

木に着目した L1L2* スキャン設計の適用手法

Tree-focused L1L2* Scan Design

上岡 真也[†]
Shin'ya Ueoka

岩田 大志[‡]
Hiroshi Iwata

山口 賢一[‡]
Ken'ichi Yamaguchi

1 まえがき

同期式回路設計では、LSI の高集積化・高速化に伴い、クロックツリーによる面積オーバーヘッドやクロックスキューが顕著となっている。そのため近年、大域クロックに依存せずに回路の制御を行う非同期式回路が注目されている。非同期式回路には省電力性やクロック遷移時の耐電磁放射線などといった利点がある。しかし非同期式回路は、組合せループの存在や、大域クロックが無いなどの要因によって、テストが困難であるため、非同期式回路のための EDA ツールやテスト容易化設計 (DFT) 手法が必要とされている。

同期式回路のための DFT 手法である L1L2* スキャン設計 [1] を非同期式回路へと適用する手法が、Beest ら [2] によって提案された。この手法は同期式回路への完全スキャン設計と同様に、非同期式回路を組合せ回路部とスキャン素子へと分割する。そのため組合せ回路部に対して任意のパタンの印加と応答の観測ができ、高い故障検出率を実現できる。またこの DFT 手法により、既存の同期式回路用の EDA ツールを使用できるという利点がある。しかし文献 [2] では L1L2* スキャン設計の適用アルゴリズムが示されていない。

文献 [3] では非同期式回路に L1L2* スキャン設計を適用するアルゴリズムが提案されている。この手法は L1L2* スキャン設計適用後の回路面積増加を最小に抑えるために、回路を網羅的に走査するアルゴリズムであった。そのため計算時間が大きく、大規模回路に対しては実用的な時間で適用できなかった。よって本稿では実用的な時間で適用できる、高速な L1L2* スキャン設計の適用アルゴリズムを提案する。

本稿は、まず 2 節で非同期式回路に対する L1L2* スキャン設計について述べる。3 節では提案手法を説明する上で必要となる諸定義を行う。4 節では本稿で提案する L1L2* スキャン設計の適用アルゴリズムを説明する。5 節では提案手法を適用した結果について議論する。最後に、6 節で本稿での結論を述べる。

2 L1L2* スキャン設計

非同期式回路では多くの組合せループが発生する。組合せループの内部状態を外部から制御することは難しく、

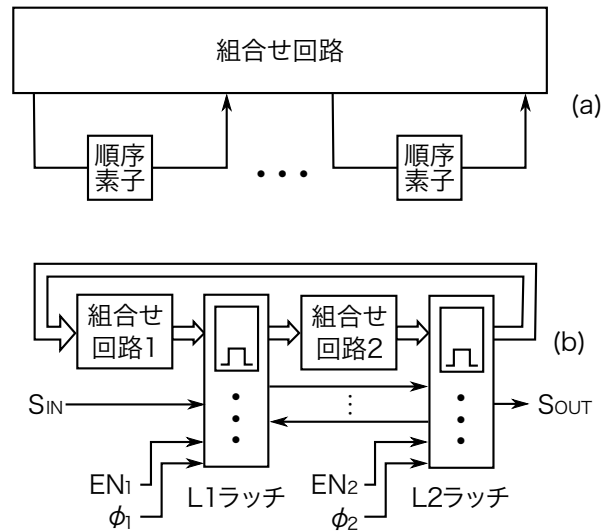


図 1: L1L2* スキャン設計の構造

テスト生成が困難である。文献 [4] では非同期式回路の組合せループに対して LSSD のスキャン素子を挿入し、非同期式回路を組合せ回路部とスキャン素子へと分割することで、LSSD を実現する手法を紹介している。しかし LSSD ではスキャン素子にダブルラッチを用いるため、DFT 適用時に大きな面積増加が生じる。L1L2* スキャン設計ではシングルラッチを用いることで LSSD で顕著となる面積増加を緩和する。本節では L1L2* スキャン設計に必要な回路の設計変更およびテスト方法を述べる。

