

並列論理シミュレータ

概要

論理シミュレーションは、設計回路の論理機能、信号伝播タイミングを検証する LSI 設計の一工程であり、多大な処理時間を必要とする。本並列論理シミュレータは、対象回路の並列性を効率良く抽出する機構を採用し、PIM に代表される大規模 MIMD 型並列マシン上での高速シミュレーションを実現する。

特徴

タイムワープ機構

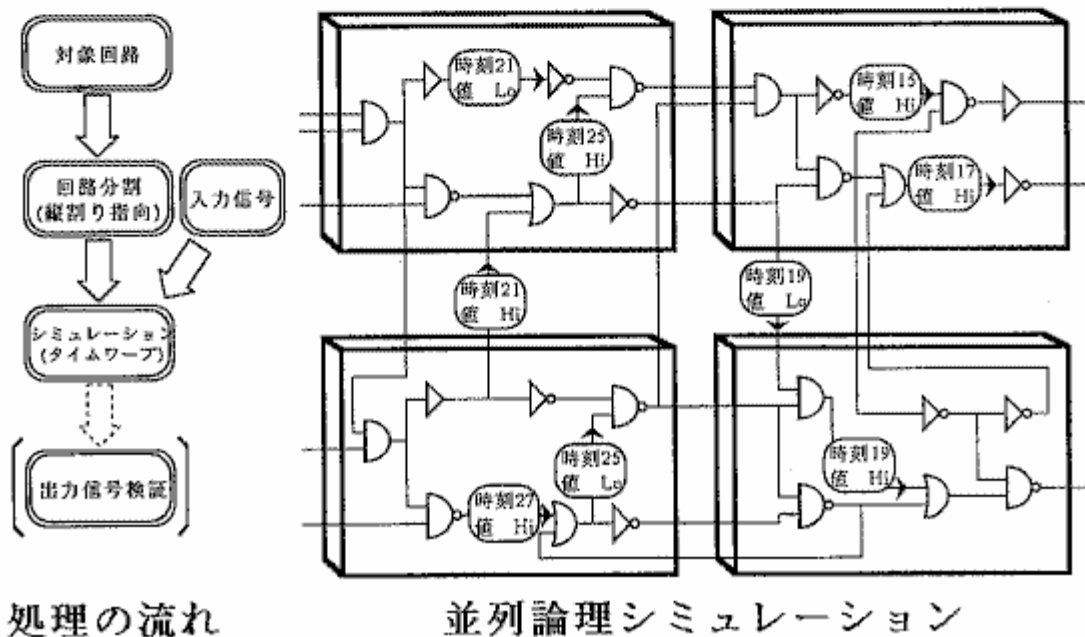
メッセージ (信号値変化情報を伝達) の評価順序を分散的に管理する。積極的な見込み処理により高い並列性を抽出する一方、誤りが判明した見込み処理については、ロールバック処理により修正することで、シミュレーション結果の正しさを保証する。

ロールバック処理のオーバーヘッド削減

アンチメッセージ削減機構、適切なメッセージスケジューリング、適応型移動時刻界の導入によってロールバック処理のコスト、頻度ともに削減する。

回路分割

縦割り指向戦略により対象回路を静的に分割し、プロセッサ間通信頻度低減、並列性抽出、負荷の均一化を図る。



背景

論理シミュレーションは、LSI 設計において、設計回路の論理機能、および信号伝播タイミングを検証する工程である。この工程は、多大な計算時間を必要とすることから高速化が強く望まれているとともに、遅延の詳細度変更など種々の要求に適應できる柔軟性も要求されている。

ソフトウェア論理シミュレータの並列化は、高速かつ柔軟なシミュレーションを実現する有望な方法である。ICOT では、将来の超並列計算機を用いた超高速シミュレーション実現の第一歩として、PIM 上に並列論理シミュレータを開発した。

システム仕様

本シミュレータは、組み合わせ回路、順序回路 (非同期回路も含む) とともに扱うことができる。信号値モデルとしては、Hi、Lo、X(不定) を用いる 3 値モデルを、また、遅延値モデルとしては、各ゲートに異なる値を割り当てることができる非単一遅延モデルを採用した。なお、本シミュレータはゲートのみを扱うため、フリップフロップなど機能ブロックは全てゲートに分解して記述する。

