ISSN 2186-5647

2-22

推定パターン数削減のための テスト並列化ドントケア割当て手法

日大生産工（院）　○徳田晴太　日大生産工　細川利典
京産大　吉村正義　日大生産工　新井雅之

1. まえがき

近年，超大規模集積回路(Very Large Scale Integrated circuits: VLSI)のテストコスト増大に伴い，テストパターン数の削減が重要視されている．設計した回路に対して生成されたテストパターン集合に対してテストパターン数を削減するテスト圧縮法 [1-2] や，同一の機能でありながら多数の故障を並列にテストできる回路構造となるよう設計するテスト容易化設計(Design-for-Testability: DFT) [3-7] が提案され，このテスト並列化のための DFT を適用した回路は，生成されたテストパターン数が削減されている．

しかしながら，テスト圧縮法 [1-2] の有効性は回路構造に依存し，テストパターン削減効果の低い回路構造が存在する．また，論理合成後のゲートレベルにおいて，テストパターン数の削減を図る DFT [5-7] を適用すると，論理最適化された回路に対するゲートの追加により，タイミングの最適性を損失する可能性がある．以上の理由から，論理合成適用前の抽象度の高いレジスタ転送レベル(Register Transfer Level: RTL)でテスト並列化を考慮したテスト容易化設計が重要である．

RTL におけるテストパターン数削減のための DFT 手法 [3-4] が提案されている．文献 [4] は，機能動作時に遷移しないコントローラの無効状態に着目し，無効状態を含む状態遷移(State Transition: ST)にハードウェア要素のテスト並列度が向上する制御信号値として設計する．この設計を適用した結果，テスト圧縮の効率が向上し，テストパターン数を平均 33.47%削減できた [4]．しかしながら，無効状態を含む状態遷移に対する特定の制御信号値を割当てる DFT 手法は, 面積オーバヘッドが平均 7.12%, 最大 38.37%となった [4].

文献 [8] では，面積オーバヘッドを抑制するために，コントローラの有効状態 [4] の状態遷移における制御信号値のドントケア X に着目して，データパス中のハードウェア要素のテスト並列度を向上させるためのドントケア割当て法が提案された．文献 [8] で提案された制御信号のドントケア割当て法は，擬似ブール最適化(Pseudo Boolean Optimization: PBO)問題として定式化され，PBO ソルバを用いて状態遷移ごとに，推定パターン数を最小化するドントケア割当てされた状態遷移を求めた．小規模な高位合成ベンチマーク回路では，論理合成によりドントケア割当てが実行された回路と比較して，テストパターン数が平均 8.06%削減されたことが報告されている．しかしながら，中規模以上の高位合成ベンチマーク回路は, 小規模回路と比較して制御信号の X 数が増大するため PBO で解を求める時間は膨大となり，現実的な時間で解くことは不可能である．

文献 [9-10] では, 中規模以上の回路に対して現実的な時間で解を求めるためのヒューリスティックアルゴリズムが提案されている．そのアルゴリズムは，各状態遷移で全ての演算器がテスト可能となるようにドントケア割当てを実行している．しかしながら，データパスのテストパターン数を支配する演算器のみのテスト可能性を考慮したドントケア割当てを実行することが，推定パターン数削減に寄与するか否かを明確に判断することは困難である．

本論文では，推定パターン数削減のためのコントローラの制御信号のドントケア割当て手法を提案する．推定パターン数の削減に貢献するハードウェア要素を絞込み，それらをテスト可能とする候補状態遷移を列挙し，テスト並列度を向上させるドントケア割当てされた候補状態遷移の選択問題を擬似ブール最適化(Pseudo Boolean Optimization: PBO)問題を用いて解く．

第 2 章で, 前提知識について説明する. 第 3 章で, 提案手法について説明する．第 4 章で，実験結果について説明し，最後に結論と今後の課題について述べる．

2. 前提知識

本論文における対象回路は，データパス，FSM で設計されたコントローラで構成された RTL 回路である．データパスはコントローラへ状態信号を供給し，コントローラはデータパスへ制御信号を供給する．

