

# Audio tömörítő egység megvalósítása FPGA-val

Nyiri Levente

2025 Október

## 1. Szabvány

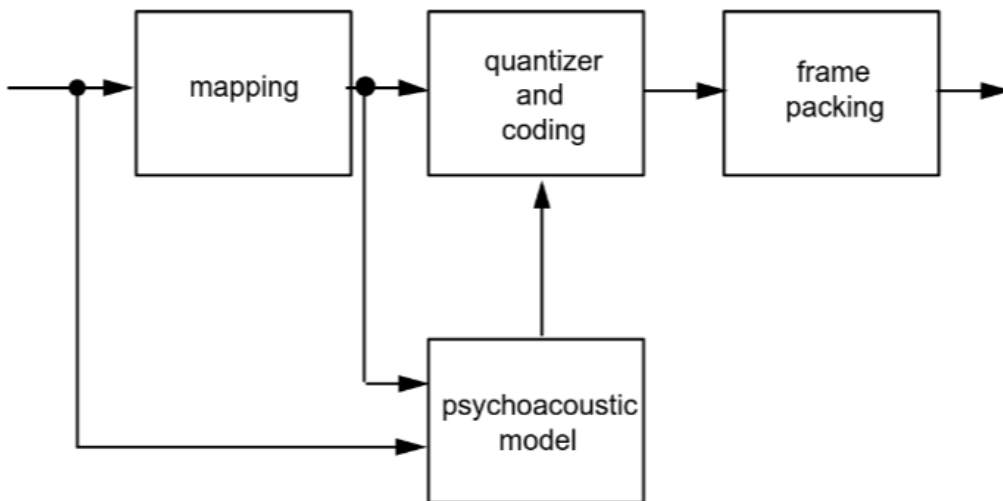
Tanulmányoztam az MPEG-1 Audio és az MPEG-2 Advanced Audio Coding (AAC) szabványokat, úgy döntöttem, hogy az MPEG-1 Audio-t fogom implementálni.

## 2. LAME

Kiindulásnak a LAME (Lame Aint an MP3 Encoder) nyílt forráskódú MP3 tömörítő szoftver forráskódját tanulmányoztam. A weboldalukról töltöttem le, fordítottam és egy példa .wav file-on kipróbáltam.

## 3. Encoder

Az encoder blokkvázlata[1] az 1. ábrán látható. Nem szabványosított az algoritmus, én a LAME source code-jából fogok kiindulni a tervezésnél. A kimeneti bitsream-nek meg kell felelnie az International Standard-nak.



1. ábra. Encoder blokkvázlat

### 3.1. Mapping

### 3.2. Frame

Egy MP3 file kisebb részegységekre van osztva, ezeket frame-eknek hívjuk. Minden frame 1152 audio mintát tartalmaz. Egy frame továbbá szét van választva 2 granule-ra, mindkettőre 576 minta jut.

Egy frame méretét byteban a(z) 1. egyenlet írja le.

$$\text{frame size} = \frac{144 \cdot \text{bitrate}}{f_s} + \text{Padding} \quad (1)$$

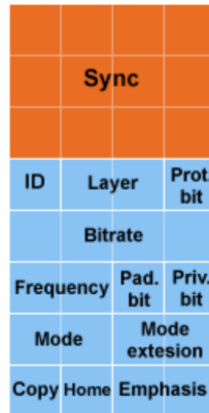
Egy frame felépítése a(z) 1. táblázatban [2] látható.

Header	CRC	Side_Information	Main_Data	Ancillary_Data
--------	-----	------------------	-----------	----------------

1. táblázat. Az MPEG frame fő mezői

#### 3.2.1. Header

A header tartalmazza a szinkronizációs biteket és egyéb információkat a frame-ről, felépítését mutatja a(z) 2. ábra [2].



