計測自動制御学会東北支部 第196回研究集会 (2001.7.30)
資料番号 196-3

拡張容易な任意精度乗除算器の構成について

Structure of Multiplier-Divider for Numbers of Arbitrary Word Length with Easy Expansibility

○眞田祐一*，恒川佳隆*

○Yuichi Sanada* and Yoshitaka Tsunekawa*

*岩手大学

*Iwate University

キーワード： 高精度 (high-precision)，加減算器 (adder-subtractor)，VLSI評価 (VLSI evaluation)，乗算器 (Multiplier)，除算器 (Divider)

連絡先： 〒020-8551 盛岡市上田4–3–5 岩手大学 工学部
眞田祐一，Tel.: (019)621-6468，Fax.: (019)621-6468，E-mail: t2301014@iwate-u.ac.jp

1. まえがき

現在，加減乗除等の算術演算は，計算機などのディジタルシステムにおいて，基本演算として広く用いられている．その中でも，乗算器，除算器はハードウェア量が加算器，減算器と比べ大きく，演算時間も大きくなる．そのため，これまでにさまざまな乗算器，除算器が考えられてきた[1)2)]．ディジタル信号処理やロボット制御などのリアルタイム処理を必要とする分野や，乗算，除算を繰り返し行うべき乗剰余演算機構を必要とするRSA暗号方式などの情報セキュリティ技術の分野においてより高速で高精度な乗算，除算が必要とされている[3)4)]．また，精度により強度を変える暗号などのアプリケーションにおいては，柔軟に精度を変更できる拡張容易性も必要とされてきている．

従来の組合せ回路方式の乗算器は，高速化を図る場合に適している．しかし，複数個の乗算器を用いて，高精度の乗算器を構成する場合，複数の乗算器を二次元的に組み合わせ，さらに乗算器以外に外部回路を必要とするので，回路が複雑化し拡張が容易にできなくなる．そのため，精度をn倍に拡張しようとすると面積がn^2倍以上になり，高精度な乗算器を構成する際，ハードウェア量と拡張性が問題となってくる．

また，除算に関しては，他の演算と異なり，比較，判断を必要とする演算であるため，複数の除算器を分散して用いることができない．ゆえに，任意の精度に拡張することが困難である．そのため，必要な精度ごとに除算器を構成する必要があり，高価格化と低効率化につながっていた．その結果，除算を用いないアルゴリズムの使用，または，除算をソフトウェアで行うことにより対処していたのが現状である．

それに対し，これまでに順序回路方式を用いた任意精度乗算器および除算器が提案されている[5)6)]．従来の組合せ回路方式では2次元的に拡張する必要があったが，これらの順序回路方式では1次元的に拡張することにより，任意精度の乗算器および除算器を構成することができた．しかし，従来型の乗算器は内部回路のビット数がモジュールの精度の2倍必要となり，ハードウェア量が大きくなっていた．また，モジュールをカスケード接続し高精度化する際，モジュール間のキャリーの伝播による処理時間の増大が問題となっていた．さらに，乗算，除算を繰り返し必要とするRSA暗号方式などの分野においては，乗算器，除算器を別々に用意する必要があり，ハードウェア量が大きくなっていた．

