

# DISEÑO Y SIMULACIÓN DE ARQUITECTURAS ARITMÉTICAS RÁPIDAS MEDIANTE HERRAMIENTAS CAD

S. García López, M. Pérez Castellanos, C. Gonzalo Martín y \* J. Arriaga García de Andoain

Universidad Politécnica de Madrid

Facultad de Informática

Departamento de Arquitectura y Tecnología de Sistemas Informáticos

Campus de Montegancedo, Boadilla del Monte, 28660 Madrid

Tfno: (91) 3367382, Fax: (91) 3367412, E-mail: marga@fi.upm.es

\* Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Departamento de Sistemas Electrónicos y de Control

Carretera de Valencia km 7, 28031 Madrid

Tfno: (91) 3367800

**RESUMEN.**- El estudio de algoritmos aritméticos rápidos, el posterior diseño de arquitecturas dedicadas que los soporten, así como el manejo de herramientas VLSI, son objetivos prioritarios en la formación tecnológica de los Ingenieros Informáticos. En este trabajo se presenta la metodología y planificación de trabajos, así como los resultados obtenidos en el desarrollo de un proyecto de diseño de una arquitectura rápida de división, que hibrida el método numérico de Newton-Rapson y el algoritmo de División por Multiplicaciones Sucesivas y que pretende cubrir los objetivos mencionados.

## 1. INTRODUCCIÓN

El conjunto de asignaturas que conforman el perfil tecnológico de los futuros Ingenieros Informáticos tiene entre otros objetivos: transmitir al alumno conocimientos básicos sobre los fundamentos de los dispositivos y sistemas electrónicos que configuran el soporte material de los sistemas informáticos; capacitarle para manejar un bagaje técnico que le dote de un adecuado nivel profesional y proporcionarle el grado de conocimientos suficiente sobre los aspectos tecnológicos específicos de esta Ingeniería. El primer objetivo y parcialmente el segundo se cubren, básicamente, con las asignaturas de primer ciclo: *Tecnología de Computadores* (15 créditos) y *Diseño Digital* (15 créditos), mientras que las del segundo ciclo, entre las que cabe citar la asignatura de *Tecnología Informática*, tienen como objeto barrer los restantes aspectos del conjunto de objetivos. A lo largo de este periodo formativo, se pretende familiarizar al alumno con ciertas herramientas de diseño de Circuitos Integrados (CI), de uso común en el ámbito universitario, o en su caso también de uso industrial. Así en la asignatura de 2º curso *Tecnología de Computadores*, se introduce al alumno en el manejo de una herramienta bastante simple de diseño de CI, pero con un gran interés académico [1], ya que facilita considerablemente la utilización de herramientas más



complejas y potentes, como son las herramientas ES2 [2] que permiten la integración de circuitos a través de EURORACTICE. Las razones para optar por estas herramientas han sido de carácter operativo, no pareciendo oportuno su discusión en este momento. Una vez superada esta etapa de formación, y dado el carácter ingenieril de la citada licenciatura, parece conveniente que dicho perfil tecnológico se complete con la realización de un proyecto dentro de este ámbito: *Proyecto Fin de Carrera*. Este trabajo, en concreto, pretende mostrar tanto los aspectos docentes, como algunos resultados obtenidos en el transcurso de la elaboración de uno de estos proyectos, utilizando para ello la herramienta de ES2, conocida como SOLO 1400.

En este trabajo se propone una arquitectura específica, que soporta de forma eficiente el algoritmo de División Reciproco [4]. La razón fundamental que nos ha llevado a diseñar una arquitectura dedicada para el algoritmo de división, es el alto coste computacional de esta operación; lo cual implica un alto grado de complejidad en ciertos algoritmos de Procesado Digital de Señal (PDS), tales como el algoritmo de Pyle, el algoritmo de Celosía de Gradiente Adaptativo, etc., que requieren la operación citada. En este sentido, resulta muy interesante investigar nuevas técnicas que permitan agilizar los algoritmos clásicos de división. No obstante, dada la naturaleza secuencial de esta operación aritmética, esto no es inmediato, ya que el resultado de cada acción determina la siguiente. El diseño, mediante herramientas CAD, de arquitecturas específicas que se adapten perfectamente a las características de un determinado algoritmo puede ser la solución más aconsejable.

