

レジスタ挿入手法を用いたFPGA上の回路試験

坪井 秀幸[†] 中田 広^{††} 宮崎 敏明[†]

[†] NTT 光ネットワークシステム研究所

〒238-03 神奈川県横須賀市武 1-2356

e-mail: hideyuki@ntttsd.ntt.jp

^{††} NTT 通信網総合研究所

〒180 武蔵野市緑町 3-9-11

あらまし 本稿では、FPGAにプログラムされた回路の試験を効率的に行う方法を提案する。FPGAを用いたシステムにおいては回路実現部以外にプログラミング機構に起因した回路の故障が考えられる。そのため従来のASICを対象として単一縮退故障モデルを仮定した試験方法および試験パターン生成法を単純にFPGA上の回路に対して適用することでは不十分である。従って、FPGAを前提に置きプログラムされた回路を対象にした新たな試験方法が必要である。本稿で提案する試験方法はFPGA内で回路として使用している部分へレジスタを挿入し、試験パターン生成などを容易にすることができる。ここでは、本手法を説明するとともに実験結果を用いて試験に必要なテストパターン数が大幅に削減されることを示す。

和文キーワード FPGA、故障検出、試験パターン生成

Testing for Circuits realized as FPGAs using Register Insertion Method

Hideyuki Tsuboi[†], Hiroshi Nakada^{††}, Toshiaki Miyazaki[†]

[†] NTT Optical Network Systems Laboratories

1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 JAPAN

^{††} NTT Telecommunication Network Laboratory Group

3-9-11 Midori-chou, Musashino-shi, 180 JAPAN

Abstract This paper presents an efficient test technique for circuits realized as Field Programmable Gate Arrays(FPGAs). In the FPGA system there are two kinds of hardware fault, the defect in configuration circuit parts and the fault in programming organization. In this reason, the conventional test techniques that treat a single stuck-fault model of ASICs are not enough to be used for FPGAs. Thus, we propose the testing method that is combined with two techniques; register insertion and pseudoexhaustive test. Using this method, the test pattern is generated easily. The number of test pattern by this method is tremendously reduced.

英文 key words FPGA, fault detection, test pattern generation

1. まえがき

近年、プログラムにより回路を書き換えられる Felid Programmable Gate Array (FPGA) [1]がそれによって実現できる回路規模の拡大と性能の向上により頻繁に使われるようになってきた。さらに、このFPGAを多数装備して、全体の仕様を変更可能にするシステム設計の研究もなされている[2]。このようなシステムでは回路情報、すなわちプログラムデータをロードすること、つまりソフトウェアによる制御がなされており、故障は回路実現部以外にプログラミング機構にあると想定される。従って、製造時に検査を受けたFPGAであっても、これに実現される回路を書き換え時毎にチェックすることは欠かせない。

従来、ゲートレベルの故障検出を目的とした試験法や試験データの生成法の研究が特定の仕様を満たすASIC等を対象に数多く行われてきている[3]。それらは主として論理ゲートの入出力ポイントの値が固定される単一縮退故障モデルを仮定している。FPGAの場合は内部に小さな論理を実現する構造を多数持ち、この論理変更可能な部分の故障モデルはその入出力ポイントの値が固定される故障に集約されるとは考え難く、むしろ局所的に間違った論理状態に変更する故障と想定するのが自然である。よって、先の単一縮退モデルを仮定した試験方法をFPGAに適用することは難しい。

さらに、FPGAの部分的な欠陥をカバーする手法が提案されている[4][5]。この場合、故障検出・診断手段となる試験は不可欠である。この時適用される試験方法は製造時の欠陥を検査する方法であって仕様変更時に行うものではない。

以上より本稿では、FPGA上の回路に対する効果的な試験として局所的に間違った論理状態を検出するように部分的に全ての入力設定の場合を尽くす試験および試験ベクトルの生成方法を提案する。本方法はFPGA内で回路に使用されている一部分のみを試験対象とし、回路経路上にある未使用なレジスタを配線や論理部分を変更しないで活性化させ、全レジスタをスキャンすることを行う。これによりFPGAの構造上、論理変更可能な部分の出力を直接観測でき、試験データの生成が容易なる。レジスタ挿入はこの効果に加えてあらゆる入力設定を行うことが可能な可制御性が保証されるため、より試験データの生成を簡単にし、試験パターン数を少なくできる。