本稿では，同一モジュールでの乗除算が可能で，

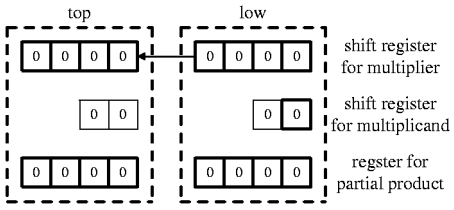

Fig. 1 Circuit behavior of Conventional Multiplier

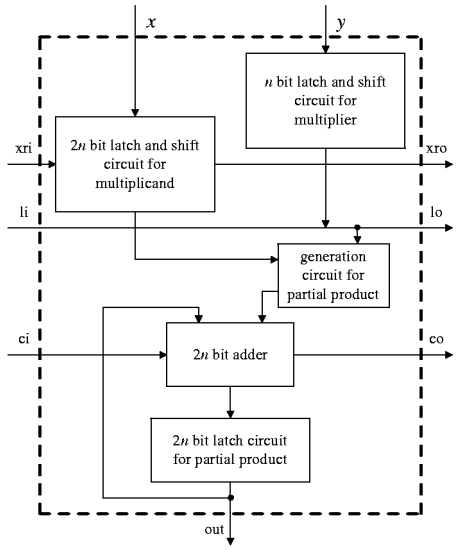

Fig. 2 Structure of Conventional Multiplier module.

任意精度に拡張容易な乗除算器の構成法を提案する．さらに，拡張性に適した高速加減算部分の構成を提案し，高速化を図る．

以上の手法を用いた本提案型が，RSA暗号方式などの乗算, 除算を繰り返し必要とする分野において極めて有効な演算器となることを明らかにする．

2. 従来の順序回路方式の乗算器，除算器

本章では，従来から提案されている順序回路方式を用いた任意精度乗算器および除算器の構成と回路動作について述べる．

2.1 従来の乗算器

図1に示すように，この乗算器は，被乗数，乗数をシフトし桁の重みを変えていく手法である．そのため，部分積レジスタ，部分積を加算する加算器は，1モジュールの精度の2倍のビット数を必要とする．その結果，被乗数も1モジュールの精度の2倍のビット数を必要とする．

モジュールの構成を図2に示す．これを複数個カスケード接続し，任意精度の乗算器を構成する．3モジュールのカスケード接続により1モジュールの3倍精度に拡張した構成を図3に示す．まず，3モジュール分の被乗数と乗数データは乗算開始信号によって，各モジュール内にラッチされる．次に，初段の乗数のLSBと全被乗数との乗算が並列に実行される．シフトパルスにより1モジュール分の乗数と全モジュール分の被乗数との乗算が終了すると，乗算終了信号が出力される．この信号が上位モジュールのシフト開始入力に与えられ，次段のモジュールの乗数と全モジュール分の被乗数との乗算も同様に行われる．さらに，そのモジュールの乗算終了信号により，次のモジュールへと順次乗算操作が移行する．最終段の乗算が終了したとき，乗算終了信号が出力され，初段のシフトパルスを制御する入力にフィードバックされる．シフトパルスの出力を遮断し，すべての乗算を終了する．

2.2 従来の除算器

図4に示すように，この除算器は被除数を1ビットずつシフトさせ，これを部分剰余とし，除数との減算を行い商を決定していく手法である．

モジュールの構成を図5に示す．これを複数個カスケード接続し，任意精度の除算器を構成する．3モジュールのカスケード接続により1モジュールの3倍精度に拡張した構成を図6に示す．除算開始信号により，除算データが各モジュール内にラッチされる．除算は，まず初段のクロックを用いて減算とシフトが行われ，部分剰余を得る．それが終了すると初段の除算終了信号が上位モジュールのシフト開始入力に与えられ，2段目のモジュール分のクロックを用いて同様に部分剰余を得る．さらに，その除算終了信号により次段のモジュールへと順次除算操作が移行し，最終段の除算終了信号が初段のシフト制御入力にフィードバックされ，シフト信号の出力を遮断し，演算が終了する．

