

非同期式回路の簡単化のための等価性判定アルゴリズムの改良：

条件ループの対応と適応的判定順序変更による高速化

Improvement of Equivalence Determination Algorithm for Asynchronous Circuits Simplification : Extension to Conditional Loops and Acceleration by Adaptive Determination Order Change

福永 哲太[†] 籠谷 裕人[†]
Tetsuta Fukunaga Hiroto Kagotani

1. はじめに

非同期式回路では、デジタル回路設計において主流である同期式回路とは異なり、クロック信号を用いずにハンドシェイクプロトコルを用いてタイミングの制御を行っている。ハンドシェイクプロトコルを用いることで、同期式回路と比べて非同期式回路は低消費電力、低ノイズ、高速性が実現できると期待されている。しかし、依存性グラフを用いてパイプライン化した非同期式回路を簡単化するための等価性判定アルゴリズムは、条件ループを持つ場合には対応できないという問題と等価性判定に長時間を要するという問題があった。本稿では、前者を解決するために、条件ループに対応可能なパイプラインイタレーションの境界を用いる手法を提案する。また、後者を解決するために、等価性判定が失敗する分岐系列の偏りに着目して、判定順序を変更することにより高速に判定する手法を提案する。

2. 依存性グラフの構造

依存性グラフ (DG) は、基本操作 (長円)、セレクト (三角形)、マージ (逆三角形)、フォーク (小正方形)、ジョイン (小円) の頂点と頂点間のトークンの移動を表す有向辺、初期トークン (黒丸) により構成されるグラフである。初期トークンから出発したトークンは有向辺上を移動し、任意の頂点上に到達した

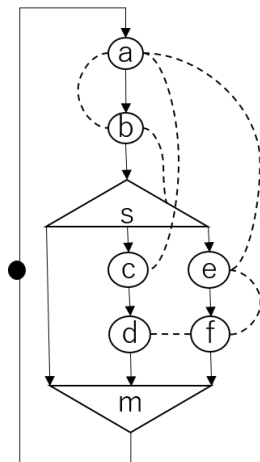


図 1 SDG

時その頂点が発火 (処理を実行) する。セレクトは発火時の変数の値などの条件からトークンをどの分岐に移動させるかを決定する。マージはセレクトにおいて分岐した合流地点である。フォークとジョインはトークンの複製、結合を行う。本論文では、基本操作、セレクト、マージのみを頂点として扱い、フォークとジョインはこれらの頂点に付随するものとして扱う。フォークとジョインを含まない DG のことを直列依存性グラフ (SDG)

と呼び、SDG を並列化した依存性グラフをパイプライン化依存性グラフ (PDG) と呼ぶ。

図 1 では SDG の頂点とその頂点ごとの依存関係 (破線) を示している。各頂点にはラベルが付加されており、セレクトでは便宜上、左から順に分岐番号を 1, 2, 3... を割り当てる。セレクトの発火時にどの分岐番号を選択したかを明示的に示す場合は、選択した分岐番号を () 内部に入れて $s(1)$ のように表す。図 2 は図 1 をパイプライン化したグラフであり、図 1 上での依存関係をすべて満足するようにセレクトとマージを複製し並列性を最大限まで高めたものとなっている。パイプライン化された際に複製されたセレクトとマージは、複製元の名前に複製番号を付加して $s1, s2, m1, m2$ のように示す。

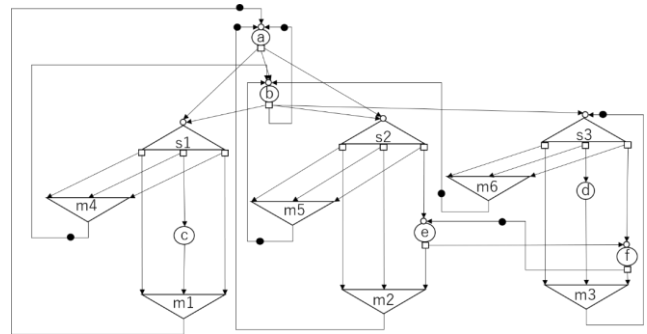


図 2 PDG

3. 強等価判定法

二つの PDG が強等価であるか判定するために本論文では PDG グラフの構造に基づいた基本操作直結因果関係グラフを用いた方法を採用している[1]。最初に、基本操作直結因果関係グラフを用いた等価判定アルゴリズムを簡単に説明する。SDG G において、初期トークンが存在する辺の終点、または条件ループ末尾のマージ要素から始まる極大な単純路のことを、 G のイタレーションと呼び、SIR と略記する。SIR を作成するアルゴリズムは[2]で示されている。PDG は SDG を並列化したものであるので、対応する SIR の頂点の順番は変わるがそれぞれの前後の関係は変化せず、またその処理内容は変わらない。したがって PDG 上のイタレーションは SIR と 1 対 1 に対応する。

