

## デジタルLSIの製造後クロック調整技術の高速化手法

諏佐達也<sup>†</sup> 村川正宏<sup>††</sup> 高橋栄一<sup>††</sup>  
 古谷立美<sup>†</sup> 樋口哲也<sup>††</sup> 古市慎治<sup>†††</sup>  
 上田佳孝<sup>†††</sup> 和田淳<sup>†††</sup>

製造ばらつきにより発生するクロック・スキューの問題を解決するための手法として、遺伝的アルゴリズムを用いたデジタルLSIの製造後クロック調整技術が提案されている。しかし、大規模なLSIの調整では、調整箇所が増大するため、調整時間が増加するという問題がある。そこで、本研究では、大規模デジタルLSIにも適用可能な製造後クロック調整の高速化手法を提案する。提案手法では、LSI設計時に行うSTA(Static Timing Analysis)の結果を用いて調整箇所を限定し、調整時間を短縮する。それに加えて、遺伝的アルゴリズムの初期集団の分布を工夫することで、さらに調整時間を短縮する。さらに、これらの手法による調整効果をLSIの設計時に検証できるようにするための調整シミュレータを開発した。このシミュレータを用いた調整実験の結果、1,031カ所のフリップフロップが存在する実用的な回路において、数秒という現実的な時間で調整が完了できる見込みを得た。

## Speeding Up Technique for Post-fabrication Clock-timing Adjustment of Digital LSIs

TATSUYA SUSU,<sup>†</sup> MASAHIRO MURAKAWA,<sup>††</sup> EIICHI TAKAHASHI,<sup>††</sup>  
 TATSUMI FURUYA,<sup>†</sup> TETSUYA HIGUCHI,<sup>††</sup> SHINJI FURUICHI,<sup>†††</sup>  
 YOSHITAKA UEDA<sup>†††</sup> and ATSUSHI WADA<sup>†††</sup>

To solve the problem of fluctuations in clock timing with large scale digital LSIs (also known as the "clock skew" problem), the post-fabrication clock-timing adjustment technique using a genetic algorithm (GA) has been proposed. However, the adjustment time increases incurred when more programmable delay circuits are incorporated within large-scale LSIs is a serious issue. For this problem, we propose a post-fabrication clock adjustment method to realize practical applications. This method reduces the adjustment time by reducing adjustment points utilizing results of static timing analysis (STA) and adopting improved distribution for initial population of GA. Moreover, we have developed an adjustment simulator to predict the adjustment results by the proposed method in design stages of LSIs. Adjustment experiments using the developed simulator demonstrate that our method can adjust practical LSIs with 1,031 flip-flops within a few seconds.

### 1. はじめに

近年、デジタルLSIの高速化や微細化にともない、クロック信号の伝播時間のずれであるクロック・スキュー(Clock Skew)の問題が深刻になってきている<sup>1)</sup>。この結果、デジタルLSIの動作歩留り(設計ど

おりのスペックで動作できるチップの比率)が低下し、チップ価格の上昇を招いている。クロック・スキューの主な発生原因の1つに製造ばらつきがあり、これにより個々のトランジスタの遅延時間や駆動能力、配線の抵抗や寄生容量が変動してしまう。このデジタルLSIのクロック・スキュー問題を解決するために、半導体MIRAIプロジェクトでは、チップのクロックラインにクロックの入力タイミングを微調整可能にするプログラマブル遅延回路を複数個挿入し、製造後にそれら遅延回路の調整値を遺伝的アルゴリズム(Genetic Algorithm; GA)<sup>2)</sup>を用いて調整する方式を提案している<sup>3),4)</sup>。遅延回路がそれぞれ44カ所と52カ所の2種類の中規模LSIでの調整実験<sup>4)</sup>の結果、歩留りを

<sup>†</sup> 東邦大学大学院理学研究科  
 Graduate School of Science, Toho University

<sup>††</sup> 半導体MIRAIプロジェクト、産業技術総合研究所次世代半導体研究センター  
 MIRAI-ASRC, National Institute of Advanced Industrial Science and Technology (AIST)

<sup>†††</sup> 三洋電機株式会社  
 Sanyo Electric Co., Ltd.

大幅に向上させることに成功している。しかし、大規模な LSI の調整では、調整箇所が増大し、調整時間が大幅に増加するという問題がある。

そこで本稿では、大規模デジタル LSI にも適用可能な製造後クロック調整の高速化手法を提案する。提案手法には 3 つの特徴がある：(1) STA (Static Timing Analysis) の結果を用いて、調整する遅延回路の数を LSI の設計時に限定する。これにより、調整時間を短縮させ、遅延回路のために必要な回路面積を削減させることができる。(2) 論理シミュレーションを行うことなく、短時間で調整シミュレーションが可能なシミュレータを開発した。このシミュレータを用いることで、LSI 設計時に (1) の効果を容易に確認することができる。(3) 調整時の GA の初期集団の分布を工夫する。この工夫により、GA の収束速度を高め、LSI 製造後の調整時間を短縮することができる。

提案手法の有効性を検証するために、1,031 カ所のフリップフロップが存在する実用的な回路に対して、提案手法を用いて遅延回路を挿入した試作 LSI を設計した。開発したシミュレータを用いて調整実験を行った結果、従来手法では実用的な時間内での調整ができなかったのに対して、実用的な時間内での調整が可能になる見通しを初めて得た。

以下、2 章ではデジタル LSI の製造後クロック調整技術についての説明を行い、大規模 LSI の調整時の問題について述べる。続いて、3 章では大規模デジタル LSI にも適用可能な製造後クロック調整の高速化手法を提案し、その詳細を説明する。4 章ではシミュレータを用いた調整実験の結果を報告し、最後に、5 章でまとめと今後の課題を述べる。

