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

2009-SLDM-138 (30)
2009 / 1 / 30

GALSシステムにおける非同期バスの実装

堀 武宏† 中村 次男† 冬爪 成人† 笠原 宏† 田中 照夫‡

†東京電機大学大学院情報環境学研究科〒270-1382 千葉県印西市武西学園台 2-1200
‡東京電機大学工学部 〒101-0054 東京都千代田区神田錦町 2-2

E-mail: † {hori, nakamura, kasahara, fuyu}@itl.sie.dendai.ac.jp, ‡ tanaka@e.dendai.ac.jp

あらまし 非同期式回路では同期式回路で発生する消費電力・速度・ノイズ・クロックスキューなどの問題を解決することができるが、その性質上、一対一の場合のみ、かつ一方向にしかデータを転送できない。そこで、同期式回路における双方向の同時通報機能を模した、非同期式回路におけるバス方式を提案する。また、同期式・非同期式の回路が混在するGALSのネットワークにも対応できるように設計した。これにより、既存の設計資産も利用可能で、省電力・高速・低ノイズで安定したチップ内ネットワーク(NoC)を実現することができる。

キーワード NoC, GALS, 非同期式回路, バスアーキテクチャ, オンチップバス

Implementation of Asynchronous Bus for GALS System

Takehiro HORI† Tsugio NAKAMURA† Narito FUYUTSUME† Hiroshi KASAHARA† and Teruo TANAKA‡

† Graduate School of Information Environment, Tokyo Denki University 2-1200 Muzaigakuendai, Inzai-shi, Chiba, 270-1382 Japan
‡ School of Engineering, Tokyo Denki University 2-2 Kandanishikichou, Chiyoda-ku, Tokyo, 565-0456 Japan

E-mail: † {hori, nakamura, kasahara, fuyu}@itl.sie.dendai.ac.jp, ‡ tanaka@e.dendai.ac.jp

Abstract Although asynchronous circuit can solve problems of power consumption, speed, noise, and clockskew, the transmission is possible only for 1 to 1 communication and uni-directional. The paper suggests a bus method in the asynchronous circuit with the imitated broadcast function used in the synchronous circuit. It is also designed to be able to cope with the GALS network where synchronous and asynchronous circuits are coexisted. As a result, low-power, high-speed, low noise and stable "network on chip(NoC)" is accomplished where the existing design resources are reusable.

Keyword NoC, GALS, Asynchronous Circuit, Bus Architecture, On Chip Bus

1. 研究背景

近年集積回路は集積度の増加が著しく、それに伴い構造が複雑になり、設計が困難になっている。これに対応するため、将来的には細分化された機能毎に設計される IP(Intellectual Property)コアと呼ばれる再利用可能な回路を単位として、一つのチップ上にこれを多数配置して接続する SoC や NoC といった集積回路技術が一般的になると考えられている。本研究は多種多様な IP コア間のネットワーク(回路ブロック)同士を接続する大局的なネットワークを想定し、回路ブロック間を接続するバスを提案するものである。

また集積度の増加によって、システムクロックを用いた同期式設計の問題点、即ち消費電力・処理速度の限界・電磁放射ノイズ・クロックスキューなど多くの課題が懸念されており、今後は非同期式設計が不可欠になると考えられる。しかし既存の同期式回路にも設計資産としての価値があるので、非同期式への移行は速やかには進まないと考えられる。

そこで、同期式・非同期式双方の回路を扱う GALS[1] というネットワーク構成が注目されてきている。我々はこの GLAS 構成のシステムを実現するため、同期式のバスの動作を模した非同期バスを用いた通信方式を提案している[2]。

本研究ではこの非同期バスを実際に設計し、これにより同期式・非同期式回路双方のインターフェースを取り、既存の同期式回路を利用しながらもネットワーク全体の省電力化を実現できる。また非同期バス自身も非同期式回路であるため、低消費電力で高速に動作し、ノイズも低く抑えられる。

