

TOPPERS[®]で学ぶ RTOS技術

第17回 Blackfin DSP への TOPPERS/JSP の移植 (2)

中村 健真

前回(本誌2006年6月号)はTOPPERSをBlackfinへ移植するにあたり、ディスパッチ部分までを解説しました。今回はCPU移植のうち難関である割り込み部分の移植を解説します。

割り込み処理とそのほかの部分

● μ ITRON4.0 仕様における割り込みの実装

割り込み処理は簡単そうに見えて、なかなか手ごわい部分です。プロセッサのアーキテクチャの割り込み機能を素直に使うのなら、単に割り込みを処理して戻ってくるだけなので単純に思えます。

しかしながら、 μ ITRON4.0仕様は割り込みからタスク・コンテキストに戻る際、必要に応じてタスク例外の処理やディスパッチを行うよう要請しています。この機能の実装は難易度が高く、処理系ごとにくふうを凝らす必要があります。したがって、この部分をどのようにするかが、割り込み処理の要^{かなめ}となります。

また、割り込みはタスク・スタックを多少なりとも消費します。そのため、アプリケーション・プログラマはアプリケーションの必要量に加えて割り込みで使うぶんもタスク・スタックを用意しなければなりません。ディスパッチャがほぼ透過だったこととは対照的です。

以下、このような点を頭に置きながら割り込み処理部の流れを見ていきましょう。

● 割り込み処理の流れ

割り込み処理は先に述べたように少しひねった部分がありますが、プログラムの流れそのものは一直線で、ディスパッチャに比べるととても単純です。TOPPERS/JSP for Blackfinの割り込み処理の流れを図1に示します。

入り口処理は割り込まれたコンテキストを保存する部分で、おもにレジスタの退避を行います。それに加えて必要に応じてタスク・スタックをイベント・スタックに切り替えます。

共通割り込み処理部は、ユーザ定義の割り込みハンドラを呼び出します。通常、ここは単純に割り込みを起こしたベクタに応じてハンドラを呼び出すだけです。しかしTOPPERS/JSP for Blackfinでは少しくふうを凝らしてユーザが使いやすいよ

うにしています。これについては後で説明します。

出口処理部は入り口と逆の操作を行います。必要に応じてイベント・スタックからタスク・スタックへの切り替えを行った後、レジスタをすべて復帰します。そして最後に割り込まれた場所に戻ります。

出口処理部は場合によって単純には元の場所に戻らないことがあります。これは μ ITRON4.0仕様の要請によるものです。 μ ITRON4.0仕様は割り込みからタスクに戻る場合、ディスパッチ要求が出ているならばタスク・ディスパッチを、タスク例外要求が出ているならばタスク例外処理を行うことを要請しています。この二つは割り込みの出口で実行されますが、いずれもタスク・コンテキストで行わなければなりません。そしてさらにディスパッチ要求があった場合には、割り込みからの戻り先は割り込まれたタスクではなく、新たにCPUを獲得してディスパッチされるタスクになります。

出口処理では、このような場合に正しくタスク・コンテキストでのディスパッチを行った後、再び割り込まれたタスクがディスパッチされたときに、タスクの割り込まれた地点に戻る処理が必要になります。この部分が割り込み処理の難しいところです。

