

## 改良カナリア FF を利用した高信頼性 VLSI 設計手法の提案

矢野 憲<sup>†,††</sup>, 吉木 崇人<sup>†</sup>  
林田 隆則<sup>†</sup>, 佐藤 寿倫<sup>†,††</sup>

半導体技術の微細化と、携帯電話などのポータブルデバイスの普及により省電力かつ高性能の LSI の需要が高まってきた。ディープサブミクロン領域での素子特性のばらつきや、経年劣化、ソフトウェアによるタイミングエラー対策としてカナリア FF を用いた高信頼性 VLSI 設計手法の提案を行った。本稿で述べる改良カナリア FF はこれまでのカナリア FF に比べて約 8% の消費電力を削減することを確認している。またカナリア FF による LSI のセル面積に与える影響についても調査を行った。

### A design approach of dependable VLSI using improved canary FF

Ken Yano<sup>†,††</sup> and Takahito Yoshiki<sup>†</sup>  
Takanori Hayashida<sup>†</sup> and Toshinori Sato<sup>†,††</sup>

#### 1. はじめに

半導体技術の微細化と携帯電話などのポータブルデバイスの普及により、LSI 動作のさらなる高速化と省電力化が求められるようになってきた。これまでの最悪ケースを考慮にいた設計ではディープサブミクロン領域での素子の特性ばらつきによる設計マージンが大きくなり性能や消費電力に与える悪影響が深刻であった。また経年劣

化、ソフトウェアなどの要因により、設計段階では想定しないタイミングエラーが発生し、集積回路の誤動作の主要な要因の一つとなっている。

本稿では典型的ケースに注力したカナリア FF を用いた高信頼性 LSI の設計手法を提案する。本研究ではカナリア FF のトランジスタレベルでの回路動作に注目し省電力、小面積の改良カナリア FF を提案した。改良カナリア FF はカナリア FF に比べて、消費電力を約 8% できることを確認している。次に、最悪ケースから得られたタイミングエラー情報からカナリア FF を挿入する位置を限定する手法について述べる。また、カナリア FF のセル面積に与える影響について調査した。本稿では第 2 章でタイミングエラー判定 FF の関連研究を紹介した後に第 3 章で改良されたカナリア FF の回路構成について述べ、第 4 章ではカナリア FF を用いた LSI 設計について、特にセルマッピングの自動化及びセル面積に与える影響について述べる。第 5 章でカナリア FF を使用した高信頼性 LSI 設計についての課題を述べる。

#### 2. タイミングエラー判定フリップフロップ

これまでにタイミングエラーを検出する研究が盛んに行われている。回路の信頼性を高める方法として、空間的な冗長性を持たせる場合と時間的な冗長性を持たせる場合とがある。本稿では、空間的な冗長性を持たせる手法として改良カナリア FF を用いたタイミングエラー検出と、それを用いた高信頼性 VLSI の設計手法について述べる。我々はこれまでにカナリア FF を用いて、LSI に用いられる素子のばらつきによるデザインマージンを削減する手法について報告を行ってきた<sup>2)</sup>。半導体プロセスの複雑度が増し、高性能で低消費電力の LSI の需要が高まる状況において、近年 LSI 素子特性のばらつきが顕在化し、そのばらつきを考慮した設計が不可欠となっている。設計段階では正確な計測が困難な経年劣化による素子特性の変化や、ソフトウェア等の問題も深刻になっている。これまでにタイミングエラーを検出する FF として、Razor Flip-Flops(RazorFF)<sup>7)</sup>、RazorII<sup>6)</sup>、Phase-adjustable Error Detection Flip-Flop(PEDFF)、Delay-Compensation Flip-Flops(DCCFF)<sup>9)</sup>などが報告されている。また、ソフトウェア対策として Built-In Soft Error Resilience(BISER)<sup>10)</sup>、Bistable Cross-coupled Dual Modular Redundancy(BCDMR)<sup>8)</sup>などの冗長 FF が提案されている。これらの FF を用いた場合の問題として冗長化した FF の追加によるセル面積の増加と消費電力がある。タイミングエラー判定 FF はメインの FF に加えてシャドウ FF (ラッチ)、遅延素子そしてエラー判定回路から構成させるために、通常の FF と比較して 2 倍から 3 倍の面積が必要となることが報告されている。逆に、タイミングエラー判定 FF を DVS と協調して動作させることによりチップ全体の消費電力を削減する提案<sup>1)</sup>も行われている。

