

# 故障検出・回復可能な FPGA とそのレイアウトモデル評価

○阿部茂樹 青木孝文 (東北大学) 樋口龍雄 (東北工業大学)  
鹿股昭雄 (仙台電波高等専門学校)

## An FPGA with Fault Detection/Recovery Capability and Its Layout-Model Evaluation

\*S. Abe, T. Aoki (Tohoku University) and T. Higuchi (Tohoku Institute of Technology) and A. Kanomata (Sendai National College of Technology)

**Abstract**— Due to the exponential growth of the density and capacity of VLSI technology, it becomes increasingly difficult to achieve higher performance while keeping a high level of system reliability. In order to construct highly reliable systems with next-generation VLSI technology, not only the improvements in fabrication technology, but also active measures are required to deal with temporary/permanent hardware faults in integrated components. This paper investigates the potential of Field-Programmable Gate Arrays (FPGAs) to realize reliable systems with self-repair capability. We propose a new FPGA architecture that can detect hardware faults during operation, and can recover the correct function by reconfiguring its structure.

**Key Words:** Fault Tolerance, FPGA (Field-Programmable Gate Array), Configurable Hardware

### 1 はじめに

集積回路の大規模化・複雑化に伴い、開発された LSI の初期段階での検査や動作中に発生する故障の検出が困難になっている。集積回路に故障が発生した場合、部品またはボードごとに交換するケースがほとんどであるが、人間が介入できない環境で用いられるシステムに対しては、自律的に故障検出および機能回復可能な LSI の開発が重要となる。本稿では、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、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 (Field-Programmable Gate Array) のアーキテクチャを示すとともに試作とその評価について述べる。また、ディープサブミクロンの集積技術の実用化により、提案する FPGA がどの程度の面積で構成できるのかレイアウトモデルを用いて評価した結果について述べる。

### 2 基本構成

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と論理ブロック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 [1]~[4]。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。提案するフォールトトレラント FPGA は、Fig.1 に示すように 2 重化された論理ブロック (DCLB: Dual Configurable Logic Block) と 2 重化されたプログラマブル配線 (DSB: Dual Switch Block, DCB: Dual Connection Block), 構成情報を書き込むコンフィグレーションレジスタから構成される [5]。

DCLB は、それ自身の内部の故障と配線経路上での故障を検出するために、その入出力に故障検出回路 (ED: Error Detector) を備える。また、故障箇所の特定と新しい構成情報を生成するため、すべての ED の検出結果を格納するエラーレジスタ (ER) が用意されている。すべてのエラーレジスタの状態の多入力論理和を計算することにより故障の有無が検出される。

