

TOPPERS/JSP カーネル ユーザズマニュアル

(Blackfin ターゲット依存部)

(Release 1.4 対応 , 最終更新: 25-Aug-2004)

TOPPERS/JSP Kernel

Toyohashi Open Platform for Embedded Real-Time Systems / Just Standard Profile Kernel

Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory Toyohashi Univ. of
Technology, JAPAN

上記著作権者は、以下の (1)~(4) の条件が、Free Software Foundation によって公表されている GNU General Public License の Version 2 に記述されている条件を満たす場合に限り、本ソフトウェア（本ソフトウェアを改変したものを含む。以下同じ）を使用・複製・改変・再配布（以下、利用と呼ぶ）することを無償で許諾する。

1. 本ソフトウェアをソースコードの形で利用する場合には、上記の著作権表示、この利用条件および下記の無保証規定が、そのままの形でソースコード中に含まれていること。
2. 本ソフトウェアを、ライブラリ形式など、他のソフトウェア開発に使用できる形で再配布する場合には、再配布に伴うドキュメント（利用者マニュアルなど）に、上記の著作権表示、この利用条件および下記の無保証規定を掲載すること。
3. 本ソフトウェアを、機器に組み込むなど、他のソフトウェア開発に使用できない形で再配布する場合には、次のいずれかの条件を満たすこと。
 - A) 再配布に伴うドキュメント（利用者マニュアルなど）に、上記の著作権表示、この利用条件および下記の無保証規定を掲載すること。
 - B) 再配布の形態を、別に定める方法によって、TOPPERS プロジェクトに報告すること。
4. 本ソフトウェアの利用により直接的または間接的に生じるいかなる損害からも、上記著作権者および TOPPERS プロジェクトを免責すること。

本ソフトウェアは、無保証で提供されているものである。上記著作権者および TOPPERS プロジェクトは、本ソフトウェアに関して、その適用可能性も含めて、いかなる保証も行わない。また、本ソフトウェアの利用により直接的または間接的に生じたいかなる損害に関しても、その責任を負わない。

1 . Blackfin ターゲット依存部の概要

1.1 ターゲットシステム

Blackfin プロセッサのターゲットシステムとしては、Analog Devices の ADSP-BF533 に対応している。これはシステムではなくプロセッサである。sys_config.h の中で入力周波数を指定することにより、どのようなボードにも対応できる。

1.2 開発環境と実行環境

開発環境には、Analog Devices の VisualDSP++ を利用する。開発は Release 3.5 で行った。

1.3 サポートする機能の概要

Blackfin 依存の機能として、割込みマスクの変更・参照 (chg_ims, get_ims) および割り込みの許可、禁止(ena_int, dis_int)、性能評価用システム時刻参照機能 (vxget_tim) をサポートしている。

1.4 他のターゲットへのポーティング

現バージョンでは、ADSP-BF533 のみをサポートしている。わずかな改造で ADSP-BF535 にも対応できるはずである。

2 . Blackfin プロセッサ依存部の機能

この節では、カーネルおよびシステムサービスの機能の中で、Blackfin 依存の部分について解説する。

2.1 データ型

signed int 型、unsigned int 型、size_t 型のサイズは、いずれも 32 ビットである。

2.2 割り込み管理機能と割り込みハンドラ

カーネル管理外の割り込みは NMI のみである。よって、CPU ロック状態や初期化ルーチン内では、NMI 以外の割り込みはすべて禁止されている。具体的には、IMASK が 0xC01F に設定される。

DEF_INH で指定する割り込みハンドラ番号 (inhno) は、ADSP-BF533 のシステム割り込み番号を表し、そのデータ型 (INHNO) は unsigned int 型に定義されている。DEF_INH に指定できる値は、sys_defs.h に INHNO_XXXX として定義してある。なお、システム割り込み番号とは、SIC_IMASK におけるビット番号である。

2.3 CPU 例外管理機能と CPU 例外ハンドラ

DEF_EXC で指定する割り込みハンドラ番号 (excno) は、単に無視される。例外ハンドラはただひとつだけが登録可能で、seqstat の excause による処理の振り分けはハンドラ内部で行う。

2.4 スタートアップモジュール

Blackfin 依存のスタートアップモジュール (start.asm) では、次の初期化処理を行う。

(A) プロセッサモードの初期化とスタックポインタの初期化

最初に LC0 と LC1 をクリアしてハードウェア・ループを無効化した後、L0-L3 を 0 にする。次にスタックをイベントスタックのトップに設定する。このスタックは dispatch() がタスク・スタックを設定するまで非タスクコンテキスト用のスタックとして使われる。プロセッサの割り込みベクトルに割り込みディスパッチャのエントリを設定する。最後にリセットモードから IVG15 モードに遷移する。

イベントスタックは LDF が定義した領域をリンカーが処理して埋め込む。

(B) カーネルの起動

kernel_start へ分岐し、カーネルを起動する。kernel_start からリターンしてくることは想定していない。

2.5 性能評価用システム時刻参照機能

Blackfin 依存部では、性能評価用システム時刻参照機能 (vxget_tim) をサポートしている。性能評価用システム時刻の精度は 1 サイクル単位である。この機能はコアのサイクルカウンタを直接読み出すため正確である。なお、SYSUTIM 型は UD 型 (64 ビットの符号無し整数型) に定義している。

