

パーシャル Max-SAT を用いた低消費電力指向 X 判定・割当て 同時最適化法

日大生産工(院) ○三澤健一郎 日大生産工 細川利典 日大生産工 山崎紘史
京産大情報工 吉村正義 日大生産工 新井雅之

1. まえがき

超大規模集積回路(Very Large Scale Integrated circuits : VLSI)は、回路の高集積化、高速化、電源電圧の低下に伴い、タイミング遅延を伴う欠陥の影響を受ける。Launch-On-Capture(LOC)方式[1]の実速度スキャンテストは、タイミング遅延を伴う欠陥を検出するために幅広く使用されている。しかしながら、回路の信号線の遷移による過度なキャプチャ時消費電力により、過度な電圧降下(IRドロップ)[2]を引き起こす可能性がある。過度な電圧降下は、タイミング遅延を伴う欠陥を発生させ、不要な歩留まり損失を引き起こす[3]。したがって、テスト時のキャプチャ時消費電力の削減が重要な課題として挙げられる。

キャプチャ時消費電力の削減は、キャプチャ動作時の信号線の遷移数(Launch Switching Activity : LSA)を削減することが非常に重要である。テスト生成ツールで生成される初期テストベクトル集合は、キャプチャ時消費電力の閾値により、キャプチャセーフテストベクトル[5]とキャプチャアンセーフテストベクトル[5]に分類される。キャプチャ時消費電力の閾値を超えるキャプチャアンセーフテストベクトルを実速度スキャンテストに使用した場合、実速度スキャンテスト時に過度なキャプチャ時消費電力が発生する可能性がある。過度なキャプチャ時消費電力により、キャプチャアンセーフテストベクトルでのみ検出されるアンセーフ故障[10]が検出されない可能性があり、故障検出率の低下を引き起こす。したがって、キャプチャアンセーフテストベクトルは、実速度スキャンテストに使用困難である。故障検出率の低下を防ぐためには、初期テストベクトル集合のキャプチャアンセーフテストベクトル数及び初期テストベクトル集合で検出されるアンセーフ故障数を削減することが非常に重要である。

キャプチャ時消費電力の削減のための手法は、回路構造変更による手法[4]、テストデータ変更[5-15]による手法に分類される。回路構造変更による手法は、テスト対象回路にテスト容易化のためのテスト専用回路を付加する。テストデータ変更による手法は、テストデータを低消費電力化されたテストデータにデータ変更する。テストデータ変更による手法は、X判定[6]とX割当て[6-15]から構成される。X判定は、故障検出率を損失させることなく、テストベクトルの故障検出に必要なビットをXビットに割当てる。Xビットを含んだテストデータをテストキューブと呼ぶ。X割当ては、テストキューブのXビットに論理値(0または1)を割当てる。低消費電力指向X割当て手法は、J-Fill[7]、P-Fill[8]、JP-Fill[9]、SAT-Fill[10]、PMSAT-Fill[11]などの多くの手法が提案されている。文献[7-10]は、フリップフロップ(Flip-Flop : FF)の信号線の遷移数を削減することにより、キャプチャ時消費電力を削減している。しかしながら、文献[7-10]は、平均WSA[12]と最大WSA[12]の削減率のみについて言及しており、アンセーフテストベクトル及びアンセーフ故障の削減

率について言及していない。また、低消費電力指向X割当て手法として、PMSAT-Fill[11]が提案されている。文献[11]は、キャプチャアンセーフテストベクトル数及びアンセーフ故障数の大幅な削減に成功している。しかしながら、いくつかのベンチマーク回路において、キャプチャアンセーフテストベクトル及びアンセーフ故障を完全に削除できていない。

低消費電力指向のX判定とX割当てでは、低消費電力化の効果が互いの結果に大きく依存する可能性がある。しかしながら、従来の低消費電力指向のX判定手法とX割当て手法は、キャプチャ時消費電力削減の最適化問題を独立に考えている問題点が挙げられる。本論文は、テストデータのさらなる低消費電力化のためには、X判定とX割当てを同時に適用して、キャプチャ時消費電力削減の最適化問題を同時に考えることが重要であると考える。