2. 非同期式回路の実装法

2.1. 遅延モデル

非同期式回路の遅延モデルには、BD(bounded delay)モデル、DI(delay insensitive)モデル、SDI(scalable delay

insensitive)モデル、SI(speed independent)モデル、QDI(quasi delay insensitive)モデルなどがある[1][3][4]。

本研究では、全ての配線とゲートにおける遅延がどのような値でも動作する DI モデルに基づいて回路を設計する。

2.2. 回路方式

非同期式回路では処理の完了を検出する必要があり、これによって回路方式が束データ方式と 2 線 2 相式に大別される[3]。束データ方式では回路の遅延より大きな遅延を持つ遅延回路によって、一方、2 線 2 相式では(0,1)または(1,0)に符号化された論理値の到着によって処理の完了を検出する。

本研究では最も安定して動作する DI モデルに基づいて回路を設計するため、2 線 2 相式を採用する。

2.3. ハンドシェーキング・プロトコル

非同期式回路では回路ブロック間の通信にハンドシェーキング・プロトコルが使われる。使われるプロトコルは 2 相ハンドシェーキングと 4 相ハンドシェーキングがあり、それぞれ処理時間が短いが回路規模が大きい、処理時間が長いが回路規模が小さい、という特徴がある。

本研究では構造の簡単な 4 相ハンドシェーキングを用いる。

2.4. 非同期通信方式

2 線 2 相式の回路で通信を行う際は、受信側が送信側に全てのデータを受信したことを通知することで通信を行うため、CHAIN パイプラインラッチ回路[5](図 3)が一般的に用いられている。本研究もこれに倣う。

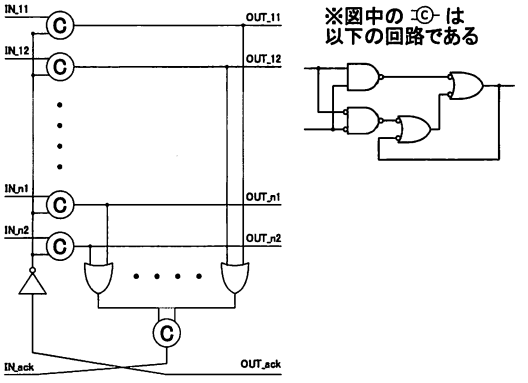

図 1 CHAIN パイプラインラッチ回路

3. ネットワーク構成

本研究ではまず IP コア間を接続する任意のネットワーク(回路ブロック)を想定し(図 2)、それらを接続するバスを提案する(図 3)。

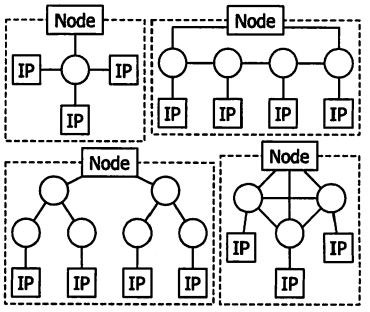

図 2 回路ブロック

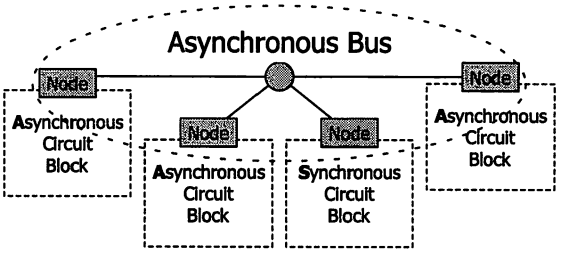

図 3 ネットワーク全体

バスは非同期式回路で設計し、回路ブロックが同期式の場合でも非同期式の場合でも動作する GALS 構成とする。これは既存の同期式回路の資産を活用しながら、全体としては非同期式回路の低消費電力などの特徴を活かすためである。

