

# 低電圧・長寿命動作に向けた クリティカルパス・アイソレーション手法

増田 豊<sup>1,a)</sup> 尾上 孝雄<sup>1</sup> 橋本 昌宜<sup>1,b)</sup>

**概要:** 半導体製造プロセスの微細化に伴い、製造ばらつきや経年劣化による回路性能のばらつきが顕在化している。本研究は、性能ばらつきへの対策としてクリティカルパス・アイソレーションに着目する。アイソレーションは、本質的なクリティカルパス以外にスラックを与えて、遅延故障発生率を削減する。本稿では、低電圧・長寿命動作可能なアイソレーション手法を提案する。提案手法は、整数線形計画法を用いて、ゲートの故障率の総和を最大限削減し得る FF 組を選択する。アイソレーションの効果を実験で評価したところ、Vdd の 25% の削減効果を実験的に確認した。同一の動作電圧で動作させた場合、MTTF(平均故障発生時間)を 14 桁以上向上させた。面積オーバーヘッドは 1.4% であった。

## Critical Path Isolation for Time-to-Failure Extension and Lower Voltage Operation

MASUDA YUTAKA<sup>1,a)</sup> ONOYE TAKAO<sup>1</sup> HASHIMOTO MASANORI<sup>1,b)</sup>

**Abstract:** Device miniaturization due to technology scaling has made manufacturing variability and aging more significant, and lower supply voltage makes circuits sensitive to dynamic environmental fluctuation. These may shorten the time to failure (TTF) of fabricated chips unexpectedly. This paper focuses on critical path isolation, which increases timing slack of non-intrinsic critical paths and decreases timing error occurrence probability in the circuit, and proposes a design methodology of isolated circuits for TTF extension and/or lower voltage operation. The proposed methodology selects a set of FFs for isolation using ILP so that it maximumly reduces the sum of gate-wise failure probabilities. We evaluated MTTF (Mean Time To Failure) of circuits with/without critical path isolation and examined how much supply voltage could be reduced without MTTF degradation. Evaluation results show that circuits with the proposed critical path isolation achieved 25% supply voltage reduction with 1.4% area overhead. With the same supply voltage, MTTF was improved by 14 orders of magnitude.

### 1. 序章

同期式順序回路では、組み合わせ論理回路の信号伝播時間がクロック周期を超過し、FF に本来の値が取り込まれなかった際に、遅延故障が起こる。信号伝播時間は、製造ばらつき、電源ノイズ等の動作環境変動や経年劣化に応じて変動する。製造ばらつきとは、閾値電圧、ゲート長、酸化膜厚等のデバイスパラメータや配線形状が設計時の想定値からずれる現象を指す。近年の半導体製造プロセスの微細化に伴い、素子特性と劣化特性のばらつきや電源ノイズによる回路遅延変動が複雑化している。これらの遅延ばら

つきは回路内で遅延故障が起こる可能性を高め、チップが故障するまでの時間 (TTF : Time To Failure) を短くする。

製造ばらつき、動作環境変動や経年劣化による遅延故障を防止するために、従来設計では一定の動作電圧/速度マージンを設けている。しかし、性能ばらつきの複雑化の結果、より多くのマージンが必要となり、達成できる回路性能が大きく低下する。マージンを必要最小量にするためには、設計時にチップの TTF を適切に見積もる必要がある。

図 1 に、製造ばらつきと経年劣化により TTF がばらつく様子を示す。製造ばらつきについては多くの研究がなされ、確率的なモデル化が一般化している。また、経年劣化の一つである NBTI (Negative Bias Temperature Instability) [1][2] に関しても、劣化のばらつきは確率的に取り扱われている。これらの確率的なばらつきにより、TTF も図 1 に

<sup>1</sup> 大阪大学大学院情報科学研究科  
Dept. Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University

<sup>a)</sup> masuda.yutaka@ist.osaka-u.ac.jp

<sup>b)</sup> hashimoto@ist.osaka-u.ac.jp

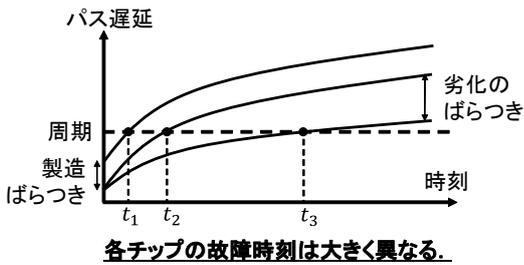


図1 製造ばらつきと経年劣化による TTF のばらつき。

示すように  $t_1$ ,  $t_2$ ,  $t_3$  とばらつく。チップの TTF を適切に見積もるためには、上記の確率的なばらつきを考慮可能な手法が必要である。近年、TTF の平均 (MTTF: Mean TTF) を確率的に見積もる手法が飯塚らによって提案されている ([3][4])。この手法では、製造ばらつき、電源ノイズ等の動作環境変動や経年劣化を考慮して MTTF を算出する。したがって、例えば、動作電圧と MTTF のトレードオフを取得できる。

