社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2009-SLDM-138 (12)
2009 / 1 / 29

任意精度計算機アーキテクチャの提案

橋本 将平† 戸塚 雄太† 牧野 政道† 安田 光†
中村 次男† 冬爪 成人† 笠原 宏†

†東京電機大学 情報環境学部 〒270-1382 千葉県印西市武西学園台 2-1200

E-mail: †{den04066, den04055, den04081, den04091}@nifty.com,
{nakamura, fuyu, kasahara}@itl.sie.dendai.ac.jp

あらまし 演算処理データ長の制限を受けない任意精度計算機のアーキテクチャを提案する．周波数の異なるクロックによる独自の計算方式を開発し，かつ，多バス方式を採用して処理の効率化と高速化を図る．そのためワード長の異なる ALU を 2 つ実装するなどのアーキテクチャを採用し，FPGA で試作して，正常に動作することを確認した．

キーワード 任意精度，多バス方式，計算機アーキテクチャ，可変長演算

A Proposal of the Computer Architecture for Numbers of Arbitrary Word Length

Shohei HASHIMOTO† Yuta TOTSUKA† Masamichi MAKINO† Hikaru YASUDA†
Tsugio NAKAMURA† Narito FUYUTSUME† and Hiroshi KASAHARA†

† Department of Information Environment, School of Information Environment, Tokyo Denki University
Muzai Gtakuendai2-1200, Inzai, Chiba, 270-1382

E-mail: †{den04066, den04055, den04081, den04091}@nifty.com,
{nakamura, fuyu, nakamura}@itl.sie.dendai.ac.jp

Abstract We propose a computer architecture for numbers of arbitrary word length with unlimited processing data length. In this design, we developed some original calculation methods using the clocks with different frequencies, and the multi-bus system for efficient processing and speedup. Two ALUs with different word length are also for the same purpose. We carried out experiments on a prototype system on FPGA and confirmed the proper behavior.

Keyword numbers of arbitrary word length, multi-bus system, computer architecture, scalable operation

1. まえがき

近年，半導体の微細加工技術や半導体材料の開発技術の発展によりプロセッサの性能が向上し，計算機システムの演算処理能力が飛躍的に進歩してきた．それに伴い，仕様の変更が容易であることやコスト面などから，ハードウェアよりもソフトウェアを主体として実装することが顕著になった[1]．しかしソフトウェアによって膨大なデータ量の計算をする場合，その都度演算データを全てソフト側に渡さなければならないため，演算誤差の累積や処理時間の増大を招く可能性がある．これに対してこのような計算をハードウェアでサポートすることにより，安定かつ高性能な処理を実現することができる．

本稿では，命令をワード単位で拡張する可変長命令にすることにより任意精度対応とし，3 バス方式による並列処理によって高速な演算が可能になる計算機の機構設計とアーキテクチャを提案する．計算機システムは現在まで，用途によりさまざまなアーキテクチャが考案されてきた．その中でも計算機の動作制御の基本である単一バスで動作させるシステムのメリットとして，データ制御やハードウェア化が容易でデッドロックが生じないという点が上げられる．しかし，データ流量に制限があるためにシステムの待ち時間が長くなり，乗算・除算など計算量が膨大になった際に演算に時間がかかることや，各機構が多岐に渡り複雑に構

