

# エラー補償手続きとブロック統合による FPGA 回路変換

1 L-5

幸田武範 上林彌彦  
京都大学工学研究科

## 1 はじめに

FPGA(Field Programmable Gate Arrays)は、実現論理を自在に変更可能な論理ブロックを二次元的に配置したゲートアレイである。本素子はその特徴のため、回路の動作検証のためのプロトタイプなどに利用されることが多い。一般に、プロトタイプ回路作成時は設計にかかる時間を短縮させるために、多くの時間を必要とする強力な最適化を適用せず、単純なマッピングのみを適用し、回路を生成することが多い。しかし、使用可能な素子面積には制限が存在することから、何らかの回路規模を小さくする手法の適用が望まれる。

本稿では、代表的な FPGA である XILINX 社の FPGA 及び同様の構成をもつ FPGA について、その論理ブロックが一般的な仮定である「5変数以下の任意関数を1つだけ実現可能」というだけでなく、実際には1つの任意の5変数関数、1つの4変数関数と1つの特定形の5変数関数又は2つの独立な4変数関数のいずれかを実現可能であるという点に注目し、この特徴を利用した回路変換手法を提案する。本手法においては、仕様変更の際などに元回路に生じた差異をエラーとみなし、そのエラーを論理ブロックの内部論理変更や入力修正などによって補償するエラー補償手続きを用い、元来単独で1ブロックを占有していた5変数関数を同一ブロック内に他の関数を実現可能な形に変形する。本手法を適用することで、時間がかかる強力な FPGA 回路最適化手法を適用しなくても、高速に回路内のブロック使用率を向上させ、最適化手法と同等かそれ以上の必要ブロック数の減少を実現できる。ここでは、本手法の有用性を示すために、提案手法を MCNC ベンチマーク回路に適用し評価を行ったので、その実験結果も示す。

## 2 基本的事項

### 2.1 FPGA

FPGA は自由に内部論理を変更可能な論理ブロックとそれらのブロック間の内部結線により実現されるリコンフィギュラブル素子である。いままでに種々の FPGA が開発・発売されており、それぞれ特徴は大きく異なるため、すべての FPGA を統一的に扱うことは難しい。そこで本稿では、最も多く利用されている XILINX 社の XC4000 シリーズを対象としているが、他社の FPGA への応用も可能である。XC4000 シリーズは図1に示す4変数又は5変数の任意関数を実現可能な論理ブロックとブロック間に用意された内部結線によって構成されている。

図中の  $f_1, \dots, f_4, g_1, \dots, g_4, h_1$  はブロック入力、 $X, Y$  はブロック出力である。また、 $G'$  はブロック入力  $g_1, \dots, g_4$  の組合わせ、 $F'$  は  $f_1, \dots, f_4$  の組合わせで実現される任意の論理関数であり、 $H'$  は  $F', G', h_1$  の任意の組合わせで実現される関数である。各出力はマルチプレクサの設定によって、 $F'$  か  $H'$ 、又は  $G'$  か  $H'$  を選択可能である。

### 2.2 エラー補償手続き

エラー補償手続きは、回路内で生じたエラーを修正する手続きである。エラーの修正の方法により、以下の3種類の手続きがある。

- ・内部論理変更 (BLM:Block Logic Modification)  
ブロックの内部論理を修正し、エラーを補償する。

Transformation of FPGA Circuits Based on Error Compensation Procedures and Block Integration.

Takenori KOUDA, Yahiko KAMBAYASHI

Department of Information Science, Kyoto University

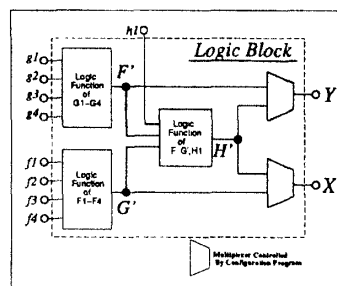


図1: XC4000 シリーズの論理ブロック

要求出力をブロックの入力関数でシャノン展開し、手続き成功後の内部論理を計算する。

- ・入力補償 (IC:Input Compensation)

BLMを拡張したもので、ブロックの内部論理のみならず入力関数の論理の一部も修正し、エラー補償を試みる。入力側ブロックの関数を変更するためその変更が回路出力に影響をおよぼさないようにする必要があるので、そこで、トランスダクション法<sup>[1]</sup>の回路冗長度を示す概念 CSPF を用い、不都合を生じさせないようにする。

- ・接続変換 (SC:Substitution Compensation)

ブロックの入力結線を他のブロックからの結線と交換し、エラー補償を試みる。本手続きのみでエラー補償を実現することは困難なため、単独で用いるのではなく前述の内部論理補償手続きや入力補償手続きと組み合わせて用いる。