本論文では、Partial Max-SATを用いたX判定・X割当て同時最適化法を提案する。第2章は、提案手法であるPartial Max-SATを用いたX判定・X割当て同時最適化法について説明する。第3章は、実験結果について説明する。第4章は、本論文のまとめについて説明する。

2. 低消費電力指向X判定・割当て同時最適化法

提案手法である低消費電力指向X判定・割当て同時最適化法では、初期テストベクトル集合を低消費電力化されたテストベクトル集合にテストデータ変更することで、初期テストベクトル集合のアンセーフテストベクトル数及び初期テストベクトル集合で検出されるアンセーフ故障数の削減することを目指す。提案手法では、まず、低消費電力指向X判定により、初期テストベクトル集合からテストキューブ集合を生成する。次に、Partial Max-SATを用いた低消費電力指向X判定・割当てにより、テストキューブ集合から低消費電力化されたテストベクトル集合を生成する。低消費電力指向X判定・割当てでは、X判定において、Xが割当てられた信号線及び正当化操作で論理値が割当てられた信号線に対して、低消費電力指向な論理値の再割当てを行う。論理値の再割当てする対象を増加させることで、従来のX割当ての狭い解空間ではなく、広い解空間でX割当てを行うことができる。

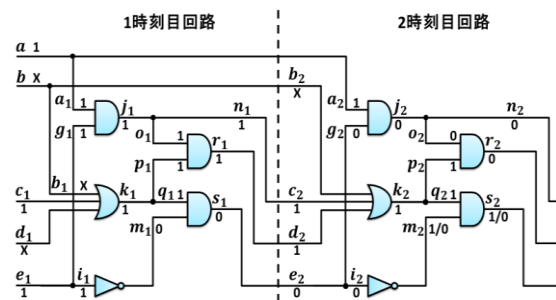


図 1. X判定の例

A Don't Care Identification-Filling Co-Optimization Method for Low Capture Power Testing Using Partial MaxSAT

Kenichiro MISAWA, Toshinori HOSOKAWA, Hiroshi YAMAZAKI,
Masayoshi YOSHIMURA and Masayuki ARAI

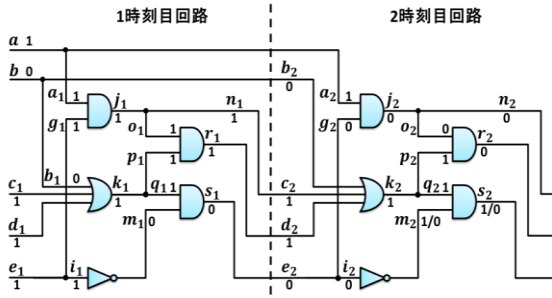


図 2. X 割当ての例

2.1 PMSAT-Fillの問題点

図1に、LOCテスト生成モデルにおけるX判定の例を示す。図1において、 a, b は外部入力信号線、 c_1, d_1, e_1 は疑似外部入力信号線、 n_2, r_2, s_2 は疑似外部出力信号線である。また、各信号線名の数字は、時間展開の時刻を示す。図1の場合、信号線 m の立上り遷移故障を検出するテストベクトル $(a, b, c_1, d_1, e_1) = (1, 0, 1, 0, 1)$ に対して、X判定を行う。まず、信号線 m_1, m_2, q_2, s_2 に対して、故障励起と故障伝搬に必要な論理値が割当てられる。次に、含意操作が行われ、信号線 $e_1, g_1, i_1, s_1, e_2, g_2, i_2, j_2, k_2, n_2, o_2, p_2, r_2$ に対して、論理値が割当てられる。次に、正当化操作及び含意操作が行われ、信号線 $a, a_1, c_1, j_1, k_1, n_1, o_1, p_1, q_1, r_1, a_2, c_2, d_2$ に対して、論理値が割当てられる。X判定によって、テストベクトル $(a, b, c_1, d_1, e_1) = (1, 0, 1, 0, 1)$ からテストキューブ $(a, b, c_1, d_1, e_1) = (1, X, 1, X, 1)$ が生成される。

