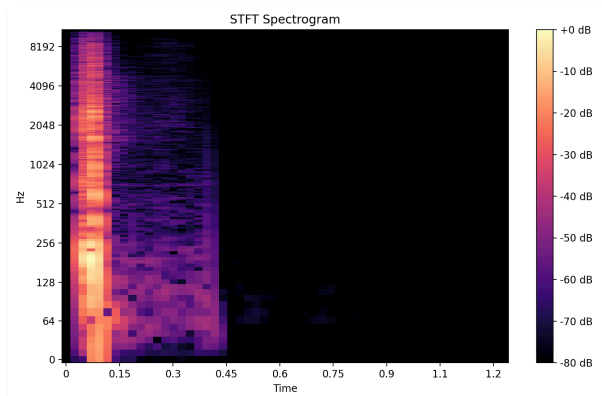# Assignment-1

Question 1. Make a Real-Time infinite loop Beat Generator.

To begin, I take the chosen audio file as input. Employing the **Short-Time Fourier Transform (STFT)**, I identify significant points in the audio, known as onsets. STFT yields a spectrogram, essentially a matrix that holds magnitude values of frequency elements for each time segment.

STFT spectrogram of a drum beat

Spectral flux, a measure of how the audio's frequency characteristics change over time, is computed next. This involves contrasting consecutive frames' magnitude spectrograms and summing the squared differences across frequency bins. By calculating the derivative of the magnitude spectrogram along the time axis and finding the **Euclidean norm of the differences**, I determine spectral flux values.
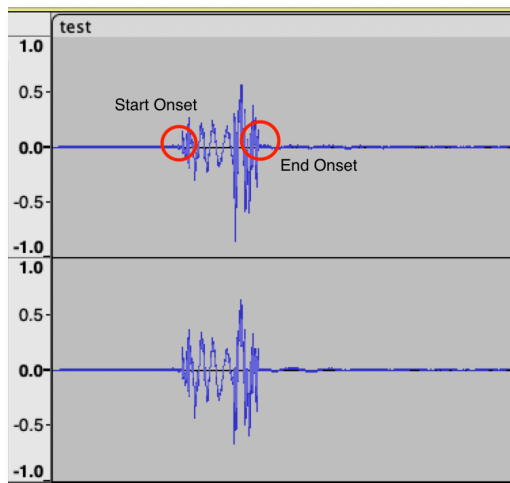
Audio Signal of a Drum beat

These spectral flux values **unveil shifts in the audio's frequency** components over time. By setting a **threshold**(I am using the default threshold as given in librosa library python), I discern onsets. Specifically, I pinpoint **two onsets**: one indicating a beat's commencement within the audio and another marking its conclusion. These onsets are transformed into sample indices by multiplying with the sample rate.

If the audio has a **gradual decrease in frequency** and no second onset can be found, I take the entire length of the beat.

Subsequently, I feed these sample indices alongside the **beat interval** (calculated as 60 divided by Beats Per Minute) into a function. This function establishes a repetitive loop mechanism. The loop iterates ceaselessly, ensuring playback of the extracted beat with intervals matching the defined beat interval. This playback loop persists until the program is terminated, providing continuous rhythmic playback. If we want to increase BPM dynamically can just enter the new BPM, and the code will shift itself to the new beat time interval just when the previous loop (with the previous BPM) is wholly executed, ensuring continuous rhythm in the signal.