

# 遅延時間最適化方式の検討

5V-1

影山 直洋 清水 嗣雄 宮本 俊介  
(株) 日立製作所

## 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。論理設計を自動化するに当っては、自動設計された論理回路が要求性能を満足するように論理回路内の遅延時間の短縮を行う遅延時間最適化機能が必須である。これまでゲートレベルでの局所変換やゲート分割等による遅延時間改良手法の提案はなされてきた<sup>1-2)</sup>。本稿ではブール式レベル論理を対象とする大域的な遅延時間最適化方式を提案する。

## 2. 遅延時間最適化方式

### 2.1 従来手法とその問題点

従来提案されてきた遅延時間短縮手法は、クリティカルパス上のゲートの駆動能力の大きなゲートへの置換等のゲートレベルでの局所変換やゲート分割などが主であった。しかしこれらの手法はいずれも局所的で、一度に取扱うことのできる論理ゲート数に制限があり、遅延時間短縮のために大域的な論理変更が必要となった場合対処できなかった。

### 2.2 基本概念

上記問題点に対し、ブール式レベル論理を対象としたクリティカルパス逐次分割による遅延時間最適化方式を提案する。本方式の対象となる論理回路は複数の機能ブロックのネットワークによって構成され、また機能ブロックはその機能表現としてブール式を持つ。対象論理回路の例を図1に示す。

本方式では、ネットワーク上のクリティカルパスに対し、クリティカルパスに沿ったファンアウトフリーリージョン(ファンアウトのない領域、以下FFRと略す。)を用いてクリティカルパスを分割し、FFR単位に論理段数の圧縮を行う。さらに設計制約を満たさない場合には隣接する複数のFFRを融合し、順次論理段数圧縮対象範囲を拡大することで、より広範囲な論理を一括して取扱い論理段数圧縮を実行する。

この結果得られる論理は、ゲートに展開した後、より精密に遅延時間の改良を行う場合に、容易に設計制約を満足させることができる。

### 2.3 処理手順

図2に示すように論理段数を制約条件として以下の手順で論理段数圧縮を行う。

- (1) フリップフロップ間の信号経路の中で、論理段数制約を満たさないものをクリティカルパスとして検出する。
- (2) 検出されたクリティカルパスをFFR単位に分割する。

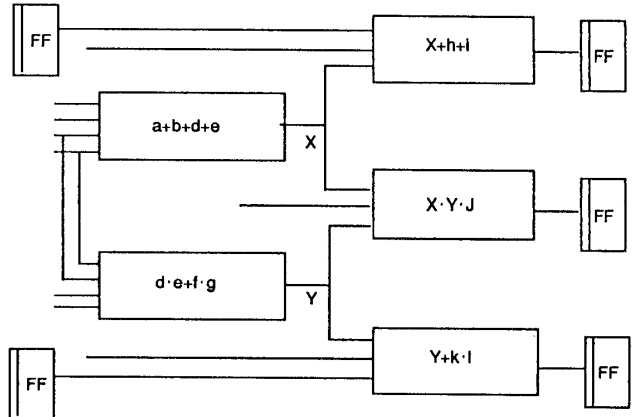


図1 処理対象

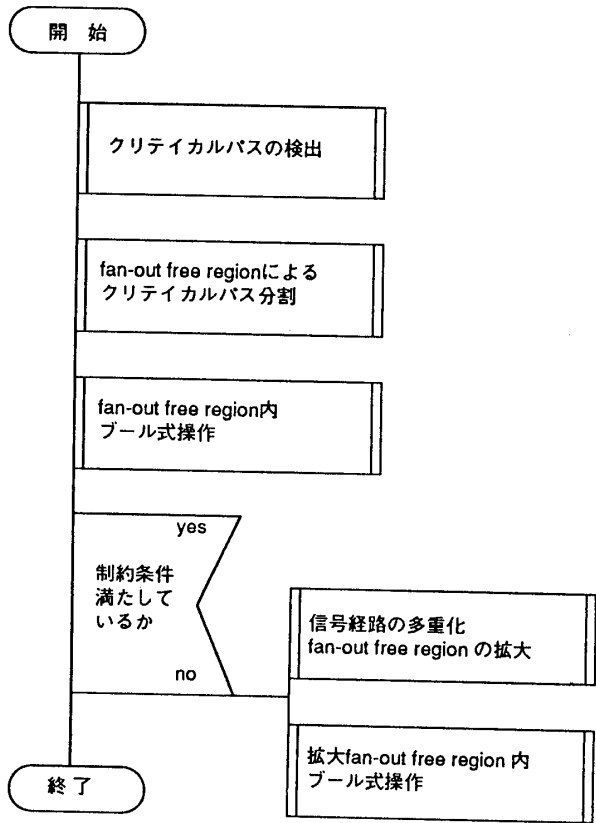


図2. 遅延時間最適化処理フロー

(3) 各FFRごとにブール式操作を行い、論理段数の圧縮を行う。

(4) 論理段数制約を満たさない場合は信号経路の多重化を行いFFRを拡大し、拡大されたFFR内で論理段数の圧縮を行う。

図3に遅延時間最適化の具体的な処理例を示す。図の例では3入力ゲートを仮定した場合、当初論理回路中のクリティカルパスの論理段数が6段((a)の状態)であったのに対し、FFR内論理段数圧縮で4段((b)の状態)に、さらに拡大FFR内の論理段数圧縮により3段((d)の状態)にまで低減できる。以下では信号経路の多重化によるFFRの拡大操作とブール式操作について詳述する。

