

3ステップ加減算手法を用いた高速冗長2進除算器の構成

Structure of High-Speed Redundant Binary Divider Using Three Steps Method for Addition/Subtraction.

工藤忠道*, 恒川佳隆*

Tadamichi Kudou*, Yoshitaka Tsunekawa*

*岩手大学大学院 工学研究科

*Faculty of Engineering, Iwate University

キーワード : 冗長2進表現(Redundant Binary Representation), 高速 (High-Speed), 加減算器 (Adder/Subtractor), 除算器(Divider), VLSI評価 (VLSI evaluation)

連絡先 : 〒020-8551 盛岡市上田4-3-5 岩手大学 工学部

工藤忠道, Tel.: (019)621-6468, Fax.: (019)621-6468, E-mail: t2303008@iwate-u.ac.jp

1. まえがき

算術演算の中で加算・減算は最も基本となるものであり, 乗算や除算をはじめあらゆる算術演算がこれらの繰り返しで行われる。故にデジタルシステムの性能は加算器・減算器の処理能力に依存し, 従来からこれらの高速化が望まれている。

また, 現在は高速性だけでなく, 幾何計算や統計処理及びネットワーク通信における情報セキュリティの分野などにおいては高精度演算の要求も高まっている。例えば情報セキュリティにおける暗号化問題では, 安全基準の問題から2048桁以上の高精度演算が要求されている。一般的な加算の高速化手法として桁上げ先見があるが, この手法を用いても演算時間は $O(\log N)$ (N は演算桁数)であり, 高精度では演算時間の増加が問題となる。

高速性と高精度性が同時に要求される問題に対応する数体系にSD(Signed Digit)表現がある¹⁾。SD数系では桁上げが1桁のみで各桁並列に加減算を

行え, 演算時間は桁数に依存しない。本報告では, このSD表現の一種である冗長2進数を用いた除算の検討を行う。これまでに, 内部演算に冗長2進数を用いた除算器が提案されている²⁾。しかし, 各演算の入出力をSD数としたシステムでの適用を目的とした除算器についてはまだ報告されていない。

本報告では, このSD数のシステムへの適用を考慮した, 入出力に冗長2進数を用いた除算器を提案する。まず除算器の構成法として, 高速性の観点から高木らが提案した高速除算用アルゴリズムに基づいて考察する²⁾。次に, 我々がこれまでに提案してきた1桁2ビット/3ビット混合表現を用いた, 符号変換器を必要としない高速加減算器の除算器への適用を試みる³⁾。この場合, 各行での演算を決定する制御信号の生成時間が影響し, 本加減算器の高速性を活かさないという問題が生じる。

そこでこの加減算器を改良し, 3つのステップで演算を行う新たな高速冗長2進加減算器の構成法を

提案する．これによりハードウェア量の増加を抑えながら，より高速な加減算器が実現できる．さらに，商を求める際に部分剰余の冗長性から生じる問題点を示し，これに対処するための除算器の構成法を提案する．最後に，本除算器の構成法に対してVLSI設計システムPARTHENONにより設計及び評価を行う⁴⁾．その結果，符号変換器を使う従来の加減算器を適用した除算器に対し，2.2倍以上の高速化が可能となることを明らかにする．

2. 高速除算アルゴリズム

本章では，除算器に適用するアルゴリズムを述べる．これまでに，冗長2進表現の特長を活かした高速除算アルゴリズムが提案されている²⁾．まず，このアルゴリズムに用いる変数を定義する．

- X : 入力の演算数 (被除数, $\frac{1}{2} \leq X < 1$)
- Y : 入力の演算数 (除数, $\frac{1}{2} \leq Y < 1$)
- Q : 出力の商 ($\frac{1}{2} < Q < 2$)
- q_i : 商の小数点 i 桁目
- $R^{(i)}$: i 回目に得られる部分剰余
- r_j : $R^{(i)}$ の j 桁目
- N : 桁数

除算アルゴリズムを以下に示す．

```

begin
  <ステップ1>
   $q_0 := 1, R^{(1)} := X - Y$ 
  <ステップ2>
  for  $i := 1$  to  $N$  do
    begin
       $q_i := \begin{cases} \bar{1} & \text{if } [r_0.r_1r_2]_{SD2} < 0 \\ 0 & \text{if } [r_0.r_1r_2]_{SD2} = 0 \\ 1 & \text{if } [r_0.r_1r_2]_{SD2} > 0 \end{cases}$ 
       $R^{(i+1)} := 2R^{(i)} - q_iY$ 
    end
  <ステップ3>
   $Q := [q_0.q_1q_2 \cdots q_N]$ 
end

