

PPC における LUT 挿入位置最適化の定量的評価

都築 匠[†] 原 祐子^{**} 山下 茂^{***} 中島 康彦[†]

[†]奈良先端科学技術大学院大学 ^{**}東京工業大学, JST PRESTO ^{***}立命館大学

概要: 新素材を用いた回路などの製造故障率が高い回路においては、単純に回路を冗長化する (DMR) だけではなく、より面積効率のよい高信頼化手法が求められる。そこで、近年、新たな高信頼論理回路モデルとして Partially-Programmable Circuits (PPC) が提案された。PPC は組み合わせ回路を対象とし、設計時に ASIC の一部を LUT で置き換え、冗長結線を追加したもので、LUT のコンフィギュレーションの書き換えのみで製造時故障を回避可能である。本論文では、LUT の挿入位置を変えることで、従来手法に対して 3 入力 LUT で合成した PPC では平均約 10%、4 入力 LUT で合成した PPC では平均約 34% の歩留まりが向上できることを示す。また、実験により得られた知見をもとに、探索する設計空間を限定することで、平均約 80% の探索時間を削減した。

1. はじめに

近年、半導体の微細化に伴い、製造故障率が増加している。プリントドエレクトロニクスなどの新素材を用いた回路においては、さらに製造故障率の問題は深刻である[1]。これらの解決策として、従来、製造プロセスや回路マスクごとに製造歩留まりを改善する手法が施されてきた。しかし、このような低い抽象度での回路修正は、長い設計時間を要し、設計生産性を低下させる。そこで、LSI 設計の高い抽象度における歩留まり向上技術が必要とされている[2]。

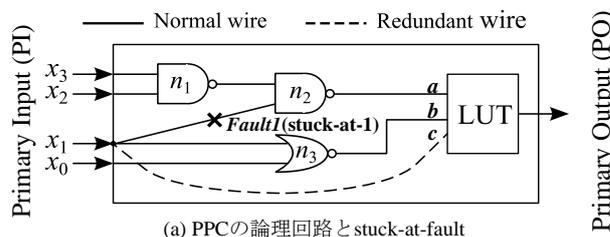
製造後に、故障を回避できるアーキテクチャの代表として FPGA (Field Programmable Gate Array) が挙げられる[3]。FPGA 中の故障していない部分を活用することで、故障を回避しつつ回路を実現する手法が数多く提案されている[4,5,6,7]。しかし、この手法は多くのリソースを使用するという問題がある。Cong らは、FPGA 中の故障を最大限活用することで、リソース消費量を削減する手法を提案している[8]。このように、FPGA の使用による有効な歩留まり改善手法は広く研究されているが、ASIC による回路を FPGA などのプログラマブルデバイスで実現した場合、実装面積が 40 倍、遅延時間が 3.2 倍、消費電力が 12 倍になるという結果も報告されており[3]、高い性能や小面積が求められるアプリケーションには適用できない。

ASIC における製造故障回避手法として、DMR (Dual Modular Redundancy) が挙げられる。DMR では、設計段階においてあらかじめ回路を二重化し、出力にマルチプレクサとメモリを追加する。製造後のテストにより、故障がないと診断された回路の出力をマルチプレクサで選択することで、回路中の故障を回避可能である。しかし、2 倍以上の面積オーバーヘッドが課せられるため、より面積効率の良い回路歩留まり向上手法が求められている。

山下らは、少ない面積オーバーヘッドで効率良く歩留まりを改善可能な論理回路の新たなモデル、PPC (Partially-Programmable Circuit) を提案した[9]。PPC は、ASIC の一部を論理の書き換えが可能な LUT (Look Up Table) で置き換え、回路中の適切な箇所から LUT の間に冗長な結線を追加

したものである。製造後に検出された故障に応じて、LUT の論理を書き換えるだけで、故障を回避できる可能性がある。PPC において少ない LUT で多くの故障に対応するためには、LUT および冗長結線の挿入箇所の最適な決定が重要である。PPC の既存の合成手法では、いずれも LUT は Primary Output の前段に置くことで、歩留まり改善が大きいとしていた[9,10,11]。しかしながら、LUT の挿入位置による PPC の歩留まり向上に関する定量的評価は行われていない。さらに、全探索により冗長結線の挿入位置を決定しており、非効率的である。すなわち、大規模な回路を PPC で実現するためには、LUT および冗長結線の最適な挿入箇所を実用的な時間内に決定する手法を確立する必要がある。