4. 回路構成

本研究の非同期バスは回路ブロックと外部とのインターフェースである Node(図 4)、データの転送路である CHAIN パイプラインラッチ、バス使用権の調停を行う通信調停機構(Communication Arbiter:以下 CA)によって構成される(図 5)。

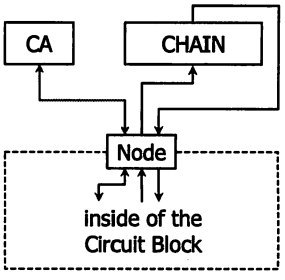

図 4 Node

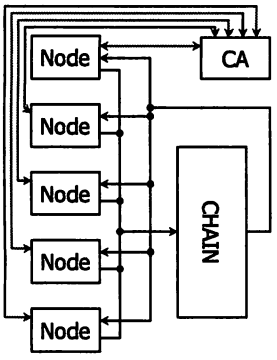

図 5 非同期バス

5. 通信方式

非同期バスを通過する際の通信方式は単純なメッセージ通信とする。データフォーマットは 1 ワードのアドレス部と任意ワードのメッセージ部からなり、1 ワード目が任意の回路ブロック内の任意の IP コアを指すアドレスであれば、メッセージ部の形式は問わない。従って精度情報や送信元アドレスなどの情報を持たせ、回路ブロック内ローカルのフォーマットで送信することも可能である。これにより、パケット通信などにも対応できる(図 6)。

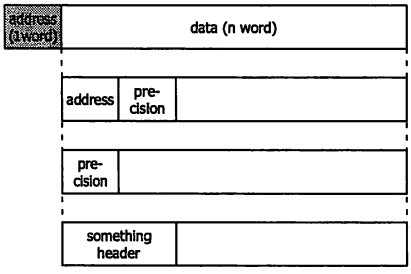

図 6 通信フォーマット

Node0 から Node4 への通信の例を図 7 に示し、以下に通信手順を説明する。

① 送信元 Node が CA に送信要求を送出
② 他の Node が通信中でなければ CA が送信元 Node に送信許可を応答
③ 送信元 Node が、自身と CHAIN を接続しているデータ線とアクノリッジ線をアクティブにする
④ 送信元 Node が CHAIN へ送信先アドレスのデータを送信、全 Node がアドレスを受信
⑤ 受信したアドレスと回路ブロック内の IP コアのアドレスを全 Node が比較し、送信先 Node のみが CA へ受信要求を送出
⑥ CA が送信先 Node に受信許可を応答
⑦ 送信先 Node が自身と CHAIN を接続しているアクノリッジ線をアクティブにし、アクノリッジ信号を送出
⑧ 送信元 Node がアクノリッジ信号を受信し、データを 1 ワード送出

(以後、送信先のアクノリッジ信号送出と送信元のデータ送出の繰り返し)

⑨ 送信元 Node がデータの終了パターンを送出、CHAIN へのデータ線とアクノリッジ線を非アクティブにし、CA への送信要求を終了
⑩ 送信先 Node がデータの終了パターンを受信、CHAIN へのアクノリッジ線を非アクティブにし、CA への受信要求を終了
⑪ CA が送信元 Node への送信許可と送信先 Node への受信許可を終了

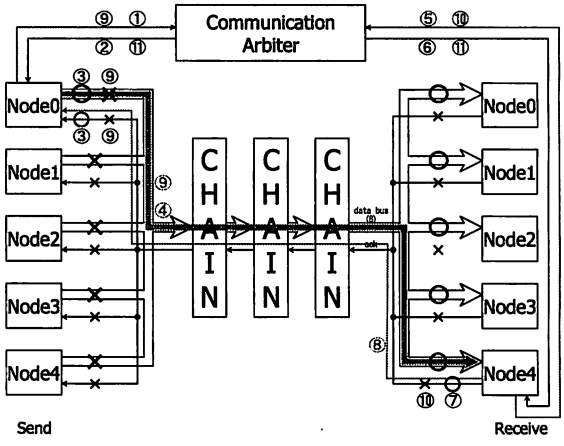

図 7 通信の手順

なおこの図では本来一つである Node を送信側と受信側の二つに分け、左右にそれぞれ配置している。

ここで送信元 Node は全 Node にデータを送出しているが、通信をしているのは 1 ワード目に送出したデータに含まれるアドレスに対応した Node だけである。これは、各 Node が自律的にデータ線やアクノリッジ線を開閉することで、多対多の通信路を 1 対 1 の通信に対応させた結果可能となっている。