まず、本手法の説明を行った後、実験結果を示し、試験パタンの圧縮効果について考察する。

2. 対象とするFPGAの構造

今回、対象とするSRAM型の書き換え可能な構造を有するFPGAは多入力の任意の論理を実現する論理ブロック(LB)と配線からなる。各LBの出力部には信号を保持するレジスタがあり、これらをフル・スキャン可能とする構造を有する。またレジスタの使用・未使用もプログラムにより選択出来るものとする。以上のようなFPGAに回路が実現されている状態を図1に示す。

3. 提案手法

本手法は以下の2つのことを組み合わせて適用するものである。

- 1) レジスタ挿入
- 2) Pseudexhaustive test (PET)[3]

3. 1 レジスタ挿入手法

元回路の配線やLBの論理は変えず回路経路上のレジスタを全て使用するようにプログラムする。このレジスタをフル・スキャンする事で、FPGAにプログラムされた回路の全ての配線セグメントに必ず入出力可能なポイントができる。図1の回路では図2の様に新たに入出力ポイントが作られる。そして、回路の論理的な関係は図3の様になる。一見して分かるように全ての入力ポイントから出力まで高々1段のLBが介在するのみとなる。従って、各LBの出力は全て直接確認でき、また各LBに対する如何なる入力も網羅的に設定可能となる。つまりLBの出力部分は全て可観測な条件で試験を考えることができる。さらに、設定するテストデータの生成も個々のLB毎に考えられる。ここで、図1の11と記されたLBの入力側の配線について考える。図4に示すように回路の経路設定には3つの配線スイッチA,B,Cが関与している。2つの配線を接続している点Aのテストでは”オフ”状態の故障を見つけることが出来ればよく、入力aの値を変えることで”オフ”状態の

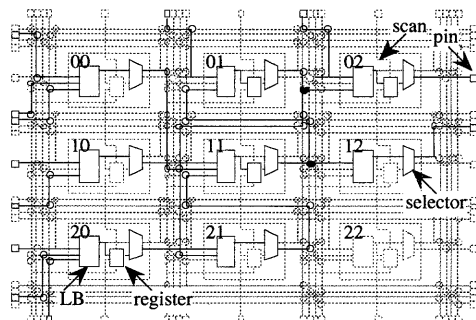


図1. FPGAと回路使用部 (実線)

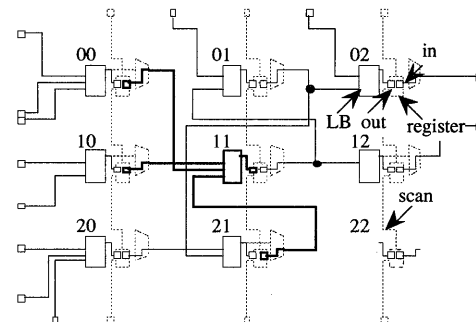


図2. レジスタ挿入と入出力ポイント

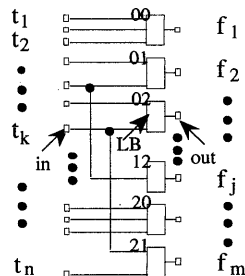


図3. レジスタ挿入後の回路

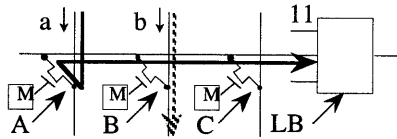


図4. 配線上のスイッチの状態

```

 $a_k = \text{row vector } k \text{ of } D(N);$ 
for (  $k = 1; k \leq n - 1; k++$  ) {
  for (  $l = k + 1; l \leq n; l++$  ) {
    if (  $a_k \cdot a_l = 0$  ) {
       $a_k = a_k + a_l;$ 
      for (  $X = l; X \leq n - 1; X++$  ) {
         $a_X = a_{X+1};$ 
      }  $n = n - 1;$ 
    }
  }
}
RDP(N) = D(N),  $p = n;$ 

```

図5. $D(N)$ から $RDP(N)$ を得る手順

故障を見つけられる。異なる信号がすれ違っている点Bでは”オン”状態の故障を見つけられればよい。それには、LB11を試験している時に、それらの入力以外、bを含み全ての設定値を”0”と”1”状態にした場合をチェックすればよい。C点では、”オン/オフ”状態の故障により、この回路の出力への影響はない。従って、Exhaustive test (ET) [6]の2倍のテストパターンの試験を行うことで配線領域のスイッチの”オン/オフ”状態の故障もチェックできる。またレジスタ挿入により、次の関連する効果が得られる。

