

# SDI モデルに基づく局所同期型非同期式 VLSI 設計方式

今井 雅<sup>†</sup> Metehan Özcan<sup>†</sup> 南谷 崇<sup>†</sup>

本論文では、SDI ( Scalable-Delay-Insensitive ) モデルに基づく非同期式システムの一設計方式として、要求-応答プロトコルに基づいて生成されたローカルタイミング信号を利用する非同期式 VLSI 設計方式を提案する。本設計方式では、セルフリセット構造を持つローカルタイミング信号生成回路を利用して要求-応答オーバーヘッドの小さい回路を実現する。また、入力データに依存して処理遅延が大きく変動する場合、繰り返し実行されるレジスタ間データ転送処理遅延の平均値で性能を評価することができ、高速化を図ることができる。さらに、タイミング信号のみ非同期で実装する設計方式であるため、制御回路とデータパスに分離してそれぞれの設計モデル・設計方式に応じた仕様記述を行わなければならない従来の非同期式システム設計手法と比較して、設計が容易である。

## An SDI Model Based Design Methodology for Locally-timed Asynchronous Circuits

MASASHI IMAI,<sup>†</sup> METEHAN ÖZCAN<sup>†</sup> and TAKASHI NANYA<sup>†</sup>

In this paper, an SDI (Scalable-Delay-Insensitive) model based design methodology for locally-timed asynchronous circuits is presented. In this methodology, register transfer level circuits are synchronized by local request-acknowledgment signals when delay values can be properly estimated and utilized. By using self-resetting circuits as local-timing generator circuits, request-acknowledgment overhead is minimized. In addition, further performance gain can be obtained by exploiting average-case behavior for input-dependent loop-structured implementations. Since only timing signals are implemented as asynchronous circuits, logic synthesis is relatively easier than the traditional asynchronous synthesis methodologies in which the general circuit is divided into data-path and control circuits and separate modeling and synthesis methods are applied on both parts.

### 1. はじめに

半導体集積回路製造技術の進歩にともなう素子の微細化とシステムの大規模化が進むに従い、配線遅延が支配的になるため、チップ全域にクロック信号を分配する同期式システムでは、期待されるスイッチング速度の向上がシステムの性能向上に反映されなくなると予測されている<sup>1)</sup>。この問題を解決するための1つの手段が、事象生起の因果関係を駆動原理とする非同期式システムによる実現である。非同期式システムはグローバルクロックの制約がなく、プロセス微細化にともない減少するゲート遅延でサイクルタイムが決定されるカスケード ALU アーキテクチャ<sup>2)</sup>の採用等により、素子の高速性をそのまま直線的にシステムの性能向上に反映しうる。また、その動作原理から Delay

Insensitivity 特性があり、予測不能な遅延変動に対する高いロバスト性が得られるとともに、モジュラー設計が容易となる利点がある。

非同期式システム設計に関しては様々な研究が行われており、遅延モデルや回路構成方式の異なる様々な設計方式<sup>3)</sup>が提案され、ツールの実装も行われている。主なものとして、以下があげられる。1つ目は、Huffman モデル<sup>4)</sup>と呼ばれる遅延仮定に基づいた非同期式順序回路の設計手法を基として、ある一定時間内の状態遷移をまとまった遷移として考え、その遷移を引き起こす入力遷移列を Burst として指定する Burst Mode<sup>5)</sup>に基づいた設計手法と設計支援ツール ACK<sup>8)</sup>、および XBM ( Extended Burst-mode ) モデルに基づいた論理合成ツール 3D<sup>6)</sup>等である。これらのモデルに基づいた設計では、予測できない入力変化タイミングはないものと仮定しているが、現実のシステムとしては回路要素が環境と複雑な相互作用を及ぼしながら動作するため、この仮定は必ずしも成立しない。2つ目は、

<sup>†</sup> 東京大学先端科学技術研究センター  
Research Center for Advanced Science and Technology,  
The University of Tokyo

Speed Independent (SI) 回路理論<sup>7)</sup>に基づき、要求-応答プロトコルによりシステムを動作させるモデルである。この設計モデルでは、仕様記述の段階では“内部状態”という概念を持たない。仕様の記述方法として、Petri-net や STG (Signal Transition Graph) 等のグラフ表現に基づいた方法があり、それらを入力仕様として論理合成を行うアルゴリズムの研究とツール Petrify 等の実装が行われている<sup>9)~11)</sup>。Petri-net や STG を仕様記述とする合成手法では、論理合成段階でいったん状態遷移図を作成し、それに基づいてハザードフリー構成を求める。この手法では、使用されている変数に対して指数関数オーダの記憶領域を必要とするため、大規模な回路では状態爆発を引き起こしやすく、論理合成が実質的に不可能となる場合がある。さらに、回路要素の遅延は Speed Independent (SI) モデルに基づいているため、要求-応答プロトコルにともなうオーバーヘッドが大きい。3 つ目は、マイクロプロセッサ等の大規模な回路を対象とした設計モデルであり、CHP (Communicating Hardware Processes) 言語を用いて動作仕様をプロセスとして記述することにより制御回路を合成する方法<sup>12)</sup>や、仕様を Tangram により記述し、中間表現 Handshake circuit に変換した後に直接制御回路を合成する方法<sup>13)</sup>、および Tangram と同様の Balsa と呼ばれる言語を用いた非同期式 VLSI 設計方式とその支援システム<sup>14),15)</sup>等がある。これらに基づいた設計では、プロセス記述という VLSI 設計者にとって馴染みの薄いものを使用する必要があるため、仕様を記述するのが困難場合がある。Balsa システムは Balsa-C と呼ばれる C 言語類似の高位記述言語からの合成環境、シミュレーション環境等が提供されているが、ハンドシェイク回路を意識して仕様を記述する必要があり、同期式システム設計者にとっては設計が困難である。これら従来の非同期式システム設計手法は、いずれも事象生起の因果関係に基づいて要求-応答信号の制御を行う非同期式制御回路と、制御回路から出力される要求信号に基づいて論理動作を開始し、その完了を応答信号によって制御回路に通知するデータバスモジュール等から構成されるモデルを対象としている。また、いずれも Delay Insensitivity 特性を重視し、それぞれの遅延モデルで規定された範囲内の遅延変動が生じても正しく動作するように回路を構成するため、速度性能に対するオーバーヘッドが大きくなりやすい。

Delay Insensitivity 特性を維持しつつ、速度性能に対するオーバーヘッドの小さい非同期式システムを実現するための 1 つの手段が、遅延情報を利用することがで

きる局所領域内において、Scalable-Delay-Insensitive (SDI) モデル<sup>16)</sup>に基づいた設計を行うことである。SDI モデルに基づいた回路設計例には、32 ビット非同期式マイクロプロセッサ TITAC-2<sup>16)</sup>や DDL 回路を用いた非同期式パイプライン<sup>17)</sup>がある。また、SDI モデルに基づいて設計された回路のタイミング検証ツールの実装<sup>18)</sup>も行われている。