図2に、PMSAT-Fillを用いたX割当ての例を示す。図2において、X割当ての対象となる信号線は、信号線 b, b_1, b_2, d_1 である。PMSAT-Fillでは、信号線の論理値遷移を抑制するため、 d_1 に1を割当て、信号線 d の立上り遷移を抑制する。また、実速度スキャンテストにおいて、外部入力信号線 b は論理値遷移が発生しないため、 b にはランダムな論理値が割当てられる。図2において、信号線の遷移数は9本である。

図2のように、PMSAT-FILLは、Xを有する信号線の数が少ない場合、信号線の遷移数を大きく削減できない可能性がある。このように、従来の低消費電力指向X割当て手法は、故障伝搬経路、正当化経路を一意に決めてしまっているため、本来の解空間よりも狭い範囲でしかX割当てが実行できない問題点が挙げられる。そのため、本論文では、X判定の正当化操作と信号線の遷移の抑制を同時に実行する低消費電力指向X判定・割当て同時最適化法を提案する。

2.2 低消費電力指向X判定・割当て同時最適化問題の定式化

提案手法である低消費電力指向X判定・割当て同時最適化法は、以下の最適化問題として定式化される。

入力：回路、初期テストベクトル集合、WSA閾値
出力：X割当て後のテストベクトル集合、

アンセーフ故障集合

制約：X割当ての試行回数

最適化：X割当て後のテストベクトル集合におけるアンセーフ故障数の最小化

2.3 低消費電力指向X判定・割当て同時最適化法の全体アルゴリズム

図3に、低消費電力指向X判定・割当て同時最適化法の全体アルゴリズムを示す。入力は、回路 C 、初期テストベクトル集合 T 、WSA閾値 W_{th} 、X割当ての試行回数 N である。出力は、最終テストベクトル集合 T_{fin} 、アンセーフ故障集合 USF である。まず、 T_{fin} を空に初期化する(行4)。次に、X割当ての試行回数を表す変数 i を1

に初期化する(行5)。X割当てのアルゴリズムを切り替えるためのフラグ変数 $switch_flag$ を0に初期化する(行6)。 T を用いて、文献[6]の低消費電力指向X判定を実行して、テストキューブ集合 T_x と検出辞書 TFS を生成する(行7)。ここで、 TFS は、 T_x のテストキューブと各テストキューブが検出を保証しなければならない故障の組合せが記述された辞書である。 i が N 以下の場合、行9から行24の処理を実行する(行8)。 T_{fin} が空でなく、 $i-1$ 回目のX割当て後のテストベクトル集合内のキャプチャセーフテストベクトル集合 $Tsafe_{i-1}$ が空である、または、 $switch_flag$ が0でない場合、行10から行12の処理を実行する。そうでなければ、行13から行14の処理を実行する(行9)。行9が条件を満たす場合、 $switch_flag$ に1を格納し(行10)、 C, T_x, TFS, i を用いて、提案手法である低消費電力指向X判定・割当てを行い、 i 回目のX割当て後のテストベクトル集合 T_i を生成する(行11)。行9が条件を満たさない場合、 C, T_x, i を用いて、文献[11]の低消費電力指向X割当てを行い、 i 回目のX割当て後のテストベクトル集合 T_i を生成する(行13)。 T_i と W_{th} を用いて、WSA計算を実行して、 T_i をキャプチャセーフテストベクトル集合 $Tsafe_i$ とキャプチャアンセーフテストベクトル集合 $Tunsafe_i$ に分類する(行15)。 $Tsafe_i$ が空である、または、 i が N に等しい場合、 T_{fin} に T_{fin} と $Tsafe_i$ の和集合を格納し(行17)、行9から行24の処理を終了する(行16)。 T_{fin} に T_{fin} と $Tsafe_i$ の和集合を格納する(行20)。 T_x から $Tsafe_i$ に対応するテストキューブを削除する(行21)。 TFS から $Tsafe_i$ に対応するテストキューブ及び故障の組合せを削除する(行22)。 i を1増加する(行23)。 $C, T_{fin}, Tunsafe_i$ を用いて、故障シミュレーションを行い、 USF を更新する(行25)。 T_{fin} に T_{fin} と $Tunsafe_i$ の和集合を格納する(行26)。最後に、 T_{fin} と USF を返し、アルゴリズムを終了する(行27)。

