

自己修復機能を有するフォールトトレラント LSIアーキテクチャの検討

Fault-Tolerant LSI Architecture with Self-Repair Function

○阿部茂樹[†], 青木孝文[‡], 樋口龍雄[‡], 鹿股昭雄[§]

○Shigeki Abe[†], Takafumi Aoki[‡], Tatsuo Higuchi[‡], Akio Kanomata[‡]

[†]東北大学工学部, [‡]東北大学大学院情報科学研究科,
[§]仙台電波工業高等専門学校

[†]Faculty of Engineering, Tohoku University

[‡]Graduate School of Information Sciences, Tohoku University

[§]Sendai National College of Technology

キーワード : フォールトトレランス (Fault Tolerance), FPGA(Field-Programmable Gate Array), 再構成型ハードウェア (Configurable Hardware), VLSI (Very Large Scale Integration), 自己修復 (Self Repair)

連絡先 : 〒 980-8579 仙台市青葉区荒巻字青葉 05 東北大学大学院情報科学研究科 樋口研究室
阿部茂樹, Tel.: (022)217-7169, Fax.: (022)263-9406, E-mail: abe@higuchi.ecei.tohoku.ac.jp

1. はじめに

ディープサブミクロンの集積技術の実用化に伴い, 単一の半導体チップに1000万ゲート以上が集積できるようになり, ささまざまな分野で高速・高性能な特殊用途向けのVLSIが開発されるようになってきた. このようなVLSIは, 歩留まりの向上に加えて, 用途によっては, 高いアベイラビリティが要求される. そのため, 開発されたVLSIの初期段階での検査や, 故障が検出された場合の対応が重要となる. しかしながら, 初期段階での故障検査では, すべての回路の機能を診断するために膨大な数のテストパターンを用意しなければならず, 限られた時間内での完全な検査は困難である. また, 稼働中のVLSIに故障が発生した場合, 迅速に対処して稼働率を向上させるために, 故障

箇所をすばやく検出し, 故障部分を交換する必要があるが, 人間の手が介入できないような環境で応用される場合には, 故障が検出されても交換することができないという問題がある.

一方, 近年EDA (Electronics Design Automation) 技術の発展に伴い, ハードウェア記述言語 (HDL: Hardware Description Language) と論理合成によるハイレベル設計手法への移行が進み, 大規模で複雑な集積回路が容易に設計できるようになってきた¹⁾. それに加えて, 基本的な論理構造がすでにチップ上に作成され, 設計者が自由にプログラムできる大規模なFPGA (Field-Programmable Gate Array) などのデバイスが利用されるようになってきた^{2),3)}. FPGAは, アレイ状に配置されたプログラム可能な論理ブロックCLB (Configurable

Logic Block) と CLB 間を接続するプログラム可能な配線領域から構成されている⁴⁾⁻⁵⁾。

本稿では、FPGA が再構成可能であることに着目し、故障が検出された場合でも同じハードウェア上で故障箇所を回避してシステムを再構成することにより正常な機能を回復できる新しい FPGA アーキテクチャを提案する。提案するアーキテクチャでは、CLB や CLB 間配線などのハードウェアリソースの徹底的な 2 重化によって、動作中に故障検出が可能である。さらに、故障箇所を特定し、これを回避するように回路の再構成を行い、正常な機能を回復することが可能である。この高信頼 FPGA は、システム動作中に発生する一時故障や永久故障に対してロバストなシステムを構築する際のキーデバイスとして利用可能であり、人手が介入できないようなクリティカルな応用において有用であると考えられる。

2. 基本概念

FPGA は、構成情報を変更することにより、さまざまな回路をマッピングできるプログラマブルな IC である。この構成可能 (Configurable) な性質を利用して、たとえシステムにフォールトが存在しても、回路構造の再構成により正常な機能を回復できるようなロバストなシステムを構築することが可能であると考えられる。すなわち、システムを構成する FPGA に十分なハードウェアの冗長性が備わっている場合、もし故障位置の特定が可能であれば、その故障箇所を使用せずに回路を FPGA 上に再構成することが可能である。

上記のような再構成型デバイスによる高信頼システムを構築する場合、最も重要な問題は、いかにしてハードウェアリソース中の故障箇所を特定するかという点である。従来の FPGA をそのまま利用する場合、故障検出が極めて難しく、実用化への決め手を欠いている状況であった。これに対して、本稿では、システムの動作中に迅速な故

障検出が可能となる新しい FPGA アーキテクチャを提案し、これを用いた高信頼システムの構築手法を示す。以下では、ハードウェアの再構成が容易な SRAM 方式の FPGA アーキテクチャを想定する。