## 2. LA HERRAMIENTA SOLO 1400

La herramienta SOLO 1400 permite el diseño de CI semicustom, siendo este estilo de diseño muy adecuado a la formación que adquieren los alumnos a lo largo de la licenciatura de Informática. Dicha herramienta se caracteriza por su facilidad y sencillez de manejo y permite realizar diseños de complejidad y prestaciones medias. Se basa en células predefinidas y está orientada a la fabricación de ASIC's industriales. Soporta un generador de memorias RAM, D'PRAM, ROM y PLA's, así como un sintetizador que permite la minimización lógica de máquinas de estado finito, estos dos módulos proporcionan una inestimable ayuda a la hora de agilizar el proceso de diseño de ciertos circuitos. El diseño de circuito se puede realizar, bien de forma gráfica o textual, utilizando un lenguaje de especificación de circuitos. Hasta el momento, nos hemos centrado en el diseño gráfico, ya que los lenguajes de especificación entra dentro del ámbito de interés de otros integrantes de nuestro grupo (GTC).

## 3. PLANIFICACIÓN DEL PROYECTO

Los autores consideran que la formación integral del alumno dentro del perfil mencionado, no debe limitarse a la utilización de herramientas de diseño, sino que también debe ser capaz de abordar los problemas desde el nivel algorítmico, analizarlos con el fin de extraer las mejores y específicas características de los mismos y posteriormente, diseñar arquitecturas que se adapten lo más eficientemente posible a ellos [3]. Evidentemente, este planteamiento representa un esfuerzo excesivo para un



único PFC. La solución que nos ha parecido más adecuada es la organización de grupos de trabajo, coordinados por el autor del proyecto, e integrados por alumnos de cursos inferiores (4º o 5º cursos) que apoyen fundamentalmente la etapa de diseño de los circuitos planteados por el coordinador. De esta forma, cuando estos alumnos de cursos inferiores llegan al puesto de coordinador, conocen y manejan con un cierto nivel la herramienta y pueden dedicarse prioritariamente a las labores de análisis, optimización de los algoritmos y planteamiento de la arquitectura. Esta estructura jerárquica está dando bastante buen resultado, además de favorecer la adaptación de los alumnos al trabajo de grupo.

#### 4. DESCRIPCIÓN DEL ALGORITMO RECÍPROCO DE DIVISIÓN

El algoritmo Recíproco de División reduce esta operación a la multiplicación del dividendo por el inverso del divisor. El cálculo del inverso se ha realizado clásicamente de diferentes formas, cabe señalar el método de multiplicaciones sucesivas, así como la tabulación de aproximaciones a dicho inverso [5]. En este trabajo se propone la hibridación de ambas técnicas. Para ello, se toma como primera aproximación al inverso del divisor, un valor previamente tabulado, en vez de igualarla a la unidad, como se hace en el método de multiplicaciones sucesivas. El algoritmo global resultante de aplicar estas ideas, es equivalente a resolver la ecuación no lineal

$$f(x) = \frac{1}{x} \pm D, \quad (1)$$

mediante el método de convergencia cuadrática de Newton Rapson. Obviamente, la raíz de la ecuación (1) coincide con el inverso del denominador. Un aspecto determinante de la convergencia de este método es la primera aproximación a la raíz de la ecuación. Se puede demostrar que la convergencia está asegurada y se consigue en un número de iteraciones relativamente pequeño, cuando esta primera aproximación,  $x_0$ , está acotada entre 0 y  $2/D$  [5,6]. Con objeto de simplificar el posterior diseño de la arquitectura, se ha optado por normalizar los datos de entrada, de forma que el denominador esté siempre comprendido en el intervalo  $[1/2, 1)$ . En estas condiciones, la restricción comentada anteriormente, impone que  $x_0$  sea estrictamente menor que 2, lo que facilita la realización de la citada tabla. La ecuación iterativa

$$x_{i+1} = x_i(2 - Dx_i), \quad (2)$$

permite a partir de  $x_0$  obtener el inverso del denominador. Este proceso iterativo se controlará mediante un parámetro  $\delta$ , con el que se comparará la última aproximación obtenida mediante la ecuación (2). Cuando la diferencia entre 1 y dicha aproximación sea inferior a  $\delta$ , se considerará que la raíz de la ecuación obtenida es la mejor aproximación al valor del inverso del denominador.



