

クラスタ環境での静的タイミング解析の並列化*

4ZB-01

須藤 重 月山 友宏 月川 淳 大津 金光 横田 隆史 馬場 敬信†

宇都宮大学工学部情報工学科‡

1. はじめに

近年マイクロプロセッサの性能の向上はめざましく、それにつれて回路規模も増大しており、現在、1 LSI 当り数百万～数千万ゲート規模となっている [1]。そのため、シミュレーションや配置配線などの LSI 設計 CAD (Computer Aided Design) アプリケーションでは、このような大規模な問題を扱う必要が生じている。

本研究は、回路規模の増大に伴い処理時間が増加しているタイミング解析を高速化することを目的とする。タイミング解析とは、回路の遅延時間を計算しフリップフロップ (以後 FF と表記する) が正常な動作を行うかを解析する LSI-CAD アプリケーションである。精密な解析には処理に時間がかかり開発コストを増大させる。

そこで、本研究ではタイミング解析を並列化して高速化する。対象とするプラットフォームは、クラスタを想定している。

2. 静的タイミング解析

タイミング解析とは、論理回路によって信号が素子を通過する際に生じる遅延を計算し、FF への書き込みが正常に行なわれるかどうかを検証する事である。本研究では、組合せ回路で結ばれた FF1 と FF2 のクロックを CLK1、CLK2 としたとき、CLK1 と CLK2 が同一のクロックである時の Setup、Hold 条件のチェックを行った。

● Setup 条件式

(FF1 の出力～FF2 への入力)の遅延 <
Clock_{cycle}+CLK2 の遅延-CLK1 の遅延-FF2 の Setup 時間

● Hold 条件式

(FF1 の出力～FF2 への入力)の遅延 >
CLK2 の遅延-CLK1 の遅延+FF2 の Hold 時間

3. 並列化の方針

全回路データを各プロセスが保持しタイミング解析の開始点を分散させることにより、リニアに近い台数効果の向上が得られることが分かっている [2]。しかし、近年の回路規模の増大に伴い、全回路データを各プロセスが保持する方式は現状にそぐわない。このため、本研究では回路を分割してタイミング解析を並列化する。

回路分割を用いると、各プロセスは部分回路のデ

ータを持つことになり (図 1)、タイミング解析の実行時には自プロセスが保持している部分回路の情報だけでは不足する。そのため、プロセス間でメッセージ転送を行わなければならない。プロセス間のメッセージ転送の実装には移植性が高くプログラミングが容易な、MPI ライブラリを用いた。

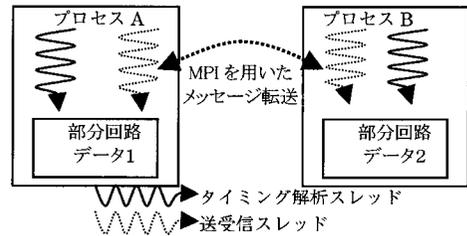


図 1 並列化の方針

4. 並列プログラミングの実装

4.1 一括転送による転送時間の削減

MPI ライブラリを用いてメッセージ転送時間の計測をおこなった。環境は PC (Pentium III Processor 800MHz/256Mbytes Memory/OS: Windows 2000 SP1) をルーター (CISCO Catalyst 6506) で接続したものを使用した。結果は、2kbytes のメッセージを送るのには約 0.9ms、4kbytes のメッセージには、約 1.3ms だった。この結果より、複数のメッセージを一括して転送することにより総転送時間を削減できることの見通しを得た。

4.2 メッセージ転送ルーチンの別スレッド化

本プログラムでは、タイミング解析ルーチンは開始点のリストから開始点を取り出して実行を行い、接続先の情報が見つからないとき、メッセージ転送ルーチンによって他のプロセスにメッセージの送信を行う。

この際、メッセージの送受信を別スレッドで実行すれば、送受信処理はタイミング解析処理が行なわれている間も実行されるため、受信側プロセスの開始点のリストが空になる事態が発生しにくい。また、タイミング解析スレッドは開始点のリストが空にならない限り、タイミング解析処理を行っているため、アイドル時間が発生しにくい。以上の理由からメッセージ転送ルーチンは別スレッドとして実装した。

更にメッセージはすぐには送信せずいったん送信用バッファに蓄積する方式とした。これにより転送一回あたりの量が大きくなり、4.1 節で述べたような転送効率の向上が期待できる。

4.3 メッセージ転送ルーチンの詳細

メッセージ転送ルーチンの詳細を以下に示す (図

*Parallel processing of static timing analysis in cluster environment

† Shigeru Sudo, Tomohiro Tsukiyama, Atushi Tsukikawa, Kanemitsu Ootsu, Takashi Yokota, Takanobu Baba

‡ Department of Information Science, Faculty of Engineering, Utsunomiya University

