REQUISITOS BÁSICOS DE UM COMPUTADOR DEDICADO À ARITMÉTICA DE INTERVALO

M.B.Correia, E.N. da Silva, C.H.F.Carvalho

SUMÁRIO

Definição dos requisitos necessários ou mesmo desejáveis para um computador dedicado à Aritmética de Intervalo. Considera-se aspectos relacionados a unidade aritmética e possibilidades de se explorar uma arquitetura paralela.

1. INTRODUÇÃO

A invenção do computador decorreu em parte do desenvolvimento das máquinas de calcular. Com a evolução das unidade aritméticas, surgiram os computadores científicos capazes de processar números em ponto-flutuante.

Em relação ao processamento numérico, os desenvolvimentos subseqüentes permitiram a redução do tempo de computação. Explorou-se os mecanismos de "pipeline", as possibilidades de unidades aritméticas em paralelo e o multiprocessamento. Essa tendência persiste nos dias atuais levando ao aparecimento de novas arquiteturas e algorítmos paralelos.

Os problemas decorrentes da inexatidão da aritmética ponto-flutuante se agravaram com o aumento da capacidade de processamen-

to das máquinas e com a proliferação dos sistemas de ponto flutuante. Mesmo a flexibilidade de certas unidades aritméticas pode vir a ser uma desvantagem, não sendo elas operadas de modo conveniente.

No caso específico dos microcomputadores, onde a portabilidade do software é um fator importante, partiu-se para o estabelecimento de um sistema de ponto flutuante padrão. Nos demais casos, as máquinas diferem entre si no modo de realizar o arredondamento, tratar o "overflow" e o "underflow", representar os números etc...

Dentre os métodos para avaliação do erro decorrente da utilização da aritmética ponto flutuante, o mais usual se baseia na suposta invariância do erro relativo. Esta propriedade não sendo satisfeita, os resultados podem ser desastrosos. Os métodos alternativos são em geral bastante sofisticados e possuem uma complexidade que cresce exponencialemnte com o grau de complexidade do problema, restringindo suas aplicações a problemas simples.

Nesse contexto surgiu a aritmética de Intervalo como uma ferramenta de verificação automática do erro. Um intervalo consiste num conjunto limitado de números reais dados por:

$$A = [\underline{A}, \overline{A}] = \{x : \underline{A} \leq x \leq \overline{A}\}$$

Um intervalo também pode ser visto como um novo tipo de número representado por um par ordenado de reais que corresponde aos seus pontos extremos: limite superior e inferior.

Nos métodos intervalares, os reais são representados por intervalos definidos de modo que os resultados exatos do processamento com números em ponto flutuante estarão contidos pelos intervalos que lhes representam. Estima-se o erro associado a um certo resultado a partir da largura do intervalo.

As principais críticas à Aritmética de Intervalo decorre dos seguintes problemas: resultados pessimistas e tempo de computação.

Um resultado é considerado pessimista se a largura do intervalo for muito grande. Esse problema pode ser contornado adotando-se a aritmética de computação científica desenvolvida pelo Kulisch et alli. A idéia consiste em se equipar e computar com recursos que permitam o controle e a validação do processo computacional. A obtenção de tais capacidades estando condicionada aos seguintes requisitos:

1) Implementação das operações básicas em ponto flutuante (soma, subtração, multiplicação e divisão) com exatidão máxima;
2) Implementação do produto escalar de dois vetores com exatidão máxima;
3) Implementação das operações básicas no espaço dos intervalos;
4) Uso de técnicas de correção residual.

O aumento do tempo de computação deve-se a necessidade de processar os dois limites do intervalo. O uso de uma arquitetura paralela apresenta-se como uma solução para esse problema.

2. *IMPLEMENTAÇÃO DAS OPERAÇÕES BÁSICAS*

A forma geral de um número em ponto flutuante, usando-se a representação sinal-magnitude, é dada por:

$$x = * \; m \; b^{e}$$

onde:

$* \in \{+,-\}$ representa o sinal;

m - um número fracionário chamado mantissa;

e - um número inteiro chamado expoente;

b - um número inteiro positivo chamado de base.

A representação única dos números em ponto flutuante é garantida pela normalização da mantissa. Evita-se a representação do sinal do expoente usando-se expoente polarizado.

O conjunto de todos os números em ponto flutuante representáveis em um sistema S depende de ℓ, e_1, e_2, e b; onde ℓ é o número de bits da mantissa, e_1 e e_2 representam o menor e maior expoente respectivamente e b é a base adotada.

Definição 1 - Seja S um sistema de ponto flutuante e $*$ uma das operações em R com $* \in \{+,-,x,/\}$. A correspondente operação em ponto flutuante denotada por $\boxed{*}$ é dada por:

(PG) $$x \boxed{*} y = \square(x * y) \quad \text{para todo} \quad y,x \in S$$

e para todo $* \in \{+,-,x,/\}$

O mapeamento $\square R \to S$ é denominado de arredondamento.

