# Before we build ATLAS, we need to build a city...

(the Keynote and movie version of these slides has embedded audio)

# Before we build ATLAS, we need to build *this* city…

This project will likely *not* involve your favorite music. Instead, we'll take inspiration from Jahred's slides, and revisit an era of music known for its over-the-top music production…

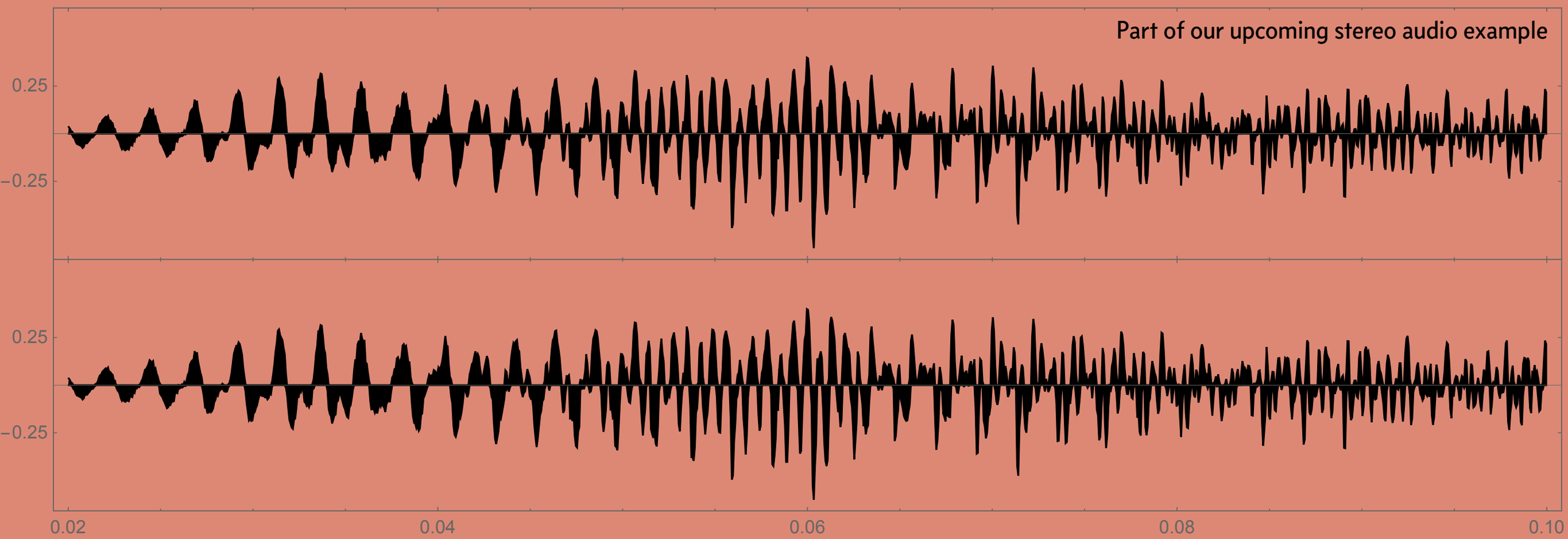
*Rolling Stone*'s "worst songs of the 1980s by a wide margin", "what practically killed rock music in the '80s". "The group's co-lead singer Grace Slick has called it 'the worst song ever'"— Wikipedia

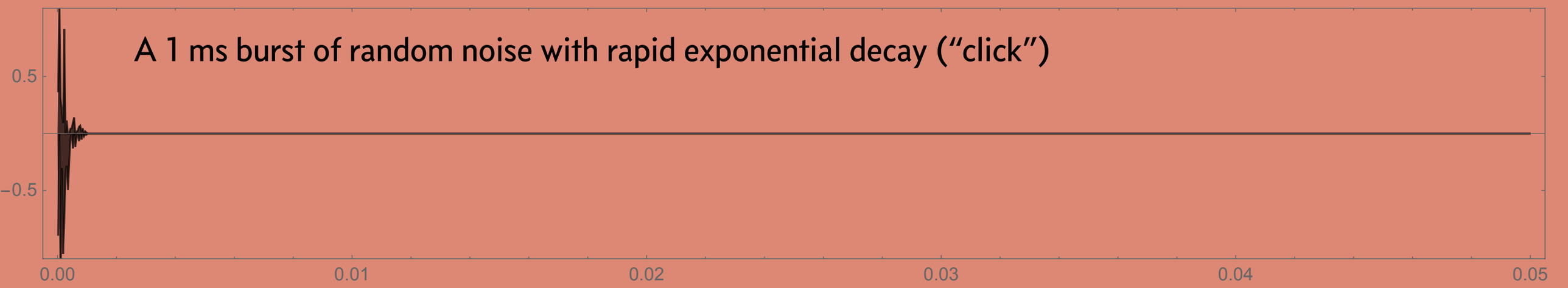
Stills from the music video for "We Built This City" by Starship (1985) including Abe Lincoln as a backup vocalist