本研究では、所望の MTTF を最小動作電圧で達成可能な回路設計手法を提案する。提案手法は、クリティカルパス・アイソレーション (Critical Path Isolation: 以下 CPI と略す) を利用する。CPI は、本質的でないクリティカルパスにスラックを与え、本質的なクリティカルパスと同等の遅延を持つパス数を削減する。本質的なクリティカルパスとは、パス遅延をそれ以上小さく出来ないクリティカルパス (例えば、低  $V_{th}$  セルのみで構成されたクリティカルパス) を意味する。CPI 回路では、本質的なクリティカルパス以外では遅延故障が起こりにくく、より長い TTF が期待される。本研究では、各 FF にスラックを付加して FF 単位の CPI を行う手法を提案する。提案手法では、対象 FF を整数線形計画法を用いて選択することで、遅延故障率を最大限削減する。FF 選択後、各 FF に付加可能な最大限のスラックをそれぞれ与える。製造ばらつき、電源ノイズと NBTI 劣化を考慮して MTTF を評価した結果、提案手法により生成した CPI 回路は、MTTF を維持しつつ動作電圧を 25.0% 削減した。

本稿の構成は以下の通りである。2 章で CPI と関連研究を紹介し、CPI 手法の最適化問題を定式化する。3 章で、提案手法を説明する。4 章で、CPI による  $V_{dd}$  削減/MTTF 延長効果を実験的に評価し、5 章で結論を述べる。

## 2. CPI 問題の定式化

本章ではまず、CPI について説明し、関連研究を紹介する。次に、所望の MTTF を満足しつつ動作電圧を削減する CPI 手法を、最適化問題として定式化する。

### 2.1 CPI

CPI は本質的でないクリティカルパスのスラックを増加させる。図 2(a) に、従来回路のパス遅延分布を示す。従来の回路設計フローでは、消費電力と面積を削減するため、クリティカルパス以外のパスに含まれるセルを、より小さ

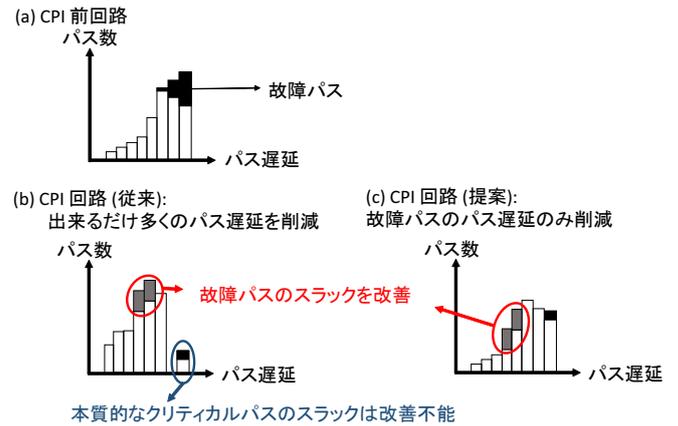


図2 パス遅延分布。(a) CPI 前、(b) CPI (従来)、(c) CPI (提案手法)。

な/高  $V_{th}$  セルに置き換える。従って、クリティカルパスの遅延に近いパスの数が増加する。一方、この置換によりスラックの小さいパス数が増加するため、遅延ばらつき時には遅延故障率がより高まる。前章で述べたように、回路遅延は製造ばらつき、電源ノイズや経年劣化により大きくばらつくため、従来設計では、タイミング違反が発生する危険性が高い。

一方、CPI 回路では、スラックが増加したパスは遅延ばらつき時であっても遅延故障が起こりにくい。図 2(b) に CPI 回路のパス遅延分布を示す。従来設計と比べて、CPI 回路では本質的でないクリティカルパスのタイミング違反率を削減出来る。しかし、従来設計で得られる電力/面積削減を諦める必要がある。

以上より、CPI を用いる際には、遅延故障率、面積や電力に関するトレードオフから、最良の設計を選択することが重要である。選択した回路は、TTF/面積/電力に対する要求を満足する必要がある。[5] で提案された従来 CPI 手法では、出来るだけ多くのパスにスラックを与えるため (図 2(b))、面積オーバーヘッドが非常に大きいという問題があった。CPI により得られる効果を維持しつつ、そのオーバーヘッドを削減するため、本研究では遅延故障に貢献するパス群を抽出し、それらに対してのみ、CPI を行う (図 2(c))。提案 CPI 手法は、3 章で詳述する。

