

汎用エンジンRM-Iの論理診断への応用

5L-8

菅沼直昭† 中田暁†† 富田昌宏† 平野浩太郎†

†神戸大学 ††(株)日立製作所

1. はじめに

複数の用途に適用可能な汎用エンジン・RM-I (Reconfigurable Machine-I) を開発し、論理診断に応用した結果を報告する。RM-Iは、5つのFPGA^[1]と384Kbyteのメモリから構成され、比較的小規模の問題を対象とする。論理診断手法の1つである拡張X-伝搬法^[2]にRM-Iを適用した。FPGA間の配線資源とメモリの利用効率、RM-Iの並列度・速度向上の鍵となる。まず、ソフトウェアによる処理方式の評価を行い、回路のFPGAへの分割とデータのメモリへの割り付けを人手で決定した。実験の結果、拡張X-伝搬法の複数の処理工程において、ソフトウェアの約200倍の高速化を達成した。

2. 論理診断エンジン

論理診断とは、ゲートレベル回路において論理設計誤りを特定・修正する問題をいう。我々は、単一出力の組み合わせ回路を対象とし、多重の誤りに対応可能な拡張X-伝搬法の一部をRM-I上に構築した。以下にその概略を述べる。

2.1 拡張X-伝搬法

拡張X-伝搬法は、誤り追跡入力と呼ばれる入力パターンを用いて、論理設計誤りの特定を行う。誤り追跡入力は1つのブール変数Xまたは \bar{X} と定数からなる。理想回路に誤り追跡入力を与えた場合には、ブール変数Xが出力まで伝搬するが、誤りを含む回路では定数が出力される。拡張X-伝搬法は誤りを含む回路がXを出力するような修正方法を提示する。拡張X-伝搬法は、以下の処理からなる。

(1) 誤り可能性の指標EPI

誤りの種類を仮定しない段階で、誤っている可能性のない箇所を早期に削除するために誤り可能性の指標EPIを用いる。誤り可能性の指標EPIは、ある箇所の修正による出力への可制御性を表す。指標EPIを用いることによって、修正後の信号値を求めずに、意味のない箇所を削除することができる。

(2) 6値シミュレーション

信号値0, 1, X, \bar{X} に加えて、 X/\bar{X} となる可能性がない不定信号値としてD, X/\bar{X} となる可能性がある不定信号値としてEの6値を用いたシミュレーションを行い、さらに候補となりえない箇所を削除する。6値シミュレーションによって、誤り種類を考慮しない箇所の段階で候補を絞り込む。

(3) 誤り候補抽出

上記(1),(2)の処理によって削減された誤り箇所に誤り候補を対応させる。誤り箇所に考えられるすべての誤り種類を対応させ、その結果外部出力にXが出力されるかどうかを4値シミュレーションによって判断する。

以上の処理のうち、6値シミュレーションと4値シミュレーションをRM-I上に構築した。

2.2 論理診断エンジンの仕様

RM-I上に構築するエンジンの仕様を述べる。

- (1) 対象とする誤り種類: ゲート誤りとして、ゲート機能選択誤り、インバータ過剰/欠落誤りに対応する。また、信号線誤りとして、信号線過剰誤りに対応する。
- (2) 対象回路規模: 4Kゲートまでの回路を扱う。
- (3) 処理速度: 1Kゲート規模の回路に100個の誤り追跡入力を加えたときに、1分程度で処理を完了する。そのために、30万イベント/秒を達成する。
- (4) 信号値: 0, 1, X, \bar{X} , D, E, Iの7値を用いる。誤り追跡入力を用いるため、0, 1の信号値を用いる他にX, \bar{X} を用いる。さらに、6値シミュレーションを行うためにD, Eの信号値を加える。また、各ゲートの入出力端子および外部入出力端子の信号値の初期値としてI (Initial) を用いる。
- (5) 処理方式: 外部入力端子からのランク順素子評価に基づくイベント駆動方式とする。

