

FPGA 高速シリアル通信を用いた ASIC エミュレータに関する研究

佐伯 亮 祐^{†1} 高橋 克 昇^{†1} 久我 守 弘^{†1}
 尼崎 太 樹^{†1} 飯田 全 広^{†1} 末吉 敏 則^{†1}

近年, ASIC は回路規模の増加に伴う開発期間の長期化が大きな問題となっている. 中でも検証は開発期間の多くを占めており, 期間全体の 6 割を検証期間が占めているといわれている. そのため, 検証期間の短縮が重要な課題である. 現在, 最も高速な検証手法は FPGA を利用した ASIC エミュレータである. しかし, FPGA は搭載できる回路規模に制限があるため, その制限を超える規模の回路は複数の FPGA に分割して搭載する必要がある. また, FPGA 間でのデータ送信手段が問題となる. そこで本稿では, 複数の FPGA を高速シリアル通信で接続することを前提にした分割回路の通信手法を提案する.

A Study of FPGA based emulator using high-speed serial communication

RYOSUKE SAEKI,^{†1} KATSUNORI TAKAHASHI,^{†1}
 MORIHIRO KUGA,^{†1} MOTOKI AMAGASAKI,^{†1}
 MASAHIRO IIDA^{†1} and TOSHINORI SUEYOSHI^{†1}

Recently, as for the ASIC (Application Specific Integrated Circuit), the on-going large scale of the circuit every year due to advances in semiconductor process miniaturization technology. At the same time, the functional verification is prolonged. The verification holds 60% of the development period. We should shorten a period of verification. Therefore, FPGA based emulator is very useful for VLSI emulation system. However, communication between FPGAs becomes the bottleneck and total operating speed largely deteriorates. In this report, we use high-speed serial communication for data transfer between FPGAs and discuss operation speedup technique for FPGA based ASIC emulator. As a result, we make high-speed serial communication mechanism between FPGAs and we show its effectiveness.

1. はじめに

特定用途向けの LSI である ASIC (Application Specific Integrated Circuit) は開発期間の長期化が問題視されている. その要因の 1 つとして ASIC の高集積化が挙げられる. 高集積化には, チップに搭載できる回路規模の増大などのメリットはあるが, 開発期間の長期化や開発コストの増大等の問題を抱えている. 中でも機能検証に非常に多くの時間を要する.

現在, 設計検証にはソフトウェアによる論理シミュレータ, ハードウェアアクセラレータ, 回路構成を変更可能な LSI である FPGA (Field Programmable Gate Array) によるエミュレータ等が用いられている. その中でも FPGA を用いたエミュレータ (以後, FPGA エミュレータと記述する) は, 他のエミュレータと比べエミュレーション速度が高速であり比較的安価なソリューションとして期待されている. しかしながら, FPGA の回路資源の制約から, 大規模な ASIC デザインは単一の FPGA に実装できない場合がある. この場合, 検証を行うためにはデザインを分割して複数の FPGA へ搭載する必要がある. このときに問題となるのが分割した回路の分割面に現れる通信信号と, FPGA における I/O 端子不足の問題である. ASIC デザインを分割した FPGA に実装する際, 分割面に膨大な数の信号線が現れる. 従来の FPGA エミュレータでは, それらをパラレルで接続する. そのため, FPGA の I/O 端子数が不足する恐れがある. また, I/O バッファによる遅延も動作速度を低下させる要因である.

そこで, 本研究では FPGA 間の通信に高速シリアル通信を用いることで, FPGA 間の通信の性能向上を図り, 複数の FPGA の動作速度向上を目指す. さらに, 従来手法と比較して使用する FPGA の端子数を削減できるため, ASIC デザインを FPGA へ移行する際の容易性の向上についても述べる. また同時に提案通信手法向けの回路分割手法についても議論を行う. 以下, 2 章で一般的な LSI 機能・論理検証とその方法について述べ, 3 章で高速シリアル通信について提案を行い, 4 章で結果を述べる. 5 章で結果についての考察を行い, 6 章でまとめる.

