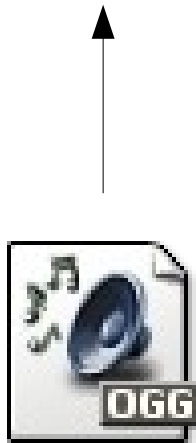
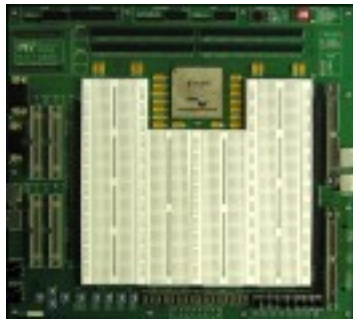


Ogg Vorbis Audio Decoder

Jon Stritar
Matt Papi

November 14, 2005



Ogg Vorbis

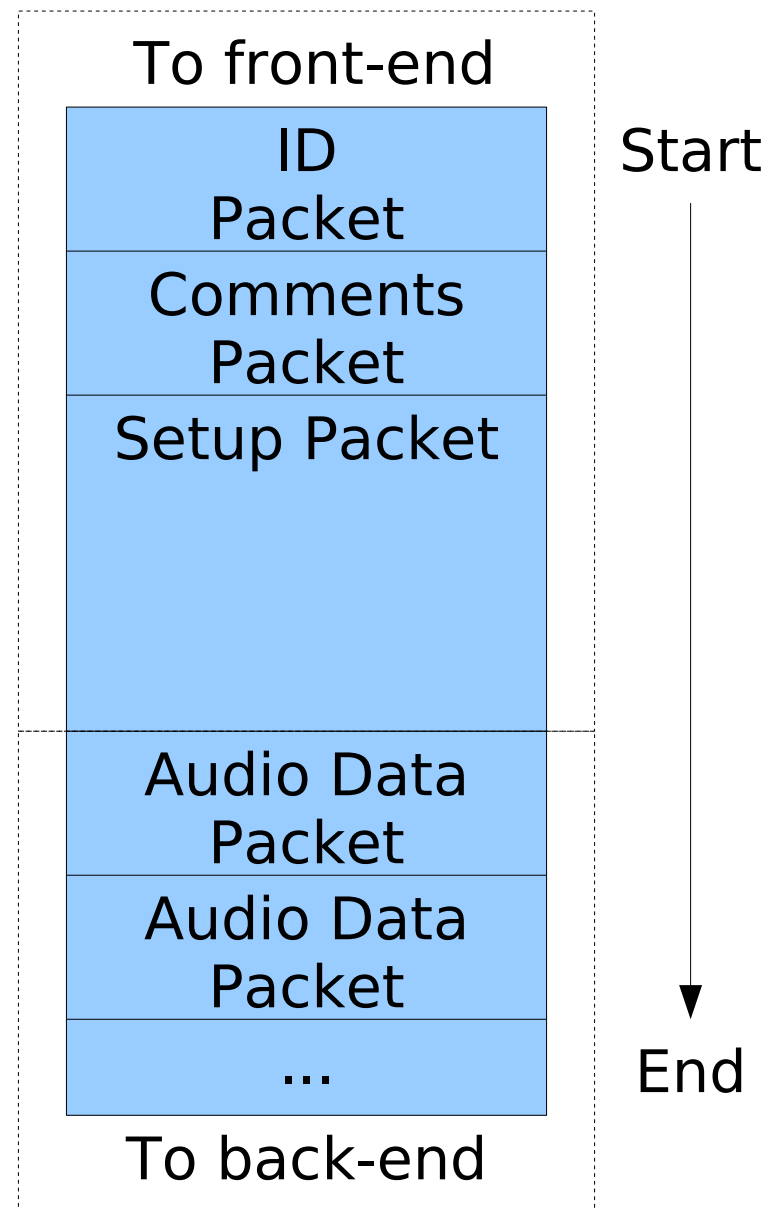
- Free and open audio codec, created in response to MP3 patent issues.
- Important: very few mainstream hardware players support Vorbis.
- Interesting: lots of things a “pure” hardware player can do that a microprocessor-based one can't.
 - higher decimal precision, parallel processing
 - reduced size, power consumption



Best. Codec. Ever.

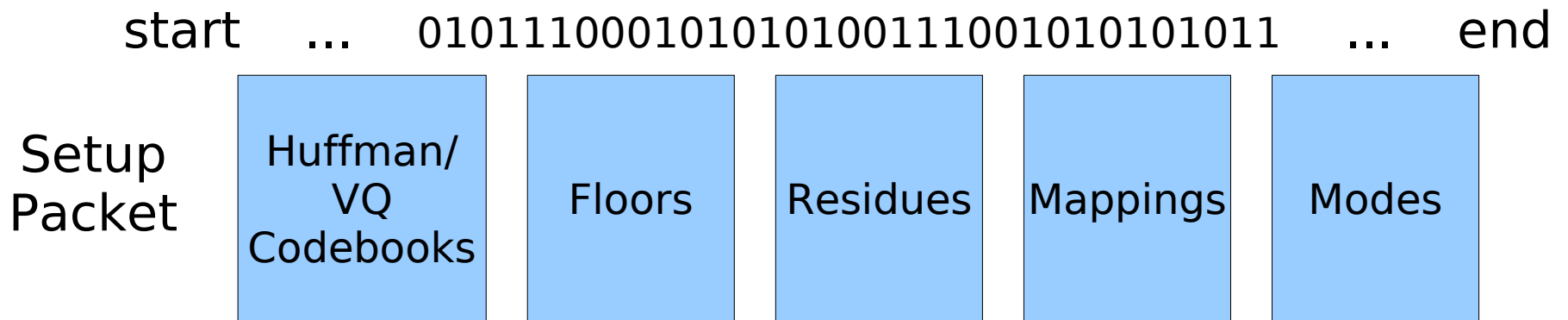
The Vorbis Stream

- Consists of 4 types of packets.
- Front-end:
 - Parses first 3 packets.
 - Configures decoder.
- Back-end:
 - Parses audio packets.
 - Converts data to PCM.