2.6 SIC_IARx の変更

何らかの理由で内蔵デバイスからの割り込み要求順位を変更したいときには SIC_IARx を変更しなければならない。変更はアプリケーション初期化コードで行い、静的サービスコール ATT_INI を使ってシステムに登録する。変更用の初期化コードは次のようになる。

```
void init_iar(VP_INT vp_int)
{
    *pSIC_IAR0 = ...;
    *pSIC_IAR1 = ...;
    ...
    make_priority_mask();
}
```

make_priority_mask は SIC_IARx から TOPPERS/JSP が内部で使用する参照ビットマップを作り出す。この初期化コードは、コンフィグレーションファイルの中で他の初期化コードより先に宣言されなければならない。

3 . ADSP-BF533 システム依存部の機能

3.1 システムクロックドライバ

システムクロックドライバが `isig_tim` を呼び出す周期は、`sys_defs.h` 中の `TIC_NUME` と `TIC_DENO` で定義されている（標準は 1 ミリ秒周期）。システムクロックのタイマーはコアタイマーか、汎用タイマー2を使用する。`sys_config.h` のなかで `USE_TIC_CORE` マクロを定義するとコアタイマーを使う。汎用タイマー2を使用する場合に必要な `SIC_IMASK` の変更は自動的におこなう。

3.2 シリアルインタフェースドライバ

シリアルインターフェースドライバには対応していない。

3.3 割り込みマスク制御

ADSP-BF533 依存の機能として、`SIC_IMASK` 中のマスク値を変えるためのサービスコール `chg_ims` と参照するためのサービスコール `get_ipm` をサポートしている。なお、マスクの値を表すデータ型 `IMS` は、`unsigned int` 型に定義されている。

これらのサービスコールは、タスク・非タスクコンテキストのいずれでも、また CPU ロック状態、ロック解除状態のいずれでも呼び出すことができる。

Blackfin 依存の割り込みマスクの変更・参照のためのサービスコールの仕様は次の通り。

(1) `chg_ims`

割り込みマスクの変更

【C言語API】

```
ER ercd = chg_ims(IMS ims);
```

【パラメータ】

IMS	ims	設定すべき SIC_IMASK の値
-----	-----	--------------------

【リターンパラメータ】

ER	ercd	エラーコード
----	------	--------

【エラーコード】

E_CTX	コンテキストエラー
E_PAR	パラメータエラー（ipm が不正）

【機能】

`SIC_IMASK` を `ims` で指定した値にする。

このサービスコールは、常に 0 を返す。

(2) get_ims

割り込みマスクの参照

【C言語API】

```
ER ercd = get_ims(IMS *p_ims);
```

【パラータ】

なし

【リターンパラメータ】

ER ercd	エラーコード
IMS ims	現在の SIC_IMASKの値

【エラーコード】

E_CTX	コンテキストエラー
-------	-----------

【機能】

現在の SIC_IMASKの値を読み出し、ims に返す。

このサービスコールは常に 0 を返す。

(3) ena_int

割り込みの許可

【C言語API】

```
ER ercd = dis_int(INTNO intno);
```

【パラメータ】

INTNO	intno	許可するシステム割り込みの番号
-------	-------	-----------------

【リターンパラメータ】

ER ercd	エラーコード
---------	--------

【エラーコード】

E_PAR	パラメータエラー (intno が不正)
-------	------------------------

【機能】

SIC_IMASKをintnoビットを1にする。対応する割り込みが許可される。パラメータとして渡すことのできる値はsys_defs.hにINTNO_XXXXとして宣言してある。

許されない番号を渡すと E_PAR を返す。

(4) dis_int

割り込みの禁止

【C言語API】

```
ER ercd = dis_int(INTNO intno);
```

【パラメータ】

INTNO intno 許可するシステム割り込みの番号

【リターンパラメータ】

ER ercd エラーコード

【エラーコード】

E_PAR パラメータエラー (intno が不正)

【機能】

SIC_IMASKをintnoビットを0にする。対応する割り込みが禁止される。パラメータとして渡すことのできる値はsys_defs.hにINTNO_XXXXとして宣言してある。

このサービスコールは常に 0 を返す。

4 . 開発環境の構築

開発環境は Analog Devices の VisualDSP++を使用した

4.1 開発環境のバージョン

動作確認したツールのバージョンは以下の通りである .

- VisualDSP++ : 3.5 July Update

コンパイラ、アセンブラ、リンカはチップのアンノーマリーに対応するため以下のオプションをコマンドラインオプションに追加しておく

-si-revision X.Y

X.Y は対応するシリコンのリビジョンである。

5 . その他

5.1 ディレクトリ・ファイル構成

Blackfin ターゲット依存部の各ファイルの概要は次の通り .

config/blackfin-vdsp/

cpu_defs.h	プロセッサ依存部のアプリケーション用定義
cpu_config.h	プロセッサ依存部の構成定義
cpu_config.c	プロセッサ依存部の関数
cpu_support.asm	プロセッサ依存部のサブルーチン
cpu_context.h	コンテキスト操作
cpu_rename.h	カーネルの内部識別名のリネーム
cpu_unrename.h	カーネルの内部識別名のリネーム解除
tool_defs.h	開発環境依存部のアプリケーション用定義 (VisualDSP++開発環境用)
tool_config.h	開発環境依存部の構成定義 (VisualDSP++開発環境用)
start.asm	スタートアップモジュール

config/blackfin-vdsp/bf533

sys_defs.h	システム依存部のアプリケーション用定義
sys_defs.c	システム依存部のアプリケーション用関数
sys_config.h	システム依存部の構成定義
sys_config.c	システム依存部の関数
sys_rename.h	カーネルの内部識別名のリネーム
sys_unrename.h	カーネルの内部識別名のリネーム解除
hw_timer.h	タイマ操作ルーチン