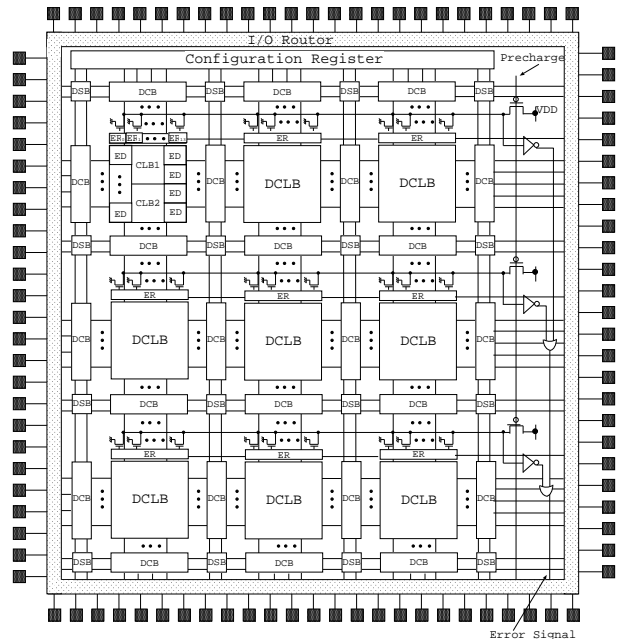


Fig. 1: Structure of a fault-tolerant FPGA

### 3 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。さらに、エラーレジスタの全データを論理和で結合し、誤

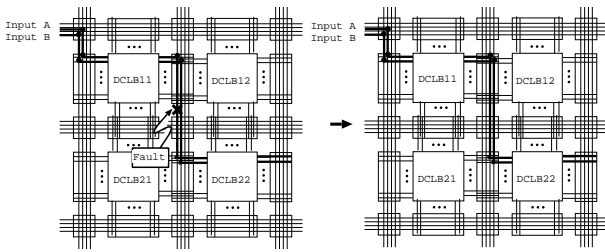


Fig. 2: Recovery from an interconnection fault

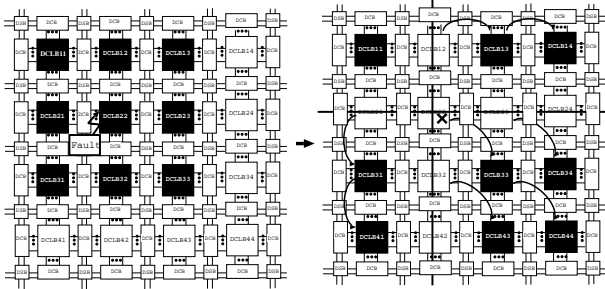


Fig. 3: Recovery from a logic block fault

り警告信号を生成する。

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、同じ箇所ですりぎが検出されたとき、あるいは、システム動作中に同一箇所の間欠的にすりぎ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、FPGA 自体を交換することなく同等機能のシステムを再構成できる。

#### 4 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、すりぎ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。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 Fig. 2 を用いて説明する。例えば、DCLB11 の出側の故障検出回路 ED においてすりぎが検出されず、DCLB22 の入力側の ED ですりぎが検出された場合、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 DCB と DSB のクロスポイントの設定を変更し、予備に用意してある配線に切り替えることにより回復が可能となる。

次に、DCLB 内部に故障が検出された場合の回復法について、Fig. 3 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。FPGA 内部に十分な数の予備 DCLB が存在する場合は、Fig.3 右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

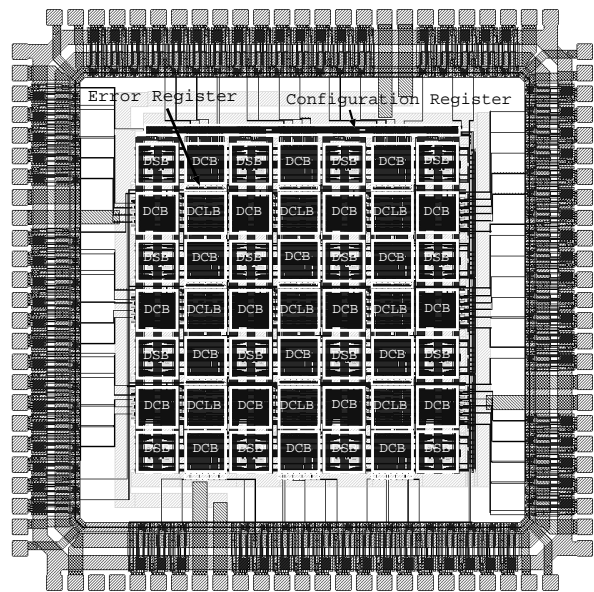


Fig. 4: Layout of the test chip

での 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。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できると考えられる。

#### 5 テストチップおよびその評価

提案する FPGA のレイアウトを Fig.4 に示す。試作した SRAM 型 FPGA は、3×3 の DCLB にチップ上に集積化しており、その仕様を Table.1 に示す。実際に、試作したチップの動作を LSI テスタ (アドバンテスト社製 T6671E) を用いて測定した結果、4 ビット加算器は 5V で約 75ns で動作し、4 ビットシフトレジスタは約 25ns 程度で動作することを確認している。また、DCLB に故障が発生したと想定し、その DCLB を含む行列をすべて使用しない局所的な再構成によって回路を構成すると、4 ビット加算器は約 80ns で動作し、4 ビットシフトレジスタは約 30ns 程度で動作することを確認している。

#### 6 レイアウトモデルによる面積評価

ディープサブミクロンの集積技術の実用化に伴い、提案する FPGA が設計ルールによってどのような面積で集積化されるのかを評価する。SRAM 方式の FPGA では、構成情報を SRAM の中に保持することや構成情報を書き込むためにシフトレジスタを使用していること、あるいは論理ブロックの構成も LUT (Lookup Table)