本研究では、少ない面積オーバーヘッドで効率良く歩留まり改善可能な PPC の合成手法の確立を目指し、LUT の挿入位置による歩留まり改善率の変化を定量的に評価する。さらに、冗長結線挿入箇所を決定する効率的な探索アルゴリズムの確立のために、冗長結線の挿入箇所と回避可能な故障数の関係、LUT の挿入箇所と回避可能な故障の関係についても評価する。これらの結果から得られた知見をもとに、



(a) PPCの論理回路とstuck-at-fault

x_2, x_3 x_0, x_1	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	1	0	1
10	0	0	0	0

(b) PIに対するLUTの論理

a, b	c	0	1	0	1
00	0	0	0	0	0
01	1	*	*	1	*
11	*	*	*	1	*
10	*	1	1	0	1

(c) Conf.0

(d) Conf.1

図1 PPCの故障訂正の例

LUT の挿入箇所および冗長結線の探索空間を効率良く削減することで、探索時間を平均約 80%削減することができた。

本論文では、2 章で PPC について説明する。3 章で実験手順と評価指標について述べ、4 章で評価実験と考察を行う。最後に、5 章でまとめと今後の課題について述べる。

2. PPC (Partially Programmable Circuit)

PPC とは組み合わせ回路を対象として、ASIC の一部を論理の書き換えが可能な LUT で置き換え、残りの回路の適切な位置より冗長な結線を LUT に追加した回路である[9]。PPC は、製造後に故障が検出された場合であっても、LUT のメモリを書き換えることで、もともとの回路の論理を維持し、製造歩留まりを向上させる新たな手法として着目されている[12]。図 1 に PPC による故障訂正の例を示す。まず、図 1(a)が合成された PPC であり、図 1(b)がこの回路における入力と出力の論理である。これを実現するための LUT のコンフィギュレーションの一つを図 1(c)に示す。ここで、入力 x_1 からゲート n_2 の間で stuck-at-1 故障が検出されたとする。その場合、図 1(d)のようにコンフィギュレーションを変更することで、正しい回路の論理を維持することができる。このように、PPC は故障が検出された場合においても、LUT の論理の変更のみで、故障を回避し、正しく動作することができる可能性がある。

2.1 PPC の合成手順

本節では PPC の合成手順について説明する[9][11]。最終的な LUT の入力数を l 、追加する冗長な結線の本数を m とし、合成した PPC で訂正可能なパターン数を n とする。Stuck-at-0 故障、stuck-at-1 故障をそれぞれ 1 パターンとして、訂正可能なパターン数 n を得る。

以下に PPC 合成の具体的な手順を示す。

- I. RTL(Register Transfer Level) 回路を入力とし、 $(l-m)$ 入力 LUT のネットワークにマッピングする。
- II. マッピングを行った LUT のネットワークから、最終的に PPC 内の LUT として残す LUT を除いた回路をスタンダードセルに再マッピングする。
- III. 残した LUT に対して、スタンダードセルに再マップした箇所のうち、LUT の入力のファンインコンの適切な位置から m 本の結線を冗長結線として追加する。すでに II.において LUT に直接接続しているゲートは、結線の追加候補から除外する。
- IV. 合成した PPC における訂正可能箇所 (n) を求める。各訂正可能箇所に対してたかだか 1 つの *config* があれば十分であり、合計 n 個の *config* を構成する。
- V. 全 *config* を併合し、LUT のメモリを最小化する。
- VI. LUT 内部の選択線の最適化を行う。

