

Empleo de FPGAs de última generación para procesamiento de señales de un Phased Array ultrasónico

Máster Universitario en Sistemas Electrónicos Avanzados.

Sistemas Inteligentes

Departamento de Electrónica

Presentado por:

Sara Gutiérrez López

Dirigido por:

Dra. Mª Carmen Pérez Rubio

Dr. Álvaro Hernández Alonso

Alcalá de Henares, a 22 de septiembre de 2014

Agradecimientos

Doy las gracias a mis directores Mª Carmen y Álvaro que han estado pendientes de mi trabajo, por su increíble apoyo y por estar siempre dispuestos a ayudarme y atenderme.

Quisiera también agradecerle a Fernando su recomendación para hacer este máster, por su motivación y por haber estado siempre ahí.

A Fran por su inestimable ayuda y cariño.

Por último, a mi familia y amigos por haber creído en mí y estar a mi lado.

Sin todos vosotros nada de esto hubiera sido posible.

Contenido

Resum	en		1
Capítu	lo 1	Introducción	3
1.1	Visi	ión general y contexto	3
1.2	Sist	temas ultrasónicos Phased-Array	3
1.3	Pla	taforma de procesamiento	5
1.4	Obj	jetivos	6
1.5	Est	ructura del documento	7
Capítu	lo 2	Arquitectura global del sistema	9
2.1	Cor	ntexto	9
2.2	Mó	dulo emisor	10
2.	.2.1	Estrategias de Codificación	10
2.	.2.2	Estrategia de Modulación	13
2.	.2.3	Configuración del Bloque Emisor	14
2.3	Mó	dulo receptor	16
2.	.3.1	Demodulador BPSK	16
2.	.3.2	Correlador LS	17
2.	.3.3	Simulación en Matlab	20
Capítu	lo 3	Implementación en FPGA	23
3.1	Est	udio del kit de evaluación Virtex-7 VC707	23
3.	.1.1	Introducción	23
3.	.1.2	Familia Virtex 7	23
3.	.1.3	AMS	25
3.2	lmp	olementación del módulo emisor	26
3.3	lmp	plementación del módulo receptor	30
3.	.3.1	Etapa de acondicionamiento de la señal	30

3.3.2	Procesamiento de bajo nivel: Demodulador y Correlador	37
Capítulo 4	Resultados	43
4.1 Sim	nulación temporal	43
4.2 Co	nsumo de recursos	45
4.2.1	Emisor	45
4.2.2	Receptor	45
Capítulo 5	Conclusiones y trabajos futuros	51
Capítulo 6	Bibliografía	53

Índice de figuras

Figura 1.1: (a) PA con un ejemplo del haz proyectado al no aplicar retardo. (b) PA con u ejemplo del haz proyectado al aplicar retardo. (c) Sectores angulares según el retard [GPHSCD13]	lc
Figura 2.1: Diagrama de bloques del sistema [G14].	9
Figura 2.2: ACF de códigos LS de longitud <i>L</i> =2303 y con IFW=5111	C
Figura 2.3: CCF de códigos LS de longitud <i>L</i> =2303 y con IFW=5111	1
Figura 2.4: Estructura de generación de K=4 códigos LS [P09]1	2
Figura 2.5: Esquema de emisión y recepción usando la técnica CDMA [GPHSCD13]1	2
Figura 2.6: Diagrama de modulación BPSK1	3
Figura 2.7: Símbolo sinusoidal usado para la modulación BPSK1	3
Figura 2.8: Modulación para 8 códigos LS con K=8, W=255, Lo=256, L=23031	5
Figura 2.9: Imagen ampliada de la modulación para 1 código LS con $K=1$, $W=255$, $L=2503$	
Figura 2.10: Esquema de modulación y demodulación1	6
Figura 2.11: Imagen de la demodulación para 1 código LS con K=1, W=255, Lo=256, con longitud L=2303	
Figura 2.12: Diagrama de bloques del correlador eficiente Golay (2-ESSC) [Bud91, Pop99, P09	-
Figura 2.13: Diagrama de bloques de un correlador eficiente LS [P09]1	8
Figura 2.14: Diagrama de bloques de la implementación del demodulador con el ELSC para detección simultánea de los ecos en los ocho sectores angulares analizados1	
Figura 2.15: Optimización propuesta para la detección simultánea de los ecos en los och sectores angulares analizados1	
Figura 2.16: Demodulación para 1 código LS con <i>K</i> =1, <i>W</i> =255, <i>L</i> o=256. <i>L</i> =23032	C
Figura 2.17: Imagen ampliada de la modulación para 1 código LS con K=1, W=255, Lo=25 L=23032	
Figura 2.18: Correlación de cada código con el código 1 de la emisión2	2
Figura 3.1: VC707 XC7VX485T-2FFG1761C <i>Evaluation Board</i> [Dig14]2	3
Figura 3.2: AMS 101 Evaluation Card [Dig14]2	4
Figura 3.3: Ejemplo de características según el tipo de FPGA [Xil14]2	4
Figura 3.4: Diagrama de bloques del AMS 101 [Xil13]	_

Figura 3.5: Diagrama de bloques del emisor [G14]	26
Figura 3.6: Diagrama de bloques implementados en la emisión	28
Figura 3.7: Simulación funcional del bloque emisor con las salidas del demodulador uso de códigos LS de longitud <i>L</i> =71.	
Figura 3.8: Simulación funcional del bloque emisor con las salidas de los DACs hacien códigos LS de longitud <i>L</i> =71	
Figura 3.9: Diagrama de bloques del receptor [G14]	30
Figura 3.10: Diagrama de bloques de la FPGA y AMS 101.	31
Figura 3.11: Puertos de la primitiva del XADC [Dig14]	32
Figura 3.12: Definición de los bits de configuración de los registros del XADC [Dig14]	32
Figura 3.13: Bits de configuración del Registro 0 del XADC [Dig14]	32
Figura 3.14: Bits de configuración del Registro 1 del XADC [Dig14]	33
Figura 3.15: Función de transferencia del modo unipolar [Dig14]	34
Figura 3.16: Modo por eventos [Dig14].	35
Figura 3.17: Operación se escritura en serie del AD5065 [An08]	36
Figura 3.18: Puertos del bloque demodulador.	38
Figura 3.19: Puertos del bloque del DCM.	39
Figura 3.20: Puertos del bloque del TIC Enable.	39
Figura 3.21: Simulación funcional de la señal de TIC enable.	40
Figura 3.22: Simulación temporal de donde se comprueba la duración de la señal TIC.	40
Figura 3.23: Puertos del bloque ESSC.	40
Figura 3.24: Puertos del bloque del correlador LS.	41
Figura 3.25: Simulación funcional del sistema completo para una secuencia de och con longitud L =71 concatenados y correlados con cada uno de los ocho correlados	_
Figura 4.1: Simulación temporal del sistema total para un código LS de longitud 71 uso de un único correlador ELSC	
Figura 4.2: Imagen ampliada de la simulación temporal del sistema total para un cóc longitud 71 haciendo uso de un único correlador ELSC	-
Figura 4.3: Simulación en Matlab de la auto-correlación para un código LS de longitud	7144

Índice de tablas

Tabla 2.1: Comparativa de las necesidades computacionales para el correlador ELSC y el ELSC con la optimización para una longitud del código LS de <i>L</i> =460720
Tabla 3.1: Grado de velocidad y rangos de temperaturas para las FPGAs de la serie 7 [Dig14]. 24
Tabla 3.2: Recursos de las distintas Virtex-7. En rojo se muestra la Virtex-7 XC7VX485T [Dig14]25
Tabla 3.3: Recursos de la familia Virtex-5 [Xil09]27
Tabla 3.4: Selección del canal del ADC [Dig14]33
Tabla 3.5: Ajustes en la operación de la secuencia [Dig14]
Tabla 3.6: Características temporales del AD5065 [An08]
Tabla 3.7: Especificaciones del sistema receptor37
Tabla 3.8: Especificaciones del sistema receptor para los ocho correladores ELSC38
Tabla 4.1: Consumo de recursos y porcentaje de ocupación del bloque Emisor para la longitud de 287 bits para la Virtex-545
Tabla 4.2: Consumo de recursos y porcentaje de ocupación del bloque Demodulador para la Virtex-7 VC707 XC7VX485T-2FFG1761C45
Tabla 4.3: Consumo de recursos y porcentaje de ocupación del Correlador sin optimizar (ESSC+Bloque de retardos) para la Virtex-7 VC707 XC7VX485T-2FFG1761C46
Tabla 4.4: Consumo de recursos y porcentaje de ocupación del bloque ESSC para la Virtex-7 VC707 XC7VX485T-2FFG1761C46
Tabla 4.5: Consumo de recursos y porcentaje de ocupación de un único Correlador Optimizado correspondiente al bloque de retardos y matriz de Hadamard. para la Virtex-7 VC707 XC7VX485T-2FFG1761C47
Tabla 4.6: Comparación del consumo de recursos a nivel de Slices ocupados para los ocho correladores sin optimizar y optimizados para la Virtex-7 VC707 XC7VX485T-2FFG1761C
Tabla 4.7: Consumo de recursos y porcentaje de ocupación del Sistema Receptor con el Correlador sin optimizar para la Virtex-7 VC707 XC7VX485T-2FFG1761C48
Tabla 4.8: Consumo de recursos y porcentaje de ocupación del Sistema Receptor con el Correlador optimizado para la Virtex-7 VC707 XC7VX485T-2FFG1761C48

Resumen

En el presente trabajo se ha desarrollado el diseño y la implementación de la optimización de un correlador eficiente basado en técnicas de codificación. Se hace uso de la técnica combinada de *Phased Array* junto con la codificación CDMA, las cuales permiten la emisión simultánea de códigos en distintos sectores angulares.

Como técnica de codificación se han elegido secuencias de códigos LS. Esta familia de códigos tiene una mayor ganancia de proceso que otros códigos estudiados, como son los CSS, permitiendo tener una mayor profundidad de exploración del medio. Para la implementación hardware usada en el procesamiento de bajo nivel de recepción de la señal, se dispone de la plataforma FPGA (*Field Programmable Gate Array*), en contreto la VC707 XC7VX485T-2FFG1761C Virtex-7 de Xilinx.

El diseño propuesto supone una optimización en cuanto al consumo de recursos de la FPGA, permitiendo de esta manera la implementación de longitudes de códigos LS mayores a las que se podrían emplear sin dicha optimización, obteniendo distancias más altas. Para llevar a cabo la implementación del diseño se ha realizado un estudio detallado de las características y la configuración de la Virtex-7, de manera que se ajuste a las especificaciones requeridas.

Con objeto de comprobar el avance de la optimización propuesta, se ha elaborado un análisis comparativo del consumo de recursos del correlador optimizado frente al correlador sin optimizar. A su vez, se exponen los resultados obtenidos en las simulaciones del diseño global, teniendo en cuenta el emisor y el receptor, para la verificación del sistema propuesto.

Capítulo 1 Introducción

1.1 Visión general y contexto

Este trabajo abarca el diseño e implementación de un sistema ultrasónico basado en *Phased Arrays*, haciendo uso de un correlador y secuencias LS como técnica de codificación. El estudio que aquí se presenta es la continuación de un trabajo anterior [G14] cuyo objetivo era el diseño y procesamiento de un *array* ultrasónico basado en técnicas de codificación a partir de conjuntos de secuencias complementarias (CSS) para la generación de imágenes por ultrasonidos en aire. La implementación hardware se realizaba a través de una FPGA que en este caso pertenecía a la familia Virtex-5 de Xilinx. La limitación de este trabajo venía dada por la baja ganancia del proceso de la codificación elegida, lo que restringía la distancia máxima de análisis.

En este trabajo, el procesamiento se realiza en una FPGA de mayor capacidad, perteneciente a la familia Virtex-7 de Xilinx, lo que ha permitido usar códigos con mayor ganancia de proceso como los LS, los cuales permiten una distancia de exploración mayor que los códigos CSS, para una longitud de secuencia similar.

1.2 Sistemas ultrasónicos Phased-Array

La velocidad del ultrasonido al atravesar un medio depende de la densidad y compresibilidad, cuanto más sólido es el medio, mayor velocidad alcanza el ultrasonido. Cuando se propaga por el aire el ultrasonido tiene una velocidad de 340m/s.

Un sistema de inspección convencional por ultrasonidos se suele basar en un único transductor piezoeléctrico emisor/receptor, el cual genera un pulso ultrasónico que se propaga en el material produciendo una reflexión en todos los puntos donde se encuentre el objeto quedando estos registrados mediante el pulso reflejado. Los sensores de ultrasonido se utilizan para estimar la distancia a objetos cercanos calculando el tiempo que le toma al pulso de ultrasonido viajar desde el transductor al objeto y regresar, esto es lo que se conoce como tiempo de vuelo. Este tiempo está directamente relacionado con la distancia recorrida según la velocidad de propagación de las ondas acústicas, por lo que se puede obtener información acerca de la posición del obstáculo detectado.

El problema cuando se utiliza un único transductor es que para analizar todo el entorno habría que mover físicamente dicho transductor, provocando así un aumento del tiempo en la obtención de los resultados. Una de las soluciones más usada para solventar este problema es lo que se conoce como *Phased Array*.

La tecnología *Phased Array* se basa en un conjunto de elementos (sensores de ultrasonido) que son activados con una serie de retardos electrónicos para crear haces con un determinado ángulo de direccionamiento que es controlado mediante *software* [RS83]. El

Phased Array ultrasónico se utiliza en una amplia variedad de aplicaciones e industrias. Hoy en día su aplicación abarca sectores como el defensa, petroquímica, manufactura nuclear, entre otros [R10] [BGL01]. Este trabajo se centra en los *Phased Array* aplicados en robótica.

Dicha tecnología posee múltiples ventajas respecto a otros métodos como pueden ser las técnicas de radiografía y los ultrasonidos convencionales.

