計測自動制御学会東北支部 第160回研究集会(1996.5.24)
資料番号 160-14

選択型セルアレイ高速除算器の設計とVLSI評価
Design and VLSI Evaluation of High-Speed Cellular Array Divider with Selection Function

○日野杉充希*，恒川佳隆*，吉田等明*，三浦守*
○Mitsuki Hinosugi*, Yoshitaka Tsunekawa*, Hitoaki Yoshida*, Mamoru Miura*
*岩手大学
*Iwate University

キーワード： 高速除算器(High-Speed Divider)，セルアレイ(Cellular Array)，
引放し法(Nonrestoring Algorithm)，VLSI評価(VLSI Evaluation)
連絡先： 〒020 盛岡市上田4-3-5 岩手大学工学部情報工学科
日野杉充希，Tel. 0196-21-6468，Fax. 0196-23-5491，E-mail: mitsuki@cis.iwate-u.ac.jp

1. はじめに

これまでに除算器は，ほとんど実用化されてこなかった．その理由として，加算や減算，乗算などと違い比較・判断をしながら演算が行われるため，回路が複雑となりコストに見合った高速性が得られない．また，普通除算の使用頻度があまり高くないことなどが挙げられる．

しかし現在，ディジタル信号処理やロボット制御など様々な分野において，高速な除算器が必要とされている．また，膨大なハードウェア量を扱える現在において，従来から提案されてきたVLSI向きのセルアレイ除算器の実現が容易となってきた．

これまでに，いくつかのセルアレイ除算器が提案されている[1)-4)]．従来の除算器は，各行の演算で部分剰余を必要とするため，1行ずつ逐次的に演算が行われていた．従って，遅延時間は語長に比例する値となる．これが，除算器を設計する上で高速化の大きな問題点である．

本論文ではディジタル信号処理などの分野での実用化を考え，被除数，除数が負の場合でも演算可能であり，また，前行の商ビットの生成と次行の部分剰余，CLA信号の生成を並列的に行うという新たな手法を用いたセルアレイ高速除算器を提案する．さらに，単位ゲート遅延を用いて本除算器の遅延時間を算出する．そして最後に，VLSI設計システムPARTHENONを用いて本除算器のVLSI評価を行う．

2. 除算アルゴリズムの選択

2.1 代表的な除算アルゴリズム

代表的な除算アルゴリズムとして，引戻し法，引放し法，SRT法，反復法などがある[5)]．

引戻し法は，被除数，除数，共に絶対値表現された正の値しか扱えない．また，部分剰余から除数を引き，その結果を負であ

るならまた除数を足して元に戻すという操作を行うため，演算時間が遅くなる．

引放し法は，被除数，除数，共に 2 の補数表現された負の値でも扱え，さらに部分剰余から除数を引き過ぎた場合でも，次の演算で部分剰余に除数を足すことにより，引戻し法に比べ元に戻す必要がなく，1 行の演算時間は常に一定である．

SRT 法は，引放し法の演算の加減算に加えシフトだけ行うという演算があるために，引放し法に比べて演算時間は速い．しかし，被除数，除数は，あらかじめ正規化しておかなければならない．また，得られた商は冗長 2 進表現されているために，最後に普通の 2 進数に変換しなければならない．

反復法は，乗算器を用いて除算を行うため，乗算器を高速化することにより除算結果も高速に得られる．しかし，直接剰余を求めることが不可能である．また，他の方法と比べて誤差が大きくなる傾向がある

これらの方法から，ディジタル信号処理，ロボット制御などの分野で実用化する場合，2 の補数表現された負の値でも扱えるアルゴリズムとして，本研究では引放し法を基に考える．

2.2 引放し法アルゴリズム

本節では引放し法アルゴリズムを説明する．語長が $2L$ ビットのとき全ての値が符号ビットを持つので，被除数，剰余は $2L+1$ ビット，除数，部分剰余，商は $L+1$ ビットである．よって，

$$\text{被除数}：N = n_0.n_1n_2\cdots n_{2L}$$
$$\text{除数}：D = d_0.d_1d_2\cdots d_L$$
$$\text{部分剰余}：P = p_0.p_1p_2\cdots p_L$$
$$\text{商}：Q = q_0.q_1q_2\cdots q_L$$
$$\text{剰余}：R = r_0.r_1r_2\cdots r_{2L}$$

とする．なお，添え字の 0 の付いたものは符号ビットである．

以下に，アルゴリズムの手順を示す．

ステップ 1

$$P := \begin{cases} N-D & if \quad N,D\text{が同符号} \\ N+D & if \quad N,D\text{が異符号} \end{cases}$$

