解　説

FPGA—その現状，将来とインパクト

4. FPGAの論理設計法†

笹　尾　　勤††

1. はじめに

現在，種々の FPGA が多数の会社から発売されている．ここでは，FPGA を4つの FPGA のアーキテクチャに分類し，各アーキテクチャに対して，代表的な設計法を述べる．また，本稿では，組合せ回路の設計法のみを述べる．

従来から用いられている FPLA (Field Programmable Logic Array) は AND—OR の二段論理回路でモデル化でき，論理設計は二段論理式の簡単化に帰着できる．そのため，設計方法は完成したといえる[1),2)]．一方，FPGA では，多段論理構造をしており，基本回路や設計の評価関数が FPLA に比べ複雑であるため，設計法も FPLA に比べはるかに複雑である．現在，多くのグループが種々の論理設計法を開発中であるが，年々改良されており，今後も大きな発展が期待される．

2. 種々の FPGA とそのモデル

テーブル参照形 FPGA

この FPGA は，**図-1** に示すように，アレイ状に並べた論理ブロックを縦横に走る配線領域で相互接続できるようにしたものである．論理素子は，LUT (Look up table) と呼ばれ，任意の k 変数関数を実現できる．k=4〜6 のものが多く，LUT と配線はスタティック RAM を用いてプログラムする．XILINX 社 LCA, AT&T 社 ORCA などがこの例である．実際の論理ブロックは，**図-2** のようにかなり複雑で，多出力を実現し，フリップ・フロップを含む．配線接続にはトランスミッションゲートを用いており，

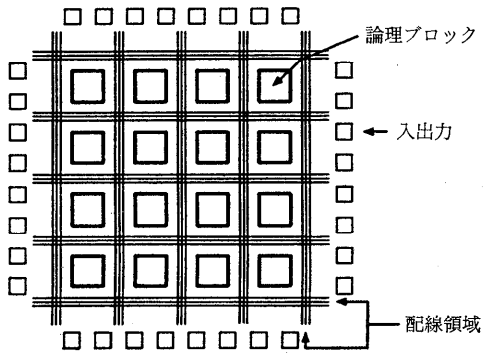


図-1　テーブル参照形 FPGA

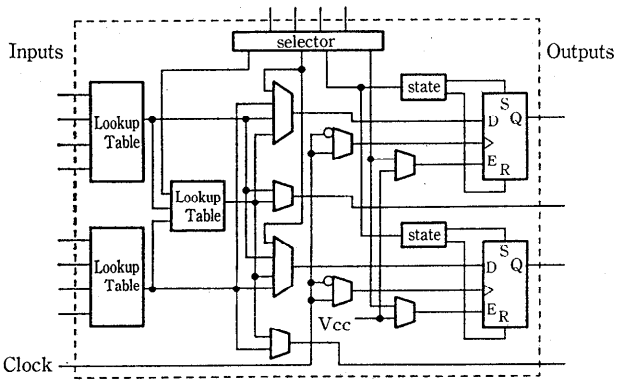


図-2　テーブル参照形 FPGA の論理ブロック
(ザイリンクス社 XC 4000)

接続時の ON 抵抗が高いため，通常のゲートアレイに比べ，配線遅延が大きくなる傾向がある．したがって，大規模 FPGA では，k を大きくしたり，配線部を大きくとって，遅延時間が大きくなり過ぎないように工夫している．この FPGA では，繰り返しプログラムでき，システム動作中にもプログラムを変更できるのが大きな特徴である．そのため，試作，教育，ロジックエミュレータなどに広く利用されている．

マルチプレクサ形 FPGA

この FPGA は，**図-3** に示すように，横に並べ

† Logic Design Methods for Field Programmable Logic Arrays, by Tsutomu SASAO (Dept. of Computer Science and Electronics, Kyushu Institute of Technology).

