割り算回路設計，あの手この手

連載 3

CSAを用いた除算回路の高速化

鈴木昌治

デバイスの記事

関連データ

非回復法による除算回路の高速化を図るため，ここではCSA（carry save adder）を用いた手法を紹介する．この手法は乗算回路の高速化にも利用されている．除算回路に適用するには部分商の判定についてくふうする必要がある．なお，本稿で設計した回路のソース・コードは本誌ホームページ（http://www.cqpub.co.jp/dwm/）からダウンロードできる．

（編集部）

連載第1回（本誌2005年8月号，pp.109-117）では回復法を用いて除算の基本を押さえ，第2回（同年10月号，pp.136-144）では非回復法で符号付きの引き数の取り扱いへ発展させて，高速化へのアプローチの土台作りを行いました．今回は，ここまで説明してきた構造をいかに高速化するかについて検討してみましょう．

1 入出力特性――前回までのおさらい

図1に回復法と非回復法における，1サイクル当たりの入出力特性を示します．今回はこの図が重要な役割を担うので，最初に図の示す意味をしっかり理解しておきましょう．同時に，回復法と非回復法のおさらいもしておきます．

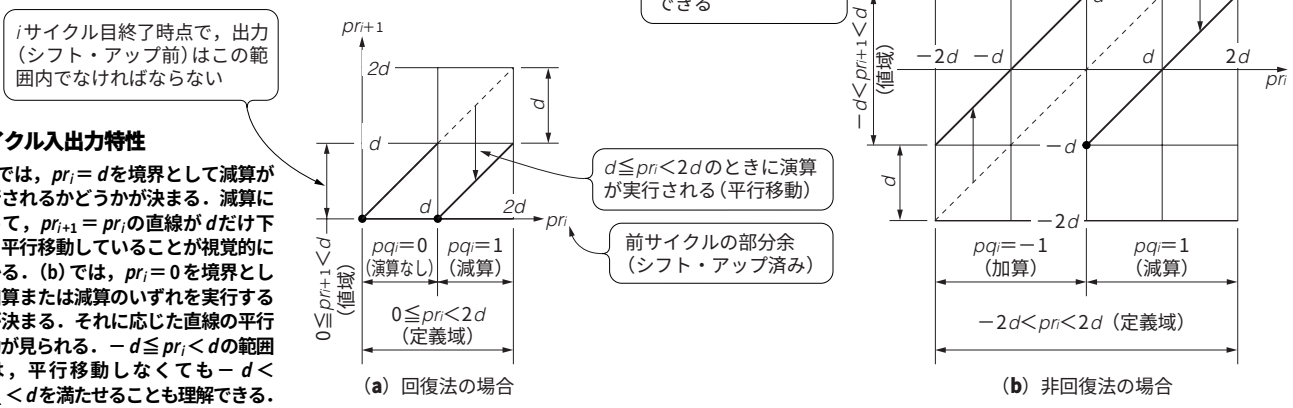

図1
サイクル入出力特性
（a）では，$pr_i = d$を境界として減算が実行されるかどうかが決まる．減算によって，$pr_{i+1} = pr_i$の直線がdだけ下方に平行移動していることが視覚的にわかる．（b）では，$pr_i = 0$を境界として加算または減算のいずれを実行するかが決まる．それに応じた直線の平行移動が見られる．$-d \leqq pr_i < d$の範囲では，平行移動しなくても$-d < pr_{i+1} < d$を満たせることも理解できる．

● 演算の条件を評価する際に利用しよう

まずは回復法から説明します．iサイクル目を開始するときの入力をpr_iとします．$i-1$サイクル目の部分余を1ビットだけシフト・アップしたものになります．除数をdとすると，回復法における除算成立条件は次式で表せます．

$$0 \leqq pr_i < 2d \quad \cdots\cdots(1)$$

式（1）を前提とし，部分商をpq_iとすると，iサイクル目では次のような操作を行います．

$$\left.\begin{aligned} pq_i &= 1 \\ pr_{i+1} &= pr_i - d \end{aligned}\right\} (pr_i \geqq d \text{の場合}) \quad \cdots\cdots(2\text{-}1)$$

$$\left.\begin{aligned} pq_i &= 0 \\ pr_{i+1} &= pr_i \end{aligned}\right\} (pr_i < d \text{の場合}) \quad \cdots\cdots(2\text{-}2)$$

これに基づいて，**図1(a)**を見てみましょう．図において点線で示した部分は，出力＝入力とした(減算が実行されなかった)場合の特性，実線で示した部分は実際の入出力特性となります．