2.3 論理診断エンジンの構成

次に、論理診断エンジンの構成について述べる。図1に構成を示す。論理診断エンジンは、ホストコンピュータ・インタフェース部および処理ユニットから構成される。それぞれの構成について述べる。

- (1) ホストコンピュータ・インタフェース部: エンジン実行に先だつエンジンの初期化、メモリへのデータの設定および終了後のデータ読みだしを行う。

- (2) 処理ユニット: 処理ユニットは、汎用エンジン上の実行モジュールに構築され、6値シミュレーションと4値シミュレーションを行う。

処理ユニットは、スケジューラ、プロパゲータ、演算・比較器、制御部の4つのサブユニットから構成される。スケジューラはイベント登録関連の処理を行い、プロパゲータはイベント伝搬に関する処理を行う。それぞれの構成について以下に述べる。

- i) スケジューラ: スケジューラは、通常素子、誤り候補(箇所)、追跡入力パターンのイベント登録を行う。また、誤り候補(箇所)のイベント再登録を行う。また、ランク順に素子評価を行うため、評価されるべきランクのイベントを取り出す。

- ii) プロパゲータ: プロパゲータは、演算・比較器から送られ

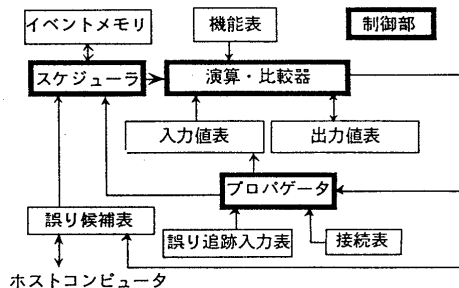


図1 論理診断エンジンの構成

RM-I and Its Application to Logic Diagnosis
Naoki SUGANUMA†, Satoru NAKATA††,
Masahiro TOMITA†, and Kotaro IHIRANO†
†Kobe University, ††HITACHI, Ltd.

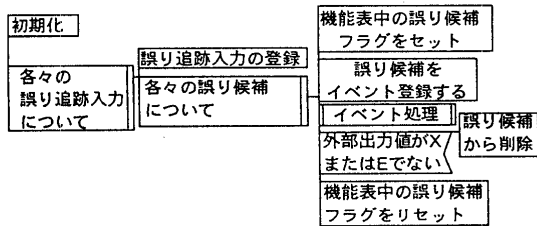
る素子出力値変化を、接続される他の素子に伝搬するとともに、スケジューラにイベントとして送る。

iii) 演算・比較器：演算器は、スケジューラから送られるイベント情報にもとづき、素子の出力値評価を行う。また、比較器は、演算器での出力値評価後、以前の出力値と比較してイベント発生の有無を判定する。

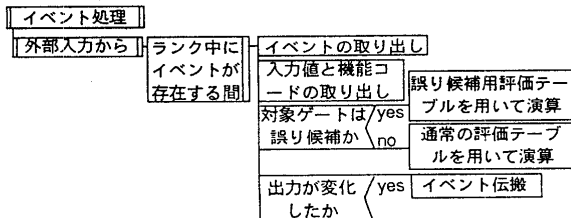
iv) 制御部：エンジン全体の実行を制御する。

次に、処理ユニットの動作を図2に示す。

- <Step 1> 初期化: シミュレーション実行に先立ち、回路データ、誤り候補(箇所)データ、誤り追跡入力パターンロードと、イベントメモリおよび信号値表の初期化を行う。
- <Step 2> 誤り追跡入力イベント登録: 誤り追跡入力をイベント登録するとともに、外部入力端子の信号値として設定する。
- <Step 3> 誤り候補(箇所)のイベント登録: 誤り候補(箇所)をイベントとして登録し、誤り候補フラグをセットする。
- <Step 4> イベント取り出し: イベントメモリから評価すべき素子番号を取り出す。
- <Step 5> 素子入力値・機能コード取り出し: 素子評価を行う際に必要な入力値、機能コードを各メモリから取り出す。
- <Step 6> 出力値演算・イベント判別: 入力信号値と機能コードをもとに素子出力値演算を行い、以前の出力値と比較してイベントを判別する。
- <Step 7> 信号値伝搬・イベント登録: イベントが生じた場合は、その出力値を接続される他の素子の入力端子へ伝搬して信号値を更新するとともに、接続される素子をイベントとして登録する。
- <Step 8> 誤り候補箇所のイベント再登録: 次の誤り候補(箇所)に対するシミュレーションに先立ち、処理が終わった誤り候補について正常素子としてイベント再登録を行う。
- <Step 9> 外部出力値評価: 誤り候補(箇所)が修正の必要条件を満たすか否かを判定する。修正の必要条件を満たさなかった場合は、誤り候補(箇所)から削除され、以後のシミュレーション対象から除かれる。



