

並列論理シミュレータWIZDOM

4 Q - 3

赤村 具尋* 中井 匡* 大窪 一郎* 寺井 順子* 中川 義浩*
 橋本 訓治** 白鳥 優子** 神田 有洋**
 *NECソフトウェア北陸 **NEC

1. はじめに

近年のコンピュータシステムは、大規模化/高集積化すると共に、その製品開発サイクルは短くなってきている。そのため装置モデルの効率的な論理シミュレーションが必要となっている。

従来、スーパーコンピュータをはじめとする大規模装置モデルの論理シミュレーションには、HAL III (専用ハードウェアを用いた並列シミュレータ) を使用していた。しかし、設計するコンピュータの大規模化と設計の短期化にともない、ハードウェア論理シミュレータの再開発が必要だが、近年のSMPの性能向上、低価格化は著しく、開発期間と開発費用の面で優位性が乏しい。そこでSMPマシンであるNX7000/P590 (HP-UX) で動作する並列論理シミュレータWIZDOMの開発を行ない、ソフトウェアでHAL IIIと同等以上のシミュレーション性能を持つ論理シミュレータを実現した。

WIZDOMは、モデル分割による並列シミュレーションの実現と、命令レベルシミュレータ、I/Oシミュレータとの協調シミュレーションの実現により、大規模装置モデルの検証を可能にした。これにより、OSの立ち上げ動作の検証を実機評価前に行なうことが可能となり、実機評価期間を大幅に短縮できた。

2. 論理シミュレーション方式

SMP上でHAL III以上の性能を出すには、種々の高速手段が必要であり、その実現手段について述べる。

2.1 並列シミュレーション

装置モデルをSMPのプロセッサ数分分割し、それぞれをシミュレーションプロセスに割り当て並列に動作させる事により、高速な論理シミュレーションを実現した。

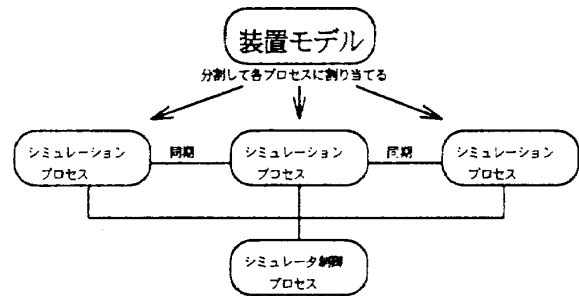


図1 並列シミュレーション

2.2 レベル付けによるサイクルベースシミュレーション

サイクルベースシミュレーションは、タイミングの検証を省略し信号値の計算をクロックエッジタイミングのみに限定することにより、高速なシミュレーションが可能である。WIZDOMにおいてはレベル付け方式を用いてサイクルベースシミュレーションを実現した。レベル付け方式とは、モデルの入力信号からレジスタ信号まで、レジスタ信号から出力先のレジスタ信号まで、レジスタ信号から出力信号までに関して、順番にレベルを付けておくことによりレジスタ及びその入力側の組合せ回路群を一つのまとまりとして、演算順序を決定する方式である。レベル順に演算を行ない、更にイベントドリブ方式と併用することにより、矛盾や無駄のないシミュレーションを行なうことができる。

2.3 高速通信

並列シミュレーションを行なう場合、分割された各モデル間でレベル同期を頻繁に行なう必要があり各モデル間の通信遅延がその性能に大きく影響する。そこで高速な並列シミュレーションを行なうために、共有メモリを用いた低通信遅延のプロセス間通信ライブラリを実現し、SMP上での高速な並列論理シミュレーションが可能となった。[1]

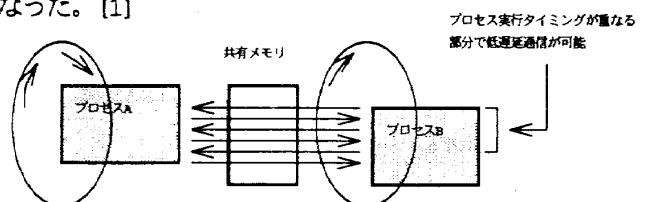


図2 SMPにおけるプロセス間通信

Parallel Logic Simulator WIZDOM
 Tomohiro Akamura*, Tadashi Nakai*, Ichiro Okubo*,
 Yoshihiro Nakagawa*, Junko Terai*
 Kuniharu Hashimoto**, Yuko Shiratori**, Arihiro Kanda**
 *NEC Software Hokuriku Ltd. **NEC Corporation