- ・簡便なレジスタ間への故障範囲の特定
- ・PETにおける試験パターン数の大幅な削減

以降、2番目の試験パターン数の削減効果について詳しく述べる。

3. 2 PET試験データの生成

J.McCluskeyがPET試験データの生成の検討を行っている[6]。図3の様な回路の入力と出力に関する行列 $D(N)$ は

$$D(N) = \begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1m} \\ \vdots & & \vdots & & \vdots \\ a_{k1} & \dots & a_{kj} & \dots & a_{kn} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nm} \end{pmatrix} \quad \dots(1)$$

$$a_{kj} = \begin{cases} 0 & (\text{if } f_j \notin \text{propa}(t_k)) \\ 1 & (\text{if } f_j \in \text{propa}(t_k)) \end{cases} \quad \dots(2)$$

ここで、 $\text{propa}(t_k)$ は、入力 t_k から信号が伝播する全ての出力を表す集合である。

ある出力が最も多数の入力に依存しているときのその数、 w を次のように定義し、それをサポート数と呼ぶ。

$$w = \max_{\text{for all } k} \sum_{j=1}^m a_{kj} \quad \dots(3)$$

このときの任意の k 行と l 行に対応するベクトル a_k と a_l との内積が”0” ($a_k \cdot a_l = 0$) ならば、 k 行をその2行の和とし

表1. PETでの試験パターン数

w	$U(p,w)$	p	Number of Tests (rows)
2	$p[0,p-1] \quad p[1,p]$	> 3	$p+1C1$
3	$p[1,p-1]$	> 4	$2pC1$
4	$p[1,p-2] \quad p[2,p-1]$	> 5	$p+1C2$
5	$p[2,p-2]$	> 6	$2pC2$
6	$p[2,p-3] \quad p[3,p-2]$	> 8	$p+1C3$
7	$p[3,p-3]$	> 9	$2pC3$
8	$p[3,p-4] \quad p[4,p-3]$	> 11	$p+1C4$
9	$p[4,p-4]$	> 12	$2pC4$
10	$p[4,p-5] \quad p[5,p-4]$	> 14	$p+1C5$

$w = 4, p = 6 \quad p[1,p-2]$

1	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
0	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
0	0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	1
p	0	0	0	1	0	1	1	0	1	1	0	1	0	1	0	0
0	0	0	0	1	0	1	1	0	1	1	0	1	0	1	0	1
0	0	0	0	0	1	1	1	1	0	1	1	0	1	0	1	0

Number of Test Patterns : $p+1C2 = 21$

図6. PETの試験データの一例

て ($a_k = a_k + a_l$)、1行は削除する。この操作を全ての行に対して繰り返し施し、どの行も内積が”0”でなくなれば、その時の $D(N)$ を $RDP(N)$ とする。

$$RDP(N) = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_p \end{pmatrix} \quad \begin{matrix} a_k \cdot a_l \neq 0 \\ k, l = 1, 2, \dots, p \\ k \neq l \end{matrix} \quad \dots(4)$$

$$a_k = (a_{k1} \dots a_{kj} \dots a_{kn}) \quad \dots(5)$$

ただし、どの2行をどの順序でその操作を行うかは様々な手順が考えられる。一例を図5に示す。この例の操作によって必ずしも最小の $RDP(N)$ が常に求まるとは限らない。

しかしながら、行列 $RDP(N)$ の行数 p は必ず先に確認したサポート数 w 以上の値となるため、 p が w の近傍値ならば、先の操作が適切であると考えられる。この時の行列 $RDP(N)$ の行数 p と w からPETに必要とされるテストパターン数は表1のように求められている[6]。さらにテストデータについても表1の $U(p,w)$ から機械的に作ることができる。そのテストデータの1つを図6に示す。

