4L－1

整数上の論理式の恒真性判定アルゴリズムを用いた組合せ論理回路の実現の正しさの証明

森岡 澄夫　北道 淳司　東野 輝夫　谷口 健一

大阪大学 基礎工学部 情報工学科

1 まえがき

組合せ論理回路が任意の入力に対して正しい演算結果を出力することを，機械的に証明できることが望ましい．本稿では，整数演算を行う回路の仕様（回路が行うべき演算）と実現（論理ゲートの結線）をそれぞれ整数上の論理式で記述し，実現が仕様を満たすことを整数上の論理式の真偽判定アルゴリズムを用いて証明した例を示す．整数演算や論理演算を行う組み合わせ論理回路の仕様は，比較や加減算を用いた整数上の論理式により，自然に分かりやすく記述できる．また，整数上の論理式の真偽判定アルゴリズムにより，仕様をブール代数の式で与える場合とは異なり，回路出力が例えば加算等の演算結果になっていることを直接証明できる．今回，整数上の論理式の真偽判定ルーチンを，組合せ論理回路の検証用に工夫して実装した．そのルーチンにより，TTL の 4 ビット ALU 74382 の正しさを 4 分程度で高速に証明できた．

2 プレスブルガー文

加算を持つ整数の理論（整数の集合 Z 上の変数, 定数, 整数の加減算, 整数同士の比較演算,∧, ∨, ¬, ∀, ∃ からなる理論）はプレスブルガー（Presburger）算術と呼ばれている．その上の閉論理式（以下，プレスブルガー文，または P 文と呼ぶ）の真偽は決定可能であり [1]，これまでに，Cooper のアルゴリズム [2] などの真偽決定手続きが知られている．

今回，Cooper のアルゴリズムを，組合せ論理回路の検証用に工夫して実装した（詳しくは 4 で述べる）．なお，論理式を簡潔に分かり易く書くことができるよう，その判定ルーチンが，論理型の変数（値として *true* または *false* を取る），if 関数，整数変数の定数倍 *，→（$b1 \rightarrow b2$ は $\neg b1 \vee b2$ を表す），論理式の等価性の比較演算（=など．$b1 = b2$ は $\neg (b1 \oplus b2)$ を表す）を扱えるようにした．

Examples of Correctness Proof of the Implementation of Combinatorial Logic Circuits Using Presburger Arithmetic Decision Procedure

Sumio MORIOKA, Junji KITAMICHI,
Teruo HIGASHINO and Kenichi TANIGUCHI
Deperment of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University
Toyonaka-shi, Osaka 560 Japan

3 回路の仕様と実現の記述

組合せ論理回路の実現の正しさの証明では，回路の仕様と実現をそれぞれ整数上の論理式で記述し，実現が仕様を満たすことを表す P 文を作成して，その P 文が真であることをプレスブルガー文真偽判定ルーチンにより示す．

回路の仕様には，整数変数（論理変数も使用できる）を使用し，回路の入出力間の関係を書く．表 1に，TTL の 2 ビットアダー 7482 [3] の仕様の記述例を示す．回路の入力 Ai と Bi の和が，Yi に出てくることが書かれている（以下，変数名の末尾が i のものは整数変数，b のものは論理変数である）．C0i は回路のキャリー入力，C2i はキャリー出力である．

表 1: 2 ビットアダー 7482 の仕様の記述

```
(0 ≦ Ai ≦ 3 ∧ 0 ≦ Bi ≦ 3 ∧ 0 ≦ C0i ≦ 1)
→ (
    if (Ai + Bi + C0i ≦ 3)
    then (
          Yi  = Ai + Bi + C0i
       ∧ C2i = 0
    ) else (
          Yi  = Ai + Bi + C0i - 4
       ∧ C2i = 1
    )
)
```

回路の実現には，(仕様とは別の）論理変数を使用し，回路の各論理ゲートの接続関係（結線）を，論理演算を用いて記述する．回路の入力から出力に至る途中のゲートの出力値も，それを表す論理変数を導入することで参照できる．表 2 に 7482 の実現の記述例を示す．

表 2: 2 ビットアダー 7482 の実現の記述

```
   Y1b = ((C0b ∧ nd1b) ∨ (A1b ∧ gate1b)
          ∨ (B1b ∧ gate1b) ∨ (C0b ∧ A1b ∧ B1b))
∧ Y2b = (¬ ((gate1b ∧ C2b) ∨ (¬ A2b ∧ C2b)
          ∨ (¬ B2b ∧ C2b)
          ∨ (gate1b ∧ ¬ A2b ∧ ¬ B2b)))
∧ C2b = (¬ ((gate1b ∧ ¬ A2b) ∨ (gate1b ∧ ¬ B2b)
          ∨ (¬ A2b ∧ ¬ B2b)))
∧ gate1b = (¬ ((C0b ∧ A1b) ∨ (C0b ∧ B1b)
          ∨ (A1b ∧ B1b)))
```

4 回路の実現の正しさの証明

仕様と実現の記述より，実現が仕様を満たすことを表す P 文を構成する．その P 文は次のような形をしている．

∨各変数 (仕様中の変数と実現中の変数の対応
　→ (実現の記述 → 仕様の記述))

「仕様中の変数と実現中の変数の対応」は，仕様の記述で使用した各 (整数) 変数の値が，実現の記述中の

(論理) 変数の値によってどのように決まるか，ということを表す論理式のことである．

表 3に 7482 の実現の正しさを検証するための P 文を示す．この P 文の真偽をプレスブルガー文真偽判定アルゴリズムで判定して，結果が真となれば，実現は正しく行われている．