```

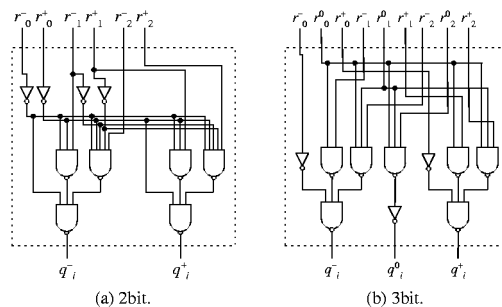


Fig. 1 Control signal generator.

このアルゴリズムをみてわかるように，部分剰余の上位3桁を参照するだけで商の各桁を決定できる．各行の演算を決定する制御信号である商の各桁を生成する回路の構成においてまず，従来の1桁2ビット表現 ($r = \{r^-r^+\}$) を用いた場合について示す．論理式は以下ようになる．

$$q_i^- = \overline{\overline{r_0^- \cdot r_0^- \cdot r_0^+ \cdot r_1^- \cdot r_0^- \cdot r_0^+ \cdot r_1^- \cdot r_1^+ \cdot r_2^-}} \quad (1)$$

$$q_i^+ = \overline{\overline{r_0^+ \cdot r_0^- \cdot r_0^+ \cdot r_1^+ \cdot r_0^- \cdot r_0^+ \cdot r_1^- \cdot r_1^+ \cdot r_2^+}} \quad (2)$$

上記の論理式に基づいた構成を図1(a)に示す．

一方，我々が提案した1桁3ビット表現 ($r = \{r^-r^0r^+\}$) を用いた場合の論理式は，以下ようになる．

$$q_i^- = \overline{\overline{r_0^- \cdot r_0^0 \cdot r_1^- \cdot r_0^0 \cdot r_1^0 \cdot r_2^-}} \quad (3)$$

$$q_i^0 = \overline{\overline{r_0^0 \cdot r_1^0 \cdot r_2^0}} \quad (4)$$

$$q_i^+ = \overline{\overline{r_0^+ \cdot r_0^0 \cdot r_1^+ \cdot r_0^0 \cdot r_1^0 \cdot r_2^+}} \quad (5)$$

上記の論理式に基づいた構成を図1(b)に示す．

これらの論理式から制御信号の生成時間を単位ゲート遅延に基づいて算出する．なお，単位ゲート遅延とは，NAND，NOR，NOTゲートを1 Δ として換算したものである⁵⁾．1桁2ビット表現した場合は3 Δ となり，3ビット表現した場合は1ビットの参照で各値を検出できるため，2 Δ となる．

3. 新たな高速加減算器

本章では，我々が既に提案してきた高速加減算器の特長を活かし，加算及び減算の並列性をより高めた除算器向け高速冗長2進加減算器を提案する．

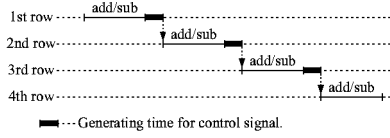


Fig. 2 Delay time of conventional divider.

3.1 従来の高速加減算器を適用した場合

本節では、既に提案してきた高速加減算器を除算器に適用した場合について考察する．この高速加減算器を除算器に適用した場合の演算の流れを図2に示す．この図からわかるように、逐次的に制御信号の生成及び加減算を繰り返す形となっている．従って、本加減算器を除算器に適用した場合、1行の遅延時間は加減算器の遅延時間 7Δ に、制御信号の生成時間 2Δ が加わるため、 9Δ に増大する．

3.2 新たな冗長2進加減算器の構成法

前節において、我々が既に提案してきた高速加減算器を除算器に適用すると、加減算器の遅延時間に制御信号の生成時間が加わり、1行の遅延時間が増加することを述べた．そこで、本節ではこの加減算器を改良し、新たな除算器向け高速冗長2進加減算器の構成法を提案する．

本構成法では、従来2つのステップで行われていた加減算を3つのステップで行う．ここで各ステップの演算を行うセルをそれぞれIセル、IIセル、IIIセルとする．本構成法では制御信号の入力はIセルではなくIIセルで行い、加減算と制御信号の生成を同時に行う．また、各セルでそれぞれ異なる表現方法を用いることにより、各セルでの演算を効率よく行い、ハードウェア量の増加を抑える．以下に本加減算器の各セルの構成法を示す．