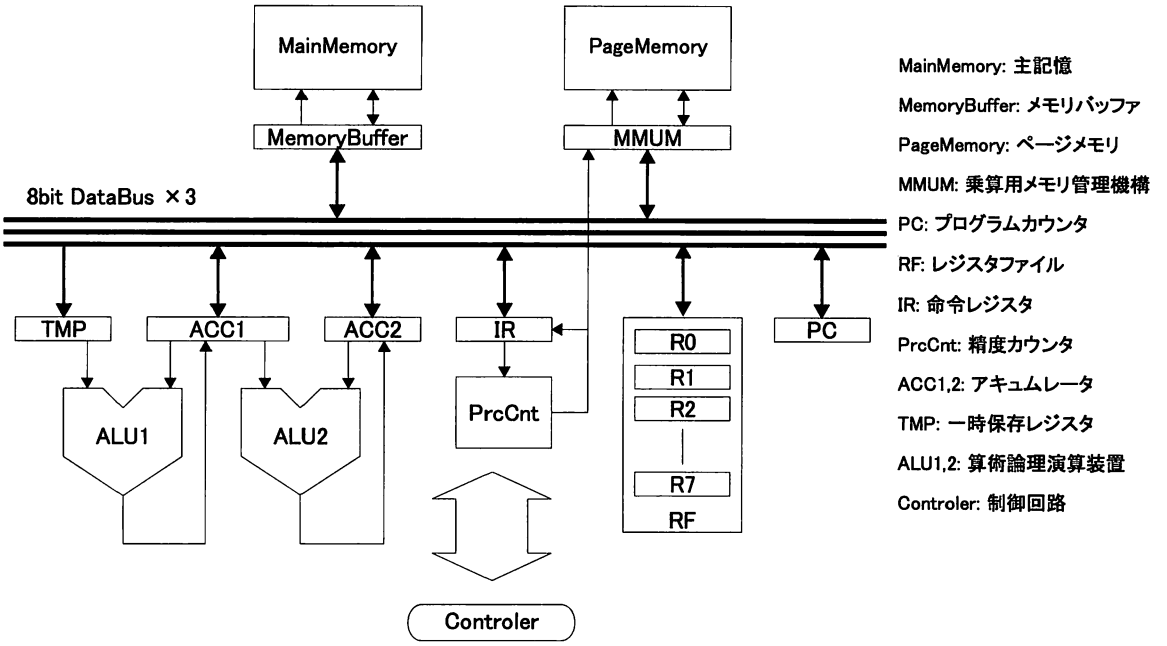

図 1　計算機のブロック図

Fig.1. Block diagram of computer architecture

成された場合，かえって制御が複雑化する可能性がある[2]．また，ビット幅を固定し命令を受け渡す固定長命令方式では，ワード数に制限があり演算量が多くなる処理には向かない．そこで，可変長命令にすることでワード数を理論上無限に拡張することができ，これにより任意精度演算を可能とし，演算ワード長の制限を受けることのない一度の処理で多桁の演算が行える計算機システムを提案する．今回の計算機システムにおいては，試作として比較的容易に開発が行えるよう，1 ワードを 8bit として設計した．

さらに 3 バス方式を用いることにより，複数の処理を並列で動作させ任意精度演算の処理効率を大幅に上げることが可能となり，出力ワード数の異なる ALU を 2 基搭載することでシステム全体の処理速度をより向上させる効果を図った．

これらのアーキテクチャを実装した FPGA(Field Programmable Gate Array)による試作結果について報告する．

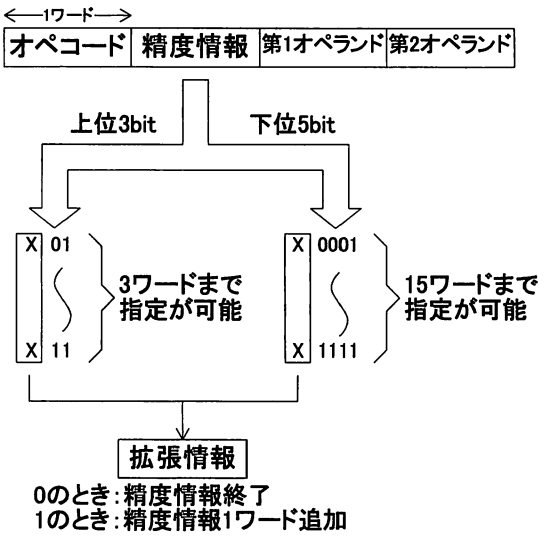

図 2　任意精度演算用命令フォーマット

Fig.2. Instruction format for operation of numbers of arbitrary word length

2. 提案する計算機アーキテクチャ

2.1.任意精度演算の実現

一般的な CPU では，単精度および倍精度の精度モードだけしかサポートしていないものが多い．そのため，倍精度以上の演算はソフトウェアで処理されることとなり，演算誤差の累積や処理時間の増大が問題となる．そこで，演算データ長に制限を受けない演算処理装置をハードウェアとして実装することにより，演算処理の高速化を図った．

提案する計算機アーキテクチャは，基本的な構成に加え，独自の機構を搭載するものとした(図 1).