本システムで対象とする故障は、以下のとおりである。

- 1) stuck-at 故障, ブリッジ故障, stuck-open 故障など製造時や動作時に発生する永久故障。
- 2) 電磁ノイズ, α 粒子の入射, 電源変動などの厳しい環境条件に起因する一時故障。

SRAM 型の FPGA は、マッピングする回路の構成情報を SRAM 中に保持する。このため、上記 2 のような一時的な故障によって SRAM が保持する構成情報が破壊された場合、システムの動作中は、1 に示したハードウェアの永久故障と区別することが不可能である。このため、故障検出の段階では SRAM データの破壊に起因する一時故障も永久故障と同様に取り扱う必要がある。ただし、故障回復の段階では、SRAM のデータ破壊に起因する一時故障は、構成情報の再ロードにより回復が可能であるのに対して、永久故障に関してはハードウェアの再構成のみによって回復が可能になる。本稿で提案する高信頼 FPGA アーキテクチャ (図 1) では、上記のような 2 種類の故障をシステム動作中に実時間検出することを目的として、ハードウェアリソースの徹底的な 2 重化を図る。高信頼 FPGA 内部では、CLB のみならずプログラマブル配線についてもすべての経路で 2 重化が施され、論理演算と信号伝達の双方に関して、常に 2 重化されたデータが比較される。この比較は、FPGA 内部に配置された多数の比較器を用いて実行される。さらに、比較結果は比較器と同数存在するエラーレジスタに、ある一定の周期で取り込まれる。このエラーレジスタの保持する値 (誤りが存在する場合 1, 存在しない場合 0) は、多入力論理和回路 (ワイヤード OR など) により一括集計され、

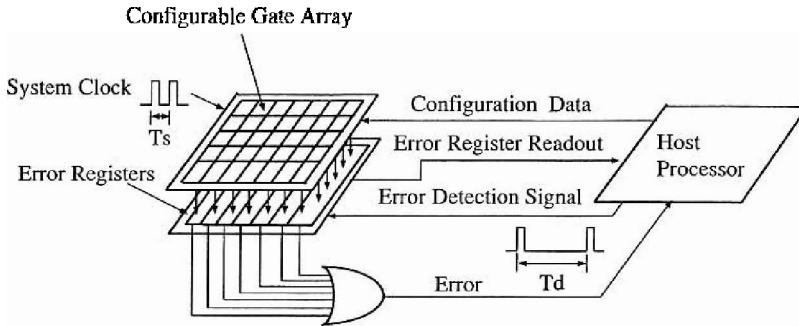


Fig. 1 システムの概念図

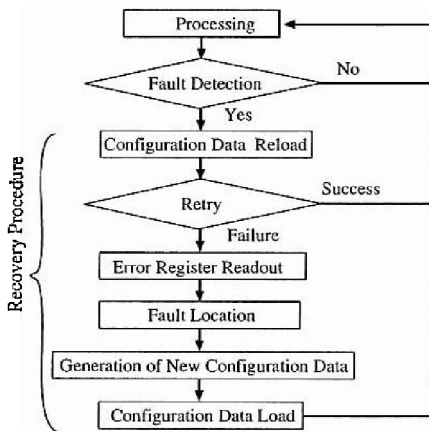


Fig. 2 故障検出および回復のフローチャート

誤りの有無のみが検出される。ここで、FPGAにマッピングしたシステムのクック周期を T_s とし、誤り検出のためにエラーレジスタの値が更新される周期を T_d とする。 T_d は、すべてのエラーレジスタの値を論理和で集計する操作にかかる時間よりも大きい必要があるため、現実的にはシステムクロック周期よりも長いと仮定する必要がある($T_s \leq T_d$)。

提案する高信頼FPGAアーキテクチャにおける故障検出および回復のためのフローチャートを図2に示す。この故障検出・回復の手順は、外部のホストプロセッサ(汎用マイクロプロセッサ等)によって制御されるものとする。本アーキテクチャの特長は、上で述べた方法により、基本的にシス

テム動作中にオンライン(周期 T_d)で誤り検出がなされることであり、誤りが検出された場合、図2に示すような回復のフェーズに入る。この回復操作は、システムの動作を停止させオフラインで実行される。システムの停止が許容されない場合は、より上位のアーキテクチャにおいて、待機FPGAモジュールや各種の冗長化技法を利用して対処することが可能である。