## 5. DISEÑO DE LA ARQUITECTURA

El algoritmo descrito en la sección anterior se puede segmentar en tres etapas bien diferenciadas. En la primera etapa se calcula la primera aproximación del inverso del divisor. En la segunda se calcula el valor final del inverso del denominador, mediante un proceso iterativo que teóricamente corresponde con el método de Newton Rapson, pero que desde el punto de vista de la implementación se puede interpretar como el método de división por multiplicaciones sucesivas, ya que la interpretación de la ecuación (2), permite reducir el proceso iterativo, salvo en una resta, a la operación de multiplicación. En la última etapa del proceso se obtiene el resultado final, al multiplicar el numerador por el valor calculado del inverso del denominador. La figura 1 muestra un diagrama de bloques de la arquitectura propuesta. El bloque 1 realiza el proceso de normalización de datos que ya ha sido comentado, correspondiendo los bloques 2, 3 y 4 a las tres etapas anteriormente descritas. Como se muestra en la figura 1, el flujo de datos desde la entrada hasta la salida, se ajusta a una típica estrategia en *pipeline*, con el fin de mejorar la velocidad del proceso y regularizar la salida de datos, en el caso de tener que realizar numerosas divisiones consecutivas. El bloque 2 de la estructura toma como entradas los datos  $D(0:n-1)$  y  $N(0:m-1)$ , utilizándose los  $k$  bits más significativos del primero de ellos para direccionar la tabla que contiene las aproximaciones al inverso de los denominadores de entrada. Evidentemente cada valor de esta tabla corresponde a todo un rango de denominadores [6]. De las cinco entradas al bloque 3, mostradas en la figura 1, tres de ellas no son utilizadas por el mismo, dado que tanto el signo del resultado como el numerador de la fracción, así como la señal *Cero* que detecta la posible división por cero, son transparentes para dicho bloque. En estas condiciones los datos con los que opera este bloque son la primera aproximación al inverso del denominador,  $x_0(0:n-1)$ , y el producto de este por el denominador normalizado,  $A_0(0:n-1)$ , obteniéndose en su salida el inverso del denominador. Como ya se ha comentado este bloque utiliza el algoritmo de División por Multiplicaciones Sucesivas. El cronograma recogido en la figura 3, muestra el funcionamiento de este bloque, en el cual se puede interpretar el algoritmo mencionado.

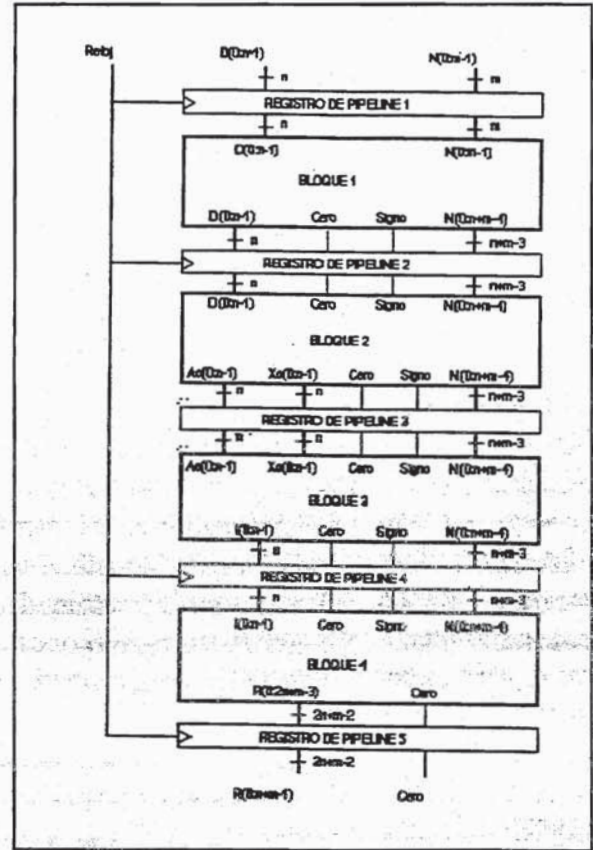


Figura 1: Esquema de la arquitectura

Se inicia el proceso con la señal *Carga* a nivel bajo, lo que activa la entrada de datos al sistema a través de los multiplexores, permaneciendo dicha señal a nivel alto durante el resto del proceso, con objeto de que el resultado de la iteración anterior se realimente. La señal *EsUno* permanecerá a nivel bajo hasta que el valor del



denominador calculado en el bucle actual, se encuentre lo suficientemente próximo a 1. En este momento el resultado de la división será el inverso deseado.