任意精度演算には，精度情報を含んだフォーマットを用いる．オペランドは被演算数および演算数の最下位ワードが格納されているメモリアドレスを指定する．オペランドの先頭には精度情報を付加し，1 ワード(8bit)のうち，上位 3bit をメモリアドレスのワード数および下位 5bit を演算数のワード数の指定に用いる．図 2 のような拡張機能を用いれば，理論上では無限大の精度指定が可能となる．しかし，実際は無限大のメモリを用意することは不可能なため，十分大きな 16bit アドレス(65535 番地)まで対応させる設計とした．

2.2. 演算データ長の異なる 2 基の ALU

演算処理の効率化を図るため，以下のような ALU とアキュムレータの接続を実装する(図 3).

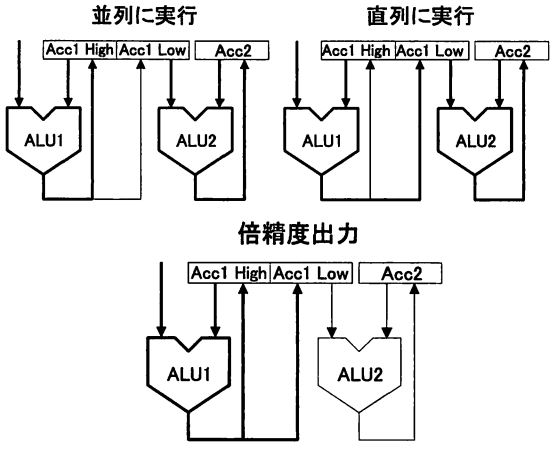

図 3　ALU と ACC の接続

Fig.3. Connection of ALU and Accumulator

このように ALU とアキュムレータの接続を行うことで，以下の動作が可能となる．

2.2.1. 通常の ALU のように，2 つの演算を並列に実行(図 3 左上)

任意精度の演算においては，次に演算するアドレスの計算や実際の演算処理(四則演算など)が頻繁に行われるため，ALU を 2 基搭載してこれらを同時実行させることで，より速い命令処理が可能となる．

2.2.2. 並列実行が不可能な演算を，直列に実行(図 3 右上)

演算順序の関係で並列演算が不可能であり，次の演算にバスを介す必要がなく，実行に 2 サイクルかかるような演算のペアが，1 サイクルで実行可能になる．

2.2.3. ALU1 の乗算時，積を倍精度で出力(図 3 下)

ACC1 を Hgih と Low に分けて ALU1 と接続することで，乗算解を ACC1 に対して倍精度出力することが可能となる．倍精度対応のレジスタを予め用意してあるため，乗算結果をすべて求めるための演算回数の削減ができ，バスへのデータ同時出力など乗算結果のデータ管理が簡易化できる．

2.2.1.と 2.2.2.に関しては，クロック周波数を下げることにより 1 サイクルでの命令実行が可能な Cascade ALU とした研究報告[3]があるが，2.2.3 に関しては，アキュムレータとの特殊な接続による倍精度出力という大きな特徴を有する．

また ALU1 には，キャリーフラグの蓄積機能付き加算を実装した．任意精度の乗算に用いる想定で実装した特殊な加算命令であり，この命令によって発生したキャリーは演算ごとに次々と蓄積されていく．これは部分積を連続で加算していく際の桁上げ対策であり，結果は別の命令によってアキュムレータに出力される．このフラグ蓄積量は演算精度に依存するものであるが，ここでは 8bit のレジスタを用意し，十分に大きな精度の乗算を行えるものとした．