3. 乗除算でのモジュールの共有化

本章では，従来の任意精度乗算器および除算器の構成をもとに，乗算除算両方の演算を行うことができる構成について検討する．

従来の乗算器および除算器は乗算，除算のみ行

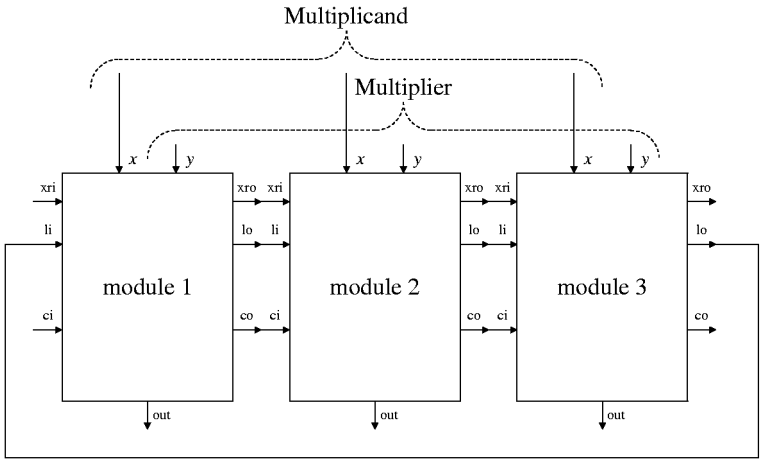

Fig. 3　Block diagram of Multiplier cascaded by three Multiplier module.

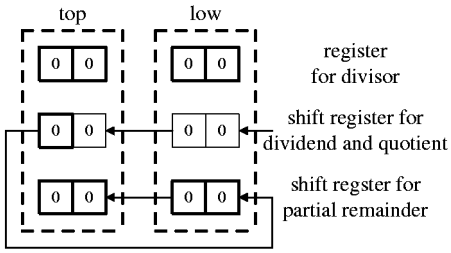

Fig. 4　Circuit behavior of Conventional Divider

えるものであり，両演算器ともに回路動作は大きく異なっていた．そこで，同一モジュールで乗算除算両演算を行うため，乗算時，除算時に使用する内部回路を最適な組合せで共有化できる構成が必要となる．

同一モジュールでの乗算除算両演算が可能な構成を考える上で，内部回路の共有化が必要となる．そのため，乗算および除算に特化したアルゴリズムでは，回路の共有化が困難となる．したがって，乗算，除算に特化したアルゴリズムを用いず，あえて従来の乗算器および除算器で用いたシフト加算，シフト減算アルゴリズムを用いることにする．

乗算時，除算時で内部回路を共有するにあたり，まず乗算時，除算時に使用する内部回路の動作を検討した．従来型の乗算器は，被乗数を左シフトすることにより，桁の重みを変え，部分積との加算を行っていた．部分積をラッチするレジスタはシフトしないので，部分積用のレジスタは1モジュールの精度の2倍のビット数必要となり，その加算を行う加算器も結果として2倍のビット数必要となっていた．

そこで，今回は図7に示すように，被乗数をシフトせず，部分積を右シフトすることにより桁の重

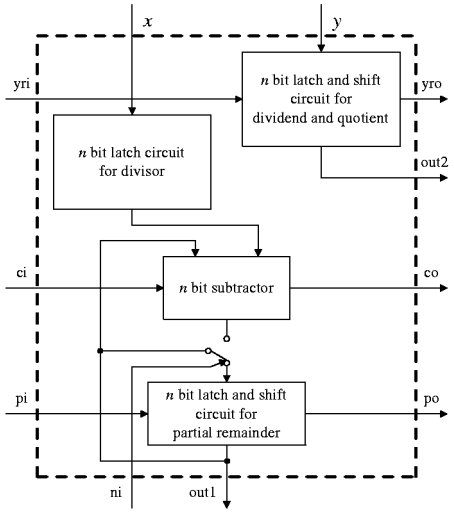

Fig. 5　Structure of Conventional Divider module.