一方、高速性を重視し、遅延は既知と仮定した高速な非同期式システムの回路構成がいくつか提案されている。主なものとして、ダイナミック CMOS 論理回路を用いた高スループット非同期式パイプライン<sup>19)</sup>、GasP と呼ばれるパルスのなタイミング信号を生成する回路を用いた FIFO 回路構成<sup>20)</sup>がある。GasP 回路はトランジスタサイズを調整してすべての素子遅延を等しくすることを仮定しており、素子 3 個で構成される最小のループ構造のサイクルタイムで動作する高速な FIFO を実現することができる<sup>20)</sup>。また、1 入力 1 出力の FIFO のほか、多入力多出力、選択分岐、アービトレーション等の機能を持った FIFO の回路構成も提案されている<sup>20),21)</sup>。これらの回路構成を用いた非同期式システムは従来の非同期式システム設計方式と異なり、タイミング信号のみ要求-応答プロトコルに基づいて実現し、データバスはそのローカルなタイミング信号に同期して制御を行うモデルとなっている。このようなシステムを本論文では局所同期型非同期式 VLSI と呼ぶ。局所同期型非同期式 VLSI は、ローカルタイミング信号としてクロック信号を用いる GALS (Globally-Asynchronous Locally-Synchronous) システム<sup>23),24)</sup>と比較して、クロック信号のように最悪遅延で制約されないため、遅延が変動するシステムでは、繰り返し実行される演算処理の平均値で速度性能を評価することができる。さらに、データバスのタイミング制約は同期式システムと同様にクリティカルバスとタイミング信号間の制約のみとなるため、データバスの設計に関して同期式システム設計支援 CAD 環境を有効利用することができる。

本論文では、SDI モデルに基づいた非同期式 VLSI の一設計方式として、局所同期型非同期式 VLSI 設計方式を提案する。次章では、設計方式の前提となる遅延モデルについて述べる。3 章では、基本回路構成モデルとその回路が SDI モデルに基づいて正しく動作するための遅延制約等を示す。4 章ではシステム全体の設計方式について述べ、5 章では設計例と性能を示し、6 章でまとめる。

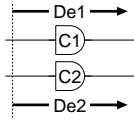


図1 予測遅延

Fig. 1 Estimated delay.

2. SDI モデル

非同期システム設計では、論理ゲートや配線の遅延に関して設ける仮定、すなわち遅延モデルが重要な役割を果たす。論理ゲートや配線の遅延の大きさは有限であるが上限は未知とする SI モデル等のように遅延変動に対して悲観的な遅延モデルに基づいた設計では、現実には起こりそうもない遅延変動が生じても正しい動作を保証するため、要求-応答プロトコルにもなうオーバーヘッドが大きく、十分な速度性能を得ることが困難となる。一方、遅延変動に対して楽観的な遅延モデルの基で設計された回路は、環境変動や設計時のプロセス変動等により誤動作を引き起こしやすい。これらの遅延モデルに対して、より現実的な仮定として、チップ製造後の論理ゲートや配線等の回路要素の遅延はそれぞれが独立に変動するのではなく、ほぼ一律に変動することに着目し、遅延変動率の相対値には上限および下限があると仮定したモデルが SDI モデルである。

SDI モデルの定義は以下のとおりである<sup>16),22)</sup>「ある回路要素  $C$  に対して、設計者が設計段階で予測した遅延を  $De$  とし、システムの生涯を通じて起こりうる実際の遅延を  $Da(t)$  とする。このとき、 $R(t) = Da(t)/De$  は回路要素  $C$  の時刻  $t$  における遅延変動率を表す。任意の 2 つの回路要素  $C1$  と  $C2$  の時刻  $t$  における遅延変動率をそれぞれ  $R1(t)$ 、 $R2(t)$  とすると、 $V(t) = R2(t)/R1(t)$  は時刻  $t$  における相対遅延変動率を表す。このとき、回路には定数  $K$  ( $K \geq 1$ ) が存在し、任意の 2 つの回路要素の間の相対遅延変動率  $V(t)$  に関して、システムの生涯を通じて  $1/K \leq V(t) \leq K$  がつねに成り立つ」

SDI モデルで仮定している遅延変動について以下に述べる。ある回路要素  $C1$ 、 $C2$  に対し、予測遅延がそれぞれ  $De1$ 、 $De2$  であり、 $De1 = De2$  であることを仮定する(図 1)。また、システムの生涯を通じた任意の時点で、その回路要素がとりうる実際の遅延値をそれぞれ  $Da1$ 、 $Da2$  とする。このとき、それぞれの回路要素の遅延変動率の大きさによって、実際の遅延値  $Da1$ 、 $Da2$  は様々な値をとりうる。

個々の回路要素の遅延変動率が等しい、すなわち相

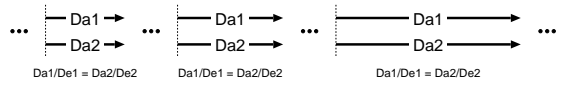


図 2 予測遅延が等しい場合の  $V = 1$  のときの実遅延値  
Fig. 2 Actual delay (1).

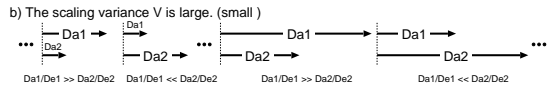
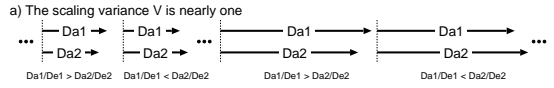


図 3 予測遅延が等しい場合の  $V \neq 1$  のときの実遅延値  
Fig. 3 Actual delay (2).

対遅延変動率  $V = 1$  のとき、実際の遅延値は図 2 に示すように、必ず同じ遅延値となる。ただし、遅延変動の大きさは有限であるが上限は未知であり、予測遅延に対して小さくなることもあれば大きくなることもある。

各回路要素の遅延変動率がそれぞれ異なる場合、遅延変動率が近い値であるか、掛け離れた値であるかによって、実遅延のとりうる値が大きく異なる。各回路要素の遅延変動率が近い値、すなわち相対遅延変動率  $V$  が 1 に近い場合、図 3 a) に示すように、実遅延もほぼ等しいものとなる。一方、各回路要素の遅延変動率が大きく異なる場合、すなわち相対遅延変動率  $V$  が 1 以上の大きい値、あるいは 1 未満の小さい正数値(分子が 1 で分母が大きな値)である場合、図 3 b) に示すように、回路要素がとりうる実遅延はその差が大きいものとなる。ただし、いずれの場合も遅延変動率はそれぞれ独立であるため、 $Da1$  が  $Da2$  よりも大きくなることもあれば、その逆となることもありうる。定義から相対遅延変動率  $V = R2/R1$  であり、遅延変動率  $R2$  が  $R1$  よりも大きい場合は  $V$  が 1 以上で大きな値となり、その逆の場合は 1 未満の小さい正数値となる。

