

DESARROLLO DE UN MODULADOR QPSK UTILIZANDO ENTORNO INTEGRADO DE TRABAJO

Jorge R. Osio

José A. Rapallini

Antonio A. Quijano

Centro de Técnicas Analógico Digitales (CeTAD) - Facultad de Ingeniería - Universidad Nacional de La Plata

josio@gioia.ing.unlp.edu.ar

josrap@ing.unlp.edu.ar

quijano@ing.unlp.edu.ar

RESUMEN

En este trabajo se presenta una aplicación de prototipado rápido de sistemas digitales, diseñando un modulador para comunicaciones digitales del tipo Quadrature Phase Shift Key (QPSK). Se realiza el diseño en alto nivel de abstracción por medio de una herramienta como PeaCE, hasta llegar a una implementación en un microcontrolador de 8 bits, para lo cual se hace uso del Entorno Integrado de Trabajo (EIT), donde se combinan distintas herramientas EDA debidamente relacionadas por medio de distintas interfases de software, que definen el flujo de diseño adoptado

1. INTRODUCCION

Cuando se intenta realizar un diseño electrónico de cierta complejidad y con características heterogéneas, es necesario contar con herramientas que auxilien al diseñador, en la realización de las mismas, en un lapso de tiempo acotado, realizando su evaluación e implementación con sus respectivas pruebas de campo, para responder al requerimiento en tiempo y forma. Se observa que estas herramientas, normalmente no se obtienen en las ofertas que brinda el mercado electrónico / informático, por lo tanto una buena solución para realizar esta tarea es reunir las distintas necesidades (programas para la simulación funcional, herramientas de particionamiento, sistemas de desarrollo tanto para hardware como para software, etc.) en un Entorno de Trabajo Integrado (ETI) [1], que reúna distintas herramientas conocidas, relacionándolas con interfases ad-hoc.

Partiendo de ambientes típicos de codiseño, se trata de potenciarlos incorporando subsistemas. En el esquema de la Fig. 1 se pueden observar distintos bloques que constituyen el ETI planteado, donde se integran 'componentes núcleos' (Ptolemy [8], PeaCE [9] Simulink [10]), con otros que responden a 'soluciones de fabricantes' de sistemas particulares (Freescale [5], Altera [3], Xilinx [4], Microchip [6], National Instruments [2], Texas [7]) a través de 'interfases de transferencia'.

1.1 Análisis comparativo del núcleo del ETI.

De las herramientas presentadas como núcleo del sistema, se pueden utilizar Matlab-Simulink, Ptolemy / PeaCE, de las cuales podemos comentar:

Simulink, forma parte del Matlab (código propietario), muy difundido en el ambiente de diseño, que permite modelar, simular y analizar sistemas dinámicos, esto es sistemas cuyas salidas y estados internos cambian con el tiempo. Es un entorno gráfico, donde se crea un modelo en bloques del sistema, utilizando librerías de bloques estándar y un editor que permite interconectar los bloques del sistema. El modelo representa gráficamente las relaciones matemáticas dependientes del tiempo o a través de las entradas, estados y salidas del sistema. Con el agregado de paquetes adicionales, permite simular máquinas de estados finitos con jerarquía, flowcharts; generar código C para diferentes plataformas o código ADA, a partir del modelo del sistema, además de poder simular en *tiempo real*, permite implementar modelos en *tiempo continuo*, en *tiempo discreto*, *mixto* (discreto y continuo), *máquinas de estado finito* (con jerarquía y concurrencia) y *flowcharts*.

Ptolemy es una buena herramienta de código libre, utilizada para el modelado y la simulación de sistemas, similar a Simulink, pero su capacidad en la síntesis de código está muy restringida por la implementación del sistema; puede producir código C desde una representación de flujo de datos (DF) pero la calidad de código no es satisfactoria. Apenas podría producir un código sintetizable VHDL, pero no del todo optimizado. No hay modo de sintetizar arquitectura desde Ptolemy. La solución fue dada cuando se creó el proyecto PeaCE que es una extensión de Ptolemy en el ambiente de Codiseño. Como está construido sobre la base de Ptolemy, básicamente PeaCE hereda todas las características positivas de Ptolemy como, la integración de diversos modelos de computación, capacidades poderosas de simulación especialmente para aplicaciones de Procesamiento de Señales Digitales (DSP), mejorando las restantes.