ステップ 2

for i:=0 step 1 until L-1 do
begin
$P := P$ shl 1;

$$q_i := \begin{cases} 1 & if \quad P,D\text{が同符号} \\ 0 & if \quad P,D\text{が異符号} \end{cases}$$

$$P := \begin{cases} P-D & if \quad q_i = 1 \\ P+D & if \quad q_i = 0 \end{cases}$$

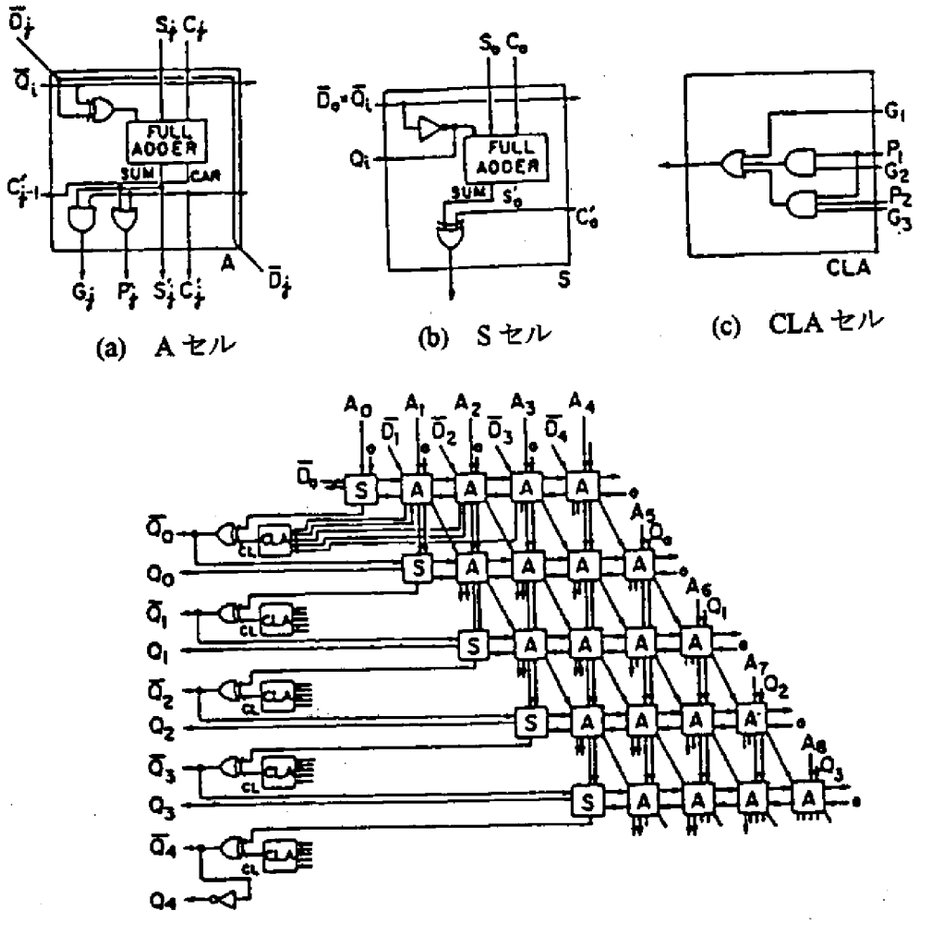
end

ステップ 3

$Q := q_o.q_1q_2\cdots q_{L-1}1$
$R := 2^{-L}\cdot P$

3. 従来の引放し法による セルアレイ除算器

これまでに，引放し法によるセルアレイ除算器が提案されている [2), 3)]．セルアレイとは，セルという最小単位の回路をアレイ上に規則正しく並べた構成をいう．よって，セルアレイは回路の設計や拡張が容易であり，VLSI 向きと言える．その中の高速な除算器として代表されるのが，キャリーセーブ方式と CLA 回路を用いたものである．本



(a) Aセル　(b) Sセル　(c) CLAセル

(d) 8ビット型CLA/キャリーセーブを用いたセルアレイ除算器

図1　CLA/キャリーセーブを用いたセルアレイ除算器

章では，この除算器について述べる．なお，この除算器では引放し法を用いているが，被除数，除数ともに正の値しか扱えない．

3.1　高速化の手法

この除算器では，CLAとキャリーセーブの手法を用いて高速な除算器を実現している．

除算器を桁上げ伝播加算器で実現する場合，各演算でキャリーが完全に伝播することを必要とするため，遅延時間は非常に大きなものとなる．しかし，この除算器では部分剰余を和とキャリーのベクトルに分けることにより，得られたキャリーベクトルをその行で伝播せずに次行の演算で加算している．これが，キャリーセーブ方式である．

