



Conciencia Tecnológica

ISSN: 1405-5597

contec@mail.ita.mx

Instituto Tecnológico de Aguascalientes
México

López Villalobos, J.J.; Mata Vázquez, R.
Propuesta de diseño de un coprocesador aritmético
Conciencia Tecnológica, núm. 15, diciembre, 2000
Instituto Tecnológico de Aguascalientes
Aguascalientes, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=94401502>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

PROPUESTA DE DISEÑO DE UN COPROCESADOR ARITMÉTICO

J.J. López Villalobos, R. Mata Vázquez

Departamento de Ingeniería Eléctrica-Electrónica

Instituto Tecnológico de Aguascalientes

Av. Adolfo López Mateos 1801 Ote, Aguascalientes, Ags. C.P. 20256. México

Teléfono: +52 4 / 910-5002 Fax: +52 4 / 970-0423

Email : jjlopez@elcaminoisp.net

Resumen

En este trabajo se presenta una propuesta de diseño de una arquitectura para procesos aritméticos y el uso de interfaces de comunicación hacia un bus PCI, analizados desde una perspectiva de integración de todos los componentes que la conforman. Se involucra el uso de herramientas CAD, dando énfasis a la interoperabilidad de todos los componentes del sistema mediante subsistemas de comunicación. Se propone el uso de una arquitectura para el procesamiento aritmético la cual se basa en la mezcla de módulos de algoritmos de multiplicación y acumulación (MAC) y estructuras combinatorias, analizándolas en cuanto a su desempeño y costes, formando una arquitectura propuesta para la realización de transformada rápida de Fourier (FFT) de los datos de entrada a procesar. Su especificación es realizada utilizando una herramienta gráfica (Ptolemy). Por otro lado, en la unidad de control se propone el uso de un lenguaje formal (Esterel) para especificar máquinas de estados finitos para llevar a cabo la asignación de actividades para las operaciones aritméticas y de comunicación externa. El campo de aplicación es el apoyo para el procesamiento distribuido de imágenes.

Palabras Clave: codiseño, diseño digital, sumadores aritméticos, multiplicadores aritméticos.

Antecedentes de herramientas

Para el desarrollo de las diversas estrategias de diseño de los sistemas electrónicos en la actualidad se tienen diferentes alternativas para llevar a cabo la construcción de un sistema, tal es el caso del Ptolemy, Mentor Graphics, StateCharts, Synopsis, etc. [1]. Todas estas herramientas tienen la característica de manejar una integración de todas las partes que componen un sistema digital como lo pueden ser procesadores, arquitecturas específicas y dispositivos programables. Esto da como resultado un ambiente conocido como Universo, estructuras más específicas definidas como Galaxias y la unidad básica conocida como Estrella [2]. Lo anterior da pie a una estructura de tipojerárquica, múltiples tipos de variables y con relación de procesos heredados, todo lo anterior demuestra una estructura del tipo objeto que es de mucha utilidad para la construcción de diseños en los cuales se pueden hacer construcciones de diversas etapas y luego ligarlas de una manera sencilla facilitando el proceso de diseño.

Las plataformas de desarrollo tienen como base alguna u algunas arquitecturas definidas dependientes del nivel definición del problema y grado de utilización, esto da la pauta al uso de herramientas de especificación formales que cuenten con la característica de independencia de la arquitectura final de implementación, esto con el fin de analizar la especificación y solución de un

problema específico antes de preocuparse de las cuestiones no funcionales y temporales [3, 5].

El uso y creación de librerías dan un criterio de selección de la herramienta, es la factibilidad de poder trasladar algoritmos de procesamiento y de control para el desarrollo de alguna aplicación particular, ejemplo de lo anterior es el caso de Ptolemy, Foundation, Max+Plus II [5].

Lo más importante para una aplicación de una metodología, es que nos aporte información completa de todo el proyecto dada su síntesis, verificación y sus simulaciones, resultantes de una propuesta de diseño y más aún si se puede intercalar la información de los diversos componentes para tener la evaluación completa de un sistema antes de su implementación, se puede lograr con la conjunción de herramientas como lo son el POLIS y el Ptolemy, ambos de la UC Berkeley.

Estructuras digitales

En esta parte resulta de una vital importancia la estrategia de la solución de un problema de acuerdo a los parámetros que se deseen obtener con la especificación del sistema, para esto contamos con estructuras digitales que ayudan en la obtención de dicho resultado. Para lo anterior cabe mencionar que el uso de sumas y multiplicaciones son cosa de diario para la obtención de FFTs que son utilizadas para el procesamiento de imágenes y así como el diseño de filtros digitales.

