

# Partially-Programmable Circuit を 用いた遅延故障の回避手法

春日井 貴通<sup>1,a)</sup> 山下 茂<sup>1</sup> 原祐子<sup>2</sup>

概要：回路の微細化の進展により，経年劣化による遅延故障が発生し，回路の耐故障性が低下している．回路の耐故障性を向上させる手法として，ABB (Adaptive Body Bias) を利用する手法がある．ただし，ABB を利用する手法では，適切な電圧を印加するために，元の回路とは別に，新しい回路が必要である．また，MOS トランジスタに電圧を印加しても回避できない遅延故障があるため，トランジスタレベルでの遅延故障を回避する手法にはたくさんの課題がある．本稿では，回路設計レベルで遅延故障を回避する手法を提案する．回路設計レベルでの遅延故障を回避する手法では，MOS トランジスタに起こる現象に関係なく遅延故障を回避することができる．本稿では，PPC (Partially-Programmable Circuit) を用いて最も多くの遅延故障を回避する回路設計手法を提案する．提案手法をベンチマーク回路に適用して実験を行った．3 入力 of LUT を使用した提案手法では，最大で全体の遅延故障の 48%，4 入力 of LUT を使用した提案手法では，最大で全体の遅延故障の 24% を回避することができた．実験結果より，PPC を利用して回路の遅延故障を回避する手法が有効である場合があることを示すことができた．

## 1. はじめに

近年，パソコンや電子デバイスの普及により，大規模集積回路の小型化や高性能化のニーズが高まっており，回路の微細化の技術が研究されている [5]．一方，回路の微細化により，経年劣化による遅延故障が増えており，耐故障性の低下が課題になっている [2]．

耐故障性を向上させる手法として，ABB (Adaptive Body Bias) を利用する手法がある [9]．ABB では，トランジスタに対して，電圧を印加することで，スイッチング速度を変えることができる．そのため，ABB を利用する手法では，遅延故障を起こしているトランジスタに対して，適切な電圧を印加することで，遅延故障を回避することができる．一方，ABB を利用する手法では，新しく挿入する回路の面積オーバーヘッドと回避できる遅延故障の総数はトレードオフの関係になる．また，さらに回路の微細化が進むと，トランジスタの特性を適切に変化させることができない場合がある．そのため，微細化による遅延故障の影響を受けない回路設計が必要である．

本稿では，回路の微細化に影響されにくい回路設計レベルで遅延故障を回避する．PPC (Partially-Programmable Circuit) を利用して遅延故障を回避する手法を提案する [7]．PPC は，既存の論理ゲートの一部を LUT (Look-Up Table) に置き換え，冗長な配線を追加した回路モデルである．提案手法により，PPC を利用することで，回路設計レベルで

遅延故障を回避することをねらう．本稿では，冗長な配線を既存のゲートの出力のみを用いて，遅延故障を回避する手法を提案する．

提案した既存のゲートの出力のみで遅延故障を回避する手法を，ベンチマーク回路に適用して実験をした．実験の結果，3 入力 of LUT を使用した提案手法では，最大で全体の遅延故障の 48%，4 入力 of LUT を使用した提案手法では，最大で全体の遅延故障の 24% を回避することができた．

本稿の構成は，以下のとおりである．まず，2 節では，遅延故障の種類と，既存手法である ABB について説明する．3 節では，本稿で用いる PPC について説明し，本稿の提案手法について説明する．4 節では，提案手法をベンチマーク回路に適用した回路で実験を行い，考察を述べる．最後に，5 節では，本研究のまとめと今後の課題について述べる．

## 2. 準備

本節では，回路の微細化により，顕在化してきた遅延故障について説明する．2.1 節では，微細化によって顕在化してきた現象について説明する．2.2 節では，トランジスタレベルでの遅延故障を回避する手法である ABB (Adaptive Body Bias) を説明する．

### 2.1 微細化による遅延故障

回路の微細化により，今まで考慮されていなかった現象が顕在化され，その現象により，遅延故障が起きやすくなった．この節では，NBTI (Negative Bias Temperature Instability) と HCI (Hot Carrier Injection) について説明

