

RTOS 利用システムのフルハードウェア化における イベントフラグの実装

Design of Eventflags for Full Hardware Implementation of RTOS-Based Systems

中原 正樹¹
Masaki Nakahara

石浦 菜岐佐²
Nagisa Ishiura

関西学院大学 大学院理工学研究科¹
Graduate School of Science and Technology, Kwansei Gakuin Univ.

関西学院大学 工学部²
School of Engineering, Kwansei Gakuin Univ.

1 はじめに

RTOS を用いたシステムの応答性能を向上させる手法として、タスク/ハンドラ及び RTOS が提供する機能の全てをハードウェア化する手法が提案されている [1] [2]. 本稿では、文献 [2] のハードウェア構成において同期・通信機能の一つであるイベントフラグを実装する。

2 イベントフラグとハードウェア構成

イベントフラグはタスク間の同期機能を提供する。タスクは 16 ビット以上のフラグに任意のビットパターンを指定して待ち、そのパターンにマッチするビットがセットされると待ちが解除される。

本稿で前提とする文献 [2] のハードウェア構成を図 1 に示す。T0~T3 はタスクをハードウェア化したもので、実行可能状態のタスクは全て並列に実行される。Manager は RTOS の機能をハードウェア化したものであり、S0~S3 は RTOS のサービスを実行するサービスモジュールである。STATUS レジスタにはタスクの状態と優先度が記憶されている。タスクはレジスタ t_i _regs の T_iF と $T_iA[4]$ にサービス番号と引数をセットすることでサービスを要求する。一度に実行できるサービスは 1 つとされているため、Request Arbiter (RA) は優先度最高のタスクからの要求を x _regs に書き込んでサービスモジュールを起動する。サービスモジュールは要求された処理を終えると x _regs に戻り値を書き込み、タスクがその戻り値を t_i _regs を介して読み込むとタスクに制御が戻る。本稿で実装するイベントフラグはサービスモジュールの 1 つである。

3 イベントフラグのハードウェア実装

本稿で実装するイベントフラグモジュールは、複数の 32 ビットフラグを 1 モジュールで扱う。

タスクがフラグ待ちを要求した場合、指定したビットパターンがフラグにセットされていれば、正常終了を表す戻り値を返し、 T_iF の要求をクリアしてタスクに制御を戻す。セットされていなかった場合には、STATUS を書き換えて当該タスクを待ち状態にする。この際、フラグを待つタスクはキューで管理するのではなく、 T_iF の要求をクリアしないことによって RA での処理待ち状態とする。

タスクがフラグのセットを要求すると、フラグを書き換えると共に待ち解除処理を行う。この際、待ち解除処理の制御はサービスモジュールではなく RA が行う。サービスモジュールは RA に対して当該フラグを待っている全てのタスクの処理待ちを一旦解除するように指示する。これにより、優先度の高いタスクから順にフラグ待ちの要求がサービスモジュールに再送されて来るため、

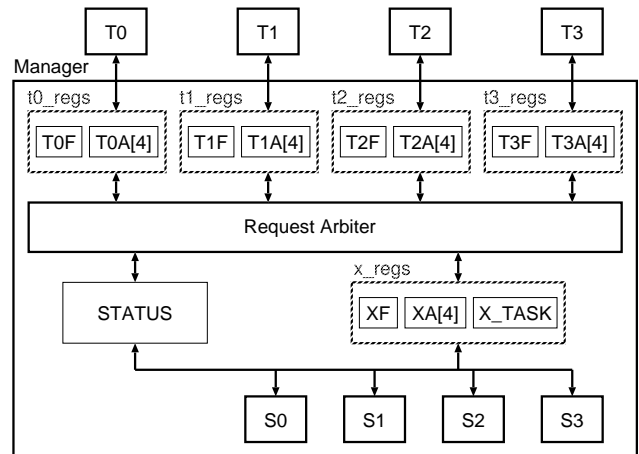


図 1 前提とするハードウェア構成 [2]

ビットパターンにマッチするもののみ待ちを解除し、そうでないものは再び RA での処理待ちにする。フラグのセットを要求したタスクには待ち解除処理を待たずに正常終了の戻り値を返して制御を戻す。

4 実験結果

本稿のイベントフラグモジュールを Verilog HDL で設計し、動作確認を行った。表 1 は優先度最大のタスクがサービスを要求してから戻り値を受け取るまでのサイクル数である。Xilinx Vivado Artix-7 をターゲットに Vivado 2016.4 を用いて論理合成を行った結果を表 2 に示す。LUT 数 135, FF 数 64, クリティカルパス遅延 6.292ns となった。

表 1: 実行サイクル数

service call	#cycle
フラグのセット	6
フラグ待ち	6

表 2: 合成結果

#LUT	#FF	delay (ns)
135	64	6.292

5 むすび

本稿では、文献 [2] のハードウェア構成におけるイベントフラグのハードウェア実装を提案した。到着順の待ち解除の実装が今後の課題である。

謝辞 本研究は一部 JSPS 科研費 19H04081 の助成による。

参考文献

- [1] Y. Oosako, N. Ishiura, H. Tomiyama, and H. Kanbara: "Synthesis of full hardware implementation of RTOS-based systems," in *Proc. RSP 2018*, pp. 1-7 (Oct. 2018).
- [2] 六車, 石浦, 安堂, 富山, 神原: "RTOS 利用システムのフルハードウェア化におけるサービス処理機能の集約," 信学技報, VLD2020-75 (Mar. 2021).