また，部分剰余の符号ビットを高速に生成するために，CLA回路を用いて符号ビットへのキャリーを生成している．

これらの手法を用いたこの除算器の遅延時間は，語長Nのとき$O(M\log N)$の値となる．

3.2　構成法

前述の手法を用いた除算器は，3種類のセルから構成される．

Aセルでは，主に和S_jとキャリーC_jのベクトルに分解した部分剰余を，全加算器によって生成する（図1(a)）．また，CLA信号生成のための生成信号G_jや伝播信号Pj

もここで生成する．

S セルは各行に 1 つずつ用いており，仮の符号ビットを生成する（図 1(b)）．そして，この信号と CLA 信号を XOR（排他的論理和）をとって符号ビットを生成し，この信号が次行の S セルに入力され商ビット Q_i を生成する．

CLA セルは各行に 1 つずつ用いており，各 A セルで生成された生成信号 G や伝播信号 P がこのセルに入力され，符号ビット生成のための CLA 信号 CL を生成する（図 1(c)）．

この除算器の語長が 8 ビットのときの全体の構成を図 1(d)に示す．

4. 選択型セルアレイ高速除算器

4.1 新たな高速化の手法

2.2 節のアルゴリズムで説明したように，前述の除算器は，各行で部分剰余と除数の符号を比較して商ビットを決め，この商ビットによって次行の演算を決定するため，演算は 1 行ずつ順次行われていく．これは，引放し法に限らず，引戻し法，SRT 法など前行の部分剰余を必要とする除算法は全てそうである．これが，一般的に他の演算器と比べて除算器が遅いといわれる最大の理由である．しかし，引放し法において各行の制御信号となる商ビットはたかだか 0 と 1 しかない．よって，前行の商ビットが決定する前に商ビットが 0 と 1 の場合の演算を行うことにより先行演算が可能となる．これが，新たな手法の基本的な考え方である．

