

論文 / 著書情報
Article / Book Information

論題	一般同期方式における最適2クラスタ分割手法
Title	An Optimum 2-Clustering Method in General-Synchronous Framework
著者	小平行秀, 高橋篤司
Authors	Yukihide Kohira, Atsushi Takahashi
出典	第25回 回路とシステムワークショップ論文集, , , pp. 178-183
Citation	Proc. the 25th Workshop on Circuits and Systems, , , pp. 178-183
発行日 / Pub. date	2012, 7
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 (c) 2012 Institute of Electronics, Information and Communication Engineers

一般同期方式における最適2クラスタ分割手法 An Optimum 2-Clustering Method in General-Synchronous Framework

小平 行秀[†]

高橋 篤司[‡]

[†]会津大学 コンピュータ理工学部

[‡]東京工業大学 大学院理工学研究科 集積システム専攻

Yukihide KOHIRA[†]

Atsushi TAKAHASHI[‡]

[†]School of Computer Science, the University of Aizu

[‡]Dept. of Communications and Integrated Systems, Tokyo Institute of Technology

1 はじめに

従来の集積回路設計では、人間の理解しやすさ、設計の容易さを第一に、回路の一つの実現方法であるクロックを各レジスタに同時に分配する完全同期方式を追求してきた。しかし、その前提の下での性能追求は限界に達しつつあるだけでなく、前提の維持のためのコスト（例えば、クロックバッファ数や消費電力）が非常に大きくなっている。一方、クロックを各レジスタに同時に分配することを前提としない一般同期方式 [1-3] では、物理遅延を完全同期方式よりも効率良く利用できる可能性があるため、クロック周期、クロック面積、消費電力、信頼性など様々な指標で計られる性能を、低コストで実現することが期待されている。

完全同期方式において、全てのレジスタに同時にクロックを供給するクロック木を設計できない場合には回路が正常に動作しないと考えられるのと同様に、一般同期方式の回路の設計においても、設定されたクロックスケジュールを満足するクロック木を設計できない場合には回路が正常に動作しないと考えられる。設定されたクロックスケジュールにより、クロック周期、クロック面積などの回路の性能だけでなく、クロック木の設計の困難さが大きく変化するので、クロックスケジュールリング問題は重要な問題の1つである。これまでに、様々なクロックスケジュールリング法が提案された [1-4]。

近年、クロック木をより容易に設計するために、レジスタをいくつかのクラスタに分割し、クラスタごとにクロックタイミングを設定し、そのクロックタイミングを満たすクロック木を設計するクラスタ分割手法が提案された [5,6]。文献 [7] において、クロックスケジュールを満たすクロック木を市販の EDA ツールで設計する場合、クラスタ数を少なくすると、完全同期方式より最小クロック

周期が小さくなり、消費電力が小さいクロック木が得られる傾向が示された。特に、クラスタ分割手法を適用した2回路とも、クラスタ数を2に限定しても完全同期方式より10%程度高速な一般同期方式の回路を実現しており、クラスタ数を2に限定しても十分にクロック周期が減少する。以後、レジスタ集合とレジスタ間の遅延情報が与えられているときに、レジスタのクラスタへの分割と、回路が正常に動作するクラスタごとのクロックタイミングを求める問題をクラスタ分割問題と呼ぶ。

パスディレイテストによりタイミング違反であると判断された回路を、正常に動作するように製造後にクロックスキューを調整する手法が検討されている [8,9]。遅延値が既知か未知かという違いがあるが、この製造後クロックスキュー調整もクラスタ分割問題とみなすことができる。

クラスタ分割問題は、クラスタ数や、設定できるクロックタイミングが連続値か、整数値か、任意の離散値かという条件により問題の計算複雑度が異なる。

クラスタ数に制限がなく、設定できるクロックタイミングが連続値である場合は、クラスタ分割をしないクロックスケジュールリングと同じ問題であるため、文献 [1-4] で提案されているアルゴリズムにより、多項式時間で最適解が求められる。また、クラスタ数に制限がなく、設定できるクロックタイミングが整数値である場合は、このクラスタ分割問題を整数計画法に定式化すると完全単模の性質を満たすため、線形計画法を解くことで多項式時間で最適解が得られることが示されている [8]。

