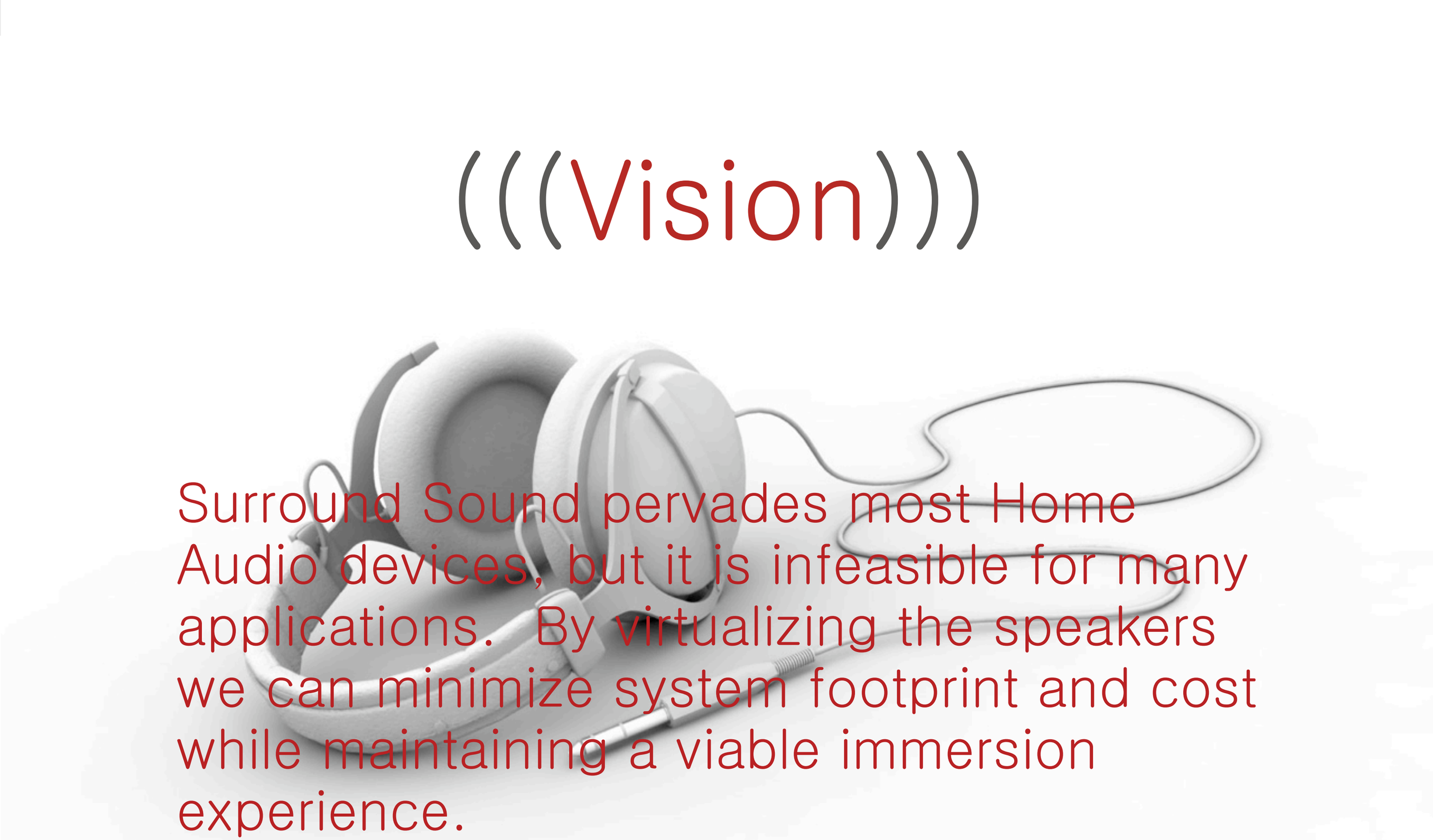
(((Virtual Surround Sound)))