並列離散事象シミュレーションとタイムワープ機構

並列論理シミュレーションは、並列離散事象シミュレーションの問題として扱われる。ここでは、ゲートに対応したオブジェクトが互いにメッセージを交換し、その状態を変化させていくことでシミュレーションが進行するように問題をモデル化する。メッセージは事象 (信号値変化) 情報を持ち、その生起時刻が刻印されている (タイムスタンプ)。正しいシミュレーション結果を得るためには、各オブジェクトで時刻順にメッセージを処理していかなければならないため、時刻管理機構が必要になる。

本シミュレータでは、分散的な時刻管理機構の一つであるタイムワープ機構を採用した。タイムワープ機構では、各オブジェクトは時刻順にメッセージが到着すると仮定して、履歴を保存しつつメッセージ処理を進める。メッセージが遅れて到着した場合は履歴を巻き戻し (ロールバック)、処理をやり直す。また、誤って送信されたメッセージは、アンチメッセージを用いて取り消す。

ロールバック処理のオーバーヘッドは性能に悪影響を及ぼすことが懸念されるため、本シミュレータでは、このオーバーヘッドを小さく抑える幾つかの工夫を加え、PIM 上での効率的な動作を実現した。

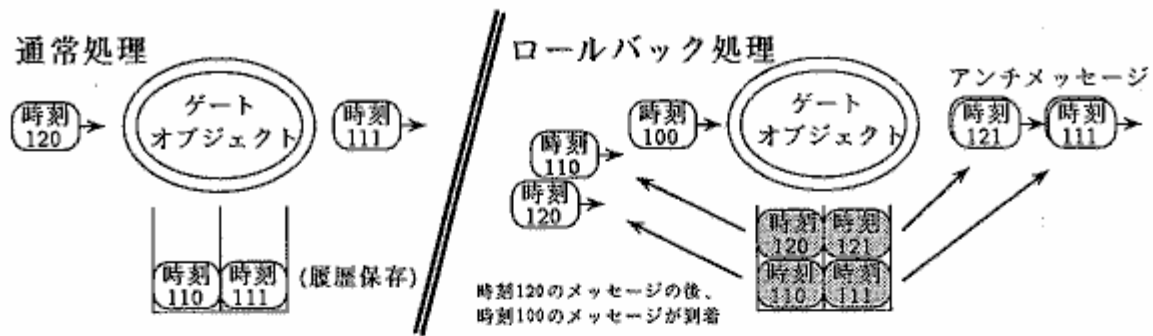


図 1: タイムワープ機構

スケジューリング

通常、シミュレーション中には、一つのプロセッサに複数のメッセージが存在するため、何らかのスケジューリング戦略が必要である。本シミュレータでは、最小タイムスタンプを持つものから処理を行う最小時刻優先の戦略を採用している。タイムワープ機構では、タイムスタンプの小さいメッセージほど、巻き戻される可能性が小さいと考えられるため、本戦略は、ロールバック頻度を小さく抑えたと考えられる。

アンチメッセージ削減

KL1 ストリーム通信では、メッセージは送信時と同じ順序で受信される。この環境においては、連続して送信されるアンチメッセージは削減することができる。メッセージ取り消しに必要なアンチメッセージ数が削減できることは、ロールバックコストの低減につながる。

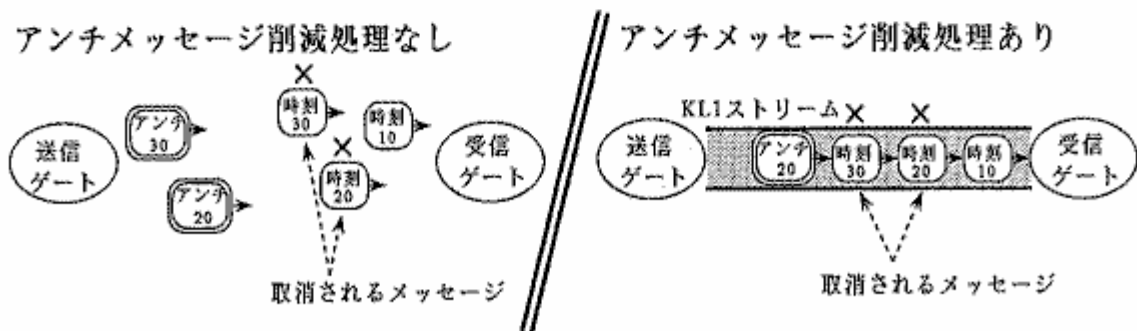


図 2: アンチメッセージ削減

適応的移動時刻界 (AMTC)