みを変えることにする．これにより，これまで1モジュールの精度の2倍必要としていた部分積，被乗数用の各レジスタ，加算器のビット数を1モジュールの精度のビット数で実現することが可能となる．また，これまで使用していた部分積の半分のビットは被乗数用に使用しているシフトレジスタを共有することにより，従来型の乗算機能を実現することが可能となる．

また，除算時に用いる構成は，最適化を行った乗算時の回路を共有できるような構成にする．図8に示すように，除算では，被除数を1ビットずつ左シフトし，出力されたビットを部分剰余とする．このため，被除数はシフトレジスタを使用する．また，部分剰余も被除数からシフト出力された値が入力されるので，シフトレジスタを使用する．ここで，部分剰余は除数との減算が行われるので，

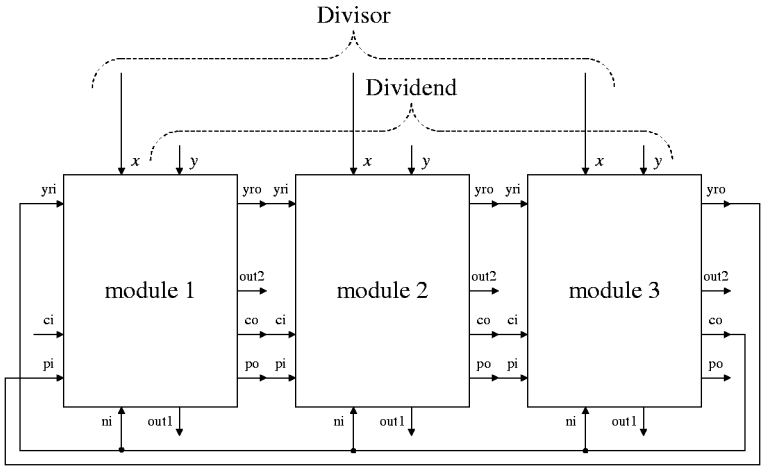

Fig. 6　Block diagram of Divider cascaded by three Divider module.

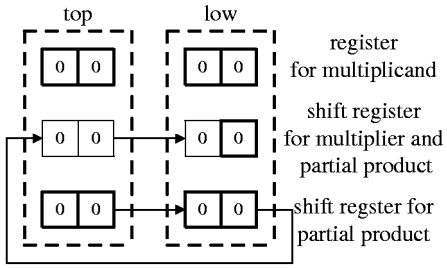

Fig. 7　Circuit behavior of Proposed Multiplier.

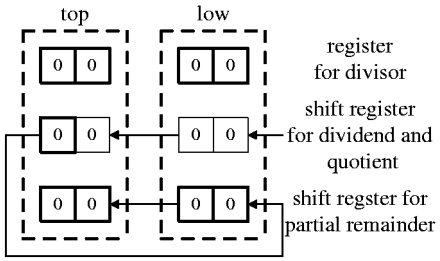

Fig. 8　Circuit behavior of Proposed Divider.

乗算時の部分積用レジスタを使用するのが最も最適である．そのため，被除数のラッチシフトに使用するシフトレジスタは乗算時の乗数用シフトレジスタを使用する．また，除数はラッチのみ行うので乗算時の被乗数用レジスタを使用する．商は部分剰余と除数の減算が行われる度に出力されるので，シフトしラッチする必要がある．そのため，被除数がシフト出力されたあとのシフトレジスタを再度利用することにより共有することができる．また，乗算時，除算時で扱う1モジュールのビット数は同じであるため，加算器を使用して減算を行うことが可能である．

このように，乗算器，除算器の回路構成を検討し，最適な方法でレジスタを共有化させた結果，図7，図8に示すように，乗算時，除算時で同じ構成にすることができる．

4.　拡張性に適した高速加減算器の構成法

拡張性を考える際，モジュール間のキャリーの伝播が問題となってくる．従来の乗算器，除算器は，複数個のモジュールを接続し，任意精度の乗算器，除算器を構成してきた．そのため，下位モジュールの加算，減算により出力されるキャリーが入力されないかぎり，次のモジュールでの加算，減算が行われない．その結果，モジュールを多数接続した際，処理時間が大幅に大きくなっていた．

