Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

リアルタイム OS と汎用 OS が混在する車載 ECU における リアルタイム性評価と改善

奥原 誠[†1]　杉山 由芳[†1]　本田 晋也[†2]

概要：
近年，車載システムの高機能化が進み，制御系 ECU と情報系システムを連携する機能統合 ECU のニーズが高まっている．このニーズを実現するためには RTOS と汎用 OS の共存が不可欠である．しかしながら，制御系 ECU は信頼性，及びリアルタイム性が重要であり，汎用 OS を統合することにより，これらの性能に影響を与え，要求性能を満足できなくなることが懸念される．そこで本論文では，この機能統合 ECU を実現するための適切なハード構成，及びそれに基づくソフトウェアプラットフォーム配置を選定し，要求性能実現に向けた改善と，その効果確認を行った．この取組みにより，機能統合 ECU を実現するための目途付けを行うことができた．

キーワード：車載 ECU，仮想化技術，VMM

Evaluation and Improvement of Real-Time performance on vehicle ECU with Real-Time OS and General-Purpose OS mixedly

MAKOTO OKUHARA[†1]　YUHO SUGIYAMA[†1]　SHINYA HONDA [†2]

Abstract:
In recent years, advanced functions of in-vehicle systems have progressed, and the need for function integrated ECUs that link control ECUs and information systems is increasing. Coexistence of RTOS and general purpose OS is indispensable to realize this need. However, the reliability and real-time nature of the control system ECU are important, and it is feared that integration of a general-purpose OS will affect these performance and fail to satisfy the required performance. Therefore, in this paper, we selected an appropriate hardware configuration for realizing this function integrated ECU, and a software platform arrangement based on it, and made improvements for achieving required performance and confirm the effect. With this approach, we were able to make a proposal to realize the function integration ECU.

Keywords:　vehicle ECU, virtualization technology,　Virtual Machin Monitor

1. はじめに

近年，車載システムの高機能化が進み，高信頼性・リアルタイム性が必要な制御系 ECU と情報系システムを連携するニーズが高まっている. 例えば, カメラなどのマルチメディア機能を使用してドライバーの健康状態や居眠り運転などを判断し，即座に制御系の駆動力を最適な動作へと切り替える使用方法などが考えられる．このように制御系 ECU に情報系システムを連携した ECU を機能統合 ECU と呼ぶ.

機能統合 ECU は高信頼性・リアルタイム性が必要な制御系処理を RTOS で実行し，画像処理・知識処理などの機能性の高いマルチメディア系処理は汎用 OS で実行する必要がある. そのため RTOS と汎用 OS の共存は不可欠となる. RTOS は最小限の構成により高信頼性・リアルタイム性を実現しており, 一方で汎用 OS は複雑な構成により高機能性を実現している．このように，この２つの OS は相反する性質である．そのため高信頼性・リアルタイム性が重要である制御系 ECU に汎用 OS を統合することは RTOS の性質を保護できず性能低下が懸念される.

このような OS の共存による課題の解決方法として，汎用 PC では仮想マシンモニタ(VMM)と仮想マシン(VM)を用いて OS 間にパーティショニングを行う方法がある. 車載システムにおいても VMM の実装提案[1]がされている. 提案は主に 2 つに分かれており，1 つ目は制御系と近年のインフォテインメントやコネクティビティおよび快適機能との相互作用のニーズに対する提案である．具体的には AUTOSAR プラットホームで用いられる RTOS と汎用 OS にマイクロカーネルを用いた VMM で共存させる提案[2]や RTOS を VMM 上にゲスト OS として実装する提案である[3]．2 つ目は ECU の増加に伴うコスト上昇，搭載スペース増加に対する解決手段として，複数の制御 ECU を統合する提案である．マイクロカーネルで VMM を実現する提案[4]や既存プロセッサにソフトウェアによる準仮想化を用いて実現する提案[5]などがある.