ステップ1では1桁3ビット表現された入力 $a_j = \{a_j^-, a_j^0, a_j^+\}$ 、 $b_j = \{b_j^-, b_j^0, b_j^+\}$ の各桁における加算($a_j + b_j$)の結果 st_j 、及び減算($a_j - b_j$)の結果 dt_j を、それぞれ $\{2, \bar{1}, 0, 1, 2\}$ の範囲で同時に求める．ここで $\bar{2}$

Table 1 Computation rules for first step.

(a) Addition				(b) Subtraction			
a_j	b_j	st_j	$pm1_j$	a_j	b_j	dt_j	$pm1_j$
$\bar{1}$	$\bar{1}$	$\bar{2}$	0	$\bar{1}$	1	$\bar{2}$	0
	0	$\bar{1}$	1		0	$\bar{1}$	1
	1	0	0		$\bar{1}$	0	0
0	$\bar{1}$	$\bar{1}$	1	0	1	$\bar{1}$	1
	0	0	0		0	0	0
	1	1	1		$\bar{1}$	1	1
1	$\bar{1}$	0	0	1	1	0	0
	0	1	1		0	1	1
	1	2	0		$\bar{1}$	2	0

a_{j+1}	b_{j+1}	nm_j
$a_{j+1} = \bar{1}$ or $b_{j+1} = \bar{1}$		0
Otherwise		1

a_{j+1}	b_{j+1}	ns_j
$a_{j+1} = 1$ or $b_{j+1} = \bar{1}$		0
Otherwise		1

は -2 を、 $\bar{1}$ は -1 を示す．これらを、それぞれ4ビットで $st_j = \{st_j^{-2}, st_j^{-1}, st_j^0, st_j^1, st_j^2\}$ 、 $dt_j = \{dt_j^{-2}, dt_j^{-1}, dt_j^0, dt_j^1, dt_j^2\}$ と表記し、 st_j 及び dt_j が $\{0000\}$ の場合は、加算結果または減算結果が0であることを示す．また、次のステップでの出力を決定するための下位桁の条件も求める³⁾⁶⁾．そして、加算または減算結果が $+1$ または -1 になる条件 $pm1_j$ を求める．これを用いることにより、次のステップでの演算を効率よく行うことができる．加減算のステップ1の計算規則を表1に示す．各変数を次のように定義すると、Iセルの論理式は以下ようになる．

- a_j, b_j : j 桁目の入力
- nm_j : 加算時での j 桁目の下位桁の条件
- nmb_j : 加算時での j 桁目の下位桁の条件($=\bar{nm}_j$)
- st_j : $a_j + b_j$ が各値になる条件
- ns_j : 減算時での j 桁目の下位桁の条件
- nsb_j : 減算時での j 桁目の下位桁の条件($=\bar{ns}_j$)
- dt_j : $a_j - b_j$ が各値になる条件
- $pm1_j$: 加算及び減算結果が $+1$ か -1 になる条件

< 加算部分 >

$$nm_j = \overline{a_{j+1}^- + b_{j+1}^-} \quad (6)$$

$$nmb_j = \overline{a_{j+1}^- + b_{j+1}^-} \quad (7)$$

$$st_j^{-2} = \overline{a_j^- \cdot b_j^-} \quad (8)$$

$$st_j^{-1} = \overline{a_j^- \cdot b_j^0 + a_j^0 \cdot b_j^-} \quad (9)$$

$$st_j^{+1} = \overline{\overline{a_j^+ \cdot b_j^0 \cdot a_j^0 \cdot b_j^+}} \quad (10)$$

$$st_j^{+2} = \overline{\overline{a_j^+ \cdot b_j^+}} \quad (11)$$

< 減算部分 >

$$ns_j = \overline{\overline{a_{j+1}^+ + b_{j+1}^-}} \quad (12)$$

$$nsb_j = \overline{\overline{a_{j+1}^+ + b_{j+1}^+}} \quad (13)$$

$$dt_j^{-2} = \overline{\overline{a_j^- \cdot b_j^+}} \quad (14)$$

$$dt_j^{-1} = \overline{\overline{a_j^- \cdot b_j^0 \cdot a_j^0 \cdot b_j^+}} \quad (15)$$

$$dt_j^{+1} = \overline{\overline{a_j^+ \cdot b_j^0 \cdot a_j^0 \cdot b_j^-}} \quad (16)$$

$$dt_j^{+2} = \overline{\overline{a_j^+ \cdot b_j^-}} \quad (17)$$

< 加算・減算共通部分 >

$$pm1_j = \overline{\overline{a_j^- \cdot b_j^0 \cdot a_j^0 \cdot b_j^- \cdot a_j^+ \cdot b_j^0 \cdot a_j^0 \cdot b_j^+}} \quad (18)$$