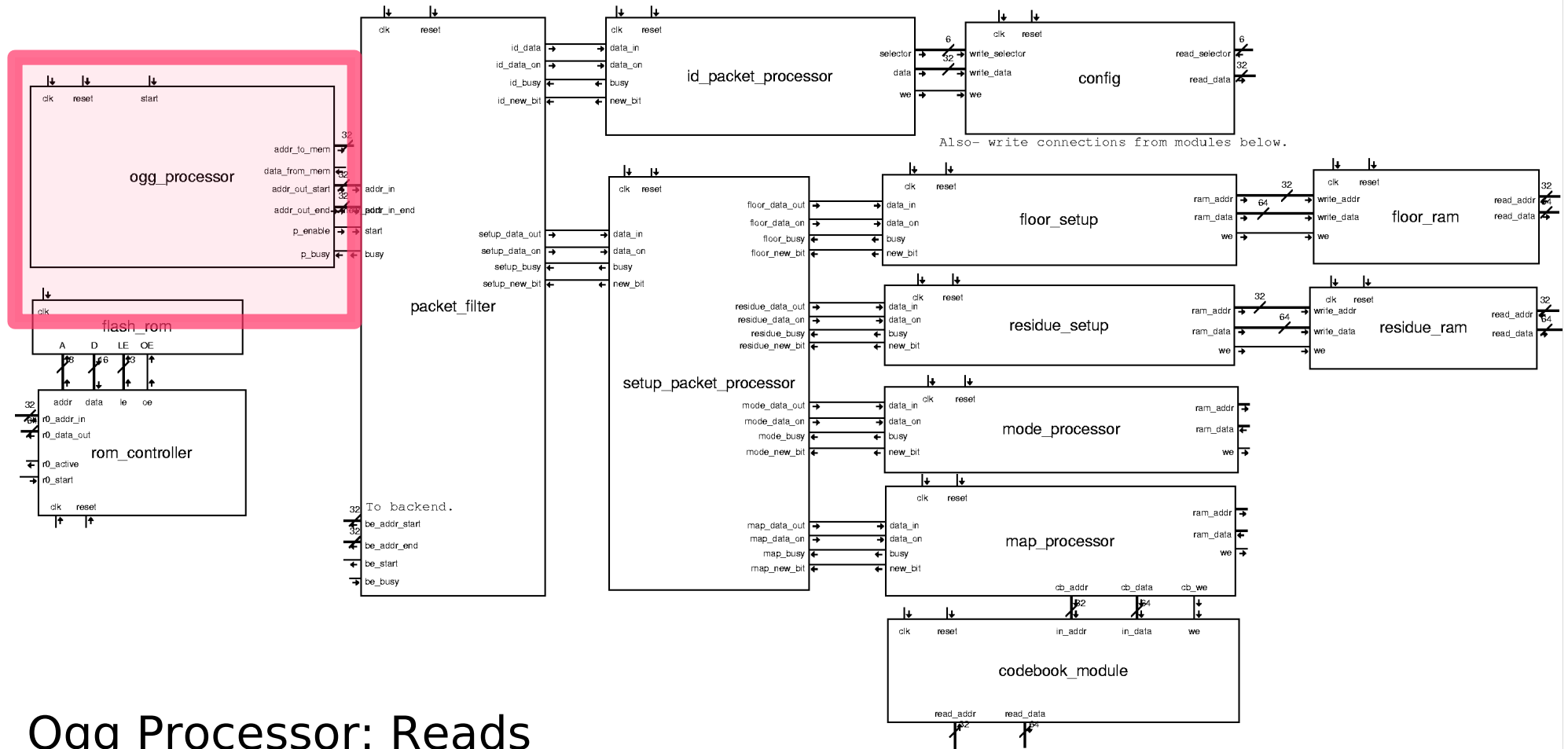


The Front-End

- Configures the decoder.
- Processes the setup and ID packets.
- Extracts Huffman codings, Vector Quantizations.
- Prepares the floor and residue spectral curves for decoding by the back end.