そこで，我々は，最下位モジュールから最上位モジュールまでの加算時間を大幅に削減できる構成法を提案する．その構成を図9に示す．セレクタの処理時間は加算器の処理時間より十分小さい．また，加減算器は，演算精度を大きくするに伴い，処理時間が大きくなる．しかし，セレクタは，精度に関係なく処理時間は一定である．加減算の結果は，キャリーが0と1の場合の2パターンしか存在しない．これらの性質を利用し，あらかじめ，各モジュールで加減算器を2個用いて，キャリーが0と1の場合の演算を同時に行う．これにより，各モジュールの加減算は下位モジュールからのキャリーを待たずに処理することが可能となる．この結果を下位モジュールからのキャリーを用いて選択することにより，加減算器1モジュール分の処理時間とモジュール数分のセレクタの処理時間で行うこ

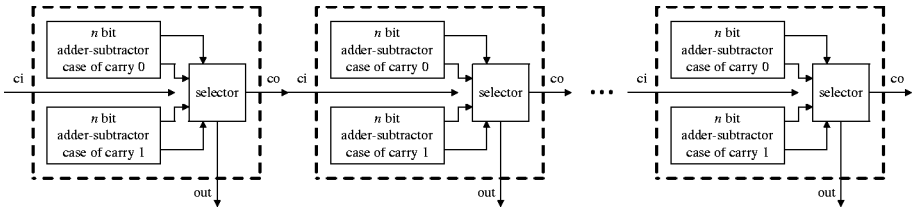

Fig. 9 Block diagram of carry selection adder-subtractor cascaded by three carry selection adder-subtractor.

とができる．

図9を用いて，回路動作を説明する．はじめに，各モジュールに被加数と加数，または被減数と減数が分割され入力される．同時に各モジュールでキャリーが0と1の場合の加算または減算が行われる．この結果を最下位モジュールのキャリー入力により順に選択していき，演算結果を出力する．

これにより，1モジュールの加減算器の処理時間をt_{as}，セレクタの処理時間をt_{sel}とすると，nモジュール接続した場合，従来のリップル型は下位のモジュールの演算が終了してから次のモジュールの処理が始まるので処理時間はnt_{as}となる．これに対し，本提案型ではすべての加減算処理が同時に行われ，その結果を下位モジュールのキャリーにより選択していくので処理時間は$t_{as} + nt_{sel}$となる．本提案型では，モジュール数nを大きくする場合，加減算の処理時間は増加したモジュール数分のセレクタの処理時間しか増加しない．そのため，モジュール数が大きくなるに伴い，処理時間の差は大きくなる．したがって，高精度演算を行う際，モジュールの精度を大きくし，モジュールを多数必要とするほど，本提案型は有効となる．

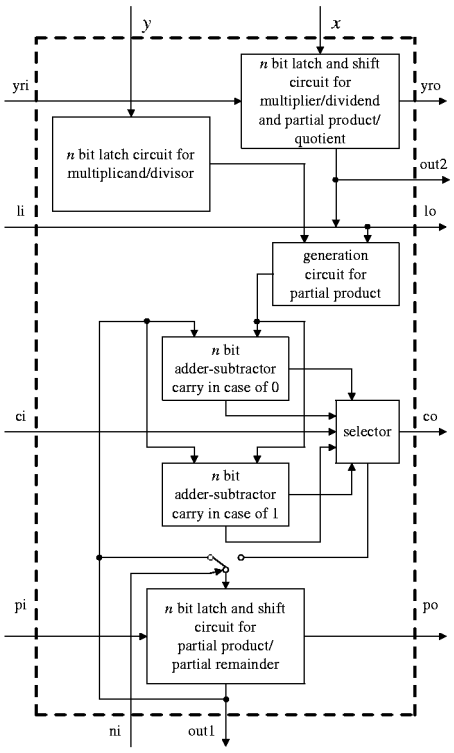

Fig. 10 Structure of Multiplier-Divider module

5. 拡張容易な任意精度乗除算器のシステム構成

本章では，任意精度乗除算器のシステム構成について述べ，拡張方法，回路動作を示す．

5.1 本提案型のモジュール構成

本モジュール回路は，モード信号により，乗算，除算の切り替えを行う．したがって，柔軟に乗算，除算に対処することができる．

本提案型は前に述べた回路の共有法と拡張性に適した加減算器の構成法に基づき，図10のようにnビットラッチ&シフト回路2個，ラッチ回路1個，加減算器2個，セレクタ1個，部分積生成回路1個で構成する．図10の端子について説明する．

