce puede verse en la Fig. 16, que representa una emisión real de una senoide con frecuencia de 80 kHz muestreada a 1.6 MHz capturada con el osciloscopio del laboratorio.

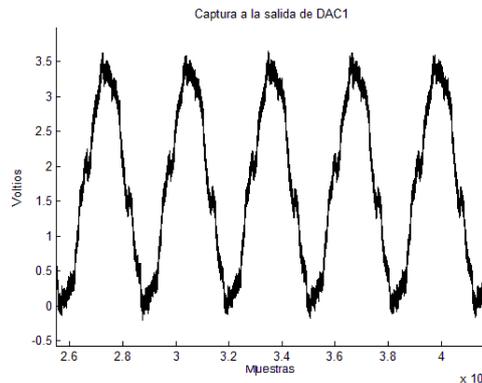


Fig. 16 Señal sinusoidal a 80 kHz obtenida a la salida del DAC121S101 con sobremuestreo de 20.

Aunque la distorsión que introduce resulta evidente, la señal sigue siendo morfológicamente similar a una senoide. Esto se debe a que el ruido que se introduce es de alta frecuencia y no afecta de forma apreciable en la obtención de las líneas B-scan, como se mostrará más adelante.

6.1.2.2 Filtrado y Pre-amplificación

Una vez realizada la conversión D/A, se obtiene una señal que varía entre $0 \text{ V} \leq V_{\text{DAC}} \leq 3.3 \text{ V}$. Esta señal debe tratarse con el fin de eliminar la componente continua y adecuar los valores de tensión para la etapa de amplificación de potencia. El material piezoeléctrico que conforma el *array* de ultrasonidos, necesita ser excitado con tensiones bipolares que varían entre $V_{\text{array}} = 75 \text{ Vp}$ y $V_{\text{array}} = 150 \text{ Vp}$. Dado que la ganancia de los amplificadores de potencia $G_{\text{pot}} = 15$, la ganancia de pre-amplificación G_{pa} debe ajustarse para una excitación del *array* $V_{\text{array}} = 75 \text{ Vp}$ (22) :

$$G_{\text{pa}} = \frac{V_{\text{array}} \cdot 2}{G_{\text{pot}} \cdot \text{span}} = 3 \quad (22)$$

Donde V_{array} es la tensión de entrada máxima que se desea en el *array* y G_{pot} es la ganancia de los amplificadores de potencia. Esta ganancia resulta sencilla de conseguir empleando componentes activos.

Por otro lado, se ha diseñado un filtro paso alto, cuyas principales especificaciones son:

- Filtro paso alto.
- Frecuencia de corte $f_c = 100 \text{ Hz}$.
- Ganancia de 0 dB.
- Oscilación de ganancia permitida en la banda de paso de 1 dB.
- Menor orden posible.

Para el diseño de esta etapa es necesario el uso de dos amplificadores operacionales A.O. por canal. A la hora de elegir el A.O. se han tenido en cuenta el *Slew Rate*, ancho de banda y coste, por lo que se ha optado por el circuito integrado TL084 [Tex99], compuesto por cuatro operacionales cuyo *Slew Rate* es $13 \text{ V}/\mu\text{s}$ y ancho de banda (BW) de 4 MHz. El principal inconveniente que presenta este dispositivo es que, al no tener una respuesta *rail to rail*, se debe dejar un margen de 1.5 V entre V_{cc} y la V_{out} máxima y para así asegurar que no se produce saturación. La salida de este circuito se lleva a los amplificadores de potencia a través de cables coaxiales compatibles con la entrada de éstos.

Para el diseño del filtro se ha utilizado la herramienta de uso gratuito *FilterPro* de *Texas Instruments* [TX11], que permite el diseño de filtros activos ajustando los valores de los componentes pasivos a los estándares normalizados. En Fig. 17 y Fig. 18 se pueden apreciar las características del filtro diseñado:

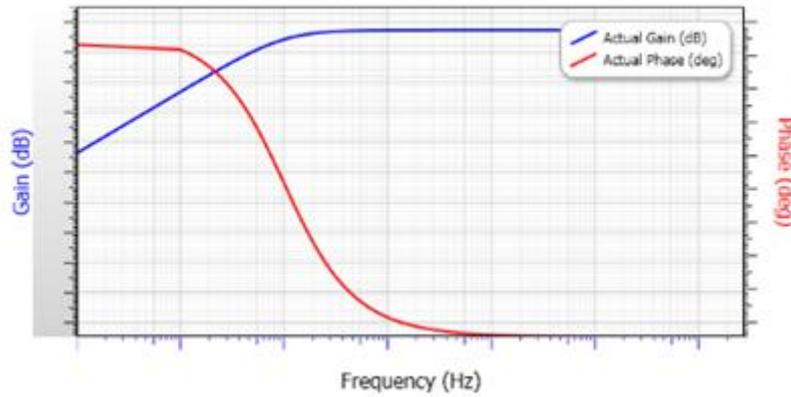


Fig. 17. Ganancia y Fase del filtro.

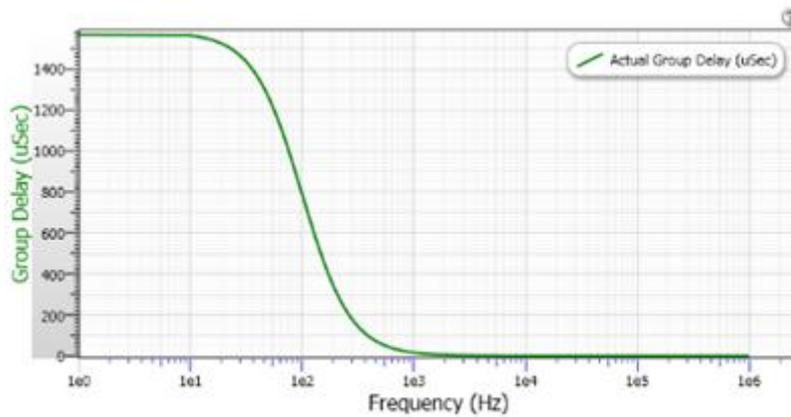


Fig. 18. Retardo de grupo del filtro.

Por otro lado, para el diseño PCB se ha empleado otra herramienta de distribución gratuita *Eagle* [Cad14], con la que se ha desarrollado esquemático y el ruteado de la PCB. En Fig. 19.a se puede ver el esquemático de las conexiones entre la etapa de filtrado y pre-amplificación y en Fig. 19.b el ruteado PCB.

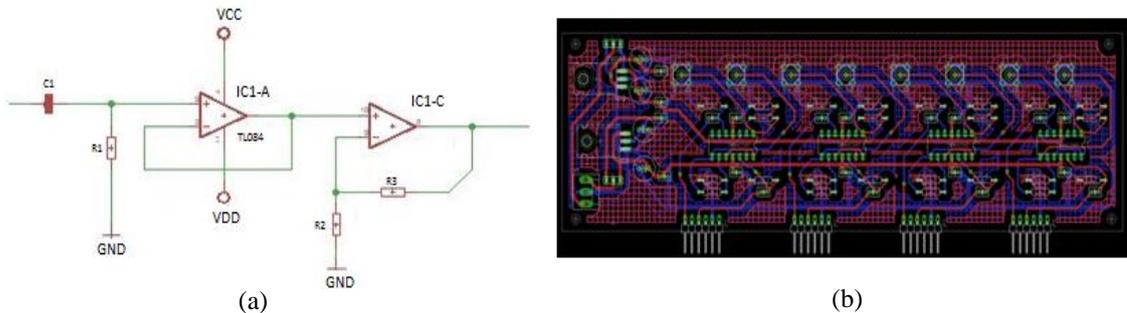


Fig. 19. (a) Esquema de filtro y amplificador. (b) Ruteado PCB.

Por último, en la Fig. 20, se puede observar el resultado.

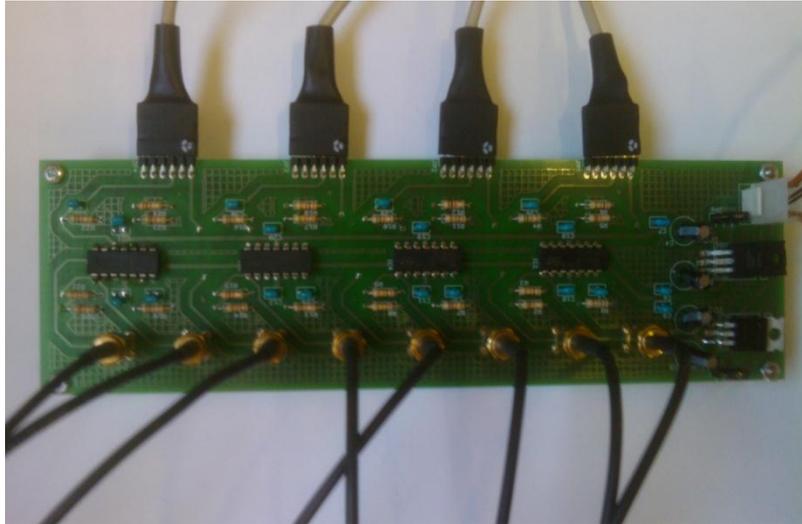


Fig. 20. Resultado PCB real.

6.1.2.3 Amplificador de potencia

Para conseguir emitir a través del *array* US es necesario aplicar tensiones relativamente altas, para lo que es necesario utilizar una etapa de amplificación de potencia. Para ello se han utilizado 4 amplificadores doble canal de la marca *Tabor ElectronicsLtd* modelo 9200, capaces de conseguir a la salida hasta 300 Vpp, y con un ancho de banda de 500 kHz para tensiones entre 150 Vpp y 300 Vpp.

6.2 Implementación hardware

6.2.1 Esquema

En la siguiente figura se muestra el diagrama de los bloques funcionales implementados sobre la plataforma hardware FPGA.

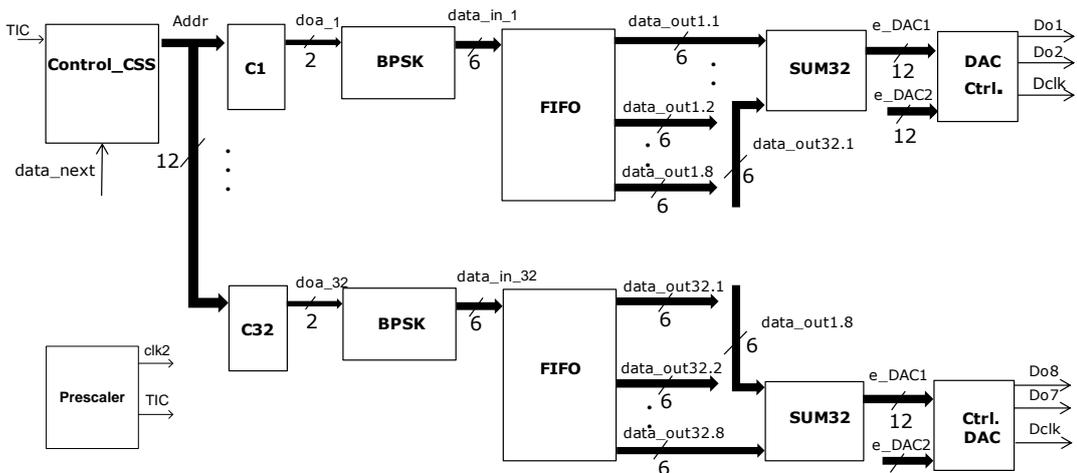


Fig. 21. Bloques HD implementados sobre plataforma hardware.

6.2.2 Bloques Lógicos

En este apartado se describe cada uno de los módulos que forman la lógica de emisión de acuerdo a su funcionamiento e interconexión.

6.2.2.1 Módulo Control_CSS

En la Tabla 5 se muestran las señales de entrada y salida del módulo Control_CSS.

Tabla 5 Puertos de entrada y salida del módulo Control_CSS.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
clk2	In	1	Clock-enable del sistema
rst	In	1	Reset del sistema
TIC	In	1	Señal de disparo de cada emisión
next_data	In	1	Señal de incremento del contador de dirección
en	Out	1	Habilita la modulación
addr	Out	14	Bus de direcciones de las memorias que contienen los códigos a emitir

Este bloque tiene como finalidad controlar el acceso a las posiciones de memoria donde se encuentran los elementos del código que deben ser enviados en cada instante. La señal *TIC* procedente del bloque **Prescaler** determina el comienzo de una nueva emisión, en este caso se produce un pulso cada segundo. Asimismo, el bus *addr* va seleccionando la posición de memoria donde se encuentran los elementos a emitir en cada uno de los bloques de memoria BRAM, para cada instante de tiempo. La lectura se realiza de forma simultánea para todos los códigos, sincronizada con la señal *next_data*. Esta última es generada en el módulo **BPSK** (siempre que la señal *en* esté habilitada) y consiste en un pulso activo cada vez que se finaliza la modulación de un bit, este pulso tendrá la frecuencia de portadora f_c y depende de la frecuencia de *clk2* (frecuencia de moduladora $O_f \cdot f_c$).

En Fig. 22 se muestra el diagrama de bloques del módulo Control_CSS.

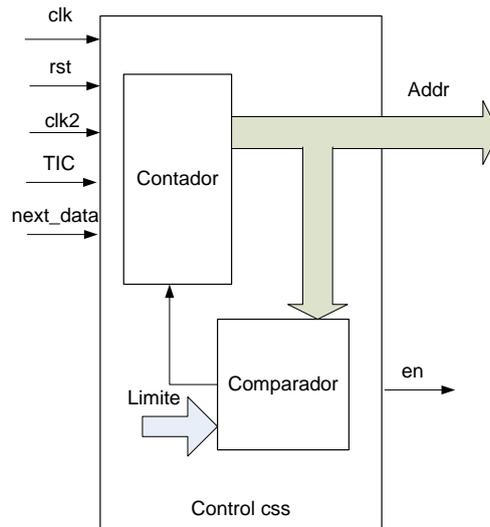


Fig. 22. Diagrama de bloques del módulo Control_CSS.

6.2.2.2 Módulo Prescaler

En la Tabla 6 se muestran las señales de entrada y salida del módulo Prescaler.

Tabla 6 Puertos de entrada y salida del módulo Prescaler.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
clk2	Out	1	Clock-enable del sistema
rst	In	1	Reset del sistema
TIC	Out	1	Señal de disparo de cada emisión

La función de este bloque es la de generar las señales de sincronización internas que son requeridas por el diseño. En primer lugar genera una señal de clock-enable *clk2* a 1,6 MHz, que determina la frecuencia del flujo de datos y viene delimitada por el periodo mínimo de conversión del conversor DAC. Asimismo, genera la señal con la que se dispara cada emisión *TIC* a 1 Hz. En Fig. 23 se muestra el diagrama de bloques del módulo Prescaler.

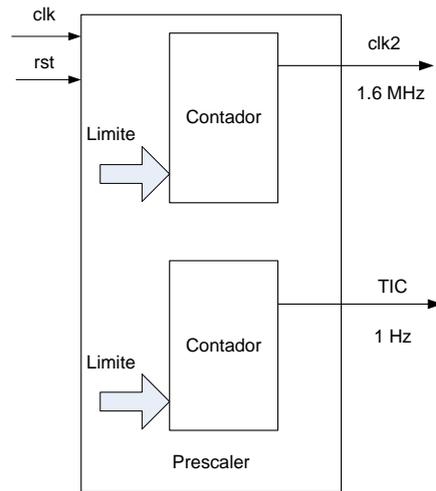


Fig. 23. Diagrama de bloques del módulo Prescaler.

6.2.2.3 Módulo BPSK

En la Tabla 7 se muestran las señales de entrada y salida del módulo BPSK.

Tabla 7 Puertos de entrada y salida del módulo BPSK.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
clk2	In	1	Clockenable del sistema
rst	In	1	Reset del sistema
doa	In	2	Dato binario del código a emitir
en	In	1	Habilita la modulación
next_data	Out	1	Señal de finalización de modulación
douta	Out	9	⁵ Dato binario modulado en BPSK

Este bloque se encuentra conectado a la salida de cada una de las memorias que contienen los códigos a emitir. Su función es la de modular dichos códigos. Está formado por una memoria ROM, que contiene el símbolo de modulación (un seno, cuyo número de elementos depende del sobremuestreo O_f . En este caso 20 muestras), así como una lógica que determina, dependiendo del elemento del código que se lee en cada instante, en qué sentido ha de leerse el símbolo. Específicamente, un 1 determina que se encuentra en fase (lectura de dirección inferior a superior); -1 en contra fase (lectura de dirección superior a inferior); y 0 valor nulo (se lee O_f veces la posición 0, igual al valor medio).

⁵ Variable en función del número total de códigos a emitir.

La amplitud máxima del símbolo de modulación viene limitado por dos parámetros. En primer lugar, el *span* del DAC y por otro lado el número de códigos que se vayan a emitir simultáneamente. Este diseño está pensado para emitir hasta 32 códigos. Por lo tanto, la amplitud máxima A_{max} será (23):

$$A_{max} = \frac{span}{2 \cdot N} = \frac{2048}{2 \cdot 32} = 32 \quad (23)$$

Donde *span* es el valor máximo de tensión de salida del DAC121S101; y *N* es el número de códigos que van a emitirse simultáneamente.

Desde este módulo se sincroniza la solicitud de nuevos bits mediante la señal *next_data*. Esta señal se habilita cuando el bloque **Control_CSS** activa la señal *en*, de esta manera el incremento en la dirección de la memoria donde se encuentran los códigos viene dado por el propio contador del modulador. En Fig. 24 se muestra el diagrama de bloques del módulo BPSK.

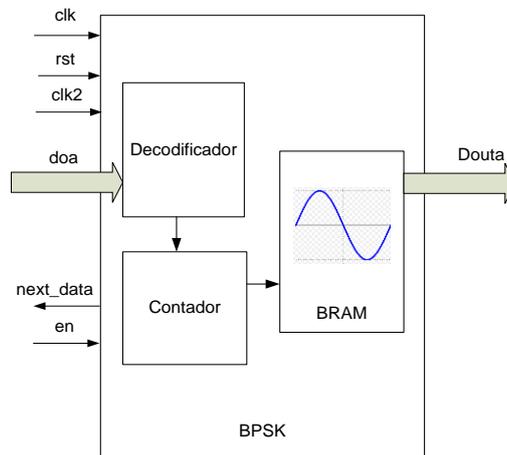


Fig. 24. Diagrama de bloques del módulo BPSK.

6.2.2.4 Módulo FIFO

En la Tabla 8 se muestran las señales de entrada y salida del módulo FIFO.

Tabla 8 Puertos de entrada y salida del módulo FIFO.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
clk2	In	1	Clock-enable del sistema
rst	In	1	Reset del sistema
data_in	In	9	Dato binario del código a emitir ⁶
limite_fifo	In	7 x 8	Vector de señales de configuración de número de retardos
data_out	Out	9 x 8	Vector de señales de salida retardadas ⁶
douta	Out	9	Dato binario modulado BPSK ⁷

La función que desempeña este elemento es la de aplicar los retardos asociados a cada código y sector del espacio, necesarios para deflectar el haz ultrasónico. Se trata de un registro de desplazamiento implementado en lógica, ya que los recursos de memoria interna de la FPGA se encuentran muy limitados debido al almacenamiento de los códigos. El número de registros que se utilizará en cada caso dependerá del número de retardos. En Fig. 25 se muestra el diagrama de bloques del módulo FIFO.

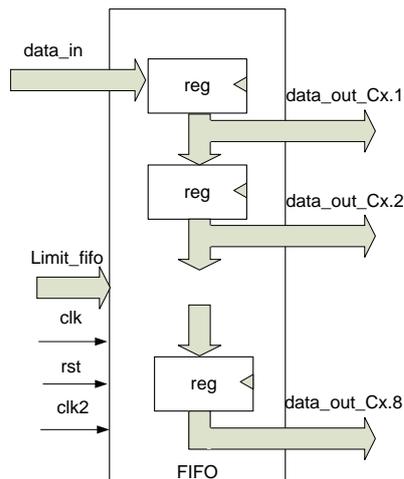


Fig. 25. Diagrama de bloques del módulo FIFO.

⁶ Función de la amplitud del símbolo de modulación (*douta* en bloque **BPSK**).

⁷ Variable en función del número total de códigos a emitir.

6.2.2.5 Módulo SUM32

En la Tabla 9 se muestran las señales de entrada y salida del módulo SUM32.

Tabla 9 Puertos de entrada y salida del módulo SUM32.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
clk2	In	1	Clock-enable del sistema
rst	In	1	Reset del sistema
data_in	In	9 x 32	Vector de Señales moduladas a sumar ⁸
data_out	Out	12	Señal de salida al controlador de DAC
start	Out	1	Señal de disparo de conversión

Este bloque suma los elementos de cada código, después de aplicar el retardo correspondiente, para cada uno de los transductores. Es decir, se suma cada elemento de los 32 códigos y se manda al controlador de DAC correspondiente. Además de realizar la suma de señales, este bloque genera una señal de sincronismo *start* que se utiliza de disparo para la conversión D/A en el bloque DAC ctrl. En Fig. 26 se muestra el diagrama de bloques del módulo SUM32.

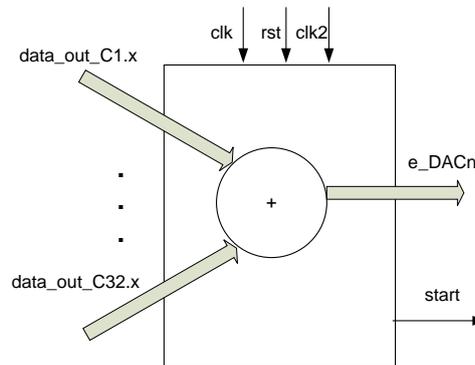


Fig. 26 Diagrama del módulo SUM32.

6.2.2.6 Módulo DAC ctrl.

En la Tabla 10 se muestran las señales de entrada y salida del módulo DAC_ctrl.

⁸ Función de la amplitud del símbolo de modulación (*douta* en bloque **BPSK**) y del número de códigos a emitir.

Tabla 10 Puertos de entrada y salida del módulo DAC_ctrl.