Diferentes tipos de arredondamento são usados pelos computadores. Em particular, a Aritmética de Intervalo opera com arredondamento direcionado que pode ser feito para cima ou para baixo. O primeiro tipo se aplica ao limite superior do intervalo e o segundo tipo, ao limite inferior.

Usando-se esse arredondamento, os tratamentos do overflow e e do underflow são vistos como casos da operação de arredondamento. Por exemplo, se o limite superior do intervalo for um número positivo e ocorrer um overflow, o mesmo será aproximado para $+\infty$; se o overflow ocorrer no limite inferior, a aproximação será para o maior número de S.

As operações básicas são então, feitas sobre S* que é uma

extensão de S de modo a incluir $+\infty$, $-\infty$ e zero Isto significa que para a implementaçao do arredondamento direcionado, a máquina deve ser capaz de reconhecer e operar os padrões especiais de bits que representam mais infinito e menos infinito.

Definição 2 - Arredondamento direcionado monotônico "para baixo":

$\nabla x := \max \{y \in S^* | y \leq x\}$ para todo $x \in R$.

Definição 3 - Arredondamento direcionado monotônico "para cima":

$\Delta x := \min \{y \in S^* | y \geq x\}$ para todo $x \in R$

O arredondamento direcionado com exatidão máxima satisfaz as seguintes propriedades:

P1 $\square x = x$ para todo $x \in S$

P2 $x \leq y$ implica que $\square x \leq \square y$

P3 $\nabla x \leq x$ e $x \leq \Delta x$ para todo $x \in R$

A exatidão das operações básicas é máxima porque entre o resultado correto e o resultado aproximado não existe nenhum número pertencente a S^*.

De acordo com PG, tem-se que $x \boxed{*} y = \square(x * y)$; em [KUL 81] são representados teoremas que provam a possibilidade de se trabalhar com o resultado aproximado $x \tilde{*} y$ de modo a assegurar a seguinte igualdade

$$\square(x \tilde{*} y) = \square(x * y).$$

Em termos de hardware, isso significa que a unidade aritmética deve operar internamente com um número de bit maior que o usado para representar o dado de modo a assegurar que o valor arredondado seja igual àquele que seria obtido aplicando-se a função de

arredondamento ao resultado de uma operação exata.

O padrão IEEE para sistemas de ponto flutuante usa 3 bits suplementares para fazer o arredondamento. O terceiro bit, chamado de "stick" é proprio do arredondamento direcionado [STE 81].

Nas operações de adição e subtração, os expoentes devem estar alinhados e para isso pode haver necessidade de se realizar uma operação de deslocamento relativo entre mantissa. Usando-se um acumulador com menos de $(e_2-e_1+\ell)$ bits, o deslocamento pode implicar em perdas de bits significativas da mantissa do menor operando.

A função do bit de "stick" é a de registrar a ocorrência de pelo menos um bit diferente de zero entre aqueles perdidos. Se a operação for exata, esse bit será zero e não haverá necessidade da operação de arredondamento alterar o valor do acumulador. Desse modo a propriedade P1 é assegurada.

3. IMPLEMENTAÇÃO DO PRODUTO ESCALAR

O fato das operações básicas serem implementadas com exatidão máxima não assegura o melhor resultado na avaliação de expressão algébrica. O erro registrado em cada operação pode se propagar através dos cálculos e se acumular além dos limites aceitáveis.

Por esse motivo, o padrão IEEE para números em ponto flutuante estabelece um formato interno chamado real temporário com um número de bit que representa a mantissa maior que aqueles usados para representar o dado (formato precisão simples e dupla). Em [HOU 81], é dado um exemplo onde essa solução não é satisfatória.

O produto escalar pode ser visto como um caso especial de avaliação de expressão. O destaque decorre de sua importância no

cálculo matricial. Para implementá-lo com exatidão máxima é necessário que se tenha os termos de produto com 2ℓ bits e que o somatório desses termos realizados de forma a assegurar que o erro esteja no último bit da mantissa.

O algorítmo proposto em [BOH 83] envolve os seguintes passos:

i) Calcular o somatório S dos termos de produto; As operações de soma parcial são feitas em um acumulador de $(4\ell+1)$ bit e o resultado arredondado para 2ℓ bits. O resíduo associado a cada operação de arredondamento de um resultado parcial é um número em ponto flutuante com representação em S* (desde que não se faça a exigência da normalização); e como tal será armazenado para os cálculos posteriores.

ii) Calcular o somatório dos resíduos. Essa operação pode gerar novos resíduos.

iii) Testar se o somatório dos resíduos Sr altera o somatório S; ou seja se

$$S \neq S \oplus Sr$$

onde $\oplus$ denota a operação de soma seguida do arredondamento. Em caso afirmativo, é feita a operação $S := S \oplus Sr$ e volta-se ao passo (ii). O resultado negativo do teste fornece a condição de parada para o algorítmo.