横軸は入力のpr_iであり，シフト・アップ済みなので式(1)の条件を満たしています．これが定義域となります．$pr_i \geqq d$の場合のみ除数の減算が実行され，部分商を1とします．このため，出力pq_{i+1}(縦軸)はdだけ下方向に平行移動した値をとります．この操作により，値域が$0 \leqq pr_{i+1} < d$の範囲に抑えられるのがよくわかると思います．

図1(b)に，非回復法の場合を示します．グラフのつごう上，$d > 0$としてあります．まず，除算成立条件を式(3)に，iサイクル目で行う操作を式(4-1)と式(4-2)に示します[注1]．

$$-2d < pr_i < 2d \quad \cdots\cdots(3)$$

$$\left.\begin{aligned} pq_i &= 1 \\ pr_{i+1} &= pr_i - d \end{aligned}\right\} (pr_i \geqq 0 \text{の場合}) \quad \cdots\cdots(4\text{-}1)$$

$$\left.\begin{aligned} pq_i &= -1 \\ pr_{i+1} &= pr_i + d \end{aligned}\right\} (pr_i < 0 \text{の場合}) \quad \cdots\cdots(4\text{-}2)$$

減算/加算が施された領域(実線)は，点線(演算が行われなかった部分)が$\pm d$だけ平行移動しているのがわかります．また，実線の端点における黒丸はその点が実在することを示しており，非回復法で$pr_i = 0$の場合は$pr_{i+1} = -d$となって，除算の成立条件である式(3)が満たされていないことがすぐにわかります．

このように，入出力の関係がひと目でわかるので，場合分けの条件や除算の成立条件などを評価するには，非常に便利な図です．

また，**図1(b)**から非回復法において注目すべき点がわかります．入力が$-2d \leqq pr_i < 2d$の場合，出力は$-d \leqq pr_{i+1} < d$の範囲で一意に決まればよいので，入力が$-d \leqq pr_i < d$の範囲にある場合は減算や加算は必要ないということです．言い換えれば，入力が$-d \leqq pr_i < d$の範囲にある場合は平行移動せず，点線の位置のままでも非回復法の条件は成り立つということです．

表1　SRT法の処理条件

pr_iの範囲	部分商	演　算
$pr_i < -1/2$	-1	$pr_{i+1} = pr_i + d$
$-1/2 \leqq pr_i < 1/2$	0	$pr_{i+1} = pr_i$
$1/2 \leqq pr_i$	1	$pr_{i+1} = pr_i - d$

注1：グラフのつごう上$d > 0$としたので，式(4-1)と式(4-2)では判断基準を$pr_i \geqq 0$としてあるが，dを符号付きで扱う場合はpr_iとdの符号が等しいか否かが判断基準となる．

2 SRT法に見る回路高速化のヒント

次に，この発想を利用したおもしろい除算法を紹介しましょう．非回復法の変型ともいえる「SRT法」です．個人的には，ここで説明するSRT法は「実用性の面ではいまひとつ」と考えていますが，その中にある発想が後述の高速化の手法につながるので，ここで紹介しておきます．

● 加減算や比較演算を省略できるメリットはあるが…

まず，除数を$1 > d \geqq 1/2$に正規化し，$-1/2 \leqq pr_i < 1/2$になるように調整しますが，これらは符号反転とシフトで行えます．そして，**表1**の条件で部分商の選択と加減算を行うこととします．

この条件下で，先ほどの入出力特性をプロットしてみましょう．これを**図2**に示します．非回復法の場合と似た特性ですが，

- 加減算を行わない領域がある
- その境界が$\pm 1/2$という固定値

という点が異なります．また，入力が$-1 \leqq pr_i < 1$，出力が$-1/2 \leqq pr_i + 1 < 1/2$で定義されることも，非回復法とは違います．

ここからわかることは，$-1/2 \leqq pr_i < 1/2$である限り$-1/2 \leqq pr_{i+1} < 1/2$も成立するので，最初のサイクルで条件が成立すれば，最後まで破たんなく除算が成立するということです．このアルゴリズムが持つメリットは，次の二

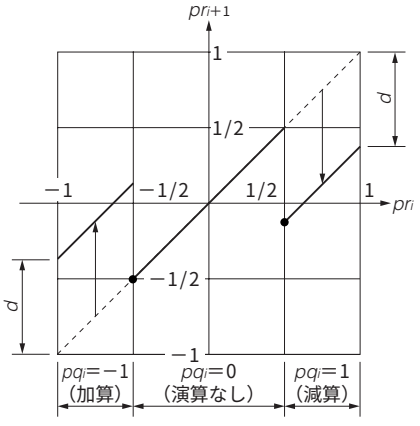

図2
SRT法におけるサイクル入出力特性
正規化された除数dに対し，$\pm 1/2$という固定値を境界として加算/減算を使い分ける方式でも，演算しない領域を持つことで除算の成立条件を満たすことができる．