また、CA は送信要求が送られた際に他の Node が通信中であれば送信許可を返さないため、通信路でデータの衝突が起こらないことが保証できる。

6. 回路の設計

本研究では Xilinx 社の ISE™ 9.2.04i を用いて回路を設計した。今回は 4 つの回路ネットワークを非同期バスで接続し、通信を行うモデルを作成し、これに基づいて設計を行った。

6.1. Communication Arbiter(CA)

CA は、内部に 4 出力の Asynchronous Demultiplexer (図 8)、4 入力の Asynchronous Multiplexer(図 9)、4 状態の Asynchronous Selector(図 10)を持つ 2 つの Polling Unit(図 11)で構成されている。

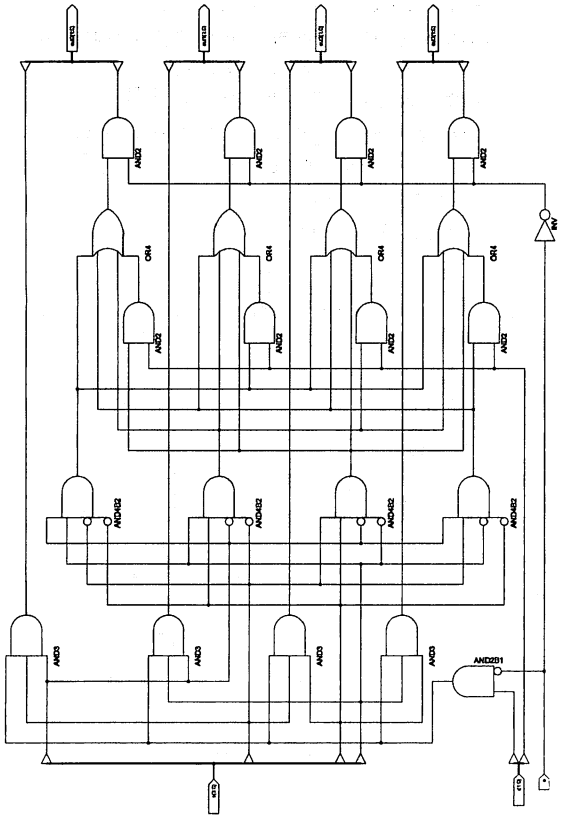

図 8 Asynchronous Demultiplexer

Asynchronous Multiplexer 及び Demultiplexer は、通常の Multiplexer・Demultipexer と同じ機能を持つ回路を 2 線 2 相式で設計したものである。入出力が 1 ビットに付き 2 線ある点と、出力が切り替わる際には途中に必ず(0, 0)で符号化されるスペーサを挟むという点が、通常の同期式の Multiplexer・Demultiplexer との相違点である。

