

## メッセージ駆動形 IP コアインタフェースの一提案

佐々木 隆太<sup>†</sup> 中村 次男<sup>†</sup> 冬爪 成人<sup>†</sup> 笠原 宏<sup>†</sup> 田中 照夫<sup>‡</sup>

<sup>†</sup> 東京電機大学大学院情報環境学研究所 〒270-1382 千葉県印西市武西学園台 2-1200

<sup>‡</sup> 東京電機大学工学部 〒101-8457 東京都千代田区神田錦町 2-2

E-mail: <sup>†</sup> {sasaki, nakamura, fuyu, kasahara}@itl.sie.dendai.ac.jp, <sup>‡</sup> tanaka@e.dendai.ac.jp

あらまし SoC のような超高集積回路では、開発元の異なる多種多様な IP コアが 1 チップに集積される。そのため、IP コア間のインタフェースの標準化、通信方式、バス使用権の制御などの課題が山積する。そこで、筆者らはこれまでバスと各 IP コア間に通信の制御を行う機構を配置することでインタフェースの標準化を容易にする研究を行ってきた。これにより、以上の諸問題を解決でき、さらに並列性を持たせた効率のよい通信が可能となる。FPGA で試作した結果を報告する。

**キーワード** メッセージ駆動, IP コア, インタフェース, 非ノイマン型, SoC

### A Proposal of Message Driven IP Core Interface

Ryuta SASAKI<sup>†</sup> Tsugio NAKAMURA<sup>†</sup> Narito FUYUTSUME<sup>†</sup> Hiroshi KASAHARA<sup>†</sup>

and Teruo TANAKA<sup>‡</sup>

<sup>†</sup> Graduate School of Information Environment, Tokyo Denki University 2-1200 Muzaigakuendai, Inzai-shi, Chiba, 270-1382 Japan

<sup>‡</sup> School of Engineering, Tokyo Denki University 2-2 Kandanshikichou, Chiyoda-ku, Tokyo, 101-8457 Japan

E-mail: <sup>†</sup> {sasaki, nakamura, fuyu, kasahara}@itl.sie.dendai.ac.jp, <sup>‡</sup> tanaka@e.dendai.ac.jp

**Abstract** In a ULSI such as SoC, various IP cores with different development firms are integrated in single-chip. Therefore problems such as standardization of the interface between the IP cores, communication method, bus use control etc. are piling up. To make the standardization of interface easy, we propose a scheme called access control unit (ACU) which controls communication between the common bus and each IP core. By using this ACU, it is possible to solve not only above mentioned problems, but also to cope with efficient concurrency operation. The results on the trial by implementing ACU on FPGA are also described.

**Keyword** Message Driven, IP core, Interface, Non-von Neumann Type Computer, SoC

### 1. 研究の背景

超大規模集積回路技術の発達により、システム LSI、システム・オン・チップ(SoC)、システム・イン・パッケージ(SiP)等では数十から数百の IP コア(Intellectual Property Core)が 1 チップに集積される<sup>[1][2]</sup>。この規模の集積化においては電源電圧と消費電力、IP コア間のインタフェースの標準化、通信方式などの課題が存在する。またこのような大規模なチップに集積化する全回路ブロックの全てを 1 つの機関で設計しようとする、設計に要する時間やコストの面で非常に困難である<sup>[2]</sup>。そこで IP コアの再利用が必要となってくる。しかし、開発元の異なる IP コアの再利用にはインタフェースの標準化が不可欠であるが、各社それぞれが独自の技術で競って開発している現状を考慮すれば、その標準化は大きな問題である。この問題に対して多くの研究報告がなされているが、本研究ではその中でも、

各 IP コア間の通信方式に関して「全体的に非同期式/局所的に同期式(GALS: Globally-Asynchronous Locally-Synchronous)」とする方式に注目しており、その中でも「局所的に同期式」の部分について、これまで完全自立形 IP コア的设计法<sup>[3]</sup>や IP コア間の通信方式についての研究を行ってきた<sup>[4][5]</sup>。IP コア間での送受信をメッセージとパラメータからなる非常に簡素なフォーマットのメッセージ通信で行うことによってインタフェースの標準化を容易にし、また IP コアとデータバス間にバス使用権など通信の制御を行う機構(ACU: Access Control Unit)を配置し、IP コアはその ACU を介してメッセージ通信によって目的とする IP コアとの通信を行う。さらに IP コアは筆者らが開発した演算処理データ長の制限を受けない完全自立形のアーキテクチャであり、この通信をデータフロー形式のように行うことにより、より並列性を持たせることが可能となる。

