

統合論理設計支援システム ILOS における

3S-7

論理検証ツール EVERY 4

飛永聡 山際肇 菅波和幸

田中英俊 河合正人

北陸日本電気ソフトウェア(株)

日本電気(株)

1. はじめに

最近の論理装置は大規模化され複雑さを増してきている。この種の論理装置の開発を短期間で行うためには、開発の初期段階において、少しでも多くの設計ミスを取り除く必要があり、そのために高性能で使い勝手の良い論理検証システムが不可欠である。

LSI化された論理装置を開発する場合、装置設計者が最も期待する論理検証方法は、装置全体のシミュレーションモデルを作成し、装置としての動作を試験する方法である。しかし、この検証方法は多大なコンピュータリソースと長大な検証期間を必要とする。従って、設計の初期段階でこの方法を使用することは、逆に設計効率の低下をまねく恐れがある。

このため設計したばかりの回路を、即座に論理検証できる設計環境が必要となる。しかし、回路の一部分だけで論理検証を行おうとすると、回路の切り口でのテストパターンを準備する必要があり、このための十分なテストパターンを作成することは、装置設計者に大きな負担を与える。またシミュレーションの結果を目視チェックするというわずらわしさも伴う。

このような状況に対応するために、筆者らはLSIを効率良く論理検証するシステムを、既にメインフレーム上に構築し、実用に供しているが、今回、更にこのシステムの運用性を向上させ、統合的な論理設計支援環境を実現するために当社のEWS4800シリーズ上に論理検証システムEVERY4を開発した。

2. EVERY4の概要

EVERY4は、機能レベル[1]で記述された回路と論理接続レベルで記述された回路の論理的等価性を検証する2階層自動論理検証システム[2]である。

図1にEVERY4の概要を示す。

EVERY4では、まず、論理検証を容易にするために機能レベルと論理接続レベルの両方の回路を組合せ回路に分割する。

次に、組合せ回路化された機能レベルの回路よりテストパターンを自動生成し、そのテストパターンを使用して機能レベルシミュレータ、論理接続レベルシミュレータで両レベルのシミュレーションを行う。最後に、両レベルのシミュレーション結果を比較して回路の論理的等価性を検証する。

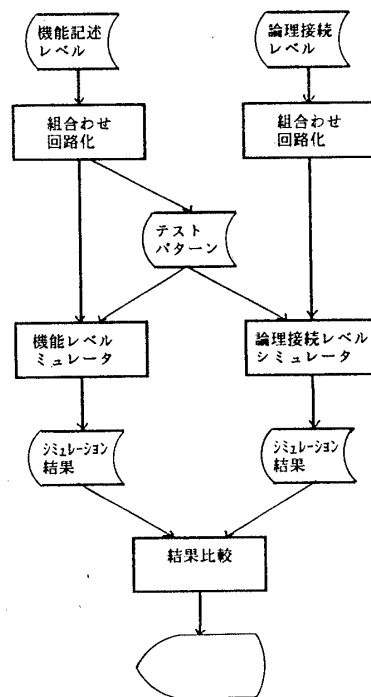


図1 EVERY4の概要

3. EVERY4の機能

3.1 組合せ回路化

組合せ回路化は、機能レベルの回路と論理接続レベルの回路の両方に対して行われる。

組合せ回路化では、回路中のレジスタの出力信号を切断して出力ピンに出す。そして、入力ピンを新たに設定してレジスタのファンアウト先の論理素子に接続する。このことにより、論理回路は入力ピンと、出力ピンと、前段にレジスタがある出力ピンとに囲まれた複数の組合せ回路に分割される。図2参照。

組合せ回路化により、テストパターン発生を容易化することができ、設計者による論理エラーの解析を容易にすることもできる。

Logic Verification Tool EVERY4 In Integrated Logic Design Support System ILOS

Satoshi TOBINAGA\*, Hajime YAMAGIWA\*, Kazuyuki SUGANAMI\*, Hidetoshi TANAKA\*\*, Masato KAWAI\*\*

\* NEC Software Hokuriku, \*\* NEC Corporation

```

INPUT A(0:4), B(0:4),...
OUTPUT C(0:4),...
.....
REG DFF(0:4) = IF CLK.UP. THEN A
                ELSE B;
C(0:4) = DFF(0:4);
.....