SIR の系列を有限長でかつ一意に特定するために、分岐系列 $\hat{B} = \{B^1, B^2 \dots\}$ を定義する。ただし、 B^i は \hat{B} の i 番目の分岐パラメータである。また、 B^i に対応するイタレーション i の頂点 x が発火するとき x^i とあらわす。

分岐系列と PDG に対して、基本操作発火 x^i からセレクトとマージのみを通過して基本操作頂点発火 y^j に到達するとき、 x^i と y^j は基本操作到達関係があるという。さらに、 (x^i, y^j) を有向辺としてみなした有向グラフを基本操作直接到達関係グラフ V と表す。 V の推移的リダクションを基本操作直結因果関係グラフ $\hat{O}(G, \hat{B})$ (または、 \hat{O}) で表す。SDG のセレクト数を M として、 $\hat{O}(G, \hat{B})$ のうち分岐系列において i 番目に対応する部分グラフを $\hat{O}_{(1)}(G_1, \hat{B})$ と表す時、[1]より長さ $4M + 4$ の任意の分岐系列 \hat{B} において $\hat{O}_{(1)}(G_1, \hat{B}) = \hat{O}_{(1)}(G_2, \hat{B}), \hat{O}_{(2)}(G_1, \hat{B}) = \hat{O}_{(2)}(G_2, \hat{B})$ を満たすだけで強等価であると判定できる。特に、条件ループを持たない場合は分岐系列長が $2M + 2$ で十分であり、 $\hat{O}_{(1)}(G_1, \hat{B}) = \hat{O}_{(1)}(G_2, \hat{B})$ のみで判定を行うことができる。

4. 条件ループに対応するための改良

探索の過程で、PDG 上の有向辺 (x, y) を経由する際に以下のいずれかの条件を満たしていれば、これをイタレーションの境界とする。ただし、複製されたセレクト、マージは複製元の SIR を参照する。

- 現在のイタレーションに対応する SIR 上では、頂点 y 発火以降に頂点 x が発火する
- 現在のイタレーションに対応する SIR 上では頂点 y が発火しない

図 3 は条件ループを持つ SDG である。図 4, 表 1 はそれぞれ図 3 の PDG, SIR である。

分岐系列 $\hat{B} = (\{s(2), t(2)\}, \{t(2)\}, \{s(1), t(1)\}, \{s(2), t(1)\}, \dots)$ を例として説明する。

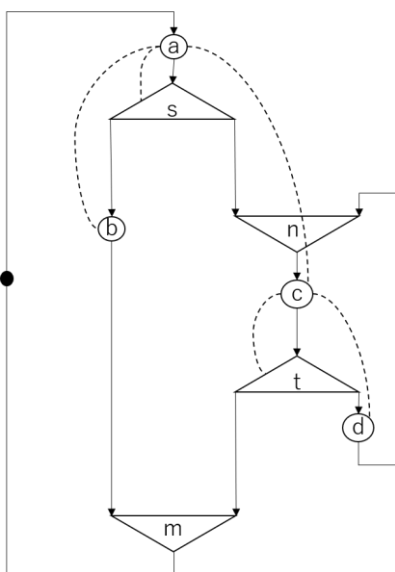


図 3 条件ループを持つ SDG

基本操作発火 d^1 から有向辺 $(d, n2)$, $(n2, c)$ を経由して到達する頂点は c である。開始点が a , 分岐系列が $\hat{B}^1 = (\{s(2), t(2)\})$ の条件を満たす SIR は $(a, s(2), n, c, t(1), d)$ であり、 n の発火後に d が発火する事が分かるので、有向辺 $(d, n2)$ は一つ目の条件満たす。よって、イタレーション番号を 1 から 2 変更し、 V 上に作成される有向辺は (d^1, c^2) になる。

この例は、イタレーションの境界にトークンを用いた場合、有向辺上にトークンがないためイタレーション番号の変更が起きなかったが、イタレーションの境界に SIR を用いた場合、イタレーション番号の変更を起こすことが可能になり、基本操作直接到達関係グラフ V を作成が可能になった。

