

オンチップリアルタイム OS の設計評価

5 Z - 7

岩崎 裕江 長沼 次郎 遠藤 真
NTT ヒューマンインタフェース研究所

1. はじめに

コアCPUを中心として、メモリ等の機能部品と専用ハードウェアをオンチップに集積化したエンベディッドシステムLSIの開発が盛んである。このようなLSIの開発では、ハードウェア開発とともに、それを制御するファームウェアの開発が重要である。MPEG2 CODECシステムのようなリアルタイム性が要求されるシステムにおいては、従来のファームウェア開発では、ハードウェアの様々なタイミングを制御するためファームウェアが複雑化し、ファームウェア開発/保守の短TAT化が図れないという問題があった。

一方、リアルタイム性の要求があるシステムでは、ソフトウェアの複雑化を解消するために、リアルタイムOS(RTOS)[1][2]を組み込み、マルチタスク環境で開発を容易に行っている。しかし、市販されているRTOSは、様々な機能を提供しているが、多くのメモリ量を必要とするため、エンベディッドシステムLSIに適用することは困難である。これに対し、我々は、使用メモリ量の低減化を図ったオンチップRTOSの検討[3]を行ってきた。

本稿では、これらの検討を踏まえて、エンベディッドシステムLSIのファームウェアに適したオンチップRTOSを開発し、MPEG2多重(MUX)LSI[4]のファームウェアに適用して、動作確認と性能評価を行ったので報告する。

2. オンチップRTOSの設計

本オンチップRTOSは、エンベディッドシステムLSIを制御するファームウェア開発に不可欠な機能として、タスク管理、割込管理、セマフォ管理に限定し、使用メモリ量の低減化を図っている。本オンチップRTOSが提供するシステムコールを表1に示し、以下に各機能の概要を述べる。

2. 1 タスク管理

複数の独立した入力を監視するために、周期的に起動される周期タスクを実現する。タスク管理では、タスクの状態遷移を実現するため、タスク/周期タスクを生成(TSK_create/TSK_cycliccreate)、起動(TSK_start/TSK_cyclicstart)、終了(TSK_exit/TSK_cyclicexit)、削除(TSK_delete/TSK_cyclicdelete)するシステムコールを提供する。

タスクスケジューリング方法は、タスク生成時に付与される優先度に従い、優先度の高いタスクから順番に実行するスケジューリング方式[1]を用いる。また、実行中に優先度の高いタスクが実行待ちになった場合には、実行中のタスクをプリエンプション(中断)し、優先度の高いタスクを実行する。

表1 システムコール一覧

項目	システムコール名	機能
初期設定	INI_kernel	カーネル初期設定
タスク管理	TSK_create	タスク生成
	TSK_start	タスク起動
	TSK_exit	タスク終了
	TSK_delete	タスク削除
	TSK_cycliccreate	周期タスク生成
	TSK_cyclicstart	周期タスク起動
	TSK_cyclicexit	周期タスク終了
	TSK_cyclicdelete	周期タスク削除
	TSK_abort	他タスク強制終了
	TSK_cyclicabort	他周期タスク強制終了
TSK_idget	自タスクid情報取得	
割込管理	INT_define	割込ハンドラ定義
	INT_cancel	割込ハンドラ定義解除
	INT_exit	割込終了
セマフォ管理	SEM_create	セマフォ生成
	SEM_operation	セマフォ操作
	SEM_delete	セマフォ削除

2. 2 割込管理

リアルタイム性の厳しい処理を行ったり、異常処理を行うために割込管理機能を実現する。割込管理では、割込ハンドラ定義(INT_define)、割込ハンドラ定義解除(INT_cancel)、割込終了(INT_exit)のシステムコールを提供する。

2. 3 セマフォ管理

オンチップ上で限られた資源を有効に利用するため、資源管理が特に重要である。このため、タスク間通信と、資源の排他制御をするセマフォ管理機能を実現する。セマフォ管理では、セマフォ生成(SEM_create)、セマフォ操作(SEM_operation)、セマフォ削除(SEM_delete)のシステムコールを提供する。セマフォの資源数の操作は、すべてセマフォ操作システムコールで行う。また、セマフォの資源数は、セマフォ生成時に指定可能とする。

3. オンチップRTOSの評価

本オンチップRTOSのフィージビリティを確認し、性能評価を行うため、リアルタイム通信のプロトコル処理を行うMPEG2多重(MUX)LSI[4]のファームウェア[5]を本オンチップRTOSと複数のタスクで構成した。

3.1 MPEG2多重(MUX) LSIへの適用

MUX LSIシステムにオンチップRTOSを導入した場合のシステム構成を図1に示す。MUX LSIのファームウェアは、Video、Audio、User、PCR、PSIなどの処理単位で、独立したタスクに分割できる。設計評価用に、Video、PCR、PSIの処理をオンチップRTOS上で動作させた。各タスクの優先度とプログラム規模を表2に示す。時刻管理を行うPCR処理は、他の処理より速い応答性が必要であるため、PCRタスクに最も高い優先度を与える。その結果、他の処理のタイミングに依存せず、PCR処理を優先させることができた。各タスクは、依存関係がなく独立な処理として記述できるため、ファームウェア開発が容易になる。

