

データパスのテスト容易化機能的 k 時間展開モデル生成のための コントローラ拡大法

日大生産工（院）○兒玉 雄佑 西間木 淳 日大生産工 細川 利典 大院大 藤原 秀雄

1. はじめに

近年、半導体技術の進展に伴い、大規模集積回路 (Large Scale Integrated circuits: LSI) の規模が飛躍的に増大し、複雑化している[1]。これにより、設計される LSI が急速に大規模化しており、設計が困難になっている。この問題を解決するためには、LSI の設計生産性を向上させる必要があり、動作合成を用いた抽象度の高い動作レベルでの設計が提案されてきた[2]。一方、LSI の大規模化、複雑化に伴い、従来の LSI のスキャンテストのコストは劇的に増加している。そのため、動作合成の段階で非スキャンテストに基づいたレジスタ転送レベル(Register Transfer Level: RTL)データパスのテスト容易性を考慮したスケジューリングやバインディングが提案されている[3][4]。RTL データパスのテスト容易化手法の多くは、回路のデータパスのみに着目した手法であり、高い故障検出効率を得るためのテスト生成を行うためには、データパスとコントローラがテスト時に分離されていることが前提となる。しかしながら、テスト時にデータパスとコントローラを分離するために必要なハードウェアオーバーヘッドは設計によっては大きく、実際分離困難な場合もある。また、RTL 設計のデータパスとコントローラを分離せずにテストすることを前提とした RTL テスト容易化設計手法も提案されている[5]。文献[5]では、データパスのテスト容易な構造に基づいて、テスト時にデータパスを動作させる制御機能をコントローラに付加している。しかしながら、テスト生成においては、これまで提案されてきた回路構造のみに着目した順序回路のテスト生成アルゴリズム[6][7]を用いており、テスト容易な構造を制御するように拡大したコントローラの機能の通りテスト生成するとは限らない。文献[8]では、コントローラの機能を網羅するように機能的時間展開モデル[9]を生成し、テスト生成を行っている。しかしながら、コントローラの機能に基づき時間展開モデルを生成するため、時間展開数が大きくなる可能性がある。

以上より、文献[5]で提案されたような RTL データパスのテスト容易な構造を制御するコントローラ機能を付加し、その動作を実現できるような機能的時間展開モデルを生成することで、高い故障検出率を得ることが期待できる。本論文では、RTL のデータパスの順序深度に着目したテスト容易な構造とコントローラの機能からテスト容易な機能的時間展開モデルを生成し、その機能的時間展開モデルを実現可能とするようにコ

ントローラの拡大法を提案する。

2. コントローラ拡大法

コントローラとデータパスを分離せずにテストすることを前提としたテスト容易化データパスの実現のためのコントローラを拡大する手法として、文献[5]が提案されている。データパスとコントローラを分離せずにテストを行う場合、データパスのみのテスト容易性を強化したとしても、コントローラからテスト容易なデータパスの動作を制御することができない場合がある。文献[5]の手法では、データパスのテスト容易な動作を実現するために、コントローラにその制御機能を付加している。一般的な順序回路のテスト生成法[6][7]である時間展開モデルを用いてテスト生成を行う場合、回路構造のみからテスト系列を生成する。本論文では回路構造のみからテスト生成を行う時間展開モデルを構造的時間展開モデル[9]と呼ぶ。構造的時間展開モデルを用いたテスト生成を適用する場合、テスト容易な構造に基づいたデータパスの動作を実現するためにコントローラに付加した制御機能を用いてテスト生成するとは限らない。

3. 機能的時間展開モデルを用いたテスト生成

機能的時間展開モデルを用いたデータパスのテスト生成[8][9]とは、コントローラ部から入力される制御信号の系列とコントローラ部へ出力する状態信号系列を制約値として、ある一定時間分展開された時間展開モデルに対してテスト生成を行う手法である。機能的時間展開モデルでは時間展開数が有限になるので、組合せテスト生成アルゴリズムが適用可能になり、さらに制御信号系列と状態信号系列を制約として付加することにより、テスト生成が高速化される。

