

機能設計支援システムOZにおける 2K-6 論理検証手法

中井 匡 飛永 聡 田中 英俊
NECソフトウェア北陸(株)

1. はじめに

コンピュータ設計の初期段階である機能設計において、回路の特性を考慮し様々な形式の回路表現を用いた設計図面が作成される。つまり、制御系回路はフローチャートや状態遷移図で表現し、データバス系回路はブロック図で表現された設計図面が、作成されることとなる。

本稿では、それらの設計図面を入力とし、直接論理シミュレーションを行いその結果を設計図面に表示するOZシミュレータについて報告する。

2. 機能設計における論理検証

コンピュータ設計においては、論理合成ツールの実用化等、設計の自動化が進められている。しかし、試行錯誤の多い機能設計では、論理変更とそれに伴う検証が頻繁に行われ、膨大な設計期間が必要となっている。この設計期間を短縮するために、機能設計手段であるブロック図やフローチャートでの論理検証が必要となる。

そこで本シミュレータは、機能設計段階の論理検証として、以下の機能を実現した。

- ・ブロック図/フローチャートの直接シミュレーション
- ・ブロック図/フローチャートの混在シミュレーション
- ・図面上での動作確認

特に、ブロック図/フローチャートを直接シミュレーションすることでの、設計TAT向上の実現や、回路動作の図面表示による設計誤り検出の容易化が、本シミュレータの目的である。

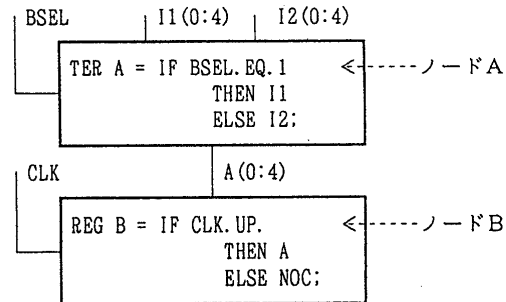
3. ブロック図シミュレーション

(1) シミュレーション方式

ブロック図内の個々のシンボルに記述されたFDL(Functional Description Language)の1文を1つのノードとして考えたイベントドリブンのシミュレーションを行う。また、各シンボルに存在するノードは、シンボル間に接続するネットと呼ばれる信号線によって接続関係が表現されており、その接続関係に従いノードの状態値が伝搬する。

図1のブロック図を説明する。例えば入力信号I1の状態値が変化したとすると、ノードAの演算が行われる。結果として、ノードAの状態値が変化したとすると、イベントを出力先である下段のシンボル内のノードBに伝搬する。

そして、それを受けてノードBの演算が行われる。



—— はネットを示す
図. 1 ブロック図

(2) ブロック図への結果表示

本シミュレータは、タイムチャート表示は無論の事、その実行結果をブロック図上に表示する事が可能である。

表示のボタンは以下の3通りがあり、これらのボタンを任意に選択し結果解析を行うことで、検証効率が向上する。

- 1) シミュレーション停止時
シミュレーションが停止した状態での信号線の状態値を表示する。
- 2) 条件成立時
シミュレーション実行中に、あるノードの状態値がある値になった時の、信号線の状態値を表示する。
- 3) 長時間シミュレーション後
長時間シミュレーション時に信号線の状態値を保持しておき、シミュレーション終了後に指定した時刻における信号線の状態値を表示する。

図2に表示例を示す。但し、図中の斜体文字は表示した状態値を表現する。

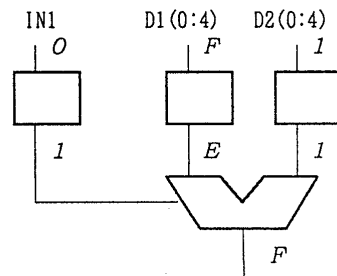


図2. ブロック図状態値表示

4. フローチャートシミュレーション

(1) シミュレーション方式

フローチャート内の各シンボルをノードとして考えてモデルを作成する。

そして、これらのノードを用いてシミュレーションすることで、イベントドリブンシミュレーションを可能とした。シミュレーションは基本的にはブロック図の場合と同様に行うが、判断シンボルと状態シンボルに関しては下記の特殊処理を行う。

