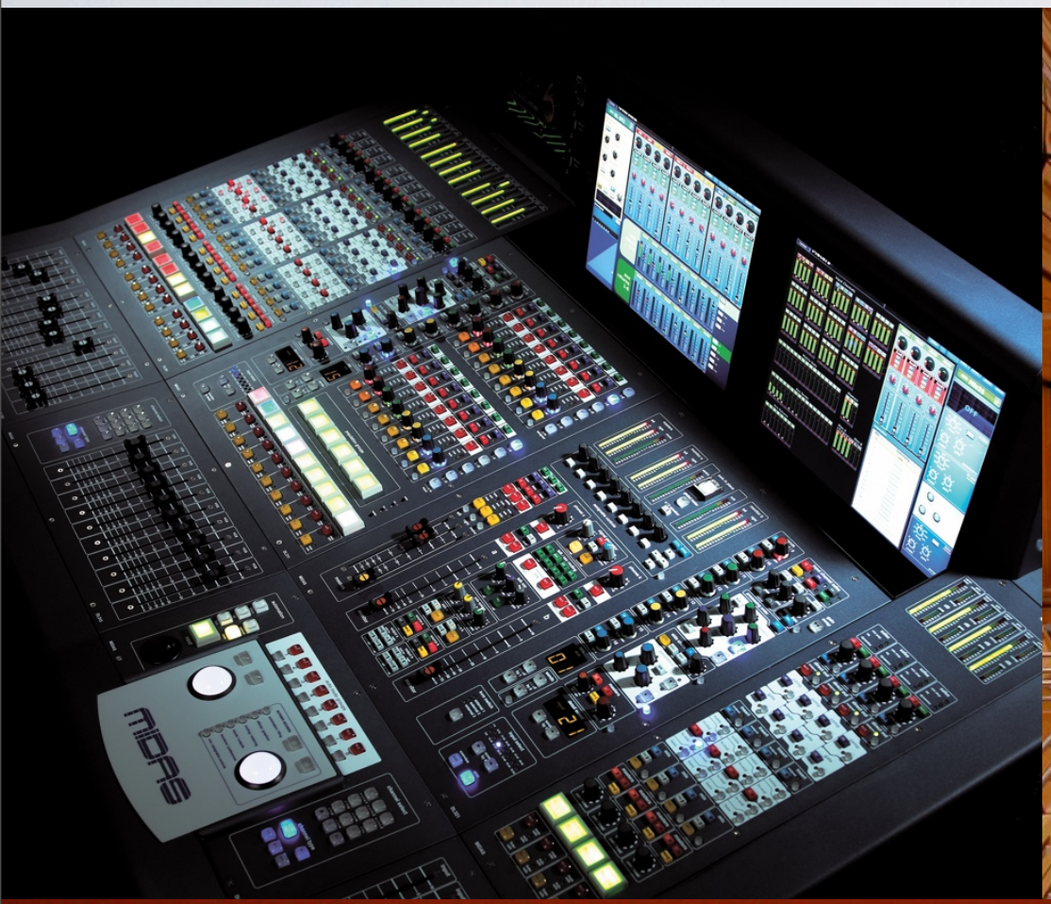
AMBISONICS

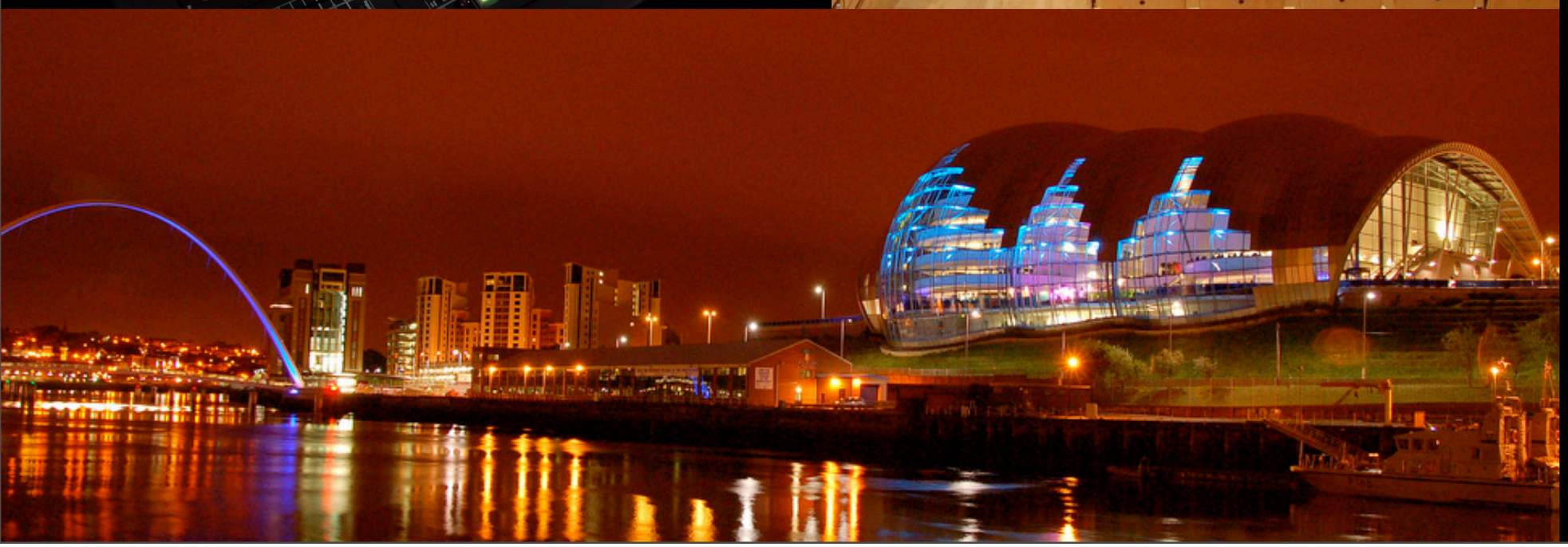
Flexible surround sound system

AMBISONICS

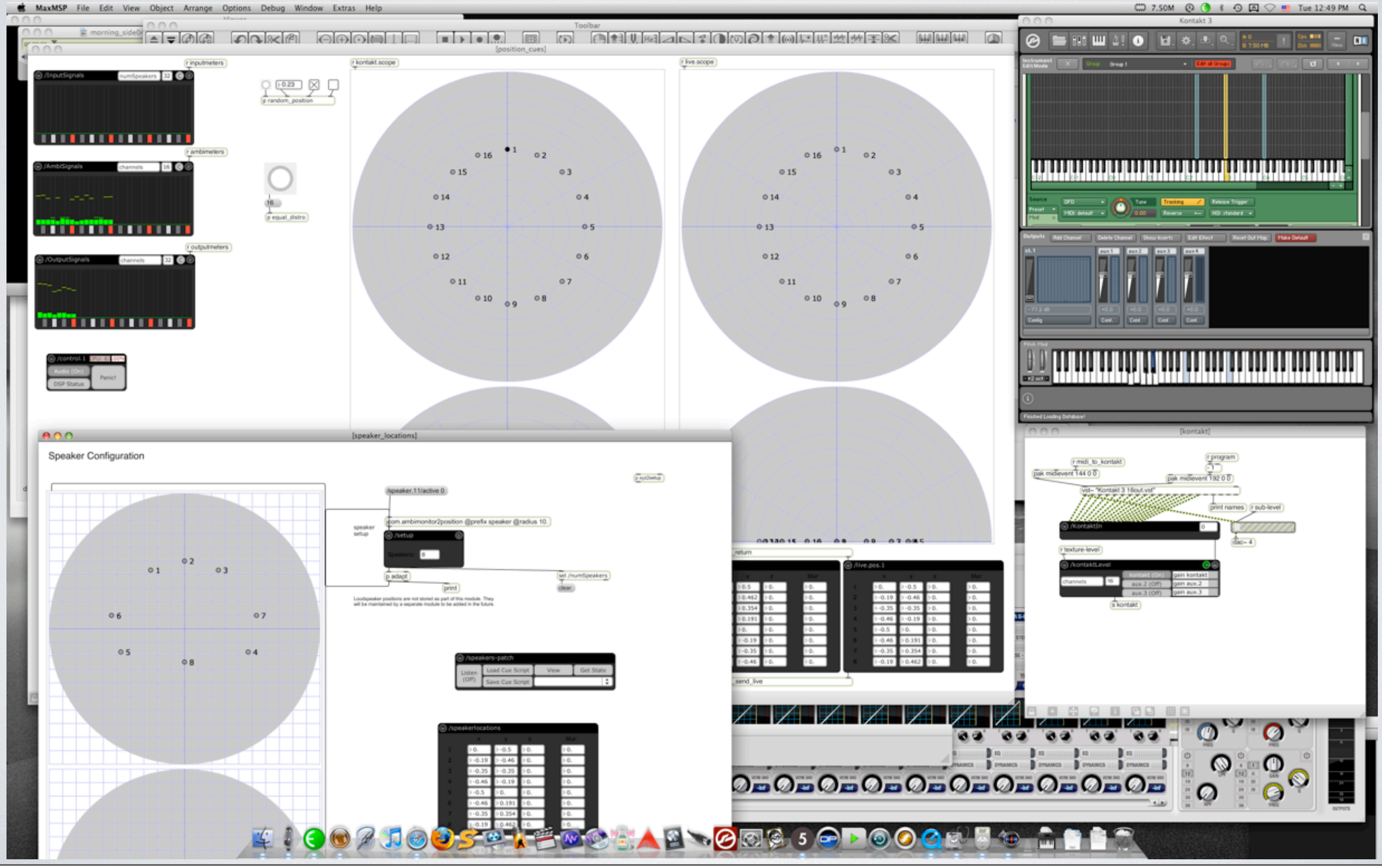
- Method of encoding mono audio and 3D location information.
- Completely arbitrary speaker configuration and layout.
- Encoded audio plays back on all systems so that virtual sources retain their location and movement given multiple speaker configurations.
- Applications in live theater, consumer home-electronics, amusement parks, digital-cinema, and car audio.

