マルチサイクルキャプチャテストを用いた フルスキャン設計回路のテスト生成法

日大生産工(院)　○大森　悠翔　　日大生産工　細川　利典
明大　山崎　浩二　　九大　吉村　正義

1　はじめに

半導体微細化技術の進歩のため，VLSIが大規模化，複雑化し，VLSIのテスト設計技術が重要になっている．これまで，VLSIのテスト方法として，縮退故障を対象としたスキャンテスト[1)2)]が，広く普及しているが，近年そのテストでは不良と判定できない欠陥VLSIが存在し，テスト品質を向上させるために，ディレイテスト[3)]や実動作速度機能テスト[4)]が必要であるということが報告されている．スキャンテストは回路構造の情報のみを利用したテスト法で，シフト動作によって，回路を無効状態に遷移させてテストする場合がある．したがって，スキャンテストはオーバテストを行っていると考えられ，そのため歩留まり損失が発生する可能性がある．そこで順序回路動作で故障を検出するようなテスト生成を行うことにより，できるだけオーバテストを防ぎ，歩留まり損失を抑制できると考えられる．

本稿では，スキャンテストのキャプチャモード時のサイクル数がkであるようなkサイクルキャプチャテストを提案する．また，kサイクルキャプチャテストに伴い順序回路動作で故障を検出するkサイクルキャプチャ縮退故障テスト生成法を提案する．

2　kサイクルキャプチャテスト

（定義１：kサイクルキャプチャテスト）

縮退故障や遅延故障を対象としたスキャンテストでは，キャプチャモード時のサイクル数は1サイクルまたは，2サイクルである．kサイクルキャプチャテストとは，キャプチャモード時のサイクル数がkであるようなスキャンテストである．k≧２のとき，マルチサイクルキャプチャテストという．

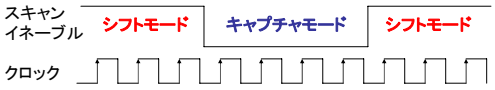

図１．4サイクルキャプチャテストの例

図1に4サイクルキャプチャテスト（k=4）の例を示す．kサイクルキャプチャテストを行うことによって，kサイクル間順序回路動作をさせるので，回路を無効状態に遷移させてテストする回数が減り，オーバテストを抑制できると考えられる．

3　kサイクルキャプチャ縮退故障テスト生成

3.1　諸定義

（定義２：kサイクルキャプチャ縮退故障テスト生成）

kサイクルキャプチャ縮退故障テスト生成とは，kサイクルキャプチャテストで縮退故障を検出するためのテスト生成である．

kサイクルキャプチャ縮退故障テスト生成の対象とする回路はフルスキャン設計回路やパーシャルスキャン設計回路であり，k時間展開モデル[4)]を用いてテスト生成を行う．また，k時間展開モデルにおいて，故障モデルは多重縮退故障に変換される．

生成されるテスト系列は，シフトインするパターンと長さkの外部入力系列とシフトアウトするパターンである．

3.2　k時間展開モデル

（定義３：k時間展開モデル）

時刻1でスキャンフリップフロップの出力を擬似外部入力とし，時刻kでスキャンフリップフロップのデータ入力を擬似外部出力としてk時間分回路を展開した組合せ回路である．

