論　文

テーブル参照型FPGAにおける論理ブロックの検査法

道西　博行[†]　横平　徳美[††]　岡本　卓爾[††]　井上　智生[†††]
藤原　秀雄[†††]

A Test Methodology for Configurable Logic Blocks of a Look-up Table Based FPGA

Hiroyuki MICHINISHI[†], Tokumi YOKOHIRA[††], Takuji OKAMOTO[††],
Tomoo INOUE[†††], and Hideo FUJIWARA[†††]

あらまし　テーブル参照型FPGA（以下，単に，FPGAと呼ぶ）を任意にプログラムするときの正当性を効率良く検査することは，出荷時の検査として重要である．本論文では，FPGAの主要な構成要素の一つである論理ブロック（CLB）を対象に，この種の正当性の検査法を提案している．CLBは一つのルックアップテーブル（LUT），二つのマルチプレクサおよび一つのDフリップフロップから構成されるものとしている．ここでは，たかだか一つのCLBに多重故障が起こり得るという前提のもとで，まず，一つのCLBを単独に検査する場合のプログラムとテストパターンを導出している．次に，FPGA上でのCLBの配置の規則性に着目して，上述したプログラムとテストパターンを，すべてのCLBに対して同時に適用する方法を与えている．最後に，FPGA内のすべてのCLBを検査するためのプログラム回数が，アレーサイズ（CLB数）に依存しないことを明らかにしている．

キーワード　FPGA，論理ブロック，ルックアップテーブル，プログラム，故障検出

1. まえがき

FPGA（Field Programmable Gate Array）[1],[2]とは，プログラム可能なゲートアレーを集積したデバイスのことである．このデバイスはアレー状に配置されたプログラム可能な論理ブロック（以下，CLBと略記する）とこれらの間を結ぶプログラム可能な配線領域から構成されている．そして，CLBの機能および配線領域の接続情報をプログラムすることによって，所望の論理回路を実現することができる．

FPGAにはテーブル参照型，マルチプレクサ型，ゲート敷詰め型など種々のタイプがあるが，中でもテーブル参照型は，SRAMの内容によりCLBの機能やCLB間の接続情報を制御しているため，論理回路の機能を容易に変更できる．このため，テーブル参照型FPGA（以下，誤解の恐れのない限り，単にFPGAと呼ぶ）は，ディジタル回路のプロトタイプの設計だけでなく，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新たな分野でも利用されるようになってきている[3]．これに伴い，FPGAの高信頼化要求も高まり，現在，効率の良いFPGAの検査法を確立することが重要な課題となっている．

FPGAを検査する場合，チップ上に所望の機能をプログラムした後，その機能の正当性を検査するという立場と，チップ上に所望の機能をプログラムする前に，FPGAが任意のプログラムに対して正しく機能するか否かを検査する立場がある．前者の立場の検査法としては，これまでにいくつかの方法[4],[5]が提案されている．また，後者の立場の検査法としては，一部のSRAMセルの内容を書き換えることによりプログラムが変更できるということを仮定した方法[6]が提案されている．しかし，プログラムの変更のたびに，すべてのSRAMセルの内容の設定（以下，一括アクセ

† 岡山大学大学院自然科学研究科，岡山市
Graduate School of Science and Technology, Okayama University, 3-1-1, Tsushima-naka, Okayama-shi, 700 Japan
†† 岡山大学工学部情報工学科，岡山市
Faculty of Engineering, Okayama University, 3-1-1, Tsushima-naka, Okayama-shi, 700 Japan
††† 奈良先端科学技術大学院大学情報科学研究科，生駒市
Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma-shi, 630-01 Japan

スと呼ぶ）を要する現状の FPGA の検査法については，これまで報告されていない．

本論文では，後者の立場に立ち，かつ，一括アクセスを前提に，FPGA の主要な構成要素の一つである CLB の Go/No-Go レベルの検査法を提案している．まず，対象とする FPGA の構造と CLB の内部構成を明らかにした上で，故障モデルについて述べる．次に，一つの CLB を単独で検査する場合のプログラムとテストパターンの導出を行う．続いて，FPGA 上での CLB の配置の規則性に着目して，単独で検査する場合のプログラムとテストパターンを，すべての CLB に対して同時に適用する方法を与える．この結果によれば，FPGA 内のすべての CLB を検査するためのプログラム回数は，アレーサイズ（CLB 数）に依存せず，$2k+4$ になる．但し，k は，CLB 内のルックアップテーブル（以下，LUT と略記する）の入力数を表す．本論文の方法は，*Xilinx* の XC2000 シリーズ [7] における CLB の内部構成を参考にした FPGA モデルに対して得られたものであるが，CLB の内部構成が多少異なる FPGA に対しても，必要に応じて多少の変更を加えるだけで容易に適用できると思われる．

2. 論理ブロックの構成と故障モデル

2.1 FPGA の構造

FPGA は，図 1 に示すように，アレー状に配置された CLB とそれらを取り囲む配線領域から構成される．配線領域の信号線が交差する部分には，それらを接続するためのスイッチマトリクス（以下，SM と記す）が配置され，更に，FPGA の外周に沿った部分には，外部信号線との接続を行うための IO ブロック（以下，IOB と記す）が配置されている．同図の場合，アレーサイズは 4×4 であるが，以下，$N\times N$ とする．また，SM に接続される信号線の本数は，上下左右ともに 4 本であるが，以下，n 本とする．