図 1 に RTL 回路例を示し，(a)にデータパス，(b)にコントローラを示す．図 1(a)において，R0~R2 は REG，+と*はそれぞれ加算器と乗算器，M1 は MUX，m1 は M1 の制御信号線，a, b, c は外部入力，y は外部出力を示す. MUX 内の数字は MUX の制御信号値に対応した入力番号を示す．また，R1 はホールド機能付き REG(HREG)であり，r1 は R1 の制御信号線である．本論文では，データパス回路内の信号線，演算器，マルチプレクサ(MUX)，MUX の制御信号値線，レジスタ(REG)，REG の制御信号線をハードウェア要素 [4] と呼ぶ．ハードウェア要素とは，データパスを構成するモジュールを示しており，特に，MUX や REG の制御信号線の故障についてもハードウェア要素と定義する．ここで，M1 は 3 入力の MUX であるが, 制御信号線の故障のテストを考慮し,

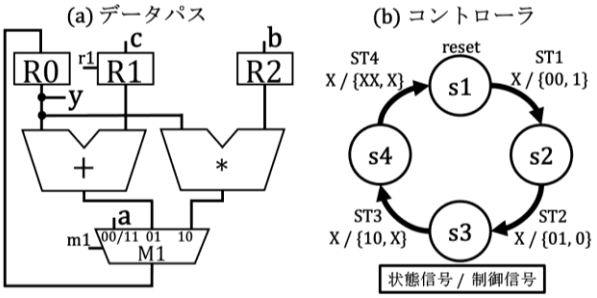

図 1. RTL 回路例

X-Filling Method for Concurrent Test to Reduce the Estimated Number of Patterns
Haruta TOKUTA, Toshinori HOSOKAWA,
Masayoshi YOSHIMURA and Masayuki ARAI

機能動作に用いない制御信号 11 を制御信号 00 と同じ出力値として MUX を設計する．提案手法はフルスキャン設計を前提としているため，機能動作に用いない制御信号は可能な限り，外部入力または REG と接続する信号線を選択する制御信号とし，制御信号の可制御性を向上させる．

図 1(b)において，s1~s4 は状態，ST1~ST4 は状態遷移，status は状態信号，m1,r1 は制御信号値である．本コントローラは各状態の遷移先が 1 状態であるため，全ての状態に供給される状態信号は X である．

2.1 テスト可能表

テスト可能表とは，構造的記号シミュレーション[8-9]を実行後，状態遷移ごとにハードウェア要素のテスト可能性を示す表である．構造的記号シミュレーションとは，与えられた状態遷移でハードウェア要素のテスト可能性を求める手法である．楽観構造的記号シミュレーションの結果により楽観テスト可能表，悲観構造的記号シミュレーションの結果により悲観テスト可能表がそれぞれ生成できる．

表 1 に，図 1 の RTL 回路を用いた楽観と悲観テスト可能表を示す．表 1 において，1 行目に各ハードウェア要素を示し，1 列目に各状態遷移を示している．2 行目 2 列目から，各ハードウェア要素が楽観的にテスト可能か否か，悲観的にテスト可能か否かを示している．並列テストが可能であれば「✓」，そうでなければ「×」である．また，楽観的にテスト可能で悲観的にテスト不能であれば「✓/×」である．例えば，表 1 における 2 行目 2 列目は，演算器+がテスト不能，3 行目 2 列目はテスト可能，5 行目 2 列目は楽観的にテスト可能で悲観的にテスト不能であることを示している．

楽観と悲観テスト可能表における「✓」とは，データパス内の MUX, HREG の制御信号値 X に依らずテスト可能であることを示している．「×」とは，にいかなるドントケア割当てを行った場合でも必ずテスト不能であることを示している．「✓/×」は，適切なドントケア割当てによりテスト可能となるかもしれないことを示している．「×/✓」は，シミュレーション上存在することはない．

表 1 の 5 行目 2 列目に着目すると，ST4 で演算器+が楽観的にテスト可能であるのに対し，悲観的にテスト不能であることがわかる．これは図 1(a)の MUX1 の制御信号のドントケア割当てによって，演算器+のテスト可能性が変化することを示しており，実際に MUX1 の制御信号線 m1 に論理値 01 を割当てることで演算器+はテスト可能となる．また，制御信号 m1 に論理値 00, 10, 11 を割当てると演算器+はテスト不能となる．あるハードウェア要素をテストするためには適切なドントケア割当てが必要である．

2.2 推定パターン数最小化のためのドントケア割当て手法

文献 [8] では，コントローラのオリジナルの状態遷移の制御信号 X に対して考えられる全通りのドントケア割当てされた状態遷移を列挙し，PBO ソルバで最も推定パターン数が最小となる状態遷移を選択する．推定パターンとは，RTL で見積られたテストパターンを示しており，各ハードウェア要素の単体で論理合成して ATPG を実行し得られた決定論的パターンのことである．

