

符号確定位置を考慮した非同期式非回復法除算器の設計

3E-6

中野栄治

今井雅

中村宏

南谷 崇

東京大学 先端科学技術研究センター

1 はじめに

同期式の加減算繰返し型の除算器では、部分剰余の符号検出の最悪遅延が大きい。部分剰余と除数の上位数ビットのみを用いて商を求める SRT 法が広く用いられている [1]。これによって高速に除算を行うことができるが、商デジットの選択や通常表現への変換等のための複雑な回路が必要になる。

一方、非同期式回路では回路の性能を決定するのは平均遅延である。そこで本稿では部分剰余に SD 表現を採用し、符号検出に要する平均遅延を小さくすることで、より簡単な非回復法を用いて高速な除算器が構成できることを示す。

2 SD 表現について

SD 表現された数 $R = [r_n \cdots r_0]$ ($r_i \in \{\bar{1}, 0, 1\}$) は、 $\sum_{i=0}^{n-1} r_i \cdot 2^i$ という値をもつ。データにタイミング情報を付与しなければならない非同期式では、各ビット r_i の値を表すのに三本の信号線 (rp_i, rz_i, rm_i) を使い、それぞれ (1, 0, 0), (0, 1, 0), (0, 0, 1) のとき、値 1, 0, $\bar{1}$ をとると解釈する。

通常表現の符号なし数 $X = [x_n \cdots x_0]$ ($x_i \in \{0, 1\}$) と SD 数との加算では、キャリーが 1 ビットしか伝播しないことが知られている [5]。また、SD 表現された数の符号を反転するには、各ビットの正の線 rp_i と負の線 rm_i を入れ替え、(rm_i, rz_i, rp_i) とすればよい。従って、減算も $R - X = -(-R + X)$ とすることにより同一の加算器で行える。演算中一定の値をもつ除数を X 、部分剰余を R とすることにより、除算における加減算に SD 加算を用いることができる。

SD 表現された数の符号検出用セルとして、図 1 を設計した。このセルは (hip, hiz, him) (sp, sz, sm) の 2 つの one-out-of-3 信号を入力とし、(hop, hoz, hom) という one-out-of-3 信号を出力する。(hop, hoz, hom) は (hip, hiz, him) = (0, 1, 0) の場合のみ (sp, sz, sm)、それ以外は (hip, hiz, him) を出力する。

Design of an asynchronous nonrestoring divider with high-speed sign detector

Eiji Nakano, Masashi Imai, Hiroshi Nakamura, Takashi Nanya

Reserch Center for Advanced Science and Technology, University of Tokyo

$$(hop, hoz, hom) = \begin{cases} (hip, hiz, him) & (hip, hiz, him) \neq (0, 1, 0) \\ (sp, sz, sm) & (hip, hiz, him) = (0, 1, 0) \end{cases}$$

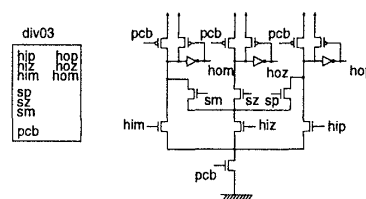


図 1: 符号検出用 DCVSL セル

3 除算器の高速化

3.1 符号検出方法の改良

部分剰余に SD 表現を用いて非回復法除算器を構成し、除数・被除数にランダムな 32 ビット正数を 1000 組入力した。そのときの部分剰余について、ゼロが上位何ビット連続するかを調べたのが図 2 である。これにより、90% 以上の場合において、上位 5 ビット以内に部分剰余の符号が確定していることが分かる。

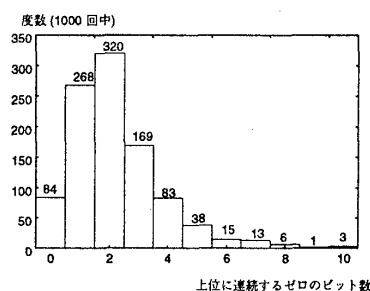


図 2: 符号確定位置

そこで、上位ビットで符号が決まる場合により高速に出力が決定するような符号検出方法を考えた。この方法を 10 ビット数 $r[9:0]$ を例にして図 3 に示す。

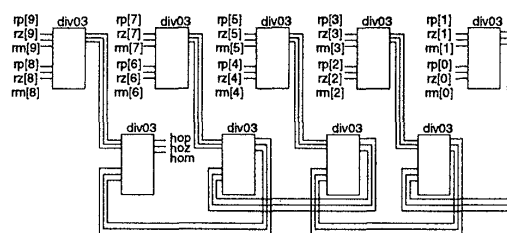


図 3: 符号検出方法

3.2 加減算と符号検出の並列化

SD 表現では、加減算と符号検出は全く独立な操作である。そこで、図 4 のように各ステージに加算器を 2 つ用いて加算・減算両方の結果を予め求めておき、符号検出が完了した時点で所望のものを選択することで高速化出来る。

