

## 平成 19 年度 情報工学コース卒業研究報告要旨

高田 研究室	氏 名	相庭裕史
卒業研究題目	マルチプロセッサシミュレーション環境の機能拡張と効率化	
<p>近年、大規模・複雑化する組込みシステム開発において、発生頻度の低い現象の再現性が低いなどの理由から、実機検証が困難となっている。このような問題を解決する方法の 1 つが、シミュレータを使った検証である。また、組込みシステムでは、性能向上・消費電力削減等の要求から、マルチプロセッサ化が進んでおり、マルチプロセッサ用リアルタイム OS (以下、RTOS) の使用が広がりつつある。したがって、マルチプロセッサ用 RTOS に対応したシミュレータが必要となっている。</p> <p>我々の研究室では、複数の命令セットシミュレータ (以下、ISS) と、その通信を仲介するツールにより、マルチプロセッサシミュレーション環境を開発している。しかし、このシミュレーション環境には次のような問題が挙げられる。まず、ISS でのシミュレーション実行では検証時間が長くなる。ISS はターゲットプロセッサ用のオブジェクトコードを実行できるため、コードレベルのシミュレーションが可能である。しかし、ホスト計算機上でネイティブ実行するシミュレータに比べて、シミュレーション実行が非常に遅くなる。次に、シングルプロセッサ用の ISS を拡張しただけであるため、プロセッサ間のサイクルレベルの同期がとれない。そのため、実行ログからの協調動作解析が困難であり、ハードウェアシミュレータとのサイクルレベルのコシミュレーションが不可能である。本研究では、上記 2 点の問題を解消するために、従来のシミュレーション環境の機能拡張と効率化を行った。</p> <p>まず、シミュレーション時間短縮のために、RTOS のアイドル状態に着目した効率化を図った。RTOS 上で実行すべきタスクが無い状態をアイドル状態という。アイドル状態はシングルプロセッサ RTOS でも発生するが、マルチプロセッサ RTOS では他のプロセッサの処理を待つ間など、シングルプロセッサ以上にアイドル状態になる状況が増えると予想される。アイドル状態中は、割込みが入り新たなタスクが起動されるまで、繰り返し処理によって待つが、この繰り返し処理中も ISS を動作させ続けるのは効率が悪い。そこで本研究では、ISS 上の RTOS がアイドル状態に入った場合に、ISS 自体を待ち状態にすることでホスト計算機の CPU を効率的に用いる手法を提案した。実験では、デュアルプロセッサ用に開発された MPEG4 デコードシステムを用い、デコード処理にかかる時間を従来手法と比較した。その結果、従来手法の約 8 倍の高速化を達成し、有効性が確かめられた。</p> <p>次に、ISS 間の任意のサイクル数による同期機能を追加した。同期機構には、柔軟な通信機構である COM を用いた方法 (以下、同期手法 1) と、ホスト OS の共有オブジェクト機構を用いた単純な同期方法 (以下、同期手法 2) を提案した。実験により、両手法ともに同期が実現されたことを確認した。さらに、コア数 2 同期サイクル数 100 の条件のもとで、シミュレーション時間の比較実験を行った。2 つの ISS を同期させた結果、同期手法 1 では、非同期時の約 40 倍の時間がかかったのに対し、同期手法 2 では、約 1.5 倍の時間に抑えることができた。したがって、適用可能な場合は同期手法 2 を用い、他のシミュレータとのコシミュレーションにおいて、同期手法 2 が適用できない場合には柔軟な同期手法 1 を用いるべきという結論に達した。</p> <p>本研究により、マルチプロセッサシミュレーションの高速化とサイクルレベルの同期の実現が可能となった。</p>		