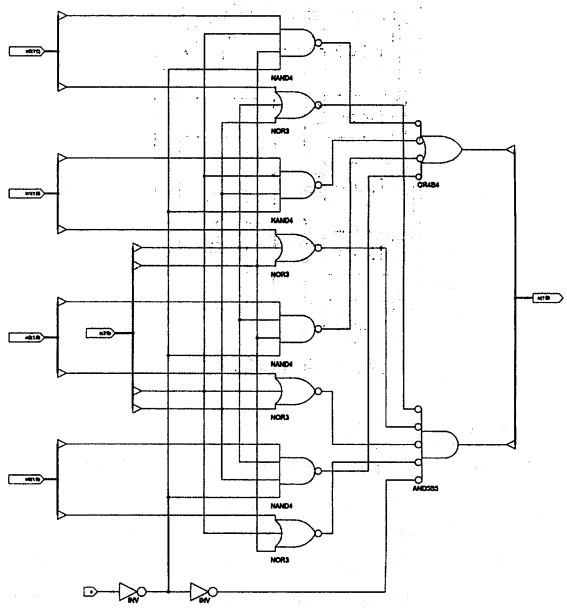

図 9 Asynchronous Multiplexer

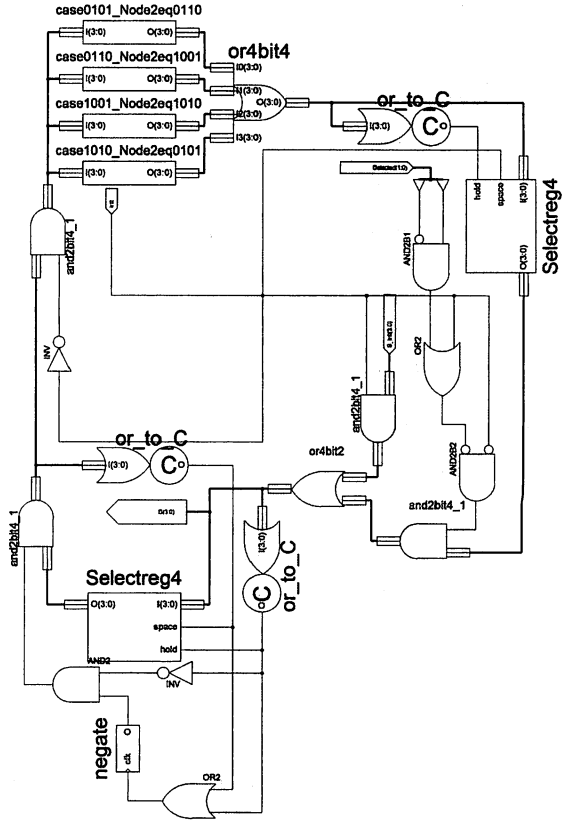

図 10 Asynchronous Selector

Asynchronous Selector は 1 ビットの入力値の変化を検出する毎に 4 ビットの出力値を＋1 する回路である。これら 3 つのモジュールを接続することで Polling Unit とする(図 11)。この Polling Unit を 2 つ持つ機構が CA である。

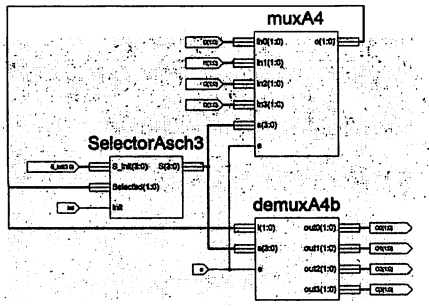

図 11 Polling Unit

Polling Unit は名前が示す通り、ポーリングの処理を行い、CA のバス制御の中核を担う部分である。それぞれの Polling Unit は送信側または受信側の Node とハンドシェーク線で接続されるが、Polling Unit 同士は接続されない。

6.2. Node

Node は Address Table、Node_ctrl、及び多数の Tri-State buffer からなる通信線路で構成される(図 12)。

Address Table は Node にアドレスが送られてきた際、当該 Node が担当する回路ブロック内に送られたアドレスと対応する IP コアが存在するかどうかを判断し、真偽値を返すという機構である。