## 2. 本研究の特徴

### 2.1 通信フォーマット概要

IP コア間の通信フォーマットを図 1 に示すように、送信先アドレスと精度情報からなるメッセージ部と処理データのパラメータ部のみからなる簡素なものとする。

また、1 通信データの開始と終了を表すために信号線(send\_m)を設ける。

このように通信方式を簡素なメッセージ通信に限定することにより、IP コア間のインタフェースの標準化が容易になり、統一した使用法で多種多様な IP コアの使用が可能となる。

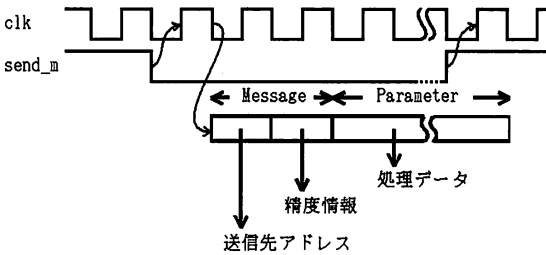


図 1 通信フォーマット

### 2.2 通信制御機構概要

主な機能は対応する IP コア宛のデータのみの取り込みや、データ送出時のアドレス管理、エラー発生時の再送要求、トークン方式によるバス使用権の制御などが挙げられる。また図 2 のように、各 IP コアと対になるように通信制御機構(ACU : Access Control Unit)を配置する。ACU はどのような IP コアに対しても全て同じ機構で構成されており、それぞれの IP コアとオン・チップ・バス間のインタフェースをとり、IP コア間の通信制御を行う。

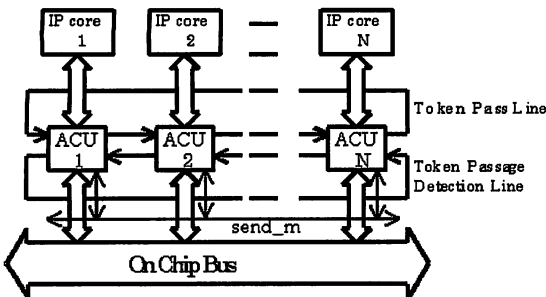


図 2 IP コア - ACU 接続概要

### 2.3 完全自立形任意精度 IP コアの概要

任意の機能を持った同一の基本モジュールを複数個カスケード接続することによりパラメータに合った任意長の演算が実行可能となる。また、この基本モ

ジュールは受け取ったデータの中の精度情報を読み取り、処理に必要な数のみが動作し、その他の基本モジュールについてはスタンバイモードとなるため、消費電力の抑制が可能となる。さらに、メッセージ通信を用いることで、IP コア自身で自立的に処理を行うので、どのような機能を持った IP コアであるのかさえ分かっていたら、内部動作等 IP コア内部の詳細を知る必要はない。処理に必要なデータをパラメータとして送信するだけで結果を得ることができる。

なお、この IP コアはすでに筆者らによって開発済みだが、現段階では、動作チェック用の簡易版の IP コアを製作し、代用している。今後実装していく予定である。

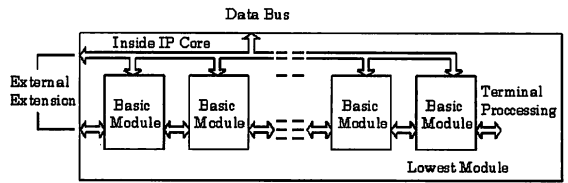


図 3 完全自立形任意精度 IP コアの概要

### 2.4 データフロー形計算機アーキテクチャとの比較

データフローマシン利点の主な利点は並列性を最大限に抽出できることにある。しかし、動的な回路構成となるとコア間の接続をメッシュ状に行うなど、回路規模の増大といった課題が生じる。これに対して提案するメッセージ駆動形アーキテクチャは、独立した演算コアにそのコアを指すメッセージと処理データを送るものであるため、ノイマン型での命令フェッチという形態はなく、また静的な回路構成で上記の利点を実現することができる。