^{†1} 熊本大学 大学院 自然科学研究科 〒 860-8555 熊本市黒髪 2-39-1

Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Kumamoto 860-8555

2. LSI の設計検証

2.1 設計検証の重要性

LSI の設計コストのうち、60%から 70%を設計検証が占めているといわれている¹⁾。さらに、1 度製造ラインに入った LSI が何らかの問題で設計に戻る確率は年々上がっており、65%が再設計となっている報告もある。その失敗の多くは論理の設計のミスであり、検証に十分なコストをかけているにも関わらず依然として検証が不十分であることが指摘されている。従って、LSI の設計期間、コストを計画通り抑えるためには設計検証を正しくかつ効率良く行うことが求められる。LSI の機能検証を正しく効率的に行う上で以下の 3 点が課題として挙げられる。

検証品質の確保： 検証ツールの高速化や高速なエミュレータが用いられているが、決められた時間やリソースの制約の中でいかに検証の品質を確保したまま検証行うかが課題となっている。

設計の複雑化： 設計が複雑化することにより検証のためのテストパターン作成が困難となり、検証漏れによる不具合が発生する。

設計期間： プロセスの微細化により設計回路の規模は大規模化するが、製品サイクルの短縮により設計期間は短縮する。

2.2 ASIC エミュレータの従来手法

設計検証を行う手法として、論理シミュレータ、ハードウェアアクセラレータおよび FPGA エミュレータの 3 つが挙げられる²⁾³⁾。

論理シミュレータは、設計した論理回路の論理機能やタイミングを検証するためのツールである。これはすべてのシミュレーションをソフトウェアベースで行っているため、非常に時間がかかる。

ハードウェアアクセラレータは論理シミュレータの処理の一部を専用ハードウェア化したものである。検証対象は基本的にはゲート・レベルの論理回路であり、ハードウェアアクセラレータの処理速度は、ワークステーション上でイベント・ドリブン型シミュレータを動作させた場合と比べて 10 倍～数百倍高速である。

FPGA エミュレータは、検証対象の論理 LSI を FPGA に実装しプロトタイピングを行う。FPGA の柔軟性を活かしたエミュレータである。論理エミュレータ上に展開した論理回路は数 MHz 前後で動作する。これは、ワークステーション上で稼働するソフトウェアの論理シミュレータと比較して 4 桁～5 桁高速である。またハードウェアアクセラレータと比

較しても 2 桁～3 桁動作速度が速い。検証期間の増大に伴い、動作速度の速い手法が求められているため、FPGA エミュレータは現在広く利用されている。

2.3 FPGA を利用した検証方法

FPGA エミュレータに FPGA の規模を超える ASIC デザインを移行する際には、ASIC デザインを分割して FPGA に搭載しなければならない。その場合は、FPGA の I/O 端子から信号を出力し他の FPGA へ通信する必要がある。その際の問題点として以下の 5 点が考えられる。

動作周波数の低下： ASIC デザインを複数の FPGA に搭載した際、ASIC デザインの分割面に表れる多くの I/O 端子がパラレルに接続される。この場合、タイミング・スキューやディレイの問題が発生しエミュレータの性能が FPGA 間の通信性能に依存してしまう。

I/O 端子の不足⁴⁾： ASIC デザインを分割した際、分割面に現れる信号数が FPGA の外部出力端子数よりも多くなる可能性がある。その場合はエミュレーションができない。

複数の FPGA 間の配線が困難： ASIC デザインの分割面に表れる多くの I/O 端子を再び配線することはユーザの負担になる。さらに、図 1 のように接続する端子数が多いほど接続は困難になり、益々設計検証の期間は増大してしまうことが考えられる。