Señal	Tipo	Tamaño (bits)	Descripción
CLK	In	1	Reloj del controlador
RST	In	1	Reset del sistema
DATA1	In	12	Datos de entrada para DAC1
DATA2	In	12	Datos de entrada para DAC2
START	In	1	Señal de disparo de conversión
D1	Out	1	Datos de salida para DAC1
D2	Out	1	Datos de salida para DAC2
CLK_OUT	Out	1	Salida de reloj utilizada por el chip DAC121S01
nSYNC	Out	1	Señal de salida enviada al DAC para almacenar los datos enviados
DONE	Out	1	Señal de salida que indica el final de conversión.

La función de este módulo es la comunicarse con el DAC [Dig06], a través de un bus SPI (Fig. 27).

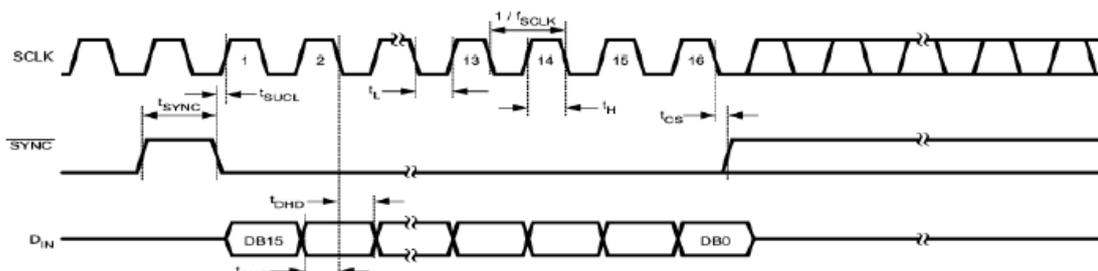


Fig. 27. Cronograma de comunicación SPI entre el DAC y el dispositivo de control [TX13].

La frecuencia máxima de funcionamiento es de 30 MHz. Cada conversión se debe realizar a una frecuencia de 1.6 MHz y es preciso enviar 16 bits por dato, más el t_{sync} (Fig. 27) cuyo valor mínimo es de 20 ns. Es decir, la frecuencia del bus SPI debe ser como mínimo $17 \cdot 1.6 = 27.2$ MHz. El modelo VHDL de este componente se basa en el que proporciona Digilent en su página web [Dig14]. El modo de funcionamiento es el siguiente (véase la Fig. 28):

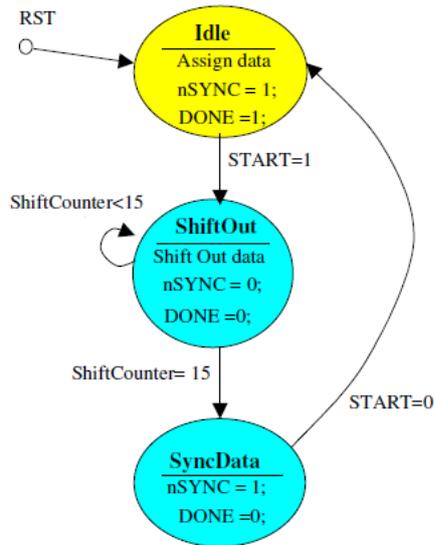


Fig. 28. Máquina de estados de funcionamiento del módulo DAC_ctrl [Dig06].

La dinámica de funcionamiento se basa en una máquina de estados como la que se puede ver en la Fig. 28 basada en tres estados: *idle*, *shiftout* y *SyncData*. Durante el estado *idle*, el vector de datos de 12 bits se actualiza junto con el registro de control de 4 bits. Para realizar una conversión es preciso que la señal *DONE* se halle a nivel alto. Cuando la señal de entrada *START* se activa durante el estado de *idle*, se pasa al estado *shiftout*. En dicho estado, los bits de datos se envían por el bus del MSB al LSB usando un contador que cuenta 16 ciclos de reloj. Cuando el contador llega a 15, la máquina de estado pasa al estado *SyncData*. En este estado, la señal *nSYNC* es llevada a nivel alto, con el fin de que los datos queden almacenados en el DAC. Si la señal de entrada *START* se encuentra a nivel bajo, se vuelve al estado *idle*.

No es posible hallar una frecuencia del bus SPI que sea múltiplo entero de 1.6 MHz y que a su vez se encuentra en el rango de 28 MHz y 30 MHz, utilizando un único reloj en el sistema. Por tanto, es necesario utilizar un reloj a diferente frecuencia para este bloque. La frecuencia de salida del DCM_DAC se ha escogido a 60 MHz, pues el bloque divide entre dos la frecuencia para obtener el reloj del bus. En otras palabras, la comunicación se realizará a 30 MHz.

Tener dos relojes dentro del mismo diseño produce un problema de sincronismo en el sistema. Para poder solventar este inconveniente se genera una señal de *start*, activada a la vez que *clk2* y cuya duración es de 4 ciclos de reloj (a 64 MHz). Este pulso permite asegurar la detección de dicha señal de disparo con una incertidumbre máxima de dos ciclos de reloj, es decir, 1 bit. Esto se traduce en que la frecuencia de conversión oscila 33.3 ns. Este error es 18.75 veces inferior a la resolución con la que se producen los retardos para lograr la deflexión, por lo que se ha considerado despreciable. Por otro lado, esta pequeña pérdida de sincronismo afecta a la señal *nSYNC*, la cual varía el tiempo que se halla a nivel alto entre 2 y 3 ciclos de reloj. No obstante, el tiempo mínimo que dicha señal se encuentra activa es de dos bits. Este tiempo es mayor que el mínimo dado por el fabricante [Dig06], por lo que la comunicación no se ve afectada.

En Fig. 29 se muestra el diagrama de bloques del módulo DAC_ctrl.

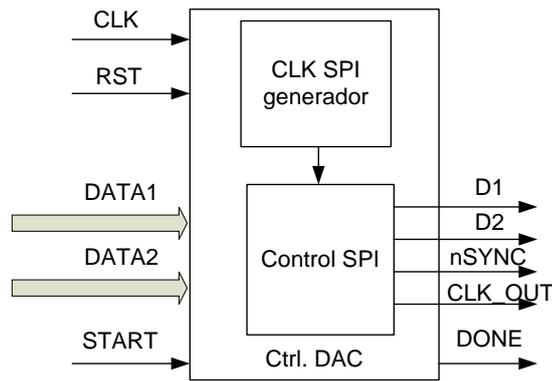


Fig. 29. Diagrama de bloques del módulo DAC_ctrl.

6.3 Simulaciones

Para comprobar el correcto funcionamiento del diseño se han realizado las siguientes simulaciones funcionales. En los siguientes cronogramas (Fig. 30 y Fig. 31) se muestra el proceso de comunicación entre el módulo emisor y el DAC. En primer lugar se tiene la señal de reloj del bus SPI a 30 MHz *clk_out*. Posteriormente se muestra, con valor numérico, cada una de las muestras antes de enviarse a través del bus, así como las señales de sincronismo y datos del bus SPI al primer DAC, *data_sum1*, *nsync,d1*, respectivamente. Por último se muestra, con valor numérico, el dato a la salida del modelo de simulación del primer DAC, que realiza la conversión de los datos serie del bus SPI, a paralelo.

En esta simulación (Fig. 30) se observa el proceso de comunicación entre el controlador del DAC y su modelo de simulación. Se puede apreciar que la conversión se realiza cada 633 ns o a 625 ns. No obstante, de cada cuatro conversiones una se realiza a 600 ns, lo que compensa el desfase producido por la falta de sincronismo entre el bloque de control del DAC y el resto del sistema. De esta manera se concluye que la emisión se realiza de forma periódica con un error máximo de conversión de 33 ns.

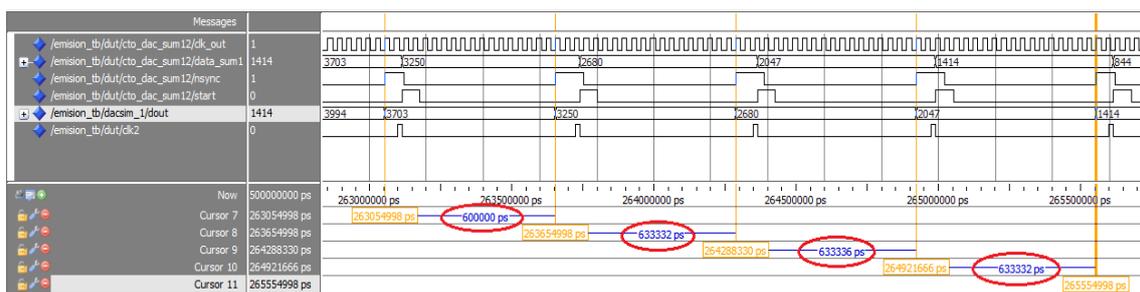


Fig. 30. Cronograma de comunicación SPI.

En la Fig. 31 es posible apreciar como el ancho del pulso de la señal *nSYNC* varía entre 2 y 3 pulsos de reloj; como ya se ha mencionado, éste debe durar al menos un pulso de reloj por lo que la temporización de la comunicación es correcta.

Implementación del bloque emisor

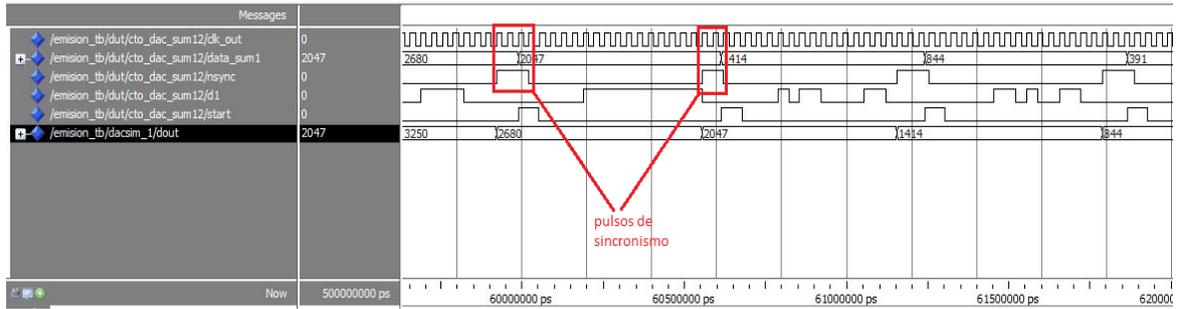


Fig. 31. Cronograma de simulación de comunicación SPI entre emisor y DAC.

A continuación se muestra el cronograma (Fig. 32) de la simulación en la que se representan, de forma simbólica, la salida de cada uno de los modelos de DAC. Esto se realiza para una emisión que pretende deflectar el haz en un ángulo de 64° emitiendo un único código y generando los retardos necesarios.

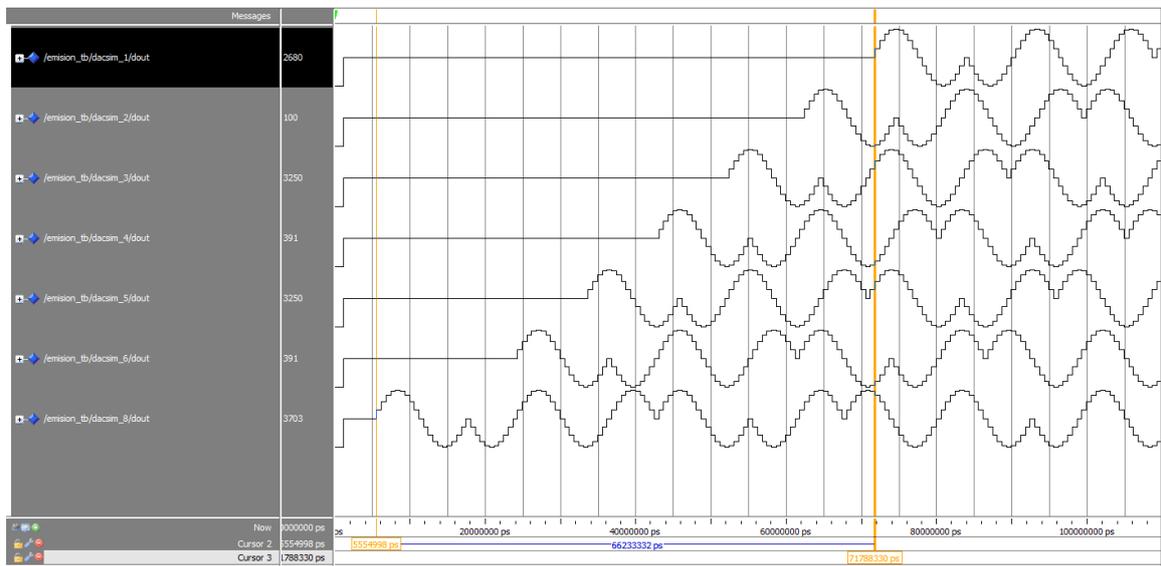


Fig. 32. Simulación de la emisión de un único código con un ángulo de 64° .

En la Tabla 11 se muestran los resultados en μs obtenidos en *Matlab* para cada uno de los retardos, con respecto al primer transductor que comienza a emitir, para la emisión de un único código a 64° .

Tabla 11 Retardos ideales en cada convertidor D/A para una deflexión a 64° .

D/A 1	D/A 2	D/A 3	D/A 4	D/A 5	D/A 6	D/A 7	D/A 8
66.03 μs	56.60 μs	47.27 μs	37.73 μs	28.30 μs	18.87 μs	9.43 μs	0 μs

La resolución del sistema hardware planteado viene dada por la frecuencia de conversión del sistema. Como ya se ha mencionado, dicha frecuencia es de 1.6 MHz, lo que indica que la resolución en la obtención de los retardos reales y por tanto también el error máximo es de 625 ns, a lo que se debe sumar el error de conversión anteriormente descrito de 33 ns, es decir, un error total de 658 ns. Los valores obtenidos tras la simulación se muestran en la Tabla 12:

Tabla 12 Retardos reales en cada convertidor D/A para una deflexión a 64°.

D/A 1	D/A 2	D/A 3	D/A 4	D/A 5	D/A 6	D/A 7	D/A 8
66.25 μ s	56.87 μ s	47.88 μ s	37.53 μ s	28.13 μ s	18.75 μ s	9.37 μ s	0 μ s

Se puede apreciar como los valores son muy parecidos a los ideales a pesar de las fuentes de error mencionadas.

6.4 Recursos FPGA

En la Tabla 13 se muestran los recursos consumidos por cada uno de los bloques del módulo emisor en su implementación en la FPGA Xilinx Virtex5 LX50T. Las principales limitaciones del diseño vienen marcadas por la longitud $L_{M_s n}$ de los códigos, debido al consumo de memoria, así como por la resolución temporal t_e escogida a la hora de generar los retardos. Esta última influye en el tamaño del bloque de retardos, así como en la frecuencia máxima de conversión f_s del DAC.

Tabla 13 Recursos consumidos por el módulo Emisor propuesto en la FPGA Xilinx Virtex5 LX50T.

Recursos emisión/ Porcentaje de ocupación		
Bloque	Slices	BRAMs
Control_CSS	62 (0,86 %)	0 (0,00%)
Cx	0 (0,00%)	64 (53,33%)
BPSK	576 (8,00%)	32 (26,67%)
FIFO	1984 (27,56%)	0 (0,00%)
SUM32	928 (12,89%)	0 (0,00%)
DAC_ctrl	260 (3,61%)	0 (0,00%)
Total	3810 (52,92%)	96 (80,00%)

7 Implementación del bloque receptor

El apartado de recepción se divide en tres partes:

1. Adquisición y correlación.
2. Envío de datos.
3. Post-procesado.

Es condición de diseño el adquirir y procesar en tiempo real. Se ha optado por usar una plataforma basada en FPGA, igual a la empleada para el procesamiento digital de la señal en el apartado de emisión, para el procesamiento de bajo nivel. Este procesamiento incluye la demodulación BPSK asíncrona de las señales capturadas y su posterior correlación con las macro-secuencias Ms_n emitidas. Los resultados de la correlación se envían a un ordenador donde se lleva a cabo el post-proceso de alto nivel (detección de envoltorio y composición de la imagen). Para gestionar el envío de los resultados de correlación se utiliza un microprocesador embebido en el FPGA SOC (*System On Chip*) de tipo *Microblaze*. La Fig. 33 muestra el flujo de ejecución del bloque receptor.

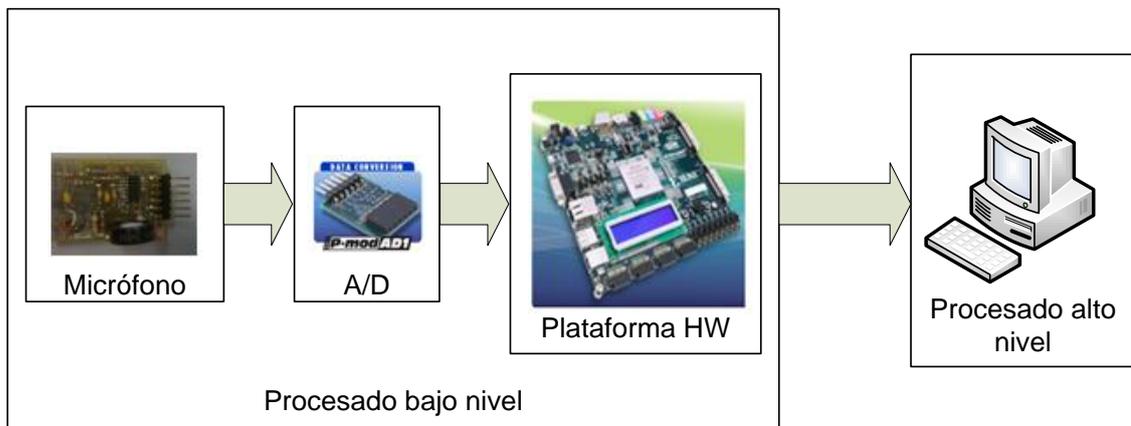


Fig. 33. Flujo de ejecución del módulo receptor.

La adquisición, demodulación y procesamiento de la señal recibida (correlación) han sido implementados sobre los bloques lógicos de la FPGA. Los resultados de esta operación se almacenan en memoria externa DDR2 a través de un DMA (*Direct Memory Access*) usando un protocolo NPI. La información es leída por *Microblaze* y enviada al PC para su procesamiento.

7.1 Plataforma Hardware

En este apartado se detallan los elementos hardware que intervienen en el sistema para el módulo de recepción. Asimismo se especifica cuáles han sido los criterios de elección o de diseño del hardware necesario, tenidos en cuenta para el desarrollo de este módulo.

7.1.1 Procesado Digital

Este diseño requiere de una plataforma que disponga de todos los recursos necesarios para la implementación de un correlador eficiente OCCS así como de conexiones con los elementos físicos que permitan comunicaciones de alta velocidad Ethernet. Puesto que se pretende realizar la correlación de macro-secuencias M_{s_n} empleando una arquitectura que lleva a cabo la correlación en paralelo de todas las secuencias $s_{n,n}$ del conjunto S_n que forma dicha macro-secuencia, es necesario un bloque retardador que permita pasar de la recepción serie de las secuencias a una paralelización de éstas que ataque al bloque de correlación. Por lo tanto, los requisitos de memoria son los siguientes (24) (25):

$$M_{\text{correlador}} = \frac{N^2 - N}{2} \cdot (DW + \log_2(N)) = 806 \text{ bytes} \quad (24)$$

$$M_{\text{retardos}} = (w_o + N) \cdot (N - 1) \cdot DW \cdot O_f = 957280 \text{ bytes} \quad (25)$$

Siendo $N=32$; $w_o=354$; $DW=8$; $L=32$ y $O_f=10$. Debido a la arquitectura del bloque de correlación, la memoria requerida por este algoritmo es implementada sobre la lógica del dispositivo mientras que la memoria empleada en el bloque de retardos es obtenida de las pastillas de memoria RAM contenidas en el dispositivo.

Por otro lado, el resultado de la operación debe ser almacenado en memoria antes de su emisión para su posterior procesado. Los datos de correlación que interesan son los correspondientes a la IFW. La memoria ($M_{\text{resultado}}$) necesaria para almacenar todos los datos deseados es de 2 Mbit (26), lo que representa el 104% de la memoria interna del dispositivo *Virtex5 LX50T*.

$$M_{\text{resultado}} = w_o \cdot D_{\text{out}} \cdot O_f \cdot N \quad (26)$$

Donde D_{out} representa el tamaño de los datos a la salida del bloque de correlación.

Además de los requisitos mencionados, es preciso el empleo de un dispositivo que permita llevar a cabo comunicaciones de alto nivel para comunicar el dispositivo con el PC, de forma eficiente y rápida. Dado que realizar un driver hardware para el control de un periférico de comunicación de tipo Ethernet sobre una FPGA resulta arduo, se ha optado por una plataforma software para el desempeño de esta función.

Teniendo estos factores en cuenta se ha utilizado una placa de evaluación, *Genesys*, desarrollada por *Digilent Inc.* basada en una FPGA Xilinx *Virtex5 LX50T* [Xi11], similar a la empleada para el módulo de emisión. Esto se debe a que dispone de memoria externa de acceso rápido DDR2 y los recursos necesarios para la implementación de un SoC de tipo *Microblaze* y para la implementación de los bloques lógicos del diseño, así como el hardware necesario para llevar a cabo una comunicación Ethernet.

7.1.1.1 *System On Chip (Microblaze)*

Ya se ha mencionado qué parte de la funcionalidad de este diseño se ha realizado sobre un SoC (*System On Chip*), concretamente un microcontrolador *soft core* instanciado sobre la lógica del dispositivo FPGA de tipo *Microblaze*. La función principal que desempeña este sistema es la de leer los datos almacenados en memoria externa DDR2 correspondientes al resultado de la operación de correlación y enviarlos a través de un puerto Ethernet a un PC.