†1 株式会社デンソーテン
DENSO TEN Limited.
†2 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University

ⓒ2019 Information Processing Society of Japan

Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

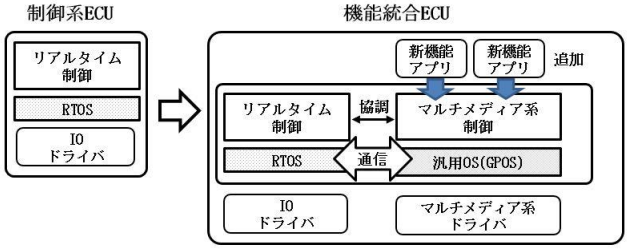

図 1　機能統合 ECU システム構成

本論文では前者の RTOS と汎用 OS の共存を対象とする．本論文の構成は次の通りである．2 章では機能統合 ECU のシステム構成を示し，システムを実現するために必要な機能要件，非機能要件を述べる．3 章では車載で実現可能なハードウェア構成及びそれに基づくソフトウェアプラットフォーム配置を列挙し，2 章で述べた要件より最適な構成を選定する．4 章では選定した構成の課題である RTOS のリアルタイム性を改善し，その改善効果を確認する．

2. 機能統合 ECU 要件

機能統合 ECU のシステム構成を示し，システムを実現するために必要な機能要件・非機能要件を述べる．

2.1 機能統合 ECU システム構成

機能統合 ECU のシステム構成を図 1 に示す．機能統合 ECU は駆動，ブレーキ制御などのリアルタイム制御とカメラや指紋認証などのマルチメディア系制御を搭載する．マルチメディア系制御については，今後の機能追加にも対応できるようにアプリケーションの追加インストールを可能とする．搭載する OS はリアルタイム制御に RTOS，マルチメディア系制御に汎用 OS を使用する．このリアルタイム制御とマルチメディア系制御は，OS 間通信で協調制御を行うことも可能な構成とする．RTOS の入出力は GPIO，AD などの IO ドライバを対象とし，汎用 OS は SDIO などのマルチメディア系ドライバを対象とする．

なお，車載制御系プロセッサはルネサスの RH850，インフィニオンの TriCore，NXP や ST マイクロの Cortex-R が主に使用されているが，これらのプロセッサでは一般的な汎用 OS である Linux は動作しない．そのため車載系でも Linux が動作可能な ARM v7/v8-A シリーズのプロセッサを前提とする．

2.2 機能要件

FR1：RTOS のメモリ保護が可能なこと

マルチメディア系制御は Web 上からアプリケーションのダウンロードが可能なものもあり，悪意のあるソフトが紛れ込む恐れがある．ハッキングによる自動車の遠隔操作の事例[6]もあり，人命に関わる制御を行う RTOS 側に，悪意のあるソフトが紛れ込むことを回避する必要がある．

FR2：RTOS 処理の実行時間が保証できること

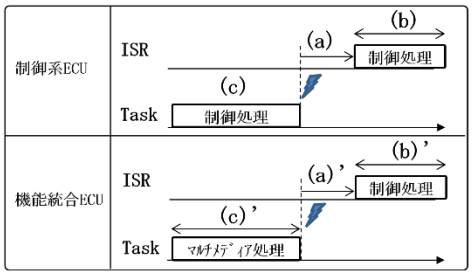

図 2　RTOS のリアルタイム性(NFR1)

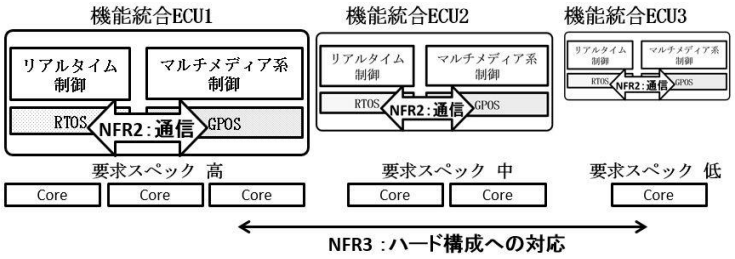

図 3　通信(NFR2)，ハード構成(NFR3)

