

IP-VLBI用デバイスドライバの修正

木村 守孝

平成17年7月8日

1 修正項目

- DMA 転送を REWIND モードへ変更 (ハングアップの対応)
- FIFO の増加分から時系列を保証する方法の採用 (データずれの対策)
- ドライバ内のバッファ構造の変更
- タスク切替えサイクルを 1000Hz へ変更

1.1 DMA 転送を REWIND モードへ変更

オリジナルのデバイスドライバの DMA 転送は RELATIVE モード (転送先アドレスが自動的にインクリメントする) が使用されている。このモードは確保したバッファ内へ順次データを転送する場合には便利であるが、バッファが全て満たされる前に新しい転送先アドレスを再設定できない場合、バッファオーバーフローが発生する。この場合、高い確率でハングアップ常態に陥る。DMA 転送を RELATIVE モードから REWIND モード (転送先アドレスは毎回同じ) へ変更することで DMA 転送でのバッファオーバーフローは原理的に発生しない。

1.2 FIFO の増加分から時系列を保証する方法の採用

オリジナルのデバイスドライバでは DMA 転送サイズを基にデータの時刻列を保証する仕組みになっている。この DMA 転送サイズは転送前後での FIFO の読み出しカウンタの差と原理的に一致するはずであるが、IP-VLBI ボードのハードウェアのバグによりこの二つは一致しない。FIFO の読み出しカウンタが示すアドレスから DMA 転送が開始するはずなので、FIFO の読み出しカウンタを用いて時系列を保証する方が信頼性が高い。

※ FIFO の読み出しカウンタが異常リセットされる IP-VLBI ボードが一部発見された。時系列データを正しく取得するためにはハードウェアのバグを解消しなくてはならない (ドライバ側では完全な解決は不可能)。

1.3 ドライバ内のバッファ構造の変更

オリジナルのデバイスドライバでは "for safety" という目的で 2MB のバッファをカーネル空間上で複数個確保している。しかし、DMA の転送サイズはハードウェア的な制約より 512KB が上限であるため、512KB 以上のバッファ確保にはメリットが無い。これは DMA 転送を RELATIVE モードで使用する時、オーバーフローを無くすための保険と考えられるが、REWIND モードで使用する場合には不要である。また、オリジナルでは 1 個のバッファに複数回に渡る DMA 転送データを保存しており構造が複雑になっていた。そこで、バッファ 1 個のサイズを 2MB から 512KByte と小さくし、1 回の DMA 転送で 1 個のバッファを使用することにした。バッファの個数を 64 個に増加させた。

1.4 タスク切替えサイクルを 1000Hz へ変更

このボードでは 1 回のハードトリガ DMA 転送で最大 256KB の転送が出来る。128Mbps の取得モードでは $128\text{Mbps}/256\text{KB}=64$ 回/sec の DMA 転送がおこなわれる。Linux の通常の kernel では 100Hz でタスク切替えがおこなわれているため、負荷の全く無い常態では 64 回/sec の制御は可能かもしれないが、通常のマルチタスク OS 上では安定した制御は困難である。そこで、Kernel コードの一部を変更し、タスク切替えを 1000Hz へ変更した。