<sup>1</sup> 立命館大学

<sup>2</sup> 東京工業大学大学院 / JST PRESTO

<sup>a)</sup> center@ngc.is.ritsumeai.ac.jp

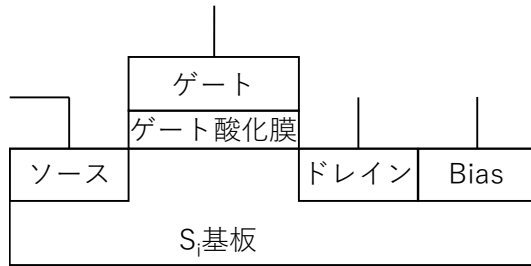


図 1 Body Bias で使われる MOS トランジスタ

する。

NBTI とは、pMOS トランジスタに対して、負の電圧の印加することにより、pMOS トランジスタの特性が変動する現象である [3]。NBTI が起こると、MOS トランジスタの特性が変動するため、遅延故障を引き起こすことがある。この現象は、回路の微細化にともない、pMOS トランジスタの内部の電界強度が大きくなるにつれ、顕在化してきている。

HCI とは、ドレイン付近の電界強度が増大し、Hot Carrier と呼ばれる高いエネルギーをもったキャリアがゲート酸化膜に注入されることにより、MOS トランジスタの特性が変動する現象である [8]。この現象も、NBTI と同様に、回路の微細化に伴い、顕在化してきた現象である。電源電圧の低電圧化は、内部信号のレベルの低下を引き起こすため、低電圧化を行っていくのが難しい傾向にある。その結果、MOS トランジスタの電界強度が増大し、Hot Carrier が存在する可能性が大きくなる。HCI により、ゲート酸化膜が劣化し、MOS トランジスタの閾値電圧が変動するため、遅延故障が起こる。上記で述べた NBTI と同様に、回路の微細化に伴い、HCI が顕在化している。そのため、NBTI と HCI の両方を考慮した設計が必要とされている。

## 2.2 Adaptive Body Bias

NBTI と HCI による遅延故障を回避するために、Body Bias を利用する手法がある [4]。Body Bias とは、MOS トランジスタに繋がっている端子を 1 つ増やし、その端子から MOS トランジスタに対して、順電圧、または逆電圧を印加することで、MOS トランジスタの特性を変化させる技術である。Body Bias を利用した MOS トランジスタは図 1 のようになる。一般的な MOS トランジスタと比べて、Bias という端子が付加されている。Body Bias では、この端子から、MOS トランジスタに対して電圧を印加することができる。

Bias 端子に、順電圧を印加すると、MOS トランジスタのスイッチング速度を高速に動作させることができる。ただし、順電圧を印加することで、リーク電流が増大してしまう問題がある。スイッチング速度とリーク電流の度合いは、順電圧の大きさによって決まる。一方、Bias 端子に逆電圧を印加すると、リーク電流を減らすことができる。ただし、逆電圧を印加することで、MOS トランジスタのスイッチング速度が低下してしまう問題がある。順電圧の場合と同じく、スイッチング速度とリーク電流の度合いは、

逆電圧の大きさによって決まる。

NBTI や HCI による遅延故障を回避するために、Body Bias では、順電圧を利用する。しかし、すべての MOS トランジスタに対して、順電圧を一律に印加すると、リーク電流の増加の問題が起こる。そこで、Body Bias の技術を利用し、なおかつ適切な順電圧を印加するための手法として、ABB (Adaptive Body Bias) と呼ばれる手法がある [9]。

ABB では、適切な順電圧を MOS トランジスタに印加するために、元の回路に対して、順電圧を制御する回路を付加する。付加される回路は、3 つの回路から構成される。その 3 つの回路は、センシング回路、スクエアリング回路、増幅回路である。センシング回路は、元の回路のなかの MOS トランジスタに対して、MOS トランジスタの実際の閾値電圧を計測する回路である。スクエアリング回路と増幅回路は、センシング回路から計測された MOS トランジスタの閾値電圧の値を利用して、適切な順電圧を生成し、その順電圧を MOS トランジスタに印加するための回路である。この 3 つの回路により、MOS トランジスタに適切な順電圧を印加することができる。

ABB では、それぞれの MOS トランジスタに対して、適切な電圧を印加するために、1 つの MOS トランジスタに対して、1 つの順電圧を制御する回路が必要である。つまり、1 つの MOS トランジスタに対して、数十以上の MOS トランジスタが順電圧の制御のために必要になる。この場合、付加回路の面積オーバーヘッドがかなり大きくなるので、ABB では、MOS トランジスタをいくつかのブロックに分け、そのブロックのなかでそれぞれ一律に順電圧を印加する。ただし、1 ブロックあたりの MOS トランジスタの数が多くなると、ABB では、それぞれの MOS トランジスタに必要な順電圧を印加することができない場合がある。つまり、付加する回路の面積オーバーヘッドと回避できる遅延故障の総数はトレードオフの関係になるといえる。また、回路の微細化が進むと、Body Bias による手法では、HCI による遅延故障を回避することが難しくなる [4]。

