任意の精度に被除数を拡張できる 並列除算器の提案

正員 中村 次男 (国際短期大)

正員 笠原 宏 (東京電機大)

A Proposal of Cascadable Parallel Divider on Dividend for Any Word Length

Tsugio Nakamura, Member, (Kokusai Junior College), Hiroshi Kasahara, Member (Tokyo Denki University)

The paper proposes a method of digital parallel divider circuit module and its one-chip integration. The multiplication and division are performed very frequently in computers, electronic control systems, signal processing for sound, and image processing. So far, we have proposed a new module of multiplier and divider, which is easily cascadable to perform arithmetic for any word length, without the use of external circuitry. As the multipliers are used more frequently, many fast multipliers with high precision have been proposed and integrated. But there are few one-chip dividers, because they are available only within the limited word length of dividend. So that, there are few merits of making one-chip integrated dividers.

The proposed parallel divider circuit module is cascadable on dividend for any word length, without the use of external circuitry. The soundness of the circuit was confirmed by computer simulation.

We can design a system without the restriction of the bit length of dividend by using this divider. The proposed method makes it possible to implement one-chip integrated dividers suitable for fast division for any word length.

キーワード：高速除算，並列除算器，拡張機能，カスケード，モジュール化，ワンチップ化

1. ま え が き

算術演算器は電子計算機をはじめとするいろいろなデジタルシステムの基本構成要素の一つであり，FA やロボットなどにおける電子制御システム⁽⁷⁾，また，音声や画像処理での信号処理演算などにおいて，特に乗除算が頻繁に行われる。

以上のようなシステムにおいてはファジー性が概念的に興味ある問題として挙げられる。例えば，ロボットの制御において，ある位置の物体に接近するまでは粗いベクトル演算で高速に移動し，その後，高精度でゆっくりと目的とする物体に接触するような制御に相当する。このような演算精度の可変長を要求するシ

テムに対し，著者らは演算モジュールを単にカスケード接続するだけで任意の精度に拡張容易な乗除算器の提案を行ってきた^{(1)~(6)}。

乗除算のうち最も使用頻度が高い乗算に関しては，高速な並列乗算器やこれらを内蔵した DSP (Digital Signal Processor) が適しており，開発が進んでいる。しかし，除算は被除数から除数が何回引けるかという演算操作から考えて，除算器の除算能力が不足した場合，除算データの分散処理が困難である。従って，1 ビットでも除算能力を超えた使用に対しては，たとえメモリーや外部回路を用いても容易には対処できない。このような理由によって専用 IC 化のメリットがなくワンチップ化された除算器は非常に少ない

(拡張機能をもった除算器は見あたらない)。DSP にも除算器を内蔵したものではなく⁽⁴⁾、ソフトウェアで対処しているのが現状である。

このような除算器に関する問題点に対し、著者らは先の乗算器と同様に、カスケード接続することによって被除数と除数を任意の精度に拡張可能な回路方式も考案した⁽⁴⁾⁽⁶⁾。その除算器は任意の精度に拡張可能であるが、 n ビット $\div$ n ビットの除算には n クロックを要するものであり、高速化に対しては必ずしも満足するものではない。高速に関しては並列形に勝るものはないがクロックを用いないだけにその実現は更に困難なものとなる。

本論文は、高速な並列除算器に対しても、並列減算器の改良と剰余のほかモジュールへの有効な伝達方法を考案して、その問題点を解決したものである。本除算器の拡張に際しては、これまでの一貫した設計思想に基づき、外部回路を用いることなく、単に除算器モジュールをカスケードに接続するだけである。本除算器によりワンチップ化のメリットが生じ、任意の高速除算が容易に可能となる。

2. 並列除算器の現状

除算は乗算ほどではないが、いろいろなアルゴリズムが考案されている⁽⁹⁾⁻⁽¹¹⁾。また、除算は乗算ほど頻繁に用いられるわけではない。しかし、信号の正規化、コンピュータ通信や情報ネットワークにおける秘匿技術⁽¹²⁾、画像処理や音声認識における特徴抽出の線形予測計算の演算⁽¹³⁾およびフォールトレラントシステムにおいて⁽¹¹⁾など、応用例が多数報告されている。しかしながら、これら既存の除算はその除算器(あるいはモジュール)の除算可能けた数の範囲内で有効であり、1 ビットでも除算能力を超えた使用は、たとえメモリーや外部回路を用意しても不可能である。高速な並列除算器に拡張機能をもたせることは更に困難であり、高速な演算を重視する場合には最大の除算能力を決めゲートアレイなどでワンチップ化して、その除算可能けた数以内で使用するなどの方法をとっているのが現状である。これに対し、順序回路方式ではあるが任意に拡張可能な除算器を著者らは先に提案した⁽⁴⁾⁽⁶⁾。

拡張機能はもっていないが、現在ワンチップ化されている除算器に MB 86040(10 ビット $\div$ 8 ビット、+5 V 単電源、40 ピン DIP; 富士通)⁽¹⁴⁾がある。MB 86040 は高速画像信号処理用に開発されたパイプライン方式高速除算器である。除算データは正の固定小数点で、被除数 10 ビット、除数 8 ビットの除算を 2 段のパイ