2.3. 精度情報の一括管理

任意精度演算では回路内での同期した精度情報の認識が重要である．各機構に別々で精度情報を認識させると，同一時間軸において処理しているアドレスやデータなどにずれが生じ，誤動作に繋がる恐れがある．そこで，任意精度演算における精度情報を一括管理する機構として，精度カウンタという機構を実装した．このカウンタ以外の機構は精度情報を知ることができず，精度情報の管理をひとつの機構にまとめることで各機構間での精度認識のずれをなくす効果を図った．

精度カウンタは，精度情報を受け取りカウントアップ動作を行う．設定したカウント値までカウントすると，カウント終了フラグを出力する．任意精度乗算のアドレス計算の際にも使用することを想定し，同一仕様の内部カウンタを 2 つ実装した．アップカウンタにすることにより，同じカウント値で繰り返しカウンタとして利用することができる．

2.4. 乗算専用のメモリ管理ユニット

任意精度乗算では，加算や減算と比べ，演算回数や管理するデータ量が膨大である．特に，今回提案する筆算形式の演算方法では，部分積の保持とレベルごとの部分積の加算が必要となる．部分積の加算については前述したが，その部分積の保存領域と加算レベルの決定方法が問題となる．そこで，任意精度の乗算時に

動作する特殊な機構として，乗算用メモリ管理機構(Memory Management Unit for Multiplication:MMUM)とページメモリを配置した(図 4).

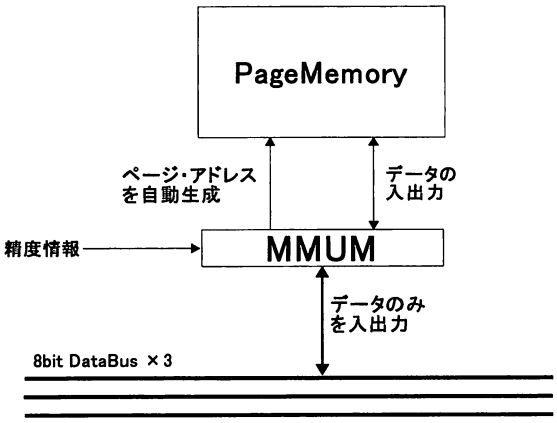

図 4　乗算用メモリ管理機構のブロック図
Fig.4. Block diagram of Memory Management Unit for Multiplication

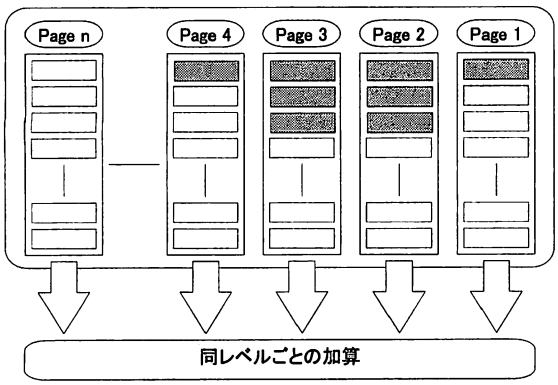

図 5　ページとアドレスの割り当て
Fig.5. Allocation of page and address

MMUM の基本的な構成はメモリバッファと同様である．大きく異なる点は，リード・ライトともに該当するページとアドレスを自動生成することである．部分積の加算を行うために，そのレベル分けをページという単位で分割する(図 5). バスから部分積を入力し値を書き込む際は，書き込みたいデータをメモリへ出力するのと同時に，この機構自身が書き込むべきページとアドレスを判断し，送られてきたデータとともにメモリに出力する．連続加算の際には，決められた順序(ページ 1 から順に)でページとアドレスを計算し，ワード単位でバスへデータを出力する．ページやアドレスの計算には，精度カウンタのフラグ情報が用いられる．制御回路からページ・アドレス管理の制御を切り離すことで，より容易な回路制御が可能となった．