### 2.2 関連研究

本節では、2 つの関連研究を紹介する。一つ目は CRISTA (CRITICAL path ISolation for Timing Adaptiveness) [5] であり、著者らが知る限り、CPI を用いた唯一の研究である。二つ目は確率的故障率見積もり手法 [3][4] である。本研究ではこの見積もり手法を用いて MTTF を算出する。

#### 2.2.1 CRISTA

Ghosh らが提案した、CRISTA と呼ばれる設計手法は、CPI により遅延故障の起こりにくい回路を生成し、その結果、動作電圧の削減を可能とする。CRISTA の重要なポイントは以下の通りである [5]。

- CPI により、非クリティカルパスのスラックを増加

する。

- CPI 後、電圧を下げて動作する。低電圧動作時には、遅延故障し得る命令のみ 2 サイクル かけて実行し、それ以外は 1 サイクルで実行する。
- クリティカルパスの活性化率を削減し、2 サイクル動作の割合を低減 ([6] と同様の手法を利用)。

文献 [5] では、平均 60% の電力削減が、18% の面積オーバーヘッドで達成できることを実験的に確認した。しかし、CRISTA の電力削減に対する有効性は、静的な製造ばらつきの下でのみ評価されており、動作環境変動や経年劣化といった動的なばらつきは考慮されていない。設計時にチップの寿命を正確に見積もり、必要最小限のマージンを設定するためには、動的なばらつきの考慮が不可欠である。

一方、本研究では、所望の MTTF を満足しつつ、動作電圧を最大限削減可能な CPI 手法を提案する。MTTF 評価時には、製造ばらつき、電源ノイズと NBTI を考慮する。

### 2.2.2 確率的遅延故障率見積もり手法

飯塚らは、[3][4] で、回路の MTTF を高速に算出可能な確率的遅延故障率見積もり手法を提案した。TTF 評価手法の一例として、ゲートレベルシミュレーションの実行が挙げられる。しかし、実際に遅延故障の起こる割合は非常に低く \*1、これらの故障を再現するのに要するシミュレーション時間は極めて長い。例として、1 ヶ月の MTTF を評価するのに  $10^8$  年以上のシミュレーション時間が必要である [3]。そこで、[3] では、動的な遅延ばらつきの下での回路動作を連続時間マルコフ過程を用いて表す手法を提案し、論理シミュレータと比較して  $10^{12}$  倍の MTTF 推定高速化を達成した。さらに [4] では、[3] の提案手法を拡張し、製造ばらつきや経年劣化の影響が考慮可能な MTTF 導出手法を実現した。本研究では、[4] の手法を用いて CPI 回路の性能評価を行う。

### 2.3 CPI 手法の定式化

図 2 ではパス単位での CPI を説明したが、パス単位での CPI は、回路内に含まれるパス数が膨大であるため、計算時間の観点で効率的ではない。そこで本研究では、図 3 の、FF 単位での CPI 手法を考える。本手法ではまず、(1)  $i$  番目の FF のセットアップ制約を  $\Delta setup_i$  だけ増加し、ECO (engineering change order) により、設計を更新する。次に、(2) 厳しくした制約を元に戻す。(1)(2) により、 $i$  番目の FF を終端とするパスに  $\Delta setup_i$  以上のスラックを付加する。

次に、全ての FF が MTTF に同程度の影響を及ぼすわけではないことを示す。図 4 に OR1200 OpenRISC プロセッサ内の FF の故障率を示す。これらの故障率は、タイミング違反率と活性化率の同時確率により算出する。図 4 より、いくつかの FF が突出して高い故障率を持ち、これらが MTTF を決定している。そこで本研究では、MTTF に大きな影響を与える少数の FF を抽出し、それらに CPI を適用する。この場合、面積オーバーヘッドを必要最小限に抑え