## 2. デジタル LSI の製造後クロック調整

デジタル LSI には、クロック・スキューによる動作歩留りの低下の問題がある。この問題を解決するために、チップを製造後に調整することによりクロック・スキューを補正する方式が提案されている (図 1)。

この方式では、あらかじめチップの設計段階で、クロック信号の到達タイミングを微調整することができるプログラマブル遅延回路を、すべてのフリップフロップに対して挿入したうえでチップを製造する。チップの製造後に、遺伝的アルゴリズム<sup>2)</sup>を用いて、挿入したプログラマブル遅延回路を最適に調整する。プログラマブル遅延回路には様々な実装方法があるが、たとえば文献 4) では 30 ps 以下の単位遅延時間が実現されている。また、プログラマブル遅延回路の遅延値の設

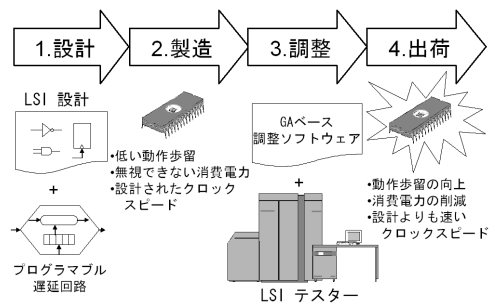


図 1 LSI の製造後クロック調整技術

Fig. 1 Post-fabrication clock-timing adjustment for digital LSIs.

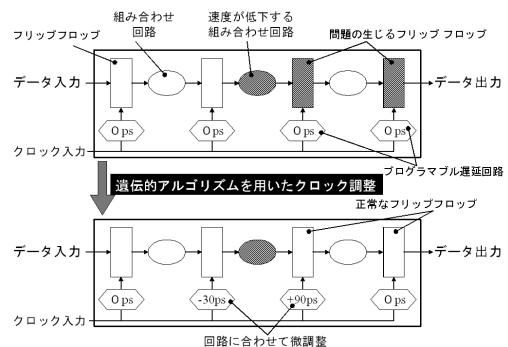


図 2 遺伝的アルゴリズムを用いた調整の様子

Fig. 2 Adjustment using genetic algorithms.

定方法としては様々な方法があるが、たとえば文献 3) のような方法を用いてもよい。

調整の様子を図 2 のようなパイプライン型の回路で説明する。段間の組合せ回路に製造誤差などによる遅延が生じた場合、その組合せ回路の直後のフリップフロップでは、組合せ回路からのデータ信号よりも早くクロック信号が到達してしまう。そのため、このフリップフロップでは本来取り込まれるべき値とは異なる不定の値が取り込まれてしまい、回路全体が誤動作を起こしてしまう。そこで、不定の値が取り込まれてしまうフリップフロップへのクロックライン上に挿入されたプログラマブル遅延回路を微調整し、クロック信号の入力タイミングを遅らせる。そうすることで、遅延回路が直前の組合せ回路で生じた遅延を吸収し、回路全体の誤動作を解消することが可能となる。同様に、クロックツリーで生じたばらつきによる遅延が、回路の誤動作を招いた場合にも本調整は有効である。

ただし、チップ内でプログラマブル遅延回路が必要な場所は多数あり、それぞれの最適な遅延時間の値は相互に密接に関係している。このため、それぞれの遅延時間の最適値を求めようとすると、広大な探索空間を持つ探索問題を解く必要がある。そこで、調整ソフ

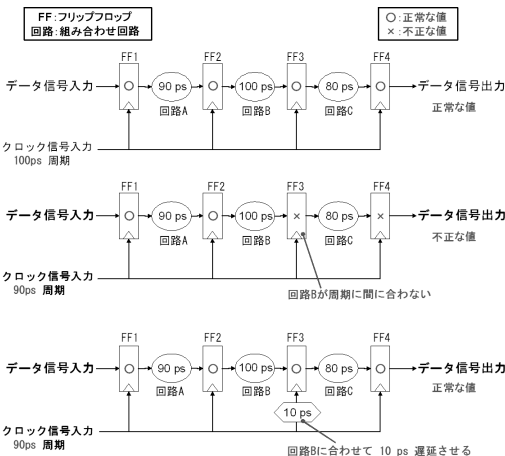


図 3 設計よりも速いクロック速度の実現例

Fig. 3 Realization of faster clock speed than design specifications.

トウェアでは、遺伝的アルゴリズムを用いることで、複数のプログラマブル遅延回路の最適な遅延時間を高速に求めることを可能にしている。製造後クロック調整で遺伝的アルゴリズムを用いる利点は、製造後のLSIで容易に測定することができる出力信号を基に、調整に必要な演算を行える点である。それに対し、他のグラフィカルアルゴリズム<sup>5)</sup>を用いる場合には、LSI製造後の製造誤差によって生じた遅延などの測定が必要になるため、時間やコストの面から現実的ではない。

2.1 製造後クロック調整の効果

この方式を用いることにより、動作歩留りの向上だけでなく、(1) 設計よりも速いクロック速度の実現、動作電源電圧を低下させることによる(2) 低消費電力化が実現されている<sup>4)</sup>。以下で、これらの原理を説明する。

図3の上段のように、それぞれ90ps, 100ps, 80psで動作する3つの組合せ回路が、レジスタと交互に直列に接続されているパイプライン型の回路を考える。このとき、レジスタで正確なデータを保持できるクロック信号の周期は、最も遅い組合せ回路Bに合わせた100psである。すなわち、この回路全体は100psの周期で動作する。ここで、図3の中段のように、回路全体の周期を90psに変更する。すると、100psで動作する組合せ回路Bを通過するデータがレジスタ3でのクロック信号到達に間に合わなくなる。そのため、回路全体では誤動作を引き起こしてしまう。そこで、図3の下段のように、データ信号の到達からクロック信号の到達までに時間的余裕がないレジスタ3へのクロック信号を、遅延回路により10ps遅らせる。このようにすることで、100psで動作する組合せ回路