I/O 端子を多用途での使用が困難： I/O 端子を FPGA 間の接続に用いすぎることによって、信号を観測するために利用することが難しくなってしまう。これにより、エミュレータとしての利便性が低下してしまふ恐れがある。

論理的等価性の問題： ASIC デザインを分割して FPGA に搭載した場合、搭載した回路が分割前の ASIC デザインと等価であるという保証が必要となる。

3. 高速シリアル通信を用いた EPGA エミュレータ

本章では、提案する Xilinx 社製 FPGA に搭載されている高速シリアル通信を利用した FPGA エミュレータの通信手法について述べる。

3.1 シリアル通信の有用性

従来パラレル通信では、バンド幅を上げるために 16bit から 32bit、32bit から 64bit へとビット幅の拡張や、転送周波数を上げるといった方法が取られてきた。しかし、FPGA 間の配線遅延のため信号線間に生じるタイミング・スキューやクロストーク等の問題により転送周波数を上げることが難しい。また、広いビット幅が必要な場合は FPGA の必要端子

数が多くなる．この場合，前述したタイミング・スキュー等の問題から，遅延を考慮した配線技術が要求される．それに対して，シリアル通信の場合は信号線が少ないためこれらの問題を緩和することが可能である．タイミング・スキューが発生しないうえに，信号線数が少ないことでクロストークの問題も減少するため，高周波数での動作が可能となる．さらに，通信する信号の数の増減した場合の対応も容易に行うことができる．

3.2 パラレル接続による FPGA エミュレータ

図 1 にパラレルで FPGA 間を接続した場合のエミュレータ図を示す．このように，FPGA 間の接続に用いる信号の数はとても多く，信号の遷移周波数を上げようとすると主に I/O バッファの遅延により上げることができない．また，図 2 にパラレル接続の回路で，何も考慮せずに分割した場合の回路分割図を示す．このように回路を分割すると分割回路間の通信を何度も行う必要がでてくる．したがって，通信遅延の影響や，各 FPGA 間のクロックスキュー等を十分に考慮する必要がある．

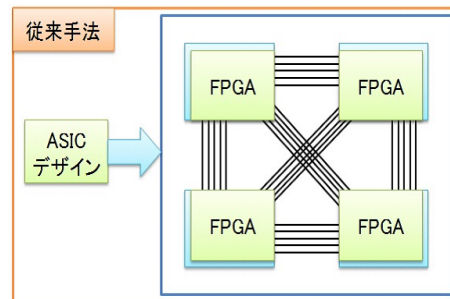


図 1 従来手法の FPGA 間接続

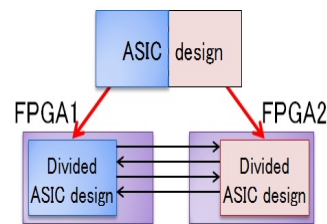


図 2 ASIC デザインの分割

3.3 シリアル通信による FPGA エミュレータ

図 3 にシリアル通信を用いた場合の FPGA 間接続図を示す．このように，通信をシリアル化することで配線を簡素化することが可能となる．シリアル通信は通信周波数を高速化できるだけでなく，リンク数を増やすことも容易である．そのため通信をシリアル化することでデータの通信を速く終了することができ，ASIC エミュレータの周波数を上げることが可能になる．また，従来のパラレル接続では通信に必要な端子数が不足する可能性があったが，高速シリアル通信 IP を用いて出力端子の仮想化を行うことで多ビット通信が可能となるため，I/O 端子数不足も緩和されると考えられる⁵⁾．

