

FPGAの配線領域に対する欠陥救済

金子 智[†] 伊藤 秀男^{††}

[†]千葉大学 自然科学研究科

^{††}千葉大学 工学部

〒263-8522 千葉市稲毛区弥生町1-33

E-MAIL:{satoshi,ito}@icsd2.tj.chiba-u.ac.jp

あらまし FPGA(Field Programmable Gate Array)は、内部回路を電氣的にプログラムすることによりユーザの手元で様々な論理回路を実現できるデバイスである。なかでも、SRAM-based FPGAは何回でもプログラムすることができるため、様々な分野で利用されるようになってきている。これに伴い、FPGAチップの歩留りの向上が要求され、FPGAに対する欠陥救済は重要になっている。FPGA内で欠陥箇所を特定することが可能であれば、ユーザは正常部分だけでFPGAを有効に使用することができる。本稿では、構成データのシフトによりFPGAの配線領域に対する欠陥救済を実現する手法を提案する。提案手法では、元の構成データを再利用するため、再マッピングが不要であり、欠陥救済に要する時間を短縮することができる。

キーワード SRAM-based FPGA, 欠陥救済, 構成データ, シフト

Defect and Fault Tolerance for Interconnections in FPGAs

Satoshi Kaneko[†] Hideo Ito^{††}

[†] Graduate School of Science and Technology, Chiba University

^{††} Faculty of Engineering, Chiba University

1-33, Yayoi-cho, Inage-ku, Chiba 263-8522, Japan

E-MAIL:{satoshi,ito}@icsd2.tj.chiba-u.ac.jp

Abstract Field Programmable Gate Arrays (FPGAs) can realize an arbitrary logic circuit in the field. One important class is the SRAM-based FPGAs which have been available for many fields, since they can be reprogrammed any number of times. Therefore, there is a critical need for defect and fault tolerance for FPGAs to increase chip yields. If faulty components in an FPGA can be located by fault diagnosis, the FPGA is made possible for the users to use effectively by means of isolating the faulty components. In this paper, a defect and fault tolerance approach for interconnections in FPGAs by shifting configuration data is proposed. The proposed approach does not need for re-mapping, because it reuses an original configuration data. Therefore, the reconfiguration time for defect and fault tolerance can be reduced.

key words SRAM-based FPGA, defect and fault tolerance, configuration data, shift

1 はじめに

FPGA(Field Programmable Gate Array)とは、内部回路を電気的にプログラムすることによりユーザーの手元で様々な論理回路を実現できるデバイスのことである [1]。なかでも、SRAM-based FPGA(以下、単に、FPGA と呼ぶ)は内部回路の機能を書き換えることができるので、デジタル回路の試作のみならず、画像処理や再構成可能な計算機アーキテクチャなど、従来の技術では実現し得なかった様々な分野で利用されるようになってきている [2]~[4]。これに伴い、FPGA チップの歩留りや信頼性の向上が要求され、FPGA に対する欠陥救済は重要になっている。欠陥救済とは、構成要素の一部に欠陥が存在しても、欠陥を回避して良品チップとして使えるようにすることである [5]。FPGA 内で欠陥箇所を特定することが可能であれば、正常部分だけで FPGA を有効に使用することができる。

FPGA の欠陥救済に対して、これまでにいくつかの手法 [6]~[10] が提案されている。文献 [6] の手法は、マッピング (FPGA 上に論理を割り当て、配置配線を行うこと) を再度行う手法であり、文献 [7] の手法は、あらかじめ複数のマッピング結果を出力・保存しておき、欠陥部分が特定されたとき、その部分を使用しないものを別のマッピング結果の中から選択して回路を再構成する手法である。これらの手法に対し、文献 [8]~[10] の手法は、構成データをシフトすることにより欠陥救済を実現する手法である。