La gran ventaja que posee Ptolemy, es estar mucho mas difundido y poseer una gran cantidad de Librerías. La desventaja es que está orientado a control, en cambio

PeaCE posee modelos de computación de control y de flujo de datos.

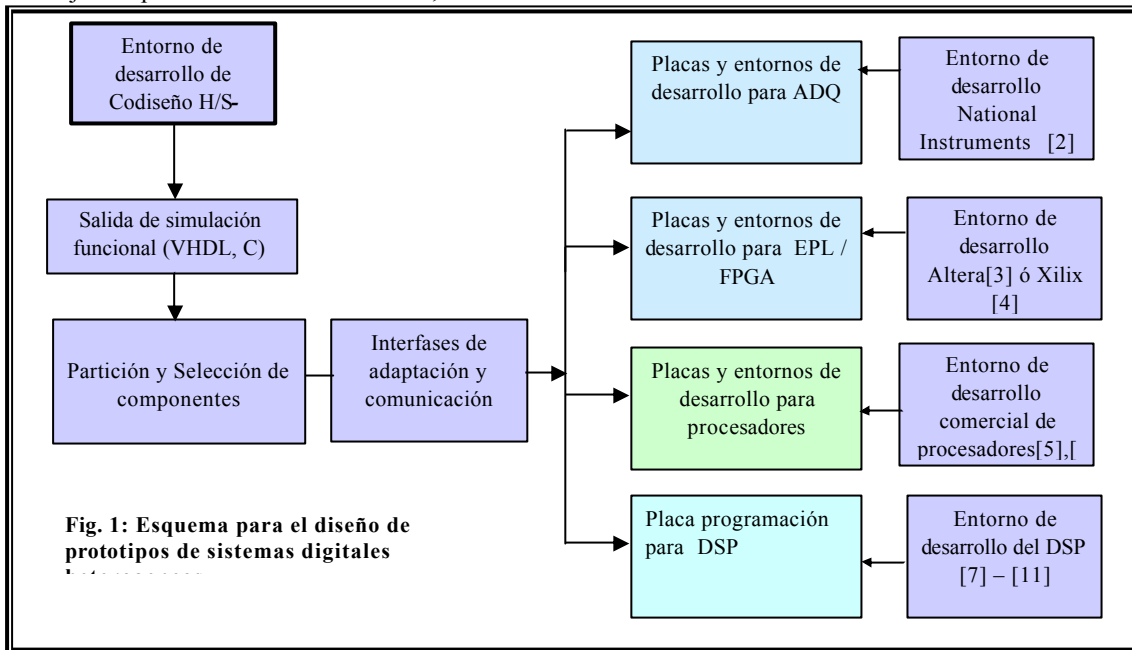


Fig. 1: Esquema para el diseño de prototipos de sistemas digitales

En el ETI, se optó por PeaCE, dado que tiene ventajas sobre Ptolemy y es de código libre y recompilable, con lo que permite, tener la posibilidad de generar características propias definidas por el grupo de diseño.

2. APLICACIÓN

Los pasos más significativos de diseño del modulador QPSK, se realizan aplicando las metodologías de Codiseño mediante el entorno PeaCE. (Figura 2).

El bloque fuente es un generador aleatorio, que entregará las señales a modular; las mismas son cuantizadas mediante un bloque cuantizador “Quant” y luego en el bloque “BitsToInt” se agrupan los datos en palabras de 2 elementos. Estas palabras entran al bloque “TableCX”, donde son mapeadas de la siguiente manera:

- 00 se mapea como (I, Q) ≡ (-1,-1)
- 10 se mapea como (I, Q) ≡ (1, -1)
- 01 se mapea como (I, Q) ≡ (-1, 1)
- 11 se mapea como (I, Q) ≡ (1, 1)

Figura 2: modulador QPSK

donde I = Canal en Fase y Q = Canal en cuadratura

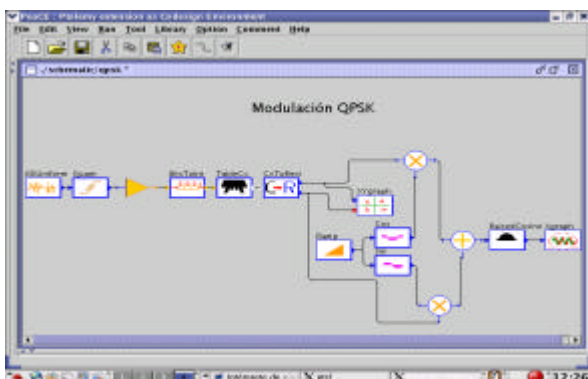
Si los datos no cambian de un periodo al siguiente, la fase de la portadora no cambia. Si hay un cambio de un bit, la portadora es desfasada 90°. Si ambos bits cambian, la fase de la portadora cambia 180°.

