

線形計画法を用いた 遅延時間最適化手法の検討

影山直洋 三浦地平 清水嗣雄
(株)日立製作所

近年、計算機論理の大規模化、複雑化に伴いその論理設計工数は増大しており、これに対処すべく論理自動生成システムの研究がなされてきた。論理自動生成システムでは生成論理の性能を人手設計論理に近ずけるために遅延時間最適化機能が必須である。しかし、従来手法では遅延時間短縮のための論理変更により冗長論理を生じ易い欠点があった。本稿で述べる手法では対象論理回路を複数の部分回路に分割し、各部分毎に機能的に等価でゲート数・遅延時間の異なる論理パターンを複数生成し、論理回路全体で一括して各部分回路の最適論理パターンを線形計画法を用いて選択する。本手法により冗長論理を生ずること無く遅延時間短縮が可能となる。

Research on the delay optimization by linear programming approach

Naohiro Kageyama Chihei Miura Tsuguo Shimizu
Hitachi, Ltd.

We propose a new algorithm for reducing the delay time without generating the redundant logic. This algorithm uses a linear programming approach and makes it possible that the delay and gate optimization processes are operated at the same time from the global point of view. In this algorithm logic circuit patterns which have same function and different number of gates or delay time are generated at the each part of logic circuit and optimal logic circuit pattern is selected simultaneously by linear programming approach. Therefore this new algorithm prevents the generation of redundant logic arising from delay time improvement.

1. 緒言

近年、計算機及びVLSIの論理規模拡大と制御論理の複雑化に伴い、計算機の開発期間及び開発工数が増加しており、特に論理設計工数の増大が問題となっている。これに対応すべくこれまで論理自動生成システムの研究がなされてきた。

論理自動生成システムの研究では、生成論理が人手設計以上の設計品質を得るための論理最適化アルゴリズムの研究が必須である。論理最適化機能は大別してゲート数削減機能と遅延時間短縮機能とがある。特に近年はLSIの高集積化に伴い、ある機能を論理回路として実現する場合、ゲート数の少ない論理回路よりむしろ遅延時間の短い論理回路として実現することに重点が移りつつあり、遅延時間短縮機能の研究が増々重要となっている。

遅延時間短縮機能を実現する場合解決しなければならぬ問題点として、遅延時間短縮のための論理変更に伴う冗長論理の生成が挙げられる。遅延時間短縮は冗長ゲートを削除することによって達成されるのでゲート数削減機能によって達成される場合も多い。反面多段化機能等のゲート数削減機能では遅延時間の増加を招き、論理段数圧縮機能等の遅延時間短縮機能では逆にゲート数の増加を招く。このため論理段数圧縮機能では適用対象範囲を限定しなければ冗長論理を生成しやすく、また遅延時間短縮機能とゲート数削減機能を組合せて最適化を行う必要がある。基本的には次の2種類のアプローチが考えられる。

- (1) 遅延時間制約を満足しゲート数最小となる論理回路を求める。
- (2) ゲート数制約を満足し、性能最適となる論理回路を求める。

本稿では(1)のアプローチを採用するが、遅延時間を制約条件として対象論理に線形計

画法を適用することで遅延時間制約を満足し、冗長ゲートの無い論理回路を生成する手法を提案する。

対象論理はブール式レベルの論理であるが、ブール式を対象とした最適化処理アルゴリズムは、これまでに様々なものが提案されており、国内外で活発に研究が進められてきた。積項数削減アルゴリズムとしては、ESPRESSO-DC¹⁾が、多段化アルゴリズムとしてはWeak Division, Strong Division²⁾が有名であり、様々なシステムに組み込まれている。しかし、遅延時間制約までも考慮したシステムは、MIS³⁾ YSC⁴⁾ SOCRATES⁵⁾等僅かしかなく機能的にも不十分なものが多い。今後ますます研究テーマとして重要となると思われる。