本稿では、構成データをシフトすることにより FPGA の配線領域に対して欠陥救済を実現する手法を提案する。われわれは既に論理ブロックに対する欠陥救済手法は提案したが、配線領域に対する欠陥救済は含んでいなかった [9][10]。本稿で提案する手法は、再マッピングを行う手法と比較して欠陥救済に要する時間を短縮することができる。文献 [8] の手法も、元の構成データを利用することにより配線領域に対する欠陥救済を実現する手法であるが、欠陥を回避するためにヒューズを必要とする。これに対し、本稿で提案する手法はヒューズを必要としないため、文献 [8] の手法と比較して面積オーバーヘッドが小さい。

本稿は次のようになっている。まず、2 節で対象とする FPGA の構造と故障モデルについて述べる。次に、3 節では提案する欠陥救済手法について説明し、4 節で提案手法の評価および考察を行う。最後に 5 節でまとめを行い、今後の課題について述べる。

2 FPGA の構造と故障モデル

この節では、FPGA の構造と故障モデルについて述べる。

2.1 FPGA の構成

FPGA は、図 1 に示すように、アレー状に配置された論理ブロック (以下、CLB と記す) とそれらを取り囲む配線領域から構成される [1]。配線領域の配線が交差する部分には、それらを接続するための Switch Block (S ブロック) が配置され、FPGA の外周に沿った部分には、外部との接続を行う I/O Block (IOB) が配置されている。CLB の入出力線と配線領域の接続は Connection Block (C ブロック) により行う。CLB 内部の論理や配線領域の接続等の設定情報 (構成データ) はメモリセルにより記憶される。FPGA には多数のメモリセルが内包されている。以下、メモリセルに構成データを書き込むことをプログラムすると言う。本稿では、Xilinx 社のアーキテクチャに見られるように、メモリセルは直列に接続されているものとし、構成データは直列に書き込まれるものとする [11]。

図 1 の場合、CLB の数は 4×4 であるが、以下、 $N \times N$ とする。また、S ブロックに接続される配線の本数は、上下左右ともに 3 本であるが、以下、 n 本とする。

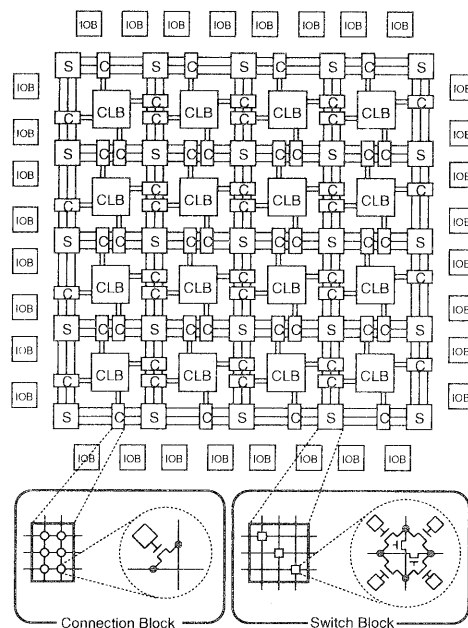


図 1 : FPGA の構成

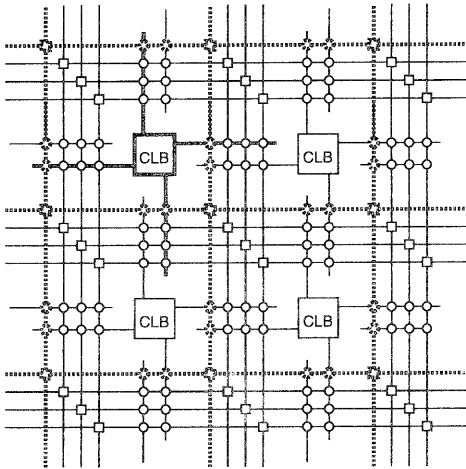


図 2: CLB と部分配線領域

2.2 CLB および配線領域の構成

一般的に、1つの CLB は、ルックアップテーブル、フリップフロップおよびマルチプレクサから構成される [1]。図 2 中の太い実線で示されているのが 1 つの CLB である。本稿では、CLB に接続されている入出力配線は CLB に属するものとする。以下では、説明の簡略化のため、CLB の上、右、下、左側に接続されている配線の本数はそれぞれ 1 本とする。