提案手法では、X割当て後のテストベクトル集合内のキャプチャアンセーフテストベクトルに対応するテストキューブに対して、Partial Max-SATの制約時間を増加させ、再度、低消費電力指向X割当てを行う。Partial Max-SATの制約時間を増加させ、テストキューブのX割当てに掛ける時間を延長することで、キャ

```

1. Input: Circuit C, Initial test set T, Threshold value of WSA  $W_{th}$ , and #Iteration N
2. Output: Final test set  $T_{fin}$ , Unsafe fault set USF
3. LCP_X-id-fill_Partial_MaxSAT_IT (C, T,  $W_{th}$ , N) {
4.    $T_{fin} = \varnothing$ ;
5.    $i = 1$ ;
6.    $switch\_flag = 0$ ;
7.    $(T_x, TFS) = X-ID (C, T)$ ;
8.   while ( $i \leq N$ ) {
9.     if ( $T_{fin} \neq \varnothing$  && ( $Tsafe_{i-1} = \varnothing \parallel switch\_flag \neq 0$ )) {
10.       $switch\_flag = 1$ ;
11.       $T_i = LCP\_X-id-fill\_Partial\_MaxSAT(C, T_x, TFS, i)$ ;
12.     } else {
13.       $T_i = LCP\_X-fill\_Partial\_MaxSAT(C, T_x, i)$ ;
14.     }
15.      $(Tsafe_i, Tunsafe_i) = Calc\_WSA(C, T_i, W_{th})$ ;
16.     if ( $(Tunsafe_i = \varnothing) \parallel (i = N)$ ) {
17.       $T_{fin} = T_{fin} \cup Tsafe_i$ ;
18.      break;
19.     }
20.      $T_{fin} = T_{fin} \cup Tsafe_i$ ;
21.      $T_x = T_x - \text{test cubes corresponding to } Tsafe_i$ ;
22.      $TFS = TFS - \text{test cube-faults pairs corresponding to } Tsafe_i$ ;
23.      $i++$ ;
24.   }
25.    $USF = \text{Fault\_sim}(C, T_{fin}, Tunsafe_i)$ ;
26.    $T_{fin} = T_{fin} \cup Tunsafe_i$ ;
27.   return( $T_{fin}, USF$ );
28. }
```

図 3. 低消費電力指向 X 判定・割当て同時最適化法の全体アルゴリズム

```

1. Input: Circuit  $C$ , Target test cube set  $T_x$ , Test cube-faults pair set  $TFS$  and  $i$ -th iteration  $i$ 
2. Output:  $i$ -th test vector set  $T_i$ 
3. LCP_X-id-fill_Partial_MaxSAT( $C, T_x, TFS, i$ ) {
4.    $T_i = \emptyset$ ;
5.    $\Phi_C = \text{Generate CNF from } C$ ;
6.   for ( $j = 0; j < |T_x|; j++$ ) {
7.      $(\Phi_{const_j}, \Phi_{tran_j}) = \text{Fault\_sim}(C, t_j, tfs_j)$ ;
8.      $\Phi = \Phi_C \cdot \Phi_{const_j} \cdot \Phi_{tran_j}$ ;
9.      $t_j = \text{Patial\_MaxSAT}(\Phi)$ ;
10.     $T_i = T_i \cup t_j$ ;
11.  }
12.  return( $T_i$ );
13. }

