Hardware Acceleration Models
and
Re-direction of Audio Streams

revision 1.0

Written by
Gary Solomon - Platform Architecture Lab
gary_solomon@ccm.jf.intel.com
&
Dan Cox - IAL Media & Interconnect Technology Lab

Intel Corporation

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein, and Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information in this document. Intel does not warrant or represent that such implementaion(s) will not infringe such rights.

* Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

1. Introduction

This paper is targeted at IHVs and OEMs who have detailed working knowledge of the current PC audio architecture. It is also recommended that the reader be familiar with the Audio Codec ‘97 Component Specification available on the Intel Web server at http://www.intel.com/pc-supp/platform/ac97/.

The goal of this paper is to identify two principal hardware acceleration models for PC audio, and justify from a platform architecture perspective why there is a preference for one over the other.

Companion white papers address two related subjects:

- “Digital audio” and the 1997 desktop PC
- Implementing legacy audio on the PCI bus

2. Hardware Acceleration Models

Hardware acceleration is most effective when applied in a manner that adds significantly to the performance of its appointed tasks without, in the process, imposing a performance cost on the host or the rest of the system.

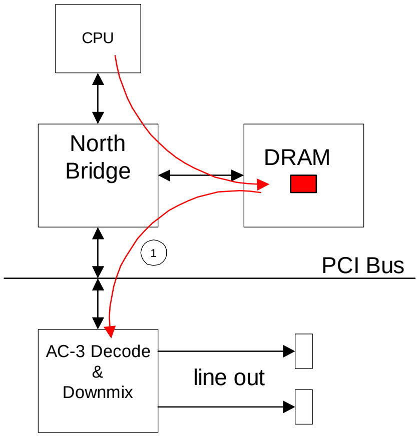
Hardware acceleration can be implemented in one of two basic models:

- an “in-line” accelerator model which is in extensive use today for PC audio
- a “multi-trip” accelerator model which has been proposed as a solution for re-directing audio output

2.1. In-line Acceleration Model

An in-line accelerator processes data in a manner similar to an “in-line filter” where the specialized hardware acts as an intermediary “active filter” situated between the host and the final destination for the data. Once a data structure has been passed from the host to the hardware accelerator, the host never has to touch that particular data structure again. Ideally an overlapped, “pipelined” channel of communication is established between the host and its accelerator which brings with it big application performance gains due to maximized parallelism.

An example of an in-line accelerator is shown in the figure below.



In the above example the system could be playing a DVD-ROM movie where the CPU has just parsed the Dolby* AC-3 encoded audio away from the MPEG-2 video, and placed it into an audio playback buffer in DRAM. At this point (1) the AC-3 hardware accelerator retrieves the encoded AC-3 data from its DRAM buffer, decodes it, and down mixes the 5.1 channel result to a 2 channel output suitable for playback in a 2 speaker volume PC platform.

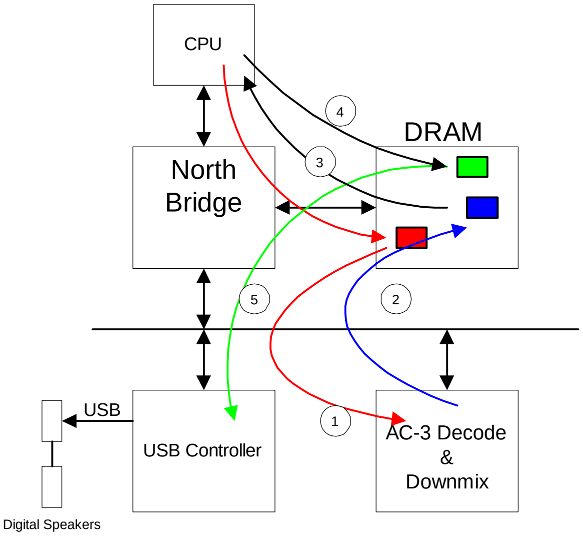
This example illustrates how the AC-3 hardware accelerator is situated between the agent it is targeted to assist (the CPU), and the final destination for the data (the speakers).

2.2. *Multi-trip Hardware Acceleration Model*

Any "multi-trip" accelerator model delivers diminishing returns relative to an in-line accelerator.

Less parallelism is attainable with the multi-trip model since the CPU will be stalled to some degree in its preparation of the next data structure by increased latencies associated with the additional system-wide traffic for the current data structure. There is also a potential requirement for additional CPU involvement[1] with the current data structure which would further diminish the effective parallelism of the accelerator.

The figure below models the same scenario described for the in-line accelerator, however in this case the resultant data is not fed to its final destination directly. In this example the hardware accelerator inside the PC is used to perform work on an audio data stream that is targeted for USB (or 1394), with digital speakers outside the PC.



As shown above, there are actually a minimum of 3 bus trips, with the potential to grow to 5 bus trips if the Microsoft* OS / Win32* Driver Model (WDM) driver infrastructure cannot effectively pass the resultant AC-3 decoded buffer pointer directly to the USB software stack. If efficient buffer pointer passing cannot be accomplished given the current OS/WDM driver infrastructure, a memory to memory move will be required to stage the resultant buffer for transmission down the USB. In either case, the CPU's ability to prepare and deliver the next data structure is impacted.

The sequence of events is as follows:

1) Hardware accelerator retrieves the data structure that has been prepared for it by the CPU.

2) AC-3 decode and stereo down mix (in this example) performed, and resultant stream written into a new buffer area of DRAM. The OS is notified of the new buffer being ready (interrupt).

[1] Additional work following the passing off of the data structure to the hardware accelerator.

3), 4) OS (CPU) may now be required to move the data to a new buffer area, staging the data for the USB pipe. Passes the output buffer information to the USB driver.[2]

5) USB host interface retrieves the resultant data and sends it down the USB.

Given that this example will be typical of DVD-ROM movie playback, with the addition of an MPEG-2 video stream also propagating throughout the system, the impact on the application's performance, relative to the in-line model, could be dramatic.

The synchronization of the audio and video streams needs to be carefully considered with the increased impact to system responsiveness that the multi-trip model imposes. The impact on the accuracy of audio sample position status needs to be fully understood given longer system-wide latencies.

With the drive for better 3D graphics (AGP) and cost effective MMX technology, the DRAM memory subsystem has become the limiting factor. This "ping pong" pattern of bus traffic will only make this problem worse. In the end, the application performance benefits promised by the hardware accelerator are dampened by the hit to the system's responsiveness as a result of the increased traffic for each data structure.

3. Recommendations

If a function requires hardware acceleration, the in-line acceleration model is preferred.

A PCI based AC '97 controller which implements in-line AC-3 decode and generates ~90dB SNR *analog* output at the line out stereo mini jack may be preferable from a performance standpoint to the same system with re-directed *digital* audio output.

OEMs who are considering adopting the multi-trip accelerator model as a way to re-direct audio output streams to USB (or 1394) in support of a high end "Living Room PC", or "modular" high volume PC with a USB hub and digital speakers built into the monitor, should carefully evaluate the impact of 2-4 additional 16-bit 48Kss stereo streams on overall system performance:

- CPU and memory performance
- latency and A/V synchronization

4. Conclusion

As the host and memory subsystems become more powerful, the host will transition into the role of orchestra as well as conductor (performing many of the acceleration tasks), regardless of whether the final destination for the audio is an internal Codec with line out to speakers, or digital speakers situated on USB (or 1394), an in-line model is achieved. In the near term, OEMs may choose to deliver solutions based on the multi-trip accelerator model.

[2] This is an issue that could be resolved with software (OS/WDM driver) infrastructure work.