# Audio data

Raw digital audio data consists of a time series of “samples” of an analog waveform at a fixed sample rate.

- In this project, the data are all stereo waveforms sampled at 48 kHz (.wav and .txt formats).

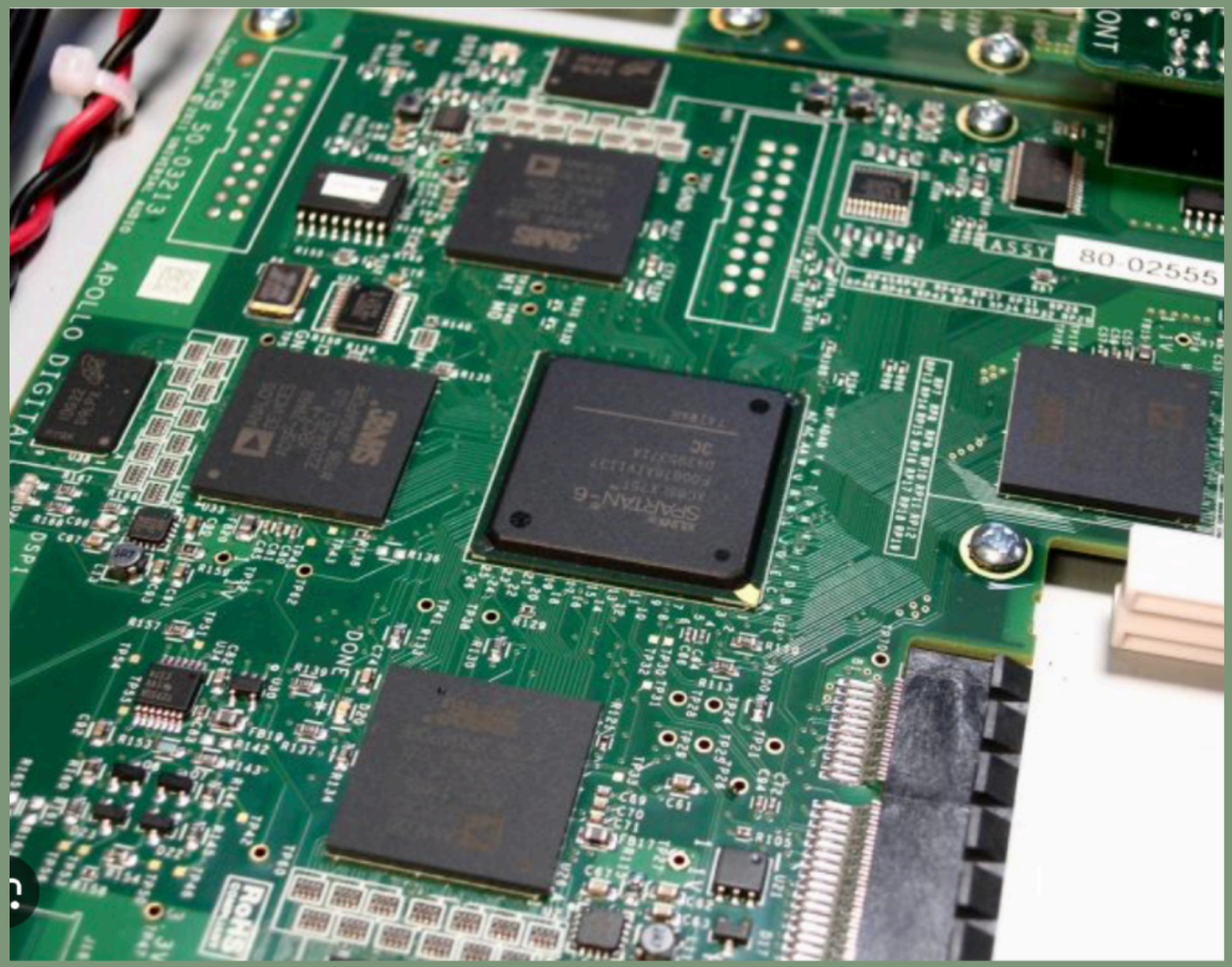


# Digital signal processing on FPGAs

FPGAs excel at manipulating audio data in real time (low latency). This can be used for “tracking” audio in a recording studio or at the “front of house” sound boards at music venues to enhance the sound that the musician or audience hears, to generate new sounds in synthesizers and other electronic instruments, etc.

If recorded in the last decade, the polished sound of your favorite artists almost certainly involves extensive digital processing, such as through DSP emulation of classical analog recording hardware.





