

報告番号	甲	第	号
------	---	---	---

## 主 論 文 の 要 旨

論文題目 組込みリアルタイムシステムを対象とした  
マルチコアプラットフォーム技術  
氏 名 一場 利幸

### 論 文 内 容 の 要 旨

組込みリアルタイムシステムを開発するには、まず、アプリケーションが動作するための基盤となるハードウェアおよびソフトウェア（プラットフォームと呼ぶ）を定めてから、アプリケーションの開発を行う。プラットフォームを構成するソフトウェア（ソフトウェアプラットフォーム）の代表例は、OSである。ソフトウェアプラットフォームを利用することで、アプリケーションソフトウェアの保守性や再利用性の向上を実現することができる。組込みリアルタイムシステムでは、特に、実行時間の予測可能性を持ったリアルタイム OS を用いるのが通常である。これは、組込みリアルタイムシステムでは、満たすべき時間要件（時間制約）に従って動作する性質（リアルタイム性）を要求されるためである。パーソナルコンピュータやスーパーコンピュータなどの汎用システムで用いられている汎用 OS では、リアルタイム性を満たすことができない。

現在、汎用システムのほとんどは、マルチコアシステムとなっている。ハードウェアリソースへの制約が厳しい組込みシステムにおいても、その重要性が増している。これは、現在の半導体技術においては、消費電力の増大を抑えつつシステム全体のスループットの向上を図るためには、クロック周波数を上げるよりも、コア数を増やしたほうが有利なためである。

マルチコアシステムの重要性が増すにつれて、マルチコアシステムをサポートするマルチコアプラットフォームの重要性も増している。ソフトウェアプラットフォームの役割の1つは、ハードウェアの隠蔽である。マルチコアシステムにおいては、対象となるハードウェアに関する知識を持った上でプログラミングしなければ、マルチコアシステムの性能を十分に発揮できない。ソフトウェアプラットフォームを利用することにより、アプリケーション開発者に求められるハードウェア知識を減らすことができる。マルチコアシステム上のソフトウェアは、複雑なも

のになりやすく，マルチコアプラットフォームの果たす役割は大きいといえる．

マルチコアプラットフォームを構築するにあたって解決すべき課題は多く，組込みリアルタイムシステムで用いるために重要な点として，次の3つが考えられる．1点目はリアルタイム性である．組込みリアルタイムシステムで用いるために必要な具体的な性質は，プラットフォームとしてアプリケーションに提供するAPIの最悪実行時間が定まることや，割込み応答時間が短いこと，である．2点目は実行オーバーヘッドである．これは，実行時間の増分である組込みリアルタイムシステムにおいても，実行オーバーヘッドを抑えることが要求される場合が多い．本論文ではコア数を増やしたことによる増分に着目する．3点目は信頼性である．プラットフォームは様々なアプリケーションで利用されることを想定するため，その動作による不具合が存在すると多数のシステムに影響を与えてしまう．そのため，プラットフォームが提供する動作による不具合が発生しないことが求められる．特に，組込みリアルタイムシステムでは，製品化された後でソフトウェアの更新により不具合を修正することが困難であったり，非常に多くの費用を要したりする場合があるため，不具合のないことが重視されることが多い．不具合が発生しやすいのは，割込み処理や排他制御に関連する部分であり，マルチコアプラットフォームにおいてはデッドロックが発生しないことが重要である．デッドロックが発生するのは非常にまれなケースで，再現が困難であることが多いため，発生しないことを保証することが課題となる．以上の3つの課題を解決する，組込みリアルタイムシステムに適したマルチコアプラットフォームが必要である

本研究の目的は，組込みシステムに適したマルチコアプラットフォームを構築するため，3つの課題を解決する技術を提案することである．特に，3つの課題すべてに影響を与える，排他制御に着目する．排他制御による影響については，次の通りである．排他制御によって他のコアに処理を待たされると，APIの最悪実行時間が長くなる．他のコアの実行状態が関わるため，最悪実行時間がいくつであるかを容易に知ることができず，リアルタイム性を損なう恐れがある．他のコアに処理を待たされない場合においても，排他制御のための処理を実行する時間が長いと，実行オーバーヘッドも長くなる．さらに，排他制御はマルチコアプラットフォームの果たす大きな役割であるので，排他制御が正しく動くこと，信頼性のあることが重要である．

本研究の具体的な内容は以下の3つである．

まず，コア間の共有リソースを排他する際に用いる排他制御アルゴリズムを提案する．マルチコアプロセッサを用いたリアルタイムシステム向けの排他制御アルゴリズムとして，数多くのキューイングスピンロックアルゴリズムが提案されている．これらの多くは，単一のロックを取得する状況を想定しているため，複数のロックを同時に取得する場合には，コア数に対するスケラビリティとリアルタイ

ム性を両立できないという問題がある．本研究では，スピンロックが満たすべき要件を示し，提案アルゴリズムが全ての要件を満たすことを示す．さらに，提案アルゴリズムをハードウェアで実装することで高速に実行できる手法について述べる．

2つ目として，マルチコアプラットフォームを構成するソフトウェアのテスト効率化手法を提案する．マルチコアシステムにおいては，ソフトウェアプラットフォーム内に各コアの実行順序に依存してパスが定まる分岐（実行順序依存分岐と呼ぶ）が存在する．この分岐は，各コアが並列に実行することによって発生する不都合な状態に対応するために，ソフトウェアプラットフォームの開発者が意図的に作成したものである．分岐を網羅する方法として，実行順序依存分岐を実行する可能性のあるプログラムを繰り返し実行する手法が考えられる．しかし，特定のパスを実行する可能性が低い場合は，実行順序依存分岐を網羅するために非常に多くの繰り返し回数を要する．プログラムを繰り返し実行する手法は，テストの実施や不具合の分析に多くの手間を要するため，テストの効率が悪い．本研究では，プログラムからコアの実行を制御できる機構を用いたテスト効率化手法を提案する．テストプログラムにコアの実行を制御するための記述を追加しコアの実行を制御することで，プログラム中の特定のパスを決定的に実行し，テストの効率化を実現できる．提案手法をマルチコアに対応したリアルタイム OS である TOPPERS/FMP カーネルに適用し，提案手法の有用性を示す．

最後に，リアルタイム性と実行オーバヘッドの両立を目指した通信ミドルウェアを提案する．対象は，車載ソフトウェアアーキテクチャ仕様である AUTOSAR の通信ミドルウェアである．本研究では，まず，通信ミドルウェアをマルチコア拡張する際に求められる要件を示し，既存手法では，実行オーバヘッドとロック取得時間の低減を両立できないことと，実行並列性がないことを明らかにする．そして，これらの問題の解決を目指した手法を2つ提案し，実装および評価を行う．各手法の評価結果と要件とを比較し，各手法の特徴を明らかにする．また，既存手法は通信ミドルウェアが持っている定期通信の機能を考慮しておらず，リアルタイム性を損なわないように実装を行うと，実行オーバヘッドが大きくなってしまいう問題があることを示す．

本研究では，マルチコアプラットフォームが果たす基本的な役割である排他制御について，リアルタイム性と実行オーバヘッドを両立する手法を提案した．また，マルチコアプラットフォームについて，テストの効率化手法を提案した．以上の内容により，マルチコアプラットフォームの構築に関する3つの課題を解決する技術を示した．