2.1 回路の設計変更

L1L2* スキャン設計で対象とする回路モデルは図 1(a) に示すように、組合せ回路部と順序素子から成る回路とする¹。L1L2* スキャン設計の適用には全ての順序素子をスキャン機能を持つスキャン素子に置換する。そして図 1(b) に示すように、全てのスキャン素子を L1 ラッチ集合、L2 ラッチ集合へと分割する。このとき対象回路が 2 分割不能な構造の場合、シングルラッチ (図 2(a)) をダブルラッチ (図 2(b)) へと置換することで、2 分割可能となるよう設計変更する。スキャンパスは、L1 ラッチおよび L2 ラッチを交互に接続し、それぞれ対を成すラッチをマスタースレーブ FF によるスキャン FF とし

[†] 奈良工業高等専門学校 電子情報工学専攻

[‡] 奈良工業高等専門学校 情報工学科

¹ ただし対象回路に組合せループが存在する場合は LSSD のスキャン素子を挿入し、全ての組合せループを取り除く。

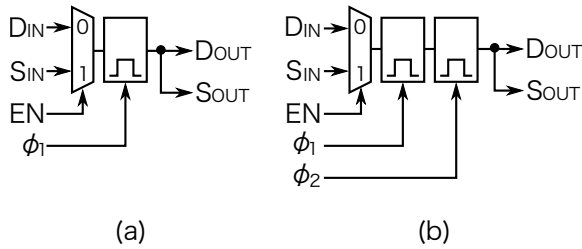


図2: シングルラッチ (a) とダブルラッチ (b)

表1: L1L2*スキャン設計のテスト信号

	ϕ_1	ϕ_2	EN ₁	EN ₂
スキャンシフト	\square	\square	1	1
L1 テスト	\square	\square	0	1
L2 テスト	\square	\square	1	0
通常動作	1	1	0	0

て動作させることで、任意のパタンの印加と応答の観測を可能とする。

2.2 テスト

L1L2*スキャン設計を適用した回路のテストは、L1テストおよびL2テストに分割して行う。L1テストではL1ラッチにより組合せ回路1にボタンを印加し、L2ラッチによって応答を取り込む。またL2テストではL2ラッチにより組合せ回路2にボタンを印加し、L1ラッチによって応答を取り込む。

L1L2*スキャン設計の信号の値を表1に示す。スキャンシフト時にはスキャンイネーブル EN₁ = 1, EN₂ = 1 に設定し、スキャンイン S_{IN} およびスキャンアウト S_{OUT} からスキャンシフトを行う。L1テスト時には EN₁ = 0, EN₂ = 1 に設定し、組合せ回路1に対してテストを行う。またL2テスト時には EN₁ = 1, EN₂ = 0 に設定し、組合せ回路2に対してテストを行う。そして通常動作時には EN₁ = 0, EN₂ = 0 に設定する。

3 諸定義

非同期式回路に対しL1L2*スキャン設計を適用するために、シングルラッチをダブルラッチに置換して回路を2分割する。本稿ではダブルラッチに置換するシングルラッチを求める手法を提案する。文献[3]では対象となる非同期式回路をグラフ化したモデルが定義されており、本稿でもこのモデルを用いる。

定義 1. 非同期式回路の順序素子を頂点集合 V 、および任意の順序素子間の接続関係を表す辺集合 E からなる有向グラフ $G = (V, E)$ を **S グラフ** と呼ぶ。□

L1L2*スキャン設計適用後の回路は、L1ラッチは組合せ回路1のみを駆動し、その応答はL2ラッチによって取り込まれる。またL2ラッチは組合せ回路2のみを駆

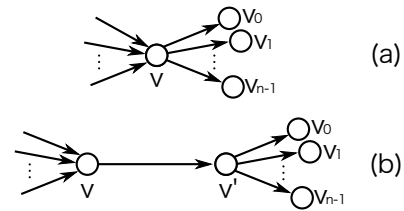


図3: 頂点の多重化操作

動し、その応答はL1ラッチによって取り込まれる。したがってL1L2*スキャン設計の適用後のSグラフは定理1の性質がある。

定理 1. ある回路のSグラフが2部グラフであれば、L1L2*スキャン設計が可能である。□

L1L2*スキャン設計適用時のシングルラッチからダブルラッチへの置換をSグラフで表現すると、図3となる。ラッチの追加前を図3(a)とすると、追加後の頂点は図3(b)となる。グラフ上でこれらの操作を**頂点の多重化**と定義する。