FPGA には，CLB 内部の論理や配線領域の接続を決定するために，多数の SRAM セルが内包されており，電源投入時に順次情報が書き込まれるが，以下，このような書込みを行うことをプログラムすると言い，書き込まれる内容のことをプログラムと言う．

2.2 論理ブロックの構成

一つの CLB は，図 2 に示すように，LUT，D フリップフロップ（以下，DFF と記す）および二つのマルチプレクサ（以下，MUX と記す）からなり，$k+1$ 本の入力線と 1 本の出力線をもつ．入力線のうち，k 本は図面上，上側，左側および下側の各配線領域から，それぞれ $k_1(k=3k_1, k_1<n)$ 本ずつ LUT に接続され，残る 1 本は，左側の配線領域から直接 MUX1 に接続される．

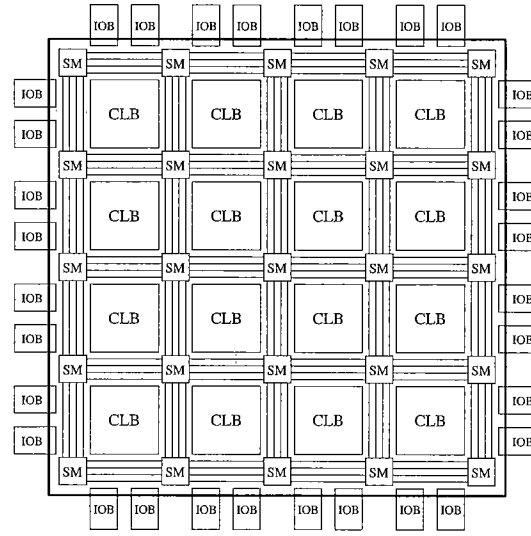


図 1 4×4 FPGA の構成
Fig. 1 Structure of 4×4 FPGA.

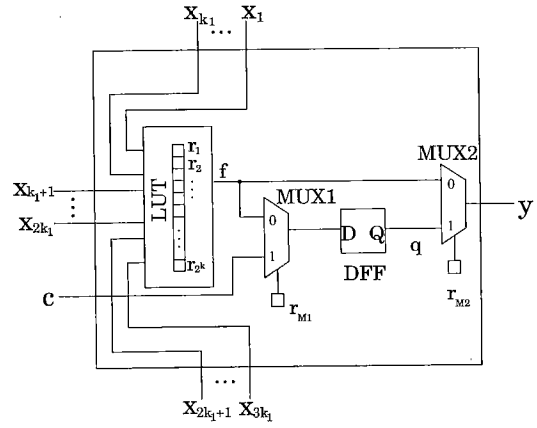


図 2 論理ブロックの構成
Fig. 2 Configuration of a CLB.

LUT には 2^k 個の SRAM セル $(r_1, r_2, \cdots, r_{2^k})$ が内包されており，いずれもあらかじめ 0 または 1 にプログラムされる．そして，k 本の入力線の論理値 $(x_1, x_2, \cdots, x_k)$ をアドレスとして，これらを読み出すことにより，任意の k 変数論理関数 f が実現できる．このときのアドレスデコーダは，プログラムするとき

に用いられるデコーダとは異なっており，以下，LUTデコーダと呼ぶ．また，MUX1は，SRAMセルr_{M1}の内容により制御されており，DFFにはこれを介して，上記の関数fの値またはCLBの外部信号の値cを記憶させることができる．更に，MUX2は，SRAMセルr_{M2}の内容により制御されており，これを介して，CLBの出力値yを，関数fの値またはDFFの出力値qとすることができる．

2.3 故障モデル

本論文の検査における分解能は，Go/No-Goレベルであるが，故障については，次のような仮定を設けている．

故障は，FPGA内のたかだか一つのCLBにのみ存在するものとする［仮定1］．また，故障の存在するCLBにおいては，多重故障も存在し得るものとする［仮定2］．但し，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，SRAMセルは，縮退していない限り，正しくプログラムされるものとする［仮定3］．

CLBの故障は，SRAMセルの縮退故障，LUTの入出力信号線の縮退故障，LUTデコーダの機能故障，DFFの機能故障およびMUXの機能故障とする．このうち，LUTデコーダの機能故障は，更に，無選択故障，誤選択故障および多重選択故障に分けられる．無選択故障が生起した場合，アドレス（$(x_1, x_2, \cdots, x_k)$のビットパターン）が印加されたにもかかわらず，いずれのSRAMセルも選択されないため，LUTの出力はハイインピーダンスとなる．そこで，アドレスを変更しながら連続的にSRAMセルの値を読み出そうとした場合，無選択故障が生じたアドレスに対するLUTの出力値は，直前の読出しにおけるLUTの出力値と等しくなるものとする．これに対して，プログラムを行った直後に無選択故障が生じたアドレスを印加した場合，LUTの出力は，0または1の一定値になるものとする（0となるか1となるかは，デバイスに依存して決まる）．また，誤選択故障が生起した場合，印加したアドレスとは異なるアドレスに対応するSRAMセルが選択される．更に，多重選択故障は，複数のアドレスに対応するSRAMセルが同時に選択される故障であるが，これが生起した場合，LUTの出力には，選択されたすべてのSRAMセルの値の論理和または論理積が出力されるものとする（論理和，論理積のいずれとなるかは，デバイスに依存して決まる）．DFFの機能故障は，状態数が増加しないという制限のもとで，他の順序回路に変換される故障であり，MUXの機能故障は，順序回路にならないという制限のもとで，出力関数が他の関数に変換される故障である．

