

オーディオ信号処理で学ぶ DSP

第4回 Blackfinの周辺機能制御と割り込み

堀江 誠一

本連載ではDSP (Blackfin) を使ったオーディオ・プログラミングについて解説している。一般的なオーディオ・プログラミングであれば、これまで提示されたフレームワークを利用するだけで作成できる。しかし高度なプログラムを作るためには割り込み処理が必須となる。そこで連載第4回目の今回は、Blackfin において割り込み処理を行うためのプログラミング手法について解説する。(編集部)

本連載では、プログラムを容易に作成できるように、枠組みとなるフレームワークを導入しています。このフレームワークを元にして改造を行えば、簡単に信号処理の実験が行えて便利です。例えば、`afw::processData()` 関数の中を取り替えるだけで、トーク・スルーやカラオケといったアプリケーションに変更できることは以前に記述した通りです(2007年1月号, pp.136-146)。また、初期化関数である `afw::init ProcessData()` まで手を加えるなら、フィルタの実験などができることも紹介しました。

フレームワークはこのように便利に使えますが、もう少し踏み込んだ実験を行うとなると、これだけを使っているわけにはいきません。例えば「プッシュ・ボタンを押すたびに音の大きさが変わる」といった実験を考えてみましょう。この場合、プッシュ・ボタンを監視するか、あるいはプッシュ・ボタンからの割り込みを受けて、その変化を検出しなければなりません。

汎用マイクロプロセッサによるアプリケーションの場合、よほど単純なものでない限り、ボタンの変化は割り込みで検出します。そこで、今回は ADSP-BF537 の評価基板である EZ-KIT Lite を例に、プッシュ・ボタンによる割り込みを処理する方法を説明します^{注1}。

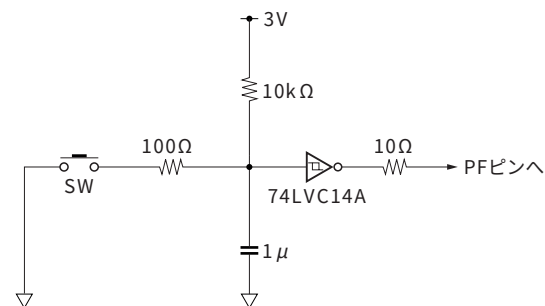


図1 EZ-KIT Lite のプッシュ・ボタン回路

Blackfin プロセッサは汎用 CPU と同じように割り込みを取り扱うことができます。また、VisualDSP++ のライブラリは割り込み支援機能を持っています。プログラマはこれらの機能を使うことによって、高級言語だけで割り込み処理プログラムを書くことができます。

1. プッシュ・ボタン割り込みの流れ

図1にEZ-KIT Lite上のプッシュ・ボタン・スイッチの回路図を示します。

プッシュ・ボタン・スイッチはプルアップ抵抗とチャタリング除去用のCR回路に接続されています。プッシュ・ボタン・スイッチの他端はGNDに接続されているので、ボタンを押すと電圧レベルは“L”になります。この電圧レベルはシュミット・トリガ入力の論理反転回路で受けてADSP-BF537のPORTF フラグI/O(GPIO)に接続されています。

結果的に、ユーザがプッシュ・ボタンを押すと、ADSP-BF537のPORTF入力には“H”レベルが入力されます。

フラグI/Oは名前の通り、ビット単位のON/OFFが可能な入出力ピンですが、割り込み入力としても使えます。今回はこの周辺機能を割り込み源とします。

ADSP-BF537内部では、図2に示すように、フラグI/O→割り込みコントローラ→イベント・コントローラの順で信号が流れていき、最後に割り込みハンドラが起動されます。以下では、この信号の流れの順にADSP-BF537の割り込みの流れを詳しく説明します。

注1：サンプル・プログラムはCQ出版社のWebサイトからダウンロードできる。<http://www.cqpub.co.jp/interface/download/>

