

平成 30 年 6 月 12 日現在

機関番号：62615

研究種目：挑戦的萌芽研究

研究期間：2015～2017

課題番号：15K12005

研究課題名(和文) 遷移型非同期式回路の新たな設計手法に関する研究

研究課題名(英文) New Approach to Design of Transition Signaling Asynchronous Circuits

研究代表者

米田 友洋 (Yoneda, Tomohiro)

国立情報学研究所・アーキテクチャ科学研究系・教授

研究者番号：30182851

交付決定額(研究期間全体)：(直接経費) 2,800,000円

研究成果の概要(和文)：遷移型非同期式回路は、要求・応答信号の、レベルではなく遷移(立ち上がり、立ち下がり)を用いるため、高速な回路が実現できるが、設計が難しく、その長所を活かすには非常に高い専門知識を必要とする。本研究では、複数のクロック入力を持つ、新しいフリップフロップに基づく遷移型制御回路のテンプレートを構築することで、特別な専門知識なしに容易に遷移型非同期式回路を実現できる、新しい設計手法を確立し、応用例として遷移型非同期式NoCルータを実現・評価した。また、発展研究として遷移型非同期式回路用再構成デバイスについて検討し、シミュレーションにより動作を確認するとともに、提案手法の有効性を確認した。

研究成果の概要(英文)：Transition signaling asynchronous circuits work based on the transitions of request and acknowledgement signals. Such a design methodology has potential to implement high performance circuits, but also requires specialized design techniques, which has prevented this approach from becoming popular among designers. In this research project, we utilize new and special flip-flops that have multiple clock inputs for the same outputs, and we provide new design method based on templates using them, in order to make it possible for even designers without specialized design techniques to easily implement transition signaling asynchronous circuits. An asynchronous NoC router has been designed for demonstrating the proposed method. Furthermore, this idea has been extended to an asynchronous reconfigurable device that contains those special flip-flops in a configurable logic block. Its naive version has been evaluated by simulation, and we have concluded that such an approach is promising.

研究分野：非同期式回路設計

キーワード：非同期式回路 遷移型 設計手法 新フリップフロップ NoCルータ 再構成可能デバイス

1. 研究開始当初の背景

(1) 現在、主に使われている同期式設計では、クロック分配のための配線面積増、消費電力増、電磁妨害等のグローバルクロックに起因する問題を避けるために、相当の工夫・改良が必要で、グローバルクロックの高速化にともしない設計上の大きな問題になりつつある。一方、グローバルなクロックシステムを用いない非同期式設計は、根本的にこのような問題が生じない。そのため、研究レベルのみならず、実用システムにも非同期式回路が用いられるようになってきた。例えば、Intel FM6000 イーサネット・スイッチで使われているメインチップはその 90%が非同期式回路で構成されている。

(2) 非同期式回路は、グローバルクロックを用いる代わりに、要求・応答の因果関係により実行制御を行う。その実現方法は大きく、要求・応答信号のレベル（電圧の高低）を用いたレベル型制御方式と、レベルではなく遷移（立ち上がり、立ち下がり）を用いた遷移型制御方式に分類できる。遷移型制御方式は要求・応答信号のレベルを 0 に戻す休止相が不要であるため、高速な回路が実現できる。一方、現実のゲートは電圧レベルに基づいて動作するため、遷移に基づく動作を実現するのは直感的ではなく、その長所を活かすには非常に高い専門知識を必要とした。

(3) このような理由で従来敬遠されていた遷移型制御方式ではあるが、非同期式回路のハンドシェイクに関わるオーバーヘッドを大きく削減できる可能性を有するため、最近その実装が試みられるようになってきた。上記イーサネット・スイッチを開発した Intel/USC(Univ. of South California)のグループは、遷移型制御回路を扱えるように、彼らの設計 CAD を改良しつつある。我々のグループでも、非同期式 NoC(Network-on-Chip)のルータを遷移型制御方式で実現し、ASIC チップを試作している[1]。しかし、現状では、遷移型制御方式の非同期式回路を設計するコストは大きいと言わざるを得ない。

[1] M. Imai and T. Yoneda: Improving Dependability and Performance of Fully Asynchronous On-chip Networks, Proc. IEEE Int. Symp. Asynchronous Circuits and Systems 2011, pp.65-76, (Apr., 2011).

