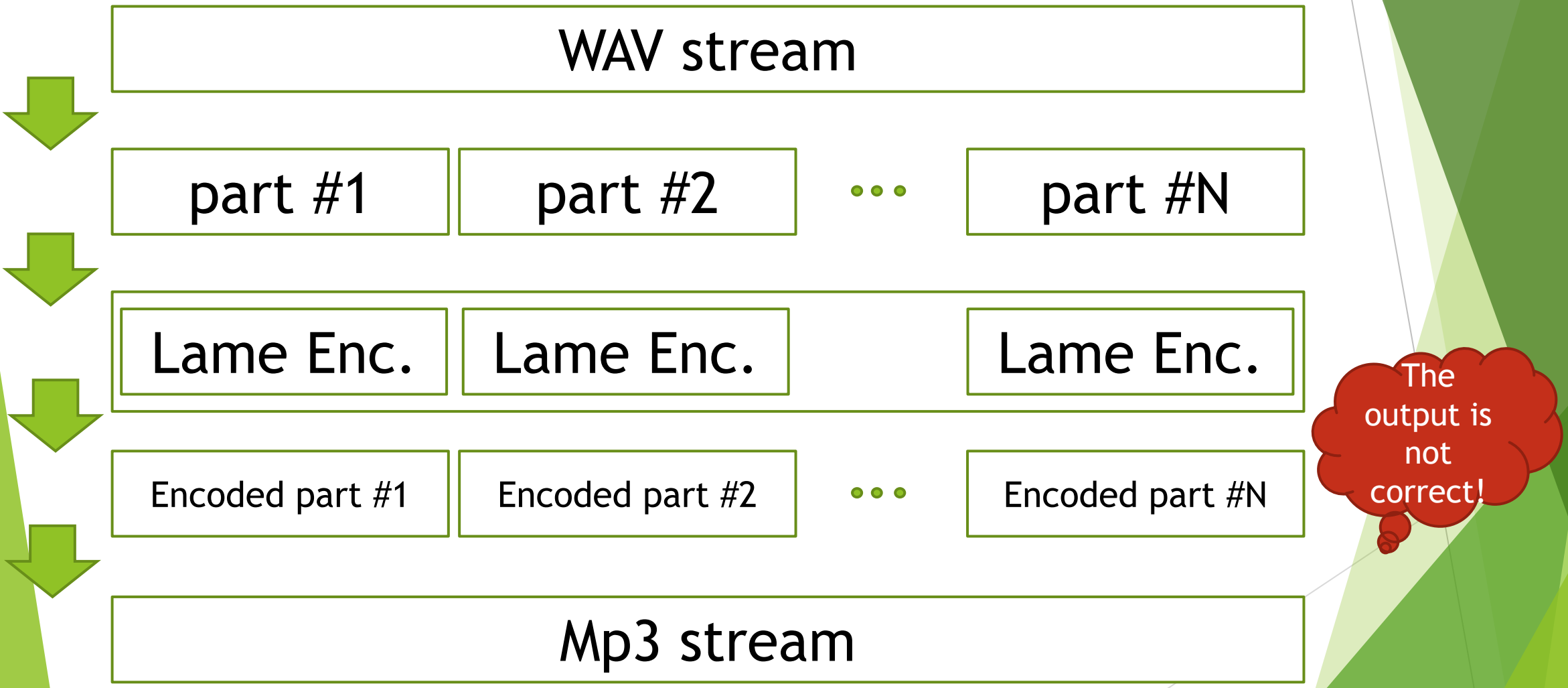
# Multi-threaded mp3 encoder based on LAME