Bに10psの時間的余裕をつくり、回路全体の誤動作を解消することができる。しかし、遅延回路によりレジスタ3のクロック到達タイミングが10ps遅れてしまったため、その直後の組合せ回路Cは、通常より10ps速い80psで動作しなければならない。だが、組合せ回路Cは元々80psで動作する回路なので、誤動作は起こらない。したがって、1つのレジスタのクロックタイミングを遅延させることで、回路全体のクロック周波数を向上させることが可能となる。製造後調整では、製造後のチップそれぞれについて、このような状態を実現する各遅延回路の設定値を自動的に探索する。この結果、設計よりも速いクロック速度を実現させることが可能である。

同様の原理でLSIの低消費電力化も実現可能である。LSIの消費電力は電源電圧の二乗に比例して大きくなるという性質がある<sup>6)</sup>。すなわち、LSIの電源電圧を低くすることができれば、消費電力の削減が可能である。ただし、電源電圧を低下させることで、回路内部のトランジスタの駆動能力が低下し発生した遅延のために、一部の組合せ回路ではレジスタへのクロック信号到達にデータ処理が間に合わなくなってしまい、誤動作が起こる。そこで、クロック速度を向上させる場合と同様の原理で、遅延回路を利用することで、電源電圧を低下させたにもかかわらず、回路を正常に動作させることができる。このようにして、製造後クロック調整により、消費電力を削減することも可能になる。

2.2 大規模デジタルLSI調整時の問題点

しかし、より大規模なデジタルLSIを調整するとなると、従来の調整手法では限界があった。中規模回路の調整実験<sup>4)</sup>では調整対象のLSI内部のすべてのフリップフロップに対してプログラマブル遅延回路を挿入していた。この従来手法をそのまま大規模LSIに適用したのでは、挿入する遅延回路の数が増大し、チップ面積が増大することにより、チップ製造コストが増える。それに加え、調整箇所が増大によって遺伝的アルゴリズムの探索空間が指数関数的に増大する。その結果、調整にかかる時間が増大し、LSIテストコストの増大につながる。つまり、LSIの生産コストは大きく分けて(1)チップ面積に応じた製造コスト、(2)パッケージ代、(3)LSIテストコストの3つの要素からなる。大規模LSIに従来調整手法をそのまま適用すると、(1)と(3)が増大するため、調整により歩留りが向上したとしても、結果的にコストの増大につながってしまう。

### 3. 製造後クロック調整の高速化手法の提案

本稿では、大規模デジタル LSI にも適用可能な製造後クロック調整の高速化手法を提案する．図 4 に提案手法のフローチャートを示す．提案手法では、大規模回路を現実的な時間内で調整可能にするために、調整箇所を限定と遺伝的アルゴリズムの初期集団の分布を改良することで調整時間を短縮させる．以下で、図 4 中の(1),(2),(3)の工程についてそれぞれ 3.1, 3.2, 3.3 節で詳しく説明する．

#### 3.1 調整箇所の限定

この工程では、調整箇所を限定することによって調整時間を短縮させることが目的である．ここで、LSI の設計時に調整箇所を限定することでチップ面積も減少させることができる．そこでこの工程では、図 5 のようにして調整箇所を限定する．以下でその手順を説明する．

- (1) LSI の設計時に行う STA (Static Timing Analysis) の結果から、パス遅延時間のリストを生成する．パス遅延時間とは、フリップフロップからフリップフロップまでの間の組合せ回路をデータ信号が通過するのにかかる時間のことである．調整時に電源電圧を変更する場合には、事前に用意しておいた、想定する各電源電圧ごとの STA の結果を用いればよい．
- (2) 作成したリストを基に、タイミング的な余裕に関してパス遅延時間を順位付けする．
- (3) 最も余裕のないパス遅延時間の両端のフリップフロップから順に  $N$  カ所を調整箇所に決定する．

このようにして、調整の効果が見込める場所にのみ調整箇所を限定し、調整時間を短縮する．ただしこのとき、限定する調整箇所の個数  $N$  の最適値は調整による歩留り向上と調整時間増大とのトレードオフで決まるので、対象とする LSI によって異なる．そこで、図 4 に示すように、次節で説明する調整シミュレータを用いて繰り返し調整シミュレーションを行い、最適な  $N$  の値を対象とする LSI ごとに決定する．

#### 3.2 調整シミュレーション

調整の効果を設計時に確認するためには、GA による調整シミュレーションを行う必要がある．GA では回路がどの程度正常動作するかを評価値として使用し、遺伝的操作により染色体の修正を行った後に再度評価値の計算を行う．このような繰り返し作業を何千回と行うことになるため、回路が正常動作するかを調べるために、組合せ回路の論理シミュレーションを行ったのでは、1 度の評価値計算に時間がかかり、調整シミュ

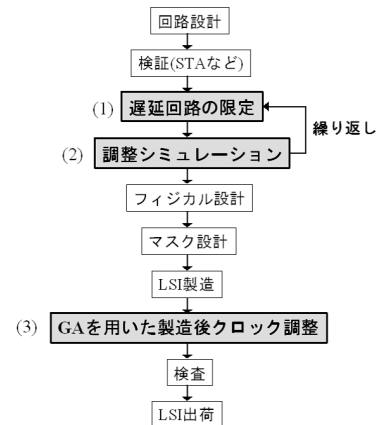


図 4 提案手法のフローチャート  
Fig. 4 Flow chart for the proposed method.

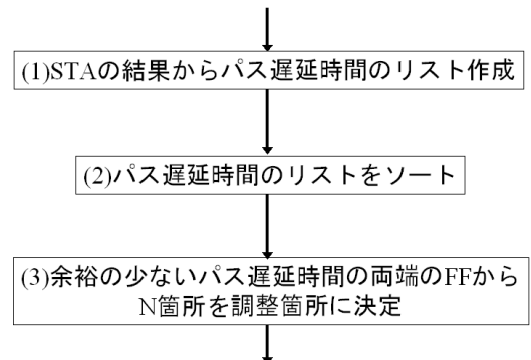


図 5 調整箇所の限定のフローチャート  
Fig. 5 Flow chart for reduction of adjustment points.