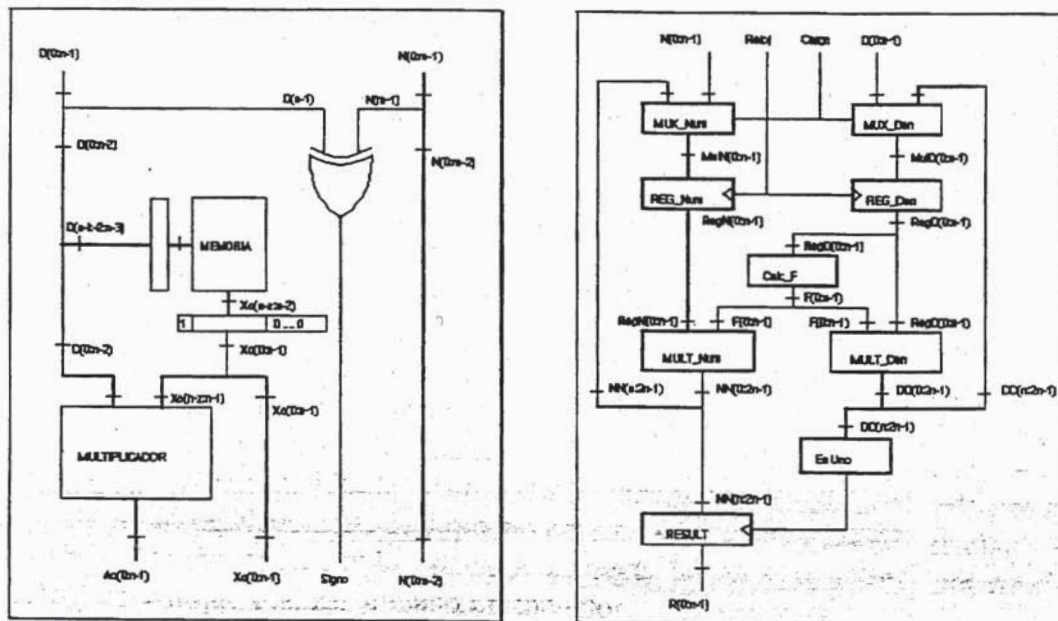


Figura 2: Estructura de (a) bloque 2 y (b) bloque 3

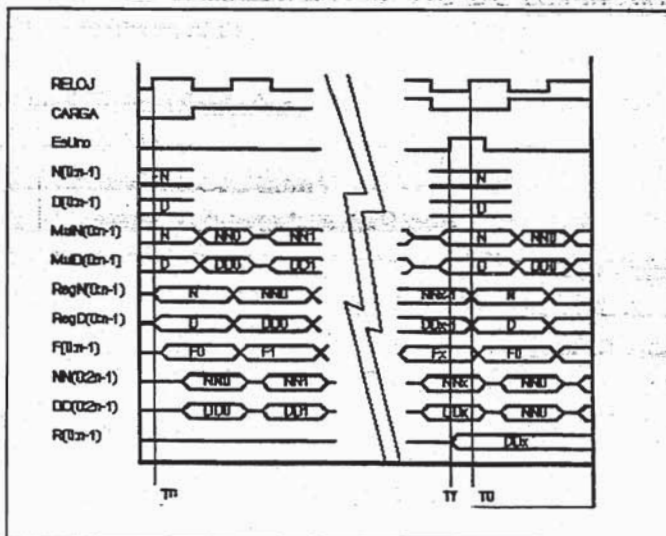


Figura 3: Cronograma de operación de división

na la primera aproximación,  $x_0$  (3:7), se obtienen las entradas al divisor por multiplicaciones sucesivas (bloque 3): la primera aproximación al inverso,  $x_0$ , y el producto de este valor por el denominador,  $a_0$  (4:11). En esta simulación se han considerado 4 valores iniciales del denominador, que corresponden a

Limitaciones de la herramienta de diseño no permiten mostrar claramente simulaciones del proceso global, cuando este requiere un gran número de señales. En la figura 4 se incluye el resultado de simular los dos primeros bloques del divisor, representando sólo aquellas señales que en algún momento del proceso toman un valor alto. Se puede observar que en el flanco de subida de la señal *pipe*, se carga en el segundo bloque el valor normalizado del denominador de entrada,  $dd$  (0:6). Con este último dato normalizado y mediante la tabla que proporcio-

D	0,937	0,687	0,812	0,875
1/D	1,066	1,454	1,230	1,143
$x_0$	1	1,375	1,187	1,125

Tabla I



los casos más desfavorables; estos valores se muestran en la primera fila de la tabla 1, representándose en la segunda sus inversos. La tercera fila muestra los valores tabulados de la primera aproximación al inverso del denominador. Se puede observar que hasta para los casos más desfavorables la aproximación obtenida es suficientemente buena, de hecho solamente se requiere una iteración para que el proceso converja.