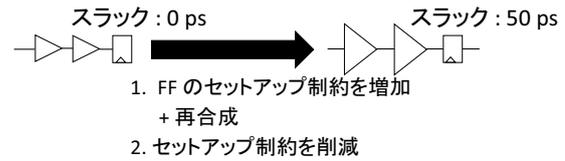


図 3 FF 単位での CPI。

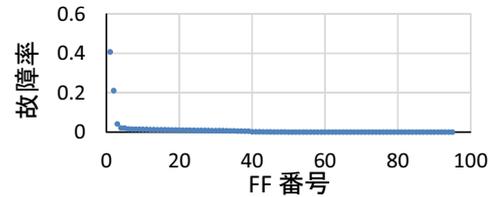


図 4 FF 毎に大きく異なる故障率。

ることが出来る。

以上より、CPI 問題を以下のように定式化する。

- 目的関数
  - Minimize :  $V_{dd}$
- 制約条件
  - $MTTF \geq MTTF_{min}$
  - $Area \leq Area_{max}$
- 変数
  - $\Delta setup_i$  ( $1 \leq i \leq N_{FF}$ )

この最適化問題の目的は、MTTF の下限 ( $MTTF_{min}$ ) と面積の上限 ( $Area_{max}$ ) を満足しつつ動作電圧を最小化することである。変数  $\Delta setup_i$  は、 $i$  番目の FF に与えるスラック、 $N_{FF}$  は回路内の FF の総数である。  $\Delta setup_i=0$  は、 $i$  番目の FF が CPI 対象の FF でないことを意味する。つまり、対象 FF の総数  $N_{CPI}$  は、0 より大きい  $\Delta setup_i$  を持つ FF の総数に等しい。ここで、MTTF は  $\Delta setup_i$ 、 $V_{dd}$  に依存し、これらの関係は、前節の確率的故障率見積もり手法を用いて取得できる。Area は  $\Delta setup_i$  に依存し、この値は ECO 再合成することで取得する。

## 3. 提案 CPI 手法

### 3.1 概略

上記の最適化問題では、 $MTTF$ 、 $Area$ 、 $\Delta setup_i$  が非線形の関係を持ち、 $MTTF$ 、 $Area$  の評価に比較的長い CPU 時間を要する。そこで本研究では、以下の解法を用いる。

様々な  $N_{CPI}$  に対して、MTTF を最大化する FF 組と  $\Delta setup_i$  を決定する。これは、MTTF の長い回路はより大きく  $V_{dd}$  が削減可能である、という仮定に基づく。次に、各  $\Delta setup_i$  組を用いて ECO 再合成し、Area を取得する。その後、確率的故障率見積もり手法を用いて  $V_{dd}$  と  $MTTF$  のトレードオフを評価する。評価結果より、 $MTTF$  と Area に関する制約を満足しつつ  $V_{dd}$  を最小化する  $\Delta setup_i$  組を決定する。

本手法では、与えられた  $N_{CPI}$  に基づき、CPI 対象 FF を選択し、各 FF に対して  $\Delta setup_i$  を決定する。3.2 節で対象 FF の選択、3.3 節で  $\Delta setup_i$  の決定方法をそれぞれ述

\*1 頻繁に遅延故障が起こる回路は、実用的な設計とは言えない。

べる。

### 3.2 CPI 対象 FF の選択

まず、MTTF 最大化に向けた、対象 FF の選択方法を提案する。提案手法は、各ゲートの故障率の総和を最小化するように FF を選択する。例として、図 5 に、10 個の組み合わせ論理セルと 4 個の FF から構成される回路を示す。各ゲートの下に記載されている数は故障率である。この故障率の算出方法は後述する。この例では  $N_{CPI} = 2$  とする。

図 5(a) のように FF2 と FF4 のスラックを増加すると、L1, L3, L4, L5, L6, L7, L9, L10 の 8 つのセルのスラックが増加する。この場合、これらの 8 つのセル遅延がばらついたとしても、増加したスラックにより遅延故障発生が防止できる。従って、 $0.21 (= 0.02 + 0.02 + 0.02 + 0.03 + 0.03 + 0.03 + 0.03)$  だけ故障率を削減出来る。一方、図 5(b) のように FF1 と FF2 を選択した場合は、L1, L2, L3, L4 のスラックのみが増加するため、故障率の削減は  $0.08 (= 0.02 \times 4)$  に留まる。この場合、TTF は短くなる。

本研究では、この FF 選択問題の厳密な解を得るため、以下のように整数線形計画問題に帰着させて解を求める。

- 目的関数
  - Maximize :  $\sum_{k=1}^{N_{inst}} (inst\_fail_k \times inst_k)$
- 制約条件
  - $0 \leq inst_k \leq 1 \quad (1 \leq k \leq N_{inst})$
  - $0 \leq FF_i \leq 1 \quad (1 \leq i \leq N_{FF})$
  - $\sum_{i=1}^{N_{FF}} FF_i \leq N_{CPI}$
  - $inst_k \leq \sum_{i=1}^{N_{FF}} (FF_i \times FF\_inst_{i,k})$
- 変数
  - $FF_i \quad (1 \leq i \leq N_{FF})$