### 2.5 並列性

$y=(a+b)*(c-d)$  の  $a+b, c-d$  が並列に処理できることと同じように、本研究では並列に行える処理をすべて順次それぞれの IP コアへ送出する。依頼された処理を終えた IP コアから返送されるので、個々に送出し返送待ちの後に送出するものと比べ、非常に効率のよい動作が可能となる。

### 3. IP コア間の通信例

IP コア間の通信の流れを以下の簡単な(1)式と IP コアの接続図 4 を例に説明する。

$$y=(a+b)*(c-d) \cdots (1)$$

IP コア間の通信概要を以下に説明する。

式 1 では  $a+b$  及び  $c-d$  の両計算は同時に行う事ができるので、プロセッサコア P (以下 P) から ACU P' (以下 P'), On Chip Bus と経由し、各対応 ACU が自宛デ

ータのみを受け取り a,b は加算器 A (以下 A) に, c,d は減算器 (以下 S) へ届けられる. その後, 与えられた処理を終え, 逆のルートを通りそれぞれの値が P へ返送される.

また, P に, 加算結果, 減算結果がそろった状態になった時に, それぞれの処理結果を乗算器 M (以下 M) へ先ほどと同様に送し, その結果が返送されれば, 式 1 の結果である y が得られることとなる.

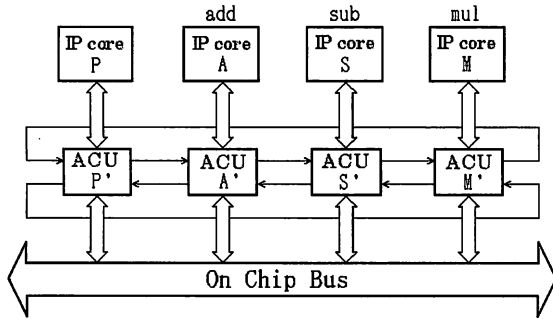


図 4 IP-ACU 接続例

具体的な流れを追っていくと, 以下のようなになる.

(1) P は, P' がトークンを保持し, バスの使用権を得ると a,b を A に, c,d を S へ送信するために, P' へデータを送出する. P からデータを受け取った P' は図 5 のようにアドレス変換を行いバス上へ送出する. また, トークンの保持はここで終了し, 次段の ACU へトークンをパスする.

(2) バス上に送出されたデータについて, 各 ACU は先頭アドレスから自宛のデータであるかを判別し, 自宛であれば続けてデータの取り込みを行い, 自宛でない場合はそれ以下のデータは無視する. また, 2 つ目のデータについても同様の判別を行う (図 6).

(3) 自宛のデータを取り込んだ ACU (A', S') はここでもアドレス変換を行い, そのデータを対応する IP コア (それぞれ A, S) へ送る (図 7).

これで, 処理を依頼したい IP コアまでのデータの送付と取り込みは完了する. 次から処理済みデータの返送時の動作となる.

(4) 処理を終え返送準備のできている状態に対応 ACU へトークンが回ってくると, トークンを保持し, IP コア (A, S) から ACU (A', S') へデータが送付され, 先ほどの送信元 (P') へ処理結果を返送するようアドレス変換を行い, そのデータをバスへ送付する (図 8).

(5) 処理後のデータ取り込みは (3) と同様に, P' のみがデータを取り込み P へ処理結果が返される. そして, すべての処理結果が返ってきたら, 次の処理依頼を同様に行って, 式 1 の結果 y を求める.