2. ábra. Frame header

**Sync (12 bit)** Szerepe a szinkronizálás, mind a 12 bitnek 1-esnek kell lennie: `sync = 12'b1111_1111_1111`

**ID (1 bit)** Az MPEG verziót határozza meg (MPEG-1 vagy MPEG-2)

**Layer (2 bit)** Layer I, II vagy III

**Protection bit (1 bit)** Meghatározza, használunk-e CRC-t

**Bitrate (4 bit)** A bitrate beállítása

**Frequency (2 bit)** A mintavételi frekvencia meghatározása

**Padding bit (1 bit)** Néhány frame-nek szüksége van rá, hogy a bitrate pontos legyen

**Private bit (1 bit)** Applikáció-specifikus trigger

**Mode (2 bit)** Csatornamód

**Mode extension (2 bit)** Csak joint stereo esetén használatos, további specifikáció

**Copyright bit (1 bit)** Jelzi, ha a tartalom szerzői jogi védelem alatt áll

**Home bit (1 bit)** A frame az eredeti adathordozón található-e

**Emphasis (2 bit)** A dekódernek szükséges-e de-emphasist alkalmaznia zajcsökkentés után

### 3.2.2. Side information

A side information további információt tartalmaz arra vonatkozóan, hogy hogyan kell dekódolni a frame-et. A felépítését a(z) 2. táblázat mutatja.

main_data_begin	private_bits	scfsi	Side_info gr. 0	Side_info gr. 1
-----------------	--------------	-------	-----------------	-----------------

2. táblázat. A side information mezői

**main\_data\_begin (9 bit)** Layer III-nál bit reservoir használatával egy adott frame main data helyén szabadon maradt helyet másik frame-ek is használhatják. Ez a mező azt a negatív offsetet adja meg, amennyivel korábban kezdődik egy frame-nek a main data-ja a sync word-höz képest.

**private\_bits (5 bit)** Szabad felhasználás

**scfsi (4-4 bit)** ScaleFactor Selection Information, azt határozza meg, hogy az egyes scalefactor-ok mindkét granule-ra használhatóak-e vagy külön kell mindkettőre küldeni. Ha elég csak egyet küldeni, azzal biteket nyerünk, amit Huffman kódoláshoz lehet használni. Layer III-nál a scalefactorok 4 csoportba vannak osztva a(z) 3. táblázat szerint.

Mindkét csatornára vonatkozóan, ha egy bit 1-esbe van, akkor az adott csoporthoz tartozó scalefactorokat a frameben lévő második granule újra fogja használni.

Ha short window-t használunk (`block_type == 2`) bármely granule-ban, akkor a scalefactorokat mindig külön küldjük mindkét csatornára.

group	scalefactor bands
0	0,1,2,3,4,5
1	6,7,8,9,10
2	11,12,13,14,15
3	16,17,18,19,20

3. táblázat. Scalefactor csoportok

A granule-okhoz tartozó **side\_info** még további mezőkből áll, ezt mutatja a(z) 4. táblázat.

<b>part2_3_length</b>	<b>big_values</b>	<b>global_gain</b>	<b>scalefac_compress</b>
<b>windows_switching_flag</b>	<b>block_type</b>	<b>mixed_block_flag</b>	<b>table_select</b>
<b>subblock_gain</b>	<b>region0_count</b>	<b>region1_count</b>	<b>preflag</b>
<b>scalefac_scale</b>	<b>count1table_select</b>		

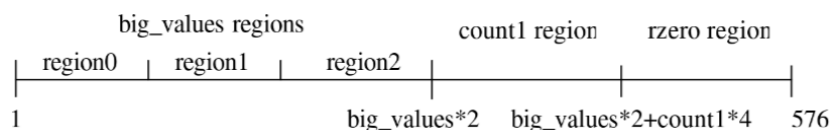
4. táblázat. Granule side information mezői

**par2\_3\_length (12-12 bit)** Megmondja, hogy a main data részében a frame-nek hány bit van allokalva scalefactor-oknak (part2) és Huffman kódolt adatnak (part3).