```

図 4. 低消費電力指向 X 判定・割当てのアルゴリズム

ブチャセーフテストベクトルが生成される可能性が増加すると考えられる。

2.4 低消費電力指向 X 判定・割当てのアルゴリズム

図4に、低消費電力指向X判定・割当てのアルゴリズムを示す。入力、回路 C 、初期テストキューブ集合 T_x 、検出辞書 TFS 、X割当ての試行回数を表す変数 i である。出力は、 i 回目のX割当て後のテストベクトル集合 T_i である。変数 i は、何回目のX割当てを行っているかの試行回数を表す。まず、X割当て後のテストベクトルを格納するテストベクトル集合 T_i を空に初期化する(行4)。次に、 C を用いて、 C 内のゲートの動作を表す命題論理式を生成して、 Φ_C に格納する(行5)。 T 内のテストキューブ t_j に対して、行7から行11の処理を実行する(行6)。 C, t_j, tfs_j を用いて、故障シミュレーションを実行する。ここで、 tfs_j に記述されている故障に対して、故障シミュレーションが実行される。故障シミュレーション後、故障伝搬に必要な論理値を有する信号線に対して、信号線の論理値を有するための節が生成され、 Φ_{const} に格納される。また、故障伝搬に必要な論理値を有する信号線に対して、信号線の論理値遷移を抑制するための節が生成され、 Φ_{tran_j} に格納される(行7)。 t_j の命題論理式として、 $\Phi_C, \Phi_{const_j}, \Phi_{tran_j}$ を Φ に格納する(行8)。 Φ を入力として、 t_j のPartial Max-SATを用いた低消費電力指向X割当てを実行する(行9)。 T_j に t_j と t_j の和集合を格納する(行10)。 T_j を返して、アルゴリズムを終了する(行12)。

2.5 命題論理式の生成

提案手法では、低消費電力指向X判定・割当て問題を命題論理式 Φ に変換して、命題論理式 Φ をPartial Max-SATソルバーに与えることで、テストキューブの低消費電力指向X判定・割当てを行う。命題論理式 Φ は、 $\Phi_C, \Phi_{const}, \Phi_{tran}$ から構成される。 Φ_C は回路モデルを表すための命題論理式である。 Φ_{const} は、故障を励起及び伝搬するための命題論理式である。 Φ_{tran} は、信号線の論理値遷移を抑制するための命題論理式である。図5に、命題論理式の生成例を示す。

まず、命題論理式 Φ_C の生成について説明する。提案手法では、2時間展開された回路モデルを表すために命題論理式 Φ_C をハード節として生成する。ハード節の重みは、無限大に設定する。

次に、命題論理式 Φ_{const} の生成について説明する。提案手法では、信号線が故障励起及び故障伝搬に必要な論理値を有するために、ハード節を生成する。ハード節の重みは、無限大に設定する。

図5において、信号線 $e_1, g_1, i_1, m_1, s_1, e_2, g_2, i_2, j_2, k_2, m_2, n_2, o_2, p_2, q_2, r_2, s_2$ は、X判定において、故障励起及び故障伝搬に必要な論理値が割当てられた信号線である。これらの信号線の論理値は、信号線 m の立上り遷移故障が励起され、故障影響が疑

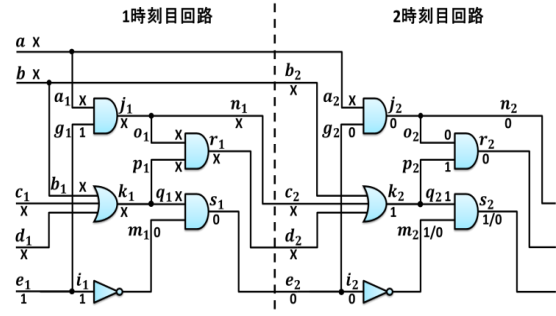


図 5. 命題論理式の生成例

似外部出力 s_2 に伝搬されるために必要となる。したがって、これらの信号線に対して、信号線が故障励起及び故障伝搬に必要な論理値を有するための命題論理式 Φ_{const} をハード節として生成する。式(2.1)に図5におけるハード節の例を示す。

$$\begin{aligned} \Phi_{const} = & (e_1, \infty) \cdot (g_1, \infty) \cdot (i_1, \infty) \cdot (\neg m_1, \infty) \cdot \\ & (\neg s_1, \infty) \cdot (\neg e_2, \infty) \cdot (\neg g_2, \infty) \cdot (\neg i_2, \infty) \cdot \\ & (\neg j_2, \infty) \cdot (k_2, \infty) \cdot (m_2, \infty) \cdot (\neg n_2, \infty) \cdot \\ & (\neg o_2, \infty) \cdot (p_2, \infty) \cdot (q_2, \infty) \cdot (\neg r_2, \infty) \cdot (s_2, \infty) \quad (2.1) \end{aligned}$$