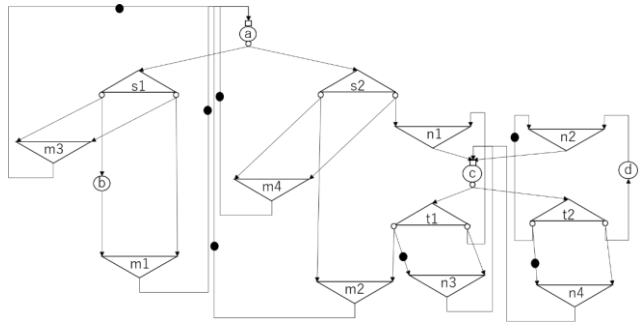


図 4 条件ループを持つ PDG

表 1 図 3 の SIR 一覧

SIR の開始点	次の SIR の開始点	分岐系列	SIR
a	a	s(1)	a, s(1), b, m
a	a	s(2), t(1)	a, s(2), n, c, t(1), m
a	n	s(2), t(2)	a, s(2), n, c, t(1), d
n	m	s(1), t(1)	n, c, t(1), m, a, s(1), b
n	n	s(2), t(1)	n, c, t(1), m, a, s(2)
n	n	t(2)	n, c, t(2), d
m	m	s(1)	m, a, s(1), b
m	m	s(2), t(1)	m, a, s(2), n, c, t(1)
m	n	s(2), t(2)	m, a, s(2), n, c, t(2), d

5. 分岐系列の適応的判定順序変更

従来手法において、強等価ではないと判定される場合、同一ではない基本操作直結因果関係グラフが作成されるまで、分岐系列を変更しながら基本操作直結因果関係グラフを作成する必要がある。しかし、強等価ではないことを判定するには、同一ではない基本操作直結因果関係グラフを作成する分岐系列のみが必要であり、必要ない分岐系列から同一基本操作直結因果関係グラフを作成するための時間が気になる問題が発生している。この問題を解決するために、多くの DG で単一化した頂点に関係なく、同一の分岐系列で失敗していることから、すべての分岐系列で \hat{O} の一致を判定するさいに、別の単一化パターンが失敗した分岐系列を優先的に判定を行い、早期に失敗を判定する可能性を高くすることで、判定に必要な基本操作直結因果関係グラフの作成個数を削減し、アルゴリズムの高速化を行った。

図 6 は図 2 の $s2$ と $s3$ を単一化した PDG G' である。図 2 と図 6 の強等価判定において、分岐系列 $\hat{B} = (\{s(2)\}, \{s(1)\}, \{s(1)\}, \{s(1)\})$ を用いて作成された $\hat{O}_{(1)}$ がそれぞれ図 5 と図 7 となり、 (d^1, b^3) の関係が (d^1, a^3) の関係に変化したため、強等価ではないと分かる。また、他の組み合わせで単一化した PDG の強等価判定においても、 $\hat{B} = (\{s(2)\}, \{s(1)\},$

$\{s(1), \{s(1)\}$ で同様の変化が起こっているため、分岐系列 $\bar{B} = (\{s(2), \{s(1), \{s(1), \{s(1)\}$ は単一化した頂点に関係なく、 (d^1, b^3) の関係を (d^1, a^3) 変化させる可能性があることが分かる。

表 2 は従来手法における分岐系列の使用順番の一部である。表 3 は提案手法における変更後の分岐系列の使用順番の一部である。表 3 のように、分岐系列 $\bar{B} = (\{s(2), \{s(1), \{s(1), \{s(1)\}$ が 1 番目に使用されるような使用順番に変更することで、以降の強等価判定の最初に (d^1, b^3) の関係が (d^1, a^3) に変化しているか 1 番目に調べることができる。変化していた場合、表 2 の順番 1 番から 3 番の分岐系列を判定に使用せずに、1 個の分岐系列のみで強等価ではないと判定できたことになる。このように、判定を失敗した分岐系列を発見する度に 1 番目に使用されるように変更することで、以降に同一の分岐系列で失敗する強等価判定は、最大で使用順番を変更した回数のみで失敗の判定を行うことが可能になる。