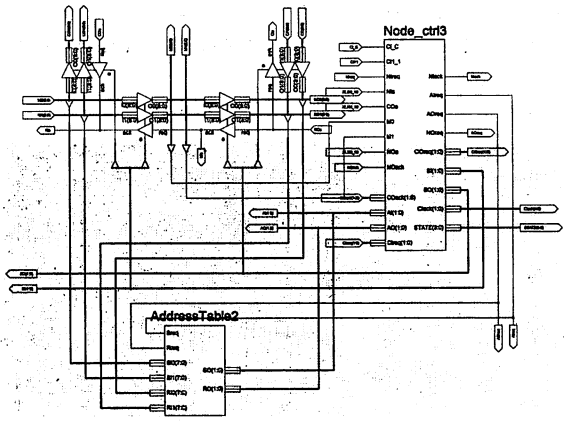

図 12 Node

Node_ctrl は回路ブロック内外とのハンドシェーク、通信路の開閉などを司る機構であり、Node の最も基本となる機能を担う。

これら Address Table 及び Node_ctrl に関しては Verilog-HDL という言語を用いて設計したが、ページの関係上ソースの掲載は省略する。

7. 検証

Mentor Graphics®社の ModelSim® XE III/Starter 6.2c でシミュレーションをおこない、回路の動作を検証した。行ったシミュレーションの内容を表すタイミングチャートを以下に示す。

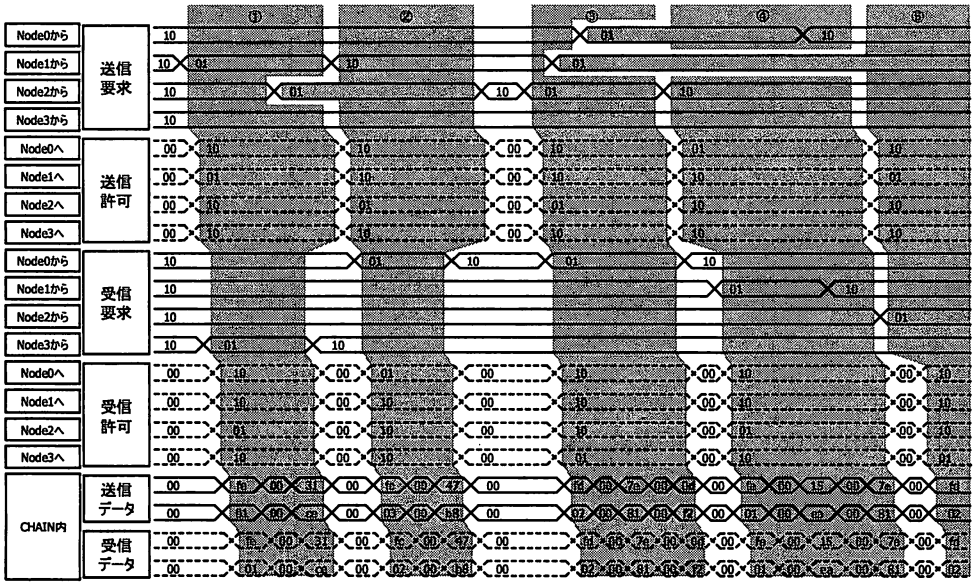

図 13 シミュレーションのタイミングチャート

このように、①Node1 から 2 へ、②Node2 から 0 へ、③Node2 から 3 へ、④Node0 から 1 へ、⑤Node1 から 3 へ、という 5 通りの通信データを与え、このデータを CA が正しく処理し、送信側 Node から CHAIN を通って受信側 Node へデータが送られるかどうかを確認する。