2. 研究の目的

本研究では、遷移型非同期式回路の新たな設計手法を提案し、その可能性、実用性を評価する。従来の遷移型制御回路の設計方法は、(i)遷移信号のレベル信号への変換、(ii)変換後の仕様を信号遷移グラフ(STG: Signal Transition Graph)にて記述、(iii)合成可能な形への変換、(iv)合成ツールにて回路を生成、というステップを取っていた(図1左)。この方法では、(ii)および(iii)のステップが

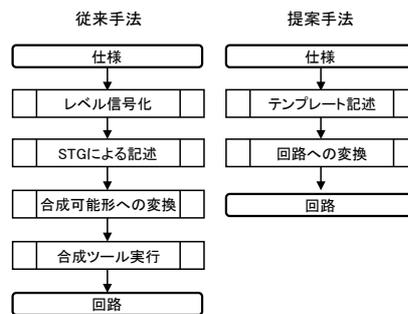


図1

特に非同期式回路設計の高度な専門知識を要した。また、(iv)のステップは状態空間探索に基づくため、中規模以上の回路の場合、その計算量が非常に大きくなるという問題点があった。さらに、(i)のステップにおいて、性能を追求した回路設計を行うと、レベル化を行う EXOR ゲート出力に比較的短いパルスが生じることがある。そのような短いパルスは、チップとして実装する際に遅延や負荷容量により消失しないように、配置・配線に細心の注意を必要とし、実装を難しくする要因の一つとなっていた。

一方、本研究では、(i)仕様に従って、テンプレートを組み合わせた記述を作成、(ii)テンプレートを直接回路へ変換、というステップで遷移型制御回路を生成できるようにする(図1右)。任意の遷移型制御回路の仕様を、その組み合わせとして記述できるようなテンプレートを開発すれば、非同期式回路設計の高度な専門知識を不要とした設計手法の実現が期待できる。また、テンプレート方式は置換に基づく処理が中心であり、中規模以上の回路も容易に扱える。

3. 研究の方法

(1) 遷移型非同期式回路の場合、(立ち上がり)と(立ち下がり)という信号の遷移が要求や応答というイベントを表す。しかし、データの処理には、有効や無効、あるいは属性などを表すために信号のレベル(値)が必要となり、遷移とレベルの相互変換が必要不可欠となる。従来は上述したように、このレベル変換に EXOR ゲート等を用い、アドホックな方法でレベル化された信号を扱っていた。例えば、「header=1 のときに iack が遷移すると gate 出力を 0 とし、tail=1 のときに iack が遷移すると gate 出力を 1 とする。ただし、初期化時 (rstb=0) には gate 出力は 1 とする。」という動作を行う制御回路は図2のようになる。この回路では、レベル化のために、補助信号である oreq を用い、EXOR ゲートでレベル化した後、header あるいは tail でゲートした信号 A, B を作る。それをもとに、STG を用いた非同期式回路設計方法により図2下部の回路を得ている。提案手法では、このレベル変換にエッジトリガ方式のフリップフロップを用いる。しかし、通常のエッジトリガフリップフロップ(図3左)は、立ち上がりあるいは立ち下がりの方にのみ反応し、また、ひとつの