Universal Audio (Xilinx Spartan-6, SHARC processors) e.g. as used for Billie Eilish's entire debut album (2020 Grammy for audio engineering).

## Delay effects

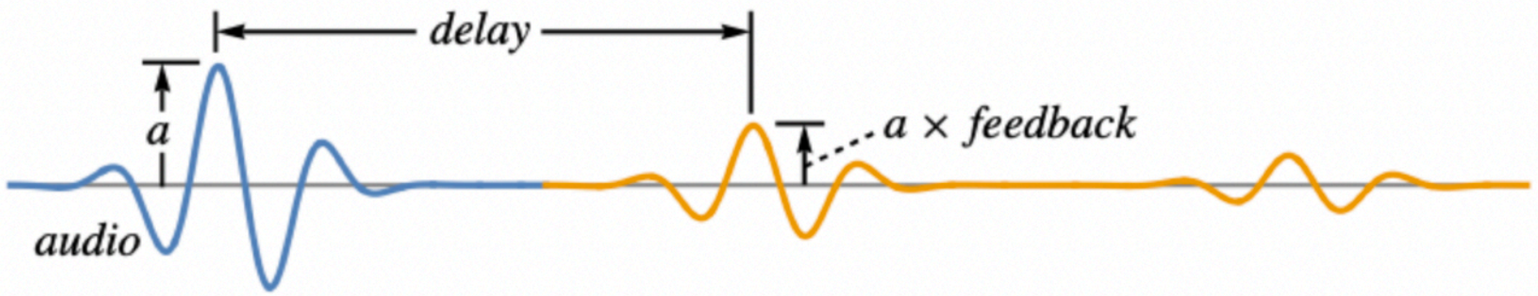
Delay/echo effects on audio are ubiquitous and come in a huge variety of tonal coloration (originally achieved with a loop of magnetic tape, analog “bucket brigade” circuits, or more high-fidelity methods).

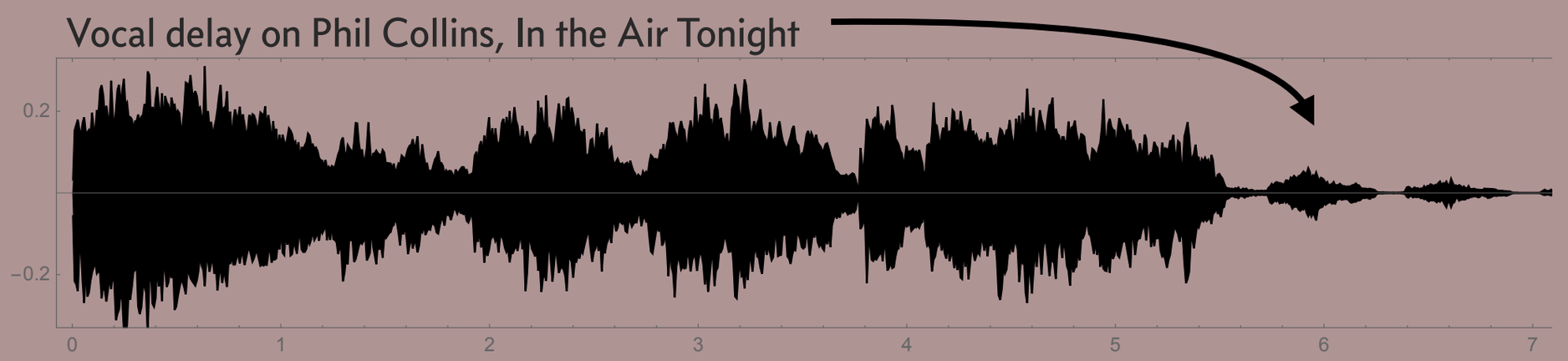
In its simplest form, a transparent “digital delay” just repeats the input waveform at a later time and mixes it back into the output. The delayed output can be renormalized, filtered or modulated, and fed back into the input for more than one repeat. This can be accomplished with a simple “delay line”, circular buffer, or many other circuits.