Mostafa Jabaroutimoghaddam

# Main Challenge

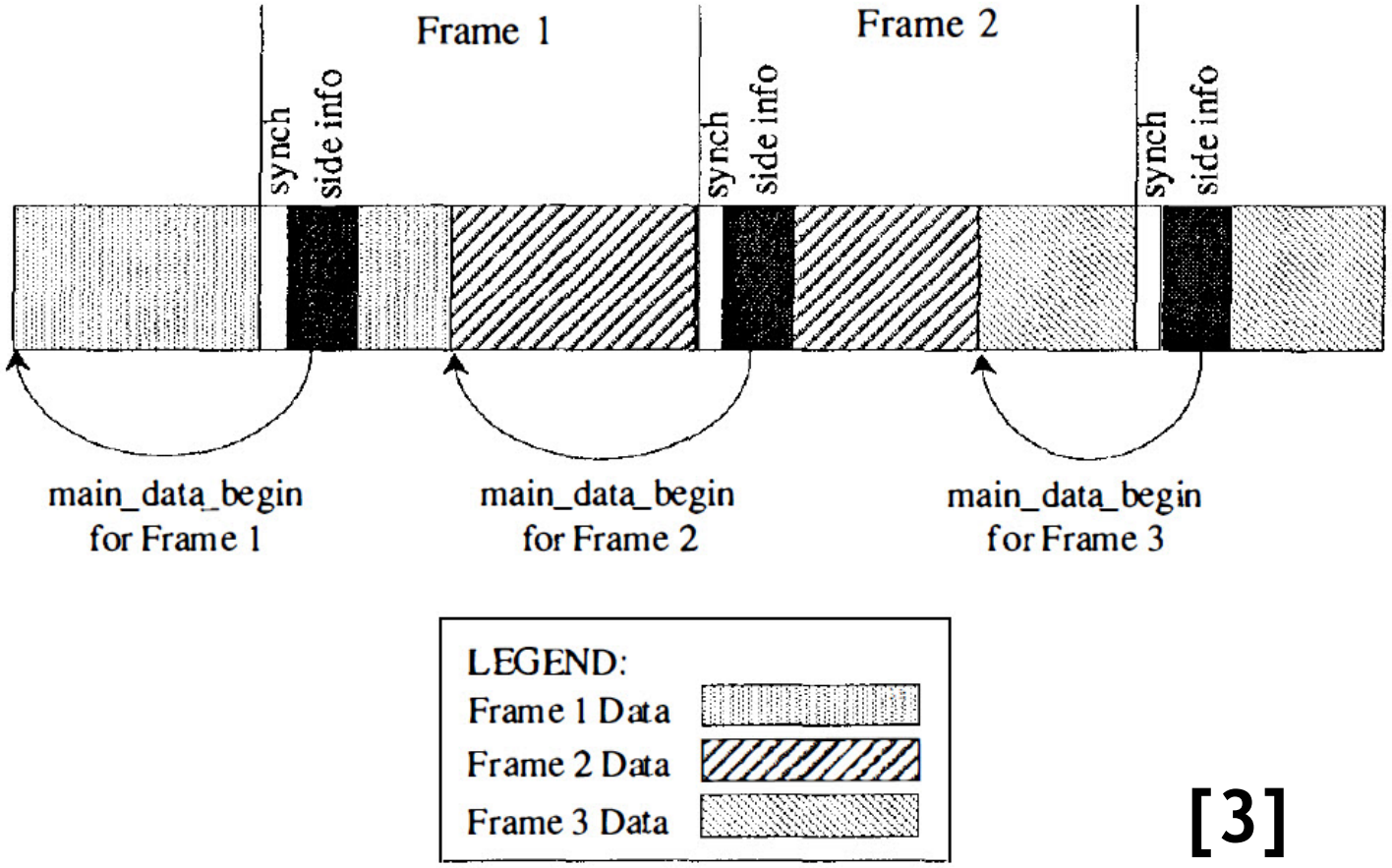


# Main Challenge

- The mp3 output file in this case has a gaps between parts.
- For finding the reason, following strategy was selected:

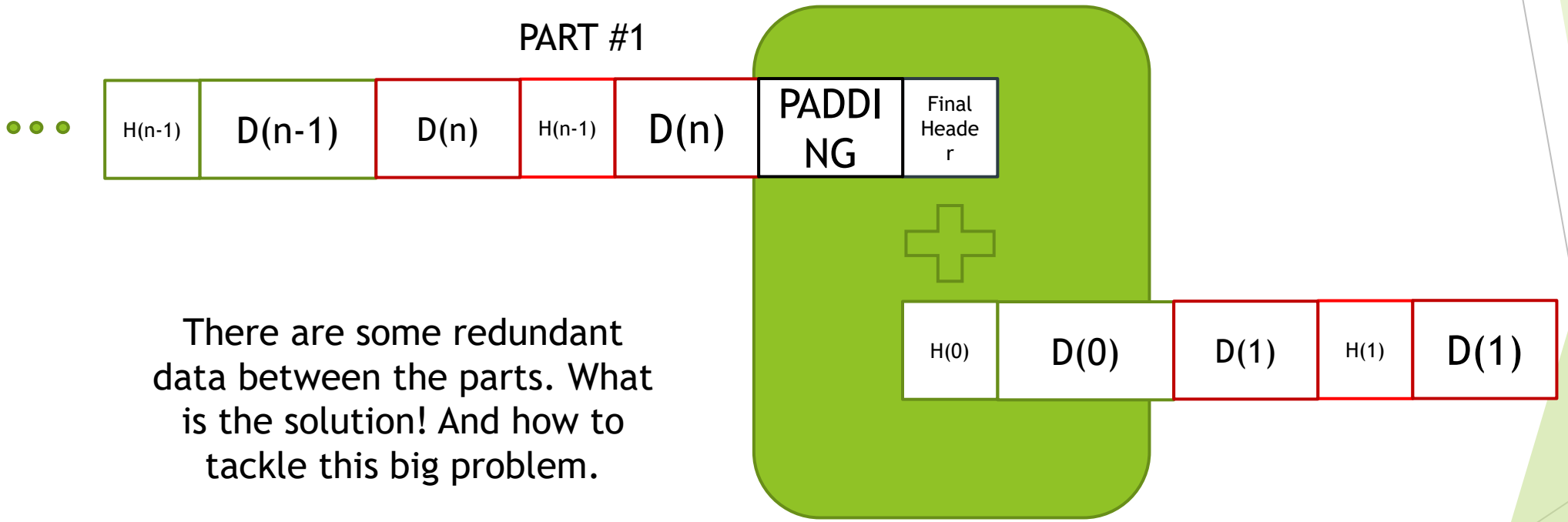
1. Understand MP3 encoder algorithm in detail.
2. Find the problem.
3. Propose some methods to solve.
4. Implementation and test.