S ブロック内には n 個のスイッチが対角線上に配置されている。本稿では、図 2 中の太い点線で示されている配線およびスイッチの集合を部分配線領域と呼ぶことにする。S ブロック内のスイッチの数と部分配線領域の数は同じである。本稿では、C ブロック内のスイッチは配線領域に属するものとする。以下では、説明の簡略化のため、IOB については考慮しない。

2.3 故障モデル

本稿では、故障に関して、以下の仮定を設ける。

[仮定 1] 故障は、配線領域におけるただか一つの部分配線領域にのみ存在するものとする。配線領域の故障は配線の縮退故障またはスイッチの交点故障のいずれかである。スイッチの交点故障とは、スイッチにより互いに接続される配線同士が恒久的に接続状態になったり、逆に恒久的に非接続状態になる故障である。

[仮定 2] 一つの部分配線領域については、同時に複数の故障が存在してもよいものとする。

[仮定 3] FPGA をプログラムするための機構はすべて正常であるとする。すなわち、構成データはメモリセルに正しく書き込まれるものとする。

3 配線領域の欠陥救済

本稿で提案する配線領域の欠陥救済手法の手順は以下のとおりである。

まず、あらかじめいくつかの CLB や配線、スイッチをスペアとして確保しておき、マッピングを行う。スペアとして確保されているスイッチに関するメモリセルには、そのスイッチを非接続状態にするような構成データをあらかじめ格納しておく。しかし、スペアとして確保されている CLB に関するメモリセルには、どのような論理を実現する構成データを格納しておいてもよい。これは、CLB 内部でどのような論理が実現されていたとしても、その CLB の周囲にある C ブロック内のスイッチを非接続状態にすることにより、配線領域や他の CLB に対して影響を与えないようにすることができるからである。そして、欠陥が発生した時、欠陥箇所を特定することが可能であるなら、スペアとして確保しておいた部分に関する構成データを欠陥箇所のメモリセルにシフトすることにより、欠陥箇所を回避することができる。また、スペア部分に関する構成データのシフトに合わせて、欠陥箇所に関係のない部分に関する構成データもシフトする必要がある。

このように、提案手法の基本的方針は、構成データ全体をシフトすることによって欠陥救済を実現するというものである。したがって、欠陥部分に関係のない部分の構成データもシフトされることになるが、FPGA の構造は規則的であることから、シフト後においてもシフト前と同じ論理を実現することができる。

この節では、提案手法を実現するために必要なスペアの配置方法およびメモリセルの接続方法を示した後、それらに基づいた配線領域の欠陥救済手法について説明する。

3.1 スペアの配置

構成データのシフトにより配線領域の欠陥救済を実現するためには、いくつかの CLB と 1 つの部分配線領域をスペアとして確保しておく必要がある。

CLB については、スペアの配置方法がいくつか考えられる。われわれは、FPGA 全体 (CLB や配線領域、メモリセル等) に対する欠陥救済手法を考案することを最終的な目的にしている。よって、本稿では、スペアの CLB の配置はわれわれが提案した配置方法と同じものとする (図 3 中の黒い四角で示してある CLB がスペアである) [9][10]。

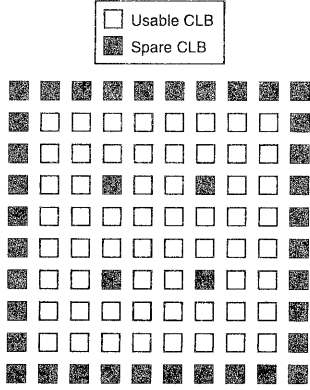


図 3: CLB に対するスベアの配置

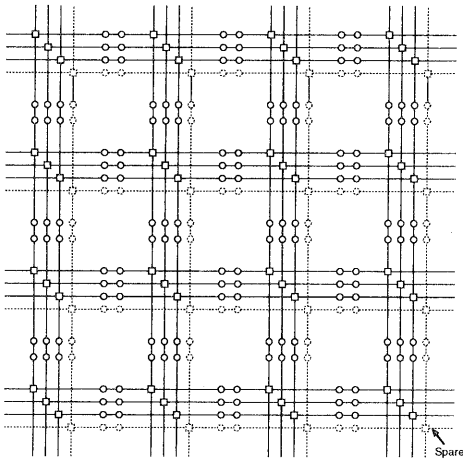


図 4: 配線領域におけるスベアの配置

配線領域については、故障モデルより、ある1つの部分配線領域をスベアとして確保しておく必要がある。本稿では、例として、Sブロック内の一番右下スイッチを含むような部分配線領域をスベアとして確保しておくことにする(図4中の点線で示してある部分がスベアの部分配線領域である)。