ABB を利用する手法とは別に、HCI による遅延故障を回避するために、パルス状の逆電圧を印加する手法がある [10]。この手法では、回路の微細化が進んでも、HCI による遅延故障を回避することができる。そのために、ABB で付加した回路とは別の回路が必要となる。つまり、HCI と NBTI による遅延故障を同時に回避するために、2 つの付加回路が必要になる。回路の微細化による遅延故障を回避するために、HCI と NBTI による遅延故障を同時に回避し、なおかつ、付加する回路の面積オーバーヘッドを抑え、多くの遅延故障を回避できる設計が必要である。

## 3. PPC を利用した遅延故障の回避手法

本節では、PPC (Partially-Programmable Circuit) を利用した遅延故障を回避する提案手法について説明する。3.1 節では、PPC について説明する。3.2 節では、本稿で適用する遅延モデルについて説明する。3.3 節では、PPC を利用した遅延故障を回避する提案手法を説明する。

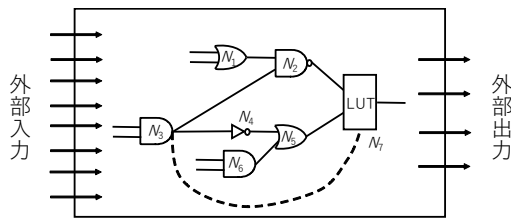


図 2 PPC

### 3.1 Partially-Programmable Circuit

PPC とは、冗長性をもった新しい回路モデルである [7] . PPC は、既存のノードの一部を LUT に置き換え、冗長な結線を新たに追加したものである . PPC で想定する故障は、ノードの出力または結線における単一の縮退故障である . PPC の一例を図 2 に示す .  $N_i$  は、回路内のあるノードをあらわす .  $C_{[i,j]}$  は、 $N_i$  の出力から  $N_j$  の入力への結線をあらわす . 図 2 中の破線  $C_{[3,7]}$  は、冗長な結線をあらわす . 回路内に故障がない場合、冗長な結線  $C_{[3,7]}$  を利用しなくても、回路は正常に動作する . 図 2 の結線  $C_{[4,5]}$  が故障した場合、PPC では、冗長な結線  $C_{[3,7]}$  を利用して、LUT の内部論理を適切に書き換えることによって、回路を正常に動作させることができる . 一方、図 2 の結線  $C_{[1,2]}$  が故障した場合、結線  $C_{[1,2]}$  に対する冗長な結線が存在しないので、回路を正常に動作させることができない . そのため、PPC では、すべての故障を回避することができない . しかし、PPC では、LUT とマルチプレクサのみで一部の故障を回避することができるので、面積オーバーヘッドを抑えつつ、多くの故障を回避することができる場合がある .

PPC の設計手法では、回避可能な故障の数が最も多くなるようにノードを LUT に置き換えることが重要である . 本稿では、外部出力に一番近いノードを LUT で置き換えるヒューリスティックによって、LUT の配置を決定する . このヒューリスティックを用いて LUT を配置した回路は、回路内すべての故障に対して、故障を回避できる冗長な結線の条件を必ず求めることができる .

### 3.2 遅延モデル

PPC に関する既存研究において、想定する故障は、ノードの出力または結線における単一の縮退故障である . そのため、本稿では、PPC における遅延モデルを定義する必要がある .

本稿では、遅延故障は 1 ヶ所のみ起こり、その故障はノードの MOS トランジスタ、または LUT の MOS トランジスタに発生すると仮定する . 遅延故障を起こしたノード、または LUT は遅延時間が  $k$  倍になると仮定する . 既存ゲートの遅延時間について、様々な研究で既存ゲートの遅延時間が測定されている . 本稿では、それらの研究から既存ゲートごとの遅延時間の比率を利用して、遅延時間を計算する .