†† 九州工業大学情報工学部電子情報工学科

た論理ブロックを横に走る配線領域で相互接続できるようにしたものである．論理ブロックには，**図-4** のようなマルチプレクサ (MUX) が入っている．ACTEL 社の ACT, QUICK—LOGIC 社 pASIC などがこの例である．配線接続には，低抵抗のアンチフューズを用いており，テーブル参照形 FPGA よりも高速性を狙っている．したがって，高速性を必要とする応用に適している．ただし，プログラムは一度だけしか実行できず，変更はできない．

階層形 FPGA

この FPGA は，**図-5** のように，いくつかの FPLA (Field Programmable Logic Array) を中央の配線領域で，相互接続できるようにしたものである．この FPGA を CPLD (Complex Programmable Logic Device) と呼び，FPGA の範ちゅうに入れない人もいる．ALTERA 社の MAX, NS 社の MAPL, AMD 社の MACH などがこの例である．このタイプの FPGA の配線接続は高速で，遅延時間は論理ブロックに比べ無視できる．したがって，遅延時間は論理ブロック段数の整数倍となり，予測が容易である．プログラムは電気的に消去可能なものが多く，試作にも適し，いくつかの FPLA をまとめてパッケージ数を減らすという応用に最適である．

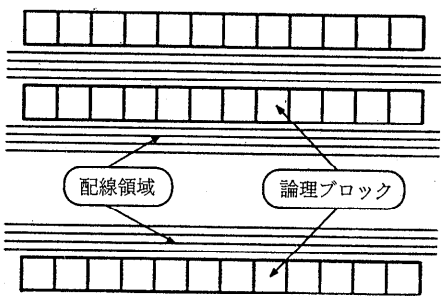


図-3 マルチプレクサ形 FPGA

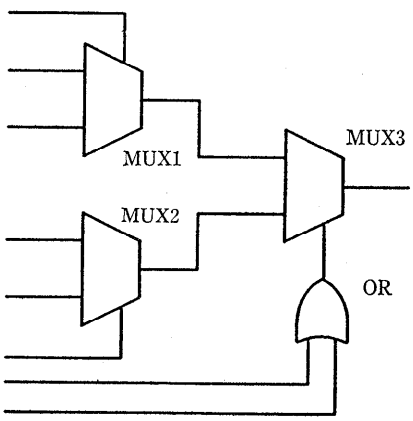


図-4 マルチプレクサ形 FPGA の論理ブロック (アクテル社 ACT-1)

ゲート敷き詰め形 FPGA

これは，通常のゲートアレイと非常によく似ており，設計法もゲートアレイのものが流用できる．CROSSPOINT 社の CP 20 K や ATMEL 社の CLI などがこの例である．配線接続には，低抵抗のアンチフューズを用いており高速である．この FPGA では，設計データが，そのままマスクタイプのゲートアレイの開発に使用できるため，量産用のゲートアレイのプロトタイプとして利用できる．ただし，プログラムは一度だけしかできない．

3. FPGA の論理設計法

FPGA の論理設計法は，そのアーキテクチャによって大きく異なる．実際の応用では，所与の FPGA に納まればよく，論理段数は少ないほうがよい．設計の良さを示す評価関数としては，必要なモジュール数と論理段数を用いる．ただし，遅延時間は，階層型 FPGA を除き，配置配線によって大きく変化する．一般に，モジュール数と論理段数を同時に減らすのは，容易ではない．

3.1 テーブル参照型 FPGA の論理設計

論理設計は，与えられた論理関数を k 入力 LUT を用いて実現する問題として定式化できる．ここで，各 LUT は，k 変数の任意の論理関数を実現する．実際の FPGA では，配線部分の遅延時間が，論理部分の遅延時間より大きくなりがちなので，配置配線には特に配慮が必要である．以下では，代表的な手法を述べる．

テクノロジマッピング法

与えられた論理関数を既存のツールで，AND,

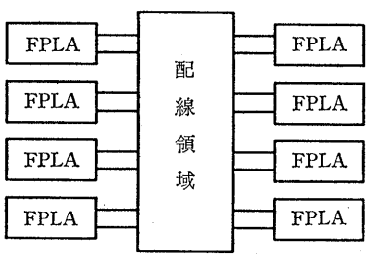


図-5 階層形 FPGA

OR などのゲートの多段論理回路に変換し，次に各ゲートを LUT にマッピングする方法である(**図-6** 参照)．多段論理回路の各ゲートのファンインが k 以下の場合には，比較的容易に設計できる[3]．この手法では，多段論理合成ツールや既存の回路など，既存の資産を利用できる利点があるが，基本論理素子として，AND, OR などのみを考えており，LUT のすべての論理能力を発揮させる手法ではない．

