



# УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

## ЗАВРШНИ (BACHELOR) РАД

Кандидат: **Јована Милановић**

Број индекса: **ра 175/2016**

Тема рада: **Реализација модула за еквализацију спектра аудио сигнала на платформи заснованој на ARM процесорском језгру**

Ментор рада: **доц. др Јелена Ковачевић**

Нови Сад, јул, 2021. год.



## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска документација	
Тип записа, ТЗ:	Текстуални штампани материјал	
Врста рада, ВР:	Завршни (Bachelor) рад	
Аутор, АУ:	Јована Милановић	
Ментор, МН:	доц. др Јелена Ковачевић	
Наслов рада, НР:	Реализација модула за еквализацију спектра аудио сигнала на платформи заснованој на ARM процесорском језгру	
Језик публикације, ЈП:	Српски / латиница	
Језик извода, ЈИ:	Српски	
Земља публиковања, ЗП:	Република Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2021	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/41/7/1/27/0/0	
Научна област, НО:	Електротехника и рачунарство	
Научна дисциплина, НД:	Рачунарска техника	
Предметна одредница/Кључне речи, ПО:	Дигитална обрада сигнала, уједначавање сигнала, еквилизатор	
УДК		
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У оквиру рада је имплементирано једно решење модула за уједначавање спектра аудио сигнала на платформи производача NXP, заснованој на ARM процесорском језгру	
Датум прихватања теме, ДП:		
Датум одбране, ДО:		
Чланови комисије, КО:	Председник:	Проф. др Мирослав Поповић
	Члан:	Проф. др Иван Каштелан
	Члан, ментор:	Доц. др Јелена Ковачевић
		Потпис ментора



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO:</b>		
Identification number, <b>INO:</b>		
Document type, <b>DT:</b>	Monographic publication	
Type of record, <b>TR:</b>	Textual printed material	
Contents code, <b>CC:</b>	Bachelor Thesis	
Author, <b>AU:</b>	Jovana Milanović	
Mentor, <b>MN:</b>	PhD Jelena Kovačević	
Title, <b>TI:</b>	Digital audio equalizer implementation on platform based on ARM processor core	
Language of text, <b>LT:</b>	Serbian	
Language of abstract, <b>LA:</b>	Serbian	
Country of publication, <b>CP:</b>	Republic of Serbia	
Locality of publication, <b>LP:</b>	Vojvodina	
Publication year, <b>PY:</b>	2021	
Publisher, <b>PB:</b>	Author's reprint	
Publication place, <b>PP:</b>	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, <b>PD:</b> (chapters/pages/ref.tables/pictures/graphs/applications)	7/41/7/1/27/0/0	
Scientific field, <b>SF:</b>	Electrical Engineering	
Scientific discipline, <b>SD:</b>	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, <b>S/KW:</b>	Digital signal processing, equalization, equalizer	
<b>UC</b>		
Holding data, <b>HD:</b>	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, <b>N:</b>		
Abstract, <b>AB:</b>	This thesis describes one solution for an implementation of digital audio equalizer on NXP platform based on ARM processor core	
Accepted by the Scientific Board on, <b>ASB:</b>		
Defended on, <b>DE:</b>		
Defended Board, <b>DB:</b>	President:	PhD Miroslav Popovic
	Member:	PhD Ivan Kastelan
	Member, Mentor:	PhD Jelena Kovacevic
		Menthor's sign

## **Zahvalnost**

Za izradu ovog rada zahvalnost dugujem svom tehničkom mentoru Nenadu Pekezu, kao i teorijskom mentoru doc. Dr Jeleni Kovačević. Takođe, zahvalila bih se i kolegama iz kancelarije, koji su mi pružali podršku tokom procesa pisanja diplomskog rada.

Posebnu zahvalnost dugujem svojoj čerki, koja mi je pružila glavnu motivaciju da istrajam na svom putu i svom suprugu, koji me je bodrio da pratim svoje želje.



## **SADRŽAJ**

1.	Uvod .....	8
2.	Teorijske osnove.....	10
2.1	2.1 Istorijski osvrt .....	10
2.2	2.2 Parametarski ekvilizator.....	11
2.3	Grafički ekvilizator .....	11
2.3.1	Osnovni pojmovi .....	11
2.3.2	Opis implementacije grafičkog ekvilizatora.....	12
2.3.3	Postojeće realizacije .....	12
2.3.4	Bikvad filter.....	14
2.4	Izbor ciljne hardverske platforme .....	15
3.	Koncept rešenja.....	16
3.1	Razvoj aplikacije za DS procesor .....	16
3.2	Razvoj aplikacije za platformu zasnovanu na ARM procesorskom jezgru .....	17
3.3	Razvoj GEQ za platformu zasnovanu na ARM procesorskom jezgru .....	17
4.	Implementacija .....	19
4.1	Faza 1 - analiza referentnih modela .....	19
4.2	Faza 2 – izrada C referentnog modela .....	21
4.3	Faza 3 .....	22
4.4	Karakteristike ciljne platforme .....	22
4.5	Razvojno okruženje .....	23
4.5.1	Faza 1 .....	23
4.5.2	Faza 2.....	24
4.5.3	Faza 3.....	25
5.	Programsko rešenje.....	27
5.1	Struktura history.....	27



# УНИВЕРЗИТЕТ У НОВОМ САДУ

## ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



5.2	Struktura biquad.....	27
5.3	Funkcija init .....	27
5.4	Funkcija coeffsGEQI .....	27
5.5	Funkcija coeffsGEQII .....	28
5.6	Funkcija process.....	28
6.	Ispitivanje predloženog rešenja .....	29
6.1	Faza 2 .....	29
6.1.1	Analiza rezultata dobijenih primenom GEQ I.....	29
6.1.2	Analiza rezultata dobijenih primenom GEQ II .....	32
6.1.3	Poređenje rezultata dobijenih primenom GEQ I i GEQ II algoritma.....	36
6.2	Faza 3 – verifikacija rezultata u simulatoru .....	38
7.	Zaključak .....	39
8.	Literatura .....	41

## SPISAK SLIKA

Slika 2.1: Grafički ekvilizator programskog paketa Audacity .....	11
Slika 2.2: Kaskada filtera grafičkog ekvilizatora [1].....	13
Slika 2.3: Direktna forma II.....	14
Slika 2.4: Transponovana direktna forma II .....	14
Slika 3.1: Tok implementacije softvera na digitalnim signal procesorima sa aritmetikom u nepokretnom zarezu [5] .....	17
Slika 4.1: i.MX 8M Mini sistem blok dijagram [7] .....	23
Slika 4.2: CLIDE integrисано razvojno okruženje.....	24
Slika 4.3: Visual Studio - kontrolisano izvršavanje programa .....	25
Slika 4.4: Trace32 simulator - kontrolisano izvršavanje programa .....	26
Slika 6.1: test Killed, GEQ I.....	30
Slika 6.2: test AllUp, GEQ I.....	30
Slika 6.3: test BP4, GEQ I .....	31
Slika 6.4: test Highpass, GEQ I .....	31
Slika 6.5: test Lowpass, GEQ I .....	32
Slika 6.6: test killedAll, GEQ II .....	33
Slika 6.7: test UpDown, GEQ II .....	33
Slika 6.8: test BP3, GEQ II .....	34
Slika 6.9: test BP4, GEQ II .....	34
Slika 6.10: test AllUp, GEQ II .....	34
Slika 6.11: test Lowpass, GEQ II .....	35
Slika 6.12: test Highpass, GEQ II .....	35

Slika 6.13: test Highpass.....	36
Slika 6.14: test Highpass.....	36
Slika 6.15: test AllUp.....	37
Slika 6.16: test BP4.....	37
Slika 6.17: test UpDown.....	37
Slika 6.18: Ručno bit poređenje izlaza GEQ I i simulatorskog projekta .....	38

## **SPISAK TABELA**

Tabela 2.1: ISO standard . . ..... 13

## SKRAĆENICE

- **GEQ** – grafički ekvilizator
- **GREQ** – grafički referentni ekvilizator
- **PEQ** – parametarski ekvilizator
- **PREQ** – parametarski referentni ekvilizator
- **DSP** – digitalni signal procesor
- **IDE** – integrisano razvojno okruženje

## 1. Uvod

Digitalna obrada signala je omogućila savremenom čoveku da svoju svakodnevnicu obogati ili olakša putem performansi širokog spektra potrošačke elektronike. Kako bi očekivanja i potrebe potrošača bile zadovoljene, digitalna obrada signala se konstantno razvija, unapređuje i ubrzava.