しかしながら、コントローラの動作を解析して生成した機能的時間展開モデルは、データパスの通常動作をモデル化したものであるため、時間展開数が大きくなる場合など、そのモデルの構造がテスト容易であるとは限らない。

4. テスト容易な機能的 k 時間展開モデル生成のためのコントローラ拡大法

本論文では、RTL のデータパス構造とコントローラの動作からテスト容易な機能的 k 時間展開モデルを生成し、その機能的時間展開モデルを回路動作可能なようにコントローラの拡大を行う方法を提案する。デー

タパス部は演算器、レジスタ、マルチプレクサ及びそれらを接続する信号線から構成されるので、コントローラの拡大を行うことにより、データパス部の大部分の故障はデータパス部のすべての演算器とレジスタを含んだテスト容易な機能的 k 時間展開モデルでテスト生成可能と考えられる。また、本手法では、データパス部の性能低下を避けるためにその部分のテスト容易化設計を行わないものとする。

機能的時間展開モデルを生成する上で、データパスの時間展開数を削減するために、コントローラを拡大すると、面積オーバーヘッドが増大する。また、小さなコントローラの拡大の場合、データパスの時間展開数が大きくなるためにテスト容易性が低くなる可能性がある。データパスの時間展開数とコントローラ的面積オーバーヘッドのトレードオフを考慮し、下記に示す問題を定式化する。

<テスト容易化機能的 k 時間展開モデル生成問題>

入力：

データパス動作グラフ[10]集合： $G_1, G_2, \dots, G_j$

状態遷移グラフ集合[10]： $H_1, H_2, \dots, H_j$

時間展開数： k

テスト容易化機能的 k 時間展開モデル数： m

出力：要素が j (自然数) 個の集合 $\{(G_1, H_1), (G_2, H_2), \dots, (G_j, H_j)\}$ から導かれるテスト容易化機能的 k 時間展開モデル集合： $M = \{(G^1, H^1), (G^2, H^2), \dots, (G^n, H^n)\}$

制約：下記の(1), (2), (3)のすべてを満たす。

- (1) $|M| = n \leq m$
- (2) H^i ($1 \leq i \leq n$: 整数) において、状態数が k 以下である
- (3) G^i ($1 \leq i \leq n$: 整数) 中のモジュール集合とレジスタ集合をそれぞれ MO_i , REG_i とすると

$\bigcup_{i=1}^n MO_i$ にはすべての演算器が、 $\bigcup_{i=1}^n REG_i$ に

はすべてのレジスタが含まれる。

最適化：評価関数 $COST$ の最大化

$$COST = \text{MAX} \left(\sum_{i=1}^n \text{Eval}(G_i, H_i) \right)$$

$$\text{Eval}(G_i, H_i) = \alpha \left(\frac{1}{1+T_i} \right) \times \beta \left(\frac{1}{k \times (\delta \times (C_i + RC_i) + 1)} \right) \times \gamma (R_i + O_i)$$

ただし、 α , β , γ は係数であり、 T_i はテスト時に追加する状態遷移数、 C_i は定数により制御される演算器数、 RC_i は再収斂構造を構成している演算器数、 R_i はレジスタ数、 O_i は演算器数である。 δ は定数であり、定数により制御されている演算器数と再収斂構造を構成している演算器数の和の係数と使用する。係数 α の項はコントローラ拡大による面積オーバーヘッドのコス