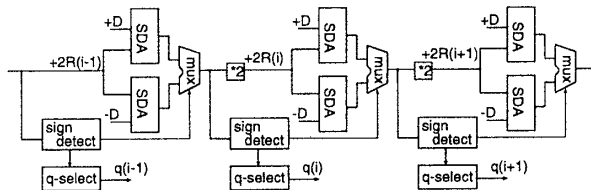


図 4: 符号検出と加減算の並列化

4 シミュレーション結果

まず、2 つの正規化された 32 ビット数 X, Y が入力され、32 ビットの商 $Q = [q_{31} \cdots q_0]$ を出力する組み合わせ回路による除算器を主に DCVSL で構成した。0.25 μm CMOS プロセスを想定しており、各セルはインバータ 3 つに相当する容量と 0.24 mm の配線容量を駆動することを仮定している。

SD 表現のものは加減算器を並列にしないもの (SD1) とするもの (SD2)、通常表現のものは加減算器に順次桁上げ方式を用いたもの (RCA) とキャリールックアヘッドを用いたもの (CLA) によって、組み合わせ回路による非回復法除算器を構成し比較した (表 1)。入力値にはランダムな 32 ビット正数を用い、1000 回シミュレーションを行いその平均遅延を求めた。

この表で「平均遅延」とは商 32 ビットを求めるのに要した平均時間である。「各段の遅延」はそれを 32 で割ったものであり、「回路量」は 1 段分のトランジスタ数である。表から SD1 が RCA は勿論 CLA よりも高速であることが分かる。SD2 は SD1 に比べて回路量は 83% 増えるが、40% 高速である。

表 1: 非回復法除算器の遅延

	SD 1	SD 2	RCA	CLA
平均遅延 (ns)	55.3	39.4	60.7	56.4
各段の遅延 (ns)	1.73	1.23	1.90	1.76
回路量	2742	5018	2112	3600

さらに基数 2 の SRT 除算との比較を行った結果を表 2 に示す。「SRT1-1」「SRT1-2」は文献 [4] にあるような、部分剰余を SD 数で表した場合であり、「SRT1-1」は加算器 1 つ、「SRT1-2」は加算器を 2 つ並列に用いたものである。また、「SRT2」は文献 [2][3] にあるような通常の桁上げ保存加算器を用いた場合である。

これにより、非同期式 SRT 除算器を設計する場合にも、部分剰余に SD 表現を用い、加算器を 2 つ配置することで高速になることが分かる。桁上げ保存型は、性能面・コスト面で共に SD 表現を用いた二つの場合の中間にあたる。

ここで、提案した非回復法除算器は SRT 法に迫るコストパフォーマンスを達成していることに注目すべきである。実用化の際に、SRT 法では商の変換のための回路量や遅延がわずかに増加し、最終段では剰余の正確な符号が必要となることを考えると、非同期式回路においては非回復法も有効な手段であるといえる。

表 2: SRT 法除算器の遅延

	SRT1-1	SRT1-2	SRT2
平均遅延 (ns)	55.1	36.5	51.1
各段の遅延 (ns)	1.72	1.14	1.60
回路量	3327	4944	3589

5 まとめ

部分剰余に SD 表現を用いると、符号が決定されるビット位置の度数分布が上位に片寄っていた。これを利用して平均的に高速に符号を検出する手法を提案した。これにより、性能が平均遅延で決定される非同期式の除算器では、基数 2 の SRT 法に迫る高速な除算器が非回復法によって構成できることを示した。

なお、本研究の一部は科学研究費補助金 (基盤研究 (B)09480049)、及び (株) 半導体理工学研究センターとの共同研究によるものである。

参考文献

- [1] S. F. Oberman and J. Flynn : "Division Algorithms and Implementations," *IEEE Trans. Computers*, Vol.46, No.8, pp.833-854, Aug. 1997.
- [2] T. E. Williams and M. A. Horowitz : "A Zero-Overhead Self-Timed 160-ns 54-b CMOS Divider," *IEEE Journal of Solid State Circuits*, pp.1651-1661, Nov. 1991.
- [3] 松原, 井出, 鈴木 : "非同期回路を用いた高速除算/平方根演算器の設計," 電子情報通信学会論文誌, pp.197-206, March 1997.
- [4] 高木, 安浦, 矢島 : "冗長 2 進表現を利用した VLSI 向き高速除算器," 電子通信学会論文誌, Vol.J 67-D, No.4, pp.450-457, April 1984.
- [5] Kai Hwang : "Computer Arithmetic," New York, Wiley, 1979.