- No contamina el medio ambiente con residuos químicos ni radiactivos eliminando la probabilidad de incidentes radiactivos.
- Admite la exploración electrónica a gran velocidad y sin partes móviles, lo que permite que el resultado de la prueba sea inmediato, esto genera una disminución de tiempo en la obtención del resultado final.
- Permite el registro en el modo más adecuado para su interpretación y evaluación (S-scan, B-scan, C-scan y otras imágenes 2D y 3D).
- Se puede realizar una gran variedad de tipos de exploración para la detección de diferentes tipos de objetos localizados en distintas configuraciones por lo que tiene una gran flexibilidad.
- La técnica tiene una gran resolución para su dimensionado y posicionado en tiempo real.
- Mejora la calidad de la inspección al permitir el control mediante software de las características del haz sonoro.
- Facilita una optimización del ángulo y de la focalización.
- Permite el almacenamiento digital de los datos.

El objetivo final es el de formar imágenes *B-scan* del entorno. Para ello, cada sector angular se divide en *N* subsectores de igual tamaño. El *array* emite la señal ultrasónica en cada sector con un ángulo $\theta_{n=1...N}$, controlando dicho ángulo según se indica en (2.1),

$$\tau_n = \frac{1}{c} \cdot x_n \cdot \sin\left(\theta_n\right) \tag{2.1}$$

donde c es la velocidad del sonido, considerada constante ($c = 340 \, m/s$).

Los transductores son activados en fase en diferentes tiempos, por lo que el haz de ultrasonido es emitido en un ángulo determinado formando una imagen sectorial como se observa en la Figura 1.1. Esto tiene el inconveniente que por cada sector angular a evaluar hay que hacer una nueva transmisión con todos los elementos del *array*, lo que provoca que haya una baja tasa en la generación de imágenes.

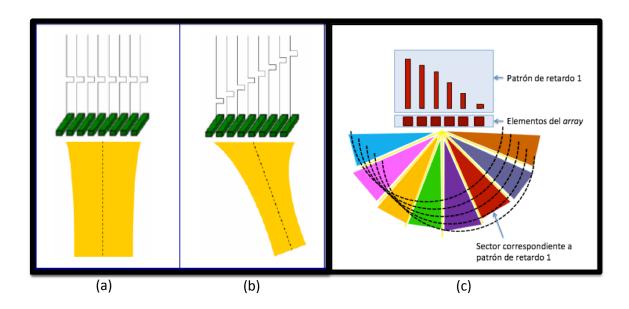


Figura 1.1: (a) PA con un ejemplo del haz proyectado al no aplicar retardo.
(b) PA con un ejemplo del haz proyectado al aplicar retardo.
(c) Sectores angulares según el retardo [GPHSCD13].

Una solución a este problema son las técnicas basadas en acceso múltiple por división de código o CDMA (*Code Division Multiple Access*). Combinando la técnica *Phased Array* con CDMA y asignando un código a todos los elementos del *array* con un retardo correspondiente a un sector angular determinado, se puede realizar una emisión en todos los elementos del *array* de forma simultánea en todas las direcciones. Otra ventaja de la técnica CDMA es que aumenta la relación señal-ruido haciendo este sistema más inmune frente a interferencias externas causadas por el medio.

1.3 Plataforma de procesamiento

Actualmente los dispositivos lógicos programables tipo FPGA están presentes en una gran cantidad de productos industriales, ya que presentan la ventaja de tener la capacidad de ser reprogramables y que su fabricación no es excesivamente costosa y larga, comparado con otros dispositivos integrados como los ASIC. Éste es el motivo por el cual se eligen a las FPGAs para una gran variedad de aplicaciones [O03]. Dada su arquitectura interna pueden trabajar de forma flexible en diseños síncronos y asíncronos.

En este trabajo se hace uso de la plataforma Xilinx Virtex-7 VC707 *Evaluation Kit* la cual se explica con mayor detalle en el Capítulo 4.

1.4 Objetivos

El principal objetivo de este trabajo se centra en estudiar una mejora para el sistema de generación de imágenes, de tal manera que se pueda aumentar la distancia de exploración del *Phased Array* ultrasónico haciendo uso de técnicas de codificación. Para ello el trabajo puede dividirse en varias tareas para alcanzar el objetivo principal.

Estudio de la implementación previa

Se realiza un estudio previo de las técnicas de codificación y correlación, así como simulaciones de distintos códigos y longitudes de códigos con Matlab, empleando el algoritmo del trabajo [G14] para tener una mayor comprensión de dichas técnicas y sus propiedades.

Migración de la técnica de codificación

El estudio previo permite la realización de uno de los objetivos de este trabajo que consiste en migrar la técnica de codificación de códigos CSS a LS, ya que estos códigos tienen una mayor ganancia de proceso y por lo tanto permiten una mayor profundidad de exploración del entorno.

Optimización del procesamiento

Continuando con el trabajo se realiza una optimización del procesamiento paralelo para dichos códigos LS. Se realiza un estudio detallado de la optimización de un correlador eficiente que permitirá una reducción en el consumo de recursos del sistema. A su vez se lleva a cabo un análisis comparativo de dicho correlador con la optimización propuesta y sin dicha optimización, para así comprobar la ventaja que supone.

Implementación en FPGA

Se debe ajustar el sistema propuesto a la implementación en la FPGA Virtex-7. Para ello, se estudia dicha plataforma de procesamiento y sus principales características.

Diseño del sistema receptor para la Virtex-7

Para finalizar, una vez comprendido el funcionamiento y configuración de la Virtex-7, se realiza el diseño del sistema total junto con la implementación de la optimización del correlador eficiente estudiado.

1.5 Estructura del documento

El presente trabajo se estructura en siete capítulos, comenzando en el Capítulo 2 con un análisis de la arquitectura global del sistema propuesto, donde se estudia la arquitectura del módulo emisor y receptor, ambos necesarios para implementar el sistema, así como las estrategias de codificación, modulación y correlación.

A continuación, en el Capítulo 3 se lleva a cabo el estudio de la implementación en la FPGA analizando en mayor detalle la plataforma Virtex-7, y el diseño e implementación del módulo emisor y receptor en FPGA.

Posteriormente, en el Capítulo 4 se muestran los resultados obtenidos realizando un análisis del consumo de recursos, separando el módulo emisor del receptor. A su vez, se exponen los resultados obtenidos en las simulaciones necesarias para la correcta verificación del sistema propuesto.

Para finalizar, en el Capítulo 5 se exponen las conclusiones del estudio realizado así como las líneas futuras de investigación que podrían derivar de dicho trabajo.

Capítulo 2 Arquitectura global del sistema

2.1 Contexto

El sistema propuesto consta de un bloque emisor y otro receptor tal y como se muestra en la Figura 2.1.

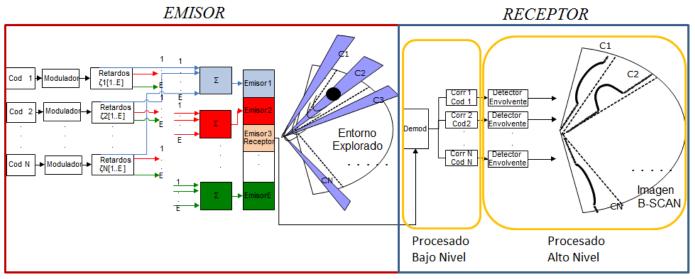


Figura 2.1: Diagrama de bloques del sistema [G14].

El módulo emisor se encarga de almacenar y modular los códigos que se van a transmitir, con los retardos correspondientes dependiendo del sector angular a explorar. Después de este proceso todos los códigos que se van a emitir son sumados para así emitirlos de forma simultánea en todos los sectores angulares.

En módulo receptor se divide en un procesado de bajo nivel y otro de alto nivel. El procesado de bajo nivel demodula la señal recibida y hace la correlación con los patrones emitidos. El bloque de alto nivel se encarga de generar la imagen B-scan.

Este trabajo se ha centrado en acondicionar el bloque emisor para los códigos LS y modularlos para así poder emitirlos. Respecto al bloque receptor, se ha abordado el procesado de bajo nivel, lo que corresponde a realizar la demodulación de dichos códigos LS y hacer la correlación con un correlador LS eficiente¹. A su vez se ha realizado una optimización de dicho correlador eficiente para reducir el consumo de recursos en la implementación.

¹ Se utiliza el término eficiente para indicar que la correlación puede realizarse con un menor número de operaciones, y por tanto, en un tiempo menor que usando un método de correlación convencional.

2.2 Módulo emisor

2.2.1 Estrategias de Codificación

Los sistemas de codificación son usados en diversas áreas de la ingeniería como comunicaciones, robótica, sistemas de radar/sonar, ensayos no destructivos de materiales, etc. [G14]. Sus propiedades matemáticas los hacen ventajosos para aquellas aplicaciones donde haya que recuperar información digital contenida en una señal que está afectada por ruido, atenuación del canal e interferencia con otras fuentes.

En este trabajo se estudia el uso de códigos LS (*Loosely Synchronous codes*) [P09]. Son códigos ternarios, derivados de códigos CSS (*Complementary Sets of Sequences*) [ZYH05] o de parejas Golay [SBH01]. Se usan en entornos cuasi-síncronos y poseen una ventana libre de interferencias IFW (*Interference Free Window*) alrededor del origen, donde los lóbulos laterales de la función aperiódica de auto-correlación (ACF) y de correlación cruzada (CCF) son nulos.

Este trabajo se va a centrar en códigos LS construidos a partir de dos pares Golay incorrelados [SBH01]. Estos pares son incorrelados debido a que la suma de las correlaciones entre las correspondientes secuencias de cada par es cero para cualquier desplazamiento. En la Figura 2.2 y Figura 2.3, se muestra un ejemplo gráfico de función de auto-correlación ACF y de correlación cruzada CCF de códigos LS para una longitud de L=2303 bits.

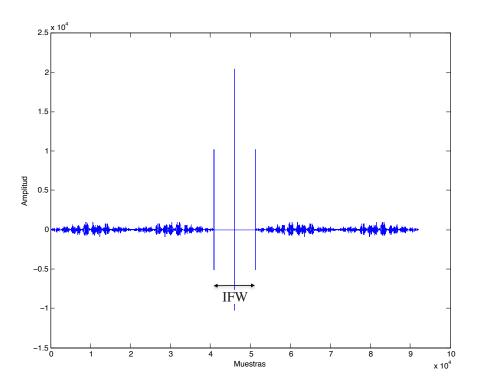


Figura 2.2: ACF de códigos LS de longitud L=2303 y con IFW=511.

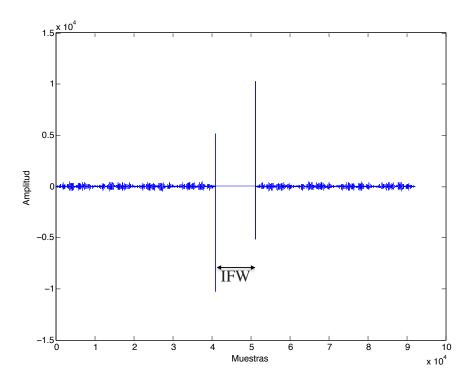


Figura 2.3: CCF de códigos LS de longitud L=2303 y con IFW=511.

Sean así dos pares Golay incorrelados $S_0=(s_{0,0},s_{0,1})$ y $S_1=(s_{1,0},s_{1,1})$ de longitud L_0 , compuestos por elementos $\{-1,+1\}$, se puede obtener una familia $\{G=g_k[l]; 0\leq k\leq K-1; 0\leq l\leq L-1\}$ con K códigos LS de longitud L como se indica en (3.1), siendo K un número potencia de dos.

$$G_k[z] = \sum_{i=0}^{\frac{K}{2}-1} h_{k,i} \cdot z^{-i \cdot L_0} \cdot [S_{\pi_{i},0}[z] + z^{-\left(\frac{K}{2}L_0 + W\right)} S_{\pi_{i},1}[z]]$$
(3.1)

que representa la transformada Z de $g_k[\tau]$, donde los coeficientes $h_{k,i} \in \{-1,+1\}$ son los elementos de una matriz de Hadamard con tamaño $\left(\frac{K}{2} \times \frac{K}{2}\right)$ que indican la polaridad con la que se concatenan las secuencias de los pares Golay $S_{j,i}[z]$, $0 \le i,j \le 1$, siguiendo el orden establecido por π_i ; siendo el vector $\Pi = \left[\pi_0, \pi_1, \dots, \pi_{\frac{K}{2}-1}\right] \in \{0,1\}$ la representación binaria de un número natural $n,0 \le n \le 2^{K/2}$; W es el número de ceros a introducir entre las secuencias 0 de los pares y las secuencias 1, e indica el tamaño de la semi-ventana libre de interferencias.

Con esto se generan los primeros K/2 códigos LS y para los siguientes K/2 códigos hay sustituir en la ecuación el vector Π por su vector complementario, siendo éste,

$$\Pi^* = \left[\pi_0^*, \pi_1^*, \dots, \pi_{\frac{K}{2} - 1}^*\right], \pi_k^* = \pi_k + 1 \pmod{2}, 0 \le k \le \frac{K}{2} - 1 \tag{3.2}$$

Si las emisiones son periódicas hay que establecer un intervalo de guarda con un mínimo de W ceros. La longitud final del código LS es,

$$L = K \cdot L_0 + W \tag{3.3}$$

siendo K el número de secuencias LS obtenidas; L_0 la longitud de la pareja Golay a partir de la cual se generan las secuencias LS; y W corresponde con la semi-ventana libre de interferencias. La longitud de la IFW es 2W+1 donde $W \leq L_0-1$.

En la Figura 2.4 se muestra la estructura de K=4 códigos LS a partir de un vector $[0\ 1]$ y una matriz de Hadamard $h_0=[1\ 1], h_1=[1-1].$

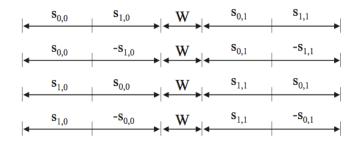


Figura 2.4: Estructura de generación de K=4 códigos LS [P09].

Debido a que estos códigos tienen una IFW, si todas las recepciones se reciben dentro de esa ventana es posible mitigar las interferencias ISI (*Inter-Symbol Interference*) y MAI (*Multiple Access Interference*) permitiendo así una detección más eficiente. Las ISI son causadas por valores no nulos en la ACF; y las MAI se producen debido a que las CCF entre códigos pueden no ser cero. Para evitarlas los lóbulos laterales de ACF y CCF deben ser lo más pequeños posible. Como en los códigos LS las interferencias fuera de la IFW son más elevadas que en otros códigos, hay que asegurar que la diferencia de tiempos máxima entre recepciones sea menor que W.