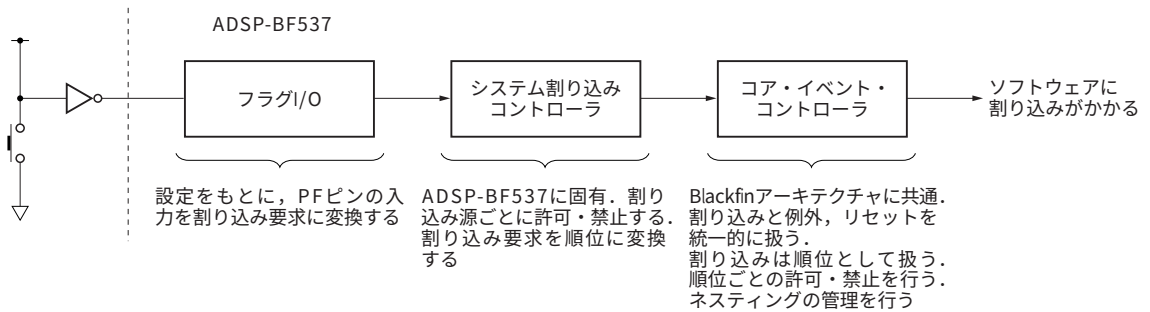


図2
割り込みの流れ

2. BlackfinのフラグI/O

● フラグI/O とは何か

Blackfinの持つ「フラグI/O」は、ビット単位でON/OFFを制御できる周辺機能です。一般にGPIOと呼ばれているものと同じような機能を持っています。ADSP-BF537ではピン数を節約するために、フラグI/Oはほかの機能の周辺機能とピンを共有しています。

ADSP-BF537は3系統48本のフラグI/Oを持っています。これらのうち、EZ-KIT Liteがプッシュ・ボタンに接続しているのはPORTFなので、今回はフラグI/OといえばPORTFを指します。ただし、どのポートのフラグI/Oも機能は同じなので、PORTG、PORTHも同じように使うことができます。

● 入力/出力/割り込み入力機能を持つ

フラグI/Oの機能は、大きく分けて三つあります。一つ目は入力で、スイッチのON/OFFなどの状態を読み込むために使います。二つ目は出力で、これはLEDの点灯などの機能に使います。三つ目は割り込み入力で、外部からの信号をトリガにして、ADSP-BF537内部で割り込みイベントを発生します。これらの機能の設定は、メモリ・マップされたレジスタから行うことができます(図3)。

関連レジスタ群はこの3種類の機能用に管理用を加えた、計4種類のレジスタからなります。レジスタはすべて16ビット幅で、レジスタ内の各ビットはポートの各ピンに対応しています。つまり、PORTFレジスタのビット3は、PF3ピンに対応しています。

● フラグI/Oレジスタの説明

以下、各レジスタの説明を行います。

PF12				PF8				PF4				PF0			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTFレジスタはすべて16ビット幅で、各ビットが各ピンに1対1に対応している

管理用レジスタ

PORTF_FER(Function Enable Register)
0: フラグI/O
1: ほかの周辺機能
PORTFIO_DIR(Direction Register)
0: 入力
1: 出力

入力設定用レジスタ

PORTFIO_INEN(Input Enable Register)
0: 入力バッファ切り離し
1: 入力バッファ接続
PORTFIO_POLAR(Polarity Register)
0: 正論理
1: 負論理

出力用レジスタ

PORTFIO
0: 出力ピンがLになる
1: 出力ピンがHになる
PORTFIO_SET
0: 出力ピン変化なし
1: 出力ピンがHになる
PORTFIO_CLEAR
0: 出力ピン変化なし
1: 出力ピンがLになる
PORTFIO_TOGGLE
0: 出力ピン変化なし
1: 出力ピンがLになる

割り込み設定用レジスタ

PORTFIO_EDGE
0: レベル割り込み
1: エッジ割り込み
PORTFIO_BOTH
0: 片エッジで割り込み
1: 両エッジで割り込み
PORTFIO_MASKA
0: PORTF割り込みAは起きない
1: PORTF割り込みAが起きる
PORTFIO_MASKB
0: PORTF割り込みBは起きない
1: PORTF割り込みBが起きる

注意: データ入力時は、PORTFIOレジスタを読む。
割り込みがかったかどうかの確認もPORTFIOレジスタを読んで判断する。

図3 レジスタと機能の一覧