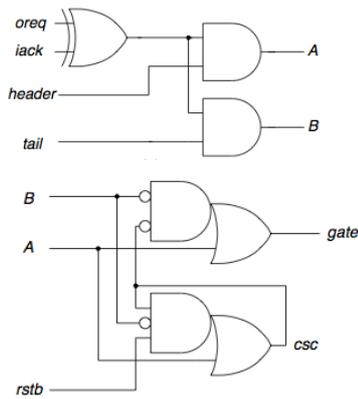


図 2

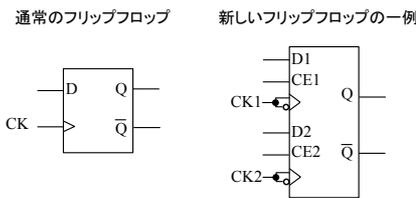


図 3

出力に伝搬する信号入力はただひとつである。これでは、さまざまな動作要求に対応する回路を構成することは容易ではない。そこで、立ち上がりと立ち下りの両方に反応し、かつ、複数の入力により、一つの出力を駆動できる、新しいフリップフロップ[2]を用いて遷移型非同期式回路を構成する。図3右の新フリップフロップの例では、一つの出力Qに対し、二つの入力とクロックおよびイネーブル入力を持つ。二つのクロックが同時にイネーブルされて入力されることは禁止されるが、そうでない場合、それぞれの入力で自由に一つの出力を制御できる。このため、例えば「条件Aのとき出力を1とし、条件Bのとき出力を0にする」を実現したい場合、条件Aと条件Bの駆動回路を独立に実現し、それぞれの入力に接続するだけで、所望の制御回路が実現できる。この「条件別に駆動回路を独立に構成できる点」がテンプレートによる回路合成手法の鍵となっている。この考え方を利用すると、上述の回路は図4のようにただひとつの新フリップフロップにより構成できる。このようなフリップフロップとしては、3以上の入力を持つもの、立ち上がりや立ち下りのみ動作するものなどのバリエーションを用意することは容易であり、それらにより、さまざまな条件に対応した制御回路を、わかりやすい記述から直接生成することが可能となる。

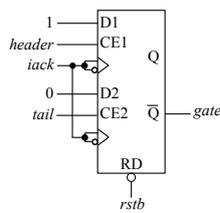
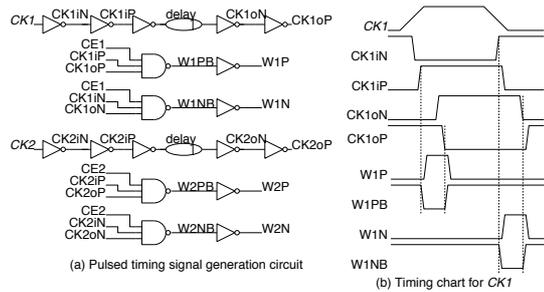


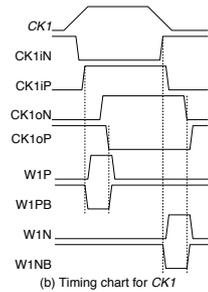
図 4

[2] 出願番号：特願 2013-134719，発明者：米田友洋，今井雅，発明の名称「フリップフロップ回路」，出願人：情報・システム研究機構，出願日：2013年6月27日

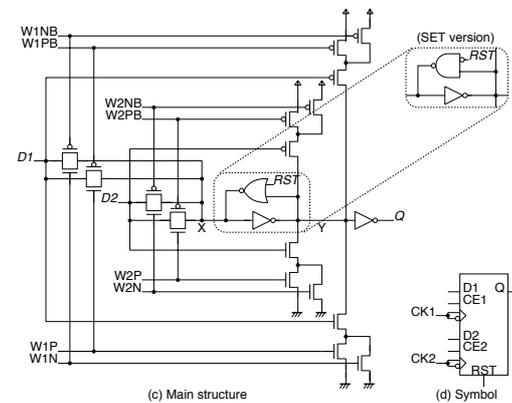
(2) 上記のような新フリップフロップは具体的には図5のように実現できる。また、その真理値表を表1に示す。前述したように、遷移信号のレベル化を行うとゲート出力に比較的短いパルスが生じることがある。このフリップフロップでも内部でそのようなパルスを発生させているが、それらはフリップフロップ内に閉じ込められ、フリップフロップ間の配線には現れない。このため、フリップフロップのセル設計をしっかりと行うことで、ユーザが行う配置・配線は大幅に簡易化される。



