3G-04

# 安全プロセッサのためのリアルタイムオペレーティングシステムの提案

斎藤 弘樹 †　　北道 淳司 ‡

† 会津大学コンピュータ理工学部　　‡ 会津大学大学院コンピュータ理工学研究科

## 1 はじめに

組込みシステムは, 多くのリアルタイムアプリケーションから構成される車載機器, 電力供給システムあるいは医療機器など, 重要な社会インフラを支えている. 技術発展に伴い様々な外部機器への対応, アプリケーションの並列処理の要求など, 高度かつ複雑なシステムにおける故障あるいは誤動作は, 人命に関与したり, 経済損出の原因になる. 信頼性向上のために, 搭載されるプロセッサをロックステッププロセッサや Triple modular redundant(TMR) にするなどの fault-tolerant design[1], 故障検知機能や故障診断機能を用いた解析手法 [2] が提案されている. Real Time Operating Sytem(以下 RTOS) の利用も考えられる. RTOS を用いた開発は, タスクの同期やリソースの管理, 時間管理などが容易であり, ソフトウエアの再利用性も高く開発期間の短縮が可能である. 現在, 多くの RTOS があり, μ C-OS/II, μ TKernel などが知られている. しかし, RTOS を用いた対故障機能の実装に関する研究あるいは実装については十分であるとはいえない.

本稿では, 安全性の高いプロセッサのための高信頼 RTOS を提案する. 既存 RTOS をベースに, ターゲットプロセッサのエラー検知機能やメモリ保護機能などの故障検知機能に対応させる. ターゲットプロセッサは, テキサスインスツルメンツ社 (以下 Ti 社) の TMS570LS3137[3] とした. これは, Arm Cortex-r4f プロセッサを搭載し, ECC や Memory Protection Unit(MPU) によるメモリ保護, CPU の故障を検知するモジュールなどの安全性のための機能を有する. ベース RTOS は, 高信頼性システムのための TOPPERS/HRP2 kernel[4](以下 HRP2) を採用した. この RTOS では, 採用するプロセッサが有する故障検知機能の実装などが行われていない. 本研究では, HRP2 に対して, ターゲットプロセッサのための機能を実装する.

## 2 TOPPERS/HRP2 とターゲットプロセッサ

Toyohashi OPen Platform for Embedded Real-time Systems(TOPPERS)/High Reliable system Profile 2(HRP2) kernel は, 信頼性や安全性の要求が高いシステム向けのものであり, TOPPERS/ASP kernel に対して, メモリ保護機能, オーバランハンドラ機能などが強化されている.

メモリ保護機能は, 各メモリオブジェクトについて, 読み書き実行の操作が, どの保護ドメインから許可されるのかを管理し, 不正メモリアクセスを検出する.

オーバランハンドラ機能は, タスクが使用したプロセッサ時間が, 指定時間を越えた場合に起動され, 時間を超過していることを通知する. この機能により, 時間制約の厳しいタスクに対して制約を守れているかどうかをユーザが検知することができる.

拡張サービス保護機能は, 下位のタスクから上位のタスクを呼び出す場合, アクセス権限に応じて, 実行するか, 不正エラーを返すかを判断する. これを, ハードウエアを用いずに管理できる.

ターゲットマイクロプロセッサは, 高安全性かつリアルタイム制約の厳しいアプリケーション向けに開発されている. CPU を二つ持ち, それぞれの実行結果を比較し, 演算結果の一致, 不一致を判断し, CPU の故障を検知でき, メモリ保護機能ユニットなどリアルタイムアプリケーション向けの機能を持つ. プロセッサは,2 つの CPU, データの比較モジュール (CCM-R4F) などから構成されている (図 1). FIFO によって 2 クロック時間を違えて実行された演算結果は CCM-R4F において比較され, 不一致だった場合, 比較エラー (cpu コンペアエラー) を出力し, CPU の故障を検知できる.

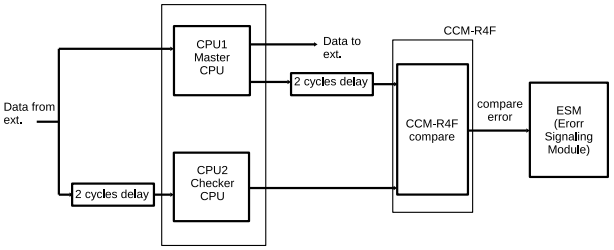


図 1: ターゲットプロセッサの構成の概要

メモリ保護ユニットは, 外部メモリに対して, 最大 12 個の領域のアクセス保護を行う. この機能は, システム制御コプロセッサのレジスタを用いて実現される. 提案 RTOS では, コプロセッサに対する処理を実装し, メモリ保護機能を実現する.

Real Time Interrupt Module は, タイマ機能を提供し, リアルタイムスケジューリングを可能としている. OS

Proposal of a Real Time Operating System for a Safety Processor
†Hiroki SAITO　‡Junji KITAMICHI
†School of Computer Science and Engineering, The University of Aizu
‡Graduate School of Computer Science and Engineering, The University of Aizu

用のティックを生成するために, 構成可能なコンペアレジスタを 4 つ有し, 個別の割り込みを実行する. 提案 RTOS では, タイマ機能とオーバランハンドラ機能のために, 割り込みを 2 つ使用する実装を追加する.

