C-020

組込み向けマルチコア対応ドメイン分離技術

Domain-partitioning Technology for Multi-core Embedded Systems

高橋 明生† 近藤 雄樹† 野尻 徹† 入江 直彦†
Akio Takahashi Yuki Kondoh Tohru Nojiri Naohiko Irie

1. はじめに

近年、組込みシステムでのマルチコア SoC (System-on-a-Chip) の活用が検討されており、既存ソフトウェア資産を低コストでマルチコア環境に移行する方法が求められている。これまでもハードウェア制御などの制御系機能とマルチメディア処理やネットワーク通信などの情報系機能を備えたシステムを想定し、マルチコア上で複数の OS (Operating System) を実行する方式が研究されている[1]。

この方式には、既存 OS の利用によってアプリケーションプログラムが容易に移行できるという利点があるが、従来は個別のハードウェアを使用していた情報系機能と制御系機能が同一のハードウェア資源を共有するようになるため、一方の異常動作がシステム全体に影響してしまうという課題がある。

この課題に対し、マルチコア上に複数の実行環境の相互作用を防止する技術（ドメイン分離技術）によって解決を図った。本技術を実際にマルチコア SoC に適用し、評価を行った結果、既存システムをマルチコアで同時実行する際の信頼性が確保できることを確認した。

2. マルチコアにおける複数 OS 実行の課題

制御系機能と情報系機能で構成されるシステムとして車載情報システム例に挙げる。図 1 に車載情報システムの従来構成と、マルチコアを活用した構成を示す。

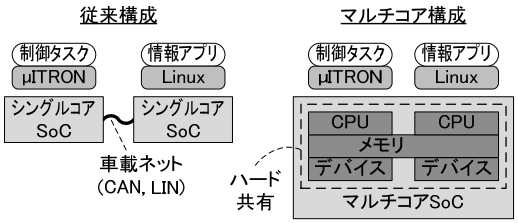

図 1　従来構成とマルチコア構成

従来構成では、制御系機能と情報系機能が別々のハードウェア上に実装されており、CAN (Controller Area Network) などの通信路で接続されていた。通信路以外には共有部分がないため、ネットワークデバイスや受信データさえ正しく扱えば、他方の影響は受けなかった。

次に従来構成をマルチコア構成に移行することを考える。従来構成で使用していた OS をマルチコア構成でも使用することにより、既存のアプリケーションプログラムをマルチコア構成でも利用することができる。しかしながら、マルチコア構成では制御系機能と情報系機能が共通のハードウェア上で動作するため、従来環境では一方のハードウェア内に限定されていた異常動作の影響範囲が、マルチコア構成ではシステム全体に拡大してしまうという課題がある。

この課題に対し、我々は OS などのソフトウェアや動作に必要なハードウェアなどシステムの構成単位を「ドメイン」と定義し、異常動作の影響をドメイン内に限定する技術によって解決を図った。我々はこれをドメイン分離技術と呼んでおり、図 2 に概要を示す。

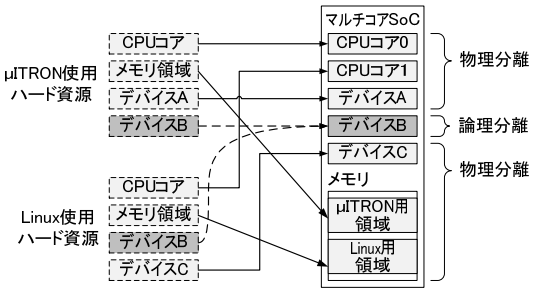

図 2　ドメイン分離技術の概要

各ドメインはそれぞれ動作に必要なハードウェア資源の集合を持ち（図の左側）、一方マルチコア SoC にもハードウェアの実体集合がある（図の右側）。実際の実行においてはドメインのハードウェア資源と実ハードウェアとの対応付けを行われるが（図の矢印線）、その対応付けによってハードウェアを次のように分類できる。

(1) 1 対 1 関係の資源：メモリや、一方のドメインのみ使用するデバイス、CPU コアである。これらの資源については他のドメインが操作できないようにする必要がある（物理分離）。

(2) 多対 1 関係の資源：ドメイン間の通信や SoC 資源活用などの目的で、ドメイン間で共有される資源である。これらの資源については、目的と資源特性に応じたドメイン分離を実施する（論理分離）。

ドメイン分離では、これらの種類に応じた分離手法を適用する。本報告では、ドメイン分離において特に(1)の資源に適用する物理分離について述べる。

3. ドメイン分離における物理分離の実現

一般に、OS およびアプリケーションプログラムの実行に必要な資源は CPU コアとメモリ、いくつかの I/O デバイスである。これらのハードウェア資源に対する物理分離の内容を表 1 に示す。

表 1　各ハードウェア資源に対する分離

項目	分離対象	干渉手段
CPU コア	実行コンテキスト	電源設定の操作や強制リセットなど
メモリ	物理領域	読み書きアクセス
I/O デバイス	レジスタ	読み書きアクセス