<sup>†</sup> 福岡大学

<sup>††</sup> 独立行政法人科学技術振興機、CREST

### 3. 改良カナリアフリップフロップ

タイミングエラーを検出するFFとしてカナリアFFが提案されている<sup>1)</sup>。本稿では、カナリアFFを使用して実際にLSIチップの作成を行うことを検討するために、カナリアFFのトランジスタ回路および、セル化の検討を行った。評価環境としてライブラリはROHM社0.18 $\mu\text{m}$ 京大版ライブラリ<sup>1)</sup>を用いた。表1にライブラリで定義しているプロセス条件を示した。図1で示すカナリアFFの回路では、同等のFFを2つ必要とするため、必要なセル面積や回路で消費される電力も2倍以上になることが予想される。ここで回路を最適化することを考える。タイミングエラーが発生する場合は、マスターFFとシャドウFFのマスターラッチにデータが到着している時点で判別できるためシャドウFFのスレーブラッチは不要となる。一方で、エラー検出回路には半周期後のタイミングでエラー判定を行うためにNANDゲートが追加となる。図2に提案する改良カナリアFFの回路を示した。2つの回路で消費される平均消費電力と、最大消費電力をHSPICEのシミュレーションにより計測した結果を表2に示した。カナリアFFではDFFに比べて平均消費電力が2.7倍、最大省電力が1.06倍であった。またカナリアFFに比べて改良カナリアFFは平均消費電力で約7.8%、最大消費電力で約7.6%削減できることが確認できた。

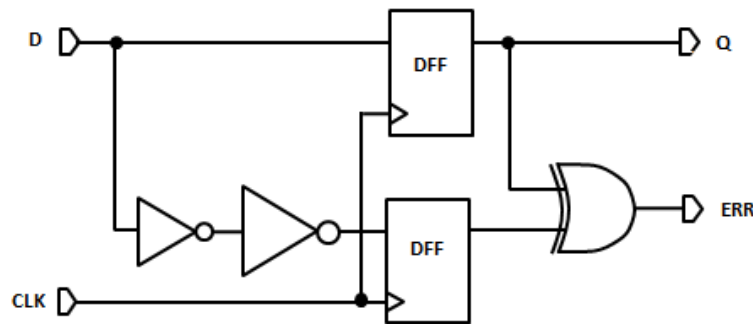


図1 カナリアFF

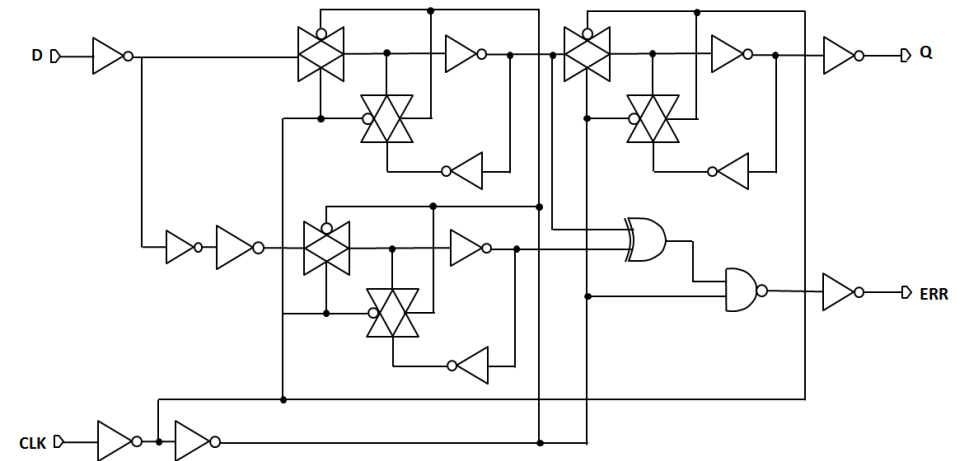


図2 改良カナリアFF回路図

表1 ROHM社0.18 $\mu\text{m}$ 京大版ライブラリ プロセス条件

Vdd	Temperature	Process
1.6V	85	Max
1.8V	25	Typ
2.0V	-40	Min

表2 HSPICEでの各FF方式での最大消費電力、平均消費電力の見積もり  
 (クロック周期は20nsに設定)