回路内のインスタンスと FF の総数をそれぞれ  $N_{inst}$ ,  $N_{FF}$  とする。この整数線形計画問題は、 $(inst\_fail_k \times inst_k)$  の総和の最大化を目的とする。 $inst\_fail_k$  は、 $k$  番目のゲートの故障率を表す。 $inst_k$  は、 $k$  番目のインスタンスが対象 FF を終端とするいずれかのパスに含まれるか、を表すバイナリ変数であり、含まれる場合 1 をとる。つまり、 $inst\_fail_k \times inst_k$  の総和は、CPI により削減された故障率

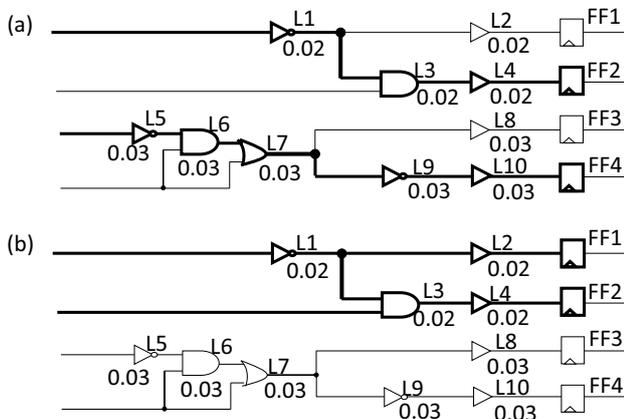


図 5 CPI 対象 FF の選択例。

を表す。本問題では、バイナリ変数  $FF_i$  に対して、 $i$  番目の FF が対象 FF に含まれる時のみ 1 を与える。

第一、第二の制約条件により、 $inst_k$  と  $FF_i$  をバイナリ変数として扱うことが出来る。第三の制約は対象 FF の数が  $N_{CPI}$  以下であることを意味する。第四の制約は  $inst_k$  と  $FF_i$  の関係を表す。 $FF\_inst_{i,k}$  は回路構造により決定されるバイナリ変数であり、 $k$  番目のインスタンスが  $i$  番目の FF を終端とするいずれかのパスに含まれる場合 1 をとる。 $inst_k$  は  $FF_i$  と  $FF\_inst_{i,k}$  の積が、全ての FF に対して 0 の時のみ 0 を取る。一方、もし  $k$  番目のインスタンスが対象 FF を終端とするパスに含まれる場合、 $FF_i$  と  $FF\_inst_{i,k}$  の積のいずれかは必ず 1 になる。この場合、 $inst_k$  は 0 でも 1 でも制約は満足するが、目的関数で  $(inst\_fail_k \times inst_k)$  の最大化を目指しているため、 $inst_k$  は必ず 1 となる。

残る問題は  $inst\_fail_k$  の算出である。[4] に従って MTTF を導出すると、各 FF の故障率  $FF\_fail_i$  が算出される。本研究では、この  $FF\_fail_i$  を用いて  $inst\_fail_k$  を取得する。

$$inst\_fail_k = \max_i \left( \frac{FF\_fail_i}{\sum_{i=1}^{i_{max}} (FF\_inst_{i,k})} \right) \quad (1 \leq i \leq N_{FF}) \quad (1)$$

上の式は、ある FF を終端とするパスに含まれるインスタンスが、その FF の故障率に等しく貢献する、という仮定に基づいている。インスタンスが、複数の FF を終端とするパスに含まれる場合を考慮するために、 $\max$  演算を行っている。

### 3.3 $\Delta setup_i$ の決定

次に、前節で選択した FF に対して  $\Delta setup_i$  を決定する。図 6 に、 $\Delta setup_i$  と再合成後のスラックの関係を示す。スラックを無限に増加することは出来ず、上限 ( $\Delta slack_i^{UB}$ ) が存在する。本研究では、簡単化のため、 $\Delta setup_i$  に  $\Delta slack_i^{UB}$  を与える。実際には、 $\Delta setup_i$  の最適値は 0 から  $\Delta slack_i^{UB}$  の間に存在する。この区間からの最適な  $\Delta setup_i$  の決定は、今後の課題の一つとする。

## 4. 評価結果

### 4.1 評価環境

本研究では、CPI 前回路として OR1200 OpenRISC プロセッサを用いた。本プロセッサの RTL ネットリストを NanGate 45nm Open Cell Library と商用ツールを用いて論理合成し、評価に用いた。合成後ネットリストは 24000 個のスタンダードセルと 2500 個の FF を持つ。つまり、 $N_{inst} = 21500$ ,  $N_{FF} = 2500$  である。整数線形計画問題を解くにあたって、Gurobi Optimizer 6.5 を用いた。このソルバを、OS が Red Hat Enterprise Linux 6 で 1024 GB のメモリを搭載した、2.4 GHz Xeon CPU マシンで実行した。実行時間は最大 0.05 秒であった。