3.2 メモリセルの接続方法

2.1節では、メモリセルは直列に接続されていると仮定したが、提案手法を実現するためには、メモリセルを以下の3つの部分に分けて考える必要がある。

- CLBに関するメモリセル
- Sに関するメモリセル
- Cに関するメモリセル

説明のため、図5, 6のようにラベル付けを行う。図5ではCLB, Sブロック, および, Cブロックについて

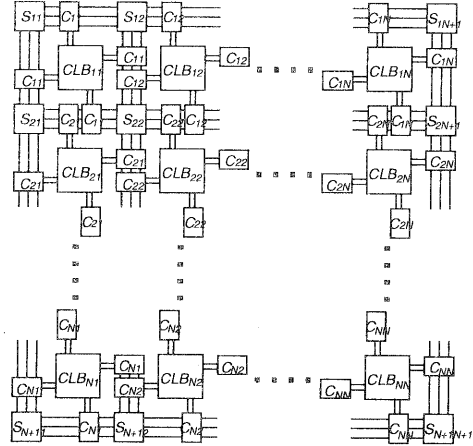


図 5: CLB, S ブロック, C ブロックのラベル付け

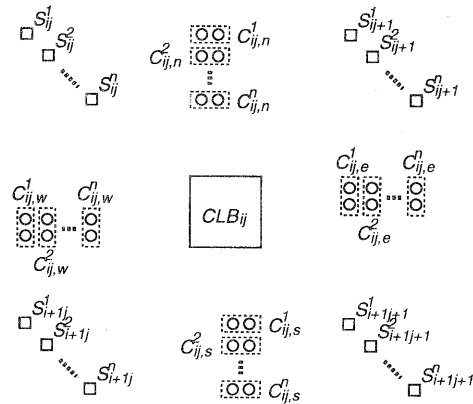


図 6: S ブロックおよび C ブロック内のラベル付け

てのラベルを示しており、図6では CLB_{ij} の周辺にあるSブロックおよびCブロック内のスイッチについてのラベルを示している。

ここで、 $S_{ij}^k, C_{ij}^k, CLB_{ij}$ に関するメモリセルをそれぞれ、 $M(S_{ij}^k), M(C_{ij}^k), M(CL B_{ij})$ と表記する(ただし、 $i = 1 \sim N, j = 1 \sim N, k = 1 \sim n$)。また、マッピングにより生成された $S_{ij}^k, C_{ij}^k, CLB_{ij}$ に関する構成データをそれぞれ、 $D(S_{ij}^k), D(C_{ij}^k), D(CL B_{ij})$ と表記する。構成データ $D(S_{ij}^k), D(C_{ij}^k), D(CL B_{ij})$ は、メモリセル $M(S_{ij}^k), M(C_{ij}^k), M(CL B_{ij})$ にそれぞれ格納される。また、 C_{ij}^k に対して次の関係があるものとする。

