Capítulo I

Introducción

1. Motivación
2. Objetivos
3. Conocimiento actual y problemas abiertos
4. Formulación del problema y propuesta de solución

I - 1

Motivación

Pensemos por un momento en el siguiente problema:

> *Sea la ecuación de primer orden*
>
> $10x = 6$
>
> *Su resultado es de sobra conocido y fácil de calcular, despejando x se obtiene un valor de* $x = 0{,}6_{10}$
>
> *Ese número, expresado en notación fraccionaria posicional, indica con precisión absoluta cuál es el valor de la variable x que cumple la relación anterior. Sin embargo, con la mayor parte de la algoritmia y de la tecnología disponible actual no se obtiene su resultado exacto, sino tan sólo una aproximación.*

Esta circunstancia se produce por la naturaleza de la representación binaria posicional del número decimal 0,6 que se compone de infinitas cifras fraccionarias. A este respecto surgen multitud de interrogantes acerca de las características de la representación numérica: ¿cuántas cifras significativas posee un número? ¿en qué medida influye la base de representación? ¿qué conjuntos numéricos provocan esta situación? … Su análisis nos llevará a dilucidar si existen codificaciones numéricas alternativas para estos números o bien se tiene que renunciar a la expresión de su valor exacto mediante representación posicional. La medida que frecuentemente se adopta pasa por acometer aproximaciones al valor correcto utilizando una cierta cantidad de cifras [IEEE, 1985].

Determinadas actividades requieren elaborados cálculos matemáticos y, adicionalmente, cuentan con importantes restricciones de precisión. De inmediato se plantean situaciones en las que resulta conveniente, si no necesario, disponer de la expresión del valor exacto, al menos para dominios numéricos como el del problema anterior: cálculo de trayectorias en sistemas de guiado y posicionamiento, comunicaciones de alta frecuencia, alineamiento de antenas y, en general, en aquellos casos en los que la relación entre el tamaño de los operandos sea desproporcionada. En estos problemas, imprecisiones poco significativas en los cálculos pueden provocar desviaciones de consideración en los resultados que se obtienen. En relación con esta cuestión se deben tener en cuenta las incorrecciones que pueden surgir de las limitaciones de los actuales esquemas de representación y los errores que provocan en las operaciones [Schulte, 2000], [Michelucci y Moreau, 1997], [Goldberg, 1991], [Bohlender, 1990], [Ratz, 1990] y [Hoffmann, 1989]. Por otro lado, en algunas aplicaciones existen restricciones temporales que exigen disponer de la respuesta en un tiempo determinado y reducido. En este contexto, el empleo de métodos de alto nivel puede no ser adecuado para cumplir satisfactoriamente con estos requerimientos. Una posible línea de actuación consiste en traerse al nivel físico soluciones que tradicionalmente han sido del dominio superior: el desarrollo e implementación de operadores que junto con otros elementos y mecanismos del procesador regulan la precisión de los operandos y resultados y alcanzan su representación sin error y, al mismo tiempo, abren la posibilidad de proveer una respuesta ajustable en el tiempo desde el nivel más bajo de la arquitectura [García et al, 2003a], [García et al, 2003c], [Mora, 2001].

I - 2

Objetivos

El objetivo general de este trabajo consiste en avanzar en el conocimiento y en el desarrollo de modelos para procesadores de alto rendimiento y prestaciones. En particular, proporcionar el soporte formal para la concepción de métodos que permitan la representación numérica fraccionaria posicional adecuada para la operatoria con control de la precisión. La casuística es amplia, por lo que difícilmente se puede establecer un modelo general que aporte soluciones de interés aplicado para todos los casos. Se busca, en cambio, una solución que resuelva satisfactoriamente una familia de problemas desde el nivel bajo de la arquitectura. En concreto la investigación se centra en el conjunto de los números racionales y en algunas operaciones básicas.

Debido a la estrecha relación que existe entre la precisión de los resultados y el tiempo de procesamiento, se establece como consideración complementaria la consistencia del modelo con las restricciones temporales. Es decir, se debe proporcionar la flexibilidad necesaria para gestionar el par precisión y tiempo de procesamiento para cada problema.

El objetivo general se concreta en otros parciales relacionados con la representación de los datos y el modelo aritmético. Con respecto al método de representación de la información numérica se establecen los siguientes fines:

- Proporcionar suficiente capacidad expresiva para representar de manera exacta cualquier número racional en notación fraccionaria posicional.
- Extender el esquema de coma flotante tradicional para proporcionar mayor amplitud de expresión numérica.
- Facilitar la implantación hardware y el diseño de técnicas de cálculo flexibles que no penalicen el rendimiento.

En cuanto al modelo de cálculo aritmético se contemplan los siguientes cometidos a alcanzar:

- Constituir un conjunto de operadores sobre la base del esquema de representación propuesto para los números.
- Concebir las operaciones matemáticas con capacidad para realizar cálculos exactos al operar con números racionales y dotarlas de la flexibilidad suficiente para adaptarse a los requerimientos del hardware.

La investigación se orienta a la concepción de un coprocesador específico: Procesador Racional Flexible (FRP— *Flexible Rational Processor*) capaz de procesar números racionales expresados en el formato de representación posicional exacta. Dicho procesador incorporará las operaciones básicas suma y producto como base para la construcción de otros operadores por combinación. Se propone la arquitectura para la realización de dichos métodos de cálculo, teniendo en cuenta la propia implementación de los operadores elementales y el estudio de los elementos de memoria necesarios.

De forma esquemática, la funcionalidad que deberá proporcionar este coprocesador se resume a continuación:

- Almacenar números racionales sin error.
- Procesar números racionales de manera exacta para las operaciones suma y producto.

- Gestionar la precisión de los resultados en función de las restricciones existentes.

La operatoria se orienta hacia técnicas iterativas de procesamiento paralelo. Este trabajo engloba, tanto aspectos del ámbito de la investigación básica (concepción y especificación de modelos aritméticos de representación y procesamiento parametrizables), como aplicada (realización de las propuestas de procesador).

