

AMBISONICS

Flexible surround sound system

AMBISONICS

- 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

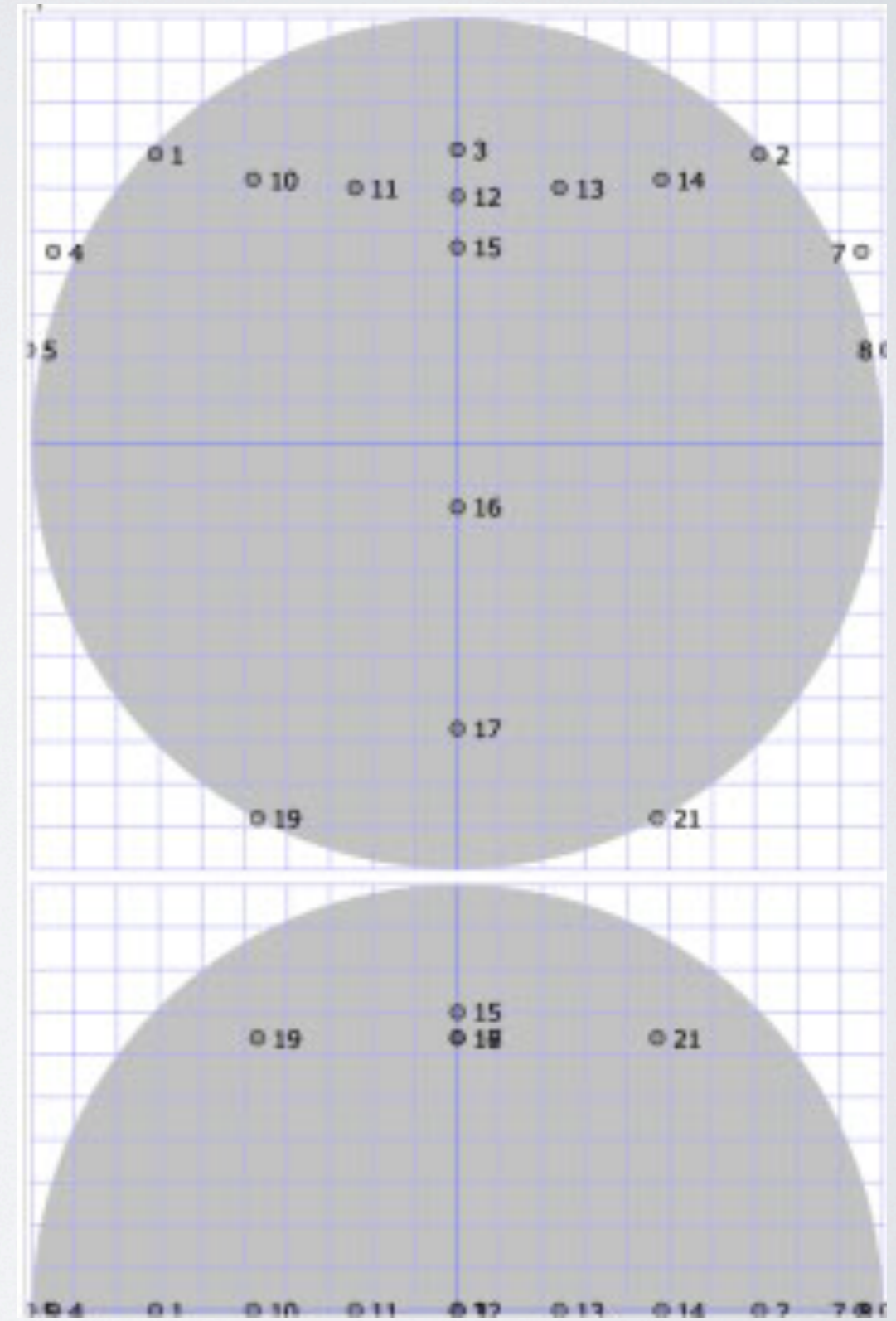
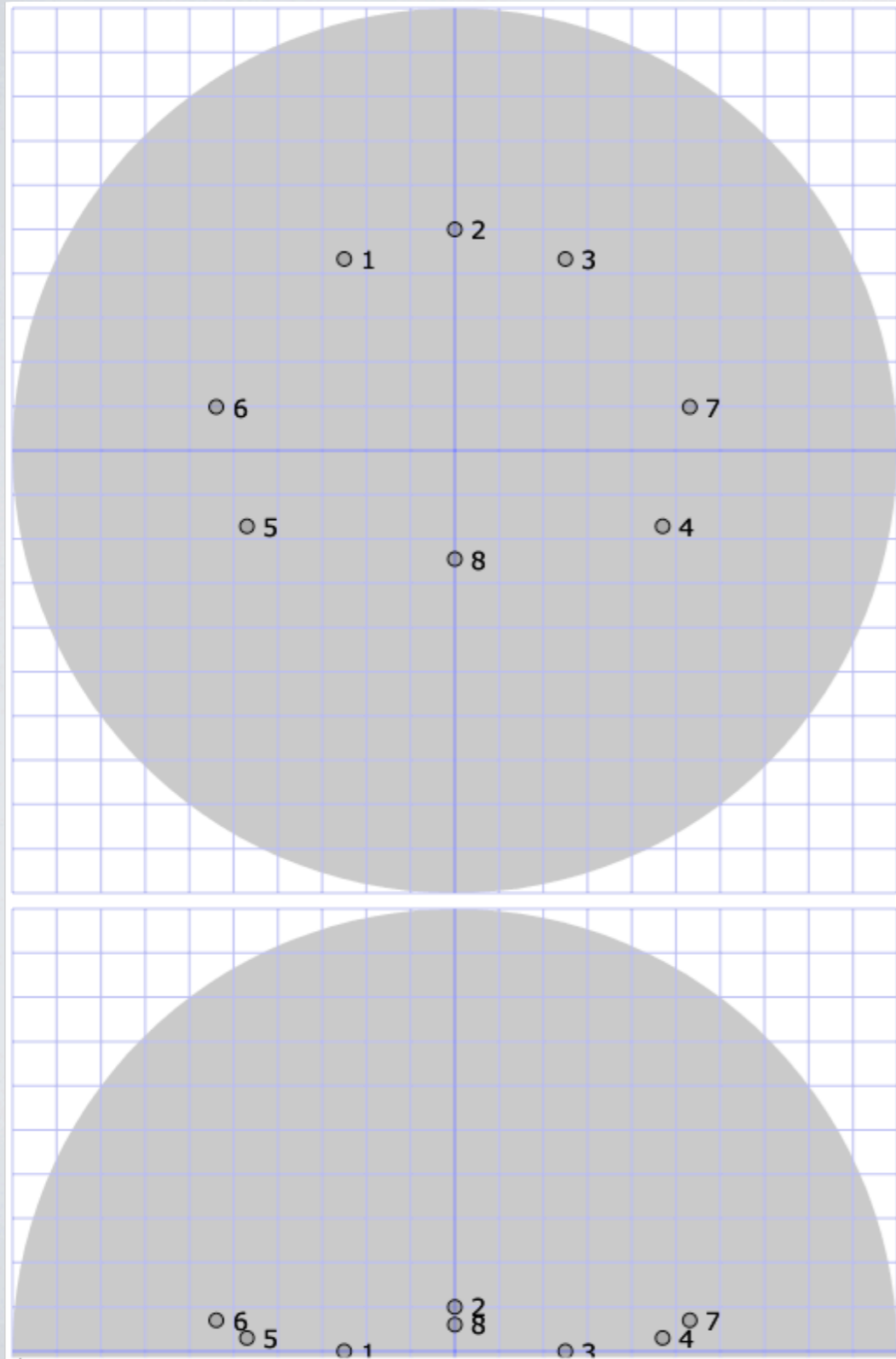