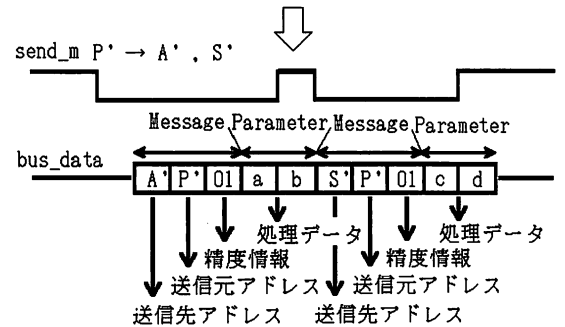
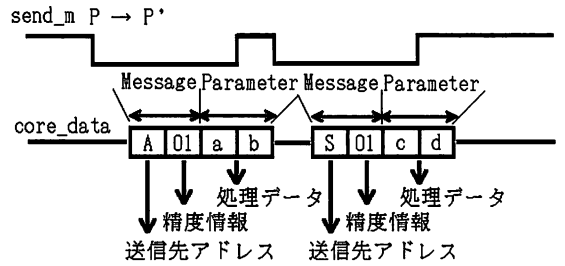


図 5 アドレス変換時

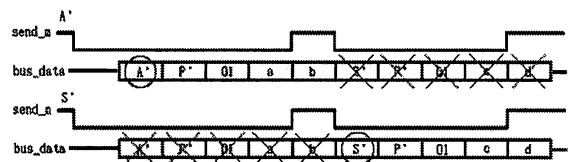


図 6 アドレス判別時

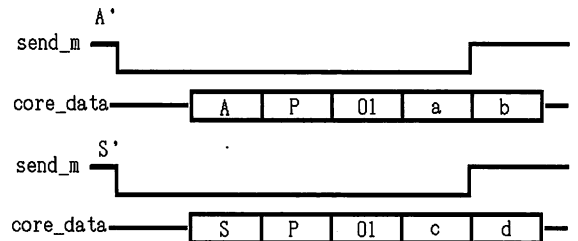


図 7 IP コア取り込み時

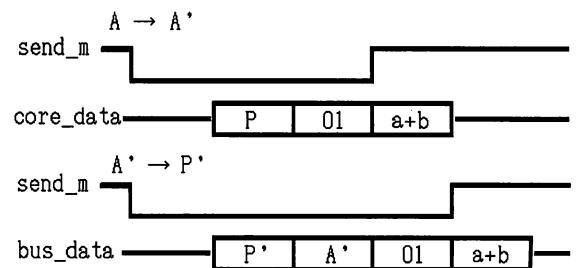


図 8 処理済みデータ返送時

#### 4. 通信制御機構内回路構成および検証

通信制御機構 (ACU) 内は大きく分類すると、図 9 に示すようにトークン制御機構、アドレス変換機構、アドレス判別機構、エラー判別機構の 4 つの機構から成る。各機構の主な動作について以下に述べる。

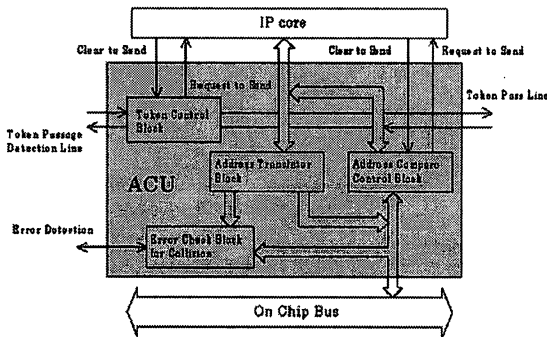


図 9 ACU 内構成

##### 4.1 トークン制御機構

バスの使用权を管理するために本研究ではトークンをリング状にパスする方式をとっている。そのトークンの保持、終了、次段 ACU へパスなどの動作を主に行っているのがトークン制御機構である。

対応する IP コアから通信要求が出ている状態でトークンが回ってきた場合には、前段 ACU から入力されるトークンを保持する。その後データのバスへの送出手続きが完了すると、前段 ACU へのトークン保持終了通知をアクティブにする。トークンを出力している状態で次段 ACU からトークン終了通知が入力された場合はトークンの出力を終了し、トークンは次段の ACU へと移っていく。対応 IP コアから通信要求が出ない場合、すなわちバスを使用する必要のない場合はトークンは保持せず、次段 ACU へそのままトークンをパスする。

また、トークンの有無はバスの使用权に直結するので、トークンが同時に 2 つ発生してしまうことのないようにトークンの出力とトークン終了通知はハンドシェイクの形をとっている。