Delay in Mathematica

- AudioDelay simulates the effect of a decaying echo. »

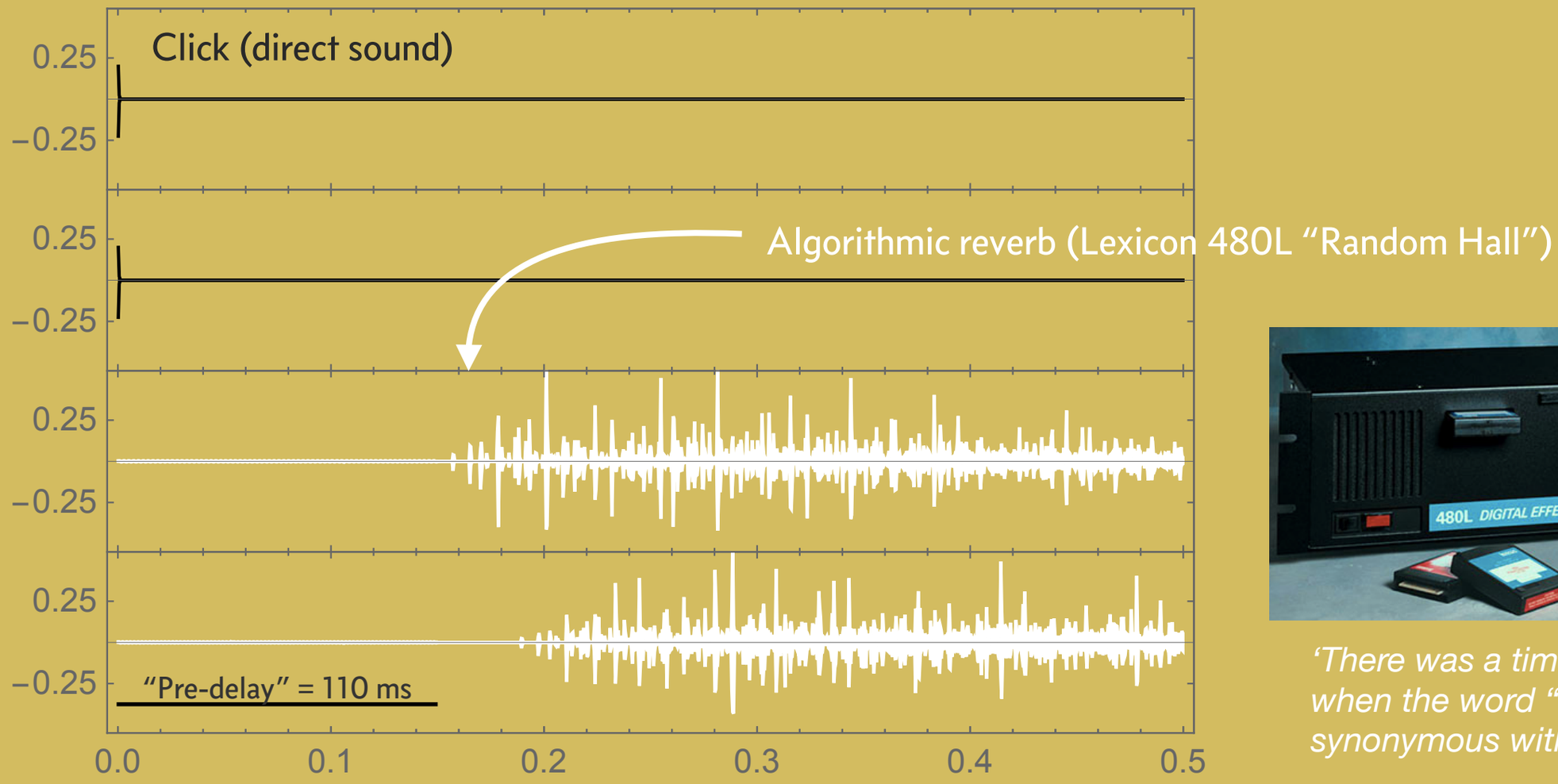
In[ ]:=



# Reverb

Reverb simulates how sound waves from a source reflect around a space and sum to the waveforms arriving at each of your ears. A realistic reverb sound has thousands or millions of reflections. The audio processing in your brain tends to focus on the earliest reflections for key information about a space and to localize the source of the sound, but all of the reflections are important for realism.