RTOS はリアルタイムに処理を実行するため，任意の単位時間あたりに必要な CPU 使用時間を確保できる必要がある．

2.3 非機能要件

NFR1：RTOS のリアルタイム性が悪化しないこと

図 2 に示すように機能統合 ECU により RTOS の以下 2 つのリアルタイム性が悪化しないこと．

NFR1-1：RTOS の割込み応答時間

割込み処理に即時対応できるように制御系 ECU の割込み応答時間(a)が機能統合 ECU(a)’で悪化しないこと．

NFR1-2：RTOS の処理時間

制御系処理には割込み，タスク処理時間の制約があるため，制御系 ECU の処理時間(b)(c)が機能統合 ECU の処理時間(b)’(c)’で悪化しないこと．

NFR2：OS 間通信が効率的に行えること

機能統合 ECU では RTOS-汎用 OS 間で協調した制御を行うため，通信レートの高い効率的な OS 間通信を行う必要がある．これを実現するためにキャッシュを活用した OS 間通信を行うことが望ましい．

NFR3：ハードウェア構成への対応が可能なこと

図 3 のように車載システムは車種により要求スペックが異なるため，柔軟に対応できるシステムを構築する必要がある．そのためシングルコア，マルチコアどちらも対応できることが望ましい．

以上，要件について述べた．

なお RTOS，汎用 OS それぞれ制御対象ドライバが異なるためドライバに関する要件は不要とした．

3. 構成検討

車載で実現可能なハードウェア構成及びそれに基づく

ⓒ2019 Information Processing Society of Japan

Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

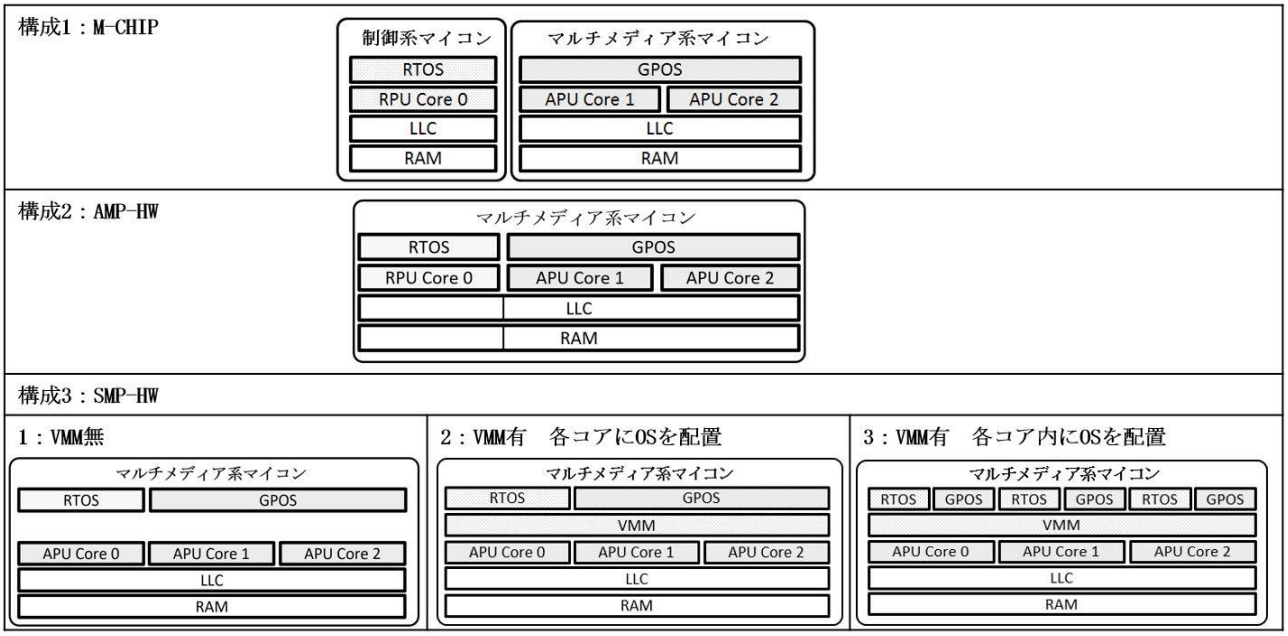

図 4　RTOS と汎用 OS を共存させるシステム構成

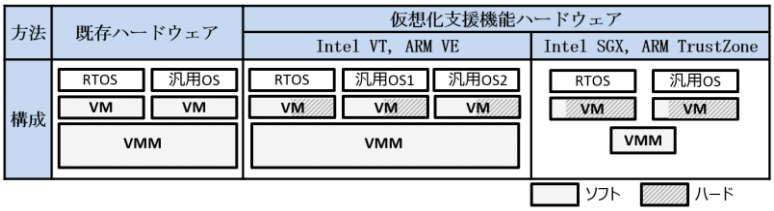

図 5　VMM 実現方法

ソフトウェアプラットフォーム配置を列挙し，要件充足を確認する．その中から車載に最適な構成を選定する．

ハードウェア構成

車載系でも主流になりつつあるマルチコアを想定した場合，ハードウェア構成は次の 3 種類を挙げることができる．

構成 1：マルチチップ(M-CHIP)
構成 2：ヘテロジニアスマルチコアプロセッサ（AMP-HW）
構成 3：対象型マルチコアプロセッサ（SMP-HW）

構成 1：M-CHIP は，リアルタイム性と低消費電力に優れた Real-time Processing Unit(RPU)を備えた制御系マイコンと高機能で演算性能の高い Application Processing Unit(APU)を備えた汎用マイコンで構成されたマルチマイコン方式であり，制御系処理とマルチメディア系処理は完全に分離されている．

