

多段論理最適化方式の検討

5S-3

影山直洋 清水剛雄 宮本俊介
(株)日立製作所 中央研究所

1. 緒言

汎用大型計算機やVLSI論理の大規模化及び複雑化に伴い論理設計の自動化が重要な課題となっている。論理設計の自動化ではゲート数削減手法として二段論理最適化、多段論理最適化など様々な手法¹⁾⁻³⁾が提案されているが論理最適化ではゲート数と同時に遅延時間短縮を考慮することが必須である。本稿では自動論理設計システム⁴⁾におけるブール式レベル論理を対象とした多段論理最適化方式について述べる。

2. ゲート数削減方式

2.1 概要

ブール式を対象としたゲート数削減方式には、積和形ブール式の積項数の削減を行う簡約機能と、単一ブール式内の共通項のくくりだしや、積和形ブール式間の共通項の検出による多段化機能とがある。以下では積和形ブール式の多段化機能について述べる。

2.2 基本方式

多段化機能は複数ブール式間の共通項の統合機能と単一ブール式の除算機能の2種類からなる。以下では、 \cdot は論理積、 $+$ は論理和を示す。

統合機能により得られるブール式 F, G 間の共通項 S を $S = \cap (F, G)$ のように表す。また、ブール式 F の積項 P による除算を次のように表す。

$$F = P \cdot Q + R$$

但し P : 除算因子(積項)

Q : 商(積和形ブール式)

R : 剰余(積和形ブール式)

以下に多段化の基本方式を述べる。

ブール式の集合を $F = \{F_i \mid i=1 \sim n\}$ とする。全てのブール式 F_i に対し、除算因子 P_i を求め、除算機能により商 Q_i と剰余 R_i を求める。全てのブール式の商からなる商集号及び剰余からなる剰余集合をそれぞれ Q, R とする。

$$F_i = P_i \cdot Q_i + R_i$$

$$Q = \{Q_i \mid i=1 \sim n\}, R = \{R_i \mid i=1 \sim n\}$$

次に統合機能により複数の商の間の共通項を求める。

$$S = \{S_{ij} \mid S_{ij} = \cap(Q_i, Q_j), Q_i \in Q, Q_j \in Q\}$$

この時、商 Q_i, Q_j は次のようになる。

$$Q_i = S_{ij} + W_i, Q_j = S_{ij} + W_j$$

ここで、 W_i, W_j は商 Q_i, Q_j での非共通部分であり、これを共通項剰余と称する。従ってもとのブール式 F_i, F_j は次式で表現される。

$$F_i = P_i \cdot (S_{ij} + W_i) + R_i$$

$$F_j = P_j \cdot (S_{ij} + W_j) + R_j$$

このようにブール式を多段化することにより除算因子 P_i, P_j 及び、共通項 S_{ij} に対応する論理ゲートが削減されることになる。

ブール式 W_i, R_i に対しては、適切な除算因子を選択することにより、さらに多段化が可能な場合がある。このため、改めて論理関数の集合 F を

$$F = Q' \cup R \cup W$$

とにおいて、上記の除算機能及び統合機能を繰り返す。ここで Q' は共通項が検出されなかった商の集合であり ($Q' \subset Q$)、 R は剰余集合、また、 W は商 Q_i から共通項 S_{ij} を除いた共通項剰余 W_i の集合を表す。

2.3 ブール式の多段化における留意点

ブール式は除算因子 P を変えることにより多様な多段化が可能であり、同時に統合機能によるゲート削減効果も変化するため、除算因子の選択の方法が重要なポイントとなる。

- (1) 複数のブール式を同時に多段化する場合、除算因子により商が異なるため、検出できる共通項に差が出る。図1の例では、2個のブール式を同時に多段化する場合、 $P = b$ で除算することにより共通項 ' $c+a$ ' の検出が可能となり、ゲート数削減効果が大きくなる。

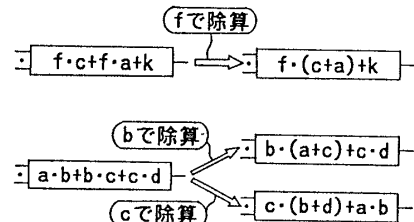


図1 複数ブール式間の共通項検出への影響

(2) 商 Q_i および、余り R_i に対してできるだけ多くの多段化ができるよう除算因子 P を選択する必要がある。図2の例では $P = a$ で除算した後その商を $P = b$ で除算した方がゲート数削減効果が大きくなる。