(a) Pulsed timing signal generation circuit



(b) Timing chart for CK1



(c) Main structure

(d) Symbol

図 5

D1	CE1	CK1	D2	CE2	CK2	RST	Q _n
a	1	↑	X	X	0/1	0	a
a	1	↓	X	X	0/1	0	a
X	1	0/1	X	X	0/1	0	Q _{n-1}
X	0	X	X	X	0/1	0	Q _{n-1}
X	X	0/1	b	1	↑	0	b
X	X	0/1	b	1	↓	0	b
X	X	0/1	X	1	0/1	0	Q _{n-1}
X	X	0/1	X	0	X	0	Q _{n-1}
X	X	X	X	X	X	1	0
X	1	↑/↓	X	1	↑/↓	0	(prohibited)

表 1

(3) 図4のような回路を生成するために次のような記述を定義する。

```
x = [0 | 1] if init_cond
  exp1 else if enable_cond1 @(edge_sel1 edge_cond1)
  exp2 else if enable_cond2 @(edge_sel2 edge_cond2)
  ...
```

init_cond, enable_condn, edge_condn, expn :: ブール式
edge_sel :: [posedge | negedge | bothedge]

これは出力信号 x の動作を規定するもので、init_cond が真となるタイミングで、x は非同期的に 0 または 1 にセットされる。また、enable_condn が真であるとき、edge_condn の立ち上がり (posedge)、立ち下がり (negedge)、

あるいは、変化時(**bothedge**)に、*expn* の値が *x* にストアされ、次の条件成立までそれを保持する。この記述を用いれば、前述の例題における信号 *gate* は次のように表される。

$$\begin{aligned} gate &= 1 \text{ if } \neg rstb \\ &0 \text{ else if header @ (bothedge iack)} \\ &1 \text{ else if tail @ (bothedge iack)} \end{aligned}$$

この記述から図 4 のような回路を生成するのは容易である (ただし、図 4 は非同期リセットを持つ素子を用いているため、D 入力の信号値が反転していることに注意する)。

(4) このような考え方を用いて、非同期式 NoC ルータを実際に設計し、必要となるテンプレートの選定を行う。また、提案手法により実現された回路の各種評価を行う。

4. 研究成果

(1) 非同期式 NoC ルータの仕様に基づき、その設計に要するテンプレートを定めた。その一部を図 6 に示す。TR-gate は遷移信号のゲーティングを行うテンプレートで、ある条件が成立するまで、信号の伝搬を停止するために用いる。レベル型制御回路では、このような目的のために AND ゲートが用いられるが、遷移型制御回路では、信号の遷移が意味を持つため AND ゲートは用いることができない。TR-gate は透過型ラッチにより実現できる。TR-MRG は、レベル型での OR ゲートに相当するもので、遷移信号をマージする。一方、遷移信号をフォークするためには、伝搬先毎に次の遷移のためのレベルを覚えておく必要がある。そのためトグルフリップフロップを用いて、TR-DEMUX のように実現する。そのほか、レベル信号を扱う必要もあるので、いくつかのテンプレートを用意する (Level-LT, LVL-gate 等)。