# What is the MP3 format file



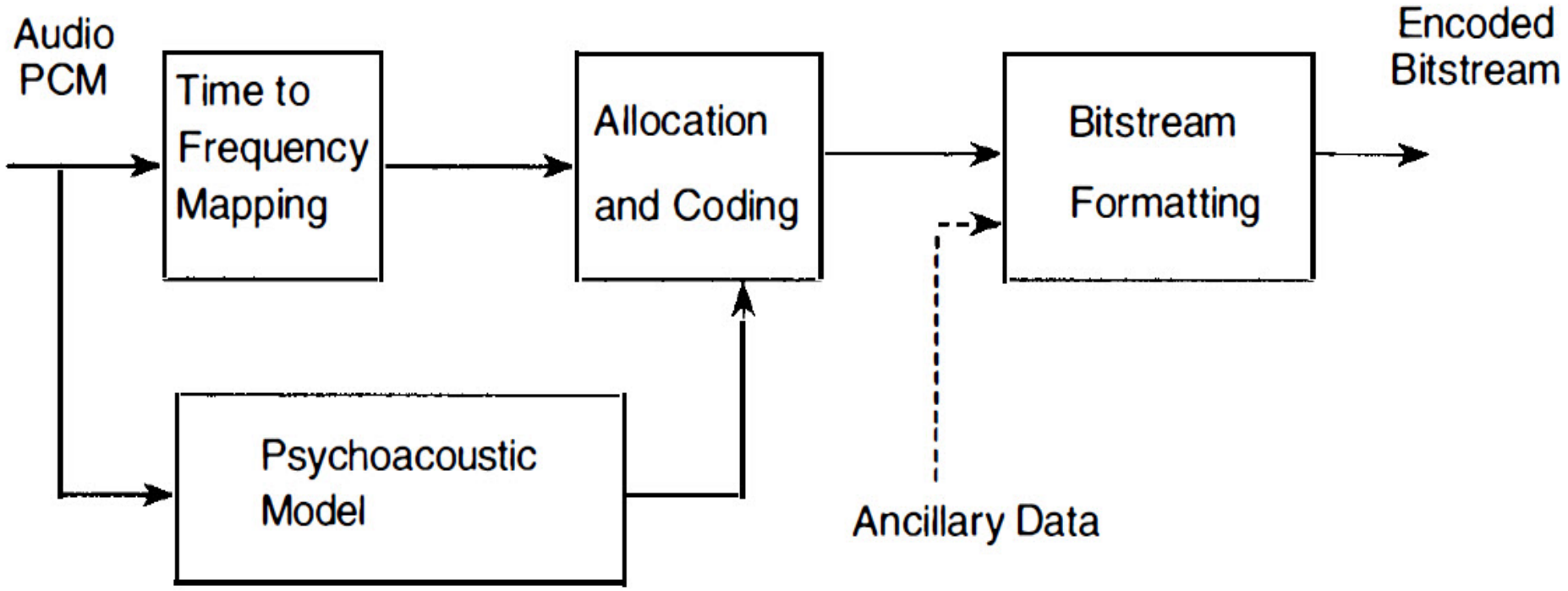

[3]

# The main challenge




There are some redundant data between the parts. What is the solution! And how to tackle this big problem.

# What is MP3 algorithm




[3]