論理式分解法

与えられた論理関数を積和形論理式に変換し，次に，積項をいくつかの集合に分割する．各集合が依存する変数の個数を k 以下にすれば，各積項の集合は，k 入力の LUT で実現できる．LUT 数を最小化する問題は，ビン・パッキング問題として定式化できる[4]．論理関数の中間表現として，積和形論理式を用いているが，必ずしも，積和形を用いる必要はなく，他の論理式（たとえば AND—EXOR) を使えばさらに改良できる可能性がある．

関数分解法

与えられた論理関数 f が $f=g(h(X_1), X_2)$ の形で表現可能なとき，f は**図-7** の回路で実現できる．これを論理関数の分解という．回路 H の入力数が k 以下の場合，回路 H は一つの LUT で実現できる．この操作を繰り返し，すべて k 入力以下の回路ブロックに分解できれば，与えられた論理関数を FPGA で実現できたことになる．論理関数の分解可能性を調べるため，従来，分解表が用いられてきた．分解表とは，X_1 の変数の値のすべての組合せを列のラベルとし，X_2 の変数の値のすべての組合せを行のラベルとした，関数 f の真理値表である．列のパターン数(列複雑度)が 2 以下の場合は，図-7 の回路構造で実現できる．ただし，分解表の大きさが 2^n のため，入力変数の個数 n が大きい場合には，分解表は使えない．最近 BDD (後述) を利用して，分解表の列複雑度を能率良く計算する方法が考案された．まだ実験データが少ないが，入力変数の少ない場合には，他の方法に比べ LUT の少ない回路を生成できる[5)~7)]．LUT 形 FPGA の設計では，各 LUT が実現する論理の内容には関係なく，依存する変数の個数のみが問題となる．関数分解法では，依存する変数の個数のみを考慮しており，LUT 形 FPGA の設計に向いている．

このほか，依存変数の個数のみに着目した方法として別の方法も考案されている[8]．また，FPGA では，配置配線に時間がかかるので，配置配線を変化させずに LUT の内容のみを変更し再設計を高速にする手法も考案されている[9]．

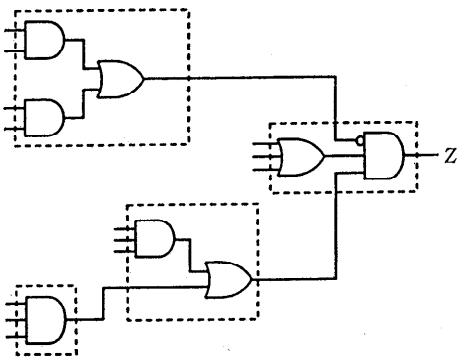


図-6 テーブル参照形 FPGA へのマッピング

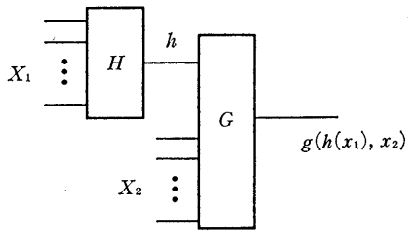


図-7 関数分解法

3.2 マルチプレクサ型 FPGA の論理設計 BDD を用いた方法

BDD (Binary Decision Diagram) とは，シャノン展開を繰返し適用して展開したものをグラフ表現したものである[10]．たとえば，論理関数 $f=x\cdot y \vee z$ は，**図-8** の BDD で表現できる．制御変数が 1 個の MUX は，BDD の一つの節点に対応する．

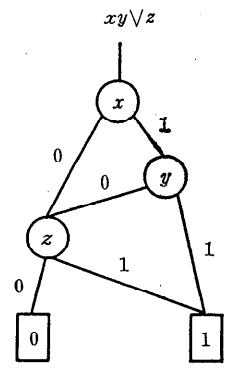


図-8 $f=x\cdot y \vee z$ の BDD

したがって，与えられた論理関数 f を BDD で表現し，各節点を MUX に置き換えれば，f の回路が実現できる．この方法で得られる回路の段数は，入力変数の個数 n に比例する．与えられた論理関数をそのまま一つの BDD で表現すると，大きくなりすぎる場合は，既存の手法で設計した多段論理回路を適当な大きさに分解し，各部分を BDD に変換する[3]．

ITE DAG を用いた方法

ITE DAG (if then else directed acyclic graph) を用いると，論理式 $F=\bar{x}F_0 \vee xF_1 \vee F_2$ は，

if F_2 then 1, else (if x then F_1 else F_0)

