ISSN 2186-5647

2-36

テスト容易化データパスに対する 機能的時間展開モデル生成のためのコントローラ再合成法

日大生産工(院)　○森田 菜留美
日大生産工　　細川　利典

1. はじめに

近年，半導体技術の急速な進歩により，大規模集積回路 (Large Scale Integrated circuits :LSI) が大規模化し，回路が複雑化してきている．従来 LSI はレジスタ転送レベル(Register Transfer Level: RTL)での設計が主流であるが，LSI の大規模化に伴い，RTL での設計が困難になってきている[1]．それゆえ，RTL より抽象度の高い動作レベルで設計された記述を RTL 回路に変換する技術である動作合成[2]が注目されている．

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を基本とした順序回路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[3-7]．これらの手法はデータパスのみをテスト容易化しているため，データパスとコントローラを分離してテストするか，あるいはコントローラをスキャン設計することが前提となる．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テスト容易化手法も提案されている[8][9]．文献[8]では，テスト実行時間の短縮を目的とした RTL データパスの階層テスト生成[10]のためのテスト環境生成容易化のデータパスとコントローラのテスト容易化設計を提案している．文献[9]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，これまで提案されてきた順序回路のテスト生成[11]では，回路構造のみからテスト系列を生成するアルゴリズムを用いており，テスト容易な構造を制御するように付加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ず，テスト容易化の意図通りテスト生成できない可能性がある．

本論文では，RTL のデータパスの構造とコントローラの動作からテスト容易な機能的時間展開モデル[12][13]を生成し，その機能的時間展開モデルを実現可能なようにコントローラの再設計を行う方法を提案する．

2. 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[12][13]を用いたデータパスのテスト生成とは，コントローラ部に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．

回路構造のみに注目した時間展開モデルである構造的な時間展開モデルの展開数は回路中にフィードバックループ[1]が存在すると不確定になり，図 1 のように無限に展開可能なモデルとなる．一方，図 2 のように機能動作の情報として制御信号系列と状態信号系列を利用し，時間展開モデルを機能的に生成した機能的時間展開モデルの時間展開数は有限になる．展開数が有限ならば，組合せテスト生成アルゴリズムが適用可能になる．

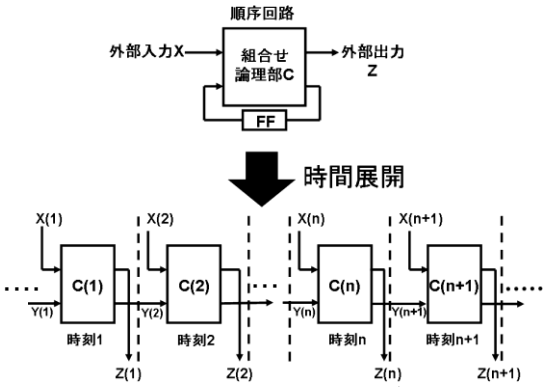

図 1. 構造的時間展開モデル

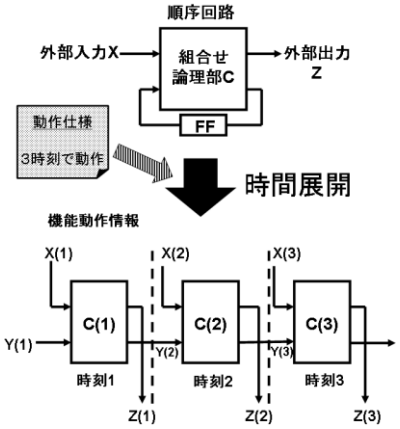

図 2. 機能的時間展開モデル

Re-synthesis of cntroller for testability
to genetrate functional time expansion models

Narumi MORITA and Toshinori HOSOKAWA

3. テスト容易なデータパス実現のためのコントローラ再合成

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にコントローラを再合成する手法として，文献[9]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．そのため，文献[9]の手法では，データパスにのみテスト容易化を考慮した回路に対し, テスト容易なデータパスの動作を制御可能なように, その制御機能をコントローラに付加している．