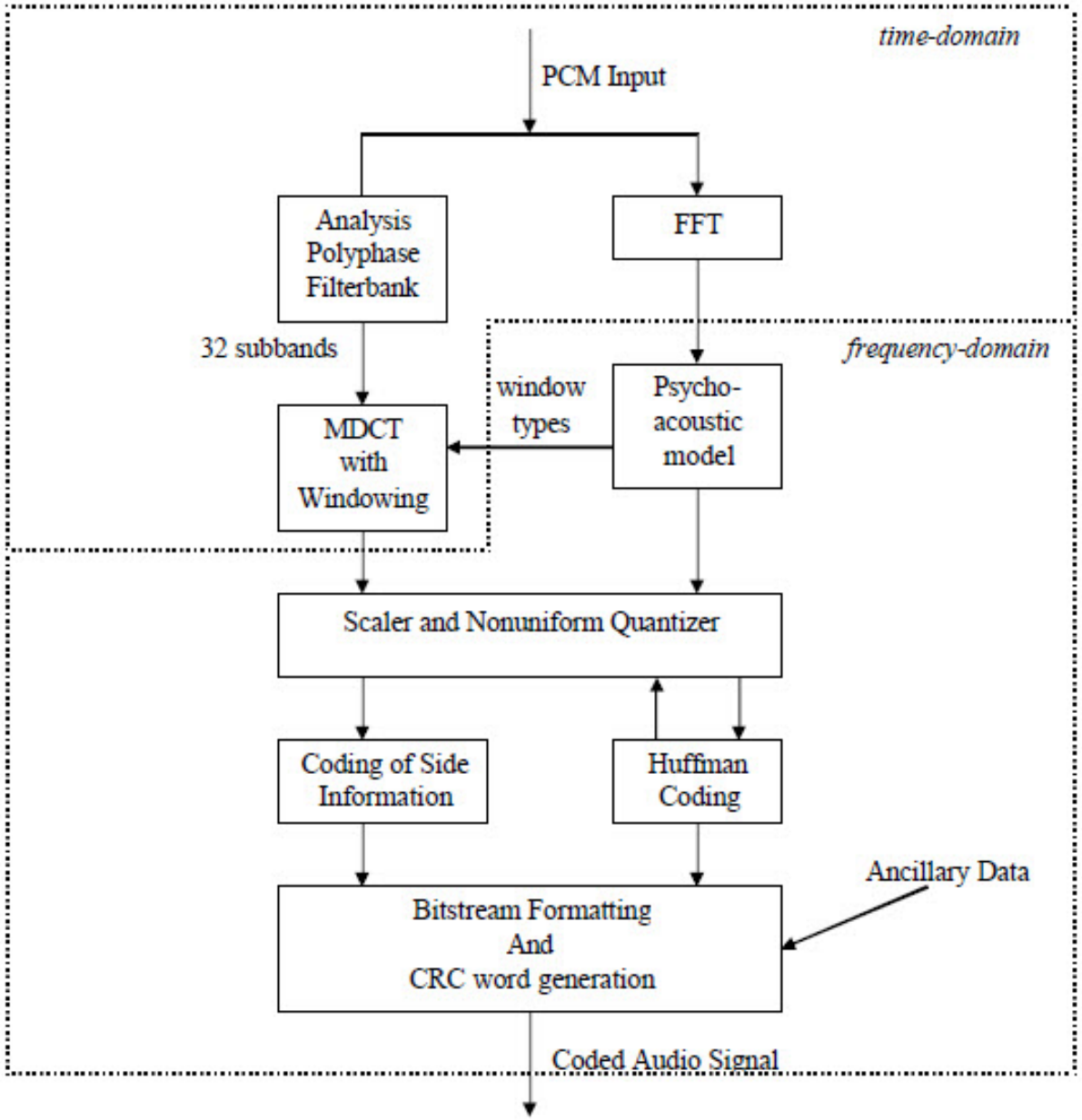
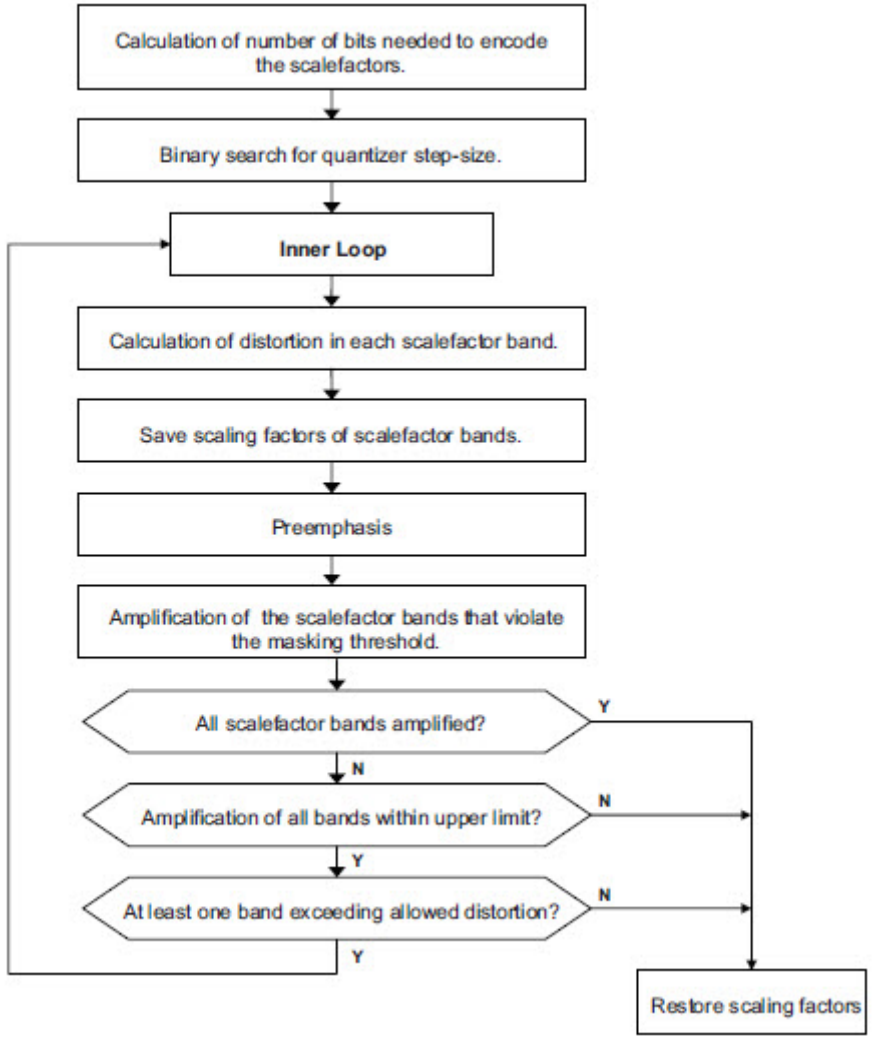