2.5. 任意精度演算方式

演算データ長に制限を受けない四則演算を実現するため，各々の演算について筆算と同様の方式を提案した．加減算・乗算については以下に示す方法を用いて任意精度演算対応とした．

除算については計算パターンの多岐化問題があり現在検討中であるが，制限付きの除算方法について提案を行い，以下のような条件における除算方法を実装した．

・被除数，除数ともに同ワード長であること
・除数の最上位ワードが 0 でないこと
・商および余が 1 ワードであること

これらの制限を設けることにより，演算途中で用いる仮定した商と余が 1 ワードの固定となるため演算パターンが容易に組める．さらに一時的に使用する格納領域を限定でき，制約はあるが任意精度の除算が可能となる．現在，更なる機能向上のため改良中である．

2.5.1. 加算(減算)

被加数と加数それぞれの最下位ワード同士を加算し，その結果(キャリーフラグ)を踏まえ次のワード同士を加算する．精度情報で指定された分だけワード単位での加算を繰り返す(図 6).

減算に関しても，サインフラグを考慮して同様の処理を繰り返すだけである．

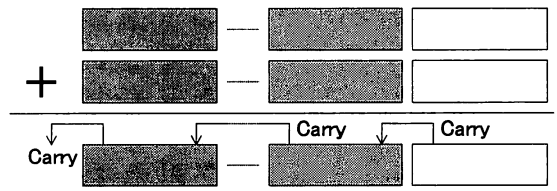

図 6　任意精度加算方法
Fig.6. Method of addition of numbers of arbitrary word length

2.5.2. 乗算

前述した ALU や乗算専用の MMUM を用いて，筆算と同形式の方法で行う．最下位ワード同士の乗算を行い，その部分積を MMUM を介しページメモリへ渡す(図 5). 同様の処理を繰り返し，すべての組み合わせの乗算を終えると，MMUM にて振り分けたレベルごとに連続的に加算を行う(図 7). なお，アドレス計算や部分積の連続加算には，精度カウンタに用意した 2 つのカウンタを利用する．

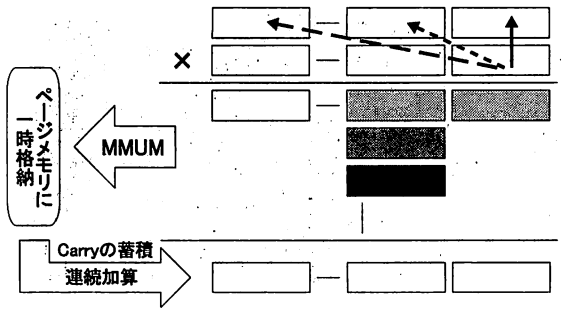

図 7　任意精度乗算方法

Fig.7. Method of multiplication of numbers of arbitrary word length

2.6. 多バス方式による並列処理

計算機の回路構成の基本とされる単一バス方式では，処理データの取り扱いが簡易化できる一方で，動作速度の低下や構成する機構の数によってはかえって制御が複雑になることがある．そこで，システム全体の処理効率と回路構成のバランスを考慮し，3 本のデータバスを配置した計算機回路を提案する．ALU を 2 つ搭載することによる演算数などのデータや，メモリにおけるアドレスデータ管理の複雑化の回避，およびシステム全体の動作速度の向上を目的とした．

3 本のデータバスの制御には，各機構に接続したバス設定線を利用する．3 本のバスに 1bit ずつ割り当てた 3bit のバス設定線によりバスの有効･無効化を操作することが可能となる．1 本のバスにおけるデータの衝突を避けるため，制御回路によりクロック同期でバス設定線と入出力命令を制御し，命令処理を実行する．

3 本のデータバスを利用し，フェッチ･デコード処理と実行処理を分離して段階的に命令データを処理する(図 8)．単一バス方式のデータ処理は単純な直列であったが，3 バス方式では複数のデータを同時に処理することで並列的になり比較的高速な処理が可能となる．