3. 3 レジスタ挿入とPETを組み合わせる例

図7に示す様な回路においてETおよびPET試験データを生成する。比較のために、まずレジスタ挿入を行っていない場合について述べる。図7の右側に出力 f_j に信号が伝播する入力 t_k を示す表を挙げた。ここで、各出力 f_j に対応した入力、サポートの数を w_j とすればETのパターン数は次式で求められる。ここで j は出力の順を示す。

$$ET = \sum_{j=1}^m 2^{w_j} \quad (j = 1, 2, \dots, m) \quad \dots(6)$$

この例では $ET=30$ となる。

次に表2に $D(N)$ を示す。先に行列で示していた $D(N)$ 、 $RDP(N)$ をここでは表にして示す。入力数 n のとき、PET1は最大サポート数 w に応じて $p=n$ 生成できる。生成法は表1および

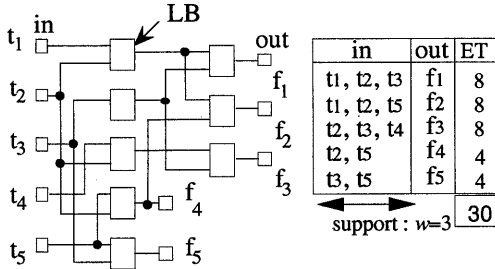


図7. レジスタ挿入前

表2. PET1 data (レジスタ挿入前)

in	out	in	PET1
t1	f1, f2	t1	1 0 0 0 0 0 1 1 1 1
t2	f1, f2, f3, f4	t2	0 1 0 0 0 0 1 0 1 1 1
t3	f1, f3, f5	t3	0 0 1 0 0 1 1 1 0 1 1
t4	f3	t4	0 0 0 1 0 1 1 1 1 0 1
t5	f2, f4, f5	t5	0 0 0 0 1 1 1 1 1 1 0

= 10

表3. RDP(N), PET2 data (レジスタ挿入前)

in	out	in	PET2
t1, t4	f1, f2, f3	t1, t4	1 0 0 0 0 1 1 1 1
t2	f1, f2, f3, f4	t2	0 1 0 0 1 0 1 1 1
t3	f1, f3, f5	t3	0 0 1 0 1 1 1 0 1
t5	f2, f4, f5	t5	0 0 0 1 1 1 1 1 0

= 8

図6に従う。PET1=10となる。

さらにPET2の生成は先のD(N)を各行について出力が全て異なるかを調べ、異なっていれば一行にまとめ、表3の様にRDP(N)を求める。圧縮された入力数に対しPET1と同様の方法でPET2を求める。PET2=8である。

次にレジスタ挿入を行う場合について考察する。回路および出力f_jに信号が伝播する入力t_kの関係は図8の様になる(ET=30)。またD(N)は表7の様になる(PET1=10)。次はD(N)からRDP(N)を求める。行間の圧縮の手法は一意に決まった方法はないので、ここでは表5に示す1行目から最終行まで順に圧縮できる(出力が全く重複しない)ものはまとめてしまう方法1と、表6に示す圧縮する前に出力の数が多い順に並べ変えて、その多い順に圧縮を行う方法2を試した。並べ変えてから圧縮をした方が圧縮結果は良い。しかし圧縮操作に並べ替え操作が加わるため、この並べ替えにかかる処理でRDP(N)を求めるのに必要な時間が多くなる。この所用時間の増加を避けて、方法1で得た結果を用いても十分な場合が多い。この例においても双方ともPET2=4という結果が得られる。

4. 実験および考察

4.1 実験

図9にFPGAを用いたシステム設計の流れと今回の試験データの作成および試験方法の流れを示す。動作記述によって与えられた回路仕様が最終的にハードウェアであるFPGAにプログラムされる途中段階で試験データの生成に必要な情報、すなわちLBの割り当てとその間の結線の情報を得る。

これに全てのLBの出力部にレジスタを挿入する操作とPET試験データ生成の操作を行った。ここでLBの入力数すなわち

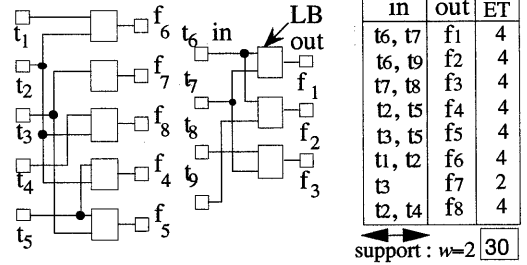


図8. レジスタ挿入後

表4. PET1 data (レジスタ挿入後)