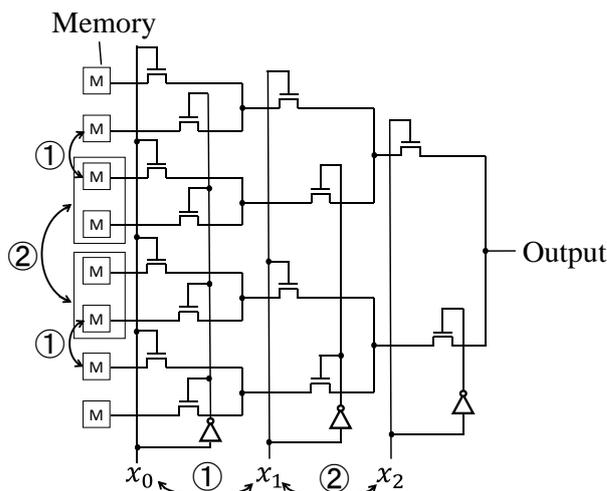


図2 3入力 LUT

$a \backslash b \ c$	0	1
00	*	0
01	*	*
11	1	*
10	0	1

Conf. 0

	0	1
0	0	0
1	0	*
1	1	*
0	0	1

Conf. 1

	0	1
0	*	0
0	*	*
1	1	*
1	1	1

Conf. 2

	0	1
0	0	*
*	*	*
0	0	*
0	0	*

Conf. 3

	0	1
0	*	*
*	*	0
*	*	1
*	*	1

Conf. 4

	0	1
0	0	0
0	0	0
M	1	1
M	1	1

Conf. LUT

図3 LUT のメモリ削減例

既存研究では、LUT の挿入位置は Primary Output の直前に限定されていた[9,10,11]。これは、Primary Output の前段に入れることで LUT の入力のファンインコンに含まれるゲート数が最大となり、より多くの故障が訂正できると考えられたからである。しかし、実際に LUT を Primary Output 直前以外に入れる場合について、LUT の挿入位置と歩留まり向上の関係の定量的な評価はされていない。また、これまで最適な PPC を合成するために冗長結線の追加位置に対して全探索を行ってきた。しかし、LUT の追加位置も変更することや、より大きなベンチマーク回路に対して計算を行う上で、全探索では膨大な計算時間がかかる。

本研究では、LUT の配置 (前述 II) や冗長な結線の追加位置 (前述 III) を変更し、PPC の歩留まり改善効率にどのように影響するのかを評価・考察する。さらに、冗長結線の挿入箇所について、全探索ではない効率的なアルゴリズムを用いて探索時間を削減することを目指し、冗長結線の挿入位置と歩留まり向上の関係、訂正可能箇所と LUT の位置との関係についての評価を行う。

2.2 PPC における LUT の実装と面積削減手法

前節で説明した PPC の具体的な LUT のモデルやそのトランジスタ数削減手法について示す。

図 2 に 3 入力 LUT のハードウェアでの実現を示す。この LUT では $2^3 = 256$ 通りの論理が実現できる。しかしながら、前節で述べたとおり、1 つの故障に対してただか 1 つのコンフィギュレーションがあれば十分である。そこで、訂正可能な故障の数 n の論理を表現するために必要な最低限のメモリと選択線とで LUT を構成することで、歩留まりの改善効果を維持したまま、LUT の面積を削減する手法を[11]で提案した。また、図 2 において①や②のように LUT の入力線を入れ替えた場合でも、対応するメモリの配置を入れ替えることで、より少ない面積で同等の論理を実現できる。この性質を生かし、選択線のトランジスタの削減も行う手法も提案した[11]。

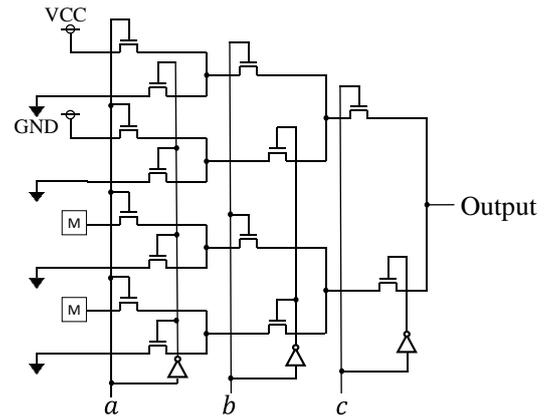
メモリの削減について図 3 を用いて説明する。図 3 ではもともとの回路の論理である $conf.0$ と訂正可能箇所ごとの $conf.1\sim 4$ を示している。そこで、これらの $conf$ を併合し、LUT に最終的に必要なメモリを以下のように求める。

- すべての $conf$ において真理値表の値が 1 またはドントケアである場合、最終的な真理値は 1 とする。
- すべての $conf$ において真理値表の値が 0 またはドントケアである場合、最終的な真理値は 0 とする。
- 真理値表の値に 1 と 0 が混在している場合、どちらも表現できる必要があるためメモリとする。