構成 2：AMP-HW は，メインコアとサブコアの複数種類のプロセッサにより構成されており，APU と RPU による構成が一般的である．一部バスや外部メモリは両者のコアで共有する．ルネサスの R-CAR シリーズなどが該当する．

構成 3：SMP-HW は，複数キャッシュを共有する同一命令セットのコアにより構成されている．プログラムは基本的にどのコアでも実現可能であり，多くのハードウェアリソースは各コアで共有される．NXP の i.mx6 Quad や S32S などが該当する．

ソフトウェアプラットフォーム配置

構成 1，2 に対しては RPU に RTOS，APU に汎用 OS を配置する．構成 3 については VMM の使用を含めて以下 3 パターンを挙げることができる．

構成 3-1：(VMM 無)各コアに OS を配置
構成 3-2：(VMM 有)各コアに OS を配置
構成 3-3：(VMM 有)各コア内に OS を配置

各構成を図 4 に示す．構成要素として OS，プロセッサ APU/RPU，キャッシュ(LCC)，RAM を表す．

ここで構成 3-2，3-3 の VMM の実現方法について記す．

VMM 実現方法

図 5 に示すように VMM の実現方法として 3 つ挙げることができる．既存ハードウェアを使用する方法では，ハードウェアを VM で多重化する必要があり，ソフトウェアの負荷が大きく，実行オーバーヘッドも大きい．仮想化支援機能ハードウェアで実現する方法では，Intel の Intel VT や ARM の Virtualization Extensions(VE)を使用する方法，Intel の SGX や ARM の TrustZone[7]を使用する方法がある．Intel については，2 章のプロセッサの選定で ARM を前提としているため除外する．ARM の VE と TrustZone は，OS の搭載数

ⓒ2019 Information Processing Society of Japan

Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

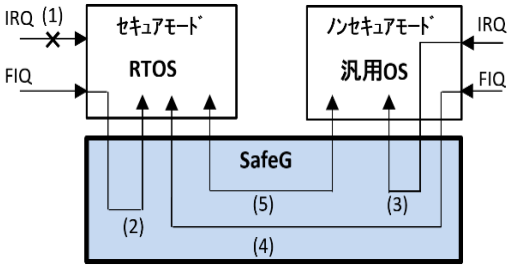

図 6 割込みハンドリング

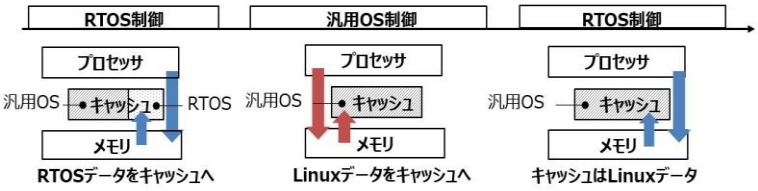

図 7 キャッシュ共有による影響

が複数か 2 つまでの違いがある．VE では複数 OS を動作させるために割込みの仮想化，ページテーブルを多重化する必要があり，リアルタイム性が悪化する．一方で TrustZone は 2 つの OS 動作前提のため，ソフト負荷も小さくリアルタイム性が高い．機能統合 ECU は RTOS と汎用 OS の 2 つまでが動作するシステム構成のため，TrustZone が有効と言える．VMM はこの TrustZone 方式の SafeG[8]を使用する前提とし，構成 3-2, 3-3 が対象となる．