1) 判断シンボル

判断シンボルの演算を行い、保持しておいたFANOUT情報の条件値と一致するFANOUT先のシンボルに、FANOUTする。

2) 状態シンボル

保持しておいたクロック信号のトリガが有るまでイベントを止めておき、トリガがかかったら次のシンボルにFANOUTする。

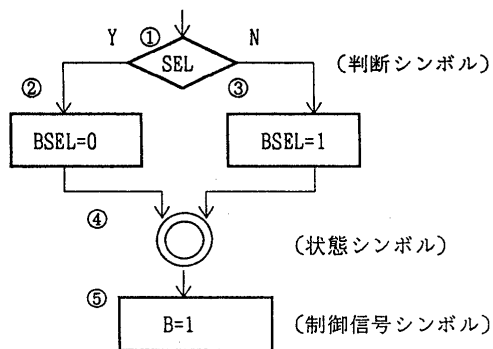


図3. フローチャート

図3のフローチャートを用い説明する。まず、シンボル①にFANOUTしたとする。そこでシンボル①の演算を行い真であればシンボル②にFANOUTするし、偽であればシンボル③にFANOUTする。例えばシンボル③にFANOUTしたとすると今度はシンボル③の演算を行い信号BSELの状態値を1にして次のシンボル④にFANOUTする。シンボル④は状態シンボルであるので、クロック信号のトリガを待ち、トリガがかかったら演算を行い次のシンボルにFANOUTする。

(2) フローチャート結果表示

本シミュレータは、タイムチャート表示は無論の事、その実行結果をフローチャート上に表示する事が可能である。

表示のパタンは以下の2通りがあり、これらのパタンを任意に選択し結果解析を行うことで、検証効率が向上する。

1) シミュレーション停止時

シミュレーションが停止した状態で、どのシンボルの実行が行われたのか、またどのシンボルからFANOUTしてきたのか、またどこにFANOUTしようとしているのかが判別できるように色付けをシンボル及びネットに行う。

2) 長時間シミュレーション後

長時間シミュレーション時にどのようにフローチャートが動いたかを保持しておき、シミュレーション終了後にステップ実行でシミュレーション停止時と同様の表示を行う。

図4に表示例を示す。但し、図中では色付けの代わりに下記の線種を用いている。

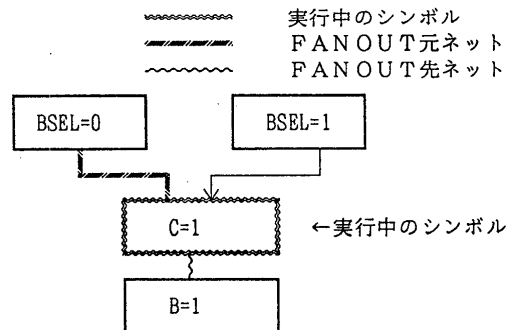


図4. フローチャート実行表示

5. 異種図面混在シミュレーション

(1) シミュレーション方式

制御回路をフローチャートで記述し、データバス系をブロック図で記述した場合、両者をリンクした状態でのシミュレーションを行う。シミュレーション時には、基本的にはフローチャートシンボルから作成されたノードも、ブロック図から作成されたノードも、同一処理が可能となっている。

但し、フローチャートのシンボルから作成されたものの内、判断シンボルと状態シンボルに関しては3章で説明した通りに特殊処理を行っている。また、ブロック図／フローチャート間の接続については、フローチャートの入力において入出力信号の指定を行っておき、それらの信号とブロック図上の入出力信号を接続し、状態値の受け渡しを行っている。

6. おわりに

本稿では、機能設計支援におけるブロック図・フローチャートのシミュレーションについて報告を行った。このように、従来のハードウェア記述言語による設計と検証方法から一歩進めて、機能設計に対する新しい設計と検証方法を提案した。現在、状態遷移図シミュレータを開発中であり、今後さらに設計者に理解し易いシステムに発展させる予定である。