De esta manera si en una primera etapa se emiten todos los códigos simultáneamente en todos los elementos del *array*, al realizar la correlación con cada código en la etapa de recepción se tendría un pico de correlación en aquel sector angular en el que haya un obstáculo, ya que cada sector angular se encuentra perfectamente identificado por un único código pseudo-incorrelado con los demás (véase la Figura 2.5).

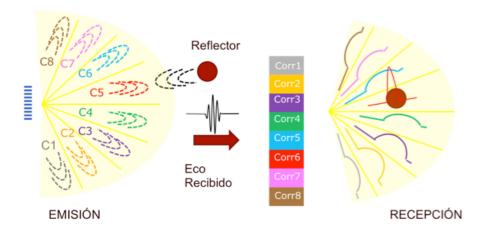


Figura 2.5: Esquema de emisión y recepción usando la técnica CDMA [GPHSCD13].

2.2.2 Estrategia de Modulación

La modulación engloba el conjunto de técnicas para transportar información sobre una onda portadora. Básicamente, la modulación consiste en hacer que un parámetro de la onda portadora cambie de valor de acuerdo con las variaciones de la señal moduladora, que es la información que se desea transmitir. Se ha usado una modulación de fase caracterizada porque la fase de la onda portadora, de amplitud constante, varía con la señal de modulación. Debido a que en las modulaciones PSK (*Phase Shift Keying*) la potencia de todos los símbolos es igual, se simplifica el diseño de los amplificadores y las etapas receptoras, reduciendo costes y posee cierta inmunidad frente al ruido.

Específicamente, se ha usado una modulación BPSK (*Binary Phase Shift Keying*). En este tipo de modulación, la portadora sinusoidal toma dos valores de fase separadas 180°, una indica el estado bajo "-1" y otra a 0° para el estado alto "1". A mayor separación entre los estados, para el receptor será más fácil distinguir entre ellos en presencia de ruido e interferencias. Sin embargo, la modulación BPSK tiene como desventaja que la velocidad de transmisión es la más baja de todas las modulaciones de fase, ya que cada símbolo sólo aporta un bit de información. En la Figura 2.6 se puede ver un esquema del diagrama de la modulación BPSK.

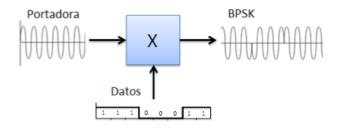


Figura 2.6: Diagrama de modulación BPSK.

Por lo tanto, una vez generados los códigos, se aplica la modulación. En el presente trabajo, se ha usado una modulación BPSK haciendo uso de un símbolo sinusoidal con un único ciclo de portadora (véase la Figura 2.7).

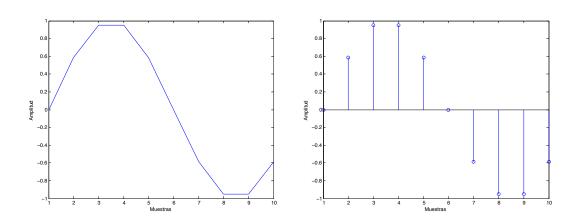


Figura 2.7: Símbolo sinusoidal usado para la modulación BPSK.

2.2.3 Configuración del Bloque Emisor

El trabajo previo del que se parte consiste en la codificación de las transmisiones ultrasónicas haciendo uso de códigos derivados de conjuntos de secuencias complementarias (CSS) con el objetivo de realizar una exploración de varios sectores angulares simultáneamente. Para ello se usaron 32 sectores diferentes, por lo que se necesitaban 32 códigos, cada uno de longitud $L=11998\ bits$. La emisión se realizaba a través de un array de 8 elementos. El bloque emisor se encargaba de almacenar y modular los códigos que se iban a transmitir y de generar los retardos necesarios correspondientes a cada sector. La implementación hardware se realizaba sobre la plataforma Genesys desarrollada por Digilent Inc., basada en la FPGA Virtex-5 LX50T.

Las diferencias más significativas respecto al trabajo del que se parte es que en este trabajo para la codificación se han usado códigos LS y se han emitido 8 códigos debido a la limitación respecto a la resolución angular que tiene el *array* ultrasónico. Aunque las emisiones de los códigos se realizan con la misma FPGA Virtex-5, la recepción se lleva a cabo a través de la FPGA Virtex-7 explicada en el Capítulo 3 La ventaja del uso de códigos LS frente a los códigos CSS es que estos códigos tienen una mayor ganancia por lo que se consigue un pico de correlación mayor.

Para la configuración de dicho bloque, inicialmente se verán las especificaciones del sistema propuesto.

Especificaciones del sistema

- Frecuencia de portadora: $f_c = 40kHz$
- Frecuencia de muestreo: $f_s = 400kHz$

Para la generación de los códigos LS en Matlab a partir de parejas Golay hay que especificar las siguientes variables,

- K: número de emisiones simultáneas, que es igual al número de códigos que se van a generar. Este valor debe ser potencia de 2.
- W: número de ceros a introducir en el código, que es igual al tamaño de la semiventana sin interferencias que queda alrededor del origen. Se fija su valor a L_0 -1. $W \geq 354$ para cubrir al menos distancias desde 0.30m. Esto obliga a usar N=512 y por tanto W=511.
- L_0 : longitud inicial de las parejas Golay, debiendo cumplir que sea potencia de 2.
- Sem: Semilla para la generación de parejas Golay iniciales. Esta señal representa al
 conjunto de los coeficientes de generación de la secuencia. Por lo tanto existirán
 tantos pares de secuencias como semillas diferentes existan.

Para el array con el que se trabaja en el laboratorio, se tiene que,

- K = 8 sectores angulares.
- W = 511.
- $L_0 = 512$.
- Sem = 0.

Con estos datos se obtiene una familia de ocho códigos de longitud $L=4607\ bits$. Esto supone un ahorro de $L=7391\ bits$ frente a lo que necesitaba el sistema anterior con 32 sectores angulares y longitud $L=11998\ bits$.

Simulación

Realizando la simulación en Matlab con estas especificaciones, los resultados gráficos de la modulación para los ocho códigos usando una portadora sinusoidal son los siguientes (véase la Figura 2.8):

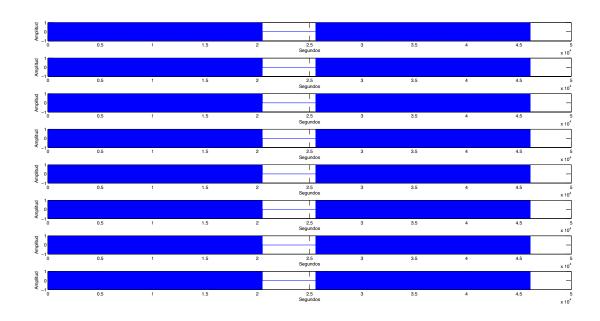


Figura 2.8: Modulación para 8 códigos LS con *K*=8, *W*=255, *L*o=256, *L*=2303.

Analizando el primer código, mostrado en la Figura 2.9, se pueden observar dichos resultados de la modulación.

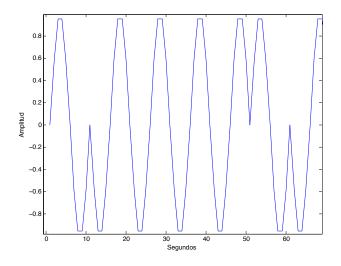


Figura 2.9: Imagen ampliada de la modulación para 1 código LS con K=1, W=255, L0=256, L=2303.

2.3 Módulo receptor

2.3.1 Demodulador BPSK

Después de enviar la información usando una modulación BPSK, se debe recuperar la dicha información captada por un ADC y después dirigirla al demodulador BPSK. Éste tiene que tener la capacidad de rescatar la información que fue codificada anteriormente (véase la Figura 2.10).

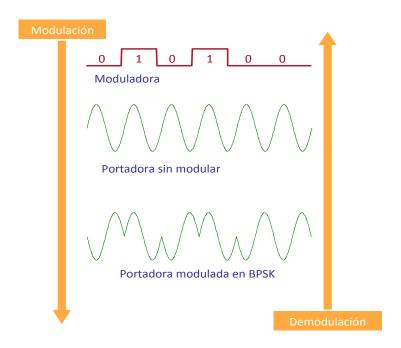


Figura 2.10: Esquema de modulación y demodulación.

Al configurar el demodulador hay que tener en cuenta todos los parámetros que se establecieron en el modulador para así obtener un resultado correcto. Dicha demodulación se lleva a cabo a través de un bloque demodulador al que le llega la señal del modulador BPSK y la portadora sinusoidal que tiene la misma frecuencia y fase que la que se usó en la emisión.

Para la demodulación, se tiene,

$$Sd_{dem} = \frac{A}{2} \cdot O_f \cdot \cos \theta \tag{3.4}$$

Donde Sd_{dem} es la señal demodulada digitalizada ausente de ruido, A es la amplitud de la señal modulada y O_f el número de muestras por símbolo. La demodulación se ha realizado de forma aperiódica por lo que la señal resultante se comporta de forma sinusoidal y el valor de fase de los símbolos adquiridos se obtiene cada O_f muestras.

Se ha realizado una simulación en Matlab de la demodulación BPSK con A=1 y $O_f=10$.

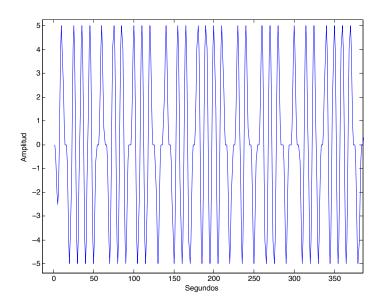


Figura 2.11: Imagen de la demodulación para 1 código LS con K=1, W=255, Lo=256, de longitud L=2303.

2.3.2 Correlador LS

En términos generales, un correlador es un proceso digital o analógico que trata de encontrar una información determinada contenida en una secuencia de datos. En el correlador se realiza el producto entre la señal recibida y la muestra patrón. Se estudia un correlador eficiente donde el número de operaciones son reducidas, frente a uno directo.

Existen métodos optimizados de correladores eficientes donde se simplifica el proceso de detección como proponen [Bud91] y [Pop99] para parejas de secuencias Golay donde se implementan las multiplicaciones como sumas y restas (véase la Figura 2.12), de la misma forma que lo hace el correlador directo. Teniendo como resultado la correlación de la secuencia de entrada con las dos secuencias del par Golay.

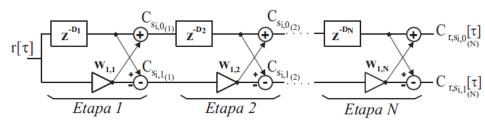


Figura 2.12: Diagrama de bloques del correlador eficiente Golay (2-ESSC) [Bud91, Pop99, P09].

La correlación de códigos LS puede realizarse de forma eficiente (ELSC, Efficient LS Correlator) mediante la arquitectura propuesta en la Figura 2.13; en donde el bloque 2-ESSC representa el correlador eficiente Golay de la Figura 2.12.

La correlación de una señal de entrada $r[\tau]$ con un código LS $g_k[\tau]$, puede expresarse según 3.4 y 3.5,

$$C_{R,G_k}[z] = \sum_{i=0}^{\frac{K}{2}-1} h_{k,i} z^{-\left(\frac{K}{2}-i-1\right) \cdot L_0} \left[z^{-\left(\frac{K}{2}\cdot L_0+W\right)} C_{R,S_{\pi_{i},0}}[z] + C_{R,S_{\pi_{i},1}}[z] \right]$$
(3.4)

$$C_{R,G_{k+K/2}}[z] = \sum_{i=0}^{\frac{K}{2}-1} h_{k,i} z^{-\left(\frac{K}{2}-i-1\right) \cdot L_0} \left[z^{-\left(\frac{K}{2} \cdot L_0 + W\right)} C_{R,S_{\pi_i^*,0}}[z] + C_{R,S_{\pi_i^*,1}}[z] \right]$$
(3.5)

donde $C_{R,G_k}[\mathbf{z}]$ con $0 \leq k \leq K/2$ es la transformada Z de la correlación de la señal de entrada $r[\tau]$; y el código LS $g_k[\tau]$. $C_{R,S_{\pi_i,0}}[z]$ y $C_{R,S_{\pi_i,1}}[z]$ son la transformada Z de la correlación entre la señal de entrada $r[\tau]$ y las secuencias S_0 y S_1 que pueden llevarse a cabo usando un correlador eficiente Golay (EGC o 2-ESSC). En la Figura 2.13 se muestra el diagrama de bloques del ELSC, donde en el Paso 1 cada 2-ESSC realiza la correlación de la secuencia de entrada con las dos secuencias del par dando a la salida las correlaciones $(C_{r,S_{0,0}},C_{r,S_{0,1}})$ y $(C_{r,S_{1,0}},C_{r,S_{1,1}})$. En los Pasos 2 y 3 se retardan las correlaciones obtenidas y se multiplican por los elementos de la matriz de Hadamard. El resultado final es $C_{r,g_k}[\tau]$ resultado de correlar $r[\tau]$ y $g_k[\tau]$.

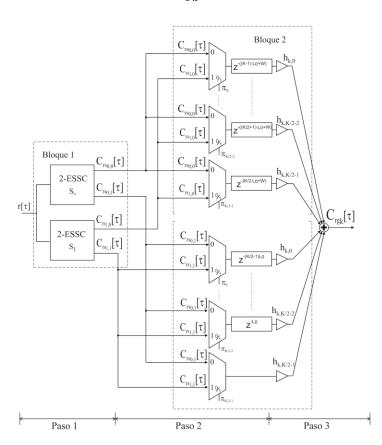


Figura 2.13: Diagrama de bloques de un correlador eficiente LS [P09].

Debido a que se van a analizar ocho sectores angulares, se implementará ocho veces el correlador LS de la Figura 2.13. Por lo tanto el diagrama de bloques para la obtención simultánea de la correlación en los ocho sectores angulares contemplados se muestra en la Figura 2.14.

