

Audio tömörítő egység megvalósítása FPGA-val

Nyiri Levente

2025 Október

1. Szabvány

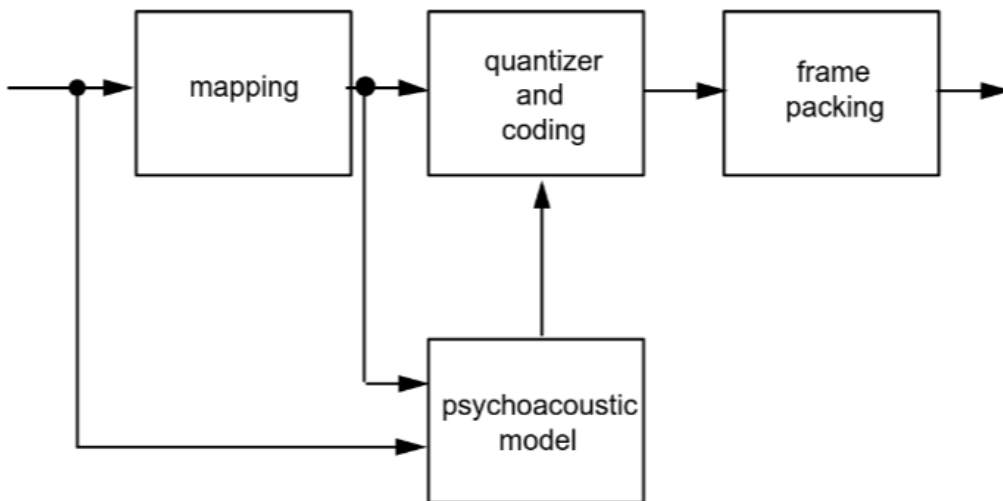
Tanulmányoztam az MPEG-1 Audio és az MPEG-2 Advanced Audio Coding (AAC) szabványokat, úgy döntöttem, hogy az MPEG-1 Audio-t fogom implementálni.

2. LAME

Kiindulásnak a LAME (Lame Aint an MP3 Encoder) nyílt forráskódú MP3 tömörítő szoftver forráskódját tanulmányoztam.

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. ábra írja le.

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

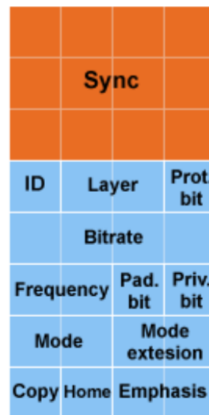
Egy frame felépítése a(z) 1. táblázatban [2] látható.

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.

part2_3_length	big_values	global_gain	scalefac_compress
windows_switching_flag	block_type	mixed_block_flag	table_select
subblock_gain	region0_count	region1_count	preflag
scalefac_scale	count1table_select		

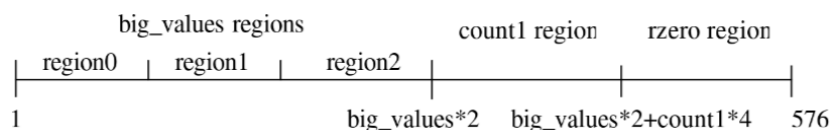
4. táblázat. Granule side information mezői

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 particionálás 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, hogyha 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 használunk.

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.

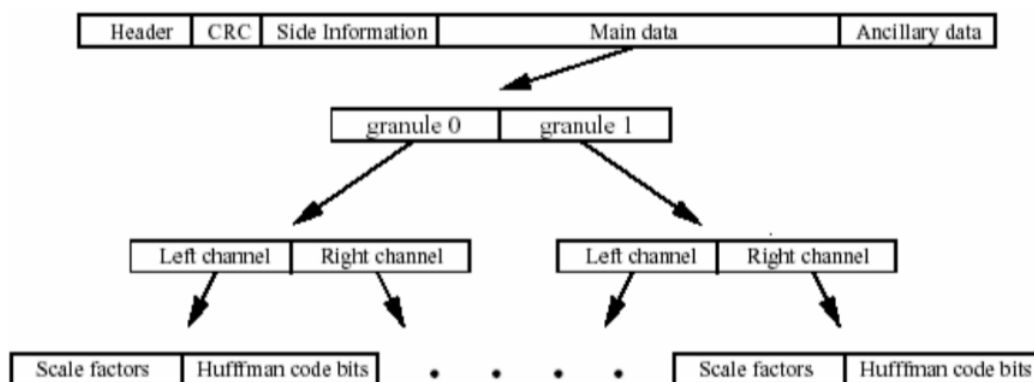
scalefac_scale	step size
0	$\sqrt{2}$
1	2

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.

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>.