レーションが完了するまでに膨大な時間がかかってしまう．

そこで、フリップフロップが正常な情報を伝達できるかという、タイミング情報のみをモデル化したシミュレータを新たに作成した．このシミュレータを用いることで、最適な調整箇所の数をより高速に見出せる．本調整シミュレータでは、図 6 のようなシミュレーションモデルを用いた．この図では、入力データが組合せ回路やフリップフロップを通過し、出力されるまでの様子を有向グラフで表現している．1 つ 1 つの組合せ回路の情報や、通過データの細かいタイミング関係などを省いているため、少ない計算量でタイミングに関する動作確認が可能となる．具体的には、以下のような手順でチップの動作確認を行う．

- (1) シミュレーションの初期状態では、すべてのフリップフロップ  $i$  はすべて正常に動作すると仮定する．また、ネットワークの接続関係は、STA の結果を用いることで構築する．ここで、

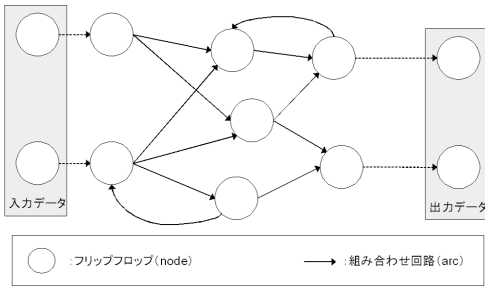


図 6 有向グラフで表した調整シミュレータのシミュレーションモデル

Fig. 6 Simulation model for adjustment simulator using directed graphs.

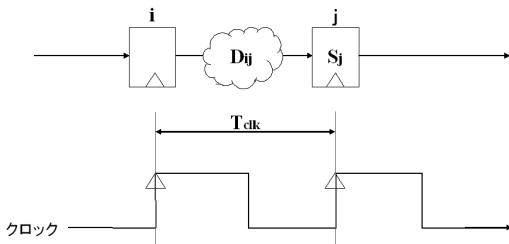


図 7 フリップフロップの動作モデル

Fig. 7 Model for operation of a flipflop.

フリップフロップ  $i$  が正常に動作するかどうかを表す値を  $S_i$  とし、正常動作するならば 1, 正常動作しないならば 0 とする ( $n$  は全フリップフロップ数).

$$S_i = 1 \quad : \quad 0 \leq i \leq n-1 \quad (1)$$

- (2) チップ内のフリップフロップ  $i, j$  を前後に持つ組合せ回路を通過する時間  $D_{ij}$  が、クロック信号の 1 周期分の時間  $T_{clk}$  以下であれば、演算されるデータが正常に伝達されると仮定し、フリップフロップ  $j$  が正常に動作するかどうかを表す値  $S_j$  を 1 とする. また、データが正常に伝達されていないフリップフロップ  $j$  には正常なデータでないという印  $S_j = 0$  をつける (図 7).

$$D_{ij} \leq T_{clk} \quad : \quad S_j = 1 \quad (2)$$

$$D_{ij} > T_{clk} \quad : \quad S_j = 0 \quad (3)$$

- (3) この正常なデータを持たない印  $S_j = 0$  のついたフリップフロップ  $j$  を入力を持つ組合せ回路  $D_{jk}$  では、正常な値を出力することができないため、印のついたフリップフロップ  $j$  の出力値を入力とするフリップフロップ  $k$  には、同様に印  $S_k = 0$  をつける (図 8). このようにして、チップ内のすべての組合せ回路でデータが正常に伝達されているかをチェックする.

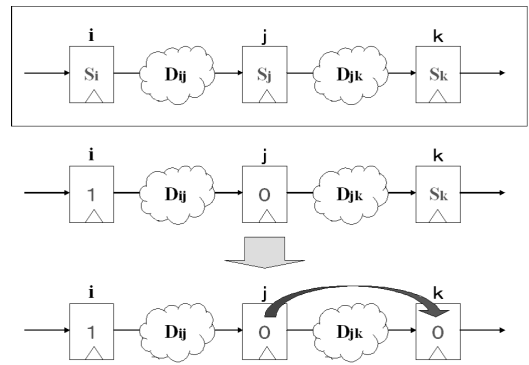


図 8 フリップフロップの動作状態の遷移モデル

Fig. 8 State transition model for operation of flipflops.

- (4) 最終的に出力に関する各フリップフロップ  $out_i$  が、正常なデータ  $S_{out_i} = 1$  をすべて保持できていればそのチップは正常に動作していると判断する. また、出力に関するフリップフロップ  $out_i$  が 1 つでも正常なデータを保持できていなければ ( $S_{out_i} = 0$ ), そのチップは誤作動してしまうチップだと判断する.

ここで、出力に関するフリップフロップ  $out_i$  の総数  $N_{out}$  のうち、正常なデータを保持しているフリップフロップ  $out_i$  の個数  $n_{out}$  の割合  $n_{out}/N_{out}$  を、遺伝的アルゴリズムでの適応度  $fit$  に対応させる. このようにして、遺伝的アルゴリズムでは適応度  $fit$  が 1 になる染色体が現れるまで遺伝的操作を繰り返すことになる. このとき、遺伝的アルゴリズムでは、LSI の出力の正誤情報のみを使用し、各パスの遅延時間情報を使用していないことに留意されたい.