OpenRISC のワークロードとして、MIBenchmark[9] から CRC32, SHA1, Dijkstra の 3 プログラムと 30 種類の入力

データパターン (計 90 ワークロード) を用いた。

有意な MTF の算出には、現実的な遅延ばらつきの考慮が不可欠である。本評価では以下のばらつきを考慮した。

- 電源ノイズ (-50mV から +50mV までランダムに変動)
- チップ内+チップ間ばらつき
- NBTI 劣化 ([7] の実測劣化データに、T/D モデル ([8]) でフィッティングし、劣化特性を得た。)

本評価では、クロック周期、 $MTTF_{min}$  をそれぞれ 2.1ns、 $1.00 \times 10^{16}$  サイクル (= 8.0 ヶ月) とした。  $Area_{max}$  は CPI 前回路の 101.5%, 103%, 103.5% とした。 MTF 評価において、遅延故障が起こらない状況 ( $MTTF = \infty$ ) が発生した。これらの結果も図中に含めるため MTF の最大値を  $1.00 \times 10^{17}$  (= 3.3 年) とした。

#### 4.2 評価結果

提案手法で 10/20/30 個の FF を CPI した回路を用いて ( $N_{CPI} = 10/20/30$ )、 $N_{CPI}$  と面積オーバーヘッドの関係を調べた。  $Area_{max} = 103.5%/103%/101.5%$  を満足するためには、 $N_{CPI}$  はそれぞれ 30/20/10 以下の必要がある (図 7)。

図 8 に MTF の評価結果を示す。  $V_{dd}$  は 1.2V から 0.85V まで 50mV 刻みの 9 種類を設定した。図 8 より、 $N_{CPI} = 10$  の CPI 回路が、0.9V において  $MTTF_{min}$  を達成した。つまり、所望の MTF を維持しつつ、 $V_{dd}$  を 25.0% 削減している。これは、動的電力の 44% の削減に相当する。 MTF に関しては、 $N_{CPI} = 10$  の回路において、0.9V 動作時に  $1.38 \times 10^2$  サイクルから  $1.00 \times 10^{17}$  サイクルまで、 $7.24 \times 10^{14}$  倍の延長効果を達成した。

以上より、提案手法は、CPI による面積オーバーヘッドを数% に抑えつつ、大きな省電力効果/MTF 延長効果を達成出来ることを示した。 MTF の長い設計と故障率の低い設計は等価なため、CPI 回路は、発生した故障を検出して回復する Razor[10]、TRC (Tunable Replica Circuit) [11] や故障発生を予測する TEP-FF (Timing Error Prediction

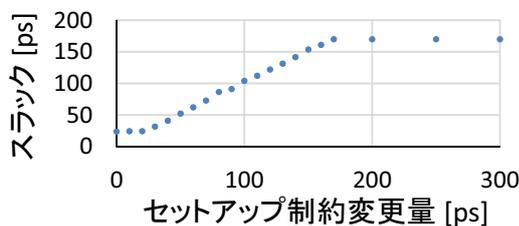


図 6 制約変更量  $\Delta setup$  と合成後のスラックの関係。

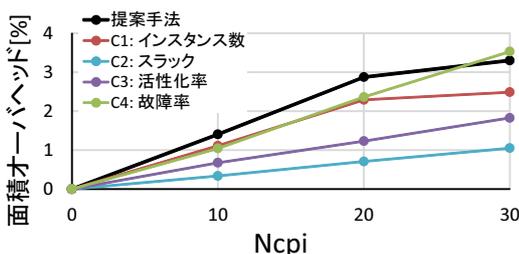


図 7 CPI による面積オーバーヘッド。

Flip-Flop)[12] の、適用対象回路としても望ましい。

#### 4.3 比較手法

次に、提案手法と以下の 4 手法を比較した。

- C1: タイミングクリティカルでないセルの数を最大化するように、FF 組を選択 (整数線形計画法を使用)。
- C2: スラックの厳しい FF から選択。
- C3: 活性化率の高い FF から選択。
- C4: 故障率の高い FF から選択。