3. 論理ブロックにおけるサブブロックの検査

CLBを構成するLUT，DFF，MUX1およびMUX2の各々をサブブロックと言う．各サブブロックの検査においては，便宜上，そのサブブロック以外は正常であると仮定する［仮定4］．また，テストパターンの設定およびそれに対する応答の観測は，FPGA全体の検査に拡張することを念頭において，それぞれ，仮想的にCLBの入力端および出力端で行うものとする．

3.1 LUTの検査

既に述べたように，LUTには2^k個のSRAMセルが内包されており，各SRAMセルはkビットの入力$x_1, x_2, \cdots, x_k$のいずれかのビットパターン（アドレス）に対応している．よって，プログラム時（書込み時）と読出し時とで，異なるデコーダが用いらていることを除けば，LUTは一種のRAMと考えることができる．

従来のRAMの検査法[8],[9]は，書き込んだデータが正しく読み出されるか否かにより，良否判別するものであり，その多くは，アドレスごとに書込みと読出しを行うことにより検査する方法である．しかし，FPGAでは，プログラム時において，すべてのSRAMセルの内容を一括して書き込まなければならない．このため，これらの方法をそのままLUTに適用することはできない．また，RAMの検査法として，すべてのアドレスに書込みを行った後に，すべてのアドレスから読出しを行う方法もあるが，LUTでは，書込みと読出しとでデコーダが異なるため，この方法を適用した場合には，LUTデコーダの機能故障がマスクされる可能性がある．以上のことから，本論文では，FPGAの全アドレスへの書込み（プログラム）とLUTの全アドレスの読出しを交互に行うという前提のもとで，以下に示す検査手続き（TP-LUT）を導出した．

［TP-LUT］

(1) $p = 1, 2, \cdots, k$に対して，以下の手続き(1.1)および(1.2)を行う．

(1.1) $f = x_p$, $r_{M1} = 1$, $r_{M2} = 0$となるようにプログラムを行う．

(1.2) テストパターンとして，CLBの外部入力にすべてのビットパターン（アドレス$(x_1, x_2, \cdots, x_k)$）

を印加し（但し，入力線 c は0または1に固定しておく），その応答を出力 y として観測する．

(2) $p=k+1,k+2,\cdots,2k$ に対して，以下の手続き (2.1) および (2.2) を行う．

(2.1) $f=\bar{x}_{p-k}$, $r_{M1}=1$, $r_{M2}=0$ となるようにプログラムを行う．

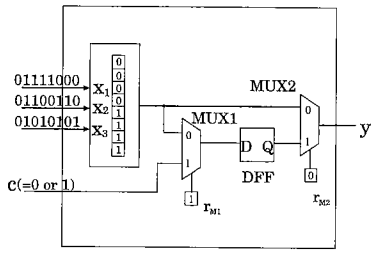
(2.2) 手続き (1.2) と同じ手続きを行う．

上述の手続き (1.2) におけるテストパターンの印加順序は，$(0,0,\cdots,0)$ および $(1,1,\cdots,1)$ 以外のアドレスから開始するという条件さえ満たせば任意でよいが，すべての p に対して，その順序は同じでなければならない（以下，制約Aと呼ぶ）．これに対して，手続き (2.2) におけるテストパターンの印加順序は全く任意であり，異なる p に対しても同一順序とする必要はない．

2^k 個の各SRAMセルに対して，対応するアドレスの2進表現を系列とみなし，その系列自身とその反転系列をこの順に連接した系列（以下，連接系列といい，アドレス i の連接系列を C_i で表す）を考える．$^{\forall}i,^{\forall}j(i\neq j)$ に対して，$C_i\neq C_j$，すなわち，$C_i(p)=0$ かつ $C_j(p)=1$ であるような p $(1\leq p\leq 2k)$ が必ず存在する．但し，$X(p)$ は系列 X の p 番目の要素を表す．従って，上述のTP-LUTを実行した場合，もし，故障が存在しなければ，$p(1\leq p\leq 2k)$ 番目のプログラムの後に，任意のアドレスから読み出される値は，そのアドレスに対応する連接系列の p ビット目に一致する．すなわち，各プログラムの後にアドレス i において読み出される値を順に連接させた系列 R_i（以下，応答系列という）は，C_i と一致する．例として，3入力LUTを検査するときの1番目のプログラム $(f=x_1)$ および連接系列を，それぞれ，図3(a)および(b)に示す．

次に，故障の検出について述べる．仮定1および4から，TP-LUTを実行したときのLUTの出力値はCLBの出力値と一致する．そこで以下，CLBの出力値の観測によりLUTのすべての故障が検出できることを示す．LUTの故障は**2.3**述べたように，LUTの入力線および出力線の縮退故障，SRAMセルの縮退故障，LUTデコーダの機能故障（無選択故障，誤選択故障，多重選択故障）であるが，このうち，LUTの出力線の縮退故障は，LUTの他の故障の有無に関係なく検出できることは自明である．よって，以下では，LUTの出力線の縮退故障は存在しないものとする．

[補題1] TP-LUTの実行結果が正常であれば，LUTの入力線の縮退故障は存在しない．



(a)

i	C_i
000	000111
001	001110
010	010101
011	011100
100	100011
101	101010
110	110001
111	111000

(b)

図3 LUTの検査のためのプログラムとテストパターンおよび連接系列の一例

Fig. 3 An example of programs, test patterns and concatenated sequences for testing of LUT.