次に、組合せ回路のタイミングに関するモデルについて説明する. フリップフロップ  $i, j$  のクロックラインに挿入したプログラマブル遅延回路の遅延値を  $D_i, D_j$  とした場合、調整後のフリップフロップ  $i, j$  間のパス遅延時間  $D'_{ij}$  は式 (4) で求められる.

$$D'_{ij} = D_{ij} + D_j - D_i \quad (4)$$

このとき、遅延時間  $D_{ij}$  は、LSI 設計時の STA の解析結果を用いればよい. 製造工程で、2 つのフリップフロップ  $i, j$  のクロックタイミングに生じたばらつきと、フリップフロップ  $i, j$  間のパス遅延時間に生じたばらつきを合算したものを  $Var_{ij}$  とすると、ばらつきを考慮した遅延時間  $D''_{ij}$  は、式 (5) のように定義できる.

$$D''_{ij} = D'_{ij} + Var_{ij} \quad (5)$$

ばらつき値  $Var_{ij}$  は、簡単のために相対値  $\sigma_{rel}$  で正規分布すると仮定し、正規乱数  $N(0, 1)$  を用いて式 (6) で求める.

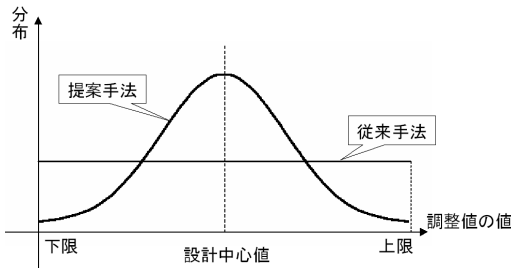


図 9 個体の初期パラメータの分布の比較

Fig. 9 Comparison of distribution for initial parameters of an individual.

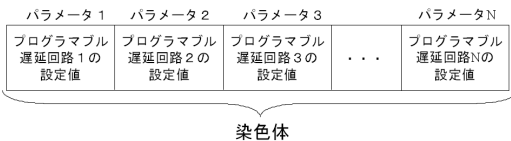


図 10 染色体の設定

Fig. 10 Coding of the chromosome.

$$Var_{ij} = D_{ij} \times \sigma_{rel} \times N(0,1) \quad (6)$$

こうして求めた  $D'_{ij}$  を式 (2), 式 (3) の  $D_{ij}$  の代わりに用いる. このようにして, すべてのフリップフロップ間のパス遅延をモデル化した仮想的なチップに対し, GA ソフトウェアを用いて, 調整シミュレーションを実施する.

### 3.3 GA 初期集団の分布の改良

GA で調整を行う際, 従来は個体の初期パラメータの値を一様乱数により設定していた. しかし, 製造誤差は設計値を中心として前後にばらついて発生している<sup>7)</sup>. そこで本手法ではこの性質を生かすために, 個体の初期パラメータの値を設計値を中心として正規乱数により分布させる (図 9). 具体的には,  $N$  個のプログラマブル遅延回路のそれぞれの設定値  $D_i$  ( $0 \leq i \leq N - 1$ ) は, 以下の式 (7) で設定する.

$$D_i = \sigma_{GA} \times N(0,1) \quad (7)$$

このようにプログラマブル遅延回路の初期値を設計値を中心として正規乱数により分布させることで, GA の収束速度が高まり, 解候補をより迅速に見出せる.

本手法では, GA の染色体はチップ内の  $N$  個のプログラマブル遅延回路の各遅延値をパラメータとし, そのパラメータ群を一列に並べたものを 1 組の染色体とする (図 10). 各遅延値は, 遅延の割合に対応する. 遅延値が負の場合は, 1 周期前のクロック信号を大幅に遅らせることにより, 擬似的にクロック信号を速めることに相当する.

GA を用いた調整のフローチャートは図 11 のようになる. 以下でその手順を説明する.

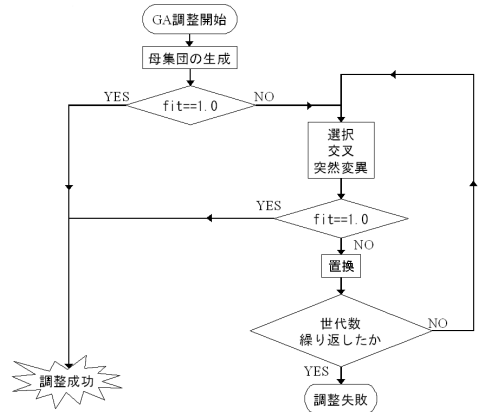


図 11 GA を用いた調整のフローチャート

Fig. 11 Flow chart of adjustment using GA.

- (1) 式 (7) に従って初期集団を発生させ, 母集団数だけ個体を生成し, 評価値  $fit$  を求める. ここで評価値  $fit$  が 1.0 になる個体が存在すれば調整成功と見なし終了となる. 評価値  $fit$  が 1.0 になる個体が 1 つもない場合には, (2) へ進む.
- (2) 母集団の中からランダムに親の 2 個体を選択し, 交叉, 突然変異の処理を行い, 子の 2 個体を生成する. 交叉, 突然変異にはそれぞれ一様交叉と Gaussian Mutation (正規分布を用いた突然変異<sup>8)</sup>) を用いる.
- (3) 子の 2 個体に対し評価値  $fit$  を求める. ここで子の個体の一方でも評価値  $fit$  が 1.0 ならば調整成功と見なし終了となる. どちらの個体も評価値  $fit$  が 1.0 に満たないならば, (4) へ進む.
- (4) 親の 2 個体と子の 2 個体で評価値  $fit$  を比較し, 評価値  $fit$  の大きい 2 個体を元の母集団に戻す (置換).
- (5) 母集団の数/2 の回数だけ (2) から (4) を繰り返した場合, 1 世代終了したと見なす.
- (6) (2) から (4) を一定世代数繰り返しても調整が終了しなければ, 調整失敗と見なし終了する.

## 4. シミュレーションによる提案手法の有効性の検証

提案手法の有効性を検証するために, 調整可能な箇所が 1,000 カ所を超える回路を設計し, その設計データを基にシミュレータで仮想的に生成したチップを用いて調整実験を行う. 従来の 50 カ所程度の調整実験と比較すると, 大規模な調整対象であるといえる.