一方、クラスタ数が制限されている場合や、設定できるクロックタイミングが任意の離散値である場合には、多項式時間で最適解を求める手法は知られておらず、発見的な手法 [5,9] や、整数計画法に定式化し、そのソルバにより解を求める手法 [6,9] が提案されている。発見的な手法は

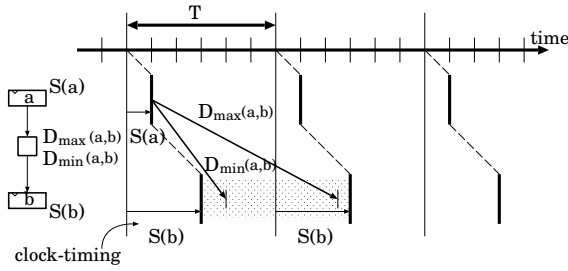


図 1: 一般同期方式の動作制約 .

短時間で解が得られるが、最適性が保証されていない、整数計画法による手法は、解が得られれば最適性が保証できるが、多項式時間で解が得られる保証がない。

本稿では、クラスタ数が 2 以下に制限され、設定できるクロックタイミングが連続値のときに、2-SAT 問題に定式化することでクラスタ分割問題を解く多項式時間アルゴリズムを提案する。2-SAT 問題は多項式時間で解ける [10] ため、提案手法は多項式時間で最適解が得られる。提案手法により、一般同期方式において低コストなクロック木を実現することが期待されるだけでなく、製造後クロックスキュー調整手法において遅延の調整量が 2 つの値に限定されている問題でも、提案手法によりクロックスキュー調整を求めることで歩留まりが向上することが期待される。

2 準備

一般同期方式の回路が正常に動作するための条件は、信号が伝搬する全てのレジスタ対が以下の 2 つのタイミング制約を満たすことであると知られている [1](図 1)。

セットアップ制約: 0 クロック制約

$$S(a) - S(b) \leq T - D_{\max}(a, b)$$

ホールド制約: 二重クロック制約

$$S(b) - S(a) \leq D_{\min}(a, b)$$

ここで、 T はクロック周期、 $D_{\max}(a, b)$ ($D_{\min}(a, b)$) はレジスタ a , b 間の最大遅延 (最小遅延)、 $S(a)$ ($S(b)$) はレジスタ a (レジスタ b) のクロックタイミングを表す。

完全同期方式では、全てのレジスタに同じタイミングで同一周期のクロックが与えられることを前提とするため、クロック周期がレジスタ間の最大遅延未満のとき、回路が正常に動作しないと考える。一方、一般同期方式では、クロック周期がレジスタ間の最大遅延値未満で完全同期方式では正常に動作しないとみなされる回路でも、全てのタイミング制約を満たすのであれば回路は正常に動作すると正確に判断する。なお、クロック木において遅延

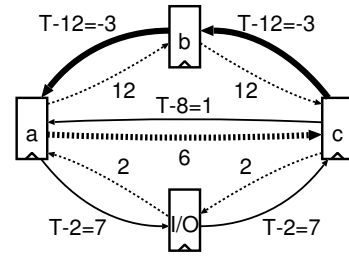


図 2: 制約グラフ $H(G)$ とクロック周期が 9 のときの制約グラフ $H(G, 9)$.