文献 [8] で提案された手法について説明する．例えば，図 1 においてオリジナルの ST3 の制御信号は m1=10, r1=X であるが，X に対して論理値 0,1 を割当てた m1=10, r=0 と m1=10, r1=1 の 2 通りがドントケア割当て状態遷移であり，それぞれ ST3_0, ST3_1 としている．また，オリジナルの ST4 の制御信号に 3 つの X があり，全部で 8 通りのドントケア割当てされた状態遷移が列挙される．つまり，オリジナルのある状態遷移の制御信号の X 数を n とすると，列挙される状態遷移の数は2^n通りとなる．しかしながら，X 数が多い大規模回路に対しては現実的な時間で解を求めるのは困難である．

表 1. 楽観と悲観テスト可能表

	+	*	M1in0	M1in1	M1in2	M1out	M1_1b_sa0	M1_1b_sa1	M1_2b_sa0	M1_2b_sa1	R0	R1	R1_1b_sa0	R1_1b_sa1	R2
ST1	×	×	✓	×	×	✓	×	✓	×	✓	✓	×	✓	×	×
ST2	✓	×	×	✓	×	✓	✓	×	×	×	✓	✓	×	✓	×
ST3	×	✓	×	×	✓	✓	×	×	✓	×	✓	✓/×	✓/×	✓/×	✓
ST4	✓/×	✓/×	✓/×	✓/×	✓/×	✓	✓/×	✓/×	✓/×	✓/×	✓	✓	✓	✓/×	✓/×

3. 提案手法

3.1 ヒューリスティックアルゴリズム

本論文では，推定パターン数削減のためのヒューリスティックアルゴリズムを提案する．提案手法は 2 つの要素から構成され，一つは推定パターン数削減において考慮するハードウェア要素と状態遷移の絞り込みであり，もう一つは複数の候補状態遷移の圧縮による候補状態遷移数削減である．ここで，候補状態遷移とはオリジナルの状態遷移の制御信号 X に対して全てまたは一部ドントケア割当てされた状態遷移のことを指す．

一つ目の絞り込み手法は，楽観テスト可能表から下界となる推定パターンを求め，その推定パターンと悲観テスト可能表において単体テストパターン数以上となるハードウェア要素を特定することである．

図 2 に，楽観推定パターンと悲観テスト可能表を用いたテスト並列化ハードウェア要素列挙例を示す．例に用いるハードウェア要素は+, *, R0, R1_1b_sa0 とする．まず楽観推定パターン数ETP_oを求める．ETP_oは，その状態遷移で各ハードウェア要素を楽観的にテスト可能と判定するため，推定パターン数の下界を示す．また，悲観テスト可能表は状態遷移ごとにドントケア割当ての値に関わらず必ずテスト可能なハードウェア要素を示す．図 2 において，各ハードウェア要素が悲観的にテスト可能と判定された状態遷移と同じ状態遷移の楽観推定パターン数を用いて，ハードウェア要素ごとの下界パターン数を求めた．次に各ハードウェア要素の下界テストパターン数が単体テストパターン数未満かを判定した．その結果，ハードウェア要素+, *, R1_1b_sa0 は下界パターン数が単体テストパターン数未満であるため，これら 3

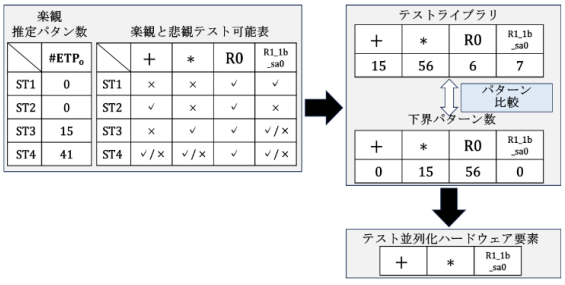

図 2. 楽観推定パターン数と悲観テスト可能表を用いたテスト並列化ハードウェア要素列挙例

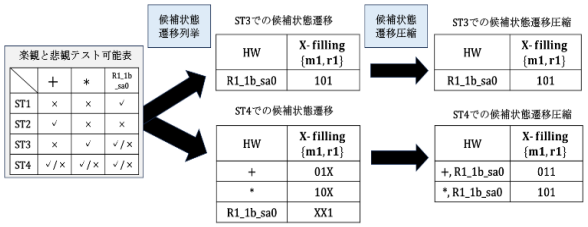

図 3. テスト並列化ハードウェア要素の候補状態遷移と圧縮例

つのハードウェア要素がテスト並列化ハードウェア要素と判定した.