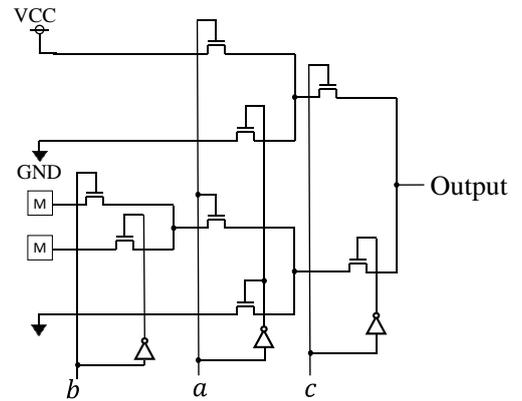
これにより得られたコンフィギュレーションが、図 3 の $Conf.LUT$ である。さらに、このコンフィギュレーションから作成した LUT を図 4(a) に示す。最後に、LUT の入力線入れ替えを行い、選択線のトランジスタ削減を行う。図 4 では、図 2 の①の入れ替えと②の入れ替えを交互に行い、入力 a, b, c の入力順の全組み合わせを試し、トランジスタ数最小の組み合わせを求め、最終的に図 4(b) の LUT を得る。このようにして、LUT のメモリと選択線のトランジスタを必要最低限に削減することで、PPC の面積削減に寄与している。

3. 実験手順と評価指標

本研究では、論理合成ツールである ABC[13] をベースとした、PPC の合成フレームワークを開発し、実験・評価を行った。本章では、ベンチマークおよび実験環境について説明する。表 1 に本実験で使用したベンチマーク回路を示す。ベンチマーク回路は MCNC ベンチマークから、多入力 1 出力回路を選んだ。Node Counts は各回路をスタンダードセル (2 入力 NAND と Inverter) でマップしたネットワークのゲートの総数で、Depth はその深さである。また、LUT Network size は各回路を、LUT のみでマップした際の LUT 数で、Inputs は Primary Input の数である。



(a) 最終 $conf$ から作成した LUT



(b) 入力線を入れ替えた LUT

図 4 入力線入れ替えによる選択線のトランジスタ削減

表 1 ベンチマーク回路

Circuit Name	Node Counts	Depth	LUT Network size (l=3)	LUT Network size (l=4)	Inputs
majority	20	6	8	10	5
cm152a	35	6	21	43	11
parity	91	9	15	4	16
cm150a	113	13	61	7	21
mux	136	16	76	16	21
9symml	298	17	196	112	9