のばらつきが大きい場合やレジスタのホールド時間が大きい場合にはホールド制約を違反することはあるが、本稿では説明のため、完全同期方式ではホールド制約は違反しないと仮定する。以後、レジスタからクロックタイミングへの写像 S をクロックスケジュールと呼ぶ。また、クロック周期 T で、あるクロックスケジュールが全てのタイミング制約を満たす場合、そのクロックスケジュールを T で許容なクロックスケジュールと呼ぶ。

ここで、各レジスタに任意のクロックタイミングを設定できると仮定したときの回路 G の一般同期方式の最小クロック周期を $T_S(G)$ とする。一般同期方式の最小クロック周期 $T_S(G)$ は、回路 G から得られる制約グラフ $H(V_r(G), E_r(G))$ によって求められる [3]。制約グラフ $H(V_r(G), E_r(G))$ の点集合 $V_r(G)$ はレジスタの集合である。制約グラフの有向辺集合 $E_r(G)$ は 2 つのタイミング制約に対応する。レジスタ a から b への重み $D_{\min}(a, b)$ の有向辺はホールド制約に対応する。レジスタ b から a への重み $T - D_{\max}(a, b)$ の有向辺はセットアップ制約に対応する。以後、制約グラフ $H(V_r(G), E_r(G))$ を簡単に $H(G)$ と表記する。また、制約グラフの有向辺の重みはクロック周期 T の関数である。クロック周期 $T = t$ とした制約グラフを $H(G, t)$ と表記する。

一般同期方式の最小クロック周期 $T_S(G)$ は、制約グラフ $H(G, t)$ が負閉路を持たない最小の t であることが知られている [3]。したがって、一般同期方式では、制約グラフ $H(G, T_S(G))$ において 0 閉路であり、制約グラフ $H(G, t)$ ($t < T_S(G)$) において負閉路である閉路が最小クロック周期を決める。

例として図 2 に示す制約グラフ $H(G)$ について考える。レジスタ間の最大遅延が 12 なので、完全同期方式の最小クロック周期は 12 である。クロック周期が 9 のとき、制約グラフ $H(G, 9)$ に負閉路はなく、閉路 (a, c, b, a) のみ 0 閉路である。また、閉路 (a, c, b, a) は $H(G, t)$ ($t < 9$) で負閉路となるので、一般同期方式の最小クロック周期 $T_S(G)$ は 9 である。

3 提案手法

3.1 問題の定義

本稿では，クロックタイミングが同一のレジスタの集合をクラスタと呼ぶ．また，レジスタとクラスタの所属関係をクラスタ分割と呼ぶ．クロックスケジュールを定めるとクラスタ分割は一意に定まることに注意されたい．

本稿では，設定できるクロックタイミングが連続値で，クラスタ数が高々2以下であるとき，クロック周期を最小化するクラスタ分割問題を扱う．設定できるクロックタイミングが連続値で，クラスタ数が2以下に制限されたとき，許容なクロックスケジュールが得られる最小のクロック周期を2クラスタ分割における最小クロック周期 T_2 と呼ぶ．

本稿で扱う，クラスタ数が2以下に制限された2つのクラスタ分割問題を下記に定義する．

2 クラスタ分割問題

入力：制約グラフ $H(G)$

出力：2クラスタ分割における最小クロック周期 T_2 ， T_2 で許容なクロックスケジュール $(S(a) = \{0, s\} \text{ for } \forall a \in V_r(G))$

2 クラスタ分割判定問題

入力：制約グラフ $H(G)$ ，設定クロック周期 T ，タイミング差 $s (\geq 0)$

質問：各レジスタに0か s のクロックタイミングを設定することで， T で許容なクロックスケジュールを得られるか？

以後，設定クロック周期 T ，タイミング差 s で，2クラスタ分割判定問題の答えが“yes”となるとき， T & s で2クラスタ分割可能と呼ぶ

3.2 提案手法

クラスタ数が1より大きくレジスタ数より少ない数に制限されている場合の最小クロック周期は，一般同期の最小クロック周期 T_S 以上，完全同期の最小クロック周期 T_C 以下である [6]．したがって，2クラスタ分割における最小クロック周期の下界は T_S であり，上界は T_C である．提案手法では，設定クロック周期 T を T_C と T_S の間の値に，タイミング差 s を $T_C - T$ に設定し，2クラスタ分割判定問題の答えが“yes”となる最小の T を二分探索法により求めることで，2クラスタ分割問題を解く． s の値が異なる場合，許容なクロックスケジュールが得られる最小のクロック周期も異なるが，下記の定理により，クロッ