in	out	in	PET1
t1	f6	t1	1 1 0 0 0 0 0 0 0 0 0
t2	f4, f6, f8	t2	1 0 1 0 0 0 0 0 0 0 0
t3	f5, f7	t3	1 0 0 1 0 0 0 0 0 0 0
t4	f8	t4	1 0 0 0 1 0 0 0 0 0 0
t5	f4, f5	t5	1 0 0 0 0 1 0 0 0 0 0
t6	f1, f2	t6	1 0 0 0 0 0 1 0 0 0 0
t7	f1, f3	t7	1 0 0 0 0 0 0 1 0 0 0
t8	f3	t8	1 0 0 0 0 0 0 0 1 0 0
t9	f2	t9	1 0 0 0 0 0 0 0 0 0 1

= 10

表5. RDP(N), PET2 data (方法1)

in	out
t1, t3, t4, t6, t8	f1, f2, f3, f5, f6, f7, f8
t2, t7, t9	f1, f2, f3, f4, f6, f8
t5	f4, f5

in	PET2
t1, t3, t4, t6, t8	1 1 0 0
t2, t7, t9	1 0 1 0
t5	1 0 0 1

= 4

表6. RDP(N), PET2 data (方法2)

in	out
t2, t3, t7, t9	f1, f2, f3, f4, f5, f6, f7, f8
t1, t4, t5, t6, t8	f1, f2, f3, f4, f5, f6, f8

in	PET2
t2, t3, t7, t9	0 0 1 1
t1, t4, t5, t6, t8	0 1 0 1

= 4

サポート数wを3とした。比較のために以下に示す異なるデータも生成・検討した。

- (1) レジスタ挿入を行わない場合
- (2) マッピングを行わない場合
- (3) レジスタにクロック・イネーブル機能が付いている場合

ここで、(1)は元の回路にレジスタがある部分のみを入出力端子とすることで、組み合わせ回路(CC)を扱い、ET試験データ生成の操作を行う。また、(2)は論理合成の際の仮想セルのままの状態、仮想セルをLBに見立てて、全セルの出力にレジスタを追加させることでレジスタ挿入と同じ効果を狙った。ただし3入力以下の仮想セルを使用した。これによりマッピング手法に無関係な単にレジスタ挿入する効果を調べた。最後の(3)は元の3入力にクロック・イネーブル信号の入力とレジスタに保持される値を加えて、LBの入力、w=5として対処した。反対にクロック・イネーブルがない場合にク

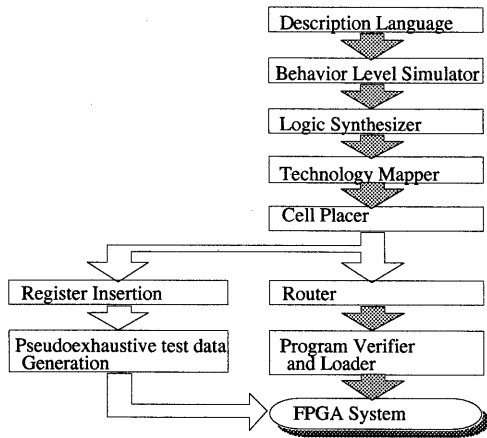


図9. FPGAのシステム設計と試験

表7. 各方法の条件

レジスタ	レジスタ挿入	なし	あり
クロック・イネーブルあり	方法.A	方法.C	方法.B
クロック・イネーブルなし	方法.C	方法.C	方法.D

表8. マッピングを行わない場合の試験パターン数

回路名	レジスタ				CC ET	レジスタ挿入手法						
	gates	input	output	register		input	cell	ET	PET1	PET2	p1	p2
au4gen	307	13	8	13	2 ²⁴	105	150	1200	210	8	3	3
stm1gen	1696	12	40	92	2 ¹⁰²	558	727	5816	1116	8	4	3
vc4gen	335	13	10	13	2 ²⁴	121	179	1432	242	8	4	3
sigg	2576	15	40	131	2 ¹⁴⁴	823	1164	9312	1646	8	4	3
au4t	1564	21	3	73	2 ⁹²	550	800	6400	1100	8	4	3
stm1t	3063	12	31	183	2 ¹⁹³	871	1462	11696	1742	8	4	3
vc4t	292	13	10	13	2 ²⁴	99	147	1176	198	8	4	3
sigt	4571	12	31	246	2 ²⁵⁶	1375	2248	17984	2750	8	4	3
fsync	213	13	6	8	2 ¹⁹	102	141	1128	204	8	3	3
fcount	254	5	13	13	2 ¹⁶	72	123	984	144	8	4	3
osync	329	12	8	18	2 ²⁸	139	156	1248	278	8	4	3
vc32det	3922	17	17	174	2 ¹⁸⁹	1391	1953	15624	2782	8	4	3

