

OPTIMIZACIÓN DE CÓDIGO C, MEDIANTE SUBRUTINAS DE PUNTO FLOTANTE PARA EL DISEÑO DE SISTEMAS EMBEBIDOS

Jorge R. Osio, Federico Costantino, Sebastián Ledesma, José A. Rapallini, Antonio A. Quijano

Centro de Técnicas Analógico Digitales (CeTAD) - Codiseño Hardware/Software (CoHS)
Facultad de Ingeniería. Universidad Nacional de La Plata
Calle 48 y 116, La Plata 1900, Argentina

josio@gioia.ing.unlp.edu.ar sledesma@barcala.ing.unlp.edu.ar
fcostant@barcala.ing.unlp.edu.ar josrap@ing.unlp.edu.ar quijano@ing.unlp.edu.ar

RESUMEN

El Entorno de Diseño PeaCE se caracteriza por ser una herramienta eficiente para el prototipado rápido de sistemas embebidos, reconfigurables y de bajo costo, mediante técnicas de codiseño HW/SW. Una de las principales ventajas que ofrece es la generación de código eficiente en la síntesis de sistemas, tanto en C como en VHDL.

El código C generado está orientado a implementaciones en DSPs (variables en Punto Flotante de 32 bits). Por lo tanto para el Diseño de Sistemas Embebidos simples con procesadores de 8 bits, que necesiten precisión numérica, son necesarias transformaciones del código. Esto lleva a la creación de subrutinas de operaciones en Punto Flotante (suma, resta multiplicación y división) eficientes para ser utilizadas en forma directa por el compilador a código de máquina del MCU. Como aplicación, se implementa un generador de señal sinusoidal para los MCUs 908 (utilizable para un modulador PSK).

1. INTRODUCCIÓN

Los sistemas embebidos [1] se aplican a infinidad de soluciones electrónicas, resolviendo problemáticas cada vez más complejas.

La planificación del sistema se realiza aplicando técnicas de codiseño hardware / software, tratando de desarrollar métodos y herramientas que pueden usarse para mejorar la calidad de la aplicación final.

El diseño de un sistema comienza con una descripción funcional que da origen a la especificación y a su simulación, obteniendo luego una síntesis del código a utilizar.

El Entorno de desarrollo PeaCE [2] es una de las herramientas más eficientes para el prototipado rápido de sistemas embebidos, que permite realizar la especificación, simulación y síntesis del mismo.

PeaCE integra diversos modelos de computación [3] (para control y flujo de datos), capacidades poderosas de simulación especialmente para aplicaciones de Procesamiento de Señales Digitales (DSP).

El código C generado por PeaCE es muy eficiente para aplicaciones en DSPs. Pero si las operaciones definidas en lenguaje C, que utilizan valores en punto flotante, como la función seno, son utilizadas por el traductor de C a Código de Máquina, se genera un código que ocupa aproximadamente el 80% de la memoria en un Microcontrolador Freescale de la Familia HC908. Para que este código sea portable por otras arquitecturas, se deben tener en cuenta algunas consideraciones.

En el caso de Arquitecturas MCUs de 8bits, si se desea mantener la precisión y una ocupación razonable de la memoria, al código C generado por PeaCE se le deben agregar subrutinas que realicen operaciones en punto flotante, desarrolladas especialmente para estas arquitecturas, ya sea como archivos "include" o dentro del mismo código.

2. FLUJO DE DISEÑO

La Figura 1 muestra el flujo de diseño, en particular los bloques sombreados corresponden a los pasos realizados para su uso en microprocesadores genéricos de 8 bit, que se desarrollan en el presente trabajo.

Las rutinas de punto flotante realizan la suma, resta, multiplicación y división. Mediante estas operaciones y la utilización de algoritmos matemáticos se pueden implementar las demás operaciones.