素朴に実装されたタイムワープ機構では、ゲートにメッセージが到着すれば、即座にその評価を行う。しかしながら、極度に将来の時刻を持ったメッセージについては、評価後に巻き戻される可能性が極めて高い。

MTC(移動時刻界) は、評価を行うメッセージの時刻の上限値を一時的に与えるものである。MTC の導入によって、極度に将来の時刻を持つメッセージの評価および生成を抑制できる。したがって、MTC はロールバック頻度を低下させると考えられる。

MTC は適当なタイミングで、適当な時間幅だけ将来の方向に進んで行く。しかしながら、MTC の適切な移動幅を予め静的に決定しておくことは難しい。例えば、MTC の移動幅を過小に設定した場合、ロールバック頻度は低減できるものの、プロセッサの遊び時間が長くなる。一方、MTC の移動幅を過大に設定した場合、ロールバック頻度はほとんど低減できない。

本シミュレータでは、ロールバック頻度や、以前の MTC 移動幅が性能に与えた影響の情報を元に、適応的に新たな MTC 上昇幅を決定してゆく (AMTC)。この方法により、対象回路、シミュレーション条件に依存せず、適切な MTC 上昇幅を得る。

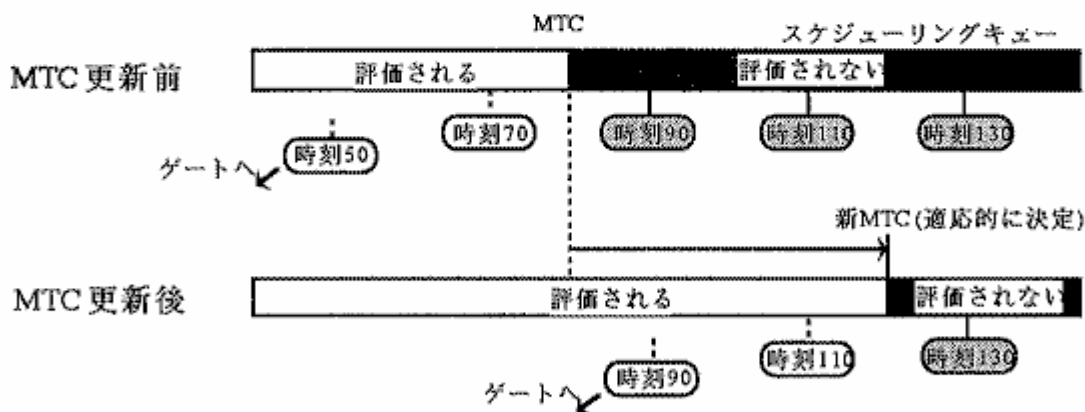


図 3: 適応的移動時刻界 (AMTC)

回路分割

本システムでは、縦割り指向戦略により、回路を静的に分割する。この戦略では、回路入力端子から順にゲート接続関係をたどり、縦方向につながったゲートをグループにまとめることによって、回路を幾つかのクラスタに分割する。全ゲートのグループ化終了後、クラスタの大きさの均一化処理を行い、最後にクラスタをランダムに各 PE に割り当てる。