表9. マッピングを行ったときの試験パターン数

回路名	A, B, C, D		方法.A		方法.B		support:w=5			方法.D			support:w=3
	LBs	input	PET2	PET2	input	ET	PET1	PET2	input	ET	PET1	PET2	
au4gen	63	70	24	2 ¹⁶	74	2016	5402	32	81	560	162	8	
stm1gen	300	369	102	2 ²⁸	310	9600	95790	42	379	2952	758	8	
vc4gen	75	86	24	2 ¹⁶	86	2400	7310	42	96	688	194	10	
sigg	466	539	144	2 ⁶⁹	506	14912	228962	42	552	4312	1104	10	
au4t	356	385	84	---	367	11392	134322	72	396	3080	798	14	
stm1t	482	613	193	2 ³⁰	492	15424	241572	56	623	4904	1246	10	
vc4t	59	70	24	2 ¹⁶	70	2240	4830	42	81	560	162	10	
sigt	874	1025	256	---	884	27968	780572	56	1035	8200	2070	12	
fsync	75	75	19	2 ¹³	86	2400	7310	42	86	600	172	8	
fcount	43	53	16	2 ¹⁶	46	1376	2070	42	56	424	112	10	
osync	104	106	28	2 ¹²	114	3328	12882	32	114	848	228	8	
vc32det	986	1016	189	---	1003	31552	1001000	42	1031	8128	2062	10	

ロック・イネーブル付きのレジスタが回路にあるところにはLBを複数使い、フィードバックの配線とセクタを加える形でクロック・イネーブル機能をもつレジスタを実現した。

4. 2 結果および考察

表8にマッピングを行わない場合の結果を示す。レジスタ挿入を行わないCCではETのパタン数、レジスタを全セル出力に追加するレジスタ挿入ではセル数、ET、PETのテストパタン数PET1およびPET2を挙げた。ここでPET1は3.3節で説明した圧縮操作を行わず、入力端子から直接求め、PET2は圧縮操作を行い求めた。PET2において3.3節で説明した行列を圧縮する2通りの方法1と方法2でpの値を求めたが(表7のp1、p2)、両者圧縮の差は小さい。結果、両者のパタン数は同じとなった。

また、表9はマッピングを行った結果を示す。この中で方法A、B、CとDはクロック・イネーブルの使用、レジスタ挿入の有無の違いを表7のように決めている。ただし、レジスタ挿入をしない場合の一部データは有効時間内に求められず、表には---で示している。

まずレジスタ挿入による試験パタン数の抑制効果を調べるため、レジスタ挿入後のパタン数と挿入しないときとを比較した。レジスタ挿入を行わない場合の試験データとしてCCのETおよびFPGAにマッピングされた方法A、CのPET2とレジスタ

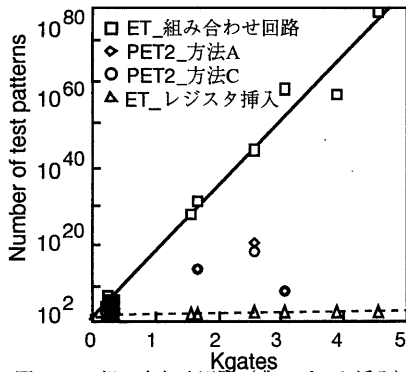


図10. 組み合わせ回路 (非レジスタ挿入)

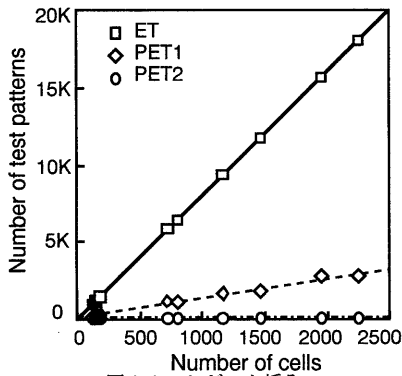


図11. レジスタ挿入

挿入後のETの結果を図10に示す。方法.A/方法.C (レジスタのクロックイネーブルの有無)によりPET2の値にあまり差はない。レジスタ挿入を行わない場合はパターン数が回路規模に対し急激に増大する。レジスタ挿入後のETはCCのETの値やその増加の程度より小さいPET2と比べ、共に小さく、レジスタ挿入により試験パターン数増加の抑制効果が明確に分かる。