C1 は、CPI によりスラックが増加するセル数の最大化と、CPI 回路の MTF の最大化には相関があると考えている。本提案手法と C1 との主な差は、C1 が FF の故障率を考慮していない点である。 C2 は、スラックの厳しい FF が最も故障しやすいと想定している。この C2 は、STA (もしくは SSTA) のみで対象 FF を決定できるため、実装が容易である。 C3 は、活性化率に重点を置いている。もしタイミングクリティカルな FF であっても、活性化されなければ遅延故障は起こり得ない。活性化率を取得するためには、論理シミュレーションの実行、もしくは信号の伝播確率を確率的に算出 ([13]) する必要がある。 C4 は C2 と C3 を組み合わせた手法である。本研究では故障率を、タイミング違反率と活性化率の同時確率として算出した。

本稿では、タイミング違反率をモンテカルロ統計的静的タイミング解析 (SSTA : Statistical Static Timing Analysis) により算出し、活性化率を論理シミュレーションの遷移時刻と STA のパス遅延の対応付けにより求めた (図 9)。

図 10 に、提案手法と比較手法 (C1, C2, C3, C4) の MTF 比較結果を示す ( $N_{CPI} = 10$ )。図 10 より、提案手法が、MTF と  $V_{dd}$  に関する最も良いトレードオフを示し、最も大きな  $V_{dd}$  削減効果を達成した。提案手法の  $V_{dd}$  削減効果が 25.0% であるのに対し、C1, C2, C3, C4 の削減効果はそれぞれ 8.3%, 0%, 8.3%, 12.5% であった。

図 11 に提案手法と比較手法の  $V_{dd}$  削減効果の概要を示す。  $Area_{max}$  は 101.5%, 103%, 103.5% に設定した ( $N_{CPI}$  は 10, 20, 30)。図 11 より、 $N_{CPI} = 10, 20, 30$  の全てで、提案手法が最も大きな  $V_{dd}$  削減効果を達成した。  $N_{CPI} = 30$  では、C4 と提案手法は同等の効果を達成した。

#### 4.4 提案手法が最も大きな $V_{dd}$ 削減効果を達成した要因

前節より、提案手法に基づき CPI を行った回路が最も大きな  $V_{dd}$  削減効果を示した。本節はその要因を議論する。図 12 に、提案手法、比較手法で CPI を行った回路と、CPI

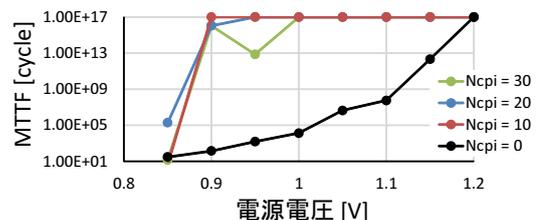


図 8 CPI 回路の MTF (提案手法)。

前回路のパス遅延分布を示す。この分布では、図2と同様、各パスが故障し得るかどうか分類している。図12より、提案手法が故障し得るパス数を最も削減した(144パスから13パスに削減。C4では49パスが故障した)。図13に各FFの故障率を示す。提案手法でCPIを行った回路が、故障率と故障し得るFF数の両方を最も良く削減出来たことが分かる。この故障FF数の削減は、MTTFの延長のみならず、TRC[11]やTEP-FF[12]といった、センサベースの速度制御機構の実装を容易化出来ると期待される。

### 5. 結論

本研究では、TTF延長/低電圧動作を可能とする、CPI手法を提案した。提案手法はFF単位のCPIを採用し、ゲートの故障率の総和を最大限削減するFF組を整数線形計画法により選択した。CPI前後の回路に対してMTTFを評価した結果、提案手法でCPIを行った回路は1.4%の面積オーバーヘッドで25%の $V_{dd}$ 削減効果を達成した。同一の動作電圧で動作させた場合、MTTFを14桁以上向上させた。

### ACKNOWLEDGEMENT

本研究はSTARCとの共同研究による。また、一部ICOMによる研究助成に基づく。

### 参考文献

[1] B. Zang, M. Orshansky, "Modeling of nbtI-induced pmos degradation under arbitrary dynamic temperature variation,"