なお、図中の点線は与えた入力に対して期待される出力結果である。

シミュレーション結果を以下に示す。

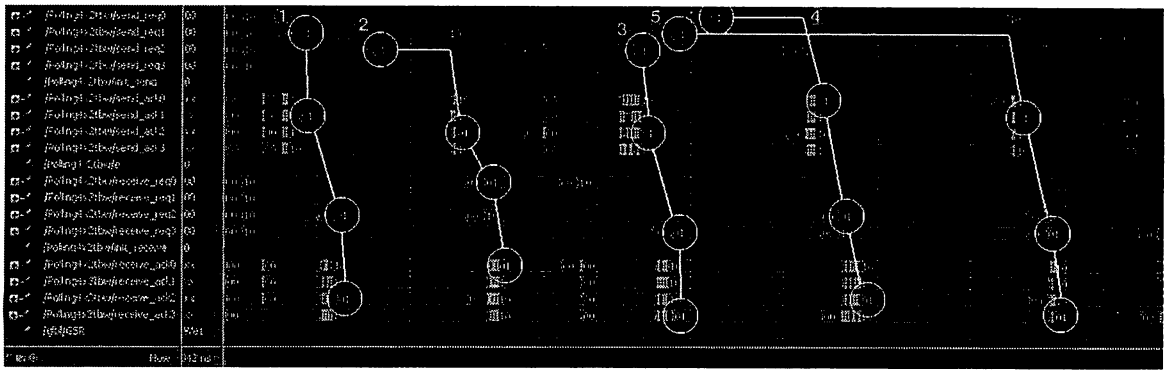

図 14 シミュレーション結果波形（CA）

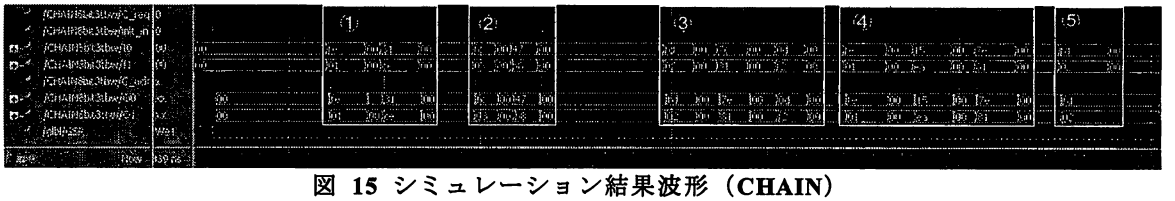

図 15 シミュレーション結果波形（CHAIN）

これらの結果から、回路ブロック同士が実際に非同期バスを介して正しく通信できることが確認できる。

8. まとめ

一般的に 2 線 2 相非同期式回路でバス方式は困難だと言われていたが、今回の設計とシミュレーションにより、実際に動作させることが可能であるという確証を得た。

また設計した非同期バスにより、同期式・非同期式に関わらず、IP コアによる回路ブロック間の通信が、低消費電力・高速・低ノイズかつ安定した環境で実現できる。これにより、昨今特に厳しく問われる消費電力を始めとして処理速度やノイズなどの問題を非同期式回路で解決しながら、同時に非同期式回路を使う場合のデメリットである、既存の同期式の設計資産が利用できなくなるという問題にも対応することが可能である。

このように、集積度の向上により発生してきた新しい問題を解決しながらも旧来の技術との互換性も保っているので、広く使われる可能性を持った技術であると言える。

今後はこの非同期バスの回路を FPGA にダウンロードして実動作を行うとともに、より高速・小規模で低消費電力な回路に改良していく予定である。

文　献

[1] 唐木信雄, “非同期回路設計のすすめ,” Design Wave Magazine, vol.10, no.7 pp.64-69, Jul.2005.

[2] 堀武宏, 中村次男, 冬爪成人, 笠原宏, 田中照夫, “ネットワーク・オン・チップにおける非同期バスの一提案”, 第 60 回電関学会九支連大, no.09-1P-09, p.214, Sep 2007

[3] 斎藤誠司, 渡邊誠也, 正木亮, “2 線 2 相式非同期回路用 FPGA アーキテクチャ”, 信学論 D-I, Vol.J86-D-I, no.2, pp.108-116, Feb.2003.

[4] Chris J. Myers, 非同期式回路の設計, 共立出版株式会社, 東京, 2003

[5] John Bainbrige, and Steve Furber, “CHAIN : A Delay-Insensitive Chip Area Interconnect”, IEEE Micro, Vol.22, No.5, pp.16-23, Sep/Oct 2002