*‘There was a time in the ‘80s and ‘90s when the word “Lexicon” became synonymous with reverb.’*—Tape Op

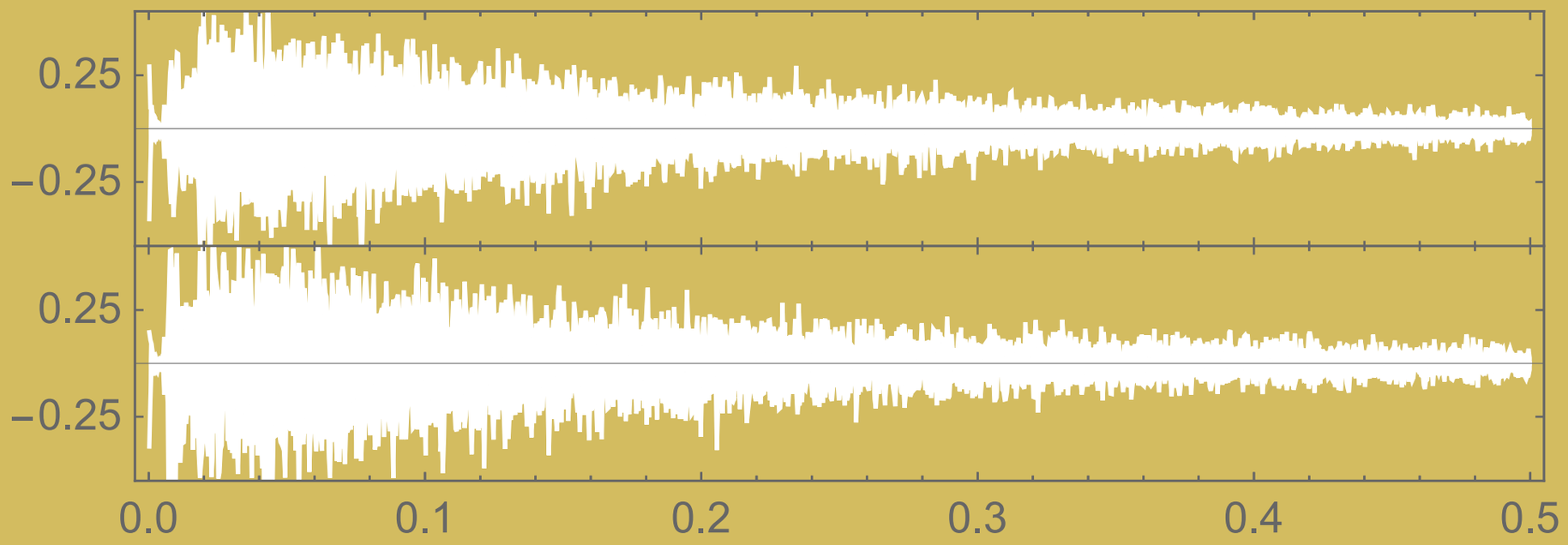
Algorithmic and analog hardware (plate, spring, chamber) reverbs were most likely the ones used on the original production in our source material, but we will use a more recent for ease of implementation in this exercise.

A modern method to capture the reverberations in a space is called “convolutional reverb,” which involves creating a brief impulsive source (delta function, clap, click, …) and recording the reverberations as a *impulse response (IR)*.