万が一、途中でトークンが消滅してしまった場合には、任意の一定期間経過後に設定した 1 つの ACU においてトークンを再発生させることで対応している。

また、同一 IP コアから他 IP コアへの処理要求データが複数ある場合には、処理を要求したい全てのデータを流し終えるまでトークンを保持することで、同時に複数 IP コアへの要求が可能となり、効率のよい並列処理が可能となる。

に送出されているデータが同一のものではなくなる。するとその状態を検知し、バス上でエラーの発生を判断

図 10 においては、通信要求が出ている IP コアに対応した ACU へトークンがまわってきたときにトークンを保持する。この場合は 3 つのデータを続けてバスへ送出している (図 10①)。全てのデータをバス上に送出し終わると次の段の ACU へトークンをパスする。このとき次の段の ACU に対応した IP コアは演算処理を終えているので、トークンを保持し送信元である IP コア宛へデータを返送する (図 10②)。以降③、④と同様の動作を行う。通信要求の出ない IP コアに対応した ACU へトークンがまわってきたときは、トークンを保持せずそのまま次段へパスする (図 10⑤)。

##### 4.2 アドレス変換機構

IP コアから ACU へは送信先の IP コアが指定されたデータが送出され、そのアドレスを対応 ACU 宛へと変換する。また、どのデータがどこから送られてきたものなのかを判断するために、送信先アドレスと精度情報の間に自 ACU のアドレス (送信元アドレス) を追加している。この追加された自アドレスは、ACU 内にて保持しておき、IP コアにて処理が終わり返送するときに、このアドレスを送信先アドレスとして先頭に付加するといった変換も行っている。

以上のように、複数データを続けて複数の IP コアに送った場合でも、自 IP コアで処理されたデータの返信先が判明しているため、依頼元である送信元の IP コアへ正しく返送することができる。

##### 4.3 アドレス判別機構

アドレス変換を終え、バスへ送出されたデータは全ての ACU へと入力される。しかし、必要なデータとなるのは、自宛のデータのみとなるのでアドレス情報を見て自分宛のデータのみを最後まで取り込んで、自宛でないものについては、その時点で以降のデータを無視する。

なお、複数データが続けて送出された場合に、例えば 1 つ目のデータが自宛のものでなくても、それ以降のものに自宛のデータがあれば、該当データの取り込みを行うことができる。これはデータの開始と終了を表すための信号線 (send\_m) で判断することができる。

##### 4.4 エラー判別機構

データが何らかの理由で IP コアへ送出することができなかった場合や、バスへ送出した時点で、他のデータと衝突してしまった場合など、万が一正しくデータの受け渡しが不成立となってしまった場合に、正しく送出することができなかったデータについて再送要求を行う。例えば、何らかの理由で同時に 2 か所からバスへデータが送出されると、バス上でデータの衝突が起き、ACU が送出しようとしたデータと、現在バス

して、エラー信号をアクティブにする。その後、次にトークンがまわってくると、バスへ送出したデータの送信元へ再送要求を送信する。また、この信号がアクティブになってから任意の期間はトークンを一時的に消滅させ、その後トークンを1つ発生させる、といった動作となる。

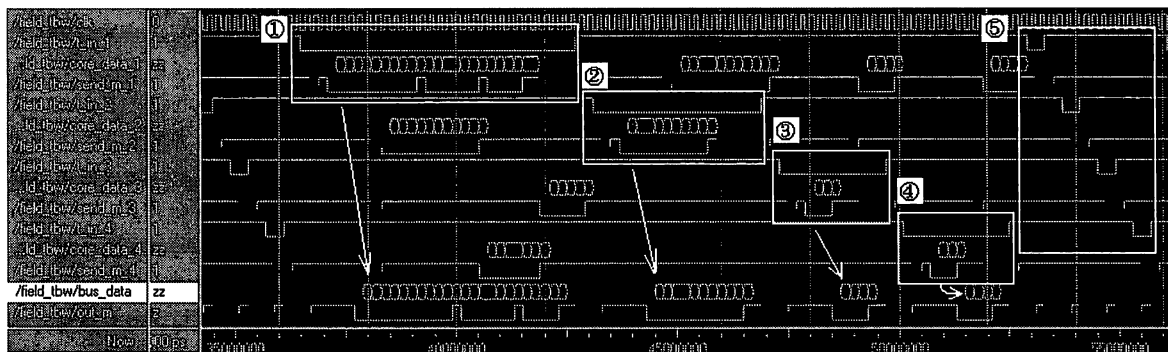


図 10 トークン保持とバスの使用権の遷移

#### 4.5 IP コア間の通信検証

IP コア間での通信を行う際のシミュレーション結果を以下に示す。なお、回路合成には xilinx 社 ISE9.2i を、シミュレーションには MntorGraphics 社 ModelSim XE3 6.2g を使用した。動作説明は次ページに示す。