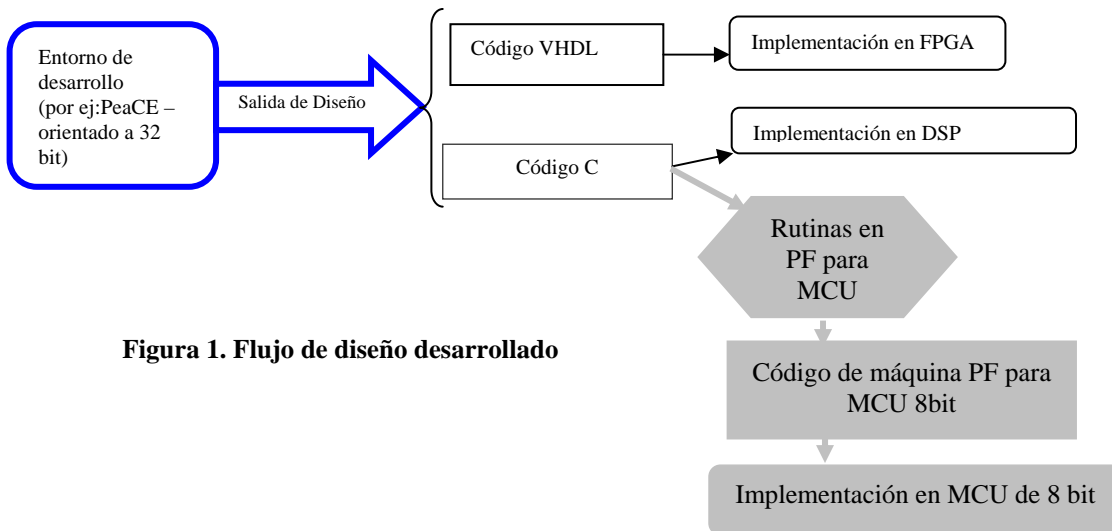


Figura 1. Flujo de diseño desarrollado

3. RUTINAS EN PUNTO FLOTANTE

Las rutinas de multiplicación, división, suma y resta de números en formato de punto flotante [4], siguiendo las especificaciones de punto flotante descritas por la Norma 754 de la IEEE. Las operaciones se realizan con valores en punto flotantes en simple precisión, es decir, que cada número se representa con 32 bits.

Un número en punto flotante según la norma está formado por los siguientes campos: **S** es el bit de signo y **Exp** es el campo exponente. El signo del exponente **s** es de un bit y la **Fracción o mantisa** es de 23 bits.

Signo del número real S :	1 bit
Signo del exponente s :	1 bit
Exponente (entero Exp):	7 bits
Mantisa (número real [fracción]):	23 bits

3.1. Datos de interés sobre las operaciones en el MCU.

Al hacer una operación matemática si el resultado es negativo, el microcontrolador lo expresara en CA2 y seteará la bandera N del CCR [5].

En la multiplicación (o división) el resultado de la suma (o resta) de los exponentes debe estar dentro del rango permitido, siendo el resultado de la operación infinito si excede el máximo o cero si es por debajo del mínimo

Cuando los dos números sean muy chicos, entonces los exponentes normalizados serán menores a cero. Este es el caso en el que el resultado es cero.

4. APLICACIÓN

Para la generación de una onda sinusoidal se comparan el teorema de Taylor y la fórmula de Mac Laurin,

utilizando este último ya que tiene una rápida ejecución y ocupa poco espacio en memoria. Para formalizar la aplicación se utilizan las herramientas indicadas en [6] y [7], con el correspondiente kit de programación para el MCU HC908 de Freescale.

5. CONCLUSIONES

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. Se demuestra que mediante las subrutinas de operaciones, en punto flotante, creadas y la utilización de algoritmos matemáticos que hacen uso de ellas, se pueden diseñar sistemas embebidos simples que requieren de precisión numérica pero no gran procesamiento matemático.

6. REFERENCIAS

- [1] S. Edwards, L. Lavagno, E. A. Lee, and A. Sangiovanni-Vincentelli, "Design of Embedded Systems: Formal Models, Validation, and Synthesis," IEEE Proceedings, pp. 366-390, March 1997.
- [2] User's Manual: PeaCE User's Manual, Version 1.0, Linux, CAP Laboratory of Seoul National University and the Pringet corporation, may 28, 2003.
- [3] Jorge Osio, José Rapallini, Antonio Quijano, "Análisis de Modelos Computacionales para Sistemas Embebidos," Iberchip, San José de Costa Rica, Marzo de 2006.
- [4] An American National Standard, "IEEE Standard for Binary Floating-Point Arithmetic", Approved July 26, 1985 Reaffirmed May 21, 1991. American National Standards Institute.
- [5] Ing. Gabriel Dubatti. <http://www.ingdubatti.com.ar/artmath.htm>
- [6] "Embedded C Development tools" <http://www.imagecraft.com>, 1994.

[7] "P&E Microcomputers Ssystems"
<http://www.pemicro.com>, 2004.