ホストプロセッサは、誤りが検出されたFPGAのシステムクロックおよび誤り検出クロックを停止させ、特定された誤りが一時故障に起因する可能性を考慮して、FPGAの構成情報を再ロードし、システムを再起動する。これによって正常な機能が回復しない場合は、ハードウェアの永久故障が発生していると判断する。永久故障と判断された場合には、直前に保持しているエラーレジスタのデータを外部へ読み出す。このエラーレジスタのデータから、誤り発生箇所の特定を行う。この場合、ホストプロセッサは、FPGAの故障箇所を使用せずに迂回する新たな構成情報を生成し、これをFPGAにダウンロードし、システムの再起動を行う。

3. 自己修復可能なFPGAアーキテクチャ

本章では、上で述べたような機構を組み込んだSRAM型FPGAのアーキテクチャについて述

べる。今回は、下記のような設計方針を採用した。

- 1) FPGA の基本アーキテクチャとしては、XILINX 社の XC4000 シリーズを参考にした。
- 2) 故障の検出のために、すべての論理ブロック CLB(Configurable Logic Block) を 2 重化するとともに、プログラマブル配線についてもすべての経路について 2 重化を施す。
- 3) 各 CLB(Configurable Logic Block) の入出力に比較器を配置し、2 重化されたすべての入力および出力信号の誤りをオンラインで検出可能とする。これにより CLB 内部の故障ならびに CLB 間を接続する配線経路上の故障が検出できる。
- 4) すべての比較器について、誤り検出結果を周期 T_d でサンプリング保持するエラーレジスタならびに外部への読み出し回路を用意する。
- 5) エラーレジスタに格納された誤り信号の有無のみをオンラインで検出するための誤り検出回路(多入力論理和回路)を備える。
- 6) 誤り信号の存在が検出された場合に、すべてのエラーレジスタの内容をホストプロセッサに読み込んで故障箇所の特定を行うために、エラーレジスタの読み出し機構を備える。
- 7) CLB の故障が検出された場合は、予備 CLB を用いて機能回復を行う。一方、CLB 同士を接続する配線経路上の故障については、経路に含まれるすべての配線を予備配線によって代替するものとする。

高信頼 FPGA の構成を図 3 に示す。CLB を 2 重化した DCLB (Dual CLB) とその間を 2 重化配線でプログラマブルに接続するための DSB (Dual Switch Block), DCB (Dual Connection Block), 構成情報を書き込むためのコンフィギュレーション

レジスタ、故障検出結果を保存するエラーレジスタ (ER) および故障の有無を検出するための多入力 OR 回路から構成される。また再構成する場合、配線の競合を考慮することなく新しい構成情報を容易に生成するための I/O ルータ (IOR) を備えている。

7 章で述べるテストチップの例では、 3×3 の DCLB マトリックスをチップ上に集積化している。この FPGA の構成情報は縦 54 個 \times 横 88 個のマトリックス状に配置された合計 4,752 ビットの SRAM に格納される。構成情報を入力する際には、ホストプロセッサの制御により、上端に配置された 88 ビットのコンフィギュレーションレジスタにデータが直列に入力された後、SRAM マトリックス内部に並列に書き込まれる。これをコンフィギュレーションレジスタマトリックスの縦の行数である 54 回繰り返すことによりすべての構成情報が書き込まれる。この構成情報は、FPGA 内の DCLB の論理機能や DSB, DCB のスイッチの設定を定義するものであり、ユーザは構成情報を書き換えることによって FPGA の機能をプログラムすることができる。

以下に、高信頼 FPGA の主要構成要素およびその機能について説明する。

DCLB (Dual Configurable Logic Block) : 図 4 に示すように論理ブロックは CLB1 と CLB2 で 2 重化し、故障検出を可能にするために論理ブロックの入出力側にエラーディテクタ ED を配置している。1 つの CLB は、3 個の LUT (LookUp Table) と 2 個のフリップフロップおよびマルチプレクサから構成され、最大 9 変数関数を構成することができる⁶⁾。また、DCLB 間を接続するための配線もすべて 2 重化され、配線経路上の故障の検出が可能になっている。

DSB (Dual Switch Block) : 配線経路を 2 重化するために、図 5 に示すように SB1 と SB2 の 2 重化によって構成される。今回のテストチップでは冗長な配線を含めて 20×20 本の信号線をスイッチ