*Figure 6.1: MPEG-1 Layer III encoding scheme*

[3]

# Bit and noise allocation




[2]

# The solutions

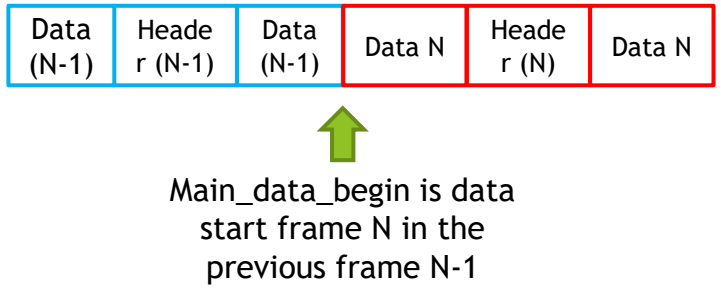
1. Analyzing the output mp3 frames. Unpack them and try to re-align by re-encoding the frames such as an algorithm here (https://freac.org/developer-blog-mainmenu-9/14-freac/287-superfastlame).
2. Modify the LAME that encode the part that can merge it with the next part. It seems easy. But It is so complex. In this task, I modify the LAME to support the multithreaded encoder.

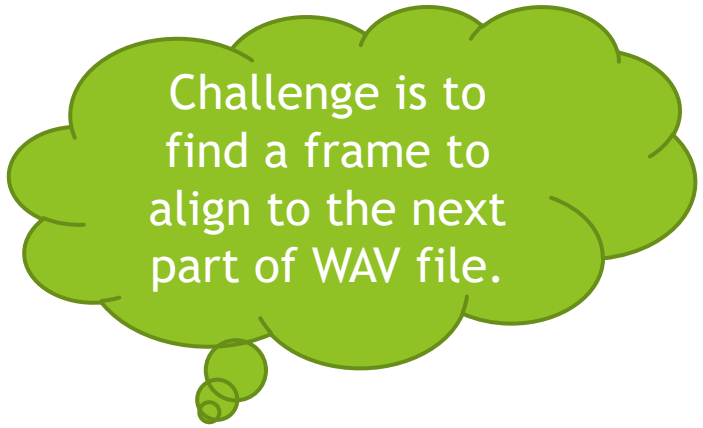
# LAME modification procedure

1. find how LAME encodes the WAV by detail.
2. Find the exact procedure to implement the MP3 stages.
3. Propose a way to tackle this problem.
4. Implement it and test.

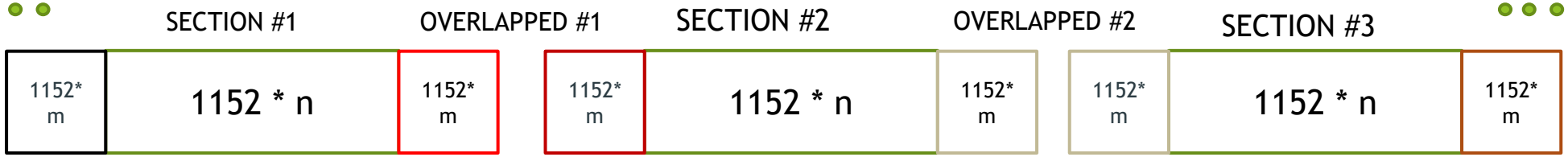
# MP3 Frame

- Each frame of MP3 contains 1152 samples of audio data after decoding.
- The beginning of the data for each frame is defined by main_data_begin parameter with is the number of bytes before starting header containing the data of header.



# Proposed Method : Overlapped sections




1. Partition the WAV file to the overlapped sections.
2. The information of OVERLAPPED section is encoded first.
3. header information and main_data_begin are passed to the section encoding process.
4. After encoding (n+m)* 1152, the search process to align the two parts is start.

# Search process to align two parts

Encoder is to encode the frame n+m+3
(the beginning frame of overlapped section is the best one. Because the encoder was not stable on the number of main_data_begin

Encode the frame and calculate the main_data_begin for the next frame and compare it to the desired main_data_begin of the corresponding frame in the overlapped section

If the calculated main_data_begin is larger than the main_data_begin the solution is find BUT HOW!!!

