形式的論理検証における一手法

7P－5

古田康幸† 小澤彰一† Yung-Te Lai‡ 鈴木五郎†
†（株）日立製作所 大みか工場 ‡HMSI

1.はじめに

形式的論理検証システムHIFORM (Hitachi Hierarchical Formal Verification) を開発した．BDD (Binary Decision Diagram) [1][2][3]をベースにしているが，論理の階層構造を利用し，また構造の不一致個所だけを切り出してBDDを作成する等の工夫をして機能一致検証の高速化を図った．

2.機能一致検証の高速化

まず，BDDだけを用いたプロトタイプを作成し，ゲートレベルの二つの論理を比較する処理時間の評価を行なった．図１のグラフから分かるように，ゲート数が50Kゲート以上の大規模LSIでは，実用的な時間内での検証は不可能である．そこで，処理の80％程度はBDD作成に費やされることから，BDDを作成する対象範囲をできるだけ絞り込むことによって高速化を図ることにした．

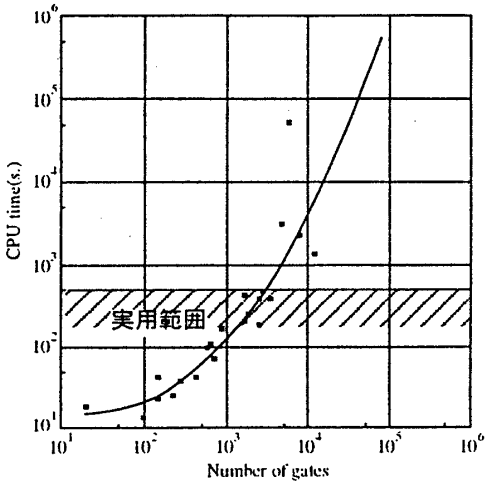

図１　プロトタイプの検証時間

2.1 論理階層構造の利用

第一の戦略は，論理階層構造の利用である．通常LSIは階層設計されるが，特にASICの場合は論理設計の最初からレイアウト設計まで同一階層で設計を行なう．この性質を利用して，機能とゲート，ゲートとゲートいずれの比較の場合でも構造上の比較をまず行なうことにした．

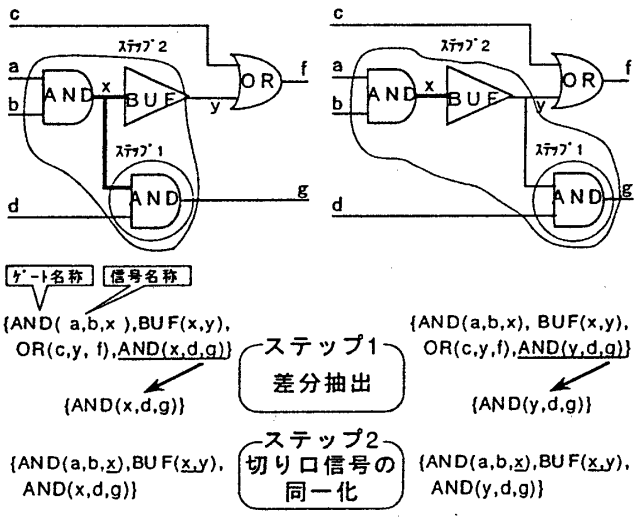

図２　構造不一致個所の抽出処理

2.2 構造不一致個所の抽出

第二の戦略は，ゲートとゲートの比較の場合の構造不一致個所の抽出である．論理構造の異なる部分だけを切り出してBDDを作成し論理の一致を検証する．構造不一致部分の抽出処理を図２を用いて説明する．回路のゲート記述から，ゲート名称と信号名称を用いて接続関係をそれぞれ

A Speeding Up Method of Formal Verification

Yasuyuki Furuta†, Shouichi Ozawa†, Yung-Te Lai‡, Goro Suzuki†
† Hitachi, Ltd. Omika Works. ‡ Hitachi Micro Systems, Inc.

{AND(a,b,x),BUF(x,y),OR(c,y,f),AND(x,d,g)}, {AND(a,b,x),BUF(x,y),OR(c,y,f),AND(y,d,g)}

と表す．ステップ１として接続関係を比較し，他方に含まれていない接続関係つまり差分を抽出すると

AND(x,d,g), AND(y,d,g)

となり，取り出した回路の切り口の信号線は(x,d,g), (y,d,g)となるが，構造不一致部分を抽出するには切り口の信号線を同じにする必要がある．そこでステップ２としてxにつながるゲート名称と信号名称を付加する．この場合xに接続しているANDとBUFを付加すると，切り口の信号線はどちらも(a,b,d,y,g)となり，それぞれを構造不一致部分として抽出できる．

3. 評価結果

機能一致検証の高速化を図った後のHIFORMにおいて，それぞれゲート規模の異なる４つのLSIで評価を行った．論理階層数は４，エラーとしてはパワーゲート置換の誤り，使用したマシンはSparc Station 10である．

(1) 検証時間

ゲートとゲートの比較を行った場合，200Kゲートの論理でCPUタイム300秒を実現しており，ゲート数をNとすると$O(N)$の計算複雑度を持った処理時間である．ちなみにシミュレータVerilog_XL™を用いて同一のエラーを見つける場合3000秒ほど必要となった．

(2)エラー的中率

HIFORMのプロトタイプにおいて，ゲート記述同士の検証を行った際に指摘したエラーの的中率を表１に示す．的中率とはエラーと指摘した中に真のエラーの含まれている割合のことである．BDDの性質上，検証したブロックが等しいかどうかを示すだけなので，検証して等価でないとされたブロックの全てのゲート数がこの指摘エラーの値となる．６Kゲートの回路で指摘されたエラー数585個中の１個だけが真のエラーであり，その的中率は0.2%であった．４つのケースの的中率は平均すると1.1%と非常に低かった．しかし，機能一致検証の高速化で述べた第二の戦略，つまり構造不一致個所のみを抽出する処理を施したことによって平均的中率68%まで大幅に改善された．100%には遠い数字であるが，摘出エラー数が一桁であることから実用上は問題無いと考える．

4. まとめ

従来，形式的論理検証におけるエラー個所の同定にはシミュレータを用いるが，提案している手法では論理分割を行うことにより，実用的な的中率での同定を可能にした．又，階層構造の利用と論理分割により，時間のかかるBDDの作成処理を最小限とし，高速化を図ることができた．

表１　HIFORM指摘エラー的中率

ケース	ゲート数	真ｴﾗｰ/摘出ｴﾗｰ	的中率 (%)
(1)	6 kｹﾞｰﾄ	1 / 2	50
(2)	50 kｹﾞｰﾄ	4 / 5	80
(3)	85 kｹﾞｰﾄ	4 / 6	67
(4)	200 kｹﾞｰﾄ	6 / 8	75

平均 68%

参考文献

[1] R.E. Bryant : Graph-based algorithms for Boolean Function Manipulation : IEEE Trans. on Computer, Vol. C-35, No.8, pp.677-691 (1986 Aug.)

[2] R.Rudell : Variable Ordering for Ordered Binary Decision Diagram : ICCAD-93, pp.42-47 (1993 Oct.)

[3] Steven Schulz : Ready For Prime Time : Integrated System Design, pp.18-30 (1995 Apr.)