SDI モデルに基づいた設計では、相対遅延変動率が 1 に近いことが予測される場合は、相対遅延変動率の上限、下限を規定する定数  $K$  ( $K \geq 1$ ) を 1 に近い値として設計時に与えることができる。一方、個々の回路要素の遅延変動率がそれぞれ大きく異なり、相対遅延変動率が 1 より大きな値、あるいは 1 未満の小さい正数値をとることが予測される場合は、定数  $K$  を大きな値として設計時に与えなければならない。定数  $K$  の値は設計経験、使用テクノロジー、製造プロセス等により決定される値である。本論文では定数  $K$  は設計

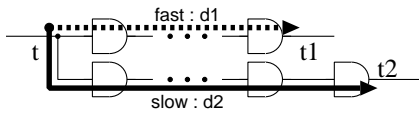


図4 SDIモデルに基づいた回路実現の遅延制約

Fig. 4 Timing constraint in SDI model based circuits.

時に与えられるものと仮定する。

非同期システム設計では、データパスにおけるデータ信号とその安定を示す完了信号との関係のように、順序関係を保証しなければならない信号遷移が存在する。SDIモデルに基づいた設計では、回路中の信号遷移  $t_1$  が信号遷移  $t_2$  よりも早く生じるように仕様で定められている場合、図4に示すように、信号遷移  $t_1, t_2$  の共通原因となる信号遷移  $t$  からの各パスの予測遅延値をそれぞれ  $d_1, d_2$  としたとき、 $K \cdot d_1 < d_2$  が成り立つように回路を構成することで、任意の時刻における信号遷移  $t_1$  と  $t_2$  の順序関係が保証される<sup>22)</sup>。

実際の設計では、すべての信号遷移間で、共通遷移元と信号遷移の順序関係を求め、不等式を満たすように設計する必要はない。順序関係を保証する信号遷移間に関してのみ、共通信号遷移元からのそれぞれのパスの遅延が不等式を満たせばよい。たとえば要求信号を共通の信号遷移元とした場合、ソースレジスタからデスティネーションレジスタまでの演算処理のパスが速いパスであり、その演算処理の完了を示す信号を生成するまでのパスが遅いパスとなる。

### 3. 設計モデル

#### 3.1 基本回路構成モデル

局所同期型非同期式 VLSI の基本となる回路構成モデルを図5に示す。なお、図5のように、書き込み要求信号 ( $\_wr$ ) と読み出し要求信号 ( $rr$ ) を入力とし、書き込み応答信号 ( $\_wa$ ) と読み出し応答信号 ( $\_ra$ ) およびローカルタイミング信号を生成する回路と、そのローカルタイミング信号に同期して値が更新される記憶素子および組合せ回路を持つ構成を「基本モジュール」と呼ぶものとする。また、先頭に「 $\_$ 」が付加された信号名は負論理の信号であることを表す。システム全体は基本モジュールを結合することにより実現される。ある基本モジュールに対して、入力側に接続された基本モジュールの読み出し応答信号 ( $\_ra$ ) が書き込み要求信号 ( $\_wr$ ) として接続され、書き込み応答信号 ( $\_wa$ ) が入力側基本モジュールの読み出し要求信号 ( $rr$ ) として接続される。出力側も同様で

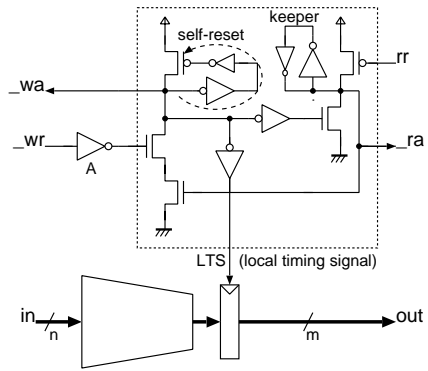


図5 基本モジュールの構成

Fig. 5 Structure of the basic module.

ある。

図5において、ローカルタイミング信号 (LTS) 生成回路は、高速な FIFO を実現する GasP 回路<sup>20)</sup> に対し、回路構成自体は変更せず、書き込み側・読み出し側それぞれで4相ハンドシェイクを実現するように構成単位を変更したものであり、基本モジュールは容量1の非同期 FIFO として動作する。GasP FIFO 回路では、トランジスタサイズを調整することによりすべての素子遅延を等しくし、正しく動作することを保証している。しかしながら、実際の回路構成において、すべての素子遅延を等しくするためにはフルカスタムでレイアウトをする必要があり、設計が困難である。また、データパスもパルスのローカルタイミング信号の生成間隔に合わせて細かく分割する必要がある。そこで、遅延モデルをSDIモデルに基づいたものとし、構成単位を図5ととらえ、破線で囲った回路を基本セルとして設計し、スタンダードセル方式でデータパスの遅延に合わせたローカルタイミング信号を生成することができるようにする。これにより設計の容易化が図れる。以降、破線で囲われたブロックは1つの基本セルとして構成することを表すものとする。

基本モジュールがSDIモデルに基づいて正しく動作するために満たさなければならない制約を図6に示す。実線で示す遅いパスの遅延が破線で示す速いパスの遅延の  $K$  倍より大きくなるように回路を構成する。また、図6において、「 $-$ 」は立ち下がり遷移、「 $+$ 」は立ち上がり遷移であることを表す。図6(a)はセルフリセットの方が遅くなければならない制約であり、これが満たされないと keeper (図5) が保持している値が遷移しないため、データの消失が生じる。また、slow1のセルフリセットの方が遅く遷移しないと、VDDからGNDまでショートし、故障の原因となる。図6(b)はセルフリセットの方が遅くなければならない

