Opérateurs arithmétiques matériels
Additions et décalages pour les fonctions complexes

Florent de Dinechin

5 janvier 2005

Introduction

- On a déjà vu la division et la racine carrée
- E-méthode pour les polynomes et les fractions rationnelles
- Cordic et ses enfants pour certaines fonctions élémentaires

La E-Méthode

Cékoi

- proposée par M. Ercegovac dans les années 70
- résoudre certains systèmes linéaires, à diagonale dominante,
- à l'aide d'une itération simple et régulière à base d'additions et de décalages
- version vectorielle de la division SRT

$$\begin{pmatrix} 1 & -x & 0 & & \cdots & & & 0 \\ q_1 & 1 & -x & 0 & & \cdots & & 0 \\ q_2 & 0 & 1 & -x & 0 & & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & 0 \\ & & & & & & 1 & -x \\ q_n & & & \cdots & & & 0 & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ \vdots \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix}$$

Désormais on notera

$$\mathcal{A}y = b$$

La solution de $\mathcal{A}y = b$ est $y = [y_0, y_1, \ldots, y_n]^t$ telle que

$$y_0 = R(x) = \frac{P(x)}{Q(x)} = \frac{p_n x^n + p_{n-1} x^{n-1} + \cdots + p_0}{q_n x^n + q_{n-1} x^{n-1} + \cdots + 1}.$$

Condition : faut que $q_0 = 1$ (sinon scaling).
Remarque: les polynômes sont un cas particulier.

Et pourquoi on fait cela ?

Un genre d'algo de division SRT, mais avec un vecteur de restes partiels (on divise par la matrice):

initialisation :

$w[0] \longleftarrow b$

$d[0] \longleftarrow 0$

itération sur les chiffres :

pour j de 1 à m faire

$w[j] \longleftarrow \beta \times \left(w[j-1] - \mathcal{A} \times d[j-1]\right)$

$d[j] \longleftarrow S(w[j])$

résultat :

$y_0[m] = \sum_{i=1}^{m} d_0[i]\beta^{-i}$

$d[j]$ et $w[j]$ sont des vecteurs.

Allez lire la thèse d'Ercegovac

- Plein de conditions pour que cela converge.
- Décidément cela ressemble à la division SRT
- Typiquement il faut que $|x| < 1/8$

Ercegovac l'a fait en base 2 (chiffres signés). Arnaud et cie sont en train de le généraliser en base quelconque:

Romain Michard, Arnaud Tisserand and Nicolas Veyrat-Charvillon,
Evaluation de polynômes et de fractions rationnelles sur FPGA avec des opérateurs à additions et décalages en grande base.
RR LIP 2004-62.

Plus en détail

- Calcul effectué par ligne:

$$w_i[j] = \beta \times \left(w_i[j-1] - d_0[j-1]q_i - d_i[j-1] + d_{i+1}[j-1]x\right)$$

- évidemment la multiplication par β est un décalage
- Dans les cas $i = 0$ et $i = n$, le calcul se simplifie en

$$w_0[j] = \beta \times \left(w_0[j-1] - d_0[j-1] + d_1[j-1]x\right)$$

et

$$w_n[j] = \beta \times \left(w_n[j-1] - d_0[j-1]q_n - d_n[j-1]\right)$$

- la sélection de chaque chiffre du vecteur d se fait aussi ligne par ligne
- pour implémenter, i sera un indice d'espace et j un indice de temps.

Architecture

- Une unité de calcul par ligne
- Une unité:
  - 2 ou 3 additions/soustractions,
  - des multiplications d'un chiffre par un nombre
  - et la fonction de sélection S
- Entrées/sorties de l'unité : des chiffres seulement (peu de communications)
- Comme d'hab, on s'intéresse à
  - $\beta = 2$, chiffres $\{-1, 0, 1\}$
  - $\beta = 4$, chiffres $\{-3...3\}$ avec $3x = 2x + x$

Fonction de sélection

Comme pour la division, elle donne les chiffres en fonction des restes partiels :

- $d[j] \longleftarrow S(w[j])$
- Elle se calcule composant par composant
- Fonction idéale:

$$S(x) = \begin{cases} \text{signe}(x) \times \lfloor |x + 1/2| \rfloor, & \text{si } |x| \leq 1 \\ \text{signe}(x) \times \lfloor |x| \rfloor, & \textit{sinon} \end{cases}$$

- En principe il faut tout $w[j-1]$
- Pour aller plus vite, on peut utiliser une approximation

À quoi Arnaud utilise ses esclaves

Fonction de sélection pour la base 2:

	$\beta = 2, \mathcal{E}_{2,1}$	
$\hat{w}[j-1]$	$d[j]$	$w[j] \in$
0000	0	$[0, 1/2[$
0001	1	$[1/2, 1[$
0010	1	$[1, 3/2[$
0011	1	$[3/2, 2[$
0100	n/a	impossible
⋮		
1011	n/a	impossible
1100	-1	$[-2, -3/2[$
1101	-1	$[-3/2, -1[$
1110	-1	$[-1, -1/2[$
1111	0	$[-1/2, 0[$

À quoi Arnaud utilise ses esclaves

Fonction de sélection pour la base 4:

	$\beta = 4, \mathcal{E}_{4,2}$		$\beta = 4, \mathcal{E}_{4,3}$	
$\hat{w}[j-1]$	$d[j]$	$w[j] \in$	$d[j]$	$w[j] \in$
00000	0	$[0, 1/2[$	0	$[0, 1/2[$
00001	1	$[1/2, 1[$	1	$[1/2, 1[$
00010	1	$[1, 3/2[$	1	$[1, 3/2[$
00011	2	$[3/2, 2[$	2	$[3/2, 2[$
00100	2	$[2, 5/2[$	2	$[2, 5/2[$
00101	2	$[5/2, 3[$	3	$[5/2, 3[$
00110	n/a	impossible	3	$[3, 7/2[$
00111	n/a	impossible	3	$[7/2, 4[$
01000	n/a	impossible	n/a	impossible
⋮				
10111	n/a	impossible	n/a	impossible
11000	n/a	impossible	-3	$[-4, -7/2[$
11001	n/a	impossible	-3	$[-7/2, -3[$
11010	-2	$[-3, -5/2[$	-3	$[-3, -5/2[$

Et alors, cela sert à quelquechose ?

Ils viennent juste de s'y mettre, ils n'ont pas fait des comparaisons sérieusement. En gros :

- C'est plus rapide et plus petit que du Horner bête ou du Horner/SRT
- Plus petit que les méthodes à base de table de Jérémie à partir de 20 bits.
- Toujours au moins 10 fois plus lent (latence totale pour avoir tous les chiffres).
- Donc utile pour de grandes précisions (24, 32, 64)
- Le gros concurrent, cela sera Cordic.

Cordic et ses avatars