- yri： 下位モジュールからのシフト入力
- yro： 上位モジュールへのシフト出力
- ci： 下位モジュールからの桁上げ(乗算時)／下位モジュールからの桁借り(除算時)
- co： 上位モジュールへの桁上げ(乗算時)／上位モジュールへの桁借り(除算時)
- pi： 下位モジュールからのシフト入力
- po： 上位モジュールへのシフト出力
- li： 乗数入力
- lo： 乗数出力
- ni： 減算抑止信号入力

このモジュールを，一次元的にカスケード接続す

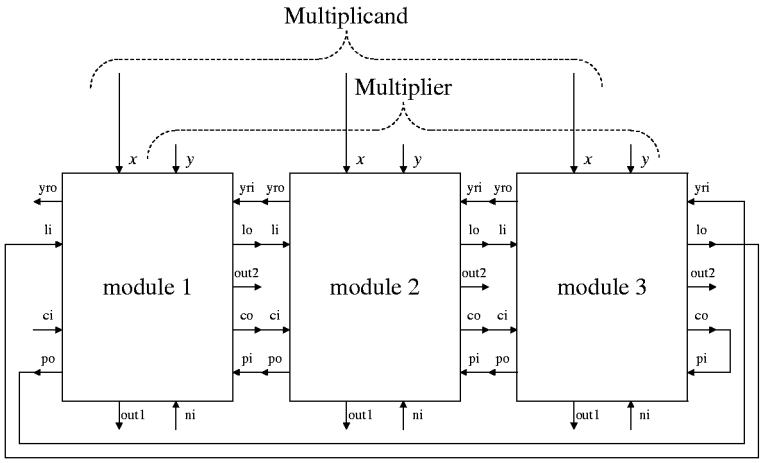

Fig. 11　Block diagram of multiplier cascaded by three Multiplier-Divider module

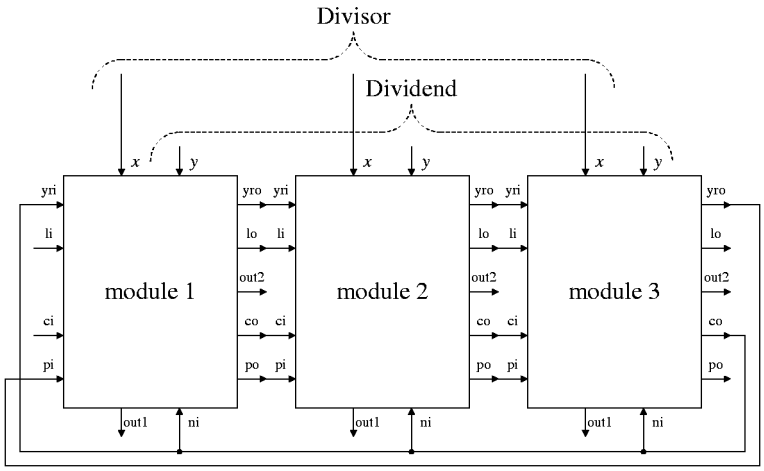

Fig. 12　Block diagram of divider cascaded by three Multiplier-Divider module

ることにより，容易に任意の精度に拡張することが可能となる．次に，このモジュールを用いて3倍精度に拡張した構成を示し，任意精度への拡張容易性を示す．

5.2　ビットスライス化による本提案型の拡張容易性

本節では，図10のモジュールを3倍精度に拡張した際の接続方法を述べる．3倍精度に拡張した乗算時および除算時の構成を図11，除算器を図12に示す．

乗算時の接続方法は，図11に示すように，モジュール1とモジュール2，モジュール2とモジュール3の間でそれぞれ対応した入力，出力端子を接続する．また，最上位モジュールと最下位モジュールの接続については，部分積の加算結果をシフトする際に最下位モジュールのpoから出力したビットは部分積の下位nビットとして最上位モジュールのyriに接続される．また，最上位モジュールのcoは部分積の最上位ビットとして最上位モジュールpiに接続される．

除算時の接続方法は，図12に示すように，モジュール1とモジュール2，モジュール2とモジュール3の間でそれぞれ対応した入力，出力端子を接続する．最上位モジュールと最下位モジュールの接続については，最上位モジュールのキャリーが商，減算抑止信号となるので，この出力端子coは最下位モジュールのyriと各モジュールのniに接続される．また，最上位モジュールの被除数が1ビットずつシフトされ部分剰余となるので，最上位モジュールのyroと最下位モジュールのpiが接続される．