各構成に対して要件充足を確認する．

3.1 構成 1：各 OS を M-CHIP に配置

全てのリソースが物理的に異なるため，OS も完全分離している．そのため機能要件は実現できる．

一方で非機能要件については，外部バスを使用するため OS 間通信の実行効率は悪い(NFR2)．またマルチチップのため，ハードウェア構成が限定され，コスト効率が悪い(NFR3)．

3.2 構成 2：各 OS を AMP-HW に配置

各 OS は異なるプロセッサを使用するため，共有するハードウェア資源が少なく干渉を受けにくい．そのため機能要件は実現できる．一方で非機能要件については，OS 間でキャッシュを共有できないため，OS 間通信の実行効率は悪い(NFR2)．またプロセッサは APU，RPU を必要とするため構成 1 と同様，ハードウェア構成が限定され，コスト効率が悪い(NFR3)．

3.3 構成 3-1：SMP-HW(VMM 無)各コアに OS を配置

OS 間で RAM を共有しているが，保護する機能がないため機能要件を満たすことができない(FR1)．

3.4 構成 3-2：SMP-HW(VMM 有)各コアに OS を配置

VMM を使用することにより RTOS のメモリ保護を実現できる(FR1)．具体的には，SafeG では TrustZone を使用し，RTOS が用いるメモリをセキュアモードに，汎用 OS が用いるメモリをノンセキュアモードに配置している．これにより，汎用 OS から RTOS へのメモリアクセスは不可能となる．メモリアクセスが必要な場合は SafeG の API 経由で可能である．RTOS の実行時間保証は OS 搭載コアが異なるため実現可能である(FR2)．非機能要件については，VMM によるオーバーヘッドによりリアルタイム性が悪化する恐れがある(NFR1)．OS 間通信は多くの場合，仮想化支援機能ハードウェアでコヒーレントの切り替えがサポートされているため，OS 間のキャッシュの利用が可能であり実行効率が高い(NFR2)．一方でマルチコア前提のためハードウェア構成は限られておりコスト効率が悪い(NFR3)．

3.5 構成 3-3：SMP-HW(VMM 有)各コア内に OS を配置

コア内で汎用 OS と RTOS を共存させる．構成 3-2 と同様，VMM を使用しているため RTOS のメモリ保護は実現可能である(FR1)．この構成はコア内で OS を共有するため，同一コア内で OS を切替えて使用する必要がある．SafeG では ARM の高速割込み FIQ と通常割込み IRQ にそれぞれ FIQ に RTOS を，IRQ に汎用 OS を割当てる．FIQ は IRQ より割込み優先度が高く，またハードウェア的に割込み禁止設定は不可能となっている．そのため，図 6 に示すように RTOS の割込み処理実行中は汎用 OS の割込みは受付けない．また RTOS のタスク実行中も IRQ を常に禁止することにより受付けない(図 6(1))．よって RTOS がアイドル状態時のみ汎用 OS が動作することになる．各 OS 実行中に同 OS の割込みが発生した場合は優先度によって割込みが切替わる(図 6(2), (3))．汎用 OS の処理中に RTOS の割込みが発生した場合は即 RTOS に切替わる(図 6(4))．各 OS 実行中に別の OS に切替える場合は SafeG の API を使用する(図 6(5))．このように SafeG は RTOS を優先する構成のため RTOS の時間保証が可能である(FR2)．非機能要件の RTOS のリアルタイム性については，参考文献[9]より RTOS と汎用 OS のキャッシュ共有により RTOS 側のタスク処理時間に影響があると想定される．これは図 7 のようにキャッシュの共有により汎用 OS がキャッシュをフルに使用してしまうため，RTOS に切替わる毎にキャッシュがリフレッシュされている状態となる．よって RTOS はキャッシュミスヒットを繰り返し，RTOS の処理時間に影響を与えていると考えられる．この対策は SafeG に折込まれていないため改善が必要とある(NFR1)．