Zadatak ovde prezentovanog rada je upravo jedan aspekt digitalne obrade signala, tj. Realizacija modula za ujednačavanje spektra audio signala na platformi zasnovanoj na ARM procesorskom jezgru. U tu svrhu je bilo potrebno implementirati grafički ekvilizator sa 31 frekvencijskim opsegom namenjen izvršavanju na i.MX 8M Mini platformi kompanije NXP, koja sadrži ARM Cortex-A53 I Cortex-M4 procesorska jezgra. Kao referentni modeli su korišćene već postojeće dve implementacije GEQ-a za digitalne signal procesore (Cirrus Logic CS498xx familija digitalnih signal procesora i TMS320C6000<sup>TM</sup> DSP proizvođača Texas Instruments). Kompletna aplikacija je razvijena u C programskom jeziku na računaru opšte namene u softverskom razvojnog alatu Visual Studio 2019.

Ekvilizator modul je bilo potrebno realizovati kao zasebnu biblioteku koja bi kasnije bila uključena u osnovne gradivne blokove aplikacije za grafičko programiranje audio obrade. Verifikaciju modula je potrebno izvršiti kroz nekoliko iterativnih koraka, od računara opšte namene, preko simulatora za ciljnu platformu i konačno na samoj platformi.

Cilj rada je bio upoznavanje sa razvijanjem aplikacija u realnom vremenu za platformu zasnovanu na ARM procesorskom jezgru sa ograničenim resursima, počev od razvoja algoritma, njegove implementacije u programskom jeziku C, pa do različitih mehanizama debagovanja, te automatizacije testiranja.

Rad se sastoji od osam poglavlja. U drugom poglavlju je predstavljena teorijska pozadina ekvilizatora, uz poseban akcenat na struktura rešenja i dizajn. Treće poglavlje je namenjeno

opisu koncepta rešenja, dok je u četvrtom dat detaljan opis koraka u implementaciji modula, uz osvrt na razvojno okruženje i ciljnu platformu. U petom poglavlju su opisane glavne funkcije kreiranog programa u programskom jeziku C. Šesto poglavlje je rezervisano za iznošenje rezultata na osnovu primene testova poređenja spektralnih slika subjektivnom, vizuelnom metodom, ali i objektivnim ručnim bit-identičnim testovima. U narednom, sedmom poglavlju, dat je rezime i izvedeni su zaključci na osnovu dobijenih rezultata, dok je u poslednjem, osmom poglavlju navedena literatura na koju se autor oslanjao prilikom izrade rada.

## 2. Teorijske osnove

Ekvalizacija je proces podešavanja balansa među frekvencijskim komponentama u okviru elektronskog signala. Najpoznatija upotreba ekvalizacije je u audio snimanju i reprodukciji, ali postoje i mnoge druge upotrebe u elektronici i telekomunikacijama.

U audio obradi signala, ekvalizacija je proces modifikacije frekvencijskog odziva audio sistema korišćenjem linearnih filtera. Uređaj koji se koristi da bi se postigla ekvalizacija se naziva ekvilibrator. Ovaj uređaj omogućava korisniku podešavanje amplitude konkretnih frekvencijskih opsega audio signala. Postoje grafički i parametarski ekvilibrator koji se odlikuju značajnom fleksibilnošću u modifikaciji frekvencija u okviru datog signala, a razlikuju se pre svega po tome što parametarski ekvilibrator dozvoljava korisniku veći broj kontrola, što je u narednim poglavljima detaljnije objašnjeno [1].

### 2.1 Istorijski osvrt

Termin ekvilibrator potiče iz rane faze razvoja telefonske mreže kada je bilo potrebno obezbediti da frekventni odziv sistema bude jednak za sve frekvencije u spektru signala koji se prenosi mrežom. Od tada se ovaj termin koristi za sve procedure podešavanja magnitude frekventnog odziva.

Prva upotreba grafičkog ekvilizatora je bila u filmovima 50-tih godina XX veka u svrhe poboljšanja kvaliteta zvuka. Ekvilibrator koji je omogućio arbitarnu selekciju centralnih frekvencija i pojačanja u tri preklapajuća opsega je predstavljen 1971. Godine. Bio je to prvi parametarski ekvilibrator.

U samom početku razvoja, ekvilizatori su bili analogni, da bi se tokom 1980-ih godina razvila i prva digitalna rešenja. Vremenom, ekvilizatori postaju standardni alat u muzičkoj produkciji i audio sistemima [2].

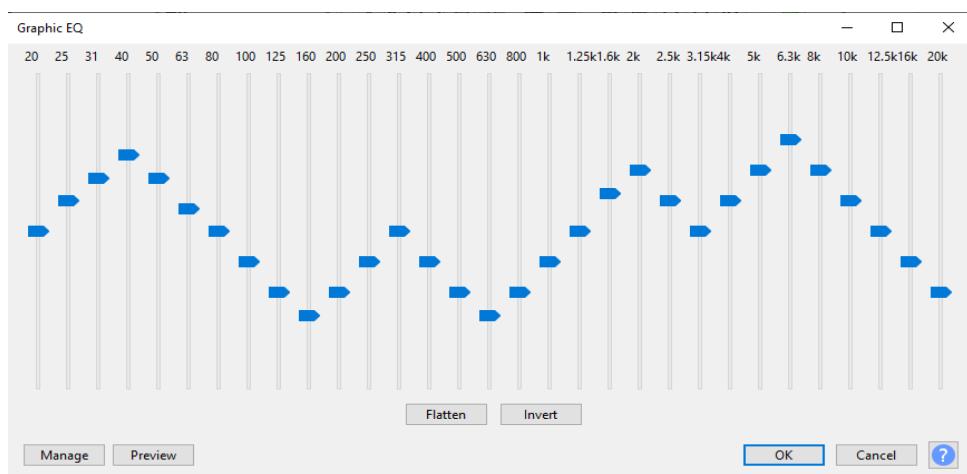
## 2.2 Parametarski ekvilizator

Parametarski ekvilizator daje mogućnost korisniku da podešava tri parametra: pojačanje, centralnu frekvenciju i širinu opsega (zakošenost). Time se postiže da korisnik može da pojača ili smanji potpuno proizvoljnu frekvenciju u spektru audio signala. Ostale frekvencije, koje su na dovoljno velikoj udaljenosti od one koju podešavamo, ostaju nedirnute. Pojačanje određenih frekvencija može nam pomoći da učinimo zvuk pojedinačnog instrumenta u orkestru izražajnijim, ili da damo kolorit zvuku modifikacijom određenih frekvencijskih opsega, na primer. Utišavanje izvesnih frekvencija nam omogućuje da eliminišemo šum ili druge neželjene zvuke. Kako bismo što preciznije uticali na konkretnu frekvenciju u spektru, a da pritom ne utičemo na ostale spektralne komponente, koristićemo uzan opseg. Za realizaciju parametarskog ekvilizatora se najčešće koriste peaking/notch filteri drugog reda povezani u kaskadu [1].

## 2.3 Grafički ekvilizator

### 2.3.1 Osnovni pojmovi

Grafički ekvilizator je, poput parametarskog, alat za podešavanje pojačanja višestrukih frekvencijskih opsega audio signala. Ono što razlikuje ova dva tipa ekvilizatora je to što je grafički ekvilizator manje fleksibilan. Jedina kontrola kojom korisnik može da manipuliše je pojačanje, dok su centralne frekvencije i širina opsega unapred definisani. Strukturno, on predstavlja skup filtera drugog reda. Odrednica „grafički“ ukazuje na činjenicu da se položaj dugmadi klizača može interpretirati kao graf odziva, što grafički ekvilizator čini intuitivnim za upotrebu. Zbog toga je veoma popularan kao uređaj za poboljšanje zvuka, iako je restriktivniji u kontrolama od parametarskog ekvilizatora [1].



Slika 2.1: Grafički ekvilizator programskog paketa Audacity

### 2.3.2 Opis implementacije grafičkog ekvilizatora

Opseg je osnovna jedinica grafičkog ekvilizatora. Opsezi su tipično distribuirani logaritamski, kako bi odgovarali percepciji čoveka. Širina opsega  $B_m$  za  $m = 1, \dots, M$  je razlika između gornje  $w_{u,m}$  i donje granice opsega  $w_{l,m}$ , tj. najniže i najviše frekvencije koje pripadaju datom frekvencijskom opsegu (Formula 1).

$$B_m = w_{u,m} - w_{l,m} \quad (1)$$

Gornja granica m-tog opsega je donja granica opsega m+1. Logaritamska distribucija opsega je osigurana korišćenjem fiksiranog racia R, tj.