La posibilidad de realizar operaciones sucesivas sin acumular error apunta hacia una algoritmia donde la planificación de operaciones se realiza según parámetros de tiempo y precisión. Su gestión conjunta permite adaptar las capacidades del procesador a situaciones que interactúan fuertemente con el mundo real: sistemas de guiado por visión, sistemas de seguridad y control o en sistemas multimedia. En todos ellos se plantea el problema del procesamiento intensivo de datos proporcionados por periféricos. En estos casos se produce, necesariamente, una coordinación estrecha entre sensores y actuadores del propio dispositivo y de éste con el resto de los niveles del sistema, debiéndose profundizar en el determinismo de los tiempos de respuesta de los métodos empleados así como de la precisión de los cálculos. Ello, favorece su integración en la planificación junto con el resto de las tareas del sistema.

A continuación se trata el estado actual de la investigación referente a los métodos de representación numérica y los modelos de computación variable más relevantes para disponer de una base rigurosa sobre la que terminar de centrar los objetivos de la investigación. Posteriormente, se formaliza el problema y se propone la solución que será desarrollada a lo largo de esta memoria.

I - 3

Conocimiento actual y problemas abiertos

En este apartado se desarrolla el estado del arte acerca de la representación numérica en un computador y de su operatoria asociada. La relevancia de este problema se constata en la abundancia de propuestas y trabajos que tratan de resolver determinados aspectos del mismo. El análisis se realiza a varios niveles y abarca las soluciones de alto nivel, los formatos de codificación numérica, los modelos aritméticos para el cálculo y las unidades procesadoras que operan sobre estos esquemas. La exposición del estado actual de la investigación parte desde las herramientas software por su amplia utilización y flexibilidad, seguidamente se analiza la algoritmia para computación numérica de alta precisión y los métodos convencionales de representación para continuar con aquellas propuestas a bajo nivel que tratan de implementar capacidades específicas que no proveen los sistemas convencionales.

La práctica más habitual consiste en recurrir a herramientas software de precisión variable por razones de portabilidad. Esta solución forma una capa de abstracción por encima del hardware del computador que dota al sistema de estructuras y operaciones de longitud ajustable a las necesidades de cada problema. Entre estas herramientas se encuentran librerías de funciones y tipos de datos, paquetes aritméticos y otras extensiones de los lenguajes de programación comunes (C, Pascal, Fortran, MatLab) [Press et al, 1994], [Aberth y Schaefer, 1992], [Klatte et al, 1991], [Smith, 1991]. Su principal contribución se encuentra en ampliar el espacio de memoria de la representación para almacenar una mayor cantidad de cifras significativas y alcanzar una mayor precisión. Sin embargo, la complejidad en el cálculo de las operaciones limita su uso para el caso general. Por otra parte, la representación última se apoya en formatos y esquemas de representación numérica soportados por el hardware de la máquina que propaga en cierta medida sus restricciones.

La algoritmia para el cálculo de alta precisión está dirigida generalmente a operar con datos de considerable longitud y a compensar las limitaciones de los formatos de representación estándar. En este último aspecto cobran especial interés las propuestas de procesamiento numérico decimal que evitan los errores de conversión a binario en la introducción de datos [Cowlishaw, 2003], [EU, 1999], [Bohlender, 1991]. Para el cálculo de operandos de un número elevado de cifras destacan los métodos basados en procedimientos que reducen su tamaño [Karatsuba y Ofman, 1963], [Toom, 1963] y los métodos recursivos de obtención progresiva de cifras significativas del resultado. En este conjunto se encuentran los conocidos algoritmos de *Newton-Raphson*, *Goldschmidt* y *Taylor* [Ercegovac et al, 2000b], [Ito et al, 1997], [Schulte, 1994] y la *aritmética on-line* [Schneider et al, 2000], [McIlhenny y Ercegovac, 1999], [Muller, 1991], [Ercegovac y Trivedi, 1987], [Ercegovac y Trivedi, 1977]. Ésta última se caracteriza por realizar cada operación desde las posiciones más significativas a las menos significativas de los números. Algunos de estos algoritmos se incluyen en las herramientas software mencionadas, por lo que adquieren sus inconvenientes, si bien se han desarrollado propuestas hardware de su implementación que serán mencionadas más adelante.

Los formatos de codificación numérica vienen condicionados fundamentalmente por las características de los conjuntos numéricos a representar, existiendo distintos convenios y formatos según se representen números naturales, enteros o reales [Patterson y Hennessy, 2002], [Kornerup, 1994], [Omondi, 1994]. Para la representación de estos últimos, la notación fraccionaria posicional ofrece una expresión directa del número donde cada cantidad consiste en una parte entera y en una parte fraccionaria que indican fielmente su valor.

El esquema de representación fraccionaria más sencillo es el de coma fija. Este formato se caracteriza por destinar una cantidad determinada de memoria a la parte entera y a la parte fraccionaria de modo que se representan siempre las cifras del número en torno a su coma fraccionaria. Esta característica proporciona ventajas en el diseño de unidades aritméticas al utilizar en su procesamiento aritmética de enteros que posee una menor complejidad en los diseños [Patterson y Hennessy, 2002], aunque como inconveniente se destaca que sólo es capaz de representar de manera exacta números que coinciden con los elementos discretos codificables y realiza aproximaciones para el resto. Este formato es ampliamente utilizado en arquitecturas de procesadores digitales de señal DSP [Kim et al, 1998], [Inacio y Ombres, 1996].

Con el fin de abarcar un mayor intervalo de representación toman especial relevancia los sistemas de representación en coma flotante, en los que la parte entera se reduce a su mínima expresión, desarrollando ampliamente la parte fraccionaria. Esta representación se asemeja a la expresión científica de los valores numéricos reales y mejora la capacidad expresiva del formato de coma fija. [Patterson y Hennessy, 2002], [Sun Microsystems, 2000]. La codificación del número se estructura en tres campos: signo, exponente y mantisa. La mantisa constituye la parte significativa del número mientras que el exponente representa su orden de magnitud. Al igual que en la representación en coma fija, estos formatos establecen un esquema discreto de representación de los números reales en un computador. Como modelo característico de los esquemas de coma flotante cabe destacar la *norma de representación de datos en coma flotante IEEE-754* [IEEE, 1985], ampliamente adoptada por la gran mayoría de los sistemas informáticos. Este modelo describe varios formatos de representación, modos de redondeo y manejadores de excepciones.