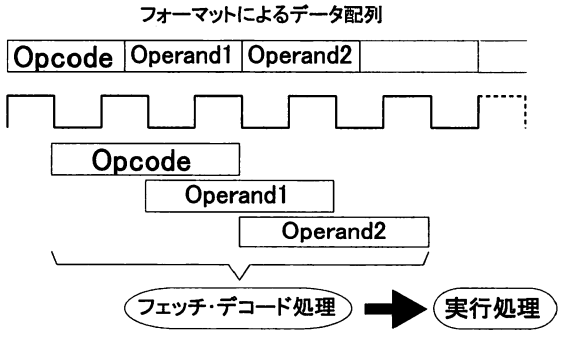

図 8　命令データの並列処理

Fig.8. Parallel processing of instruction data

3. シミュレーションと試作結果

提案するアーキテクチャを実装した上で，3 バス方式での命令実行や任意精度演算が可能であることを，シミュレーションと FPGA による試作によって確認した．Xilinx 社の FPGA Spartan2，XC2S150 をターゲットデバイスとして動作確認を行った．設計ツールに Xilinx 社 ISE Foundation 9.1i を，シミュレータには Mentor Graphics 社 ModelSim XE Ⅲ 6.3c を使用した．シミュレーション結果(データバスの様子)について図 9 と図 10 に示す．

図 9 は，3 バス方式計算機の Load Memory の命令実行結果の一部を示している．PC からロードされたアドレスからメモリバッファを介しオペコードを読み取り、IR を通して命令のデコードが行われる。次にデコードされた命令フォーマットに従い，それぞれのオペランドが次々に読み出される．ここまでのフェッチ･デコード動作において，3 本のバスを利用して並列なデータ処理が行われていることがわかる．オペランドとオペコードが揃い次第，命令の実行処理へ移行する．この実行処理においても，アドレスや演算数値などデータの種類によってバスを使い分けて同時に複数のデータ処理を行い，命令処理時間の短縮が実現されている．

図 9 と同等の命令処理を単一バス方式の計算機で実行し，3 バス方式との命令処理時間を比較すると，単一バス方式ではおよそ 16 クロック，3 バス方式ではおよそ 7 クロックであり，半分程度の時間に短縮できていることがわかった．

図 10 は，任意精度乗算結果の一部を示したものである．任意精度の演算命令においては，初期化処理として演算数が格納されているアドレスや解の格納アドレスを，予めレジスタファイルの規定の場所に保持する．任意精度の乗算命令では，オペコードと精度情報を読み出した後，各オペランドを読み出すと共にこの初期化処理を行い乗算までの準備をする．初期化処理が完了すると，2.5.2.で示した方式で演算を行う．

図 10 の左側は，1 ワードごとの乗算結果を倍精度で出力し，MMUM を介してページメモリへ書き込む動作である．図 10 の右側では，ページメモリへ格納した部分積データをページごとに順次読み出し，連続的に加算を行っている．このとき生成された ALU1 による桁上げフラグの蓄積結果は，次のページの加算に反映される．ページメモリとのデータの入出力において，ページとアドレスの自動生成が正常になされていることも確認できた．

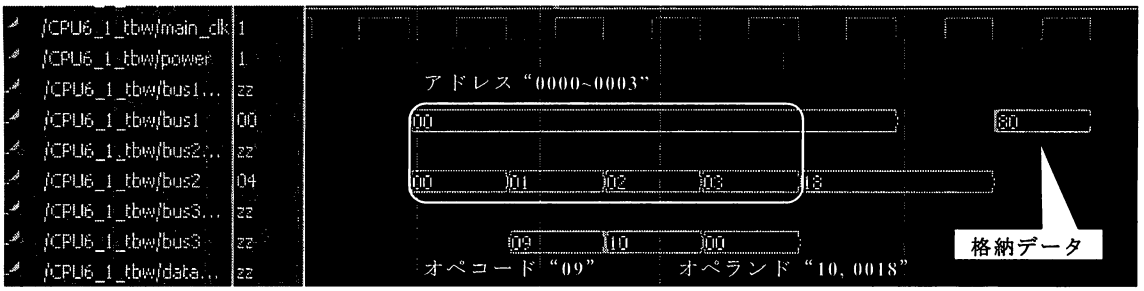

図 9　Load Memory 命令実行結果(3 バス方式)