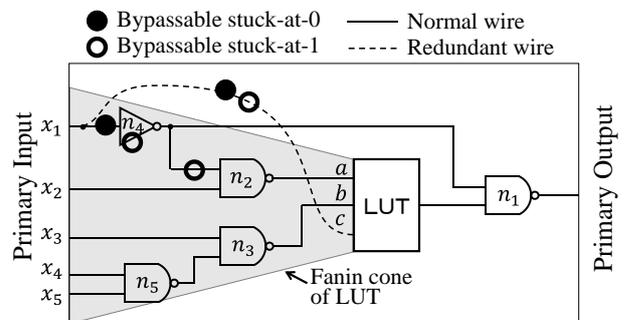


図 5 PPC と訂正可能箇所の例

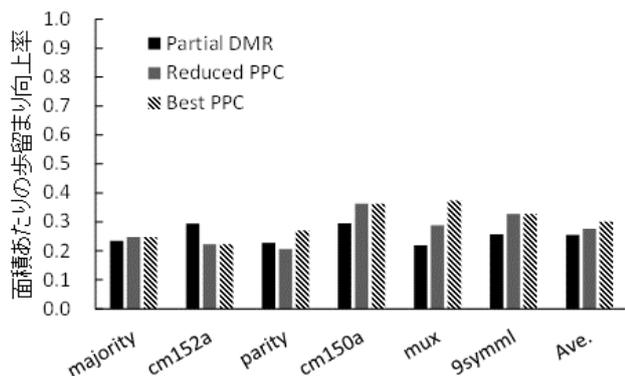


図6 3入力LUTで合成したPPCの単位面積オーバーヘッドに対する歩留まり向上率

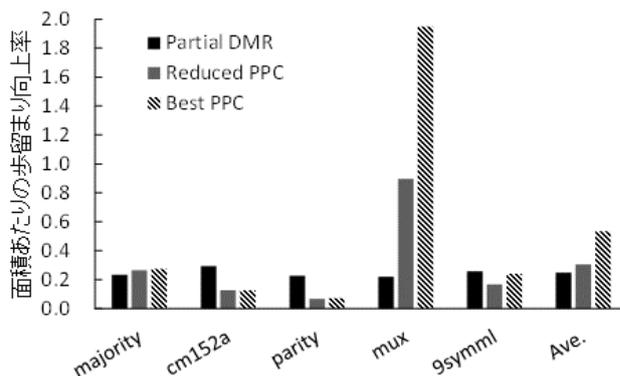


図7 4入力LUTで合成したPPCの単位面積オーバーヘッドに対する歩留まり向上率

3.1 実験条件

本実験では、各ベンチマーク回路に対して2.1節で述べた方法でPPCを合成した。PPCとして残すLUTは1つのみとし、各ベンチマーク回路のLUTの全候補に対して3.2節で説明する指標について評価した。3入力および4入力の2種のLUTを用い、LUTへの入力のうち1本は冗長結線に使用した。

図5にPPCとその故障訂正可能位置の例を示す。本実験で対象とする故障は、stuck-at-0故障とstuck-at-1故障であり、回路中の結線、ゲート、LUTの出力における単一故障を仮定した。図5では、冗長結線を含めて13本の結線と5つのゲート、および、1つのLUTで、計19カ所で故障が起こりうるため、故障のパターン数はstuck-at-0故障とstuck-at-1故障を合わせて38となる。この全故障パターン数を p_{total} と定義する。また、LUTのファンインコーン外において故障を訂正できる可能性は極めて低く、今回の実験では訂正可能箇所の探索範囲から除外しても評価に大きな影響はない[9]。そのため、訂正可能箇所は回路中の全故障箇所のうち、LUTのファンインコーン(図5灰色の範囲)内のゲートと結線が候補となり、それらに対して起こる故障がLUTで訂正可能か判別する処理を行う。その結果、図5ではLUTのファンインコーン内に5つの訂正可能箇所が検出されたと仮定する。しかし、LUTをPrimary Outputの前段以外に挿入する場合、LUTの出力では訂正可能であっても、LUTをすり抜け、故障が出力に伝播する場合がある。そこで、各訂正可能箇所に対し故障が伝播するゲートを求め、LUTのファンインコーンに含まれないゲートにも伝播する場合、その故障は訂正不可能とした。したがって、図5では、ゲート n_4 の出力がLUTをすり抜け、ゲート n_1 の入力に接続されているため、ゲート n_4 のstuck-at-1故障および x_1-n_4 間の結線のstuck-at-0故障は訂正不可能となる。すなわち、図5における訂正可能数は3となる。このときの訂正可能数を $p_{bypassable}$ と定義する。

3.2 評価指標

歩留まり改善手法により、回路面積のオーバーヘッドは異なる。本実験では、それらの手法について定量的に歩留まりの改善効率を比較するため、式(2)に示す、面積当たりの歩留まり向上率という指標を用いる。歩留まり向上率、面積当たりの歩留まり向上率は、結線、ゲート、およびLUTの総数(p_{total})、冗長化により単一縮退故障に対して故障回避になった結線およびゲートの総数($p_{bypassable}$)、冗長化による面積オーバーヘッド($A_{overhead}$)、冗長化前の回路面積(A_{orig})より以下の式で表される。

$$\text{歩留まり向上率} = \frac{p_{bypassable}}{p_{total}} \quad (1)$$

$$\text{面積当たりの歩留まり向上率} = \frac{p_{bypassable}/p_{total}}{A_{overhead}/A_{orig}} \quad (2)$$

面積オーバーヘッドが10%に対して、歩留まりの向上が5%、10%、20%の場合、式(2)の値は0.5、1.0、2.0となる。DMRはこの指標では1.0を超えることはないが、PPCでは1.0以上の値を取ることがある。回路全体の面積 A_{orig} と面積オーバーヘッド $A_{overhead}$ は、トランジスタ数で評価した。本実験では、NANDは4個、Inverterは2個、メモリは18個のトランジスタで構成されているとして、面積を求めた。

また、本実験では、各ベンチマーク回路に対して、歩留まり改善効率を含めたさまざまな視点から、以下の3つの手法の比較・評価を行う。

Partial DMR: 部分的に回路の二重化を行い、故障訂正能力を持つ回路で、先行研究[11]で比較対象として用いられた手法。本実験では、Reduced PPCと同程度の故障訂正性能を持つ回路をPartial DMRにより生成した。

