

予測ヌルメッセージによる並列論理シミュレーション*

和田 耕一¹

Neo Teck Keng^{2†}

¹筑波大学 電子・情報工学系[‡]

²(株)横河ヒューレットパカード[§]

概要

並列論理シミュレーションの新しいアルゴリズムとして、下流ゲートのふるまいを予測することで、効率良くデッドロックを回避できる予測ヌルメッセージ法を提案している。本方式の詳細について述べ、汎用並列計算機上で開発したシミュレータの構成と性能評価について述べている。本方式により、従来のヌルメッセージ法の47倍、問い合わせ法の2倍の性能が得られた。また、適切な予測を行うことで、メッセージ数を大幅に削減し、かつデッドロックを回避出来ることを明らかにした。

Abstract

A speculative null message, which is a new deadlock avoidance algorithm for parallel logic simulation, is proposed. The algorithm can eliminate deadlocks efficiently by speculating behavior of the succeeding elements. The results of the evaluation show that the proposed algorithm exhibits the speed of 47 times faster than the conventional null message algorithm with eight processors. The result also shows our algorithm can maintain parallelism with very few messages for deadlock avoidance.

1 はじめに

近年のLSIの大規模化に伴い、論理シミュレーションに要する時間と費用が急増しており、柔軟で高速な論理シミュレータの必要性が高まっている。また、CAD(Computer Aided Design)ツールの発達により、LSIの設計からシミュレーションによる動作の検証、開発までをワークステーション上で実行できるようになってきた。ソフトウェアによる論理シミュレータは、CADツールの隣接工程との接続性も良く、高速化が達成できれば、LSI開発効率の向上に大きく寄与できる。

本研究では、下流ゲートのふるまいを予測することで、効率良い並列実行を可能とする並列論理シミュレーション方式を提案する。本論文では、本方式の詳細について述べ、汎用並列計算機上で開発したシミュレータの構成と性能評価について述べる。

本方式は、並列離散系シミュレーションアルゴリズムの一つであるコンサーバティブ法に基づいている。コンサーバティブ法では各プロセスにおいてすべての入力にイベントが到着するまで出力イベントを生

成しないため、デッドロックが生じる可能性がある。コンサーバティブ法に基づいた並列論理シミュレーションでは、このデッドロックをいかに効率良く回避、解決できるかが性能向上の鍵を握る。

2章では、コンサーバティブ法について述べ、3章では、提案する予測ヌルメッセージ法の詳細と位置付けについて述べる。4章では、本シミュレータの構成について述べる。5章では、本方式に基づいたシミュレータの性能、デッドロック回避、解決のためのメッセージ数などをもとに、本方式の有効性について評価を行う。

2 コンサーバティブ法

コンサーバティブ法におけるデッドロック対処法としては、以下の二種類が知られている [2] [3] [9]。

2.1 ヌルメッセージ法

ヌルメッセージ法は、ゲートの出力が変化しない場合にも下流ゲートに時刻を通知するメッセージを発行し、デッドロックを回避するものである。時刻通知のために発行されるメッセージをヌルメッセージと呼ぶ。ヌルメッセージを用いることでデッドロックを必ず回避できるが、次のような問題がある。

ある時刻に出力が変化しているゲート数は、多くの場合、全ゲートの数%以下である [5] [1]。ヌルメッ

*Parallel Logic Simulation Using Speculative Null Message

[†]Koichi Wada¹ and Neo Teck Keng²

[‡]Institute of Information Sciences and Electronics, University of Tsukuba

[§]Yokogawa-Hewlett-Packard, Ltd.

メッセージ法では、出力が変化しないときにもメッセージが発生するため、その数は真のメッセージ数をはるかに上回る。その結果、メッセージ数の爆発を招き、ヌルメッセージの処理に多くの時間を消費することになり、場合によっては膨大なメッセージ数によってネットワークが輻輳状態に陥る可能性がある。松本らによる文献 [4] では、ヌルメッセージ法の評価もなされており、全メッセージの 97% がヌルメッセージであったと報告されている。

2.2 問い合わせ法

問い合わせ法では、イベントが入力に揃わないためにデッドロックに陥ったことを検出したら、イベント未到着の入力ラインを介して上流ゲートに時刻を問い合わせる。上流ゲートが、デッドロックの解決を保証できる時刻に到達していたら、その時刻をメッセージにより返答する。保証できない場合は、さらに上流に問い合わせを転送する。時刻の問い合わせに用いられるメッセージを問い合わせメッセージと呼ぶ。