```

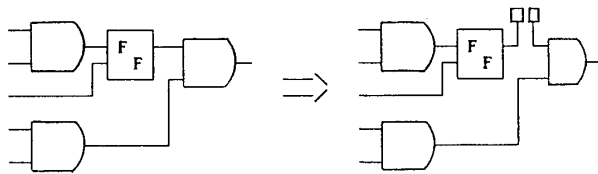


```

INPUT A(0:4), B(0:4),... ;
INPUT DFF(0:4);
OUTPUT C(0:4),...;
OUTPUT XXX(0:4);
.....
REG XXX(0:4) = IF CLK.UP. THEN A
                ELSE B;
C(0:4) = DFF(0:4);
.....

```

機能レベル回路の組合せ回路化



論理接続回路の組合せ回路化

図2 組合せ回路化

### 3.2 テストパターン自動生成

テストパターン自動生成では、機能レベルの回路記述を解析して、この記述に表されている機能がすべて網羅されるようなパターンを発生する[3]。これは、組合せ回路に分割された回路の入力から出力へ至るすべてのパスを活性化することによって行う。

パターンの発生手順としては、ひとつの出力ピンから入力ピンに至るまでのすべてのパスをたどり、たどったパスがすべて活性化されるようなパターンを発生する。これをすべての出力ピンについて行う。

このようにして網羅率の高いテストパターンを自動生成している。

### 3.3 シミュレーション・結果解析

自動生成されたテストパターンを使用して、機能レベルシミュレータで機能レベルの回路のシミュレーションを実行する。機能レベルと論理接続レベルの回路の入力ピンの対応を自動的に取り、同一のテストパターンを論理接続レベルの入力ピンに設定して、論理接続レベルシミュレータでシミュレーションを実行する。論理接続レベルのシミュレーションは、回路規模の増加に伴い処理時間が増大するため、専用のシミュレーションエンジンを付加してシミュレーション時間を短縮している。

ここでのシミュレーションは、入力パターンを1パターン分入力ピンに設定して、1サイクル分のクロックを印加した後、出力ピンの値が確定するまで行う。

シミュレーション後、機能レベルと論理接続レベルの回路の出力ピンの対応を自動的に取り、両レベルのシミュレーション結果を毎パターンごと比較する。シミュレーション結果が異なる場合、どの出力ピンでエラーが検出されたかを表示する。

全ての回路は組合せ回路に分割されているので、エラーの原因は、エラーのあった出力ピンからファンイン方向に回路をトレースしていった場合の、入力ピンに到達するまでの範囲の回路にあるとして求めることができる。

## 4. 評価

本システムは設計初期の論理検証のほか、FUSION（論理合成システム）が合成した回路に人手修正が入った場合に、論理的にエラーが混入していないかということを検証するためにも用いられている。

当社のメインフレーム上で動作する同様の機能を持ったシステムと実行時間のTATの比較を行った結果、被検証回路は約2KGで、テストパターンは1000パターンの場合、約4分の1のTATで検証を終了した。

## 5. おわりに

以上述べた様に今回開発した論理検証システムEVERY4は、設計者が論理回路の機能レベル記述と論理接続レベルの記述を用意するだけで、簡単に論理検証ができることにより、設計初期段階において、大幅な設計期間の短縮を可能としている。

また、個々のLSIの初期設計品質が向上するため、メインフレームによるシステムレベルのシミュレーションが、効率良く行えるという効果が出ている。

今後、処理速度、機能、ユーザーインターフェース等に関して、より一層の向上を目指し開発を進める予定である。

## 参考文献

- [1] 佐々木他,「MIXSの機能記述言語(FDL)」情報処理学会第21回全国大会
- [2] 田中, 野水他,「2階層論理検証」, 情報処理学会第29回全国大会
- [3] M.Kawai et al.,「A High Level Test Pattern Generation Algorithm」, ITC(1983)