(証明) LUTの入力線のうち任意の ℓ 本 $(1\leq\ell\leq k)$ が縮退しているとする．このとき，TP-LUTにおける手続き (1.2) および (2.2) においてLUTに与えられるアドレスは，たかだか $2^{k-\ell}$ 通りとなり，異なる応答系列もたかだか $2^{k-\ell}$ 通りしか得らないことになる．よって，$R_i\neq C_i$ となるアドレス i が存在する．

(証明終)

よって，以下では，LUTの入力線の縮退故障も存在しないものとする．

[補題2] TP-LUTの実行結果が正常であれば，LUTデコーダの無選択故障は存在しない．

(証明) 応答系列 R_i の先頭から h ビット目までの部分系列を，R_i^h と表すものとする．アドレス i が無選択故障であると仮定すれば，故障仮定より，TP-LUTにおけるアドレス i に対するLUTの出力値は，他の故障の有無に関係なく，直前に与えたアドレス j $(j\neq i)$ に対するLUTの出力値と等しくなり，$R_i^h=R_j^h$ が成り立つ．また，プログラムの直後，最初に与えるアドレスが無選択故障の場合，R_i^h はすべて0の系列（以下，S_0 で表す）またはすべて1の系列（以下，S_1 で表す）となる．

(証明終)

[補題3] TP-LUTの実行結果が正常であれば，LUTデコーダの誤選択故障は存在しない．

(証明) アドレス i を与えたとき，誤ってアドレス j $(j\neq i)$ のSRAMセルが選択されると仮定する．このSRAMセルが縮退している場合，$R_i=S_0(S_1)$ となる．また，そのSRAMセルが縮退していない場合，$R_i=C_j(\neq C_i)$ となる．

(証明終)

そこで以下，LUTデコーダの無選択故障および誤選択故障も存在しない（LUTデコーダの多重選択故

障および SRAM セルの縮退故障のみが存在する）ものとする．

[補題 4] TP-LUT の実行結果が正常であれば，LUT デコーダの多重選択故障は存在しない．

(証明) TP-LUT における $2k$ 回のプログラムによって，アドレス i の SRAM セルに書き込まれた値の系列を W_i で表す．アドレス i の SRAM セルが縮退していなければ，$W_i = C_i$ となり，0(1) 縮退していれば，$W_i = S_0(S_1)$ となる．

LUT デコーダに多重選択故障が存在する場合，ここでは，便宜上，LUT の出力値が選択された SRAM セルの値の論理和となるものとして証明を行う（論理積の場合にも同様にして証明できる）．

LUT の入力に，アドレス j を与えたとき，アドレス $j_1, j_2, \cdots, j_\ell (\ell \geqq 2)$ の SRAM セルが選択されるものとすれば，次式が成り立つ．

$$R_j = \bigvee_{i=1}^{\ell} W_{j_i}$$

但し，$\bigvee$ は長さ $2k$ の ℓ 個の系列のビットごとの論理和を示している．

（1） $j_1, j_2, \cdots, j_\ell$ の SRAM セルの中に 1 縮退のものが存在する場合

一般性を失うことなく，j_m $(1 \leqq m \leqq \ell)$ の SRAM セルが 1 縮退しているとする．このとき，$W_{j_m} = R_j = S_1$ となり，明らかに正常な結果が得られない．そこで以下，$j_1, j_2, \cdots, j_\ell$ に対応する SRAM セルは 1 縮退していないものとする．

（2） $j_1, j_2, \cdots, j_\ell$ の SRAM セルがすべて 0 縮退している場合

$R_j = S_0$ となり，明らかに正常な結果が得られない．よって以下，$j_1, j_2, \cdots, j_\ell$ の SRAM セルのうち，少なくとも一つは縮退していないと仮定し，一般性を失うことなく，そのアドレスを j_1 とする．

（3） $j_1 \neq j$ の場合

連接系列の性質から，$C_{j_1}(p) = W_{j_1}(p) = 1$, $C_j(p) = 0$ となる p が必ず存在する．故に，$R_j \neq C_j$ $(R_j(p) = 1, C_j(p) = 0)$ となる．

（4） $j_1 = j$ の場合

$j_2 \cdots, j_\ell$ の SRAM セルの中に縮退してないものが存在すれば，**3.** と同様になる．そこで以下，これら SRAM セルがすべて 0 縮退しているものとする．このとき，$R_{j_1}(= R_j) = C_{j_1} \bigvee S_0 = C_{j_1}$ となり，応答系列は正常時と一致する．しかし，j_a $(2 \leqq a \leqq \ell)$ の選択が正常であれば，$R_{j_a} = S_0$ となり，j_a の選択が多重選択故障であれば，j_a に対応する SRAM セルが 0 縮退しているので，上述の 1., 2., 3. のいずれかの場合に該当する． (証明終)

よって，以下の補題が成り立つことは明らかである．

[補題 5] TP-LUT の実行結果が正常であれば，SRAM セルの縮退故障は存在しない．

補題 1〜5 より，次の定理が成り立つ．

[定理 1] LUT 以外のサブブロックが正常であるなら，TP-LUT により，LUT の故障はすべて検出される．

3.2 DFF および MUX の検査

DFF および MUX の機能故障は，プログラムとテストパターンを適宜組み合せて生成される全数パターンを用いることで，すべて検出できる．以下に示す TP-DFF および TP-MUX は，それぞれ，DFF および MUX の検査手続きである．