本稿では、既存ゲートと同様に、LUT の遅延について定義する必要がある . そのため、本稿では、パストランジスタの段数ごとに LUT の遅延時間を決めるモデルを考える . 図 3 は、3 入力 LUT の構造を示す . LUT では、LUT の

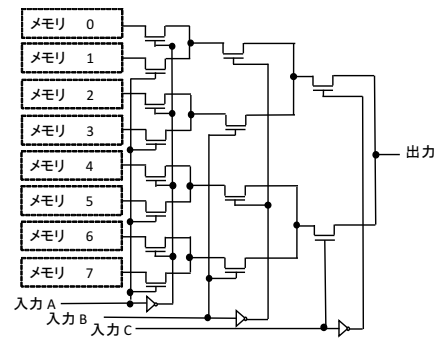


図 3 3 入力 LUT の構造

入力をパストランジスタに繋げることで、その入力によって出力するメモリを選択している . 図 3 の入力 C に関して、出力から入力 C に繋がっているパストランジスタへのパス上には、パストランジスタが 1 つ存在する . そのため、入力 C から出力までの遅延時間をパストランジスタ 1 段と仮定する . 同様に、図 3 の入力 B に関して、出力から入力 B に繋がっているパストランジスタへのパス上には、パストランジスタが 2 つ存在する . そのため、入力 B から出力までの遅延時間をパストランジスタ 2 段と仮定する . つまり、本稿では、各入力から出力までのパストランジスタの段数ごとに遅延時間を決めることで、LUT の遅延時間を計算する . このように、本稿では、既存ゲートの遅延時間と LUT の遅延時間を計算することで、回路の全体の遅延時間を計算する .

### 3.3 既存ゲートを用いた提案手法

本稿では、回避可能な遅延故障が最も多くなるような LUT の配置と冗長な配線を定める手法を提案する . 遅延故障は 1 ヶ所のみ起こり、その遅延故障は既存ゲートの MOS トランジスタ、または LUT のパストランジスタと NOT ゲートの MOS トランジスタに起こると仮定する .

初めに、回路内のすべてのパスに対して、遅延故障が起きていないときの回路の遅延時間を計算する . その後、回路内のノードのなかで、1 つのノードが遅延故障を起こしたと仮定し、そのときの回路の遅延時間を計算する . このとき、遅延故障を起こしたノードの遅延時間が遅延故障が起きていない時のノードの遅延時間と比べて、 $k$  倍になると仮定する . 遅延故障を起こしたノードにより、回路内のパスの一部の遅延時間が元の回路のパスの遅延時間より長くなる場合がある . 元の回路の最長パスと比べて、遅延時間が長くなったパスを、そのノードが遅延故障を起こしたときの故障パターンと呼ぶ . すべてのノードにおける故障パターンのパスの集合  $\{P_1, P_2, \dots, P_m\}$  を計算する . 本稿では、故障パターンのパスの集合をもつ回路内のすべてのノードを遅延故障の対象とする . そのノードにおけるすべての故障パターンのパスを回避できるとき、そのノードの遅延故障は回避できるとする .

それぞれのノードにおけるパスの故障パターンを探索したあと、回路内のノードのなかで、外部出力に一番近いノードを LUT に置き換える . このとき、すべてのノードのバ

**Algorithm 1** 冗長な配線の条件の計算手順

```

Ensure: LUT の配置が決まっている
for  $N_i \leftarrow$  故障パターンのパスをもつすべてのノード do
  for  $P_m \leftarrow N_i$  の故障パターンのパスの集合 do
     $N_i$  の活性化論理関数を計算
     $P_m$  上に存在する LUT の入力に関する SPFD を計算
    SPFD と活性化論理関数から Remaining SPFD を計算
  end for
end for
  