	DFF	カナリアFF	改良カナリアFF
平均消費電力[W]	2.53e-5	6.82e-5	6.28e-5
最大消費電力[W]	4.99e-3	5.26e-3	4.86e-3

#### 3.1 改良カナリアFF置き換え位置の判定

高信頼性のみを追求するのであれば、回路のすべてのFFをカナリアFFに置き換えることも可能であるが、マイクロプロセッサ等では、数千から数万オーダーの数のFFが使用されており、カナリアFFによる面積増加と消費電力を考慮した場合、最適な

手法とはならない。また全カナリア FF から報告されるエラー報知の収集の問題も深刻となる。そこで、タイミングマージンの小さい FF のみカナリア FF に置き換える方法を採用した<sup>3)</sup>。RTL による機能レベルの HW 記述を論理合成ツールにより基本論理ゲートからなるネットリストへ変換すると、使用している標準セルライブラリに記述されている各論理ゲートの遅延情報から各回路パスの遅延時間が求まる。本研究では、RTL として、東芝の MeP プロセッサコア<sup>11)</sup>と miniMips プロセッサ<sup>12)</sup> (Mips プロセッサの縮小命令セット版) を用い、Synopsys Design Compiler (D-2010.03-SP5) で論理合成を行った。

クリティカルパスの結果からクロック周期をプロセス条件が Typ の時にタイミングエラーが発生しない最小値に設定して論理合成を行った。合成の結果 Typ と Min のプロセス条件ではタイミングエラーとなるパスは存在しなかったが、プロセス条件が Max の場合には両方ともタイミングエラーとなる回路パスが存在した。表 3 に各プロセッサで使用されている FF の合計数とタイミングエラーとなった回路パスの終端 FF の合計数を示す。MeP プロセッサは全体の 1.6%、miniMips プロセッサでは全体の 11.6% の割合の FF でタイミングエラーが発生することが分かった。

表 3 タイミングエラーとなった回路パスの終端 FF の数

Processor	Total # of FF	# of FF (Timing Error)	Percentage
MeP	3732	60	1.6%
miniMips	1967	228	11.6%

タイミングエラーとなるパスは設定するクロック周期に依存するが、ここではプロセス条件が Typ の時にタイミングエラーが発生しない最小のクロック周期が選択されることを想定している。つまり通常の動作環境ではタイミングエラーは発生しないが、電圧降下等の動作環境の変化では、タイミングエラーが発生する可能性が高い。そこで、改良カナリア FF へ置き換える FF はプロセス条件が Max の場合にタイミングエラーを発生する回路パスの終端 FF に限定することにした。こうすることにより、カナリア FF へ置き換える FF を大幅に減少させることが可能になる。以下にその詳細を述べる。

### 3.2 改良カナリア FF 置き換え自動化

初めに、クロック周期を求めるためにプロセス条件を Typ として論理合成を行い、クリティカルパスから求められる最小のクロック周期を求めこれを Clk\_typ とする。またこの時、論理合成で作成されたネットリストを NetList\_org として保存しておく。次にプロセス条件を Max に変更し、クロック周期を Clk\_typ として論理合成を行う。

プロセス条件 Max では電源電圧が Typ 条件の場合よりも、低く定義されているため、各ゲートの遅延が増加し、いくつかのパスがタイミングエラーとなる。論理合成のログ情報よりタイミングエラーが発生した回路パスを抽出し、各回路パスの終端 FF を置き換え対象 FF リスト FF\_rep に登録する。元となるネットリスト Netlist\_org と、置き換え対象 FF のリスト FF\_rep が得られたら、それらを FF 置き換えスクリプトへ与える。スクリプトは Netlist\_org と FF\_rep を入力ファイルとして読み込み、ネットリスト Netlist\_org で定義している FF のインスタンスが FF\_rep に登録されていれば、そのセル名を改良カナリア FF のセル名に書き換える。全ての FF の置き換えが終了した時点で、変換後ネットリスト NetList\_chg を出力する。図 3 に通常 FF を改良カナリア FF に置き換える手順を示した。図 4 は MeP プロセッサの変換前ネットリスト、変換後ネットリストの抜粋である。FF のインスタンスである rbidiPCount\_reg\_1\_と rbidiPCount\_reg\_5\_が FF\_rep に登録されているため、変換後のネットリストではそのセル名が改良カナリア FF のセル名に変更されている。なお、変換前のネットリストではドライブ能力の違いにより専用の FF セルが使用されているが、ここでは単純にすべての FF セルを同一の改良カナリア FF で置き換えることにしている。