次に、命題論理式のソフト節について説明する。提案手法では、信号線の論理値遷移を抑制するために、ソフト節を生成する。ソフト節の重みは、1に設定する。

提案手法では、X割当てにより、信号線の遷移が発生する可能性がある信号線に対して、以下の5種類のソフト節からいずれかのソフト節を生成する。以下では、ある信号線 s の1時刻目論理値がX、2時刻目論理値が0の場合、各論理値を(X, 0)と表している。

(1) (X, 0)

信号線 s が(X, 0)の場合、X割当てにおいて、信号線 s の立下り遷移故障が発生する可能性があるため、以下のソフト節が生成される。

$$(\neg s_2, 1)$$

(2) (X, 1)

信号線 s が(X, 1)の場合、X割当てにおいて、信号線 s の立上り遷移故障が発生する可能性があるため、以下のソフト節が生成される。

$$(s_2, 1)$$

(3) (0, X)

信号線 s が(0, X)の場合、X割当てにおいて、信号線 s の立上り遷移故障が発生する可能性があるため、以下のソフト節が生成される。

$$(\neg s_1, 1)$$

(4) (1, X)

信号線 s が(1, X)の場合、X割当てにおいて、信号線 s の立下り遷移故障が発生する可能性があるため、以下のソフト節が生成される。

$$(s_1, 1)$$

(5) (X, X)

信号線 s が(X, X)の場合、X割当てにおいて、信号線 s の立上り遷移故障か立下り遷移故障が発生する可能性があるため、以下のソフト節が生成される。

$$(s_1 \vee \neg s_2, 1) \cdot (\neg s_1 \vee s_2, 1)$$

図5の場合、信号線 $c, d, j, k, n, o, p, q, r$ は、X判定において、故障励起及び故障伝搬に必要な論理値が割当てられた信号線である。これらの信号線は、割当てにより、立上り遷移故障か立下り遷移故障が発生する可能性がある。したがって、これらの信号線に対して、信号線の論理値遷移を抑制するための命題論理式をソフト節として生成する。式(2.2)に図5におけるソフト節の例を示す。

$$\Phi_{tran} = (\neg c_1 \vee c_2, 1) \cdot (c_1 \vee \neg c_2, 1) \cdot (\neg d_1 \vee d_2, 1) \cdot (d_1 \vee \neg d_2, 1) \cdot (\neg j_1, 1) \cdot (k_1, 1) \cdot (\neg n_1, 1) \cdot (\neg o_1, \infty) \cdot (p_1, \infty) \cdot (q_1, \infty) \cdot (\neg r, \infty) \quad (2.2)$$

3. 実験結果

本章では、実験結果について説明する。提案手法は、C言語で実装された。実験対象回路は、ISCAS'89ベンチマーク回路及びITC'99ベンチマーク回路である。実験には、Core i7 4790(3.6GHz)のプロセッサ及び8GBのメモリを搭載したCPUを使用した。初期テストベクトル集合は、自動テストパターン生成ツールのTetraMaxで生成した。テストキューブ集合は、文献[6]の低消費電力指向X判定法でX判定した。WSA閾値は、テスト時に遷移が発生する可能性がある信号線において、論理値遷移が発生した場合のWSA値の20%に設定した。Partial Max-SATは、Intel Xeon(R) CPU E51660v4(3.2GHz×16)のプロセッサ及び32GBのメモリを搭載したCPUを用いて、DistでPartial Max-SATを行った。1つのテストキューブにおけるPartial Max-SATの制約時間は、i回目(1≤i≤11)のX割当てに対して、0.1秒、1秒、2秒、3秒、4秒、5秒、6秒、7秒、8秒、9秒、10秒に設定した。

表1に、提案手法である低消費電力指向X判定・割当て同時最適化法の実験結果を示す。表1において、「Circuit」は回路名、「#TV」は初期テストベクトル数、「PMSAT-Fill」はPMSAT-Fillの実験結果、「Proposed」は提案手法の実験結果、「#STV」はキャプチャセーフテストベクトル数、「#UTV」はキャプチャアンセーフテストベクトル数、「#USF」はアンセーフ故障数、「Time」はX割当ての実行時間である。提案手法は、PMSAT-Fill[11]と比較して、キャプチャセーフテストベクトル数を平均7.01%増加、キャプチャアンセーフテストベクトル数を平均31.63%削減することに成功した。また、アンセーフ故障数を平均31.84%削減することに成功した。さらに、PMSAT-Fill[11]と比較して、全ての回路において、X割当ての実行時間を削減することに成功した。