ク周期 T ，ある非負定数 s において， T で許容なクロックスケジュールが得られるならば，非負定数 $T_C - T$ でも T で許容なクロックスケジュールが得られることが保証できる．したがって，提案手法により，2クラスタ分割における最小クロック周期 T_2 が得られる．

定理 1 $T \leq T_C$ ，ある非負定数 s において， T & s で2クラスタ分割可能であるとき， $s \geq T_C - T$ であり， T & $(T_C - T)$ で2クラスタ分割可能である．

証明 $s' = T_C - T$ とおく．まず， $s \geq s'$ であることを示す．

$T = T_C$ のときは， s が非負定数という条件より， $s \geq 0 = T_C - T = s'$ は自明．

また， $T < T_C$ のとき， $T < T_C = \max\{D_{\max}(a, b)\}$ より， $T < D_{\max}(a, b)$ となるレジスタ a, b が存在し，

$$S(a) - S(b) \leq T - D_{\max}(a, b) < 0$$

が成り立つ．上式を満たすのは， $(S(a), S(b)) = (0, s)$ のときのみであり，

$$s \geq D_{\max}(a, b) - T$$

が成り立つ．上式は， $T < D_{\max}(a, b)$ となる任意のレジスタ対で成り立つので，

$$s \geq \max\{D_{\max}(a, b)\} - T = T_C - T = s'$$

を得る．よって，常に $s \geq s'$ が成り立つ．

次に， T & s' で2クラスタ分割可能であることを示す．

信号伝搬がある任意のレジスタ対 (a, b) に対して，レジスタ a, b のクロックタイミングの割り当ては， $(S(a), S(b)) = (0, 0), (0, s), (s, 0), (s, s)$ の4パターンある．このうち， $(S(a), S(b)) = (0, 0), (s, s)$ の2パターンについては，セットアップ制約とホールド制約の両方とも，左辺の値が変わらないので，タイミング制約違反の有無は変わらない．つまり， $(S(a), S(b)) = (0, 0), (s, s)$ の2パターンについては $(S(a), S(b)) = (0, 0), (s', s')$ としてもタイミング制約を違反しない．

そこで，レジスタ a, b のクロックタイミングの割り当てで， $(S(a), S(b)) = (0, s), (s, 0)$ の2パターンについて， $(S(a), S(b)) = (0, s'), (s', 0)$ とするとタイミング制約を違反するかを考える．

まず，セットアップ制約について考える． $s \geq s'$ より， $(S(a), S(b)) = (s, 0)$ のときには，

$$s' \leq s \leq T - D_{\max}(a, b)$$

より， $(S(a), S(b)) = (s', 0)$ としてもセットアップ制約が満たされる．また， $(S(a), S(b)) = (0, s)$ のときには，

$$-s' = T - T_C \leq T - D_{\max}(a, b)$$

が成り立つので, $(S(a), S(b)) = (0, s')$ としてもセットアップ制約が満たされる. したがって, $s = s'$ としても, 全てのレジスタ対でセットアップ制約が満たされる.

最後に, ホールド制約について考える. 本稿では完全同期の場合にホールド制約が満たされると仮定しているので, $(S(a), S(b)) = (s, 0)$ のときには,

$$-s' \leq 0 \leq D_{\min}(a, b)$$

であり, $(S(a), S(b)) = (s', 0)$ としてもホールド制約が満たされる. また, $(S(a), S(b)) = (0, s)$ のときには, $s \geq s'$ より,

$$s' \leq s \leq D_{\min}(a, b)$$

であるので, $(S(a), S(b)) = (0, s')$ としても, ホールド制約を違反することはない. したがって, $s = s'$ としても, 全てのレジスタ対でホールド制約が満たされる.