2. 遅延時間最適化手法の問題点

従来より用いられてきた遅延時間最適化手法には以下の問題点があった。

- (1) 局所的・逐次的に論理回路を改良する。

論理回路中遅延時間短縮不要の部分までも処理し、冗長な論理回路を生成する。

- (2) 遅延時間の短縮とゲート数の削減を独立して行う。

遅延時間短縮機能とゲート数削減機能は相反する効果をもつため両機能間の調整が困難である。

以下では具体例を用いて説明する。

到着時刻のみを考慮した場合、信号源側より逐次的に各ノードの到着時刻を改良してゆけば遅延時間最小の論理回路が得られる。しかしその結果冗長な論理回路を生ずる場合がある。図2.1において node1～node5のブール式の内、遅延時間短縮の対象となるのは node3 と node4 のブール式である。node3 は、(a) に示す式変形が可能である。node4 は (b) 及び (c) の形に式変形できるが、遅

延時間を優先して逐次的にノードの変形を行う手法では (c) の変形が選択される。しかし、node3 の式変形結果と node5 を同時に考慮した場合、node4 の式変形は (b)、(c) いずれでも node5 の出力信号の到着時刻に変化はない。従って (b) を選択した方がゲート数を考慮した場合有利である。

このようにノード内ブール式の変形を行う場合、論理回路全体を一括して、ゲート数削減と遅延時間の改良を同時に考慮することにより、最適な論理回路を得ることができる。

3. 線形計画法を用いた論理最適化手法

3.1 用語の定義

以下では、本稿で用いる用語をまとめて定義する。

(1) Boolean Network

論理回路を有向・非循環グラフでモデル化したもので、このグラフを対象として遅延時間の改良を行う。グラフの各ノードは、一次

入出力ピン、ゲート、2段論理を表現する積和形ブール式又は真理値表に対応する。各エッジはノード間の結線関係を表現する。

(2) 到着時刻³⁾

信号がノード内で合成されノードの出力端子に出力される時刻のことである。本稿では時間の単位を、 n 入力ゲートを仮定しこのゲートを通過するときの信号の遅れを1単位とする。また、配線内の信号の遅れは生じないものとする。

図 3.1 においてノードの入力信号を A, B, C, D とし、これらの信号のノードの入力ピンでの到着時刻を $\alpha, \beta, \gamma, \delta$ とする。さらに着目したノード内での論理段数をそれぞれ a, b, c, d とすると着目したノードの出力ピンの信号到着時刻 AT は、

$$AT = \text{Max}(\alpha + a, \beta + b, \gamma + c, \delta + d)$$

このように、本手法ではノードの出力ピンでの信号の到着時刻はノード内の組合せ回路の種類に関わらず入力ピンの到着時刻とノード内論理段数を足したものが最大となる信号によって決定されるモデルを採用する。到着