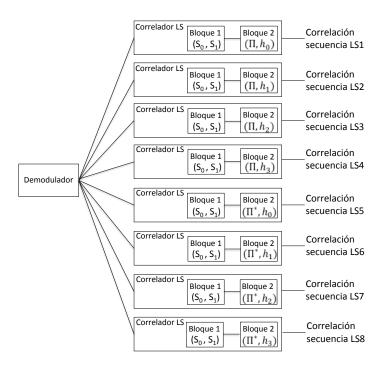


Figura 2.14: Diagrama de bloques de la implementación del demodulador con el ELSC para la detección simultánea de los ecos en los ocho sectores angulares analizados.

En este esquema se tiene un bloque que demodula la señal que le llega del emisor para en una segunda etapa correlarla con los ELSC, cada uno de ellos correspondiente a un sector angular determinado.

Para optimizar recursos se propone un nuevo correlador en donde el bloque 1 se comparte por los ocho correladores, debido a que es común a todos ellos. De este modo, la implementación quedaría según se muestra en la Figura 2.15.

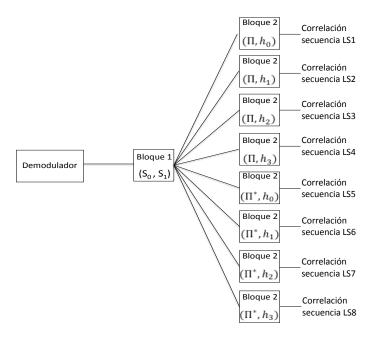


Figura 2.15: Optimización propuesta para la detección simultánea de los ecos en los ocho sectores angulares analizados.

Con esta optimización la implementación es más sencilla, además del ahorro computacional que supone. A continuación se hará una tabla comparativa de las necesidades computacionales en productos, sumas y bits de memoria, expresados en Tabla 2.1, para una longitud de 4607 y W=511. Se compara el ELSC del sistema total, descrito en la Figura 2.14, es decir, implementado para los 8 correladores; y la optimización propuesta teniendo en cuenta los 8 correladores pero el bloque 1 común a todos ellos, visto en la Figura 2.15.

Tabla 2.1: Comparativa de las necesidades computacionales para el correlador ELSC y el ELSC con la optimización para una longitud del código LS de *L*=4607.

Consumo de recursos	ELSC	ELSC optimizado				
Productos	208	82				
Sumas	344	92				
Bits de Memoria	13 183 360	11 968 440				

Se observa que hay una gran ventaja respecto al consumo de recursos en el ELSC con la optimización realizada frente al ELSC sin dicha optimización.

2.3.3 Simulación en Matlab

El proceso de recepción de la señal se divide en dos fases. Una primera donde se realiza la demodulación mediante BPSK con una portadora sinusoidal, es decir, la correlación con el símbolo de modulación; y una segunda en la cual se hará la correlación con dicho símbolo espaciado.

Inicialmente se han hecho todos los cálculos para la emisión de un único código para poder observar mejor los resultados. En la primera fase, como se ha dicho, se ha calculado la demodulación, para un único código, dando como resultado (Figura 2.16):

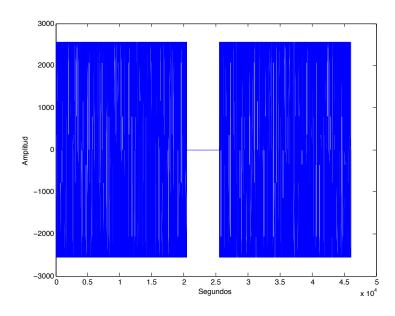


Figura 2.16: Demodulación para 1 código LS con *K*=1, *W*=255, *L*0=256. *L*=2303.

Ampliando la Figura 2.16 se puede observar la demodulación de la señal en la Figura 2.17.

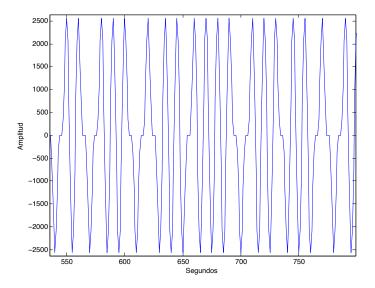


Figura 2.17: Imagen ampliada de la modulación para 1 código LS con K=1, W=255, Lo=256, L=2303.

En la segunda fase, se ha espaciado el código teniendo en cuenta el símbolo de modulación para que corresponda con la emisión y se ha hecho la correlación de la demodulación anterior con el nuevo código espaciado, dando como resultado las correlaciones de la Figura 2.18. En dicha figura, se realiza la correlación del código emitido con cada una de las 8 secuencias LS que se tienen. Cuando el pico de correlación del código emitido coincide con el pico de alguna de las secuencias LS, se ha realizado una auto-correlación, como se muestra en el primer caso de la figura. Para el resto de los casos se tiene la correlación cruzada con valores nulos en la IFW, por lo que el código emitido no coincide con ninguna de las secuencias restantes, no consiguiéndose así un máximo de correlación.

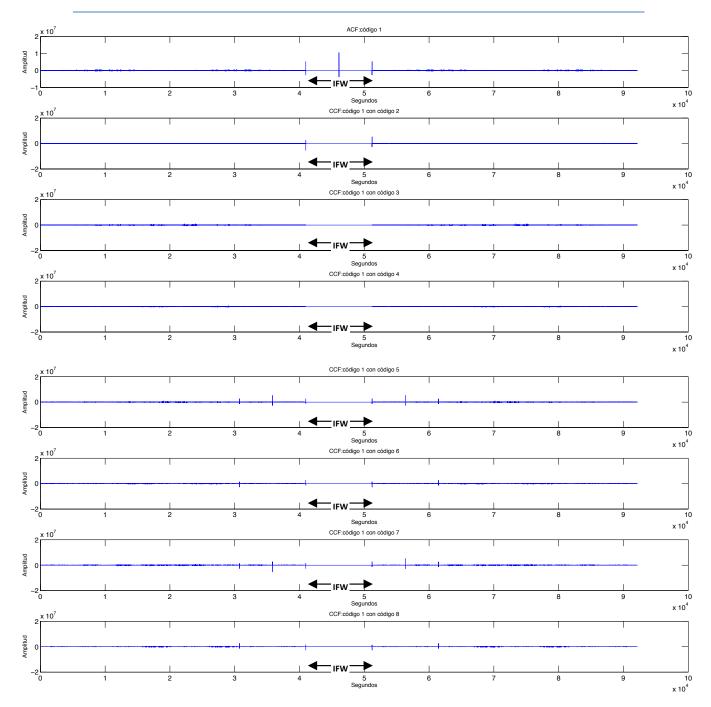


Figura 2.18: Correlación de cada código con el código 1 de la emisión.

Capítulo 3 Implementación en FPGA

3.1 Estudio del kit de evaluación Virtex-7 VC707

3.1.1 Introducción

Las FPGAs (*Field Programmable Gate Arrays*) son dispositivos semiconductores reprogramables, que se basan en matrices de bloques de lógica configurables (conocidos como CLBs) cuyas conexiones se pueden modificar según las necesidades del diseñador. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional, hasta complejos sistemas en un chip. Ofrecen velocidades temporizadas por hardware y fiabilidad, pero sin requerir altos volúmenes de recursos para compensar el gran gasto que genera un diseño personalizado de ASIC. Utilizar tecnología basada en FPGAs ofrece a ingenieros y científicos la habilidad de crear sistemas personalizados reconfigurables con procesamiento integrado.

3.1.2 Familia Virtex 7

En este trabajo se hace uso de la FPGA de Xilinx Virtex-7 VC707 *Evaluation Kit* para el procesamiento en bajo nivel de los datos adquiridos. En la Figura 3.1 se muestra un diagrama con todos los componentes de esta FPGA.

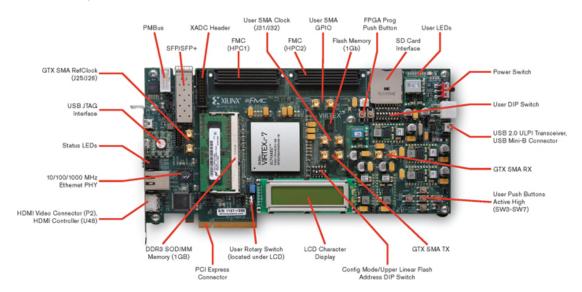


Figura 3.1: VC707 XC7VX485T-2FFG1761C Evaluation Board [Dig14].

Se incluye además la Tarjeta de Evaluación, mostrada en la Figura 3.2.



Figura 3.2: AMS 101 Evaluation Card [Dig14].

El nombre de la FPGA indica algunas de las características de dicha placa. A modo de ejemplo (Figura 3.3) para la FPGA XC7K325T-2FBG900C se tiene.

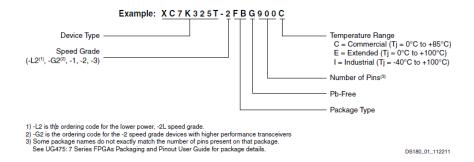


Figura 3.3: Ejemplo de características según el tipo de FPGA [Xil14].

Dado que el modelo de FPGA del que se dispone es la XC7VX485T-2FFG1761C, esto indica que el rango de temperatura es comercial y tiene 1761 pines. A continuación se muestra en la Tabla 3.1 el grado de velocidad y temperatura que puede soportar la FPGA, ya que algunos dispositivos tienen restricciones en cuanto a velocidad y temperatura.

Tabla 3.1: Grado de velocidad y rangos de temperaturas para las FPGAs de la serie 7 [Dig14].

		Speed Grade and Temperature Range									
Device Family	Devices	Commercial (C) 0°C to +85°C	Extended (E) 0°C to +100°C	Industrial (I) -40°C to +100°C							
Artix-7	All	-1, -2	-2L ⁽¹⁾ , -3	-1, -2							
Kintex-7	All	-1, -2	-2L, -3	-1, -2							
Virtex-7 T	XC7V585T	-1, -2	-2L, -3	-1, -2							
	XC7V2000T	-1, -2	-2L, -2G ⁽²⁾	-1							
Virtex-7 XT	XC7VX330T XC7VX415T	-1 -2	-2L, -3	-1, -2							
	XC7VX485T XC7VX550T XC7VX690T	_									
	XC7VX980T	-1, -2	-2L	-1							
	XC7VX1140T	-1, -2	-2L, -2G	-1							
Virtex-7 HT	All	-1, -2	-2L, -2G	N/A							

Notes:

- -2L speed grade offers reduced maximum power consumption. Artix-7 and Kintex-7 FPGAs are capable of operating at lower core voltage.
- -2G speed grade offers -2 FPGA logic performance with higher performance transceivers.

Las Virtex-7 dispone de distintos grados de velocidad: -3, -2 y -1, siendo -3 la configuración más alta. A continuación se muestra en la Tabla 3.2 los recursos disponibles para distintas familias de Virtex-7.

Tabla 3.2: Recursos de las distintas Virtex-7. En rojo se muestra la Virtex-7 XC7VX485T [Dig14].

Device ⁽¹⁾	Logio	Logic	Logio		rable Logic s (CLBs)	DSP	Bloc	k RAM Blo	ocks ⁽⁴⁾	CMTs	PCIe				XADC	Total I/O	Max	
	Logic Cells	Slices ⁽²⁾	Max Distributed RAM (Kb)	Slices ⁽³⁾	18 Kb	36 Kb	Max (Kb)	(5)	(6)	GTX	GTH	GTZ	Blocks	Banks ⁽⁷⁾	User I/O ⁽⁸⁾	SLRs ⁽⁹⁾		
XC7V585T	582,720	91,050	6,938	1,260	1,590	795	28,620	18	3	36	0	0	1	17	850	N/A		
XC7V2000T	1,954,560	305,400	21,550	2,160	2,584	1,292	46,512	24	4	36	0	0	1	24	1,200	4		
XC7VX330T	326,400	51,000	4,388	1,120	1,500	750	27,000	14	2	0	28	0	1	14	700	N/A		
XC7VX415T	412,160	64,400	6,525	2,160	1,760	880	31,680	12	2	0	48	0	1	12	600	N/A		
XC7VX485T	485,760	75,900	8,175	2,800	2,060	1,030	37,080	14	4	56	0	0	1	14	700	N/A		
XC7VX550T	554,240	86,600	8,725	2,880	2,360	1,180	42,480	20	2	0	80	0	1	16	600	N/A		
XC7VX690T	693,120	108,300	10,888	3,600	2,940	1,470	52,920	20	3	0	80	0	1	20	1,000	N/A		
XC7VX980T	979,200	153,000	13,838	3,600	3,000	1,500	54,000	18	3	0	72	0	1	18	900	N/A		
XC7VX1140T	1,139,200	178,000	17,700	3,360	3,760	1,880	67,680	24	4	0	96	0	1	22	1,100	4		
XC7VH580T	580,480	90,700	8,850	1,680	1,880	940	33,840	12	2	0	48	8	1	12	600	2		
XC7VH870T	876,160	136,900	13,275	2,520	2,820	1,410	50,760	18	3	0	72	16	1	6	300	3		

3.1.3 AMS

La tarjeta de evaluación que tiene la Virtex-7 es la AMS 101. La Figura 3.4 muestra el diagrama de bloques para visualizar el contenido de dicha tarjeta.

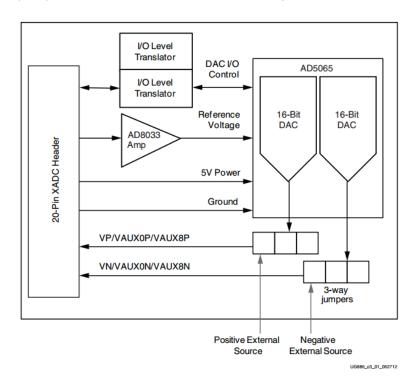


Figura 3.4: Diagrama de bloques del AMS 101 [Xil13].

La Tarjeta de Evaluación se divide en seis sub-bloques:

- Un conector XADC de 20 pines para la interfaz con la placa base de la FPGA.
- Dos DAC de 16 bits (Analog Devices DAC AD5065).
- Tiene un bloque de I/O para la interfaz del periférico serie (SPI) al DAC.
- Un buffer amplificador (AD8033).
- Pines para fuentes externas positivas o negativas.
- Tres jumpers para la fuente externa o no del DAC al conector del XADC.

Posteriomente, en 3.3 se explica con más detalle la configuración del módulo del XADC y del DAC aplicada a este trabajo.