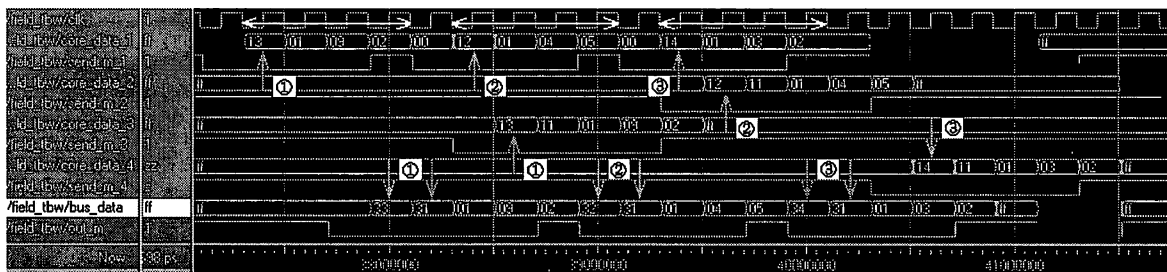


図 11 シミュレーション 1 結果前半

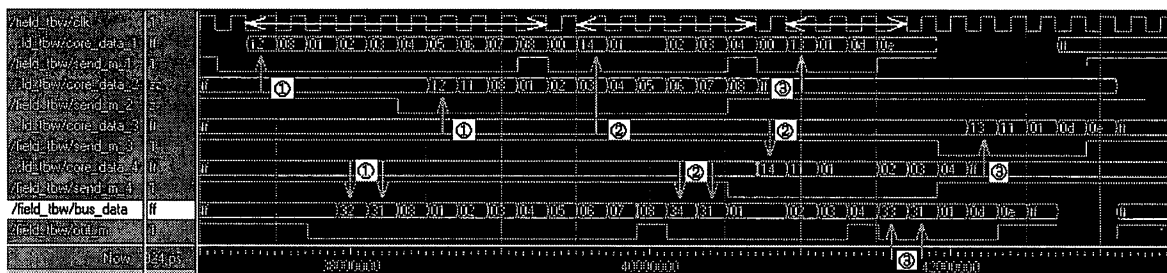


図 12 シミュレーション 2 結果前半

#### 入出力端子説明

core\_data\_1…IP コア 1, ACU1 間入出力  
 send\_m\_1…core\_data\_1 のデータ開始, 終了通知線  
 core\_data\_2…IP コア 2, ACU2 間入出力  
 send\_m\_2…core\_data\_2 のデータ開始, 終了通知線  
 core\_data\_3…IP コア 3, ACU3 間入出力

send\_m\_3…core\_data\_3 のデータ開始, 終了通知線  
 core\_data\_4…IP コア 4, ACU4 間入出力  
 send\_m\_4…core\_data\_4 のデータ開始, 終了通知線  
 bus\_data…バス上のデータ  
 out\_m…bus\_data のデータ開始, 終了通知線

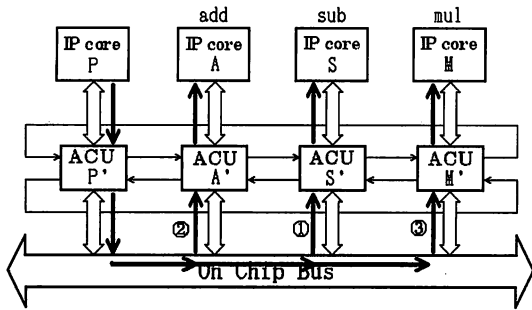


図 13 シミュレーション 1 流れ

表 1 アドレス対応表

IP	P	A	S	M
	11	12	13	14
ACU	P'	A'	S'	M'
	31	32	33	34

図 11 のシミュレーション 1 結果は図 13 のように core\_P から, core\_S, core\_A, core\_M へ続けて処理依頼を出力している. 図 11 中の①, ②, ③は各データのアドレス部を示している.

