

# グラフに基づく論理照合アルゴリズム の評価と改良

3U-5

藤沢 久典      藤田 昌宏      川戸 信明  
富士通研究所

## 1. はじめに

回路の自動合成においては、2つの回路の論理的な一致の検証は非常に重要な問題である。従来は、入力に対する出力を比較することにより検証を行う方法が主であったが、この方法では入力数の増加に伴い入力パターン数が指数的に増加し入力数40の回路を扱うのが精一杯であった。そこで我々は Bryant によって提案された回路をグラフに変換する方法 [1] を論理照合に対して適用し評価改良をおこなった。

## 2. グラフへの変換

グラフとは入力変数に対して順序付けが行われた2分決定木のことであり、等しい論理を持つ回路は同じ形を持つグラフに変換することができる。そのためグラフの形を調べることにより論理検証が可能となる。

グラフへの変換における手間はグラフの大きさに依存し、比較的入力数の多い回路に対しても高速な検証が期待できる。ところがグラフの大きさは入力変数の順序付けに対して大きく依存するため、最適な順序付けを行うことが必要であるが、未だ有効なアルゴリズムは発見されていない。そこで、最適解ではないが多くの回路において実用的に有効な解を、小さな手間で見つけるアルゴリズムを提案する。アルゴリズムは2つの基本的な考えに基づいている。

- (1) 同じゲートに繋がった入力は隣あった順序とする。
- (2) 複数ゲートに繋がっている入力は先に順序付けを行う

アルゴリズムを Fig. 1 に示す。

## 3. 実行結果

ISCAS のベンチマーク用回路 [2] を対象に、上記の方法を適用し評価をおこなった。その結果を Table. 2 に示す。但し検証は各出力ごとにおこなっている。プログラム言語にはCを用い、SUN3/260ワークステーション上で実行した。変数の順序付けの方法としては、ISCAS のデータに現れている順、ランダムな順及び上で述べたアルゴリズムによる順の3種類の順序付けを行って比較した。その結果上述のアルゴリズムを用いることにより、入力数207ゲート数3827の回路を含む大部分の回路に対して実用的な時間で検証が可能であった。

その他、回路を多出力のままグラフに変換して検証を行った結果についても示す。

## 4. まとめ

この順序付けアルゴリズムを含むグラフへの変換法は論理照合において、特に入力数が多く論理が簡単な回路に対しては、非常に有効であることがわかった。今後は多段論理の簡単化への適用等を考えている。

- [1] R.E.Bryant : Graph-based Algorithms for Boolean Function Manipulation, IEEE Trans. Computer, C-35(8) 667-691, August 1986.
- [2] F.Brglez and H.Pujiwara : A Neutral List of 10 Benchmark Circuits and a Target Translator in Fortran, Proc. IEEE int. Symp. on Circuits and Systems, 663-698, June 1985.

```
{ Initially all nets must be marked off }
procedure makeOrder(N);
begin
  foreach I <- Set of all input nets of the gate to which N is connected do
  begin
    if I is marked then continue;
    if I is directly connected to a primal input then
      if I is connected more than one gate then
        begin
          FANOUT2UP := I;
          if I is not in ORDER then ORDER := append(ORDER,I);
        end else FANOUT1LIST := append(FANOUT1LIST, I);
        else makeOrder(I);
      end;
    if FANOUT2UP <> NIL then
      begin
        Insert FANOUT1LIST into ORDER just after FANOUT2UP;
        FANOUT1LIST := NIL;
      end;
    Mark N;
  end;
  return;
end;
```

Fig. 1 変数の順序付けアルゴリズム

Name	Inputs	Outputs	Original oder		Random oder		Our argorithm oder		Multiple output handling time
			Time	Maximum nodes	Time	Maximum nodes	Time	Maximum nodes	
SN181	14	8	8	578	6	213	6	213	3
c432	36	7	109	1146	unable	>10 <sup>5</sup>	1423	6196	529
c499	41	32	928	9020	unable	>10 <sup>5</sup>	673	4661	312
c880	60	26	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	55	3359	34
c1355	41	32	675	9020	unable	>10 <sup>5</sup>	1009	4661	—
c1908	33	25	965	2919	>5×10 <sup>4</sup>	—	778	3076	441
c2670	233	140	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	338	14763	336
c3540	50	22	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	12620	53460	7613
c5315	178	123	2140	11807	unable	>10 <sup>5</sup>	498	3441	253
c6288	32	32	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	unable
c7552	207	108	unable	>10 <sup>5</sup>	unable	>10 <sup>5</sup>	383	2096	unable

Table. 1 実行結果

Evaluation and Improvement of Graph Based Boolean Comparison Algorithm

Hisanori FUJISAWA Masato FUJITA Nobuaki KAWATO  
FUJITSU LABORATORIES Ltd.