[TP-DFF]

(1) $r_{M1} = r_{M2} = 1$ となるようにプログラム（LUT の出力関数は任意）する．

(2) LUT のすべての入力値を固定した上で，CLB の入力 c には DFF の固有系列における入力系列（例えば，000110）を印加し，CLB の出力を観測する（図 4 参照）．

[TP-MUX]

(1) $r_{M1} = r_{M2} = 1$，かつ，LUT の出力関数が TP-LUT におけるいずれかの関数 f となるようにプログラムする．

(1.1) LUT のすべての入力に 0 を印加した状態で，CLB の入力 c に 01 の系列を印加し，CLB の出

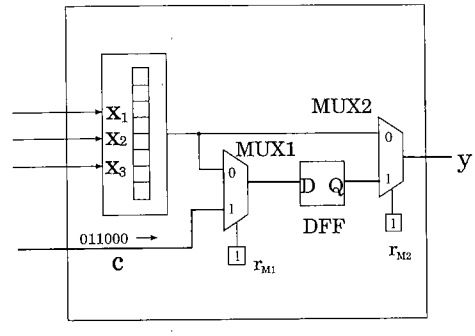


図 4 DFF の検査のためのプログラムとテストパターンの一例

Fig. 4 An example of programs and test patterns for testing pf DFF.

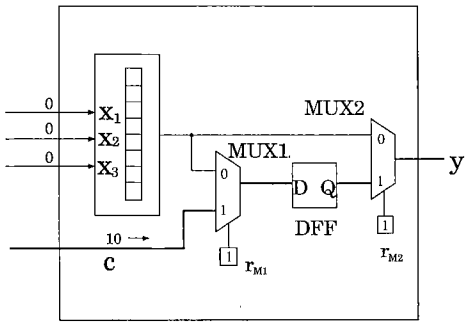


図5 MUXの検査のためのプログラムとテストパターンの一例
Fig. 5 An example of programs and test patterns for testing of MUX.

力を観測する（図5参照）.

(1.2) LUTのすべての入力に1を印加した状態で，CLBの入力 c に01の系列を印加し，CLBの出力を観測する.

(2) プログラムを $r_{M1}=1, r_{M2}=0$ に変更して，手続き(1)を行う.

(3) プログラムを $r_{M1}=0, r_{M2}=1$ に変更して，手続き(1)を行う.

4. 複数のサブブロックにおける多重故障の可検査性

3.では，仮定4のもとで，各サブブロックの検査手続き（TP-LUT,TP-DFF,TP-MUX）を導出した．本章では，これらの手続きをすべて行えば，仮定4を除外しても（複数のサブブロックに同時に故障が存在する場合），冗長故障以外はすべて検出できることを示す.

各サブブロックが常に正常な場合の出力値の否定を出力するような故障を，以下，反転故障と呼ぶ．但し，MUXの場合，選択されない入力の値が0あるいは1のいずれか一方のときのみ，選択すべき入力値の否定を出力する場合も反転故障に含めるものとする.

[補題6] TP-DFFの実行結果が正常であれば，反転故障以外のDFFの故障は存在しない.

(証明) TP-DFFでは，CLBの故障の有無にかかわらず，LUTの出力が不変に保たれる．このため，正常時に選択されないMUX1およびMUX2の入力値も不変となる．また，r_{M1} および r_{M2} の値も不変である．従って，故障の存在も考慮した場合，MUX1,MUX2の出力値は，(1)常に0，(2)常に1，(3)正常時に選択される入力値，(4)正常時に選択される入力値の否定のいずれかとなる．MUX2が(1)または(2)の場合，TP-DFFの実行結果が正常となることはあり得ないことから，以下では，MUX2は(3)または(4)であるとする.

MUX1が(1)または(2)の場合，TP-DFFにおいて，DFFの入力には常に同じ値が印加され続けることになる．従って，DFFのいかなる故障に対しても，固有系列における出力系列若しくはその系列の各ビットを反転させた系列（以下，反転系列という）がDFFの出力に現れることはない．よって，MUX1も(3)または(4)の場合のみを考えればよい.

MUX1,MUX2のいずれも(3)の場合，固有系列における入力系列が正しくDFFに印加され，そのときのDFFの出力系列がそのままCLBの出力に現れるので，補題6は成り立つ．MUX1,MUX2のいずれも(4)の場合，DFFの固有系列の反転系列も固有系列となることから同様である.

MUX1が(3)，MUX2が(4)の場合も，DFFの入力には固有系列における入力系列が正しく印加され，DFFの出力系列の反転系列がCLBの出力に現れる．よって，DFFに反転故障以外の故障が存在すれば，明らかにTP-DFFの実行結果は正常でない．MUX1が(4)，MUX2が(3)の場合も同様である． (証明終)

そこで以下，DFFは機能的に正常であるか，あるいは，反転故障が生じているのいずれかとする［仮定I］.

[補題7] TP-MUXの実行結果が正常であれば，$r_{M1}=1(r_{M2}=1)$ における反転故障以外のMUX1（MUX2）の故障は存在しない.

(証明) TP-LUTの実行結果で異常が見つかれば，TP-MUXの実行は不要となるので，この補題の証明に際しては，TP-LUTの実行結果は正常であるとする．このことから，TP-MUXにおいて，LUTのすべての入力に0を印加した場合と1を印加した場合とでLUTの出力値が異なることは明らかである．よって，TP-MUXの手続き(1)では，$r_{M1}=1$ としたときのMUX1のすべての入力の組合せが印加されることになる．このことと仮定Iから，補題6と同様にして，MUX1に関する補題7が容易に示される．また，$r_{M2}=1$ のときのMUX2についても同様である． (証明終)

そこで，以下，$r_{M1}=1(r_{M2}=1)$ としたときのMUX1(MUX2)の機能は正常，あるいは，反転故障とする［仮定II］.