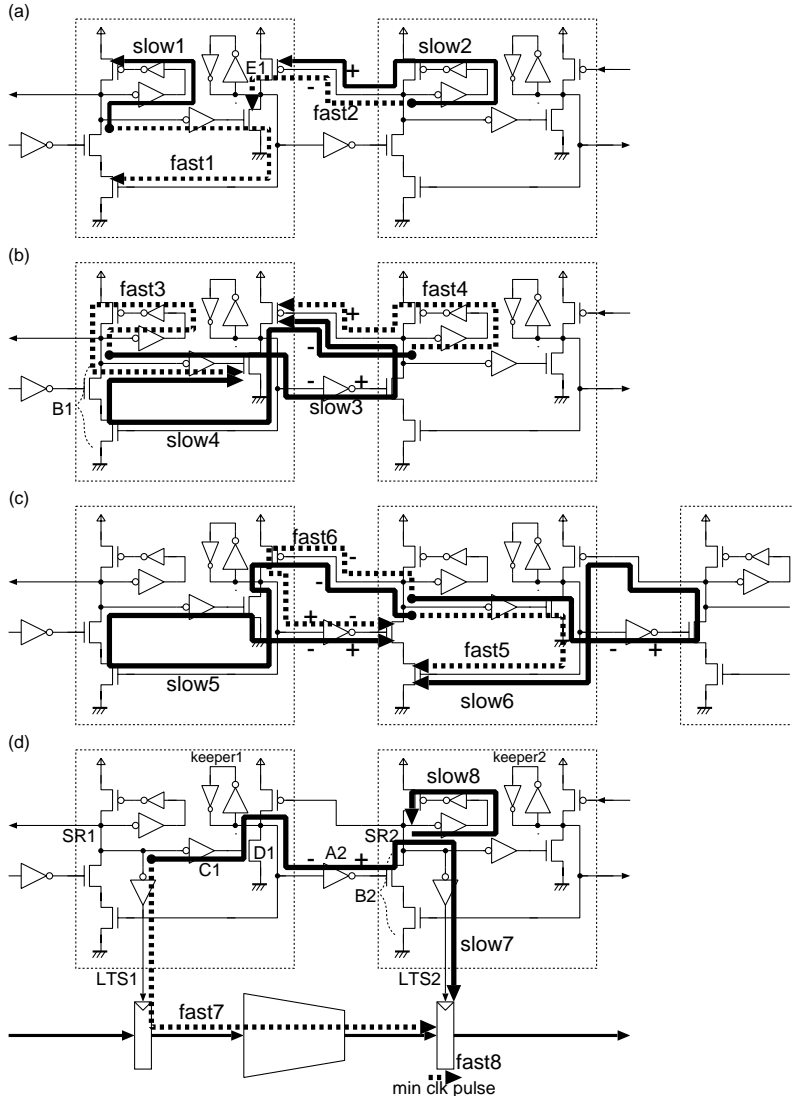


図6 SDIモデルに基づいた設計で満たさなければならない遅延制約  
Fig. 6 Timing constraints to be observed.

い制約であり、これが満たされないと keeper への入力となる INV 構造で VDD から GND までのパスが通り、ショートする。図 6 (c) はデータの二度読み・二度書きを防ぐための制約である。図 6 (d) は、ローカルタイミング信号生成回路とそれに同期して値の更新等の制御を行うデータパスとの遅延関係である。セルフリセットの開始を始点として、遅いパス slow7 の遅延が fast7 として示すデータパスの遅延の  $K$  倍よりも大きくなるように回路を構成する。また、セルフリセットの遅延が記憶素子の最小クロックパルス幅の  $K$  倍よりも大きくなるように回路を構成する。

遅延の調整はセル設計段階で行うものと、論理設計・

レイアウト設計段階で行うものとに分けて考えることができる。図 6 において、fast1-slow1 の関係は完全にセル内に閉じた関係であるため、セルの設計の際に遅延調整を行うことで容易に大小関係式を満たすことができる。また、fast8-slow8 の関係も、あらかじめ定められた最小クロックパルス幅が速いパスであり、セル設計段階で調整することができる。slow2 と fast4 は同一のパスであるため、 $K^2 \cdot \text{fast2} < K \cdot \text{slow2} = K \cdot \text{fast4} < \text{slow4}$  となるように図 6 (a) E1 の PMOS のサイズを大きく、(b) の B1 の NMOS のサイズを小さくセルを設計する。また、fast3-slow3, fast5-slow5, fast6-slow6, fast7-slow7 の関係はいずれも基本モジュール間のインパー

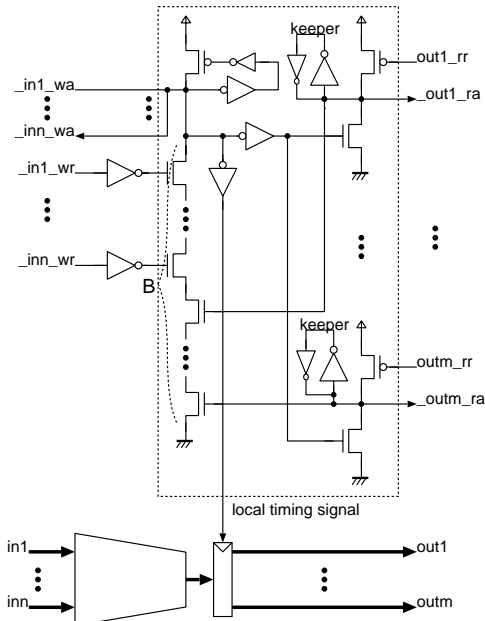


図7 n 入力, m 出力を持つ場合の回路構成モデル

Fig.7 Structure of the basic module for n-inputs and m-outputs.

データの立ち上がり遅延を論理設計・レイアウト段階で調整することで、大小関係性を満たすように設計することができる。データパスの遅延の方が支配的である場合、基本モジュール間のインバータの立ち上がり遅延は以下のように求められる。組合せ回路の遅延を  $T_e$ 、記憶素子のセットアップタイムを  $T_{setup}$ 、記憶素子の書き込み遅延を  $T_w$ 、ローカルタイミング信号を生成するインバータの遅延を  $T_{inv}$  とすると、

$$K \cdot (T_{inv} + T_w + T_e + T_{setup}) < T_{C1+} + T_{D1} + T_{A2+} + T_{B2} + T_{inv} \quad (1)$$

を満たすようにインバータ A2 の立ち上がり遅延  $T_{A2+}$  を調整する。これにより、ローカルタイミング信号 LTS1 が 1 に遷移してから、次のローカルタイミング信号 LTS2 が 1 に遷移するまでの遅延をデータパスの遅延に合わせたものとする事ができる。

### 3.2 複数の入出力を持つ基本回路構成モデル

図5に示した基本モジュールは1入力1出力の回路構成であり、単純なFIFOを表していた。しかしながら、一般的な回路構成では、異なるローカルタイミング信号に基づいて動作している複数のレジスタ間でデータのやりとりを行う場合がある。そこで、 $n$  入力,  $m$  出力を持つ場合の基本モジュールを図7に示す。図7において、 $in1 \sim inn$  はそれぞれローカルタイミング信号が異なるレジスタから出力されたデータを現し、 $out1 \sim outm$  はそれぞれローカルタイミング信

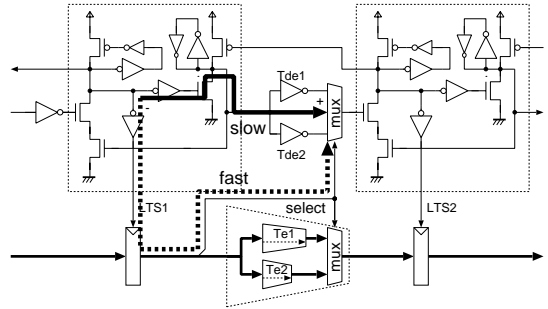


図8 入力データに依存したローカルタイミング信号の生成  
Fig.8 Utilization of variable delay inverter.