なお，この手法を用いた本除算器を選択型セルアレイ高速除算器と名付ける．

本手法の具体的手順を以下に示す．

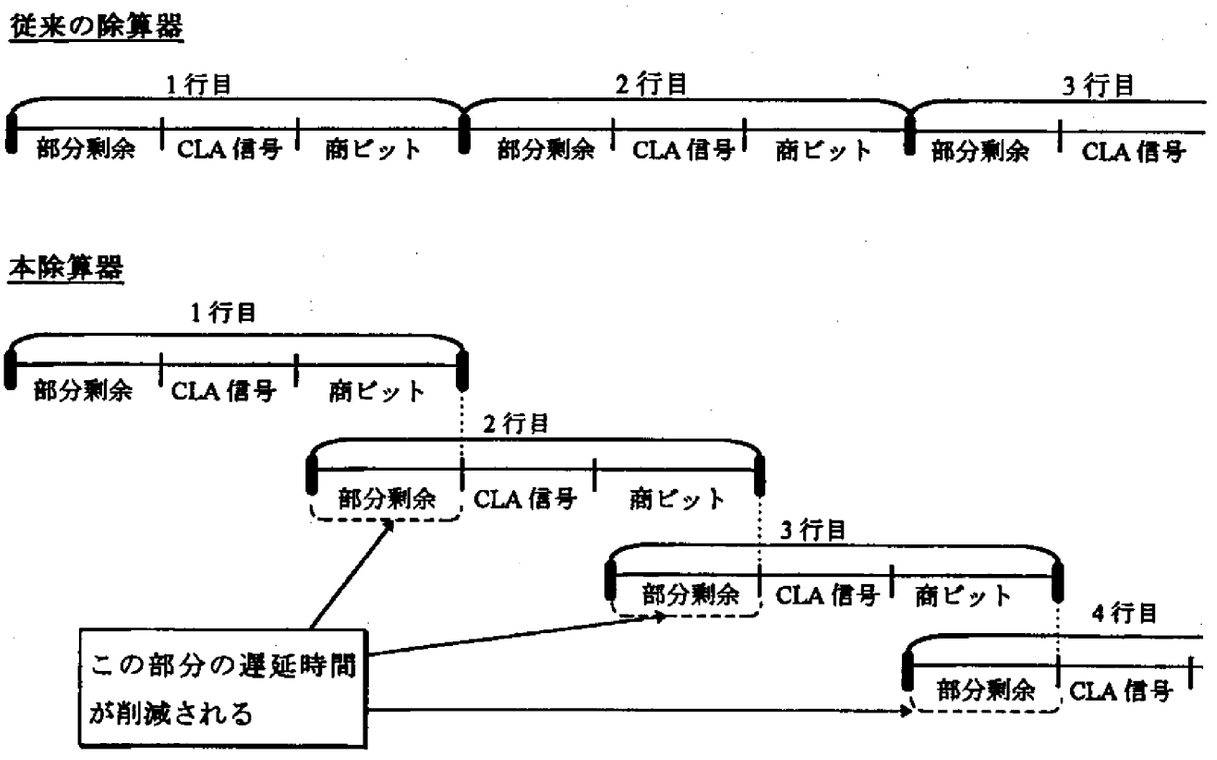


図 2　部分剰余生成時間削減の実現

● 部分剰余生成時間の削減

まず本手法の第1段として，前行の商ビットが決定する前に，商ビットが0と1の場合の次行の部分剰余を先に生成するように考える．これによって，加減算の遅延時間を削減することができる．この考え方を図2に示す．

この実現法として，部分剰余を生成するセルに加算用と減算用の2つの全加算器を用いる．また前述の除算器では，図1(d)に示すように，最右端のセルのキャリー入力に前行の商ビットを入力するようになっていた．しかし，本手法でそうした場合，そのセルだけが商ビットが決定しないかぎり演算を行うことができない．そこで，新たに最右端用のセルとしてRセルという新しいセルを用いる．

● CLA信号生成時間の削減

前述の実現法によって，部分剰余の生成時間を削減することができた．しかし，さらに検討してみた結果，CLA信号の生成時間が最大遅延パスとなることがわかった．そこでCLA信号も，商ビットが0と1の場合を先行演算するように考える．よって，CLA信号の生成時間を削減することが可能となる．

この実現法として，まず部分剰余を生成するセルでCLA信号生成に必要な生成信号，伝播信号をそれぞれ商ビットが0と1の場合を生成する．そして，各行に商ビットが0と1の場合のCLA信号生成を行うCLAセルを2つ用いる．

4.2 構成法

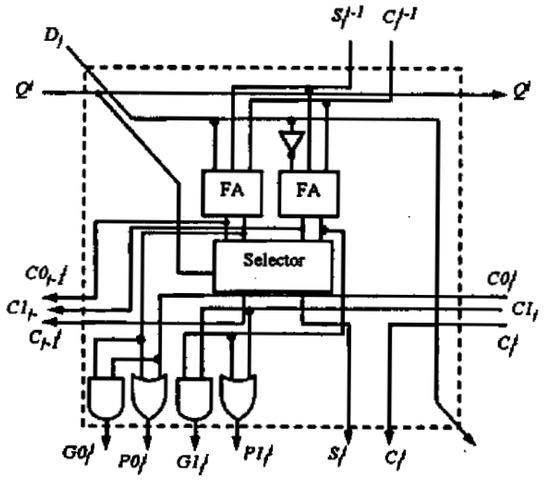
本除算器の構成は4種類のセルで構成する．

Bセルでは，商ビットが0の場合の部分剰余（和$S0_j^i$，キャリー$C0_j^i$）と1の場合の部分剰余（和$S1_j^i$，キャリー$C1_j^i$）を2つの全加算器によって生成する（図3(a)）．そして，商ビットが決まってから部分剰余をセレクタによって選択する．また，CLA信号生成のための生成信号や伝播信号も商ビットが0（生成信号$G0_j^i$，伝播信号$P0_j^i$）と1（生成信号$G1_j^i$，伝播信号$P1_j^i$）の場合をAND，ORゲートを2つずつ用いて生成する．

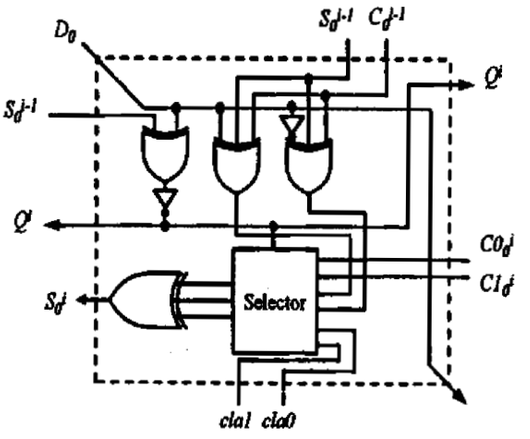
Rセルは各行の最右端に用いるセルである（図3(b)）．動作はBセルとほぼ同じで，最下位ビットの部分剰余の生成を目的としている．しかし，この部分にRセルという新しいセルを用いた理由は，4.1節で述べたようにこの部分にBセルを用いてキャリー入力に商ビットを入力するよう設計すると，このセルが最大遅延パスとなり本除算器の利点が活かされない．よって，全加算器のキャリー入力に加算演算用には'0'を，また減算演算用には'1'の定数を入力することによって，この問題を解決している．