Para lograr dicho objetivo tenemos que conocer estrategias como lo son los algoritmos y las arquitecturas específicas [6, 7], y partiendo de ahí seleccionar la forma en la cual analizar los resultados de un diseño digital más especializado para una aplicación.

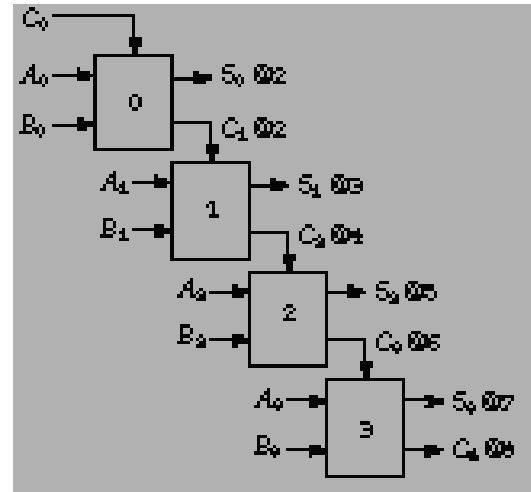


Figura 1. Sumador típico con acarreo suprimido

En el caso de la implementación de software en hardware de algoritmos cíclicos comparando los resultados de ejecución con estaciones de trabajo se ven mejoradas en relación tres a uno [8], pero aún en los casos del acomodo del hardware este nos puede ofrecer una mejor reducción en los tiempos de ejecución.

En primer instancia tenemos la arquitectura de hardware más simple para la construcción de sumadores en una arquitectura dedicada construida con Dispositivos Lógicos Programables Complejos (CPLDs) o Arreglos de Puertas Programables de Campo (FPGAs). Véase Fig. 1. En este caso se dispone de un sumador el cual tiene un registro con corrimiento a su salida para los fines ejecución del algoritmo, se tiene también una retroalimentación para guardar el dato precedente para proseguir con el algoritmo, este tipo de arquitectura denominada como Unidad Aritmética Lógica basada en Registros (RALU), tiene la característica de ser aplicado en algoritmos aritméticos de sumas y multiplicadores implementados mediante una unidad de control la cual se le programa el algoritmo correspondiente.

A continuación tenemos la arquitectura de sumadores combinatoriales, Fig. 1, que se caracteriza por ser mucho más rápida pero con pérdida de algo de precisión [9], pero con la utilización de espacio físico.

En el caso particular de paralelismo y el concepto de Bit Slice se tiene una buena aportación de opciones para lograr mayores números que pueden mejorar el desempeño de cálculos aritméticos, como lo puede ser el diseño de sistemas de procesamiento de señales. El uso de sistemas de conmutación por medio de arreglos tipo mariposa se pueden llevar a cabo por medio de multiplexores que ya son aportados por diversas herramientas para su desarrollo, al igual que otros tipos de estructuras que pueden utilizarse para ciertas aplicaciones i.e. redes de procesadores.

Con esta gama de opciones podemos tener como resultado múltiples variables para su análisis, pero el secreto esta en la metodología de diseño que implementemos para poder hacer dicha evaluación con las herramientas adecuadas que puedan soportar dichas características de construcción de sistemas digitales, que como se menciona anteriormente depende del alcance y participación de diferentes dispositivos para resolver el problema planteado.

Codiseño, la estrategia

Dentro de los métodos de solución a problemas digitales actuales, todas ellas involucran el uso de equipos de computo para llevar a cabo el desarrollo de soluciones, la diferencia radica en la forma en la cual se planteen los pasos a seguir que tantos elementos se involucren.

Según [10], un caso típico para el diseño de alguna arquitectura se parte del uso de lenguajes de especificación de hardware como lo son el VHDL, Verilog y AHDL, de ese punto se procede a la síntesis del problema y a su simulación, en lo general, se tiene una metodología descendente en el cual se modela y se simula llegando a una etapa

de generación de hardware en la cual se sintetiza y se simula nuevamente, pero todo lo anterior se encajona en un solo dispositivo del tipo OTP y muticonfigurable [11]. En el caso de microcontroladores, microprocesadores y DSPs se tienen lenguajes ensambladores de transferencia de registros RT que están más especializados en estos únicamente. La pregunta es ¿Podemos integrar todo?, la respuesta es si, dando con esto una mejor forma de visualizar todo el problema así como la integración de todas las fases del desarrollo que lo conforma i.e., especificación, síntesis, verificación y simulación. Un punto que hace la diferencia es el concepto de partición dentro de la metodología, que en el caso de los lenguajes descriptivos se tienen los