$$C_{ij}^k = C_{ij,n}^k C_{ij,e}^k C_{ij,s}^k C_{ij,w}^k$$

CLB, Sブロック, Cブロックに関するメモリセルの接続とそれらのメモリセルに書き込まれている構成データの関係を、図7, 8, 9にそれぞれ示す。

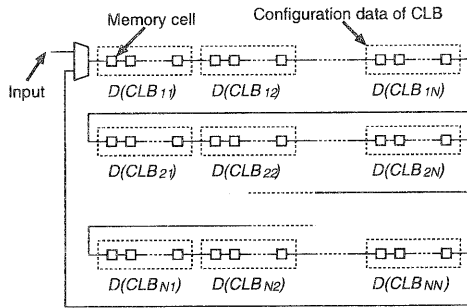


図 7: CLB に関するメモリセルの接続

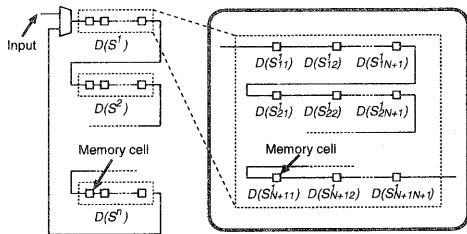


図 8: S ブロックに関するメモリセルの接続

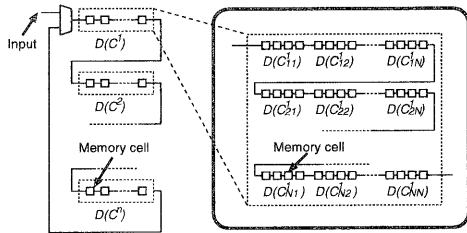


図 9: C ブロックに関するメモリセルの接続

3.3 配線領域に対する欠陥救済

配線領域に対する欠陥救済方法は、欠陥が特定された箇所によって次の 2 つの場合に分けられる。

- 配線または S ブロック内のスイッチに欠陥があると特定された時
- C ブロック内のスイッチに欠陥があると特定された時

以下では、これらの 2 つの場合について、具体的な例を挙げて配線領域の欠陥救済について説明する。マッピングの結果、図 10 に示してあるような回路が構成されるとする。ここでは、点線で示されている CLB および部分配線領域をスペアとしてあらかじめ確保しておくものとする。

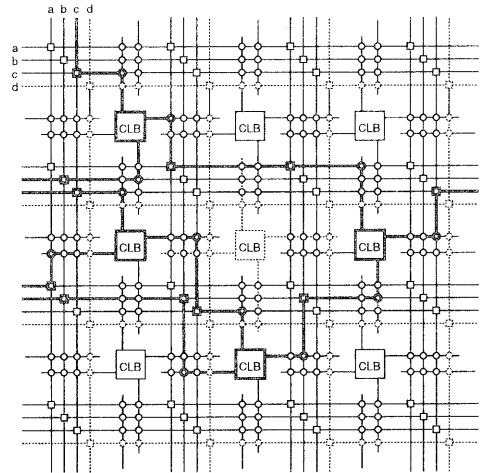


図 10: 回路の構成例

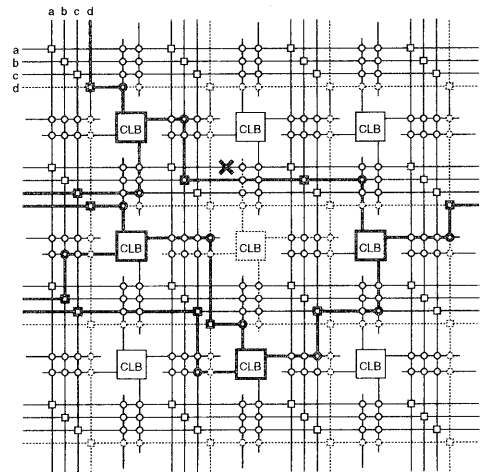


図 11: 配線または S 内のスイッチに欠陥があると特定された時の欠陥救済

[配線または S ブロック内のスイッチに欠陥があると特定された時]

この場合は、CLB に関する構成データはそのままにしておき、配線領域に関する構成データのみをシフトする。スペアの部分配線領域に関する構成データを欠陥が特定された配線またはスイッチを含んでいる部分配線領域に関するメモリセルにシフトすることにより欠陥救済を行う。スペアの部分配線領域に関する構成データのシフトに伴い、スペアではない部分配線領域に関する構成データもシフトされる。

例えば、図 11 の X 印の場所に欠陥があると特定された時には、図 12 に示されているように、S ブロックおよび C ブロックに関する構成データをシフトする。ただし、CLB に関する構成データはそのままである。

