6.111 Final Project Proposal
Harrison Hall

Virtual Surround Sound from a Dolby Digital Source

The goal of this project is to extract the data coded in a continuous stream of Dolby Digital media and simulate the 5 satellite speakers and subwoofer on a pair of conventional, stereo headphones. The basic breakdown is to decode the information in the stream into the 6-channels of PCM data that typically flow to a home theater audio system and then to use this data as input to a Head-Related Transfer Function(HRTF) for each ear of the user to simulate surround sound.

Similarly, this is the most logical breakdown for the modules in the system. Therefore the main modules are the Dolby Decoder and the HRTF calculation unit. There are other modules used incorporated into the model, namely the Spatial Location Interface, but these are not the primary focus of the project and the degree of planning and expansion that goes into preparing them will depend greatly on the primary modules in the system.

The module diagram corresponding to the system can be seen below. It is a very primitive diagram at this point as each module lacks a full specification and the necessary control signals have yet to be determined. However, the flow of audio data through the system will follow a path almost identical to the main path described, in the diagram.

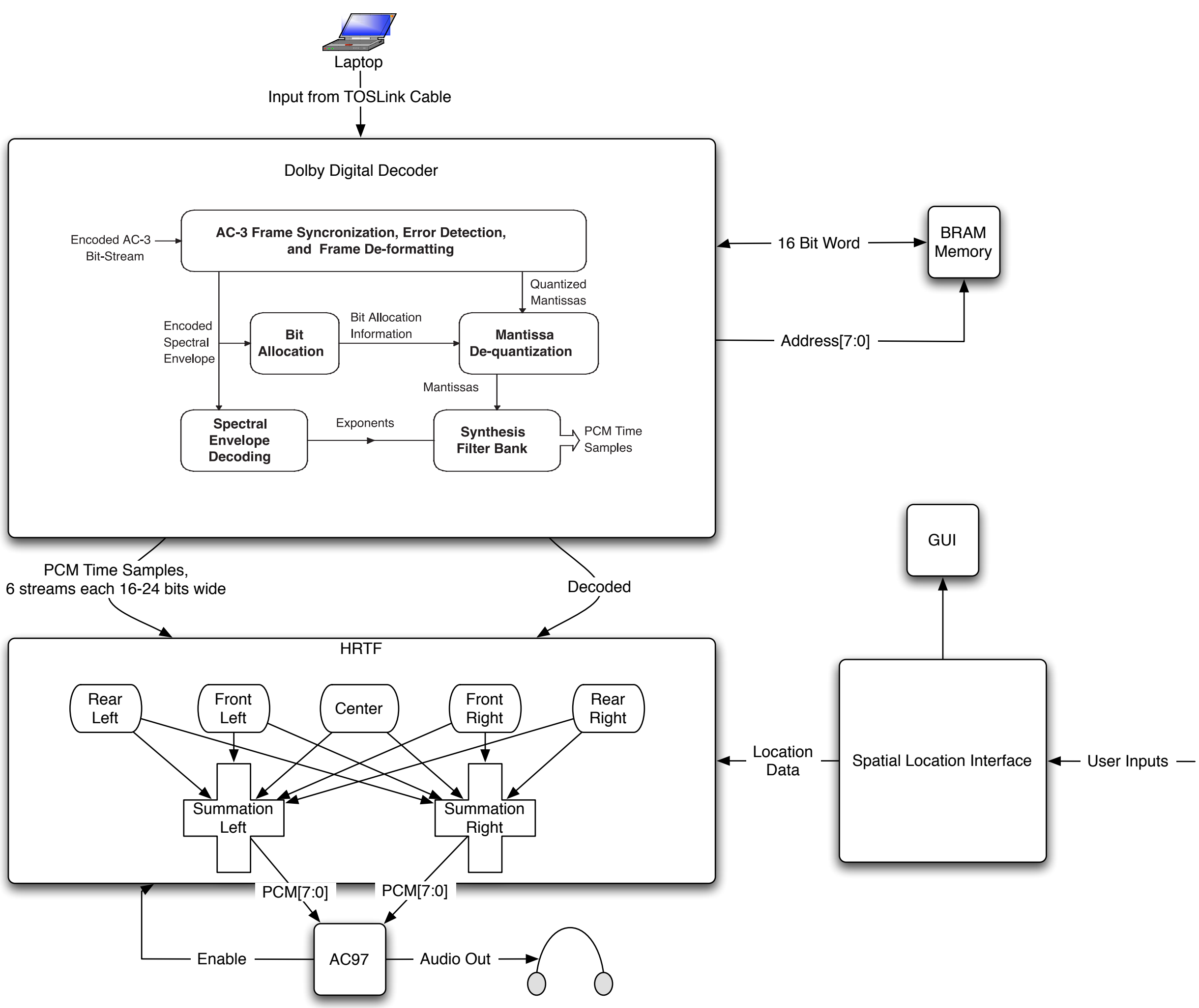

Figure 1: Primitive Module Diagram of the Major Components of the System

<u>Dolby Digital Decoder</u>

The signal we will be processing will be brought to the board via a TOSlink cable through through a TOSLink receiver package that takes the light pulses and converts them directly to digital signals at a theoretical maximum of 13.2 megabits per second, but for this application we only are interested in 320 kilobits per second to 640 kilobits per second, or , similarly, we have to clock it to at least 640kHz and then compare the synchronizer packet at the head of each frame.