このように，拡張する際外部回路を必要としないので，モジュールを複数個カスケード接続することにより，容易に任意精度の乗除算器を構成す

ることができる．

5.3 乗除算の回路動作

本節では，乗算時，除算時の回路動作を述べる．

＜乗算時の回路動作＞

モード信号が入力されると，乗数，被乗数はビット分割され図11で示した各モジュールのx，yに入力される．その後，乗数（被除数）ラッチ&シフト回路，被乗数（除数）ラッチ回路でラッチされる．次のクロックにおいて，最下位モジュールの乗数のLSBがloから各モジュールに出力され，被乗数との乗算により部分積が決定される．その値と部分積（部分剰余）ラッチ&シフト回路にラッチされていた値との加算が2個の加減算器によりすべてのモジュールで同時に行われる．発生したキャリーはcoから出力され上位モジュールのciに入力される．最上位モジュールから発生したキャリーは最上位モジュールのpiに入力される．そして加算結果は部分積（部分剰余）ラッチ&シフト回路で右シフトされラッチされる．シフトした結果，あふれ出たビットは，poから出力され，下位モジュールのpiに入力される．最下位モジュールのpoから出力された値は，最上位モジュールのyriに入力される．以降のクロックでは，このような回路動作を繰り返し行って，積を求める．

動作クロック数は，入力値のラッチと演算結果の出力でそれぞれ1クロック使用されるので，演算精度をnとすると$n+2$クロック必要となる．

＜除算時の回路動作＞

モード信号が入力されると，被除数，除数はビット分割され図12で示した各モジュールのx，yに入力される．被除数は1ビット左シフトし被除数（乗数）ラッチ&シフト回路に，除数は除数（被乗数）ラッチ回路に入力される．また，被除数のMSBは，yroから出力され，上位モジュールのyriに入力される．最上位モジュールのyroから出力されたビットは，最下位モジュールのpiに入力される．その後，部分剰余（部分積）ラッチ&シフト回路にラッチされ，仮の部分剰余となる．次のクロックで，ラッチされている部分剰余と除数との減算がすべてのモジュールの2個の加減算器で同時に減算される．減算結果が正ならば1，負ならば0が最上位モジュールのcoから出力され，各モジュールのniに入力される．入力値が1の場合は減算結果，0の場合は減算前の部分剰余が部分剰余（部分積）ラッチ&シフト回路に入力後，左シフトされラッチされる．また，coの値は商として最上位モジュールのyriに入力される．このクロックの回路動作が以降のクロックにおいて繰り返し行われ，商と剰余が算出される．