$$B_{m+1} = R * B_m \quad (2)$$

$$w_{l,m+1} = R * w_{l,m} \quad (3)$$

$$w_{u,m+1} = R * w_{u,m} \quad (4)$$

$$w_{M,m} = \sqrt{w_{l,m} * w_{u,m}} \quad (5)$$

$$B_m = w_{u,m} - w_{l,m} = (\sqrt{R} - 1/\sqrt{R}) * w_{M,m} \quad (6)$$

Širina m-tog opsega je približno proporcionalna centralnoj frekvenciji  $w_{c,m}$  što za posledicu ima da će opsezi sa rastom centralnih frekvencija biti sve širi. To Q faktor, tj. zakošenost, čini gotovo konstantnim za sve frekvencijske opsege grafičkog ekvilizatora:

$$Q = w_{c,m} / B_m = \sqrt{R} / (R-1) \quad (7)$$

### 2.3.3 Postojeće realizacije

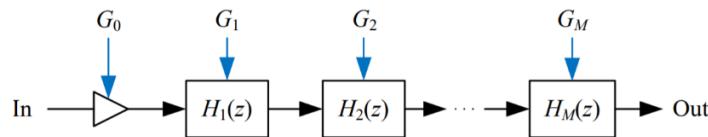
Kada je u pitanju broj frekvencijskih opsega i njihova širina, dva su uobičajena dizajna grafičkog ekvilizatora: oktava filter i trećina oktave filter. U oktava grafičkom ekvilizatoru, jedna oktava odgovara jednom frekvencijskom opsegu, tj. jednom filteru, pa je  $R = 2$  između svih opsega. U tom slučaju je  $Q = \sqrt{2} \approx 1.41$ . U trećina oktave dizajnu, jednu oktavu prepokrivaju čak tri filtera, te je u ovom slučaju  $R^3 = 2$ , tj.  $R \approx 1.26$ , dok je  $Q = 4.32$ .

$f_l$ (Hz)	$f_c$ (Hz)	$f_u$ (Hz)	$f_l$ (Hz)	$f_c$ (Hz)	$f_u$ (Hz)
22.4	25	28.2	708	800	891
28.2	31.5	35.5	891	1,000	1,122
35.5	40	44.7	1,122	1,250	1,413
44.7	50	56.2	1,413	1,600	1,778
56.2	63	70.8	1,778	2,000	2,239
70.8	80	89.1	2,239	2,500	2,818
89.1	100	112	2,818	3,150	3,548
112	125	141	3,548	4,000	4,467
141	160	178	4,467	5,000	5,623
178	200	224	5,623	6,300	7,079
224	250	282	7,079	8,000	8,913
282	315	355	8,913	10,000	11,220
355	400	447	11,220	12,500	14,130
447	500	562	14,130	16,000	17,780
562	630	708	17,780	20,000	22,390

**Tabela 2.1:** ISO standard za frekvencijski opseg od 1/3 oktave.  $f_c$  - centralna frekvencija ,  $f_l$  - donja granica opsega,  $f_u$  - gornja granica opsega [1]

Kada je u pitanju struktura grafičkog ekvilizatora, razlikujemo takođe dva moguća dizajna.

Jedan od načina da se konstruiše grafički ekvilizator je uvezivanje u kaskadu više peaking/notch filtera, kao što je prikazano na slici, gde je  $M$  broj filtera. Obično su prvi i poslednji filter u kaskadi šelving filteri. Može se opcionalno uključiti i opšte (isto za sve) pojačanje  $G_0$  izračunato kao aritmetička sredina ili medijana pojačanja svih korišćenih filtera. Ono bi bilo korisno u slučajevima kada većina filtera ima isto pojačanje (Slika 2.2) [1].



Slika 2.2: Kaskada filtera grafičkog ekvilizatora [1]

Funkcija prenosa ovako realizovanog grafičkog ekvilizatora bi bila sledeća:

$$H_{CGEQ}(z) = G_0 \prod_{m=1}^M H_m(z) \quad (8)$$

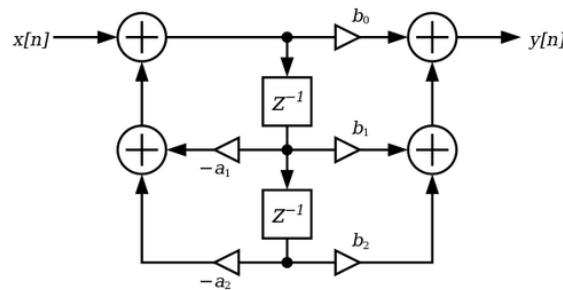
gde je  $H_m(z)$  filter m-tog opsega ( $m = 1, \dots, M$ ). Iz ove formule bi se moglo zaključiti da funkcija prenosa ekvilizatora ima iste polove i nule kao i svi korišćeni filteri zajedno. Idealno, svaki pojedinačni filter bi imao pojačanje  $G_m$  na svojoj centralnoj frekvenciji i  $\sqrt{G_m}$  na krajevima opsega. Međutim u praksi postoji interakcija filtera.

### 2.3.4 Bikvad filter

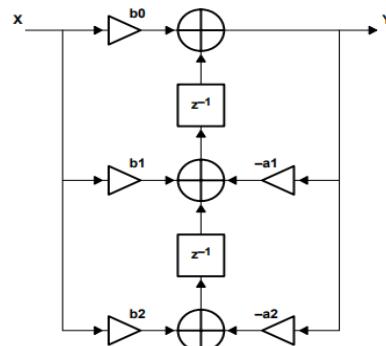
Grafički ekvilizator se najčešće realizuje kao skup bikvad filtera. Bikvad filter je IIR filter drugog reda, tj. sadrži dva pola i dve nule. Prenosna funkcija u Z domenu bikvad filtera izgleda ovako:

$$H(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2}) / (a_0 + a_1 z^{-1} + a_2 z^{-2}) \quad (9)$$

Zbog stabilnosti, umesto IIR filtera višeg reda, bikvad se koristi kao osnovna gradivna jedinica složenijih struktura, kao što je, između ostalog, i grafički ekvilizator. Može se realizovati u dve različite forme: direktnoj formi I i direktnoj formi II. Direktna forma I je pogodnija za implementaciju na procesorima sa aritmetikom u fiksnom zarezu, koji su osetljivi na prekoračenje, obzirom da poseduje samo jedan sabirač. Direktna forma II je adekvatnija za korišćenje kada su u pitanju procesori sa aritmetikom u pokretnom zarezu, posebno njena transponovana verzija, iz razloga što su ovakvi procesori precizniji pri sabiranju približnijih vrednosti, nego kada su razlike velike.



Slika 2.3: Direktna forma II



Slika 2.4: Transponovana direktna forma II

Diferencne jednačine transponovane direktne forme II su:

$$y[n] = b_0 * x[n] + w_1[n-1] \quad (10)$$

$$w_1[n] = b_1 * x[n] - a_1 * y[n] + w_2[n-1] \quad (11)$$

$$w_2[n] = b_2 * x[n] - a_2 * y[n] \quad (12)$$

## 2.4 Izbor ciljne hardverske platforme

Spektar uređaja koji koriste neki vid ekvalizacije audio signala je širok, a neki od njih su: kućni bioskopi, stereo uređaji u automobilima, aplikacije u telefonima i sl. U zavisnosti od tipa uređaja proističu i kriterijumi odabira namenskog sistema, koji je zadužen za samu ekvalizaciju. Kriterijumi odabira se mogu ticati cene proizvoda, potrošnje, zagrevanja, postojećih algoritama, mogućnosti integracije u širi sistem, brzine i sl. Digitalni signal procesori su decenijama bili najzastupljenija hardverska platforma u obradi audio signala. Međutim, u poslednje vreme sve više se koriste i platforme zasnovane na ARM procesorskom jezgru.

Najvažnije prednosti platforme zasnovane na ARM procesorskom jezgru su: niska cena, mala potrošnja, jednostavniji razvoj aplikacija.

Sa druge strane, digitalni signal procesori su mikroprocesori specijalizovani za računske operacije nad digitalnim signalima. DSP obezbeđuje veoma brze sekvene instrukcija, što je veoma važno kada su u pitanju aplikacije za obradu signala, koje obiluju matematičkim operacijama.

Ipak, digitalni signal procesori su specijalizovani za konkretnu aplikaciju koju izvršavaju. ARM procesorska jezgra to nisu, što ih čini značajno fleksibilnijima [3]. Takođe, usled niske potrošnje i cene, ARM procesorska jezgra su veoma poželjna u sistemima sa ograničenim resursima, kao što su telefoni, na primer. Sa druge strane, razvoj aplikacije koja će se odvijati na platformi zasnovanoj na ARM procesorskom jezgru je značajno olakšan činjenicom da se većim delom vrši u višem programskom jeziku. Iz svega navedenog, čini se da postoje opravdani razlozi da se kao ciljna platforma za ekvalizaciju spektra audio signala koristi platforma zasnovana na ARM procesorskom jezgru.

