

PCI Expressを本格的に使うには割り込みの実装が必須です。高速なデータ転送にはDMAが必須ですが、DMAを実装するなら転送完了を知るために割り込みを使わなければならないからです。そこで今回は、PCI Expressにおける割り込みの実装方法と、Windowsのデバイス・ドライバでそれを受け取るためのしくみについて解説します。(筆者)

1. 割り込みのハードウェア

● PCIの割り込み処理

レガシなPCIにはINTA#～INTD#という4本の割り込み信号線(以下、INTxと略)があり、アドイン・カードはこの信号線のどれか一つをLレベルにすることによって、システムに割り込みをかけられるようになっていました。

この信号はマザーボード上に並んだPCIのロットで共有されていて、順番にひねったようにつながっています(図1)。そして各信号がワイヤードORでつながっていて、複数のカードで同一の割り込みラインを共有することも可能でした。たとえば、図1の例では、ロット1のINTA#とロット2のINTB#は同一の線を使っています。それぞれのアドイン・カードがINTA#とINTB#を使うなら、これらの割り込みは共有になります。

また、INTxはエッジではなくレベル・トリガです。つまり、INTxがLレベルの状態にある限り、システムに割り込みがかかり続けます。

レガシPCIは、レベル・トリガとワイヤードORによって複数の割り込みが共有できていました。

注1: Message Signaled Interrupt. PCI 3.0でオプションとして規定された方式。

● PCI Expressの割り込み発生方法

PCI Expressの割り込みには、従来のレガシPCIで使われていたINTA#～INTD#をエミュレートする方法と、MSI^{注1}またはMSI-Xを用いる方法があります。レガシなINTxをエミュレートする方法は下位互換性のために残されています。INTxを用いる方法は簡単であり、システムのブート時からすぐに使えるというメリットがあります。しかし、複数の割り込み要因が共有されるというデメリットもあります。

PCI Expressでは、INTx割り込みを発生させるデバイスは、MSIまたはMSI-Xか、もしくはその両方をサポートすることが必須となりました。デバイスが両方の割り込み方法をサポートしていても、同時に使われるのはどちらか一方です。

● INTxエミュレーションによる割り込み発生

PCI ExpressにはINTxの物理的な制御線はないので、仮想ワイヤというしくみでそれをエミュレートします。これは、エンドポイントとルート・コンプレックスはINTxの状態を覚えておいて、メッセージ・パケットを使って変化するタイミングを知らせるというものです(図2)。

INTxをLレベルに下げたい場合にはAssert_INTxメッセージを送信し、Hレベルに戻したい場合はDeassert_INTxメッセージを送信します。このメッセージ・パケットは図3に示す構造をしています。

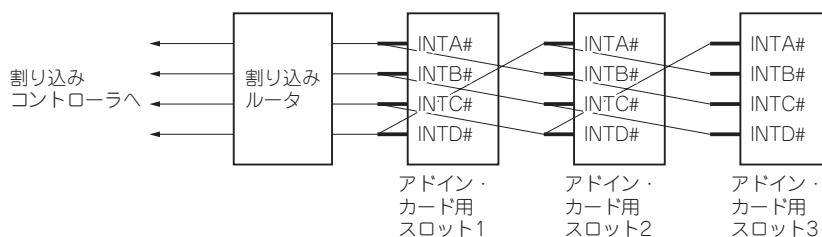


図1
レガシPCIにおける割り込みのルーティング
割り込みルーティングは、PCIデバイスの割り込み線をシステムの割り込み線に割り当てて接続する。