## 3 FPGA 回路変換手法

### 3.1 ブロック統合

今回扱う XC4000 シリーズの場合、一つの論理ブロックについて任意の4変数関数を独立に2つ、又は任意の5変数関数を1つ実現可能である。しかし一般的な FPGA 設計ツールでは、この特徴を生かすことなく1ブロックを4変数以下の関数1つで占有するようなマッピングをするため、ブロックの利用効率が大変悪い。そこで、本稿ではこの特徴を生かした手法を考える。そのための準備として、FPGA用にマッピングされた回路データを4変数関数2つを1ブロックで実現（以下、これを「ブロック統合」と呼ぶ）し、マッピングされた回路データの再評価を行った。以下、この評価法のことを「ブロック統合評価」と呼ぶ。尚、FPGAは内部結線網の設定自由度が大きいので、一つのブロックに入力/出力からの深さ（入力/出力レベル）が全く異なる2関数を実現することも可能である。そのため、ここでは同一ブロックで実現する2関数の入力/出力レベルに差があっても、ブロック統合が可能であるとみなした。

### 3.2 FPGA 回路変換手法

ブロック統合可能となるのは4変数以下の関数だけではない。5変数関数でも仮に  $Y = H' = h_1 * G'$  の形で表現可能であれば、もう一つの出力  $X = F'$  とともに一つの論理ブロック内で実現可能となる。このような形で表現できるようブロックの内部論理や入力関数を変更する手続きを本稿では、「4+1分割」と呼ぶことにする。

本回路変換手法は、5変数関数に対してエラー補償手続きを利用した冗長入力の削除と4+1分割を試み、ブロック統合評価における論理ブロック数の削減を実現する。

表 1: 回路変換手法適用結果 及び 最適化手法との比較

| 回路名      | 初期回路             |                 | 適用手法             |                  | 比較手法             |                 |
|----------|------------------|-----------------|------------------|------------------|------------------|-----------------|
|          | 5入力マッピング回路       |                 | 提案手法 (BLM,IC)    |                  | 4入力マッピング回路       |                 |
|          | LB(LB)/Conn/Lev  | LB/Conn/Lev/CPU | LB/Conn/Lev/CPU  | LB/Conn/Lev/CPU  | LB(LB)/Conn/Lev  | LB/Conn/Lev/CPU |
| C432     | 60(80)/304/14    | 52/304/14/9     | 52/299/14/778    | 51(101)/360/13   | 54/277/14/1892   |                 |
| alu4     | 183(236)/1009/23 | 136/981/23/6    | 129/935/23/232   | 151(302)/1059/28 | 158/900/23/610   |                 |
| dalu     | 291(414)/1772/11 | 233/1727/11/24  | 232/1721/11/1303 | 310(619)/2088/1  | 237/1447/10/3893 |                 |
| example2 | 78(109)/413/5    | 57/409/5/1      | 57/392/5/80      | 65(129)/402/6    | 70/378/5/107     |                 |
| f51m     | 31(44)/173/10    | 29/173/12/0.1   | 26/167/10/9      | 33(67)/225/1     | 27/152/10/29     |                 |
| i5       | 82(112)/459/6    | 81/459/6/1      | 57/369/6/346     | 78(156)/513/6    | 79/447/6/382     |                 |
| i7       | 181(325)/1219/2  | 163/1219/2/2    | 163/1219/2/149   | 198(395)/1331/3  | 181/1219/2/151   |                 |
| vda      | 394(566)/1866/5  | 285/1866/5/4    | 283/1760/5/359   | 366(732)/2046/5  | 276/1367/5/484   |                 |

(統合ありブロック数 (統合なしブロック数) / 結線数 / 段数) 又は (統合ありブロック数 / 結線数 / 段数 / CPU 時間 (.sec))

### 3.2.1 冗長入力結線の削除

5入力ブロックの各入力について、以下の処理を行う。

- 入力結線切断
- エラーが発生したら、エラー補償手続き適用。成功なら処理終了。失敗なら切断した結線を元に戻す。

処理に成功した場合、当該ブロックで実現されていた5変数関数は4変数関数になるため、元々4変数の関数と同様にブロック統合を行うことが可能となる。

### 3.2.2 4+1 分解

5変数関数  $Z_5$  は  $a_1, \dots, a_5$  からなる関数で、4変数関数  $Z_4$  は  $b_1, \dots, b_4$  からなる関数であるとする。ここで、5変数関数  $Z_5$  を4つの変数  $a_1, \dots, a_4$  からなる2つの関数  $A_1, A_2$  と変数  $a_5$  を用いたで表現した場合、以下のようになる。

$$Z_5 = a_5 * A_1 + \bar{a}_5 * A_2$$

ここで、 $Z_5$  を5変数のまま  $Z_4$  とのブロック統合可能な形にするためには、以下の条件のうちのどれかを満たせばよい。

[ブロック統合可能となる条件]