Para la generación del SoC sobre la plataforma FPGA Virtex5 es preciso el uso de la herramienta XPS (*Xilinx Platform Studio*) contenida en el paquete EDK (*Edit Desing Kit*) de Xilinx. Esta herramienta permite crear mediante una interface gráfica el SoC configurando diferentes parámetros tales como:

- Número de procesadores del sistema (un solo procesador o coprocesador). En este diseño se ha generado un sistema de un único procesador.
- Configuración del procesador: procesador de tipo *Microblaze* con frecuencia de trabajo de 100 MHz.
- Tipo de bus interno utilizado en el sistema (PLB o AXI). En este caso se han usado buses de tipo PLB ya que los buses de tipo AXI no son soportados sobre la plataforma Virtex 5.
- Los periféricos que va a tener el sistema. En este diseño se han utilizado: controlador de DDR, MAC de Ethernet y UART RS232.
- Cantidad de memoria interna y cache que va a tener el sistema. En este caso se cuentan con 32 kb de memoria interna.

Para facilitar la configuración de estos parámetros se hace uso del *wizard* BSB (*Board System Builder*) que permite crear un sistema completo sin necesidad de partir de cero. Esto es posible gracias al fichero .bsb proporcionado por *Digilent inc.*⁹ en su página web. En la Fig. 34 se muestra el flujo de configuración de las especificaciones del sistema.

⁹ <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,819&Prod=GENESYS>. consultado 05/05/2014.

Implementación del bloque receptor

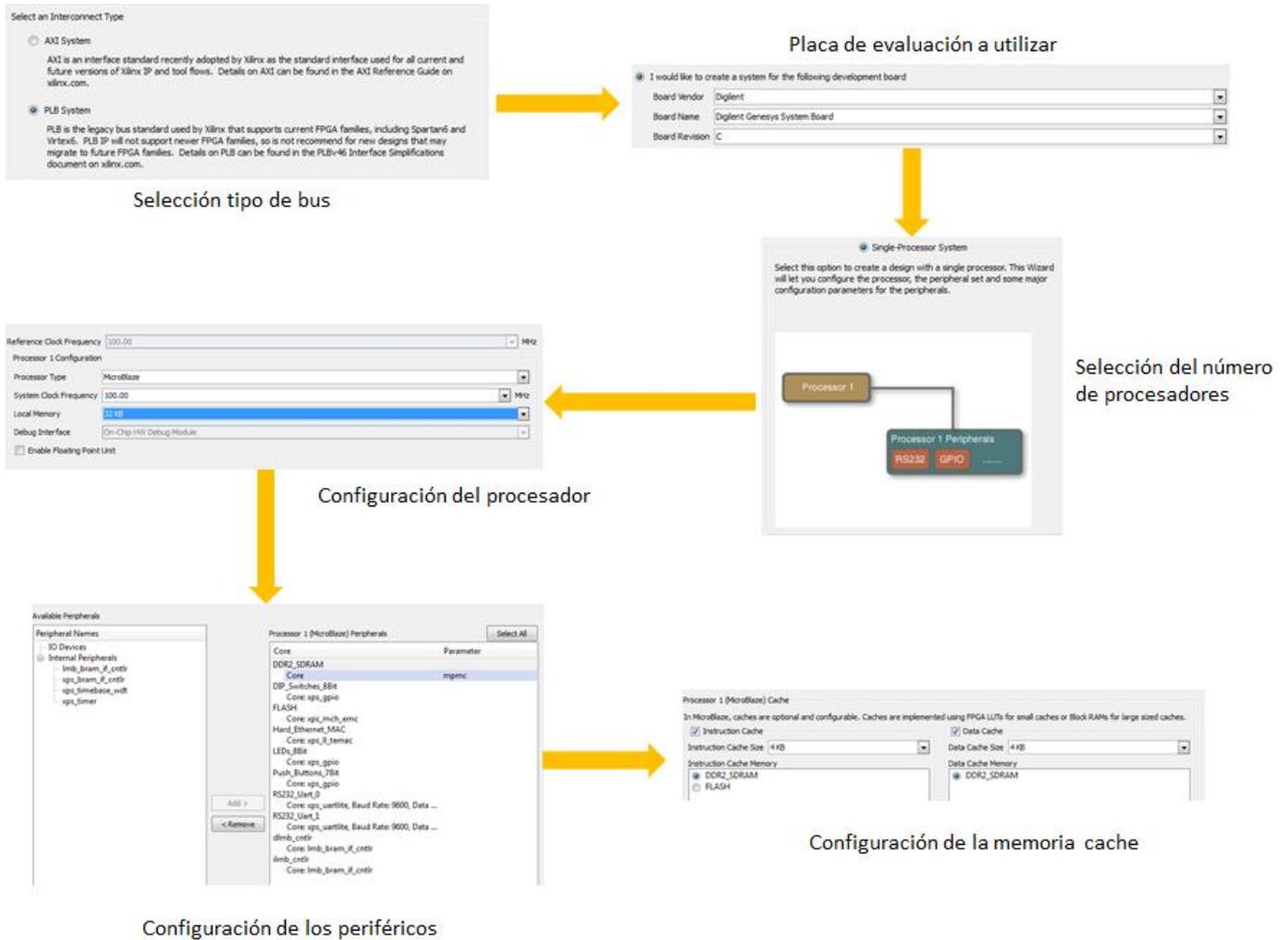


Fig. 34. Creación del SoC. Especificaciones del sistema.

Una vez completadas las especificaciones del sistema se definen las interconexiones de los diferentes elementos que forman el SoC. De esta manera, según muestra en la Fig. 35 a través de una interfaz gráfica se conectan todos los periféricos al bus PLB, el cual permite la comunicación con *Microblaze*. Asimismo, se conecta el periférico *custom*, que realiza la función de correlación, al controlador de memoria DDR2 a través un bus NPI (*Native Peripheral Interface*) (7.1.2), y dicho controlador de memoria es conectado a *Microblaze* por medio del bus PLB.

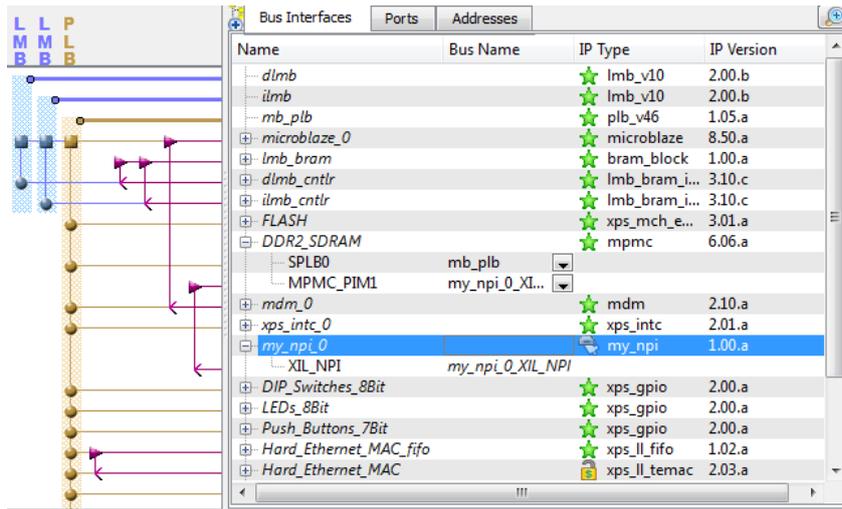


Fig. 35. Bus interface entre *Microblaze* y los periféricos que forman el SoC.

A continuación se determina el conexionado de los puertos del sistema (Fig. 36). Esto permite realizar conexiones a nivel de bit entre elementos internos del sistema y acceder a los pines de la FPGA. Por último se asigna una dirección a cada uno de los periféricos conectados al bus PLB (Fig. 37).

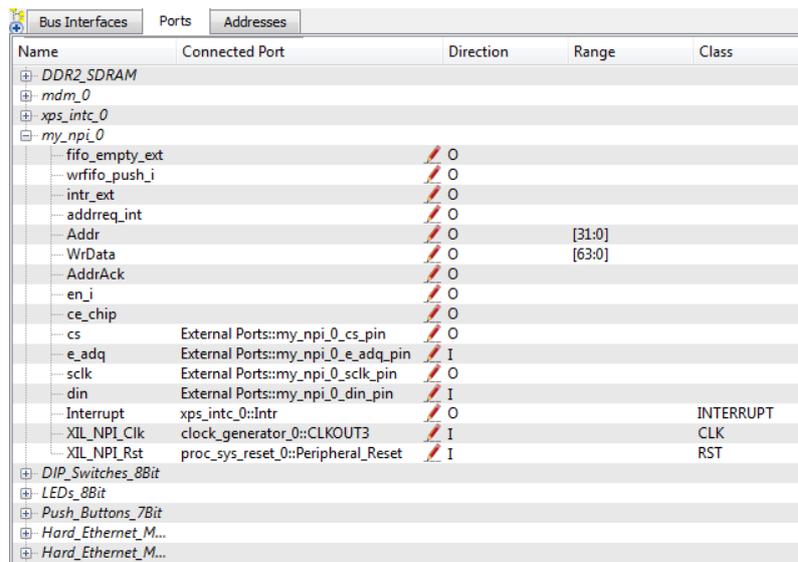


Fig. 36. Asignación de puertos del SoC.

Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name	Lock
microblaze_0's Address Map							
dmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF	32K	SLMB	dmb	<input type="checkbox"/>
ilmb_cntlr	C_BASEADDR	0x00000000	0x00007FFF	32K	SLMB	ilmb	<input type="checkbox"/>
Push_Buttons_7Bit	C_BASEADDR	0x81400000	0x8140FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
LEDs_8Bit	C_BASEADDR	0x81420000	0x8142FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
DIP_Switches_8Bit	C_BASEADDR	0x81440000	0x8144FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
xps_intc_0	C_BASEADDR	0x81800000	0x8180FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
Hard_Ethernet_MAC_fifo	C_BASEADDR	0x81A00000	0x81A0FFFF	64K	SPLB	mb_plb	<input checked="" type="checkbox"/>
xps_timer_0	C_BASEADDR	0x83C00000	0x83C0FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
RS232_Uart_1	C_BASEADDR	0x84000000	0x8400FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
RS232_Uart_0	C_BASEADDR	0x84020000	0x8402FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
mdm_0	C_BASEADDR	0x84400000	0x8440FFFF	64K	SPLB	mb_plb	<input type="checkbox"/>
Hard_Ethernet_MAC	C_BASEADDR	0x87000000	0x8707FFFF	512K	SPLB	mb_plb	<input type="checkbox"/>
FLASH	C_MEM0_BASE...	0x8C000000	0x8DF7FFFF	32M	SPLB	mb_plb	<input type="checkbox"/>
DDR2_SDRAM	C_MPMC_BASE...	0x90000000	0x9FFFFFFF	256M	SPLB0	mb_plb	<input checked="" type="checkbox"/>

Fig. 37. Asignación de direcciones a los periféricos del SoC conectados al bus PLB.

Una vez finalizados estos pasos se tiene definido el SoC utilizado para este diseño.

7.1.2 Acceso a Memoria

En este diseño la adquisición, demodulación y procesado de la señal recibida (correlación) han sido implementados sobre los bloques lógicos de FPGA. Los resultados de esta operación se almacenan en memoria externa DDR2 a través de un DMA (*Direct Memory Access*) usando un protocolo NPI. La información es leída y enviada a través de *Microblaze*. Estos factores tienen dos implicaciones importantes: se deben escribir datos en tiempo real en memoria externa y compartir memoria entre dos elementos del sistema (lógica del sistema y *Microblaze*).

Por este motivo es preciso el uso de un controlador de memoria multipuerto MPMC (*Multi Port Memory Controller*) [Xil11]. Este controlador permite la conexión de hasta 8 puertos y configurar el tamaño del ancho de datos de 4 a 64 bits.

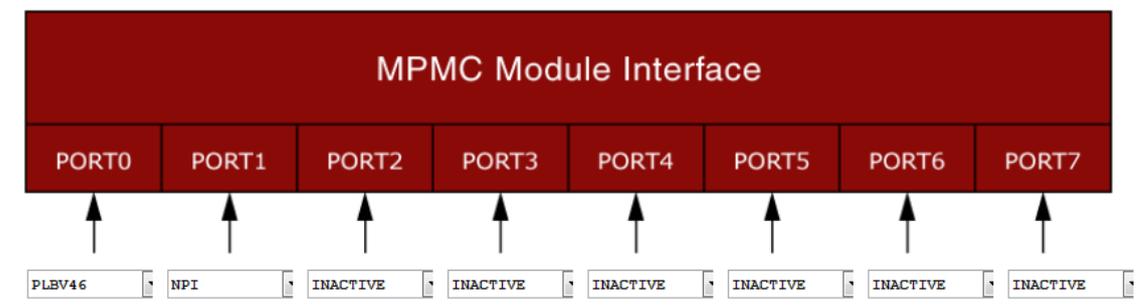


Fig. 38. Interface del controlador de memoria multipuerto MPMC.

Es posible acceder al MPMC utilizando varios protocolos de comunicación. En este caso van a utilizarse dos:

1. PLB (*Peripheral Local Bus*) empleado en las comunicaciones entre periféricos. Este protocolo es el que se utiliza para el acceso a memoria externa desde *Microblaze* y resulta transparente al diseñador.

2. NPI (*Native Peripheral Interface*) bus de comunicación punto a punto. Permite el acceso a memoria de forma sencilla y rápida. Este protocolo es el utilizado para la escritura de datos desde los bloques de lógica a memoria externa.

La siguiente figura muestra la conexión física entre del *data path* del controlador MPMC con memoria DDR.

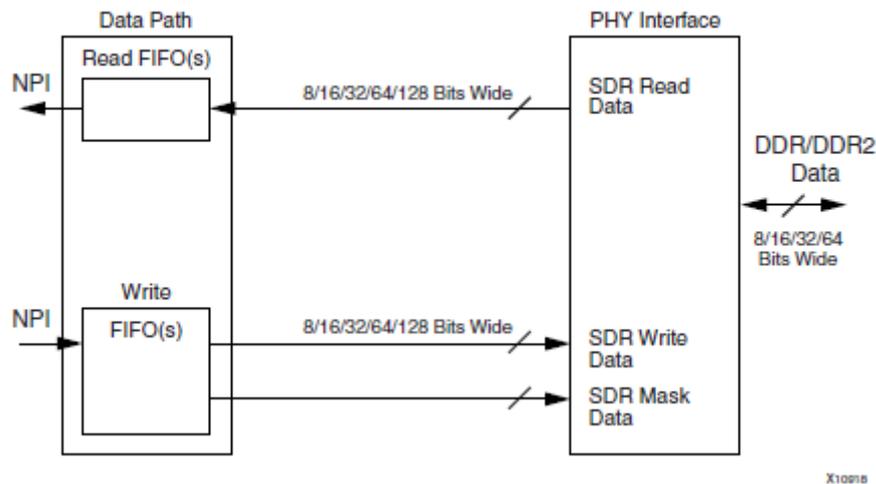


Fig. 39. Conexión física del Data path del MPMC [Xil11].

La interface física convierte el bus *Double Data Rate* (DDR) de la memoria externa en *Single Data Rate* (SDR). Esto es dos veces el ancho de datos de memoria externa. Para memorias SDRAM los datos siguen siendo del mismo tamaño al pasar por la interface física. Ya que el MPMC soporta 8,16,32 y 64 bits de memorias SDRAM y DDR/DDR2, esto se traduce en un bus SDR de 8, 16, 32, 64 o 128 bits conectado con bloque de *data path*.

El protocolo de más bajo nivel utilizado para comunicar los distintos elementos con el *data path* del controlador es el NPI.

7.1.2.1 Protocolo NPI

Como acaba de mencionarse este protocolo es relativamente sencillo de implementar y permite la comunicación punto a punto con el controlador multipuerto MPMC. El protocolo NPI tiene las siguientes características:

- Permite la conexión del MPMC adaptada a las necesidades del sistema.
- Proporciona señales de dirección, datos y control para habilitar cualquier petición de acceso a memoria.
- Permite hacer simultáneamente un *push* y un *pull* sobre las FIFO de los puertos del MPMC.
- Permite escribir datos de 32 ó 64 bits
- Se soportan transferencias de: *byte*, *half-word*, *word*, *4-word cacheline*, *8-word cacheline*, ráfaga de 32 *word* y ráfaga de 64 *word*.

La frecuencia de funcionamiento del bus NPI debe ser la misma que la del MPMC, que en este caso es de 200 MHz. De estos modos de funcionamiento, en este trabajo se ha usado el modo de ráfaga de 32 *word* (cada *word* formado por dos bytes) y un ancho de bus de 64 bits, por lo que en cada dato a escribir se mandan dos *word*.

La Fig. 40 muestra el cronograma obtenido del *datasheet* del MPMC que refleja el funcionamiento de la escritura en ráfaga de 16 datos (señal *Size*) con un ancho de 64 bits. Al activar la señal *WrFIFO_Push* se comienzan a guardar los datos de la ráfaga en la memoria RAM del controlador. Cuando la memoria RAM empieza a almacenar los datos el controlador desactiva la señal *WrFIFO_Empty*. Una vez se ha escrito el último dato D15 se realiza la petición de volcado de datos sobre memoria externa mediante la señal *AddrReq*, el controlador valida dicha petición activando la señal *AddrAck*. Llevar a cabo esta petición en este instante implica que no se pierden datos, ya que se asegura que todos los datos de la ráfaga ya se encuentran en la memoria RAM. Una vez finalizado el volcado de los datos de la RAM en memoria externa el controlador vuelve a activar la señal *WrFIFO_Push* lo que indica que puede volverse a hacer una nueva escritura.

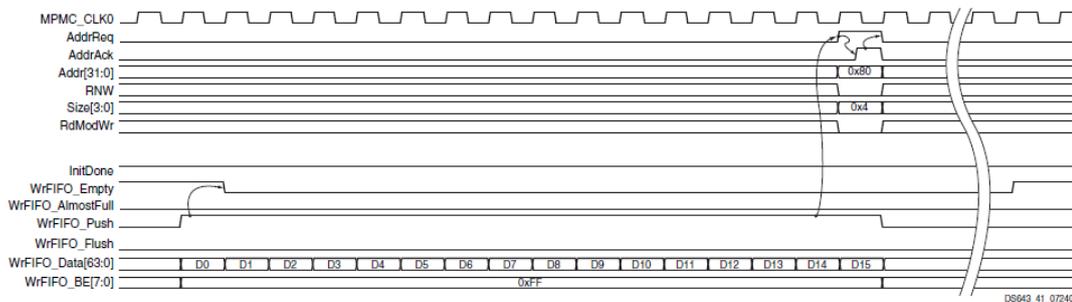


Fig. 40. Cronograma protocolo NPI en modo ráfaga de 16 datos de 64 bits [Xil11].

7.1.3 Conversor Analógico Digital

Ya que es necesario convertir las señales analógicas de los ecos recibidos a señales digitales para su procesamiento, se ha utilizado un conversor A/D AD7476A SPI de 12 bits de resolución montado sobre un módulo P-MOD PmodAD1 (Fig. 41) creado por Digilent [Dig11].

Las principales características del conversor AD7476A son:

- Frecuencia de conversión de hasta 1 MSPS.
- Bajo consumo.
- SPAN: 0 a 4096.
- Rango de alimentación de 2.35 V a 5.25 V
- Comunicación vía SPI con una frecuencia máxima de 20 MHz.

El PmodAD1 además del dispositivo AD7476A contiene un filtro *anti-aliasing Sallen-Key* de orden dos cuyos polos se encuentran situados en 500 kHz. Este filtrado limita el ancho de banda en función de la frecuencia de muestreo.

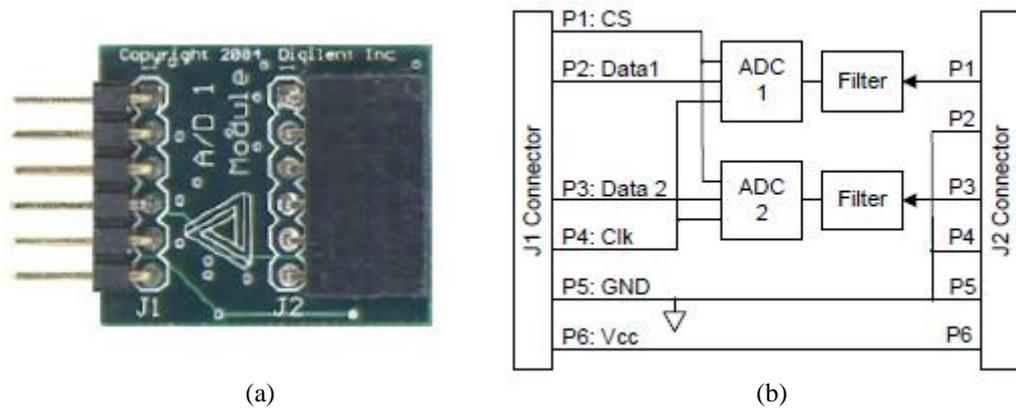


Fig. 41. (a) PmodAD1. (b) Diagrama de bloques de PmodAD1.

7.2 Implementación Hardware

7.2.1 Esquema

A continuación se muestra el diagrama de bloques que reflejan la funcionalidad de los bloques hardware que se han implementados sobre la plataforma FPGA.

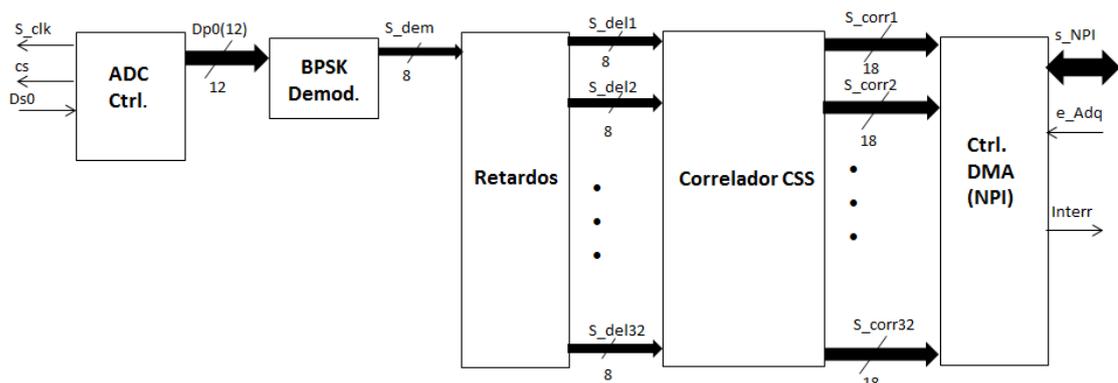


Fig. 42. Diagrama de bloques HW del módulo receptor.

