

# Implementación en FPGA de un Transmisor QPSK para un Conformador de Haz

Javier Fernandez, Pablo Sonna, Gabriel R. Caballero, Raúl A. Gastaldi, Marcela B. Busnardo, Sergio A. Medina

Instituto Universitario Aeronáutico

Departamento Electrónica y Telecomunicaciones, Área I+D

Córdoba, Argentina

Email: {jfernandez, psonna}@iua.edu.ar

**Resumen**—La conformación de un haz de radiación en el campo de antenas inteligentes se lleva a cabo alimentando cada una de las antenas de un arreglo, con la misma señal desfasada apropiadamente para cada elemento del arreglo. El propósito de este trabajo es presentar un modulador QPSK<sup>1</sup> al cual se le puede modificar la fase de la portadora en tiempo real sin interrumpir su funcionamiento, mediante la variación de la fase del oscilador local en la etapa de frecuencia intermedia. Este modulador tiene como fin ser utilizado en la implementación de un haz de radiación, en la línea de trabajo de antenas inteligentes.

**Indice de Términos**—Conformador de Haz, Field Programmable Gate Arrays, Quadrature Phase Shift Keying.

## I. INTRODUCCIÓN

Los arreglos de antenas inteligentes están siendo cada vez más utilizados en diversidad de aplicaciones gracias a los avances de la tecnología. En un transmisor de múltiples antenas, el haz se conforma en una dirección particular variando la fase de alimentación de cada antena en forma individual. La forma clásica de generar estos desfases es utilizando elemento capaz de generar una variación de fase y amplitud sobre la señal proveniente del transmisor para cada antena. Esta arquitectura se muestra en la figura 1. Uno de los factores limitantes en la integración de transceptores de arreglos de antenas inteligentes, es que para generar el desfase necesario se requieren desfasadores de radio-frecuencia cuyo costo y complejidad son elevados [1],[2]. Una forma de evitar el uso de estos desfasadores de RF es generar los desfases en una frecuencia intermedia dentro de la cadena del transmisor. Si además se agrega que la generación de portadora se hace de forma digital utilizando técnicas de síntesis de señales, se pueden generar los desfases de forma digital.

En este trabajo se muestra una implementación de un conformador de haz completamente digital. Dentro de la FPGA<sup>2</sup> se genera tanto la modulación, como así también la portadora y los desfases necesarios para alimentar cada una de las antenas.

El cálculo del desfase necesario para direccionar el haz en la dirección deseada ya fue derivado en [3]. Aquí solo se utilizarán los resultados de dicho estudio.

<sup>1</sup>Quadrature Phase-Shift Keying

<sup>2</sup>Field Programmable Gate Arrays

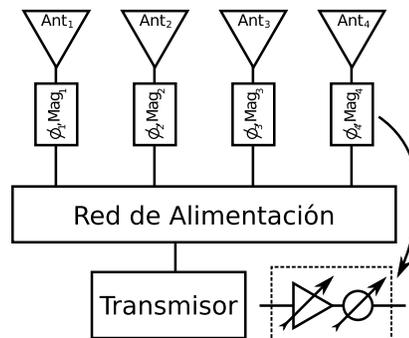


Figura 1. Diagrama de Bloques

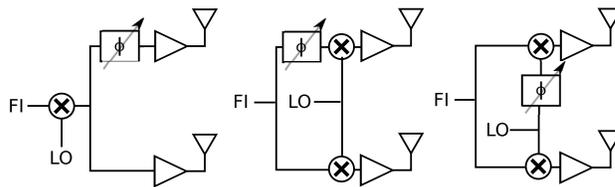


Figura 2. Conformadores de Haz

Las aplicaciones de esta utilidad son variadas: En el ámbito de la aviación, complementado con un sistema de sistema tracking, se puede direccionar un haz de transmisión para concentrar la energía de la señal y para minimizar la posibilidad de interferencias. En el caso de la telefonía celular, se logra incrementar considerablemente el ancho de banda para cada móvil ya que se reduce la interferencia.