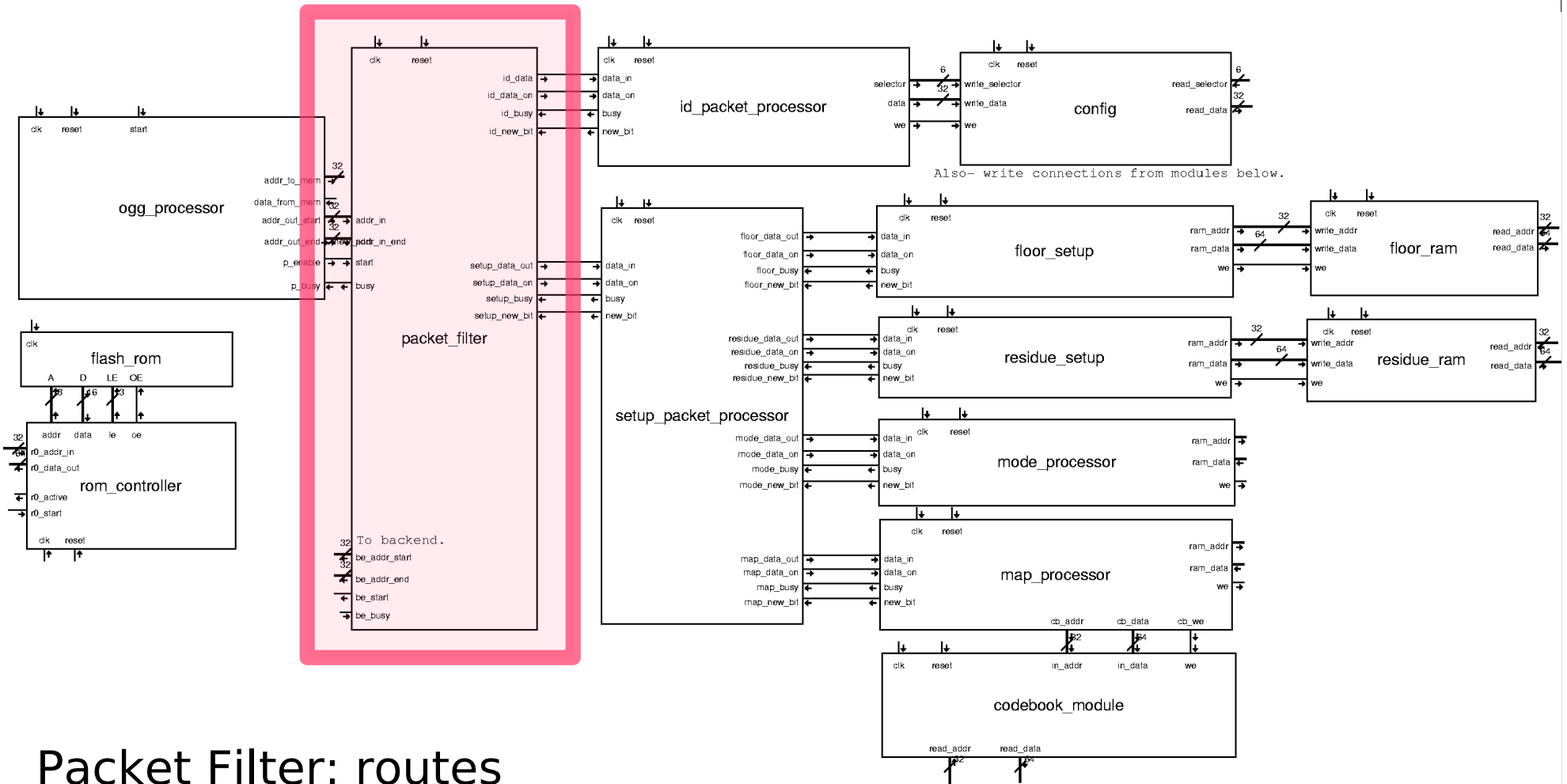


The Front-End



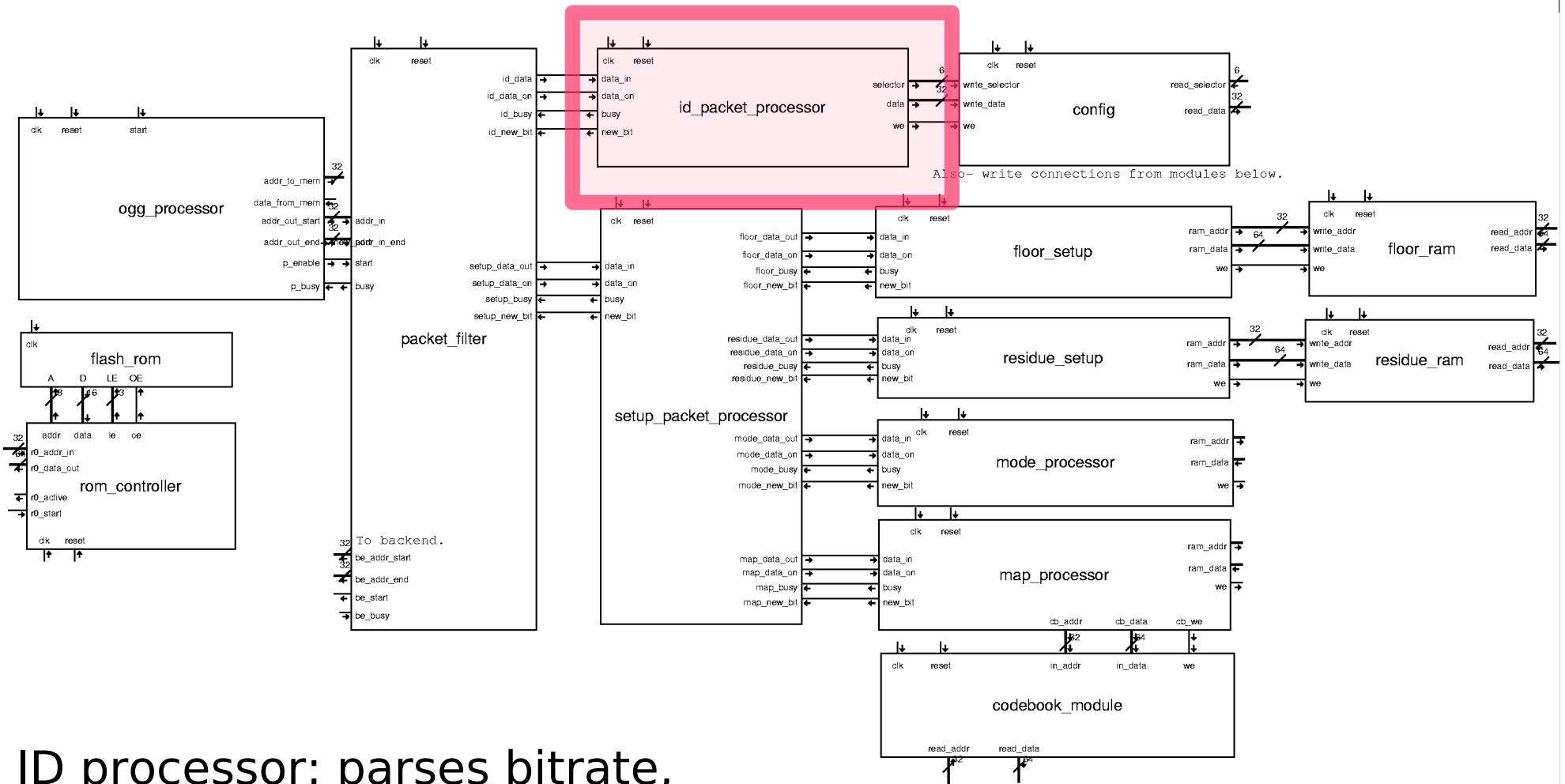
Ogg Processor: Reads file from ROM, extracts Vorbis.