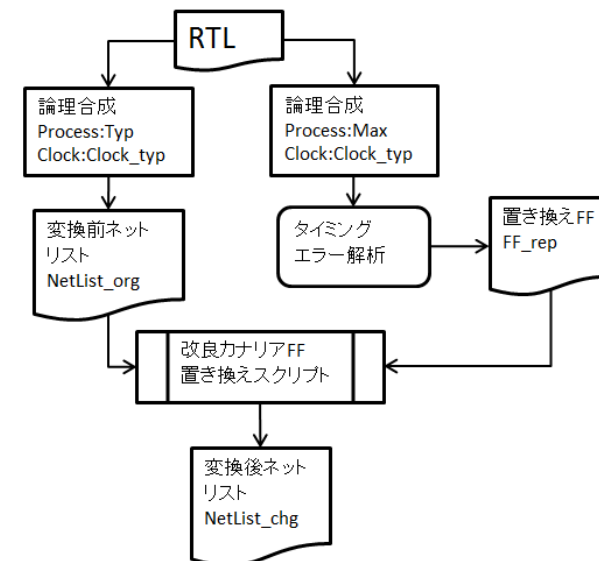


図 3 改良カナリア FF 置き換え手順

変換前ネットリスト (抜粋)

```
ROHM18DFP120 rbidiPSS_reg_0 ( .D(n22), .C(cpuclk), .Q(rbidiPSS[0]) );
ROHM18INV040 U32 ( .A(n140), .Y(n29) );
ROHM18DFP010 rbidiPCount_reg_1 ( .D(n12), .C(cpuclk), .Q(rbidiPCount[1]) );
ROHM18DFP010 rbidiPCount_reg_5 ( .D(n4), .C(cpuclk), .Q(rbidiPCount[5]) );
ROHM18NOR2P040S U9 ( .A(n73), .B(n7), .Y(n8) );
ROHM18NOR2P080S U31 ( .A(n80), .B(n29), .Y(n30) );
```

変換後ネットリスト (抜粋)

```
ROHM18DFP120 rbidiPSS_reg_0 ( .D(n22), .C(cpuclk), .Q(rbidiPSS[0]) );
ROHM18INV040 U32 ( .A(n140), .Y(n29) );
canary FF rbidiPCount_reg_1 ( .D(n12), .C(cpuclk), .Q(rbidiPCount[1]) );
canary FF rbidiPCount_reg_5 ( .D(n4), .C(cpuclk), .Q(rbidiPCount[5]) );
ROHM18NOR2P040S U9 ( .A(n73), .B(n7), .Y(n8) );
ROHM18NOR2P080S U31 ( .A(n80), .B(n29), .Y(n30) );
```

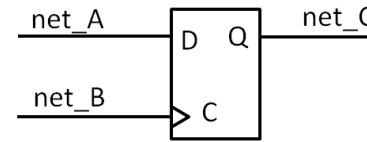
図4 変換前、変換後ネットリスト例

#### 4. カナリア FF を取り入れた配置配線

前章では改良カナリア FF がすでに標準セル化されていることを前提にネットリストを書き換える方法について説明を行った。改良カナリア FF のセル化については、現在開発途中であるため配置配線では使用できない。そのため、ここでは既存の標準セルを使用してカナリア FF を実装し、配置配線を行うことにした。