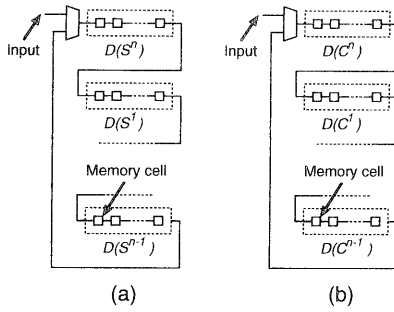


図 12 : SブロックおよびCブロックに関する構成データのシフト (a : Sブロック, b : Cブロック)

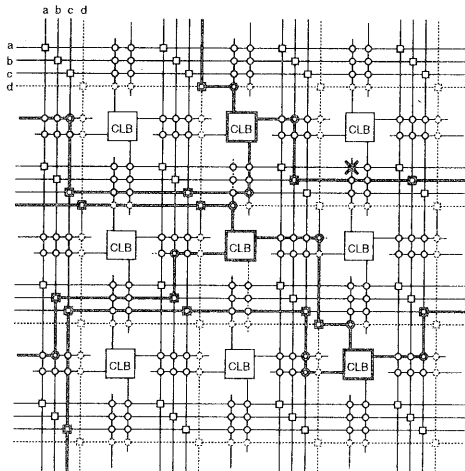


図 13 : Cブロック内のスイッチに欠陥があると特定された時の欠陥救済

[Cブロック内のスイッチに欠陥があると特定された時]

この場合は、CLBに関する構成データおよび配線領域に関する構成データをシフトする。スペアのCLBに関する構成データを欠陥があると特定されたスイッチを含んでいるCブロックと関連のあるCLBに関するメモリセルにシフトする。また、スペアの部分配線領域に関する構成データを欠陥があると特定されたスイッチを含んでいる部分配線領域に関するメモリセルにシフトすることにより欠陥救済を行う。スペアのCLBおよび部分配線領域に関する構成データのシフトに伴い、スペアでないCLBおよび部分配線領域に関する構成データもそれぞれシフトされる。

例えば、図 13 の×印の場所に欠陥があると特定されたならば、図 14, 15, 16 に示されているように、CLBに関する構成データおよび配線領域に関する構成データをシフトする。

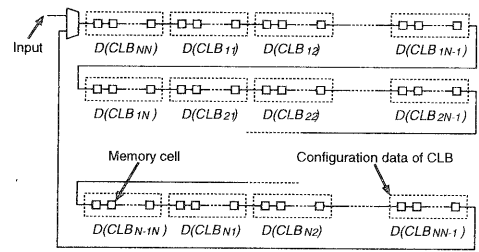


図 14 : CLBに関する構成データのシフト

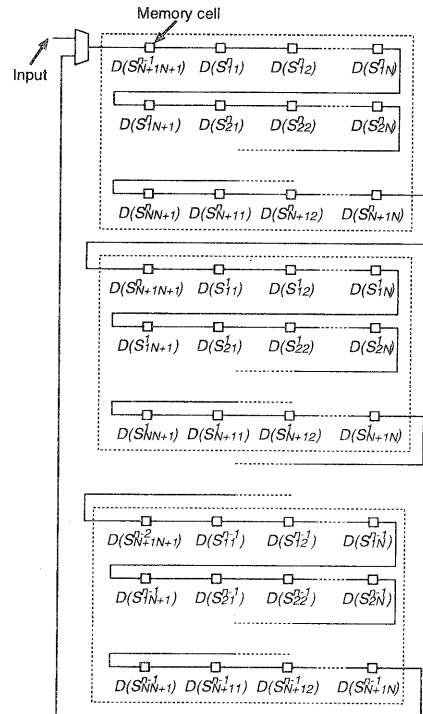


図 15 : Sブロックに関する構成データのシフト

4 評価

この節では、提案手法の評価項目として、使用可能なCLBおよび配線領域の割合とチップの歩留りを考え、それらを評価した結果から考察を行う。

4.1 使用可能なCLBおよび配線領域の割合

提案手法を適用したFPGA内の使用可能なCLBの割合を、CLBの数が 10×10 , 24×24 , 40×40 , 56×56 の場合について評価した結果を表1に示す。