表 3: 2 ビットアダー 7482 の実現の正しさを証明するためのプレスブルガー文

```
∀ Ai ∀ Bi ∀ C0i ∀ C2i ∀ Yi ∀ A1b ∀ A2b ∀ B1b
∀ B2b ∀ C0b ∀ Y1b ∀ Y2b ∀ C2b ∀ gate1b
(
    (  仕様中の変数の値と 実現中の変数の値の 対応
       (Ai = (if A2b then 2 else 0)
             + (if A1b then 1 else 0))
     ∧ (Bi = (if B2b then 2 else 0)
             + (if B1b then 1 else 0))
     ∧ (Yi = (if Y2b then 2 else 0)
             + (if Y1b then 1 else 0))
     ∧ (C0i = (if C0b then 1 else 0))
     ∧ (C2i = (if C2b then 1 else 0))
    )
    → (
        回路の実現の記述
        →   回路の仕様の記述
    )
)
```

上の P 文は全ての変数が∀で束縛された冠頭標準形（以下，**∀冠頭標準形 P 文**と呼ぶ）になっている．

Cooper のアルゴリズムでは，判定する P 文があるとき，それと等価で変数の数が 1 つ少ない P 文を構成する操作（**変数消去**）を繰り返し，最終的に変数を含まない（整数定数だけからなる）P 文を得て真偽を決定する．∀冠頭標準形 P 文の真偽を Cooper のアルゴリズムで判定する場合，**変数の消去順**（どの変数から消去していくか）**により判定にかかる時間が大幅に変わる**ことが経験的に分かっている．今回，判定を高速に行えるようにするため，以下の規則に従って変数の消去順を決定するプレスブルガー文真偽判定ルーチンを作成した．

[**規則 1**] 回路の実現の記述中の変数を，仕様中の変数より先に消去する．

[**規則 2**] 実現中の変数は，実現の論理式の真偽値の決定に効きやすいもの（現在は，論理式の構文木を考えたとき，その根に近い変数としている）から消去する．

[**規則 3**] 変数消去の途中で P 文が$\exists \ldots \exists x \exists \ldots (x = exp \wedge F(x))$（$x$ は変数，exp は x を含まない式）の形になった場合（判定ルーチンは，∀冠頭標準形 P 文の判定を，∃冠頭標準形 P 文の判定に帰着して行っている），$F(exp)$ を求めることにより x を消去する．

判定ルーチンは以下を繰り返して真偽判定を行う．

(1) まず [規則 1] より，実現の式を真にするような（実現中の）論理変数の値の組を 1 つ求める．1 組は実現の真理値表（その表では，変数は don't care も値として取ることができるとする）の 1 行分に相当する．

(2) (1) で求めた組に対応する仕様の変数の値の組を，対応関係の式から求める．対応関係の式中の実現の (論理) 変数に真偽値を代入して，論理式を $false \wedge exp = false$ などの規則により簡単化すると，対応する仕様の変数の値は，変数 = 定数という式の形で自動的に求まる．

(3) 求まった値の組を仕様の各変数に代入し，仕様の式が全て真になるかどうかを調べる．各代入は [規則 3] により自動的に行われる．

以上を，(1) で求まる実現の変数値の組全てについて繰り返す．このとき [規則 2] は，(1) で求める実現の変数値のある 1 組に対しては，そのうちの don't care をなるべく多くするように働く．これにより (1) で求まる変数値の組の総数は減少すると思われる．

5 証明例

3, 4 で述べた方法で，TTL の 2 ビットアダー 7482, 4 ビットコンパレータ 7485, 4 ビットセレクタ 74157, 4 ビット ALU 74382 の検証を行ってみた．表 4に，各 P 文の大きさ（式中の演算子や変数の総数），真であることの判定に要した時間，判定に要した主記憶の量を示す（Sony NEWS−5000 使用．速度は約 100MIPS, 主記憶は 64M バイト）．74382 については，ALU の各機能（加減算や論理演算）毎に別々の P 文で証明した結果も示す．いずれの場合も真であることを証明する P 文は大きくなったが，少ないメモリで高速に判定を行うことができた．

また，4 で述べたような変数消去順の工夫を行わない（ランダムな順番で変数を消去する）判定ルーチンでは，例えば，74382 の証明に十数時間を要した．

なお，実現に誤りがある場合は，表に示した時間よりも高速に（誤りの内容にもよるが，誤りのない場合と比べて，大体数百分の一程度の時間で）判定できる．

表 4: 各回路の証明で判定した P 文の大きさと判定時間 (Sony NEWS−5000)

回路	P 文の大きさ	CPU 時間 (秒)	主記憶使用量 (K バイト)
7482	519	0.6	50
7485	1155	58	186
74157	583	20	94
74382 全機能	2935	238.7	801
74382 (加算のみ)	2075	12.8	536
74382 (減算のみ)	2343	28.5	617
74382 (and のみ)	1891	4.8	422
74382 (or のみ)	1891	2.2	406
74382 (xor のみ)	1955	7	434
74382 (clear のみ)	1811	0.7	348
74382 (preset のみ)	1811	0.5	315

6 あとがき

本稿では，回路の実現が仕様を満たすことをプレスブルガー文で記述し，その論理式が恒真であることをプレスブルガー文真偽判定アルゴリズムを用いて判定した例について述べた．

参考文献

[1] J.E.Hopcroft, and J.D.Ullmann: Introduction to Automata Theory, Languages and Computation, Addison-Wesley,1979（邦訳：野崎昭弘他:オートマトン言語理論 計算論 I,II, サイエンス社, 1986）．

[2] D.C.Cooper: Theorem Proving in Arithmetic without Multiplication, Machine Intelligence, No.7(1972),pp.91-99.

[3] '94 74 シリーズ IC 規格表, CQ 出版社.