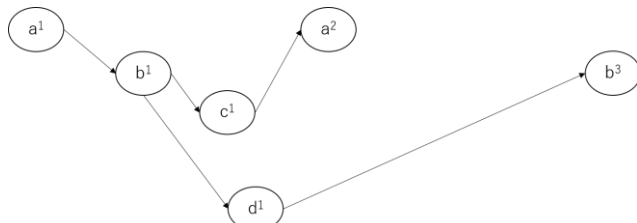


図 5 図 2 の基本操作因果関係グラフ $O_{(1)}$

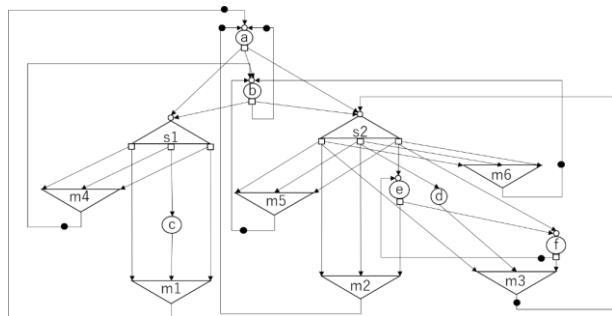


図 6 PDG G'

表 2 分岐系列の使用順

使用順番	従来手法のイタレーション
1	$(\{s(1), \{s(1), \{s(1), \{s(1)\}$
2	$(\{s(1), \{s(1), \{s(1), \{s(2)\}$
3	$(\{s(1), \{s(1), \{s(1), \{s(3)\}$
4	$(\{s(2), \{s(1), \{s(1), \{s(1)\}$
5	$(\{s(2), \{s(1), \{s(2), \{s(1)\}$
⋮	⋮

表 3 変更後の分岐系列の使用順

使用順番	従来手法のイタレーション
1	$(\{s(2), \{s(1), \{s(1), \{s(1)\}$
2	$(\{s(1), \{s(1), \{s(1), \{s(1)\}$
3	$(\{s(1), \{s(1), \{s(1), \{s(2)\}$
4	$(\{s(1), \{s(1), \{s(1), \{s(3)\}$
5	$(\{s(2), \{s(1), \{s(2), \{s(1)\}$
⋮	⋮

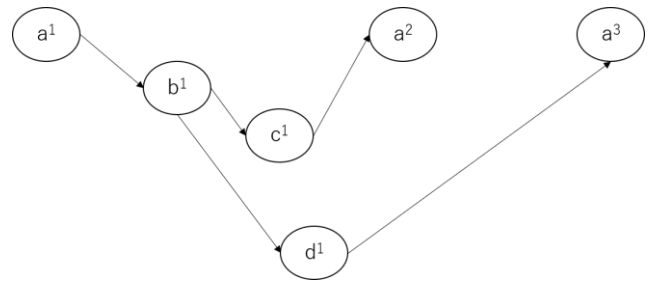


図 7 図 6 の基本操作因果関係グラフ $O_{(1)}$

6. 高速化の評価

単一化を行いうる全ての頂点对について、単一化前後の強等価判定アルゴリズムを実行し、提案手法と従来手法の作成した基本操作直結因果関係グラフの総数と実行時間を計測をした。ベンチマークとして用いた SDG を図 8~図 13 に示す。CPU は Intel Core i7-8700 を用いた。

6.1 測定結果

提案手法と従来手法の分岐系列数などの情報を表 4、作成した基本操作直結因果関係グラフの総数と削減個数を表 5、実行時間を表 6 に示す。ただし、表 5 の削減率は次の式で求めている。削減率 = $(\text{従来手法の } O \text{ の作成個数} - \text{提案手法の } O \text{ の作成個数}) / (\text{従来手法の } O \text{ の作成個数}) \times 100$

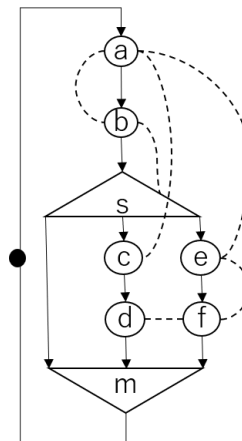


図 8 SDG(1)

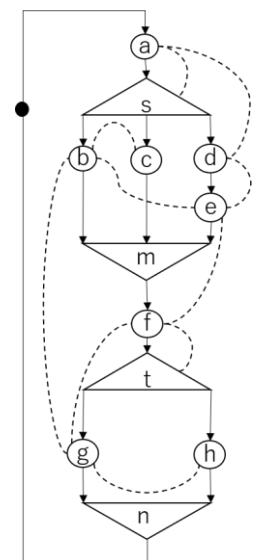


図 9 SDG(2)

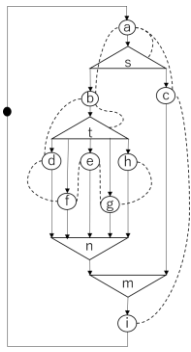


図 10 SDG(3)