CLAセルは各行に2つずつ用いており，それぞれ商ビットが0と1の場合のCLA信号を生成する．なお，CLAセルは前述の除算器で使われたCLAセルと全く同様のものである（図1(c)）．

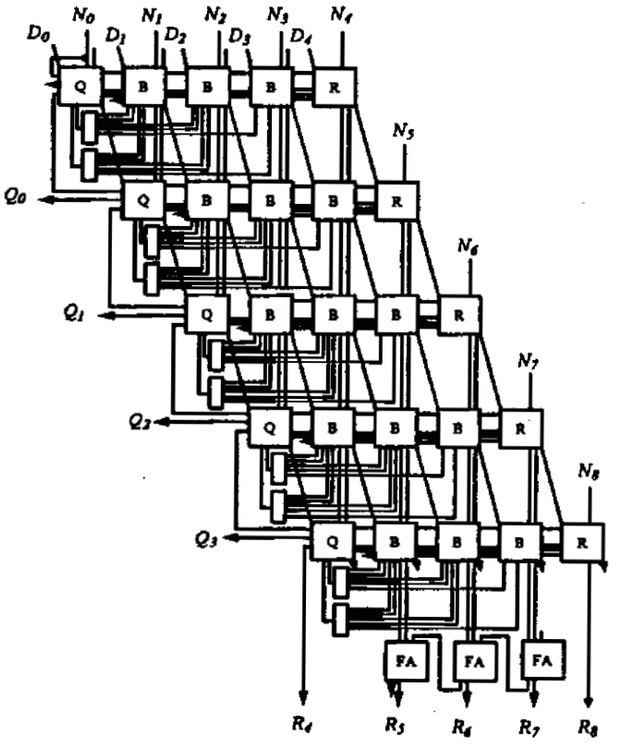
Qセルは各行に1つずつ用いており，符号ビット，商ビットの生成を目的としている（図3(c)）．まずQセルでは，商ビットが0と1の場合の最上位ビットの部分剰余