The Front-End



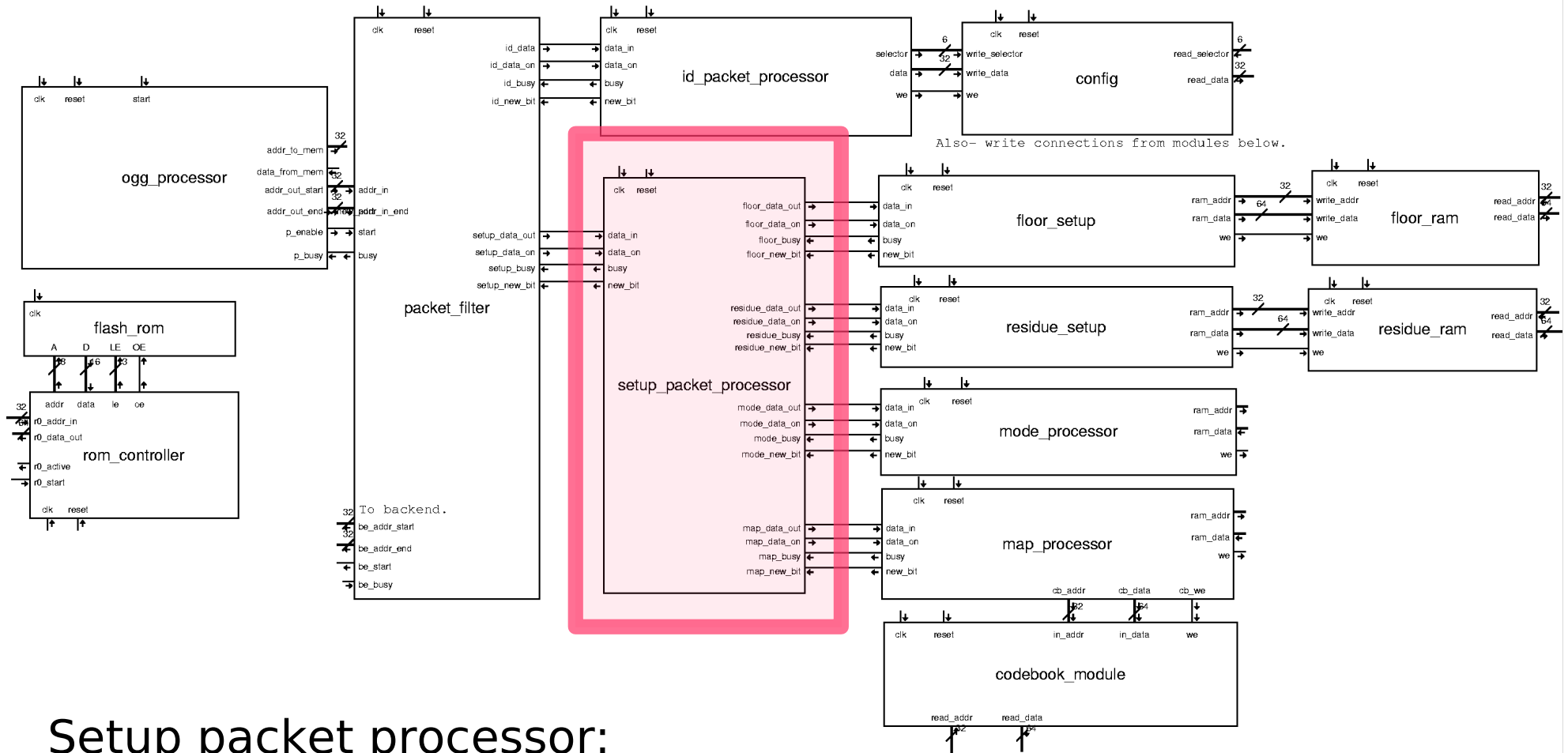
Packet Filter: routes packet to destination FSMs.

The Front-End



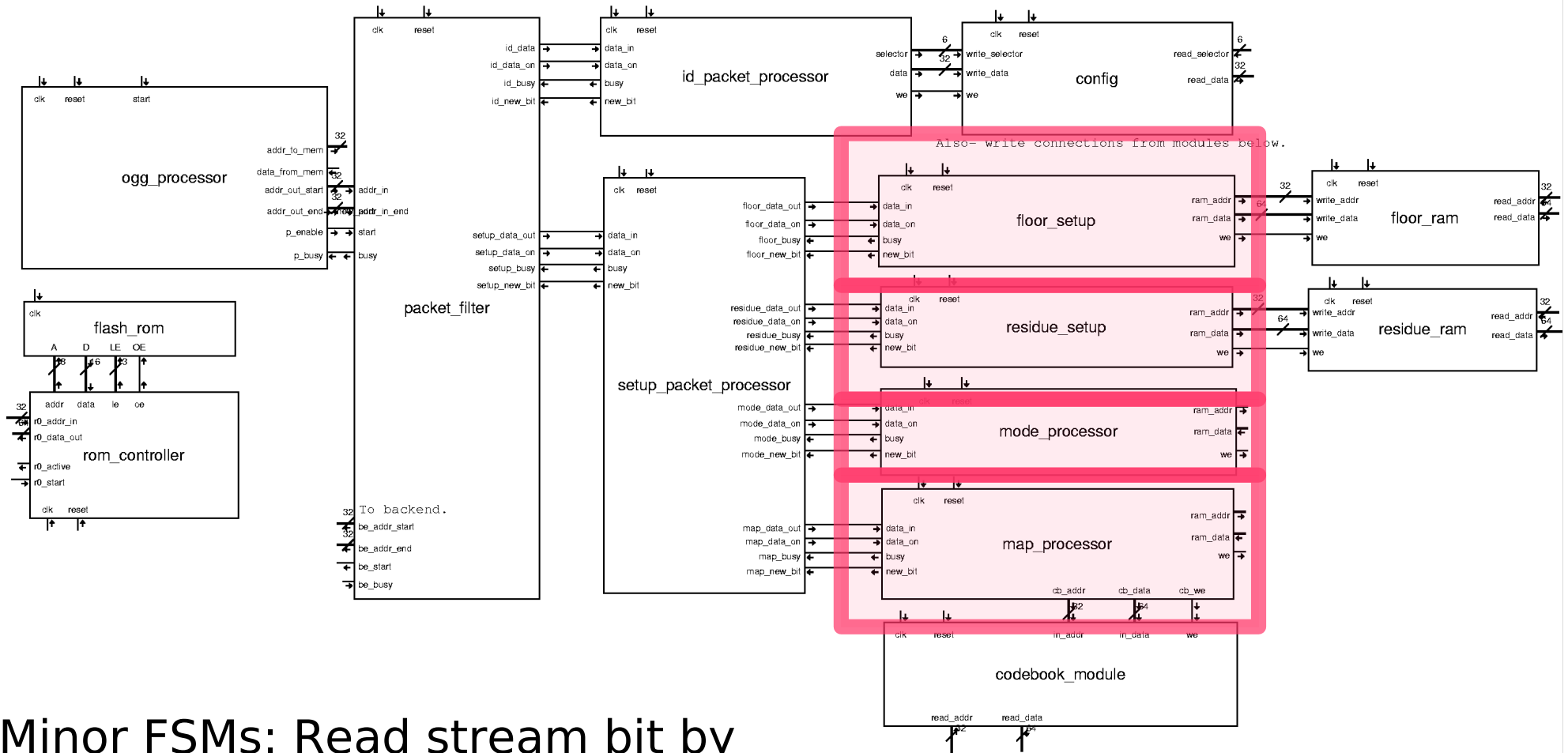
ID processor: parses bitrate, Vorbis version, etc. from ID packet and saves values to config module