メモリに対しては物理アドレスで分割された領域ごとの

†株式会社日立製作所 中央研究所 組込みシステム基盤研究所

アクセス制御が必要で、I/O デバイスに対してはレジスタへのアクセス制御が必要である。CPU (Central Processing Unit) コアに対しては、WDT (Watch-dog Timer) などリセットなどによって動作を制御できるデバイスの保護が必要だが、これは結局関係デバイスのレジスタへのアクセス制御によって実現できる。

ここで、アクセス制御の方式はバスアーキテクチャに依存する部分が大きいため、本研究では同一の物理メモリ空間上にメモリとレジスタがマップされるメモリマップト I/O を想定した。このアーキテクチャでは、アクセスの対象（メモリなのか I/O デバイスのレジスタなのか）はアドレスによって区別されるため、アクセス制御は、アクセス対象アドレスとアクセス元デバイスの組に関して許可または禁止を判断し、禁止アクセスを無効化することで実現できる。

3.1. アクセス制御方式

バス上で実際にアクセスを制御する方式として、A) バスからアクセス先デバイスへ送られるアクセスを制御する方式、B) バスイニシエータからバスへ出されるアクセスを制御する方式が考えられる。図 3 に各方式でアクセス制御を行う位置を模式的に示す。

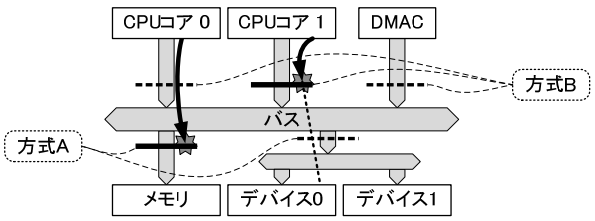

図 3　各方式におけるアクセス制御の位置

各方式とも、図に示した位置に制御ハードウェアを挿入し、アクセス元デバイスとアクセス先デバイスの組を判断してアクセス制御をする点では同様である。しかしながら、実施位置の違いにより、制御ハードウェアに設定する情報が表 2 のように異なる。

表 2　制御ハードウェアに設定が必要な情報

設定情報	方式 A	方式 B
アクセス元デバイス	必要	不要
アクセス先デバイス (アクセス先アドレス)	必要	必要

方式 B では、アクセス元が自明であるため、アクセスを許可するアクセス先デバイスの情報のみ必要である。一方、方式 A では制御ハードウェアから先に複数のアクセス制御対象が存在するため、アクセス元とアクセス先の両方の情報が必要になる。

3.2. アクセス制御ハードウェア

実際のアクセス制御は、Logical Partition Controller (LPC) という新規ハードウェアで実施することにした。

LPC はアクセス許可条件を複数記憶しており、あるアクセスについての許可または禁止をこの条件に従って判断する。アクセス許可条件にはアクセス制御対象のアドレス範囲とアクセス元デバイスを指定でき、ソフトウェアから設定可能である。

LPC は禁止と判断したアクセスについて、読み込みアクセスには 0 を返し、書き込みアクセスは破棄することで、アクセス先のメモリやデバイスを保護する。

4. 評価

4.1. アクセス制御方式比較

方式 A と方式 B との比較を表 3 に示す。ここで、N はアクセス先デバイスの数、M はアクセス元デバイスの数である。

表 3　分離に必要な LPC 数の比較

比較項目	方式 A	方式 B
N 個のデバイスの分離に必要な LPC の個数	N 個	M 個
1 個のデバイスの分離に必要な LPC の個数	1 個	M 個

物理分離を実現するために必要な LPC の数について比較すると、すべてのハードウェア資源を分離した場合は方式 A と方式 B で本質的な違いはない。しかし、方式 B ではすべてのアクセス元デバイスについてアクセス制御を実施しないと保護が成り立たないが、方式 A では保護したいハードウェア資源についてアクセス制御を実施すれば確実に保護できること、今後コア数が増加した場合には方式 A の方がスケーラビリティを確保しやすいことを鑑み、今回は方式 A を採用することにした。

4.2. 複数 OS の実行とアクセス制御の効果

方式 A での LPC を搭載したマルチコア試作チップを用いて、実際に評価した。μITRON を OS とする制御ドメインと、Linux を OS とする情報ドメインを同時に実行するシステムを構築し、各 OS の正常動作を確認した。

次に、Linux の異常動作させ、μITRON のメモリ領域を破壊し、μITRON で使用中デバイスを操作したときに μITRON の動作に与える影響を、ドメイン分離の実施環境と未実施環境で比較した。その結果、未実施の環境では μITRON がリセットしたが、実施環境ではタイマーによる定周期処理が継続できるなど深刻な影響がないことを確認した。

5. 結論

ドメインという概念でマルチコアの実行環境を分離するドメイン分離技術を開発し、本論文では特にアクセス制御ハードウェアを用いた物理分離技術について報告した。本技術によって、メモリとデバイスに対する他ドメインからの悪影響を阻止可能であり、マルチコア上に複数ドメインのシステムを構築した場合の信頼性を確保することが可能になる。

参考文献

[1] 高田 広章, 他, "機能分散マルチプロセッサ向けのリアルタイム OS", 情報処理学会誌, vol.47, No.1, pp.41-47, 2006.