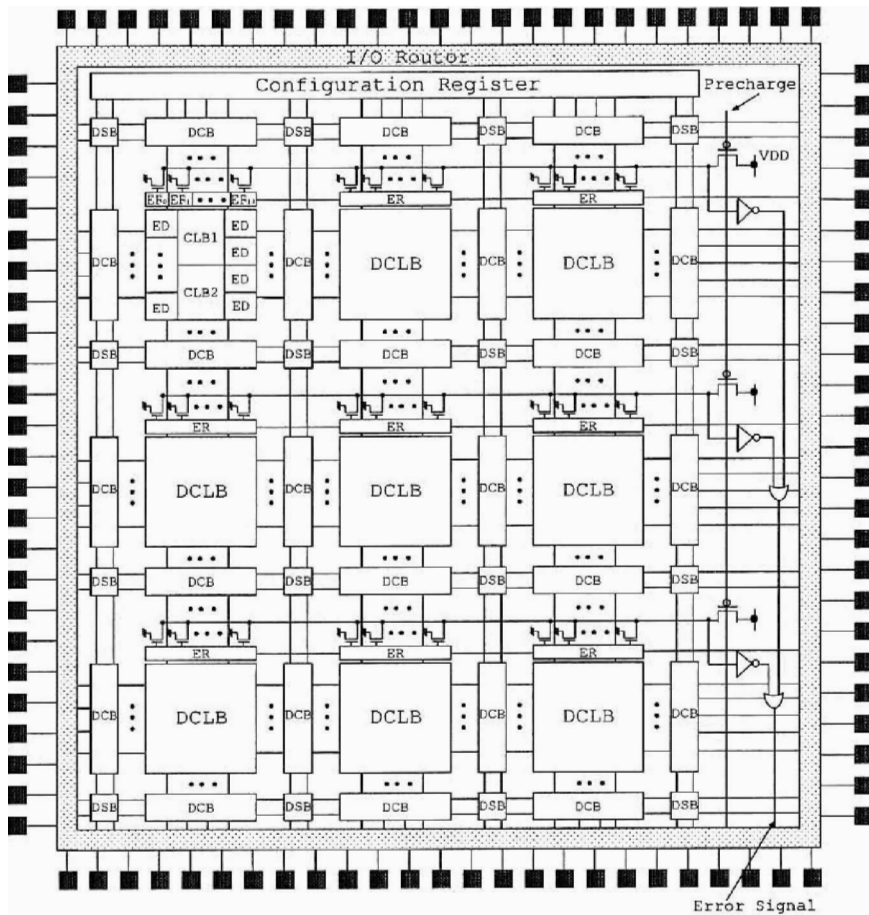


Fig. 3 高信頼FPGAの構成(3×3論理ブロック配列の例)

することが可能である。DSB内部の信号線のクロスポイントには6トランジスタスイッチが配置され、ユーザはSRAMに構成情報を書き込むことで配線のルーティングをプログラムできる。

DCB (Dual Connection Block): 図6に示すようにCB1とCB2で2重化され、DSBに接続する横の配線(今回の試作では20本)に対し、上下のDCLBからの入出力線数だけ交差している。横型のDCBに関しても同様である。

ED (Error Detector): 2重化された論理ブロックおよび配線における誤りを検出するために、DCLBの入力および出力には誤り検出回路EDが配置さ

れている(図4)。ここで、入力側の誤り検出回路は、配線経路上の故障を検出し、出力側の誤り検出回路はDCLB内部の故障を検出する。この誤り検出回路EDの構成を図7に示す。2重化された入力信号(x, x')が一致する場合、この回路はそのまま2重化信号(x, x')を出力する。一方、 x と x' が一致しない場合は、対応するエラーレジスタへ1を出力する。

ER (Error Register): エラーディテクタからの出力を格納し、そのデータは故障の有無を検出するためおよび故障箇所を特定するときに使用される。読み出し回路については、故障が検出された

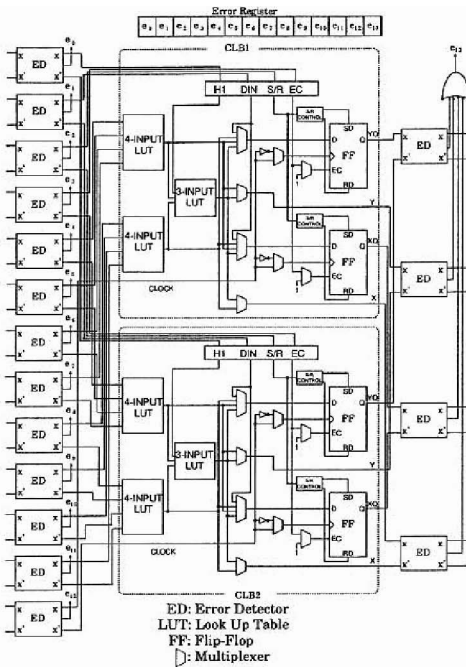


Fig. 4 Dual Configurable Logic Block (DCLB) の構成

場合に故障箇所を特定するためすべてのエラーレジスタの内容をホストプロセッサに取り込むために必要となる。

多入力 OR 回路:多入力 OR 回路は、すべてのエラーレジスタの論理和をとり故障の有無をホストプロセッサに知らせるために備えている。

IOR (I/O Router):入力パッドを仮想的に移動することにより再構成が容易になるため、入力パッドとマトリクスを自由に接続可能とするために備える。

4. 故障の検出

提案するアーキテクチャでは、FPGA 内部の論理・配線ブロック単位での細粒度冗長構成を採用しているため、故障箇所の特定は論理・配線ブロック単位で行うことができ、故障状態からの回復も故障箇所を避けるように構成情報を変更することにより、FPGA 自体を交換することなく実行する