The Dolby Digital(AC-3) signal is a variable length encoding depending upon the frequency of the samples in the original work and the bit quality in the original work. The encoding is very high gain meaning that a studio master copy of a 5.1 channel audio program is converted from a PCM representation requiring more than 5 Mbps (6 channels × 48 kHz × 18 bits = 5.184 Mbps) into a 384 kbps serial bit stream in AC-3. Despite this major reduction in size, a memory module will still be needed to store the data. Clearly the on chip BRAM is enough for streaming data, since we don't want more more audio lag than necessary. We will store the data in 8-bit segments because each Sync Frame is a multiple of the word-length. Then we can read out a sync frame as necessary to buffer the audio effectively.

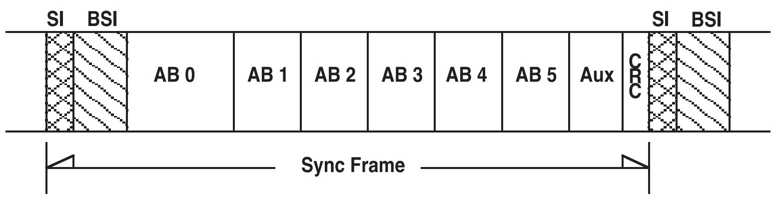

Figure 2: Sync Frame Space Allocation and Ordering

Thus the first step in building the compliant PCM data stream from the encoded one is to check whether the data stream is not AC-3 compliant and if it is to continue with the processing by determining the control bits of the for the frame, which are numerous. The bit allocation information can be found in the Bit Stream Information(BSI) packet in the frame as illustrated by the Sync Frame structure pictured in Figure 2. However, if the stream does not comply with the standard the system will output no sound.

The packets AB0-5 are the 6 audio streams in the frequency realm and encoded using a exponential-modulus scheme to make the number of collisions small while keeping the bit-length small. To counteract this there is some processing that occurs before the inverse FFT is taken, namely taking the logarithm of each stream, and decoupling the streams from one another. To finish the decode, the inverse FFT is done on each stream and certain streams are melded together giving us the final 6 streams of PCM data. The exact order of the decode can be seen in Figure 3.

There is also error detection in the Sync Frame located in both the BSI and in the CRC portions. This error detection does not correct the error but merely signals the system that it should be silent for this frame. It is roughly broken into the first 5/8 of the sync frame and the last 3/8, though without the first 5/8 there will be no audio data from the system. Merely checking the first 5/8 however gives an error rate less than one percent.

