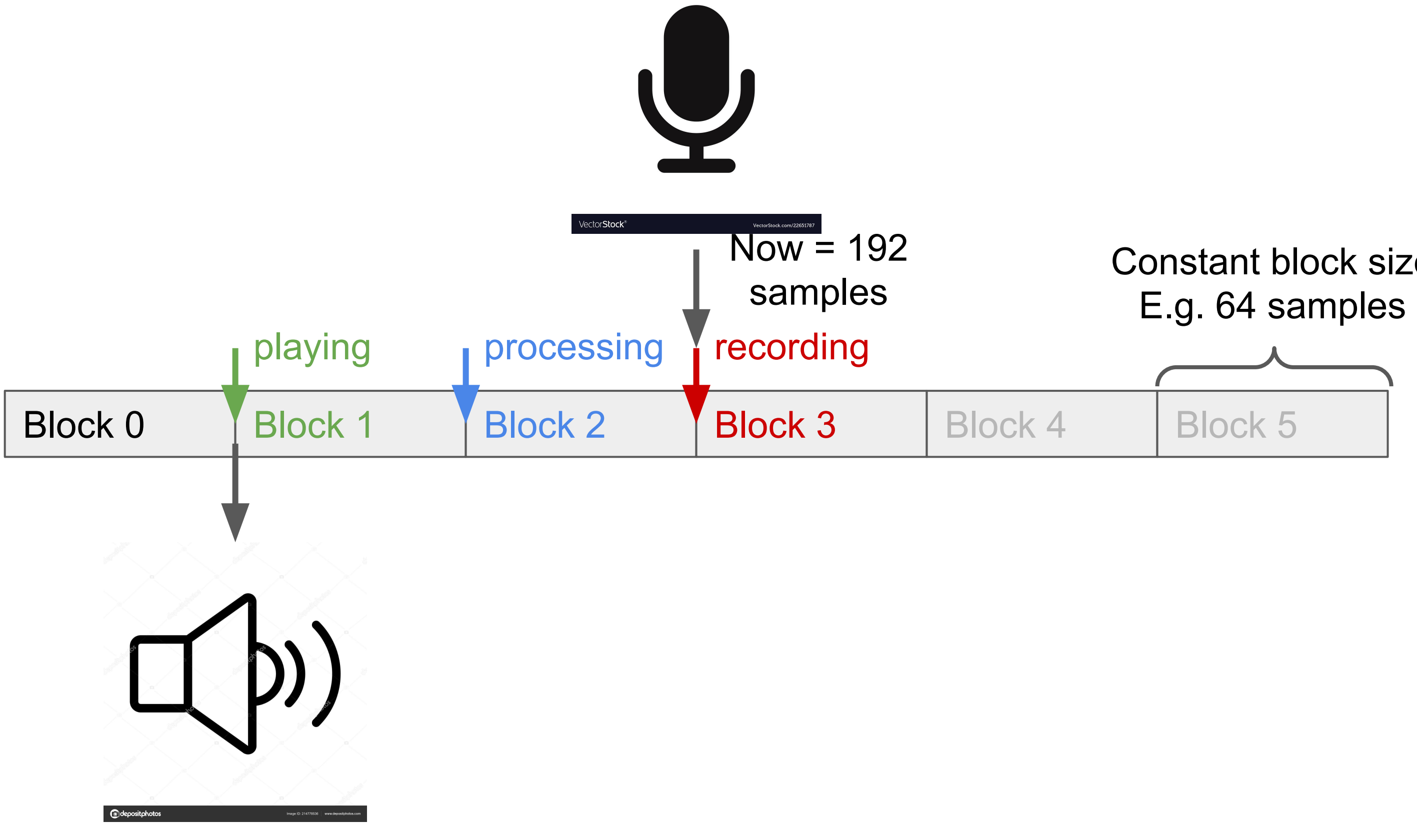
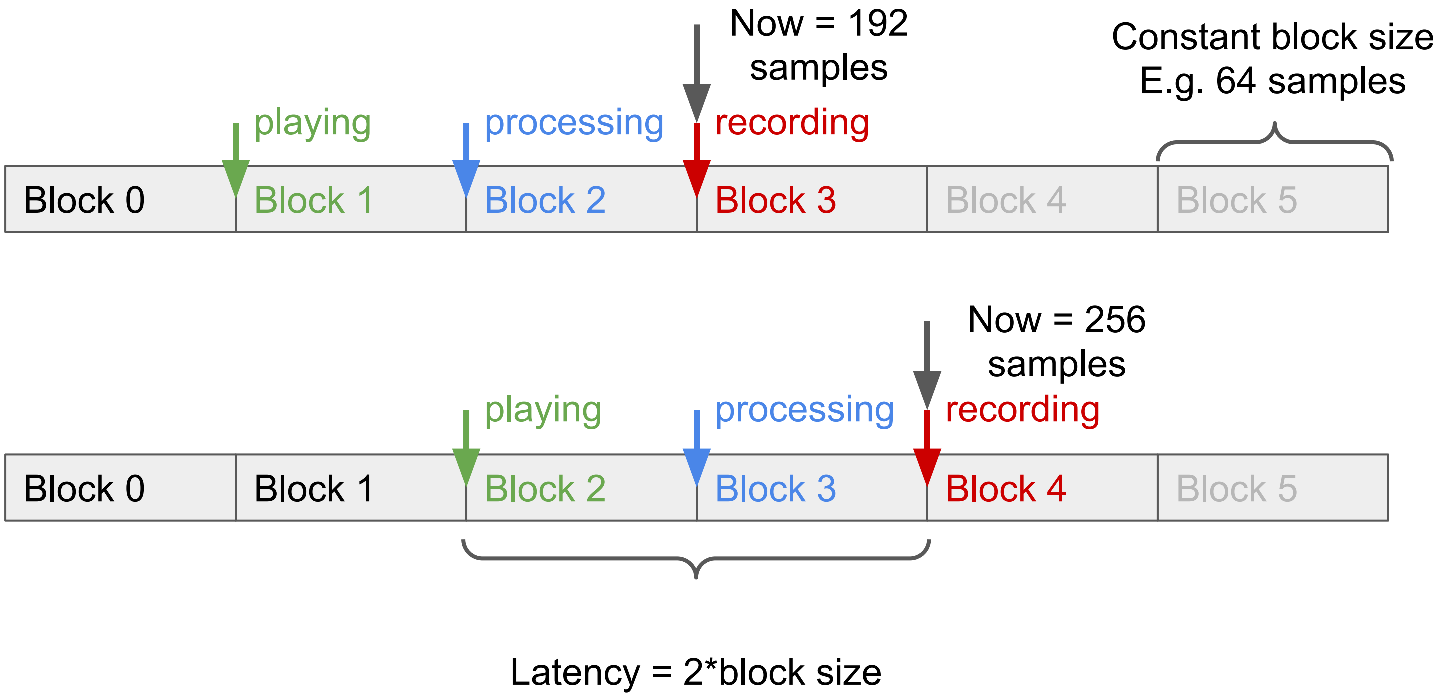
JUCE pt. 1