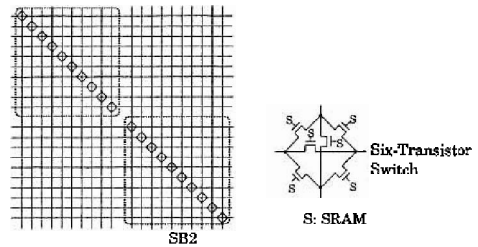


Fig. 5 Dual Switch Block (DSB) の構成

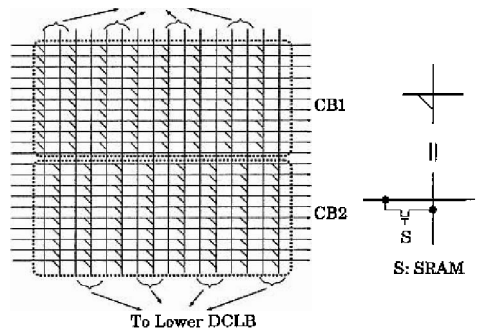


Fig. 6 Dual Connection Block (DCB) の構成

ことができる。このようなシステムは、FPGA やプリント基板そのものを実装後に交換することが困難なミッションクリティカルな応用（例えば宇宙技術など）において有効であると考えられる。

故障の検出は、DCLB の入出力に配置された 14 個の誤り検出回路 (ED) によって行われる。このうち入力側の誤り検出回路は、前段の DCLB から接続される配線経路上の故障に起因する誤りを検出する。また、出力側の誤り検出回路は、DCLB 内部の故障に起因する誤りを検出することができ、4 個の検出結果は論理和によりまとめられてエラーレジスタへ送られる。これらの故障検出の結果は、適切な周期 T_d でエラーレジスタに並列に取り込まれる。さらに、エラーレジスタの全データを論理和で結合し、誤り警告信号を生成する。

図 2 に示したように、誤りが発見された場合、回復処理のフェーズに入る。まず、第 1 段階は、この誤りが一時故障によるものであると仮定して、

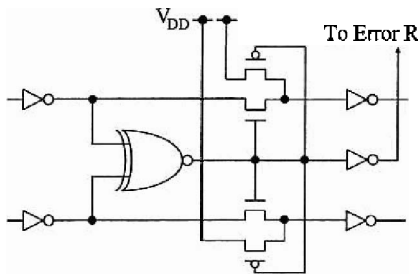


Fig. 7 Error Detector (ED) の構成

一時故障からの回復を図る。FPGA においては、信号線にのるノイズなどによって、最悪の場合、RAM に格納されている構成情報が書き換わる可能性がある。これに対処するために、まず、同じ構成情報をホストプロセッサから再ロードするとともに、直前の処理を再度実行する。これによって、同じ箇所でも誤りが検出されたとき、あるいは、システム動作中に同一箇所でも間欠的に誤りが再発するときには、ハードウェアの永久故障とみなし、故障箇所を使わないようにシステムの再構成を行う。提案する FPGA では、故障箇所が特定できるため、その箇所を回避するように構成情報を書き換え、同等機能のシステムを再構成する。

5. システム再構成

提案する FPGA を用いたシステムの再構成の手法について述べる。まず、上記のような再発する誤りが発見された場合には、すべてのエラーレジスタのデータをホストプロセッサ内に取り込み、その情報に基づいて故障箇所が特定される。この故障箇所の特定は、前章の最後で述べた誤り検出回路の特性により、容易に自動化することができる。この結果、検出される故障は、論理ブロック DCLB における故障と DCLB 間を接続する配線系路上の故障の 2 種類に分類できる。