3.2 Implementación del módulo emisor

El bloque del módulo emisor se ha diseñado para la implementación en la FPGA Virtex-5 LX50T. El diagrama de bloques general es el mostrado en la Figura 3.5.

EMISOR Retardos Modulado ζ1[1..E] Retardos Modulado Cod 2 ζ2[1..Ε] Entorno Emisor3 Recepto Explorado Retardos Cod N Modulado ζN[1..E]

Figura 3.5: Diagrama de bloques del emisor [G14].

Su diseño se puede dividir en cuatro partes:

- · Códificación.
- Modulación de los códigos.
- Aplicación de los distintos retardos.
- Bloque sumador de las señales.

Recordando las especificaciones del sistema vistas en el apartado 2.2.3 del Capítulo 2 se tiene:

- Frecuencia de portadora: $f_c = 40kHz$.
- Frecuencia de muestreo: $f_s = 400kHz$.
- Número de emisiones simultáneas: K = 8.
- Ceros a introducir en la semi-ventana libre de interferencias: W = 511.
- Longitud inicial de las parejas Golay: $L_0 = 512$.
- Semilla para la generación de las parejas Golay: Sem = 0.

Dicha FPGA dispone de los recursos expuestos en la Tabla 3.3.

Tabla 3.3: Recursos de la familia Virtex-5 [Xil09].

	Configurable Logic Blocks (CLBs)		DEDARE	Block RA		ock RAM Blocks		PowerPC	Endpoint Blocks for	Ethernet	Max RocketiO Transceivers ⁽⁶⁾		Total	Max	
Device	Array (Row x Col)	Virtex-5 Slices(1)	Max Distributed RAM (Kb)	Slices ⁽²⁾	18 Kb ⁽³⁾	36 Kb	Max (Kb)	CMTs ⁽⁴⁾	Processor Blocks	PCI Express	MACs ⁽⁵⁾	GTP	GTX	Banks ⁽⁸⁾	Use I/O/7
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
C5VLX50	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	80
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	80
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	80
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	1,20
XC5VLX20T	60 x 26	3,120	210	24	52	26	936	1	N/A	1	2	4	N/A	7	17
XC5VLX30T	80 x 30	4.800	320	32	72	36	1.296	2	N/A	1	4	8	N/A	12	36
XC5VLX50T	120 x 30	7,200	480	48	120	60	2,160	6	N/A	1	4	12	N/A	15	48
XC5VLX85T	120 x 54	12,960	840	48	216	108	3,888	6	N/A	1	4	12	N/A	15	48
XC5VLX110T	160 x 54	17,280	1,120	64	296	148	5,328	6	N/A	1	4	16	N/A	20	68
XC5VLX155T	160 x 76	24,320	1,640	128	424	212	7,632	6	N/A	1	4	16	N/A	20	68
XC5VLX220T	160 x 108	34,560	2,280	128	424	212	7,632	6	N/A	1	4	16	N/A	20	68
XC5VLX330T	240 x 108	51,840	3,420	192	648	324	11,664	6	N/A	1	4	24	N/A	27	96
XC5VSX35T	80 x 34	5,440	520	192	168	84	3,024	2	N/A	1	4	8	N/A	12	36
XC5VSX50T	120 x 34	8,160	780	288	264	132	4,752	6	N/A	1	4	12	N/A	15	48
XC5VSX95T	160 x 46	14,720	1,520	640	488	244	8,784	6	N/A	1	4	16	N/A	19	64
XC5VSX240T	240 x 78	37,440	4,200	1,056	1,032	516	18,576	6	N/A	1	4	24	N/A	27	96
XC5VTX150T	200 x 58	23,200	1,500	80	456	228	8,208	6	N/A	1	4	N/A	40	20	68
XC5VTX240T	240 x 78	37,440	2,400	96	648	324	11,664	6	N/A	1	4	N/A	48	20	68
XC5VFX30T	80 x 38	5,120	380	64	136	68	2,448	2	1	1	4	N/A	8	12	36
XC5VFX70T	160 x 38	11,200	820	128	296	148	5,328	6	1	3	4	N/A	16	19	64
XC5VFX100T	160 x 56	16,000	1,240	256	456	228	8,208	6	2	3	4	N/A	16	20	68
C5VFX130T	200 x 56	20,480	1,580	320	596	298	10,728	6	2	3	6	N/A	20	24	84
C5VFX200T	240 x 68	30,720	2,280	384	912	456	16,416	6	2	4	8	N/A	24	27	96

- Virtex-5 FPGA slices are organized differently from previous generations. Each Virtex-5 FPGA slice contains four LUTs and four flip-flops (previously it was two LUTs and two flip-flops.)

 Each DSP48E slice contains a 25 x 18 multiplier, an adder, and an accumulator.

 Block RAMs are fundamentally 36 Kbits in size. Each block can also be used as two independent 18-Kbit blocks.

 Each Clock Management Tille (CMT) contains two DCMs and one PLL.

- This table lists separate Ethernet MACs per device.

 RocketlO GTP transceivers are designed to run from 100 Mb/s to 3.75 Gb/s. RocketlO GTX transceivers are designed to run from 150 Mb/s to 6.5 Gb/s.
- This number does not include RocketIO transceivers.
- Includes configuration Bank 0.

Esta FPGA tiene un reloj de 100MHz y puede funcionar internamente hasta 200MHz. El DAC proporcionado por Digilent es el PmodDA2. Las características relevantes es que su frecuencia máxima de funcionamiento de su reloj serie es 30MHz y el número de bits por trama de datos es de 16 bits de los cuales 4 son de control y los 12 restantes son bits de datos. En él se hayan dos DAC121S101 [TX13] SPI de 12 bits de resolución.

Respecto a la implementación hardware, se ha partido de un modelo de códigos CSS [G14] y se ha modificado para códigos LS, teniendo en cuenta las especificaciones del sistema. En la Figura 3.6 se muestra el esquema del diagrama de bloques funcionales.

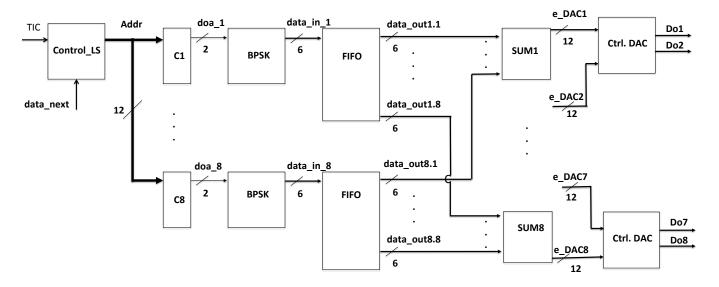


Figura 3.6: Diagrama de bloques implementados en la emisión.

El bloque **Control_LS** gestiona el acceso a las posiciones de memoria donde están los elementos del código a enviar. La señal TIC a 1 Hz indica el comienzo de una nueva emisión. La lectura se realiza simultáneamente para todos los códigos.

El módulo **BPSK** se encarga de realizar la modulación, usando para ello una memoria ROM con el símbolo de modulación. El bloque **FIFO** aplica los retardos asociados a cada código dependiendo del sector angular en el que se quiere deflectar el haz. El bloque **SUM** se encarga de sumar los elementos de cada código y se manda al controlador de DAC que corresponda. Por último, el módulo **Ctrl.DAC** se encarga de controlar la gestión del DAC a través de un bus SPI. La frecuencia máxima del reloj SPI es 30MHz y la conversión se realiza a una frecuencia de 1.6MHz. Cada controlador del DAC da salida a los dos DAC en paralelo que tiene el PmodDA2.

A continuación se muestra la simulación funcional del bloque emisor (Figura 3.7) donde se puede ver la emisión de un único código LS modulado con una BPSK y con los retardos correspondientes.

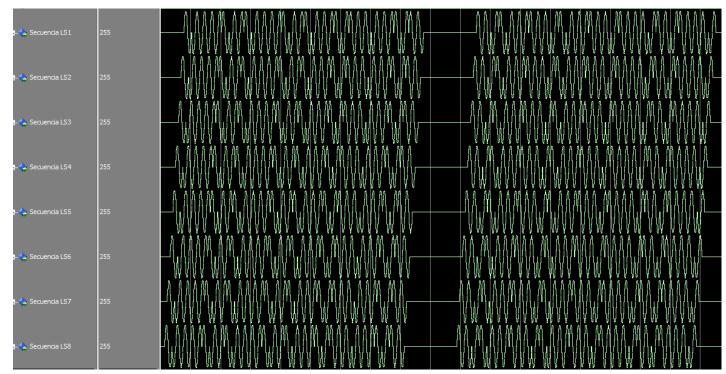


Figura 3.7: Simulación funcional del bloque emisor con las salidas del demodulador haciendo uso de códigos LS de longitud L=71.

En esta simulación se observa también los ceros introducidos en los códigos LS. En la siguiente simulación funcional (véase la Figura 3.8) se expone la salida de los DACs para cada una de las 8 señales. Ésta es el resultado de sumar cada elemento de los 8 códigos después de haber aplicado el correspondiente retardo para cada transductor.

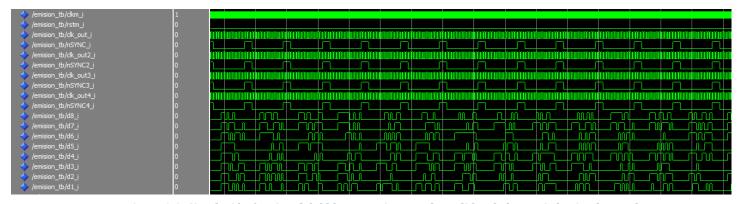


Figura 3.8: Simulación funcional del bloque emisor con las salidas de los DACs haciendo uso de códigos LS de longitud *L*=71.

3.3 Implementación del módulo receptor

En este apartado se explica lo referente al bloque receptor centrándose en el procesado de bajo nivel (véase la Figura 3.9).

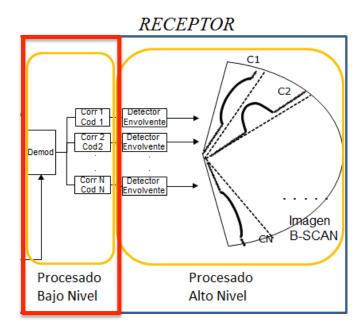


Figura 3.9: Diagrama de bloques del receptor [G14].

En el bajo nivel se lleva a cabo la adquisición de la señal a través de un XADC para ser demodulada y procesada mediante la correlación con el código correspondiente a cada sector angular.

3.3.1 Etapa de acondicionamiento de la señal

La FPGA Virtex-7 es usada para la etapa de recepción de la señal. Para entender mejor el diagrama de bloques que se va a implementar se tiene el esquema propuesto mostrado en la Figura 3.10 el cual incluye:

- **DCM**: módulo que reducirá el reloj externo de 200MHz a 100Mhz.
- EN: señal de *enable* que será usada en cada uno de los bloques y que tendrá un período de $3\mu s$.
- XADC: instanciación del XADC que se explicará a continuación.
- **Demo**: bloque demodulador de la señal recibida.
- Corr: bloque correlador de la señal demodulada.
- Ctrl DAC: controlador del DAC.
- AMS: tarjeta de Evaluación que incluye el XADC y los DAC.

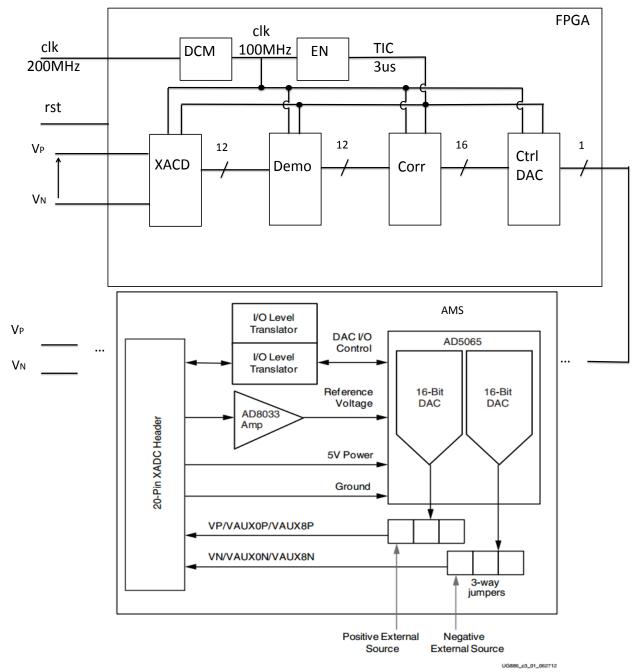


Figura 3.10: Diagrama de bloques de la FPGA y AMS 101.

Primitiva XADC

Previo a la etapa de demodulación es necesario convertir a digital la señal emitida por el modulo emisor, debiendo incorporar un ADC al sistema. En este caso Xilinx proporciona un componente cuya primitiva se muestra en la Figura 3.11.

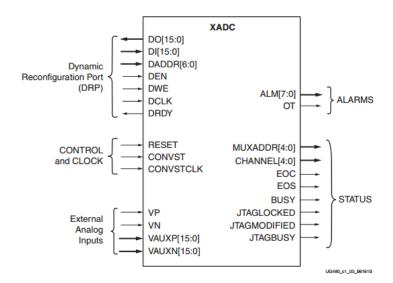


Figura 3.11: Puertos de la primitiva del XADC [Dig14].

A continuación se explica la configuración de registros del XADC necesarios para el funcionamiento del sistema, en la Figura 3.12.

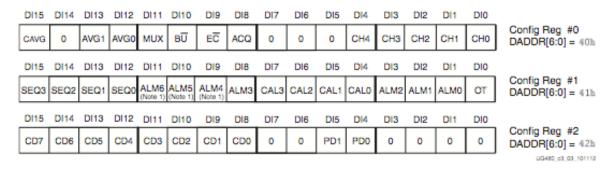


Figura 3.12: Definición de los bits de configuración de los registros del XADC [Dig14].

Para la simulación se crea un archivo en Matlab con el formato $\{TIME, V_P, V_N\}$, siendo TIME el tiempo en el que cambia el valor de las entradas analógicas; V_P es la tensión de la señal de entrada comprendida entre los valores descritos anteriormente dependiendo de si se quiere unipolar o bipolar, por lo que V_N y V_P son cero en la implementación en VHDL si se tiene dicho archivo. Para la implementación en un sistema real se tiene V_N a cero, siendo la diferencia de tensiones la entrada al XADC que en este caso es V_P .