MY WORK

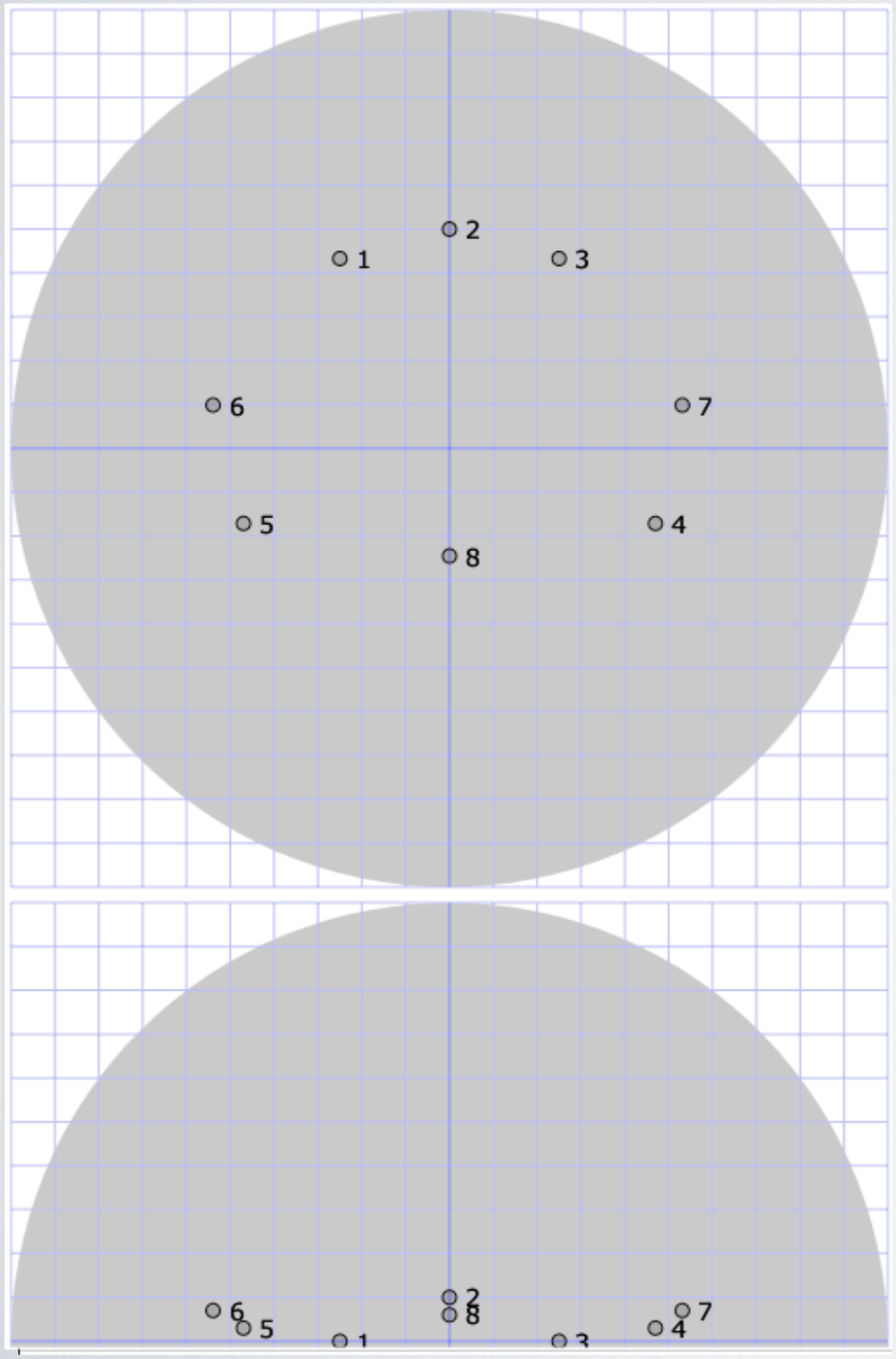


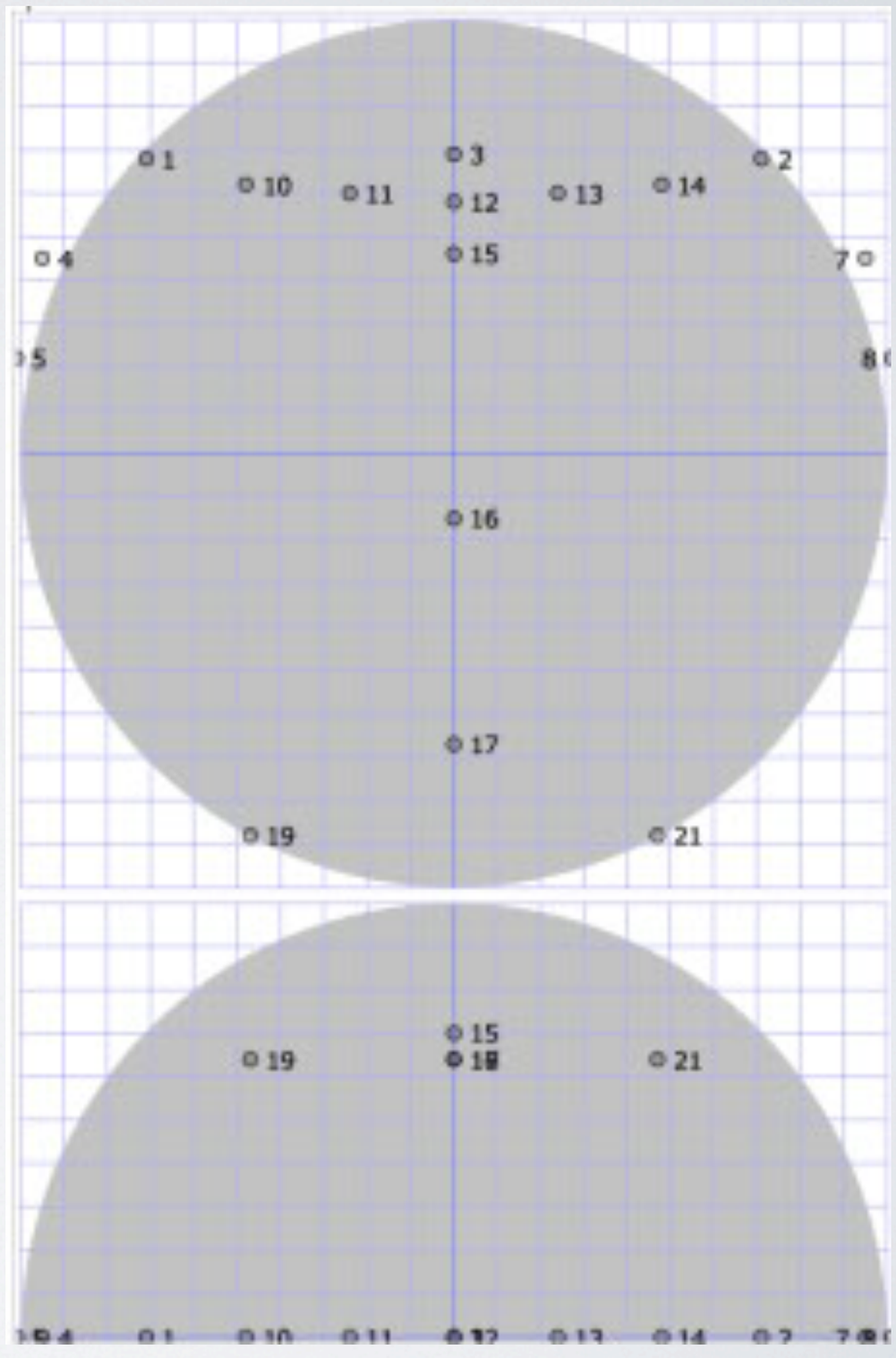


MY WORK


MaxMSP File Edit View Object Arrange Options Debug Window Extras Help
Toolbar
[position_cues]
r inputmeters
/InputSignals
numSpeakers
r ambimeters
/AmbiSignals
channels
r outputmeters
/OutputSignals
p random_position
p equal_distro
r kontakt.scope
r live.scope
/control.1
Audio (On)
DSP Status
Panic!
[speaker_locations]
Speaker Configuration
/speaker.11/active 0
speaker setup
/setup
p adapt
print
sel /numSpeakers
clear
/speakers-patch
Listen (Off)
Load Cue Script
View
Get State
Save Cue Script
/speakerlocations
Kontakt 3
[kontakt]
r midi_to_kontakt
r program
pak midievent 144 0 0
pak midievent 192 0 0
vst~ "Kontakt 3 16out.vst"
print names
r sub-level
/KontaktIn
dac~ 4
r texture-level
/kontaktLevel
s kontakt
/return
/live.pos.1
send_live