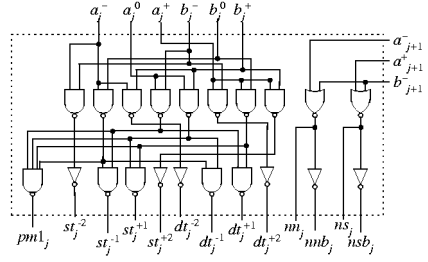
上記の論理式に基づいた構成を図3(a)に示す。

次にステップ2ではIセルの各信号と制御信号の入力を行い、桁上げが伝搬しないような中間桁上げまたは中間桁借り c_j 、中間和または中間差 s_j を求める。以降、中間和・中間差の2つを中間和、中間桁上げ・中間桁借りの2つを中間桁上げと記述する。Iセルの演算時間は上式から 2Δ となり、制御信号の生成時間と同じであるため、IIセルの演算は制御信号の生成を待たずに開始可能である。

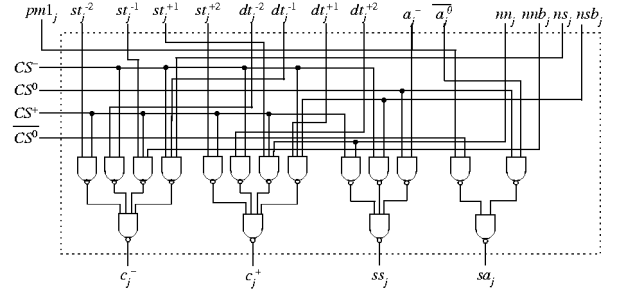
加算時及び減算時の計算規則をそれぞれ表2(a), (b)に示す³⁾⁶⁾。ここで中間和に注目すると、中間和はIセルで求めた $pm1_j$ から絶対値を、下位桁の条件から符号を容易に求められることがわかる。

そこでIIセルの出力は、中間桁上げには一般的な冗長2進表現($c_j = \{c_j^- c_j^+\}$)を用い、中間和は従来とは異なる表現($s_j = \{ss_j sa_j\}$)を用いる。ここで ss_j は中間和の符号が負になる条件を示し、 sa_j は絶対値が1になる条件を示す。

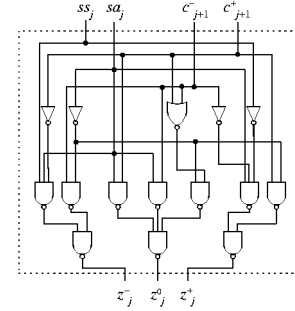
次に、制御信号 CS を $\{CS^+ CS^0 CS^-\}$ と表記し、 CS が $\{100\}$ のときは加算、 $\{001\}$ のときは減算、そして $\{010\}$ のときは演算をせず中間桁上げを0、中間和を a_j として出力する。以上からIIセルの論理式は以下ようになる。



(a) I cell.



(b) II cell.



(c) III cell.

Fig. 3 Structure of proposed adder/subtractor.

$$c_j^- = \frac{\overline{\overline{CS^+ \cdot st_j^{-1} \cdot nnb_j \cdot CS^- \cdot dt_j^{-1} \cdot ns_j}}}{\overline{\overline{CS^+ \cdot st_j^{-2} \cdot CS^- \cdot dt_j^{-2}}}} \quad (19)$$

$$c_j^+ = \frac{\overline{\overline{CS^+ \cdot st_j^{+1} \cdot nn_j \cdot CS^- \cdot dt_j^{+1} \cdot nsb_j}}}{\overline{\overline{CS^+ \cdot st_j^{+2} \cdot CS^- \cdot dt_j^{+2}}}} \quad (20)$$

$$ss_j = \overline{\overline{CS^+ \cdot nm_j \cdot CS^- \cdot nsb_j \cdot CS^0 \cdot a_j^-}} \quad (21)$$

$$sa_j = \overline{\overline{CS^0 \cdot pm1_j \cdot CS^0 \cdot a_j^0}} \quad (22)$$