Reduced PPC: 先行研究[11]で提案された手法。Primary Outputの前段のLUTを残してPPCを合成した。

Best PPC: すべてのLUTの挿入候補と冗長結線の追加候補を対象に全探索を行い、その中で式(2)の値が最良のPPCを選択した。

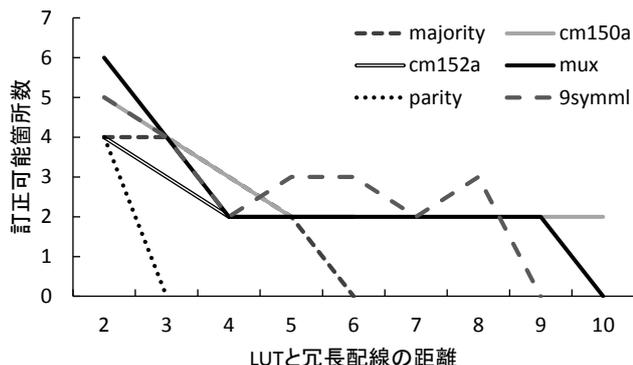


図8 3入力 LUT で合成した PPC の冗長結線の追加位置

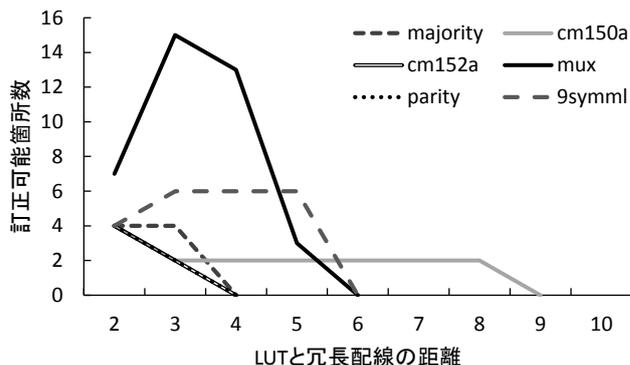


図9 4入力 LUT で合成した PPC の冗長結線の追加位置

4. 評価実験と考察

各ベンチマーク回路に対して、3.2 節の評価指標について比較・評価を行い、考察を述べる。

4.1 単位面積あたりの歩留まり向上率

図 6 に 3 入力 LUT, 図 7 に 4 入力 LUT で合成した PPC について、3.2 節の式(2)から求めた単位面積オーバーヘッドに対する歩留まり向上率のグラフを示す。図 7 において cm150a の項目がないのは、合成した PPC のトランジスタ数が冗長化前の ASIC におけるトランジスタ数より小さく、オーバーヘッドが負の値となり、単位面積あたりの歩留まり向上率では比較できなかったためである。DMR では、冗長化後の回路が小さくなるということは決してないことに留意されたい。すなわち、PPC の面積効率の高さが顕著に表れた結果である。

まず、図 6 では、Reduced PPC の歩留まり向上率は Partial DMR と同程度か若干の改善があり、文献[11]で示された傾向とも一致する。次に、Reduced PPC と Best PPC を比較すると、従来手法である Reduced PPC に比べ、Best PPC は平均約 10%の改善を得ている。図 7 においては、Reduced PPC は cm152a, parity, 9symml で、Partial DMR よりも低い結果になっている。これは、回路全体の面積に対する 4 入力 LUT の面積の割合が大きいためであると考えられる。一方、majority や mux は LUT の面積を効率良く削減でき、Partial DMR よりも高い改善効率を得ている。また、Best PPC は、Reduced PPC に比べ、平均約 34%の改善を得た。以上の結果より、ほとんどの回路において、Primary Output の前段以外に LUT の最適な挿入箇所が存在することが明らかになった。

4.2 冗長結線の追加箇所と訂正可能数

図 8 に 3 入力 LUT, 図 9 に 4 入力 LUT で合成した PPC の冗長結線の追加位置と訂正可能箇所の関係を示す。このグラフでは、縦軸に訂正可能数、横軸に冗長結線の LUT からの距離 (段数) を示している。各ベンチマーク回路にお

いて Best PPC を求め、その LUT の挿入位置に対し、LUT から 2 段目以降のゲートを候補とし冗長結線の追加を行い、その位置で訂正可能箇所数がどのように変化するかを評価した。ここで、LUT から 2 段目のゲートとは、図 5 の場合、ゲート n_4 , n_5 を指す。同じゲートから LUT にたどり着くパスが複数ある場合は、最短パスの段数を評価に用いた。