ここでは、まず、後者の DCLB 間の配線経路に故障が発生した場合の回復例を図 8 を用いて説明する。例えば、DCLB11 の出側の故障検出回路

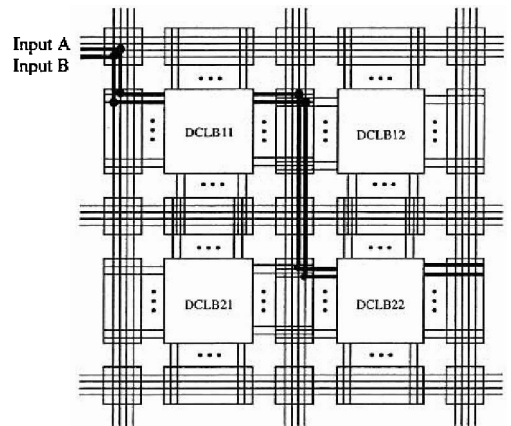
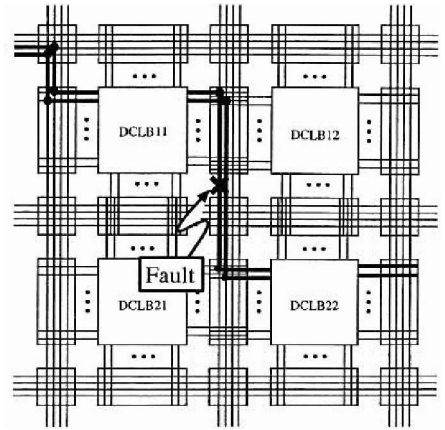


Fig. 8 配線故障からの回復 (局所的再構成)

ED において誤りが検出されず、DCLB22 の入力側の ED で誤りが検出された場合、その間を結ぶ配線経路上のいずれかの地点で故障が発生したと判断される。そこで、この経路上のすべての配線を CB と SB のクロスポイントの設定を変更することにより、予備に用意してある配線に切り替える。これを実現するために、設計初期の段階で FPGA へ回路をマッピングする際には、故障確率に応じた十分な数の予備配線を残すように設計を行う必要がある。このように予備配線が使用可能な状況では、「局所的な再構成」によって故障からの回復が可能である。また、局所的な再構成のための構成情報は、ホストプロセッサが保持するオリジナ

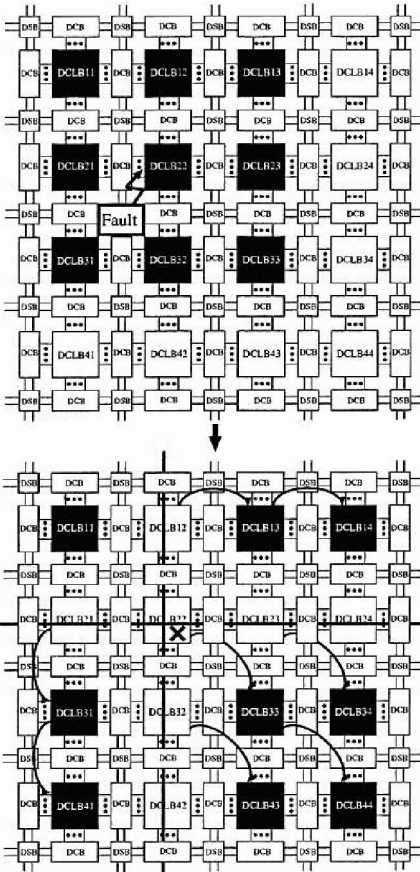


Fig. 9 論理ブロックの故障からの回復（局所的再構成）

ルの構成情報を元に容易に自動生成することができる。

次に、DCLB 内部に故障が検出された場合の回復法について、図 9 を例に説明する。この場合、黒の DCLB が使用中の DCLB であり、DCLB22 に故障が特定された状況を想定している。上で述べた配線経路の再構成の場合と同様に、FPGA 内部に十分な数の予備 DCLB が存在する場合は、図 9 下段の矢印で示すように、故障が検出された DCLB の行と列に含まれるすべての DCLB を使用しないように再構成することにより回復が可能である。このように十分な数の予備 DCLB が使用可能な状況では、「局所的な再構成」によって故障からの回

復が可能である。また、局所的な再構成のための構成情報は、ホストプロセッサが保持するオリジナルの構成情報を元に容易に自動生成することができる。例えば DCLB22 に故障が検出された場合の局所的再構成では、DCLB および DCB については故障した DCLB と同じ行列上にある DCLB を回避するように構成情報をシフトさせることで新しい構成情報が生成できる。

