

## 時間記号論理シミュレータの高速化と性能評価

3U-3

高橋瑞樹 石浦菜岐佐 矢島脩三  
(京都大学工学部)

### 1. はじめに

論理回路の設計支援ツールとしては、従来から論理シミュレーションが広く用いられており、いまや論理設計支援システムには欠くことのできないものとなっている<sup>[1]</sup>。設計検証の中でもタイミングに関する検証は複雑で困難な場合が多い。特に非同期回路として設計されるものに対しては微妙なタイミングが問題になり、素子遅延のばらつきまで考慮する必要がある。論理シミュレーションでは、このような遅延のばらつきを最大/最小遅延シミュレーションにより解析するが、現実よりも悲観的な結果しか得られないことが知られている。我々は、これらの問題を解決するものとして時間記号論理シミュレーションを提案している。時間記号論理シミュレーションは、素子の遅延時間や入力の変化する時刻を、変数を含む式で表現して遅延のばらつきをモデル化し、精密なタイミング解析を行おうとするものである。本稿では、従来の時間記号論理シミュレータで用いていた線形計画法のルーチンの改良による高速化とその評価、非同期順序回路への応用について述べる。

### 2. 時間記号論理シミュレーションの概要<sup>[2]</sup>

時間記号論理シミュレーションでは、ゲートの遅延時間、入力の変化時刻を変数で表わしシミュレーションを行う。これを、時間変数と呼ぶ。各時間変数には、その取り得る値の範囲があり、それを変数制限条件という。信号線の信号値系列は、イベントまたは代数式でラベル付けされた節点からなるイベント木で表わされる(図1)。シミュレーションは、回路内の各ゲートについて、入力イベント木から出力のイベント木を計算することにより行われる。

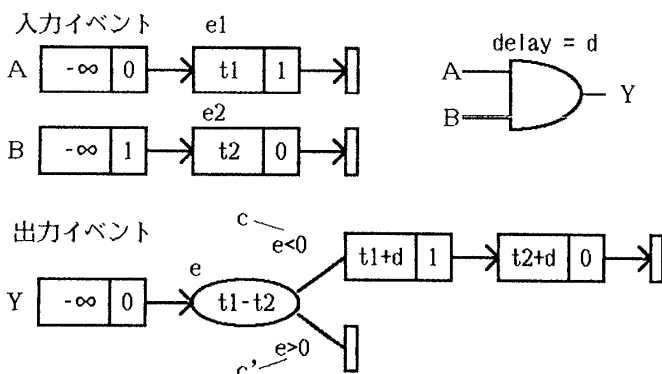


図1 イベント木の構造

イベントの発生時刻が定数ではなく変数であることに起因して、入力イベントの前後関係によって場合分けが行われ、出力のイベント木上に分岐が生ずる。この際、変数制限条件とそこまでのパス条件(その時点までの分岐条件の集合)より、注目している2つのイベントの前後関係が一意に決まり分岐が生じない場合がある。時間記号論理シミュレータでは、変数制限条件、パス条件、分岐条件を時間変数に関する連立一次不等式とみなし、分岐の可能性の判定を、線形計画法を用いた連立一次不等式の解の存在判定により行う。

前に作成したシミュレータのプロトタイプでは、シミュレーション時間の約90%が線形計画法のルーチンに費やされており、適用回路規模の面においてもボトルネックとなっていた。本稿では、高速化のための線形計画法のルーチンの改良について述べる。

### 3. シミュレーションの高速化

時間記号論理シミュレーションにおける連立一次不等式の解の判定問題には次に示す2つの特徴がある。

- (1) 変数制限条件とパス条件の集合に分岐条件が1つずつ追加されながら判定が行われていく。
- (2) 1つの分岐の可能性判定に、'変数制限条件+パス条件+分岐条件'と、'変数制限条件+パス条件+分岐条件の否定'の2つの解の存在判定が必要である。

(1)については、従来のシミュレータでは、毎回全ての条件を不等式の形で与えて、初めから計算を行っていた。しかし、与えたn個の条件のうちn-1個は前段階の判定に用いた条件と同じであり、前段階における計算結果であるタブローを再利用すれば無駄なピボット変換を避けることができる。条件の追加は、前段階におけるタブローとピボット変換の情報を用いて、追加条件を基底形式に変換しタブローに加えて解の存在判定をすることで行える。

(2)は、変数制限条件の集合をVC、パス条件をPC、分岐条件をc、その否定をc'としたとき、実際に分岐が起こるのは、 $VC+PC+\{c\}$ と $VC+PC+\{c'\}$ の両方が可能解を持つ場合であり、従来のシミュレータではその両方について線形計画法のルーチンを適用していた。しかし、この2つの判定は前段階のタブローが同じであるから、c'を追加する際の基底形式への変換は、cを基底形式に変換した式の係数の符号を変えるだけで行えるので、一度に、c、c'の両方に対する判定を行ったほうが効率がよい。