号が異なるレジスタへ入力するデータを表す。

図7のローカルタイミング信号生成回路の基本構成は GasP family<sup>20)</sup>の中の Two successor places を拡張したものであり、1入力1出力の基本モジュールと同様、入出力でそれぞれ4相ハンドシェイクを実現するように構成したものである。また、図5同様、破線で囲まれた部分をセルとして用意する。なお、初期状態でレジスタに初期値が代入されるときは、 $m$  個ある keeper はすべて full (0) とし、初期値が不要な場合は keeper はすべて empty (1) とする。

回路動作は1入力1出力のときと同じく、書き込み、読み出しの要求-応答信号遷移に基づいてFIFO動作を実現する。また、SDIモデルに基づいて正しく動作するための制約は基本的に図6に示したものと同一である。図6では速いパスと遅いパスの関係が1対1として示されているが、入力数  $n$ 、あるいは出力数  $m$  に応じて、1対  $n$ 、あるいは1対  $m$  すべてに関して遅いパスの遅延が速いパスの遅延の  $K$  倍となるように回路を構成する。

### 3.3 平均遅延動作による速度性能の向上

前述のとおり、ある基本モジュールのローカルタイミング信号が1に遷移してから、次の基本モジュールのローカルタイミング信号が1に遷移するまでの遅延は、図5におけるインバータAの立ち上がり遅延により決定される。そこで、状態機械における状態ごとにクリティカルパスが変化する構成、あるいは図8に示すように、データパスが遅延の異なる演算処理を行い、入力データに依存していずれかの処理をマルチプレクサで選択する構成の場合、インバータの立ち上がり遅延をデータパスの処理遅延に合わせてそれぞれ用意し、マルチプレクサで選択することで、ローカルタイミング信号 LTS1 を1に遷移させてから、LTS2 を1に遷移させるまでの遅延を変化させることができる。選択されたパス単体で考えると、その中の最悪遅延に

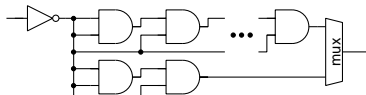


図9 入力データに依存した遅延の生成回路

Fig. 9 Implementation of variable delay inverter.

合わせたタイミング信号を生成する束データ方式<sup>1)</sup>となるが、システム全体としてはサイクルごと、あるいは入力データごとのクリティカルパスの遅延と、そのパスの選択される確率を掛け合わせた平均遅延により性能評価を行うことができる。

ただし、インバータの遅延制約は図6に示したとおりであり、いずれのインバータを選択したときでも、立ち下がり遅延はfastの制約を満たさなければならない。図9はこの制約を満たす回路構成の例である。立ち上がりはANDゲートの段数分の遅延となるが、立ち下がりはずねに一定の遅延で構成することができる。また、図8に破線の速いパスと実線の遅いパスとして示すように、マルチプレクサへのインバータの入力が立ち上がる前に選択信号が確定するように回路を設計しなければならない。

### 3.4 ループ構造や選択分岐構造の実現

これまで示してきた基本モジュールは容量1のFIFOであり、データパスが完全な直線構造であれば、単純に基本モジュールを結合することで、システム全体を構成することができる。しかしながら、実際のシステムではループ構造や選択分岐・合流構造が存在する。

最初にループ構造を実現する場合の制約を示す。図5および図7に示すローカルタイミング信号生成回路を用いる場合、その出力側に接続されているローカルタイミング信号生成回路のkeeperがempty状態にならないと要求-応答信号が遷移せず、データ転送が行われない。したがって、ループ構造は少なくとも2つのローカルタイミング信号生成回路を持ち、初期状態でkeeperがfullとemptyとなっていないなければならない。すなわち、ループ構造を実現するためには少なくとも2つの基本モジュールが必要となる。また、ループ構造でも基本モジュールの回路構成自体は変わらないため、SDIモデルに基づいて正しく動作するために満たさなければならない遅延制約は図6と同じものとなる。

次に、選択分岐・合流構造を実現する基本モジュールの構成として、GasP family<sup>20)</sup>のData conditional GasPを拡張した2出力の選択を行う基本モジュール構造を図10、合流を行う基本モジュール構造を図11

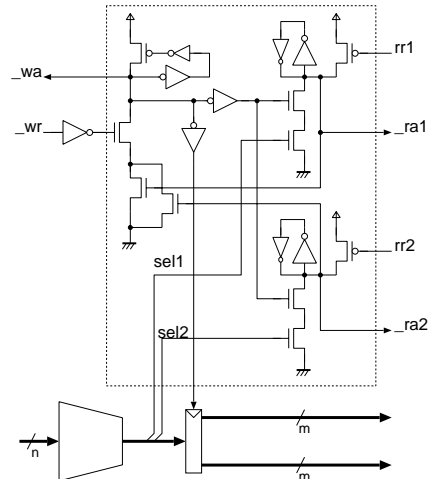


図10 選択分岐構造の基本モジュール

Fig. 10 Structure of the basic module with output selection.

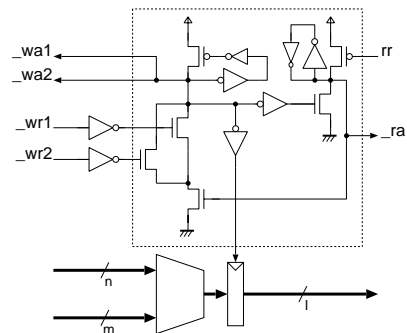


図11 合流構造の基本モジュール

Fig. 11 Structure of the basic module with input selection.

に示す。選択分岐構造では、1-out-of-2信号として入力された選択信号(sel1, sel2)により、読み出し応答信号 $_{ra1}$ ,  $_{ra2}$ いずれか一方のみが0に遷移し、選択分岐構造と対になる合流構造では、1-out-of-2信号として入力された書き込み要求信号( $_{wr1}$ ,  $_{wr2}$ )に基づいて書き込みが行われる。

3.5 遅延制約を満たすことが困難な場合の回路構成  
SDIモデルに基づいて正しく動作するための制約は図6に示すとおりであり、図6(a)のslow2と(b)のfast4のように、同じパスが一方では速いパス、他方では遅いパスとなるため、遅延制約を満たすのが困難な場合がある。また、基本モジュール間の遅延も予測可能でなければならないが、実際の設計では基本モジュール間の遅延が予測できない場合もある。このような場合、基本モジュールの構成として、図12、および図13に示す構成を用いる。図12は出力側の遅

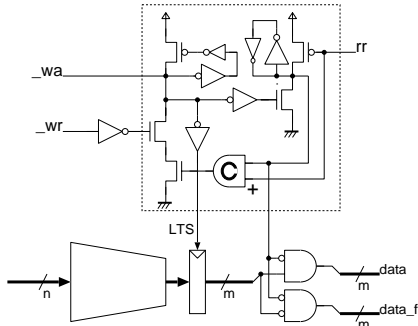


図 12 出力側の遅延が不明な場合の基本モジュールの構成  
Fig. 12 Structure of the basic module with DI-output.