Harrison King Hall

November 14, 2006 MIT Course 6.111 Project Presentation

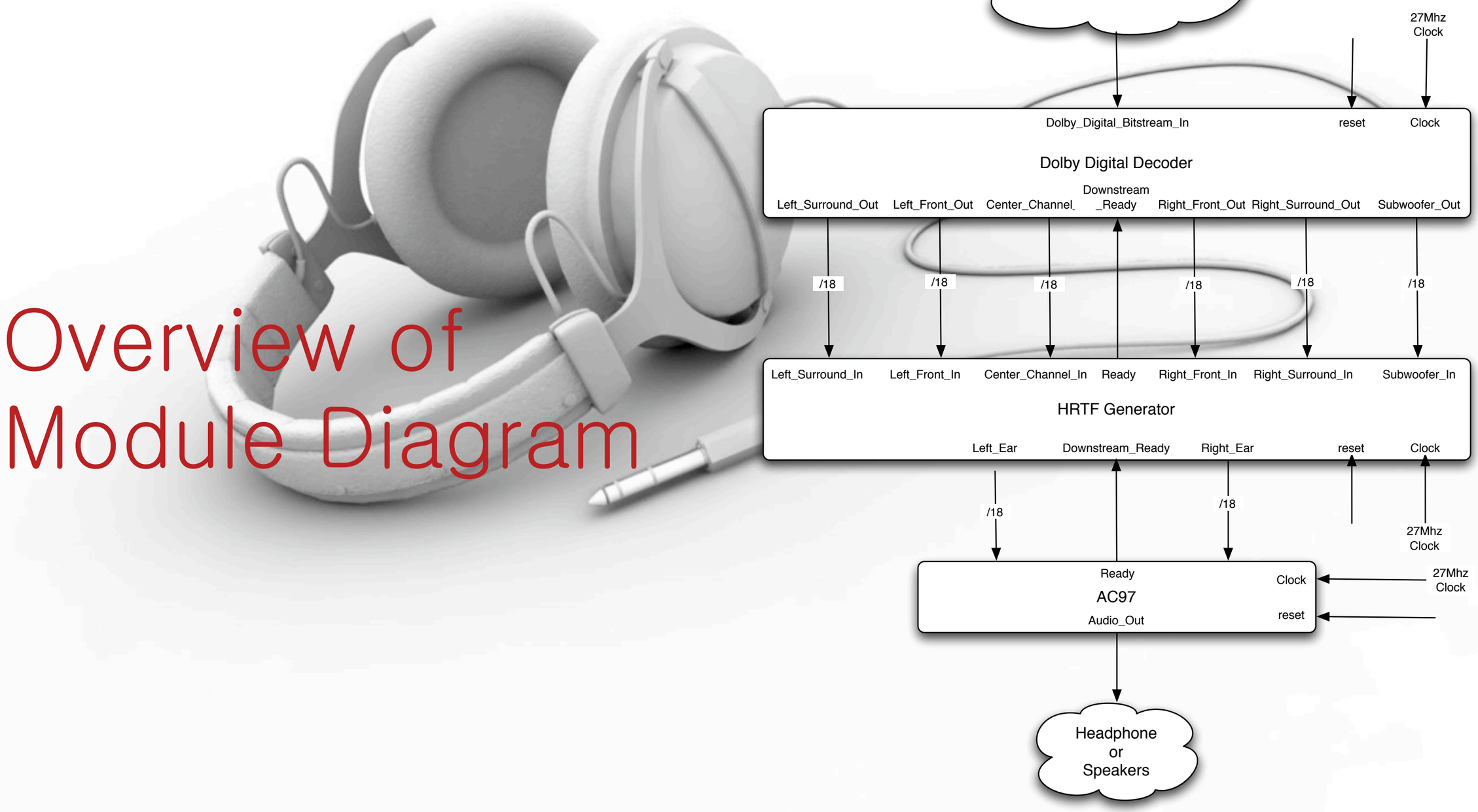
(((Vision)))



Surround Sound pervades most Home Audio devices, but it is infeasible for many applications. By virtualizing the speakers we can minimize system footprint and cost while maintaining a viable immersion experience.

(((Steps)))