En esos esquemas, llamados de precisión finita por utilizar una cantidad finita de bits en la representación, resulta patente la imposibilidad de representar cualquier valor fraccionario que supere la cantidad de cifras significativas del formato [Sun Microsystems, 2000]. La consecuencia es que se provoca un error en el cálculo [Goldberg, 1991] que dificulta resolver satisfactoriamente problemas que requieren una alta precisión en los resultados, como por ejemplo [Clark, 1998], [Hoffmann, 1989]: comunicaciones de alta frecuencia, simulaciones científicas, navegación y posicionamiento, etc.

En aquellos problemas que requieren de capacidades que desbordan los sistemas actuales, las investigaciones se orientan hacia diseños dedicados que tratan de cubrir a bajo nivel las carencias concretas existentes. Estas soluciones potencian determinadas características que no han sido suficientemente desarrolladas o bien añaden nuevas funcionalidades que las hacen apropiadas para resolver cierta clase de problemas. Ejemplos de propuestas de este tipo se encuentran en [Wolf y Franklin, 2002], [Villalba et al, 2002], [Mora, 2001], [Paulin et al, 2001], [Wires, 2000], [Nielsen y Kornerup, 1999] y [Nielsen, 1997]. La representación numérica no es ajena a esa situación y del mismo modo abundan las soluciones a medida válidas para determinados entornos. Las investigaciones realizadas se concretan en la búsqueda de métodos alternativos de expresión numérica y de cálculo aritmético implementados en diseños específicos de procesadores especializados. En lo referente a estas aportaciones, se presentan los avances realizados así como las carencias más relevantes. Se ha realizado una ordenación de las propuestas según su capacidad de expresión, entendiendo como tal, la precisión a la hora de expresar tanto los operandos como el resultado que produce su procesamiento.

En primer lugar, como método de mayor expresividad, se encuentra el modelo de computación simbólica, caracterizado por realizar una representación exacta de los datos con los que opera. Esta técnica, que constituye el esquema de *aritmética exacta* [Mencer, 2000], consiste en expresar las operaciones matemáticas mediante expresiones algebraicas que corresponden con valores numéricos. Basándose en estos principios se dispone del diseño de una unidad aritmética [Kornerup y Matula, 1983a], [Kornerup y Matula, 1983b] en la que se realiza la representación de los números de forma simbólica mediante fracciones. Se contemplan las operaciones básicas de suma, resta, multiplicación y división, operando sobre los numeradores y denominadores de las fracciones con aritmética de enteros. En esta propuesta se representan y operan los números racionales sin error, presentando igualmente una expresión analítica del resultado. No obstante, el procesamiento con estas expresiones es costoso computacionalmente en especial cuando no se encuentran formas de simplificación [Buchberger, 1991]. Además, para expresar el resultado mediante un único valor numérico es necesario realizar una operación de división que puede provocar errores de aproximación.

Las fracciones continuas también ofrecen un método de representación exacta del conjunto de los números racionales. En este caso se realiza la codificación de los números mediante sucesivas fracciones de enteros [Brezinski, 1980], [Moore, 1964]. A pesar de ello, los diseños hardware realizados [Mencer, 2000], [Mencer et al, 1999], [Seidensticker, 1983], [Robertson y Trivedi, 1977], revelan una alta complejidad de las operaciones aritméticas [Vuillemin, 1990] y, al igual que en la computación simbólica, en caso de que se quiera un resultado numérico se deben realizar operaciones adicionales cometiéndose imprecisiones al transformar las expresiones.

Una interesante propuesta para la codificación sin error de números racionales consiste en la representación explícita del desarrollo periódico de los números fraccionarios [Hehner y Horspool, 1979]. Sin embargo este trabajo se limitó sólo a su formulación y no se presentaron procedimientos adecuados para su procesamiento ni arquitecturas que lo implementaran. En consecuencia, no captó el suficiente interés por la comunidad científica y fue abandonado al ser una propuesta carente de realismo.

Otro conjunto de propuestas se fundamenta en un modelo de aritmética de intervalos. El objetivo que persiguen consiste en acotar la imprecisión cometida al proveer un resultado numérico. En este caso, un número viene expresado por los extremos del intervalo en el que se encuentra, los cuales son codificados mediante notación en coma flotante [Arnold et al, 2003], [Schulte, 2000], [Hormigo el al, 2000], [Schulte y

Swartzlander, 2000], [Hormigo et al, 1999], [Sáez et al, 1998], [Knuppel, 1994], [Neumaier, 1990], [Alefeld y Herzberger, 1983], [Moore, 1979]. Las operaciones aritméticas se realizan sobre los extremos del intervalo conservando los resultados incluidos entre sus límites. Los diseños hardware basados en esta técnica mejoran notablemente el coste temporal frente a paquetes software de simulación con la misma funcionalidad. Como ejemplos representativos se destacan los siguientes:

- La *arquitectura AIX* [Kolla et al, 1999] empaqueta los extremos de los intervalos en la misma palabra del computador para que sea tratada en paralelo por la unidad aritmética. Se sugiere el uso de datos de doble precisión para almacenar cada intervalo de manera que el cálculo sea realizado en una unidad compatible con la norma de IEEE. Su principal limitación se encuentra en que la cantidad de cifras destinadas a representar cada intervalo es fija y acotada por un formato de representación de precisión finita.
- El *procesador VP de aritmética de intervalos* [Schulte, 2000], [Schulte, 1996] consiste en una estructura de representación en coma flotante en la que el campo que almacena la mantisa admite una longitud variable aunque limitada por el propio formato. Ofrece una precisión considerablemente mayor que el modelo anterior utilizando intervalos más estrechos. Sin embargo, aunque se aumente la precisión de los extremos, el grado de aproximación está acotado por la cantidad de cifras significativas a representar.