7.2.2 Bloques lógicos

El funcionamiento del bloque receptor consiste en:

1. Realiza la conversión de la señal recibida a una frecuencia de muestreo de 800 kHz.
2. Demodulación BPSK de la señal adquirida.
3. Paralelización de la señal obtenida para conseguir que todas las secuencias $s_{n,n}$ de una misma macro-secuencia Ms_n entren al correlador de CSS de forma simultánea.
4. Correlación de la señal recibida
5. Escritura en memoria externa DDR2.

A continuación se enfatiza sobre cada uno de los bloques lógicos que forman el módulo receptor.

7.2.2.1 Módulo ADC ctrl.

En la Tabla 14 se muestran las señales de entrada y salida del módulo ADC_ctrl.

Tabla 14 Puertos de entrada y salida del módulo ADC_ctrl.

Señal	Tipo	Tamaño (bits)	Descripción
Mclk	In	1	Reloj del sistema
Mrst	In	1	Reset del sistema
Init	In	1	Inicio de conversión
ds0	In	1	Dato serie recibido del ADC
Sclk	Out	1	Reloj del bus SPI
Cs	Out	1	Chip select del bus SPI
dp0	Out	12	Dato adquirido digitalizado

Este módulo tiene como función la comunicación vía SPI con el ADC AD7476A [AD14]. El inicio de una conversión y de la comunicación se realiza con el flanco de bajada de la señal *cs*. Esta señal lleva al ADC a modo *hold*, quita el estado del bus de alta impedancia y muestrea la señal analógica en ese instante. La conversión necesita 16 flancos de bajada de la señal *sclk* para completarse. Cuando esta conversión termina el bus vuelve al estado de alta impedancia. Los datos de la línea *ds0* son válidos en los flancos de bajada de la señal *sclk*. De los 16 bits que se transmiten, los 4 primeros son 0 y los siguientes 12 corresponden al dato adquirido. En la Fig. 43 se muestra el cronograma de la comunicación SPI con el AD7476A.

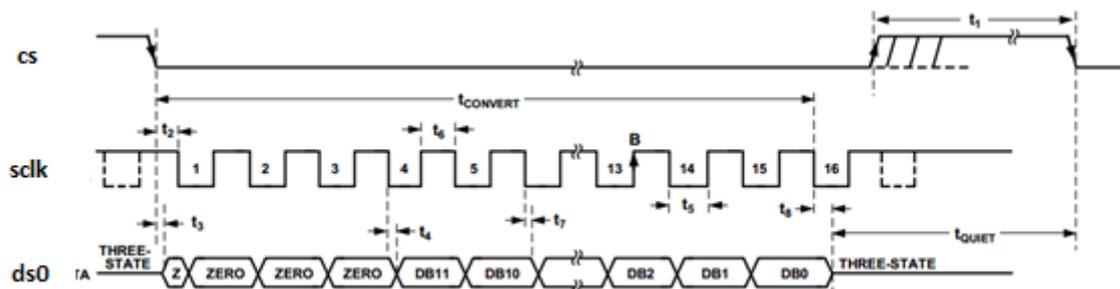


Fig. 43. Cronograma del protocolo de comunicación SPI con AD7476A [AD14].

Cada conversión está sincronizada con la señal de *Init* que genera un pulso a nivel con una frecuencia de 800 kHz. En este momento, pasa del estado de *Idle* al estado de *qReading*, donde se permanece durante 17 flancos de bajada de la señal *sclk*. En los 16 primeros se captura la señal entrante mientras que en el último se actualiza la señal *dp0*, esta cuenta se refleja en la señal interna *cntbit*. Por último, se activa la señal interna *EndConv* que indica el final del proceso de adquisición y se vuelve al estado de *Idle* a la espera de una nueva conversión (véase la Fig. 44).

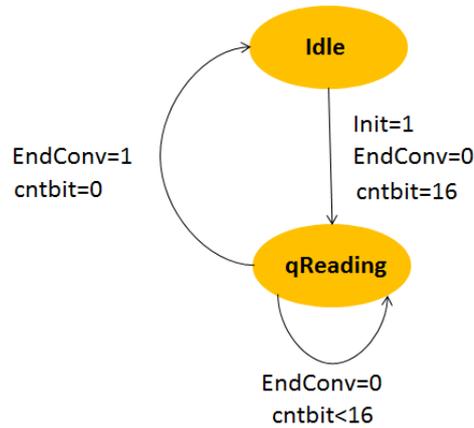


Fig. 44. Máquina de estados del funcionamiento del módulo ADC_ctrl.

La frecuencia de funcionamiento del bus SPI es de 16 MHz con lo que a una frecuencia de conversión de 800 kHz no existen problemas de pérdida de datos ya que entre conversiones se tiene una latencia de 250 ns y el tiempo mínimo que la señal *cs* puede estar a nivel alto es de 10 ns.

7.2.2.2 Módulo Demod. BPSK

En la Tabla 15 se muestran las señales de entrada y salida del módulo Demod BPSK.

Tabla 15 Puertos de entrada y salida del módulo demod BPSK.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
reset	In	1	Reset del sistema
enable	In	1	Clock enable del bloque
din	In	12	Datos obtenidos del controlador de ADC
dout	Out	8	Datos de salida demodulados

Este bloque tiene como función la demodulación de la señal adquirida con el fin de extraer los códigos emitidos. Esta demodulación se lleva a cabo según se describió en el apartado 5.3.1, realizando la operación de correlación de la señal entrante con el símbolo de modulación.

En la siguiente Fig. 45 se puede observar la arquitectura interna del demodulador: el bloque RAM es la memoria que contiene el símbolo de modulación; *Buf_adq* es el registro de desplazamiento, del mismo tamaño que el símbolo de modulación, que recibe los datos entrantes del bloque **ADC_ctrl**; *MUL* es un multiplicador de 18 bits; *SUM* es un sumador de 12 bits y por último *Acc* es el registro acumulador donde se almacena el resultado de la suma el cual se realimenta para sumarse al resultado de la siguiente multiplicación

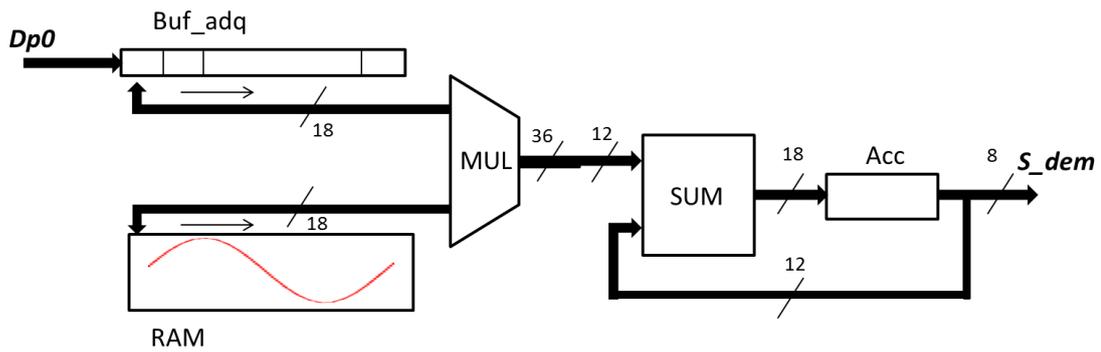


Fig. 45. Diagrama de bloques del proceso de demodulación.

Cada vez que una muestra es adquirida y almacenada en *Buf_adq* se lleva a cabo una multiplicación de todos los datos almacenados en *Buf_adq* con el elemento correspondiente del símbolo de modulación que se encuentra en el bloque *RAM*. Los resultados se suman a la operación anterior acumulada en el registro *Acc* a través de *SUM*. Esta operación se lleva a cabo tantas veces como elementos tenga el símbolo de modulación. De esta manera, la latencia que este bloque necesita para alcanzar el resultado final es el periodo de dicho símbolo.

7.2.2.3 Módulo Retardos

En la Tabla 16 se muestran las señales de entrada y salida del módulo Retardos.

Tabla 16 Puertos de entrada y salida del módulo Retardos.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
rst	In	1	Reset del sistema
ce	In	1	Clock enable del bloque
data_in	In	8	Datos truncados obtenidos del demodulador BPSK
data_out	Out	$8 \cdot M$	Datos de salida demodulados

Este bloque tiene como función convertir las secuencias $s_{n,n}$ recibidas en serie en la macro-secuencia Ms_n . De esta manera, se necesitan $N-1$ memorias FIFO, diseñadas a partir de bloques BRAM obtenidos de las primitivas *RAMB_16_9S_9S* de 16 kbit de memoria de datos. Cada una de estas FIFO está formada por dos primitivas, con lo que tienen un tamaño de 4 kbyte. El espacio de memoria que se utiliza en cada una de las FIFO es $O_f \cdot (w_o + N) = 3860 \text{ bytes}$. Son necesarios tantos de estos bloques como secuencias se han utilizado en la formación de las macro-secuencias. En Fig. 46 se muestra el diagrama de bloques del módulo Retardos.

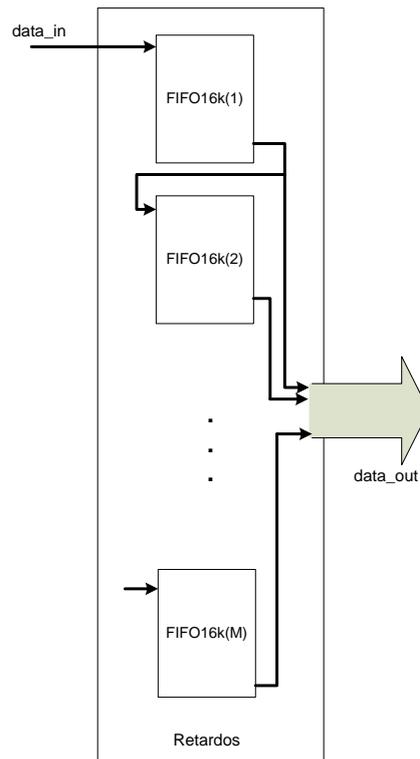


Fig. 46. Diagrama de bloques del módulo Retardos.

7.2.2.4 Módulo Correlador CSS

En la Tabla 17 se muestran las señales de entrada y salida del módulo Correlador CSS.

Tabla 17 Puertos de entrada y salida del módulo Correlador CSS.

Señal	Tipo	Tamaño (bits)	Descripción
clk	In	1	Reloj del sistema
rst	In	1	Reset del sistema
ce	In	1	clock enable del bloque
e_Adq	In	1	Señal de sincronización con el emisor
ent	In	$8 \cdot N$	Datos de entrada al correlador
semilla	In	$\log_2 N$	Semilla ¹⁰ de generación del código
sal	Out	$N \cdot (\log_2 N \cdot 2 + 8)$	Salida de correlador CSS ¹¹
enable	Out	1	Salida de habilitación de escritura en memoria externa

¹⁰ En este diseño el ancho de esta señal es de 5 bits

¹¹ En este diseño el ancho de esta señal es de 576 bits

La función principal de este bloque es la de realizar la operación de correlación con la arquitectura mencionada en el punto 5.3.2. En este trabajo, la correlación se realiza de 32 conjuntos formados por 32 secuencias. De esta manera, se tienen 32 entradas de 8 bits y 32 salidas de 18 bits. Cada una de estas salidas corresponde con la función de correlación de la señal adquirida con la correspondiente macro-secuencia M_{S_n} .

Además del proceso de correlación este bloque habilita la escritura en memoria externa de los datos de interés. Es decir, se sincroniza con el emisor para habilitar la escritura de datos durante la ventana temporal correspondiente a la obtención de los resultados de correlación en la IFW.

La figura Fig. 47 muestra la sencilla máquina de estados en la que se basa el proceso de sincronización y habilitación de escritura en memoria externa. Se parte del estado de espera $st0$. Al recibirse la señal de sincronismo e_Adq , procedente del bloque emisor, se comienza a contar en función de la longitud del código (27) y de la conformación del haz (campo lejano), y se pasa al estado $st1$.

$$cuenta_{sync} = (N + wo) \cdot (N - 1) \cdot O_f + v \cdot 2 \cdot NF(m) \cdot f_{sr} \quad (27)$$

Al finalizar $cuenta_{sync}$ se transita al siguiente estado $st2$ donde se activa la señal de $enable$, que habilita la escritura en memoria externa. Dicha señal se lleva a nivel alto el tiempo necesario para completar la escritura de la semi-ventana libre de interferencias wo correspondiente a la ventana temporal de exploración (28).

$$cuenta_{enable} = wo \cdot O_f \quad (28)$$

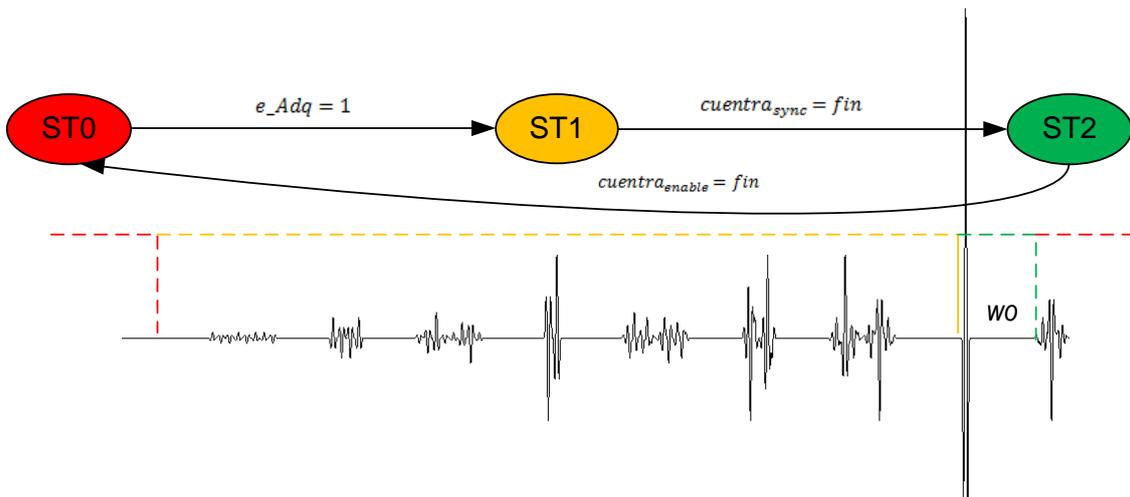


Fig. 47. Representación del proceso de sincronización y habilitación de escritura de datos de la función de correlación obtenida del correlador CSS para una macro-secuencia de $N=8$ y $wo=16$; $f_{sr}=80$ kHz y $O_f=20$.

7.2.2.5 Módulo DMA_NPI

En la Tabla 18 se muestran las señales de entrada y salida del módulo DMA_NPI.

Tabla 18 Puertos de entrada y salida del módulo DMA_NPI.

Señal	Tipo	Tamaño (bits)	Descripción
XIL_NPI_Clk	In	1	Reloj del bus NPI
XIL_NPI_Rst	In	1	Reset del bus NPI
Ce	In	1	Clock enable del bloque
Interrupt			
XIL_NPI_Addr	Out	32	Dirección sobre la que se quiere escribir en memoria externa.
XIL_NPI_AddrReq	Out	1	Solicitud de volcado de datos de memoria FIFO a DDR2
XIL_NPI_AddrAck		1	Reconocimiento de solicitud de volcado de datos memoria FIFO a DDR2
XIL_NPI_Size		4	Modo de escritura
XIL_NPI_WrFIFO_Data		64	Bus de datos
XIL_NPI_WrFIFO_BE		8	Byte strobe
XIL_NPI_WrFIFO_Push		1	Escritura de datos en FIFO
XIL_NPI_WrFIFO_Empty		1	FIFO vacía
XIL_NPI_InitDone		1	Estado del MPMC
XIL_NPI_RNW		1	Señal de lectura o escritura

La función de este bloque consiste en gestionar el proceso de escritura sobre la memoria externa DDR2 de los datos obtenidos en tiempo real del bloque **CorreladorCSS**. Una vez terminado el proceso de escritura se genera una señal de interrupción durante 10 ciclos de reloj a 200 MHz (puede valer con 4 ciclos de reloj a 200 MHz) para asegurar que *Microblaze* detecta dicha señal.

La Fig. 48 representa la máquina de estados en la que se basa el funcionamiento de la escritura en memoria externa. El estado de *IDLE* representa el estado de espera para comenzar la escritura de una trama de 16 datos de 64 bits. Cada dato de 64 bits se construye con los datos de salida obtenidos de dos operaciones de correlación. Dichos datos de salida, en este caso, tienen un tamaño de 18 bits, por esta razón es preciso usar anchos de 32 bits para codificar cada dato de salida. La activación de la señal *enable* que procede del bloque **CorreladorCSS** y *ce* determina el comienzo de una escritura, pasando al estado *TX_DATA* e incrementando la dirección *XIL_NPI_Addr* 128 posiciones (direccionamiento a byte). Este estado corresponde con el proceso de volcado de datos en la memoria FIFO del *MPMC*. Como la comunicación se encuentra configurada a modo ráfaga de 64 *word* (señal *XIL_NPI_Size="0100"*) se van

volcando los datos de forma controlada a través de un contador *data_count* mientras la señal *XIL_NPI_WrFIFO_Push* se mantiene a nivel alto. Al volcar el último dato se activa la señal *XIL_NPI_AddrReq*. Una vez se ha recibido la contestación del *MPMC* se pasa al estado *TX_ADDR*, en el que se permanece hasta que se vacía el contenido de la memoria FIFO y se pone a nivel bajo la señal *XIL_NPI_WrFIFO_Push*. Al vaciarse dicha memoria, se activa la señal *XIL_NPI_WrFIFO_Empty*, lo que produce una transición al estado *INTE*. En este estado se evalúa si se han terminado de escribir todos los datos de interés a almacenar, para lo que se chequea la señal *enable*. Si *enable* ha sido desactivada se produce la activación de la señal *Interrupt* (que produce una interrupción en *Microblaze*), se actualiza la señal *XIL_NPI_Addr* con la dirección de base y se transita al estado *IDLE*. Si *enable* no ha sido desactivada, *Interrupt* no se activa y se pasa a *IDLE* igualmente.

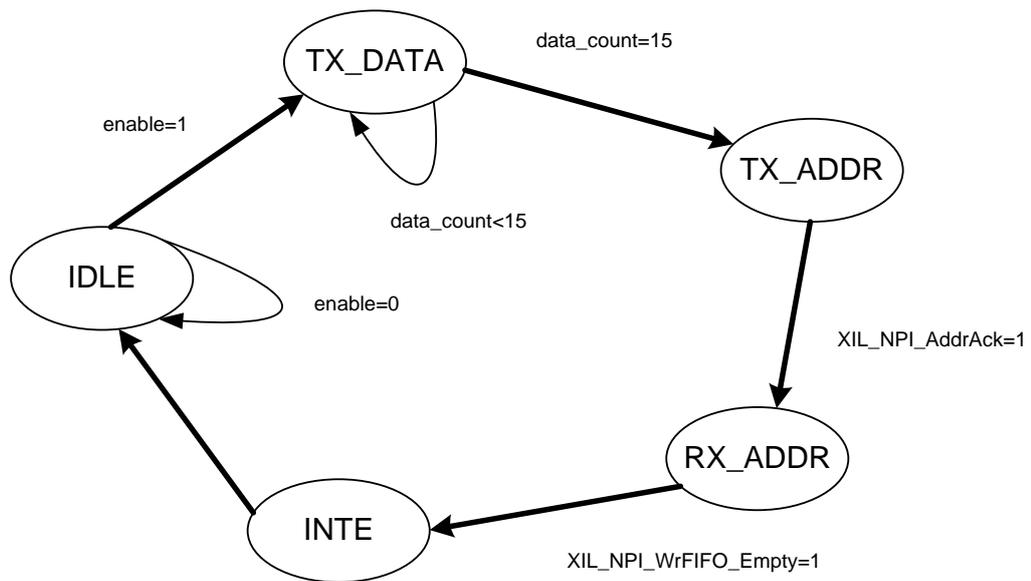


Fig. 48. Máquina de estados del funcionamiento del bloque DMA_NPI.

7.2.3 Microblaze

Para transmitir los datos procesados sobre la placa de desarrollo al PC se ha llevado a cabo una comunicación Ethernet basada en el protocolo TCP/IP. *Microblaze* se define como servidor, mientras que el PC es el cliente el cual solicita los datos. De esta manera, cada vez que *Microblaze* es interrumpido cuando los datos de correlación han sido almacenados en memoria externa, se chequea si ha habido alguna solicitud del PC. Si es así, se realiza el envío de información hacia la IP que realizó la solicitud. Cuando se realiza la solicitud y se produce la interrupción, los datos se re-almacenan en otra posición de memoria externa para evitar el solapamiento de información en el caso en el que se vuelva a producir una escritura de los resultados de correlación mientras se está produciendo el envío de información. Esto ocurre ya que el periodo con el que se ejecuta el procesado en bajo nivel es menor que el tiempo en el que se lleva a cabo la comunicación. En la Fig. 49 se refleja el flujograma de funcionamiento de la aplicación desarrollada sobre *Microblaze*.