4. まとめ

本論文では、Partial Max-SATを用いて低消費電力指向X判定・割当て同時最適化法を提案した。提案手法は、従来手法であるPMSAT-Fill[11]と比較して、キャプチャアンセーフテストベクトル数を平均31.63%、最大100%削減することに成功した。また、キャプチャアンセーフ故障数を平均31.84%、最大100%削減することにも成功した。

参考文献

- 1) J. Savir and S. Patil, "Scan-based transition test," IEEE Trans. Comput. Aided Design Int. Circuits & Syst., vol. 13, no. 8, pp. 1057-1064, 1994.
- 2) J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreepakash and M. Hachinger, "A case study of IR-drop in structured at-speed testing," Proc. ITC, pp. 1098-1104, 2003.
- 3) Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," Proc. VTS, pp. 4-9, 1993.
- 4) N. Ahmed, M. Tehranipoor and V. Jayaram, "Transition delay fault test pattern generation considering supply voltage noise in a SOC design," Proc. DAC, pp. 533-538, 2007.
- 5) X. Wen, K. Miyase, S. Kajihara, H. Furukawa, Y. Yamato, A. Takashima, K. Noda, H. Ito, K. Hatayama, T. Aikyo and K. K. Saluja, "A Capture-Safe Test Generation Scheme for At-Speed Scan Testing," Proc. ETS, pp. 55-60, 2008.
- 6) K. Miyase, K. Noda, H. Ito, K. Hatayama, T. Aikyo, Y. Yamato, H. Furukawa, X. Wen, and S. Kajihara, "Effective IR-Drop Reduction in At-Speed Scan Testing Using Distribution-Controlling X-Identification," Proc. ICCAD, pp. 52-58, 2008.
- 7) X. Wen, Y. Yamashita, S. Kajihara, L. T. Wang, K. K. Saluja and K. Kinoshita, "On Low-Capture-Power Test Generation for Scan Testing," Proc. VTS, pp. 265-270, 2005.
- 8) S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz and J. Rajski, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," Proc. ITC, paper 32.2, 2006.
- 9) X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi and L. T. Wang, "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," Proc. ITC, paper 25.1, 2007.
- 10) M. Yoshimura, Y. Takahashi, H. Yamazaki, and T. Hosokawa, "A Don't Care Filling Method for Low Capture Power based on Correlation of FF Transitions Using SAT," IEICE Trans. Fundamentals, Vol. E100-A, no. 12, pp. 2824-2833, 2017.
- 11) T. Hosoka, K. Misawa, Y. Hirama, H. Yamazaki, M. Yoshimura, and M. Arai, "A Low Capture Oriented X-filling Method Using Partial MaxSAT Iteratively," Proc. DFTS, 2019 (Will appear).
- 12) I. Pomeranz and S. M. Reddy, "Switching activity as a test compaction heuristic for transition faults," IEEE Trans. VLSI Syst., vol. 18, no. 9, pp. 1357-1361, 2010.

表 1. 提案手法の実験結果

Circuit	#TV	PMSAT-FILL[18]				Proposed			
		#STV	#UTV	#USF	Time (s)	#STV	#UTV	#USF	Time (s)
s5378	160	142	18	310	1070.5	150	10	175	651.0
s9234	319	227	92	2535	5296.4	280	39	901	2593.9
s13207	310	309	1	43	428.6	309	1	43	380.0
s15850	199	198	1	20	287.2	199	0	0	212.9
s35932	62	47	15	3508	951.8	53	9	1699	712.2
s38417	238	169	69	12475	4107.7	192	46	9860	3070.8
s38584	412	404	8	2942	1721.1	406	6	2544	1251.2
b14	1185	715	470	12207	26941.6	811	374	10572	23202.5
b15	995	972	23	235	2456.4	973	22	202	2180.5
b20	1399	1015	384	8832	24013.5	1036	363	8278	22354.9
b21	1484	1099	385	11265	24172.9	1119	365	10851	22532.4
b22	1513	1177	336	8750	24802.8	1324	189	4318	22318.3