## II. CONFORMADOR DE HAZ DIGITAL

En un conformador de haz de banda ancha, se requieren retardos temporales reales en las cadenas de RF para lograr la conformación del haz. Sin embargo, si el ancho de banda de la señal a transmitir es pequeño, este retardo puede aproximarse por un desfase constante en la frecuencia central de transmisión. Dicho desfase puede estar en la salida del transmisor, o como se mencionó anteriormente, puede incluirse en la cadena de radio-frecuencia. En la figura 2 se muestran tres opciones posibles para generar el desfase. Para una implementación completamente digital, la arquitectura c) es la más conveniente, ya que el oscilador local se sintetiza

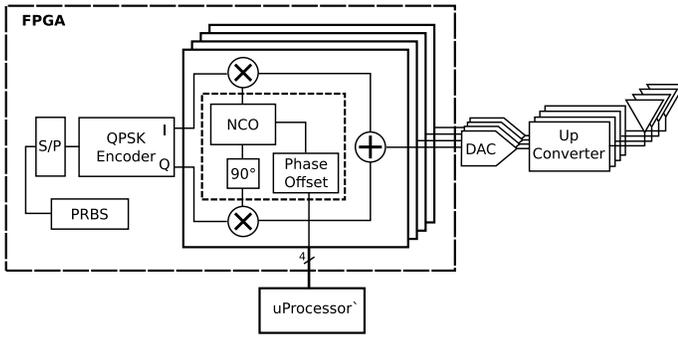


Figura 3. Diagrama de Bloques

digitalmente utilizando un DDS<sup>3</sup> y la fase puede ser fácilmente modificada en el dominio digital. Al hacer esto se evita completamente el uso de desfasadores analógicos y la cadena de RF se reduce a una serie de simples upconverters. La arquitectura propuesta en este trabajo se muestra en la figura 3. A los fines de comprobar la arquitectura, se generan en la FPGA datos aleatorios con un generador de PRBS<sup>4</sup>. Estos datos simulados serán reemplazados por información real en una futura implementación. Los datos se convierten en una palabra de dos bits que entra al codificador QPSK. Luego se repite N veces el bloque modulador, donde N es la cantidad de elementos en el arreglo. El bloque modulador consta de un oscilador en cuadratura controlado numéricamente, al que se le puede controlar la fase. Esta señal en cuadratura se multiplica por los canales I y Q que salen del codificador QPSK y se suman. A la salida del modulador se obtiene la señal QPSK en la frecuencia intermedia y con un desfase controlado digitalmente. Estas señales digitales se convierten al dominio analógico en los conversores digital a analógico y entran a los upconverters que la llevan a la frecuencia de transmisión. La selección de los desfases relativos entre elementos se calculan en un microprocesador y se transmiten a la FPGA a través de un puerto serial.

### III. DISEÑO Y SIMULACIÓN

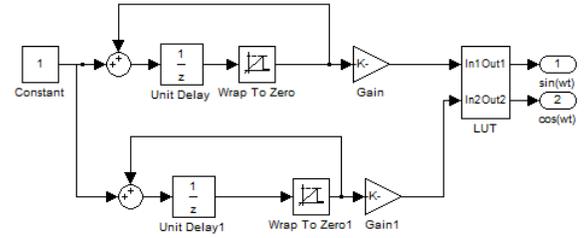
En esta sección se detallan las metodologías de diseño y simulación del sistema completo. El sistema consta de un generador de datos aleatorios, un conversor de serie a paralelo, un codificador QPSK y un generador de portadora en cuadratura con fase variable. En la siguiente sección abarca la etapa de simulación del sistema. En las siguientes secciones se describe el funcionamiento y la metodología de diseño utilizada en cada uno de estos bloques. Cada uno de estos bloques fue diseñado y simulado utilizando el programa de simulación Simulink®.

#### III-A. Generador de Seno y Coseno

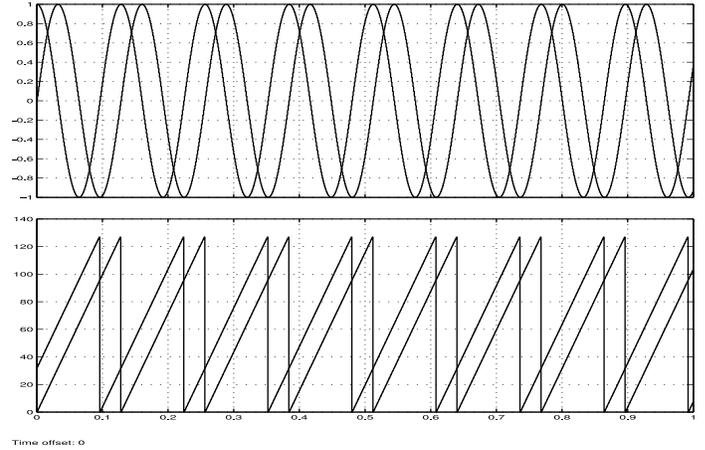
Para generar una señal senoidal digitalmente, se utiliza una técnica conocida como DDS o Direct Digital Synthesis [4],[5]. Con un integrador de N bits se genera una rampa periódica,