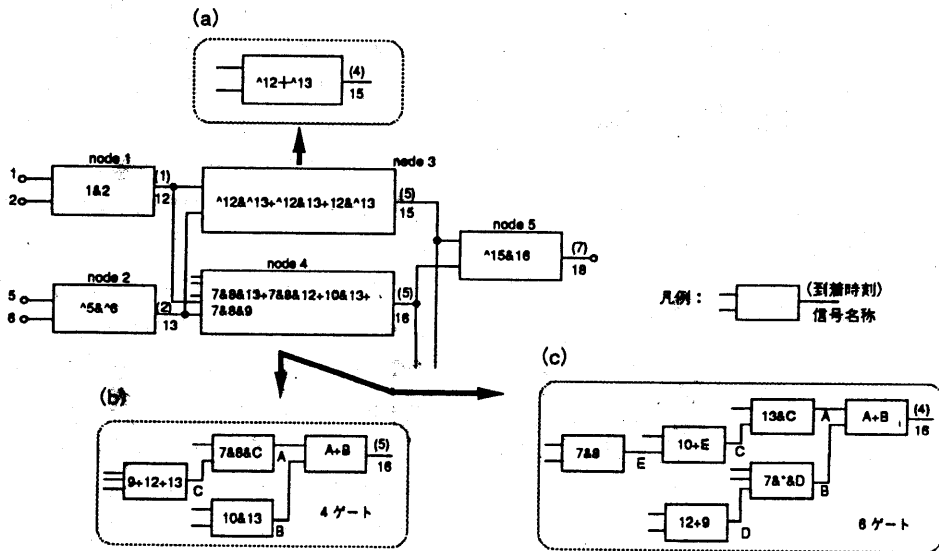


図 2.1 従来手法の問題点

時刻は、論理回路中入力ピン側のノードより出力ピン側のノードに向かって計算される。

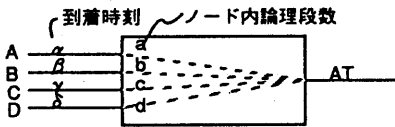
(3) スラック**

スラックは、各ノードにおいて信号がどれだけ遅延を許容できるかを示す値である。ユーザの指定する一次出力ピンの信号の到着要求時刻を基に一次入力ピンに向かって後方検索し、各ノードの出力ピンの信号到着要求時刻を求める。スラックは、この到着要求時刻より到着時刻を引いた値である。

図3.2 において、一次出力ピンに対するユーザが指定した信号到着要求時刻(遅延時間の制限値)をTとし、到着時刻を $t(v)$ とする。一次出力ピンに対するスラック $s(v,T)$ は次のようになる。

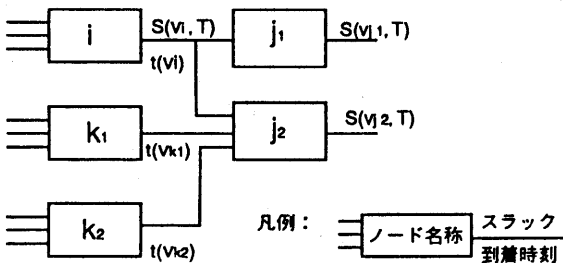
$$s(v_i, T) = t(v_i) - T$$

また、一次出力ピン以外の任意のノードにおけるスラックは次のように定義される。



$$AT = \text{MAX}\{ \alpha+a, \beta+b, \gamma+c, \delta+d \}$$

図 3.1 到着時刻の定義



$$s(v_i, T) = \min \{ s(v_k, T) + \max_k t(v_k) - t(v_i) \}$$

図 3.2 スラックの定義

$$s(v_i, T) =$$

$$\min \{ s(v_j, T) + \max_k t(v_k) - t(v_i) \}$$

3. 2 処理概念

2章で述べたように従来手法では、対象論理回路に対しゲート数削減と遅延時間改良を独立に、しかも局所的かつ逐次的に行うため、処理時間が増大しかつ冗長な論理回路を生成しやすかった。そこで、本節では従来の手法に代わるものとして対象論理回路を分割し、分割された部分単位に機能的に等価でゲート数・論理段数の異なる論理回路(これを論理パターンと呼ぶ。)を複数生成し、論理回路全体で一括して各部分単位に最適な論理パターンを選択する手法を提案する。このような手法は、数理計画法を用いることによって可能である。

本節で提案する手法の目的は、複数の Flip-Flop 間、入力ピン - Flip-Flop 間或いは Flip-Flop - 出力ピン間の信号到着時刻がユーザ指定の制限値内に収まる論理回路の内、ゲート数が最小となるものを求めることにある。対象となる論理回路はいわゆる Boolean Network で表現され、各ノード内の論理回路はファンアウトを持たないいわゆるファンアウトフリーレジョン (FFR) とし、機能を表現するブール式は積和形ブール式とする。

図3.3 に示すように本手法では、対象論理回路の各ノードごとに機能が等価でゲート数、遅延時間の異なる論理パターンを複数用意する。ユーザが指定する遅延時間制約をもとに各バスごとに制約条件式を作成し、ゲート数を目的関数として、数理計画法を適用することにより、各ノード単位に最適な論理パターンを論理回路全体で一括して選択する。

本手法により、遅延時間に余裕のあるノ-

ドはゲート数の少ない論理パターンが選択され、余裕のないノードは遅延時間の短縮された論理パターンが選択される。スラックが負のノードでは、その解消はこのノードを含む信号経路全体で、ゲート数の増加を考慮して一括して行なわれ、1ノードだけでの局所的な解消は行われぬ。このように、ゲート数の削減と遅延時間の改良とが同時にしかも論理回路全体で一括して可能である。