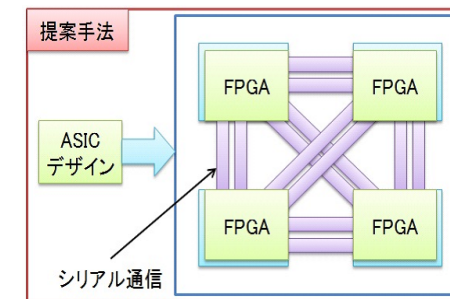


図 3 シリアル通信での FPGA 間接続

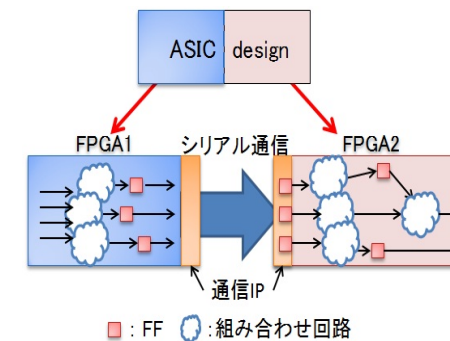


図 4 2 分割時の回路イメージ

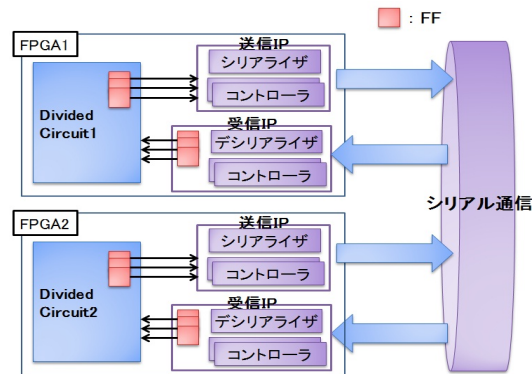


図5 2つのFPGAを用いたASICエミュレータ回路の概要

図4に、シリアル通信で接続する際の回路分割の制約に従った回路分割後の図を示す。分割された回路を高速シリアル通信で動作させ、高速なエミュレーションを実現するために、分割面にFF (Flip Flop) が並ぶように分割を行う。また、通信IPの受信部には送信側が送ったFFのデータを保存するためのFFが並ぶように配置する。このように論理回路内のFFに着目して分割を行えば、FF内のデータを最初にすべて通信した後に次クロックの状態遷移を行うようにすることで同期動作させることが可能なため、論理エミュレーションを行うことができる。

3.4 提案FPGAエミュレータの概要

図5に提案するFPGAエミュレータの概要を示す。分割されたASICデザインのデータ転送は、高速シリアル通信を用いる。また、Xilinx社製VirtexFPGAが搭載する高速シリアル通信ブロックであるRocketIOのMGTブロック⁶⁾を用いて通信を行う。RocketIOのMGTブロックは、さまざまなシリアル通信仕様に柔軟に対応している。なお、今回はXilinx社が提供するIPであるAuroraで用いられている8b/10bあるいは、64b/66bエンコード・デコード処理を行わない方法を選択した⁷⁾。それは、著者らが事前に行った調査から、複雑なエンコード・デコード処理を行うと、それに伴う通信レイテンシが大きくなりエミュレーション周波数を上げることができないことを確認したからである³⁾。そのため、特別なエンコード・デコード処理を行わず、データをそのまま転送する構成にしている。分割された回路に送信IPおよび受信IPを接続してシリアル通信を行う。提案するエミュレータでは、高速シリアル通信のクロックは通信IP用動作クロックをFPGA内の

DCM (Digital Clock Manager) で通倍したものを用いる。DCMとはクロックの周波数を変更するためのモジュールで、クロックを必要に応じて通倍、分周することが可能である。

3.5 提案エミュレータ詳細

図6に今回提案するFPGAエミュレータの詳細図を示す。FPGA2つに分割回路を搭載して動作させる。その際送信IPおよび受信IPを同時に搭載する。どちらのIPにもレジスタの制御とタイミングの制御のために2つのコントローラを備えている。

送信IP：分割された回路の分割面にFFが来るように分割されている。そのFFの中にデータが格納されるため、FFに格納されたデータをパラレルでシリアライザに入れ、シリアルデータに変換する。変換されたデータをシリアル通信を用いて送信する。また、データを送信する際には前述した通り、クロック (SEND_CLK) を同時に送信する。これは、データと同時にデータと同じ数のパルスとなるクロックを送信することで送っているデータが有効であることを示すためである。