Overview of Module Diagram



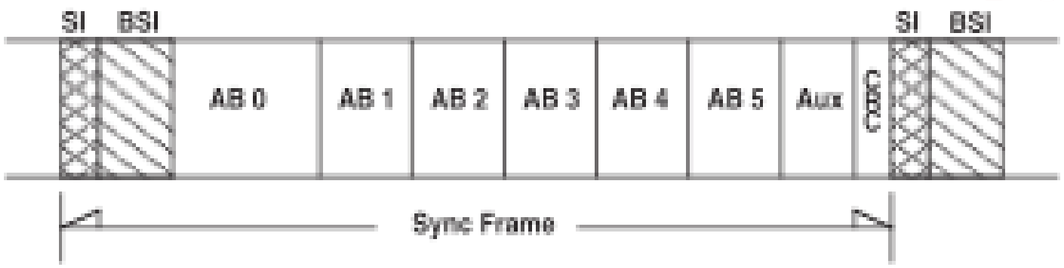
(((Steps))).1

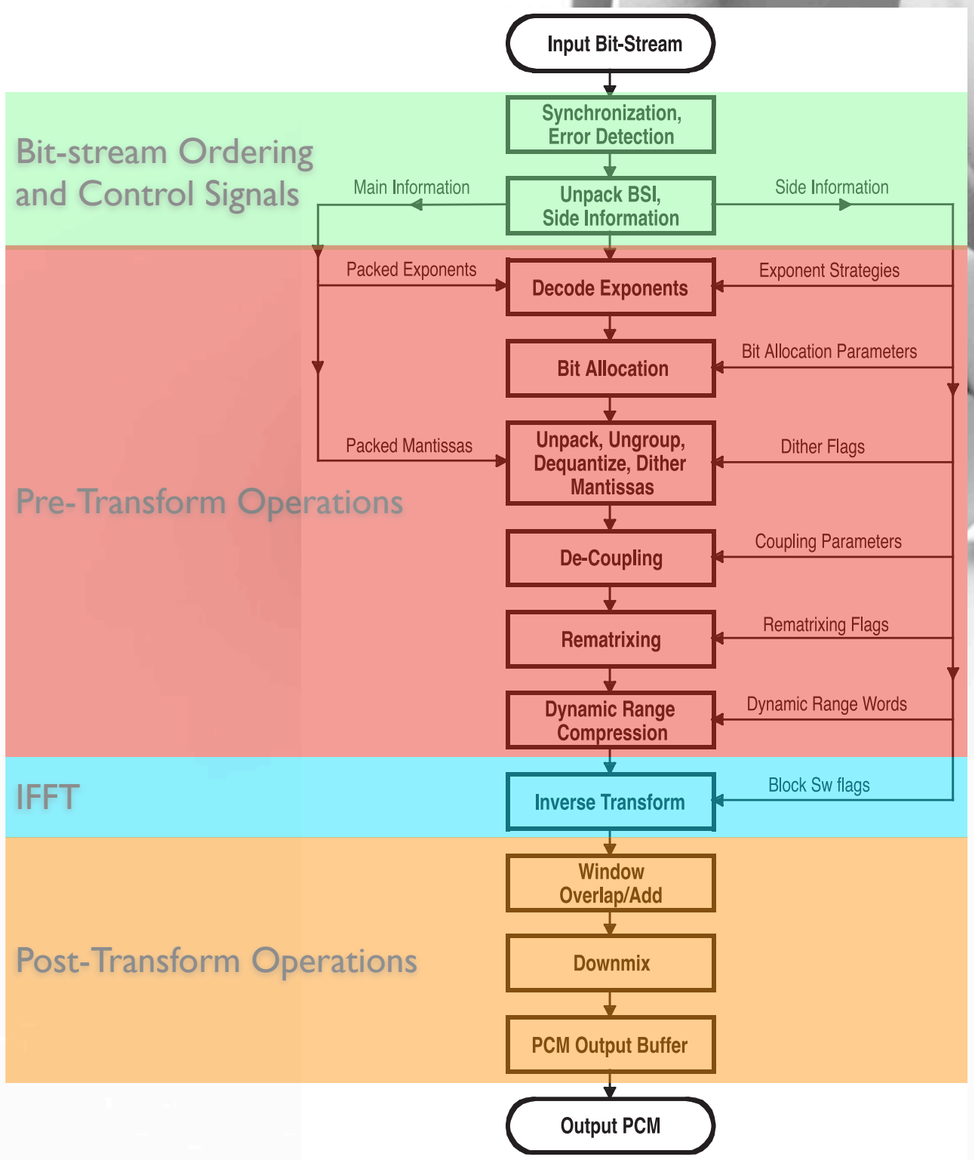
Find a Source

- Take disparate streams of PCM data
- “Freely” available encoded source
- Encode our own(daunting)