The most difficult portion of the AC-3 Decoder will most likely lie in the bit allocation. Since there is a continuous stream of data flowing through it is vital to keep the proper information for each sync frame. Most of the processing information can be readily computed from lookup tables or ISE-made programming blocks. It is attaching the proper information to these blocks and determining whether there is error in the sync frame that will be the challenge.

Testing for this module will be a challenge. The simplest, and lest beneficial, will be the speaker test and seeing if it produces sonic output that I believe to be correct. Time permitting, a more precise testing procedure will be written in Matlab or similar language to the same specification as the hardware then by passing in some data, we will examine the output of both systems for equality.

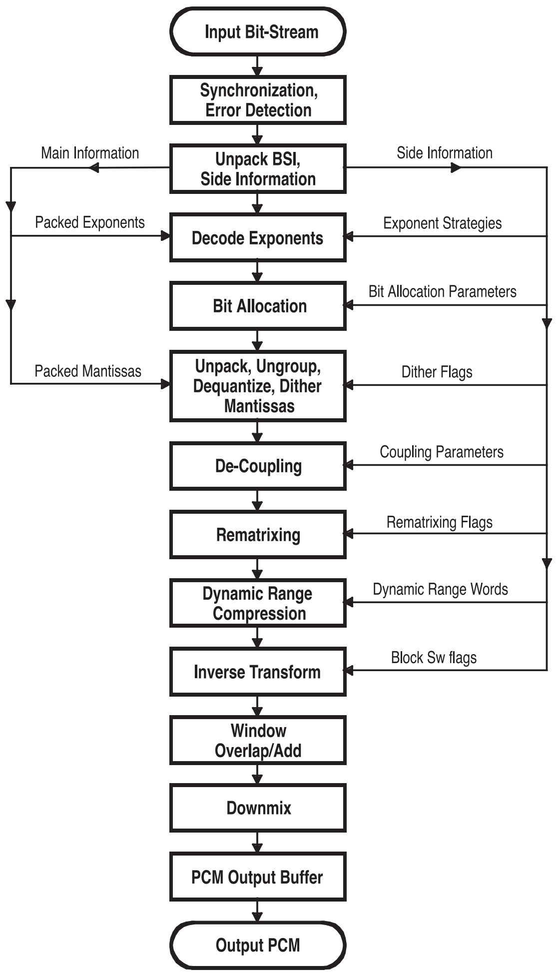

Figure 3: Dolby Digital Decoding Main Operation Ordering

<u>HRTF</u>

Now that we have the individual data streams and there is a well-defined specification for the ideal location for placement around a user in a particular orientation we can calculate the appropriate time and frequency shifts for the source to make it appear that the audio on that channel is coming from any location within the sphere around the user.

The reason this technology works is that sound signals do not arrive at the same time or with the same frequency at each ear. This is in fact different for every individual, but without extensive individualized user testing there can not be anyone ideal solution produced. The solution is to take the average of a wide variety of different users to get an idea of a good average value for coefficients. This information is freely available and we will be consulting several of them to get the best fit possible.

This processing basically boils down to a FFT of each source for each ear and combined with the signal as it would appear to a microphone at the center of the head. This then is converted back to PCM data that has been shifted in frequency and phase so that it appears to be coming from a virtual speaker. There are also small effects that impact our perception of where data is coming from, namely the shape of one's chest and the shape of the ear and ear cavity. Much of this is dealt with in Pinna multipliers where 10-20 different calculations, representing different measurements of the ear, sum to form a better image of the sound as it would appear from a surround sound system.