<sup>3</sup>Direct Digital Synthesizer

<sup>4</sup>Pseudo-Random Bit Sequence



a)



b)

Figura 4. Generador de Seno y Coseno

cuyo período puede ajustarse variando la entrada al integrador  $f_{in}$ . La frecuencia fundamental de la rampa periódica queda determinada por  $f_{in} \frac{f_{clk}}{2^N - 1}$ , en donde  $f_{in}$  es un número que puede ir desde 0 hasta  $2^N - 1$ . Si se pasa esta rampa digital a través de una Look Up Table (LUT<sup>5</sup>) que convierte los valores de memoria ingresados en su correspondiente valor de seno o coseno, se obtiene la señal deseada. Para generar dos señales senoidales con un desfase de  $\pi/2$  entre sí, se colocan dos integradores, y a uno de ellos se lo inicializa en  $2^N/4$ . De esta forma las dos rampas siempre quedarán un cuarto de período desfasadas entre sí. En la figura 4a se muestra el diagrama de bloques del generador de seno y coseno y en la figura 4b se muestran las salidas tanto del integrador como de la LUT.

#### III-B. Generador de Datos Aleatorios

Para simular una serie de datos aleatorios, se crea un generador de PRBS o secuencia de bits pseudo-aleatoria. Este bloque consiste en un shift register realimentado. En este caso se eligió una PRBS7 por simplicidad, que consiste en un registro de 7 bits.

#### III-C. Desfasador en Cuadratura

Este módulo tiene como función escalar y desfasar las portadoras I y Q. Si se expresa a las portadoras I y Q como una sola señal compleja  $e^{j\omega t}$ , entonces este bloque

<sup>5</sup>Look-Up Table

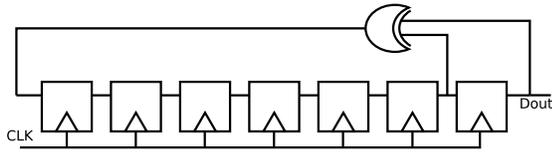


Figura 5. Generador de Datos Pseudo-aleatorios

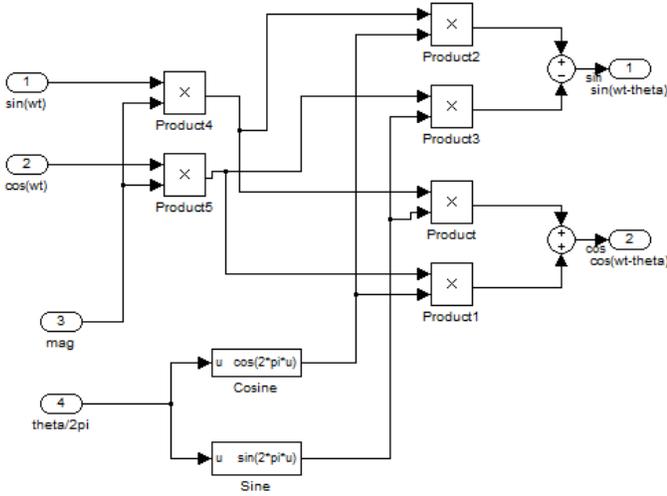


Figura 6. Desfaseador en Cuadratura

debe realizar la multiplicación de esta señal por un factor complejo  $Ae^{-j\theta}$ . Al descomponer esta multiplicación en parte real e imaginaria, se obtiene el diagrama de bloques de la figura 6. La parte real (I) queda como  $A \cos \omega_c t \cos \theta + A \sin \omega_c t \sin \theta = A \cos(\omega_c t - \theta)$  y la parte imaginaria (Q) queda  $A \sin \omega_c t \cos \theta - A \cos \omega_c t \sin \theta = A \sin(\omega_c t - \theta)$ . El módulo recibe como entradas la magnitud A y el seno y coseno del ángulo por el que se desea desfasar la portadora. Este módulo es de vital importancia ya que es el que permite generar los desfases relativos entre las portadoras y eventualmente direccionar el patrón de radiación del arreglo de antenas.