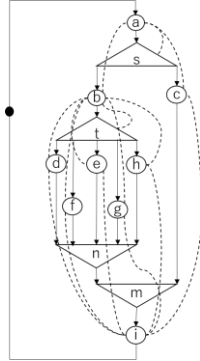


図 11 SDG(4)

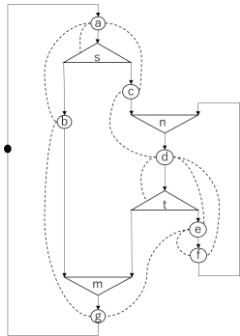


図 12 SDG(5)

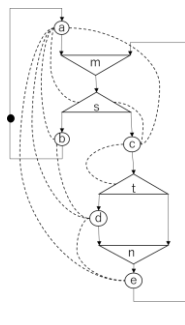


図 13 SDG(6)

表 4 分岐系列の削減回数

SDGの種類	強等価判定回数	分岐系列数を削減できた判定回数
SDG(1)	13	8
SDG(2)	52	42
SDG(3)	173	51
SDG(4)	47	36
SDG(5)	26	13
SDG(6)	46	9

表 5 提案手法の削減効果

SDGの種類	従来手法(個)	提案手法(個)	削減個数(個)	削減率(%)
SDG(1)	150	126	24	16.0
SDG(2)	34673	21707	12966	37.4
SDG(3)	4276956	4150138	126818	2.97
SDG(4)	1526804	971261	555543	36.39
SDG(5)	4582391	4574716	7675	0.17
SDG(6)	2651378	2651342	36	0.0014

表 6 強等価判定の実行時間

SDGの種類	従来手法(s)	提案手法(s)	削減率(%)
SDG(1)	0.009	0.008	11.11
SDG(2)	3.889	2.485	36.10
SDG(3)	457.279	440.274	3.72
SDG(4)	155.527	102.999	33.77
SDG(5)	1222.771	1199.163	1.93
SDG(6)	654.505	652.775	0.26

7. 評価

表 5 と表 6 から、従来手法と比べて、提案手法は作成する基本操作直接因果関係グラフの数と実行時間を削減をすることができている。SDG(3), SDG(5), SDG(6)は削減率は低い。なぜなら、提案手法は強等価判定に失敗する場合に作成する基本操作直接因果関係グラフの個数の削減を行うものであり、強等価判定に成功する場合には影響がないため、強等価判定の回数に対して分岐系列数を削減できた判定回数が半分以下であるこれらの SDG には影響が少なかったからである。強等価判定の回数に対して分岐系列数を削減できた判定回数が半分以下になる理由として、SDG(1)の依存関係(d,f)のような対になるセレクト、マージの内部で完結する依存関係を持たないからである。このような依存関係は、外部の頂点と関係を持たないため、単一化によって新たな基本操作直接到達関係を持つことで、並列性を失い、判定を失敗させる可能性を高くする性質がある。さらに、SDG(5)と SDG(6)は、強等価の場合に必要な分岐系列数に対して削減できる分岐系列数の平均の割合が他の SDG と比べて小さいため、より削減率への影響が少なくなっている。

SDG(1), SDG(2), SDG(4)は削減率は高くなっている。この理由として、強等価判定の回数に対して削減できた判定回数が半分以上であるため削減個数が増加するからである。

以上より、提案した手法は、基本操作直接因果関係グラフの数を削減することで実行時間を削減できることを確認できた。

8. おわりに

研究ではイタレーション境界としてトークンの代わりに SIR を用いることにより条件ループを持つ SDG に対応した強等価判定アルゴリズムの実装を行った。また、別の単一化パターンで失敗した分岐系列を優先的に判定に用いることによる、強等価判定アルゴリズムの高速化法を提案した。この分岐系列の適応的判定順序変更法の実装を行い、従来手法と比較して0.26%~36.10%高速化されたことを確認した。

参考文献

- [1] 籠谷裕人, 杉山裕二, 岡本卓爾, "パイプライン化依存性グラフを単純化するための効率的な等価性判定アルゴリズム", 電子情報通信学会論文誌 D, vol. J100-D, no. 6, pp. 616-626, June 2017.
- [2] 渡辺康輔, "パイプライン化依存性グラフ単純化のためのイタレーション列挙アルゴリズムの実装" 岡山大学特別研究報告書, Feb. 2017.
谷田翔吾, "依存性グラフ単純化のための頂点再統合アルゴリズムの効率化と実装" 岡山大学特別研究報告書, Feb. 2018.