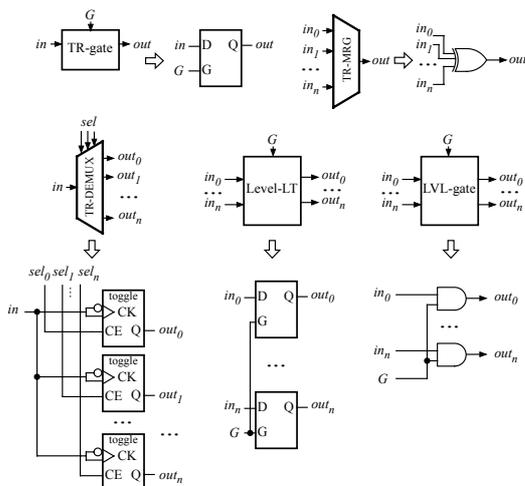


図 6

(2) 本研究の評価に用いた非同期式 NoC ルータの全体構成を図 7 に示す。このルータは 5 つの入力チャンネル (IC)、クロスバ入力部 (CBIN)、クロスバ出力部 (CBOU) からなる。IC

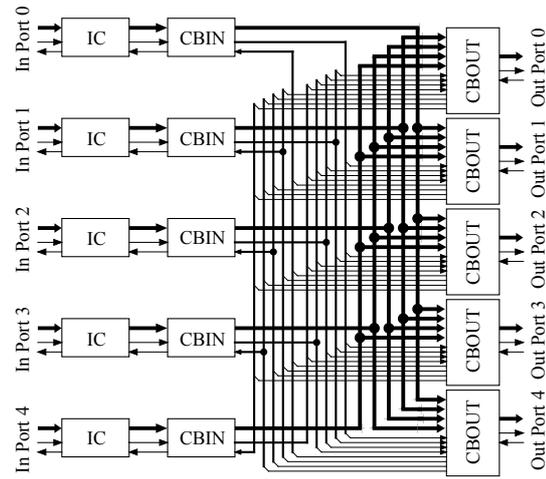


図 7

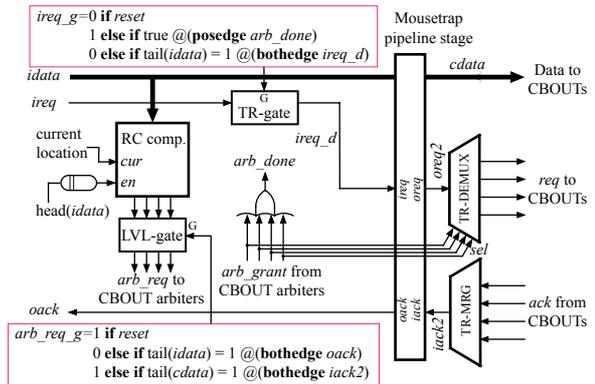


図 8

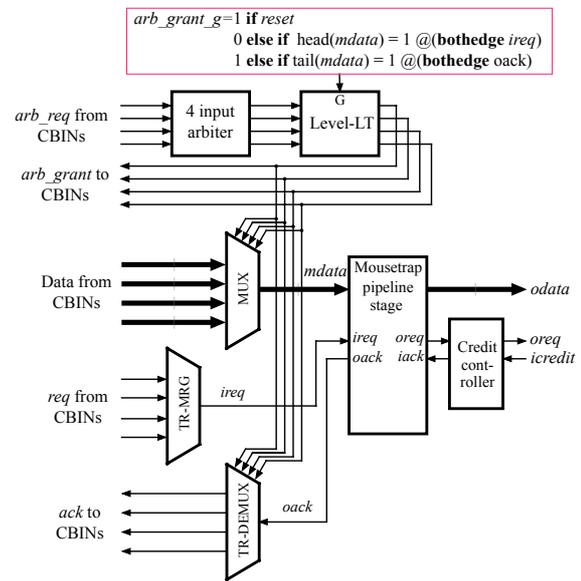


図 9

は、基本的な線形非同期式パイプラインで構成したもの (Linear FIFO) と、リング式の非同期式パイプラインで構成したもの (Circular FIFO) を用いたが、その詳細は省略する。CBIN は図 8 のように構成される。特に TR-gate と LVL-gate は遷移型制御回路特有の制御を行う部分であり、その G 入力には前述の記述を用いて簡潔かつ、直感的に記述できている。また、図 9 に CBOU の構成を示す。