Fig.9. Result of Load Memory(3bus system)

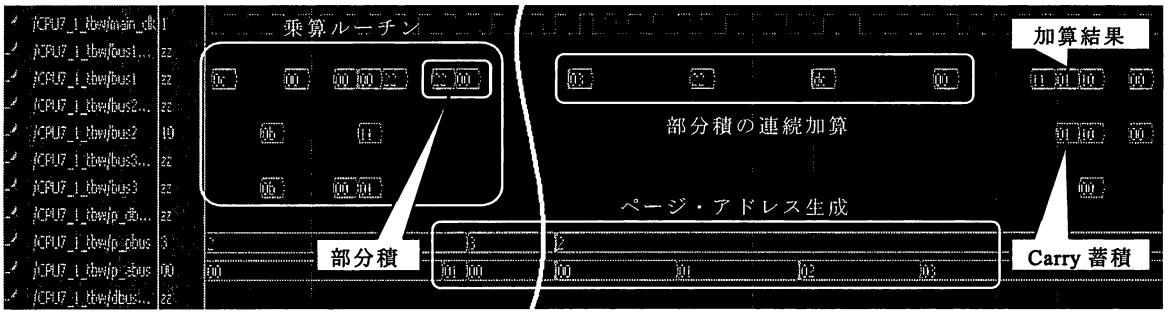

図 10 任意精度乗算結果(一部)

Fig.10. Result of multiplication of number of arbitrary word length

4. まとめ

本稿では，演算ワード数の拡張可能な任意精度計算機アーキテクチャを提案し，任意精度対応演算の実装と各機構の特徴を述べ，シミュレーションおよび FPGA による試作結果を報告した．試作した結果，比較的高速な任意精度演算をハードウェアにおいて実現できることが確認できた．

今後の課題は，任意精度除算方法の更なる改良である．除算では，加減算や乗算と異なり演算数を全て認識できないと解を求めることは困難であり，さらに演算数により計算パターンが複雑化してしまうため，現在，任意精度除算法の改良を行っている．演算数の上位部どうしを除算し商を仮定した後，乗算と減算の組み合わせで計算する方式であるが，これを実装するためには，除算の途中計算で必要となる余りの格納領域の確保や，商を仮定する機能を ALU に追加するなど各機構の再検討が必要となる．

提案した任意精度計算機の研究では，動作確認を目的に試作を行い，FPGA において正しく動作することを確認した．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おり，特にハードウェア量や拡張性が問題となる乗算器や除算器に関しては，これまで様々な演算回路が考えられてきた[4][5]．本稿で提案した拡張可能な任意精度計算機は，このような高精度な演算処理を求められる場面においては特に有効であるといえる．これまでの研究ではゲート数の抑制などには重点を置いていなかったため，今後の展望として，回路規模の削減や処理速度の向上，ソフトウェアとの比較および任意精度除算の実装に向け検討中である．

文　献

[1] 吉瀬謙二，“高速化と高信頼化に関する研究動向”，電子情報技術産業協会(JEITA) 計算機システム技術専門委員会 報告書，1.1.1.1，(2001 年)

[2] 出口光一郎，森下巌，小笠原司，小野輝，平沢裕，渡辺淳，“単一バス同期データ交換型マルチコンピュータシステムの高能率化”，情報処理学会論文誌，Vol.20 No.4，(1979 年 7 月)

[3] 佐々木広，“GALS 型マイクロプロセッサの低消費電力化に関する研究”，東京大学大学院 情報理工学系研究科 修士論文，(2004 年)

[4] 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)

[5] 眞田祐一，恒川佳隆，“拡張容易な任意精度乗除算器の構成について”，計測自動制御学会東北支部第 196 回研究集会，資料番号 196-3，(2001 年 7 月 30 日)