Table 1: Specification of the test chip

Input Data	25 bits
Output Data	26 bits
Configuration Data	78 × 96 bits
Error Register	126 bits
Chip Organization	9 DCLBs, 24 DCBs, 16 DSBs, 117 EDs
Supply Voltage	5V
Active Area Size	2.6mm × 2.6mm
Transistor Count	61,923
Process	0.6μm CMOS 3 metal layers VDEC ROHM

から構成されるなど SRAM が多数組み込まれている。そこで、提案する FPGA では横 36λ × 縦 65λ で設計した SRAM を基本とし、設計ルールはゲート幅およびゲートスペースを 2λ、メタル幅を 3.5λ、メタルスペースを 2.5λ として面積評価を行った。

面積評価は、Fig.5(a) で示す非冗長の FPGA、Fig.5(b) で示す提案する 2 重化した FPGA とともに再構成を容易にするためレイアウトを規則的となる配置にする。

FPGA 全体の面積は、論理ブロックが  $n \times n$  集積化されているとすれば、非冗長で構成された場合式 (1) で表され、2 重化された FPGA 全体の面積は、式 (2) のように表すことができる。

$$A_{fpga} = ((n+1)\alpha + n \cdot \beta + w_{co}) \cdot ((n+1)\delta + n \cdot \gamma + h_{co}) \quad (1)$$

$$A_{dfpga} = ((n+1)\alpha' + n \cdot \beta' + w_{co}) \cdot ((n+1)\delta' + n \cdot \gamma' + h_{co}) \quad (2)$$

ここで、 $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  は以下の条件に基づき、行列のそれぞれサイズの大きい方が選択される。

$$\alpha = \{ \max(w_{cbv}, w_{sb}) \}$$

$$\beta = \{ \max(w_{clb}, w_{cbh}) \}$$

$$\gamma = \{ \max(h_{cbv}, h_{clb}) \}$$

$$\delta = \{ \max(h_{cbh}, h_{sb}) \}$$

また、2 重化した場合も同様に  $\alpha'$ ,  $\beta'$ ,  $\gamma'$ ,  $\delta'$  を選択する。

ここで、 $w_{co}, h_{co}$  は、構成情報を書き込むためのコンフィグレーションレジスタで面積計算に使用する横と縦方向の長さを示す。

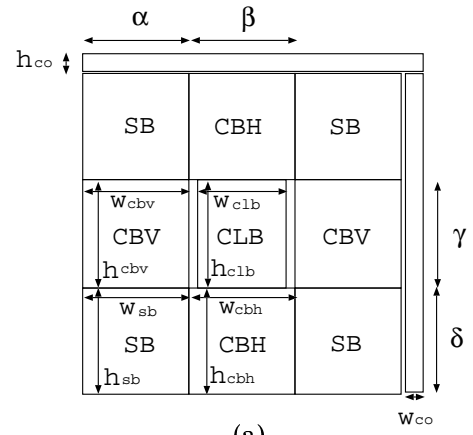
以下では、各ブロック毎の横と縦の大きさおよび面積について評価する。

まず、論理ブロックは、Fig.6 に示すように 2 重化された論理ブロック、入出力側に配置されたエラーディテクタ、検出結果を格納するエラーレジスタおよび故障の有無を検出するダイナミック回路で構成された多入力 OR 回路から構成されている。

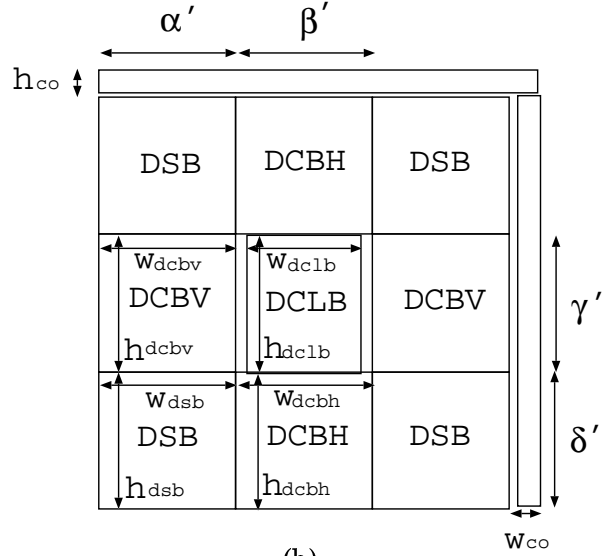
非冗長の論理ブロックの面積 ( $A_{clb}$ ) と 2 重化された論理ブロックの面積 ( $A_{dclb}$ ) は、それぞれ式 (3), (4) で表される。