(((Steps))).2

Decode the Source

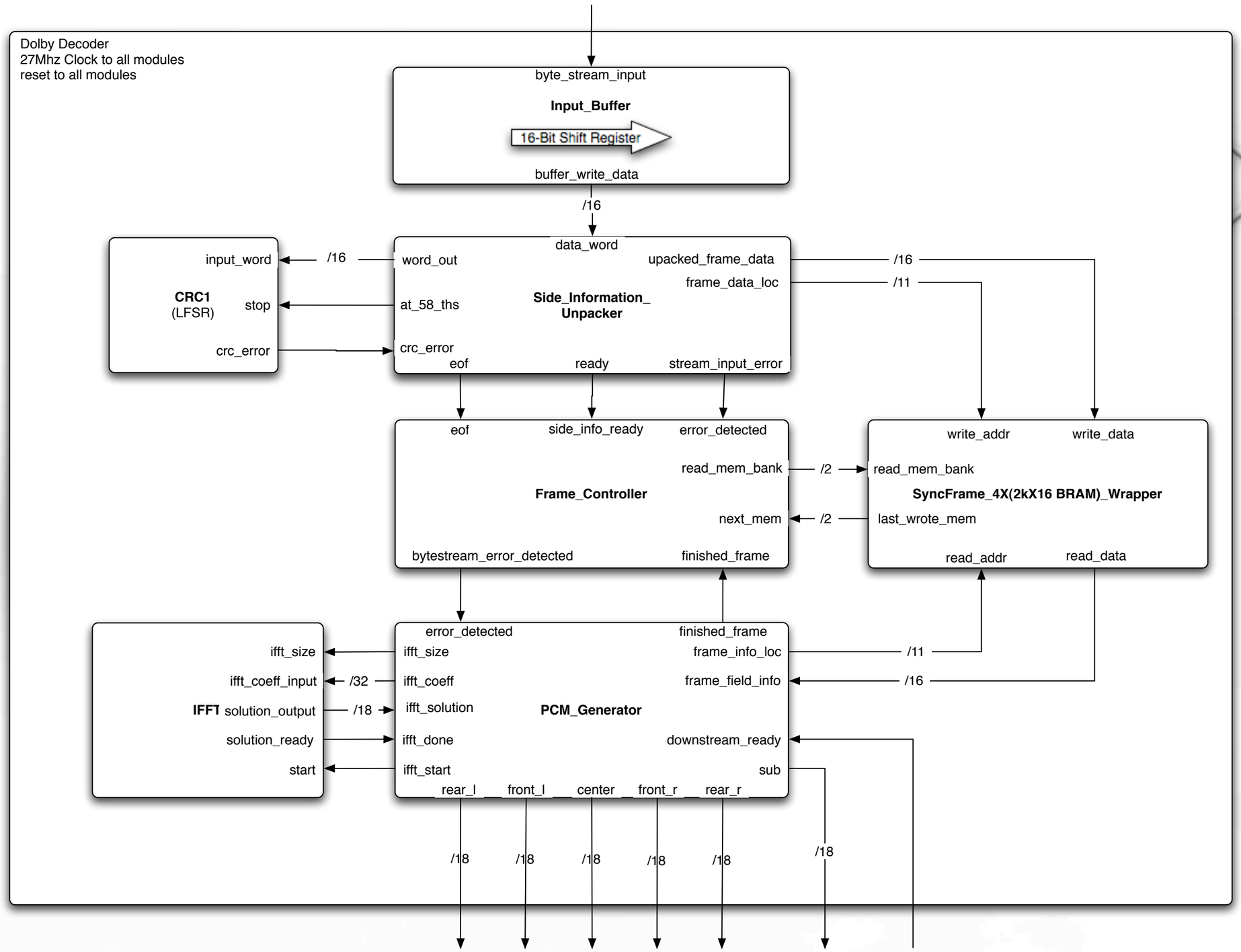




1. Bit-stream Ordering and Control Signals
2. Pre-Transform Operations
3. IFFT Transform
4. Post-Transform Operations

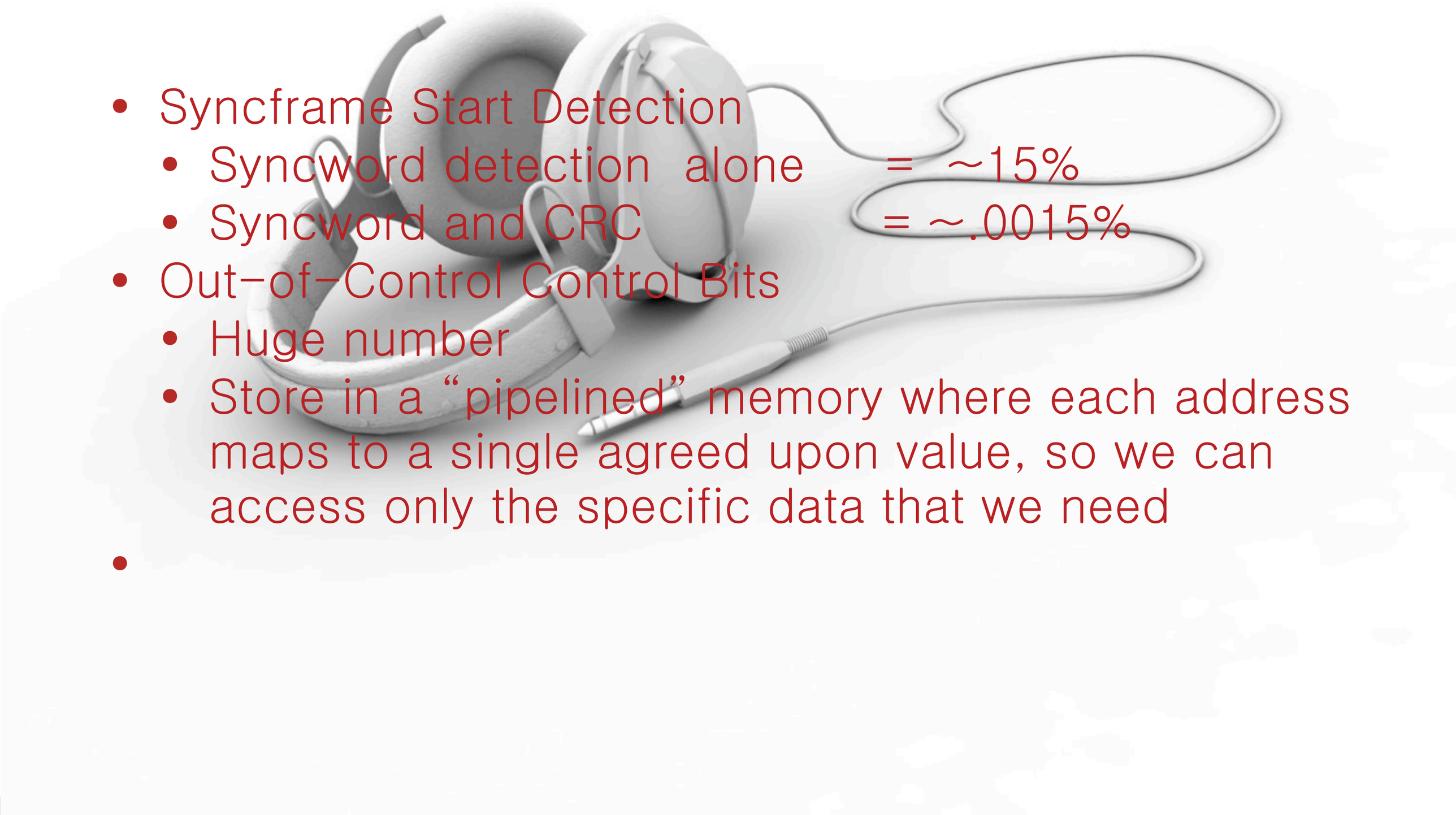
(((Steps))).2

Decode the Source: Module Diagram



(((Steps))).2.1

Bit-stream syncing and Control Signals



- Syncframe Start Detection
  - Syncword detection alone = ~15%
  - Syncword and CRC = ~.0015%
- Out-of-Control Control Bits
  - Huge number
  - Store in a “pipelined” memory where each address maps to a single agreed upon value, so we can access only the specific data that we need
-

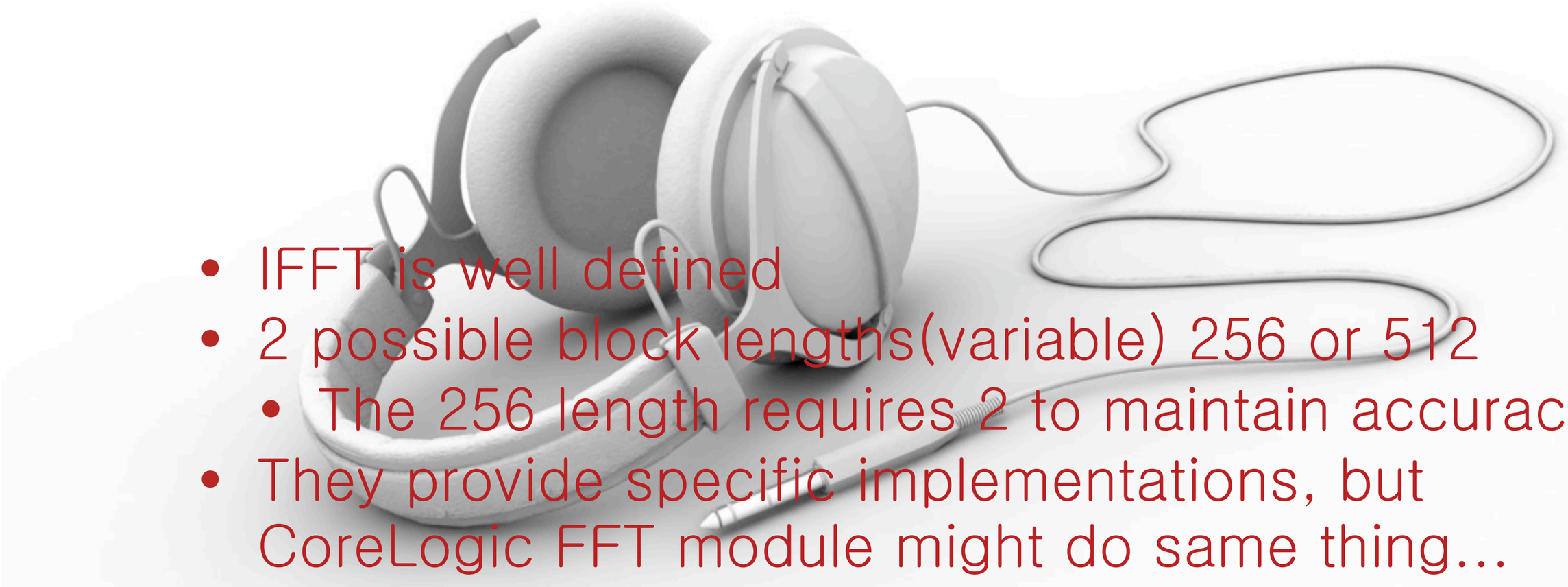
(((Steps))).2.2

Pre-Transform Processing

- What are we really looking at in an AC-3 stream?
  - “The actual audio information conveyed by the AC-3 bit stream consists of the quantized frequency coefficients. The coefficients are delivered in floating point form, with each coefficient consisting of an exponent and a mantissa.”
- Steps:
  - Need to generate the set of exponents for each AudioBlock or for all 6 audio blocks(determined by A0)
  - Take set of exponents and determine number of bits to assign the mantissa for decoding
  - Decouple and re-matrix the input (if necessary)
- Scary huge complications and will likely take a long time to write(i.e. Thanksgiving...maybe)

