AMBISONICS
Flexible surround sound system
AMBISSONICS

• Method of encoding mono audio and 3D location information.

• Completely arbitrary speaker configuration and layout.

• Encoded audio plays back on all systems so that virtual sources retain their location and movement given multiple speaker configurations.

• Applications in live theater, consumer home-electronics, amusement parks, digital-cinema, and car audio.
MY WORK

Tuesday, November 17, 2009
MY WORK
USEFUL TERMS

• **Virtual Sources**: Mono audio that will be spatialized.

• **Physical Sources**: Physical speakers in the venue whose output contributes to the ambisonic sound field.

• **Encoded stream**: An audio stream which contains location data for one or more virtual sources.
SIMPLE AMBISONIC SYSTEM

Explosion
Ambisonic Encoder

Airplane Sound
Ambisonic Encoder

Footsteps
Ambisonic Encoder

Summing Device

Ambisonic Decoder
Speaker

Ambisonic Decoder
Speaker

Ambisonic Decoder
Speaker

Ambisonic Decoder
Speaker

Ambisonic Decoder
Speaker

Ambisonic Decoder
Speaker
ENCODING & DECODING

• **Encoding**: Take incoming audio sample, make 16 copies of the sample and multiply each by a different coefficient calculated using location coordinates.

• Encoded stream is represented by 16 channels resulting from each sample’s multiplication by coefficients which change over time. Encoded streams may be summed together.

• **Decoding**: To decode the stream for a particular physical source, multiply each channel in the encoded stream by a corresponding coefficient calculated using physical source coordinates and sum them together.
ENCODING & DECODING

- Spherical Harmonic Coefficients
- Really slow to calculate in hardware

<table>
<thead>
<tr>
<th>Order</th>
<th>Name</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0th</td>
<td>W</td>
<td>$\sqrt{2}/2$</td>
</tr>
<tr>
<td>1st</td>
<td>X</td>
<td>$\cos(\theta) \cos(\delta)$</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>$\sin(\theta) \cos(\delta)$</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>$\sin(\delta)$</td>
</tr>
<tr>
<td>2nd</td>
<td>R</td>
<td>$1.5 \sin^2(\delta) - 0.5$</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>$\cos(\theta) \sin(2\delta)$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$\sin(\theta) \sin(2\delta)$</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>$\cos(2\theta) \cos^2(\delta)$</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>$\sin(2\theta) \cos^2(\delta)$</td>
</tr>
<tr>
<td>3rd</td>
<td>K</td>
<td>$\sin(\delta) (5.0 \sin^2(\delta) - 3) - 0.5$</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>$\cos(\theta) \cos(\delta) (5 \sin^2(\delta) - 1)$</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>$\sin(\theta) \cos(\delta) (5 \sin^2(\delta) - 1)$</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>$\cos(2\theta) \sin(\delta) \cos^2(\delta)$</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>$\sin(2\theta) \sin(\delta) \cos^2(\delta)$</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>$\cos(3\theta) \cos^3(\delta)$</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>$\sin(3\theta) \cos^3(\delta)$</td>
</tr>
</tbody>
</table>

Table 1. The coefficients for encoding/decoding into third order Ambisonics
6.1 I I I FINAL PROJECT!

- Create an ambisonic encoder and decoder that works in real-time.
- Stream virtual sources from the computer to the labkit via USB.
- Output via 8 channel Texas Instruments PCM1681 DAC.
- Control via buttons using preset locations and speed lock on interpolation.

Tuesday, November 17, 2009
TIMING

• System Cycles (27Mhz): 562.5 system cycles for each audio cycle (48Khz Audio)

• USB_Reader: 8 system cycles per byte, 8 bytes for each 4 channel sample.

• ZBT Memory: 2 memory cycles per read, 128 cycles for 64 coefficients (can be run at higher clock speed)

• Multipliers: 4 cycle latency (64-bit by 16-bit CoreGen)

• Audio Timing: TI PCM-1681 uses a 128x f_s clock speed (~6Mhz)
ALTERNATE DESIGNS

• First tests using AC97 output (could be used in final project across two lab kits)

• One channel of USB input direct to AC97

• Full Ambisonic panning across stereo outputs using AC97

• One channel of new DAC, using input from USB

• Alternative timing options (BRAM to store presets only)

• Interpolation system for coefficient lookup (more resolution)
QUESTIONS
TIMELINE

• Nov. 19: Finish USB to AC97 system
• Nov. 22: Finish 2 channel AC97 Ambisonic System
• Nov. 24: Finish Coordinate UI/Display System
• Nov. 25: Finish USB direct to PCM-1681
• Nov. 29: Finish 8-channel
• Dec 7: Debugging and Report Complete