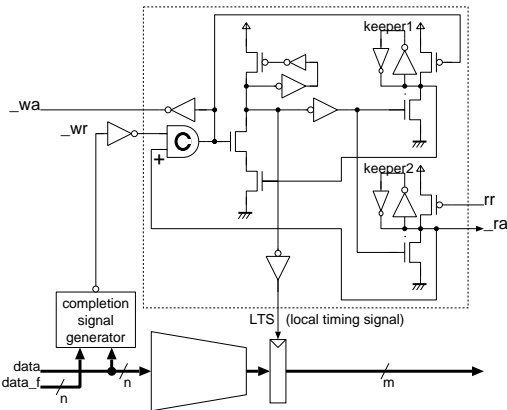


図 13 入力側の遅延が不明な場合の基本モジュールの構成  
Fig. 13 Structure of the basic module with DI-input.

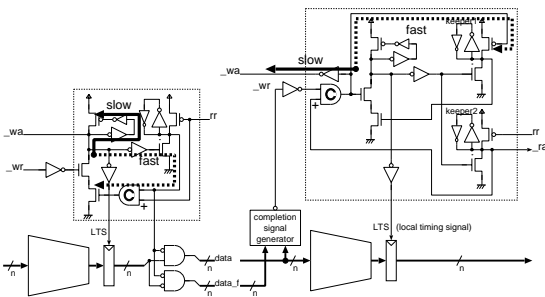


図 14 SDI モデルに基づく遅延制約  
Fig. 14 Timing constraints to be observed.

延が予測できない箇所、図 13 は入力側の遅延が予測できない箇所に使用する回路構成である。これらは入力と出力で対として使用する。

遅延の予測ができない箇所ではデータ自身に遅延情報を持たせるため、2 線化を行う。図 12, 図 13 において、信号線名に「\_f」の付加されたデータ信号線は 2 線式論理の否定成分であることを表す。入力側では 2 線化されたデータ信号がすべて符号語あるいはスペース (0,0) となったことを完了信号生成回路

(図 13, completion signal generator) により検知し、書き込み要求信号を生成する。

図 12 および図 13 に示す回路構成が SDI モデルに基づいて正しく動作するための遅延制約は、図 6 に加えて、図 14 に示す制約を満たす必要がある。

#### 4. 同期式構造モデルを用いた設計

前章に示した基本モジュールを用いた局所同期型非同期式 VLSI の設計方式を以下に示す。

- (1) グローバルクロックの立ち上がりエッジに基づいて値を更新するレジスタと組合せ回路から構成される同期式 RTL 構造記述を用意する。
- (2) 同一のタイミング信号に基づいて動作するレジスタ群を以下に基づいて決定する。
  - ソースレジスタとデスティネーションレジスタがまったく重複しない完全な直線構造の場合、レジスタは完全な FIFO として動作すると考えてよいため、ソースごと、およびデスティネーションごとにそれぞれ異なるタイミング信号に同期して動作するものとする。
  - パイプライン構造ではパイプラインステージごとにそれぞれ異なるタイミング信号に同期して動作するものとする。
  - 直線構造、パイプライン構造以外の場合、全レジスタが 1 つのローカルタイミング信号で動作するものとする。
- (3) システムが完全な直線構造であり、記憶素子に初期値を入力する必要がない場合、すべてのローカルタイミング信号生成回路の keeper を empty 状態にしておき、データが入力されたところから順次ローカルタイミング信号を 1 に遷移させることができるため、レジスタ間を分割する必要はない。したがって、データパスはそのままとして組合せ回路とレジスタごとに基本モジュールを構成する。

それ以外の場合は初期値の入力されたレジスタ間に存在する keeper のうち、少なくとも 1 つを empty としないとデータ転送が行われないため、組合せ回路を 2 分割し、レジスタ間に 2 つの基本モジュールを構成する。

- (4) SDI モデルに基づいて正しく動作するように、図 5 のインパータ A の立ち上がり遅延を式 (1) に基づいて決定する。

図 8 に示すようにデータパスがマルチプレクサでいずれかの演算を選択する構成の場合、そ



それぞれの処理遅延に応じて図9の可変遅延回路を付加する。

- (5) レイアウトを行った後の配線遅延情報を用いて、SDIモデルに基づいて正しく動作するための遅延制約を満たしているかどうかを確認する。遅延制約を満たしていない場合は基本モジュールを図12および図13の構成とする。また、基本モジュール間のデータ信号線を2線化し、入力側に完了信号生成回路を付加する。

本設計方式では、組合せ回路とレジスタから構成され、レジスタはグローバルクロックの立ち上がりエッジに基づいて値を更新する同期式システム構造記述を入力として、それを直接局所同期型非同期式VLSIとして実現する。したがって、制御回路とデータパスに分離してそれぞれの設計モデル・設計方式に応じた仕様記述を行わなければならない従来の非同期式システム設計手法と比較して、設計が容易となる。また、入力が同期式RTL構造記述であるため、同期式システム設計支援CADを有効利用することも可能と思われる。

5. 設計例と評価

本章では、同期式構造記述を利用した局所同期型非同期式VLSIの設計例として、フィルタ回路の設計例と評価結果を示す。また、 $K = 1$  と  $K = 1.5$  でそれぞれ設計し、 $K$  の値と速度性能とを比較した結果を示す ( $K = 1$  は遅延変動がないことを表すものではなく、各回路要素の遅延変動率がすべて同じであることを表す)。なお、評価条件として、テクノロジーは  $0.25 \mu\text{m}$  ルールのスタンダードセルライブラリを用い、ゲートの遅延は  $0.24 \text{ nm}$  の配線とインバータ3個の負荷がすべてのゲートに一律で課せられているとしたときの値を用いるものとする。

2nd order IIR Filter は  $o1 = (c0 * w1 + c1 * w2 +$

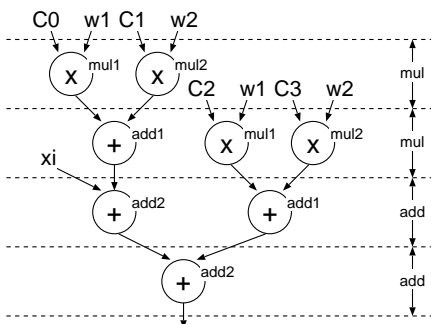


図15 IIRフィルタのデータフローグラフ  
Fig. 15 Data flow graph of 2nd order IIR filter.