diferentes bloques que lo conforman, pero todos ellos son ubicados muy probablemente en estructuras idénticas i.e. FPGAs de una sola marca comercial. En el caso del codiseño, se tienen diferentes dispositivos que lo conforman dando con esto un aspecto muy interesante para el diseñador de sistemas heterogéneos dado que la partición se estandariza de dispositivos de hardware (lo programable en VHDL) y de software (lo programable en C o ensamblador) que se pueden diseñar conjuntamente en una sola plataforma.

En el caso de la figura 2 se puede observar la forma en que se dividen las dos partes del diseño y la generación de código correspondiente para cada etapa, lo más importante es que sólo se hace un solo algoritmo y/o especificación para todo el modelo y después se hace la división con la ayuda de herramientas, en el caso del ejemplo anterior se esta diseñando para arquitecturas de DSPs y FPGAs.

Un aspecto importante es la herramienta de partición en este caso se tiene una herramienta

denominada POLIS creada por la UC Berkeley cuya metodología de diseño la cual se describe a continuación.

“El aspecto principal del POLIS, que lo distingue de otros métodos de diseño, ambos tanto diseño de software y hardware, es el uso de un modelo formal globalmente asincronico localmente sincrónico del diseño, este se denomina como Máquinas de Estado Finitas de Codiseño (CFSMs)” [15].

Con esta herramienta se logra la partición a partir de un algoritmo escrito en un lenguaje reactivo como lo es ESTEREL, en el cual no importa el hardware involucrado y además su tiempo de reacción es cero dada su característica reactiva [13].

Por otro lado propone la generación de código para diferentes aplicaciones como lo son microcontroladores, FPGAs, VHDL, C y Ptolemy, esto da una amplia variedad de aplicación entorno al diseño cooperativo.

Ptolemy da una interfaz gráfica mediante la cual se puede plantear la simulación y la síntesis del proyecto general y dado la generación de código inherente se puede hacer factible que este se genere para cada una de los elementos que componen el prototipo físico. Antes de lo anterior el prototipo virtual ya fue probado, ahorrando tiempo en su diseño.

Algoritmo “Divide y vencerás”

Para hacer una propuesta de diseño, tenemos primeramente que ver cual es el problema a resolver y mediante el desarrollo de algún algoritmo lograr su solución.

$$X(p, q) := \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m) \cdot (W_N)^{(Mp+q)(mL+1)}$$

Normalmente cuando se tiene una muestra de una imagen, por ejemplo, se llena una matriz de

ecn. 1

almacenamiento de una secuencia de datos, la cual se le aplica la ecuación 1.

Con la aplicación de algunas simplificaciones que no se mostrarán la expresión queda de la siguiente manera

$$X(p, q) = \sum_{l=0}^{L-1} \left[(W_N)^{lq} \cdot \left[\sum_{m=0}^{M-1} x(l, m) \cdot (W_M)^{mq} \right] \right] \cdot (W_L)^{lp}$$

ecn. 2

La ecuación 2 es dividida en tres partes para calcular la transformada que buscamos que se divide en:

1. Se calcula la FFT de los puntos M
2. Calculamos la matriz G(l,q)
3. Finalmente, se calcula la FFT de los puntos L

De lo anterior de acuerdo a [14] este procedimiento consume N(M+L+1) multiplicaciones y N(M+L-2) sumas.

Para este caso cabe tomar en cuenta que el desarrollo del algoritmo involucra una cantidad importante de multiplicaciones y de sumas, dada la experiencia del diseñador podrá optar por alguna metodología por proponer, de la cual se describirá a continuación.