受信IP：送られてきたシリアルデータをクロック (RX_CLK) のタイミングに準拠してデータをFFに格納していく。クロックのタイミングでデータを格納するため、送られてくるデータの最初のビットはクロックによって確認することができる。また、同時にカウンタを設けておき、送られてきたクロックが規定した個数のパルスを受け取ることによりデータの送信が終了したことを受信側で判断する。最後にFFのデータを分割回路に入れることで通信完了となるような動作となっている。

4. 評価・考察

シリアル通信を用いたASICエミュレータの開発・評価環境を表1に示す。

表1 開発・評価環境

開発ボード	ML605 Evaluation Platform
評価デバイス	Virtex-6 XC6VLX240T-1FF1156
設計ツール	Integrated Software Environment (ISE) 13.1
検証ツール	ISim 13.1
検証回路	itc99 ベンチマーク (b05)

本来の仮想化回路は図5に示す回路構成である。しかし、今回は提案システムの転送速度を評価し、転送ビット数が増加した場合の転送速度など見るため、送信データを同一FPGA

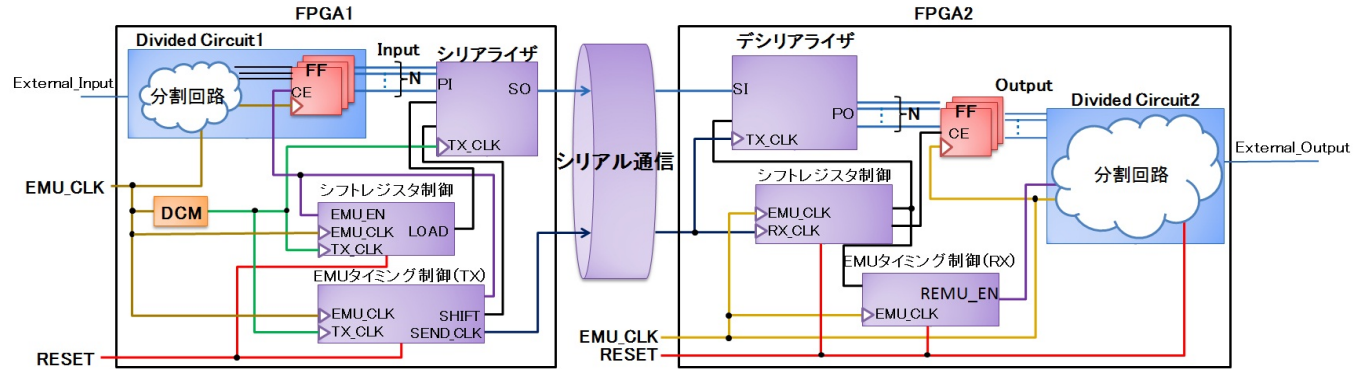


図 6 提案するエミュレータ全体のブロック図

に受信させるループバック回路で評価を行った．シリアル通信部分はループバックしており，単一 FPGA でシミュレーションを行っている．また，今回用いた itc99 ベンチマーク b05 は定期的に値を生成し，生成した値をメモリに書き込む回路である．回路規模はゲート数 998 個，FF 数 34 個の回路である．また，回路分割により複製されるゲート数は 383 となっており，分割した回路は 633 個および，771 個のゲートを持つ 2 つの回路に分割している．この分割は著者らが研究を行っている回路分割ツールを用いて分割を行った⁸⁾．この分割ツールでは，分割回路の分割面に FF が並ぶように分割を行い，入出力端子数およびゲート数の複製が増加しないように考慮した分割を行う．