#### III-D. Modulador QPSK

El siguiente bloque se muestra en la Figura 7 y se trata del Modulador QPSK (QPSK - Modulator)

El bloque modulador realiza la función de recibir los datos seriales y mapearlos a una constelación QPSK. El demultiplexor o conversor serie/paralelo, recibe datos en forma serial y los agrupa en una palabra de dos bits. Los datos que salen del demultiplexor (Demux) se convierten de unipolar a bipolar, es decir que el 0 se mapea a -1 y el 1 se mapea a 1. Uno de los datos ingresados se multiplica por la componente I de señal y el otro dato por la componente Q. La salida del módulo junto con la entrada de datos serial se muestran en la Figura 8. En la ventana superior se pueden ver la sucesión de datos ingresando en serie y en la ventana inferior se puede observar el resultado de la modulación para dos bloques iguales, pero con sus portadoras desfasadas. Se puede observar que la salida del modulador varía cada dos datos ingresados y

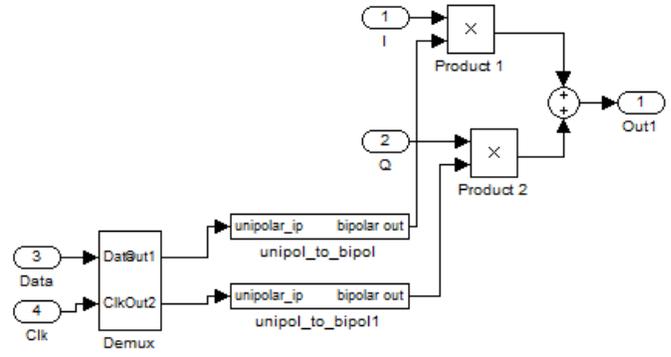


Figura 7. Codificador QPSK

que al ingresar datos agrupados de a dos bits, existen cuatro posibles combinaciones, las cuales están asociadas cada una a un desfase de la señal de salida con la portadora generada.

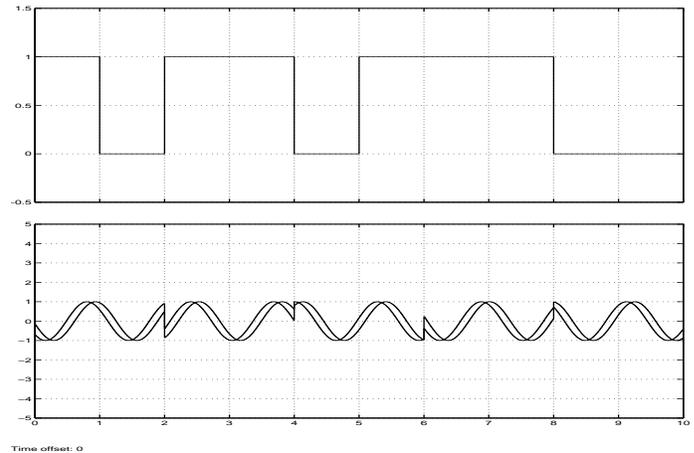


Figura 8. Simulación del Modulador

#### IV. IMPLEMENTACIÓN

A los fines de verificar la arquitectura se implementa en FPGA una sola rama del arreglo de moduladores. El diagrama en bloques de la arquitectura implementada se muestra en la figura 9. Todo el código fue escrito en el lenguaje de descripción de hardware Verilog. Las simulaciones y verificaciones se realizaron utilizando la herramienta ISE®. En el cuadro I se muestra un detalle de los recursos utilizados por la rama del modulador implementada. Esto solo considera los recursos de los bloques que se repiten N veces para una implementación en un arreglo de N antenas. Quedan fuera de esta tabla tanto la LUT como el generador de PRBS. Para llevar a cabo la implementación, se utilizó una placa de desarrollo Spartan-3E Starter Kit Board en la cual se integra una FPGA con distintos periféricos. La FPGA utilizada es XC3S500E de marca Xilinx.

El conversor analógico-digital cuenta con 4 canales de 12

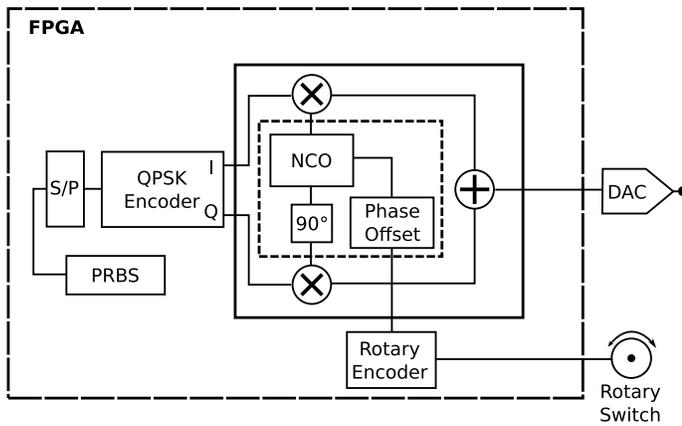


Figura 9. Diagrama de los bloques implementados

Bloques	Cantidad
Multiplicadores	4
Sumadores/Restadores	10
Contadores	2
Registros	60
Comparadores	3
XORs	1