プライン処理により、除算結果の商 10 ビットを最高 10 MHz で出力する。ただし、剰余は出力されず切捨てられる。従って、数チップに分散する拡張した使用は不可能であり、除算能力を超えた場合にはソフトウェアで対処する以外にない。

3. 本除算器の回路方式

高速な並列除算器に拡張機能をもたせることは前述したようにその実現は容易ではない。そこで、大は小を兼ねる的な考え方で最大能力の除算器を選定すれば当然コストアップにつながる⁽³⁾。もし、複数の演算器を使用することにより、容易に演算能力を任意の精度に拡張可能であればその有効性は大きいと考えられる。その場合でも大きな演算器モジュールは高価になるだけではなく、ゲートの入力数(ファンイン)やピン/ネット(ファンアウト)が増大し、必ずしも性能が向上するとは言いがたく、比較的小さい精度にしたほうが、安価で効率的な拡張が実現できる。そこで、まず拡張機能設計上の問題点を考えてみる。

乗算器のように、数チップに除算データを分割し、それぞれの結果から拡張した除算の解を得るのは困難である⁽⁵⁾⁽⁶⁾。ここでは除数を固定にし、被除数を任意の精度に拡張可能な方法を考えてみる。例えば、簡単なために 1 モジュールの除算能力を 4 ビット $\div$ 4 ビットとし、図 1 に示すように、被除数の 12 ビットをそ

図 1 除算の筆算過程

Fig. 1. Manual calculation steps for division.

れぞれ4ビットずつ三つのモジュールに分割した場合、3モジュールの被除数を接続して、除数を1ビット右に順次ずらした状態で減算を実行するような回路になっていなければならない。図に示すように減算が他モジュール間に分かれて行われるため、けた数が増えるとピン数（モジュール化した場合は接続線数）が増大し、容易ではない。

図1では除算データを数モジュールに分割した場合に生じる除算固有の問題点を説明するために、簡単に4ビット除算器で示したが、現実には4ビットでは精度が粗く、拡張容易であっても多くのモジュールを必要とするため実用的ではない。実用性を考慮して、1モジュールは8ビットの回路構成とした。つまり、1モジュールの除算能力は16ビット÷8ビットで商と余りがそれぞれ8ビットである。拡張機能をもたせるため丸め操作は行わない。

除算操作は被除数の上位8ビットから順次除数の8ビットとの減算を繰返すため、並列減算器8個を1ビット右に順次ずらした状態で配置してある。そのため

(a) 1ビットの減算ユニット

減算は1ビット右にずれた状態で行われるため、被減数は9ビットの並列減算器にしてある。拡張機能を考慮しない従来の除算器においては被除数の上位ビットを0、例えば15ビット÷8ビット除算器とすることによって、並列減算は一般的には8ビットで行っている。しかし、図1に示したような拡張機能を考慮した場合、そのような方法では、数モジュールに被除数を分割したときに0がその被除数の間に挿入されてしまい、減算とその剰余の伝搬が適切に行われない。この問題は除数 n ビットに対して $n+1$ ビットの並列減算器を用いることによって解決できる。更に減算結果を出力するか、または引けなかった場合、前の結果を出力するかを切換え回路を付加してある（図2）。

そして、図2の並列減算器を1ビット右にずらして図3のように構成したのが、本除算器の1モジュールである。1モジュールの除数は8ビットであるため、16ビット÷8ビットの除算結果の商は8ビット以内でなければならない。また拡張した使用に際しては被除数の上位8ビットには部分剰余が与えられる。そのため、1モジュールの除算能力は8ビット÷8ビットになる。3モジュールを用いて24ビット÷8ビットに拡張した動作を論理的にシミュレーションし、除算結果の商24ビットと余り8ビットに満足するものが得られた。

遅延によるグリッジなどの影響については、本方式はクロックを用いる順序回路方式とは異なり並列形であることからその影響は少ない。除算データの切換え時にはデータの変化が直ちに出力に影響し、グリッジを生じることもある。それは、演算中の出力変化であり回路の誤動作の原因となるようなグリッジとは異なるものである。除算演算の論理シミュレーションでは

(b) $n+1$ ビット並列減算器構成

図2 改良形並列減算器

Fig. 2. Improved parallel subtractor.

図 3 拡張機能をもった本除算器モジュール
Fig. 3. Cascadable divider module.

図 4 3 モジュールに拡張した使用例

Fig. 4. Expanded application of three divider modules.