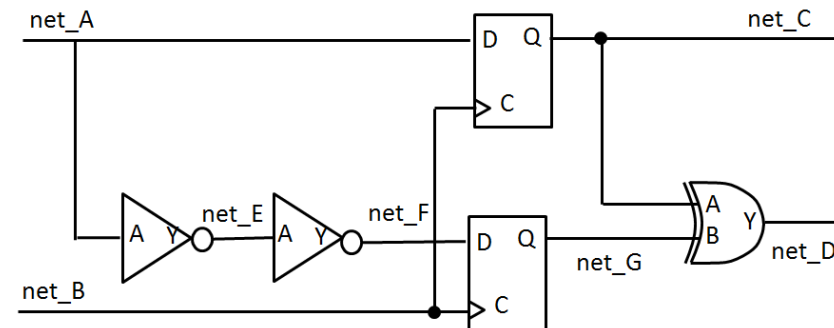
##### 4.1 カナリア FF のセルマッピング

説明のため、図5に変更前のFFの回路ブロックとネットリストに示した。ここで、FFのインスタンスである reg\_1 をカナリア FF へ変更することを考える。説明のためにインスタンス名とシグナル名は、わかりやすい名前に変更しており、net\_A、net\_B、net\_C については、FF 置き換えスクリプトによりカナリア FF へ置き換える個々の FF で使用している実際のシグナル名を抽出する必要がある。使用する FF 置き換えスクリプトは第3章で説明したものとほぼ同様であるが、FF を単純にカナリア FF へ置き換えるのではなく、カナリア FF を構成するセル群で置き換える必要がある。詳細について以降で述べる。



```
ROHM18DFP010 reg_1 ( .D(net_A), .C(net_B), .Q(net_C) );
```

図5 DFFとネットリスト (カナリア FF 変更前)



```
ROHM18DFP010 reg_1 ( .D(net_A), .C(net_B), .Q(net_C) );
ROHM18DFP010 UU_1 ( .D(net_F), .C(net_B), .Q(net_G) );
ROHM18INV010 UU_2 ( .A(net_A), .Y(net_E) );
ROHM18INV010 UU_3 ( .A(net_E), .Y(net_F) );
ROHM18XOR2P010 UU_4 ( .A(net_C), .B(net_G), .Y(net_D) );
```

図6 変更後カナリア FF とネットリスト

カナリア FF への変更後の回路ブロックとそのネットリストは図6に示すようになる。INV、EXOR は、使用しているスタンダードセルライブラリにすでに登録されているため、各ゲートをそれらのセルへ置き換える。使用するセルはすべて最小のドラ

イブ能力のものを採用した。UU\_1,UU\_2,UU\_3,UU\_4 は追加されたセルのインスタンス名、net\_D,net\_E,net\_F,net\_G は追加されたシグナル名である。これらの追加された名前は、元のネットリストで使用されている名前と重複を避けるために、適当なプレフィックス名と通し番号を組み合わせて作成すればよい。これらの名前は各モジュール定義内で一意であればよいため、通し番号はモジュール定義の開始部を FF 置き換えスクリプトが構文解析 (パース) した時にリセットし、新たな名前が使用される度に通し番号をインクリメントするようにしている。新規に追加されたシグナル名については、それが使用されているモジュールのモジュール定義部で宣言されている必要がある。しかし、モジュール定義部を FF 置き換えスクリプトがパースした段階では、どのシグナル名を宣言すればよいか分かっていないため、ここでは 2 重パース方式でシグナル宣言を行うことにした(図 7)。最初のパースでは、該当 FF をカナリア FF セルに置き換える処理を行うとともに、モジュール宣言の先頭部分のシグナル宣言を行う部分に目印としてのプレースホルダー{@wire\_decl@}を記述しておく。2 回目のパースでそのプレースホルダーを追加されたシグナル名の宣言文で置き換える。追加されたシグナル名は最初のパースでそれが宣言される予定のモジュール名から参照される連想配列を使用して管理するようにしている。

Netlist after 1st parsing	Netlist after 2st parsing
<pre> module module_A(a, b, c); input a; input b; output c; wire net_A, net_B, net_C; @wire_decl@ . DFF reg_1 (.D(net_A), .C(net_B), .Q(net_C)); DFF UU_1 (.D(net_F), .C(net_B), .Q(net_G)); INV UU_2 (.A(net_A), .Y(net_E)); INV UU_3 (.A(net_E), .Y(net_F)); XOR UU_4 (.A(net_C), .B(net_G), .Y(net_D)); . end module                     </pre>	<pre> module module_A(a, b, c); input a; input b; output c; wire net_A, net_B, net_C; <b>wire net_D, net_E, net_F, net_G;</b> . DFF reg_1 (.D(net_A), .C(net_B), .Q(net_C)); DFF UU_1 (.D(net_F), .C(net_B), .Q(net_G)); INV UU_2 (.A(net_A), .Y(net_E)); INV UU_3 (.A(net_E), .Y(net_F)); XOR UU_4 (.A(net_C), .B(net_G), .Y(net_D)); . end module                     </pre>