Cuadro I  
RECURSOS DE UNA RAMA

bits de resolución con una velocidad de 2MSPS<sup>6</sup>. El conversor se comunica por medio de una interfase serie SPI<sup>7</sup> de tres cables, con un código de 24 bits. Para realizar la interfase entre la salida digital y el conversor Digital-Analógico se implementa un bloque que toma la palabra de salida del modulador y la convierte a una secuencia serial. La señal de clock de salida hacia el conversor se invierte respecto de la señal de clock interna del bloque para cumplir con el timing requerido por el dispositivo y se utiliza una maquina de estados simple para habilitar y deshabilitar el bit de habilitación del puerto SPI.

El bit de habilitación del puerto SPI sirve como reloj maestro para todo el sistema, ya que es el que define el tiempo de muestreo de la señal de salida.

Debido a que la placa de evaluación tiene un reloj de 50MHz y el tiempo de muestreo esta limitado a 24 ciclos de reloj, se puede aprovechar al máximo la frecuencia de muestreo máxima de 2MSPS. La señal senoidal de mayor frecuencia que se puede obtener entonces es de 1MHz, teniendo sólo dos muestras por ciclo.

Para comprobar el funcionamiento del bloque desfasador en cuadratura, se utiliza una potenciómetro digital que selecciona distintos desfasajes de la portadora. En la figura 10 se muestra la salida del modulador para dos desfasajes diferentes. Las simulaciones fueron generadas en Isim y luego dibujadas desde Matlab.

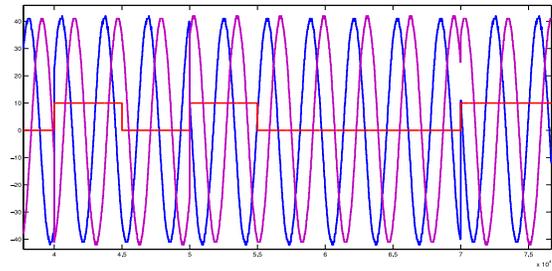


Figura 10. Simulación para distintos desfasajes

## V. TRABAJO FUTURO

Como trabajo futuro y como complemento a este trabajo, se comenzará a trabajar en un receptor QPSK digital para ser implementado en una FPGA. El objetivo es poder cerrar un lazo de comunicaciones en frecuencia intermedia, sin pasar por las etapas de radio-frecuencia. Finalmente, en la última etapa del proyecto se incorporarán las etapas de radio-frecuencia y los arreglos de antenas para comprobar

Además queda pendiente la tarea de estudiar e implementar mecanismos de codificación para optimizar recursos de la FPGA, por ejemplo el uso de los multiplexores internos para ahorrar lógica y el uso de la RAM interna para la implementación de LUTs al fin de optimizar el uso de registros de la FPGA, o la alternativa de utilizar la memoria SRAM incluida como periférico en la placa con el mismo fin de optimizar el uso de registros de la FPGA.

## VI. CONCLUSIONES

Este trabajo forma parte de un proyecto de mayor envergadura que es el de construir un sistema de comunicaciones con antenas inteligentes en el que se pueda controlar el patrón de radiación de acuerdo a diferentes criterios. En este trabajo se ha mostrado una forma de implementar el conformador de haz en una plataforma digital, que simplifica el diseño de las etapas subsiguientes de radiofrecuencia.

## REFERENCIAS

- [1] A. Natarajan, A. Komijani, and A. Hajimiri, "A fully integrated 24-GHz phased-array transmitter in CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 12, pp. 2502 – 2514, dec. 2005.
- [2] D. Parker and D. Zimmermann, "Phased arrays-part ii: implementations, applications, and future trends," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 50, no. 3, pp. 688 –698, mar 2002.
- [3] R. Gastaldi, J. Fernández, G. Caballero, J. Galleguillo, M. Busnardo, and S. Medina, "Desarrollo e implementación de síntesis de patrones de radiación sobre plataforma FPGA," *UEA2011*, Agosto 2011.
- [4] J. Tierney, C. Rader, and B. Gold, "A digital frequency synthesizer," *Audio and Electroacoustics, IEEE Transactions on*, vol. 19, no. 1, pp. 48 – 57, mar 1971.
- [5] L. Cordesses, "Direct digital synthesis: a tool for periodic wave generation (part 1)," *Signal Processing Magazine, IEEE*, vol. 21, no. 4, pp. 50 – 54, july 2004.

<sup>6</sup>Mega-Samples per Second

<sup>7</sup>Serial Peripheral Interface