### 2.4 FFRの拡大操作

論理段数の圧縮を行う場合、ある領域内で論理段数の圧縮が不可能な場合でも、領域を拡大することでさらに論理段数の圧縮が可能となる場合がある。論理段数の圧縮は、ブール式の式変形の等価性を保つためにFFR単位に行う手法をとり、順次隣接するFFRを併合し処理対象領域を拡大する手法を採用した。具体的には信号経路を多重化することでファンアウトを解消しFFRの拡大を行う。

図3において(c)、(d)は信号経路の多重化及びFFRの拡大を示した図である。

図3(c)では機能ブロックを複製することで信号経路の多重化がなされ、この結果ファンアウトが解消されFFRが拡大される。

図3(d)では、拡大FFR内の機能ブロックが1つにまとめられ、ブール式を積和形ブール式に展開した結果が示されている。

このように複数のFFRを連結し拡大FFRを構成する場合、どのFFRを連結するかによって論理段数の圧縮効果は異なる。これに対して複数のFFR連結候補を作成し、その内最も論理段数圧縮効果の大きいものを選択する手法を採用した。

### 2.5 ブール式操作

FFR及び拡大FFR内のブール式に対しては以下の操作を施す。

#### (a) 簡約化処理

機能ブロックのブール式の冗長表現を削除し積項数を削減する。

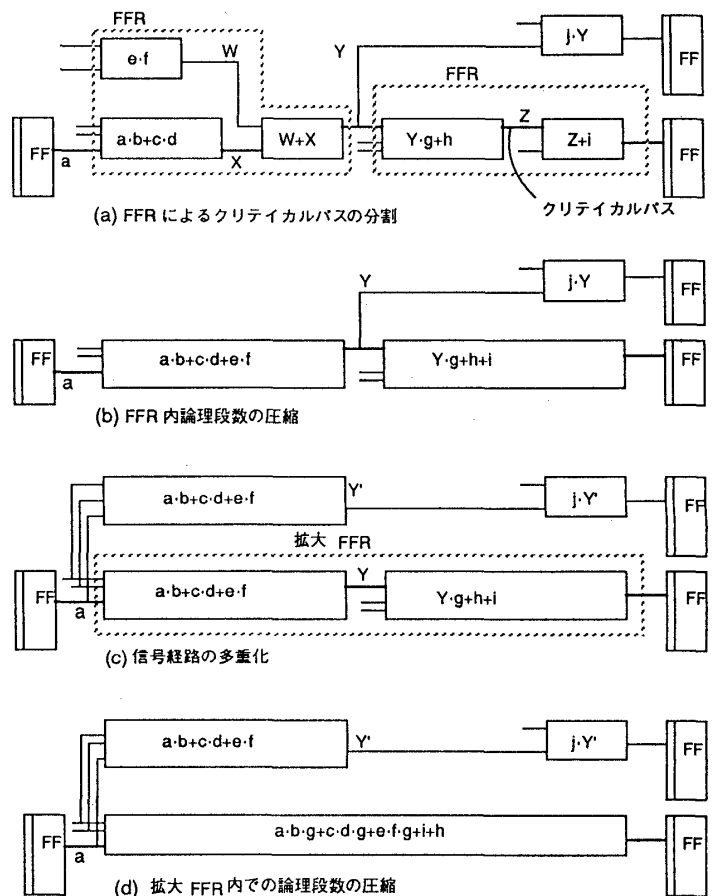
#### (b) 括弧の展開

FFR内の機能ブロックをまとめて得られるブール式は多段論理を表現する括弧を含むブール式である。よって論理段数圧縮のためその展開を行う。この時、クリティカルパス上のリテラルに関してのみ展開を行う。

### 3. 評価結果

上記方式を持つ自動論理設計システムの最適化処理部ではユーザから指定されるゲート優先モードと遅延優先モードに応じて最適化を行う。ここでゲート優先モードとは、ゲート数を優先的に削減するモードであり、その機能としては参考文献3)に示す先行探索型多段化手法のほか、積和形ブール式の簡約機能や冗長な機能ブロックの削除機能を持つ。また遅延優先モードとは、論理段数の圧縮を優先的にを行うモードである。

1750ゲート規模のLSIを対象に評価を行った結果、ゲート優先モードでは人手設計に比べゲート数10%増の結果を得た。また遅延優先モードでは、特に拡大FFRを用いた場合、クリティカルパスは完全



FFR : fan-out free region

図3 遅延時間最適化処理例

に解消でき、しかもゲート数の増加はゲート優先モードに比較し5%以内であった。

### 4. 結言

ブール式で表現された論理回路を対象とする遅延時間最適化方式について提案した。本方式では、ブール式レベルの論理を対象とし、クリティカルパスに沿ったFFR単位に論理段数圧縮を実行する。さらに論理段数制約を満たさない場合は順次論理段数圧縮対象のFFRを拡大し対処する手法である。

この結果、1750ゲート規模のLSIを対象とした評価では、ゲート数を優先的に削減するモードに比較して、ゲート数が5%増加しただけでクリティカルパスを全て解消できた。

### 5. 参考文献

- 1) A.J.de Geus, et al., A Rule-Based System for Optimizing Combinational Logic; IEEE Design & Test of Computers, pp.22-32(Aug.'85)
- 2) M.Hofmann, et al., Delay Optimization of Combinational Static CMOS Logic; DAC'87 pp. 125-132
- 3) 影山他、多段論理最適化方式の検討; 情報処理学会 第38回全国大会 pp.1359-1360