Error Signaling Module(ESM) は, 128 個のエラーチャンネルを, 3 つの重要度に分類して管理する. グループ 1 は, 重要度の低いチャンネルから構成され, ペリフェラルのパリティチェックや, メモリの修正可能なエラーなどが分類される. グループ 2 は, cpu コンペアエラーやメモリの修正不可能エラーなどから構成され, NMI を発生させる. グループ 3 は, ECC メモリの修正不可能エラーやプログラマブルヒューズのロードエラーなどである. 外部割り込みは発生せず, エラー出力端子を low にする.

Vectored Interrupt Manager(VIM) は, 95 個の割り込みチャンネルに対して, 通常割り込みおよび高速割り込みについて, ユーザ指定の優先度にて管理できる. 信頼性のために, ソフトウエアエラーに対してパリティ保護された割り込みテーブルを持つ. 提案 RTOS にこのモジュールを扱う機能を組み込むことにより, 故障検知機能, タイマ, シリアル通信の割り込みなどを可能にする.

ターゲットプロセッサを搭載している HERCULES Development Kit ボードを用い, RTOS を動作させる. 本ボードは, Integrated USB JTAG Emulator, 8MB SDRAM などの周辺モジュールと,SD カード, CAN, Ethernet などのインターフェースを有する.

## 3 提案 RTOS

使用した RTOS の開発環境について述べる. ターゲットプロセッサは, big endian を採用している. Ti 社からは, 開発環境 Code Composer Studio が提供されているが, HRP2 開発に必要な Arm Cortex-r4f CPU big endian 用の gcc complier は含まれていない. Arm Cortex-r4f CPU big endian 用の GCC 開発環境の作成を行った (gcc 4.9, binutils 2.24 など). gcc を build する際に, multilib の生成方法を変更するなどの必要があった.

HRP2 に対して拡張する機能として, スタートアップルーチン, 故障検知機能, ターゲット依存の機能などを作成した.

スタートアップルーチンでは, CPU レジスタ, メモリおよび PLL の初期化ならびに各モジュールのセルフテストエラーチェックを行う.

故障検知機能は, ESM の機能を有効にし, 実装を行った. ESM は各ペリフェラルのエラー状態を管理し, 各エラーの重要度 (優先度) に応じて処理を行わせる. 今回用いるボードではエラー出力端子は LED に接続されており, 目視による確認が行える. 一般には, エラー処理を行う外部モジュールに接続され, CPU を用いずにそのモジュールによるエラー処理が行われる. API として, 故障検知機能の初期化のための error_detection_init, 低優先故障検知の動作開始のための low_error_detection_start など, 現時点で 8 種の API を実装した. 低優先度および高優先度故障検知は ESM の管理するそれぞれグループ 1, 2 に対応し, VIM が管理する. グループ 3 に関しては割り込みハンドラ API の開発は行わない.

ターゲット依存の機能として以下を作成した. メモリ保護機能の実現のために, システム制御コプロセッサのレジスタを用いた実装を行った. 8 つの管理領域を使用し, タスクごとに開始アドレス, 属性, サイズの設定を行う. そのため, 各設定情報を保持するコンテキストブロックを用意した. Real Time Interrupt(RTI) module のコンペアレジスタを使用することで, RTOS のタイマ機能及びオーバランハンドラ機能を利用可能とした. ベースとなる RTOS ではシリアル通信の実装が必要である. これは Serial Communication Interface (SCI) module を利用して実現した. RTI および SCI における割り込みに対するハンドラは, VIM によって管理される.

## 4 まとめ

本稿では, 高信頼, 高安全のために開発されたプロセッサのための RTOS を提案した. 提案 RTOS は, ターゲットプロセッサ独自機能を活用するために, TOPPERS/HRP2 を拡張したものである. 今後, 開発した RTOS を搭載したプロセッサを, 異常に対する動作やコスト , タスク管理の性能を評価することによって, 信頼性の高いシステムが構築できることを示したい.

## 参考文献

[1] M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni- Vincentelli, M. Peri and S. Pezzini, “Fault-Tolerant Platforms for Automotive Safety-Critical Applications,” Proceedings of CASES’03, pp.170–177, 2003.

[2] R. Isermann, and P. Balle, “Trends in the application of model-based fault detection and diagnosis of technical processes,” Contrrol enginnering practice, Vol. 5, No. 5, pp. 709–719, 1997.

[3] TEXAS INSTRUMENTS, “TMS570LS31x/21x 16/32-Bit RISC Flash Microcontroller Technical Reference Manual,” http://www.tij.co.jp/jp/lit/ug/spnu499b/spnu499b.pdf, 2013.

[4] TOPPERS Project, “TOPPERS New Generation Kernel Specification,” https://www.toppers.jp/docs/tech/ngkispec-171.pdf(in Japanese), 2015.