### 4.1 実験条件

シミュレーション実験では, 設計動作電源電圧 1.0 V, 設計動作周波数 200 MHz で設計された画像圧縮伸長

処理に用いる DCT 演算回路<sup>9)</sup>を基に、プログラマブル遅延回路を組み込んだ試作 LSI の設計情報を用いる。この回路内のフリップフロップ数は 1,031 個、回路規模は 3 万ゲートであり、現在の平均的規模の機能ブロックといえる。この回路の STA の結果を基にシミュレータ上で仮想的な LSI を 100 チップ作成し、(1) 調整箇所を限定したときの効果と (2) GA 初期集団の分布を改良したときの効果を、動作周波数方向の調整実験と動作電源電圧方向の調整実験でそれぞれ比較する。

動作周波数方向の調整実験では、動作電源電圧を 1.0 V で固定し、動作周波数を 200 MHz から 320 MHz まで 10 MHz 刻みで増加させ、各動作周波数ごとに歩留りを調査する。動作電源電圧方向の調整実験では、動作周波数を 200 MHz で固定し、動作電源電圧を 1.0 V から 0.8 V まで 0.01 V 刻みで減少させ、各動作電源電圧ごとに歩留りを調査する。

GA の諸条件は以下のように設定した。

- 母集団数：50
- 染色体長：調整箇所数  $N$
- 打ち切り評価回数：3,000 (60 世代)
- 交叉確率：0.5
- 突然変異確率：1.0
- $\sigma_{GA}$ ：72 ps
- $D_i$  の上下限值：-504 ps ~ 576 ps

打ち切り評価回数を 3,000 回に設定した理由として、実際の LSI では 1 回の評価時間に 1 ミリ秒程度要すると見積もっているため<sup>4)</sup>、1 チップあたりの調整時間を実用的な限度である数秒程度に抑えるためには、3,000 回が妥当であると考えた。 $\sigma_{GA}$  は想定している製造ばらつきの度合いによって決定すべき値である。今回は予備実験により  $\sigma_{GA}$  の値を決定した。

#### 4.2 実験手順

上記の条件のもと、実験は以下のような手順で行う。

- (1) 動作周波数 (動作電源電圧) を設定し、遅延回路の調整値を 0 にした状態 (無調整状態) で正常動作するかどうかを確認する。
- (2) (1) で正常動作したチップに対しては調整を行う必要がないため、(4) へ。(1) で正常動作しなかったチップは、(3) へ。
- (3) GA を用いた遅延回路の調整を行う。調整が成功すれば正常動作するチップと判断し、(4) へ。調整が失敗した場合は正常動作しないチップと判断し、(4) へ。
- (4) 次のチップで (1) から (3) を繰り返す。ただし、一定チップ数の実験を終えた場合は (5) へ。

- (5) 動作周波数 (動作電源電圧) を変更し、(1) から (4) を繰り返す。調査する動作周波数 (動作電源電圧) がすべて終了した場合は、実験を終了する。

このようにして得られる歩留りの増加量をそれぞれの条件ごとに比較する。

#### 4.3 調整シミュレーション時間

開発したシミュレータを用いて本実験を行ったところ、上記実験手順を 1 試行完了するのにかかった時間は平均 14 時間程度であった。なお、調整シミュレーションを行った際に使用した計算機の仕様は、CPU が Intel (R) Xeon (R) の 3.40 GHz、メモリが 4.0 GB、OS が SUSE LINUX EnterpriseServer9 である。最適な調整箇所数を確認するシミュレーション実験にかかる時間としては、十分現実的な時間だといえる。仮想 LSI の数を増やすことや検査する動作周波数 (動作電源電圧) の刻みを細かく設定することで、より正確な情報を得ることが可能だが、それに反して、調整シミュレーションにかかる時間が増大してしまう。このため、仮想 LSI の数については、全設計工程内で許される時間内で調整シミュレーション実験が終了するように決定する必要がある。

#### 4.4 調整箇所の限定による歩留り向上効果

限定した調整箇所の個数  $N$  について、100, 300, 600, 1,031 (全箇所調整) の 4 通りを設定し、それぞれの調整実験を行った。

図 12 に動作周波数方向における調整実験、図 13 に動作電源電圧方向における調整実験の結果を示す。横軸には動作周波数 (動作電源電圧)、縦軸にはチップの歩留りを設定し、調整実験で求めた調整前の平均歩留り、調整箇所数ごとの調整後の歩留りをそれぞれ折れ線グラフで表している。図 12、図 13 いずれにおいても 1,031 カ所の全箇所調整ではほとんど歩留りが向上していないことが分かる。これは、探索空間が広いために、3,000 回の打ち切り条件では GA が十分に収束していないためと考えられる。それに比べて、調整箇所を限定した場合の調整では歩留りの向上が確認でき、調整の効果を得られることが分かる。たとえば、図 13 の動作電源電圧 0.85 V での歩留りを比較すると、1,031 カ所の調整では歩留りがおよそ 10% 程度しか上昇していないのに対し、100 カ所や 300 カ所の調整では歩留りが約 80% にまで上昇している。これらの結果は、探索空間を縮小することで、GA の収束速度を向上させることができたためである。

#### 4.5 GA 初期集団の分布を改良したときの効果

次に、調整で効果の大きかった 100 カ所と 300 カ所

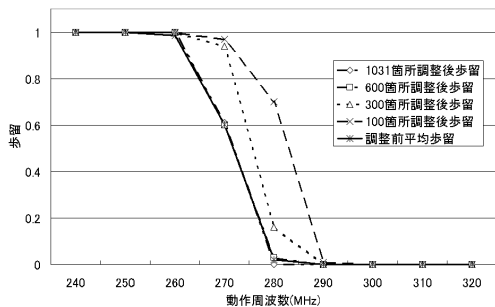


図 12 動作周波数方向における調整箇所限定時の調整実験結果  
 Fig. 12 Adjustment results using reduced adjustment points when operational frequency is raised.