(((Steps))).2.3

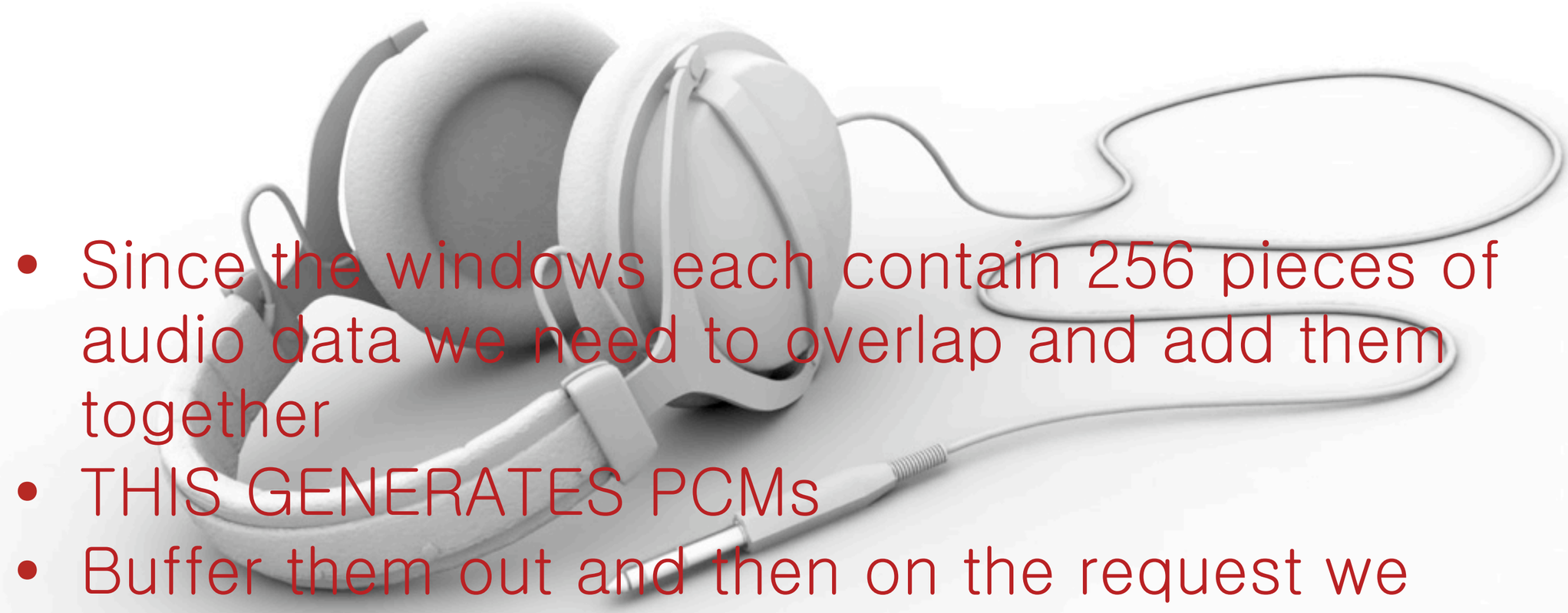
Inverse Fast Fourier Transform



- IFFT is well defined
- 2 possible block lengths(variable) 256 or 512
  - The 256 length requires 2 to maintain accuracy
- They provide specific implementations, but CoreLogic FFT module might do same thing...