本戦略は、回路分割を行う時の目標である、
 ・ 負荷の均一化、
 ・ プロセッサ間通信低減、
 ・ 高い並列性抽出の 3 点を満たす。

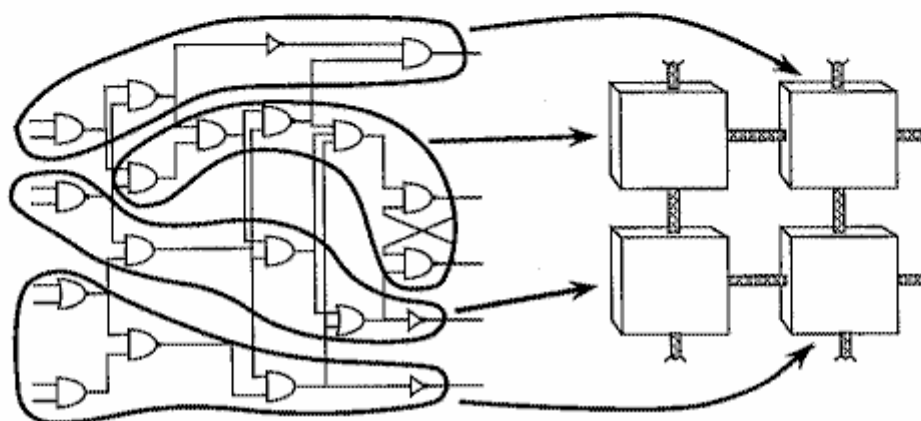


図 4: 縦割り指向戦略

性能評価

ISCAS'89 のベンチマークのうち、五つの順序回路について、PIM/m を用いてシミュレーションを行った。各回路規模 (ゲート数) を表 1 に、また、性能とプロセッサ台数の関係を図 5 に示す。

回路	s38584	s38417	s35932	s15850	s13207
ゲート数	27,965	31,995	26,433	13,354	11,965

表 1: 回路規模

256 プロセッサを用いた場合、最高 537K イベント / 秒の性能を得た。また、この時の台数効果は 166 倍であった。

性能 (K イベント / 秒)

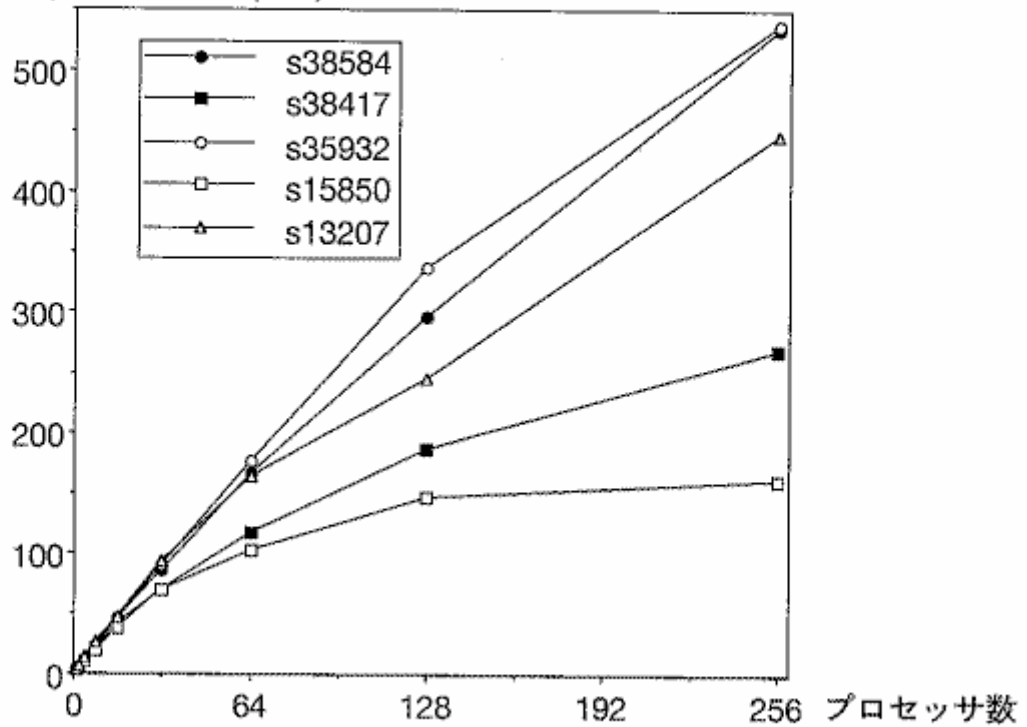


図 5: 性能と使用プロセッサ数

デモ概要

ISCAS'89 のベンチマーク s15850 および s38584 を対象回路とし、PIM/m 256 プロセッサを用いて、シミュレーションを行う。

特に、以下の点に注目して、デモンストレーションを進める。

- 全プロセッサの効率良い稼働状況
- 移動時刻界の移動幅が適応的に決定されていく状況
- 最終的なロールバック数が少なく、高い性能が得られている点