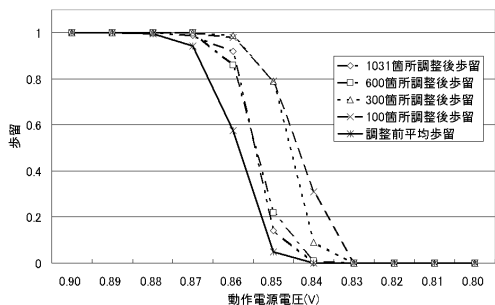


図 13 動作電源電圧方向における調整箇所限定時の調整実験結果  
 Fig. 13 Adjustment results using reduced adjustment points when Vdd is lowered.

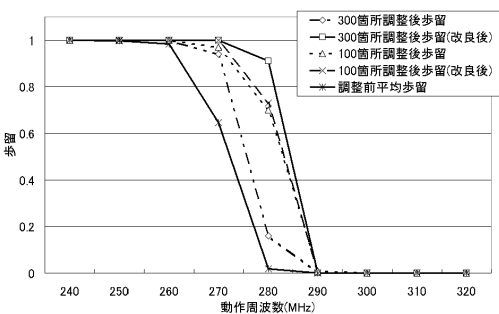


図 14 動作周波数方向における GA 改良時の調整実験結果  
 Fig. 14 Adjustment results using improved GA when operational frequency is raised.

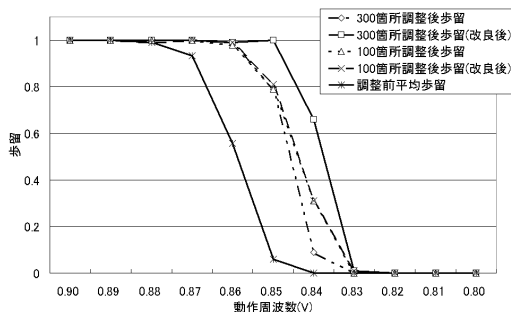


図 15 動作電源電圧方向における GA 改良時の調整実験結果  
 Fig. 15 Adjustment results using improved GA when Vdd is lowered.

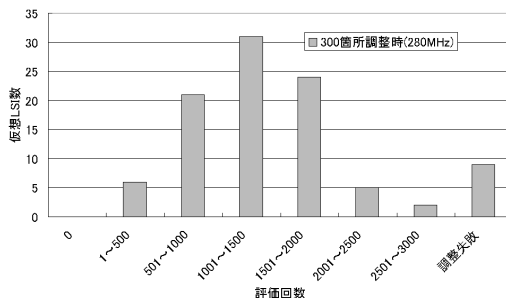


図 16 動作周波数 280 MHz における調整に要した評価回数の度数分布  
 Fig. 16 Frequency distribution of evaluation iterations for adjustment at operational frequency 280 MHz.

波数が 280 MHz のときの、調整完了までにかかった評価回数ごとに度数分布表を作成した (図 16)。

図 14, 図 15 のグラフより, GA の初期集団の分布を改良することで, 300 カ所の調整での歩留りが大幅に上昇していることが分かる。特に, 図 14 の動作周波数 280 MHz でのチップの歩留りをみると, 従来の GA による 300 カ所の調整では約 15% であった歩留りを, 本改良を加えた GA による 300 カ所の調整を行うことで, 歩留りを約 90% にまで向上させている。これは, 図 16 に示すように, 本手法を用いることで GA が 3,000 回の打ち切り回数以内で十分に収束できたためである。一方, 100 カ所の調整の歩留りが初期集団の分布改良後にほとんど上昇していない。これは, 100 カ所の調整では必要な遅延回路が足りず, 歩留りをこれ以上向上させられないことが考えられる。

以上より, 調整箇所を 300 カ所に限定した場合の提案手法による調整効果が最も大きかった。調整効果について, 調整前と調整後のグラフの横軸方向の移動量を観察すると, 動作周波数方向で 10% 程度の向上, 動作電源電圧方向で 5% 程度の低減が見込める。また, 調整時間に関しても, 1 回の評価に 1 ミリ秒要すると

での調整実験において, GA 初期集団の分布を改良したときの調整実験を行った。図 14 に動作周波数方向における調整実験, 図 15 に動作電源電圧方向における調整実験の結果を示す。横軸には動作周波数 (動作電源電圧), 縦軸にはチップの歩留りを設定し, 前実験で比較的效果の見られた 100 カ所調整の歩留りと 300 カ所調整の歩留りに加え, 本実験で求めた GA 初期集団の分布を改良したときの 100 カ所調整の歩留りと 300 カ所調整の歩留りをそれぞれ折れ線グラフで表している。また, 300 カ所の調整実験において, 動作周



仮定すると、1チップあたり3秒以内に調整を完了させることができる。この結果、提案手法を用いた調整は実用的な機能ブロックの調整に有効であるといえる。

## 5. おわりに

本研究では、実用的な規模の回路を現実的な時間内で調整できる新たな製造後クロック調整手法を提案した。具体的には、STAの結果を用いてLSI内部の調整箇所を限定し、GAの初期集団の分布を改良することで調整時間を大幅に短縮した。また、調整の効果をLSI設計時に短期間でフィードバックすることのできるシミュレータを開発した。

提案手法の有効性を検証するために、1,031カ所のフリップフロップが存在する実用的なLSIの設計情報を基に、調整実験用LSIを設計した。調整シミュレーションの結果、調整時間が数秒以内という現実的な制約の下では、従来手法ではまったく見られなかった調整効果が、動作周波数方向および動作電源電圧方向の両方において、確認することができた。たとえば、動作周波数280MHzにおいて調整前歩留りはほぼ0であったが、提案手法により歩留りを約90%にまで向上させることができた。また、動作電源電圧0.85Vにおいて調整前の歩留りは5%程度であったが、提案手法によりすべてのチップを動作させることができた。さらにこれらの調整シミュレーションは、開発したシミュレータを用いて1試行14時間程度で完了できたので、LSIの設計フローに組み込むことができる。