MY WORK





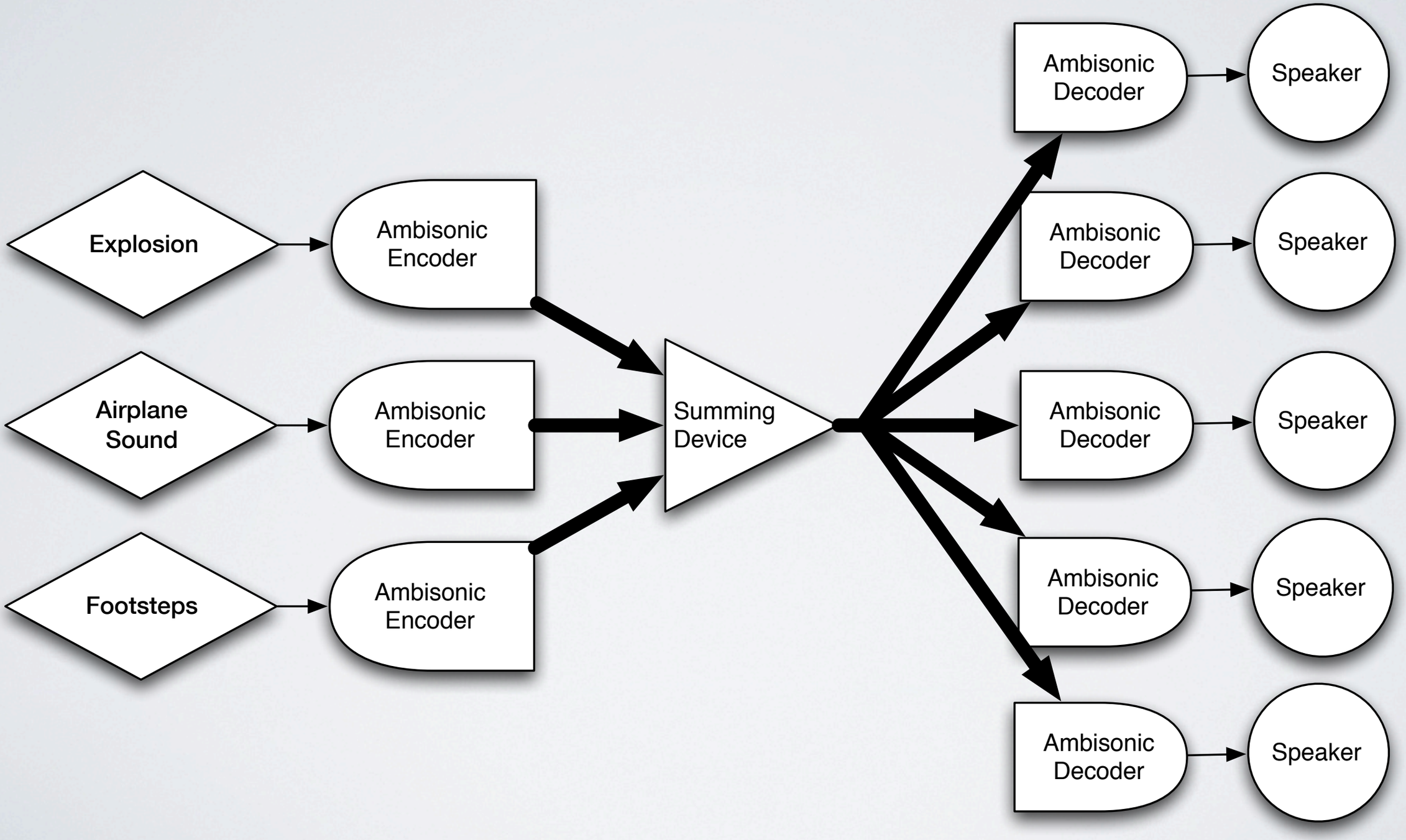
USEFUL TERMS

- **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

Order	*Name*	*Coefficient*
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!