問い合わせ法は、要求駆動型のヌルメッセージ法とみなすことができ、少ないメッセージ数でデッドロックを解決できる。しかし、問い合わせを発行した後、返答が返されるまで当該ゲートがブロックするため並列度が抑制される問題がある。

3 予測ヌルメッセージ法

本論文では、前章で述べたデッドロック対処法の問題点を克服する方法として、予測ヌルメッセージ法と呼ぶデッドロック回避・解決法を提案する。

予測ヌルメッセージ法の基本概念の説明のため、図 1 に 3 個の AND ゲート P, Q, R から成る論理回路と単純化した動作例を示す。入力信号線は、a, b, c, d とし、a は "H" に、b は "L" に固定されているとする。さらに、入力 d には、先頭から順にタイムスタンプ 50, 100 のイベントが受信されているとする。

予測ヌルメッセージ法では、問い合わせ法に基づいてシミュレーションを開始する。まず、ゲート P と R 間で起こり得るメッセージの授受について述べる。ゲート P は、b 入力 "L" であるため入力 c に受信したイベントにかかわらず出力が変化しない。このため、P は出力信号線上にイベントを送出せず、R は、d 上のタイムスタンプ 50 のイベントの評価を

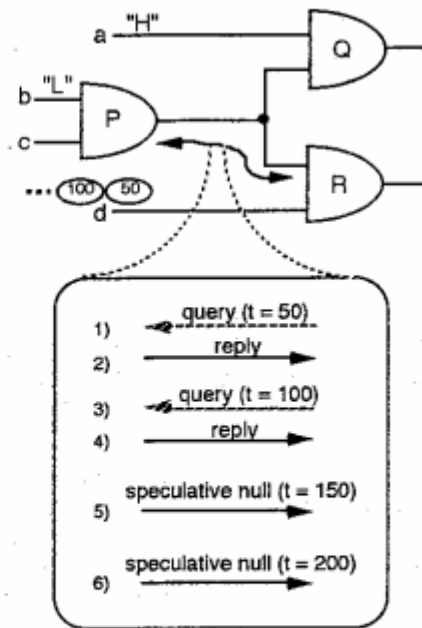


図 1 : 予測ヌルメッセージ法

進めることが出来ない。R は、問い合わせ法に従って、P に対して時刻 50 以前のイベントが生じ得るかどうかの問い合わせメッセージを送る。P の時刻が 50 以上であれば、返答メッセージを送信し、R はタイムスタンプ 50 のイベントの評価を行う。続いて R は、前回と同様に時刻 100 に対する問い合わせを P に発行する。

ゲート P は、この時点までに受け取った時刻 50 と 100 の問い合わせをもとに、50 単位時間刻みで R がデッドロックに陥ると予測する。この予測に従って、P は、時刻 150 から 50 単位時間毎にヌルメッセージを R に送信する。

一方ゲート Q は、入力 a が常時 "H" であるため、さらに下流のゲートからの問い合わせを転送する場合を除き、ゲート P に対して自発的に問い合わせを行うことはない。従って、P も Q に対しては、ヌルメッセージを送信しない。もしも、入力 a にもある時刻間隔でイベントが到着していた場合は、その間隔に見合った予測時刻に、ヌルメッセージが P から Q へ送信される。

以下に、予測ヌルメッセージ法の手順と予測につ

いてまとめ、位置付けについて述べる。

3.1 デッドロック回避、解決の手順

予測ヌルメッセージ法によるデッドロック回避、解決は以下の手順で行われる。

1. シミュレーションは、基本的に問い合わせ法で開始される。
2. 下流ゲートから問い合わせを受け取った場合、問い合わせの発行元とその時刻を記録する。
3. 記録された問い合わせの履歴をもとに、将来デッドロックを起こす可能性がある下流ゲートと時刻を予測する。
4. 予測された時刻に到達したら、該当する下流ゲートにヌルメッセージを発行する。

すなわち、予測ヌルメッセージ法は、論理回路のふるまいをシミュレーション実行時に監視し、下流ゲートがデッドロックに陥る可能性を予測して、選択的にヌルメッセージを発行することでデッドロックを回避させるものである。

3.2 予測

本方式では、二つの意味での予測を行っている。

問い合わせを発行するゲートの予測

ヌルメッセージを送るべき下流ゲートを動的に予測することで、ヌルメッセージ数の削減を図る。過去に予測を受け取ったゲートのみに対して選択的にヌルメッセージを送出する。