尚、本手法ではゲート数及び遅延時間は全てn入力ゲートを仮定し、配線の遅延は考慮しない。

次に具体的な定式化方法について述べる。

3. 3 定式化

数理計画法を適用するためには制約条件式及び、目的関数が必要である。本手法では制

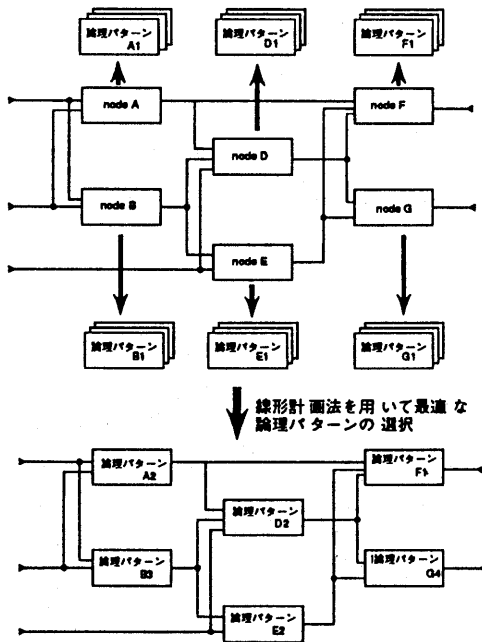


図 3.3 処理概念

約条件式としてユーザの指定する遅延時間をまた目的関数としてゲート数を採用する。

(a) 制約条件式

複数のFlip-Flop 間の制約条件式は次のように求めることができる(Input-Flip-Flop 間、Flip-Flop - Output 間の制約条件も同様)。図3.4 に示すように node1 の出力信号 X の到着時刻は

$$\text{MAX}(a+a, \beta+b, \gamma+c)$$

となる。さらに node2 の出力信号 Y の到着時刻は、

$$\text{MAX}(a+a+x, \beta+b+x, \gamma+c+x, \delta+d, \epsilon+e, \zeta+f)$$

以上のことから、論理段数を遅延時間とするモデルで考えた場合、Flip-Flop 間の論理回路がどのようなものであっても出力ピンに到る最長論理段数を持つパスが出力ピンの到着時刻を決定する。よって任意のパスについて遅延時間の制約条件を満たせば遅延制約は満足されたことになり、式で表現すると次のようになる。

C をユーザが指定した Flip-Flop 間の最大遅延時間制約とすると、

$$T(\text{path}(\text{any})) < C$$

ノード i の j 番目の論理パターンに対応する選択変数 $\lambda(i, j)$ を用いて表現すると以下ようになる。ただし選択変数 $\lambda(i, j) = 0, 1$ であり、 $\lambda(i, j) = 1$ の時、

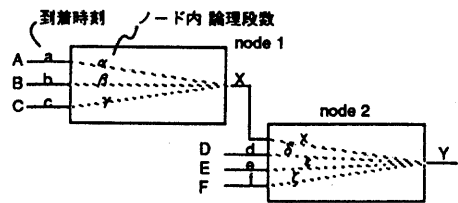


図 3.4 制約条件式の作成

ノード i の j 番目の論理パターンを選択することを意味する。

あるパス k に関して、 $i \in k$ として

$$\sum_i \sum_j \lambda(i,j) * d(i,j,k) < C(k) \quad (1)$$

$$\sum_j \lambda(i,j) = 1 \quad \forall i \quad (2)$$

$$\lambda(i,j) = 0, 1 \quad (3)$$

ここで、 $d(i,j,k)$ はノード i の j 番目の論理パターンのパス k に関する遅延時間。

さらに上式において、

$$0 \leq \lambda(i,j) \leq 1 \quad (4)$$

として線形条件とする。

(b) 目的関数

ゲート数の目的関数は、 $\lambda(i,j)$ を用いて以下のように表現できる。

$$G = \sum_i \sum_j \lambda(i,j) * g(i,j) \quad (5)$$

ただし、 $g(i,j)$ はノード i の j 番目の論理パターンのゲート数。

以上が条件式及び目的関数であるが、条件式 (1)、(2)、(4) の制約の下、目的関数 (5) を最小にすれば良い。

3. 4 論理パターン生成方法

各ノードごとに積和形ブール式を対象に、大別して遅延時間短縮論理パターンとゲート数削減論理パターンの2種類を生成する。生成方法として、参考文献3) に述べられている因子化及びブール式分割の2種類の方法を基にした。