(((Steps))).2.4

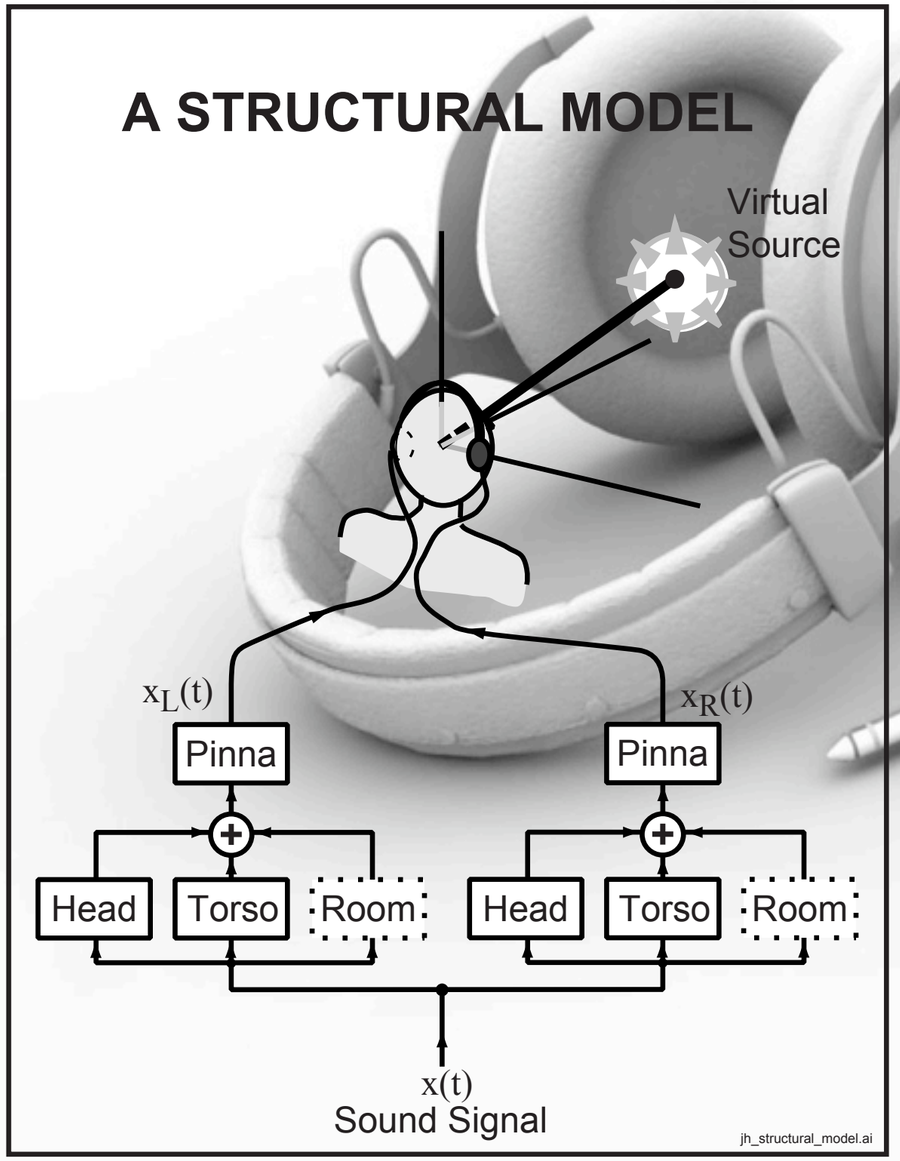
Post-Transform Processing



- Since the windows each contain 256 pieces of audio data we need to overlap and add them together
- THIS GENERATES PCMs
- Buffer them out and then on the request we shuttle them off

(((Steps))).3

Head-Related Transfer Function



- Use phase and frequency shift to make a virtual speaker appear at some location

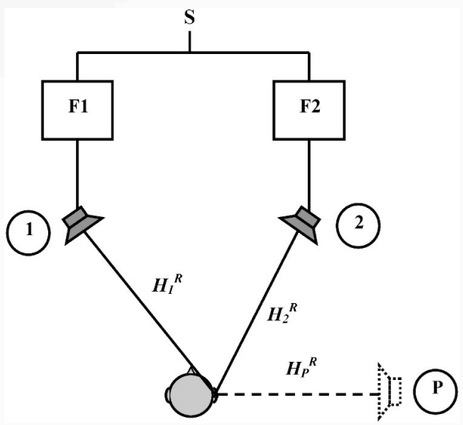
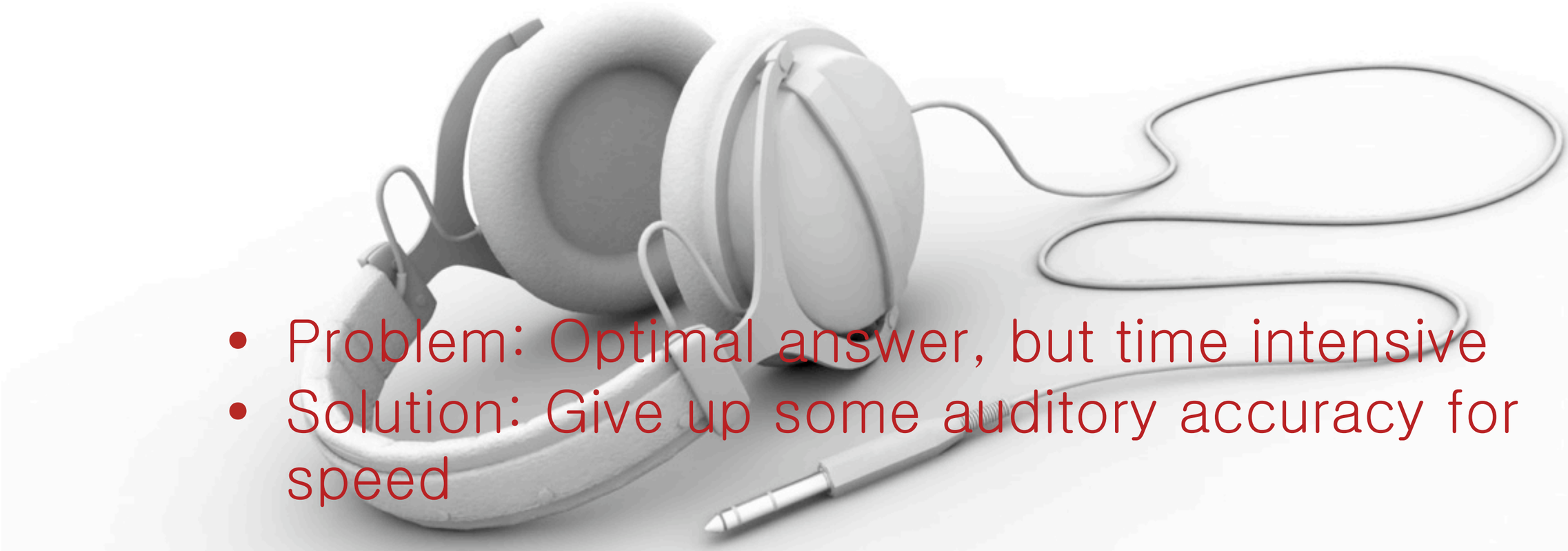