$$A_{clb} = w_{clb} \times h_{clb} \quad (3)$$

$$A_{dclb} = w_{dclb} \times h_{dclb} = (2w_{clb} + 2w_{ed}) \times (h_{clb} + h_{er} + h_{dor}) \quad (4)$$



(a)



(b)

Fig. 5: (a) Layout of the non-redundant FPGA (b) Layout of the proposed FPGA

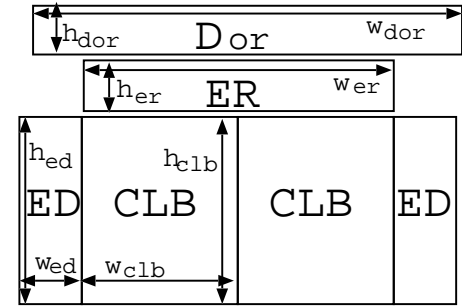


Fig. 6: Layout of the Configuration Logic Block

2 重された論理ブロックは、非冗長の場合と比較し式 (5) で示すように約 3.4 倍程度となる。

$$A_{dclb} = 2.58w_{clb} \times 1.35h_{clb} = 3.4A_{clb} \quad (5)$$

次に、スイッチブロックのレイアウトは、Fig.7 で示すように 6 方向トランジスタとそれを制御する SRAM から構成される。

スイッチブロックの面積は、配線数に依存し、それに伴う 6 トランジスタスイッチを構成する領域の菱形の面積が大きくなる構成となっている。したがって、ス

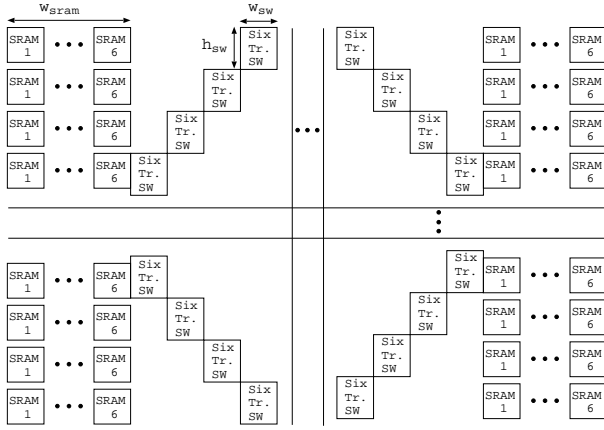


Fig. 7: Layout of the switch block

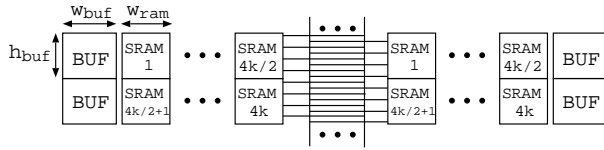


Fig. 8: Layout of the connection block

イチブロックを非冗長で構成した場合の面積 ( $A_{sb}$ ) および 2 重化した場合の面積 ( $A_{dsb}$ ) は、配線数が  $4k$  本のときそれぞれ式 (6) 式 (7) で表すことができる。

$$\begin{aligned}
 A_{sb} &= w_{sb} \times h_{sb} \\
 &= (2w_{sram} + 2kw_{sw} + 4k \cdot 3.5\lambda + (4k + 1) \cdot 2.5\lambda) \\
 &\quad \times (2k \cdot h_{sw} + 4k \cdot 3.5\lambda + (4k + 1) \cdot 2.5\lambda) \quad (6)
 \end{aligned}$$

$$\begin{aligned}
 A_{dsb} &= w_{dsb} \times h_{dsb} \\
 &= (2w_{sram} + 4kw_{sw} + 8k \cdot 3.5\lambda + (8k + 1) \cdot 2.5\lambda) \\
 &\quad \times (4k \cdot h_{sw} + 8k \cdot 3.5\lambda + (8k + 1) \cdot 2.5\lambda) \quad (7)
 \end{aligned}$$