図 8 では、各ベンチマークにおいて訂正可能箇所数のピークが 2 段目にあり、冗長結線を LUT の 2 段目のゲートに接続した場合に最も訂正可能箇所数が多い。一方、図 9 では、2 段目あるいは 3 段目に訂正可能箇所数のピークが存在している。このことから、本実験では LUT から 2~3 段目と、比較的 LUT から近い位置に冗長結線を追加することが望ましいと言える。

4.3 訂正可能箇所の LUT からの距離の分布

各ベンチマークについて、Best PPC における LUT の挿入位置、および、冗長結線の追加位置のすべての組み合わせに対して、訂正可能箇所数を求めた。LUT からの各距離における訂正可能候補数に対する訂正可能箇所数の割合について、図 10 に 3 入力 LUT, 図 11 に 4 入力 LUT で合成した PPC の結果を示す。例えば、図 10 の majority では約 80%の割合で 1 段目¹が訂正可能であることを示している。

図 10, 図 11 とともに、1~3 段目に訂正可能箇所が集中しており、LUT から近い箇所に分布していることが分かる。したがって、限定された範囲 (例えば、1~3 段目の範囲) のみで訂正可能箇所数の評価を行っても、その精度にはほとんど影響がないと考えられる。

4.4 合成時間削減についての考察

本実験では、比較的規模の小さいベンチマーク回路に対して評価を行った。これは、LUT の挿入箇所に自由度を持たせ、Primary Output 前段のみでなく、回路全体を考慮することで、PPC 合成の計算量²が

冗長結線の候補数 \times 訂正可能箇所候補数 = $O(mn)$ から、

¹ 1 段目とは、図 5 の場合、ゲート n_2 , n_3 と結線 $n_2 - a$, $n_3 - b$, $x_1 - c$ を指す。

² n は冗長結線の候補数、 m は訂正可能箇所候補数、 n は LUT の挿入候補数を表す

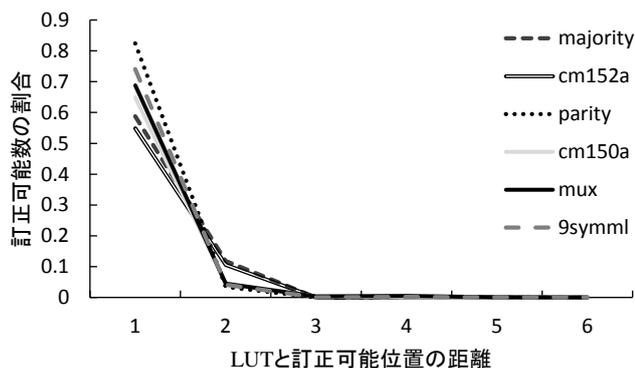


図 10 3 入力 LUT で合成した PPC の訂正可能箇所
LUT からの距離と訂正可能割合

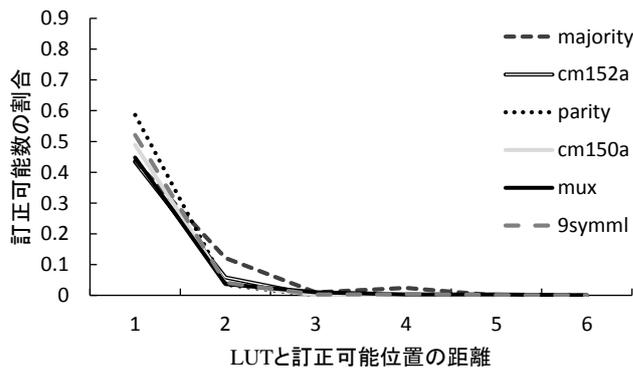


図 11 4 入力 LUT で合成した PPC の訂正可能箇所
LUT からの距離と訂正可能割合

$$\begin{aligned} & \text{冗長結線の候補数} \times \text{訂正可能箇所候補数} \\ & \quad \times \text{LUT の挿入候補数} = O(lmn) \end{aligned}$$