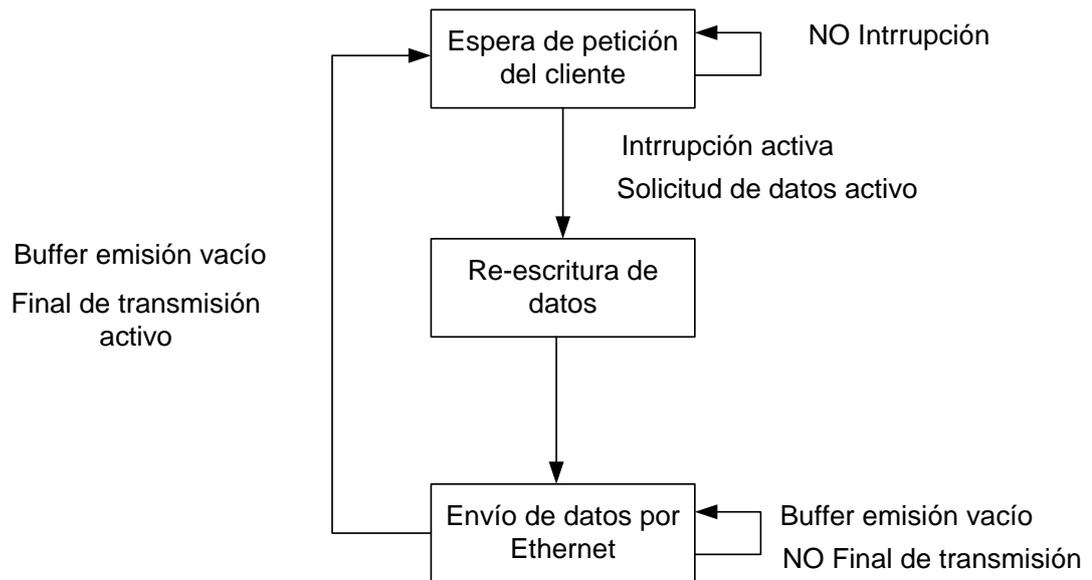


Fig. 49. Flujograma de la aplicación desarrollada sobre *Microblaze*.

Microblaze lee los datos almacenados en DDR y a continuación los envía al periférico de Ethernet a través del bus PLB. Esto produce que la comunicación se haga más lenta, con lo cual, el ratio de transmisión vía Ethernet que se obtiene es de 2250 ms. Dicha medida se ha realizado con una precisión de 25 ms. El total de datos a transmitir es de 453120 bytes.

7.3 Simulaciones

7.3.1 Simulación adquisición

La Fig. 50 representa el modelo de funcionamiento de la comunicación SPI entre el módulo **ADC_ctrl** y el modelo de simulación del ADC. En ella se aprecia como cada $1.25 \mu\text{s}$ se activa la señal de *Init* lo que inicia una nueva conversión. La frecuencia de comunicación del bus es de 16 MHz lo que permite realizar una conversión cada $1 \mu\text{s}$. Esto deja 250 ns de margen entre cada adquisición.

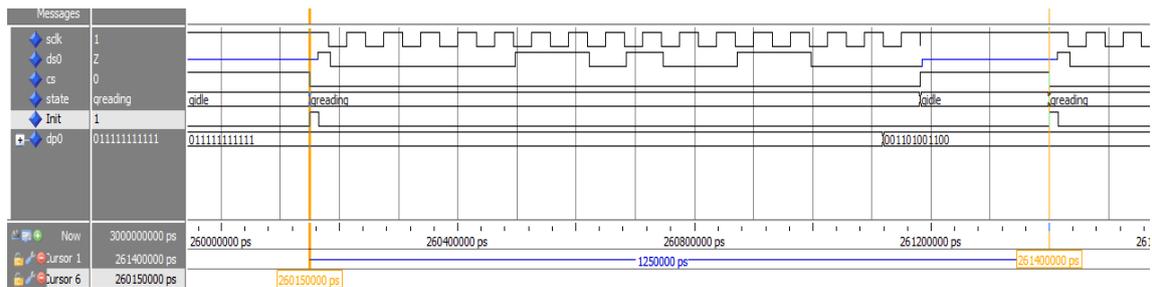


Fig. 50. Simulación de la comunicación con el ADC AD7476A.

7.3.2 Simulación demodulación

En esta simulación se puede apreciar la entrada al bloque **demodBPSK** correspondiente con la señal adquirida y digitalizada en el bloque **ADC_ctrl**, y la señal de salida tras haber realizado el proceso de demodulación. La amplitud máxima de salida tiene como valor 9 y la mínima -9 (Fig. 51). Esto se debe a que al coger los 8 bits de mayor peso de la operación de demodulación, a la salida obtenemos el valor tras la operación dividido entre 16 (4 bits menos).

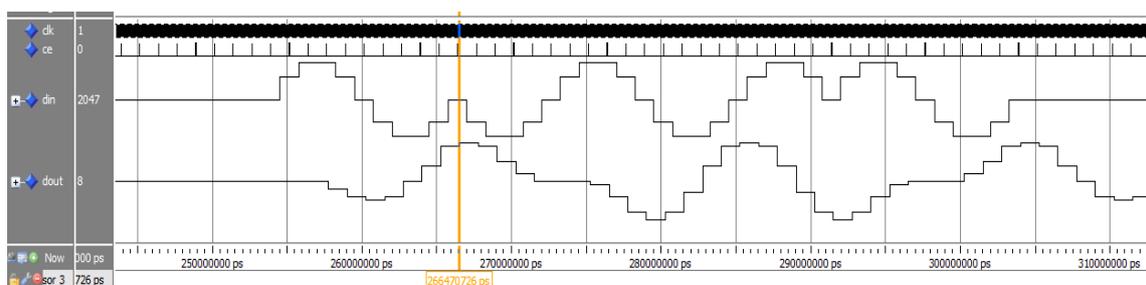


Fig. 51. Simulación de la señal resultante tras realizar la demodulación de una secuencia de 4 elementos con modulación BPSK y portadora a 80 kHz.

En la Fig. 52 se refleja más en detalle el funcionamiento del demodulador. Se aprecia como la amplitud máxima en la señal demodulada se alcanza cuando las muestras almacenadas en el buffer son iguales o inversas a las del símbolo de modulación.

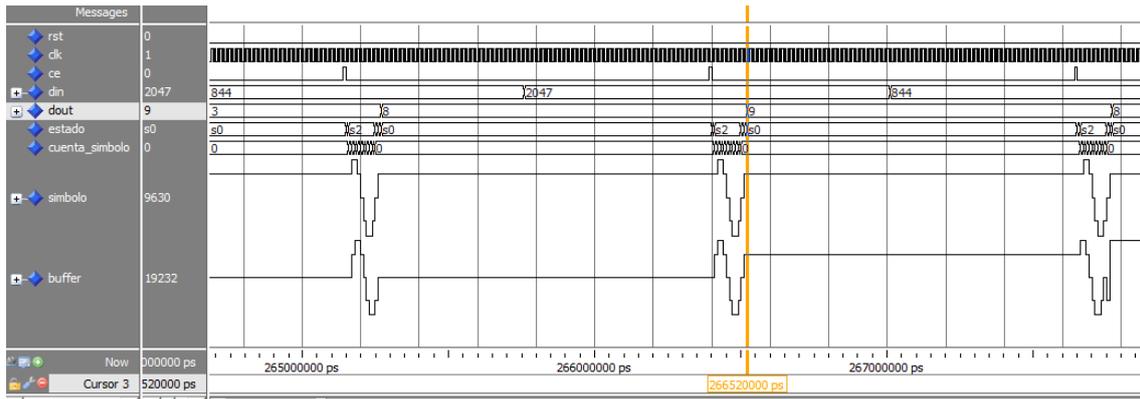


Fig. 52. Simulación del proceso de demodulación.

7.3.2.1 Simulación de retardos

La siguiente simulación muestra el funcionamiento del bloque de retardos que produce la paralelización de las secuencias $s_{n,n}$ que conforman las macro-secuencias Ms_n recibidas de forma serie. Se han calculado unos retardos para la recepción de macro-secuencias formadas por 4 secuencias CSS de 4 bits y w_0 de 16 bits, utilizando una modulación *BPSK* con una portadora de 80 kHz. Al comprobar el retardo teórico (29) y el obtenido en simulación se observa que concuerdan (Fig. 53).

$$Retardo (seg) = \frac{1}{f_{sr}} \cdot (w_0 + N) = 25 \mu s \quad (29)$$

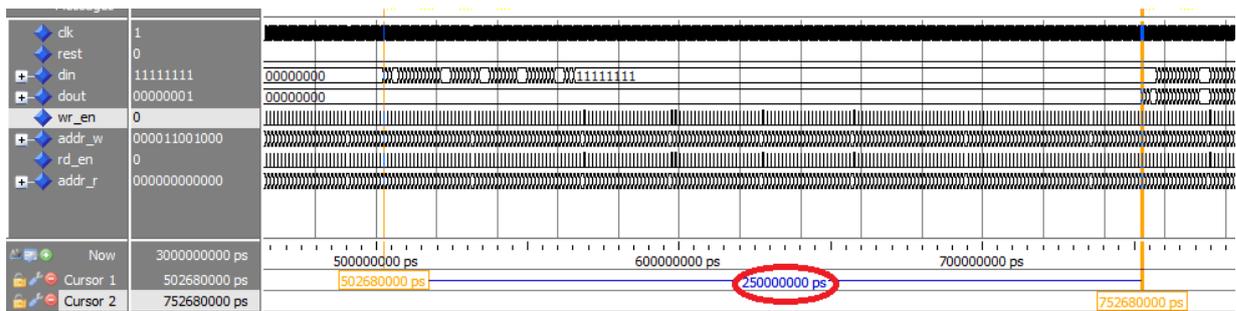


Fig. 53. Simulación del bloque de Retardos.

7.3.3 Simulaciones correlación

En la siguiente simulación (Fig. 54) es posible apreciar las señales de entrada al módulo **Correlador_CSS** tras el proceso de demodulación. Esta simulación se ha planteado para la recepción de macro-secuencias con las mismas características que las usadas en el apartado anterior. En este caso se ha simulado la adquisición de la macro-secuencia número 4. Las señales *Retardos_s1*, *Retardos_s2*, *Retardos_s3* y *Retardos_s4*, representan las salidas del módulo **Retardos** que atacan al módulo **Correlador_CSS**. Las señales *sal_i1*, *sal_i2*, *sal_i3* y

sal_{i4} , corresponden con las salidas de correlación ACF y CCF para cada uno de los cuatro conjuntos que forman las macro-secuencias.

Se observa como el pico de auto-correlación se obtiene cuando las 4 secuencias, que forman la macro-secuencia, se introducen simultáneamente en el correlador. Por otro lado, aparece un ruido producido por la cuantificación debido a que se trabaja con señales discretas. Este error se hace más evidente al perderse resolución tras la demodulación (de 12 a 8 bits).

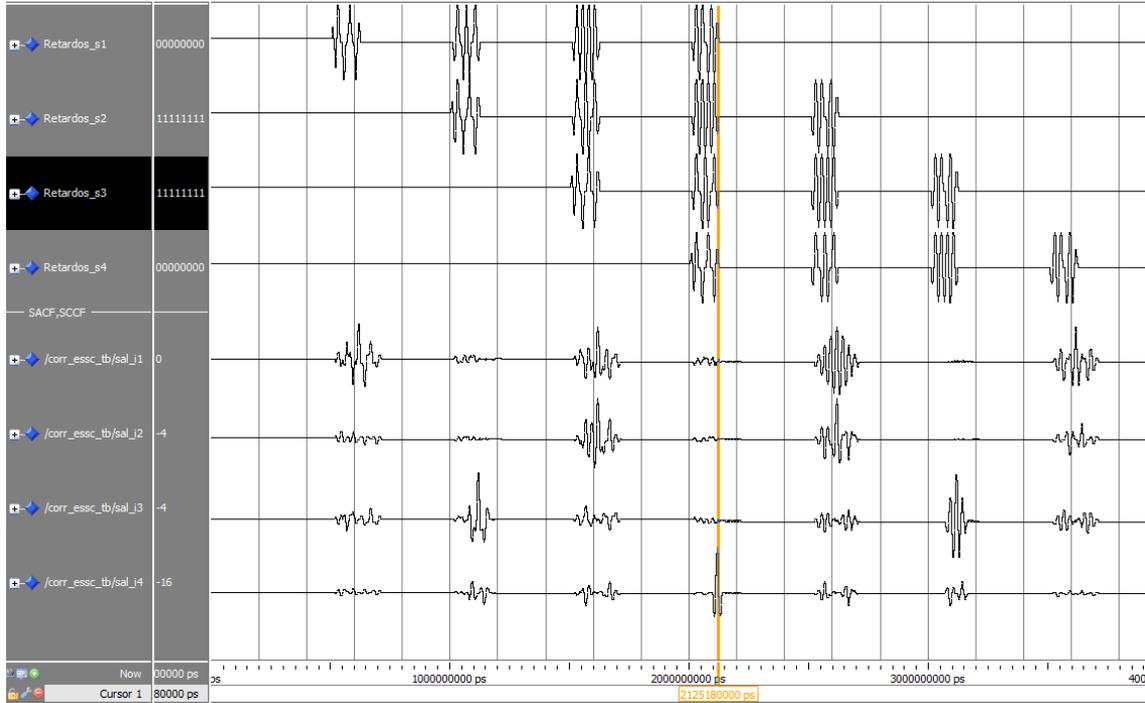


Fig. 54. Simulación de entradas al correlador CSS y las funciones CCF y ACF a su salida para la recepción de la macro-secuencia Ms_4 formada por 4 secuencias de 4 bits y una w_0 de 16 bits.

Este ruido también es posible apreciarlo en las simulaciones llevadas a cabo en *Matlab* tal y como se representa en (Fig. 55). Asimismo, al realizar la comparativa entre los resultados obtenidos tras la simulación hardware y los simulados mediante la herramienta *Matlab*, se aprecia que ambos resultan idénticos, lo que indica el correcto funcionamiento del proceso implementado en hardware.

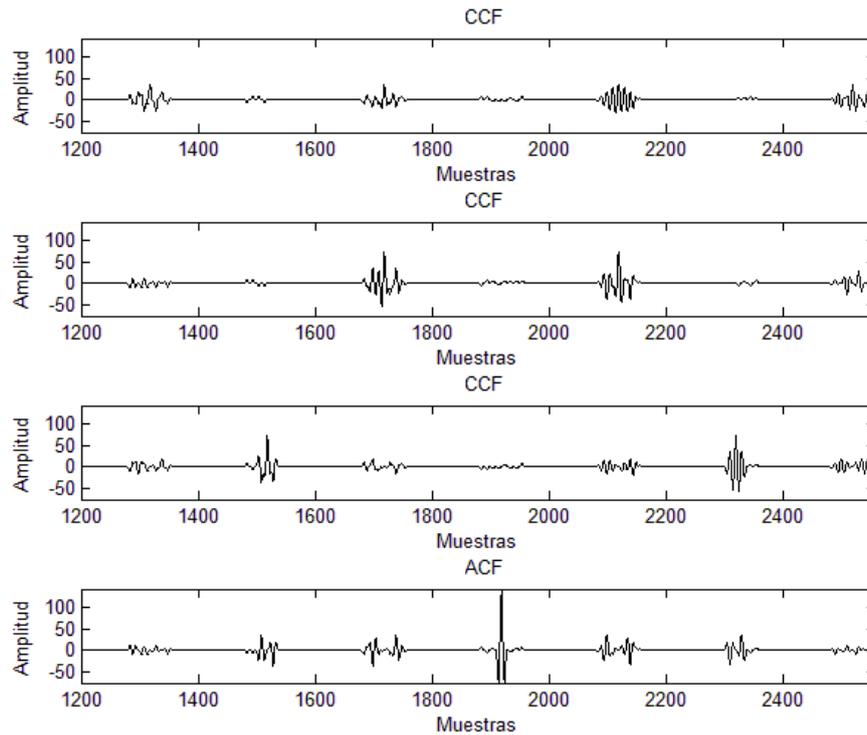


Fig. 55. Funciones CCF y ACF para la recepción de la macro-secuencia Ms_4 formada por 4 secuencias de 4 bits y una w_0 de 16 bits.

A continuación, y para conseguir la validación de este diseño se plantea el siguiente escenario real de simulación (Fig. 56). Este escenario está formado por dos placas de evaluación *Genesys Virtex5*, un módulo PmodDA2 y un módulo PmodAD1. En la primera placa de evaluación se carga el diseño de emisión para emitir únicamente la macro-secuencia Ms_1 . Por otro lado, en la segunda placa de evaluación se carga un diseño formado por los bloques de **ADC_ctrl**, **demodBPSK**, **Retardos** y **Correlador_CSS**.

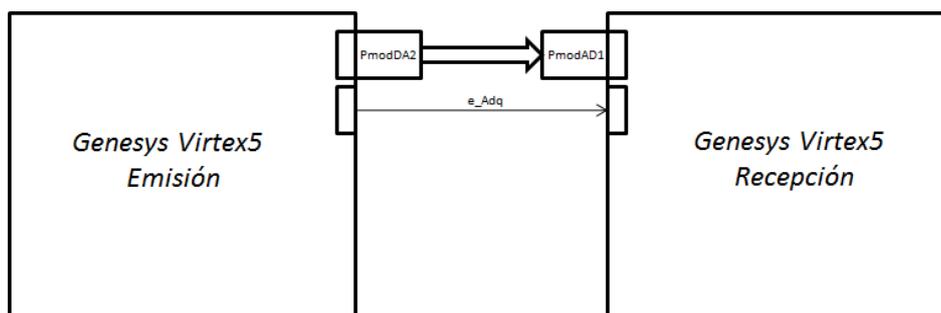


Fig. 56. Escenario real de simulación.

Utilizando la herramienta *ChipScope* se obtienen los siguientes resultados (véase la Fig. 57), al capturar la salida de correlación que representa la ACF de la macro-secuencia Ms_1 formada por 32 secuencias de 32 bits, w_0 de 354 bits, portadora $f_c=40\text{ kHz}$ y frecuencia de muestreo $f_{sr}=400\text{ kHz}$.

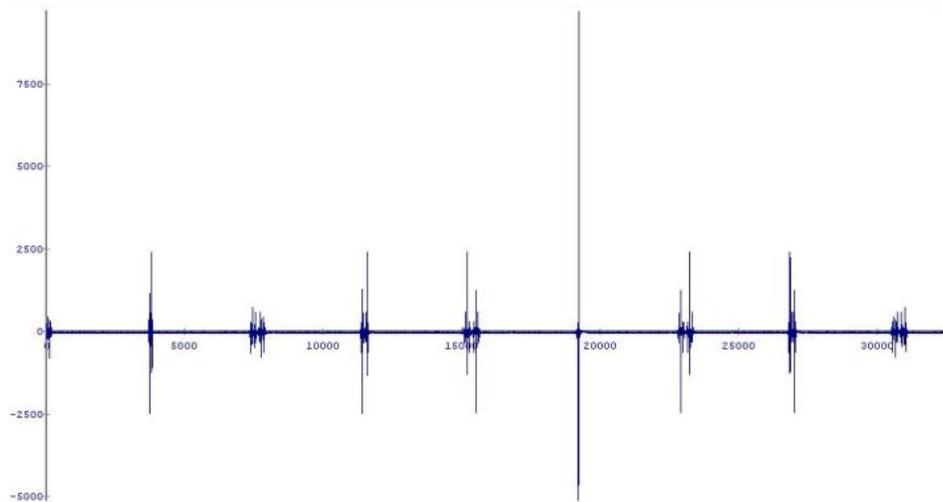


Fig. 57. segmento de ACF de la macro-secuencia M_{s1} formada por 32 secuencias de 32 bits, w_0 de 354 bits, portadora $f_c=40\text{ kHz}$ y frecuencia de muestreo $f_{sr}=400\text{ kHz}$ obtenida con *ChipScope*.

Asimismo, en la Fig. 58 se muestra el mismo segmento de la ACF para la misma macro secuencia realizada en *Matlab*.

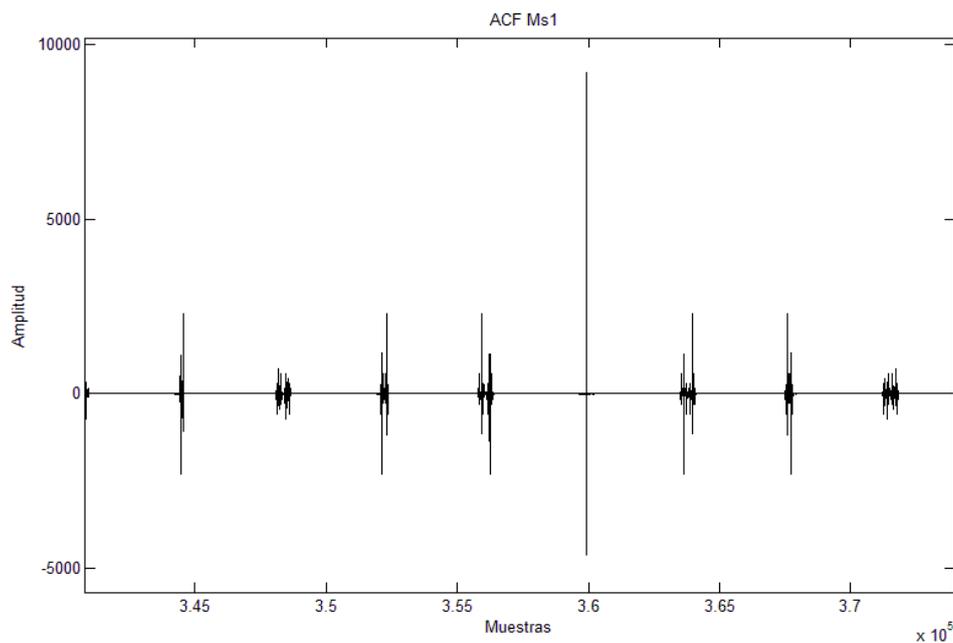


Fig. 58. segmento de ACF de la macro-secuencia M_{s1} formada por 32 secuencias de 32 bits, w_0 de 354 bits, portadora $f_c=40\text{ kHz}$ y frecuencia de muestreo $f_{sr}=400\text{ kHz}$ obtenida con *Matlab*.