4.1 評価結果

得られた b05 回路の提案エミュレータによるシミュレーション評価波形を図 7 に示す．図中① が送信 IP の動作に必要な信号の遷移である．また，図中② がデータ転送に必要な信号の遷移を示しており，③ が受信 IP に必要な信号の遷移を示している．まず①において，EMU_EN を有効にすることで FF の内容をシリアルライザ用のシフトレジスタの入力 PI としてコピーする．シリアルライザに LOAD を入力し，データの格納が完了したことを伝える．次に SHIFT 信号を出し逐次的に②のようにデータを送信する．同時に SEND_CLK も送信し，SEND_CLK 信号の立ち上りに合わせて受信側のデシリアライザに受信データを格納する．全ての転送データを受信した後，③で規定個数のクロックを受けることで Rshift 信号が生成される．Rshift 信号のタイミングで，PO より分割回路へデータの入力を行う．そ

の後 REMU_EN 信号を出しエミュレータ全体のクロックを進める．ここで，処理①と処理③を足し合わせたクロック数 5 が通信 IP に必要なクロック数となる．また，図中④のように，エミュレートする回路のクリティカルパス遅延分の時間を確保する．近年の FPGA は高速動作が可能であるため，分割回路の実装結果にもよるが，クリティカルパス遅延としては 20[ns] (50MHz 動作) 程度を確保すれば十分動作可能であると考えている．

4.2 考察

提案するエミュレータのエミュレーションクロック周波数についての考察を行う．エミュレーションクロックを f_{emu} ，高速シリアル通信のデータ転送レートを f_G ，通信 IP のベースクロックを f_{base} ，エミュレーション 1 周期に必要な時間を T_{emu} とすると，提案する論理エミュレーションの動作周波数は以下のように求められる．まず，通信 IP のベースクロックである f_{base} を X 倍に通倍したクロックを転送周波数 f_G として用いている (式 1)．

$$f_G[\text{bps}] = f_{base}[\text{MHz}] \times X \times 10^9 \quad (1)$$

次に，エミュレーション 1 周期に必要な時間である T_{emu} は式 (2) で求められる．

$$T_{emu} = \frac{5}{f_{base}[\text{Hz}]} + \frac{N}{f_G[\text{bps}]} + M \times 10^{-9}[\text{s}] \quad (2)$$

ここで，第 1 項の $5/f_{base}$ は通信 IP の動作にかかる時間を示している．これは送信 IP の動作に必要なレイテンシと，受信 IP の動作に必要なレイテンシを足し合わせたものである．次に，第 2 項の N/f_G は実際にデータ転送にかかる時間を示している．N はデータ転