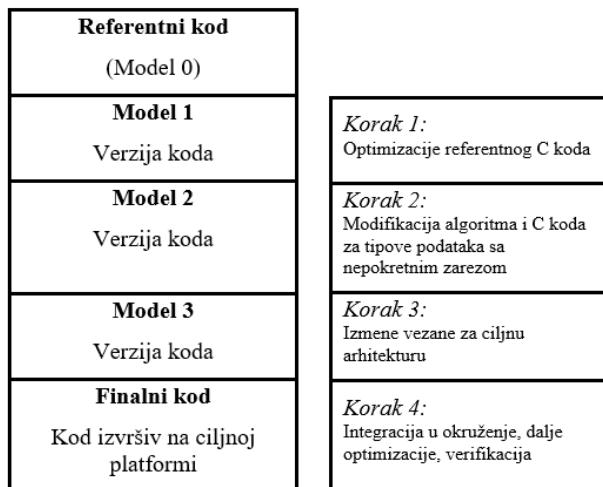
### 3. Koncept rešenja

Proces razvoja softverskih aplikacija je u velikoj meri određen samom ciljnom arhitekturom, čije karakteristike određuju skup mogućnosti programera [4]. Kako bi se ispunio zadatak ovde prezentovanog rada, tj. razvoj aplikacije GEQ za platformu zasnovanu na ARM procesorskom jezgru pri čemu se kao referentni model koristi aplikacija namenjena digitalnom signal procesoru, neophodno je uzeti u obzir specifičnosti razvoja aplikacija za oba tipa platforme.

#### 3.1 Razvoj aplikacije za digitalni signal procesor

Metodologija razvoja aplikacija za obradu audio signala na digitalnim signal procesorima propisuje korake, koji ubrzavaju i olakšavaju razvoj i održavanje koda.

Prvi korak je analiza ulaznih ispitnih vektora, zadatog algoritma i formiranje referentnog C koda. Rezultat ovog koraka je referentni kod, tj. Model 0 i skup izlaznih vektora, koji služe za proveru ispravnosti svake naredne faze. Model 1 uvodi funkcionalne optimizacije koda. Model 2 podrazumeva prilagođenje koda aritmetici ciljne arhitekture. Model 3 predstavlja potpuno prevodiv C kod za ciljnu platformu. Ukoliko su rezultati profilisanja zadovoljavajući, kod se povezuje sa programskim okruženjem ciljne platforme. Model 0, Model 1 i Model 2 se prevode opštenamenskim kompjlerom, dok se Model 3 prevodi kompjlerom za ciljnu platformu (Slika 3.1) [5].



Slika 3.1: Tok implementacije softvera na digitalnim signal procesorima sa aritmetikom u neprekidnom zarezu [5]

## 3.2 Razvoj aplikacije za platformu zasnovanu na ARM procesorskom jezgru

Razvoj softvera za ciljnu platformu NXP i.MX 8M Mini, usled njenih karakteristika, pre svega podržane aritmetike u pokretnom zarezu, odgovara u mnogome razvoju softvera za računare opšte namene. Naime, programiranje je moguće i bez znanja o specifičnostima i organizaciji resursa fizičke arhitekture. Pritom se koriste viši programski jezici, što doprinosi preglednosti i prenosivosti samog koda. Programski jezik C se u tu svrhu nameće kao dobar izbor usled rasprotranjenosti i pogodnosti za razvoj softvera [4].

## 3.3 Razvoj GEQ za platformu zasnovanu na ARM procesorskom jezgru

Proces razvoja modula za ekvalizaciju spektra audio signala se može opisati kroz tri faze.

Rezultat prve faze je konstruisan algoritam grafičkog ekvilizatora na osnovu dva referentna modela. Jedan matematički model je potrebno kreirati putem analize asemblerских instrukcija, koje stoje u osnovi primitive parametarskog ekvilizatora u grafičkom okruženju CLIDE integrisanog razvojnog okruženja. Drugi matematički model proističe iz implementacije parametarskog ekvilizatora za ciljni TMS320C6713 audio digitalni signal procesor proizvodjača Texas Instruments.

Druga faza se ogleda u kreiranju aplikacije grafičkog ekvilizatora u C programskom jeziku na računaru opšte namene, koji će biti referentni C model. U toku razvoja C referentnog modela, potrebno je pristupiti proveri ispravnosti programskih rešenja tako što bi se izlazni vektori modula poredili sa izlaznim vektorima primitive parametarskog ekvilizatora. Poređenje

rezultata se može vršiti na nekoliko načina: slušnim testovima, poređenjem datoteka na nivou bita, ali i poređenjem spektralnih slika [4].

Treća faza podrazumeva da se tako realizovani model, uz izvesne modifikacije radnog okvira ukoliko je to potrebno, pokreće u simulatoru NXP okruženja, te se njegovi izlazni vektori upoređuju sa izlazima dobijenim u prethodnoj fazi, putem već pomenutih testova.

## 4. Implementacija

Sledi opis detaljnih koraka u implementaciji modula za ekvalizaciju spektra audio signala po fazama razvoja definisanim u prethodnom poglavlju.

### 4.1 Faza 1 - analiza referentnih modela

Kao prvi referentni model za realizaciju grafičkog ekvilizatora je korišćena postojeća implementacija grafičkog ekvilizatora namenjena za Cirrus Logic CS498xx familiju digitalnih signal procesora (u daljem tekstu grafički referentni ekvilizator I, GREQ I).

Nakon upoznavanja sa načinom realizacije GREQ I putem analize asemblerских datoteka implementiranih u CLIDE integrисаном razvojном okruženju, zaključilo se sledeće:

- Kao osnovna struktorna jedinica ekvilizatora je korišćen bikvad filter
- Bikvad filteri su korišćeni u transponovanoj direktnoj formi II
- Ekvilizatori su implementirani kao kaskada osnovnih struktornih jedinica
- Način računanja koeficijenata bikvad filtera je sledeći:

$$w_c = 2\pi * F_c / F_s \quad (13)$$

$$soc_{64} = \sin(w_c) / 64 \quad (14)$$

$$coc = \cos(w_c) \quad (15)$$

$$w_3 = w_c * (\sqrt{Q^2/1024 + 1/4096} - 1/64) / (Q / 32) \quad (16)$$

$$Q_{b32} = \sin(w_3) * soc_{64} / (\cos(w_3) - coc) \quad (17)$$

$$A = 10^{g/20} \quad (18)$$

$$\text{Za } g < 0: Q_{b32} = Q_{b32} * A \quad (19)$$

$$den = Q_{b32} + soc_{64} \quad (20)$$

---

$c_{a1} = Q_{b32} * \text{coc} / \text{den}$	(21)	
$c_{a2} = (\text{soc}_{64} - Q_{b32}) / 2\text{den}$	(22)	
$\text{opa2} = 0.5 + c_{a2}$	(23)	<b>Legenda:</b>
$\text{gtoma2} = A * (0.5 + c_{a2})$	(24)	$F_s$ – frekvencija odabiranja
$b_0 = \text{opa2} + \text{gtoma2}$	(25)	$F_c$ – centralne frekvencije frekvencijskih opsega
$b_1 = -c_{a1} / 2$	(26)	$g$ - pojačanje izraženo u dB
$b_2 = \text{opa2} - \text{gtoma2}$	(27)	$A$ – linearno pojačanje
$a_1 = 2c_{a1}$	(28)	$Q$ – zakošenost
$a_2 = 2c_{a2}$	(29)	$b_0, b_1, b_2, a_0, a_1, a_2$ – koeficijenti bikvad filtera

U toku analize referentnog modela je primećeno da GREQ I nema odgovarajuće performanse. Razlog leži u činjenici da su centralne frekvencije svih opsega fiksirane na 1000 Hz, što praktično znači da bez obzira na broj opsega, kojih maksimalno može biti 31, suštinski postoji samo jedan frekvencijski opseg na koji korisnik može uticati.

Zbog toga se kao, referentni model koristio PREQ I (parametarski ekvilizator razvijen za Cirrus Logic CS498xx familiju digitalnih signal procesora), koji se pokazao kao bolji u kreiranju očekivanog izlaza, a po pitanju strukture i načina računanja bikvad filtera je identičan GREQ I aplikaciji.

Kao drugi referentni model je korišćena implementacija parametarskog ekvilizatora (u daljem tekstu PREQ II) na TMS320C6713 audio digitalnom signal procesoru proizvodjača Texas Instruments [6]. Ovaj ekvilizator je takođe implementiran kao kaskada rekurzivnog filtera drugog reda, tj. bikvad filtera u transponovanoj direktnoj formi II. Razlika u odnosu na PREQ I je u načinu računanja bikvad koeficijenata. U slučaju PREQ II ovi koeficijenti su računati na sledeći način:

$$A = 10g/40 \quad (30)$$

$$w = 2\pi * F_c / F \quad (31)$$

$$\sin = \sin(w) \quad (32)$$

$$\cos = \cos(w) \quad (33)$$

$$\alpha = \sin / 2Q \quad (34)$$

$$a_0 = 1 + \alpha/A \quad (35)$$

$$a_1 = -2\cos/\alpha \quad (36)$$

$$a_2 = (1 - \alpha/A)/a_0 \quad (37)$$

$$b_0 = (1 + \alpha A)/a_0 \quad (38)$$

$$b_1 = -2\cos/\alpha \quad (39)$$

$$b_2 = (1 - \alpha A)/a_0 \quad (40)$$

**Legenda:**

Fc – centralne frekvencije frekvencijskih opsega

g – pojačanje izraženo u dB

Q – zakošenost

A – linearno pojačanje

b0, b1, b2, a0, a1, a2 – koeficijenti bikvad filtera

## 4.2 Faza 2 – izrada C referentnog modela

Uzimajući u obzir sve prethodno izloženo, pristupilo se implementaciji grafičkog ekvilizatora u Visual Studio 2019 softverskom alatu u programskom jeziku C, ugledajući se na formirane algoritme, koji su integrisani u jedan pri čemu se daje korisniku izbor da li će koristiti način računanja bikvad koeficijenata po ugledu na PREQ I (u daljem tekstu GEQ I) ili po ugledu na PREQ II (u daljem tekstu GEQ II).