一般的な順序回路のテスト生成法である構造的時間展開モデルを用いてテスト生成を行った場合, 回路構造のみからテスト系列を生成するため, テスト容易な構造に基づいてデータパスの動作を制御するように付加した機能の通りテスト生成するとは限らない．一方，文献[13]で提案されている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を用いれば, テスト生成は高速になると報告されている．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ので, そ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時間展開モデル生成のためのコントローラ再合成

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時間展開モデル[12]を生成し，その機能的時間展開モデルを実現可能なようにコントローラの再設計を行う方法を提案する．それゆえ，データパス部の大部分の故障はテスト容易な時間展開モデルでテスト生成可能と考えられる.

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小さくするために, コントローラを大幅に変更すると, 面積オーバーヘッドが増大する．また，オリジナルのコントローラをほとんど変更しない場合, 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の面積オーバーヘッドのトレードオフを考慮し，合成された RTL データパスとコントローラを入力とし, 以下の指針で機能的時間展開モデルの生成とコントローラの再合成を行う.

(指針 1)
オリジナルのコントローラの機能を可能な限り使用する.

(指針 2)
すべての機能的時間展開モデルの時間展開数を k 以下にする(k は正の整数).

(指針 3)
コントローラの状態レジスタ数をオリジナルから j ビット以下の増加に抑える（j は 0 以上の整数）.

(指針 4)
生成した機能的時間展開モデルを動作させるための状態信号を生成するためのブロックを付加する.

(指針 5)
機能的時間展開モデルの種類を m 個以下にする（m は 2 のべき乗－1 が望ましい）.

(指針 6)
機能的時間展開モデル内に同じ演算器やレジスタの出現回数をできる限り抑える.

上記の指針に基づき，以下の問題を考える.

＜問題定式化＞
入力 : RTL データパス，コントローラ,
時間展開数 k，機能的時間展開モデル数 m
追加状態レジスタ数 j
出力 : テスト容易化データパスを制御する
コントローラ,
m 個以下の機能的時間展開モデル
(時間展開数は k 以下)
制約 : すべてのレジスタ・演算器は少なくとも 1 つ以上の機能的時間展開モデルに存在する.
最適化 : コントローラ面積最小化

また，6 つの指針により，コントローラの再合成を行うため, 機能的時間展開モデルを生成するとき，以下の優先順位で生成する.
(1)オリジナルのコントローラを利用
(2)コントローラの制御信号の X を利用
(3)できる限り状態レジスタを増加させず，状態と状態遷移数追加

図 3 にバインディング済み SDFG，図 4 に図 3 の SDFG の RTL データパス回路，図 5 に図 3 のコントローラの状態遷移図を示す. 乗算器 M1 のテスト生成について説明する．k=5，j=0 とする．例として，M1 を必ず含むテスト容易化時