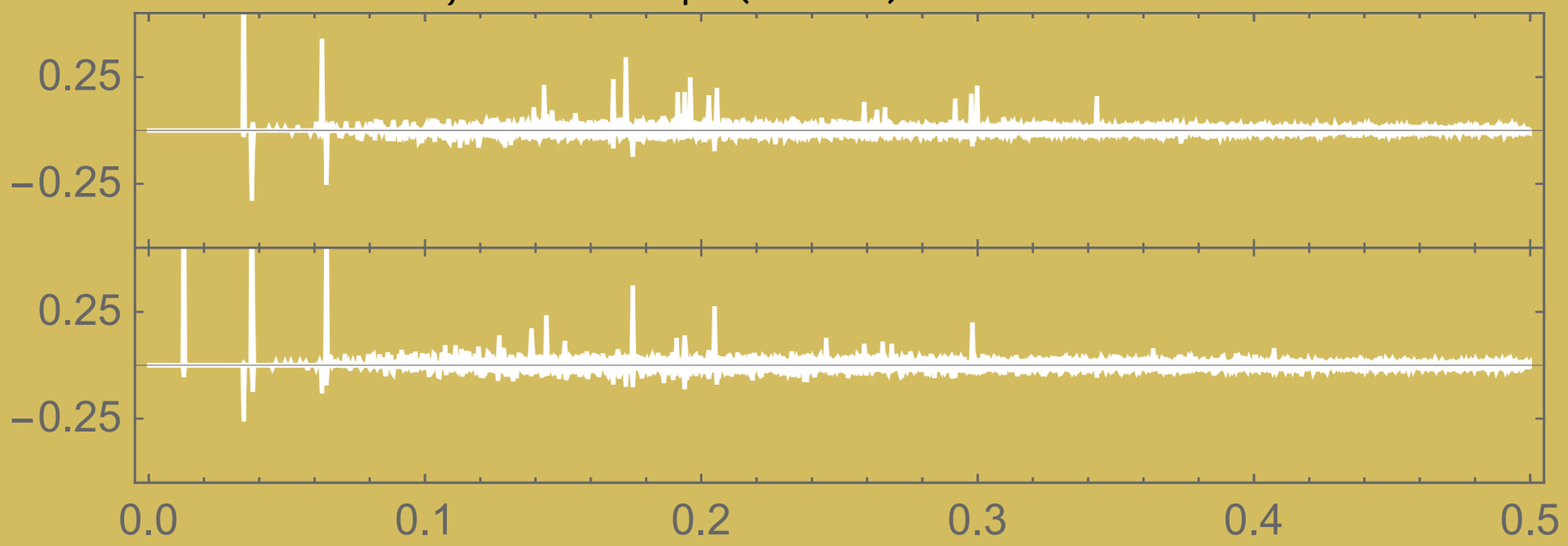
# Convolutional reverb

The git folder contains impulse response (IR) recordings for several reverbs. One can convolve a “dry” source audio waveform with the IR as a kernel to produce a “wet” audio signal, $y(n) = \Sigma_k\, x(k)\, IR(n-k)$, which is then mixed in a linear combination with the original signal.

One “hall”-style reverb example (darkHallIR.txt)




A different “hall”-style reverb example (hallIR.txt)



## Assignment

Implement (either/both) a digital delay and a convolutional reverb. Using both, you should be able to transform the dry signal *dry.txt/dry.wav* without these effects to a "wet" signal with these effects. Without much work, it is possible to reach a close approximation of the original studio production (*wet.wav*).

*Some tips*

Start with the "click" sound to validate that your implementation works, then move on to the vocal sample.

Try feeding the direct+delayed signal into the reverb and then mixing this back into the final sound.

For the delay, use a "musical" rational fraction of the tempo (144 bpm, or 416.6 ms per quarter note). To my ear, there are multiple repeats on the vocal delay, and the delay is definitely stereo. Try a "ping-pong" between the two stereo channels.

For the reverb, use one of the hall IRs. You may want to truncate ("gate") the IR sample to the first 500 ms or so and/or introduce a faster exponential decay envelope to the IR. Most reverb has some kind of pre-delay (the hall IRs do not have this; try adding something like 15 ms).

The direct sound should remain the most prominent. Try starting with a 15% wet/dry mix for both the delay and reverb effects.

There is a script in the git folder to turn your output into a .wav file that you can play.