Kako su kao referentni programske modeli korišćeni parametarski ekvilizatori i ovde opisana implemetacija jeste suštinski parametarski ekvilizator. Međutim, testni slučajevi su kreirani tako da bi mogli da se porede sa nekim komercijalnim rešenjima grafičkog ekvilizatora, koji su nam dostupni.

Broj korišćenih frekvencijskih opsega je 31, što je maksimalan broj opsega u PREQ I modulu. Rešeno je, po ugledu na neke od komercijalnih rešenja implementacije grafičkog ekvilizatora (Audacity, Adobe Audition) da se koristi trećina oktave dizajn. S toga su centralne frekvencije opsega grafičkog ekvilizatora i izabrane po ugledu na Tabelu 2.1. Zakošenost je konstantna i iznosi 5.

Kako bi izlazi GEQ I i njemu referentnog PREQ I modela bili podložniji objektivnoj analizi prilikom međusobnog poređenja, postojala je težnja da se kreira referentni kod u Clide integrisanom razvojnog okruženju na osnovu dostupnih asemblerskih datoteka. U tome se i uspelo, kada je u pitanju pojedinačni bikvad filter, kao osnovna strukturna jedinica ekvilizatora. Usپoredbom izlaznih vektora novokreirane bikvad primitive u grafičkom okruženju Clide IDE-a sa izlazima pojedinačnog bikvad filtera realizovanom u programskom jeziku C u Visual Studiu 2019, došlo se do zaključka da postoje razlike, koje se mogu pripisati razlici u

podržanom tipu aritmetike (Cirrus Logic CS498xx familija digitalnih signal procesora podržava aritmetiku u fiksnom zarezu, dok računar opšte namene podržava aritmetiku u pokretnom zarezu). Kako je grafički ekvilizator zamišljen kao kaskada bikvad filtera, ova razlika, tj. greška bi se s povećanjem broja filtera i sama uvećavala.

Izlazni vektori aplikacije u slučaju kada se koristi način računanja bikvad koeficijenata GEQ II su poređeni sa izlazima grafičkog ekvilizatora programskega paketa Adobe Audition, kao jednog od komercijalnih rešenja.

Efikasnost implementacije, koja se odvijala postupno, se utvrđivala poređenjem dobijenih izlaza sa referentnim uporednom vizuelnom analizom frekvencijskog spectra u Adobe Audition programskom paketu.

### 4.3 Faza 3

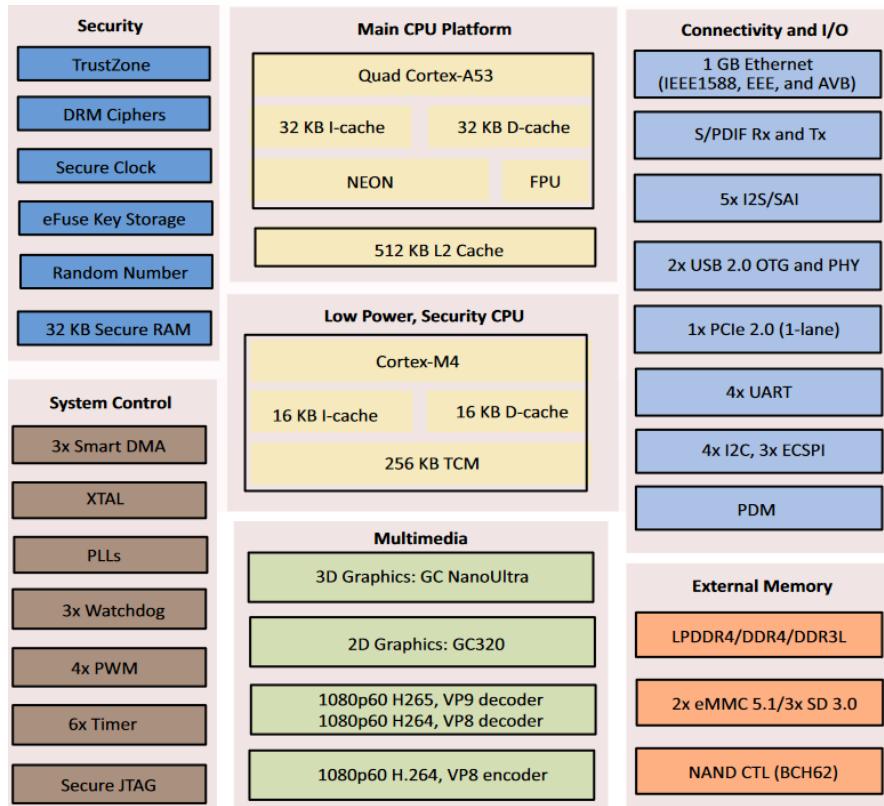
U trećoj fazi razvoja razvijena aplikacija je prilagođena postojećem simulatorskom projektu uz minimalne modifikacije istog, sa ciljem verifikacije rezultata modula namenjenog NXP platformi. Testiranje, tj. poređenje izlaznih sa referentnim vektorima kreiranim u prethodnoj fazi, vrši se na osnovu bit-identičnog testa.

### 4.4 Karakteristike ciljne platforme

Ciljna platforma za koju je razvijana aplikacija GEQ tokom izrade ovde prezentovanog rada je i.MX 8M Mini procesor. Ova platforma je, između ostalog, predviđena za razvoj aplikacija surround sistema, sound barova, audio-video prijemnika i sl.

Sadrži četiri ARM Cortex-A53 jezgra i jedno Cortex-M4 jezgro za heterogenu višejezgarsku obradu. Cortex-M4 jezgro je brzine 400MHz, dok Cortex-A53 jezgra rade na brzinama i do 1.8GHz. Takođe, nudi LPDDR4 memorijski interfejs za visoku performansu Sadrži 20 I2S kanala.

Podržava aritmetiku u pokretnom zarezu [7].



Slika 4.1: i.MX 8M Mini sistem blok dijagram [7]