間展開モデルを生成するために，M1 周辺の制御の状況を解析する．

まずは M1 から外部出力までの経路を探索する．M1 から外部出力までの最短経路は

M1→R2→SU→R0→u1

の経路である．この経路は機能的に実現されているので，オリジナルのコントローラの

s2→s3→s4

を使用することができる．

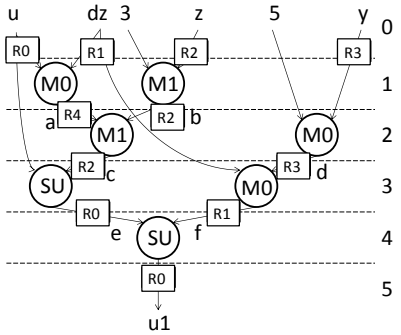

図 3. バインディング済み SDFG

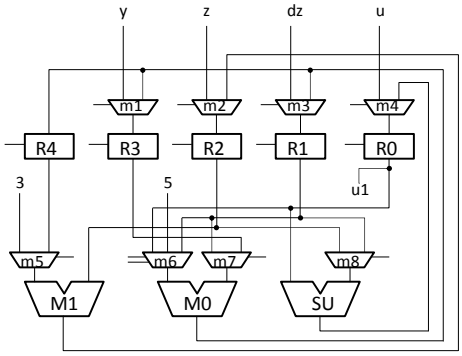

図 4. データパス

また，M1 に対し，テスト生成するためには，入力となるレジスタ R4，R2 の値を制御する必要がある．R2 は外部入力 z の値を格納することができるので，s0 の状態で R2 には z の値が格納される．しかしながら，1 時刻目の M1 の左入力は定数 3 であるため，M1 の左入力を制御することができない．さらに，R4 は内部変数 a を記憶するため，R4 の入力となる値を格納する必要がある．R4 を制御するためには，M0 を制御する必要があり，その入力は R0 と R1 でそれぞれ外部入力 u，dz の値が格納される．s0→s1→s2 とオリジナルのコントローラを使用した場合，s1 において M1 の左入力に定数 3 が存在するため，時刻 2 の R2 の値をを制御できない．よって，優先順位(1)を満たすことができないため，優先順位の(2)を考える．s0 の次状態で M1 を制御可能にするため，s1 でコントローラの制御信号の X を利用して M1 の制御の可能性を検討する．しかしながら，機能上，s1 で M1 の両入力を任意の値に制御することは不可能であるので，コントローラの制御信号の X を利用することはできない．

次に優先順位の(3)について考える．オリジナルの状態レジスタ数は 3 であり，状態は 2 の 3 乗個表現することができる．よって， 8 状態表現でき，無効状態が 2 状態あるので，状態を 1 つ追加しても，状態レジスタ数が変化することはない．状態 s2 において，M1 の左入力を制御するためには 1 つ前の時刻で M0 の演算操作を実行し，M1 の左入力レジスタ R4 に値を設定する必要がある．さらに，M1 の右入力を制御するために，1 つ前の時刻で外部入力 z から M1 の右入力レジスタ R2 に値を設定しておく必要がある．M0 の演算を実行し，R4 に値を設定し，かつ外部入力 z から R2 に値を設定するような状態はオリジナルのコントローラには存在しないため，新たに状態 s6 を追加し，s0 から s6，s6 から s2 への状態遷移を追加する．よって，M1 をテストするためのコントローラの動きは

s0→s6→s2→s3→s4

となり，指定した k＝5 以下の時間展開数で機能的時間展開モデルを実現することができる．

同様にして，他すべての演算器 M0，S0 に対して機能的時間展開モデルを生成するためにコントローラを再合成すると，図 6 のようなコントローラが合成できる．

機能的時間展開モデルは以下の通りになる．

M1：

u→R0，dz→R1 (s0) ⇒ z→R2，R0×R1→R4 (s6)
⇒ R4×R2→R2 (s2) ⇒ R0−R2→R0 (s3) ⇒ e (s4)

M0：

u→R0，dz→R1，y→R3，dz→R1 (s0)
⇒ R1×R3→R1，R0−R2→R0 (s3)
⇒ R0−R1→R0 (s4) ⇒ u1 (s5)

S0：

u→R0, dz→R1 (s0) ⇒ R0−R1→R0 (s4) ⇒u1 (s5)

よってデータパスに対してテスト生成を行う場合，図 6 に示すように 1 つの状態と 4 つの状態遷移を追加することで，最大でも 5 時間の時間展開数で機能的時間展開モデルを実現することができる．

5. 実験結果

本論文では，図 3 のバインディング済み SDFG に対して，動作合成ツール PICTHY を利用し生成した RTL データパスとコントローラを基に，

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なようにコントローラを再設計した．また，再設計した回路に対し，機能的時間展開モデルによるテスト生成を行った．比較対象として，PICTHY で生成した RTL 回路に対し，構造的時間展開モデルによるテスト生成を行った．

表 1 に再設計前の回路(ex2(LEA＋org))と再設計後の回路(ex2(LEA+re_ctrl))のコントローラの状態遷移数、状態レジスタ数、状態数を示している．表 2 に 2 つの回路に対し，テスト生成を行った結果を示している．