上記2つの特徴を考慮して、双対単体法<sup>[3]</sup>を用いた線形計画法のルーチンの改良版を作成

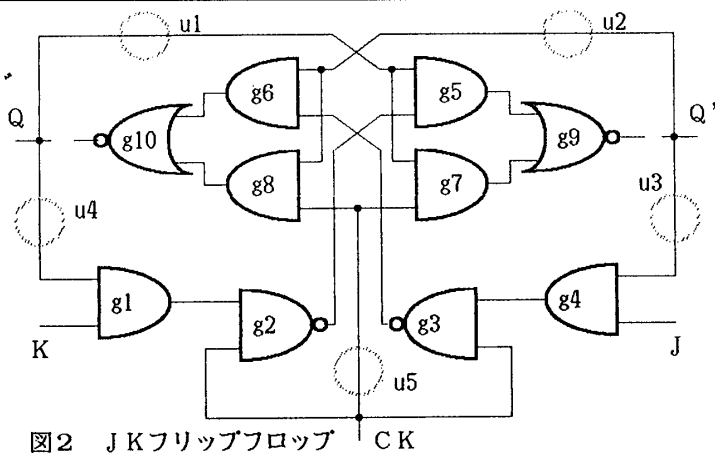


図2 JKフリップフロップ CK

| J | K | Q | Q' | CK    |
|---|---|---|----|-------|
| 0 | 0 | — | —  | ①↑,②↓ |
| 0 | 1 | — | —  | ③↑,④↓ |
| 0 | 1 | ⌋ | ⌋  | ⑤↑    |
| 0 | 1 | — | —  | ⑥↓    |
| 1 | 1 | ⌋ | ⌋  | ⑦↑    |
| 1 | 1 | — | —  | ⑧↓    |

図3 入力パターン

し実験を行った。実験結果については次章で述べる。

4. JK-FFの検証への応用と性能評価

時間記号論理シミュレータの応用として、図2に示したJK-フリップフロップの検証の例を示す。

時間記号論理シミュレータは組合せ回路を対象としているが、非同期順序回路についても次に示す手続きによって、その組合せ回路部分のシミュレーションにより検証を行うことができる<sup>[4]</sup>。

Step 1 状態変数に対応するフィードバックループを切断することにより組合せ回路を得る。

Step 2 状態遷移表から切断された信号線上に発生するイベントの系列を得る。

Step 3 外部入力とStep 2で得たイベントの系列を与えて、時間記号シミュレーションを行う。

Step 4 回路の外部出力と切断点について、Step 3で得られたシミュレーション結果と期待値とを比較し正常動作しているかどうかを調べる。

上記の手続きを用いて、図2に示すJK-FFの検証を行った。外部入力は、JK=FFの各動作(状態遷移)に対応させ、また回路の対称性を考慮して全部で8通りのパターン(図3)を用い、そのそれぞれについてシミュレーションを行い動作を確認した。はじめに、図2においてu1~u4の4箇所遅延を設定しシミュレーションを行った結果、出力として正常動作しているものは得られなかった。次に、u5にも遅延を設定し同様にシミュレーションを行ったところ、ポジティブエッジ型のJK-FFとして正常に動作している場合が得られた。これにより、u5に遅延を入れることにより図2のJK-FFが正常に動作するということを確認することができた。

表1にu1~u5の5箇所遅延を設定した場合の各入力パターンに対するシミュレーションのCPU時間(SONY NWS-830上)を、旧版、改良版の両バージョンについて示す。時間変数の数は17個である。表中のイベント数とは回路中に発生したイベントのうち初期イベントと終了イベントを除いたものの数を示す。また、入力パターン⑤+⑦は⑤と⑦の2つの動作を続けた場合の入力パターンを示す。このときの時間変数の数は19個である。表1の結果より、線形計画法ルーチンの改良により約20倍の高速化の実現が確認された。

5. まとめ

時間記号シミュレーションにおいて最も時間のかかる線形計画法のルーチンを、応用の特殊性を考慮して改良することによりシミュレーションの高速化を図ることができた。今後は、出力結果の照合系について研究していきたい。

《謝辞》 御討論頂いた本学平石裕実助教授、高木直史博士および矢島研究室の諸氏に感謝致します。

<参考文献>

- [1] 樹下行三, 浅田邦博, 唐津修: VLSIの設計II, 岩波講座マイクロエレクトロニクス, 岩波書店, (1985).
- [2] 高橋瑞樹, 石浦菜岐佐, 矢島脩三: 時間記号論理シミュレータについて, 第36回情処全大3x-2, pp.1923~1924(Mar. 1988).
- [3] 伊理正夫: 線形計画法, 共立出版(1986).
- [4] 木村晋二, 羽根田博正: 系列集合論理シミュレーション手法に基づく非同期式順序回路の検証, 信学技報VL087-118, pp.15~22(Feb. 1988).

| 入力パターン | イベント数 | CPU時間(sec.) |      |
|--------|-------|-------------|------|
|        |       | 旧版          | 改良版  |
| ①      | 2     | 0.1         | 0.1  |
| ②      | 2     | 0.1         | 0.1  |
| ③      | 2     | 0.1         | 0.1  |
| ④      | 2     | 0.1         | 0.1  |
| ⑤      | 5 4   | 15.2        | 0.6  |
| ⑥      | 3 1   | 6.6         | 0.4  |
| ⑦      | 6 3   | 18.4        | 0.7  |
| ⑧      | 3 2   | 6.9         | 0.4  |
| ⑤+⑦    | 1239  | 352         | 13.2 |

表1 実行時間