問い合わせを発行する時刻の予測

問い合わせを受け取ったら、問い合わせの送出元ゲートとその時刻を問い合わせ履歴として保存する。この履歴をもとに、下流ゲートが再度問い合わせを発行するであろう時刻を予測するわけであるが、予測処理そのもののオーバーヘッドを極力抑える必要がある。予測時刻の算出には、過去の問い合わせ間隔の最大値、平均値、最小値などを用いることが考えられる。これらは、計算量が少なく、必要メモリ容量も小さい。

3.3 予測ヌルメッセージ法の位置付け

Misra は文献 [2] において、ヌルメッセージ法の改善策としてタイムアウトメカニズムを提案している。タイムアウトメカニズムは、一定時間 τ だけヌルメッセージの送信を遅らせることでヌルメッセージ数の削減を図るものである。しかし、[2] では、 τ の決定方法については述べられていない。予測ヌルメッセージ法は、問い合わせの履歴によって τ を決定する方式と見ることができる。

また、一度シミュレーションを実行してデッドロックを起こすゲートをマークし、次回以降のシミュレーションでは、マークされたゲートに対してのみヌルメッセージを送信するキャッシング法が、Soule らによって提案されている [1]。これに対し、予測ヌルメッセージ法は、過去に受け取った問い合わせをもとに、ヌルメッセージを送信すべきゲートを動的に決定する方式であるといえる。

4 シミュレータの構成

本シミュレータは、汎用並列計算機である米 Cogent Research 社の XLNT 上で開発した [6]。XLNT の要素プロセッサ (PE: Processor Element) は、トランスピュータ T800 である。本シミュレータは、10 プロセッサのシステム上で開発した。

並列プログラムの開発言語には XLNT 上の Linda [7] [8] である kernel Linda [6] と C++ 言語を用いた。Kernel Linda では、多重タプル空間が提供されている。すなわち、全プロセスから参照可能なタプル空間のみでなく、特定のプロセス間でのみ参照できるタプル空間を生成できる。各プロセスの通信パターンを考慮して、タプル空間を適正に配置することにより通信を局所化できる。本シミュレータでは、プロセス毎に受信用タプル空間を生成し、メッセージの受け渡しに使用している。

負荷分散のための回路分割は、回路全体の出力から入力方向へゲートを辿りつつ分割領域を決定する縦割り分割とした。

5 評価

ISCAS'89 でのベンチマーク回路である s1494 を用いて評価を行った。s1494 は以下のような回路である。

表1: 絶対性能

PE数	(events/sec)		
	予測ヌルメッセージ法	ヌルメッセージ法	問い合わせ法
2	3486	84	1679
4	6300	154	3107
8	10169	216	4982