- $A_1$  か  $A_2$  のどちらかが恒偽
- $A_1 = A_2$

仮に5変数関数  $Z_5$  が1つ目の条件を満たしている場合(例:  $A_2$  が恒偽)、又は2つ目の条件を満たしている場合、当該ブロックで以下に示す2関数  $Z_5, Z_4$  を同時に実現可能となる。尚、 $Z_4$  が3変数以下の場合や  $G'$  と  $F'$  を交換した場合も同様である。

5変数関数  $Z_5 = Y = H' = h_1$  と  $F'$  の積和形  
( $F' = A_1, h_1 = a_5$ )

4変数関数  $Z_4 = X = G'$

しかし、上記の条件を修正を行わなくても満たす5変数関数はテクノロジーマッパーの生成した回路の中にはほとんど存在しない。そこで本手法では、このような場合には処理対象入力を  $a_5$  とみなし、 $A_1$  について下記の方針に従い理想値を求め、 $a_5$  以外の入力の組み合わせで理想値を実現するためにエラー補償手続きを適用する。手続きに成功した場合、当該関数は上記の統合条件を満たし他の関数と統合可能な形になる。失敗した場合は、次の入力について同様の試みを行い、統合可能な形を求める。

[ $A_1$  の理想値]

以下のいずれかを満たす関数

a)  $A_1 = \text{恒偽}$

b)  $A_1 = a_5 \cdot Z_5 + \bar{a}_5 \cdot \bar{Z}_5$

尚、 $A_2$  についても  $A_1$  の場合と同様である。

### 3.3 実験結果とその考察

前節で述べた回路変換手法をC言語を用いて実装し、SUN ULTRA1にて実験を行った。表1にその実験結果を示す。初期回路には、MCNCベンチマーク回路をSISで

用いて5入力論理ブロックからなるFPGAにマッピングした回路を用いた。提案手法との比較を行うために、SISで4入力論理ブロックにマッピングした結果と、提案手法と同様にエラー補償手続きを利用したFPGA回路最適化手法<sup>[2]</sup>を適用した結果も表の最後の欄に加えてある。これらの欄内のブロック数は、全てブロック統合を行ったときの数である。ブロック数について、括弧の中に数が示してあるものは統合を行わないときのブロック数である。提案手法については、冗長入力削除の際に4+1分解を行う際に利用したエラー補償手続きの種類により、BLMとICのみ使用した「提案手法 (BLM,IC)」と、BLMとIC,SCを利用した「提案手法 (all)」を分けて表現してある。太字の部分の結果がよかった個所である。

実験結果から、提案手法は5入力ブロックにマッピングした時より結線数や段数を増加させることなく、4入力ブロックにマッピングした時以上に必要ブロック数の少ない回路を短時間で生成できることを示した。また、最適化手法と比較しても、結線数などでは及ばないもののブロック数の減少効果では優れていることが示された。また、SCまで利用したときと比較し能力は落ちるもののBLMとICのみを使った回路変換手法は、高速に回路変換を行い、プロトタイプ作成用途では十分要求を満たすものと思われる。

## 4 結論

本稿で提案した手法は、特定形の関数であれば1つのブロックで2つ同時に実現できるという論理ブロックの特徴を利用するものである。本手法では、元来ブロック統合不可能な5入力関数をエラー補償手続きを用いて4変数関数に変形させたり、5変数関数のままでもブロック統合可能な形に変形させることで、全体としてのブロック数の減少を実現した。回路最適化手法との比較においても、結線数などでは劣るもののブロック数削減に関しては短時間で十分な効果を得られることが示された。しかし、提案手法では回路冗長性の表現法としてCSPFを採用しているが、CSPFの代わりに文献<sup>[3]</sup>にて発表されたFPGA向けの冗長性表現法であるSPFDを採用することで、提案手法のさらなる改良が可能と思われる。今後の課題としては、これらのより強力な回路冗長性表現法を用いた回路変換手法の実装と実験、結果比較などがあげられる。

## 参考文献

- [1] S.Muroga, Y.Kambayashi, H.C.Lai, J.N.Culliney, "The Transduction Method-Design of Logic Networks Based on Permissible Functions", IEEE Trans.Comput., pp.356-359, Nov.1989
- [2] 幸田武範, 上林彌彦, "内部論理補償と入力補償によるFPGA回路設計", 情報処理学会 第52回全国大会, Vol 6, 1K-10, pp19, Mar.1996
- [3] S.Yamashita, H.Sawada, A.Nagoya, "A New Method to Express Functional Permissibilities for LUT based FPGAs and its Applications", In Proc. of IEEE/ACM ICCAD'96, pp.254-pp.261, Nov.1996