El mayor inconveniente de esta técnica radica en la renuncia a la obtención de un único valor exacto, provocando falta de precisión al producir intervalos demasiado amplios como consecuencia de determinadas operaciones y cálculos. No obstante, se produce un avance con respecto a las técnicas anteriores al mantener el error del resultado numérico acotado por los límites del intervalo. Aprovechando esta circunstancia se han realizado propuestas combinadas entre la aritmética de intervalos y la notación simbólica: la técnica de aritmética relajada, comúnmente conocida como *lazy arithmetic* [Michelucci y Moreau, 1997], [Benouamer et al, 1993], describe los resultados mediante una expresión matemática simbólica y al mismo tiempo los expresa numéricamente a través de un intervalo. Con este procedimiento, en caso de requerir más precisión se realizan de nuevo los cálculos obteniendo una mejor aproximación.

Otras investigaciones tratan de incrementar la precisión de los valores numéricos con los que trabajan a costa de utilizar una mayor cantidad de cifras significativas para representar los operandos y los resultados. De este modo se pretende alcanzar una aproximación *adecuada* a las necesidades del problema sobre el que se aplica. No obstante, aunque estos sistemas proporcionan una única expresión numérica de los datos, son incapaces de codificar números con una cantidad infinita de cifras fraccionarias, perdiendo la representación exacta de valores racionales y cometiendo errores en su procesamiento. En este conjunto de propuestas se encuentran aquellas basadas en la *aritmética escalonada* y la *aritmética on-line*.

La *aritmética escalonada* [Priest, 1991], [Dekker, 1971] representa cada número mediante una lista variable de valores en coma flotante no solapados. La suma de estos elementos proporciona el valor del número que codifican. Las operaciones aritméticas se realizan sobre esa lista de números, obteniendo como resultado otro valor representado con el mismo formato. Esta propuesta ha sido implementada en el *procesador para aritmética de intervalos escalonada* [Schulte y Swartzlander, 1995]. Su principal ventaja es la de poder usar las unidades de coma flotante convencionales para datos en formato estándar de IEEE, mientras que como inconvenientes se tiene que la precisión de los valores escalonados es limitada y el proceso de cálculo y conversión a este formato es complicado y costoso. Por ejemplo, la simple comparación entre dos números se convierte en un algoritmo complejo por el hecho de que para un mismo número pueden existir múltiples representaciones distintas.

En la *aritmética on-line* los operandos se introducen en serie y el procesamiento se realiza dígito a dígito a partir del conocimiento parcial de los datos de entrada. De esta forma pueden diseñarse métodos de cálculo segmentados que permitan un encadenamiento de las operaciones y, además, debido a sus propiedades iterativas, es posible establecer estructuras regulares que pueden ser usadas para el cálculo de las operaciones elementales con un número variable de cifras. Entre las propuestas que se basan en este tipo de aritmética se encuentran varios diseños:

- El *coprocesador JANUS* [Guyot et al, 1989] considera una precisión máxima de 600 dígitos, implementando tan sólo la operación de multiplicación.
- El *coprocesador VLP* [Ferreira, 1998] consiste en un coprocesador aritmético desarrollado bajo plataforma FPGA con capacidades de reconfiguración dependiendo de las operaciones a realizar.

Finalmente, con el mismo objetivo de aumentar la cantidad de cifras significativas de los números, existen variantes hardware capaces de trabajar con datos de una cantidad variable de dígitos:

- El *procesador CACAD* [Cohen et al, 1983] realiza la codificación de los datos mediante palabras de longitud variable. Cada palabra contiene los campos de signo, exponente, longitud de la mantisa, mantisa y un indicador. La mantisa se representa en BCD [Hull et al, 1991]. En su diseño se evitan los errores de representación en las interacciones de entrada y salida consecuencia de las transformaciones decimal a binario. Presenta como inconvenientes la complejidad adicional de la unidad aritmética al operar con datos expresados en BCD así como la limitación en la precisión representable.

- El *coprocesador VP para FPGA* [Hsu, 1996] supone un avance respecto al diseño anterior al no limitar la cantidad de cifras de la mantisa significativa del número. En este caso, se utiliza una estructura basada en una cantidad variable de palabras de 64 bits, destinando una cantidad fija de cifras a la representación de la mantisa y al exponente en cada bloque. Se da la posibilidad de concatenar varias palabras hasta completar la representación del número. Asimismo, se describen los algoritmos necesarios para operar con este tipo de datos, detallando los accesos a la memoria en la que se almacenan.

Las representaciones aproximadas con una cantidad determinada de cifras, entre las que se incluyen los formatos de representación convencionales, suelen disponer de métodos de refinamiento o redondeo de la codificación que producen con la misión de ajustarla al espacio disponible y cometer así el mínimo error. El efecto que provocan es la modificación de las cifras menos significativas de la representación respecto a las cifras exactas del número. Entre los criterios de redondeo más usuales se encuentran el redondeo por exceso, por defecto, al más cercano y el redondeo al par [Bruguera y Lang, 2000], [Park et al, 1999], [Quach et al, 1991] y [IEEE, 1985]. El procedimiento que se utiliza en su instrumentación está estrechamente relacionado tanto con el método de representación que se utilice como con la operación que provea el resultado a redondear [Even y Seidel, 2000], [Oberman, 1996].

La investigación en procesamiento numérico y precisión variable, frecuentemente, forma parte de algún proyecto de diseño y desarrollo de un dispositivo de procesamiento específico. En el entorno geográfico más cercano se mencionan los siguientes grupos de investigación con trabajos en la materia:

El *departamento de arquitectura de computadores* de la *Universidad de Málaga* realiza investigaciones para el desarrollo de procesadores especializados en cálculo numérico. Con este objetivo se destacan los trabajos encaminados a concebir una arquitectura de precisión variable basada en aritmética de intervalos [Hormigo el al, 2000], [Hormigo et al, 1999], [Sáez et al, 1998].

El *grupo de arquitectura de computadores* de la *Universidad de Santiago de Compostela* se centra en varios aspectos de la disciplina destacando sus trabajos sobre la aritmética del computador. En esta materia desarrollan algoritmia de cálculo de altas prestaciones para procesadores tanto de propósito general como de aplicación específica [Piñeiro y Bruguera, 2002], [Piso et al, 2002], [Bruguera y Lang, 2001].