図 7 カナリア FF 置き換え後ネットリスト  
 (左：1 回目パース終了時、右：2 回目パース終了時)

#### 4.2 カナリア FF によるセル面積への影響

元のネットリストとカナリア FF に変更されたネットリストを用いて、Synopsys IC Compiler (D-2010.03-ICC-SP2-1)で実際に配置配線を行い、カナリア FF 置き換えによるチップ全体のセル面積への影響について調べた。ただし、カナリア FF から報知されるエラー信号については、現在のところ処理を行っていないため、固定的に LOW または HIGH いずれかの論理値に設定した。配置配線を行った結果を表 4 に示した。表からカナリア FF の置き換えによるセル面積増加は MeP プロセッサで約 5.5%、miniMips プロセッサで約 3.9%であり、いずれの場合もセル面積増加に対する影響を低く抑えることができることが分かった。比較のために、miniMips プロセッサのすべての FF をカナリア FF に変更した場合のセル面積増加率を調べた結果、表 4 の miniMips(\*)に示すように約 30%の面積増加となった。なお、個々のカナリア FF を構成するセル群の配置場所を調べたところ、どのセルも近くに配置されていることが確認できた。

表 4 カナリア FF 置き換えによるチップ全体のセル面積

プロセッサ	セル面積(カナリア FF 置き換え前) [μm <sup>2</sup> ]	セル面積(カナリア FF 置き換え後) [μm <sup>2</sup> ]	セル面積増加率
MeP	3664956.96	3865980.58	5.5%
miniMips	874156.95	908135.43	3.9%
miniMips(*)	874156.95	1136549.83	30.0%

以上の結果から、本稿で提案したカナリア FF の置き換え位置を限定する方法はセル面積の増加を抑制する効果的な手法であることが分かった。しかし、本稿では述べていないが、チップ全体の消費電力への影響についても同様に考慮しながら、カナリア FF の挿入位置を限定する必要があると思われる。図 8 は miniMips プロセッサで配置配線を行った結果のチップ全体を表しており、白い部分がカナリア FF セルで置き換えた部分である。図の左が全ての FF をカナリア FF に置き換えた場合で、右がカナリア FF の置き換え位置を限定した場合である。

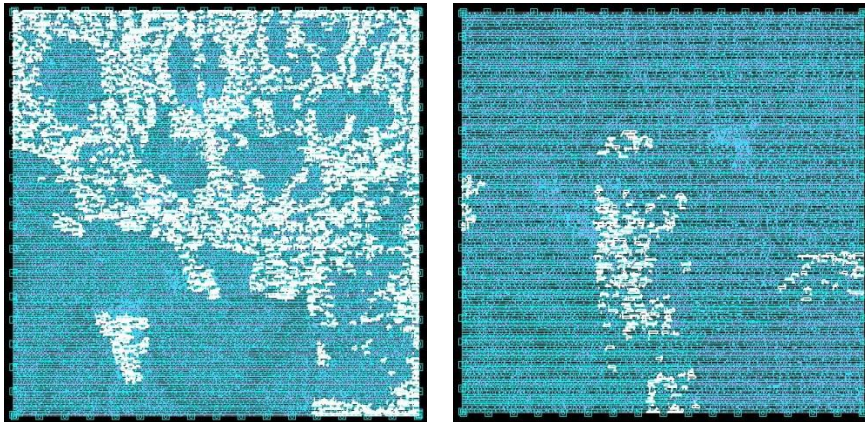


図 8 miniMips プロセッサ

(左：すべての FF をカナリア FF に変更 右：カナリア FF の置き換え位置を限定)