FIG. 2. Illustration of how a virtual stimulus is implemented by using HRTFs from the presentation loudspeaker positions and from the desired virtual position (P). Representations are given for only one of the two ears. See Eqs. (1) and (2).

(((Steps))).3.1

The Truth about Head-Related Transfer Functions

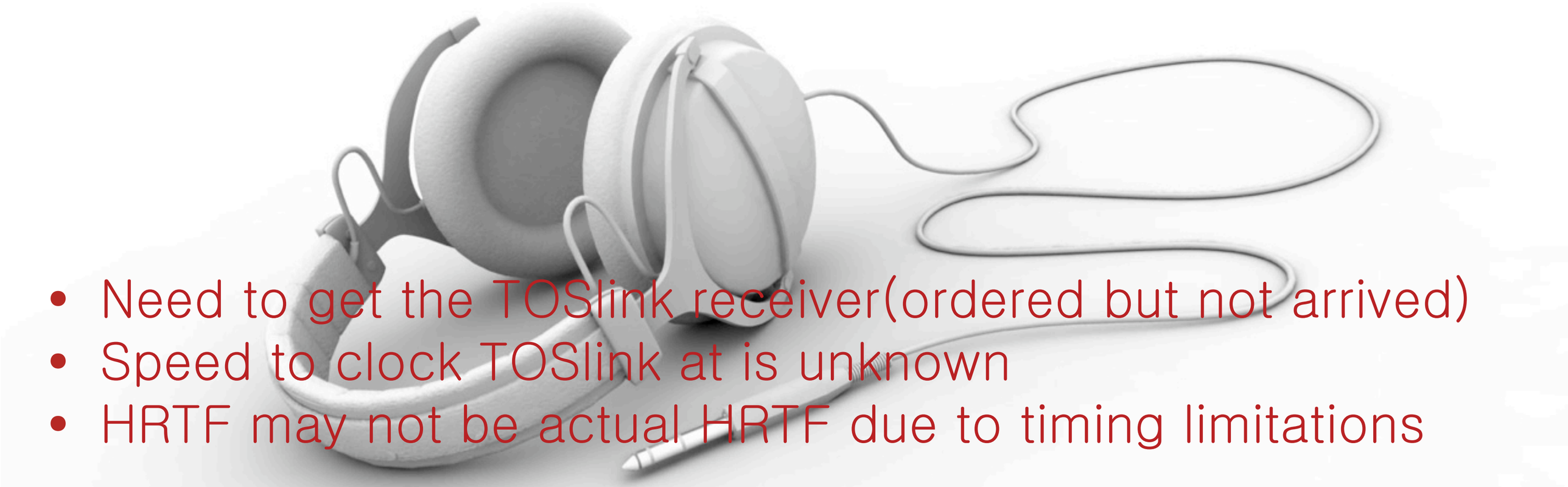


- Problem: Optimal answer, but time intensive
- Solution: Give up some auditory accuracy for speed

(((Contributions)))

Now that everything pertinent to my work progressing is spec'd I can get to the meat of the code.

(((Problems)))



- Need to get the TOSlink receiver(ordered but not arrived)
- Speed to clock TOSlink at is unknown
- HRTF may not be actual HRTF due to timing limitations