$xi) + c2 * w1 + c2 * w2$  を実現するフィルタであり、乗算回路2個、加算回路2個、およびデータレジスタ4個を使用できる状況においてスケジューリングとリソースアロケーションを行うと図15に示すものが得られる。1サイクル目と2サイクル目は乗算回路がクリティカルパスとなり、サイクルタイムは乗算回路の遅延で制約され、3サイクル目と4サイクル目のサイクルタイムは加算回路の遅延で制約される。これをグローバルクロック信号に同期して制御を行う同期式RTL構造記述として実現したものが図16(a)であり、本論文で提案した手法に基づいて局所同期型非同期式VLSIとして実現したものが図16(b)である。図16(a)は自己ループ構造を含むため、1サイクルの処理を2つに分割して間にレジスタを挿入し、それぞれのレジスタを制御するローカルタイミング信号を生成する回路を付加する。また、1, 2サイクル目は乗算回路の遅延に合わせ、3, 4サイクル目は加算回路の遅延に合わせた立ち上がり遅延を持つように図9の遅延選択回路を設計し、選択信号によっていずれかを選択するようにする。

図16に示す各回路の遅延は表1に示すとおりである。

このとき、それぞれの回路のサイクルタイムをVerilog-XLによるシミュレーションにより求めた結果を表2に示す。同期式システムでは回路全体のクリティカルパスで制約されるサイクルタイムであり、 $6.63 \text{ (ns)}$  となる。一方、局所同期型非同期式VLSIでは平均サイクルタイムであり、 $K = 1$  のとき  $6.23 \text{ (ns)}$  となる。平均動作となることによるサイクルタイムの減少は  $0.40 \text{ (ns)}$  であり、これは乗算回路と加算回路

表1 演算回路のスペック  
Table 1 Specifications for filter circuit.

	遅延 (ns)
next state	0.549
datapath control	0.769
mux	0.912
mul	4.09
add	1.58
dff(write)	0.727
(setup)	0.130
(min clk pulse)	0.68

表2 速度性能の評価結果  
Table 2 Performance analysis.

	サイクルタイム (ns)
同期式システム	6.63
局所同期型非同期式システム ( $K = 1$ )	6.23 (1.00)
局所同期型非同期式システム ( $K = 1.5$ )	9.67 (1.55)

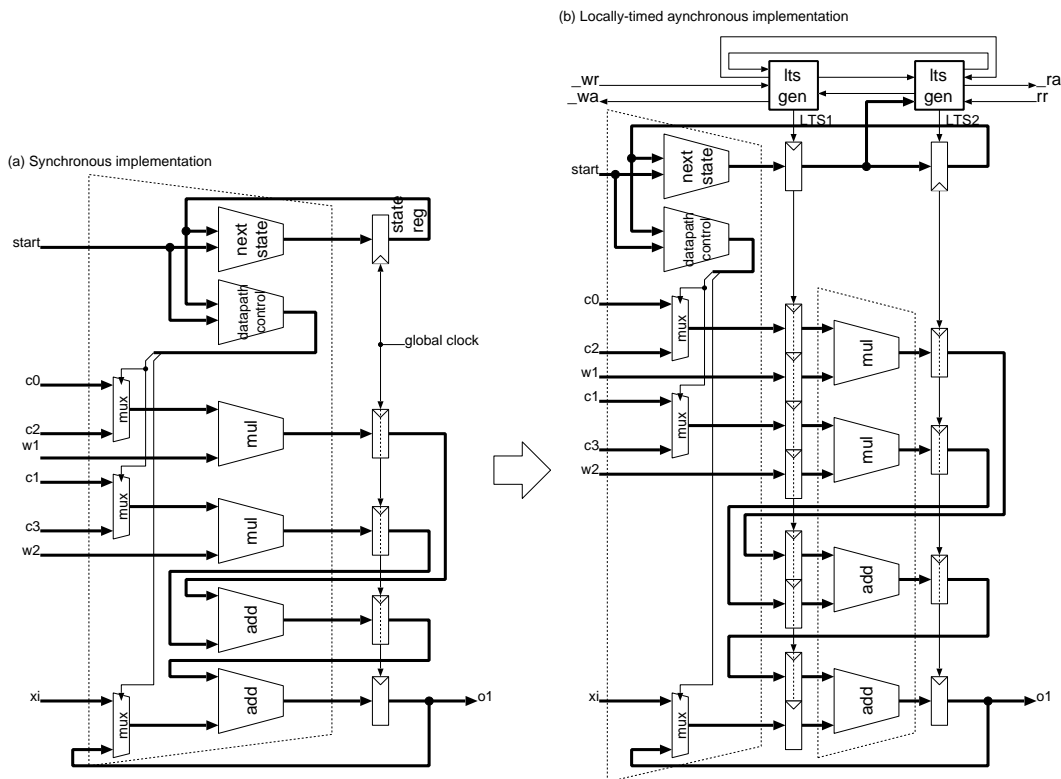


図 16 フィルタ回路の設計  
Fig. 16 An example filter circuit.

の遅延差の平均から挿入したレジスタの遅延を引いた値  $((4.09 - (4.09 + 4.09 + 1.58 + 1.58)/4) - (0.727 + 0.130)) = 0.398$  である。したがって、局所同期型非同期式 VLSI は、入力データに依存して処理遅延が変動する場合、「(処理遅延の最悪値) - (処理遅延の平均値) - (挿入する記憶素子のセットアップ遅延 + 書き込み遅延)」の分、同期式システムよりも高速化を実現できるといえる。また、 $K = 1$  と  $K = 1.5$  で設計した回路を比較した場合、 $K$  の値にほぼ比例して遅延が大きくなることが確認できる。これは基本的にデータ転送方式が束データ方式のデータ転送であり、ストローブ信号の遷移遅延をデータパスの遅延の  $K$  倍にするためである。

6. ま と め

本論文では、SDI モデルに基づく非同期式 VLSI の一設計方式として、要求-応答プロトコルに基づいて生成されたローカルなタイミング信号を利用する回路構成とシステム全体の設計方式を提案した。

ローカルタイミング信号生成には、GasP 回路と呼ばれるパルスのなタイミング信号を生成する FIFO 制

御回路を利用し、要求-応答プロトコルにともなうオーバヘッドの小さい回路を実現した。また、提案した回路構成は、あるローカルタイミング信号が 1 に遷移してから次のローカルタイミング信号が 1 に遷移するまでの遅延をデータパスの遅延に合わせて動的に変動させることができるため、入力データに依存して処理遅延が大きく変動するシステムでは、繰り返し実行されるレジスタ間データ転送処理遅延の平均値で性能を評価することができる。また、タイミング信号のみ非同期で実装する方式であるため、制御回路とデータパスに分離してそれぞれの設計モデル・設計方式に応じた仕様記述を行わなければならない従来の非同期式システム設計手法と比較して、設計が容易である。さらに、システム全体の設計は同期式システム RTL 構造モデルを入力とし、それを局所同期型非同期式システムとして実現する方式であるため、同期式システム設計支援 CAD を有効利用することができると思われる。

謝辞 本研究に対して適切な助言をいただいた中村宏東京大学先端科学技術研究センター助教授に深く感謝いたします。また、本研究に対する適切な助言、および本設計方式に基づいた非同期式システム設計支