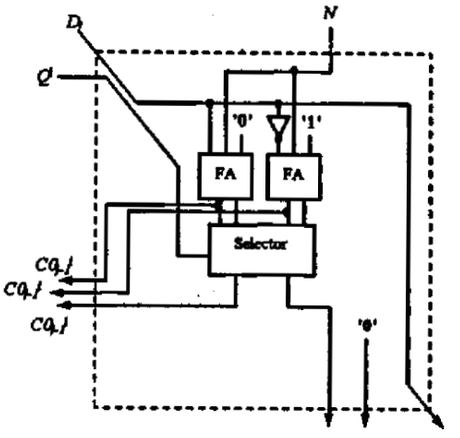


(a) Bセル



(b) Qセル





(c) Rセル

(d) 8ビット型選択型セルアレイ高速除算器

図3 選択型セルアレイ高速除算器の構成

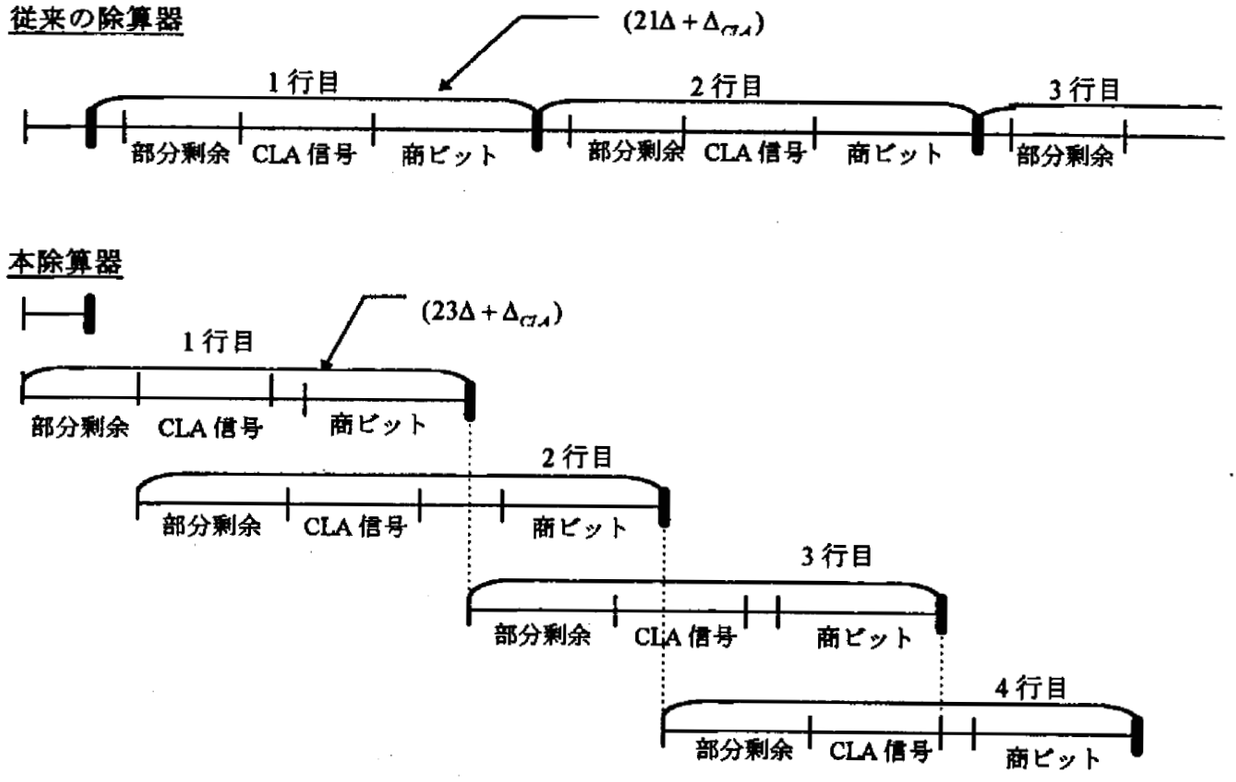


図4　遅延時間による比較

を生成する．そして，商ビットが0と1の場合の右隣のBセルからのキャリーをセレクタによって選択し，その結果を XOR ゲートによって符号ビット S' を生成する．さらに，その符号ビットが次行のQセルに入力され，除数の符号と比較し商ビット Q' を生成する．なお，従来の除算器では被除数，除数が正の値しか扱えないが，本除算器では負の値でも扱える機能をこのセルのに拡張した．

本除算器の語長が8ビットのときの全体の構成を図3(d)に示す．

5　遅延時間

5.1　演算プロセス

本除算器の演算プロセスは，図4に示すように従来の除算器と異なる．以下に本除算器の演算プロセスを説明するが，そのために図4の1行目，2行目，3行目に注目していただきたい．

基本的に，従来の除算器と本除算器の各行の演算プロセスはほとんど同じである．まず1行目の演算は，初回の制御信号の生成を待たずに開始している．これは，本除算器では制御信号が0と1の場合の部分剰余を先に演算できるからである．そして1行目の演算は順次に行われていくが，その間にも2行目の演算がすでに1行目の部分剰余の生成が終わった時点で開始している．これは，初回の制御信号がすでに決まっているので，2つのパターンの部分剰余を制御信号によって選択するのみとなっているためである．そして2行目の演算も順に行われていく．さらに，その間にも1行目の商ビットが生成された時点で3行目の演算

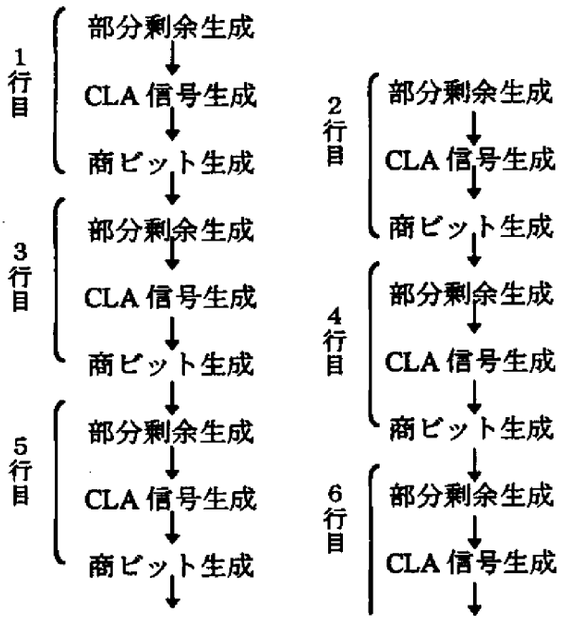


図5　本除算器の演算プロセス

が開始している．これもすでに2つのパターンの部分剰余は2行目の演算ですでに生成されており，1行目の商ビットによって選択するのみとなっているためである．

よって，この図から容易に2行目，4行目など偶数行の遅延時間が削減できることがわかる（図5）．このことから，従来の除算器の遅延時間は語長に比例する値となるが，本除算器では語長の半分に比例する値となる．