次に3入力以下のセルのみで回路合成し、全セルの出力側にレジスタを加えた形のレジスタ挿入を行った場合、試験データの作成の違いを調べるために各生成法による試験パターン数の比較を行った。図11にET/PET1/PET2の比較結果を示す。この場合ET/PET1/PET2の順にその値は小さく、特にPET2では圧縮効果が大きく働いているといえる。またセル数の増加に対してETがPET1より常に上回って増えていくが、PET2の増加は見られない。従って、PET2を用いた試験は大規模回路にも適用可能であると考えられる。

最後にFPGAにマッピング後のレジスタ挿入場合の試験パターンを調べるために各生成方法でパターン数を求めた。クロック・イネーブル付きレジスタを有する結果を図12に示す。イネーブル信号は先に説明したように $w=5$ として扱った。PET1はLBの数に対し二次関数的に、ETはリニアに増加している。またPET2はETより一層少ない増加に留まっている。

一方、レジスタにクロック・イネーブルがない場合を図13に示した。PET2はクロック・イネーブルがある場合よりもさらに少ないパターン数に抑えられ、回路規模に対する増加は極めて小さくなっている。

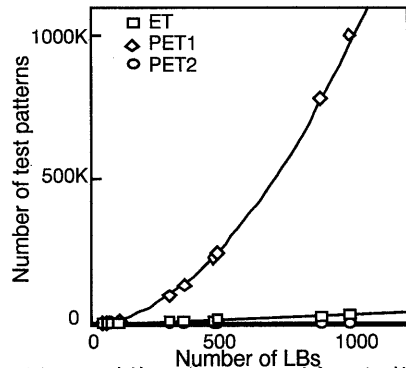


図12. 方法.B (クロック・イネーブル付き)

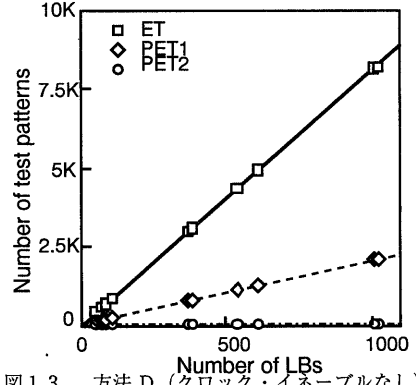


図13. 方法.D (クロック・イネーブルなし)

5. まとめ

FPGA上にプログラムされた回路の試験方法として、レジスタ挿入とPETを組み合わせることを提案し、以下の結果を得た。

- ・レジスタ挿入およびスキャンパスを活用することで被試験回路が論理段数一段の部分回路に置き換えられ、試験生成が容易となった。
- ・各論理ブロックの出力と入力関係の重複が減少するために、試験を並行的に行うことが可能となり、試験パターンの大幅な圧縮が図られた。

参考文献

- [1] S.D.Brown, et al.: "Field-Programmable Gate Arrays", Kluwer Academic Publishers, 1992.
- [2] K.Yamada, et al.: "High-Speed Emulation of Communication Circuits on a Multiple-FPGA System", ACM International Workshop on FPGAs, Berkeley California USA., Feb. 1994.
- [3] M.Abramovic, et al.: "Digital Systems Testing and Testable Design", Computer Science Press, N.Y., 1990.
- [4] N.J.Howard, et al.: "The Yield Enhancement of Field-Programmable Gate Arrays", IEEE Trans. VLSI, vol.2, no.1, Mar. 1994.
- [5] J.L.Kelly, et al.: "A Novel Approach to Defect Tolerant Design for SRAM Based FPGAs", ACM International Workshop on FPGAs, Berkeley California USA., Feb. 1994.
- [6] E.J.McCluskey: "Verification Testing - A Pseudoexhaustive Test Technique", IEEE Trans. Comput., vol.c-33, no.6, pp.541-546, Jun. 1984.