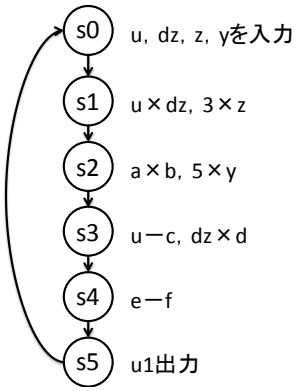

図 5. オリジナル状態遷移図

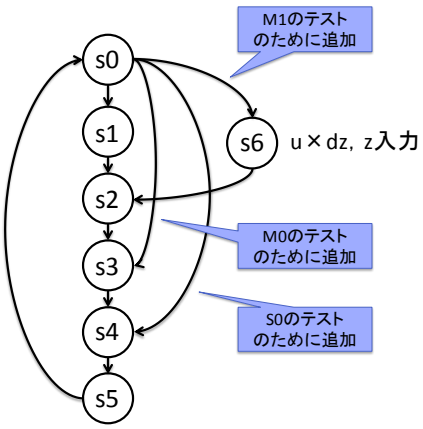

図 6. 再合成後状態遷移図

表 1. コントローラの状態遷移

回路名	状態遷移数	状態レジスタ数	状態
ex2(LEA＋org)	6	3	6
ex2(LEA＋re_ctrl)	10	3	7

表 2. テスト生成結果

回路名	時間展開モデル	故障検出効率	故障検出率	CPUtime（sec）	冗長故障	打ち切り故障
ex2(LEA＋org)	構造的					
	機能的	99.67%	99.65%	1067.33	6	116
ex2(LEA＋re_ctrl)	構造的					
	機能的	98.71%	98.36%	1660.45	125	481

参考文献

1) 藤原 秀雄,"ディ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) 高崎智也,井上智生,藤原秀雄,"無閉路部分スキャン設計に基づくデータパスのテスト容易化高位合成におけるバインディング手法", 電子情報通信学会論文誌(DI),Vol.J83-D-I No.2 ,2000.

4) V.Femandez and P.Shchez,"Partial Scan High-Level Synthesis", 1996 European Design and Test Conference (ED&TC '96),1996.

5) M.T.-C. Lee, W. H. Wolf, and N. K. Jha, "Behavioral synthesis for easy testability in data path allocation" , Int. Conf. Computer Design, Cambridge, MA, Oct.1992.

6) M.T.-C. Lee, W. H. Wolf, "Behavioral synthesis of highly testable data paths under the non-scan and partial scan environments" , Dallas, Tx, June 1993.

7) M.T.-C. Lee, W. H. Wolf, and N. K. Jha, "Behavioral synthesis for easy testability in data path scheduling", Santa Clara, CA,1992.

8) Michiko Inoue, Kazuhiro Suzuki, Hiroyuki Okamoto and Hideo Fujiwara, "Test synthesis for datapaths using datapath-controller functions," IEEE the 12th Asian Test Symposium (ATS '03), pp.294-299, Nov. 2003.

9) L.M.FLottes, B.Rouzeyre, L.Volpe,"A CONTROLLER RESYNTHESIS BASED METHOS FOR IMPROVING DATAPATH TESTABILITY", IEEE International Symposium on Circuits and Systems, May 2000.

10)B.T.MurrayandJ.P.Hayes,"Hierarchical test generation using pre computed tests for modules,"IEEE Trans.Computer-Aided Design, Integrated Circuits & Syst., vol.9, no.6, pp.594-603, June 1990.

11)M.Abramovici,M.A.Breuer,and.D.Friedman, "Digital systems testing and testable design", IEEE Pres, 1995

12)Kazuya SUGIKI, Toshinori HOSOKAWA , Masayoshi YOSHIMURA , " A Test Generation Method for Datapath Circuits Using Functional Time Expansion Models" , WRTLT'08, 2008.

13)HAYAKAWA