に増大したためである。本実験結果により得られた知見から、訂正可能箇所の候補と冗長結線の追加候補を 3 段目までに限定した場合、計算量は

$$\begin{aligned} & \text{定数(3 段目までの冗長結線の候補数)} \\ & \times \text{定数(3 段目までの訂正可能箇所候補数)} \\ & \times \text{LUT の挿入候補数} = O(l) \end{aligned}$$

に抑えることができる。この手法を本実験で使用したベンチマーク回路に適応したところ、3 入力 LUT で合成した PPC の計算量が平均約 81%、4 入力 LUT で合成した PPC の計算量が平均約 80%削減できることが分かった。これにより、実用的な規模の回路に対しても PPC 化が可能となり、従来手法 (DMR など) より高い歩留まり改善効率が期待される。

5. おわりに

本研究では、より少ない面積オーバーヘッドで高い歩留まり改善効果を得る PPC の合成手法の確立を目指し、PPC 中の LUT の挿入位置による面積当たりの歩留まり向上率、冗長結線の追加箇所と訂正可能数、LUT の距離に対する訂正可能割合について評価を行った。LUT の挿入位置を最適化することで、既存の PPC 合成手法に比べ、3 入力 LUT で合成した場合には平均約 10%、4 入力 LUT で合成した場合には平均約 34%の歩留まり改善効率的向上を得た。さらに、最適な冗長結線追位置は LUT から 2~3 段目、訂正可能位置のほとんどが LUT から 3 段目以内であることが明らかになった。これらの知見を生かし、探索範囲を限定することで、PPC の合成時間を平均約 80%削減できた。

今後は、本研究で得られた知見を実用的な規模の回路に適用し、より効率良く LUT および冗長結線の挿入箇所を決定するアルゴリズムの開発を行う。

謝辞

本研究の一部はカシオ科学振興財団、科学研究費補助金基盤(A)24240005、基盤(B)23300019、萌芽 24650020 の助成

を受けて行われたものである。本研究に対し有益な助言とフレームワークの提供をして頂いた、Fujitsu Laboratories of America の吉田浩章博士、および、早苗駿一氏に感謝する。

参考文献

- [1] J. Henkel et al., "Design and Architectures for Dependable Embedded Systems", *Proc. CODES+ISSS*, pp.69-78, 2011.
- [2] M. Bühler et al., "DFM/DFY design for manufacturability and yield-influence of process variations in digital, analog and mixed-signal circuit design", *Proc. DATE*, pp.387-392, 2006.
- [3] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp.203-215, 2007.
- [4] B. Ghavami et al., "Defect and Variation Issues on Design Mapping of Reconfigurable Nanoscale Crossbars", *Proc. ISVLSI*, pp.173-178, 2010.
- [5] R. Chakraborty et al., "Low-power hybrid complementary metal-oxide-semiconductor-nano-electro-mechanical systems field programmable gate array: circuit level analysis and defect-aware mapping", *IET Computers and Digital Techniques*, vol. 3, no. 6, pp.609-624, 2009.
- [6] Y. Su and W. Rao, "Defect-Tolerant Logic Implementation onto Nanocrossbars by Exploiting Mapping and Morphing Simultaneously", *Proc. ICCAD*, pp. 456-462, 2011.
- [7] A. Agarwal et al., "Fault Tolerant Placement and Defect Reconfiguration for nano-FPGAs", *Proc. ICCAD*, pp.714-721, 2008.
- [8] J. Cong and B. Xiao, "Defect Tolerance in Nanodevice-Based Programmable Interconnects: Utilization Beyond Avoidance", *Proc. DAC*, pp.1-8, 2013.
- [9] S. Yamashita, H. Yoshida, and M. Fujita, "Increasing Yield Using Partially-Programmable Circuits", *Proc. SASIMI*, pp.237-242, 2010.
- [10] 早苗駿一, 原祐子, 山下茂, 中島康彦, "Partially-Programmable Circuit の歩留まり向上のための LUT の最適化手法", *DA シンポジウム*, pp.27-32, 2013 年 8 月.
- [11] S. Sanae et al., "Better-than-DMR Techniques for Yield Improvement", *Proc. FCCM*, p.34, 2014.
- [12] Y. Hara-Azumi et al., "High-Level Synthesis Using Partially-Programmable Resources for Yield Improvement", *Proc. SASIMI*, pp.414-419, 2012.
- [13] Berkeley Logic Synthesis and Verification Group, "ABC: A System for Sequential Synthesis and Verification", <http://www.eecs.berkeley.edu/alanmi/abc/>