定義 2. 有向グラフ $G = (V, E)$ に含まれる、ある頂点 $v \in V$ において、新たな頂点 v' を V に追加し、 v の出辺 $e_i = (v, v_i^t)$ それぞれを辺 $e'_i = (v', v_i^t)$ に置換する。そして新たな辺 $e = (v, v')$ を E に追加する。これらの操作を**頂点の多重化**と定義する。□

本稿で提案するL1L2*スキャン設計の適用手法は、定義2に示す頂点の多重化操作のみでSグラフを2部グラフに変換する手法である。そのため2部グラフ変換可能な、多重化する頂点を求める必要がある。この問題を定義3に定式化する。

定義 3. ある有向グラフ $G = (V, E)$ において、ある頂点集合 $V' \subseteq V$ に含まれる全ての頂点を多重化することで、 G が2部グラフとなるような V' を求める問題を**2部グラフ変換における頂点多重化問題**と定義する。□

全ての頂点を多重化することにより対象のSグラフは2部グラフとなるが、それによって発生する面積オーバーヘッドはLSSDと等しくなる。したがって $|V'|$ が小さくなるような V' を求めることで、適用後の面積オーバーヘッドを抑える。

次に提案アルゴリズムを説明をする上で必要となる**補辺**を定義し、有向連結グラフの2部グラフに関する定理について説明する。

定義 4. あるグラフ $G = (V, E)$ および G の全域木 $G^T = (V, E^T)$, $E^T \subseteq E$ としたとき、辺 $e \in (E - E^T)$ を G の G^T に対する**補辺**と定義する。□

定理 2. ある連結グラフを G 、 G の全域木を G^T 、そして G の G^T に対する補辺集合を E^C とする。このとき

任意の補辺 $e \in E^C$ と全域木 G^T が作る閉路の頂点数が偶数のとき、グラフ G は2部グラフである。□

証明. G^T は木であるので、2彩色可能である。そのため G^T の2彩色時に、任意の補辺 $e = (v^s, v^t)$ において v^s, v^t がそれぞれ異色となると、 G は2彩色可能であるといえる。ここで、閉路の頂点数が偶数のときかつその時に限り、閉路は2部グラフである。補辺 e と G^T が作る閉路に着目すると、任意の v^s, v^t がそれぞれ異色となるための必要十分条件は、 e と G^T が作る閉路長が偶数であることである。したがって G^T の2彩色時に任意の v^s, v^t がそれぞれ異色となるとき、 G は2彩色可能となり、 G は2部グラフであるといえる。□

定理2は無向グラフについて議論したが、有向グラフを扱うには、辺の方向を無視すると閉路となる部分グラフの頂点数に着目することで、定理2が成り立つ。このような部分グラフを**半有向閉路**と定義する。

定義 5. 有向グラフにおいて、有向辺を無向辺に置換したとき閉路となる部分グラフを、**半有向閉路**と定義する。□

4 提案手法

本節では、2部グラフ変換における頂点多重化問題の多重頂点数が小さくなる解を求める手法を提案する。本手法は、与えられたSグラフが2部グラフを満たす制約条件と最適化目標を設定し、整数計画問題によってその解を求める。文献[3]はSグラフを2部グラフへと変換するために、任意の閉路を偶数とする制約条件を生成していた。そのため回路の全ての閉路を抽出する必要があり、非常に大きな時間を要した。提案手法では木が2部グラフであることに着目し、グラフの走査回数を減らすことで制約条件の生成を高速化する。