逐次版: 2262 events/sec

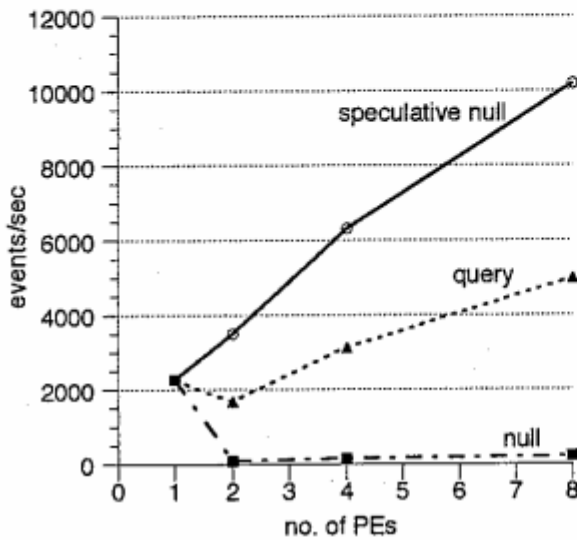


図2: 絶対性能

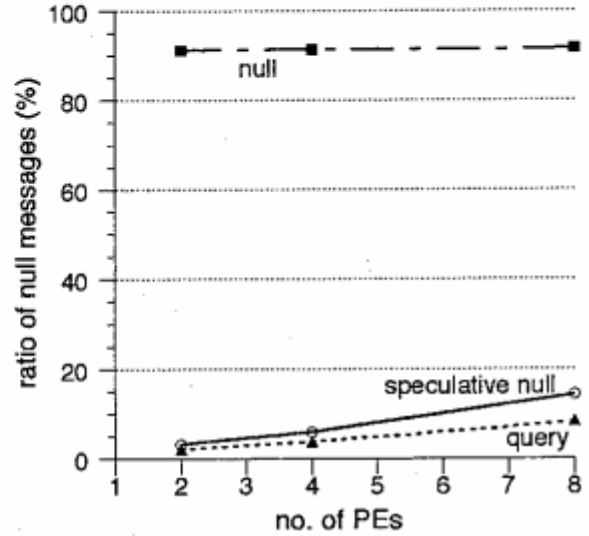


図3: ヌルメッセージの占める割合

ゲート数: 638

信号線数: 1490

平均ファンアウト数: 2.08

入力イベントの生成では、乱数を用いて生起時刻を決定している。プロセッサ数を1台から8台まで変化させ、実行時間とヌルメッセージ数、問い合わせメッセージ数などを計測し、予測ヌルメッセージ方式の有効性の評価を行った。

予測ヌルメッセージ法における時刻予測には、過去の問い合わせ間隔の最大値を用いている。また、従来のヌルメッセージ法によるシミュレーションでは、入力信号線のキュー中の不要なヌルメッセージを削除し、ヌルメッセージ数を削減する手法を取り入れている。

5.1 絶対性能と台数効果

予測ヌルメッセージ法とヌルメッセージ法、問い合わせ法によるシミュレータの絶対性能を表1に、グラフ化したものを図2に示す。PE1台の場合は、逐次シミュレーションの性能を示している。

PE数2台からの性能向上率は、各方式とも同程度であるが、ヌルメッセージ方式の絶対性能は非常に悪い。PE8台の場合、予測ヌルメッセージ方式は、問い合わせ法の約2倍、ヌルメッセージ法の約47倍の性能が得られていることがわかる。

5.2 ヌルメッセージの割合

図3に、全メッセージの中でヌルメッセージが占める割合を示す。問い合わせ法については、問い合わせに対する返答メッセージをヌルメッセージとし

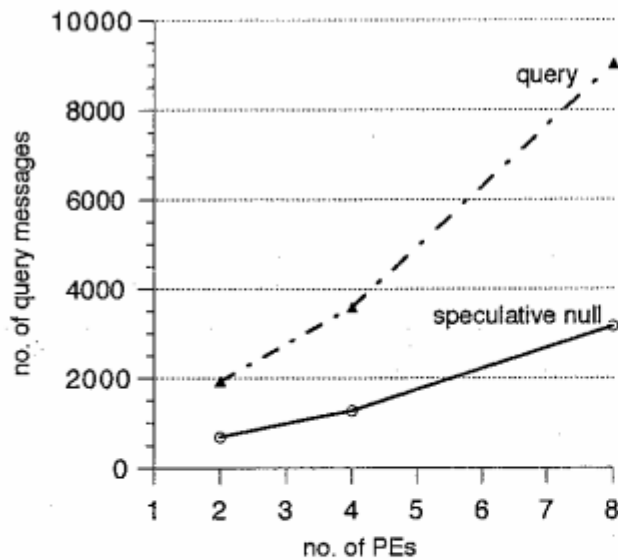


図4: 問い合わせメッセージ数

で勘定している。8PEで従来のヌルメッセージ法を用いた場合、全メッセージ数の約90%がヌルメッセージである。ヌルメッセージ法では、多量のヌルメッセージに要する処理が絶対性能を低下させていることが確認できた。

一方、予測ヌルメッセージでは14%と、ヌルメッセージ法に比べて大幅にヌルメッセージを削減できていることがわかった。この大きな差が生じた理由は、次のように考えられる。ヌルメッセージ法では、受信されたヌルメッセージは、ファンアウト数の複製が作られ、下流ゲートに送信される。すなわち、ヌルメッセージの受信と送信には強い因果関係がある。これに対して、予測ヌルメッセージ法では、ヌルメッセージを送信するか否かは、下流ゲートからの問い合わせの有無にのみ依存する。このように、ヌルメッセージ送信の要件を個々のゲート間の関係に限定したことが、メッセージ数の爆発を避けられた主たる理由であると考えられる。

5.3 問い合わせメッセージ数

図4に、問い合わせメッセージ数を示す。PE8台の場合、問い合わせ法における問い合わせメッセージ数が約9000であるのに対し、予測ヌルメッセージ法では約3000である。これは、予測ヌルメッセー