3.2 実験環境と動作確認

仮想LSI環境[6]上で、本ファームウェアの動作確認を行った。伝送レート、ビデオ符号化レートを6Mbit/s、5Mbit/s、PCR周期、PSI周期を100ms、1sに設定した。図2に本RTOS上のタスクの動作状態を示す。最も優先度の高いPCRタスクが、Video監視タスクなどPCRより低い優先度のタスクをプリエンプションして、優先的に実行されている。また、出力のビットストリームを解析した結果、PCR/PSI周期が守られ、Videoデータが所望の割合で出力されていることを確認した。

3.3 メモリ量の評価

本RTOSが必要とする動的な使用メモリ量は、アプリケーションのタスク数に依存する。表3に本RTOSのデータ領域とスタック領域に使用したメモリ量を示す。また、MUX LSIアプリケーションのデータ領域とスタック領域についても表3に示す。本RTOS部分の動的な使用メモリ量が約2Kbyte程度で実現でき、MUXファームウェア部分が約1.4Kbyteで実現できた。本RTOSと複数のタスクで構成したファームウェアで、4Kbyte以下のメモリ量で実現することができた。

4. おわりに

エンベディッドシステム LSI用のオンチップRTOSを開発した。MPEG2 MUX LSIファームウェアに、本オンチップRTOSを適用し、動作確認と性能評価を行った結果、動的な使用メモリ量を約2Kbyteで実現することができた。今後、本オンチップRTOSをエンベディッドシステム LSIの中核として、マルチメディアアプリケーションに適用して行く。

[参考文献]

[1]J.L.Perterson他、オペレーティングシステム の概念、培風館、1987。

[2]Sakamura, K., ed., μ ITRON3.0 Specification, TRON Association.

[3]岩崎他、オンチップリアルタイム OS 構成法の検討 - MPEG2多重/分離 LSIへの適用 -、1996年電子情報通信学会ソサエティ大会。

[4]M.Inamori, et al., A Memory-based Architecture for MPEG2 System Protocol LSIs, ED&TC 96, Mar. 1996.

[5]J.Naganuma, et al., A Polling-based Real-time Software for MPEG2 System Protocol LSIs, ASP-DAC97, Jan. 1997.

[6]速藤他、仮想 LSI 環境によるリアルタイム OS と MPEG2 MUXファームウェアの開発、1997年電子情報通信学会ソサエティ大会。

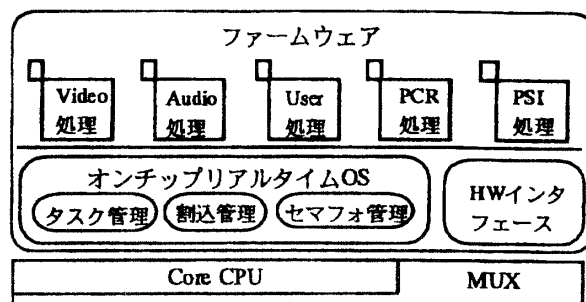


図1 MPEG2多重(MUX)LSIへの適用例

表2 タスクの優先度とプログラム規模

タスク	優先度	プログラム規模 (Kline)
初期化タスク	9	0.50
PCR初期化タスク	3	0.05
PSI初期化タスク	3	0.10
Video初期化タスク	3	0.10
PCRタスク	1	0.25
PSIタスク	2	1.28
Video監視タスク	8	0.25
Video header処理タスク	3	0.97
ヘッダファイル等	-	5.00
計		8.50

表3 動的な使用メモリ量

項目	使用メモリ量(Kbyte)
オンチップRTOSデータ領域	0.16*タスク数(8)+0.5
オンチップRTOSスタック領域	0.3
アプリケーションデータ領域	1.0
アプリケーションスタック領域	0.4
計	3.48

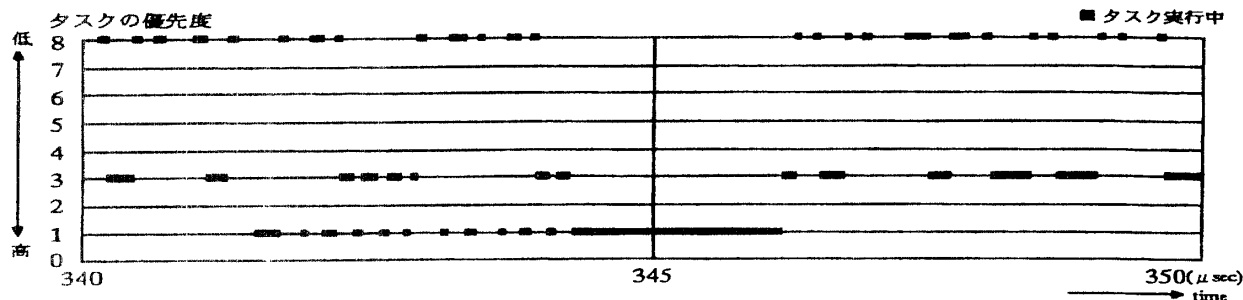


図2 リアルタイムOS上での MUX LSIファームウェアの動作