動作クロック数は，入力値のラッチと演算結果の出力でそれぞれ1クロック使用されるので，演算精度をnとすると$n+2$クロック必要となる．

6. VLSI評価

本章では，これまでに述べてきた構成法に対して，VLSI設計システムPARTHENONを用いて設計および評価を行う[7]．評価対象を消費電力，面積，ゲート数，マシンサイクル，演算時間とし，従来の任意精度乗算器と任意精度除算器，提案型の任意精度乗除算器で比較を行った．なお，今回用いる設計ルールは，$0.6\mu m$CMOSスタンダードセル（VLSIテクノロジ社）であり，電源電圧は5.0Vとする．

今回，比較対象として用いた演算器は，1モジュールの精度が4ビットの従来型の乗算器および除算器，本提案型の乗除算器とする[5][6]．また，今回はすべての演算器の加減算部分にリップルキャリー加算器を用い，各モジュールを4個カスケード接続して，16ビットの乗算，除算について比較評価を行った．これらの評価結果を表1に示す．

表1より，本提案型はマシンサイクル，演算時間においては，比較したすべての演算器の中で最も高速であることがわかる．これは，従来型において，下位モジュールのキャリーが入力されない限り，加減算を開始することができなかったが，本提案型においてはキャリーが0と1の場合についてあらかじめ計算し，これをキャリーにより選択していくので，1モジュール分の加減算時間と各モジュールの選択時間で処理できるためである．

また，ハードウェア量においては，乗算，除算両方の演算を必要とする場合，乗算器と除算器2つの演算器を用いるのに対し，本提案型を用いることにより，消費電力で14%，面積で25%，ゲート数で23%削減することができた．

また，今回は加減算器にリップルキャリー加減算器を用いたが，CLA加減算器などを用いることにより，さらに高速化を図ることができる．

7. むすび

本稿では，従来の任意精度乗算器および除算器をもとに，同一モジュールでの乗除算が可能な任意精度乗除算器のシステム構成を提案した．これ

Table 1　VLSI evaluation of each 16bit multiplier-divider.

	Power (mW)	Area (mm^2)	Gates (gates)	Machine cycle (ns)	Calculation time (ns)
Conventional Multiplier[5]	0.044	0.231	2013	37	666
Conventional Divider[6]	0.035	0.165	1427	33	594
Conventional Multiplier and Divider	0.079	0.396	3440	-	-
Proposed Multiplier-Divider	0.068	0.300	2634	29	522

を提案するため，従来型の任意精度乗算器および除算器の回路構成を考察し，同じ動作を行う回路を最適な組合せで共有化する方法を明らかにした．また，モジュール間のキャリーの伝播を考慮し，拡張性に適した高速加減算部分の構成法を明らかにした．さらに，本提案型の任意精度乗除算器，従来型の任意精度乗算器および除算器に対してVLSI設計システムPARTHENONを用いてVLSI設計および評価を行った．その結果，本提案型の任意精度乗除算器は，従来の乗算器および除算器に対し高速化が可能となった．さらに乗算器，除算器を別々に用意した場合に対し，消費電力で14%，面積で25%，ゲート数で23%削減できることを明らかにした．

このことから，RSA暗号方式などの乗算，除算を繰り返し必要とする分野において本提案型は極めて有効といえる．

参考文献

1) A. R. Omondi: Computer Arithmetic Systems/Algorithms, Architecture and Implementations, Prentice Hall (1994)

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

3) 山中喜義: 情報セキュリティと電子商取引，電気学会誌，**119**，No.6 (1997)

4) 中村次男，大石博朗，笠原宏: RSA公開鍵暗号システム実装におけるビットスライス化の一方式，電気学会論文誌，**118-C**, No.7/8, 1073/1081 (1998)

5) 中村次男，笠原宏: 拡張容易な乗算器モジュールとそのワンチップ化の提案，電気学会論文誌，**110-C**, No.2, 95/100 (1990)

6) 中村次男，笠原宏: 任意の精度に拡張容易な除算器の提案，電気学会論文誌，**111-C**, No.3, 123/128 (1991)

7) NTTデータ通信株式会社: PARTHENON User's Manual (1990).