[補題 8] TP-LUT の実行結果が正常であれば，LUT の故障は存在しない．

(証明) TP-LUT においては，入力 c の値が不変に保たれることと，仮定 I および II から，正常時に選択されない MUX2 の入力値も不変となる．よって，このときの MUX2 の出力値は補題 6 の (1)~(4) のいずれかとなることから，補題 6 と同様にして，補題 8 を証明できる． (証明終)

よって，以下では，LUT は機能的に正常であるとする [仮定 III]．

[補題 9] TP-MUX の実行結果が正常であれば，$r_{M1}=0(r_{M2}=0)$ における反転故障以外の MUX1 (MUX2) の故障は存在しない．

(証明) 仮定 III から，TP-MUX における手続き (3) により，$r_{M1}=0$ としたときの MUX1 のすべての入力の組合せが印加される．また，仮定 I および II から，MUX1 の出力値あるいはその否定のいずれかが CLB の出力に現れる．よって，MUX1 に関する補題 9 が成り立つ．$r_{M2}=0$ のときの MUX2 についても同様である． (証明終)

補題 6~9 より，TP-LUT,TP-DFF および TP-MUX のすべての実行結果が正常であれば，DFF,MUX の反転故障以外の故障は存在しないことは明らかである．また，TP-MUX においては，CLB の使用可能なすべての経路を検査しているので，TP-LUT,TP-DFF および TP-MUX のすべての実行結果が正常であれば，上述の反転故障は冗長故障であることになる．以上から，次の定理が成り立つ．

[定理 2] TP-LUT，TP-DFF および TP-MUX により，CLB における多重故障は，冗長故障以外すべて検出される．

5. 論理ブロックの同時検査法

本章では，テストパターンの印加および応答の観測をすべて IO 端子で行うという前提のもとで，**3.** で述べた各サブブロックの各検査手続きを複数の CLB に対して同時に適用することを考える．各手続きにおけるテストパターンが，すべての CLB に対して同時に印加でき，かつ，その結果が同時に観測できれば，一つの CLB の検査の場合と同じプログラム回数ですべての CLB を検査できることは明らかである．しかし，通常の FPGA においては，CLB 数が N^2 であるのに対して，IO 端子数 $(8N\sim 16N)$ が $O(N)$ であるために，上述した同時の観測が単純には実現できない．そこで，本論文では，CLB の出力を他の CLB の入力として利用し（CLB をカスケード接続し），一部の CLB の出力のみを観測することで，等価的に同時の観測を実現する．

5.1 LUT の同時検査法

3.1 の TP-LUT では，プログラムされる LUT の出力関数が，$f=x_i(f=\bar{x}_i)$ であるので，CLB が正常である限り，その出力には，LUT の入力 x_i に印加した系列がそのまま（反転して）現れる．従って，CLB の出力は別の CLB 内の LUT のテスト入力として利用できる．以下に示す手続き TP-LUT$_{ALL}$ は，このような着想のもとに，LUT の同時検査のための手続きとして得られたものである．

記述を容易にするために，図 1 の上から t 行目，左から u 列目に位置する CLB およびその CLB 内の LUT を，それぞれ，t 行 u 列目の CLB および LUT と呼ぶ．また，同図の左側，右側および上側の IO 端子を，それぞれ，左端，右端および上端の IO 端子と呼び，左端および右端の IO 端子を，紙面上の上から順に，それぞれ，I_1, I_2, … および O_1, O_2, … で表す．

[TP-LUT$_{ALL}$] (図 6 参照)

(1) $p=1,2,\cdots,k$ に対して，以下の手続き (1.1),(1.2) および (1.3) を行う．

(1.1) t 行目のすべての CLB に対して，$f=x_{m_t}$, $r_{M1}=1, r_{M2}=0$ となるようにプログラムする．

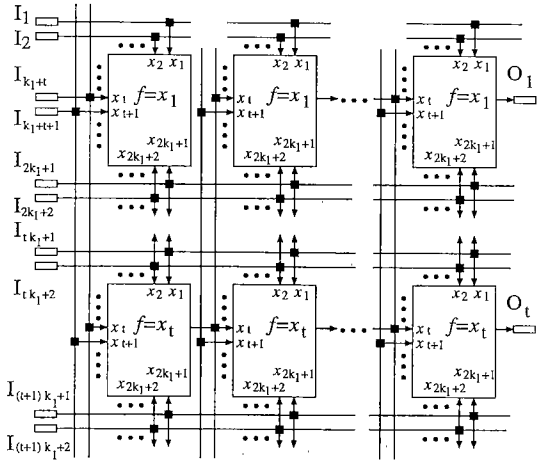


図 6 LUT の同時検査のための配線領域のプログラム
Fig. 6 Program of wiring segments for simultaneous test of all LUTs.