Test Generation Method for Full Scan Circuit Using Multi Cycle Capture Test

Yusho OMORI, Toshinori HOSOKAWA,
Koji YAMAZAKI and Masayoshi YOSHIMURA

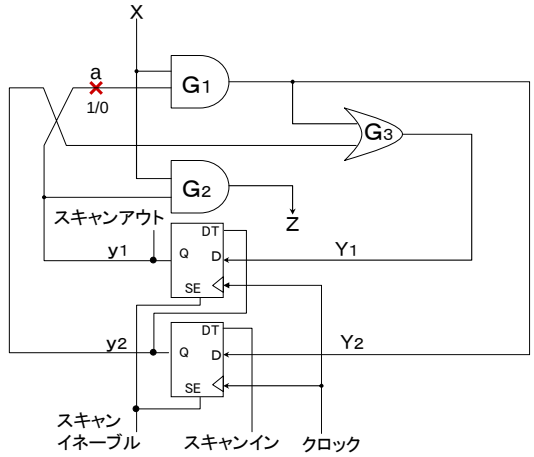

図２．フルスキャン回路例

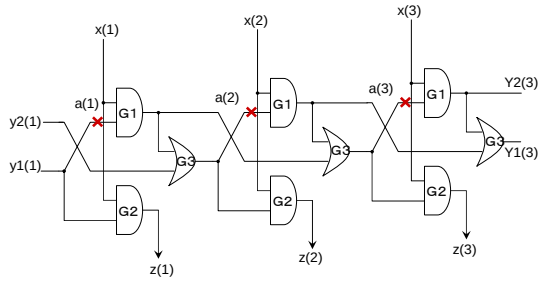

図３．時間展開モデル例（k=3）

本テスト生成法では，図2のようなフルスキャン回路を図3のようにk時間展開（k=3）した時間展開モデルを生成しテスト生成を行う．図3は3時間展開の例である．x(i)は時刻iにおける外部入力を表し，z(i)は時刻iにおける外部出力を表している．y1(1)，y2(1)は擬似外部入力を表し，Y1(3)，Y2(3)は擬似外部出力を表している．

3.3　テスト生成

kサイクルキャプチャ縮退故障テスト生成では，展開した回路モデルの各時刻に故障が表れる．例えば，図2の信号線aの0縮退故障を仮定すると，3時間展開モデルでは図3の信号線a(1～3)の各時刻において故障の影響を考えなければならない．

以下に本テスト生成アルゴリズムを示す．
(t:現在時刻，k:時間展開数，z:故障信号線)
(step1)
　信号線zの故障を選択する．バックトラック回数を0にする．t=k+1.
(step2)
　t=t-1．信号線zの時刻tの故障を励起し，外部出力か擬似外部出力まで伝搬する．未正当化信号線がある場合は正当化を行う．
(step3)
　信号線zの時刻tの故障を検出するか，バックトラックリミットが上限を超えるか，tが1ならstep4へ．それ以外はstep2へ．
(step4)
　tが1かつ全時刻で信号線zの故障を検出できない場合は順序冗長と判定する．バックトラックリミットが上限を超えた場合は打切り故障と判定する．まだテスト生成を行っていない信号線があるならstep1へ．全信号線のテスト生成を行えば終了．

信号線aの0縮退故障のテスト生成例を上記アルゴリズムと図3を用いて示す．step1とstep2より信号線a(3)の0縮退故障を励起することを考える．信号線a(3)に正常値1故障値0を割当てる．a(3)の故障の影響を伝搬するために活性化操作を行い，x(3)に論理値1を割当てる．含意操作を行うと故障の影響が擬似外部出力Y2(3)に伝搬する．次に，未正当化信号線a(3)の正常値1の正当化を行う．a(3)に論理値1を割当てるために後方追跡を行いx(1)に論理値1を割当てる．含意操作を行なってもa(3)は未正当化信号線のままなので，再度後方追跡を行い，擬似外部入力y1(1)に論理値1を割当てる．含意操作を行うと，未正当化信号線がなくなり信号線aの0縮退故障のテスト生成を終了する．step3より故障を検出できているのでstep4へ．step4より，まだテスト生成を行っていない信号線に対してテストを行うのだが，例はここで終了する．

4　順序冗長故障の判定

kサイクルキャプチャ縮退故障テスト生成では，展開時間kを大きくするほどテスト生成対象とする回路モデルの規模が大きくなる．すると，故障の伝搬や値の正当化が不可能になる信号線が増え，順序冗長が増えると考えられる．従来の１サイクル・２サイクルキャプチャのスキャンテストでは，回路を無効状態に遷移させてテストしているためオーバテストを行っている割合が高いと考えられる．一方，kサイクルキャプチャテストでは従来の方法にくらべて展開時間数を大きくするほど順序冗長故障を判定することができるためオーバテストを抑制することができるといえる．

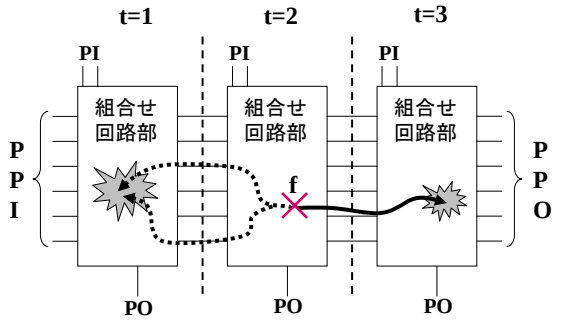

図４．故障伝搬と正当化の不可能例

図4に故障伝搬と正当化が不可能な例を示す．図4は順序回路を3時間展開した回路モデルである．各時刻の組合せ回路部は組合せ回路ブロックとして表現する. 例として2時刻目の信号線fの故障を伝搬することを考える．fの故障の影響を観測しようとすると故障の影響が3時刻目に伝搬されると仮定する．ここで, 3時刻目の故障の影響を外部出力か擬似外部出力に伝搬させたいが, 3時刻目の組合せ回路部内での故障伝搬が不可能になることが考えられる．さらに，例としてfの正当化を考える．fの励起や故障の活性化の影響が1時刻目に現れると仮定する. その結果1時刻目の組合せ回路部内で値の衝突が起こる可能性が考えられる．このように通常のフルスキャンテストでは擬似外部出力で故障の影響を観測できる故障もkサイクルキャプチャテストでは故障伝搬が不可能になることが考えられる．