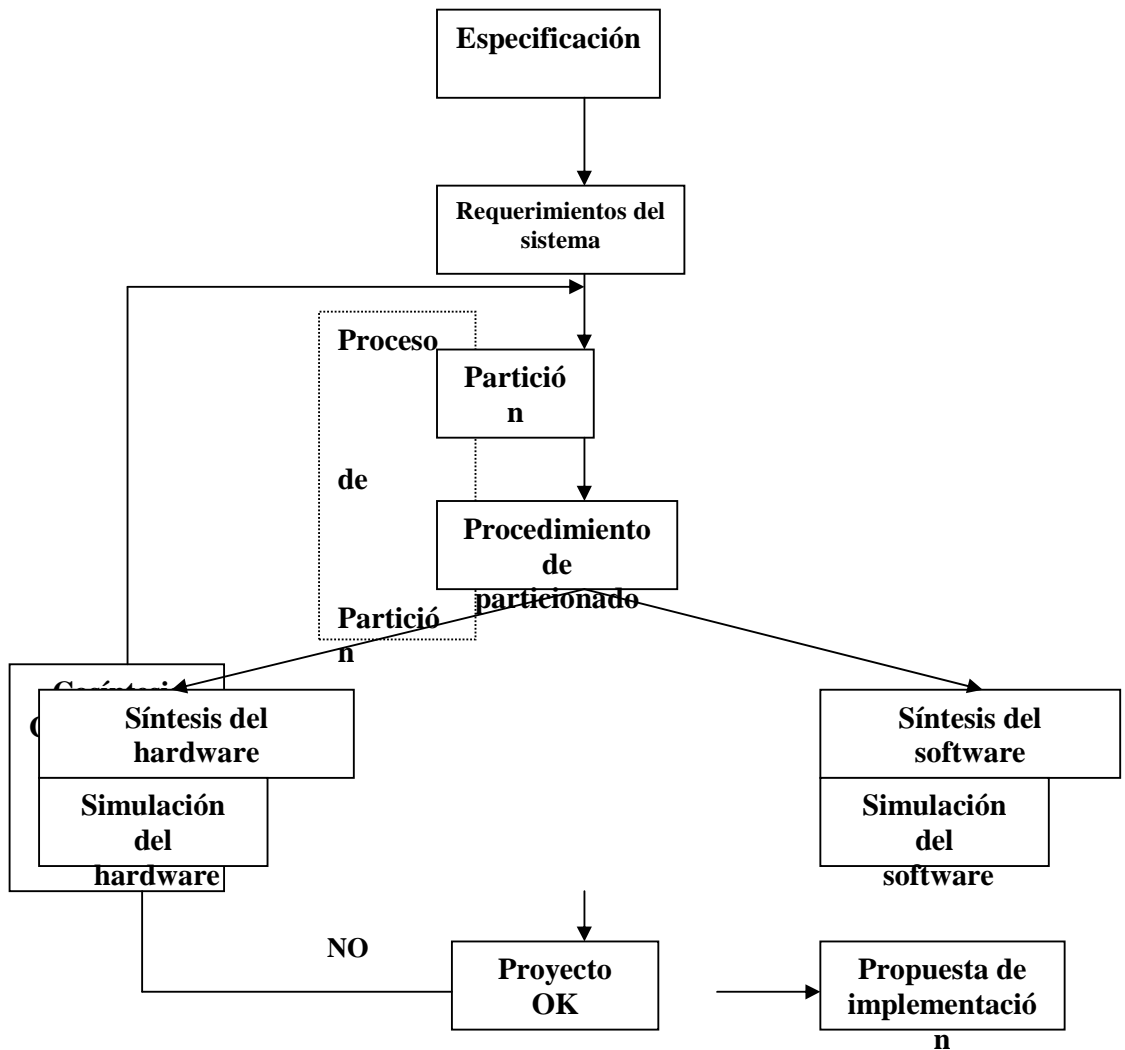


Figura 2 Estructura del Codiseño [12]

Estrategia de solución

Para resolver el problema se dispone de dos herramientas que nos pueden ayudar para hacer la especificación inicial del problema, en este caso el Ptolemy, que ofrece una gran variedad de librerías con las cuales se puede diseñar un sistema no sólo con las FFT de manera directa sino que también puede uno desarrollar otros algoritmos o bien desarrollarla mediante las librerías ya existentes.

En cuanto a la generación de código se puede especificar directamente e inclusive coadyuvar a la partición que la podemos hacer de una manera manual. Por otro lado, la síntesis se va desarrollando conforme a las pruebas de las respectivas simulaciones y verificaciones que en el transcurso del diseño se van teniendo, dando también una aproximación del tiempo de consumo por el proceso globalizado y de manera parcial. Tomando en cuenta los promedios de desempeño de la arquitectura deseada podemos estimar el

tamaño de código por cada una de los elementos programables del sistema.