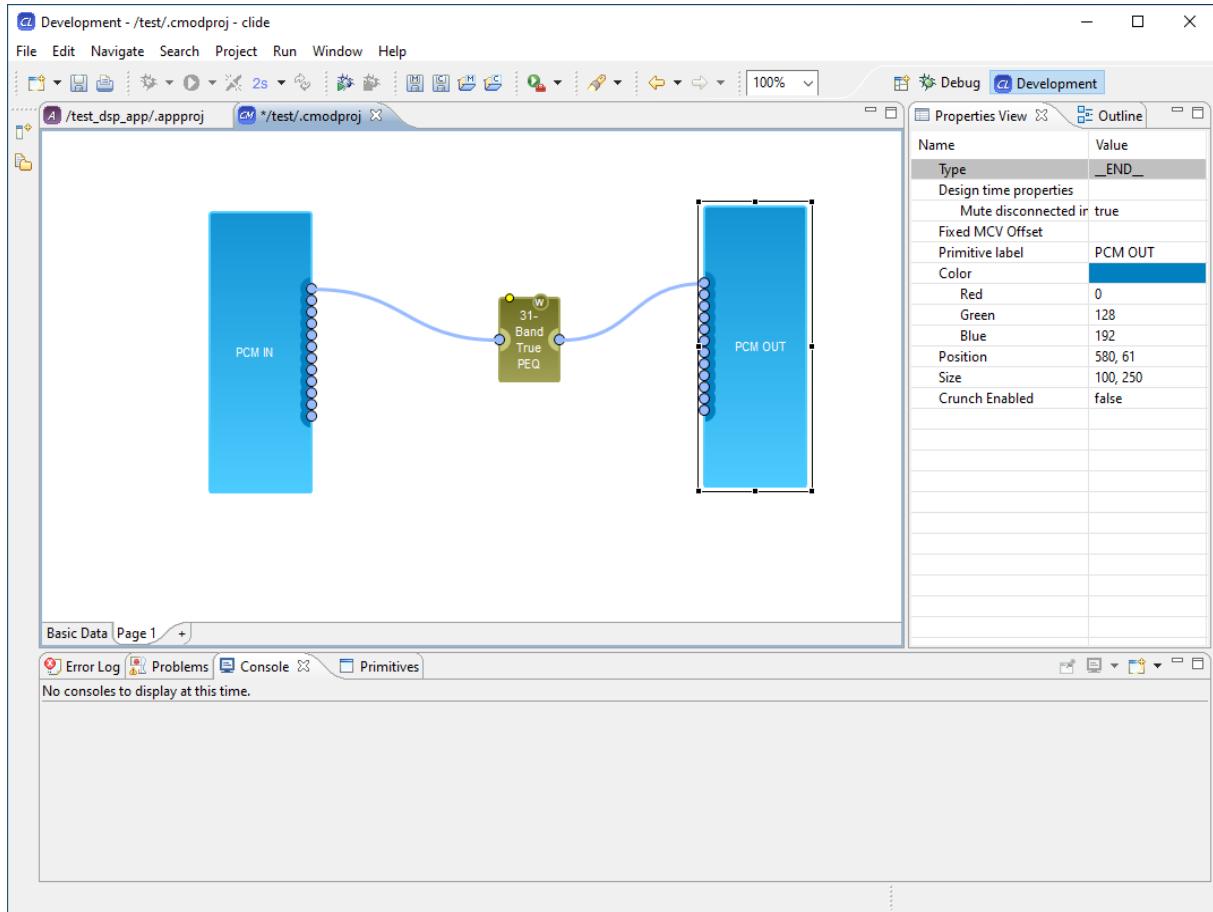
## 4.5 Razvojno okruženje

U zavisnosti od aktuelne faze razvoja koristila su se različita razvojna okruženja.

### 4.5.1 Faza 1

U fazi 1 razvoja modula za ekvalizaciju spektra audio signala koristilo se CLIDE integrisano razvojno okruženje (Slika 4.2). U njemu je vršena analiza postojećih realizacija bikvad filtera, grafičkog i parametarskog ekvilizatora. Njime su kreirani i izlazni vektori koji su poslužili kao referentni u fazi 2. Takođe, u njemu je formirana i nova bikvad primitiva sa ciljem preciznijeg praćenja vrednosti odgovarajućih parametara koji bi se upoređivali sa vrednostima kreiranog C referentnog modela u narednoj fazi.

CLIDE nudi alate za kontrolisano izvršavanje programa uz uvid u stanje memorije i registara. Daje mogućnost prevođenja aplikacije i podršku za spuštanje DSP aplikacije na razvojnu ploču. Poseduje i grafičko okruženje gde se različite primitive u vidu grafičkih blokova mogu proizvoljno kombinovati u složenije aplikacije.

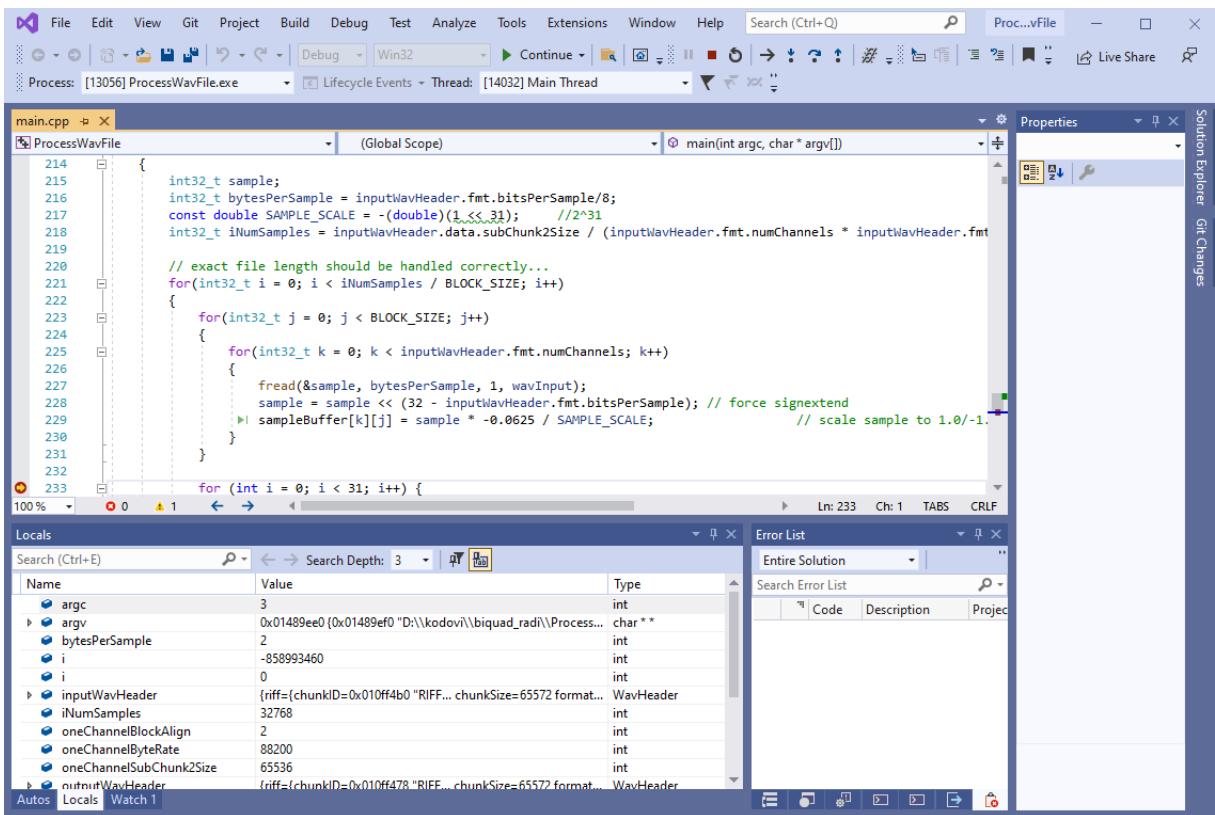


Slika 4.2: CLIDE integrisano razvojno okruženje

U ovoj fazi je vršeno testiranje novokreirane bikvad primitive sa već postojećom upoređivanjem njihovih izlaznih vekتورа na bitskom i spektralnom nivou.

#### 4.5.2 Faza 2

Visual Studio je integrisano razvojno okruženje u kome je moguće uređivati, razvijati kod kao i otkrivati i uklanjati greške u istom. Radi se o softverskom paketu bogatom funkcijama pogodnim za mnoge aspekte razvoja softvera. Pored standardnog uređivača i programa za kontrolisano izvršavanje koje vićina IDE-a nudi, Visual Studio uključuje i kompajlere, alate za generisanje koda, alate za grafički dizajn i mnoge druge koji alakšavaju proces razvoja softvera.



Slika 4.3: Visual Studio - kontrolisano izvršavanje programa

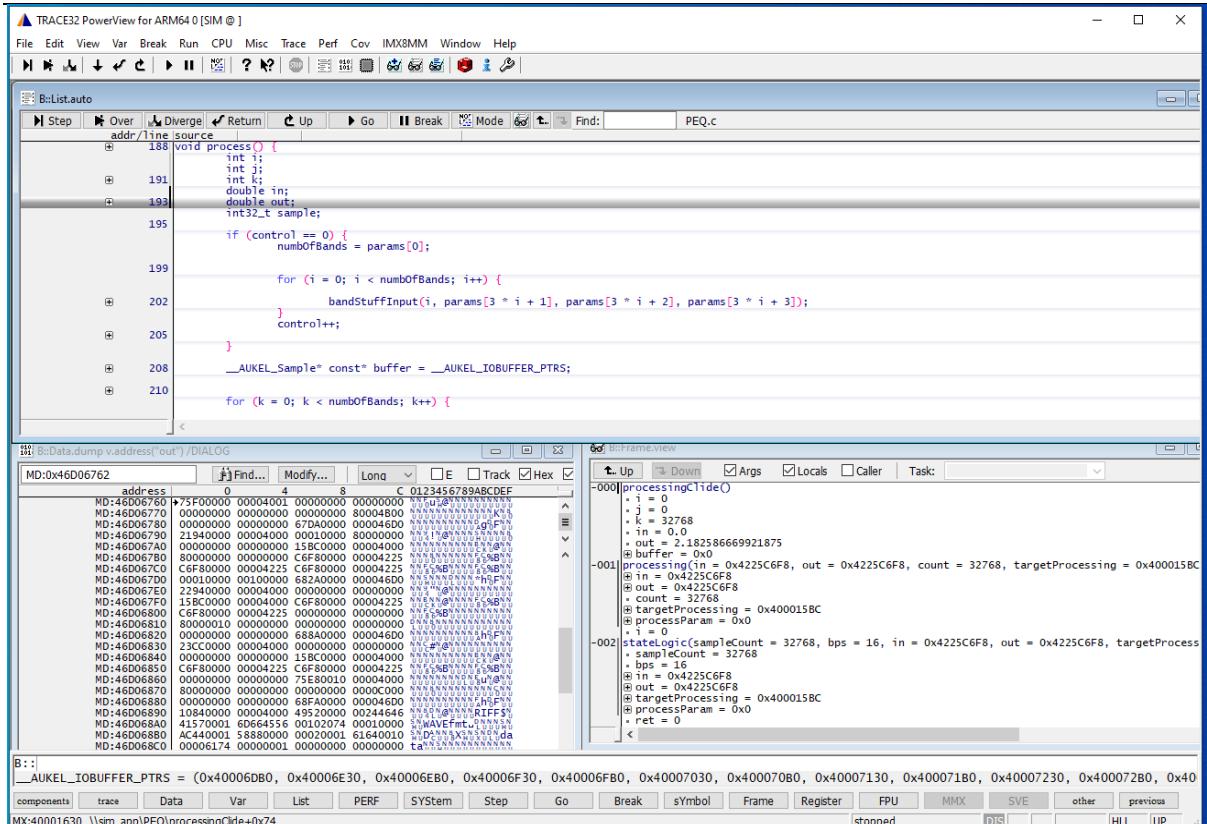
U fazi 2 je u svrhu poređenja referentnih i izlaznih vektora koršćen Audacity softverski paket za analizu spektralnih slika. Takođe se koristilo i poređenje na bitskom nivou putem PCMCompare alata.