2).はじめにメッセージ転送スレッドが起動されると、ノンブロッキング受信を発行し、他プロセスからのメッセージ転送に備える(1)。メッセージを受信した場合(2)、送られて来た接続先のない素子情報を開始点のリストに加え(3)、ノンブロッキング受信を再発行する。次に、複数のタイミング解析スレッドが接続先のない素子の情報を蓄積していく(A)送信用バッファをチェックする(4)。規定数以上のメッセージが蓄積されるか、 α が連続して規定数回実行された場合、送信用バッファに蓄積された分全てを転送する(5)。

ノンブロッキング受信は、受信手続きを行ったあと別の処理を行うことができるため、受信が完了していない場合にも送信用バッファのチェックやメッセージ転送などの処理が可能となる。

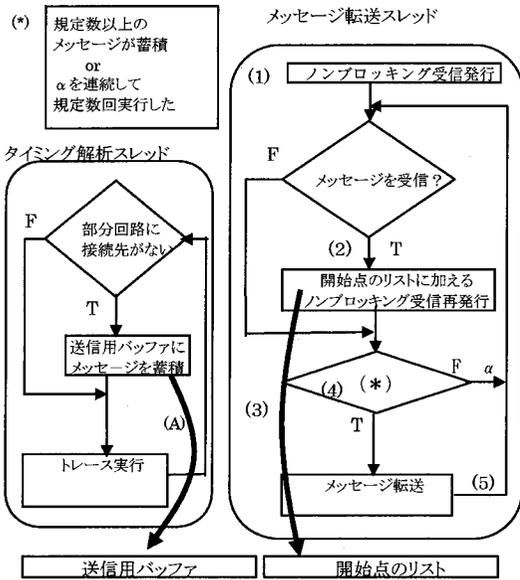


図 2 メッセージ転送の詳細

4.3 部分回路の作成

素子データは素子の種類、各素子に与えられた通し番号(素子番号)、ピン番号からなっており、回路分割は素子番号によって分割する。

この方式にする理由は、素子番号が近い(ネットリストにおいて素子情報が近い)ものを同一の部分回路に保持することにより、回路の接続先が見つからない事態(メッセージ転送が発生する)が減少することが期待されるためである。

4.4 回路分割時の境界上素子

回路分割時の境界上周辺部分を両方のプロセスが重複して保持することを許した。これは、各プロセスが境界上の素子情報を多く保持することで、メッセージの転送量を減らせることが期待されるためである。重複を許す幅は1プロセスに割り当てられる素子数に対する割合によって決定する。

5. 評価結果

実験にはPCをルーターで接続したもの(4.1節と同様)を使用した。実験データに素子数466万の回路を用いた場合の台数効果を図3に示す。図中の線はリニアな速度向上比をあらわすものである。4.4節の境界上周辺部分の重複幅は0.3とした。

この実験データでは1台では素子情報が仮想メモリに格納しきれなくなり、実行することはできない。そのため、2台のときの台数効果を2とした。

計算機4台で台数効果4.5、6台で7.9、7台で9.1の台数効果を得られた。スーパーリニアになるのは計算機が2,3台のときには割り当てられている部分回路が非常に大きいのでメモリとハードディスクの間でスラッシングが発生するためである。それに比べ4台以上では格納する素子情報が少なくなり、スラッシングが発生しにくくなる。また、格納している素子情報が少ないということは接続先の素子の検索時間が短くなるため、図3のような結果になったと考えられる。また、素子数466万の回路データには、素子数46.6万の回路データを10個ならべたものを使用した。そのため、5台での実行時には通信の発生が少なくなり台数効果が他の台数に比較し突出したと思われる。

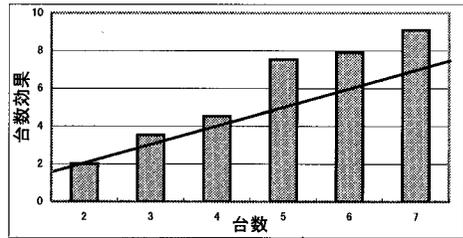


図 3 台数効果

6. おわりに

本稿では、MPIを用いた並列タイミング解析プログラムを作成し評価を行った。そして、大規模回路(素子数466万)による実験では、スーパーリニアとなる台数効果を得た。今後の課題は、更に計算機の台数を増し、より大規模なクラスタ環境での性能評価を行うことである。また、本研究のタイミング解析は完全同期回路のみを対象としていたため、他の同期関係の解析を行うことも課題である。

謝辞

本研究は、一部日本学術振興会科学研究費補助金(基盤研究(C)課題番号12680328)の援助による。

参考文献

[1]若林, 渡部, 小出“システムオンシリコンを支えるCAD技術”, 電子情報学会誌 Vol.81, No9, pp. 903-907, 1998
 [2] 土屋竜太, 大津金光, 吉永努, 馬場敬信 “WS クラスタを用いた並列論理回路タイミング解析の高速化”, 情報処理学会研究報告, Vol. 98, No. 72, pp. 55-60, 1998