OS 間通信については構成 3-2 と同様，OS 間でキャッシュの使用が可能であり，実行効率が高い(NFR2)．またシングルコア，マルチコアどちらのハードウェア構成に対応可能でありコスト効率も高い(NFR3)．

ⓒ2019 Information Processing Society of Japan

Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

表 1　システム構成比較

	FR1:RTOSの保護	FR2:RTOS処理の時間保証	NFR1：RTOSのリアルタイム性	NFR2:OS間通信の実行効率	NFR3:ハード構成対応
構成1:M-CHIP	○	○	○	×	×
構成2:AMP-HW	○	○	○	△	×
構成3-1:SMP-HW	×				
構成3-2:↑各コアOS配置	○	○	△	○	△
構成3-3:↑各コア内OS配置	○	○	△	○	○

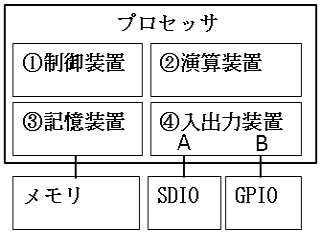

図 8 プロセッサ装置

3.6 機能統合 ECU に対するシステム構成の選定

システム構成毎の要件充足結果を表 1 にまとめた．機能要件では，構成 3-1 が FR1 を実現できない．非機能要件で比較すると要件を最も満たしているのは構成 3-3 となる．RTOS のリアルタイム性(NFR1)に課題はあるが， 参考文献[9]よりキャッシュロックダウンによる改善が見込まれるため，この機構を SafeG に織り込み効果を確認する．

＜キャッシュロックダウン機構＞

キャッシュロックダウン機構とは指定したキャッシュ領域のデータを固定化することができる.これにより,その領域はキャッシュの再割り当てが禁止され,固定化されたデータに対するアクセスは常にキャッシュヒットとなる．RTOS のキャッシュ領域を固定化することにより,RTOS のキャッシュミスヒットによる処理時間の改善が期待できる．

4. 評価

キャッシュロックダウン機構を使用する前後での RTOS タスク処理時間の効果確認を行う．これは図 8 プロセッサ装置の③記憶装置に対する対策となるが,RTOS のリアルタイム性に影響を与えるプロセッサ装置が他にないか，①制御装置②演算装置④入出力装置も併せて確認を行う.なお,評価の第一段階として今回はシングルコアを対象とする．

4.1 評価項目

・評価１．RTOS 処理時間比較

・評価 2．プロセッサ装置の処理時間影響調査

4.2 評価環境

NXP 製 i.mx6 Solo を用いて評価を行った．i.mx6 は ARM Cortex-A9 のシングルコアである．

i.mx6 の構成として，L1 キャッシュは命令，データそれぞれコア毎に 32kByte 搭載，L2 キャッシュは 1Mbyte，メモリは外付けメモリで DDR3 経由となる．

OS について，RTOS は TOPPERS 製 FMP 1.4.0 を使用し，汎用 OS は Linux NXP 公式リポジトリ版 imx-3.10.53-1.1.0ga を使用，SafeG は Ver1.2.4 を使用した．構成を図 9 に示す．

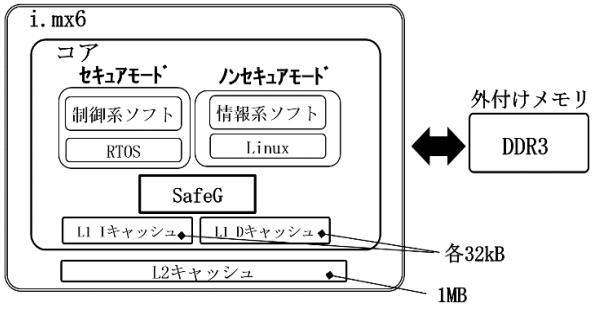

図 9　評価環境

4.3 評価方法

各評価項目について評価方法を示す．

4.3.1 評価 1．RTOS 処理時間比較

まず汎用 OS 側の処理でキャッシュをフルに使用する．その後に RTOS の処理に切替え,RTOS の処理時間を測定する．RTOS 単独で実行した場合の RTOS 処理時間は 50μｓである．汎用 OS から RTOS へ処理を切替えることにより，キャッシュロックダウン機構折込み前後での改善時間を確認する．