また、提案手法を適用したFPGA内の使用可能な配線領域の割合を、部分配線領域の数が6,8,10の場合について評価した結果を表2に示す。

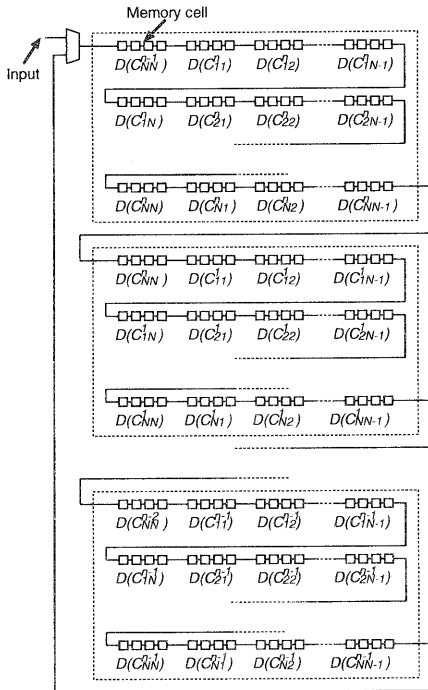


図 16：Cブロックに関する構成データのシフト

4.2 歩留り

チップの歩留りとは、製造されたチップの何%が良品を示す割合のことである。

ここでは、欠陥密度の値を $0 \sim 3.0$ 個/cm²程度をパラメータとして与えて評価する。ただし、チップのトランジスタ集積度を $7M$ (個/cm²)(設計ルール $0.25\mu m$)、システムゲートあたりのトランジスタ数(ユーザが使用可能なゲート数)を U_g 、1システムゲートあたりのトランジスタ数を 100 とし、FPGA の配線領域が 70% であると仮定した。これらの仮定から、チップ面積は以下の式によって算出される。

$$\begin{aligned}
 S &= \frac{100 \times U_g}{7 \times 10^5} \cdot \frac{10}{3} \\
 &= \frac{U_g}{21} \times 10^{-3}
 \end{aligned}$$

システムゲート数については文献 [11] を参考にした。CLB の数から算出されたチップ面積を表 3 示す。

以上の仮定のもとに、提案手法を適用した FPGA チップの歩留りを評価した。論理ブロックの数が 10×10 、 24×24 、 40×40 である FPGA チップに対し、提案手法を適用した場合の歩留りを評価した結果を図 17～図 19 に示す。ただし、部分配線領域の数は $n = 8$ として評価を行った。

表 1：使用可能な CLB の割合 (単位：%)

CLB の数	使用可能な CLB の割合
10x10	60.0
24x24	75.5
40x40	81.2
56x56	82.7

表 2：使用可能な配線領域の割合 (単位：%)

部分配線領域の数	使用可能な配線領域の割合
6	83.3
8	87.5
10	90.0

表 3：システムゲート数とチップ面積

CLB の数	システムゲート数	チップ面積 [cm ²]
10x10	5,000	0.24
24x24	30,000	1.43
40x40	80,000	3.81
56x56	180,000	8.57

4.3 考察

使用可能な CLB の割合については、表 1 に示されている値は十分なものではないと考えられる。また、配線領域についても $1/n$ が使用不可能であり、これらの要素がマッピングの制約を厳しくすると考えられる。

図 17～図 19 は、部分配線領域の数を $n = 8$ とした場合の歩留りの評価結果を示している。欠陥密度が 1.0 のときは、CLB の数が 10×10 、 24×24 、 40×40 の場合の歩留りの向上はそれぞれ、 15% 、 40% 、 20% 程度となった。また、欠陥密度が 2.0 のときは、CLB の数が 10×10 、 24×24 、 40×40 の場合の歩留りの向上はそれぞれ、 25% 、 30% 、 5% 程度となった。これらの評価結果から、提案手法は比較的小・中規模 (CLB の数が 10×10 や 24×24) の FPGA チップに対して、大きな歩留り向上を実現すると考えられる。