表2: デッドロック回避、解決のためのメッセージ数

PE数	予測ヌルメッセージ法	問い合わせ法
2	3587	3858
4	7018	7170
8	18740	18030

ジによって、2/3のデッドロックが回避できたことを示している。

5.4 デッドロック回避、解決のためのメッセージ数

表2に、予測ヌルメッセージ法と問い合わせ法において、デッドロックを回避、解決するために発行されたメッセージ数を示す。予測ヌルメッセージ法では、予測ヌルメッセージ数、問い合わせメッセージ数、返答メッセージ数の合計が示されている。問い合わせ法については、問い合わせメッセージ数と返答メッセージ数の合計が示されている。

表より、予測ヌルメッセージ法は、問い合わせ法とほぼ同じメッセージ数によってデッドロックを回避・解決していることがわかる。しかし、絶対性能では、問い合わせ法は予測ヌルメッセージ法の約半分であり、大きな差が存在する。これは、問い合わせ法では、問い合わせに対する返答を待つために並列性が抑制された結果であると考えられる。逆に、予測ヌルメッセージ法では、予測ヌルメッセージによってデッドロックを回避することで、並列性を維持できていることがわかる。

5.5 予測の有効性

本節では、問い合わせ履歴からの予測の有効性について考察する。予測ヌルメッセージの発行時刻を

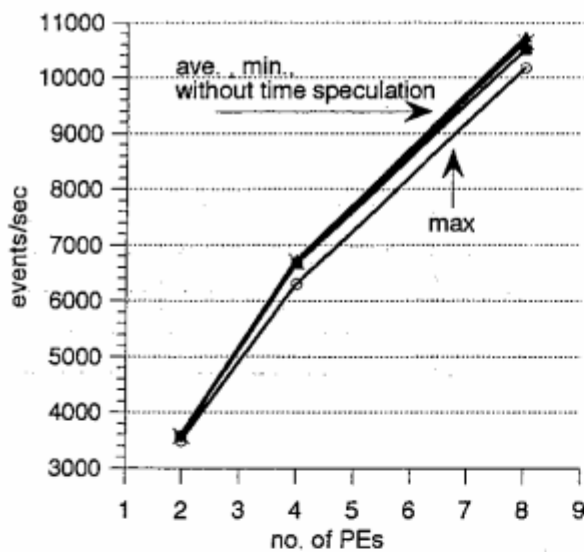


図 5 : 予測法と性能

履歴から算出する方法として、問い合わせ間隔の最大値、平均値、最小値を用いるものをそれぞれ実装した。また、比較のため、時刻の予測を行わない場合、すなわち、過去に問い合わせを受けた先のゲートに、従来のマルチメッセージ法と同じタイミングでマルチメッセージを発行する場合の実行データも示す。

表 3 と図 5 に、予測時刻の 3 種類の算出法と時刻予測を行わない場合についての絶対性能を示す。高い性能を示した予測法は最小値を用いた場合であり、以降順に平均値、最大値となっている。しかし、その差は高々 6% であり大きな差はない。注目すべき点は、時刻の予測を行わない場合の性能で、2 番目に良い結果が得られた。このことより、絶対性能に関する限り、時刻の予測の有効性は認められない。

次に、予測法とデッドロック回避、解決のためのメッセージ数の関係を図 6 に示す。図より、最小値を用いる場合、および予測を行わない場合は、最大値を用いる場合の約 6 倍のメッセージが発行されている。本結果と絶対性能の比較結果から、最小値の場合と予測を行わない場合に発行されるデッドロック回避メッセージのうち約 8 割は、性能向上に寄与していないことがわかる。逆に、最大値による予測を用いることで、極めて少ないメッセージ数で性能

表 3 : 予測法と性能

PE数	予測法			時刻予測無し
	最大値	平均値	最小値	
2	3486	3571	3590	3583
4	6300	6665	6736	6694
8	10169	10503	10734	10642

を維持できることがわかった。以上より、適切な方法による時刻の予測は、メッセージ数の削減に有効と言える。

より大規模な回路を多数のプロセッサで実行する場合、必要メッセージ数の少ない最大値をとる方法は、性能上も優位になる可能性がある。

6 おわりに

本論文では、コンサーバティブ法による並列論理シミュレーションでの大きな問題であるデッドロックを効率良く回避、解決できる予測マルチメッセージ法について述べた。

現在までに提案されているマルチメッセージ法は、メッセージ数の爆発が性能を抑制する要因となり、問い合わせ法は返答が返されるまでの待ち時間が並列性を阻害する。

予測マルチメッセージによる並列論理シミュレータを開発し、性能評価を行なった。その結果、8 台のプロセッサを用いた場合、1 台の場合の 4.6 倍の速度向上が得られた。絶対性能は、従来のマルチメッセージ法の 47 倍、問い合わせ法の 2 倍であることを示した。