今回開発した低通信遅延のプロセス間通信ライブラリ(共有メモリ)性能と、標準的な関数(セマフォ、メッセージキュー)の性能を、NX7000上でプロセス数を変化させながら測定した結果を表1に示す。

プロセス数	共有メモリ	セマフォ	メッセージキュー
2	26	54	116
3	47	86	232
4	69	140	474

表1 通信プロセス数と1同期あたりの遅延(μsec)

3. OSシミュレーション

大規模モデルの論理検証としては、装置起動時に実行される装置診断プログラムやランダム試験を、装置モデルでシミュレーションすることで十分に行っている。しかし、実機評価において、OS立ち上げでバグが発見された場合、LSIを再作するまでは評価を先に進めることができず、開発遅延をまねいてしまう。故に、シミュレーションにてOSの立ち上げを行なうことが必須である。しかし、OSの起動動作を装置モデルでシミュレーションする場合、入出力装置へのアクセスが必要となってくる。ところが、入出力装置を装置モデルにモデリングするのは困難である。そこで、命令レベルシミュレータ、I/Oシミュレータと協調シミュレーションを行なうことで、OSの立ち上げ、シミュレーションを実現した。以下にOSシミュレーションの実現方法を述べる。

3.1 割り込み処理

WIZDOMではシミュレーション中にI/Oシミュレータからの割り込み処理を常に監視し、割り込み処理が発生した場合、それがメモリへの書き込み命令ならば、I/Oシミュレータより必要な情報を受け取り、装置モデル上のメモリに状態値の書き込み処理を行なう。メモリからの読み出し命令ならば装置モデル上のメモリの状態値を読み出し、I/Oシミュレータに送信する(図3の①部分)。

以上により、論理シミュレーションモデル上に入出力装置がモデル化されていなくとも、入出力装置へのアクセスが可能となる。

3.2 解析時間の短縮

OSの立ち上げシミュレーションは、1回のシミュレーション時間がとても長くなる。バグが検出された場合、

どのOSの命令列からおかしくなったかを限定できても、装置モデルをその時点の状態に戻すには、最初からシミュレーション行なう必要がある。そのため、解析には非常に時間を要してしまう。WIZDOMでは、命令レベルシミュレータと協調シミュレーションを行なうことにより、短時間で、指定した命令列の状態に戻すことを可能とした。その方法として

1. 命令レベルシミュレータは、指定した命令列までシミュレーションを短時間ででき(図3の②部分)、その時点でのメモリ、ソフトウェアレジスタの状態値をWIZDOMに送信する(図3の③部分)。

2. 命令レベルシミュレータから受け取った情報(図3の④部分)をWIZDOMは、装置モデル上の該当するメモリ、レジスタに状態値を設定する。

以上により指定した命令列の状態に短時間で戻すことが可能となった。

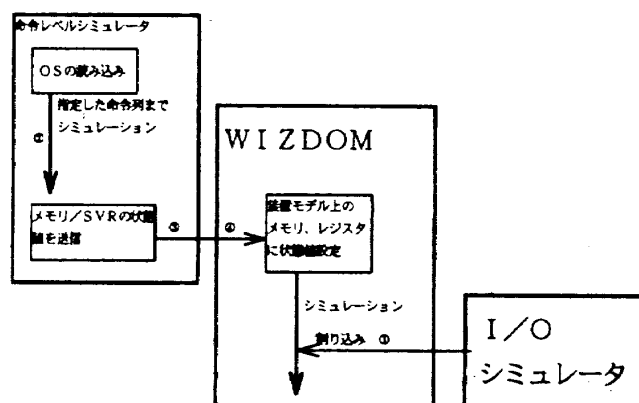


図3 OSシミュレーションの概要

4. まとめ

WIZDOMはハードウェアシミュレータHALⅢと同等以上の性能を実現し、また、ソフトウェアシミュレータであるため、マシン需要に対して柔軟に対応できるようになった。

HALⅢで行なっていた評価期間に対して1/2削減が行なえた。

[参考文献]

- [1] 神田有洋他『並列論理シミュレータのプロセス間通信』情報処理学会第55回全国大会
- [2] 中川義浩他『ハードウェアシミュレータHALⅢを用いた超大规模シミュレーション』情報処理学会第52回全国大会