Izlazni vektori kreirani u fazi 2 su poslužili kao referentni u narednoj fazi razvoja.

### 4.5.3 Faza 3

Kako bi se omogućilo kontrolisano izvršavanje simulatorskog projekta korišćen je PowerDebug USB3 hardverski alat, koji, povezan na računar opšte namene, simulira preko 100 različitih arhitektura, a između ostalih i ARM koji je od interesa za ovde prezentovan rad.

Trace32 Simulator pokreće izvršnu datoteku od interesa koristeći skript programski jezik Practice. Pritom omogućava grafički prikaz memorije, vrednosti lokalnih promenljivih, parametre funkcija tokom izvršavanja pojedinačnih instrukcija pružajući podršku za sve vodeće operativne sisteme za rad u realnom vremenu.



Slika 4.4: Trace32 simulator - kontrolisano izvršavanje programa

## 5. Programsko rešenje

Sledi opis realizovanih programske funkcije i struktura pomoću kojih je postignuta tražena funkcionalnost modula.

### 5.1 Struktura history

Struktura history sadrži dve vrednosti,  $w_1$  i  $w_2$ , koje predstavljaju vrednosti diferencnih jednačina bikvad filtera ((11), (12)) izračunatih u funkciji *process*. Njihove početne vrednosti su 0.

### 5.2 Struktura biquad

Ovo je struktura, koja sadrži osnovne informacije o konkretnom bikvad filteru: koeficijente, pojačanje izraženo u dB, centralnu frekvenciju, zakošenost, kao i niz *history* struktura, koji ima onoliko članova koliko ima kanala u ulaznom signalu.

### 5.3 Funkcija init

Ovo je funkcija, koja na osnovu ulaznih parametara popunjava strukturu biquad odgovarajućim vrednostima. U tu svrhu poziva funkciju *coeffsGEQI*, tj. *coeffsGEQII* u zavisnosti od toga po ugledu na koji referentni model želimo da računamo koeficijente bikvada.

### 5.4 Funkcija coeffsGEQI

Ovo je funkcija koja za konkretni bikvad filter u zavisnosti od parametara frekvencijskog opsega za koji je zadužen, tj. centralne frekvencije, pojačanja i zakošenosti, računa koeficijente po ugledu na GREQ I ((13) – (29)).

## 5.5 Funkcija coeffsGEQII

Funkcija, koja računa koeficijente konkretnog bikvad filtera po ugledu na način računanja koeficijenata u PREQ II modelu ((30) – (40)).

## 5.6 Funkcija process

Ovo je funkcija, koja putem diferencnih jednačina ((10) – (12)) kreira izlazni signal primenjući sve postojeće bikvad filtere, nad svim kanalima.

## 6. Ispitivanje predloženog rešenja

U ovom poglavlju su opisani metodologija i rezultati testiranja razvijene aplikacije za ekvalizaciju spektra audio signala namenjenu izvršavanju na i.MX 8M Mini platformi zasnovanoj na ARM procesorskom jezgru kompanije NXP. Testiranje se vršilo u skladu fazama razvoja aplikacije. Od interesa za procenu efikasnosti ovde perezentovanog rešenja su testiranja u fazama izrade 2 i 3.

### 6.1 Faza 2

U fazi 2 izrade modula za ekvalizaciju spektra od interesa su sledeća tri ispitivanja:

- ispitivanje podudaranja izlaznih vektora aplikacije GEQ I sa referentnim vektorima kreiranim CLIDE primitivom parametarskog ekvilizatora
- ispitivanje efikasnosti aplikacije GEQ II kroz poređenje izlaznih vektora sa izlaznim vektorima nekog od komercijalnih rešenja grafičkog ekvilizatora
- usporedba izlaznih vektora aplikacija GEQ I i GEQ II

#### 6.1.1 Analiza rezultata dobijenih primenom GEQ I

Izlazni signali iz modula za realizaciju GEQ I (u daljem tekstu izlazni signali) su se poredili sa referentnim koji su formirani primenom PREQ I aplikacije nad unapred utvrđenim ulaznim signalima: white noise, pink noise, square.

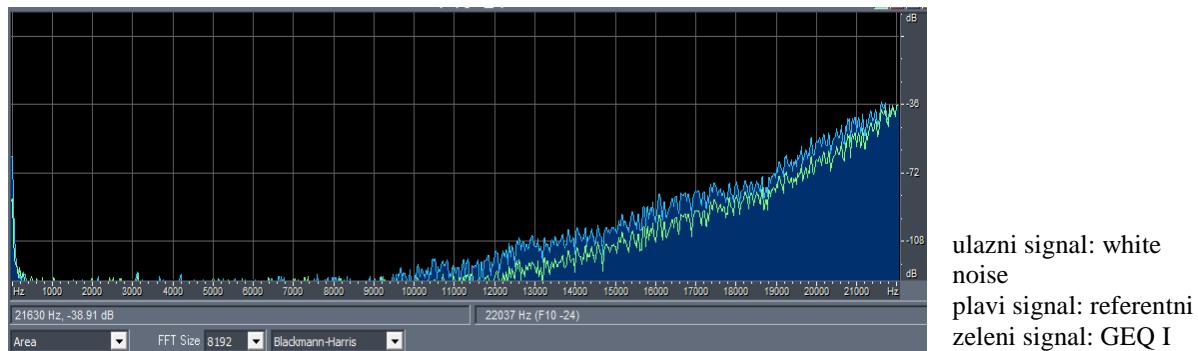
Kreirano je sledećih 9 testnih situacija:

- Highpass –  $f_c = 800\text{Hz}$ , Gain = -24dB
- Lowpass –  $f_c = 800\text{Hz}$ , Gain = -24dB
- Killed – ceo signal smanjen za -24dB
- AllUp – ceo signal pojačan za 24dB
- Bandpass1 – frekvencije manje od 1000Hz I veće od 5000Hz su pojačane za 24 dB
- Bandpass2 – frekvencije manje od 1000Hz I veće od 5000 Hz su smanjene za -24 dB

- Bandpass3 – frekvencije od 1000 Hz do 5000Hz su smanjene za -24dB
- Bandpass4 – frekvencije od 1000Hz do 5000Hz su pojačane za 24dB
- UpDown – naizmenično pojačavanje za 24dB I smanjivanje za -24dB frekvencijskih opsega

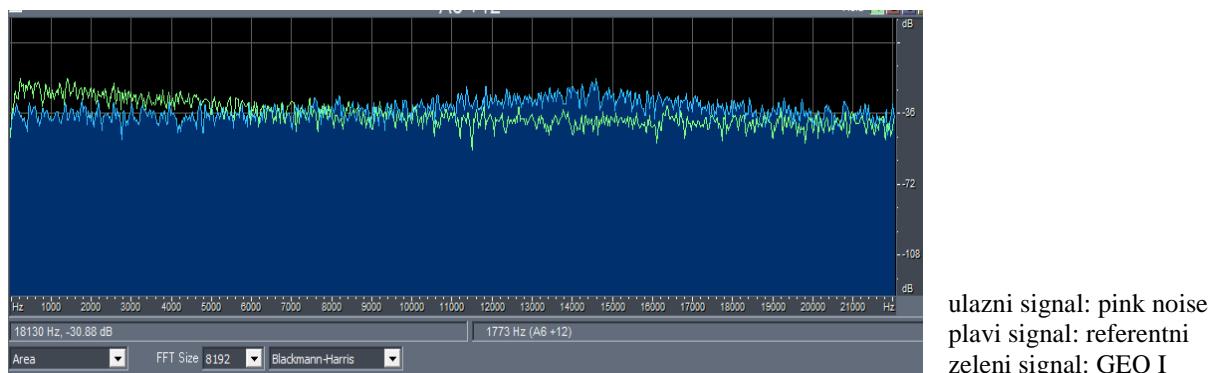
Poređenje se vršilo vizuelnom analizom frekvencijskog spektra referentnih i izlaznih signala u programskom paketu Adobe Audition.