```

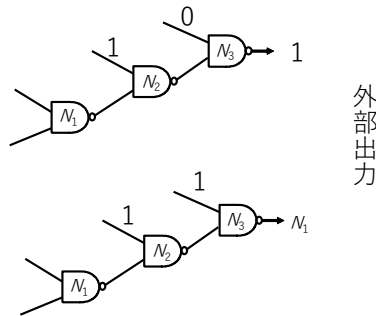


図 4 活性化論理関数の説明のための回路

スの故障パターンにおいて、そのパス上に存在するノードのみを LUT に置き換える。

LUT の配置が決まったあと、遅延故障の対象になっているノードの各故障パターンのパスを回避できる冗長な配線の条件を計算する。冗長な配線の条件を計算する手順を、Algorithm 1 に示す。まず、故障パターンのパスの集合をもつノード  $N_i$  について、 $N_i$  から  $P_m$  上に存在する LUT の入力までのパスが活性化する外部入力パターンを計算する。 $N_i$  から  $P_m$  上に存在する LUT の入力までのパスが活性化する外部入力パターンが 1 となる論理関数を  $N_i$  の活性化論理関数と呼ぶ。つまり、 $N_i$  の出力が、回路の出力に影響を及ぼすような外部入力パターンの時に 1 になる関数である。活性化論理関数の例を図 4 に示す。図 4 の上と下の回路は同じ 2 入力 NAND で構成されている。図 4 の上の回路では、 $N_3$  の入力のうち、一方の入力の値が 0 なので、外部出力の値は、1 になる。つまり、 $N_1$  の出力の値は、外部出力の値に影響しないといえる。そのため、 $N_1$  が遅延故障を起こしても、外部出力の遅延時間が長くない。しかし、図 4 の下の回路では、 $N_2, N_3$  の入力のうち、一方の入力の値が 1 なので、外部出力の値は、 $N_1$  の出力の値に依存する。つまり、 $N_1$  の出力の値は、外部出力の値に影響するといえる。そのため、 $N_1$  が遅延故障を起こした場合、外部出力の遅延時間が長くなる。各ノード  $N_i$  から  $P_m$  上に存在する LUT の入力までのパスが活性化する外部入力パターンを計算することで、ノードの遅延故障の影響を受ける最小限の外部出力のパターンのみ抽出することができる。

次に、 $P_m$  上に存在する LUT の入力について、その LUT の入力の SPFD (Set of Pairs of Functions to be Distinguished) を計算する。SPFD[11] とは、LUT の論理関数の自由度を表現するための手法である。その後、LUT の入力の SPFD と  $N_i$  の活性化論理関数から、 $N_i$  における故障

**Algorithm 2** 冗長な配線の条件を満たす配線の探索手順

```

Ensure:  $SPFD_{[i,m]}$ :  $N_i$  における故障パターンのパス  $P_m$  を回避するための冗長な配線の条件
for  $N_i \leftarrow$  遅延故障の対象であるすべてのノード do
  for  $P_m \leftarrow N_i$  の故障パターンのパスの集合 do
     $N_i$  が故障した場合の LUT の遅延時間を計算
    if LUT の遅延時間が元の回路の遅延時間と同じ、または短い then
      for  $N_j \leftarrow$  回路内のすべてのノード do
         $N_j$  が  $SPFD_{[i,m]}$  を満たすかどうか計算
        if  $N_j$  が  $SPFD_{[i,m]}$  を満たす then
           $N_j$  が LUT の入力に繋がった場合の LUT 遅延時間を計算
          if LUT の遅延時間が元の回路と同じ、または短い then
            LUT の入力の構成を  $LUT_{[i,m]}$  として保存
          end if
        end if
      end for
    end if
  end for
end for
end for
  
```

パターンのパス  $P_m$  を回避することができる冗長な配線の条件である Remaining SPFD[12] を求めることができる。LUT の入力の SPFD と活性化論理関数を利用して、すべての  $N_i$  における故障パターンのパス  $P_m$  を回避するための必要最小限の冗長な配線の条件を計算することができる。

遅延故障の対象となるノードにおける各故障パターンのパスに対する冗長な配線の条件を計算したあと、その条件を満たす配線を探す必要がある。冗長な配線の条件を満たす配線を探る手順を、Algorithm 2 に示す。まず、遅延故障の対象のノード  $N_i$  が故障した場合の LUT の遅延時間を計算する。もし、冗長な配線が繋がっていない状態で LUT の遅延時間が元の回路より長くなる場合、 $N_i$  の遅延故障を回避することができない。

LUT を配置しても遅延時間が長くない場合、次に、回路内のすべての配線から冗長な配線の条件を満たす配線を探る。もし、冗長な配線の条件を満たす配線が存在したとき、その  $N_j$  の配線を LUT の入力に繋げる。そして、冗長な配線を繋げた LUT の遅延時間を計算する。もし、冗長な配線を繋げた LUT の遅延時間が元の回路より長くない場合、 $N_i$  における故障パターンのパス  $P_m$  は、冗長な配線によって回避できる。このようにして、すべての  $N_i$  における故障パターンのパス  $P_m$  に対する冗長な配線の条件を満たす配線を探ることができる。このとき、冗長な配線を繋げた LUT の入力の構成を  $LUT_{[i,m]}$  として保存する。遅延故障の対象になっているすべてのノードに対して、冗長な配線を探ることで、各ノード  $N_i$  における各故障パターンのパス  $P_m$  を回避するときの LUT の構成  $LUT_{[i,m]}$  を計算することができる。