制約条件については、Sグラフの任意の補辺と全域木が作る半有向閉路の頂点数を偶数とするための制約条件式を作成する(定理2)。与えられたSグラフを G 、 G の全域木を G^T 、 G の G^T に対する補辺集合を E^C とする。ここである補辺 $e \in E^C$ について考える。 e と G^T が作る半有向閉路を $P = (V^P, E^P)$ とすると、多重化により P の頂点数に影響する頂点は、 P に含まれる頂点のうち入辺および出辺が存在する頂点である。このような頂点集合を $V' \subseteq V^P$ とする。ここで、多重化前の P の半有向閉路長が偶数の場合 $|V'|$ は偶数となる。そのため V' のうち偶数個の頂点を多重化すると P の閉路長は偶数となる。また多重化前の P の半有向閉路長が奇数の場合 $|V'|$ は奇数となるため、 V' のうち奇数個の頂点を多重化することで P の閉路長は偶数となる。

V' の各頂点 $v_i \in V'$ に対応付けられた変数を $x_i \in \{0, 1\}$ とし、 $x_i = 1$ のとき頂点を多重化し、 $x_i = 0$ の

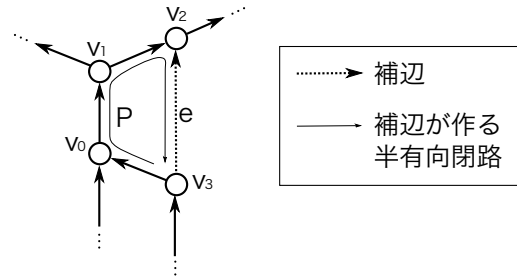


図4: e と全域木が作る半有向閉路 P

とき頂点を多重化しないとする。このような変数集合を $\{x_0, x_1, \dots, x_{n-1}\}$ とすると、制約条件は式(1)となる。

$$\sum_{i=0}^{n-1} x_i = 2m + (n \bmod 2) \quad (1)$$

$(x_i \in \{0, 1\}, m \text{ は整数})$

図4では例として補辺 e と全域木が作る半有向閉路の P を示している。 P において入次数・出次数が共に1である頂点は $V' = \{v_0, v_1\}$ である。

式(1)を与えられた連結なSグラフ G に適用するアルゴリズムを次に示す。

1. G からある全域木 G^T を作成し、 G の G^T に対する補辺集合 E^C を抽出する。
2. ある補辺 $e_j \in E^C$ について、 e_j と全域木 G^T が作る半有向閉路を $P_j = (V_j^P, E_j^P)$ をつくる。
3. V_j^P から入次数・出次数が共に1である頂点集合 $V'_j = \{v_{0,j}, v_{1,j}, \dots, v_{n-1,j}\} \subseteq V_j^P$ を抽出し、式(2)を制約条件式に追加する。

$$\sum_{i=0}^{n-1} x_{i,j} = 2m_j + (n \bmod 2) \quad (2)$$

$(x_{i,j} \in \{0, 1\}, m_j \text{ は整数})$

4. 手順2から手順3の操作を全ての補辺 e_j に対して適用する。

最適化目標は2部グラフ変換における多重化頂点を少なくすることである。制約条件の $x_i = 1$ となる変数は多重化する頂点を示すので、Sグラフ G の各頂点に対応する変数集合を $\{x_0, x_1, \dots, x_{n-1}\}$ とすると、整数計画問題の目的関数は式(3)となる。

$$\text{Minimize} : \sum_{i=0}^{n-1} x_i \quad (3)$$

表2: 提案手法による追加頂点数と実行時間

回路名	頂点数			処理時間		回路名	頂点数			処理時間	
	適用前	適用後	増加率	制約条件	線形計画法		適用前	適用後	増加率	制約条件	線形計画法
s27	3	6	100%	0.007s	0.001s	s832	5	10	100%	0.091s	0.001s
s298	14	28	100%	0.032s	0.000s	s953	29	35	21%	0.127s	0.002s
s344	15	30	100%	0.039s	0.002s	s1196	18	20	11%	0.182s	0.000s
s349	15	30	100%	0.039s	0.002s	s1238	18	20	11%	0.177s	0.001s
s382	21	36	71%	0.042s	0.001s	s1423	74	145	96%	0.495s	0.028s
s386	6	12	100%	0.038s	0.001s	s1488	6	12	100%	0.254s	0.001s
s400	21	36	71%	0.042s	0.001s	s1494	6	12	100%	0.251s	0.002s
s444	21	36	71%	0.050s	0.002s	s5378	179	277	55%	3.554s	0.234s
s510	6	12	100%	0.052s	0.001s	s9234	228	416	82%	14.773s	0.320s
s526	21	42	100%	0.055s	0.002s	s13207	669	1154	72%	41.347s	2m1s
s641	19	34	79%	0.102s	0.003s	s15850	597	1142	91%	18m16s	52.625s
s713	19	34	79%	0.117s	0.002s	s35932	1728	3184	84%	3m42s	203m37s
s820	5	10	100%	0.089s	0.002s	s38584	1452	2851	96%	4m12s	47m18s