上記の論理式に基づいた構成を図3(b)に示す。

この論理式から遅延時間を算出すると、 $\overline{CS^0}$ と $\overline{a_j^0}$ が影響して 3Δ となる。しかし、これらの値はIIセルの演算開始までに生成可能であるため、IIセルの遅延時間は実質 2Δ である。

最後にステップ3ではIIセルで得られた中間桁上げ、中間和に対して $z_j = s_j + c_{j+1}$ を計算し、加減

Table 2 Computation rules for second step.

(a) Addition($CS = 100$)				(b) Subtraction($CS = 001$)			
st_j	nm_j, nmb_j	c_j	s_j	dt_j	ns_j, nsb_j	c_j	s_j
2	—	1	0	2	—	1	0
1	$nmb_j=1$	1	1	1	$nsb_j=1$	0	1
	$nm_j=1$	0	1		$ns_j=1$	1	1
0	—	0	0	0	—	0	0
1	$nmb_j=1$	0	1	1	$nsb_j=1$	1	1
	$nm_j=1$	1	1		$ns_j=1$	0	1
2	—	1	0	2	—	1	0

Table 3 Computation rules for third step.

c_{j+1}	ss_j	sa_j	z_j
1	—	0	1
	—	1	0
0	—	0	0
	0	1	1
	1	1	1
1	—	0	1
	—	1	0

算の結果を1桁3ビット表現で得る。IIIセルの計算規則を表3に示す。この計算規則に基づいたIIIセルの論理式は以下ようになる。

$$z_j^- = \overline{\overline{ss_j \cdot sa_j \cdot c_{j+1}^+ \cdot sa_j \cdot c_{j+1}^-}} \quad (23)$$

$$z_j^0 = \overline{\overline{sa_j \cdot c_{j+1}^- \cdot sa_j \cdot c_{j+1}^+ \cdot sa_j \cdot c_{j+1}^- + c_{j+1}^+}} \quad (24)$$

$$z_j^+ = \overline{\overline{ss_j \cdot sa_j \cdot c_{j+1}^- \cdot sa_j \cdot c_{j+1}^+}} \quad (25)$$

上記の論理式に基づいた構成を3(c)に示す。

この論理式から遅延時間を算出すると、 3Δ となる。IセルとIIセルの遅延時間が共に 2Δ であるため、本提案の加減算器の演算時間は従来の高速加減算器と等しい 7Δ となる。また、本提案の加減算器を適用した場合の除算器の演算の流れは図4のような形になり、制御信号の生成時間は全体の演算時間に影響しない。従って、除算器1行の遅延時間は加減算器自体の遅延時間と同等となる。

4. 加減算器上位桁部の構成

本章では冗長2進数を用いた除算器に適用するための、加減算器上位桁の構成法を提案する。

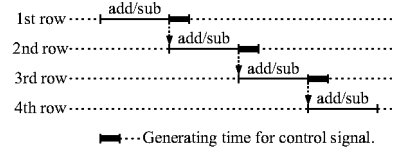


Fig. 4 Delay time of proposed divider.

4.1 商を決定する際の問題

本除算器に適用する2章のアルゴリズムは、部分剰余の上位3桁（整数部1桁と小数部2桁）の符号を検出して商の各桁を生成する²⁾。故に、このアルゴリズムを適用するには部分剰余の整数部が1桁で出力される必要があるが、部分剰余に冗長2進数を用いているため、1桁で出力されない場合もある。この場合、商生成の際に正しい商が選択されないため、部分剰余生成において整数部を1桁で出力するための対策が必要となる。

4.2 部分剰余の整数部

部分剰余を R 、除数を Y とすると、2章のアルゴリズムを用いた場合、 $|R| < 1$ の関係が成り立つ²⁾。これは、例えば整数部2桁目が1の場合、1桁目は必ず1となることを示す。つまり整数部は $1\bar{1}$ となり、 01 と表現できる。同様に、整数部2桁目が $\bar{1}$ の場合は1桁目は必ず1になり、 $0\bar{1}$ と表現できる。従って、部分剰余の整数部 $1\bar{1}$ を 01 に、 $\bar{1}1$ を $0\bar{1}$ に変換することで整数部を1桁に変換できる。