正しい結果が得られ、不都合のないことを確認した。

このような前段の部分剰余を後段の被除数の上位 8 ビットに順次接続した構成により、拡張した際に生じる適切な部分剰余の伝搬の問題を解決した。

3 モジュールに拡張した構成図を図 4 に示す。

4. モジュール化とワンチップ化の考察

拡張機能をもった本並列除算器は直列除算に比べ並列除算器であるため制御回路が不要であり、高精度でかつ高速除算が可能である。しかも使用方法は極めて簡単である。1 モジュール (チップ) の除算能力が 16 ビット ÷ 8 ビットの端子数は 42 (電源 2 ピンを含む) で、使用ゲート数は 856 である。伝搬遅延時間は NAND, NOR, NOT の基本ゲートの伝搬遅延を 1Δ (1Δ はゲートの単位遅延時間) として最長伝搬経路を算出する⁽⁹⁾と、 $2n$ ビット ÷ n ビットでは $3n(n+2)\Delta$ になる。参考までに $n=8, 16$ および 32 としたときのゲート数と伝搬遅延時間および端子数の関係を表 1 に示す。 $n=16$ と 32 ではワンチップ化する場合、端子数が多くなるが、82 ピンは被除数と除算結果を多重化することにより、標準パッケージの 64 ピンに納めることも可能である。 $n=32$ の場合も同様な多重化

表 1 1 モジュールの除算能力に対する比較
Table 1. Comparison of one module capability.

	除 算 能 力		
	16ビット ÷ 8 ビット	32ビット ÷ 16ビット	64ビット ÷ 32ビット
使用ゲート数	856	3,248	12,640
伝搬遅延時間	240 (Δ)	864 (Δ)	3,264 (Δ)
端 子 数	42	82	162

により、またはフラットタイプや PGA タイプのパッケージにより、収納は可能である。しかし、除算専用器であることを考慮し、実用的には集積度、端子数および 1 チップあたりの除算効率から $n=8$ が最適と考えられる。

本除算器は、先に考案した任意の精度に拡張可能な順序回路方式による乗除算器^{(1)~(6)}と同様に、ワンチップ化または VLSI のマクロセルとしての利用が考えられる。除数が固定という特徴から基数変換の演算にも適しているといえる。

5. ま と め

本除算器は「符号なし数と符号絶対値表示数」の演算として考案したものである。演算結果の符号は除数

と被除数の符号の排他的論理和をとって容易に得ることができる。2の補数表現の場合は通常、「符号補数表示数の間接形演算」によって対処する⁽¹⁰⁾。2の補数、符号なし、混成表示可能な機能をもたせるには数値表現設定信号を入力し、内部で補数器の切換えを行うことによって容易に解決できる。このような補数器を内蔵することは容易であり、演算部分はいずれにおいても上記の「符号なし数と符号絶対値表示数」の演算として扱い得るものである。

除算器はその除算能力以内でのみ使用可能であるという現状に対し、本除算器によってその制約から解放され、ワンチップ化のメリットが生じたことの意義が大きいといえる。除算器の専用ICとDSPなどのマイクロセルとして普及していくものと確信する。

(平成2年7月25日受付, 同3年2月19日再受付)

文 献

- (1) 中村・笠原:「無限の拡張機能を持った乗算器の開発とそのワンチップ化」, 昭63電気関係学会東北支部連大, 1D2
- (2) 佐野・三村・笠原・中村:「乗算桁数と並列処理数をダイナミックに変えられる補助乗算システム」, 平元電気関係学会中国支部連大, 122516
- (3) 中村・笠原:「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」, 電学論C, 110, 95 (平2-2)
- (4) 中村・笠原:「拡張容易な除算器モジュールとそのワンチップ化の提案」, 平元電気関係学会中国支部連大, 122515
- (5) 中村・笠原:「高速に任意の可変精度で演算が可能な並列除算器の提案」, 平2電子情報通信学会CAS, 89-138
- (6) 中村・笠原:「任意の精度に拡張容易な除算器の提案」, 電学論C, 111, 123 (平3-3)
- (7) 春田・亀山・樋口:「ロバストフォールトトレラント乗算器の構成」, 信学論D, J73-D-I, 141 (平-2)
- (8) 持田:「DSPの現状と動向」, 情報処理, 30, No. 11 (昭63-8)
- (9) 久津輪:「並列乗除算配列におけるCLA回路の効果」, 信学

論D, J64-D, 797 (昭56-8)

- (10) HWANG: コンピュータの高速演算方式 (昭55) 近代科学社
- (11) 高木・矢島:「オンライン誤り検出可能な高速配列型除算器」, 信学論D-I, J73-D-I, 148 (平-2)
- (12) 岡本・中村:「DSPの秘匿への応用」, 情報処理, 30, No. 11 (昭63-8)
- (13) 児玉・市川:「DSPの音声処理への応用」, 同上, 30, No. 11 (昭63-8)
- (14) 10ビット÷8ビット除算器MB 86040 (昭63-4) 富士通

中 村 次 男 (正員)

平成2年東京電機大学大学院修士課程修了。昭和46年日本電気精器(株)入社。現在、国際短期大学専任。ディジタル演算器の研究に従事。電子情報通信学会会員。

笠 原 宏 (正員)

昭和15年生。45年東京電機大学博士課程満期退学。同大学助手、講師、助教授を経て現在、工学部電気工学科教授。工学博士。パワーエレクトロニクス、計算機システム、制御用分散処理システム、オブジェクト指向システム設計法の研究に従事。情報処理学会、ロボット学会、IEEE、ACM会員。