(a) 誤り追跡処理



(b) イベント処理

図2 処理ユニットの動作

表1 実験結果

回路	誤りの多重度	ゲート数	6値シミュレーション 処理速度 (Kevent/sec)	4値シミュレーション 処理速度 (Kevent/sec)
Y1	1	47	550	660
Y1	2	47	430	210
Y1	3	47	420	170
Y2	2	82	340	330
Y3	2	94	340	270
Y4	2	141	320	110
C432	2	203	190	440

2.4 論理診断エンジンのRM-1上への割り付け

次の点を考慮して、回路のFPGAへの割り付けとデータのメモリへの割り付けを行った。

- (1) ハードウェア資源の分散: FPGA内での配置・配線効率を考慮して、それぞれのFPGAにできるだけ均等にハードウェア資源を割り付ける。
- (2) 並列化: 同じステートに動作できる箇所を増やし、並列度を向上させるために、同時に行われるプロセスは異なるFPGAに割り付ける。また、同時に参照される可能性のあるデータは異なるメモリに割り付ける。
- (3) FPGA間接続: 対象とする複数の工程について、FPGA内部機能は定義プログラムによって変更できるが、FPGA間接続は変更できない。FPGA間の接続資源を有効に利用することが不可欠である。

3. 実験

実験結果を表1に示す。実験に用いた回路は、ISCASベンチマーク回路¹⁾と我々が設計したシミュレーション・エンジンであるTASSE II⁴⁾から抽出した回路である。TASSE IIから抽出した回路については、設計時の誤りの履歴に基づき、これらを優先的に仮定した。ISCASベンチマーク回路では、ランダムに誤りを仮定した。誤りの探索は、回路によって多重度1, 2または3まで行った。なお、用いた誤り追跡入力の数100である。

6値および4値シミュレーションの処理速度について述べる。処理速度の平均値は6値シミュレーションでは370Kイベント/秒、4値シミュレーションでは310Kイベント/秒を達成した。ソフトウェアによる6値および4値シミュレーションの処理速度に対して、それぞれ170倍、190倍の処理速度を達成した。

前処理を含む全体の処理時間は、平均で33%短縮されており、高速化が達成されたといえる。また、CLBの使用率は最大45%、平均33%、IOBの使用率は最大70%、平均60%であった。

4. まとめ

汎用エンジンRM-1上に論理診断エンジンを構築し、論理診断への応用を行った結果について報告した。FPGA間の配線資源とメモリの有効利用が処理速度向上のための鍵となる。現在は、人手によって、アーキテクチャ分割を行っているが、これらの工程の自動化が今後の課題として挙げられる。

参考文献

[1] プログラマブル・ゲートアレイ データブック, ザイリンクス社 (1990).
 [2] 上田伸人, 菅沼直昭, 山本保, 富田昌宏, "多重誤りを対象とする自動追跡手法," 情報研報, Vol.91, No.110, 91-DA-60, pp.185-192 (1991).
 [3] F.Brglez, H.Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translation in FORTRAN," Special Session on ATPG and Fault Simulation, Proc. ISCAS'85, 1985.
 [4] N.Suganuma, M.Tomita, K.Hirano, "A Compact Simulation Engine with Flexible Logic Model Expansion," IEEE International Conf. on Systems Engineering, 1992.