もう一つの複数の候補状態遷移の圧縮による候補状態遷移数削減手法は，あるテスト並列化ハードウェア要素をテスト可能となるように制御信号の一部の X に値を割当てた候補状態遷移を求め，その候補状態遷移同士を圧縮し，候補状態遷移集合を生成する．この集合は X に全通りの値を割当てた候補状態遷移集合より小さなサイズとなる.

二つ目の絞り込み手法は，テスト並列化ハードウェア要素を実際にテスト可能とする候補状態遷移を求め，求まった候補状態遷移同士を支配する候補状態遷移となるまで圧縮を行う.

図 3 に，テスト並列化ハードウェア要素の候補状態遷移と圧縮例を示す．まずテスト並列化ハードウェア要素がドントケア割当てによってテスト可能となる状態遷移を列挙する．例えば，演算器+は，ST4 で楽観的にテスト可能であり，テスト可能となるドントケア割当ては m1=01, r1=X である．この ST4 の制御信号の一部の X に m1=01 と値を割当てた状態遷移を候補状態遷移とする．テスト並列化ハードウェア要素に対して，それぞれ候補状態遷移を求める．次に，同じ状態遷移で列挙された候補状態遷移同士を圧縮する．圧縮された候補状態遷移は複数のテスト並列化ハードウェア要素をテスト可能とする．例えば，ST4 において R1_1b_sa0 をテスト可能とする候補状態遷移 m1=XX,r1=1 があり，+をテスト可能とする候補状態遷移 m1=01, r1=X がある．圧縮された候補状態遷移 m1=01, r1=1 は R1_1b_sa0 と+を同時にテスト可能な候補状態遷移である．この圧縮の繰返しにより，多数のハードウェア要素を同時にテスト可能とする候補状態遷移を生成する．圧縮された候補状態遷移のみを対象とすることで探索対象となる候補状態遷移数を削減する.

3.2 PBO

PBO とは，与えた制約式を充足しつつ最適化関数を最小化するような論理変数割当てを求める問題である．PBO は，一般的に 0-1 整数計画問題と同一の問題を解くことができる.

本手法における PBO の入力は，候補状態遷移集合のテスト可能表である．出力はドントケア割当てコントローラと推定パターン数の総和である．ただし，オリジナルの状態遷移のテスト可能表を入力とした場合，出力は推定パターン数の総和である．以下に，PBO の制約式と最適化関数について，X_{sij}とY_{si}の導入変数を用いて説明する.

X_{sij}は，状態遷移 s の i 番目の候補状態遷移においてハードウェア要素 j がテスト可能か否かを判断する変数である．X_{sij}が 1 の場合テスト可能であり，0 の場合はテスト不能を表す．Y_{si}は，状態遷移 s の i 番目の候補状態遷移を選択するか否かを判断する変数である．Y_{si}が 1 の場合は，その候補状態遷移を選択し，0 の場合は選択しないことを表す．制約式(1)を示す.

$$\sum_{i=1}^{N_a} Y_{si} = 1 \quad (1)$$

式(1)は，候補状態遷移のうち選択される候補状態遷移は 1 つのみであることを示す．N_aは，状態遷移 s の候補状態遷移の総数である．制約式(2)を示す.

$$\prod_{j=1}^{N_m} \sum_{s=1}^{N_s} \sum_{i=1}^{N_a} Y_{si} \times X_{sij} \geq 1 \quad (2)$$

式(2)は，全ハードウェア要素が少なくとも 1 つ以上の候補状態遷移でテスト可能とする制約である．N_mはデータパスのハードウェア要素数，N_sはオリジナルの状態遷移数である．つまり，各ハードウェア要素が 1 つ以上の状態遷移でテスト可能となる候補状態遷移を選択する制約である．制約式(3)を示す.

$$\sum_{s=1}^{N_s} \sum_{i=1}^{N_a} Y_{si} \times X_{sij} \times A_s \geq TP_j \quad (3)$$

式(3)は，ハードウェア要素 j をテスト可能な状態遷移 s に割当てるテストパターン数A_sがハードウェア要素 j の単体テストパターン数以上であることを示す. ハードウェア要素 j をテスト可能な各状態遷移に対して，各状態遷移に割当てる推定パターン数の総和が j の単体テストパターン数以上とするための制約である．次に，最適化関数(4)を示す.

$$min \sum_{s=1}^{N_s} \sum_{i=1}^{N_a} Y_{si} \times A_s \quad (4)$$

式(4)は，推定パターン数の総和を最小化とする最適化関数である.