MY WORK

The image displays a complex Max/MSP patch titled "morning_side0" designed for Kontakt 3. The patch is divided into several functional areas:

- Input and Metering:** On the left, there are three metering windows: "/inputmeters" (showing 16 channels), "/ambimeters" (showing 16 channels), and "/outputmeters" (showing 16 channels). Below them is a "/control.1" window with "Audio (On)" and "Panic" buttons.
- Speaker Location Visualizations:** The central part of the patch features two large circular diagrams labeled "kontak.scope" and "live.scope". Each diagram shows a 16-speaker layout with speakers numbered 1 through 16. A third, smaller diagram labeled "speaker_locations" is at the bottom left, showing a similar layout on a grid.
- Speaker Configuration:** A sub-window titled "Speaker Configuration" contains a "speaker.setup" object with a "radius" parameter set to 10. It includes a "p.adapt" object and a "print" object. A note states: "Loudspeaker positions are not stored as part of this module. They will be maintained by a separate module to be added in the future." Below this is a "/speakers-patch" window with "Listen (Off)", "Load Cue Script", "View", "Get State", and "Save Cue Script" buttons.
- Speaker Location Data:** At the bottom center, there is a "/speakerlocations" window displaying a table of speaker coordinates. The table has 16 rows and 4 columns of numerical values.
- MIDI and Kontakt Integration:** On the right, the patch connects to Kontakt 3. It includes an "r.midi.to.kontakt" object, a "pak.midievent" object, and a "vst~ 'Kontakt 3 16out.vst'" object. A "i.program" object is also present. Below these are "i.texture-level" and "i.kontaktLevel" objects.
- Performance and Control:** On the far right, the Kontakt 3 interface is visible, showing a piano roll, a mixer with four aux channels, and a keyboard. At the bottom right, there is a detailed view of Kontakt's mixer controls, including EQ, dynamics, and volume faders.