**big\_values (9-9 bit)** Az egyes granule-ok spektrális komponensei más Huffman kód táblázatokkal vannak kódolva. A teljes spektrum 0-tól a Nyquist frekvenciáig több részre van bontva, és ezek a részek máshogy vannak kódolva. A partícionálás a maximális kvantált értékek alapján

történik. A magasabb frekvenciájú komponenseknek várhatóan alacsonyabb amplitúdójuk van, vagy nem is kell őket kódolni. Megszámoljuk, hogy minden frekvencián összesen hány 0 érték van, ezeknek a számát az **rzero**-ban tároljuk. A **count1** mezőben 4-esével vannak értékek, és az abszolútértéke nem haladja meg az 1-et (az egyes értékek -1, 0 vagy 1 lehetnek). A többi érték a **big\_values** mezőben van, ezen belül is 3 részre osztva. Az abszolút maximum értéke ennek a régióknak 8191.

A **big\_values** mező felosztása a(z) 3. ábrán látható.



3. ábra. **big\_values** felosztása

**global\_gain (8-8 bit)** Kvantálás lépésköz.

**scalefac\_compress (4-4 bit)** Hány bitet használjon a scalefactorok átviteléhez. Hogy az értéke alapján az első és második scalefactor group hány bitet kap az a(z) 5. táblázatban látható.

**windows\_switching\_flag (1-1 bit)** Megmutatja, ha a normálon kívül másmilyen ablaktípus van használatban.

**block\_type (2-2 bit)** Ha nem normál ablakot használunk, mutatja, hogy milyen. A lehetőségek a 6. táblázatban vannak.

**mixed\_blockflag (1-1 bit)** Akkor használható, ha a **windows\_switching\_flag** be van állítva. Azt jelzi, hogy más ablaktípust használunk alacsonyabb és magasabb frekvenciákon. Az alsó 2 sávban normál, a maradék 30-ban pedig a **block\_type**-ban megadottal.

**table\_select (10-10 bit)** Megadja milyen Huffman kódolást használjunk.

**subblock\_gain (9-9 bit)** Ha **windows\_switching\_flag** set és **block\_type** == 10, akkor a gain offsetet mutatja a global gain-től.

**region\_address1 (4-4 bit) region\_address2 (3-3 bit)** A spektrumot további részekre osztjuk a Huffman kódoláshoz, ezek a változók ezeknek a régióknak a kezdőcímét tartalmazzák.

scalefac_compress	slen1	slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2
13	3	3
14	4	2
15	4	3

5. táblázat. Scalefac compress értékek

block_type	window type
00	forbidden
01	start
10	3 short windows
11	end

6. táblázat. Block type értékek

**preflag (1-1 bit)** További nagyfrekvenciás erősítése a kvantált mintáknak.  
Ha set, akkor értékeket ad hozzá a scalefactorokhoz.

**scalefac\_scale (1-1 bit)** A scalefactorok logaritmikusan kvantáltak a(z)  
7. táblázat szerint.

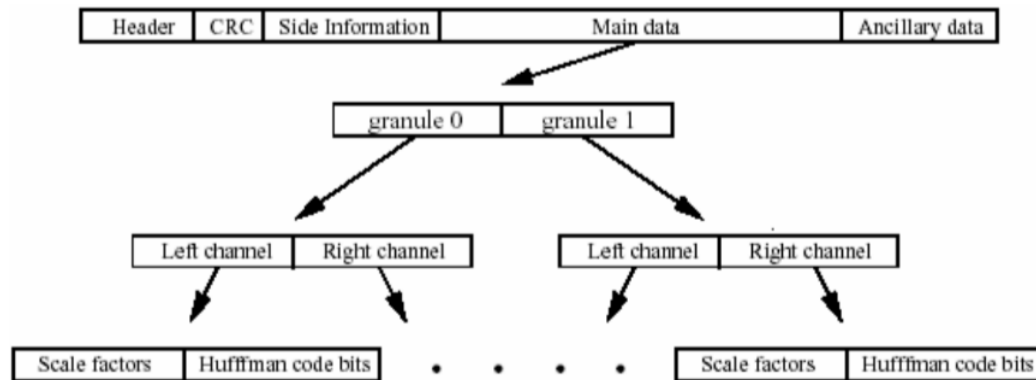
scalefac_scale	step size
0	$\sqrt{2}$
1	2

7. táblázat. Scalefac scale értékek

**count1table\_select (1-1 bit)** Huffman kódolást választ a count1 mező-  
ben lévő értékekhez.

### 3.2.3. Main Data

Scalefactorokból, Huffman kódolt bitekból és ancillary data-ból áll. Az elrendezés a(z) 4. ábrán látható.