4. 実験結果

本章では，実験結果について述べる．故障モデルは縮退故障を対象とし，回路は RTL ベンチマーク回路 ARF,BPF,ex2,kim,maha,sehwa の 6 つの回路を対象とする．計算機は，Corei7-11700(2.5GHz)及びメモリ 32GB のマシンを用いる．また，本実験における PBO solver は clasp を用いており，実行時間を 3600s と設定した．論理合成ツールは Synopsys 社の Design Compiler を使用した．4.1 節に，推定パターン数の実験結果について述べ，4.2 節に，ATPG 実行結果について述べる.

4.1 推定パターン数

本節では，オリジナル回路 org，フルランダムドントケア割当て回路 rand，提案手法適用回路 pro それぞれの悲観推定パターン数の総和と面積オーバヘッドについて述べる．本実験におけるフルランダムドントケア割当て回路とは，コントローラの制御信号の全ての X に対してランダムにドントケア割当てを実行したコントローラを 10,000 個生成し，その中で最も推定パターン数が少ない回路である.

本実験では，制御信号中の X が 100 個以上の回路には，PBO によるドントケア割当ての探索空間を更に削減するためのパーシャルランダムドントケア割当てを実行した．制御信号 X の30%にランダムにドントケア割当てを実行したコントローラを 10,000 個

求め，それぞれに対して楽観推定パターン数と悲観推定パターン数を求め，楽観推定パターン数と悲観推定パターン数と候補状態遷移数が少ないコントローラを実験対象とした．

表 2 に，ドントケア割当て結果を示す．Circuit は回路名，#HEs はハードウェア要素数，#STs は状態遷移数，method は org がオリジナルコントローラ，rand がフルランダムドントケア割当てコントローラ，pro が提案手法適用コントローラを示している．#Xs は制御信号 X の総数，$\#ETP_p$は悲観推定パターン数の総和，Runtime(s)は X 割当て実行時間，#Area は回路面積，AO(%)は，org と比較したときの rand，pro の面積オーバヘッド，Red(%)は org に対する pro の悲観推定パターン数削減率を示している．

Red 列では，全回路における悲観推定パターン数の削減を示し，平均 32.64%，最 48.87%削減した．pro の面積オーバヘッドは org より平均 1.37%，最大 3.31%であった．しかしながら，回路規模に対して平均 1.37%の面積オーバヘッドは，僅かであると言える．

4.2 ATPG 実行結果

本節では，org と pro に対して有効状態のみ用いた ATPG を行う内製 ATPG ツールを用いた．内製 ATPG ツールは，多数の目標故障を同時検出するテストパターンを生成する．今回は目標故障数を 100 とした．表 3 に ATPG 実行結果を示す．Circuit は回路名，#Total faults は総故障数，#DT は検出故障数，#UT はテスト不能故障数，#VUT は有効状態でテスト不能な故障の数，#ND は未検出故障数，#TP はテストパターン数，FE(%)は故障検出効率，FC(%)は故障検出率，Runtime(s)は ATPG 実行時間，Red(%)はテストパターン削減率を示す．

テストパターン数は平均 35.31%，最大 74.60%削減した．故障検出率は，ex2,sehwa を除く 4 つの回路において org 以上の故障検出率を達成していることがわかる．

5. まとめと今後の課題

本論文では，スキャン設計を前提とした推定パターン数削減のためのコントローラの制御信号のドントケア割当て手法を提案した．

推定パターン数は，平均 32.64%，最大 48.87%削減した．テストパターン数削減率は，平均 35.31%，最大 74.60%であった．また，面積オーバヘッドは平均 1.37%と僅かであった．

今後の課題として，多くの回路を用いた実験と，遷移故障モデルに対応した本手法の拡張が挙げられる．

表 2. ドントケア割当て結果

Circuit	#HEs	#STs	method	#Xs	#ETPp	Runtime(s)	#Area	AO(%)	Red(%)
ex2	93	6	org	46	68	---	12,006	---	17.65
			pro	24	56	3606.213	12,041	0.29	
BPF	235	10	org	210	125	---	23,318	---	15.20
			rand	0	106	---	23,601	1.21	
			pro	101	106	3,678.360	23,312	-0.03	
ARF	272	10	org	253	136	---	40,064	---	27.94
			rand	0	100	---	40,457	0.98	
			pro	96	98	4,823.671	40,247	0.46	
kim	157	26	org	405	192	---	5,173	---	44.27
			rand	0	118	---	5,209	0.70	
			pro	192	107	3,667.589	5,239	1.28	
maha	138	29	org	505	133	---	4,413	---	48.87
			rand	0	68	---	4,495	1.86	
			pro	212	68	3,790.705	4,542	2.92	
sehwa	149	31	org	523	117	---	5,133	---	41.88
			rand	0	68	---	5,254	2.36	
			pro	296	68	3,689.968	5,303	3.31	
Average (compared to org)								1.37	32.64