援 CAD システム「AINOS」の開発にご尽力いただいた NEC マルチメディア研究所若林一敏, 高橋渡両氏, NEC 情報システムズ中越優佳, 宮沢義幸両氏に深く感謝いたします。

なお, 本研究の一部は情報処理振興事業協会「独自の情報技術育成事業」の一環として行われたものである。また, 本研究の遂行にあたり, 文部科学省科学研究費補助金基盤研究(B)(2)13480076のご支援をいただいた。

### 参 考 文 献

- 1) 南谷 崇: 非同期式マイクロプロセッサの動向, 情報処理, Vol.39, No.3, pp.181-186 (1998).
- 2) Ozawa, M., Ueno, Y., Imai, M., Nakamura, H. and Nanya, T.: A cascade ALU architecture for asynchronous superscalar processors, *IEICE Trans. Electronics*, Vol.E84-C, No.2 (2001).
- 3) Hauck, S.: Asynchronous design methodologies: An overview, *Proc. IEEE*, Vol.83, No.1, pp.69-93 (1995).
- 4) Huffman, D.A.: The synthesis of sequential switching circuits, *J. Franklin Inst.*, Vol.257, No.3, pp.161-190 (1954). *ibid.*, No.4, pp.275-303 (1954).
- 5) Nowick, S.M.: Automatic Synthesis of Burst-Mode Asynchronous Controllers, Ph.D. Thesis, Stanford University, Department of Computer Science (1993).
- 6) Chou, W.-C., Beerel, P.A. and Yun, K.Y.: Average-case technology mapping of asynchronous burst-mode circuits, *IEEE Trans. Computer-Aided Design*, Vol.18, No.10, pp.1418-1434 (1999).
- 7) Muller, D.E. and Bartky, W.S.: A theory of asynchronous circuits, *Proc. International Symposium on the Theory of Switching*, pp.204-243. Harvard University Press (April 1959).
- 8) Jacobson, H., Brunvand, E., Gopalakrishnan, G. and Kudva, P.: High-level asynchronous system design using the ACK framework, *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.93-103. IEEE Computer Society Press (April 2000).
- 9) Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L. and Yakovlev, A.: Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers, *IEICE Trans. Inf. Syst.*, Vol.E80-D, No.3, pp.315-325 (1997).
- 10) <http://www.lsi.upc.es/~jordic/petrify>
- 11) Park, S.B. and Nanya, T.: Synthesis of asynchronous circuits from signal transition graph specifications, *IEICE Trans. Inf. Syst.*, Vol.E80-D, No.3, pp.326-335 (1997).
- 12) Martin, A.J.: A synthesis method for self-timed VLSI circuits, *Proc. International Conf. Computer Design (ICCD)*, pp.224-229, Rye Brook, NY, IEEE Computer Society Press (1987).
- 13) van Berkel, K., Kessels, J., Roncken, M., Saeijs, R. and Schalijs, F.: The VLSI-programming language Tangram and its translation into handshake circuits, *Proc. European Conference on Design Automation (EDAC)*, pp.384-389, 1991.
- 14) Bardsley, A. and Edwards, D.A.: The Balsa asynchronous circuit synthesis system, *Forum on Design Languages* (Sept. 2000).
- 15) <http://www.cs.man.ac.uk/amulet/>
- 16) Takamura, A., Kuwako, M., Imai, M., Fujii, T., Ozawa, M., Fukasaku, I., Ueno, Y. and Nanya, T.: TITAC-2: An asynchronous 32-bit microprocessor based on scalable-delay-insensitive model, *Proc. International Conf. Computer Design (ICCD)*, pp.288-294 (Oct. 1997).
- 17) 今井 雅, 中村 宏, 南谷 崇: SDI モデルに基づいた非同期式パイプライン・データパスの論理合成, 情報処理学会論文誌, Vol.40, No.4, pp.1547-1556 (1999).
- 18) Özcan, M., Imai, M. and Nanya, T.: Generation and verification of timing constraints for fine-grain pipelined asynchronous data-path circuits, *Proc. Async2002*, pp.109-114 (April 2002).
- 19) Singh, M. and Nowick, S.M.: High-throughput asynchronous pipelines for fine-grain dynamic datapaths, *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.198-209, IEEE Computer Society Press (April 2000).
- 20) Sutherland, I. and Fairbanks, S.: GasP: A minimal FIFO control, *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.46-53, IEEE Computer Society Press (March 2001).
- 21) Ebergen, J.: Squaring the FIFO in GasP, *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.194-205, IEEE Computer Society Press (March 2001).
- 22) Nanya, T., Takamura, A., Kuwako, M., Imai, M., Ozawa, M., Özcan, M., Morizawa, R. and Nakamura, H.: Scalable-delay-insensitive design: A high-performance approach to de-

pendable asynchronous systems, *Proc. International Symp. on Future of Intellectual Integrated Electronics*, pp.531-540, Sendai, Japan (March 1999).

- 23) Muttersbach, J., Villiger, T. and Fichtner, W.: Practical design of globally-asynchronous locally-synchronous systems, *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.52-59 (April 2000).
- 24) Sjogren, A.E. and Myers, C.J.: Interfacing synchronous and asynchronous modules within a high-speed pipeline, *Advanced Research in VLSI*, pp.47-61 (Sept. 1997).

(平成 14 年 10 月 17 日受付)

(平成 15 年 3 月 4 日採録)



今井 雅 (正会員)

1996 年東京工業大学工学部電気電子工学科卒業。1997 年同大学院情報理工学研究科計算工学専攻修士課程修了。2000 年東京大学大学院工学系研究科先端学際工学博士後期課程満期退学。2000 年東京大学先端科学技術研究センター助手。非同期式 VLSI システムの設計方式および設計支援環境の構築に関する研究に従事。



Metehan Özcan

1993 年トルコ Bilkent 大学工学部情報工学科卒業。1996 年東京工業大学大学院工学系研究科電気・電子工学専攻修士課程修了。1999 年東京大学大学院工学系研究科先端学際工学博士後期課程満期退学。1999 年東京大学先端科学技術研究センター学術研究支援員。非同期式 VLSI システムの CAD に関する研究に従事。



南谷 崇 (正会員)

1946 年生。1969 年東京大学工学部計数工学科卒業。1971 年同大学院修士課程修了。日本電気(株)中央研究所勤務を経て、1981 年東京工業大学情報工学科助教授、1989 年同電気電子工学科教授。1995 年東京大学計数工学科教授。1996 年東京大学先端科学技術研究センター教授。論理システムの物理的実現に関する諸問題に興味を持つ。工学博士。1987 年電子情報通信学会論文賞、1994 年大川出版賞、1998 年 ASP-DAC Best Paper 賞、Outstanding Design 賞受賞。著書「順序機械」(岩波書店)、「フォールトトレラントシステムの構成と設計」(楳書店)、「フォールトトレラントコンピュータ」(オーム社)等。IEEE fellow, 電子情報通信学会フェロー