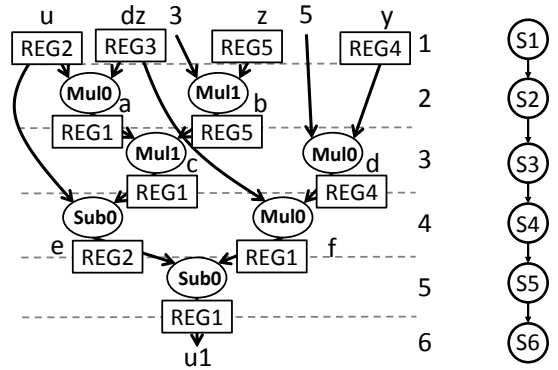
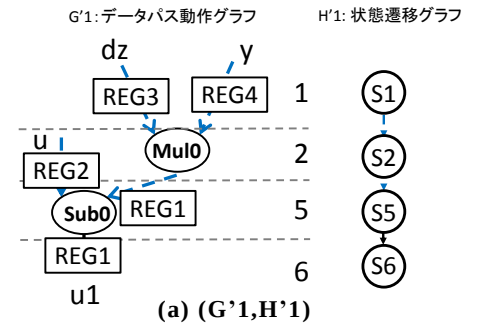
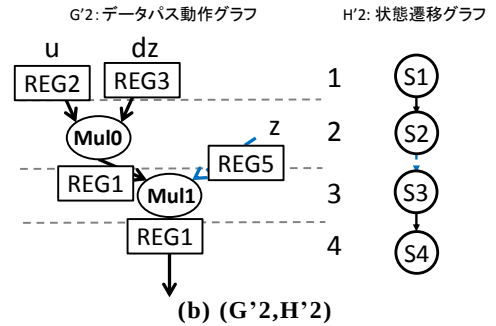


図 1. ex2 の DFG と状態遷移図



(a) (G^1, H^1)



(b) (G^2, H^2)

図 2. テスト容易化機能的 4 時間展開モデル例

トを表す。係数 β の項はテスト生成時間のコストを表す。係数 γ の項はテストカウンタ[10]の面積オーバーヘッドのコストを表す。

例 1. 図 1 に ex2[10] のスケジューリング済、バインディング済の DFG (Data Flow Graph) と ex2 のコントローラを表現する状態遷移図を示す。本論文では、バインディング済みの DFG をデータパス動作グラフ G , 状態遷移図を状態遷移グラフ H として説明する。

例 2. 図 2 は図 1 から導かれる 2 個のテスト容易化機能的 4 時間展開モデル例である。各テスト容易化機能的 4 時間展開モデルを説明する。

(a) の G^1 は G に対して変数 u_1 の REG_1 を検出レジスタとし、変数 u の REG_2 と時刻 5 の Sub_0 の左入力間の辺を無視[10]し、時刻 2 の Mul_0 と時刻 4 の Mul_0 を頂点連結[10]する。次に、変数 u の Mul_0 への左入力辺を

変数 dz の $Mul0$ への左入力辺に置換[10]し、次に、変数 dz の $Mul0$ への右入力と変数 y の $Mul0$ への右入力辺に置換する。最後に、不要な頂点や辺を削除し $G'1$ が生成される。また $H'1$ は $G'1$ と H から生成され、 $G'1$ 生成時に無視操作、頂点連結操作、置換操作、時刻更新で新たな辺を追加したため、状態 $S1$ から状態 $S2$ へのテスト用の状態遷移と、状態 $S2$ から状態 $S5$ へのテスト用の状態遷移が追加される。

(b) の $G'2$ は G に対して変数 c の $REG1$ を検出レジスタとし、変数 z の $REG5$ と時刻 3 の $Mul1$ の右入力間の頂点を無視する。最終的に、不要な頂点や辺を削除し、 $G'2$ が生成される。また $H'2$ は $G'2$ と H から生成され、 $G'2$ 生成時に無視操作、時刻更新で新たな辺を追加したため、状態 $S2$ から状態 $S3$ へのテスト用の状態遷移が追加される。

5. 実験結果

本論文では、バインディング済みの $DFG(ex2[11], ex4[11], DFCT[12])$ とバインディング済みの $CDFG(Sehwa[11], Maha[11], Kim[11])$ に対して、動作合成ツール $PICHTY$ で合成した RTL 回路を対象にテスト生成実験を行った。また、 RTL データパスとコントローラを基にコントローラの拡大を行い、テスト容易化機能的 k 時間展開モデルを生成した。テスト生成は機能的時間展開モデルを用いたテスト生成可能なツール $STAGY[9]$ を利用し、データパス内部の演算器にのみ縮退故障を設定した。比較対象として、以下の3つの手法を挙げる。