		Synchronous		Asynchronous	
		without gated-clk	with gated-clk	Linear FIFO	Circular FIFO
Synthesized	cmb.	469747	305791	234044	507092
	non-cmb.	432727	439610	353349	380568
	macro			87208	87208
	sum	902474	745401	674601	974868
1mm place & route	cmb.	782347	534905	234715	508403
	non-cmb.	433339	440119	353349	380568
	macro			87208	87208
	sum	1215686	975024	675272	976179
2mm place & route	cmb.	712054	559544	235836	517729
	non-cmb.	432773	440011	353349	380568
	macro			87208	87208
	sum	1144827	999555	676393	985505

表 2

	Synchronous		Asynchronous	
	without gated-clk	with gated-clk	Linear FIFO	Circular FIFO
1mm place&route	4.74	4.86	4.84	4.22
2mm place&route	5.48	6.74	5.13	4.43

表 3

(3) 提案手法の評価のために、同等の同期式ルータを設計し、130nm Bulk CMOS テクノロジーを用いて面積、レイテンシ、スループット、および消費電力を比較した。以後の図、表ではこの同期式ルータを Synchronous、提案手法で設計した非同期式ルータを Asynchronous と表記する。NoC 構成は 4×4 とし、フロアプランとして、ルータを 1mm 間隔で並べたものと、2mm 間隔で並べたものを用いた。表 2 に面積比較 (単位は μm^2) を示す。cmb. は組み合わせ回路部を、non-cmb. はフリップフロップ・ラッチ等を、macro は非同期式回路用素子マクロを、それぞれ表す。非同期式ルータはフリップフロップではなく透過型ラッチベースの設計となるため、面積が小さくなる傾向がある。ただし、Circular FIFO の面積は大きい。表 3 にルータのレイテンシを示す (単位は ns)。図 10 にはスループットに相当する平均フリットレイテンシを示す (上段が 1mm 間隔、下段が 2mm 間隔)。これより、トラフィック量が少ない状況では、非同期式ルータが高性能であるが、ネットワークが飽和に近く高トラフィック時では同期式ルータが効率よく処理で

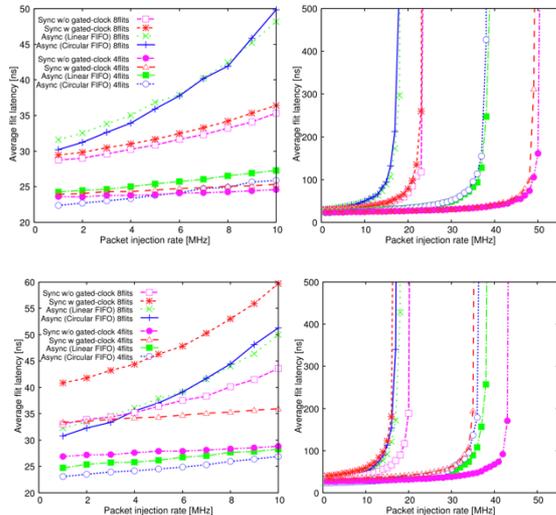


図 10

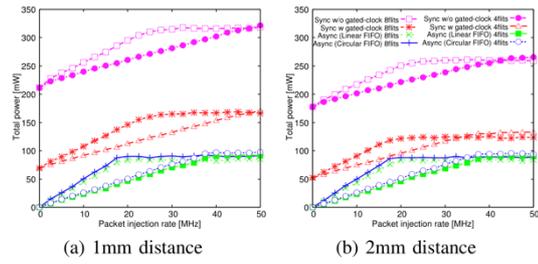


図 11

きるようになる。最後に、図 11 にそれぞれの消費電力を示す。同期式ルータはクロックゲーティングにより大きく電力消費を改善できるが、非同期式ルータの消費電力はさらにそれよりも少ないことがわかる。