一方、SB に関する再構成については、図 10 に示すような 4 つのブロックに分けて考える。まず、1 つのブロック内から外部に出ない配線について考える。例えば、図の DCLB22 から DCLB33 に配線されている場合、再構成後の配線は DCLB33 から DCLB44 の同一ブロック内での接続となり、 DSB_{α} の構成情報も $DSB_{\alpha'}$ にそのままシフトされる。すなわち、同じブロック内の配線に関する DSB の構成情報は元の情報をそのままシフトして利用できる。次に各ブロック間に跨って接続される配線を考える。ブロック A とブロック B に跨っている配線では、ブロック A に対しブロック B は横方向だけにシフトしたブロックである。例えば、DCLB11 から DCLB12 への配線が再構成後は DCLB11 から DCLB13 への配線となる。この場合、 DSB_{β} の構成情報はそのまま $DSB_{\beta'}$ にシフトし、 DSB_{β} は横方向に延長するように構成情報を設定すればよいことになる。同様な方法により、ブロック A とブロック C の場合は縦方向に配線を延長し、ブロック A とブロック D は縦と横方向に延長するように構成情報を書き換えることで DSB の構成情報が生成できる。すなわち、局所的再構成による構成情報の生成はルールベースのアルゴリズムで容易に生成することができる。

以上で述べたように、十分な配線や DCLB を予備のハードウェアリソースとして使用可能な状況では、システムの再構成による故障状態からの回復は、比較的容易である。さらに、この再構成手法によって得られた回路においては、再構成に起因する付加的な配線遅延の見積もり容易である

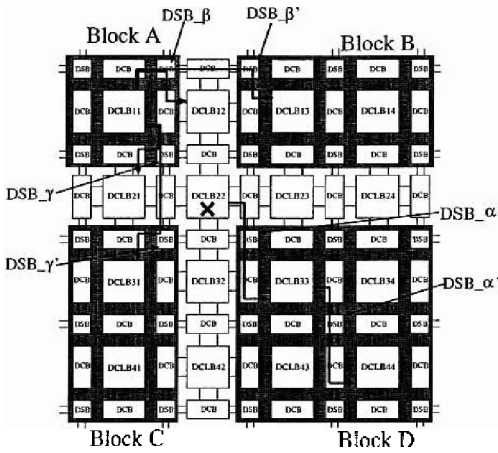


Fig. 10 SBの局所的再構成

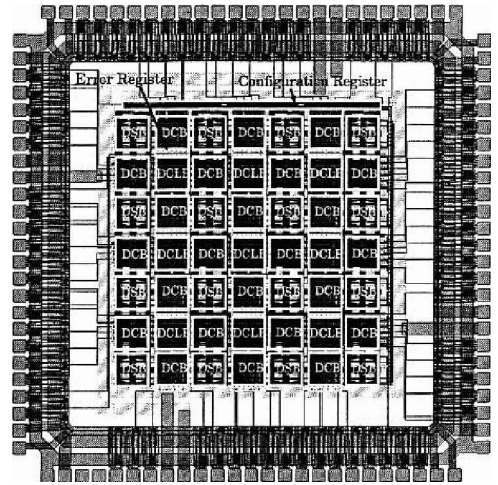


Fig. 11 レイアウト

ため、これをあらかじめ考慮して初期設計を行うことにより、再構成時のタイミング解析を不要にすることができると考えられる。これは、人間の指令を介しないで、ホストプロセッサの自律的な制御で、故障検出・回復を実行するようなシステムを構築する際に有用な性質である。これに対して、十分なハードウェアリソースを予備として使用できない場合は、回路全体の配置配線 (Place and Route) およびタイミング解析のやり直しを行って「大域的な再構成」を行う必要があり、このような手法の確立は今後の研究課題として残されている。

6. テストチップの試作および評価

今回の試作では、チップ面積の制約により故障の有無を検出する多入力論理和回路およびI/Oルータ (IOR) は集積化しないものとした。

試作した高信頼FPGAテストチップのレイアウトを図11に示す。このFPGAはVDEC (東京大学大規模集積システム設計教育研究センター) を通して試作し、その仕様は表1に示す通りである。0.6 μm CMOS 3層配線技術により、3 \times 3のDCLBを52,936個のトランジスタで集積化している。チップ面積は2.2mm \times 2.3mmとなった。2重化しない

Table 1 試作チップの仕様

Input Data	12 bits
Output Data	39 bits
Configuration Data	54 \times 88 bits
Error Register	117 bits
Chip Organization	9 DCLBs, 24 DCBs, 16 DSBs, 117 EDs
Supply Voltage	5V
Active Area Size	2.2mm \times 2.3mm
Transistor Count	52,936
Process	0.6 μm CMOS 3 metal layers VDEC ROHM

CLB単体の遅延時間は、HSPICEシミュレーションにより、1.96nsとなった。一方、これを2重化し、入出力に誤り検出回路を付加したDCLBの遅延は、2.17nsである。すなわち、故障検出機能を付加することに伴う論理ブロックにおける遅延のオーバーヘッドは、約4%程度であることがわかる。