これらの結果より、実用的な規模の回路に対して、製造後クロック調整技術を初めて適用可能となる見通しを得た。現在調整実験用のLSIを試作し、上記シミュレーションの効果を実証すべく実験を進めている。

謝辞 本研究の一部は半導体MIRAIプロジェクトの一部として、NEDO(新エネルギー・産業技術総合開発機構)からの委託により実施した。

## 参考文献

- 1) Rabaey, J., Chandrakasan, A. and Nikolic, B.: *Digital Integrated Circuits*, Prentice-Hall, Englewood Cliffs, NJ (2003).
- 2) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley (1989).
- 3) Takahashi, E., Murakawa, M., Toda, K. and Higuchi, T.: An evolvable-hardware-based clock architecture toward gigaHz digital systems, *Proc. AAAI Genetic Algorithm and Evolutionary Computation Conference (GECCO '99)*, pp.1204–1210 (1999).
- 4) Takahashi, E., Kasai, Y., Murakawa, M. and Higuchi, T.: Post-Fabrication Clock-Timing Adjustment Using Genetic Algorithms, *IEEE Journal on Solid-State Circuits*, Vol.39, No.4, pp.643–650 (2004).
- 5) 高橋篤司:[招待講演]大域クロックを用いた一般同期回路—設計方法論,それらを支えるツール群,今後の展望,情報処理学会研究報告(2006-SLDM-126), Vol.2006, No.111, pp.159–164 (2006).
- 6) Sakurai, T.: LSI design toward 2010 low-power technology, *International Conference on VLSI & CAD '99*, pp.325–334 (1999).
- 7) Masuda, H., Okawa, S. and Aoki, M.: Approach for Physical Design in Sub-100nm Era, *Proc. ISCAS 2005* (2005).
- 8) Schwefel, H.P.: *Evolution and Optimum Seeking*, John Wiley & Sons (1995).
- 9) Yamauchi, H., et al.: An 81MHz, 1280 × 720 pixels × 30 frames/s MPEG-4 Video/Audio Codec Processor, *ISSCC Digest of Technical Papers*, pp.130–131 (2005).

(平成19年2月2日受付)

(平成19年3月23日再受付)

(平成19年4月12日採録)



諏佐 達也

1981年生。2006年東邦大学大学院理学研究科情報科学専攻博士前期課程修了。現在、同研究科情報科学専攻博士後期課程在学中。LSIの製造後調整の研究に従事。



村川 正宏

1972年生。1999年東京大学大学院工学系研究科博士課程修了。工学博士。同年電子技術総合研究所(現産業技術総合研究所)入所。現在、情報技術研究部門主任研究員。東邦大学連携大学院准教授兼任。遺伝的アルゴリズム,進化的ハードウェア,強化学習の研究に従事。第2回進化システム国際会議最優秀論文賞,平成12年度つくば奨励賞,電気学会ミレニアム最優秀論文賞受賞。



高橋 栄一（正会員）

1993 年東京大学大学院工学系研究科博士課程修了。同年電子技術総合研究所（現産業技術総合研究所）入所。並列・実時間処理，適応型デジタル回路構成法，ネットワークセキュリティシステムの研究に従事。東邦大学連携大学院教授兼任。電子情報通信学会，IEEE 各会員。工学博士。



古谷 立美（正会員）

1947 年生。1973 年成蹊大学大学院工学研究科電気工学専攻修士課程修了。同年電子技術総合研究所入所。知識処理マシンの研究開発に従事。1994 年より東邦大学教授。ネットワークと知識処理を組み合わせた応用研究に従事。工学博士。1993 年市村賞。



樋口 哲也

1982 年慶應義塾大学大学院工学研究科博士課程修了。工学博士。1983 年電子技術総合研究所入所。1990～1991 年カーネギーメロン大学客員研究員。現在，産業技術総合研究所情報技術研究部門主幹研究員。筑波大学連携大学院教授兼任。進化型ハードウェア，遺伝的アルゴリズムの研究に従事。第 25 回市村学術賞，電気学会ミレニアム最優秀論文賞等受賞。電子情報通信学会会員。



古市 慎治

1964 年生。1988 年名古屋工業大学電子工学科卒業。同年三洋電機（株）入社。以来，研究開発本部でシステム LSI 設計技術，ミックスシグナル設計技術，低電力設計技術の研究開発に従事。現在，三洋電機（株）研究開発本部デバイス開発センター，ミックスシグナル開発部主管研究員。IEEE SSCS 会員。平成 16～18 年まで，技術研究組合超先端電子技術開発機構研究員。



上田 佳孝

1964 年生。1989 年岐阜大学大学院工学研究科電気工学専攻修士課程修了。同年三洋電機（株）入社。以来，研究開発本部でシステム LSI 設計技術，ミックスシグナル設計技術，低電力設計技術の研究開発に従事。現在，三洋電機（株）研究開発本部デバイス開発センター，ミックスシグナル開発部課長。電子情報通信学会会員。2004 年から，集積回路研究専門委員会専門委員。



和田 淳

1963 年生。1987 年京都大学理学部物理学卒業。同年三洋電機（株）入社。以来，研究開発本部で CMOS アナログ回路設計，ミックスシグナル設計技術，低電力設計技術の研究開発に従事。現在，三洋電機（株）研究開発本部デバイス開発センター，ミックスシグナル開発部担当部長。電子情報処理学会会員。1999～2002 年まで，IEEE VLSI Circuit Symposium のプログラム委員会メンバ。