表 3. ATPG 実行結果

Circuit	method	#Total faults	#DT	#UT	#VUT	#ND	#TP	FE(%)	FC(%)	Runtime(s)	Red(%)
ex2	org	24,662	24,618	43	1	0	53	100.00	99.82	29,986	9.43
	pro	24,550	24,500	42	8	0	48	100.00	99.80	30,446	
BPF	org	48,216	43,774	4,368	74	0	404	100.00	90.79	49,609	71.29
	pro	47,958	47,896	55	7	0	116	100.00	99.87	43,180	
ARF	org	83,909	71,061	12,823	25	0	626	100.00	84.69	81,904	74,60
	pro	84,290	84,227	49	14	0	159	100.00	99.93	89,231	
kim	org	8,193	8,065	109	19	0	114	100.00	98.48	997	35.09
	pro	8,161	8,143	14	4	0	74	100.00	99.80	148	
maha	org	6,595	6,570	25	0	0	95	100.00	99.62	137	9.47
	pro	6,597	6,583	9	5	0	86	100.00	99.79	384	
sehwa	org	7,438	7,420	16	2	0	92	100.00	99.76	369	11.96
	pro	7,492	7,472	5	15	0	81	100.00	99.73	130	
Average											35.31

参考文献

1) S. Kajihara, I. Pomeranz, and K. Kinoshita, “Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits,” IEEE Transaction s on Computer Aided Design of Integrated Circuits and Systems, Vol.14, Issue12, pp.1496-1504, Dec.1995.
2) S. Kajihara, I. Pomeranz, K. Kinoshita, and S. M. Reddy, “On Compaction Test Sets by Addition and Removal of Test Vectors,” IEEE 12th VLSI Test Symposium, pp.202-207, 1994.
3) T. Hosokawa, S. Takeda, H. Yamazaki, and M. Yoshimura, “Controller Augmentation and Test Point Insertion at RTL for Concurrent Operational Unit Testing,” IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), pp.17-20, 2017.
4) T. Hosokawa, S. Takeda, H. Yamazaki, and M. Yoshimura, “A Test Register Assignment Method Based on Controller Augmentation to Reduce the Number of Test Patterns,” Proc. Int. Symp. on On-Line Testing and Robust System Design, pp.228-231, 2018.
5) M. J. Geuzebroek, J. Th. van der Linden, and A. J. van de Goor, “Test Point Insertion for Compact Test Sets,” International Test Conference 2000, pp.292-301, 2000.
6) S. Remersaro, J. Rajski, T. Rinderknecht, Sudhakar M. Reddy, and I. Pomeranz, “ATPG Heuristics Dependant Observation Point Insertion for Enhanced Compaction and Data Volume Reduction,” IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, pp.385-393, Oct. 2008.
7) M. Yoshimura, T. Hosokawa, and M. Ohta, “A Test Point Insertion Method to Reduce the Number of Test Patterns,” IEEE the 11th Asian Test Symposium, pp.298-304, Nov. 2002.
8) 徐浩豊, 細川利典, 山崎紘史, 新井雅之, 吉村正義 “論理故障テスト並列テスト化のための制御信号のドントケア割当て法” 信学技報, CPSY2021-56, DC2021-90, pp.67-72, 2022 年 3 月.
9) 徐浩豊, 細川利典, 吉村正義, 新井雅之 “並列テストのためのコントローラの制御信号のドントケア割当てアルゴリズム” 信学技報, vol. 122, no. 205, DC2022-24, pp. 37-42, 2022 年 10 月.
10) 徳田晴太, 細川利典, 吉村正義, 新井雅之 “楽観/悲観構造的記号シミュレーションを用いたテスト並列化のためのドントケア割当て及びテストスケジューリング法” 信学技報, vol. 124, no. 149, DC2024-25, pp. 46-51, 2024 年 8 月.