- 1) オリジナルのコントローラの動作に基づく構造的時間展開モデルによるテスト生成
- 2) 機能拡大を行ったコントローラ動作に基づく構造的時間展開モデルによるテスト生成
- 3) 文献[8]の手法によるオリジナルのコントローラの動作に基づく機能的時間展開モデルに対するテスト生成

構造的時間展開モデルのテスト生成時の時間展開数は、 $ex2, ex4, DFCT$ 回路が 20, $Sehwa, Maha, Kim$ 回路は 30 に設定している。また、手法 3) の機能的時間展開モデルは、コントローラを表現する FSM の状態遷移を全て網羅して生成した。したがって、3) の機能的時間展開モデル数は $ex2, ex4, DFCT$ 回路が 1, Kim 回路が 3, $Sehwa, Maha$ 回路が 4 である。

表 1 に本手法で生成したテスト容易化機能的 k 時間展開モデルの詳細(最長機能動作レイテンシ、モデル数 m 、時間展開数 k 、追加状態遷移数、面積オーバーヘッド)を示す。 $Sehwa$ 回路に対し、テスト容易化機能的 k 時間展開モデルを生成した結果、最長機能動作レイテンシが 17 であるのに対し、時間展開数 k は 4 で実現するこ

とができた。また、コントローラ拡大による回路全体の面積オーバーヘッドも 0.9%に抑えることができた。また、全ての回路の面積オーバーヘッドは平均で 0.62%に抑えることができた。次に各回路に対し、手法 1), 手法 2), 手法 3), 提案手法でテスト生成を行った結果を表 2 に示す。 $ex2$ 回路では 4 つの手法のうち、提案手法が故障検出率 100%と、最も高い数値を得ることができた。 $ex4, DFCT$ 回路では、手法 2) や手法 3) の故障検出率が提案手法よりも高い結果となった。提案手法はテスト時に制約を与え回路動作を制限している。このことにより、ある故障を検出する時に、偶発的に検出できる他の部分の故障を検出できなかったと考えられる。手法 2) や手法 3) ではそのような偶発的に検出できる故障が多数存在したため、 $ex4, DFCT$ 回路では高い故障検出率を達成したと考える。 $Sehwa$ 回路では、手法 1) の故障検出率が 75.95%, 手法 3) の故障検出率が 95.15%であるのに対し提案手法では故障検出率 100%を達成できた。手法 2) も故障検出率 100%を達成しているが、テスト生成時間において、提案手法は手法 2) に対して、約 30 倍高速にテスト生成を行うことができた。 $Maha, Kim$ 回路に対しても同様の結果となっており、提案手法が故障検出率、テスト生成時間において最良の結果となっている。 $Sehwa, Maha, Kim$ 回路はコントローラの状態遷移に分岐が存在する回路である。 $ex2, ex4$ 回路のように回路動作が一意に定まっていなかったため、 $Sehwa, Maha, Kim$ 回路では偶発的に検出できる故障が少なく、テスト生成が困難であったと考えられる。提案手法は、コントローラの機能拡大と、拡大した機能を有効にするための制約を与えたことにより、テスト生成の探索空間を削減することができ、高速かつ高い故障検出率を達成することができたと考える。また、テスト系列長の項目でも、他の手法では長くなる傾向にある $Sehwa, Maha, Kim$ 回路において、提案手法は最良の結果を得ている。

6. おわりに

本論文では、 RTL データパス回路に対しコントローラの拡大を行い、テスト容易化機能的 k 時間展開モデルを生成し、生成したテスト容易化機能的 k 時間展開モデルによるテスト生成を行う手法を提案した。本手法では手法 2) のテスト容易化手法と比較し、 $Sehwa, Maha, Kim$ 回路では故障検出率は平均で 0.3%向上、テスト生成時間は平均で約 8 倍高速化、テスト系列長は平均で約 20%に短縮することができた。