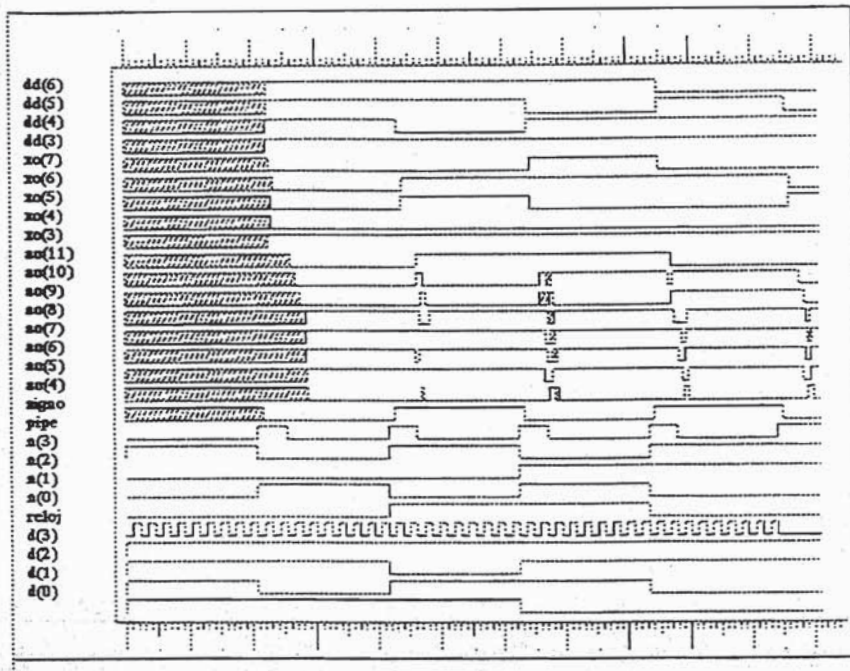


Figura 4: Simulación de los bloques 1 y 2.

## 6. CONCLUSIONES

Las características de la arquitectura presentada, se adaptan perfectamente a la filosofía de trabajo en grupo planteada, de forma que la tarea global se puede desglosar en distintas subtarefas. Cada uno de los tres bloques mencionados, ha sido diseñado y simulado mediante la herramienta CAD citada. En una primera aproximación al diseño, se han utilizado las librerías estándar disponibles [2]. Simulaciones de estos bloques, han mostrado la eficiencia de la propuesta, especialmente cuando sea preciso realizar varias operaciones de división consecutivamente [5]. La arquitectura resultante es altamente regular y modular, siendo su expansión a diferentes representaciones de datos relativamente sencilla. La eficiencia temporal de la arquitectura se puede mejorar, descendiendo a un nivel más bajo de diseño y sustituyendo las librerías utilizadas, por librerías diseñadas por los alumnos que se adapten más específicamente al algoritmo que nos ocupa; esta solución es objeto de próximos trabajos.

## BIBLIOGRAFÍA

- [1] TEDMOS V, **Manual de Usuario**, Traducción al español G.T.C, UPM, 1995.
- [2] European Silicon Structures, **SOLO 1400 Reference Manuals**, ES2 Publications Unit, U.K, 1990.
- [3] N. H. E. Weste and K. Eshraghian, **Principles of CMOS VLSI Design. A Systems Perspective**, Addison Wesley, 1993.
- [4] S. M. Triemberger, **An Introduction to CAD for VLSI**, Kluwer Academic, 1987
- [5] J. J. F. Cavanagh, **Digital Computer Arithmetic**, McGraw-Hill, 1985.
- [6] M. Perez-Castellanos, C. Gonzalo, S. García and M. Blanco, **A Modular and Regular Pipeline Architecture for the Reciprocal Divisor Algorithm**, SCAN '95, Wuppertal, Germany, september 26-29 1995.