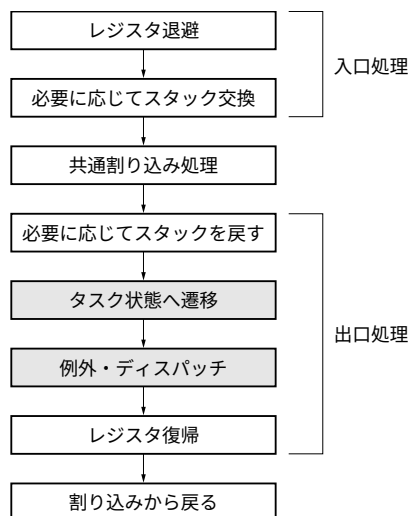


図1 TOPPERS/JSP for Blackfinの割り込み処理

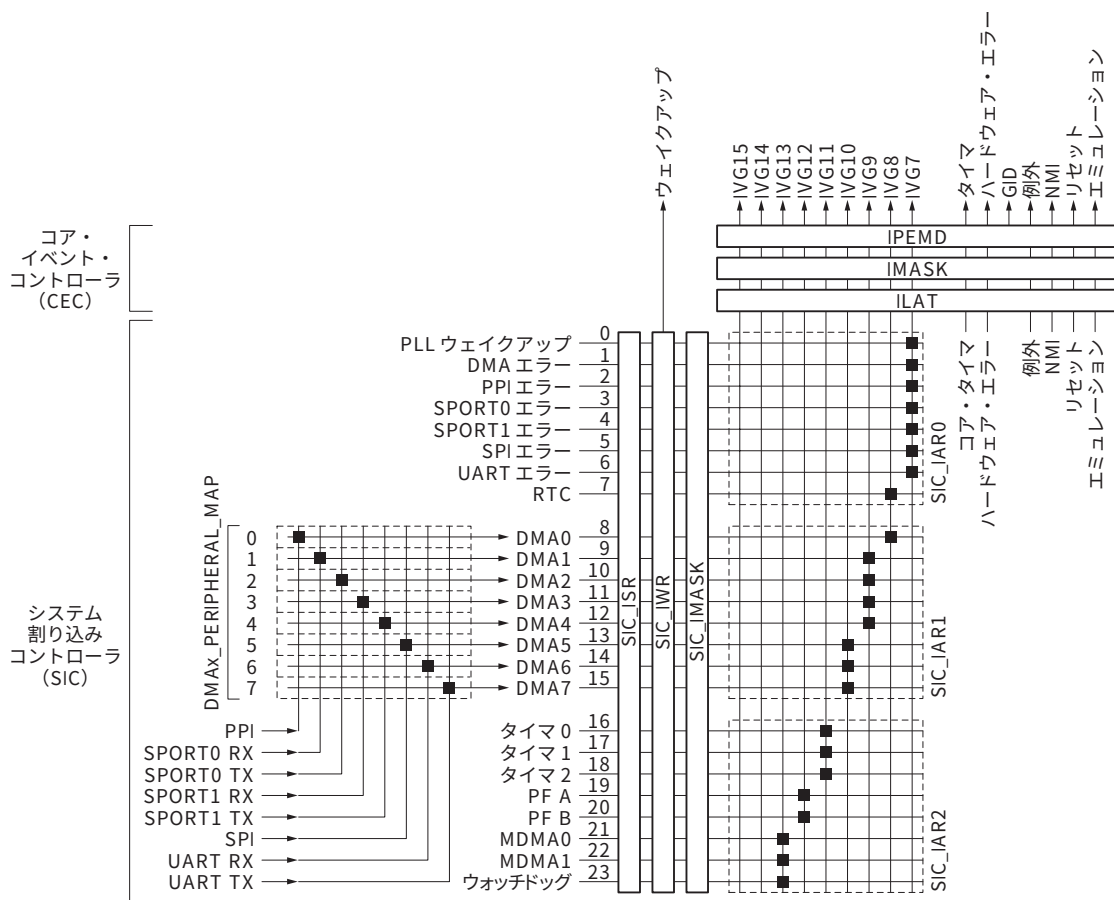


図2 ADSP-BF533の割り込み

● ADSP-BF533 の割り込み構造

割り込み処理の詳細を追う前に、もう一度 ADSP-BF533 の詳細を見ておきましょう。Blackfin ファミリは、割り込み信号を Blackfin プロセッサの統一的なイベント処理の枠組みに従って処理します。

ADSP-BF533 の割り込みは、図2に示すように二段構えになっています。内蔵周辺機能からの割り込み要求は、システム割り込みコントローラ(SIC)によって受け付けられます。SIC は SIC_IMASK と割り込み要求の論理積^{注1}をとったのち、SIC_IARx の設定に基づいてそれぞれの要求をイベント要求信号に変換します。SIC_IARx の出力はそれぞれ内蔵周辺機能からの割り込み要求信号になっていますが、SIC_IARx の出力は Blackfin コアに要求するイベント順位になっています。

変換されたイベント要求信号は Blackfin コアに入力され、ILAT に記録されます。そしてイベント・マスクである IMASK レジスタと論理積が取られます。IMASK によって許可されているイベントは、今度は現在実行中のイベント順位と比較され、

要求順位が高い場合にはイベントが Blackfin コアの実行の流れを中断させます。実行中のイベントは、ネストされたものを含めて IPEND レジスタに記録されています。IPEND レジスタを図3に示します。

イベントの種類はエミュレーション、リセット、NMI、例外、割り込みに分類されます。このうち、リセットを除くイベントは戻り番地を記録する専用レジスタをもっており、それぞれ RETE, RETN, RETX, RETI と呼ばれます。

● 入り口処理

では、TOPPERS/JSP for Blackfin の割り込み処理の流れを見てみましょう。

割り込み信号はイベントでいうと IVG15 から IVG7 の一般割り込み、タイマ専用の IVTMR、ハードウェア・エラー検出用の IVHW のいずれかに割り当てられます(表1)。このうち、IVG15 と IVG14 は TOPPERS/JSP for Blackfin が予約しているので、アプリケーション・プログラマが内部の一般的な周辺機能に割り当てることのできるイベントは、IVG13 から IVG7 ま

注1：Analog Devices の DSP は、伝統的に割り込みマスクが、「許可を表すビット・マスク」になっている。すなわち、あるビットが1であれば対応する割り込みが許可される。一方で、NMI (Non Maskable Interrupt) のマスクは禁止の意味で用いられている。混乱しやすいので注意すること。