以上より, $T \leq T_C$ のとき, $T \& s$ で 2 クラスタ分割可能であるとき, $T \& s'$ で 2 クラスタ分割可能である. ■

3.3 2 クラスタ分割判定手法

提案手法では, 2 クラスタ分割判定問題を 2-SAT 問題に帰着することで解く. 2-SAT 問題とは, 論理変数集合 $X = \{x_1, x_2, \dots, x_n\}$ に対して, 論理関数 $C(X) = c_1(X) \wedge c_2(X) \wedge \dots \wedge c_m(X)$ が 1 となる論理変数の値 $x_i = \{0, 1\}$ ($1 \leq \forall i \leq n$) が存在するか否かを判定する問題である. ただし, $c_j(X)$ ($1 \leq \forall j \leq m$) は, 2 つ以下のリテラルからなる和項 ($c_j(X) = x_a \vee \bar{x}_b$ など) である. 1 つの和項に含まれるリテラル数が 3 である, 3-SAT 問題は NP 困難である [11] が, 2-SAT 問題は強連結性を判定するグラフ問題に帰着することで多項式時間で解け, 関数が 1 となる論理変数値も求められる [10].

ここから, 2 クラスタ分割判定問題から 2-SAT 問題への変換方法を述べる. 2 クラスタ分割判定問題の各レジスタに対して, 2-SAT 問題では 1 つの論理変数を定義する. 2-SAT 問題における論理変数 x_a は 2 クラスタ分割判定問題のレジスタ a のクロックタイミング $S(a)$ に対応させ, $x_a = 1$ ($x_a = 0$) のとき, かつそのときに限り $S(a) = s$ ($S(a) = 0$) とする.

次に, 2 クラスタ分割判定問題の信号伝搬がある各レジスタ対に対して, 2-SAT 問題では 4 つの和項を定義する. 論理関数は, $C(X) = \bigwedge (c_0^{(a,b)} \wedge c_1^{(a,b)} \wedge c_2^{(a,b)} \wedge c_3^{(a,b)})$ とする. ただし, $c_k^{(a,b)}$ は信号伝搬がある任意のレジスタ対 (a, b) に対して, クロックタイミング $S(a), S(b)$ がホールド制約とセットアップ制約の 2 つのタイミング制約を満足するか違反するかにより, 表 1 に示すように定義する.

例として, $(x_a, x_b) = (0, 0)$ のときを説明する. このとき, $(S(a), S(b)) = (0, 0)$ でタイミング制約を違反する場合, $(S(a), S(b)) = (0, 0)$ を $c_0^{(a,b)}$ に代入すると, $c_0^{(a,b)} =$

表 1: 和項の定義.

x_a	x_b	$S(a)$	$S(b)$	(a, b) のタイミング制約	
				満足	違反
0	0	0	0	$c_0^{(a,b)} = 1$	$c_0^{(a,b)} = x_a \vee x_b$
0	1	0	s	$c_1^{(a,b)} = 1$	$c_1^{(a,b)} = x_a \vee \bar{x}_b$
1	0	s	0	$c_2^{(a,b)} = 1$	$c_2^{(a,b)} = \bar{x}_a \vee x_b$
1	1	s	s	$c_3^{(a,b)} = 1$	$c_3^{(a,b)} = \bar{x}_a \vee \bar{x}_b$

表 2: タイミング制約違反と論理包含関係.

x_a	x_b	制約違反	論理包含
0	0	$c_0^{(a,b)} = x_a \vee x_b$	$(\bar{x}_a \rightarrow x_b) \wedge (\bar{x}_b \rightarrow x_a)$
0	1	$c_1^{(a,b)} = x_a \vee \bar{x}_b$	$(\bar{x}_a \rightarrow \bar{x}_b) \wedge (x_b \rightarrow x_a)$
1	0	$c_2^{(a,b)} = \bar{x}_a \vee x_b$	$(x_a \rightarrow x_b) \wedge (\bar{x}_b \rightarrow \bar{x}_a)$
1	1	$c_3^{(a,b)} = \bar{x}_a \vee \bar{x}_b$	$(x_a \rightarrow \bar{x}_b) \wedge (x_b \rightarrow \bar{x}_a)$