Configuración del Registro 0:

La Figura 3.13 muestra los bits pertenecientes al registro 0 del XADC,

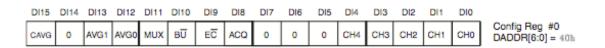


Figura 3.13: Bits de configuración del Registro 0 del XADC [Dig14].

donde solo es necesario configurar los siguientes bits:

- Modo unipolar: DI10=0.
- Activo por eventos: DI9=1. Se conecta la señal enable del sistema con la señal CONVST.

De los bits referidos al canal se selecciona aquel que visualice las entradas analógicas V_P y V_N para convertir (Tabla 3.4), siendo el valor convertido $V_P - V_N$.

ADC Channel	СН4	СНЗ	CH2	CH1	СН0	Description
0	0	0	0	0	0	On-chip temperature
1	0	0	0	0	1	V _{CCINT}
2	0	0	0	1	0	V _{CCAUX}
3	0	0	0	1	1	V _P , V _N – Dedicated analog inputs
4	U	U	1	U	U	V _{REFP} (1.25V) ^(*)
5	0	0	1	0	1	V _{REFN} (0V) ⁽¹⁾
6	0	0	1	1	0	V _{CCBRAM}
7	0	0	1	1	1	Invalid channel selection
8	0	1	0	0	0	Carry out an XADC calibration
9–12						Invalid channel selection
13	0	1	1	0	1	V _{CCPINT} ⁽³⁾
14	0	1	1	1	0	V _{CCPAUX} ⁽³⁾
15	0	1	1	1	1	V _{CCO_DDR} ⁽³⁾
16	1	0	0	0	0	VAUXP[0], VAUXN[0] – Auxiliary channel 0
17	1	0	0	0	1	VAUXP[1], VAUXN[1] – Auxiliary channel 1

Tabla 3.4: Selección del canal del ADC [Dig14].

Configuración del Registro 1:

Los bits pertenecientes al registro 1 se muestran en la Figura 3.14.

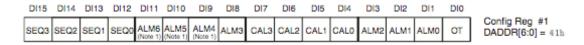


Figura 3.14: Bits de configuración del Registro 1 del XADC [Dig14].

En este registro se elegirá el modo de canal simple. Este modo se activa cuando los bits desde SEQ3 a SEQ0 en el control de registros 41h están a 0011. Para este modo se debe seleccionar el canal de conversión de analógico a digital escribiendo los bits de los canales CH4 a CH0 en el control de registro 40h. También se ha de configurar el modo como entrada analógica en $B\overline{U}$. En la Tabla 3.5 se muestra la configuración para este modo.

Tabla 3.5: Ajustes en la operación de la secuencia [Dig14].

SEQ3	SEQ2	SEQ1	SEQ0	Function
0	0	0	0	Default mode
0	0	0	1	Single pass sequence
0	0	1	0	Continuous sequence mode
0	0	1	1	Single channel mode (sequencer off)
0	1	X	X	Simultaneous sampling mode
1	0	X	X	Independent ADC mode
1	1	X	Х	Default mode

Modo unipolar XADC:

Mediante el *datasheet* de la primitiva del XADC se establece que el modo de trabajo comprende márgenes muy pequeños de tensiones analógicas de entrada, siendo en este caso de un voltio pico-pico. Existen dos modos de trabajo para este componente, diferenciando entre unipolar y bipolar, eligiendo para el trabajo modo unipolar, comprendido entre cero y un voltio. Esta característica tan especial hace que se limite mucho el rango de entrada a la FPGA, teniendo que reducir previamente la señal para su correcto procesamiento.

Existen dos formas de conseguir que la señal de entrada quede comprendida entre los valores citados anteriormente:

- 1- Incorporar una etapa previa mediante circuito externo a la FPGA que reduzca a los niveles necesarios.
 - 2- Configurar la etapa de emisión para que a su salida emita en este rango de valores.

Para convertir los valores de analógico a hexadecimal se usa la gráfica descrita en la Figura 3.15.

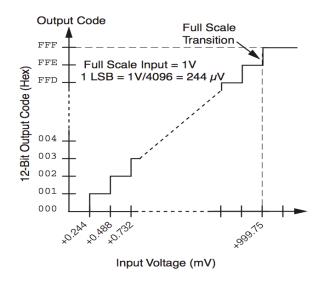


Figura 3.15: Función de transferencia del modo unipolar [Dig14].

La fórmula para obtener el valor hexadecimal es:

$$valor_hexa = \frac{(V_P - V_N)}{244\mu V}$$
 (4.1)

Modo activo por eventos:

En la Figura 3.16 se muestra el modo por eventos. En este modo de operación, el muestreo y el proceso de conversión están inicializados mediante una señal *trigger* llamada CONVST (*convert start*), usada cuando es necesario tener un control preciso sobre el muestreo.

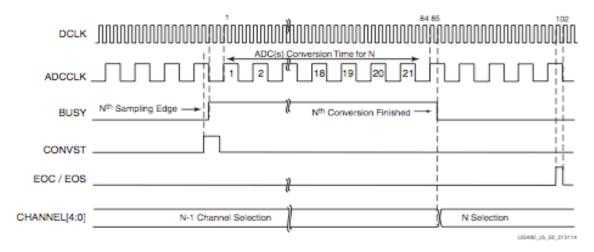


Figura 3.16: Modo por eventos [Dig14].

La señal de reloj DCLK deberá estar siempre presente cuando se hace uso de este modo. Si no estuviera presente, el XADC cambiaría a modo de tiempo continuo haciendo uso del reloj interno. La transición bajo-alto en la señal CONVST define el instante de reloj exacto para el canal analógico de entrada seleccionado expuesto en la Tabla 3.4, por lo que el tiempo de adquisición seleccionado por el canal está controlado por esta señal CONVST. La señal BUSY pasa a estado alto en el siguiente flanco de subida del reloj DCLK.

Después de que la señal de entrada analógica ha sido muestreada en el flanco de subida de CONVST, comienza la conversión en el siguiente flanco de subida del ADCCLK. Una vez iniciada la conversión mediante CONVST no es posible interrumpirla o empezar una nueva conversión hasta que BUSY no esté bajo. Cuando la señal BUSY pasa a flanco de bajada, habrá que esperar 16 ciclos de DCLK para comenzar una nueva conversión.

DAC:

Para la gestión del DAC se hace uso del siguiente cronograma mostrado en la Figura 3.17, donde cabe destacar que son necesarios 32 bits de entrada para el inicio de la conversión, haciendo necesario expandir el bit de signo obtenido a la salida del correlador.

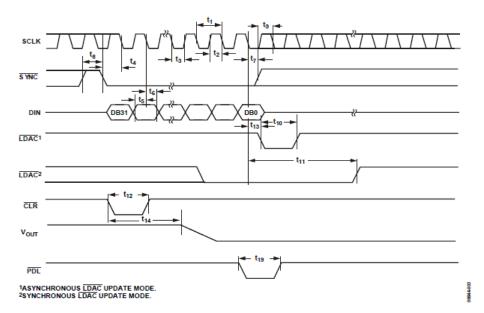


Figura 3.17: Operación se escritura en serie del AD5065 [An08].

La Tabla 3.6 muestra las especificaciones temporales del AD5065.

Tabla 3.6: Características temporales del AD5065 [An08].

Parameter	Symbol	Min	Тур	Max	Unit
SCLK Cycle Time	tı1	20			ns
SCLK High Time	t ₂	10			ns
SCLK Low Time	t ₃	10			ns
SYNC to SCLK Falling Edge Setup Time	t ₄	16.5			ns
Data Setup Time	ts	5			ns
Data Hold Time	t ₆	5			ns
SCLK Falling Edge to SYNC Rising Edge	t ₇	0		30	ns
Minimum SYNC High Time (Single Channel Update)	t ₈	2			μs
Minimum SYNC High Time (All Channel Update)	t ₈	4			μs
SYNC Rising Edge to SCLK Fall Ignore	t ₉	17			ns
LDAC Pulse Width Low	t ₁₀	20			ns
SCLK Falling Edge to LDAC Rising Edge	tıı	20			ns
CLR Pulse Width Low	t12	10			ns
SCLK Falling Edge to LDAC Falling Edge	t ₁₃	10			ns
CLR Pulse Activation Time	t ₁₄	10.6			μs
SCLK Rising Edge to SDO Valid	t ₁₅ ^{2, 3}			22	ns
SCLK Falling Edge to SYNC Rising Edge	t16 ²	5		30	ns
SYNC Rising Edge to SCLK Rising Edge	t ₁₇ 2	8			ns
SYNC Rising Edge to LDAC/CLR/PDL Falling Edge (Single Channel Update)	t ₁₈ ²	2			μs
SYNC Rising Edge to LDAC/CLR/PDL Falling Edge (All Channel Update)	t ₁₈ ²	4			μs
PDL Minimum Pulse Width	t ₁₉	20			ns

 $^{^1}$ Maximum SCLK frequency is 50 MHz at V_{00} = 4.5 V to 5.5 V. Guaranteed by design and characterization; not production tested. 2 Daisy-chain mode only. 3 Measured with the load circuit of Figure 2. th determines the maximum SCLK frequency in daisy-chain mode.

En este apartado se específica que la señal SYNC perteneciente al DAC se activa con el enable del sistema completo. Realizando un estudio de tiempos para comprobar la frecuencia máxima de trabajo se obtiene que:

$$Tmin = t8 + n^{0} de bits * t1$$
 (4.2)

$$Tmin = 2000ns + 32 * 20ns = 2640ns$$
 (4.3)

Entonces la frecuencia máxima es aproximadamente 378KHz, por lo que eligiendo un período superior el sistema seguirá las especificaciones anteriores. En este caso se elige un período de tres microsegundos (333KHz), suficiente para que el resto de bloques funcionen correctamente.

3.3.2 Procesamiento de bajo nivel: Demodulador y Correlador

Como se ha comentado anteriormente, respecto al proceso de recepción, este trabajo se centra en el procesamiento de bajo nivel, es decir, la parte correspondiente a la demodulación de los códigos recibidos y la correlación de los mismos. El bloque demodulador recupera que lo que envía el emisor y dicha señal es correlada con los correladores ELSC, donde cada uno de ellos corresponde a un sector angular.

El diseño e implementación hardware se lleva a cabo mediante el lenguaje en VHDL atendiendo a las siguientes especificaciones del sistema, las cuales están expuestas en la Tabla 3.7.

ESPECIFICACIONES GENERALES DEL SISTEMA RECEPTOR							
Genéricos	Definición	Valor					
nEtapa	Número de iteraciones del correlador Golay inicial ($L_0=2^{nEtapa}$)	5					
nK	Número de secuencias del conjunto LS ($K=2^{nK}$)	3					
Wo	Número de ceros en la semi-ventana libre de interferencias	31					
nSBMuestreo	Factor de sobremuestreo en la adquisición de la señal, es decir, número de muestras por símbolo	10					
semilla1	Semilla de la secuencia Golay 1. Comprende desde (nEtapa-1 a 0)	00001					
semilla2	Semilla de la secuencia Golay 2. Comprende desde (nEtapa-1 a 0)	00000					

Tabla 3.7: Especificaciones del sistema receptor.

Como se explicó en el apartado 2.3.2 se necesita implementar ocho veces el correlador LS (véase la Figura 2.14). A continuación se muestran los valores correspondientes de la matriz de Hadamard y del vector Π de cada uno de los correladores mostrados en la Tabla 3.8 ya que estos valores varían dependiendo del correlador.

La longitud de los códigos que se desea implementar (*L*=4607) es de mayor tamaño a la expuesta en este capítulo puesto que, como se explicará posteriormente en el Capítulo 4 no puede llevarse a cabo dicha implementación por falta de recursos de la FPGA Virtex-7.

Tabla 3.8: Especificaciones del sistema receptor para los ocho correladores ELSC.

	ESP	PECIFICACIONES DEL SISTEMA RECEPTOR: CORRELADOR LS	
		Definición	Valor
Correlador 1	had_1	Valores de la primera columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1111
	pi_1	Valores del vector pi.	0011
Correlador 2	had_2	Valores de la segunda columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1010
	pi_1	Valores del vector pi.	0011
Correlador 3	had_3	Valores de la tercera columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1100
	pi_1	Valores del vector pi.	0011
Correlador 4	had_4	Valores de la cuarta columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1}) - 1 \ a \ 0)$	1001
	pi_1	Valores del vector pi.	0011
Correlador 5	had_1	Valores de la primera columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1}) - 1 \ a \ 0)$	1111
	pi_2	Valores del vector pi.	1100
Correlador 6	had_2	Valores de la segunda columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1010
	pi_2	Valores del vector pi.	1100
Correlador 7	had_3	Valores de la tercera columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1100
	pi_2	Valores del vector pi.	1100
Correlador 8	had_4	Valores de la cuarta columna correspondiente de la matriz de Hadamard. Comprende desde $((2^{nK-1})-1\ a\ 0)$	1001
	pi_2	Valores del vector pi.	1100

A continuación se describe cada módulo por separado, donde se detallan los puertos de entrada y salida, así como otros bloques necesarios que se han diseñado para el correcto funcionamiento del sistema completo.

Demodulador

Respecto al demodulador, como se ha explicado anteriormente en el apartado 2.3.1 del Capítulo 2 este bloque es el encargado de recuperar la información proveniente del modulador BPSK. Sus puertos de entrada y salida son los explicados a continuación.

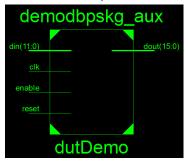


Figura 3.18: Puertos del bloque demodulador.

- din: datos de entrada provenientes de la emisión.
- clk: reloj de 100 MHz proveniente del DCM.
- **enable**: señal de enable a frecuencia de $3\mu s$ que activa la demodulación de los datos.
- reset: señal de reset del demodulador.
- dout: datos de salida del demodulador que irán al correlador LS.