次に、2重化に伴うFPGA全体のチップ面積の増加について検討する。今回試作した3 \times 3の論理ブロックを集積化した高信頼FPGAの場合、チップ面積が5.06mm² (= 2.2mm \times 2.3mm) であるのに対して、2重化をしない通常のFPGAの場合、2.04mm² (= 1.2mm \times 1.7mm) と予想される。す

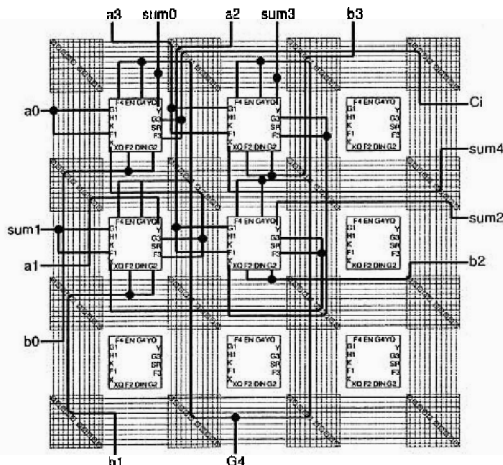


Fig. 12 加算器の配線例

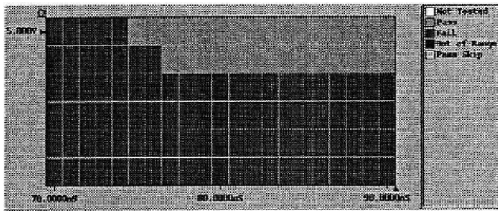


Fig. 13 加算器の速度評価結果

なわち、高信頼化に伴うFPGAの面積の増加は、約2.48倍程度であると評価できる。ただし、今回の試作では、エラーレジスタのデータ集計用の多入力論理和回路を集積化しておらず、DSBやDCBの構成情報を保持するSRAMについても2重化を行わずに面積を抑えている。これらを完全に実装したチップでは、面積の増加の割合が3倍程度であると予想される。

次に、試作したチップを用い4ビット加算器および4ビットシフトレジスタを構成し、速度評価をLSIテスタ(アドバンテスタ社製T6671E)を用いて測定した結果について示す。4ビット加算器は、入力を $a_3a_2a_1a_0$ および $b_3b_2b_1b_0$ 、加算出力を $sum_4 \dots sum_0$ として4つのDCLBでマッピングしており、DCLB内のDFFを通さないで回路を構成した(図12)。速度評価ツール(shmoo2プロ

ット)を用いて加算器の速度を測定した結果を図13に示す。縦軸が電圧、横軸が時間で表され、動作範囲は5Vで約75nsで動作することが確認できた。パッドの入出力時間が約15nsであることから、LUT1段の通過に15ns程度必要となる。同様に、DCLBのDFFのみを4個用いて4ビットシフトレジスタを構成した場合、約25ns程度で動作することも確認している。

7. おわりに

故障の検出および回復が可能なFPGAアーキテクチャについて提案した。今後の課題として、試作しているFPGAの故障注入による動作試験やホストプロセッサとのインタフェースについて検討する必要がある。また、自己修復のための構成情報を自動生成するソフトウェアの開発が重要となる。

謝辞 高信頼FPGAの設計に関して御協力頂いた伊東秀男氏ならびに津田真人氏に感謝致します。なお、本研究におけるチップ試作は、東京大学大規模集積システム設計教育研究センター(VDEC)を通し、ローム(株)、および凸版印刷(株)の協力で行われたものです。

参考文献

- 1) 桜井至:HDL設計入門,テクノプレス,1996.
- 2) S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic: Field-Programmable Gate Array, Kluwer Academic Publishers,1992.
- 3) S. M. Trimberger: Field-Programmable Gate Array Technology, Kluwer Academic Publishers, 1994.
- 4) P. Chow, S. O. Seo, J. Rose, K. Chung, G. Paez-Monzon, I. Rahardja: The Design of an SRAM-Based Field-Programmable Gate Array - PartI: Architecture, IEEE Transactions on VLSI Systems, Vol. 7, No. 2, pp. 191-197, 1999.
- 5) P. Chow, S. O. Seo, J. Rose, K. Chung, G. Paez-Monzon, I. Rahardja: The Design of an SRAM-Based Field-Programmable Gate Array - PartII: Circuit Design and Layout, IEEE Transactions on VLSI Systems, Vol. 7, No. 3, pp. 321-330, 1999.
- 6) Xilinx社: プログラマブルロジックデータブック, 1997.