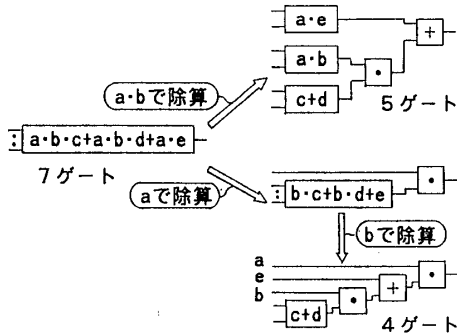


図2 単一ブール式の多段論理展開への影響

2.4 ブール式多段化アルゴリズム

除算因子の選択により生じるブール式多段化の多様性に対し、より有効なゲート数削減効果を得るために、ブール式多段化手順を tree で表現、検出可能な共通項や除算因子を先行探索する手法をとる。図3にアルゴリズムの概要を示す。

多段化の対象となるブール式は、まず剰余 R_i として登録され除算および共通項の検出がなされる。さらに得られた商、剰余、共通項剰余に対して同様の処理を繰り返す。この間の除算因子、商、剰余、共通項剰余の関係を Tree として表現する。

次にこの Tree によって表現された様々な式変形のパターンから最もゲート数削減効果の大きなものを選択する。このため、Tree のあるパスを選択した場合のゲート数削減効果効果を簡単な方法で評価し、最もゲート数削減効果の大きなものを選ぶ。

実際に Tree に展開した処理例を図4に示す。この例では第1式の除算因子の選択において、第2式との統合機能の結果得られる共通項 ($S=b·c·e+g$) によるゲート数削減効果を考慮して、リテラル a を除算因子として選択する。

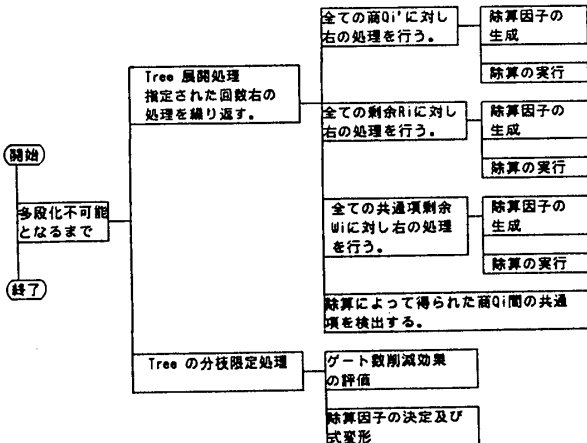


図3. アルゴリズム概要

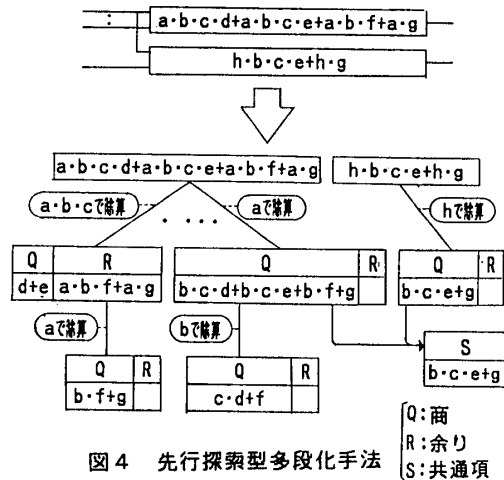


図4 先行探索型多段化手法

3. 遅延時間最適化方式

一般に単純な多段化機能は論理段数を増加させ、遅延時間制約上、問題となることがある。この問題に対してはブール式の2段化機能で対処する。

ブール式2段化機能はクリティカルパスの検出機能と、内部変数の対応するブール式への代入および括弧の展開機能とからなる。2段化機能ではクリティカルパス上にある信号名に対応するリテラルに着目し、そのリテラルに関連した部分のみブール式を2段化する。

また、ゲート数削減機能と遅延時間最適化機能の起動制御機能を用意し、論理回路全体でゲート数制約、遅延時間制約の両方を考慮した多段論理を生成する。

4. 結論

ゲート数削減方式としてブール式間の共通項の統合機能と除算機能を連動したブール式多段化アルゴリズムを提案した。本アルゴリズムでは除算における除算因子の選択により、多様な多段化が可能であり、同時に統合機能の効果が異なる点に留意し、ゲート数削減効果の大きい除算因子決定を可能とするため、式変形において先行探索機能を設けた。また同時に、遅延時間制約に対処するため、クリティカルパスを検出し、クリティカルパスに対応するリテラルに関連した部分のみを局所的に2段化する手法を提案した。

6. 参考文献

- 1) R.K.Brayton, et.al., Logic Minimization Algorithms for VLSI Synthesis; Kluwer Academic Publishers, ('84)
- 2) R.K.Brayton, et.al., Multi-Level Logic Optimization and The Rectangular Covering Problem; ICCAD'87, pp.66-69, ('87)
- 3) 影山他、論理最適化の一手法について; 情報処理学会 第34回全国大会 pp.1957-1958
- 4) 黒田他、パイプライン制御計算機の制御論理生成方式; 情報処理学会 第38回全国大会