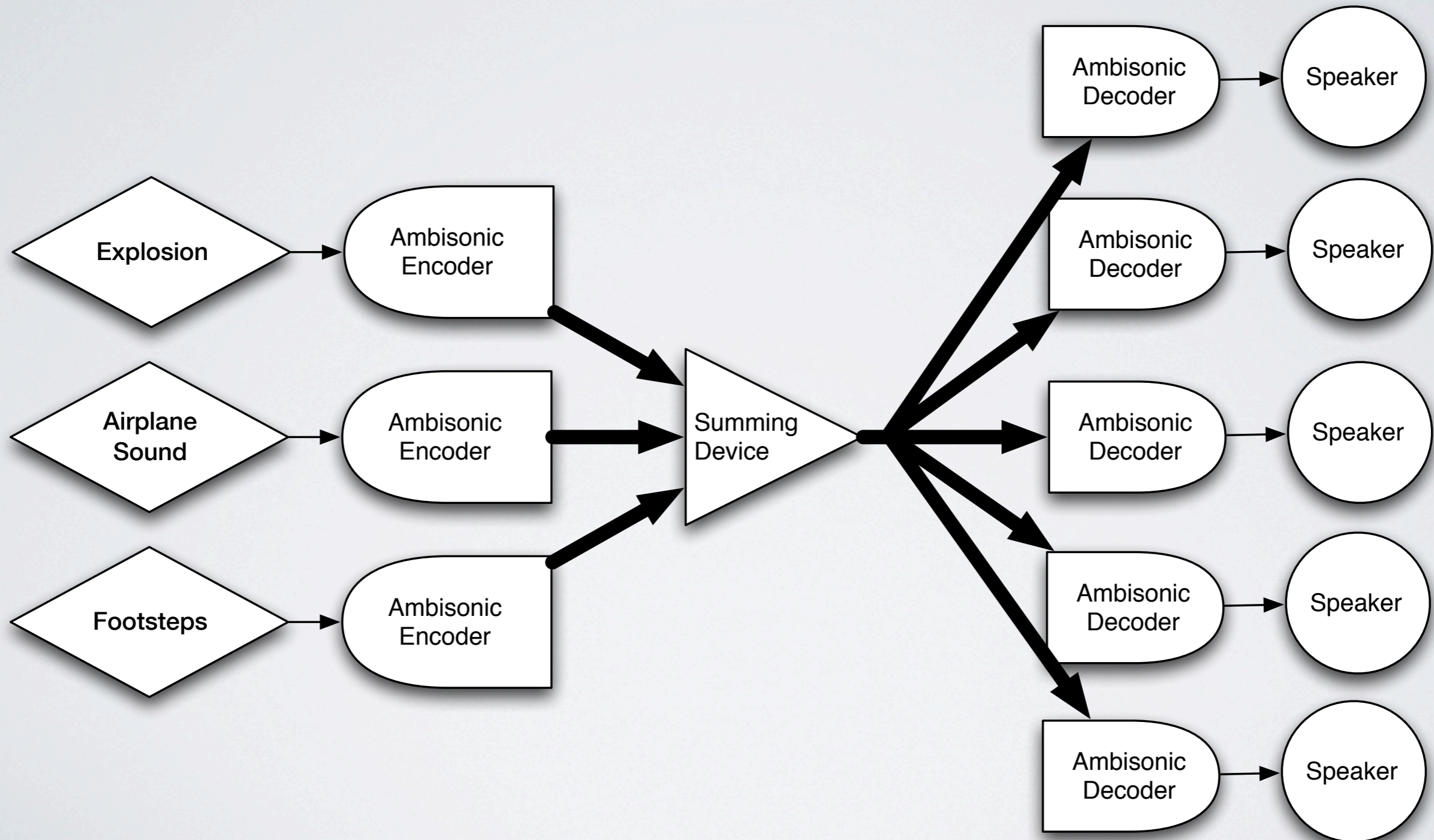
MY WORK



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



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

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

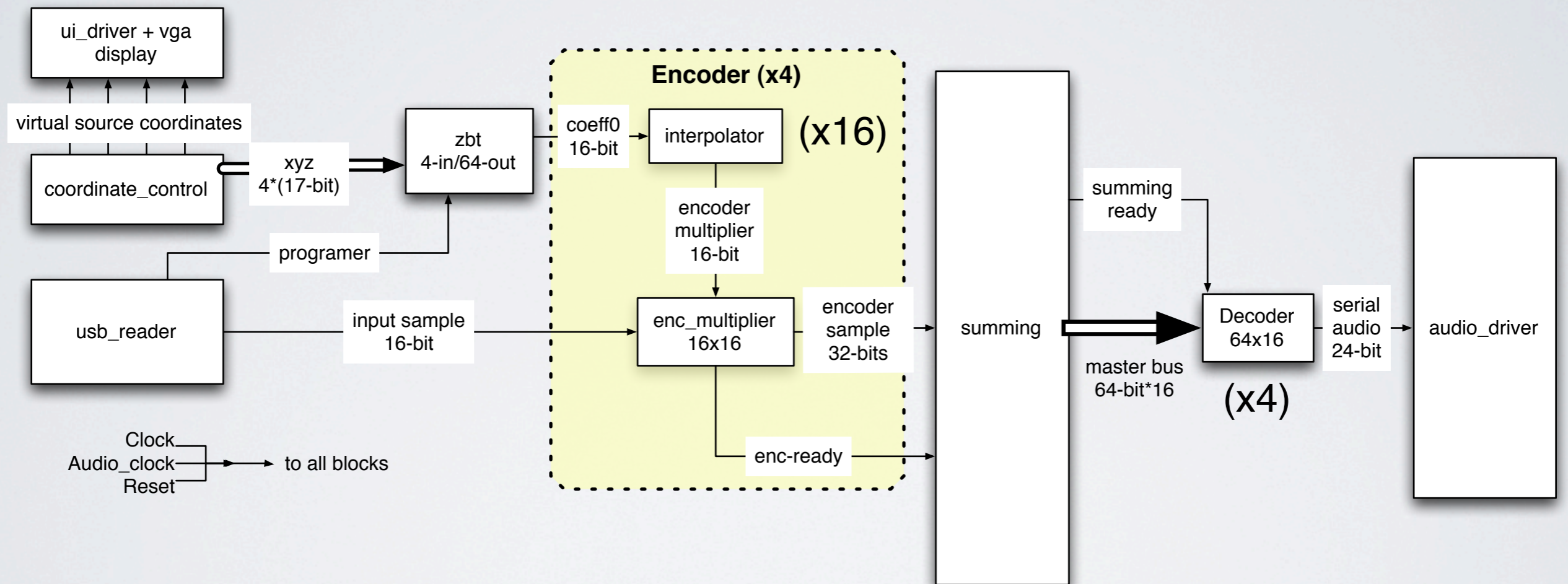
Table 1. The coefficients for encoding/decoding into third order Ambisonics

- Spherical Harmonic Coefficients
- Really slow to calculate in hardware

6.111 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.

BLOCK DIAGRAM



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 $128 \times f_s$ 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