次に、コネクションブロックは、外部および論理ブロックとの入出力を配線経路を通して行うため、Fig.8 で示すようにバッファおよび配線との接続を制御する SRAM から構成される。コネクションブロックの面積は、配線数および外部と論理ブロックに接続される入出力 ( $l$ ) 数に依存する。横型のコネクションブロックが非冗長で構成された場合の面積 ( $A_{cbh}$ ) と 2 重化した場合の面積 ( $A_{dcbh}$ ) は、配線数が  $4k$  本のとき、それぞれ式 (8)、式 (9) で表すことができる。

$$\begin{aligned}
 A_{cbh} &= w_{cbh} \times h_{cbh} \\
 &= (2w_{buf} + 4k \cdot w_{ram} + 4k \cdot 3.5\lambda + (4k + 1) \cdot 2.5\lambda) \\
 &\quad \times (2h_{buf} \times l) \quad (8)
 \end{aligned}$$

$$\begin{aligned}
 A_{dcbh} &= w_{dcbh} \times h_{dcbh} \\
 &= (2w_{buf} + 4k \cdot w_{ram} + 8k \cdot 3.5\lambda + (8k + 1) \cdot 2.5\lambda) \\
 &\quad \times (4h_{buf} \times l) \quad (9)
 \end{aligned}$$

上記の式より、提案する FPGA は、論理ブロックが固定の大きさであるのに対し、配線数や入出力数 ( $l$ ) が多くなるほどスイッチブロックおよびコネクションブ

ロックのサイズが大きくなる。規則的なレイアウトになるよう設計すると、面積はこれらのサイズにより決定される。

例として、今回の試作した論理ブロックが  $3 \times 3$  集積化され、配線数 12 本、入出力数 5 で設計した場合について、非冗長の場合と冗長化した場合のハードウェアのオーバーヘッドの比較を行った。非冗長の FPGA の面積 ( $A_{fpga}$ ) と提案する FPGA の面積 ( $A_{dfpga}$ ) は、それぞれ式 (10)、(11) で表され、冗長化した場合のオーバーヘッドが 2.53 倍程度となった。

$$\begin{aligned}
 A_{fpga} &= ((n + 1)w_{sb} + n \cdot w_{cbh} + w_{co}) \\
 &\quad \times ((n + 1)h_{cbh} + n \cdot h_{cbv} + h_{co}) \quad (10)
 \end{aligned}$$

$$\begin{aligned}
 A_{dfpga} &= ((n + 1)w_{dsb} + n \cdot w_{dcbh} + w_{co}) \\
 &\quad \times ((n + 1)h_{dcbh} + n \cdot h_{dcbv} + h_{co}) \quad (11)
 \end{aligned}$$

さらに、市販されている FPGA の論理ブロック数程度に増やした場合を想定する。配線数を 12 本、入出力数を 5 とし論理ブロック数を  $50 \times 50$  程度まで増やした場合、ハードウェアのオーバーヘッドが 2.7 倍に近づくことがわかった。また、論理ブロックを  $50 \times 50$ 、入出力数を 5 とし、配線数を増やした場合、スイッチブロックとコネクションブロックのサイズが大きくなるため、ハードウェアのオーバーヘッドが 4 倍程度になることがわかった。

## 7 むすび

今後の課題として、高信頼化によるハードウェアのオーバーヘッドを少なくできるような手法について検討することが重要となる。

**謝辞** 本研究のチップ試作は、東京大学大規模集積システム設計教育センター (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ある。

## 参考文献

- [1] S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic: Field-Programmable Gate Array, Kluwer Academic Publishers, 1992.
- [2] S. M. Trimberger: Field-Programmable Gate Array Technology, Kluwer Academic Publishers, 1994.
- [3] 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.
- [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 - PartII: Circuit Design and Layout, IEEE Transactions on VLSI Systems, Vol. 7, No. 3, pp. 321-330, 1999.
- [5] 阿部, 青木, 樋口, 鹿股: 故障の検出および回復が可能な FPGA アーキテクチャ, 信学技法, FTS2000-1, pp.1-8(2000).