4.3.1.1. 汎用 OS 処理実装内容

連続したアドレスへ R/W を繰り返す処理を実装する．L2 キャッシュは 1Mbyte あるため，1MByte のメモリ空間に対して R/W を繰り返し行うことにより,汎用 OS 側でキャッシュをフルに使用する．

4.3.1.2. RTOS 処理実装内容

汎用 OS と同様,連続したアドレスへ R/W を繰り返すソフトを時間周期タスクに実装する.時間周期タスクの実行後,システムコールを呼び出すタスクを実行し,汎用 OS に切替える．その後，再びタイマから割込みが発生すると RTOS に切替わり，時間周期タスクが実行される．

4.3.1.3. キャッシュロックダウン機構実装内容

汎用 OS と RTOS で使用するキャッシュ領域を分割するために,L2 キャッシュのキャッシュロックダウン機構を使用する．汎用 OS に切替わるときは，RTOS で使用しているキャッシュ領域をロックし，汎用 OS によって RTOS のデータが追い出されることを防ぐ．RTOS に切替わるときは，RTOS のデータがロックするキャッシュ領域外に配置されることを防ぐ．

4.3.2 評価 2．プロセッサ装置の処理時間影響調査

図 8 の①制御装置②演算装置④入出力装置を評価対象とする．④入出力装置は RTOS と汎用 OS で対象が異なるため RTOS は GPIO，汎用 OS はマルチメディア系 IO とする．また

ⓒ2019 Information Processing Society of Japan

Vol.2019-ARC-235 No.5
Vol.2019-SLDM-187 No.5
Vol.2019-EMB-50 No.5
2019/3/17

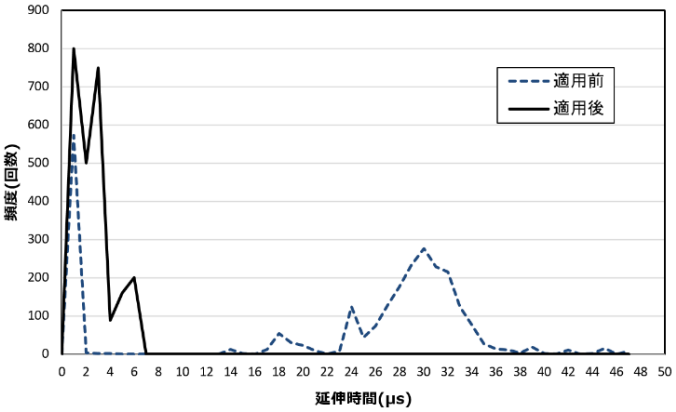

図 10　評価 1.RTOS 処理時間比較

汎用 OS は RTOS への影響が少ない⑤処理なしを用意する．①～④の各制御装置を動作させる方法として①制御装置は分岐処理を繰り返し行うことによりパイプライン，予測制御といった制御装置を動作させた．②演算装置は四則演算を行うことにより動作させた．①②は極力テンポラリで処理を行い，③記憶装置を動作させないようにした．③記憶装置は異なるアドレスへの読み書きを繰り返し行うことにより動作させた．④入出力装置では，汎用 OS は SD カードの読み書きを実施することにより，RTOS は GPIO で HI/LO を繰り返すことにより動作させた．この処理を汎用 OS で①～⑤実施後に OS を切替えて，RTOS で①～④を実施する． 全 20 パターンの制御装置動作時の RTOS が①～④を実施した時の処理時間延伸分を確認する．RTOS の処理時間は①～④すべて 50μｓとする．

4.4 評価結果

4.4.1 評価 1. RTOS 処理時間比較

評価結果を図 10 に示す．キャッシュロックダウン機構折込み前はリアルタイム処理時間の延伸が最大で 47μｓあったが，折込み後は最大 7μｓの延伸と大幅に改善できた．実際に 4000 回の RAM アクセスに対してキャッシュミスヒットが 400 回発生していたところが，キャッシュロックダウン機構を使用することにより 30 回まで改善されており，RTOS が L2 キャッシュを有効に使用できていることが確認できた．