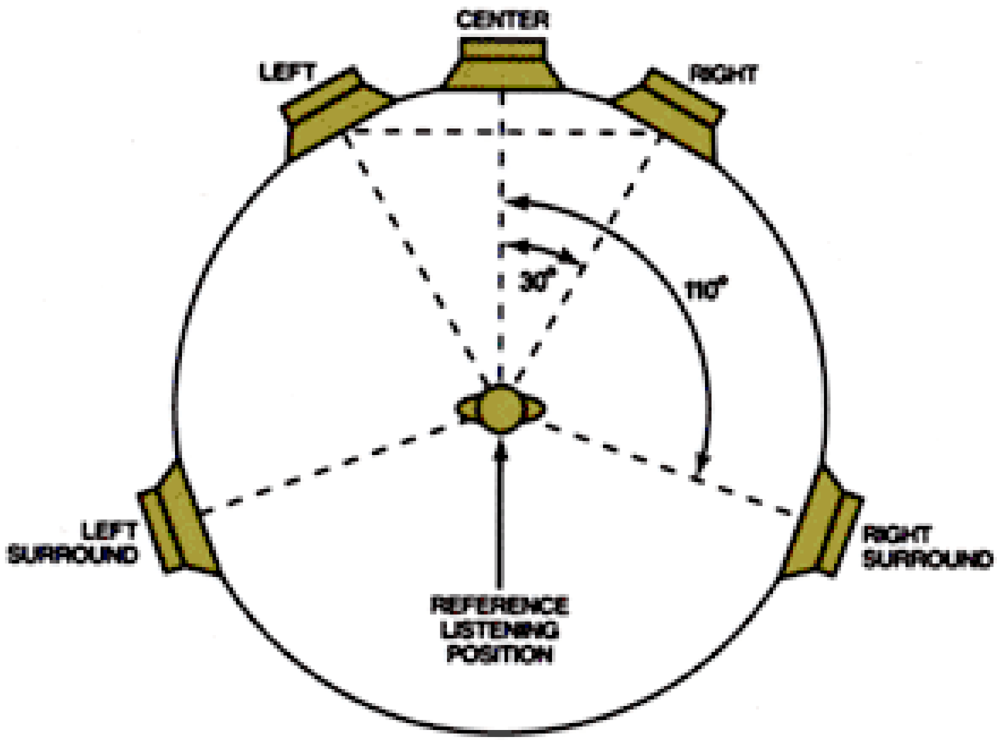

Figure 4: Ideal speaker placement in a Dolby Digital system excluding the subwoofer because low frequency suffers from poor directionality.

Much of this is a black art, but there has been some research into the area. The main issue with this approach is that it takes considerable time to synthesize this data then combine it, for multiple streams in real time. Therefore, I will be looking into models that approximate this data or reference lookup tables for it to improve efficiency. Correspondingly, I feel that this inefficiency versus quality will be the hardest portion of this project to rectify.

Testing will be done on this module utilizing freely available, on the internet, Matlab modules that calculate the HRTF data and Pinna values for certain models. This will allow us a testbed to make sure that each module is working correctly on a bit level. There will also be user tests where they are asked to identify where a sound is coming from in the space around them and compare that to where the source is producing the data to originate from. This second part will also be vital in order to get the best model possible for the widest range of users because we are not able to offer a personalized HRTF.

<u>Spatial Location Interface</u>

As this is not the main focus of the project, I am just hoping to have a GUI on the screen depicting the location of the speakers on the screen and the location and orientation of the user. The user will be able to modify an the direction they are facing in the GUI and their location in the surround sound setup, resulting in an change in the HRTF calculation so that it is correct for the user's new orientation.

The hardest part of this will be calculating the rotational difference from where the user is relative to where the sound should be coming from in their new orientation. I do no think the GUI itself will be challenging to build as it is just six to ten sprites that are simple in shape and coloration.

<u>Outlook</u>

There are a lot of daunting tasks that lie ahead, the first and foremost is to decode the Dolby logic into a hardware. This is an exciting and worthwhile project as avid moviegoers everywhere can attest. This project will be a necessary accessory for the traveling movie watcher, video gamer, or audiophile to take full enjoyment in their media.