4.3 加減算器整数部の構成

本節では、前節で示した変換規則を適用した場合の加減算器整数部の構成を示す。この場合の構成は、図5(a)になる。図5(a)において、Converterとは部分剰余の整数部の2桁を1桁に変換するための回路である。この図では、前行の部分剰余をシフトした値を r 、除数を y 、生成される部分剰余を z とし、除数の整数部は1桁とする。 $|R| < 1$ であるため、図より上位には桁上げは生じず、整数部3桁

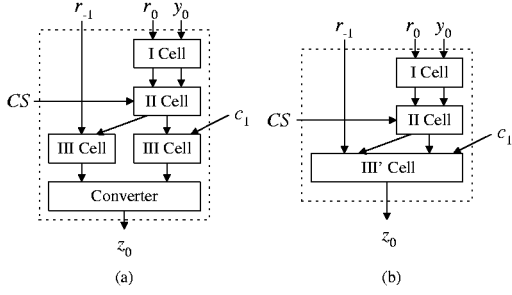


Fig. 5 Structure of integer part in adder/subtractor.

目以降は0となる．図5(a)の構成では，IIIセルの演算終了後に変換を行うため，演算時間が増加する．そこで，図5(b)のようにIIIセルの演算と変換を同時に行うIII'セルを提案し，演算時間を削減する．

4.4 III'セルの構成

本節では，前節で示したIII'セルの構成について述べる．まず，変数を以下のように定義する．

- r_{-1} : 前行の部分剰余の最上位桁 $\{r_{-1}^0 r_{-1}^1 r_{-1}^2\}$
- c_0 : 図5のIIセルからの中間桁上げ $\{c_0^- c_0^+\}$
- ss_0 : 図5のIIセルの中間和が負である条件
- sa_0 : 図5のIIセルの中間和が1か $\bar{1}$ である条件
- c_1 : 下位桁からの中間桁上げ $\{c_1^- c_1^+\}$
- cz_j : c_j が0である条件
- z_0 : 部分剰余の整数部 $\{z_0^- z_0^0 z_0^+\}$

III'セルの出力 z_0 を求める論理式を以下に示す．はじめに，各中間桁上げが0になる条件を求める．

$$cz_0 = \overline{c_0^- + c_0^+} \quad (26)$$

$$cz_1 = \overline{c_1^- + c_1^+} \quad (27)$$

整数部の変換規則に基づいて z_0 求めると，以下の式で表される．

$$z_0^- = \overline{\overline{\overline{r_{-1}^0 \cdot cz_0 \cdot r_{-1}^0 \cdot c_0^-}} \cdot \overline{r_{-1}^0 \cdot cz_0 \cdot ss_0 \cdot sa_0 \cdot cz_1 \cdot r_{-1}^0 \cdot cz_0 \cdot \overline{sa_0} \cdot c_1^-} \cdot \overline{r_{-1}^- \cdot ss_0 \cdot sa_0 \cdot cz_1 \cdot r_{-1}^- \cdot \overline{sa_0} \cdot c_1^-} \cdot \overline{c_0^- \cdot \overline{sa_0} \cdot c_1^-}} \quad (28)$$

$$z_0^0 = \overline{sa_0 \cdot c_1^- \cdot \overline{sa_0} \cdot c_1^+ \cdot \overline{sa_0} \cdot cz_1} \quad (29)$$

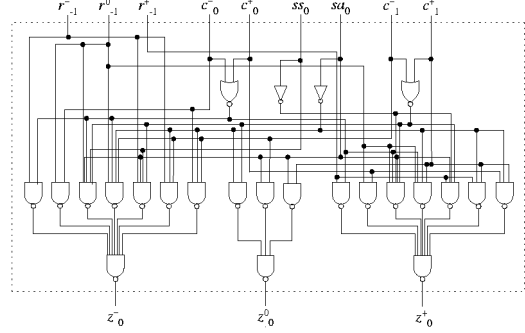


Fig. 6 Structure of III' cell.