$x_a \vee x_b = 0$ であるので, $(x_a, x_b) = (0, 0)$ のときは $C(X) = 0$ である. したがって, 2-SAT の解は $(x_a, x_b) = (0, 0)$ とはならない. つまり, $C(X) = 1$ を満たす解が得られるときは, タイミング制約を違反する $(S(a), S(b)) = (0, 0)$ とはならない. 一方, $(S(a), S(b)) = (0, 0)$ でタイミング制約を満たす場合, $c_0^{(a,b)} = 1$ であるので, この項は論理関数 $C(X)$ において消去することができる. つまり, $C(X) = 1$ を満たす解が得られるときは, $(S(a), S(b)) = (0, 0)$ となることが許容される.

3.4 2-SAT 問題の解法 [10]

2-SAT 問題は, グラフの強連結性を用いることで, 解くことができる [10]. 表 2 に示すように, 和項から対応する論理包含が定義できる. 各リテラルに点を, 各論理包含関係を有向辺に対応させたグラフを定義する. このグラフ内に, 全ての論理変数 x_a の肯定リテラル x_a に対応する点と否定リテラル \bar{x}_a に対応する点が同じ強連結成分に含まれない場合, つまり, 1 つの閉フォークに含まれないとき, かつそのときに限り, 2-SAT 問題の答えは “yes” である.

3.5 例題

図 3 に示す 2 つの例題について考える. 図 3(a) 上に示す例題では, レジスタ (a, b) 間の最大遅延が 10, 最小遅延が 3 で, 設定クロック周期 T が 9, タイミング差 s が 2 である. この例題では, $(S(a), S(b)) = (0, 0), (s, 0), (s, s)$ のときにセットアップ制約違反となるため, $C_1(X) = (x_a \vee x_b) \wedge (\bar{x}_a \vee x_b) \wedge (\bar{x}_a \vee \bar{x}_b) = (\bar{x}_a \rightarrow x_b) \wedge (\bar{x}_b \rightarrow x_a) \wedge (x_a \rightarrow x_b) \wedge (\bar{x}_b \rightarrow \bar{x}_a) \wedge (x_a \rightarrow \bar{x}_b) \wedge (x_b \rightarrow \bar{x}_a)$ である. この論理包含を表すグラフは図 3(a) 下である. このグラフは x_a, \bar{x}_a も x_b, \bar{x}_b も同じ強連結成分に含まれないため, $9 \& 2$ で 2 クラスタ分割可能であり, $C_1(X) = 1$ となる $(x_a, x_b) = (0, 1)$

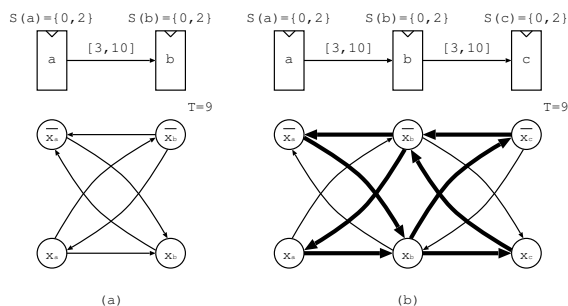


図 3: 例題. (a) は 9×2 で 2 クラスタ分割可能で, (b) は 9×2 で 2 クラスタ分割不可能.

が得られる. つまり, $(S(a), S(b)) = (0, 2)$ というクロックスケジュールが得られる.

一方, 図 3(a) 上に示す例題では, 詳細は省くが, 図 3(b) 下に示す論理包含を表すグラフでは, 全てのリテラルが 1 つの強連結成分に含まれるため, 9×2 で 2 クラスタ分割不可能である.

4 実験結果

提案手法の有効性を示すため, 提案手法と文献 [6] で提案された混合整数計画法による既存手法を実装し, 計算時間を比較した. 提案手法は, gcc4.3.2 の C++ で実装した. 既存手法の混合整数計画法を解くために, ILOG 社の CPLEX12.1.0[12] を用いた. 提案手法, 既存手法ともに, 2.93GHz Intel Core i7 CPU, 6GB RAM の PC で実行した.