5 おわりに

本稿では、配線領域に対する欠陥救済手法を提案した。提案手法の評価より、最大で約 40% の歩留りの向上を確認した。また、提案手法は構成データのシフトにより欠陥救済を行うので、再マッピングを行う場合と比較して欠陥救済に要する時間を短縮することができる。しかし、使用可能な CLB や配線領域の割合が低いという問題がある。

本稿では、FPGA の配線領域に対する欠陥救済について考察したが、他の構成要素 (例えば CLB やメモリセル) に対する欠陥救済も重要である。今後の課題としては、これらの構成要素を含めた FPGA 全体の欠陥救済手法を考案することが挙げられる。

参考文献

- [1] J. Rose, S.D. Brown, R.J. Francis and Z.G.Vranesic, "Field-Programmable Gate Arrays," Kluwer Academic Publishers, 1992.
- [2] 西田 健次編, "FPGA-その現状, 将来とインパクト," 情報処理, vol. 35, no. 6, pp. 504-539, Feb. 1994.
- [3] 三宅 常之, "-Nets 特集- FPGA の書き込み手法に新潮流," NIKKEI ELECTRONICS, pp. 204-213, Feb. 1999.
- [4] 末吉 敏則, 稲吉 宏明編, "特集: やわらかいハードウェア," 情報処理, vol. 40, no. 8, pp. 777-782, Aug. 1999.
- [5] 南谷 崇, "フォールトトレラントコンピュータ," オーム社, 1991.
- [6] W. H.Mangione-Smith, J. Lach, and M. Potkonjak, "Low Overhead Fault-Tolerant FPGA Systems," *IEEE Trans. Comput.*, vol. 6, no. 2, pp. 212-221, June 1998.
- [7] 森永 聡, "pバージョン・コンフィギュレーションの信頼度最大化," 信学論, vol. VLD98-69, pp. 185-192, Sept. 1998.
- [8] S. Dutt and F. Hanchek, "REMOD: A New Methodology for Designing Fault-Tolerant Arithmetic Circuits," *IEEE Trans. VLSI*, vol. 5, no. 1, pp. 34-56, Mar. 1997.
- [9] A. Doumar, S. Kanako, H. Ito, "Defect and Fault Tolerance FPGAs by shifting the Configuration," *The IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems*, pp. 378-385, Nov. 1999.
- [10] 金子 智, Abderrahim Doumar, 伊藤秀男, "構成データのシフトによる欠陥救済容易な FPGA," 1999 年電子情報通信学会情報・システムソサイエティ大会講演論文集, p. 83, Sept. 1999.
- [11] XILINX 株式会社, "XC4000 シリーズフィールド・プログラマブルゲートアレイ," Jun. 1998.

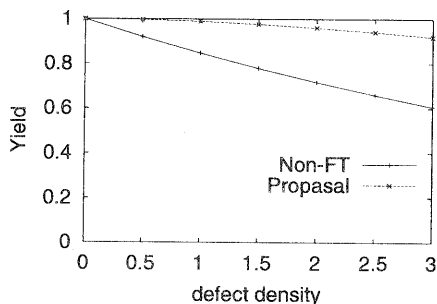


図 17: CLB の数が 10×10 の場合の歩留り

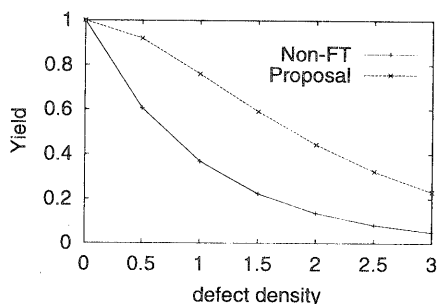


図 18: CLB の数が 24×24 の場合の歩留り

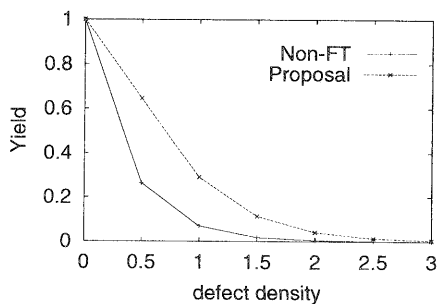


図 19: CLB の数が 40×40 の場合の歩留り