次に、すべての  $LUT_{[i,m]}$  うち、入力の構成が矛盾しない  $LUT_{[i,m]}$  の組み合わせを探る。このとき、冗長な配線は違うが、2 つの冗長な配線をマルチプレクサに繋いだときの LUT の遅延時間が長くない場合、その 2 つ

の  $LUT_{[i,m]}$  を組み合わせることができる．その  $LUT_{[i,m]}$  の組み合わせによって，回避できる遅延故障のノードの数が最も多くなる LUT の構成を決める．

上記の Algorithm 1, Algorithm 2 を利用して，すべての LUT の配置パターンで，回避できる遅延故障の数が最も多くなる LUT の構成を求める．すべての LUT の配置パターンのなかで，回避できる故障の数が最も多くなるような LUT の配置と冗長な配線を決める．

#### 4. 実験結果と考察

本節では，4.1 節で，本稿の実験で適用した既存ゲートの遅延時間と LUT の遅延時間について述べる．4.2 節では，4.1 節の遅延時間を適用して，提案手法をベンチマーク回路に適用した実験結果を述べ，その考察を述べる．

##### 4.1 既存ゲートと LUT の遅延時間

本稿の実験では，既存ゲートの遅延時間は Kumar らの遅延時間の測定の結果 [6] を参考にした．この研究では，電源電圧が 1.8V，CMOS のチャネル長が  $180\mu\text{m}$ ，温度が 25

において，2 入力の NAND ゲートの遅延時間は約 95ps，NOT ゲートの遅延時間は約 30ps と測定されている．本稿では，この結果を利用して，NAND ゲートと NOT ゲートの遅延時間の比率を 3:1 とすることで，既存のゲートの遅延時間を計算した．

本稿の実験では，Ahmed らの LUT の遅延時間を測定結果 [1] を参考に，1 段のバストランジスタの遅延時間は，NAND ゲートの遅延時間より遅くないと仮定することにする．2 段のバストランジスタの遅延時間は，2 入力の NAND ゲートと変わらないと考える．ただし，2 入力の NAND ゲートと 2 入力の LUT をそのまま置き換えることができず，2 段のバストランジスタの遅延時間は 2 入力の NAND ゲートより長いと仮定することにする．3 段のバストランジスタの遅延時間は，NAND ゲート 2 段の遅延時間と同じに，4 段のバストランジスタの遅延時間は，NAND ゲート 3 段の遅延時間と同じと仮定することにする，つまり，バストランジスタ 1 段の遅延時間を 3，バストランジスタ 2 段の遅延時間を 4，バストランジスタ 3 段の遅延時間を 6，バストランジスタ 4 段の遅延時間を 9 と仮定することにする．以上の推測で遅延についてはこのような想定値で実験したが，実際には配線遅延もあるため，実際の遅延とは必ずしも一致しない．しかし，PPC がどの程度想定する遅延故障に使えるかの目安を知る上でのある程度の目安にはなると考えている．

##### 4.2 実験結果と考察

4.1 節の遅延時間を仮定して，提案手法をベンチマーク回路に適用した実験した．本稿では，ベンチマーク回路は 2 入力の NAND ゲートと NOT ゲートで構成された回路である．ノードの遅延時間が 2 倍，3 倍，4 倍になった場合の実験結果を表 1，表 2，表 3 に示す．表の「-」は，実験時間が 2 時間以上かかり，回避できる故障数が取れなかったものを表す．3 入力の LUT を使用した提案手法では，最

表 1 ノードの遅延時間が 2 倍のときの実験結果

回路名	PI/PO	面積	故障数	3 入力の LUT の回避可能な故障数	4 入力の LUT の回避可能な故障数
comp	32/3	632	200	52	—
count	35/16	630	362	0	16
example2	85/66	1382	1140	102	58
i1	25/16	252	224	0	24
i2	201/1	1676	828	0	148
term1	34/10	1100	460	50	8
x1	51/35	1506	1012	0	40