ISCAS89 ベンチマークの 48 回路に対して, Xilinx の ISE Design Suite 12.4 を利用し, Spartan3AN XC3S700AN をターゲットデバイスとして配置配線を行い, 遅延解析を行った結果を実験対象とした. なお, 48 回路のうち 8 回路で ISE の遅延解析がメモリオーバーとなり, 遅延データが得られなかったため, 残りの 40 回路に対して, 提案手法と既存手法を適用した. 実験結果を表 3 に示す. なお, 提案手法の二分探索において, クロック周期の精度は整数とし, 小数点以下を切り上げた.

一般同期方式の最小クロック周期, つまり, クラスタ数に制限がない場合には, 最小クロック周期が 23%削減されるのに対して, クラスタ数が 2 以下に制限された場合は, 最小クロック周期が 11%削減された. また, 全ての回路に対して, クラスタ数が 2 以下に制限されても, 最小クロック周期が削減された.

さらに, 混合整数計画法を用いた既存手法ではメモリオーバーとなり解が得られない回路に対しても, 提案手法では高速に解が得られた.

5 まとめと今後の課題

本稿では, クラスタ数が 2 以下に制限され, 設定できるクロックタイミングが連続値のときに, 2-SAT 問題に定式化することでクラスタ分割問題を解く多項式時間アルゴリズムを提案した. 2-SAT 問題を多項式時間で解けるため, 提案手法は多項式時間アルゴリズムであり, かつ最適性が保証される. 計算機実験により, 高速に解が得られることを確認した.

今後の課題としては, 大規模な回路に対して提案手法を適用すること, 製造後クロックスキュー調整に提案手法を応用すること, クラスタ数が 3 より大きい数に制限されたときに対するアルゴリズムを提案することが挙げられる.

参考文献

- [1] J.P. Fishburn, "Clock skew optimization," IEEE Trans. on Computers, vol.39, no.7, pp.945–951, 1990.
- [2] R.B. Deoker and S.S. Sapatneker, "A Graph-Theoretic Approach to Clock Skew Optimization," ISCAS, pp.407–410, 1994.
- [3] A. Takahashi and Y. Kajitani, "Performance and Reliability Driven Clock Scheduling of Sequential Logic Circuits," ASP-DAC'97, pp.37–43, 1997.
- [4] A. Takahashi, "Practical Fast Clock-Schedule Design Algorithms," IEICE Trans. Fundamentals, vol.E89-A, no.4, pp.1005–1011, 2006.
- [5] 橋本浩良, 小平行秀, 高橋篤司, "EDA ツールを用いた低コスト一般同期クロックツリー合成手法," 信学技報, VLD2008–134, vol.108, no.478, pp.47–52, 2009.
- [6] 小平行秀, 高橋篤司, "一般同期方式における消費電力を抑えたクロック木構成のためのクラスタ分割法," 信学技報, VLD2008–134, vol.108, no.478, pp.47–52, 2009.
- [7] 小平行秀, 高橋篤司, "一般同期方式におけるクラスタ分割に基づくクロック木の性能評価," 電子情報通信学会 ソサイエティ大会 講演論文集 (A-3-1), vol.A, p.63, 2010.
- [8] Y. Hashizume, Y. Takashima, and Y. Nakamura, "Post-Slicicon Clock-Timing Tuning Based on Statistical Estimation," IEICE Trans. Fundamentals, vol.E91-A, no.9, pp.2322–2327, 2008.
- [9] 李 健, 金子峰雄, "タイミングテストを利用する LSI 製造後スキュー調整アルゴリズム," 電子情報通信学会 ソサイエティ大会 講演論文集 (A-3-17), vol.A, p.91, 2011.
- [10] B. Aspvall, M.F. Plass, and R.E. Tarjan, "A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas," Information Processing Letters, vol.8, no.3, pp.121–123, 1979.
- [11] M.R. Garey and D.S. Johnson, Computers and intractability a guide to the theory of np-completeness, W.H.FREEMAN AND COMPANY, 1979.
- [12] ILOG, CPLEX12.1.0, <http://www.ilog.co.jp/>.