The Front-End



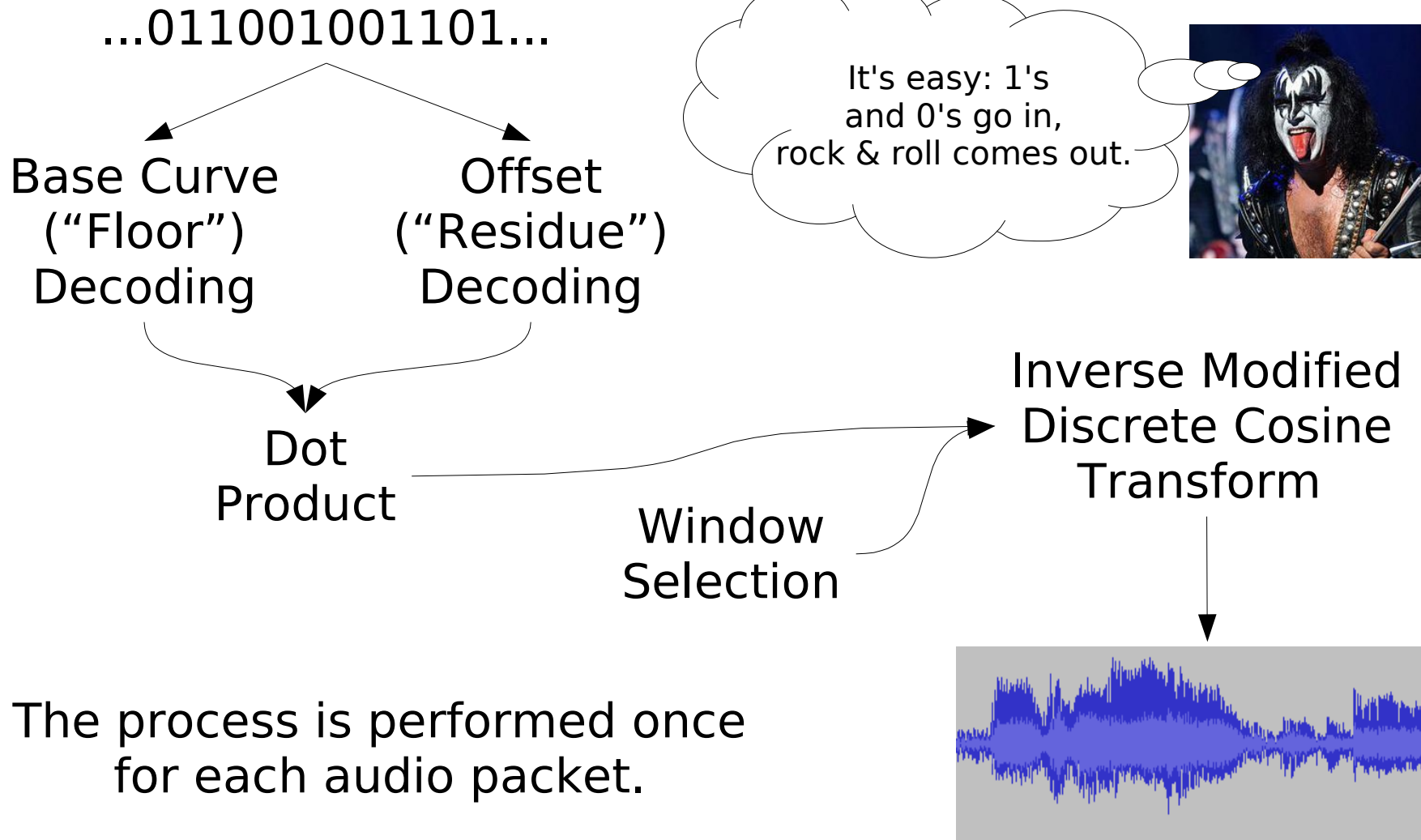
Setup packet processor:
Diverts bit stream to specific
processors.