(1) 因子化

単一積和形ブール式を対象に因子化を行い多段論理のパターンを生成する。除数のリテラルに対応する信号は出力ピンに近すぎ、商内のリテラルに対応する信号は出力ピンより遠ざかるため、各信号ごとに遅延時間が変化する。また、除算するリテラルによって得られる論理回路のゲート数も異なる。到着時刻の大きな信号に対応するリテラルを除数とし

て複数選択し除算順序を変えることで複数の論理パターンを作成する。例を図3.5 に示す。

(2) ブール式分割

n 入力ゲートを仮定し演算子により連結されたブール式の分割を行う。出力ピンに近いノードに置かれた信号の遅延時間の改良がなされる。因子化と同様到着時刻の大きな信号を複数選択し、選択された信号間でどのノードに信号を配置するかを変えることで複数の論理パターンを作成する。因子化とは異なり分割の仕方によりゲート数は変化しない。

最初に因子化を行い共通因子を全て検出した後、ブール式分割を行う。

3. 5 処理対象検出方法

複数のFlip-Flop間、入力ピンとFlip-Flop間及びFlip-Flopと出力ピン間の互いに連結した組合せ回路を一括して取扱う。切り出し手順は以下の通りである。

- (1) 信号転送先側のある1つのFlip-Flopに着目する。
- (2) 着目したFlip-Flopより信号転送源側に後方検索し、転送源側のFlip-Flopを全て検出する。

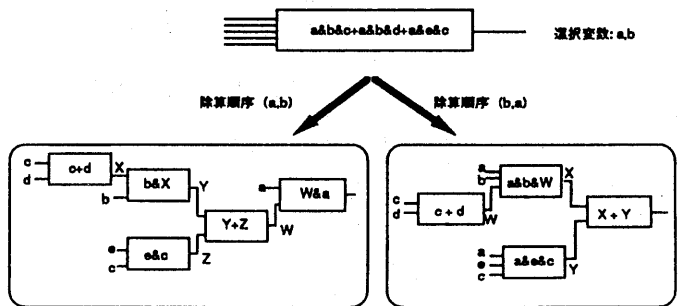


図 3.5 因子化による論理パターン作成

(3) 検出された転送源側の全ての Flip-Flop より前方検索し、転送先の全ての Flip-Flop を検出する。

(2)、(3) の処理を新に Flip-Flop が検出されなくなるまで繰返し行う。このようにして得られた転送源、転送先の Flip-Flop 群の間の組合せ回路を、処理対象論理回路として検出する。

検出された処理対象論理回路の規模が大きい場合、信号バスが多くなり一括して取扱えない場合が生ずる。この時は、スラックの大きなノードは処理対象からはずすことでバスの本数を削減する。本手法はもともと遅延時間の改良をゲート数の増加を最小にして行うことを目的としており遅延時間の改良に全く影響のないノードは一括処理対象から除いても問題ない。

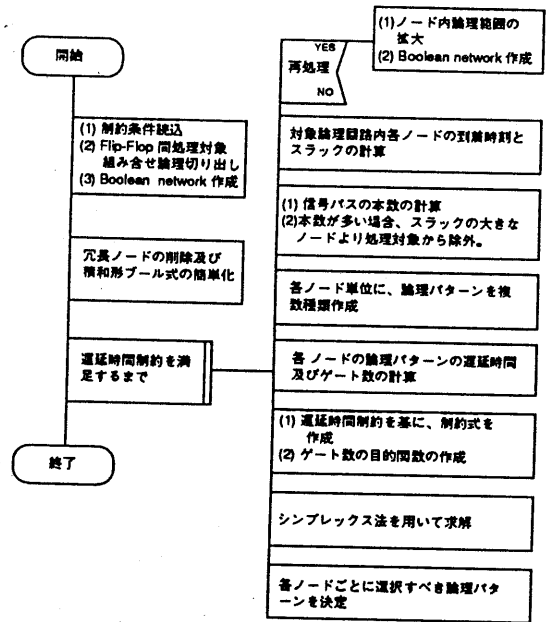


図 3.6 処理手順

3. 6 処理手順

図3.6 に処理手順を示す。図中ノード内論理範囲の拡大とは、遅延時間短縮処理を行った結果、遅延時間制約を満足する解を得られない場合に、各ノード内に含まれる論理回路の範囲を拡大し、再度処理を行う。各ノードは FFR に対応しており、FFR の拡大を行う。