As propriedades importantes desse algorítmo são: exatidão máxima e a não dependência do resultado em relação a ordem dos operandos. Ou seja, o resultado é único e o melhor possível.

Em [SIL 87] foi proposto uma modificação desse algorítmo que permite reduzir o acumulador mantendo-se as mesmas precisão e exa-

tidão e sem implicar em um aumento do tempo de computação

A idéia é que cada operação de soma seja precedida pela retirada do resíduo. Para implementá-lo, a unidade aritmética deve poder fazer uma máscara sobre os números em ponto flutuante.

4. IMPLEMENTAÇÃO DAS OPERAÇÕES BÁSICAS NO ESPAÇO DOS INTERVALOS

As operações aritméticas no espaço dos intervalos podem ser definidas como:

$$A*B = [\min(\underline{A}*\underline{B}, \underline{A}*\overline{B}, \overline{A}*\underline{B}, \overline{A}*\overline{B}),\ \max(\underline{A}*\underline{B}, \underline{A}*\overline{B}, \overline{A}*\underline{B}, \overline{A}*\overline{B})]$$

onde $* \in \{+, -, \times, /\}$

$\underline{A}, \underline{B}, \overline{A}$ e $\overline{B} \in S^*$

O uso do arredondamento direcionado garante que o resultado exato de cada operação estará contido pelo intervalo que lhe representa. A implementação das operações básicas em S^* com exatidão máxima, garante que o intervalo resultante será o menor possível.

O tempo de computação das operações aritméticas no espaço dos intervalos pode ser reduzido explorando-se a estrutura de vetor apresentado pelo intervalo. Além disso, é interessante que sejam implementadas algumas operações auxiliares muito usadas nos métodos intervalares; quais sejam: Teste de inclusão de um real em um intervalo, teste de inclusão de um intervalo noutro intervalo, valor absoluto, união e intersecção de intervalos.

5. USO DE TÉCNICAS RESIDUAIS

Trabalhando-se com matrizes de intervalos, o controle do erro torna-se bastante difícil e até mesmo inviável em alguns ca-

sos de resolução de sistemas lineares, inversão de matrizes, avaliação de polinômios etc... Usando-se a aritmética computacional avançada é possível, mesmo nesses casos, a validação automática do resultado; isto é, a verificação da existência e unidade da solução obtida pelo computador. Em problemas extremamente mal-condicionados, o usuário poderia ser notificado.

A idéia básica do método de validação é que a partir de uma aproximação inicial do resultado, obtido por um método convencional qualquer, pode-se gerar um processo iterativo de refinamento da solução tendo como fator de realimentação o erro ou resíduo cometido no passo anterior.

6. CONCLUSÃO

Em relação à Unidade Aritmética, foram examinadas duas soluções: uso de um co-processador baseado no padrão IEEE para números em ponto flutuante e o projeto de uma unidade aritmética específica.

A primeira solução permitiria a implementação das operações básicas com exatidão ótima e simplificaria consideràvelmente o circuito. A dificuldade quanto ao seu uso, estaria relacionada a precisão da operação de produto escalar.

O maior formato para representar um número em ponto flutuante, nesse padrão, tem uma mantissa com 53 bit. O algorítmo proposto por [BOL 83] precisa de $(4\ell+1)$bit, então a precisão de cada dado cairia para 13 bit o que certamente seria insuficiente. A variante do algorítmo proposto em [SIL 87], trabalhando com apenas $(2\ell+1)$ bit, viabilizaria o uso de dados com precisão simples (ℓ=23 bit). A dificuldade, nesse caso, está na realização das más-

caras para se extrair os resíduos. O co-processador 8087 por exemplo, não possui instrução que realize operações lógicas sobre números em ponto flutuante.

No que diz respeito ao paralelismo, a estrutura de vetor do intervalo e a conveniência de se implementar uma biblioteca com as operações básicas nos espaços dos vetores e das matrizes, induz a uma arquitetura SIMD. Com isso, a eficiência dos algorítmos relacionados com as técnicas residuais, aumentaria muito.

O detalhamento dessa arquitetura passaria pela definição do número de unidades aritméticas e pela obtenção da versão paralela para os algorítmos envolvidos na especificação da linguagem.

REFERÊNCIAS BIBLIOGRÁFICAS

[BOL 83] Bohlender,G., Grüner,K., "Realization of an optimal computer arithmetic - A new approach to scientific computation" Academic Press, 1983.

[HOL 81] Hough,D., "Applications of the proposed IEEE 754 standard for floating point arithmetic" - Computer march, 1981.

[KUL 81] Kulisch,V., Miraulser,W.L., "Computer arithmetic in theory and pratice" Academic Press, 1981.

[PAL 84] Palmer,J.F., Morse,S.P., "The 8087 primer" Willey Press, 1984.

[SIL 87] Silva,E.N., "Especificação da aritmética de intervalo" - Tese de mestrado - Depto de Informática - UFPE (a ser publicada).

[STE 81] Stevenson,D., "A proposed standard for binary floating point arithmetic" Computer - march 1981.