The Front-End

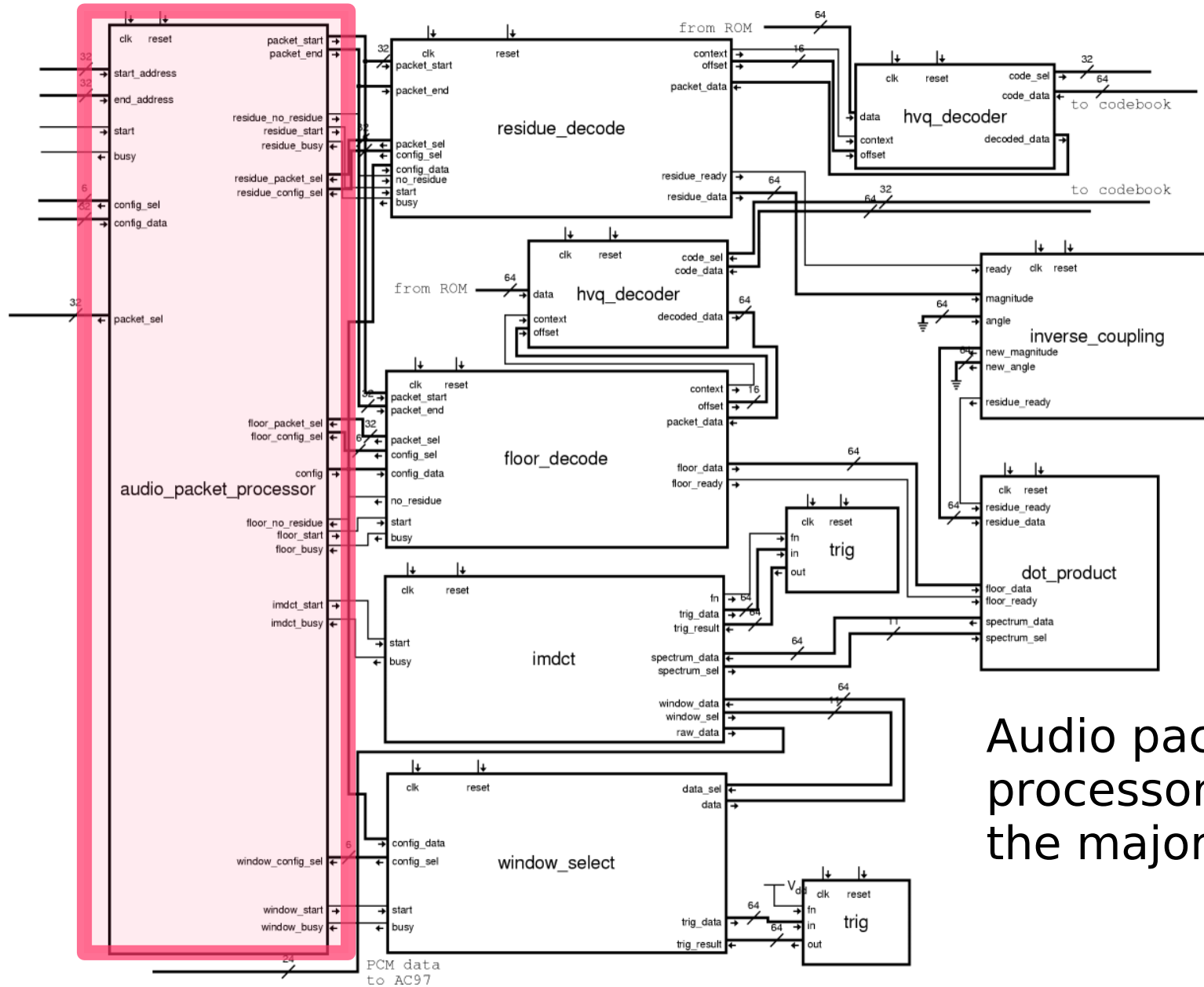


Minor FSMs: Read stream bit by bit, saving information to corresponding block RAMs for back end to use.

The Back-End

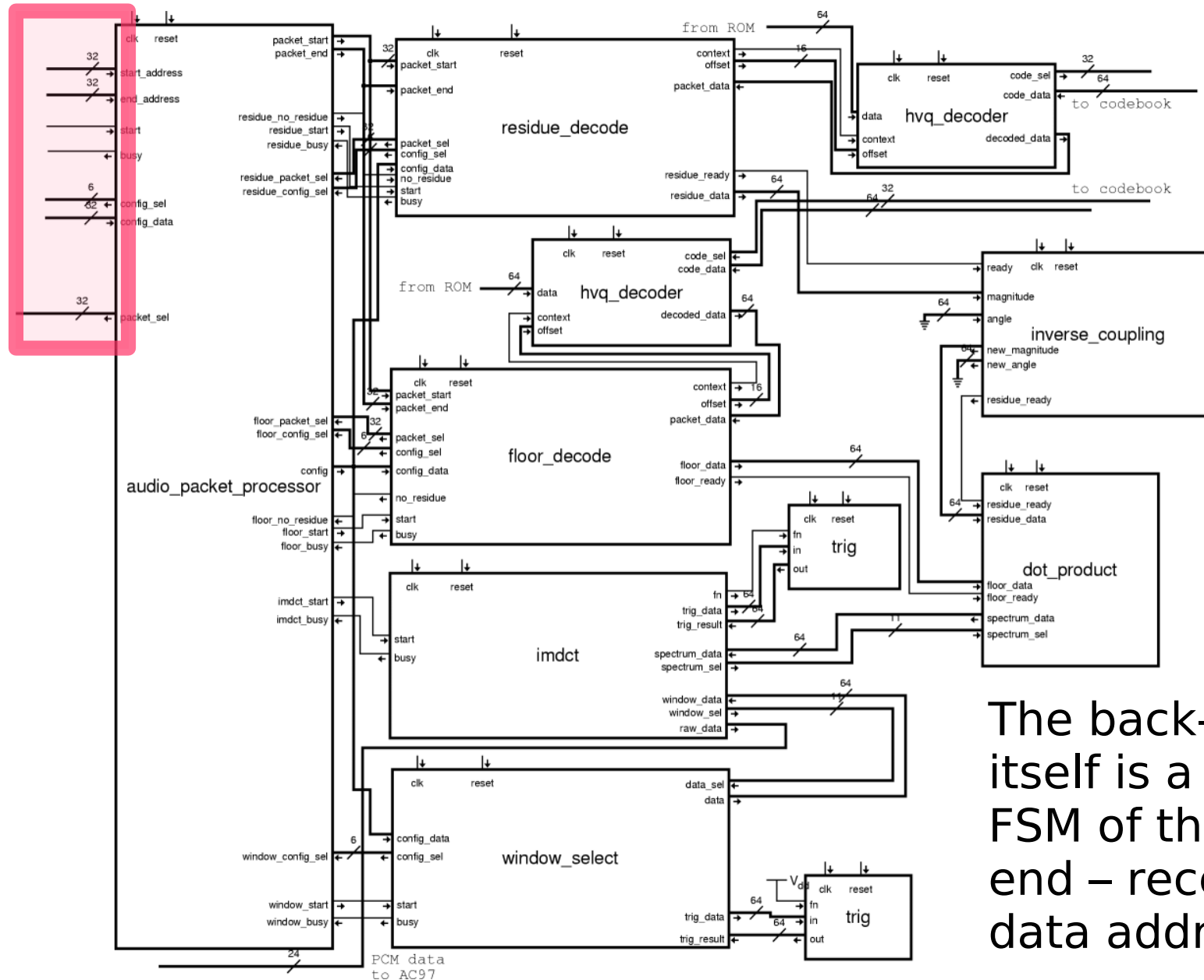


The Back-End



Audio packet processor: the major FSM.