4. システム構成

今回提案した手法は、論理自動生成アルゴリズム ASTRO (Associated Structure Oriented Control Logic Synthesis Algorithm)⁹⁾ の一部を構成する。ASTRO は、パイプライン方式の制御論理生成及び最適化を可能とするアルゴリズムであり、動作レベル論理記述を入力としブール式レベル論理記述を出力する。図4.1 に ASTRO の構成を示す。

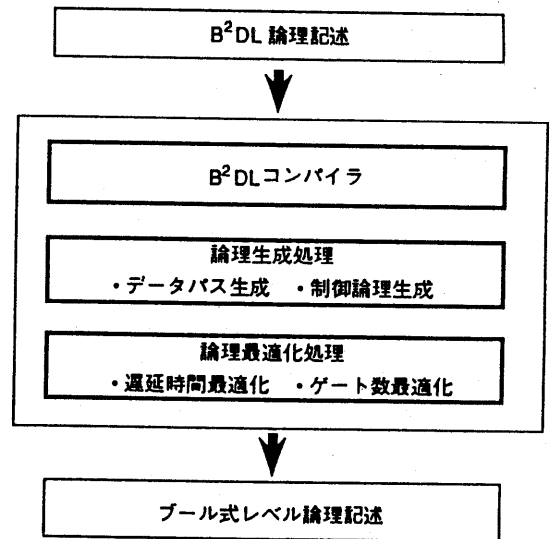


図 4.1 ASTRO の構成

5. 評価結果

DAC' 86 においてA.J.de Geus によって提唱されたベンチマーク^{*)}の内、ALUPLA を用いて評価を行った結果を表5.1 に示す。ここでは3入力ゲートを仮定した。遅延時間制約を18とした場合、元の論理回路と比較して最大で39%、平均でも23%の遅延時間の改良がみられた。また、ゲート数も22%改良された。

表5.1 評価結果

比較項目	信号名称	処理前	処理後	削減率 (%)
	遅延時間 ^{*)}			
#138		18	14	22
#139		21	17	19
#140		24	18	25
#141		20	18	10
平均				23
ゲート数 ^{*)}		205	159	22

^{*)} 遅延時間及びゲート数は3入力ゲートを仮定した。

6. 結言

遅延時間制約を満足し、ゲート数最小となる論理回路を求める手法を提案した。本手法では対象論理回路をFFR単位に分割し、各領域ごとに機能が等価でゲート数と遅延時間の異なる複数の論理パターンを作成する。さらに線形計画法を用いて対象論理回路全体で一括して各領域の最適な論理パターンを選択することにより遅延時間制約を満足し、しかもゲート数が最小となる論理回路を求める。

今回提案した手法により、従来の遅延時間短縮手法が持っていた局所的かつ逐次的な論

理回路の改良による冗長論理の発生及び処理時間の増大を抑止することが可能である。

7. 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 Optimization and The Rectangular Covering Problem," ICCAD' 87 pp.66-69
- 3) K.J.Singh et. al., "Timing Optimization of Combinational Logic," ICCAD' 88 pp.282-285
- 4) G.D.Micheli, "Performance-Oriented Synthesis of Large-Scale Domino CMOS Circuit," IEEE Trans. on CAD, Vol.6 No. 5, Sept. 1987 pp.751-765
- 5) K.Bartlett, "Synthesis and Optimization of Multilevel Logic under Timing Constraints," IEEE Trans. on CAD, Vol.5 No.4, Oct. 1986 pp.582-594
- 6) T.Shimizu, et. al., "A Control Logic Synthesis and Optimization Algorithm with an Overlap Degree Vector," ICCAD' 87, pp.124-129.
- 7) A.J.de Geus et al., "Logic Synthesis and Optimization Benchmarks for the 1986 Design Automation Conference," In Proceeding of DAC' 86, June 1986, pp78.