5 実験結果

本節では提案手法をSグラフに対して適用した結果を示す。制約条件の生成アルゴリズムをC++で実装し自動化した。実験を行った計算機はDell PowerEdge R815を使用し、CPUがAMD Opteron 6168(1.9GHz, 12コア)×4、メモリが96GiB、OSがRed Hat 6.4である。対象となるグラフは、非同期式回路とSグラフの性質が似ていると考えられる同期式回路のSグラフを使用した。同期式回路のSグラフはISCAS'89のベンチマーク回路から、規模が異なるグラフを用意した。

それぞれのSグラフに対して提案手法を適用した結果を表2に示す。頂点数はSグラフに含まれる頂点の数を表し、それぞれ適用前の頂点数、適用後の頂点数、適用前と適用後の増加率を表す。そして処理時間は、制約条件の生成と整数計画問題に要した時間である。頂点数の変化については、LSSD適用時には100%となる頂点数を、平均80%程度に抑えることができた。また処理時間に関しては、文献[3]では24時間以内に制約条件ができた回路は9つの回路(s27, s386, s510, s820, s832, s1196, s1238, s1488, s1494)であった。提案手法ではこれらの9つの回路を含む、全ての回路において実用的な時間で適用でき、これら9つの回路に対しても文献[3]の手法より短い時間で適用できた。

6 あとがき

組合セループが多く存在する非同期式回路はテストが困難であるが、L1L2*スキャン設計を適用することによって高い故障検出率を実現できる。L1L2*スキャン設計の適用には、回路のSグラフが2部グラフである必要がある。そのため対象回路のSグラフが2部グラフとなるよう、ダブルラッチに置換すべきシングルラッチを

求める必要がある。文献[3]の手法ではL1L2*スキャン設計の適用に伴う追加ラッチを最小に抑えることができないが、計算時間が大きく大規模回路に対して実用的な時間で適用できなかった。本稿では“木が2部グラフである”という性質に着目し、アルゴリズムの高速化を提案した。

実験結果から、提案手法を同期式回路のSグラフに適用するとラッチの増加率を平均80%程度に抑えることができた。また提案手法は文献[3]の手法では適用できなかった回路に対して、実用的な時間で適用することができ、その優位性を示すことができた。

今後の課題は、全域木の選択から追加ラッチ数や処理時間を抑えることである。全域木の選択法によって線形計画問題の制約条件の複雑性が異なり、計算時間が大きく異なる。したがって計算時間と部分解の関係について考察し、最適な解を求めることで、処理時間を更に短縮できることが予想される。

参考文献

- [1] S. DasGupta, P. Goel, R.G. Walther, and T.W. Williams, "A Variation of LSSD and Its Implications on Design and Test Pattern Generation in VLSI," ITC'82, pp.63-66, 1982.
- [2] F. te Beest, A. Peeters, K. van Berkel, and H. Kerckhoff, "Synchronous Full-Scan for Asynchronous Handshake Circuits," J. Electron. Test., vol.19, no.4, pp.397-406, Aug. 2003.
- [3] H. Iwata, S. Ohtake, M. Inoue, and H. Fujiwara, "Bipartite Full Scan Design: A DFT Method for Asynchronous Circuits," IEEE 19th Asian Test Symposium, pp.206-211, Dec 2010.
- [4] H. Hulgaard, S.M. Burns, and G. Borriello, "Testing Asynchronous Circuits: A Survey," Integration, the VLSI Journal., vol.19, no.3, pp.111-131, 1995.