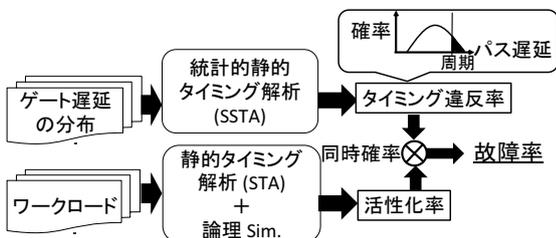


図9 故障率の算出方法。

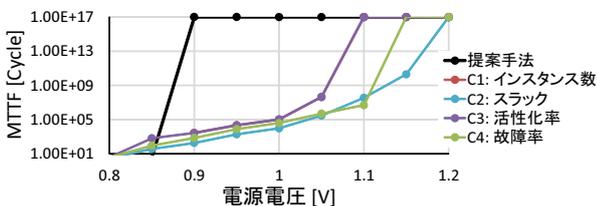


図10 MTTF比較結果 ( $N_{CPI} = 10$ )。

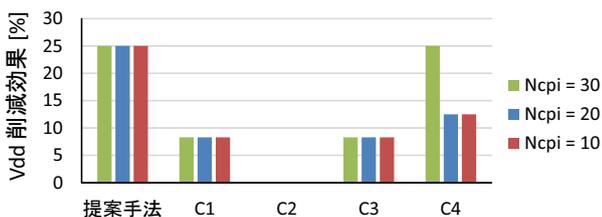


図11  $Area_{min}$  別、 $V_{dd}$ 削減効果。  $Area_{min} = 101.5/103/103.5\%$  ( $N_{CPI} = 10/20/30$ )。

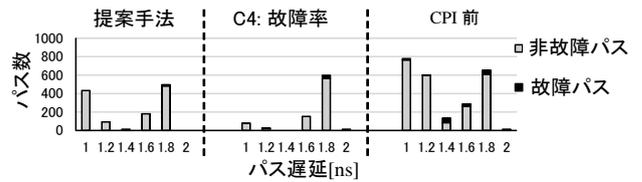


図12 パス遅延分布の比較。

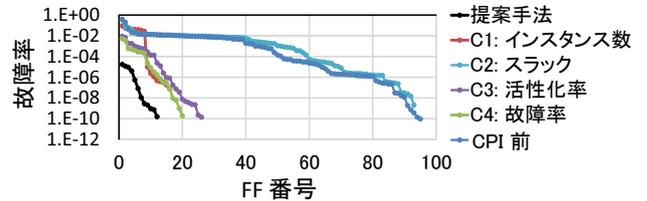


図13 故障率の比較

*Proc. ISQED*, pp.774–779, 2008.

[2] T. Wang, and Q. Xu, "On the simulation of NBTI-Induced performance degradation considering arbitrary temperature and voltage variations," *Proc. DAC*, pp.1–6, 2014.

[3] S. Iizuka, M. Mizuno, D. Kuroda, M. Hashimoto, and T. Onoye, "Stochastic error rate estimation for adaptive speed control with field delay testing," *Proc. ICCAD*, pp.107–114, 2013.

[4] S. Iizuka, Y. Masuda, M. Hashimoto, and T. Onoye, "Stochastic Timing Error Rate Estimation under Process and Temporal Variations," *Proc. ITC*, 2015.

[5] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation," *IEEE Trans. CAD*, vol.26, no.11, pp.1947–1956, Nov. 2007.

[6] X. Bai, C. Visweswariah, P. N. Strenski and D. J. Hathaway, "Uncertainty-aware circuit optimization," *Proc. DAC*, pp. 58–63, 2002.

[7] H. Awano, M. Hiromoto, and T. Sato, "Variability in device degradations: Statistical observation of NBTI for 3996 transistors," *Proc. ESSDERC*, pp.218–221, 2014.

[8] B. J. Velamala, K. B. Sutaria, H. Shimizu, H. Awano, T. Sato, G. Wirth, and Y. Cao, "Compact Modeling of Statistical BTI Under Trapping/De trapping," *IEEE Trans. ED*, vol.60, no.11, pp.3645–3654, 2013.

[9] M.R. Guthaus, J.S. Ringenberg, D. Ernst, T.M. Austin, T. Mudge, and R.B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," *Proc. Workload Characterization*, pp.3–14, 2001.

[10] S. Das, D. Roberts, L. Seokwoo, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE Journal Solid-State Circuits*, vol.41, pp.792–804, 2006.

[11] K. A. Bowman, J. W. Tschanz, S. L. Lu, P. A. Aseron, M. M. Khellah, A. Raychowdhury, B. M. Geuskens, C. Tokunaga, C. B. Wilkerson, T. Karnik, and K. D. Vivek, "A 45nm Resilient Microprocessor Core for Dynamic Variation Tolerance," *IEEE Journal Solid-State Circuits*, vol. 46, no. 1, 2011.

[12] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Adaptive Performance Compensation With In-Situ Timing Error Predictive Sensors for Subthreshold Circuits," *IEEE Trans. VLSI*, vol. 20, no. 2, pp. 333-343, 2012.

[13] F. N. Najm, "Transition density: a new measure of activity in digital circuits," *IEEE Trans. CAD*, vol. 12, no. 2, pp. 310–323, Feb 1993.