DCM

El reloj de 100MHz usado proviene del bloque DCM (*Digital Clock Manager*). Dicho bloque es una primitiva de la FPGA Virtex-7, que se encarga de multiplicar o dividir una frecuencia de reloj entrante. En este caso, la señal de reloj se dividirá, ya que la placa tiene un reloj de 200MHz, para que la resultante tenga la frecuencia de 100MHz que se necesita. Las señales del bloque son las siguientes,

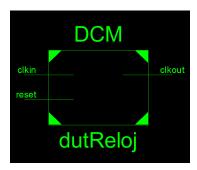


Figura 3.19: Puertos del bloque del DCM.

- **clkin**: reloj de entrada de 200MHz.
- reset: señal de reset del DCM.
- clkout: reloj de 100MHz que usan todos los bloques de la FPGA.

Enable

El bloque enable usado para crear dicha señal, que será un TIC que habilitará la entrada de datos, y deberá tener una frecuencia mínima de $3\mu s$ ya que es el tiempo mínimo necesario de conversión del XADC.

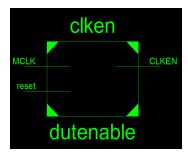


Figura 3.20: Puertos del bloque del TIC Enable.

- MCLK: es el reloj master de 100MHz usado como reloj interno en la placa.
- reset: señal de reset del bloque enable.
- **CLKEN**: es el reloj de salida de $3\mu s$ generado por el bloque.

En la siguiente figura de simulación se puede apreciar el período de duración de la señal TIC de $3\mu s$ que tiene una duración de un ciclo de reloj del reloj de 100MHz (10ns) mostrado en la Figura 3.21.

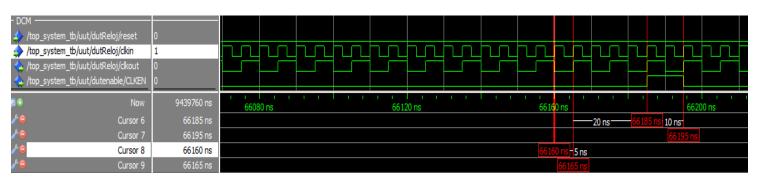


Figura 3.21: Simulación funcional de la señal de TIC enable.

Analizando la señal de TIC enable, el tiempo que debe transcurrir entre dos flancos de subida debe ser como mínimo de $3\mu s$ (3000ns), ya que, según las especificaciones del XADC, este es el tiempo mínimo de conversión de la señal (véase la Figura 3.22)

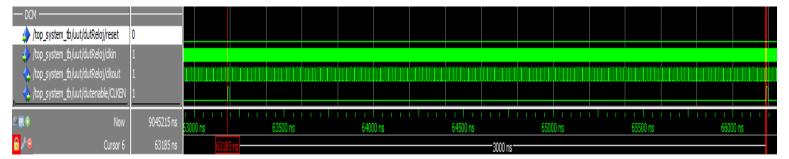


Figura 3.22: Simulación temporal de donde se comprueba la duración de la señal TIC.

ESSC

La señal de salida que se ha generado en el demodulador irá al correlador. Éste está compuesto por el bloque ESSC y el bloque correlador, ambos explicados en el apartado 2.3.2. Respecto al bloque ESSC, encargado de la correlación entre la señal de entrada $r[\tau]$ y las secuencias S_0 y S_1 .

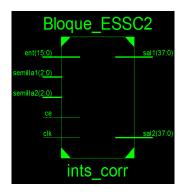


Figura 3.23: Puertos del bloque ESSC.

- ent: bus de datos de entrada al ESSC provenientes del demodulador.
- semilla1: señal que representa al conjunto de los coeficientes de generación de la secuencia Golay 1 y es dependiente del número de etapas. Para tres etapas el valor de semilla1 es 000.
- **semilla2**: señal que representa al conjunto de los coeficientes de generación de la secuencia Golay 2 y es dependiente del número de etapas. Para tres etapas el valor de semilla2 es 001.
- ce: señal de enable que habilita la entrada de datos.
- clk: reloj de 100MHz.
- sal1: bus de datos de salida del 2-ESSC correspondiente a la secuencia S_0 .
- sal2: bus de datos de salida del 2-ESSC correspondiente S_1 .

Correlador LS

Analizando el bloque del $\,$ correlador LS, explicado en la sección 2.3.2 del Capítulo 2 éste es el encargado de introducir los retardos y las señales de la matriz de Hadamard y el vector Π .

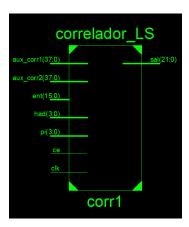


Figura 3.24: Puertos del bloque del correlador LS.

- **aux_corr1**: señal auxiliar proveniente del correlador 2-ESSC de la secuencia S_0 .
- **aux_corr2**: señal auxiliar proveniente del correlador 2-ESSC de la secuencia S_1 .
- ent: bus de entrada proveniente del demodulador que incluye los retardos.
- had: valores de la columna correspondiente a la matriz de hadamard.
- **pi**: valores del vector pi.
- ce: clock enable.
- clk: reloj del sistema a 100MHz.
- sal: bus de salida del correlador.

A continuación se muestra la simulación funcional, en la Figura 3.25, todo el bloque que constituye el demodulador y el correlador, donde se puede ver que a la salida de este último bloque se tienen los picos de auto-correlación o la correlación cruzada.

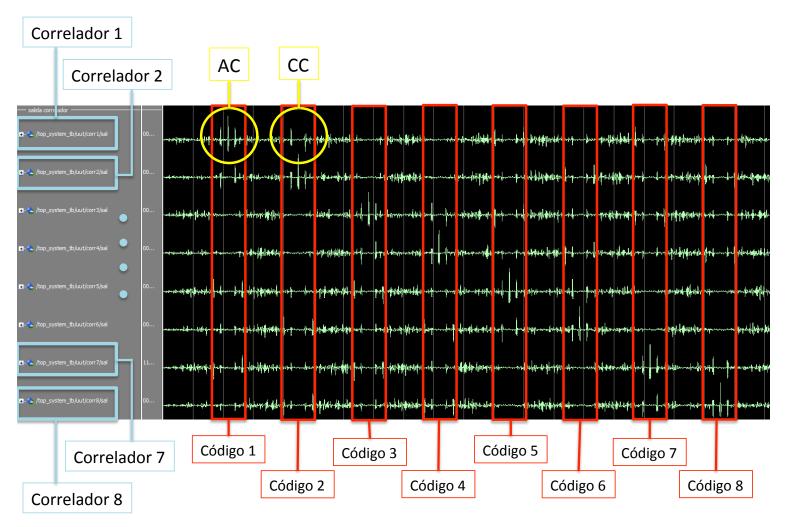


Figura 3.25: Simulación funcional del sistema completo para una secuencia de ocho códigos con longitud *L*=71 concatenados y correlados con cada uno de los ocho correladores LS.

Se ha hecho la simulación con los ocho códigos LS concatenados en un único código y se ha realizado la correlación con cada uno de los ocho correladores ELSC optimizados, por lo que se puede ver que en una primera correlación se produce AC con el primer correlador y CC con los demás, ya que es el primer código LS el que se ha analizado en ese momento. En una segunda correlación con el segundo código se observa AC con el segundo correlador y CC con el resto de los siete correladores, y así sucesivamente hasta analizar todo el código LS emitido.

En la Figura 3.25 se marca la señal de cada correlador así como la fracción correspondiente a cada código emitido y la AC y CC para poder visualizarlo más fácilmente.

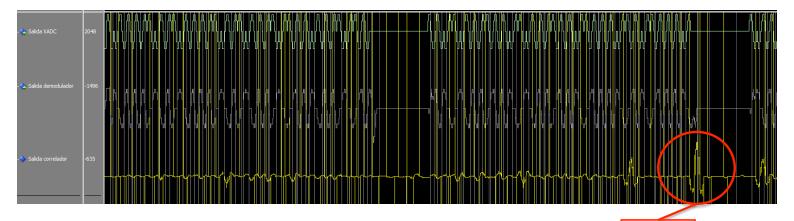
Capítulo 4 Resultados

En este capítulo se exponen las simulaciones temporales llevadas a cabo en el diseño descrito en capítulos anteriores así como el consumo de recursos. En la sección 4.1 se analiza la simulación temporal del sistema completo, emisor y receptor, expuesto en apartados anteriores y en la sección 4.2 se analiza el consumo de recursos de dichos módulos, pero en el bloque receptor teniendo en cuenta los dos tipos de diseño que se han estudiado, con un correlador LS eficiente optimizado y otro sin optimizar.

Al realizar la implementación en la Virtex-7 en el bloque receptor para los códigos LS de longitud $L=4607\ bits$ se constató que no es posible su implementación con el correlador LS optimizado, visto en el apartado 2.3.2, por falta de recursos de la FPGA, por lo que se ha realizado un estudio para diferentes longitudes de código de menor tamaño, así como un estudio del consumo de recursos de los dos tipos de correladores LS que se han visto.

4.1 Simulación temporal

En este apartado se muestra el resultado del sistema total planteado anteriormente con el emisor y el receptor en una simulación temporal. Dicha simulación se ha llevado a cabo haciendo uso del código de longitud más pequeña (*L*=71), ya que se requiere mucho tiempo para obtener resultados. En la Figura 4.1 se muestran, para códigos cortos, los picos de correlación para un único código usando un único correlador LS. Al verificar el correcto funcionamiento del sistema total, estaría validado el diseño para un sistema con un código de mayor longitud, en este caso de 287 bits, ya que como se explica a continuación en el apartado 4.2 de resultados, ésta es la longitud máxima que puede llegar a implementarse en la FPGA Virtex-7.



Pico AC

Figura 4.1: Simulación temporal del sistema total para un código LS de longitud 71 haciendo uso de un único correlador ELSC.

En la Figura 4.1 están representadas tres señales: la señal de entrada proveniente de la emisión que es capturada por el XADC, dicha señal demodulada, y por último la correlación representada en color amarillo de la señal demodulada con el primer correlador, es decir, el resultado total es una auto-correlación. En dicho resultado también pueden apreciarse los *glitches* propios de una simulación temporal. A continuación, en la Figura 4.2, se muestra la imagen ampliada de la simulación temporal expuesta en la Figura 4.1, donde se ve con mayor claridad la función de auto-correlación para un único correlador ELSC correspondiente a un sector angular.

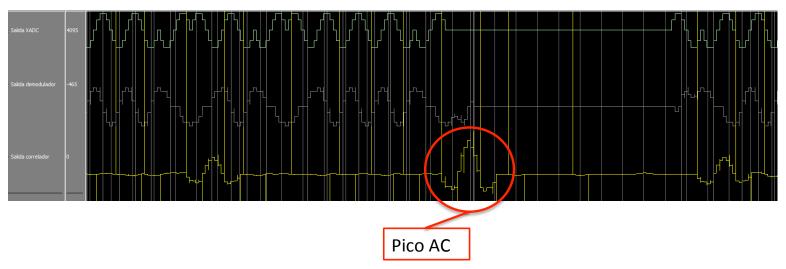


Figura 4.2: Imagen ampliada de la simulación temporal del sistema total para un código LS de longitud 71 haciendo uso de un único correlador ELSC.

En la Figura 4.3 se muestra la simulación en Matlab del mismo código usado en la simulación temporal. Se observa por lo tanto que el sistema cumple con las especificaciones diseñadas.

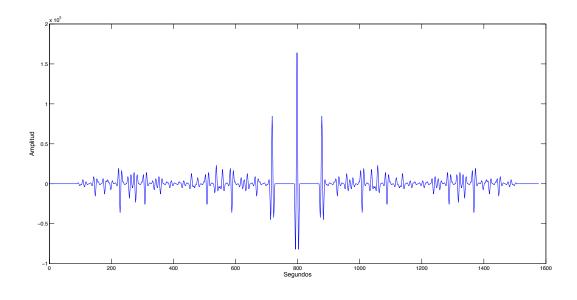


Figura 4.3: Simulación en Matlab de la auto-correlación para un código LS de longitud 71.

4.2 Consumo de recursos

4.2.1 Emisor

Para el bloque emisor, se estudia el consumo de recursos del diseño para la FPGA Virtex-5, independientemente de la longitud de código que se tenga ya que los datos están guardados en memoria y sólo afectaría a este bloque, por lo que el consumo de recursos depende directamente del diseño. La longitud del código estudiado para el emisor es 287 bits. En la Tabla 4.1 se muestra el consumo de recursos y porcentaje de ocupación, donde se observa que la mayor parte de los recursos son para los Slice LUTs usados como bloques lógicos.

Tabla 4.1: Consumo de recursos y porcentaje de ocupación del bloque Emisor para la longitud de 287 bits para la Virtex-5.

Consumo de Recursos del Emisor							
Número de Slices Ocupados	IOBs (480	Número de Slice Registers	Número de Slice LUTs usados como lógica	Número de Slice LUTs usados para memoria			
(7200 Disponibles)	Disponibles)	(280800 Disponibles)	(28800 Disponibles)	(7680 Disponibles)			
1774 (24%)	19 (3%)	2 610 (9%)	4 141 (14%)	396 (5%)			

4.2.2 Receptor

Debido a que para una longitud de $L=4607\ bits$ no es posible la implementación en la FPGA por falta de recursos, como se ve más adelante, en este apartado se analiza la recepción de la señal para distintas longitudes de códigos LS en la Virtex-7. En las tablas siguientes se tiene una comparativa del consumo de recursos para las distintas longitudes analizando el correlador LS eficiente sin optimizar y el correlador LS optimizado, expresado en número de recursos y porcentaje usado de esos recursos.

Consumo de recursos por bloque

Inicialmente, se expone el consumo de recursos del módulo receptor analizando cada bloque que lo compone por separado y teniendo en cuenta la longitud del código bajo análisis. En la Tabla 4.2 se muestra el consumo de recursos del bloque demodulador.

Tabla 4.2: Consumo de recursos y porcentaje de ocupación del bloque Demodulador para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

DEMODULADOR							
Número de Slices IOBs Número de Slice UTs Número de Slice LUT							
Ocupados (75900 Disponibles)	(700 Disponibles)	Registers (607200 Disponibles)	usados como lógica (303600 Disponibles)	usados para memoria (303600 Disponibles)			
66 (1%)	31 (4%)	189 (1%)	158 (1%)	0 (0%)			

A continuación se realiza una comparación entre el correlador no optimizado y optimizado, ambos analizados en la sección 2.3.2 donde se tenían dos bloques: un bloque 1 que contenía los correladores 2-ESSC y un bloque 2 que se encargaba de introducir los retardos y la matriz de hadamard.