## 5. まとめと今後の課題

半導体技術の微細化によるディーブサブミクロン領域での素子特性のばらつきや経年劣化等による素子不良が原因のタイミングエラーを LSI 稼働中に判定して、LSI の動作不良を回避する技術の重要性が高まってきている。本稿では、改良カナリア FF を使用した高信頼性の LSI 設計手法について提案を行った。改良カナリア FF のセル化については、現時点では完成していないが、上記で述べた検討結果をもとに、今後も引き続き作業を行う予定である。また、本稿では論じていないがタイミングエラーを検知した際のエラー信号の収集についてもさらに検討を行う必要がある<sup>4)</sup>。仮に N 個のカナリア FF があるとして、エラー信号の収集を 2 ポートの OR ゲートで構成した場合、エラー信号が  $\log_2 N$  個の OR ゲートを伝搬することになり、それによる配線遅延の問題も無視できない。また、シャドウ FF への信号の遅延バッファ量についても今後検討を行う必要がある。遅延バッファ量が過大に見積もられた場合は、必要以上にタイミングエラー報知が発生して、チップ全体の性能低下につながる。逆に過小に見積もられた場合には、タイミングエラーを見逃す可能性が高くなり信頼性が低下する。さらにカナリア FF によるチップ全体の消費電力への影響についても今後の作業として残っている。LSI の検査診断の観点からは、経年劣化、ソフトエラー、電圧降下などによりタイミングエラーを仮想的に発生させてゲートレベル、アーキテクチャレベルでシミュレーションを行い提案手法の評価を行う必要がある。

**謝辞** 本研究の一部は九州大学と共同で実施している科学技術振興機構・CREST プロジェクト「ディペンダブル VLSI システムの基盤記述」の支援によるものである。本研究で使用したライブラリは京都大学情報学研究科田丸/小野寺研究室の成果によるものであり、京都工芸繊維大学小林和淑教授によりリリースされたものである。

## 参考文献

- 1) 小野寺、平田、北村、小林、田丸：P2Lib:スタンダードセルライブラリ自動生成システム,情報処理学会論文誌,Vol.40,No.4,pp.1660-1669,1999/4.
- 2) 佐藤、国武：ばらつき耐性を持つカナリア FF を利用したデザインマージン削減による省電力化, 情報処理論文誌, Vol.49 No.6,pp.2029-2042,2008.
- 3) 国武、佐藤、山口、安浦: タイミングエラーの予報を目的とするカナリア FF の挿入位置限定, 信学技報 DC2008-42, 2008.
- 4) 栗本、鈴木、秋山、山中、大熊、高田、篠原: エラー検出 FF を用いた DVS における Short Path Penalty と OR-Tree Latency の削減手法,信学技報,VLD2008-47,2008.
- 5) 山原、美馬、千代延、佐藤: タイミングを許容する省電力加算器における違反検出回路の高速化,情報処理学会論文誌コンピューティングシステム,Vol.47,SIG18(ACS16),2006.
- 6) D.Blaauw,S.kalaiselvan,K.Lai,S.Wei-Hsiang,C.Tokunaga,S.Da,and.Bull,"RazorII:In Situ Error Detection and Correction for PVT and SER Tolerance", International Solid-State Circuits Conference, pp.400-622,2008.
- 7) D.Ernst, N.S.Kim,S.Das,S.Pant,R.Rao,T.Pham,C.Ziesler,D.Blaauw,T.Austin,K.Flautner,and T.Mudge,"Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation",36<sup>th</sup> International Symposium Microarchitecture,pp.7-18,2003.
- 8) J. Furuta, C. Hamanaka, K. Kobayashi, and H. Onodera: A 65nm Bistable Cross-coupled Dual Modular Redundancy Flip-Flop capable of protecting soft errors on the C-element, IEEE Symposium on VLSI Circuits, 2010.
- 9) K.Hirose, Y.Manzaw, M.Goshima, and S.Sakai, "Delay-Compensation Flip-Flops for Timing-Error Tolerant Circuit Design", International Conference on Solid State Device and Materials,pp.440-481,2007.
- 10) S.Mitra,N.Seifert,M.Zhang,Q.Shi,K.S.Kim,"Robust System Design with Built-In Soft-Error Resilience", IEEE Computer, Vol.28, No.2, pp. 43-52, 2005.
- 11) Toshiba MeP プロセッサ,  
<http://www.semicon.toshiba.co.jp/product/micro/selection/mep/index.html>
- 12) miniMips プロセッサ, <http://opencores.org/project.minimips>
- 13) Tokyo University VDEC, <http://www.vdec.u-tokyo.ac.jp/>