En el ámbito internacional se referencian a continuación los grupos de investigación más relevantes en esta línea:

El *laboratorio de arquitectura reconfigurable y aritmética digital* de la *Universidad de California en Los Ángeles (Estados Unidos)* destaca por su profunda investigación en nuevos métodos de cálculo de funciones matemáticas [Benowitz et al, 2002], [Ercegovac et al, 2000a], [Ercegovac et al, 2000b]. Sobresalen sus aportaciones en la aritmética on-line y en el procesamiento en precisión variable [Schneider et al, 2000], [McIlhenny y Ercegovac, 1999], [Tenca y Ercegovac, 1998], [Ferreira, 1998].

El *laboratorio de arquitectura de computadores e investigación aritmética* de la *Universidad de Lehigh en Pennsylvania (Estados Unidos)* trabaja en la realización de un proyecto para dotar de soporte hardware a la operatoria de precisión variable basándose en aritmética de intervalos [Arnold et al, 2003], [Schulte et al, 2000], [Schulte, 2000], [Schulte y Swartzlander, 2000].

El *grupo de arquitectura y aritmética de computadores* de la *Universidad de Stanford (Estados Unidos)* se orienta al desarrollo de un procesador *Stanford Nanosecond Arithmetic Processor* (proyecto SNAP) en el que se presta especial atención al tiempo de respuesta de las operaciones aritméticas y su implementación [Fanhm y Flynn, 2003], [Liddicoat y Flynn, 2001]. Se realizan otras investigaciones relacionadas con la concepción y el diseño de unidades aritméticas sobre números racionales con operatoria basada en fracciones continuas [Mencer, 2000], [Mencer et al, 1999].

El *laboratorio de informática* de la *Escuela Superior de Lyon (Francia)*, trabaja en el proyecto *Arenaire* para la creación de conocimiento en el área de aritmética de computadores. En este contexto se desarrolla investigación en métodos numéricos y en cálculo en precisión variable así como en la construcción de librerías y funciones software para su procesamiento [Boldo y Daumas, 2003], [Lefevre y Muller, 2003], [Muller, 2003].

El *departamento de matemáticas e informática* de la *Universidad de Southern Denmark (Dinamarca)* realiza investigación en el ámbito de sistemas de representación numérica y unidades aritméticas de procesamiento matemático intensivo. En esta línea destaca su aportación al procesamiento racional basado en fracciones continuas y representación simbólica de fracción [Kornerup, 2003], [Nielsen y Kornerup, 1999], [Nielsen, 1997].

El *laboratorio de sistemas informáticos avanzados* de la *Universidad de California en Davis (Estados Unidos)* se centra en estudiar la relación entre los algoritmos aritméticos y la tecnología en la que se implementan con el propósito de mejorarlos en función de la misma [Oklobdzija et al, 2003], [Yu et al, 2003], [Nedovic et al, 2002].

Otros grupos destacan por su importante investigación en arquitecturas aritméticas y en el desarrollo de algoritmia para el procesamiento de operadores matemáticos. En este conjunto se menciona el *laboratorio de procesadores numéricos de la Universidad de California en Irvine (Estados Unidos)* [Antelo et al, 2002], [Lang y Antelo, 2001]; el *departamento de informática e ingeniería* de la *Universidad Southern Methodist de Dallas (Estados Unidos)* [Paul y Seidel, 2003], [Even et al, 2003], [Seidel et al, 2001]; el *departamento de sistemas de información* de la *Universidad de Nagoya (Japón)* [Kaihara y Takagi, 2003], [Takagi y Horiyama, 1999], [Takagi, 1998] y el *grupo de arquitecturas y redes de computadores* de la *Universidad de Torino (Italia)* [Montuschi y Lang, 2001], [Montuschi y Lang, 1999], [Sanna et al, 1998].

Finalmente se señala el *laboratorio de arquitectura y sistemas de tiempo real* de la *Universidad de Massachusetts (Estados Unidos)* por su investigación en nuevas arquitecturas y algoritmos de alto rendimiento con características de tiempo real [Koren et al, 2003], [Bertoni et al, 2003], [Lakamraju et al, 2002].

Debido al carácter específico de esta materia cobran especial importancia los foros internacionales relacionados a los que acuden expertos en la disciplina así como destacados miembros de los grupos de investigación anteriores para compartir ideas, intercambiar opiniones y críticas acerca de sus investigaciones. Por su relación directa con este tema destaca el *Symposium on Computer Arithmetic* (Arith) de periodicidad bianual y la *Conference on Real Numbers and Computers* (RNC). Asimismo, se celebran otros eventos con sesiones dedicadas a arquitectura y aritmética de computadores entre los que se mencionan los siguientes: *Symposium on Digital Signal Design* (DSD), *International Conference on Electronics Circuits and Systems* (ICECS), *International Conference on Computer Design* (ICCD), *Conference on Application-specific Systems, Architectures and Processors* (ASAP), *Conference on Very Large Scale Integration* (VLSI SoC), *Symposium on High Performance Computer Architecture* (HPCA) y, *Conference on Design of Circuits and Integrated Systems* (DCIS).

Con respecto a las revistas de divulgación científica relacionadas con la materia se destacan por su alta relevancia e impacto para la comunidad investigadora el *Journal of the ACM* [JACM, ISSN: 0004-5411] y el *IEEE Transactions on Computers* [IEEETC, ISSN: 0018-9340]. Existen otras publicaciones de contenido más específico entre las que se señalan las siguientes: IEEE *Transactions on Very Large Scale Integration Systems* [IEEEVLSI, ISSN: 1063-8210], *Journal of Systems Architecture* [JSC, ISSN: 1383-7621], *Journal of Computer and System Sciences* [JCSS, ISSN: 0022-0000] y, *Journal of Circuit Systems and Computers* [JCSC, ISSN: 0218-1266].