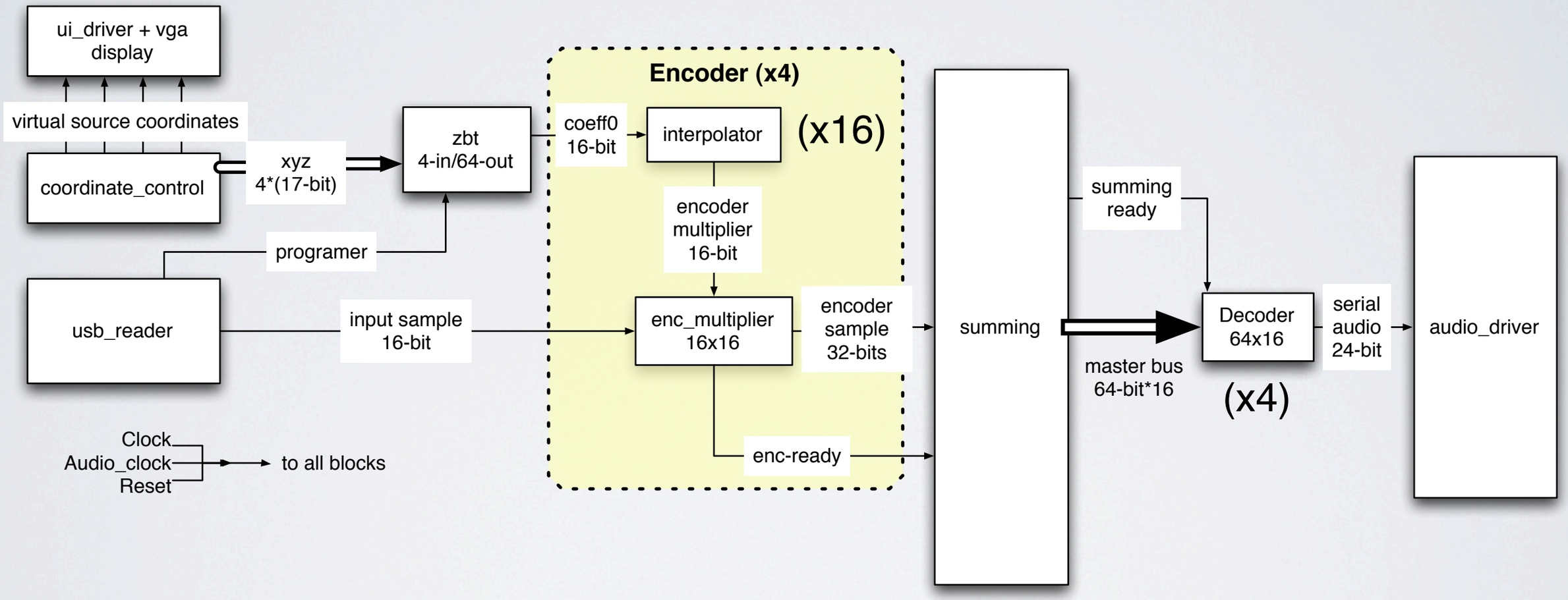
- Create an ambisonic encoder and decoder that works in real-time.

- 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 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