表 2 ノードの遅延時間が 3 倍のときの実験結果

回路名	PI/PO	面積	故障数	3 入力の LUT の回避可能な故障数	4 入力の LUT の回避可能な故障数
9symml	9/1	1078	492	0	46
comp	32/3	632	332	148	—
count	35/16	630	386	0	44
example2	85/66	1382	1222	18	32
i1	25/16	252	228	0	24
i2	201/1	1676	1606	0	272
term1	34/10	1100	672	118	112
x1	51/35	1506	1190	30	48

表 3 ノードの遅延時間が 4 倍のときの実験結果

回路名	PI/PO	面積	故障数	3 入力の LUT の回避可能な故障数	4 入力の LUT の回避可能な故障数
9symml	9/1	1078	738	0	66
comp	32/3	632	376	180	—
count	35/16	630	400	0	44
example2	85/66	1382	1290	18	32
i1	25/16	252	232	0	24
i2	201/1	1676	1606	0	272
term1	34/10	1100	858	18	120
x1	51/35	1506	1304	26	24

大で全体の遅延故障の 48%，4 入力の LUT を使用した提案手法では，最大で全体の遅延故障の 24%を回避することができた．

本稿では，回路の面積を回路内のトランジスタの数とする．故障数とは， $k$  倍の遅延故障が起きた場合に故障の対象となるトランジスタの数である．3 入力の LUT を利用する場合，その LUT の構成は，3 入力のうち 2 入力を元の回路に繋いで，残りの 1 入力を冗長な配線と繋ぐ構成となる．そのときの回避できる遅延故障の数が，表の「3 入力の LUT の回避可能な故障数」である．4 入力の LUT を利用する場合，その LUT の構成は，4 入力のうち 3 入力を元の回路に繋いで，残りの 1 入力を冗長な配線と繋ぐ構成となる．そのときの回避できる遅延故障の数が，表の「4 入力の LUT の回避可能な故障数」である．

表 1，表 2，表 3 では，すべてのベンチマーク回路のうち，2 時間以内に実験を終え，なおかつ，回避できる遅延故障があった回路である．全体の回路の数に比べて，回避できる遅延故障がある回路はあまり多くなかった．つまり，冗長な配線の条件を満たし，なおかつ，回路の遅延時間が長くなならないような冗長な配線は少ないと考えられる．また，回避できる遅延故障の数が，全体の遅延故障の数と比べて，少なかった．ただし，i2 や comp のように，多くの遅延故障を回避することができる例もあるため，回避できる遅延故障の数は，回路の規模や回路の構成に依存すると考えられる．

LUT 挿入による面積増加を考えると，実験結果の値で

は、面積増加が大きく、回避できる故障が少ないように考えられるかもしれない。しかし、提案手法では、LUT が実現しなければならない論理関数が少ないので、早苗らが提案した手法 [13] によって、LUT の面積を削減することができる。つまり、1 つの LUT に対する冗長な配線が少なければ、実現しなければならない論理関数は少ないので、LUT の面積を大幅に削減することができる。また、LUT を配置するとき、いくつかの既存ゲートを取り除くことができるので、LUT 挿入による面積の増加を抑えることができる。そのため、LUT を挿入することで発生する面積の増加は大きくないと考えられる。

ノードの遅延時間が 3 倍になる遅延故障と 4 倍になる遅延故障を比べると、回避できる遅延故障の数が変わらないものが多い。この結果から、ノードの遅延時間が 3 倍になる遅延故障から 4 倍になる遅延故障に変わることによって、増えた遅延故障は回避しづらいと考えられる。ただし、9symml や comp のようにノードの遅延時間が 4 倍になることで、増えた遅延故障を回避できる例もある。

## 5. おわりに

本研究では、遅延故障を回避する回路設計手法において、PPC を用いて遅延故障を回避する回路設計手法を提案した。提案手法では、すべての遅延故障に対する冗長な配線の条件を計算した。その後、冗長な配線を利用して、遅延故障を最も多く回避できる LUT の構成を決めた。実験結果では、遅延故障を回避できない回路もあったが、一部の回路では、多くの遅延故障を回避することができた。

今後の課題として、提案手法で決まった LUT の構成により、提案手法による面積増加を計算する必要がある。提案手法では、冗長な配線の数が多くなかったため、LUT のメモリやパストラジスタの面積を大きく減らすことができると考えている。つまり、全体の遅延故障の割合に対して、回避できる遅延故障が少ないが、LUT による面積増加がとても少ない場合があると考えられる。そのため、遅延時間を考慮した PPC の面積の計算が今後の課題である。