同様に，通常のフルスキャンテストでは擬似外部入力に値の割当てができるが, kサイクルキャプチャテストでは値衝突が発生し正当化が不可能になることが考えられる．

５　実験結果

kサイクルキャプチャ縮退故障テスト生成を実装し, ISCAS’89ベンチマーク回路に対して実験1と実験2を行った．実験は故障シミュレーションを実行せず，バックトラックリミット100回で行った．

実装はC言語で行った．マシンのOSはFedora7，CPUは3.0GHz，メモリは2GBである．

以下に実験1と実験2の実験方法を示す．

(実験1)

時間展開数の1(k=1)テスト生成を行う．この実験は通常のフルスキャンテストのことである．

(実験2)

時間展開数5(k=5)のテスト生成を行う．

以下に実験2のアルゴリズムを示す．

(k:時間展開数，U:故障集合)

(step1)

時間展開数1で冗長判定を行う. その結果非冗長な故障を故障集合Uとする．

表１．実験1と実験2のテスト生成結果

回路名	実験1								実験2							
	代表故障数	検出数	打切り数	冗長数	パターン数	検出効率(%)	検出率(%)	実行時間(秒)	代表故障数	検出数	打切り数	冗長数	パターン数	検出効率(%)	検出率(%)	実行時間(秒)
s27	29	29	0	0	29	100.00	100.00	0.00	29	29	0	0	29	100.00	100.00	0.00
s208	199	199	0	0	199	100.00	100.00	0.04	199	148	0	51	148	100.00	74.37	0.40
s298	282	282	0	0	282	100.00	100.00	0.08	282	281	0	1	281	100.00	99.65	2.06
s344	328	328	0	0	328	100.00	100.00	0.08	328	325	0	3	325	100.00	99.09	0.71
s349	336	334	0	2	334	100.00	99.40	0.08	336	331	0	5	331	100.00	98.51	0.74
s382	369	369	0	0	369	100.00	100.00	0.09	369	369	0	0	369	100.00	100.00	3.11
s386	372	372	0	0	372	100.00	100.00	0.18	372	339	0	33	339	100.00	91.13	11.05
s400	394	388	0	6	388	100.00	98.48	0.10	394	388	0	6	388	100.00	98.48	3.70
s420	398	398	0	0	398	100.00	100.00	0.10	398	222	0	176	222	100.00	55.78	3.19
s444	444	430	0	14	430	100.00	96.85	0.13	444	430	0	14	430	100.00	96.85	2.66
s510	552	552	0	0	552	100.00	100.00	0.30	552	552	0	0	552	100.00	100.00	9.67
s526	515	514	0	1	514	100.00	99.81	0.16	515	514	0	1	514	100.00	99.81	7.15
s641	463	463	0	0	463	100.00	100.00	0.10	463	463	0	0	463	100.00	100.00	0.46
s713	577	539	0	38	539	100.00	93.41	0.13	577	539	0	38	539	100.00	93.41	1.01
s820	840	840	0	0	840	100.00	100.00	0.71	840	826	0	14	826	100.00	98.33	66.06
s832	860	846	0	14	846	100.00	98.37	0.75	860	832	0	28	832	100.00	96.74	71.77
s838	793	793	0	0	793	100.00	100.00	0.31	793	402	0	391	402	100.00	50.69	14.37
s953	1021	1021	0	0	1021	100.00	100.00	0.80	1021	1018	0	3	1018	100.00	99.71	22.34
s1196	1224	1224	0	0	1224	100.00	100.00	1.08	1224	1224	0	0	1224	100.00	100.00	2.03
s1238	1336	1267	0	69	1267	100.00	94.84	1.35	1336	1267	0	69	1267	100.00	94.84	2.26
s1423	1388	1374	0	14	1374	100.00	98.99	0.53	1388	1374	0	14	1374	100.00	98.99	19.16
s1488	1474	1474	0	0	1474	100.00	100.00	2.27	1474	1451	0	23	1451	100.00	98.44	168.92
s1494	1494	1482	0	12	1482	100.00	99.20	2.31	1494	1459	0	35	1459	100.00	97.66	170.48
s5378	4361	4321	0	40	4321	100.00	99.08	2.79	4361	4015	0	346	4015	100.00	92.07	75.83
s35932	35638	31654	0	3984	31654	100.00	88.82	329.10	35638	31654	0	3984	31654	100.00	88.82	***
s38417	29770	29605	0	165	29605	100.00	99.45	45.48	29770	29605	0	165	29605	100.00	99.45	***
s38584	33537	32032	0	1505	32032	100.00	95.51	268.02	33537	32012	0	1525	32012	100.00	95.45	***