3つのデータそれぞれが core\_P からリアルタイムにアドレス変換を行い, 自アドレスも追加したアドレス変換後のデータをバスへ送出, その後, 各コア内にてそれぞれのデータの中のアドレス情報を読み取り, 自宛のデータのみが取り込まれ, 自宛でないものについては破棄する.

その後, 演算処理結果も同様にアドレス変換を行い, 処理の依頼元である core\_P へ返送される.

図 12 のシミュレーション 2 結果でも基本的には同じ動作をしているが, データ長が大きいデータについても同様にアドレス変換, アドレス判別等を行い, 同じ形式で通信が行われたことが確認できる. これにより非常に複雑なパラメータを持つ演算処理を依頼しても ACU の回路構成や通信フォーマット等を全く変化させることなく, 対応できることが確認できた.

以上のように同時に処理できる演算処理を, 1 回のトークン保持時に続けて各 IP コアへ送信することで, 並列的な処理が可能となる. また, アドレス情報と, 精度情報からなるメッセージ部とパラメータのみのデータ通信であり, 必要最低限の情報, データ長で効率的な処理が行われていることも確認できた.

## 5. まとめ

超高集積回路化において電源電圧と消費電力, IP コア間のインタフェースの標準化, 通信方式及びバス調

停機構の問題に対し, IP コアのインタフェースの標準化を解決する手段として, IP コア間の送受信をメッセージとパラメータからなる非常に簡素なフォーマットのメッセージ通信とすることによって, インタフェースの標準化を容易に実現するための設計を行った. これにより, IP コアを 1 オブジェクトと捉えることによって, 柔軟性のあるモジュールとして考えることができる. また IP コア間の通信においてバス調停機構が問題となるが, これを ACU 間のバス制御をトークン方式とすることによって, 従来のようなバス調停機構を必要としない単純な構成で実現した. これらにより, 新たに IP コアを追加しようとした場合でも容易に対応することができる. そして使用者は全ての IP コアの内部仕様を知る必要なく, 多くの IP コアを容易に集積化し, 使用することが可能となる. さらにデータフロー的な並列性を持たせた動作が可能となるので, 1 つ 1 つの処理を終えてから次の処理に移るのではなく, 複数の処理を 順次依頼でき, 処理の効率化も向上する.

なお, 今回の通信フォーマットを採用したことにより, 以前本研究室で開発していた旧 ACU に比べ, さまざまな機能を追加した上に ACU 単体でのゲート数が 3366 と非常に小規模に抑えることに成功した. さらに, 開発済みだが今回の回路にまだ実装していなかった完全自立形任意精度 IP コアの実装と, 演算処理を行うソフトも含めたシステムの試作に向けて開発中である.

## 文 献

- [1] P.P.Pande, C.Grecu, M.Jones, A.Ivanov and R.Saleh, "Performance Evaluation and Design Trade-offs for Network-on Chip Interconnect Architectures", IEEE Trans. on Computers, pp.1025-1040, Aug 2005
- [2] Luca Benini and Giovanni De Micheli, "Networks on Chips", IEEE, Cimpler, pp.70-78, Jun 2002
- [3] 佐藤正幸, 中村次男, 畠中浩行, 冬爪成人, 笠原宏, 田中照夫: "完全自立形スケーラブル IP コア的设计法と楕円曲線暗号処理システムへの適用" 情報処理学会論文誌, Vol.48, No.2, pp.868-879, Feb.2007
- [4] 早川雅文, 中村次男, 冬爪成人, 笠原宏, 田中照夫 "システム LSI 回路ブロックのインタフェースと通信方式," 電気学会論文誌, C, vol. 127, No.6, pp.867-873, Jun.2007
- [5] 緒方賢一郎, 佐々木隆太, 中村次男, 冬爪成人, 笠原宏, 田中照夫 "データフロー形スケーラブルコンピュータ" 電気関係学会九州支部連合大会, 09-1P-04, p.209, Sep 2007