と表現できる．したがって，F は**図-9** の回路で実現できる．ここで，I の枝は MUX の制御変数 (if) に対応し，$I=1$ のとき，T (then) の枝が選択され，$I=0$ のとき，E (else) の枝が選択される．本方法は，BDD を用いた方法と似ているが，制御変数に必ずしも入力変数を接続する必要がないので，BDD の方法よりも MUX 数を少なくできる可能性がある[11]．

3.3 階層形 FPLA の論理設計

この形の FPGA に関する設計法の論文は多くない．数個の FPLA から構成された回路を人手によって一つの階層 FPLA にまとめるという設計法が，最も多用されているようである．大規模 PLA を階層 FPLA に分解するのは，PLA の分解（分割）に相当する[12]．分解には，**図-10** に示すように，入力を分割する直列分解と，出力を分割する並列分解がある[13),14]．

アーキテクチャは少し異なるが，任意の論理関数を AND—OR—AND (OR—AND—OR) として設計する研究がある[15]．また，最終段のゲートの入力数を 2 に制限した研究もある[16]．

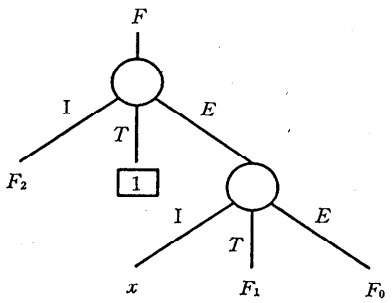
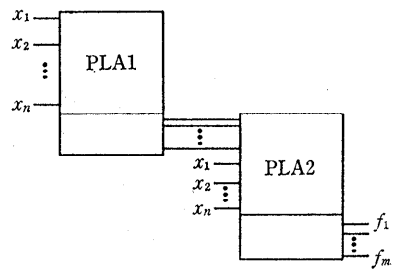
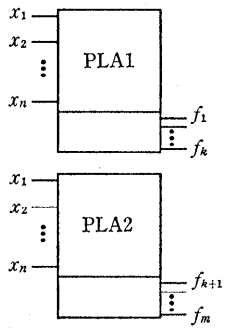


図-9 $F=\bar{x}F_0 \vee xF_1 \vee F_2$ の ITE DAG



(a) 直列分解



(b) 並列分解

図-10 PLA の分解

3.4 ゲート敷き詰め形 FPGA の論理設計

設計法はゲートアレイのものが流用できる[17]．ATMEL 社の FPGA 専用のものとして，論理関数を EXOR を用いた判定図で表現する方法も提案されているが，まだ研究段階である．

4. あとがき

FPGA は FPLA に比べ回路構造や評価関数が複雑であり，現在のところ，他の設計法に比べ圧倒的に優れているという方法は見つかっていない．したがって，多くの設計ツールでは，『種々の手法を組み合わせて回路を設計し，その中から最も適切な回路を選ぶ』方法を用いている．また，

1. 従来から研究されている AND—OR の多段回路の合成でさえも，それほど容易ではない．
2. FPGA の論理ブロックは，AND や OR などに比べると大幅に複雑である．
3. 多数の会社が毎年種々の論理構造の FPGA を開発している[18]．

ことを考えると，高性能な FPGA 設計システムを完成するには，相当な投資と年月が必要であると予測される．現在，20 以上の会社が独自のアーキテクチャの FPGA を発表している．しかし，

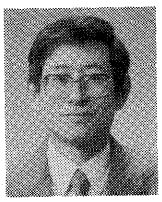
マーケットの大きさやツール開発の手間を考えると，この数は，多すぎ，あと5年程度で，いくつかの代表的なアーキテクチャに統合されるという予測もある．

FPGAの利用者にとって，デバイスごとに開発法が異なったり，ツールが異なったりするのは，大変煩わしい．標準的な記述言語で，すべてのデバイスを能率的に開発できるツールがあれば，大変便利である．各種デバイスや市販設計ツールの使い勝手などの実務的なことに関しては，日米で毎年開催される FPGA/FPLD 会議で報告されている[19]．