Conclusiones

La revisión de las diferentes metodologías así como de las estrategias para hacer frente a las nuevas necesidades computacionales pone de manifiesto que las propuestas *a medida* se abren camino como método válido y ampliamente aceptado de solución. Las técnicas convencionales de representación numérica y formatos estándar reflejan la dificultad de representar y, por tanto de operar, números de infinitas cifras fraccionarias. De las investigaciones realizadas se concluye que mejoras en la capacidad de expresión de nuevos métodos y formatos chocan frecuentemente con aumentos considerables en la complejidad de las operaciones derivadas de su utilización, al margen de que para muchos problemas y aplicaciones, resulta absolutamente necesario disponer de una cantidad numérica que represente el valor del resultado.

Las propuestas software de representación y operación, a pesar de que constituyen soluciones versátiles compatibles con la mayoría de sistemas para la construcción de aplicaciones especializadas, no dan la talla en cuanto a las exigencias de rendimiento necesarias en ciertas aplicaciones. Con respecto a las soluciones a bajo nivel, las más recientes propuestas basadas en la aritmética de intervalos y los métodos que manejan una cantidad variable de cifras significativas de los números ofrecen alternativas de interés que mejoran la precisión de los resultados. Sin embargo, no son capaces de capaces de proveer un valor exacto sino, tan sólo, una aproximación para la que, en muchas ocasiones, se carece incluso de una medida de su calidad.

En definitiva, la exigencia para muchos problemas de disponer de una gran precisión en sus cálculos a un coste computacional aceptable pone de manifiesto la necesidad de realizar una expresión apropiada de los operandos y desarrollar la algoritmia a bajo nivel correspondiente. Por todo lo anterior, la concepción de un método de codificación exacta de los operandos basado en su representación numérica constituye un avance respecto a los métodos actuales ya que, desde el primer momento, se conoce el valor de los datos y los resultados parciales que se van generando, permitiendo concebir políticas de ajuste o aproximación eficaces en caso de que los requerimientos del problema lo permitan. El camino hacia estos objetivos pasa por establecer un marco formal adecuado en el que se encuadre la definición del problema y la especificación de la solución, cuestiones éstas que se abordarán en el siguiente apartado.

I - 4

Formulación del problema y propuesta de solución

La definición del problema en términos formales elimina ambigüedad y establece el soporte expresivo necesario para presentar las posibles soluciones. El estudio se centra en los aspectos concretos del procesador aritmético. En este sentido, tal y como se ha señalado en los objetivos, se trata de avanzar en el diseño de unidades aritméticas con propiedades específicas de representación y operatoria exacta con control sobre la precisión de números racionales.

Definición del problema

Como paso previo a la formalización de la unidad aritmética, se definen los componentes necesarios que describen el modelo de cálculo así como los elementos que influyen en el grado de precisión de los resultados obtenidos ya sea fruto de la propia codificación realizada o por el funcionamiento de los operadores empleados.

Sea f una función matemática genérica. Se denomina *función instrumentación* Γ *de* f a cualquier función computable cuyo resultado se aproxima a f según una implementación particular:

$$\mathrm{codominio}(\Gamma_f(\overline{x})) \subseteq \mathrm{codominio}(f(\overline{x})) \quad [1.1]$$

tal que,

$$\forall\ \overline{x} \in \mathrm{dominio}(\Gamma_f),\ |\Gamma_f(\overline{x}) - f(\overline{x})| \leq \varepsilon \quad [1.2]$$

donde,

$\bar{x}$:Denota al conjunto de parámetros sobre los que opera la función.

ε : $\varepsilon \in \mathbb{R}^+ \cup \{0\}$. Indica el grado de aproximación de f por Γ.

El cometido general de una unidad aritmética es el de procesar funciones matemáticas. Una arquitectura A se caracteriza tanto por el conjunto de funciones que proporciona como por la forma en la que las instrumenta. La conjunción de estos dos aspectos constituye un procesador.

Dado el siguiente conjunto de funciones:

$$\Phi = \{f_1, f_2, \ldots, f_n\} \quad [1.3]$$

Una arquitectura A_Φ que proporcione dichas funcionalidades estará formada por:

$$A_\Phi = \{\Gamma_{f1}, \Gamma_{f2}, \ldots, \Gamma_{fn}\} \quad [1.4]$$

Es decir, A_Φ contendrá las instrumentaciones concretas de las funciones de Φ, donde cada Γ_{fi} produce una aproximación a su f_i correspondiente. Conforme lo mencionado anteriormente, cada f_i supone el objetivo a alcanzar por el procesador, mientras que cada Γ_{fi} corresponde con la función que finalmente se proporciona. La instrumentación de una función f no tiene por qué ser única, pueden existir varias instrumentaciones de una misma función que representen distintas aproximaciones a f, por ejemplo con un valor diferente de ε.

A lo largo de esta memoria se maneja el término instrumentación en modo general, haciendo entender que una función proporciona un resultado dependiente de su propia implementación. De este modo, se da el caso de la existencia de múltiples arquitecturas donde cada una de ellas contiene una serie de instrumentaciones concretas para un conjunto de funciones Φ. Asimismo, es posible que una misma arquitectura contenga varias instrumentaciones para una misma función f, prestando varios grados de aproximación a la función, por ejemplo las distintas implementaciones de la suma aritmética para formatos diferentes.

A continuación se define el significado de *precisión variable* y su efecto en los conceptos definidos hasta ahora.

Se denomina *función instrumentación* Γ *en precisión variable (*VP *—Variable Precision) de* f a cualquier función computable cuyo resultado tiende a f según el valor de ciertos parámetros $\overline{d}$ y de acuerdo con una instrumentación particular:

$$\Gamma_f^{VP}(\overline{x}, \overline{d}) \qquad [1.5]$$

tal que,

$$\forall\ \overline{x} \in \mathrm{dominio}(\Gamma_f),\ |\,\Gamma_f^{VP}(\overline{x}, \overline{d}) - f(\overline{x})| \le \varepsilon \qquad [1.6]$$

donde $\overline{x}$ y ε tienen el mismo significado que en la expresión [1.2].

La función Γ debe ser lo suficientemente general como para admitir diversas aproximaciones de f. No obstante, para que sea considerada función VP, el grado de similitud entre los resultados proporcionados por ambas funciones debe estar ligado al valor de $\overline{d}$.