Luego del mapeo, el bloque “CxToRect” distribuye la parte imaginaria por un lado y la parte real por el otro.

Por último los valores de I y Q se multiplican por $\cos w_c t$ y $\sin w_c t$ respectivamente. La salida del modulador va a un circuito sumador donde se combinan ambas señales:

- $IQ = 11 \rightarrow \cos w_c t + \sin w_c t$
- $IQ = 10 \rightarrow \cos w_c t - \sin w_c t$
- $IQ = 01 \rightarrow -\cos w_c t + \sin w_c t$
- $IQ = 00 \rightarrow -\cos w_c t - \sin w_c t$

El bloque “Xgraph” grafica la señal modulada en QPSK y el bloque “XYgraph” grafica la constelación de la modulación QPSK.



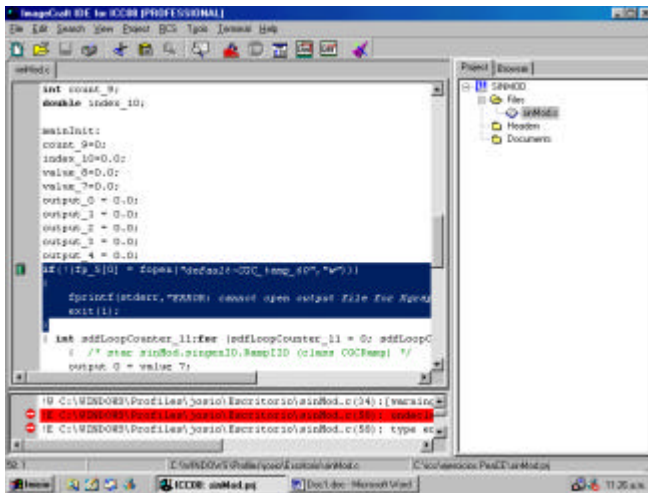


Figura 4 (a)

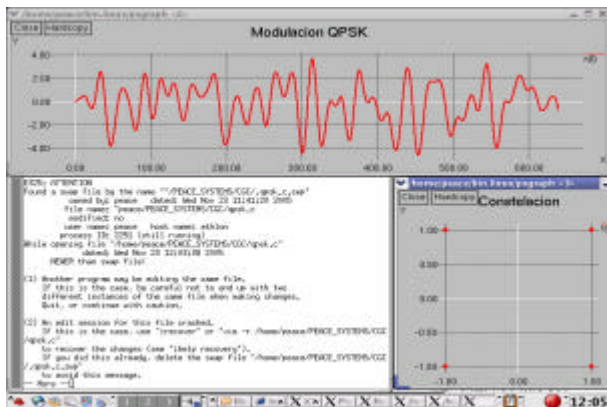


Figura 3: Ambiente de trabajo

La ejecución de este sistema da como resultado la constelación QPSK, la gráfica de la señal modulada en QPSK y un archivo que contiene código C generado automáticamente por el sistema (Figura 3).

El Código C generado se depura para ser pasado por un Traductor que genera un archivo con extensión .S19, el cual contiene código de máquina, y con este se puede programar el microcontrolador seleccionado para la implementación del sistema.

Para la traducción del código “C” se utiliza el Software ICC08 [12] (Figura 4) con el que se obtuvo el “código de máquina”.

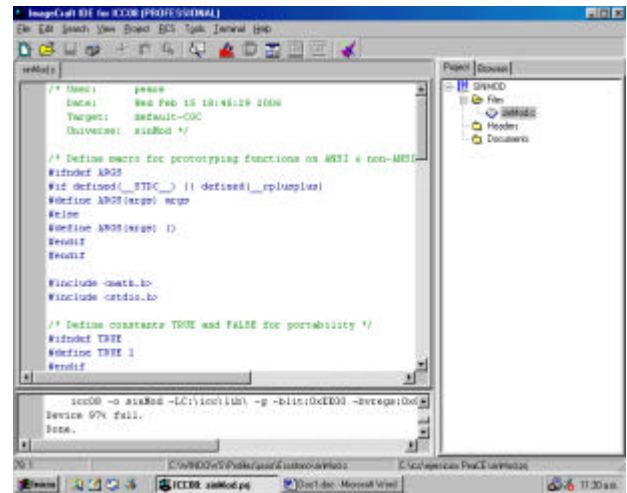


Figura 4 (b)

A través del Software ICS08 WinIDE [13] (Figura 5), se programa el componente elegido de la familia HC908 utilizando el archivo de extensión .S19 obtenido con el ICC. Logrando, finalmente la implementación de un prototipo, el cual permite realizar pruebas experimentales.