La similitud entre los resultados representados en Fig. 57 y Fig. 58 permiten validar el diseño implementado sobre la FPGA en recepción, ya que denota que el proceso de adquisición y correlación funciona correctamente. Aunque ambas placas de evaluación son iguales, no comparten oscilador. Esto puede traducirse en que pequeñas discrepancias de frecuencia entre los osciladores ambas placas, lo que produce una pérdida de sincronismo entre emisor y receptor. Es decir, si los osciladores del emisor y del receptor son diferentes, se produce un error relativo entre las correspondientes frecuencias de muestreo (alguna de las frecuencias de

muestreo es distinta de la nominal, fijada en el diseño). Al tratarse de secuencias de longitud muy elevada, este efecto se hace evidente incluso con discrepancias entre osciladores pequeñas. Este error se evidencia en recepción, ya que se produce un ensanchamiento o contracción “virtual” del código adquirido, afectando a la secuenciación con la que las señales se introducen en el correlador.

La desincronización produce una distorsión en las funciones de correlación resultantes (tanto en la ACF como en la CCF), tal y como se muestra en la Fig. 59. En la Fig. 59, además de la función de correlación distorsionada (representada en color azul), también se muestra el flanco de subida de la señal de *enable* (representada de color rojo) que habilita la escritura en memoria externa.

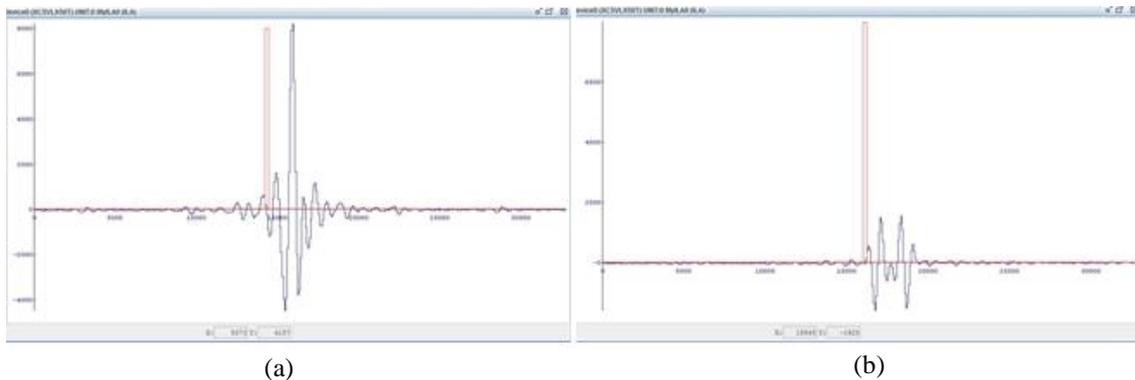


Fig. 59. (a) Pico de auto-correlación distorsionado con y flanco de subida de la señal de *enable* obtenido con *ChipScope*. (b) Correlación cruzada con distorsión y flanco de subida de la señal de *enable* obtenido con *ChipScope*.

7.3.4 Simulación NPI

Con el fin de comprobar el correcto funcionamiento de la comunicación NPI se ha llevado a cabo la simulación de una escritura en modo ráfaga de 8 word. El cronograma que describe el funcionamiento se refleja en la Fig. 60.

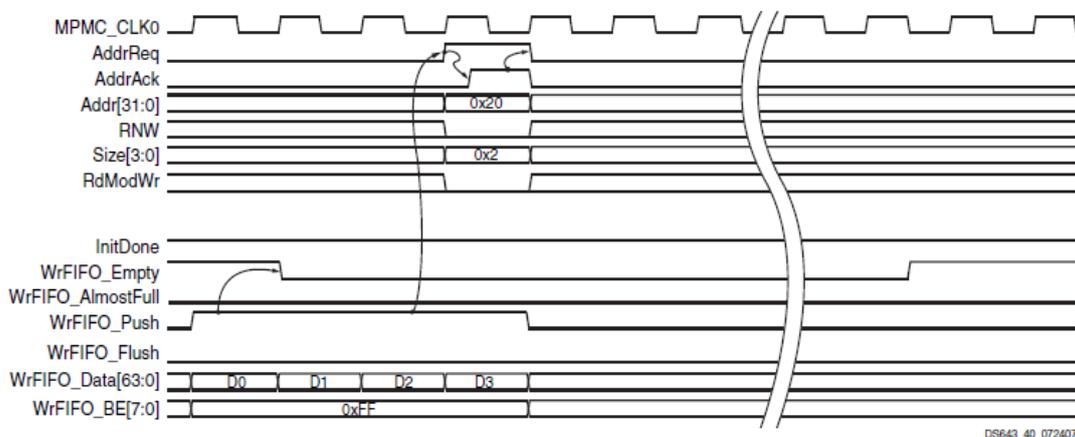


Fig. 60. Cronograma de protocolo NPI para escritura de 8 *word* en modo ráfaga [Xil11].

Para la simulación mediante la herramienta *Modelsim* se ha generado un modelo de simulación que emula la interacción entre el controlador *MPMC* y el bloque de comunicaciones **DMA_NPI**. En el documento [Xil11] la información que se suministra en cuanto a temporización es que la señal de reloj debe ser la del *MPMC* que en este caso es de 200 MHz. No se suministra información acerca del tiempo que transcurre hasta que los datos almacenados en la FIFO del controlador son volcados por completo, ni del tiempo mínimo y máximo que se tarda en obtener la señal de *Ack* del *MPMC*. Dado que el tiempo de volcado se considera despreciable con respecto a la frecuencia con la que se actualizan los datos a escribir en memoria externa, se ha tomado un valor, al azar, del doble de tiempo de la duración de la señal de *Push*. Asimismo, se ha interpretado que la señal de *Ack* es válida en el siguiente ciclo de reloj tras la activación de la señal de *Request*.

En la siguiente simulación (véase la Fig. 61) se muestran tres escrituras consecutivas en memoria externa. Cada una de estas escrituras se produce cada vez que la señal de clock-enable (*ce_chip*) se activa y a su vez la señal *enable* también se encuentra a nivel alto.

Se observa como al terminar el proceso de escritura, la señal *intr_ext*, que representa la activación de la señal de interrupción, se pone a nivel alto. Asimismo, la dirección del bus (*XIL_NPI_Addr*) vuelve a la dirección de base. Esta dirección de base es *X"80000000"* que apunta a la parte intermedia de la memoria DDR2. De esta manera, se tienen 128 Mbytes reservados para memoria de programa y 128Mbytes reservados para datos, de los cuales tan sólo se usan 450 kbytes

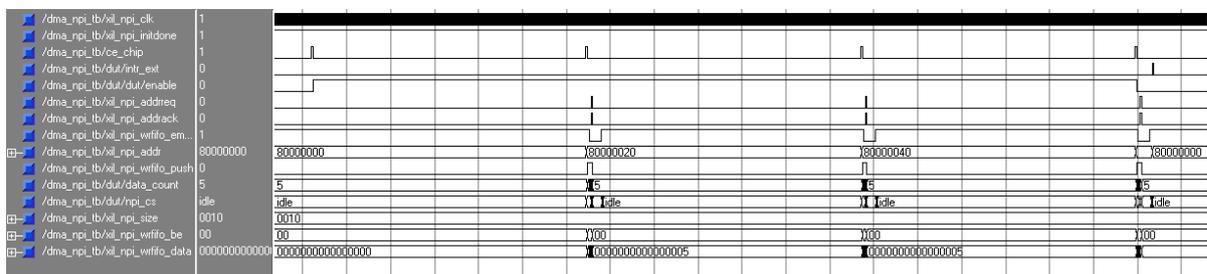


Fig. 61. Simulación del proceso completo de escritura en memoria externa.

En el siguiente cronograma se refleja una captura realizada con la herramienta *ChipScope* de *Xilinx*. En ella se comprueba el correcto funcionamiento simulado en la Fig. 62. De esta manera, las señales generadas por el módulo **NPI_DMA** se representan en color azul, mientras que las producidas por el controlador *MPMC* son representadas en color verde.

Implementación del bloque receptor

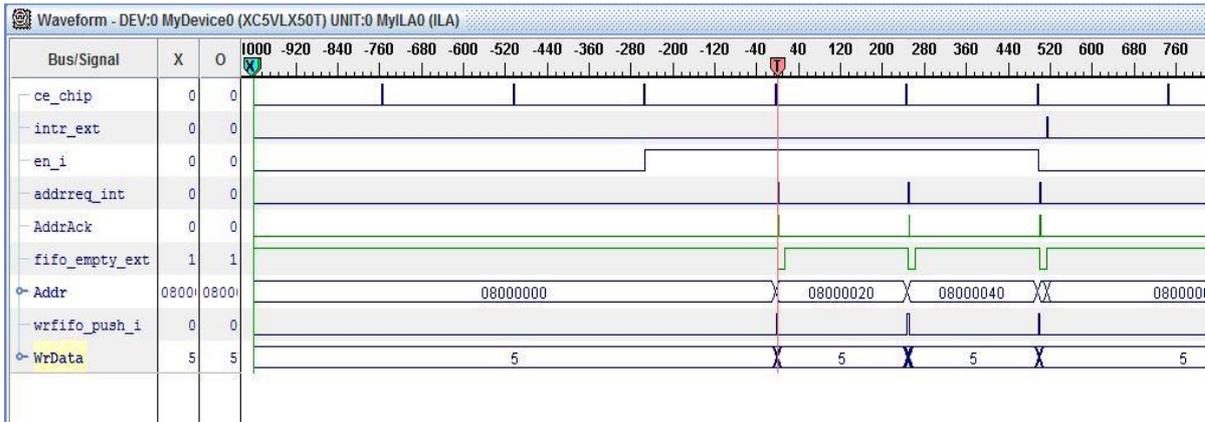


Fig. 62. Captura de *ChipScope* de proceso completo de escritura en memoria externa.

Las señales observadas en la Fig. 62 se corresponden con: *ce_chip* es la señal clock-enable del sistema; *intr_ext* representa la activación de la señal que interrumpe al procesador *Microblaze*; *en_i* corresponde a la señal de *enable* que habilita la escritura en memoria externa; *addrreq_int* representa a la señal *XIL_NPI_AddrReq* de solicitud de escritura; *AddrAck* es la señal *XIL_NPI_AddrAck* de respuesta a la solicitud de escritura; *fifo_empty_ext* corresponde con la señal *XIL_NPI_WrFIFO_Empty* que indica cuando la FIFO del *MPMC* deja de estar vacía. *Addr* corresponde con la dirección del bus *XIL_NPI_Addr*; *wrfifo_push_i* representa a la señal de push *XIL_NPI_WrFIFO_Push* y por último, *WrData* corresponde con *XIL_NPI_WrFIFO_Data* de escritura de datos en la memoria FIFO.

En el siguiente cronograma (véase la Fig. 63) se aprecia el mismo proceso focalizando sobre la escritura de una ráfaga en DDR2. En él se observa la escritura de los datos 1, 2, 3 y 4 junto con la secuenciación de los estados *npi_cs* correspondientes. El dato 5 es generado pero no almacenado en memoria.

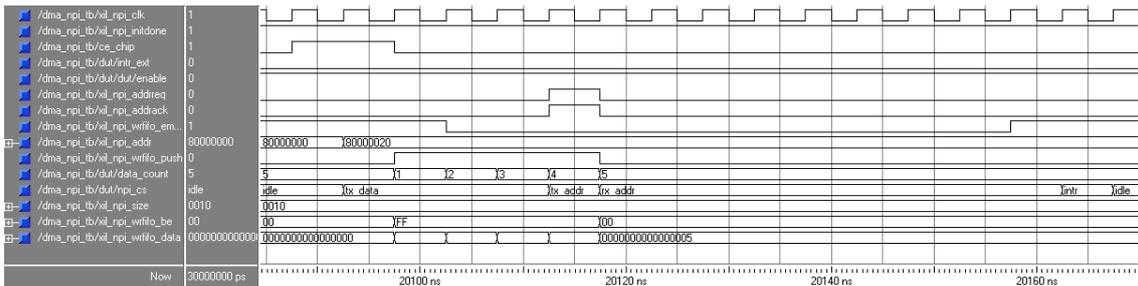


Fig. 63. Simulación del proceso de escritura de una ráfaga de 8 *word*.

En la Fig. 64 se observa la captura realizada con *ChipScope* del mismo proceso reflejado en la Fig. 63. Se aprecia como las señales reales siguen fielmente las obtenidas en simulación a partir del cronograma de la Fig. 60. No obstante, aparece una diferencia en el comportamiento de la señal de *fifo_empty_ext*, ya que su desactivación tarda más tiempo en producirse que el fijado en simulación. Este hecho no afecta al funcionamiento correcto del proceso.

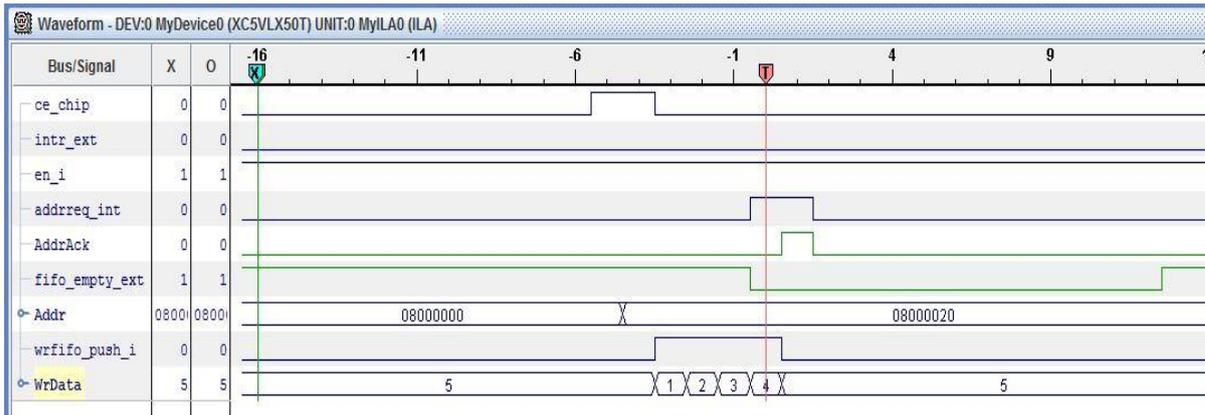


Fig. 64. Captura de ChipScope del proceso de escritura de una ráfaga de 8 word.

7.4 Recursos FPGA

La Tabla 19 recoge los resultados de implementación del módulo de procesamiento de bajo nivel y del *Microblaze*. El número de recursos consumidos en los bloques **Correlador_CSS** y **Modulo Retardos** es directamente proporcional al número códigos, a su longitud y al sobremuestreo O_f utilizado. Lo mismo ocurre con la frecuencia máxima de funcionamiento del sistema.

Tabla 19 Recursos consumidos por el módulo Receptor propuesto en la FPGA Xilinx Virtex5 LX50T.

Recursos Adquisición-Correlación/ Porcentaje de ocupación			
Bloque	Slices	BRAMs	DSPs
ADC_Ctrl	15 (0,21 %)	0 (0,00%)	0 (0,00%)
Demod. BPSK	68 (0,94%)	1 (0,83%)	1 (2,08%)
Modulo Retardos	475 (6,60%)	62 (51,67%)	0 (0,00%)
Correlador CSS	4945 (68,81%)	0 (0,00%)	0 (0,00%)
DMA_NPI	102 (1,67%)	0 (0,00%)	0 (0,00%)
Microblaze	1295 (17,99%)	52 (43,33%)	3 (6,25%)
Total	6900 (95,83%)	115 (95,83%)	4 (8,33%)

8 Resultados experimentales

8.1 Planteamiento

Las pruebas reales se han realizado con un primer prototipo del *array* EMFi, compuesto por 8 elementos, con unas dimensiones 0.26x4 cm², con una separación entre elementos (*gap*) de 1 mm y una frecuencia máxima de funcionamiento de 47 kHz; que cubre 8 sectores del espacio, comprendidos entre -52° y 60°. Se han emitido 8 códigos, en lugar de los 32 para los que está diseñado el sistema, para evitar redundancia en la información recibida. De esta manera, si se emitieran 32 códigos, la información que se recibiría por cada una de las líneas A-scan aparecería de forma redundada debido a la limitación, en cuanto a la resolución angular, de un *array* de ultrasonidos formado por 8 elementos. Asimismo, emitiendo en 8 sectores con 8 códigos se obtienen resultados similares a si se emitieran 32 códigos, en cuanto a resolución angular, pero las imágenes B-scan presentan un contraste mejor debido a que el canal se comparte entre menos códigos.

Por otro lado, en estas primeras pruebas experimentales, las tensiones máximas con las que se ha excitado el *array* de ultrasonidos son de 150 Vpp. Asimismo, debido a las características hardware del *array*, en cuanto a la separación entre elementos, la frecuencia de portadora se ha reducido a $f_c = 40$ kHz.

8.2 Escenario de pruebas

En la Fig. 65 se muestra el escenario base sobre el que se han realizado las pruebas experimentales.

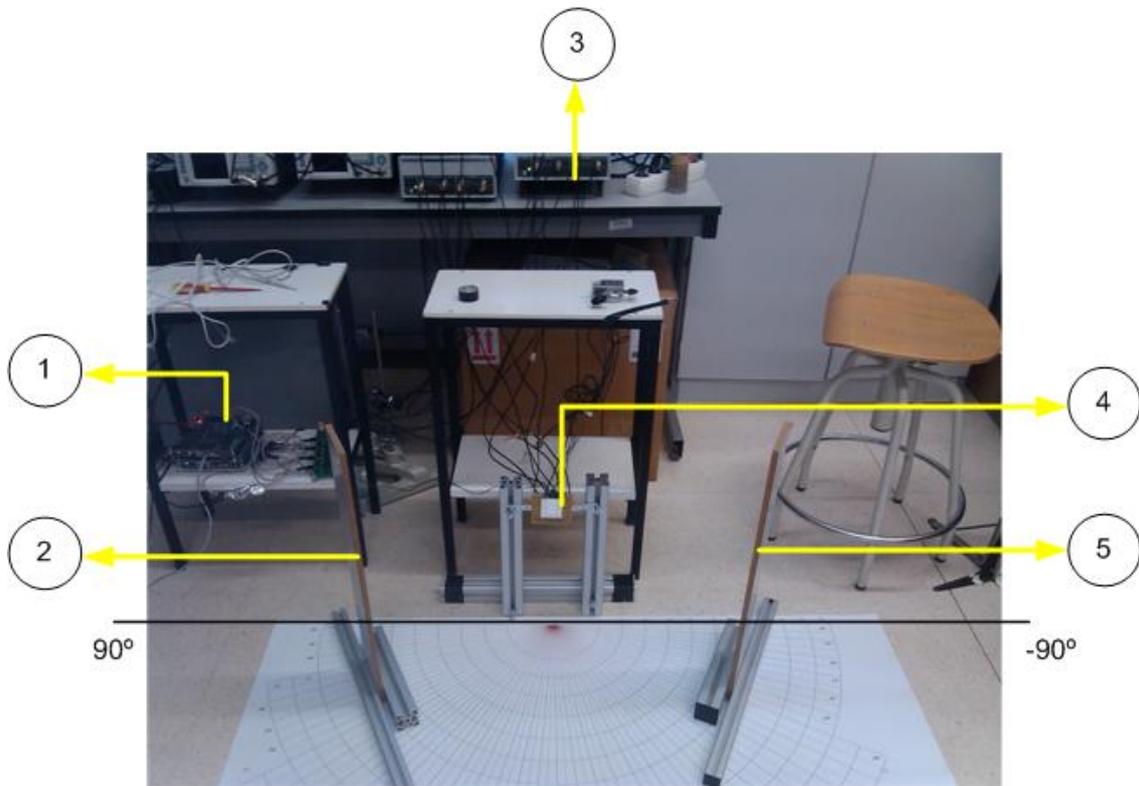


Fig. 65. Escenario base donde se han desarrollado las pruebas reales.

- 1) Emisor y receptor.
- 2) Reflector 2.
- 3) Amplificadores de potencia.
- 4) Array de ultrasonidos.
- 5) Reflector 1.

8.3 Aplicación de comunicaciones

En este trabajo, para la recepción de datos desde el PC se ha implementado una aplicación software desarrollada con la herramienta *LabWindows*. Su finalidad es la de permitir la conexión vía Ethernet mediante un protocolo TCP/IP a 100 Mb/s con la plataforma hardware de recepción. Para la obtención de los datos procesados en bajo nivel.

El aspecto que presenta la aplicación es el siguiente (Fig. 66):

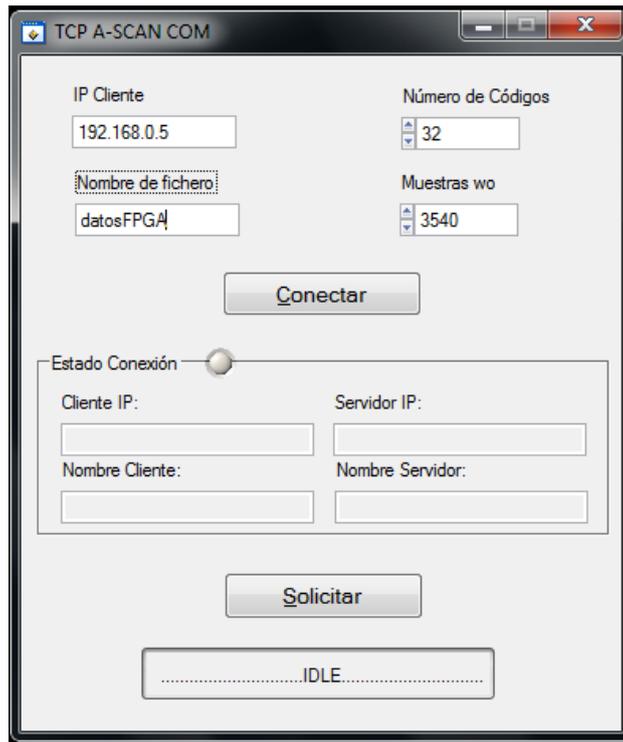


Fig. 66. Interfaz gráfica de la aplicación de comunicaciones entre el PC y la FPGA de recepción.