Un aspecto interesante es que la unidad de procesamiento aritmético es fácilmente desarrollada y analizada con Ptolemy [15], pero un aspecto muy importante para asegurar la sincronía de un algoritmo es la que desempeña la Unidad de Control, para tal fin se prevé el uso de ESTEREL que ayude a dar la secuencia de todas las funciones que deben, en este caso cada una de las multiplicaciones y sumas que son necesarias para obtener la FFT. Para lograr lo anterior se procede a construir un programa en donde se contemple cada uno de los pasos que se involucran, que dependerán tanto de sus entradas y salidas del controlador de secuencia. Con la ayuda de Xesterel, se procede a la compilación y a su respectiva simulación para comprobar que dicha secuencia de los resultados esperados. También existe una herramienta auxiliar gráfica para el desarrollo de programas en ESTEREL[16], el SyncChart[17], el cual no se contempla para este desarrollo.

El controlador se hace para este caso para una arquitectura como la de la figura 1. Las variables de salida la componen sobreflujo OVF, acarreo de salida COUT y las de entrada selector SELECTOR, reloj CLK, carga datos LOAD y acarreo de entrada CIN. De acuerdo con las condiciones se toman en cuenta para el desarrollo del programa se pueden utilizar condicionantes IF, THEN y sus respectivas AWAITs y EMITs para las variables involucradas, que son variables puras o netamente digitales, facilitando su construcción con este lenguaje formal.

Después de la verificación del funcionamiento del código se procede a convertirlo a formato POLIS en cual se procede a particionarlo en hardware o

software ejecutable, pero con su respectivo análisis adicional de desempeño en tiempo y tamaño de código. De aquí se procede a su exportación a una estrella Ptolemy para su cosimulación.

Etapas de desarrollo actuales

En estos momentos se tiene la construcción de posibles estrellas y/o galaxias en Ptolemy con algoritmos de procesamiento y su análisis de desempeño, tanto en formato de registros como de estructuras compuestas (librerías).

Por otro lado, en cuanto a ESTEREL se tiene la propuesta de secuencia de la unidad de control para la solución del algoritmo de multiplicación y/o sumas.

Cabe mencionar que esto forma parte de un proyecto de diseño integral de una tarjeta incrustada en la que se lleve a cabo el procesamiento de firmas escritas de manera distribuida.

Referencias

- [1] G. De Micheli, R. Gupta., "Hardware/Software Codesign", Proceedings of the IEEE, vol 83, No. 3, March 1997.
- [2] UC Berkeley, Dept. of Elec. Eng., "Ptolemy 0.7 User's Manual", vol 1, 1997.
- [3] M. Ercengovac and T. Lang, "Digital Systems and Hardware/Firmware Algorithms, Ed. Wiley, 1985.
- [4] J. Carter, "Digital Designing with Programmable Logic Devices", Ed. Prentice Hall, 1997.
- [5] J. Lopez, "Análisis de las Metodologías de Diseño Digitales de Sistemas Electrónicos",

Memoria del 6to Simposio Estatal de Investigación y Desarrollo Tecnológico, 1999.

[6] J. Mariño, F. Vallverdú, J. Rodríguez, A. Moreno, “Tratamiento Digital de la Señal”, Ed. Alfaomega, 1999.

[7] K. Hwang, “Advanced Computer Architecture”, Ed. Mc. Graw Hill, 1993.

[8] F. Steven, “Computer Architectures and Organization”, Ed. Mc. Graw Hill, 1992.

[9] J. Villasenor and B. Hutchings, “The Flexibility of Configurable Computing”, IEEE Signal Processing Magazine, September 1998.

[10] L. Terés, Y. Torroja, S. Olcoz and E. Villar, “VHDL: Lenguaje Estándar de Diseño Electrónico”, Ed. Mc. Graw Hill, 1998.

[11] E. García, “Curso de entrenamiento de XILINX”, XILINX, 1999.

[12] S. Pillement, L. Torres, M. Robert and G. Cambon, “Concurrent Design of Hardware/Software Dedicated Systems”, CEDEX – FRANCE, 1999.

[13] S. Edwards, “The Specification and Execution of Heterogeneous Synchronous Reactive Systems”, PhD Thesis, UC Berkeley, 1997.

[14] J. Proakis and D. Manolakis, “tratamiento Digital de Señales”, Ed. Prentice Hall, Tercera Edición, 1998.

[15] A. Kalavade and E. Lee, “Hardware/Software Co-Design Using Ptolemy – A Case Study”, Proceedings of the First Intl. Workshop on hardware/Software Codesign, Colorado, September 1992.

[16] G. Berry and Esterel Team, “The Esterel v5_21 System Manual”, Centre de Mathématiques Appliquées INRIA, March 1999.

[17] A. Charles, “SynchChart: A Case Study”, INRIA Grenoble, 1999.