Operando de manera similar, se obtiene código ‘VHDL’, implementándose el diseño en FPGA.

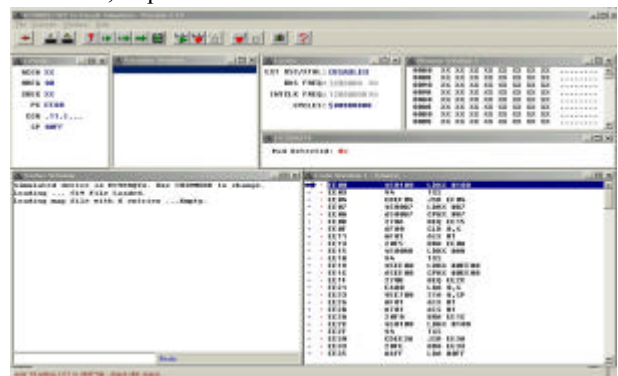


Figura 5. ICS08 WinIDE

5. CONCLUSIONES

La propuesta de un sistema de diseño integrado para el desarrollo de sistemas, genera una buena expectativa de diseño, lográndose realizar los proyectos en tiempo y forma.

La aplicación realizada, se implementó como prototipo, donde su primera utilidad corresponde para su uso didáctico. Desde este punto de vista, es muy útil, para el desarrollo de los trabajos de graduación que realizan los alumnos de ingeniería electrónica.

Los resultados de la aplicación realizada demuestran las ventajas del ambiente de codiseño PeaCE, para ser utilizado con propósitos de investigación y generar prototipos funcionales en un corto tiempo, permitiendo crear sus propios bloques o hacerle modificaciones al programa en si. Esto es

posible dado que se basa en programación orientada a objetos.

Permite la especificación de comportamiento del sistema con una composición heterogénea de tres modelos de computación: *SPDF* para las tareas de procesamiento de señal, *fFSM* para las tareas de control, y el *Modelo de tarea* de más alto nivel.

Se logra la validación fácil del diseño, el análisis estático y el principio de “construcción correcta”.

En definitiva, facilita todos los pasos de Codiseño desde la especificación del sistema hasta el prototipado. Obteniéndose resultados adicionales como:

- La estimación de desempeño de software.
- La selección de componentes y el mapeo
- La Co-simulación con exactitud en tiempo y estimación de desempeño manejada por señales.

10. REFERENCIAS

[1] Entorno de Trabajo Integrado para la Reutilización de Código en el Diseño de Sistemas, José A Rapallini, SPL06, Mar del Plata 2006

[2] www.ni.com

[3] www.altera.com

[4] www.xilinx.com

[5] www.feescale.com

[6] www.microchip.com

[7] ‘Diseño de un Equipo de desarrollo para DSP’, M. Argoitia, A. Manzo, J. Rapallini. XI Jornadas de Jóvenes Investigadores, Asociación de Universidades Grupo Montevideo (AUGM), La Plata, Argentina, 09/2003

[8] User’s Manual: *Ptolemy User’s Manual*, Version 0.7, Linux, University of California at Berkeley, College of Engineering, Department of Electrical Engineering and Computer Sciences, 1997. <http://ptolemy.eecs.berkeley.edu/>

[9] User’s Manual: PeaCE User’s Manual, Version 1.0, Linux, CAP Laboratory of Seoul National University and the Pringet corporation, may 28, 2003. PeaCE: Codesign Environment, <http://peace.snu.ac.kr>

[10] www.mathworks.com

[11] ‘Simulador del TMS320C3X v.1.0’, Alonso Alejandro - Cubillo Arribas,. Univ. de Valladolid. España.

[12]“Embedded C Development tools, ” <http://www.imagecraft.com>, 1994.

[13]”P&E Microcomputers Systems,” <http://www.pemicro.com>, 2004