En primer lugar se debe configurar la conexión rellenando con la dirección IP del dispositivo FPGA receptor, así como con el nombre del fichero en el que se desean almacenar los datos recibidos. Por otro lado, se debe especificar el tamaño de la trama de datos que va a ser recibida. Esto queda parametrizado mediante las casillas de “*Muestras wo*” y “*Número de Códigos*”. En este trabajo, ambos parámetros no cambian, pues el número de códigos recibidos es siempre el mismo y el número de muestras de la IFW tampoco varía. Independientemente del número de códigos emitidos, al PC siempre se le envían los datos de la *wo* de las salidas del módulo **Correlador CSS**, que para las macro-secuencias utilizadas Ms_n corresponden a 32 salidas.

Tras la configuración se debe pulsar el botón “*Conectar*”. Si la conexión se lleva a cabo correctamente el led de *Estado de Conexión* se encenderá y aparecerán las correspondientes IPs y nombres del cliente (PC) y del servidor (FPGA de recepción).

Por último se pulsa el botón “*Solicitar*” cada vez que se quieran obtener datos de las líneas A-scan. Mientras se esté produciendo una obtención de datos de la FPGA, el cuadro de texto que aparece en la parte inferior al botón cambiara su contenido a “*ADQUIRIENDO*”. Al finalizar la importación de datos el contenido de dicho cuadro de texto cambiara de nuevo a “*IDLE*”.

8.4 Resolución Angular

El objetivo de esta prueba es el de comprobar a qué distancia angular pueden situarse dos reflectores entre sí, para dejar de detectarse como dos reflectores independientes y pasar a detectarse como uno único. Asimismo, también se pretende comprobar cuál es el ángulo a partir del cual se comienza a detectar la presencia de cada uno de ellos.

Para conseguir esto se han colocado dos reflectores de madera y ancho 10 cm a una distancia del eje del *array* de 40 cm (aprox.) y se ha variado su posición angular desde el comienzo de la detección de los reflectores hasta la detección de ambos reflectores como uno único.

Para la representación de las líneas A-scan se utiliza la herramienta *Matlab*. Mediante el uso de este software se calcula la envolvente (utilizando el módulo de la transformada de Hilbert) del valor absoluto de las señales recibidas.

El criterio seguido para considerar que el pico de auto-correlación del eco recibido corresponde a un reflector, es el de considerar cada pico válidos hasta -3 dB sobre el valor de correlación 700 (obtenido a partir de los resultados mostrados en el apartado 8.5). La detección del eco del segundo reflector comienza a producirse en una posición angular de 85° (Fig. 67). Para esta colocación, la línea A-scan correspondiente al código 8 capta el eco cuyo pico de auto-correlación supera los -3 dB sobre el valor de pico 700. Debe tenerse en cuenta que al tratarse de reflectores de 10 cm situados a 40 cm, se tiene que cada uno de ellos cubre el equivalente a $\pm \tan^{-1} \frac{5}{40} \cong \pm 7^\circ$ con respecto a la posición angular de origen.

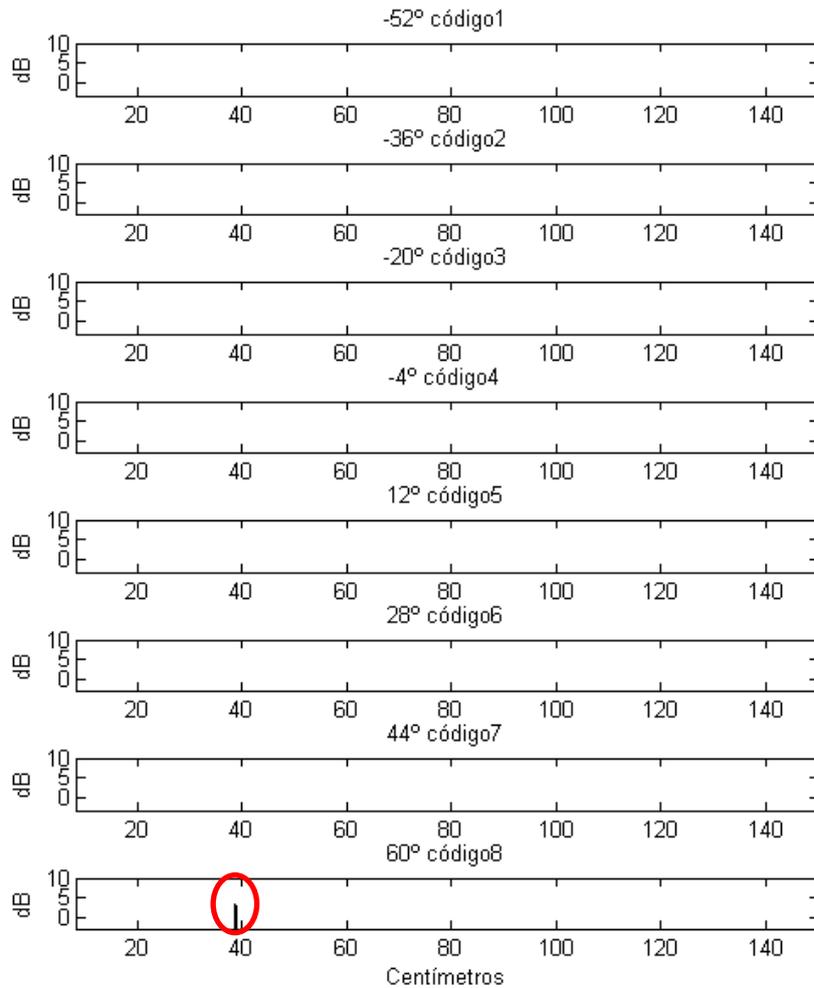


Fig. 67. Líneas A-scan con: Reflector 1 a 40 cm y -85°; Reflector 2 a 40 cm y 85°.

La detección del eco del primer reflector comienza a producirse en una posición angular de -75° (véase la Fig. 68).

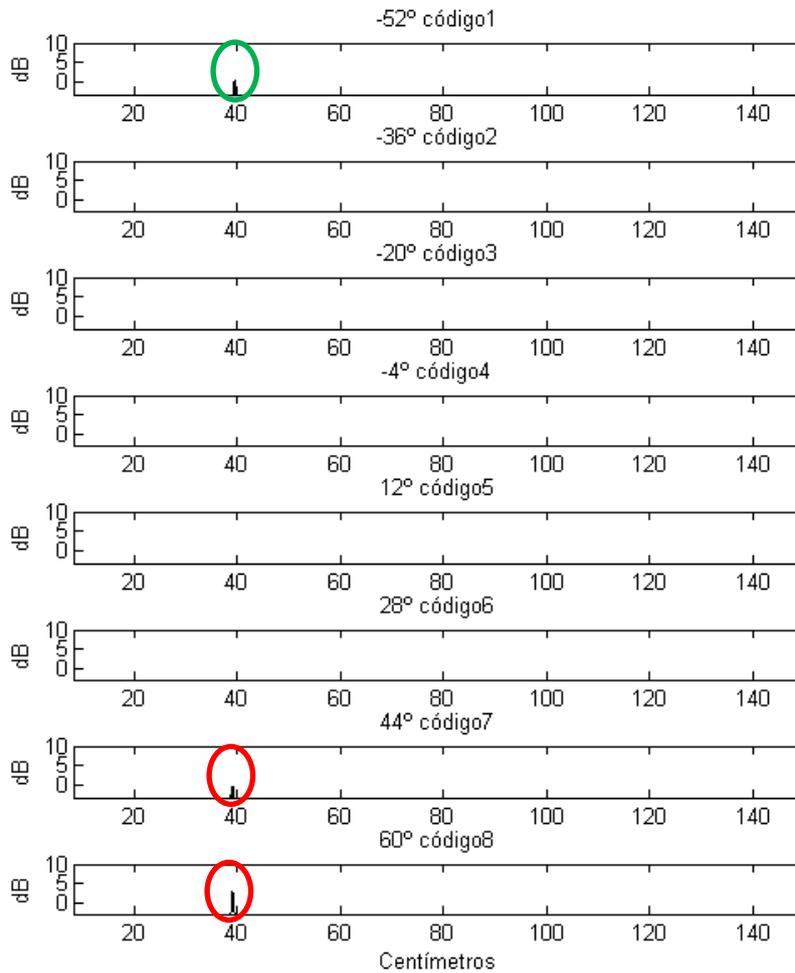


Fig. 68. Líneas A-scan con: Reflector 1 a 40 cm y -75° ; Reflector 2 a 40 cm y 75° .

Como conclusión a estas pruebas cabe destacar que el ángulo de detección máximo es mayor al esperado, pues teóricamente la resolución angular $\Delta\theta$ para un *array* de 8 elementos, reflejada en Tabla 1, es de 14.3239° lo que significa $\pm 7.1620^\circ$ con respecto al centro del haz deflectado. Por lo tanto, se puede afirmar que el rango de detección se amplía $-75^\circ - (-7^\circ - 7.1620^\circ - 52^\circ) \cong -7^\circ$ aprox. para ángulos negativos (teniendo en cuenta el ángulo cubierto por el reflector $75^\circ \pm 7^\circ$) con respecto al valor esperado, y $85^\circ - (7^\circ + 7.1620^\circ + 60^\circ) \cong 11^\circ$ para ángulos positivos (teniendo en cuenta el ángulo cubierto por el reflector $85^\circ \pm 7^\circ$) con respecto al valor esperado.

En la siguiente prueba (Fig. 69) se refleja la posición justo antes de la cual ambos reflectores pasan a detectarse como uno único. Esto sucede al colocar ambos reflectores en una posición angular de 25° y -25° respectivamente.

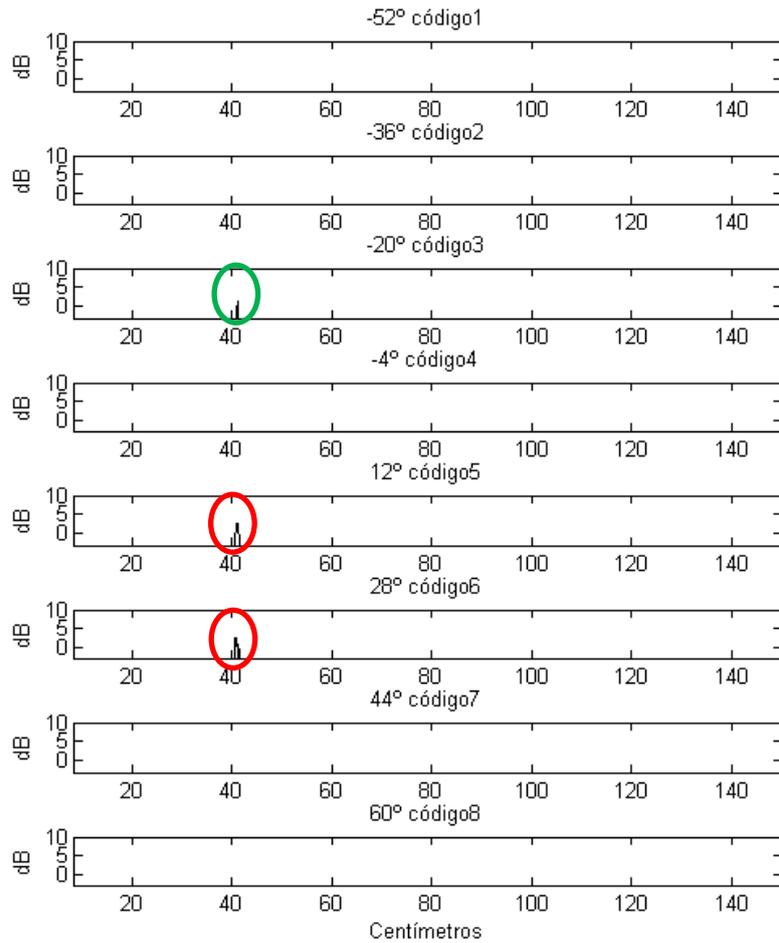


Fig. 69. Líneas A-scan con: Reflector 1 a 40 cm y -25° ; Reflector 2 a 40 cm y 25° .

De la prueba anterior se puede concluir que las líneas A-scan recibidas son las esperadas en función de las posiciones de los reflectores. En el caso del reflector 1 (en color verde), el ángulo -25° se encuentra aproximadamente entre los ángulos de deflexión de los haz ultrasónicos de los códigos 3 y 2 (-20° y 36° respectivamente). En el caso del reflector 2, el valor 25° se encuentra aproximadamente entre los ángulos de deflexión de los haz ultrasónicos de los códigos 5 y 6 (-12° y 28° respectivamente). Por último, se muestra la prueba realizada en la cual dos reflectores pasan a detectarse como uno solo (Fig. 70). Esto ocurre con una colocación de 15° y -15° respectivamente.

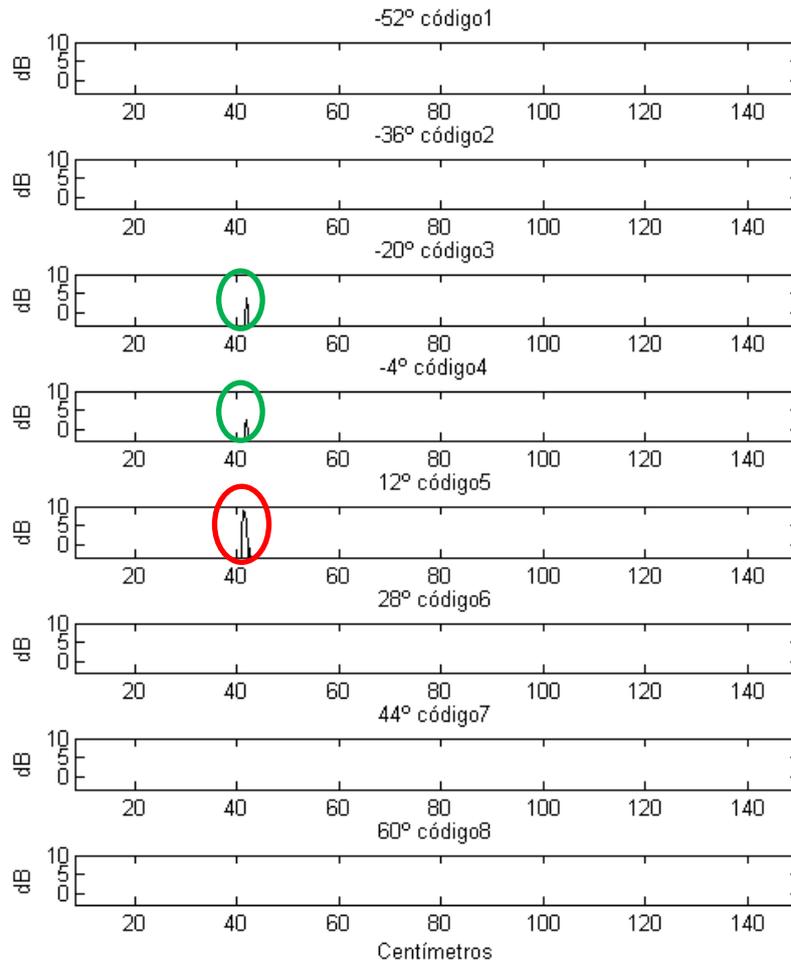


Fig. 70. Líneas A-scan con: Reflector 1 a 40 cm y -15° ; Reflector 2 a 40 cm y 15° .

Como conclusión de esta última prueba cabría decir que la resolución angular es la esperada ya que el valor -15° se encuentra aproximadamente entre los ángulos de deflexión de los haz ultrasónicos de los códigos 3 y 4 y el valor 15° se encuentra aproximadamente en el ángulo central de deflexión del haz ultrasónico del código 5. Por lo tanto, llegados a este punto, sería imposible distinguir si se trata de uno o dos reflectores los que han sido detectados.

8.5 Profundidad de exploración

Con estas pruebas se pretende comprobar cuál es la profundidad de exploración máxima alcanzable aplicando al *array* de ultrasonidos una tensión máxima de 150 Vpp, es decir, comprobar hasta qué distancia se sigue detectando un reflector. Para llevar a cabo esta prueba se ha partido de la colocación de un reflector de madera de 10 cm de ancho (similar a los utilizados en el apartado anterior (8.4)) a una distancia de 50 cm y 0° con respecto al eje del *array* de ultrasonidos y se ha ido desplazando hasta el punto en el que se ha dejado de detectar el reflector.

El criterio que se ha seguido para determinar la detección de un reflectores el de considerar la amplitud del valor absoluto del pico de auto-correlación recibido 3 dBs superior al del ruido de fondo del sistema.

La representación de estos resultados se refleja en la siguiente figura (véase la Fig. 71) que muestra las imágenes B-scan formadas por las líneas A-scan recibidas, con un contraste de -3dB. Es decir, se representa, sobre el fondo de la imagen, cualquier valor que se encuentre a menos de -3 dB del máximo valor recibido, en este caso el pico de auto-correlación correspondiente al eco del reflector.

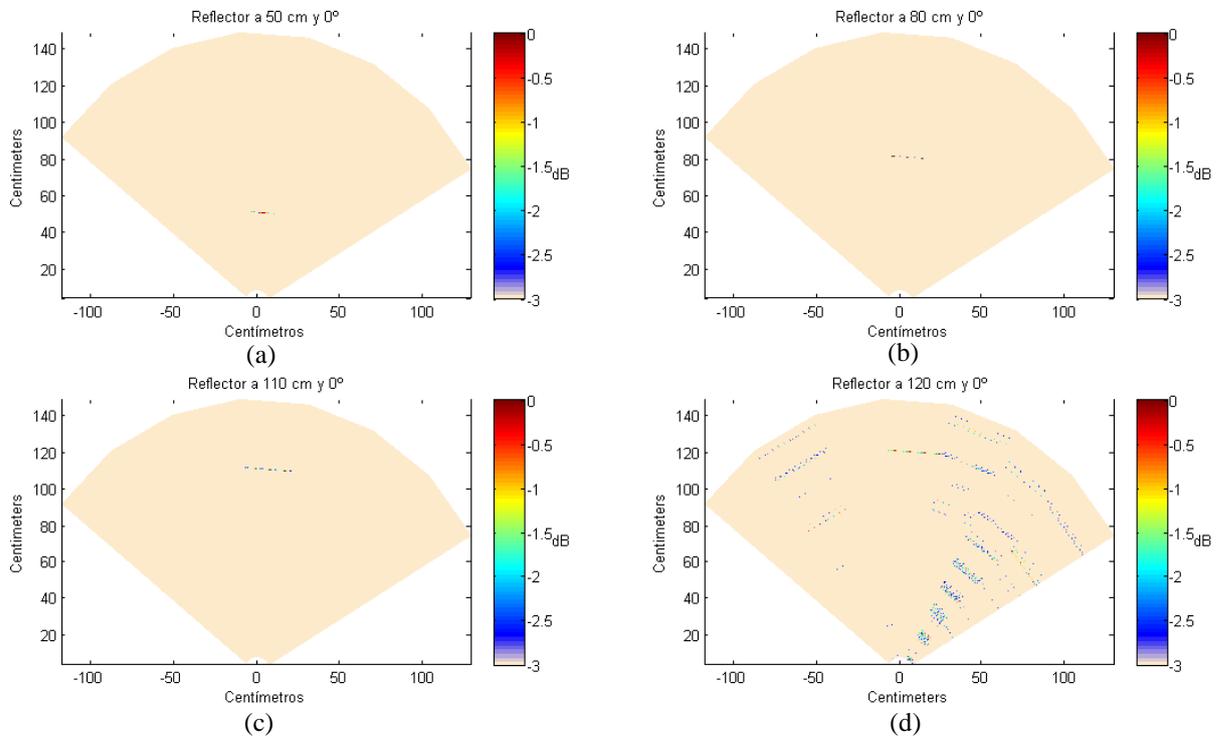


Fig. 71. Imagen B-scan con Reflector a 50 cm y 0°, contraste de -3dB (a). Imagen B-scan con Reflector a 80 cm y 0°, contraste de -3dB (b). Imagen B-scan con Reflector a 110 cm y 0°, contraste de -3dB (c). Imagen B-scan con Reflector a 120 cm y 0°, contraste de -3dB (d).

En la Fig. 71 es posible observar como a partir de 110 cm el valor del ruido de fondo del sistema supera los -3dB con respecto al valor de la señal del eco recibido debido a la presencia de un reflector. Por lo tanto, se puede afirmar que el alcance máximo del *array* ultrasónico al aplicarle tensiones máximas de 150 Vpp es de 1.1 m. El pico de auto-correlación detectado a 1.1 m es de 700.

8.6 Resolución Axial

Con esta prueba se pretende determinar cuál es la distancia mínima a la que se pueden situar dos objetos, situados en sectores angulares contiguos, antes de pasar a detectarse como un único objeto. El criterio seguido para considerar que ambos objetos son interpretados como uno único, es el de que el valle entre los picos de auto-correlación, correspondiente a cada uno de los objetos, sea mayor a -3dB, con respecto al pico de auto-correlación de menor valor.

La resolución axial ideal es de $\frac{v}{f_c} = \frac{343 \text{ m/s}}{40 \text{ kHz}} = 0.86 \text{ cm}$; y la precisión con la que se obtiene la posición real es de $\frac{v}{f_{sr}} = \frac{343 \text{ m/s}}{400 \text{ kHz}} = 0.09 \text{ cm}$.

Para llevar a cabo esta prueba se ha partido de la colocación de un reflector de madera de 10 cm de ancho (similar a los utilizados en los apartados anteriores (8.4 y 8.5)) a una distancia de 50 cm y -4° con respecto al eje del *array* de ultrasonidos y otro de similares características a una distancia de 51.5 cm y 12° .