Real-time, block-based audio processing

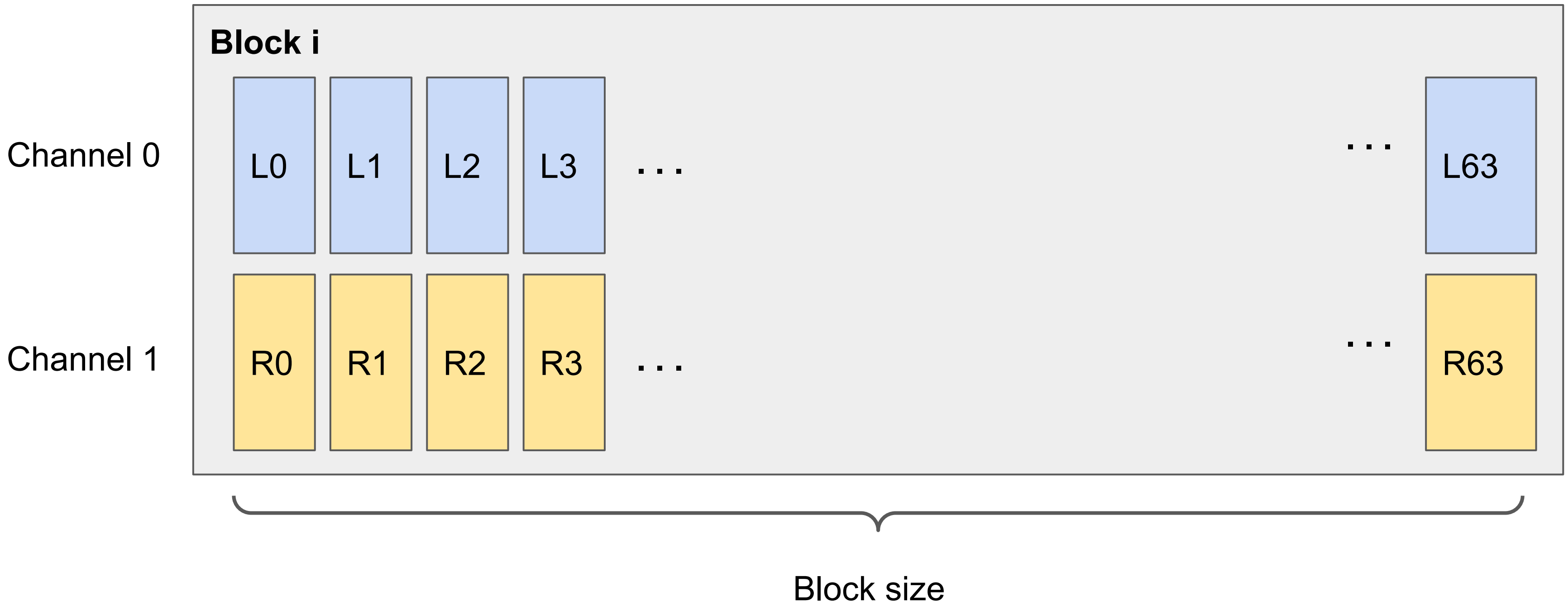
Block-based processing



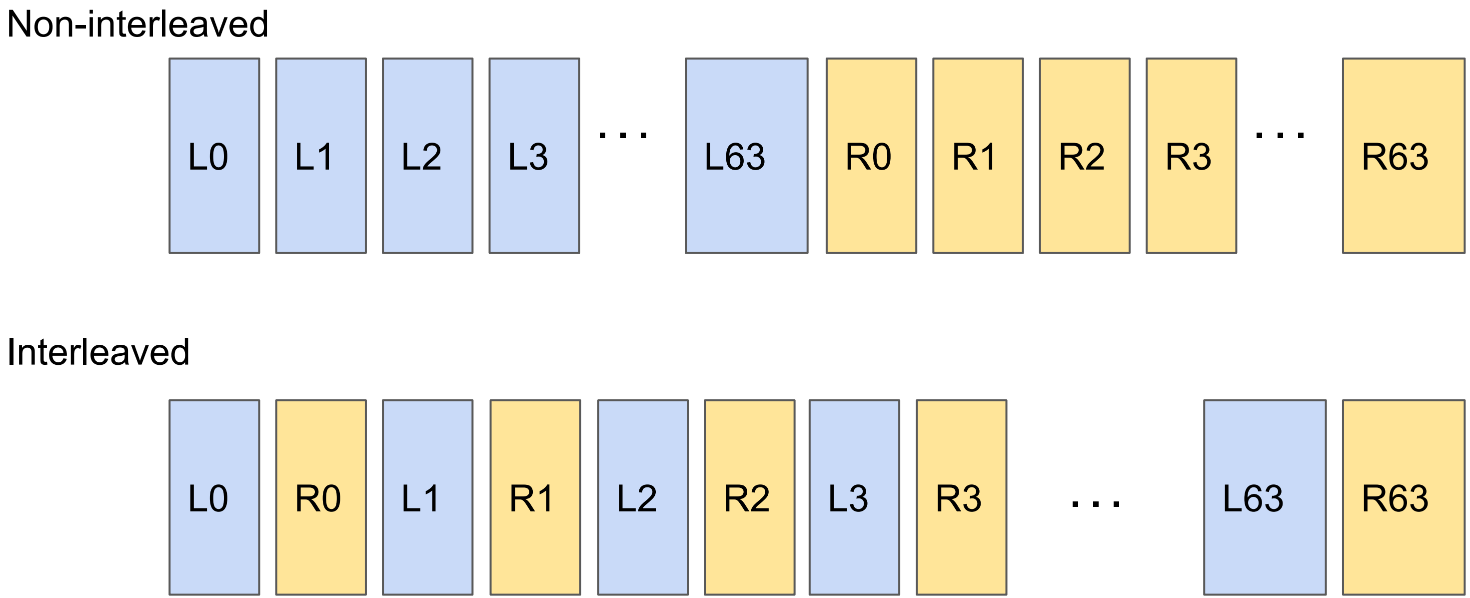
Block-based processing (latency)



Inside a block

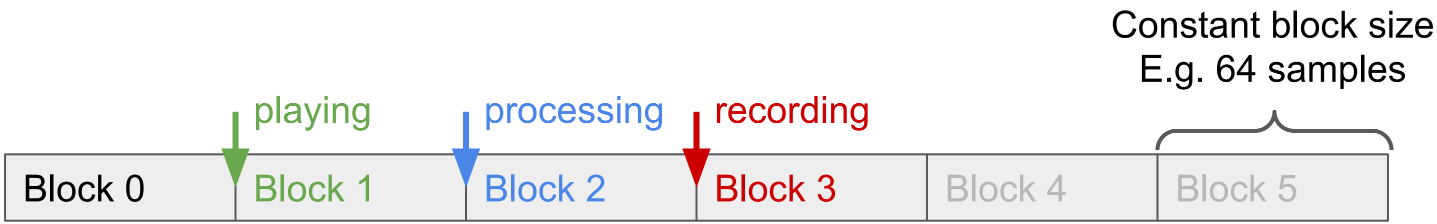


Block in computer memory



In JUCE this information is mostly hidden from us - we can retrieve data for each channel by applying the proper index (0 for left and 1 for right). We will only need this information from loading interleaved audio files and for communicating audio data with other frameworks.

Block-based processing in JUCE



Get what has been recorded (input channel 0) **as**:
auto* **inReadBuffer** = bufferToFill.buffer->**getReadPointer**(0, **bufferToFill**.startSample);

Access what needs to be processed (output channel 0) **as**:
auto* **leftWriteBuffer**= bufferToFill.buffer->**getWritePointer**(0, **bufferToFill**.startSample);

void MainComponent::**getNextAudioBlock** (const juce::AudioSourceChannelInfo& **bufferToFill**)

Common/practical issues

- In Windows 10, you might need to disable audio enhancement in audio device settings.
- To find what we have added in our examples, look for the code included in “__added__” vvv and “__added__ ^^^”