5.2　単位ゲート遅延を用いた遅延時間の算出

次に，本除算器の遅延時間を単位ゲート遅延を用いて算出する[5]．

基本的に本除算器と従来の除算器の各行の演算プロセスは同じであり，部分剰余の生成，CLA信号の生成，商ビットの生成という順に行われていく．それぞれの生成時間は，部分剰余は7Δ，CLA信号は語長が10ビット未満のとき6Δ，10ビット以上のとき8Δ（これは最大ファンイン数を8とした場合，CLA回路を2段にしなければならないことによる），また，商ビットは10Δとなることからそれぞれの除算器の1行当たりの遅延時間は，

$$\Delta_{line\ (従来)} = 21\Delta + \Delta_{CLA}$$
$$\Delta_{line\ (本除算器)} = 23\Delta + \Delta_{CLA}$$

となる．なお本除算器と従来の除算器との差が2Δあるのは，本除算器ではQセルで商ビットが0と1の場合のCLA信号，部分剰余の最上位ビットの和とキャリーを選択するためである．

よって，各除算器は$N/2$行あるので，

$$\Delta_{従来} = (21\Delta + \Delta_{CLA})N/2 + 8\Delta \cdots (1)$$
$$\Delta_{本除算器} = (23\Delta + \Delta_{CLA})N/4 - 2\Delta \cdots (2)$$

となる．なお，従来の除算器では$N/2$かけているのに対し本除算器で$N/4$かけているのは，5.1節で述べたように本除算器は偶数行の遅延時間が削減されるためである．

よって，(1)，(2)の式から容易に，

$$\Delta_{本除算器} \approx \Delta_{従来}/2$$

となることがわかる．

6.　PARTHENONによるVLSI評価

本研究ではさらに，VLSI設計システムPARTHENONにより本除算器を設計し，VLSI評価を行った．

6.1 VLSI設計

本除算器を設計するために，本除算器をSFL言語で記述した．PARTHENONではトップダウン設計が可能であるので，各セルをサブモジュールとして扱うことにより容易に設計することができる[6]．また，本除算器を任意の語長に設計できるよう，本除算器のSFLソースを自動作成するプログラムをC言語によって実現した．

6.2 VLSI評価

SFL言語で記述した本除算器を回路シミュレータSECONDSによって動作確認し，さらにシンセサイザSFLEXPで論理合成し，最適化ユーティリティOPT_MAPによってテクノロジ・マッピングと回路の最適化を行った．さらに，本除算器を語長を6ビットから32ビットまで変えてVLSI評価した．評価対象を消費電力，面積，ゲート数，商の生成時間，剰余の生成時間とし，結果を図6に示す．なお，用いた設計ルールは0.8μmCMOSスタンダードセル（VLSIテクノロジー社）である．

図6(d)の商の生成の遅延時間からみても，本除算器は従来の除算器より2倍以上高速になったことがわかる．具体的に，語長が24ビットのときで従来の除算器よりも2.35倍の高速化を図ることができた．しかも，消費電力，ゲート数ともに1.75倍，面積は1.81倍と，いずれも2倍以下となる．

7. おわりに

本研究において，前行の商ビットの生成と次行の部分剰余，CLA信号の生成を並列に行うという新たな手法を用いた選択型セルアレイ高速除算器を提案した．さらに，VLSI設計システムPARTHENONによって本除算器を設計しVLSI評価することによって，従来の除算器より2倍以上高速なものを実現できた．

参考文献

1) Dean, K. J.: Binary Division Using a Data Dependent Iterative Arrays, *Electronics Letters*, Vol. 4, 283/284(1968)

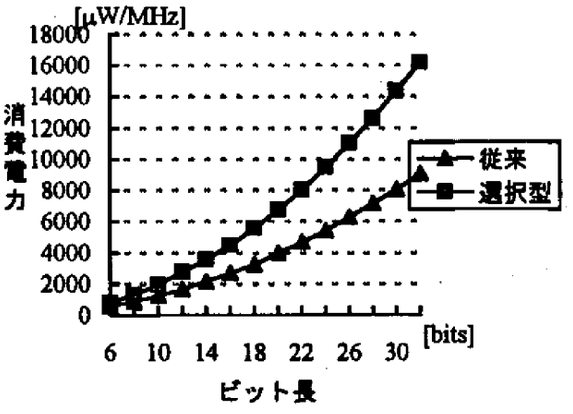
2) Guild, H. H.: Some Cellular Logic Arrays for Nonrestoring Binary Division, *The Radio and Elec. Engr.*, Vol. 39, 345/348(1970)