メッセージ数に関しても、8 プロセッサで従来のマルチメッセージ法を用いた場合、全メッセージ数の

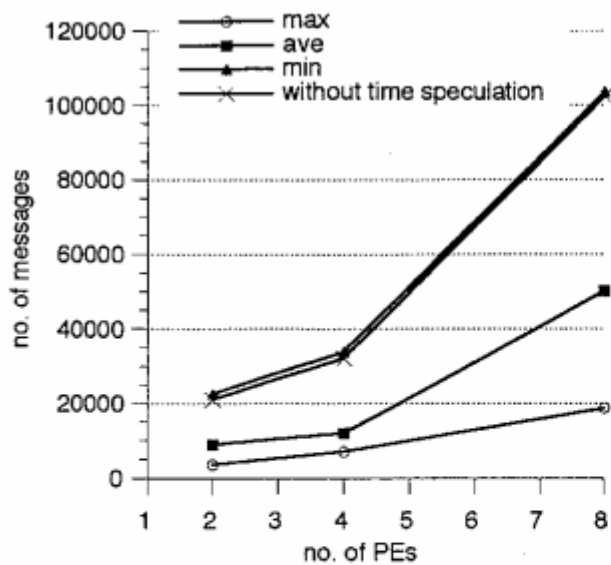


図6：予測法とデッドロック回避、解決のためのメッセージ数

約 90% がヌルメッセージであるのに対し、予測ヌルメッセージでは 14% と、大幅にヌルメッセージを削減できていることがわかった。問い合わせ数についても、予測ヌルメッセージ法によって、問い合わせ法を用いた場合の 1/3 の問い合わせ数に削減できていることを示した。また、時刻の予測法を適切に選ぶことにより、メッセージ数を 80%削減できることがわかった。

時刻の予測法については、本論文で述べた方法以外にもいくつかの方法が考えられる。論理回路の各部分は、全時間に渡って同じパターンで動作しているとは限らない。一定の時間が経過した後に、異なる動作モードに移行することも多い。例えば、このような急激な変化点で予測間隔を大幅に増減する方法も考えられる。

本研究で開発したシミュレータは、予測ヌルメッセージの実装法の検討に主眼を置いたもので、各プロセッサで実行されるシミュレータ部分は、改善すべき点が多く残されている。これらの最適化を含めて、現在本方式によるシミュレータをワークステーションクラスタ上に移植している。

また、より大きな、かつ特性の異なる回路を対象として、詳細な評価を加えることが今後の課題である。

参考文献

- [1] L. Soule and A. Gupta : *Analysis of Parallel and Deadlocks in Distributed-Time Logic Simulation*, Tech. Rep. CSL-TR-89-378, (May 1989)
- [2] J. Misra : *Distributed Discrete-event Simulation*, Comp. Surv., Vol.18, No.1, pp39-65(March 1986).
- [3] K. M. Chandy and J. Misra : *Distributed Deadlock Detection*, ACM Trans. Comp. Sys., Vol.1, No.2, pp144-156(May 1983).
- [4] 松本幸則, 瀧和男 : パーチャールタイムによる並列論理シミュレーション, 情報処理学会論文誌 Vol.33 No.3, pp387-395(March 1992).
- [5] 下郡慎太郎, 鹿毛哲郎 : メッセージドリブンによる並列論理シミュレーション, 電子情報通信学会研究報告, CAS88-110, pp.23-30, 1988.
- [6] Cogent Research Inc. : *Kernel Linda Specification : Version 4.0*, Technical Note 89.17(1989).
- [7] N. Carriero and D. Gelernter : *How to Write Parallel Program : A Guide to the Perplexed*, ACM Comp. Surv., Vol.21, No.3, pp323-357(Sept 1989).
- [8] N. Carriero and N. Gelernter : *Linda in Context*, Communication of the ACM, Vol. 32, No. 4, pp444-458(Apr 1989).
- [9] 工藤知宏, 木村哲郎, 寺沢卓也, 天野英晴 : 共有メモリを想定した並列論理シミュレータ, 電子情報通信学会, CPSY 91-23, pp151-158(July, 1991).