Es preciso resaltar que el grado de aproximación de f por Γ_f^{VP} está relacionado con la función f, la propia instrumentación Γ y el valor $\overline{d}$. Para determinadas instrumentaciones Γ_f^{VP} sólo unos valores de $\overline{d}$ realizan una aproximación, obteniendo el valor exacto de f para el resto. Otras funciones sin embargo, no producen nunca el resultado sin error para ninguna instancia de $\overline{d}$.

Se dice que una función Γ_f^{VP} *tiene una evaluación exacta de* f si existe algún valor de $\overline{d}$, tal que $\varepsilon = 0$. Según la ecuación [1.6] se deduce que:

$$\forall\ \overline{x} \in \text{dominio}(\Gamma_f),\ \exists\ \overline{d} /\ \Gamma_f^{VP}(\overline{x}, \overline{d}) = f(\overline{x}) \qquad [1.7]$$

Por el contrario, se dice que una función Γ_f^{VP} *no tiene una evaluación exacta de* f si para ningún valor de $\overline{d}$ se alcanza el valor de f.

$$\forall\ \overline{x} \in \text{dominio}(\Gamma_f),\ \neg\exists\ \overline{d} /\ \Gamma_f^{VP}(\overline{x}, \overline{d}) = f(\overline{x}) \qquad [1.8]$$

Se dice que una *arquitectura es de precisión variable* A^{VP}, si al menos una de las funciones que instrumenta es de precisión variable:

$$\exists\ \Gamma_{fi} \in A_\Phi.\ \Gamma_{f_i}^{VP} \Rightarrow A_\Phi^{VP} \qquad [1.9]$$

Para esta arquitectura, existirá al menos un conjunto de parámetros $\overline{d}$ tal que influya en el resultado de ciertas operaciones y varíe el grado de aproximación a f. De esta forma, es posible en esta arquitectura actuar sobre el valor de $\overline{d}$ para obtener diversa precisión en los resultados.

Asimismo, si la arquitectura de precisión variable A^{VP}, contiene alguna función con evaluación exacta, se dice que dicha arquitectura instrumenta de manera efectiva esa función. Es decir, cumple el objetivo de proporcionar su resultado sin error para cualquier instancia de sus operandos.

Por razones de escalabilidad de los sistemas, el diseño de una arquitectura puede organizarse en niveles. A partir de un núcleo elemental P, el de las primitivas del procesador, se construyen las demás funciones, es decir, las derivadas.

Dado el conjunto de funciones [1.3]:

$$\Phi = \{f_1, f_2, \ldots, f_n\}$$

se dice que la arquitectura

$$A = \{\Gamma_{f1}, \Gamma_{f2}, \ldots, \Gamma_{fs}\}_P \qquad [1.10]$$

constituye un *conjunto de funciones primitivas*, si y sólo si la instrumentación de cada una de dichas funciones se realiza de forma independiente, es decir:

$$A = \{\Gamma_{f1}, \Gamma_{f2}, \ldots, \Gamma_{fs}\}_P \Leftrightarrow \neg\exists\, \Gamma_{fi} \in A.\ \Gamma_{fi} = \bigotimes_{i}(A) \qquad [1.11]$$

donde la expresión $\bigotimes_{i}(A)$ denota cualquier composición de las funciones de A, sin contar la función i.

Si alguna de las funciones primitivas es de precisión variable, entonces la arquitectura formada por esas primitivas será también de precisión variable.

$$\exists\, \Gamma_{fi} \in A.\ \Gamma_{f_i}^{VP} \Rightarrow A^{VP} \qquad [1.12]$$

A partir de estas funciones elementales, el conjunto Φ de funciones podrá ser instrumentado por una arquitectura que contenga, además, una serie de funciones derivadas a partir de las primitivas.

$$A_\Phi = \{\Gamma_{f1}, \Gamma_{f2}, \ldots, \Gamma_{fs}\}_P \cup \{\Gamma_{fs+1}, \ldots, \Gamma_{fn}\} \qquad [1.13]$$

donde,

$$\forall\, \Gamma_{fi} \in \{\Gamma_{fs+1}, \ldots, \Gamma_{fn}\}.\ \Gamma_{fi} = \bigotimes_{i}(A_\Phi) \qquad [1.14]$$

Si el conjunto de funciones primitivas es de precisión variable, la arquitectura completa también lo será:

$$A^{VP} \Rightarrow A_\Phi^{VP} \qquad [1.15]$$

Por otra parte, si la propia instrumentación de una función derivada es de precisión variable, la arquitectura también será de precisión variable, independientemente de que el conjunto de primitivas lo sea.

Desde el punto de vista de la arquitectura de los computadores se considera que el procesador de una arquitectura dada está dotado de una estructura interna consistente en sus módulos estructurales y en la lógica de relación entre ellos.

$$\Pi = \Pi\ (M, \Lambda) \qquad [1.16]$$

donde,

M: Conjunto de módulos estructurales.

Λ: Lógica de relación entre los módulos estructurales.

Los criterios de diseño de M y Λ deberán ser coherentes para preservar sus características y mantener sus propiedades en su funcionamiento conjunto, de modo que la característica VP de la *arquitectura aritmética* sea mantenida y apoyada por todos los componentes para que sea efectiva. De esta forma, para que un procesador sea VP, Π^{VP}, deberá contener un núcleo funcional cuya implementación sea VP y mantener esta propiedad en toda su arquitectura. Teniendo en cuenta todo lo anterior, el enunciado del problema es el siguiente:

Definir un procesador de precisión variable, Π^{VP}.

En particular, la definición de un procesador aritmético para operandos racionales en precisión variable, Π^{VP}, con operaciones que admitan una evaluación sin error.

Las funcionalidades que proporcionará dicho procesador serán:

$$\Phi_Q = \{\text{identidad, suma, producto}\} \qquad [1.17]$$

Las operaciones en precisión variable que instrumenta dicho procesador son las siguientes:

$$P^{VP} = \{\Gamma^{VP}_{\text{identidad}},\ \Gamma^{VP}_{\text{suma}},\ \Gamma^{VP}_{\text{producto}}\}_P \qquad [1.18]$$

De tal forma que:

$$\Pi^{VP}_{\Phi_Q} = P^{VP} \qquad [1.19]$$

El problema se concreta en concebir métodos VP para evaluar cada una de las operaciones aritméticas y en incorporar los elementos necesarios para una correcta instrumentación que permitan su evaluación exacta.