$$z_0^+ = \overline{\overline{\overline{r_{-1}^+ \cdot cz_0 \cdot r_{-1}^0 \cdot c_0^+}} \cdot \overline{r_{-1}^0 \cdot cz_0 \cdot \overline{ss_0} \cdot sa_0 \cdot cz_1 \cdot r_{-1}^0 \cdot cz_0 \cdot \overline{sa_0} \cdot c_1^+} \cdot \overline{r_{-1}^+ \cdot ss_0 \cdot sa_0 \cdot cz_1 \cdot r_{-1}^+ \cdot \overline{sa_0} \cdot c_1^+} \cdot \overline{c_0^+ \cdot \overline{sa_0} \cdot c_1^+}} \quad (30)$$

上記の論理式に基づくIII'セルの構成を図6に示す．

この式よりIII'セルの遅延時間を算出すると 3Δ となり，IIIセルの遅延時間と同じである．

5. 除算器の構成法

本章では，除算器の構成について示す．構成は図7のようになる．この図は演算対象となる被除数 X ，除数 Y 及び商 Q が共に5桁の場合である．

ここで， Q セルは商の各桁を求めるセルで，前行の部分剰余の上位3桁から求める．また， R セルは小数部の各桁の部分剰余の生成を， R' セルは整数部の部分剰余を生成するセルである．

5.1 従来の加減算器適用時の構成

本節では，従来の冗長2進加減算器を除算器に適用した場合の構成法について述べる．

この場合の R セルの構成は，図8(a)となる．なお，このセルで用いられる冗長2進加算器は，従来の1桁2ビット表現による冗長2進加算器である⁶⁾．また， Q セルの構成は1桁2ビット表現を用いているので，図1(a)となる．

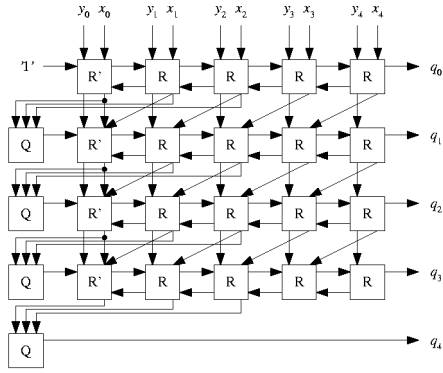


Fig. 7 Structure of redundant binary divider.

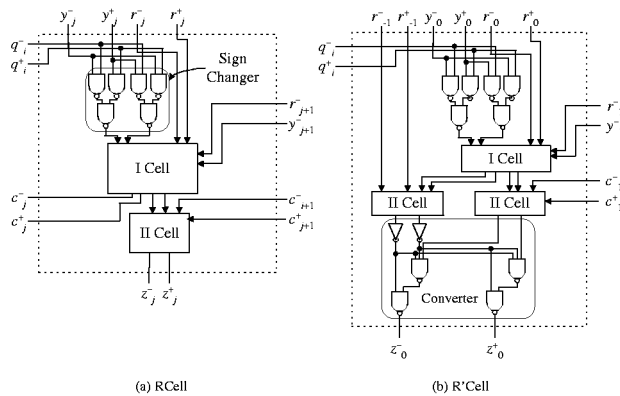


Fig. 8 Structure of conventional R and R' cell.

RセルとR'セルの遅延時間を比較した場合、加算器の遅延時間と符号変換器、変換回路の遅延時間の和となるR'セルのほうが大きい。また、演算の流れが図2となり、除算器1行あたりではこのR'セルの遅延時間にQセルの遅延時間が加わる。

5.2 本提案の加減算器適用時の構成

本節では、本提案の高速加減算器を除算器に適用した場合の構成法について述べる。

この場合のRセルの構成は、図9(a)となる。そしてR'セルの構成は、図9(b)となる。本構成は1桁3ビット表現を用いているため、Qセルの構成は図1(b)となる。この図を見てわかるように本加減算器は符号変換器を必要としない。また、制御信号の入力はIIセルで行うため、これにより制御信号の生成と加減算が同時に行える。さらにIII'セルを

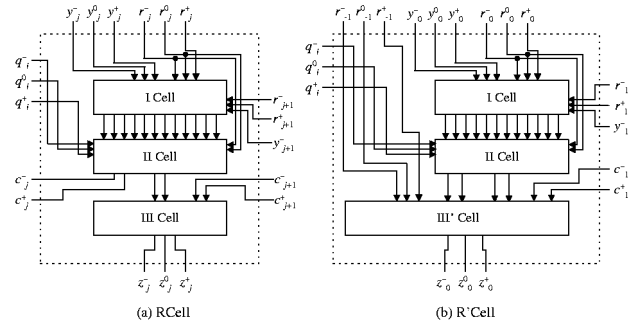


Fig. 9 Structure of proposed R and R' cell.