4.4.2 評価 2．プロセッサ装置の処理時間影響調査

評価結果を図 11 に示す．RTOS の延伸時間が大きく発生するのは汎用 OS の制御装置に関わらず，RTOS が③記憶装置を使用している場合となった．汎用 OS が⑤処理なし後に RTOS③記憶装置処理により，処理時間の延伸が発生しているのは，検査用ソフトに関わらず汎用 OS 自身が記憶装置を使用したため，RTOS 側が影響を受けたと想定できる．次に延伸時間が長い入出力装置に関してはバス競合によるものと想定できる．

評価 1 より，キャッシュロックダウン機構を使用することにより RTOS のリアルタイム性が大幅に改善でき，評価 2 より RTOS と汎用 OS の共存により大きく影響を受けるプロセッサ装置は記憶装置であることがわかった．この 2 つの評価より，リアルタイム性の改善に関係する装置は記憶装置であり，記憶装置を扱うキャッシュロックダウン機構は有効な手段と言える．

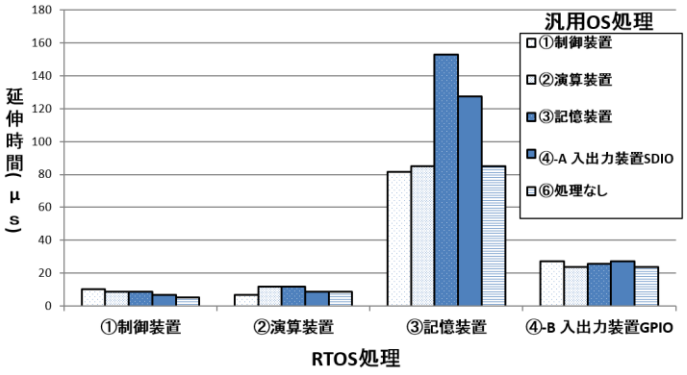

図 11　評価 2.プロセッサ装置の処理時間

5. おわりに

本研究では機能統合 ECU に必要な RTOS と汎用 OS を共存させる為の最適な構成選定と性能改善・評価を行った．この取組みにより，機能統合 ECU の実現目途付けを行うことができた．今後は ECU に実装して製品評価を行っていく予定である．また今回は 32bit 版 ARM シングルコアプロセッサを評価対象としたが 64bit 版 ARM マルチコアプロセッサ対応 SafeG を開発したので，併せて評価を行っていく予定である．

参考文献

[1] Heiser, Gernot. "Virtualizing embedded systems: why bother?〃 Proceedings of the 48th Design Automation Conference. ACM, 2011

[2] Hergenhan, Andre, and Gernot Heiser. 〃Operating systems technology for converged ECUs.〃 6th Emb. Security in Cars Conf(escar). Hamburg, Germany: ISITS. 2008.

[3] 渡邉和樹，永島力，茂田井寛隆，片山吉章，毛利公一:Xen における PCI Passthrough の性能評価 SIG Technical Report Vol31-8,2010-01-20

[4] D. Haworth, “An AUTOSAR-compatible microkernel for systems with safety-relevant components”, Informatik aktuell, Volume “Herausforderungen durch Echtzeitbetrieb”, Pages 11-20, 2012

[5] Reinhardt, Dominik, and Gary Morgan. "An embedded hypervisor for safety-relevant automotive E/E-systems." Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014). IEEE, 2014.

[6] https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[7] https://developer.arm.com/technologies/trustzone

[8] 中嶋健一郎，本田晋也，手嶋茂晴，高田広章:セキリティ支援ハードウェアによるハイブリッド OS システムの高信頼性 The IEICE Transactions on Information Systems, 93(2):7585, 2010-02-01

[9]北原裕，本田晋也，高田広章 :組込みマルチコアシステムにおけるリアルタイム性保証機構の評価 2017-SLDM-179(42),1-6 (2017-03-02) , 2188-8639

ⓒ2019 Information Processing Society of Japan