The Back-End



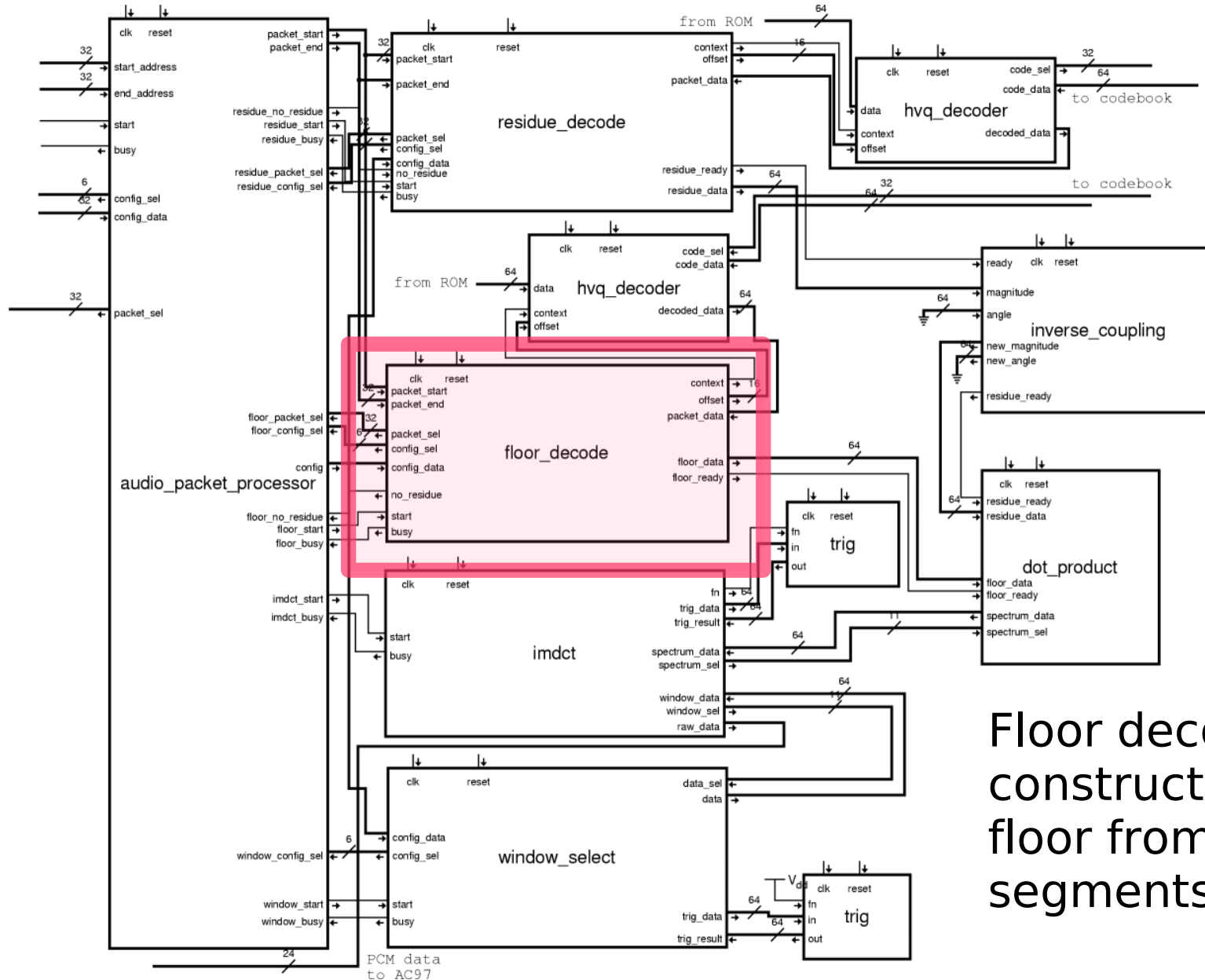
The back-end itself is a minor FSM of the front-end – receives data addresses.