但し，$m_t = \{t+p-2\} \pmod k + 1$ である．

(1.2) 配線領域を，以下のようにプログラムする（このためには，左端および右端の IO 端子数が，それぞれ，$(N+2)k_1$ 以上および N 以上で，かつ，$N \geq k$ が満足される必要があるが，通常の FPGA ではこれが満足されるので，本論文でもこれを仮定する）．

(1.2.1) 1 行 1 列目の LUT の入力 $x_{v_1}(1 \leq v_1 \leq k_1)$，$x_w(k_1+1 \leq w \leq 2k_1)$ および $x_{v_2}(2k_1+1 \leq v_2 \leq 3k_1)$ を，それぞれ，I_{v_1}，I_w および I_{v_2} に接続する．

(1.2.2) t 行 1 列目 $(2 \leq t \leq N)$ の LUT の入力 $x_{v_1}(1 \leq v_1 \leq k_1)$，$x_w(k_1+1 \leq w \leq 2k_1)$ および $x_{v_2}(2k_1+1 \leq v_2 \leq 3k_1)$ を，それぞれ，$I_{tk_1+v_1}$，I_w および $I_{(t-1)k_1+v_2}$ と接続する．

(1.2.3) t 行 u 列目 $(1 \leq t \leq N; 2 \leq u \leq N)$ の LUT の入力 x_{m_t} を，t 行 $u-1$ 列目の CLB の出力と接続する．また，t 行 u 列目 $(1 \leq t \leq N; 2 \leq u \leq N)$ の LUT の x_{m_t} 以外の入力 $x_v(1 \leq v \leq k_1; 2k_1+1 \leq v \leq 3k_1)$ および $x_w(k_1+1 \leq w \leq 2k_1)$ を，それぞれ，t 行 $u-1$ 列目の LUT の入力 x_v および $m_{t_1} = w$ となるようないずれか一つの t_1 行 $u-1$ 列の CLB の出力と接続する．

(1.2.4) N 列目のすべての CLB の出力を，互いに異なる右端の IO 端子に接続する（一般性を失うことなく，接続された N 個の端子を，O_1〜O_N で表す）．

(1.2.5) すべての CLB の入力 c と一つの上端の IO 端子を接続する（接続された端子を U_1 で表す）．

(1.3) U_1 に印加する値を 0 あるいは 1 に固定したまま，I_1〜I_k に $k(=3k_1)$ ビットのすべてのビットパターンを印加する（**3.1** の制約 A を満足するように印加する）と共に，$I_{(3+2s)k_1+r}$ および $I_{(4+2s)k_1+r}$ $(1 \leq r \leq k_1; s = 0, 1, 2, \cdots)$ に，それぞれ，I_r および I_{2k_1+r} と同じ値を印加し，このときの応答を O_1〜O_N で観測する．

(2) 手続き (1.1) における LUT の出力関数を $f = \bar{x}_{m_t}$ に変更して手続き (1) を行う．但し，手続き (1.3) において制約 A を満足する必要はない．

TP-LUT$_{ALL}$ においては，故障が存在しなければすべての LUT に対してテストパターンが正しく印加されることになる．また，仮定 1 から，故障が存在する CLB の出力は，そのまま（あるいは反転されて）観測点となる右端の IO 端子まで伝搬する．よって，TP-LUT$_{ALL}$ を用いれば，すべての LUT を $2k$ 回のプログラムと $2k \times 2^k$ 個のテストパターンにより同時に検査することが可能である．

5.2 DFF および MUX の同時検査法

TP-LUT と同様に，TP-DFF においても，正常な場合の CLB の出力に生起する系列を次段の CLB の入力 c に与えてやることで，そのままテスト系列として利用できる．よって，同行にあるすべての CLB をカスケード接続することで，TP-DFF をすべての CLB に対して同時に実行可能となる．これは，DFF をカスケード接続することにより，シフトレジスタを構成して検査を行っていることに等しい．この様子を図 7 に示す．

また，TP-MUX における手続き (1) についても全く同様にして，すべての CLB に対して同時に実行できる．同手続き (2) および (3) についても，各行の CLB の出力に生起する系列を配線領域で分岐させ，その行における次段の CLB 内の LUT のすべての入力に共通に与えると共に，すべての CLB の入力 c を一つの左端の IO 端子に接続すれば，すべての CLB に対して同時に適用できることになる．

以上から，すべての DFF および MUX は，単独検査におけるプログラム回数（1 および 3）とテストパターン数（6 および 12）により検査できる．

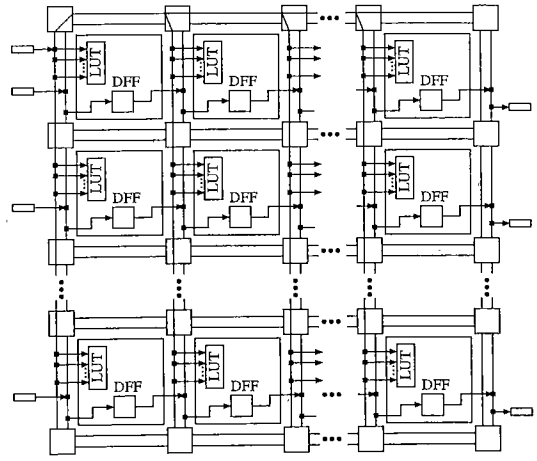


図 7 DFF の同時検査のための配線領域のプログラム
Fig. 7 Program of wiring segments for simultaneous test of all DFFs.