用いることにより、IIIセルと同じ遅延時間で整数部の変換も行うことができる。このため、従来の構成と比較して高速に演算を行える。

6. 性能評価

本章では、除算器に対して性能評価を行う。

6.1 論理的評価

本節では、単位ゲート遅延に基づく評価を行う。従来の加減算器を適用した場合は、冗長2進加算器の遅延時間が 9Δ であるため、5.1節の考察から除算器1行の遅延時間は 17Δ となる⁶⁾。

一方、本提案の加減算器を適用した場合の遅延時間は、5.2節から加減算器自体の遅延時間 7Δ に等しいことがわかる。従って、本提案の除算器は、従来の加減算器を適用した除算器と比較して、約2.4倍の高速化が得られることになる。

6.2 VLSI評価

本節では、VLSI設計システムPARTHENONを用いて評価を行う⁴⁾。評価対象を消費電力、面積、ゲート数、最大遅延時間とし、桁数を16桁から256桁まで変えて評価を行った。この結果を表4に示す。なお、実用部品として用いたセルライブラリの設計ルールは $0.6 \mu\text{m}$ CMOSスタンダードセル(VLSIテクノロジー社)であり、電源電圧は $5.0[\text{V}]$ とした。

Table 4 VLSI evaluation of each divider.

(a) Power. (mW/MHz)			(b) Area. (mm ²)		
Digit width	Conventional divider	Proposed divider	Digit width	Conventional divider	Proposed divider
16	7.93	12.88	16	1.42	2.20
32	31.66	50.21	32	5.63	8.63
64	125.91	198.16	64	22.37	34.20
128	495.46	787.34	128	88.24	136.27
256	1973.45	3142.38	256	351.25	544.47

(c) Gates. (gates)			(d) Delay time. (ns)			
Digit width	Conventional divider	Proposed divider	Digit width	Conventional divider	Proposed divider	Ratio of speed
16	11474	19115	16	437	198	2.21
32	45459	74930	32	882	396	2.23
64	180632	296417	64	1779	788	2.26
128	711401	1180657	128	3585	1578	2.27
256	2831594	4716796	256	7177	3169	2.26

この結果を見てわかるように、本提案の加減算器を適用した除算器は、従来のものと比較してハードウェア量を約1.6倍に抑えつつ、2.2倍以上の高速化が得られていることがわかる。

7. あとがき

本報告では、冗長2進数に基づく高速除算器の構成法を提案した。まず除算器の構成法として、高速性の観点から高木らが提案した除算アルゴリズムの考察を行った。次に、我々が提案してきた1桁2ビット/3ビット混合表現を用いた高速加減算器を改良した、新たな除算器向け高速加減算器を提案した。さらに、部分剰余の数表現の冗長性により商が正しく選択されない場合があることを示し、これに対応した除算器の構成法を提案した。

そして、論理的評価とVLSI設計システムPARTHENONを用いた設計及び評価を行った。その結果、本加減算器を適用した除算器は、ハードウェア量を約1.6倍に抑えつつ、2.2倍以上の高速化が可能となることを明らかにした。

今後の課題としては本加減算器の除算器以外への適用や、本除算器を実際にシステムに適用した場合の検討が挙げられる。

参考文献

- 1) A. Avizienis: Signed-Digit Number Representations for Fast Parallel Arithmetic, IEEE Trans. Elec. Comput., **EC-10-9**, 389/400 (1961).
- 2) 高木直史, 矢島脩三: 冗長2進表現を利用したVLSI向き高速除算器, 電子情報通信学会論文誌, **J67-D-4**, 450/457 (1984)
- 3) 日野杉充希, 恒川佳隆, 三浦守: 1桁2ビット/3ビット混合表現を用いた高速冗長2進加減算器, 電気学会論文誌, Vol.121-C, No.4, pp.733-741, 2000
- 4) NTTデータ通信株式会社: PARTHENON User's Manual (1990)
- 5) K. Hwang: "Computer Arithmetic/Principles, Architecture, and Design", John Wiley & Sons (1979).
- 6) 恒川佳隆, 日野杉充希, 齊藤正人, 虻川勝己, 三浦守: 1桁2ビット/3ビット混合型高性能冗長2進加算器とその乗算器への応用, 電気学会論文誌, **119-C-5**, 644/653 (1999)