表２．オーバテスト解析

回路名	実験1 検出数	冗長差	オーバテスト 削減率(%)
s208	199	51	25.63
s298	282	1	0.35
s344	328	3	0.91
s349	334	3	0.90
s386	372	33	8.87
s420	398	176	44.22
s820	840	14	1.67
s832	846	14	1.65
s838	793	391	49.31
s953	1021	3	0.29
s1488	1474	23	1.56
s1494	1482	23	1.55
s5378	4321	306	7.08
s38584	32032	20	0.06

(step2)
故障集合Uに対して時間展開数kでテスト生成を行う．その結果打切りとなった故障を故障集合Uとする．
(step3)
k=k-1．kが1以上ならstep2へ．kが0なら終了．

表1に実験1と実験2のテスト生成結果を示す．表1は，実験1と実験2を各回路に対して行った際の代表故障数，検出故障数，打切り故障数，冗長故障数，テストパターン数，故障検出効率，故障検出率，テスト生成時間を示している．

表1の結果をみると，実験1と実験2の冗長故障数に差があることがわかる．ここで，冗長故障数に差があったs208，s298，s344，s349，s386，s420，s820，s832，s838，s953，s1488，s1494，s5378，s38584の回路に注目する．

表2に冗長故障数に差があった回路のオーバテスト解析の結果を示す. 表2の各列は回路名，実験1の検出故障数，冗長差，オーバテスト削減率を示している. 冗長差とは実験1と実験2の冗長故障の差のことである．（冗長差=実験2の冗長故障数 - 実験1の冗長故障数）オーバテスト削減率とは実験1の検出故障に対する，実験2が実験1に比べて削減したオーバテストの割合のことである．（オーバテスト削減率=冗長差 / 実験1の検出故障数）

結果から実験1に比べて実験２では1%～50%程度オーバテストを削減できていることがわかる．つまり，通常のフルスキャンテストに対してkサイクルキャプチャテストを行うことでオーバテストを削減できることがわかる．

また，テストパターン数について考察してみると，実験2は実験1よりも冗長故障数が多いためテストパターン数はその分少なくなる．

６ おわりに

本研究では, kサイクルキャプチャテストとkサイクルキャプチャ縮退故障テスト生成法を提案した．kサイクルキャプチャテストは，kを大きくする程順序冗長故障を判定することができるため，オーバテストを抑制できると考えられる．

いくつかの回路で実験1に対して実験2の冗長故障が増え，通常のスキャンテストのオーバテストを抑制できることが確認できた．また，テストパターン数についても削減できることが確認できた．

今後の課題として，テスト生成アルゴリズムを改善し，順序冗長をより多く判定することや順序回路故障シミュレータの実装などが挙げられる．

「参考文献」

1) H. Fujiwara, “Logic Testing and Design for Testability,” The MIT Press, 1985.

2) M. Abramovici, M. A. Breuer, and A. D. Friedman, “Digital systems testing and testable design,” IEEE Press, 1995.

3) A. Krstic, and K.-T. Cheng, “Delay Fault Testing for VLSI Circuits,” Kluwer Academic Publishers, 1998.

4) T.Inoue, T.Hosokawa, T.Mihara and H.Fujiwara, “An Optimal Time Expantion Model Based on Combinational ATPG for TR Level Circuits”, IEEE Asian Test Symp,1998, pp.190-197.

5) P.C. Maxwell, R.C. Aitken, R. Kollitz, and A. C. Brown, “IDDQ and AC Scan: The War Against Unmodelled Defects” , Proc. of IEEE Int. Test Conf. , 1996 , pp.250-258.

6) I. Pomeranz and S. M. Reddy, “A Postprocessing Procedure to Reduce the Number of Different Test Lengths in a Test Set for Scan Circuits”, in Proc. 10th Asian Test Symposium, 2001 , pp.131 – 136.

7) J. Abraham, U. Goel and A. Kumar, “Multi-Cycle Sensitizable Transition Delay Faults”, in Proc. 24th IEEE VLSI Test Symposium, 2006, pp.306 – 313.