また、既存ゲートやパストラジスタ、配線遅延を正確に考慮した遅延故障モデルを考える必要がある。本稿では、既存ゲートの遅延時間とパストラジスタの遅延時間をそれぞれ別の研究を参考にし、推測した。そのため、実際の既存ゲートとパストラジスタの遅延時間が推測した値と違う可能性が高い。また、配線遅延を考えていないため、実際の回路の遅延時間と違う可能性がある。そのため、既存ゲートやパストラジスタ、配線遅延を正確に測定してから、提案手法を適用し LUT の構成を決めることが今後の課題である。

また、この提案手法に、LUT のみで遅延故障を回避する手法を合わせた手法を考えられる。つまり、LUT を挿入するだけで回避できる遅延故障があると考えられる。そのため、LUT のみで遅延故障を回避し、回避できなかった遅延故障を PPC を用いて回避するような手法を検討することが今後の課題である。

謝辞 本研究は JSPS 科研費 15H02679 の助成を受けた

ものです。

## 参考文献

- [1] Ahmed, E. and Rose, J.: The effect of LUT and cluster size on deep-submicron FPGA performance and density, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, Vol. 12, No. 3, pp. 288–298.
- [2] Borkar, S.: Designing reliable systems from unreliable components: the challenges of transistor variability and degradation, *Micro, IEEE*, Vol. 25, No. 6, pp. 10–16.
- [3] Cho, M., Aoulaiche, M., Degraeve, R., Kaczer, B., Franco, J., Kauerauf, T., Roussel, P., Ragnarsson, L., Tseng, J., Hoffmann, T. and Groeseneken, G.: Positive and negative bias temperature instability on subnanometer eot high-K MOSFETs, *Reliability Physics Symposium (IRPS), 2010 IEEE International*, pp. 1095–1098 (2010).
- [4] Hokazono, A., Balasubramanian, S., Ishimaru, K., Ishiuchi, H., Hu, C. and Liu, T.-J. K.: MOSFET hot-carrier reliability improvement by forward-body bias, *Electron Device Letters, IEEE*, Vol. 27, No. 7, pp. 605–608 (2006).
- [5] Kuhn, K.: Reducing Variation in Advanced Logic Technologies: Approaches to Process and Design for Manufacturability of Nanoscale CMOS, *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, pp. 471–474 (2007).
- [6] Kumar, R. and Kursun, V.: Impact of temperature fluctuations on circuit characteristics in 180nm and 65nm CMOS technologies, *Circuits and Systems, 2006. IS-CAS 2006. Proceedings. 2006 IEEE International Symposium on*, pp. 3858–3861 (2006).
- [7] S. Yamashita, H. Y. and Fujita, M.: Increasing Yield Using Partially-Programmable Circuits, *Proc. SASIMI*, pp. 237–242 (2010).
- [8] Takeda, E., Shimizu, A. and Hagiwara, T.: Role of hot-hole injection in hot-carrier effects and the small degraded channel region in MOSFET's, *Electron Device Letters, IEEE*, Vol. 4, No. 9, pp. 329–331 (1983).
- [9] Tschanz, J., Kao, J., Narendra, S., Nair, R., Antoniadis, D., Chandrakasan, A. and De, V.: Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage, *Solid-State Circuits, IEEE Journal of*, Vol. 37, No. 11, pp. 1396–1402 (2002).
- [10] Wu, R., Kawai, S., Seo, Y., Kimura, K., Sato, S., Kondo, S., Ueno, T., Fajri, N., Maki, S., Nagashima, N., Takeuchi, Y., Yamaguchi, T., Musa, A., Miyahara, M., Okada, K. and Matsuzawa, A.: 19.5 An HCI-healing 60GHz CMOS transceiver, *Solid-State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3 (2015).
- [11] Yamashita, S., Sawada, H. and Nagoya, A.: A New Method to Express Functional Permissibilities for LUT Based FPGAs and Its Applications, *Proceedings of the 1996 IEEE/ACM International Conference on Computer-aided Design, ICCAD '96*, Washington, DC, USA, IEEE Computer Society, pp. 254–261 (1996).
- [12] 山下茂, 松尾淳士: 二重化よりも面積オーバーヘッドが少ない耐故障化手法, *信学技報*, Vol. 113, No. 320, pp. 33–37 (2013).
- [13] 早苗駿一: PPC に基づく歩留まり改善に向けた論理最適化, 奈良先端科学技術大学院大学 2013 年度修士論文 (2014).