3) Cappa, M.. and Hamacher, V. C.: An Augmented Iterative Array for High-Speed Binary Division, *IEEE Trans. on Computers*, Vol. C-22, 172/175(1973)

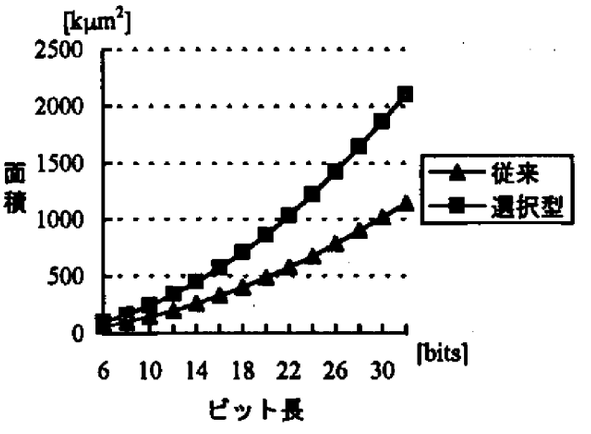
4) 高木直史，安浦寛人，矢島修三：冗長2進表現を利用したVLSI向き高速除算器，電子通信学会論文誌，Vol. J67-D No. 4, 450/457(1984)

5) K. Hwang : Computer Arithmetic/Principles, Architecture, and Design, John Wiley & Sons(1979)

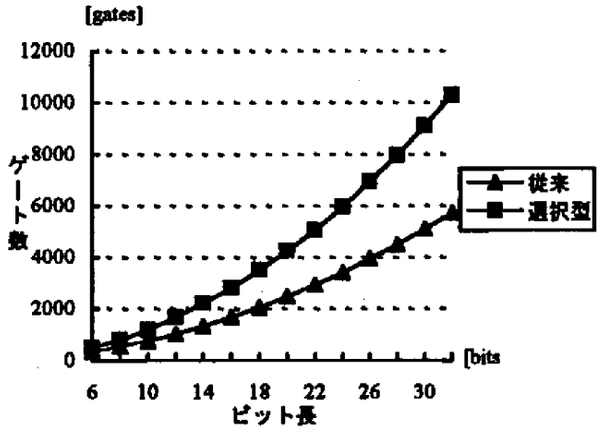
6) PARTHENON User's Manual, NTTデータ通信（株）(1990)



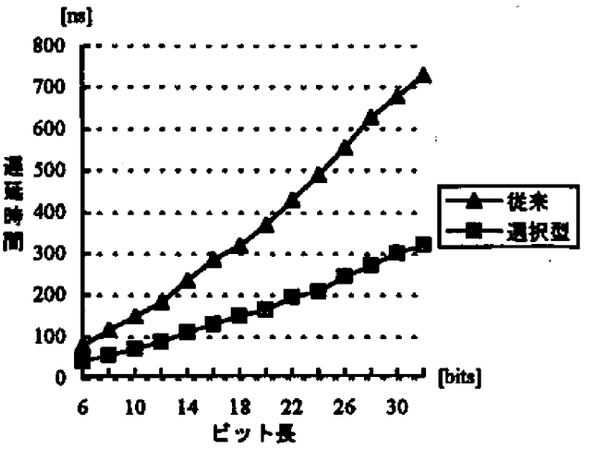
(a) 消費電力



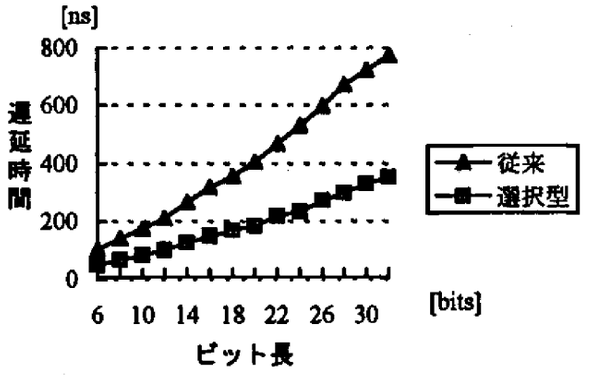
(b) 面積



(c) ゲート数



(d) 遅延時間（商の生成時間）



(e) 遅延時間（剰余の生成時間）

図 6 従来の除算器と本除算器の VLSI 評価