表 3: 実験結果 .

circuit	#FF	#delay	clock period [ps]					proposal time[s]	MILP [6] time[s]
			T_C	T_S	T_S/T_C	T_2	T_2/T_C		
s27	3	7	3451	3091	0.90	3091	0.90	<0.01	0.01
s208.1	8	36	4687	2383	0.51	3407	0.73	<0.01	0.04
s298	14	70	6185	4147	0.67	4825	0.78	<0.01	0.06
s344	15	89	6299	5187	0.82	5224	0.83	<0.01	0.25
s349	15	89	5976	4607	0.77	5140	0.86	<0.01	0.19
s382	21	146	5312	4759	0.90	5295	1.00	<0.01	0.67
s386	6	36	5490	5369	0.98	5451	0.99	<0.01	0.02
s400	21	146	5679	5122	0.90	5625	0.99	<0.01	0.37
s420.1	16	136	6640	3324	0.50	5382	0.81	<0.01	0.09
s444	21	146	5583	4564	0.82	5312	0.95	<0.01	0.59
s499	22	484	8824	8650	0.98	8733	0.99	<0.01	0.39
s510	6	36	7189	7158	1.00	7158	1.00	<0.01	0.01
s526	21	144	5868	4998	0.85	5775	0.98	<0.01	0.48
s526n	21	144	5824	5230	0.90	5789	0.99	<0.01	0.34
s635	32	528	11487	3672	0.32	9013	0.78	<0.01	0.74
s641	19	115	10526	9661	0.92	9793	0.93	<0.01	0.29
s713	19	115	10633	10082	0.95	10196	0.96	<0.01	0.22
s820	5	25	7818	7701	0.99	7701	0.99	<0.01	0.01
s832	5	25	7508	7473	1.00	7474	1.00	<0.01	0.01
s838.1	32	528	10039	3760	0.37	7681	0.77	<0.01	0.99
s938	32	528	9264	4170	0.45	7355	0.79	<0.01	0.68
s953	26	156	7540	7429	0.99	7536	1.00	<0.01	0.44
s967	26	156	7665	7646	1.00	7646	1.00	<0.01	0.07
s991	19	71	17399	4601	0.26	10089	0.58	<0.01	0.14
s1196	12	20	9157	4128	0.45	6150	0.67	<0.01	0.05
s1238	12	20	9036	2839	0.31	6392	0.71	<0.01	0.05
s1269	37	288	14908	10270	0.69	12209	0.82	<0.01	0.43
s1423	74	1764	20431	16739	0.82	18266	0.89	<0.01	4.07
s1488	6	36	8762	8719	1.00	8743	1.00	<0.01	0.02
s1494	6	36	8847	8642	0.98	8794	0.99	<0.01	0.03
s1512	57	513	8953	8579	0.96	8619	0.96	<0.01	0.85
prolog	132	549	11522	8343	0.72	10726	0.93	<0.01	O.M.
s3271	116	896	11952	5934	0.50	9740	0.81	0.01	28.17
s3330	132	549	10122	7341	0.73	8408	0.83	<0.01	7.63
s3384	183	1827	18863	15923	0.84	16589	0.88	0.01	105.73
s5378	178	1144	8432	6452	0.77	8339	0.99	<0.01	146.27
s9234	160	2000	11983	9810	0.82	10552	0.88	<0.01	28.01
s9234.1	143	1884	11593	9478	0.82	10912	0.94	<0.01	85.79
s13207.1	621	3330	14755	11335	0.77	12592	0.85	0.03	O.M.
s38584	1434	15387	16718	15343	0.92	15582	0.93	0.08	O.M.
ave.					0.77		0.89		

#FF レジスタ数
#delay レジスタ間遅延パス数
 T_C 完全同期方式における最小クロック周期
 T_S 一般同期方式における最小クロック周期
 T_2 2 クラスタ分割における最小クロック周期
<0.01 計算時間が 0.01[s] 未満
O.M. メモリオーバー