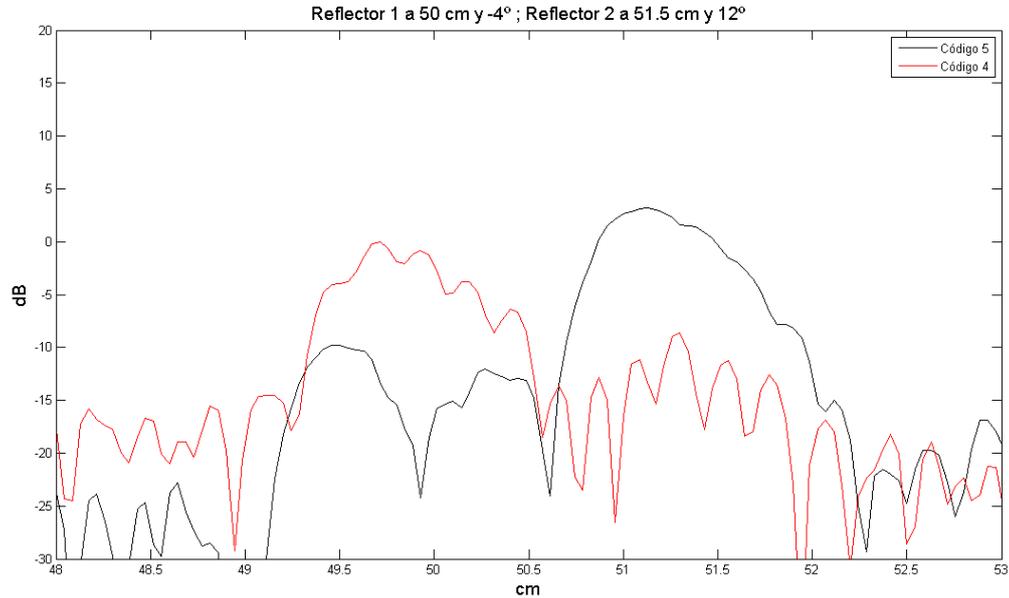


Fig. 72. Líneas A-scan correspondientes al código 5 y código 4 representadas en dB tomando como 0 dB el máximo del pico de auto-correlación de menor amplitud. Reflector 1: 50 cm y -4° ; Reflector 2: 51.5 cm y 12° .

El pico máximo de auto-correlación correspondiente al primer reflector es de 49.71 cm y el del segundo es de 51.13 cm. Este error se debe a la imprecisión en la colocación de los objetos. De esta manera, la posición relativa medida entre ambos reflectores es de 1.42 cm. Asimismo, para esta distancia se observa cómo en la Fig. 72 ambos objetos son perfectamente distinguibles.

En la Fig. 73 se refleja la captura realizada para dos reflectores situados más próximos entre sí: el primer reflector a 50 cm y -4° con respecto al eje del *array* de ultrasonidos y el segundo reflector a una distancia de 50.5 cm y 12° .

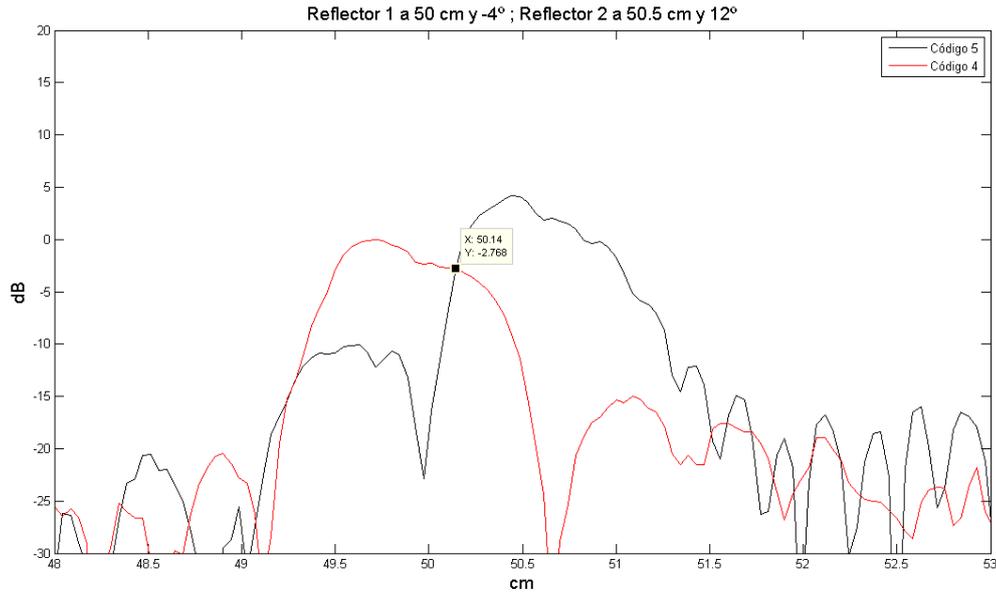


Fig. 73. Líneas A-scan correspondientes al código 5 y código 4 representadas en dB tomando como 0 dB el máximo del pico de auto-correlación de menor amplitud. Reflector 1: 50 cm y -4° ; Reflector 2: 50.5 cm y 12° .

El pico máximo de auto-correlación correspondiente al primer reflector es de 49.71 cm y el del segundo es de 50.44 cm. Este error se debe a la imprecisión en la colocación de los objetos. De esta manera, la posición relativa medida entre ambos reflectores es de 0.73 cm. Asimismo, para esta distancia se observa cómo, en la Fig. 73, ambos objetos dejarían de detectarse como uno único ya que el valle existente entre ambos picos de auto-correlación es mayor a -3 dB (-2.768 dB).

Como conclusión se obtiene que el valor de resolución axial es similar al obtenido de forma teórica y es de aproximadamente 1 cm. No obstante, este resultado no se cumple siempre y puede empeorar en función de la aparición de picos de auto-correlación secundarios, muy próximos entre sí, debidos al efecto de multicamino al recibir el eco de un mismo reflector.

8.7 Escenarios compuestos

En esta prueba se pretende mostrar las imágenes B-scan formadas a partir de distintos escenarios compuestos por dos reflectores de distintas características con un contraste de -5 dB.

En primer lugar, se muestra la imagen B-scan (Fig. 74) obtenida para dos reflectores de las mismas características que los usados en el apartado 8.4. El reflector 1 se encuentra a una distancia de 85 cm del eje del *array* de ultrasonidos y en una posición angular con respecto a éste de 0° . El reflector 2 se encuentra a una distancia de 35 cm y una posición angular de 40° .

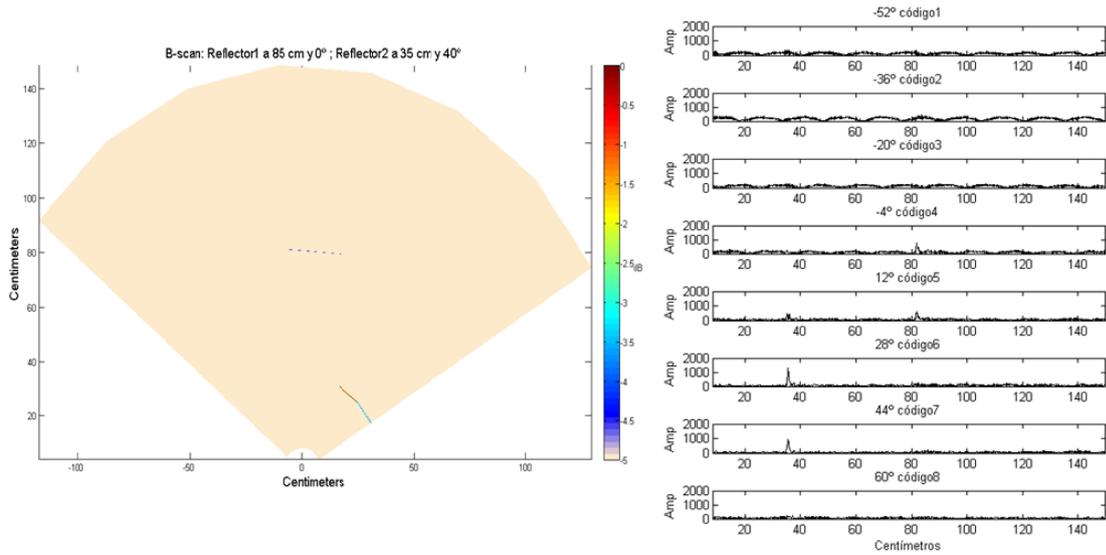


Fig. 74. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 85 cm, 0°, de madera y ancho de 10 cm; Reflector 2 a 35 cm, 40°, de madera y ancho de 10 cm.

El siguiente ensayo (Fig. 75) se ha realizado para una disposición con dos reflectores metálicos de 6 cm de ancho: uno (reflector 1) a 50 cm y desfasado -40° respecto al origen; y otro (reflector 2) a 30 cm y 20° .

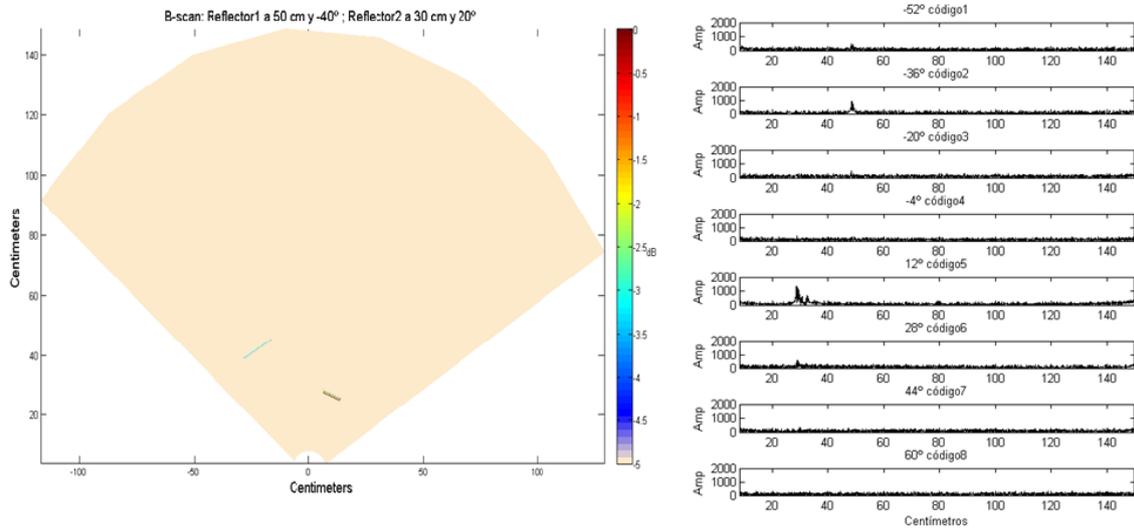


Fig. 75. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 50 cm, -40° , de metal y ancho de 6 cm; Reflector 2 a 30 cm, 20° , de metal y ancho de 6 cm.

El último ensayo (Fig. 76) se ha realizado para una disposición con dos reflectores metálicos de 6 cm de ancho: uno (reflector 1) a 30 cm y desfasado -40° respecto al origen; y otro (reflector 2) a 35 cm y 20° .

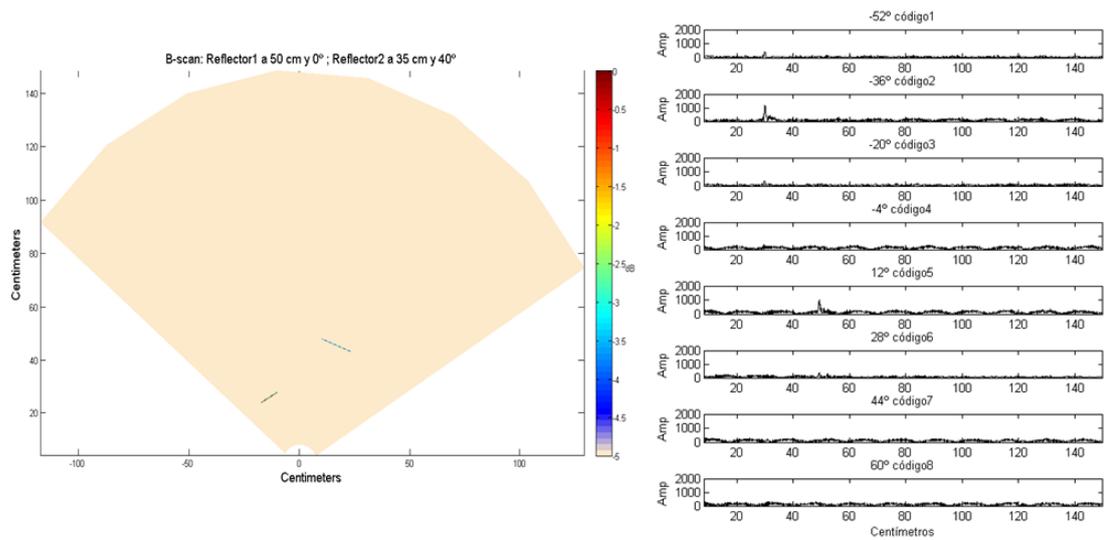


Fig. 76. Imagen B-scan y líneas A-scan asociadas para un escenario de: Reflector 1 a 30 cm, -40° , de metal y ancho de 6 cm; Reflector 2 a 50 cm, 20° , de metal y ancho de 6 cm.

Con estas pruebas queda comprobado que para reflectores de distintos materiales y tamaños se siguen obteniendo resultados de calidad suficientes como para construir imágenes de ultrasonidos B-scan.

9 Conclusiones y trabajos futuros

9.1 Conclusiones

Se ha presentado en este trabajo la implementación hardware de un sistema *Phased Array* ultrasónico para aire, que permite el barrido simultáneo en todas las direcciones. La implementación propuesta aborda el procesamiento en tiempo real, tanto en emisión como en recepción, de las señales ultrasónicas; así como el envío de los resultados de correlación a un ordenador personal para la formación de la imagen. El uso de técnicas CDMA junto con *Phased Array* permite aumentar la tasa de generación de imágenes, así como su calidad.

En este proyecto se han completado las siguientes tareas:

- Implementación hardware del módulo de emisión: se ha llevado a cabo el diseño sobre una plataforma de hardware reconfigurable de la emisión mediante técnicas *Phased Array* de macro-secuencias de CSS con ventana libre de interferencias IFW de 709 bits.
- Electrónica de acondicionamiento: se ha realizado el diseño y ruteado del prototipo de placa PCB necesario para la adecuación de niveles de tensión entre la plataforma de hardware reconfigurable y el array US.
- Implementación hardware del módulo de recepción: sobre una plataforma FPGA se ha desarrollado el proceso de adquisición y pre-procesado de los ecos ultrasónicos recibidos, así como la transmisión de los datos pre-procesados a un PC personal, a través de una comunicación basada en protocolo TCP/IP. Para ello, se ha diseñado un SoC sobre una plataforma de hardware reconfigurable.
- Software de comunicación y procesado: se ha llevado a cabo una aplicación que permite la comunicación con la plataforma hardware de recepción y controla la solicitud de datos a ésta. Asimismo, se han realizado los *scripts* necesarios para la representación de las líneas A-scan e imágenes B-scan.

Las pruebas realizadas, con un prototipo inicial de *array* US formado por 8 elementos, y llevando a cabo la exploración de 8 sectores del espacio, han permitido validar el diseño realizado.

9.2 Trabajos Futuros

Los trabajos futuros que se plantean son:

- Con un prototipo más avanzado, se pretenden realizar emisiones reales emitiendo los 32 códigos. Esto implica mejorar la ganancia del canal de emisión. Asimismo, se pretende trabajar con *arrays* de ultrasonidos de mayor longitud, con el fin de lograr mayor precisión al deflectar.

- Uso de otros códigos con mayor ganancia de proceso, como los LS o GPC que podrían mejorar las prestaciones del sistema, aunque implicarían el empleo de plataformas hardware tipo FPGA de mayor capacidad, así como la unificación de los bloques de emisión y recepción en un único dispositivo para evitar problemas de sincronización.
- Cambio del protocolo de comunicación entre el PC y la plataforma hardware a UDP, con el fin de reducir los tiempos de transmisión de datos.
- Desarrollo de un software que permita la representación de imágenes B-scan en tiempo real.

9.3 Publicaciones derivadas

Publicaciones derivadas de este trabajo:

- [GPH⁺13A] Rodrigo García, M. Carmen Pérez, Álvaro Hernández, F. Manuel Sánchez, José M. Castilla, Cristina Diego, "Arquitectura Eficiente para el Procesamiento de un *Array* Ultrasónico basado en Técnicas de Codificación" XX Seminario Anual de Automática, Electrónica Industrial e Instrumentación, ISBN: 978-84-15302-60-5, Julio 2013.
- [GPH⁺13B] Rodrigo García, M. Carmen Pérez, Álvaro Hernández, F. Manuel Sánchez, José M. Castilla, Cristina Diego, "Efficient Architecture for Ultrasonic *Array* Processing based on Encoding Techniques" International Conference on Indoor Positioning and Indoor Navigation, ISBN: 978-1-4673-1954-6, Octubre 2013.

10 Bibliografía

- [AD14] ANALOG DEVICES, “AD7476A/AD7477A/AD7478A datasheet”, 07/04/2014.
- [Cad14] CadSoft, “www.cadsoftusa.com”, consultada en Mayo 2014.
- [Che07] H. Chen, “Next Generation CDMA Technologies,” John Wiley & Son, Ltd, West Sussex PO19 8SQ, Inglaterra, 2007.
- [CK08] J. M. Carrera y A. Kurjak, “Ecografía en Diagnóstico Prenatal”, Masson, Barcelona, 2008.
- [CY06] H.-H. Chen; Y.-C. Yeh et al., “Generalized pairwise complementary codes with set-wise uniform interference-free windows,” IEEE Journal on Selected Areas in Communications, 2006, vol. 24, no.1, pp. 65-74.
- [DHJ⁺11] C. Diego, A. Hernández, A. Jiménez, F. J. Álvarez, R. Sanz. “Ultrasonic *array* for obstacle detection based on CDMA with Kasami codes,” Sensors, Vol. 11, 2011.
- [Dig06] DIGILENT Inc, “Digilent PmodDA2 Digital To Analog Module Converter Board Reference Manual”, 25/09/2006.
- [Dig11] DIGILENT Inc, “Digilent PmodAD1 Analog To Digital Module Converter Board Reference Manual”, 06/12/2011.
- [Dig11] DIGILENT Inc, “Digilent PmodAD1 Analog To Digital Module Converter Board Reference Manual”, 6/12/2011.
- [Dig12] DIGILENT Inc, “Genesys Board Reference Manual”, 28/02/2012.
- [Dig14] Digilent Inc, “www.digilentinc.com”, consultada en Mayo 2014.
- [Emf03] Emfit Ltd, “Emfit Film Specifications”. Marzo 2003.
- [P09] M^a Carmen Pérez Rubio, Tesis Doctoral, “Generación y correlación eficiente de códigos binarios derivados de conjuntos de secuencias complementarias para sistemas ultrasónicos”. 2009.
- [PJS09] D. T. Pham, Z. Ji y A. Soroka, “Ultrasonic distance scanning techniques for mobile robots”, International Journal of Computer Aided Engineering and Technology, Vol. 1, no. 2, pp. 209-224, 2009.
- [PLK00] M. Paaanen, J. Lekkala, and K. Kirjavainen, “ElectroMechanical Film EMFi_A New Multipurpose Electret Material”, Sensors and Actuators A, Vol. 84, pp. 95_102, 2000.
- [PNI⁺08] M. Parrilla, P. Nevado, A. Ibañez, J. Camacho, J. Brizuela y C. Fritsch, “Ultrasonic imaging of solid railway wheels”, Proceedings of the IEEE Ultrasonics Symposium, Beijing, China, 2008.
- [PSU⁺12] M. C. Pérez, R. Sanz, J. Ureña, A. Hernández, C. De Marziani, F. J. Álvarez, “Correlator implementation for Orthogonal CSS used in an ultrasonic LPS,” IEEE Sensors Journal, vol. 12, no. 9, pp. 2807-2816, 2012.

Bibliografía

- [PSUJ12] M. C. Perez, R. Sanz, J. Ureña, A. Hernández, C. de Marzani and F.J. Álvarez, "Correlator Implementation for Orthogonal CSS used in an Ultrasonic LPS", IEEE Sensors Journal, February 2012.
- [PUH*] M. C. Perez, J. Ureña, A. Hernández, C. De Marziani, J. J. García, and A. Jimenez, "Optimized Correlator for LS Codes-Based CDMA Systems", IEEE Communications Letters, Vol. 15, No. 2, pp. 223–225, 2010.
- [PUH09] M. C. Pérez, J. Ureña, A. Hernández, A. Jiménez, F. J. Álvarez, C. De Marziani, "Comparativa de las prestaciones de diferentes códigos en un sistema ultrasónico con detección asíncrona y acceso compartido al medio," XVI Seminario Anual de Automática, Electrónica Industrial y Automatización, SAAEI'09, Leganés, Madrid, España, 2009.
- [PUH11] M. C. Pérez, J. Ureña, A. Hernández. C. De Marziani, A. Jiménez, "Optimized Correlator for LS Codes-Based CDMA Systems," IEEE Communications Letters, Vol. 15, No. 2, 2011.
- [RS83] O. T. Von Ramm y S. W. Smith, "Beam Steering with Linear Arrays", Biomedical Engineering, BME-30, n° 8, pp. 438-452, 1983.
- [SPR91] S. W. Smith, H. G. Pavy y O. Von Ramm, "High-speed ultrasound volumetric imaging system. Part I: Transducer design and beam steering", IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, Vol. 38, no. 2, 1991.
- [Tex99] Texas Instruments Inc. "Quad JFET-Input General-Purpose Operational Amplifier TL084 datasheet", 1999
- [TL72] C. C. Tseng y C. L. Liu, "Complementary sets of sequences", IEEE Transactions on Information Theory, IT-18(5):644-652, 1972.
- [TX11] Texas Instruments, "FilterPro User's Guide", Febrero 2011.
- [TX13] Texas Instruments, "DAC121S101/DAC121S101Q 12-Bit Micro Power, RRO Digital-to-Analog Converter datasheet", Marzo 2013.
- [Xil11] Xilinx Inc, LogicCORE IP Multi-Port Memory Controller (MPMC), v6.03.a, 01/03/2011.
- [ZYH05] C. Zhang, S. Yamada, M. Hatori, "General method to construct LS codes by complete complementary sequences", IEICE Trans. On Wireless Communications Technologies, E-88-B, vol. 8, pp. 3484-3487, 2005.