結論として、提案手法を用いることで、高度な専門知識を必要とすることなく、性能の高い遷移型非同期式回路を設計することができるようになったと考える。

(4) 発展研究として、この新フリップフロップを基本要素として含む非同期式回路用再構成デバイスについて検討した。大域非同期局所同期方式の実現に特化することでアプリケーションを広めることができるのではと考え、データパス実現に特化したアーキテクチャを考案した。これにより、図 12 に示すように小～中規模の複数の同期式回路の「島」の間に、非同期式再構成可能デバイスを敷き詰める形の相互接続部が実現でき、異なるクロックドメインの接続の際に生じる様々な問題を本質的に回避できるとともに、フレキシブルかつ高効率な相互接続が期待できる。

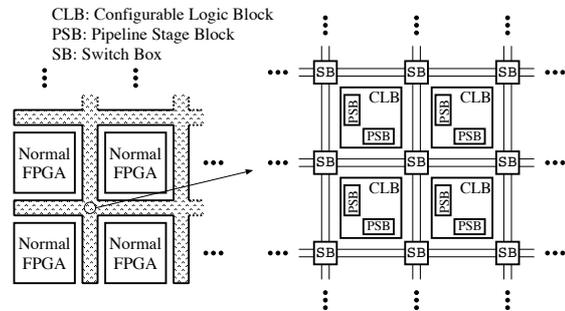


図 12

一方、再構成可能化に伴うオーバーヘッドについて検討し、いくつかの評価実験を行なった結果、特に非同期式回路を実現する本方式では、再構成可能部の基本単位 (CLB: Configurable Logic Block) を大きくした、粗粒度構成を取ることが適しているという結論を得た。例えば、図 12 の CLB にはデータバス用ワードレジスタを 4 本程度と制御信号用ルックアップテーブル等が 96 個程度含まれ、SB (Switch Box: 配線を設定して CLB を接続する) 間の配線はデータバス用と制御信号用に分離する構成を考えている。

ただし、粗粒度構成に対する配置・配線処理は複雑になることが一般的に知られている。検討中のアーキテクチャでは、配線を選択するコンフィギュレーション用マルチプレクサ

の入力数を現実的なものに抑えるため、配線可能性に制約を与える。このためさらに問題が複雑化する。現状では、これらの制約を充足可能性問題として定式化し、SAT ソルバを用いて解くというアプローチを試している。ユーザ回路が小さい場合には、このアプローチでも有効であることがわかっているが、大きなユーザ回路には対応できない可能性が高い。グリーディ算法を併用して、部分的に SAT ソルバを使う方法を検討し、実用的な規模のユーザ回路に対応できるように検討中である。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 1 件)

① Zhen Zhang, Wendelin Serwe, Jian Wu, Tomohiro Yoneda, Hao Zheng, Chris Myers: An Improved Fault-Tolerant Routing Algorithm for a Network-on-Chip Derived with Formal Analysis, Journal of Science of Computer Programming, 2016 年 3 月.

[学会発表] (計 2 件)

① Masashi Imai, Thiem Van Chu, Kenji Kise, Tomohiro Yoneda: The synchronous vs. asynchronous NoC routers: an apple-to-apple comparison between synchronous and transition signaling asynchronous designs, IEEE Conference Proceedings, 2016, (NOCS), 2016 年 8 月.

② Naoya Onizawa, Masashi Imai, Takahiro Hanyu, Tomohiro Yoneda: MTJ-Based Asynchronous Circuits for Re-initialization Free Computing against Power Failure, Proc. of ASYNC 2017, pp.118-125, 2017 年 5 月.

6. 研究組織

(1) 研究代表者

米田 友洋 (YONEDA, Tomohiro)

国立情報学研究所・アーキテクチャ科学研究系・教授

研究者番号:30182851