El procesador debe plasmar una arquitectura que instrumente de manera totalmente efectiva el conjunto de funciones sobre números racionales. Se propone implementar la función *identidad* para expresar la capacidad misma del procesador de representar y operar con números racionales. En la práctica, esta función corresponde con el formato de codificación numérico empleado, que representa sin error cualquier número racional.

Propuesta de solución

En términos generales, la solución pasa por desarrollar arquitecturas específicas que doten al computador de características que aporten propiedades de computación numérica exacta sobre los números racionales sin perjuicio de tolerar distintos niveles de aproximación al resultado. Cabe incorporar un conjunto de instrucciones con atributos de VP apoyado por otros elementos como la codificación y almacenamiento conveniente de los datos que favorezcan el funcionamiento correcto.

El punto de partida es la especificación de un formato de representación capaz de expresar cualquier elemento del conjunto de los números racionales. De esta forma se consigue su formulación exacta y única, así como su posterior manipulación. La idea parte de la combinación del método presentado por Hehner y Horspool [Hehner y Horspool, 1979] y los formatos de coma flotante. Su principal característica se encuentra en codificar cualquier valor perteneciente al conjunto de los números racionales mediante una notación fraccionaria normalizada que represente de forma separada las cifras fijas de las periódicas. Este esquema aporta una representación uniforme de los números racionales con una cantidad variable de cifras en cualquiera de sus campos pero que en todo caso será finita. Este hecho permite una representación compacta de los valores racionales periódicos sin recurrir a la notación simbólica de fracción. Adicionalmente, el esquema permite conocer directamente el valor del número y ofrece una cantidad de cifras fraccionaria ilimitada. El formato alcanza a la representación de valores extremos mediante un campo para el exponente que contiene el orden de magnitud de los números. Por otra parte, la codificación exacta de valores racionales abre otras vías de operación, dando lugar a enfoques que resuelvan sin error un conjunto de problemas del cálculo.

El almacenamiento de los datos expresados en este esquema condiciona su procesamiento. Para aprovechar todo su potencial resulta conveniente disponer de una organización flexible de memoria que permita manejar estructuras variables. Su instrumentación a bajo nivel en un contexto de registros de longitud fija sugiere la búsqueda de estrategias que aporten la flexibilidad necesaria para mantener las propiedades VP de las instrucciones desde el nivel más bajo de la arquitectura. Se propone la utilización de punteros que marquen la separación entre los campos del número, la incorporación de varios juegos de registros con distintas longitudes o la capacidad de reconfiguración de algunas de las características del espacio de almacenamiento.

El diseño de las operaciones se fundamenta en el uso de esquemas iterativos y de resultados precalculados. La composición de estas dos técnicas va a permitir la instrumentación VP de funciones que, junto con los elementos anteriores, constituirán el procesador racional flexible. En este sentido, la búsqueda de estrategias para mejorar el rendimiento de las operaciones se concreta en el aumento de la granularidad de los operandos elementales utilizando en su construcción memorias que contienen los resultados precalculados de su ejecución. Aunque estas técnicas operan al nivel de bloque, el almacenamiento de los datos precalculados impone severas restricciones a su tamaño, lo cual sugiere a su vez la división de las partes y operar con los bloques individualmente, componiendo el resultado por concatenación.

Mediante la integración de todos esos aspectos, la propuesta de solución constituye un modelo que abarca tanto las representaciones de los datos como los métodos de operación en sí, dando lugar a un esquema de cálculo exacto adaptable a las necesidades de cada problema. Algunas de las propiedades que posee este modelo son: capacidad de representación exacta de los números racionales, lo cual facilita su manipulación y procesamiento sin error; predecibilidad en el tiempo de cálculo, debido al carácter finito de la codificación y; sistematicidad del esquema algorítmico debido al carácter repetitivo de los algoritmos, lo cual propicia un alto grado de paralelismo.

Finalmente, la investigación se materializa en la propuesta de un procesador aritmético flexible que utiliza métodos que sistematizan la implementación de funciones de bajo nivel y emplean un formato flexible de representación de valores numéricos. Se construye un repertorio de instrucciones aritméticas cuyo procesamiento evoluciona iterativamente hacia su valor exacto. Se diseñan las primitivas suma y producto orientadas hacia el uso de memorias con resultados precalculados y la primitiva identidad que consiste en la representación de un número en el formato de codificación propuesto. Se deja para trabajos futuros la instrumentación de otras primitivas.

En la siguiente figura se muestra un esquema que resume el diseño de la unidad. Existe un conjunto de parámetros $\overline{\mathrm{d}}$ que condiciona el grado de precisión de los resultados y gradúa la aproximación realizada en función de los requerimientos del problema.

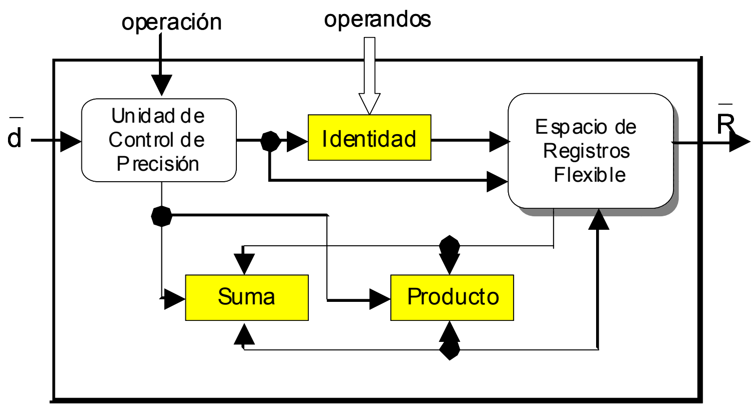

Figura 1-1: Esquema general del Procesador Racional Flexible

El sistema dispone de una unidad de control de precisión que, atendiendo a la operación solicitada y al grado de aproximación requerido, establece las características del conjunto de registros que intervienen en el procesamiento y ajusta los operadores a su precisión.