4. ábra. Main data, granule-ok és channelek elrendezése

**Scale factors** A céljuk, hogy redukálják a kvantálási zajt. Ha a minták egy adott scalefactor band-ben megfelelően vannak scale-elve, akkor a kvantálási zaj teljesen ki lesz maszkolva.

**Huffman code bits** Itt találhatóak a Huffman kódolt bitek.

**Ancillary data** Opcionális, felhasználó definiálhatja.

### 3.3. Megkötések

Mivel egy áltanános, a header-ben megadható minden opciót kiszolgáló megvalósítás túl sok időbe telne, ezért bizonyos megkötésekkel élni fogunk.

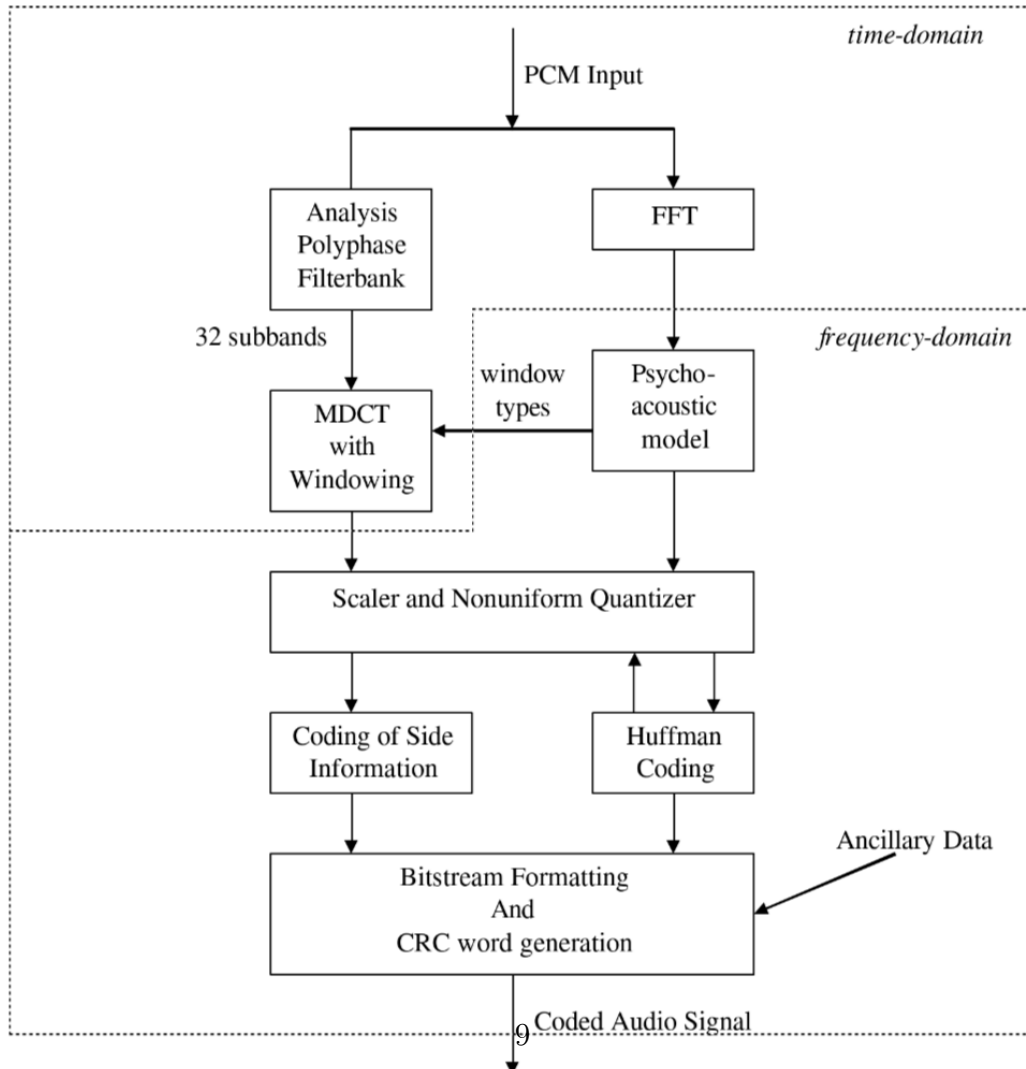
A megkötéseket a(z) 8. táblázat mutatja.