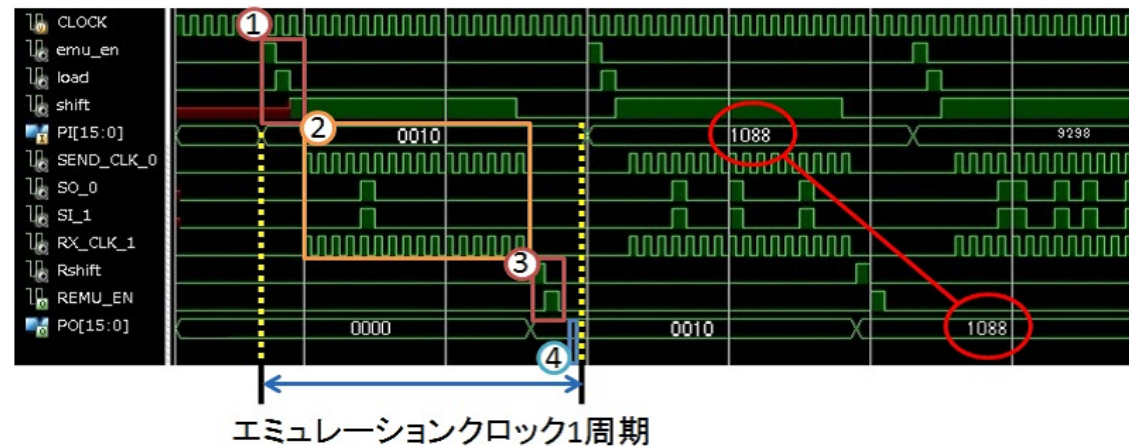


図 7 b05 のシミュレーション評価波形

送ビット数を示しており、回路によってデータ転送ビット数は異なるため、この値は可変となる。第 3 項の $M \times 10^{-9}$ は回路のクリティカルパス遅延を示しており、 f_{base} の何クロック分かで求める。回路によって必要なクリティカルパスは変化するのでパラメータとして取り扱えるようにしている。式 (1), (2) を考慮することでエミュレーションクロック周波数は式 (3) で表すことができる。

$$f_{emu}[\text{Hz}] = 1/T_{emu} \quad (3)$$

これらを踏まえたエミュレーション動作周波数の見積りを以下に示す。

表 2 通信 bit 長とエミュレーションクロックの見積り

転送レート (f_G)	転送ビット長 (N)			
	256bit	512bit	1,024bit	2,048bit
10Gbps	21.7MHz	13.9MHz	8.1MHz	4.4MHz
5Gbps	13.9MHz	8.1MHz	4.4MHz	2.3MHz
1Gbps	3.6MHz	1.9MHz	1.0MHz	0.5MHz

ここで、分割回路のクリティカルパス遅延の値は 20[ns] を想定している。現段階では f_G

を 10[GHz] および f_{base} を 100[MHz] とした場合 M は 2 となる。この場合、N が 2,048 ビットのデータを転送すると 4.4[MHz] のエミュレーション動作周波数が見込める。

今回はシミュレーションによる見積りのみであったため、今後実 FPGA 上に実装して評価を行い、どのくらいの通信レートでの論理エミュレーションが可能となるのかを実測していく予定である。

5. ま と め

本稿では FPGA 間の通信に高速シリアル通信を用いて、複数の FPGA を用いたエミュレータの性能向上手法を提案した。提案したシリアル通信を用いたシステムを用いることで、従来手法に比べ、端子数制約および送信データ量の面で有効であることを示した。また、200[MHz] のベースクロックで 2,048 ビットのデータを 10[Gbps] で通信させた場合 4.4[MHz] の動作周波数が見込めることがわかった。

参 考 文 献

- 1) 横溝憲治, “ASIC をターゲットに, ビヘイビア合成で作成した RTL コードを論理合成,” Design Wave Magazine, pp.60-67, Dec, 2005.
 - 2) 久木田透, 久我守弘, 末吉敏則, “FPGA における高速シリアル通信を用いた ASIC エミュレータの一検討,” 火の国情報シンポジウム 2010 論文集, B-1-1, Mar. 2010.
 - 3) 藪田敏生, 一ノ宮佳裕, 久我守弘, 末吉敏則, “FPGA における高速シリアル通信を用いた ASIC エミュレータ向け配線仮想化の検討,” 信学技報, vol.110, no.318, CPSY2010-33, pp.7-12, Nov, 2010.
 - 4) 磯村達樹, 稲木雅人, 高島康裕, 中村祐一, “時間多重 I/O を考慮した回路分割手法,” 信学技報, VLD2008-100, CPSY2008-62, RECONF2008-64(2009-01), pp.51-55, Jan, 2009.
 - 5) Jonathan William Babb, Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic Emulation, 1991.
 - 6) Xilinx Inc., Virtex-6 FPGA GTX Transceivers User Guide UG366, 2010.
 - 7) Xilinx Inc., LogiCORE IP Aurora 64B/66B v4.1 User Guide UG237, 2010.
 - 8) 高橋克昇, 佐伯亮祐, 久我守弘, 末吉敏則, “高速シリアル通信による FPGA ベース ASIC エミュレータ向け回路分割手法,” DA シンポジウム, 4B-1, Aug. 2011.
 - 9) Xilinx Inc., Virtex-6 ファミリ概要 DS150 (v2.3), 2011.
-