参 考 文 献

1) 笹尾 勤：VLSIにおける回路設計方式，情報処理，Vol. 28, No. 5, pp. 613-622 (May 1987).
2) 笹尾 勤：プログラマブルロジックアレーの設計支援システム，電子情報通信学会誌，Vol. 70, No. 10, pp. 1034-1037 (Oct. 1987).
3) Murgai, R., Nishizaki, Y., Shenooy, N., Brayton, R. and Sangiovanni-Vincentelli, A.: Logic Synthesis for Programmble Gate Arrays, Proc. 27th Design Automation Conference, pp. 620-625 (June 1990).
4) Brown, S.D., Francis, R.J., Rose, J. and Vranesic, Z.G.: Field Programmable Gate Arrays, Kluwer Academic Publishers, Boston (1992).
5) Sasao, T.: FPGA Design by Generalized Functional Decomposition, in (Sasao e.d.) Logic Synthesis and Optimization, Kluwer Academic Publishers (Jan. 1993).
6) 笹尾 勤，岡村康一郎：関数分解を用いたFPGAの設計手法について，情報処理学会研究会報告，Vol. 93, No. 94, 設計自動化 68-10 (Oct. 1993).
7) Lai, Y-T., Pedram, M. and Vrudhula, S.B.K.: BDD Based Decomposition of Logic Functions with Application to FPGA Synthesis, Proc. 30th Design Automation Conference (June 1993).
8) Fujita, M. and Matsunaga, Y.: Multi-Level Logic Minimization Based on Minimal Support and Its Application to the Minimization of Look-Up Table Type FPGA's, Proc. Inter, Conf. on Comput. Aided Design, pp. 560-563 (Nov. 1991).
9) Fujita, M. and Kukimoto, Y.: Patching Method for Look-Up Table Type FPGA's, Proc. Int. Conf. Comput. Aided Design, pp. 54-61 (Nov. 1992).
10) Bryant, R.E.: Graph-Based Algorithms for Boolean Function Manipulation, IEEE Trans. Comput. Vol. C-35, No. 8, pp. 677-691 (Aug. 1986).
11) Karplus, K.: Amap: A Technology Mapper for Selecter-Based Field-Programmable Gate Arrays, Proc. 28th Design Automation Conference, pp. 244-247 (June 1991).
12) Hasan, Z., Harrison, D. and Ciesielski, M.: A Fast Partitioning Method for PLA-Based FPGAs, IEEE Design & Test of Computers, pp. 34-39 (Dec. 1992).
13) 笹尾 勤，東田基樹：PLAの並列分解について，電子情報通信学会研究会資料 VLD 88-85 (1998-12).
14) Sasao, T.: Application of Multiple-Valued Logic to a Serial Decomposition of PLA's, Proc. Int. Sympo. on Multiple-Valued Logic, Zangzou, China, pp. 264-271 (May 1989).
15) Sasao, T.: On the Complexity of Three-Level Logic Circuits, International Workshop on Logic Synthesis, Research Triangle Park, North Carolina (May 1989).
16) Malik, A.A., Harrison, D. and Brayton, R.K.: Three-Level Decomposition with Application to PLDs, Proc. Inter. Conf. Computer Design, pp. 628-633 (Oct. 1991).
17) Marple, D.: An MPGA-Like FPGA, IEEE Design & Test of Computers, pp. 51-60 (Dec. 1992).
18) Clark, T.R.: Fitting Programmable Logic, IEEE Spectrum, pp. 36-39 (Mar. 1992).
19) The First Japanese FPGA/PLD Design Conference, CMP パブリケーションズ・インターナショナル (July 1993).

(平成5年10月12日受付)



笹尾 勤（正会員）

1950年生．1972年大阪大学工学部電子工学科卒業．1977年同大学院博士課程修了．工学博士．同年同大学助手．1988年九州工業大学情報工学部助教授，1993年同教授．現在に至る．この間，1982年2月より1年間 IBM ワトソン研究所客員研究員，1990年1月より3カ月間米国海軍大学院教授．1979年丹羽記念賞，1987年 ISMVL 論文賞，論理設計，スイッチング理論，多値論理などの研究に従事．最近は FPGA 設計，EXOR 論理回路に興味をもつ．著書「Logic Synthesis and Optimization」(Kluwer, 編著)，「論理設計とスイッチング理論—LSI, VLSI の設計基礎—」(共立出版，共訳)，「スイッチング理論演習」(朝倉書店，共著)，「PLA の作り方使い方」(日刊工業) など．電子情報通信学会英文論文誌編集委員，IEEE フェロー．