Mező Neve	Bit Érték	Jelentés
ID	1	MPEG-1
Layer	01	Layer III
Protection bit	1	Használunk CRC-t
Bitrate	1110	320 Kbps
Frequency	00	44.1 kHz mintavételi frekvencia
Mode	00	Stereo
Emphasis	00	Nem kell de-emphasis dekódolásnál, nem használunk noise suppression-t

8. táblázat. Megkötések a bitekre

### 3.4. Analysis Polyphase Filterbank



5. ábra. Encoder blokkvázlat részletesebben

A(z) 5. ábrán látható egy részletesebb blokkvázlat az encoderről.

Az Analysis Polyphase Filterbankba megy a PCM input, itt az 1152 PCM minták mindegyike 32 egyenletesen eloszló frekvencia sávba lesz szétosztva. Mivel  $f_s = 44.1kHz$  ezért a Nyquist frekvencia  $f_{Nyquist} = 22.05kHz$ . Minden sáv

$$\frac{22050}{32} = 689.0625Hz$$

széles lesz. Minden minta tartalmazhat 0-22.05 kHz-s komponenseket, ami a 32 sáv közül a megfelelőbe lesz szűrve. Mivel minden mintát 32 subband-re osztunk, ezért az eredeti 1152-ből  $1152 \cdot 32 = 36864$  mintánk lesz. A folyamat végén viszont 32-vel decimálunk minden subband-et, így ismét 1152 mintánk lesz.

### 3.5. FFT

Először is ki kell találni, hogy egy órajel alatt hány FFT-t kell megcsinálni, hány műveletvégző egység kell.

Layer III-nál két FFT-t kell párhuzamosan. Block length == window length.

**Short blocks** shift length = 192 samples, window length = 256 samples

**Long blocks** shift length = 576 samples, window length = 1024 samples

Mivel  $f_s = 44.1kHz$ , ezért

$$window\_size_1 = \frac{256}{44100} = 5.8ms$$

$$window\_size_2 = \frac{1024}{44100} = 23.2ms$$

$$frequency\_resolution_1 = \frac{44100}{256} = 172.3Hz$$

$$frequency\_resolution_2 = \frac{44100}{1024} = 43.1Hz$$

Látszik, hogy a kisebb ablakok jobb felbontást biztosítanak időtartományban, a hosszabb ablakok pedig frekvencia tartományban.

A cél megtalálni az AMD által biztosított FFT core-nak azt a felparaméterezését, amely a lehető legkisebb erőforrás felhasználás nélkül teljesíteni tudja a követelményeket.

**Long blocks:**

- Shift length: 576 samples
- FFT méret: 1024 points
- Overlap:  $1024 - 576 = 448$  samples (43.75%)

Frame  $N$  : samples  $[0 \dots 1023]$

Frame  $N + 1$  : samples  $[576 \dots 1599]$

Frame  $N + 2$  : samples  $[1152 \dots 2175]$

**Short blocks:**

- Shift length: 192 samples
- FFT méret: 256 points
- Overlap:  $256 - 192 = 64$  samples (25%)

Frame  $N$  : samples  $[0 \dots 255]$

Frame  $N + 1$  : samples  $[192 \dots 447]$

Frame  $N + 2$  : samples  $[384 \dots 639]$

Hosszú ablakoknál  $\frac{576}{44100} = 13.6ms$  idő van FFT-re, rövidnél  $\frac{192}{44100} = 4.35ms$

## Hivatkozások

- [1] ISO/IEC, „Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio”, International Organization for Standardization, Geneva, Switzerland, International Standard ISO/IEC 11172-3:1993, 1993, (MPEG-1 Audio).
- [2] R. Raissi, „The Theory Behind Mp3”, 2002. cím: <https://api.semanticscholar.org/CorpusID:12806533>.