Es necesario recalcar que el consumo de recursos se obtiene para un solo módulo de este correlador, diferenciándose en el caso del correlador optimizado los dos bloques mencionados: el ESSC y el bloque de retardos; y para el caso del no optimizado estos bloques se consideran como un único módulo que contiene a ambos, por lo que al estar divididos no se aprecia una mejora del consumo de recursos, pero en el caso de multiplicarlos por ocho veces, que es el número de correladores necesarios para este sistema, el consumo total de recursos difiere bastante.

En la Tabla 4.3 se estudia el consumo de recursos del correlador no optimizado para un único correlador.

Tabla 4.3: Consumo de recursos y porcentaje de ocupación del Correlador sin optimizar (ESSC+Bloque de retardos) para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

	CORRELADOR SIN OPTIMIZAR								
LONGITUD L=K·Lo+W	Número de Slices Ocupados (75900 Disponibles)	(700 Disponibles)	Número de Slice Registers (607200 Disponibles)	Número de Slice LUTs usados como lógica (303600	Número de Slice LUTs usados para memoria (130800				
				Disponibles)	Disponibles)				
L=71	3286 (4%)	54 (7%)	745 (1%)	723 (1%)	3192 (2%)				
L=143	6380 (8%)	57 (8%)	851 (1%)	859 (1%)	6692 (5%)				
L=287	10987 (14%)	60 (8%)	961 (1%)	1009 (1%)	13994 (10%)				
L=575	23715 (23%)	63 (9%)	1075 (1%)	1157 (1%)	29236 (22%)				

En la Tabla 4.4 y Tabla 4.5 se muestra el consumo de recursos del correlador ELSC optimizado que se divide en el bloque ESSC y el bloque encargado de los retardos y la matriz de hadamard, analizando para ello distintas longitudes de código.

Tabla 4.4: Consumo de recursos y porcentaje de ocupación del bloque ESSC para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

	ESSC							
LONGITUD	Número de	IOBs	Número de Slice	Número de Slice	Número de Slice			
L=K·Lo+W	Slices	(700	Registers	LUTs usados como	LUTs usados para			
	Ocupados	Disponibles)	(607200 Disponibles)	lógica	memoria			
	(75900			(303600 Disponibles)	(130800			
	Disponibles)				Disponibles)			
L=71	263 (1%)	100 (14%)	210 (1%)	426 (1%)	152 (1%)			
L=143	375 (1%)	106 (15%)	288 (1%)	584 (1%)	292 (1%)			
L=287	584 (1%)	112 (16%)	370 (1%)	750 (1%)	554 (1%)			
L=575	1255 (1%)	118 (16%)	456 (1%)	924 (1%)	1076 (1%)			

Tabla 4.5: Consumo de recursos y porcentaje de ocupación de un único Correlador Optimizado correspondiente al bloque de retardos y matriz de Hadamard. para la Virtex-7 VC707 XC7VX485T-2FFG1761C

	$Bloque_{retardo}$							
LONGITUD	Número de	IOBs	Número de Slice	Número de Slice	Número de Slice			
L=K·Lo+W	Slices	(700	Registers	LUTs usados como	LUTs usados para			
	Ocupados	Disponibles)	(607200 Disponibles)	lógica	memoria			
	(75900			(303600	(130800			
	Disponibles)			Disponibles)	Disponibles)			
L=71	1894 (2%)	108 (15%)	535 (1%)	395 (1%)	3040 (2%)			
L=143	1805 (2%)	113 (16%)	563 (1%)	415 (1%)	6400 (4%)			
L=287	3664 (4%)	118 (16%)	591 (1%)	435 (1%)	13439 (10%)			
L=575	17703 (23%)	123 (17%)	619 (1%)	455 (1%)	28160 (21%)			

En el sistema completo es necesario correlar la señal de salida del demodulador con cada uno de los ocho códigos LS perteneciente a cada sector angular. En la Tabla 4.6 se realiza un cálculo aproximado del consumo de recursos que conlleva la implementación de este sistema, analizando únicamente el número de Slices ocupados.

Tabla 4.6: Comparación del consumo de recursos a nivel de Slices ocupados para los ocho correladores sin optimizar y optimizados para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

	8 · CORRELADOR SIN OPTIMIZAR	$ESSC + 8 \cdot Bloque_{retardo}$		
LONGITUD	Número de Slices Ocupados	LONGITUD Número de Slices Ocupados		
L=K·Lo+W		L=K·Lo+W		
L=71	26288	L=71	15152	
L=143	51040	L=143	14440	
L=287	87896	L=287	29312	
L=575	189720	L=575	141624	

A la luz de los resultados obtenidos en la Tabla 4.6 se observa una clara reducción del consumo de recursos que conlleva la implementación del correlador optimizado frente al correlador sin optimizar, consiguiéndose un valor muy inferior de Slices. Recordar que los valores obtenidos en la Tabla 4.6 son cálculos aproximados, por lo que a la hora de realizar la implementación de los correladores en el sistema total no se llegará a la ocupación de estos recursos.

Consumo de recursos del sistema completo

La comprobación de la efectividad del sistema receptor total al usar un correlador LS eficiente optimizado de otro sin optimizar puede apreciarse en las tablas siguientes. Para el sistema completo usando el ELSC sin optimizar se tiene el análisis del consumo de recursos mostrado en la Tabla 4.7.

Tabla 4.7: Consumo de recursos y porcentaje de ocupación del Sistema Receptor con el Correlador sin optimizar para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

Sistema receptor con ELSC sin optimizar							
LONGITUD	Número de Slices	IOBs	Número de	Número de Slice	Número de Slice		
L=K·Lo+W	Ocupados	(700 Disponibles)	Slice Registers	LUTs usados como	LUTs usados		
	(75900 Disponibles)		(607200	lógica	para memoria		
			Disponibles)	(303600	(130800		
				Disponibles)	Disponibles)		
L=71	20721 (27%)	258 (36%)	5551 (1%)	28762 (9%)	25536 (19%)		
L=143	29617 (35%)	258 (36%)	6367 (1%)	57520 (18%)	53536 (40%)		

La longitud de 287 bits sobrepasa el consumo de recursos en cuanto a Slice LUTs usados para memoria por lo que la longitud máxima que se podría implementar es de 143 bits. El sistema completo del receptor usando el ELSC con la optimización se muestra en la Tabla 4.8.

Tabla 4.8: Consumo de recursos y porcentaje de ocupación del Sistema Receptor con el Correlador optimizado para la Virtex-7 VC707 XC7VX485T-2FFG1761C.

Sistema receptor con ELSC optimizado					
LONGITUD	Número de Slices	IOBs	Número de	Número de Slice	Número de Slice
L=K·Lo+W	Ocupados	(700 Disponibles)	Slice Registers	LUTs usados como	LUTs usados
	(75900 Disponibles)		(607200	lógica	para memoria
			Disponibles)	(303600	(130800
				Disponibles)	Disponibles)
L=71	14684 (19%)	258 (36%)	3249 (1%)	1445 (1%)	24544 (18%)
L=143	26786 (35%)	258 (36%)	3679 (1%)	1619 (1%)	51570 (39%)
L=287	32793 (43%)	258 (36%)	3924 (1%)	1761 (1%)	108151 (82%)

En la Tabla 4.7 se puede ver que, respecto al consumo de recursos del sistema total usando el ELSC sin optimizar, la FPGA se queda sin recursos para una longitud de 287 bits en los Slice LUTs para memoria, por lo que la longitud máxima que se puede llegar a usar para la recepción con la FPGA Virtex-7 es 143 bits.

Observando la Tabla 4.8, respecto al consumo de recursos haciendo uso del ELSC optimizado, la longitud que sobrepasa los recursos es de 575 Slice LUTs usados para memoria, por lo que la longitud máxima que podría implementarse es de 287 bits.

Hay una clara ventaja al comparar el consumo de recursos del sistema total con el ELSC optimizado y sin optimizar, siendo el sistema con el correlador optimizado mucho más ventajoso al usar menos recursos y poder implementar una longitud mayor que el correlador sin optimizar. Por lo que se puede concluir que optimizando más el correlador se podría llegar a implementar longitudes mucho mayores. También hay un incremento importante respecto a la evolución en las FPGAs de Xilinx. Existe un progresivo aumento de la capacidad y grandes expectativas de crecimiento en cuanto a la mejora de las prestaciones, por lo que en la próxima generación de esta familia de FPGAs se podrá implementar una longitud de códigos mucho mayor de la diseñada hasta el momento

La zona de análisis para una longitud de L=287 bits tiene un alcance de 24cm, por tanto es necesario seguir trabajando sobre arquitecturas que permitan una mayor optimización de las operaciones de correlación, de modo que puedan obtenerse alcances mayores. Algunos trabajos recientes como [GGUP10] ya están realizando propuestas de optimización al respecto. De conseguir implementar la longitud inicial estimada, de L=4607 bits la distancia que podría cubrirse es de 3.91m, con mayor inmunidad al ruido que si se utilizasen códigos CSS de similar longitud.

Capítulo 5 Conclusiones y trabajos futuros

En este trabajo se presentan los resultados del diseño y modificación del sistema de procesamiento de un *phased array* ultrasónico basado en técnicas codificación, mediante la utilización de un correlador eficiente optimizado.

Se estudia la fusión de la técnica *Phased Array* con CDMA, ya que es muy útil para explorar distintos sectores angulares de forma simultánea. Atendiendo a las técnicas de codificación, se profundiza en las propiedades y características de los códigos LS frente a los CSS utilizados en trabajos anteriores, puesto que estos códigos tienen una mayor ganancia de proceso y unas propiedades de correlación que las hacen ventajosas respecto a otros códigos.

Para una correcta implementación del sistema, se realizó un estudio detallado de cada uno de los bloques que lo componen, analizando para ello el sistema emisor y el receptor y las diferentes etapas que recorre la señal. El estudio se centró en el procesamiento de bajo nivel en cuanto al sistema receptor. Igualmente, se ha estudiado la plataforma de procesamiento Virtex-7 de Xilinx, analizando sus características y configuración necesarias para el sistema diseñado.

Se ha diseñado un sistema que permite una mejora del correlador eficiente, el cual posibilita la implementación de una longitud mayor de códigos LS. La ventaja que ofrece el aumento de la longitud es que éste permite llegar a distancias mayores de análisis del entorno. Mediante la optimización del correlador se capacita al sistema para alcanzar una mayor distancia de exploración.

Para un mejor análisis, se estudian los resultados para la implementación en el sistema total de distintas longitudes de códigos LS, exponiéndose resultados del consumo de recursos de las distintas longitudes y de los dos tipos de correladores analizados en el trabajo, con la optimización y sin ella, y cómo dicha optimización afecta al sistema completo.

Las líneas futuras de investigación que podrían derivar de dicho trabajo serían el estudiar el procesamiento de alto nivel, que consiste en realizar el envío de datos al PC (haciendo uso del microblaze), y hacer unas pruebas prácticas exhaustivas de validación del sistema. Por otro lado, se podría continuar con la optimización del correlador eficiente de códigos LS para que en un futuro se pudieran implementar longitudes mayores y así tener mayores distancias de exploración.

Capítulo 6 Bibliografía

- [An08] Analog Devices. "AD5025/AD5045/AD5065 Datasheet", consultado el 1/04/14.
- [BGL01] Alison Brown, Neil Gerein y Lee Savage. "Multipath Caracterization using Digital Phased Arrays". Proceesings of ION 57th Annual Meeting, Albuquerque, NM, June 2001.
- [G14] R. García. Trabajo fin de Máster, "Arquitectura Eficiente para el Procesamiento de un Array Ultrasónico basado en Técnicas de codificación", Universidad de Alcalá, Mayo 2014.
- [GGUP10] E. García, J. J. García, J. Ureña, M. C. Pérez. "Generation algorithm for multilevel LS codes", Electronic Letters, vol. 46, no. 21, pp. 1465-1467, Oct. 2010.
- [GPHSCD13] R. García, M. C. Pérez, A. Hernández, F. M. Sánchez, J. M. Castilla, C. Diego. Seminario anual de automática, electrónica industrial e instrumentación. "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.
- [O03] Alberto Manuel Ochoa Brust. Tesis Doctoral "Procesamiento de Señales con FPGA's", Universidad de Colima, diciembre 2003.
- [P09] Mª Carmen Pérez Rubio. Tesis Doctoral, "Generación y correlación eficiente de códigos binarios derivados de conjuntos de secuencias complementarias para sistemas ultrasónicos", Universidad de Alcalá, 2009.
- [R10] J. Richard Fisher. "Phased Array Feeds". Astro2010 Technology Development White Paper, 2010.
- [RS83] O. T. Von Ramm y S. W. Smith. "Beam Steering with Linear Arrays", Biomedical Engineering, BME-30, nº 8, pp. 438-452, 1983.
- [SBH01] S. Stanczak, H.Boche y M. Haardt. "Are LAS-codes a miracle?" En *Proc. of IEEE Global Telecommunications Conference* (GLOBECOM'2001), volumen 1, páginas 589-593, San Antonio (Estados Unidos), noviembre 2001.
- [TX13] Texas Instruments, "DAC121S101/DAC121S101Q 12-Bit Micro Power, RRO Digital-to-Analog Converter datasheet", Septiembre 2014.
 - [Xil09] Xilinx Inc. "Virtex-5 Family Overview. DS100 (v5.0)", febrero 2009.
 - [Xil13] Xilinx Inc. User Guide, "AMS101 Evaluation Card. UG886 (v1.3)", 2013.
 - [Xil14] Xilinx Inc. "7 Series FPGAs Overview. DS180 (v1.15)" February 18, 2014.
- [Xil14] Xilinx Inc. User Guide, "7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter. UG480 (v1.4)", mayo 2014.

[ZYH05] C. Zhang, S. Yamada y M. Hatori. "General method to construct LS codes by complete complementary sequences". *IEICE Trans. On Wireless Communications Technologies*, E88-B (8): 3484-3487, agosto 2005.