

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: 26-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。実際には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

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。

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。

2.4.1. 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ーを設定する。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。

2.4.2. 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。

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のでユーザーが気にする必要はない。

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. 割り込みハンドラ

`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。

`INHNO_RAISE` は `RAISE` 命令がおこなうソフトウェア割り込みのためのハンドラ番号である。`RAISE` に与えるオペランドは割り込みの順位番号を与えるが、順位がどうであれ `INHNO_RAISE` に登録されたユーザーのハンドラが呼び出される。`RAISE` 命令による割り込みを他から区別する手段がないため、`RAISE` を使う場合は他のデバイスからの割り込み要因が割り当てられてない順位を利用しなければならない。

割り込みハンドラがシステム割り込み番号ごとに呼び出されるため、アプリケーション・プログラマが割り込みハンドラ内部で実際に割り込みを起こしたデバイスを同定する必要はない。ただし、割り込みのクリアはユーザーの割り込みハンドラの責任で行う。

3.4. `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。

3.5. 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。

割り込みマスクの変更および、dis_int()を使って特定のデバイスからの割り込みを禁止する場合は、必ず CPU ロック解除状態で行うよう注意する。Blackfin コアは IMASK による割り込み禁止（CPU ロック状態に相当）であっても ILAT レジスタに割り込み要求を記録する。したがって、CPU ロック状態に入ってから dis_int()に入る前までに割り込みが生じる可能性がある。このような割り込みが起きてもいいように慎重にプログラムを組むことはできるが、単純に CPU ロック解除状態で dis_int()を呼ぶようにすれば何の問題もおきない。割り込みマスクの変更も同様である。

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。

3.5.1. 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。

3.5.2. 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。

3.5.3. 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。

3.5.4. 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

1.5. 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ー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. 独自ボードへの移植

TOPPERS/JSP for Blackfin は Analog Devices 社の評価基板である EZ-KIT Lite BF533 上で開発された。そのため、一部設定は同ボードに依存している。それらの点は `sys_config.h` のマクロを書き換えることで変更できる。また、それ以外にもユーザーが独自のボードにあわせこむための機能が用意されている。

5.1. クロック設定の変更

ユーザーが任意の周波数で Blackfin を動かすことができるよう、`sys_config.h` にクロック設定用のマクロが用意してある。これらの既定値は以下のとおり。この既定値は EZ-KIT BF533 上で 600MHz 動作を行うためのものである。

```
#define CSELVAL 1
#define SSELVAL 5
#define MSELVAL 22
#define CLKIN 27000000
```

CSELVAL, SSELVAL はそれぞれ PLL_DIV レジスタの CSEL フィールド、SSEL フィールドの設定制御用の値である。CSELVAL は 1,2,4,8 の中のいずれかの値を指定する。これはコア用の分周比であって PLL_DIV の CSEL フィールドに書き込む値ではないことに注意する。SSELVAL は SSEL フィールドに書き込む値そのものである。

MSELVAL は PLL_CTL レジスタの MSEL フィールドに書き込む値を指定する。これらの設定を元に `sys_config.c` の `sys_initialize()` が PLL の設定を行う。

CLKIN は ADSP-BF533 の CLKIN 端子に入力する周波数を Hz 単位で宣言する。EZ-KIT BF533 の場合 27MHz 入力なので既定値は 27000000 になっている。

以上の 4 つのマクロを元に、TIC (システム) タイマーの設定値と微小時間待機関数の設定も自動的に行われる。したがってこれらのマクロさえ正しく設定すれば TIC(システム)タイマーの設定を気にする必要はない。

5.2. システムタイマーの変更

ユーザーのシステムに応じてシステムタイマーとして Blackfin のコアタイマーかあるいは ADSP-BF533 の GP タイマー-2 のいずれかを利用することができる。コアタイマーを使う場合には `sys_config.h` の中で `USE_TIC_CORE` マクロを定義する。

```
#define USE_TIC_CORE
```

デフォルトではこのマクロはコメント・アウトされている。すなわち、デフォルトでは GP タイマー-2 を使っている。コアタイマーを使う場合には、`dispatch()` の割り込み待ち部分で `idle` 命令ではなく `nop` 命令をつかうようにプログラムされている。

システムタイマーの割り込みハンドラとその登録は上記マクロの状態によって自動的に条件コンパイルされるのでユーザーが手を加える必要はない。

5.3. SIC_IARx の変更

SIC_IARx の値を変えると、割り込み源と優先順位の対応付けが変わる。TOPPERS/JSP for Blackfin はこの対応を表として持っているため、SIC_IARx を書き換えたら必ずこの表をアップデートしなければならない。表のアップデートは `make_proiority_mask()`関数を呼ぶことで完了する。

SIC_IARx の変更は複雑な問題を起こしうるため、原則としてシステム起動時に一度だけ行うべきである。コンフィグレーションファイルの中で ATT_INI によってアタッチされる初期化コードの中で変更するのがもっともよい方法である。この場合、他の初期化コードより先にアタッチしなければならない。

SIC_IARx の変更に関してはサンプルプログラムを用意しているので参照されたい。

6. 他プロセッサへの移植

Blackfin コアを使った他のプロセッサへの移植はシステム依存部を書き換えることによって行う。CPU 依存部は Blackfin にのみ依存するように設計しているが、問題がある場合はこの文書の末尾に知らせた sourceforge まで連絡していただきたい。

6.1. システムタイマー制御

システムタイマーは `hw_timer.h` を書き換えて移植する。コアタイマーを使う場合にはこの部分の変更はないと思われるが、GP タイマーを使う場合は変更が必要になる。

6.2. PLL 制御

PLL の制御は `sys_config.c` で行っている。場合によっては `sys_config.h` の設定用マクロも変更が必要になるかもしれない。変更を受けうるのは `MSELVAL`, `CSELVAL`, `SSELVAL`, `CLKIN` である。一例として ADSP-BF535 の場合 `CSEL` が不要になる。

6.3. `INTNO_xxxx` と `INHNO_xxxx`

`INTNO_xxxx` と `INHNO_xxxx` は `sys_config.h` に定義してある。これらは全面的に際宣言が必要である。なお、システムタイマー割り込みハンドラ番号である `INHNO_TIMER` には `USE_TIC_CORE` の定義の有無に応じて適切な値を割り当てる。具体的には `USE_TIC_CORE` があるときにはコアタイマー割り込みの割り込みハンドラ番号を、ないときにはシステムタイマーとして使う GP タイマーの割り込みハンドラ番号を割り当てる。

ハードウェア割り込み、コアタイマー割り込み、ソフトウェア割り込みタイプのハンドラ番号は、デバイス用の割り込みハンドラ番号より大きなものを割り当てる。

6.4. `priority_mask[]` と `make_priority_mask()`

マクロ定数 `DEVICE_INTERRUPT_COUNT` に割り込み要因の数を定義すること。これは `SIC_IMASK` のビット数である。`priority_mask[]` のサイズはこのマクロの値+3 になる。

割り込み要因の数が 32 を超える場合、`priority_mask` を変更しなければならない。同様に `make_priority_mask()` も変更が必要になる。

割り込み要因の数が 32 を超えない場合でも、`priority_mask[]` の初期値はプロセッサに合わせて変更しなければならない。

7. その他

7.1. 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

7.2. 連絡先

TOPPERS/JSP for Blackfinへの要望やバグ報告は <http://sourceforge.jp/projects/toppersjsp4bf/> まで連絡されたい。なお、TOPPERS/JSP自身の使い方等は、情報共有のためにもTOPPERS プロジェクト(<http://www.toppers.jp/>)のMLで行うことをお勧めする。