5.3 FPGA の所要検査時間

FPGA の検査は，プログラムとそれに対応するテストパターン（テスト系列）の印加とを繰り返すことにより，実行される．このときの所要検査時間 T_{CLB} は，次式で表される．

$$T_{CLB} = \tau_1 N^C_{CLB} + \tau_2 N^S_{CLB} \quad (1)$$

但し，N^C_{CLB}，N^S_{CLB}，τ_1 および τ_2 は，それぞれ，所要プログラム回数，テストパターンの総数，1 回のプログラムに要する時間および一つのテストパターンの印加に要する時間である．また，**5.1** と **5.2** の結果から，次の式 (2) および式 (3) が得られる．

$$N^C_{CLB} = 2k + 4 \quad (2)$$

$$N^S_{CLB} = 2k2^k + 18 \quad (3)$$

Xilinx の XC シリーズ [7] の場合，$k = 4 \sim 6$ 程度であり，かつ，τ_1 が τ_2 の数千～数万倍となるので，式 (1) は次式で近似できる．

$$T_{CLB} \simeq \tau_1 N^C_{CLB} = (2k + 4)\tau_1 \quad (4)$$

式 (2) から，FPGA におけるすべての CLB の検査に要するプログラム回数は $O(k)$ であり，N に依存しないことがわかる．また，式 (4) における τ_1 が $O(N^2)$ であることから，*Xilinx* の XC シリーズの場合の所要検査時間は，$O(kN^2)$ となる．本検査法を XC2064 [7] と同程度の仕様をもつ FPGA（$k = 4$, $\tau_1 = 120\,\text{ms}$, $\tau_2 = 50\,\text{ns}$）に適用した場合，所要検査時間は約 1.4 秒となる．

6. むすび

本論文では，テーブル参照型 FPGA を対象に，それを構成する一つの論理ブロックを単独で検査するときのプログラムとテストパターンを導出した．また，これを拡張して，すべての論理ブロックを同時に検査する方法を提案した．この結果，FPGA におけるすべての論理ブロックを検査するためのプログラム回数は，アレーサイズに依存せず，ルックアップテーブルの入力数のオーダとなることを明らかにした．

今後の課題は，FPGA における他の構成要素との同時検査について検討することが挙げられる．また，FPGA におけるプログラム機構の検査についても検討を行う必要があろう．

文　献

[1] S.D. Brown, R.J. Francis, J. Rose, and Z.G. Vranesic, "Field-programmable gate arrays," Kluwer Academic Publishers, 1992.

[2] S.M. Trimberger, "Field-programmable gate array technology," Kluwer Academic Publishers, 1994.

[3] 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, June 1994.

[4] 坪井秀幸，中田　広，宮崎敏明，"レジスタ挿入法を用いた FPGA 上の回路試験，" 信学技報，FTS94-53, pp.55–60, Oct. 1994.

[5] I. Pomeranz and S.M. Reddy, "Testability considerations in technology mapping," Proc. ATS '94, pp.151–156, Nov. 1994.

[6] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira, and T. Okamoto, "Universal Test Complexity of Field-Programmable Gate Arrays," Proc. ATS '95, pp.259–265, Nov. 1995.

[7] The Programmable Logic Data Book, Xilinx Inc., 1995.

[8] M.S. Abadir and H.K. Reghbati, "Function testing of semiconductor random access memories," Computing surveys, vol.15, no.3, pp.118–139, Sept. 1983.

[9] A. Tuszynski, "Memory testing," in T.W. Williams (ed.), VLSI testing, Elsevier Science Publishers, pp.161–228, 1986.

[10] 樹下行三，藤原秀雄，"ディジタル回路の故障診断（上），" 工学図書，1983.

（平成 8 年 3 月 11 日受付）

道西　博行　（正員）

平 1 岡山大・工・電子卒．平 5 同大大学院博士課程了．同年岡山大・工・情報助手．平 8 同大大学院自然科学研究科助手．テスト生成，VLSI の検査容易化設計に関する研究に従事．工博．情報処理学会会員．

横平　徳美　（正員）

昭 59 阪大・基礎工・情報卒．平 1 同大大学院博士課程了．同年岡山大・工・情報助手．平 2 同講師．平 6 同助教授．計算機・通信システムの性能評価，広帯域 ISDN における実時間通信，VLSI の検査容易化設計に関する研究に従事．工博．情報処理学会，IEEE 各会員．

岡本　卓爾　（正員）

昭33阪大・工・通信卒．同年川崎重工業（株）入社．昭35三井造船（株）転社．昭42岡山大・工・奉職．現在，同情報工学科教授．主に論理回路を中心とした計算機ハードウェアの研究に従事．工博．情報処理学会，電気学会，IEEE各会員．

井上　智生　（正員）

昭63明治大・工・電子通信卒．平2同大大学院博士前期課程了．同年松下電器産業（株）に入社，明治大大学院博士後期課程を経て，現在奈良先端科学技術大学院大学助手．松下電器産業（株）においてマイクロプロセッサの研究開発に従事．明治大，奈良先端大において，テスト生成，並列処理，テスト容易化設計の研究に従事．情報処理学会，IEEE各会員．

藤原　秀雄　（正員）

昭和44阪大・工・電子卒．昭49同大大学院博士課程了．同大・工・電子助手，明治大・工・電子通信助教授，情報科学教授を経て，現在奈良先端科学技術大学院大学教授．昭56ウォータールー大客員助教授．昭59マッギル大客員準教授．論理設計論，高信頼性設計，設計自動化，テスト容易化設計，テスト生成，並列処理，計算複雑度に関する研究に従事．著書「Logic Testing and Desing for Testability」(MIT Press)など．情報処理学会会員．IEEE Fellow.