今後の課題として、コントローラを含んだ回路全体で高い故障検出率を得るために、手法を組合せて実験を行うこと、大規模回路で実験を行うことなどが挙げられる。

参考文献

- [1] 藤原 秀雄, " デジタルシステムの設計とテスト", 工学図書株式会社, 2004.
- [2] M.C.McFarland, A.C.Parker, R.Camposano, "The high-level synthesis of digital systems", Proc. IEEE, pp.301-318, 1990.
- [3] M.T.-C. Lee, W. H. Wolf, and N. K. Jha, "Behavioral synthesis for easy testability in data path allocation", Int. Conf. Computer Design, pp. 29 -32 ,Oct.1992.
- [4] M.T.-C. Lee, W. H. Wolf, and N. K. Jha, "Behavioral synthesis for easy testability in data path scheduling", IEEE International Conference on Computer-Aided Design, pp.616 -619, 1992.
- [5] L.M.FLottes, B.Rouzeyre, L.Volpe,"A CONTROLLER RESYNTHESIS BASED METHOS FOR IMPROVING DATAPATH TESTABILITY", IEEE International Symposium on Circuits and Systems, pp. 347 -350, May 2000.
- [6] W.T.Cheng, "The back algorithm for sequential test generation", Proc. 1988 IEEE Int. Conf. on Computer Design, pp. 66-69, Oct. 1988.
- [7] T.M. Niermann and J.H.Patel, "HITEC : A Test Generation Package for Sequential Circuit", in Proc. Of the European Design Automation Conf., pp.214-218, Feb.1991 .
- [8] Toshinori Hosokawa, Teppei Hayakawa, and Masayoshi Yoshimura, " A Comprehensive Functional Time Expansion Model Generation Method for Datapaths Using Controllers", The Eleventh Workshop on RTL and High Level Testing (WRTL'T'10), pp.131-138, Dec. 2010.
- [9] Kazuya Sugiki, Toshinori Hosokawa, and Masayoshi Yoshimura, "A Test Generation Method for Datapath Circuits Using Functional Time Expansion Models", The Ninth Workshop on RTL and High Level Testing (WRTL'T'08), pp.39 -44, Nov. 2008.
- [10] 兒玉 雄佑, 西間木 淳, 増田 哲也, 細川 利典, 藤原 秀雄, "データパス回路の機能的 k 時間展開モデル生成のためのコントローラ拡大法について" 第 69 回 FTC 研究会
- [11] M.T.-C.Lee, "High-Level Test Synthesis of Digital VLSI Circuits", Artech House Publishers, 1997.
- [12] Ian G. Harris and Alex Oraign , "Testability Improvement in High-Level Synthesis Through Reconvergence Reduction", In Pro-ceedings of the Asilomar Conference on Signals, Systems and Computers, pp.1919-203 ,1996.

表 1. テスト容易化機能的 k 時間展開モデル詳細

回路名	最長機能動作 レイテンシ	機能的k時間展開 モデル数m	時間展開数k	追加 状態遷移数	面積 オーバーヘッド(%)
ex2	6	2	4	3	0.20
ex4	5	2	4	1	0.12
DFCT	7	3	4	3	0.18
Sehwa	17	3	5	3	0.9
Maha	18	3	3	3	1.03
Kim	15	5	3	4	1.33

表 2. テスト生成結果

回路名	故障検出率(%)				テスト生成時間(sec)				テスト系列長			
	1)	2)	3)	提案手法	1)	2)	3)	提案手法	1)	2)	3)	提案手法
ex2	99.98	99.99	99.98	100.00	1630	1866	564	1045	940	1300	618	1036
ex4	99.27	99.91	99.86	99.59	427	344	44	55	740	620	470	331
DFCT	99.47	99.45	99.68	99.35	5261	7099	816	1959	840	1020	336	628
Sehwa	75.95	100.00	95.15	100.00	741	494	321	17	1050	3270	744	586
Maha	95.18	100.00	93.07	100.00	303	415	780	115	1260	3180	629	528
Kim	88.64	99.55	98.80	99.67	1591	347	544	20	2460	780	1807	351