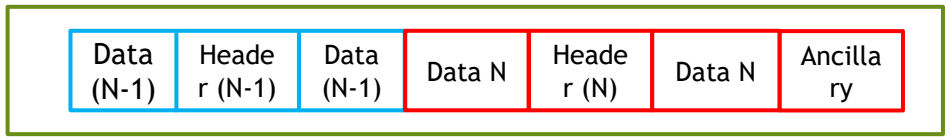
# Search process to align two parts

- If the calculated main_data_begin is larger than desired main_data_begin:
- At first the calculated frame is put in the bitstream and to compensate the rest, I put the required bit numbers to the **Ancillary data**. The ancillary data is the best choose to avoid the re-encoding the frame and change noise allocation process.
- The overlapped section is considered about 20*1152. So it is Okay to find a frame that the condition is Okay.
- But in the rare situation the calculated main_data_begin is not larger than desired main_data_begin. In this condition, I re-encode that frame to have smaller bits in the frame and the condition is satisfied.

AFTER about 15 frames the main_data_begin of the overlapped is very similar to the main_data_begin that are being calculated in this section. Because bit allocation consider about 10 previous noise to performing noise allocation.

# Search process to align two parts

Fr N> n+m SECTION #1 STOP encoding




Desired main_data_begin

SECTION #2


Truncate the next mp3 stream and combine to part

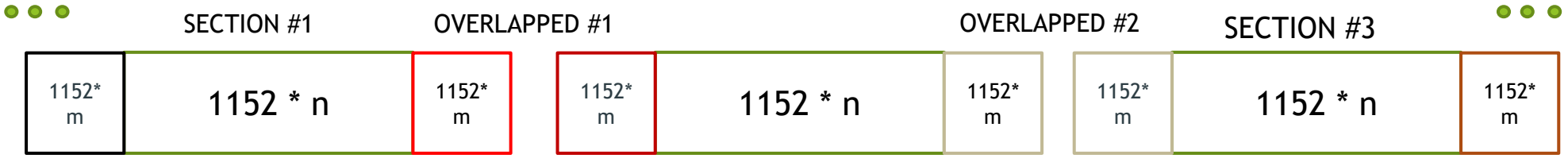
It seems the solution is found. But NOOOOOO….
There is another issue.

# Delay encoding in the beginning of the audio

- Polyphase filter bank and MDCT stages add some samples at the beginning of byte. 576 samples at the beginning of the audio file is not valid. So the beginning of the audio file should be padded by number of zero.
- So I change the encoding delay to the value that there are 1152 samples of zero at the beginning of the data. In this way, I tackle to this problem.

FINISHING the process. The LAME code was modified and all sections of my modification was commented beginning by /* jabarouti */

# Multithreaded Strategy




# Multi-cross Platform

- For supporting multi-cross, CMAKE is used to configure and generate for each platform.
- The required library is added to project in each platform by CMAKE.
- For retrieving the file in the directory, I used modern C++ std V17.

# Reading wav file

- Libsndfile is used to read the wav file.
- The LAME library can be compiled with the support of Libsndfile library. But I link it separately to the project.

# Multithreading : POSIX

- Because in the multithreading strategy, all the data is separated in each thread and each thread has it is own data. The synchronization techniques is not used unless in the output stream.
- POSIX is used. The windows version is available in Dll format.

# Conclusion

- The project is so exciting.
- It took one month that I can tackle to problem.
- It can be some other modification to the project. For example, Determinate the main data and overlapped section.

**Thank you so much.**

# References


- [1] ISO/IEC JTC1/SC29/WG11 MPEG, IS11172–3 “Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s, Part 3: Audio” 1992. (“MPEG-1”).
- [2] Analysis of the MPEG-1 Layer III (MP3) Algorithm Using MATLAB, Jayaraman J. Thiagarajan and Andreas Spanias
- [3] introduction to digital audio coding and standards, Marina Bosi
- [4] The Theory Behind Mp3, Rassol Raissi