The Back-End



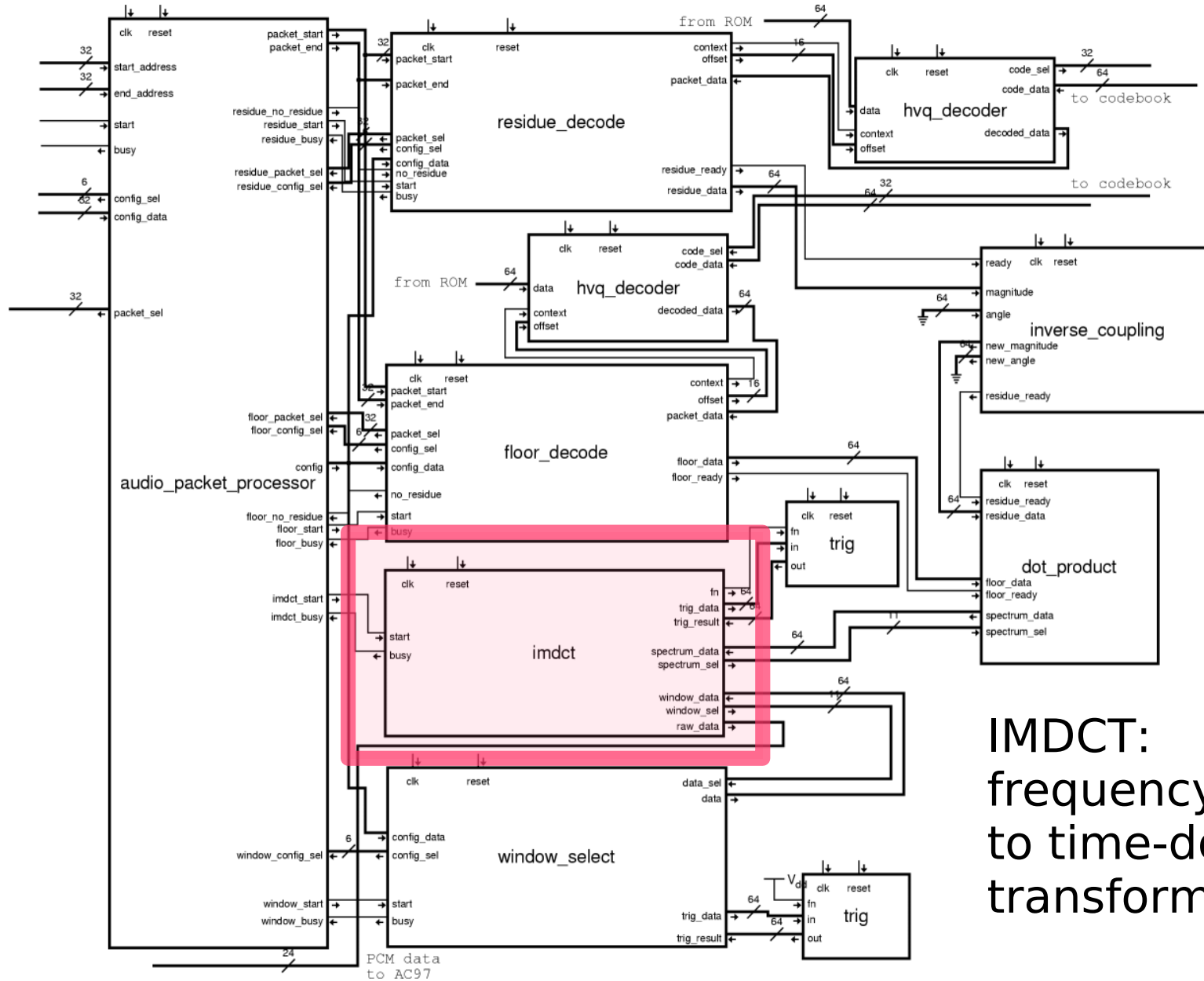
Residue decoder:
deinterlaces
stored offsets.

The Back-End



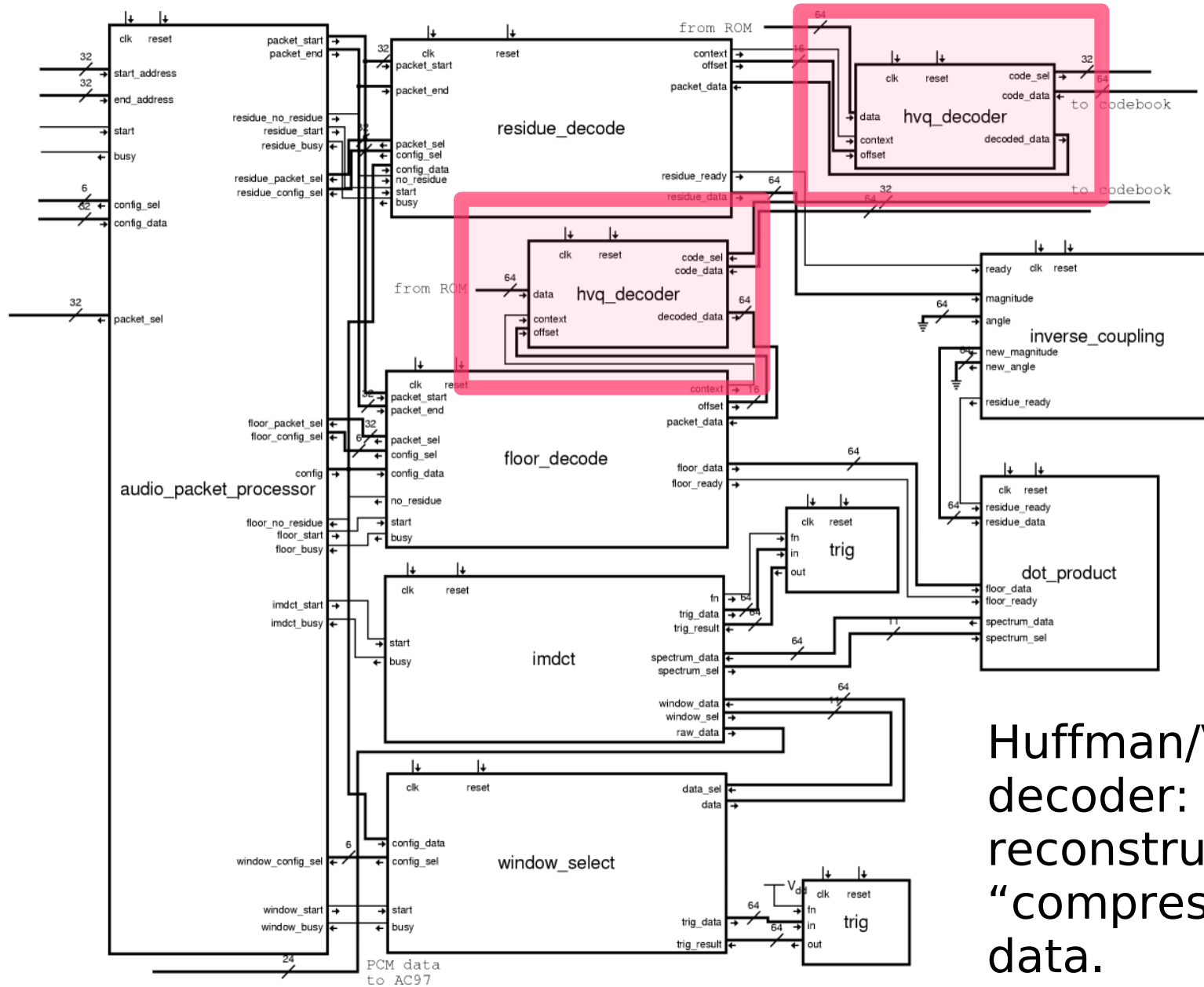
Floor decoder:
constructs
floor from line
segments.

The Back-End



IMDCT:
frequency-domain
to time-domain
transform.

The Back-End



Huffman/VQ decoder:
reconstructs
“compressed”
data.

Questions?