Zaključak je da, kada je u pitanju smanjenje amplitude nad celim signalom, postoji tendencija da GEQ I algoritam naspram referentnog uspešnije potiskuje signal u domenu viših frekvencija (kao što je ilustrovano Slikom 6.1) pri čemu je oblik anvelopa vrlo sličan.



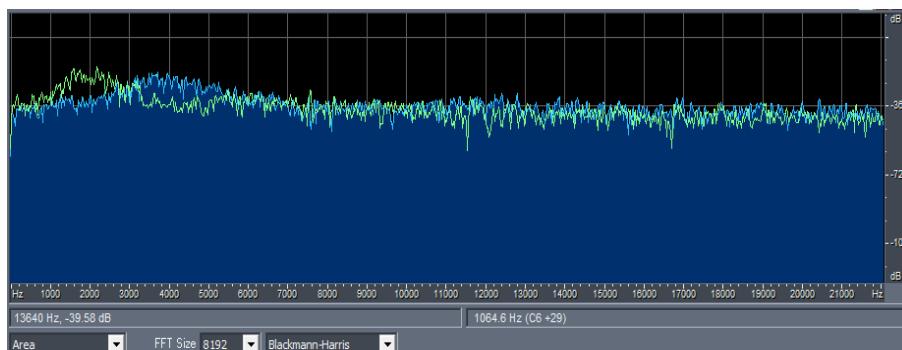
Slika 6.1: test Killed, GEQ I

Kada je u pitanju pojačavanje amplitude celog signala, u ovoj situaciji algoritmi pokazuju najveće neslaganje u izlaznim signalima u odnosu na sve ostale testove. Vizuelnom analizom spektra izlaznih signala se zaključuje da razlike postoje ne samo po pitanju amplitude izlaznih signala, već i po pitanju samog oblika anvelope. Moglo bi se reći da referentni model daje neočekivan porast amplitude na višim frekvencijama. U prilog tome govori Slika 6.2.



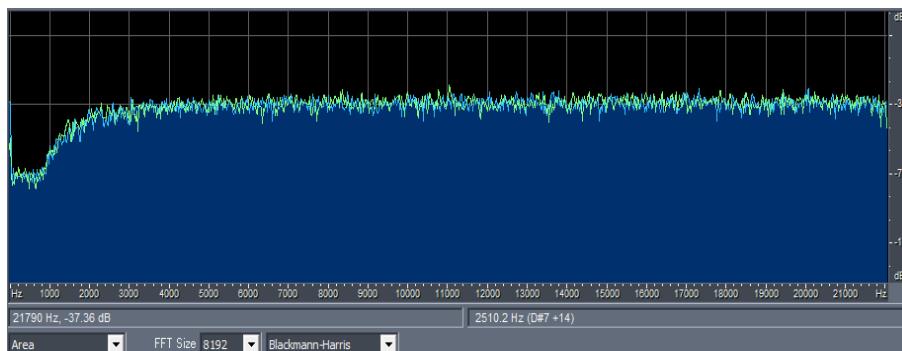
Slika 6.2: test AllUp, GEQ I

Slični rezultati se dobijaju i kada je u pitanju pojačanje izolovanog frekvencijskog opsega. Postoje nepoklapanja u poziciji lokalnog maksimuma u spektru signala koji se porede (Slika 6.3).



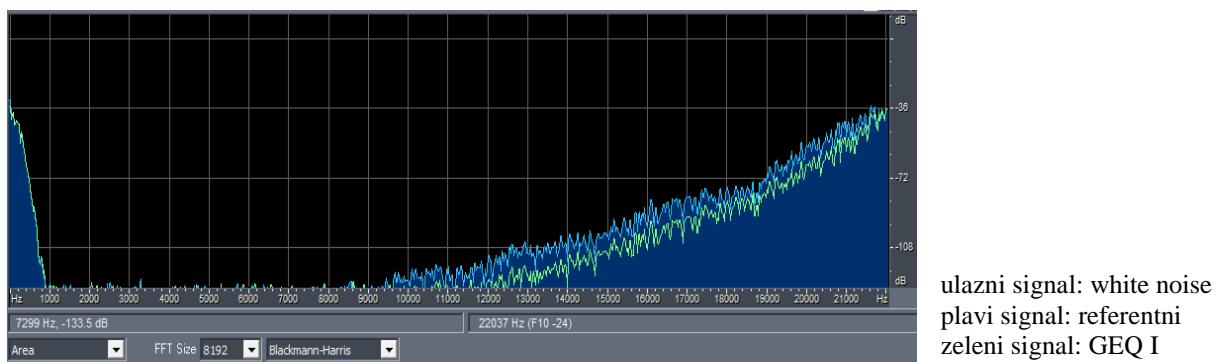
Slika 6.3: test BP4, GEQ I

Ako posmatramo propuštanje signala, bilo da je u pitanju highpass ili lowpass test, primećuje se da je postoji slaganje referentnog i izlaznog signala.



Slika 6.4: test Highpass, GEQ I

Vizuelna analiza spektra izlaznog i referentnog signala u testu Lowpass upućuje da je potiskivanje viših frekvencija uspešnije u slučaju algoritma PEQ I.



Slika 6.5: test Lowpass, GEQ I

Rezultati ostalih testova nisu detaljno analizirani iz razloga što prate već gore utvrđene zakonitosti:

- neslaganje izlaza dva modela je najveća u frekvencijskim opsezima gde se vrši povećanje amplitude
- neslaganje izlaza dva modela pri smanjivanju amplitude je posledica boljeg potiskivanja signala od strane algoritma GEQ I
- propuštanje signala ne proizvodi veliki broj razlika među izlaznim signalima dva algoritma

Takođe, primećuje se da, realizovani i referentni algoritam daju najveće razlike u izlazima kada se kao ulazni signal koristi white noise u gotovo svim testnim situacijama, što dovodi do zaključka da je realizovani algoritam osetljiv na tip ulaza.

### 6.1.2 Analiza rezultata dobijenih primenom GEQ II

Referentni signali sa kojima su se poredili izlazi modula za realizaciju GEQ II su formirani korišćenjem grafičkog ekvilizatora programskega paketa Adobe Audition nad unapred utvrđenim ulaznim signalima: white noise, pink noise, square. Provera stepena poklapanja referentnog sa izlaznim signalom se vršila alatom za analizu frekvencija Adobe Audition programskega paketa.

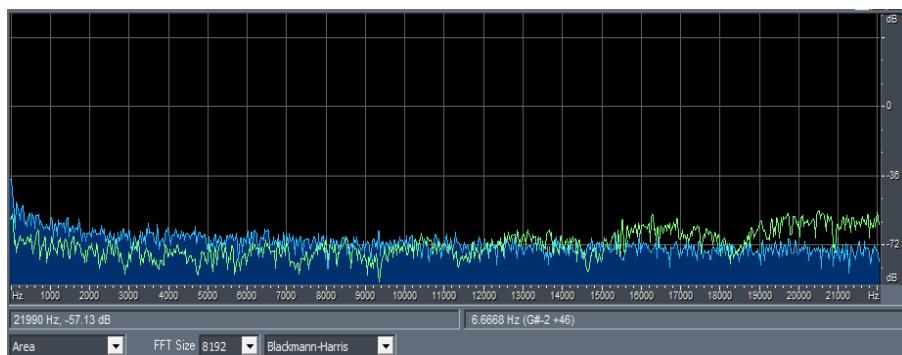
Testiranje je sprovedeno nad tri ulazna signala u sledećih 9 testnih konstelacija:

- Highpass – fc = 800Hz, Gain = -18dB
- Lowpass - fc = 800Hz, Gain = -18dB
- Killed – ceo signal smanjen za -18dB
- AllUp – ceo signal pojačan za 18dB
- Bandpass1 – frekvencije manje od 1000Hz I veće od 5000Hz su pojačane za 18 dB
- Bandpass2 – frekvencije manje od 1000Hz I veće od 5000 Hz su smanjene za -18 dB
- Bandpass3 – frekvencije od 1000 Hz do 5000Hz su smanjene za -18dB

- Bandpass4 – frekvencije od 1000Hz do 5000Hz su pojačane za 18dB
- UpDown - naizmenično pojačavanje za 18db I smanjivanje za -18dB frekvencijskih opsega

Sledi analiza rezultata sprovedenih testova.

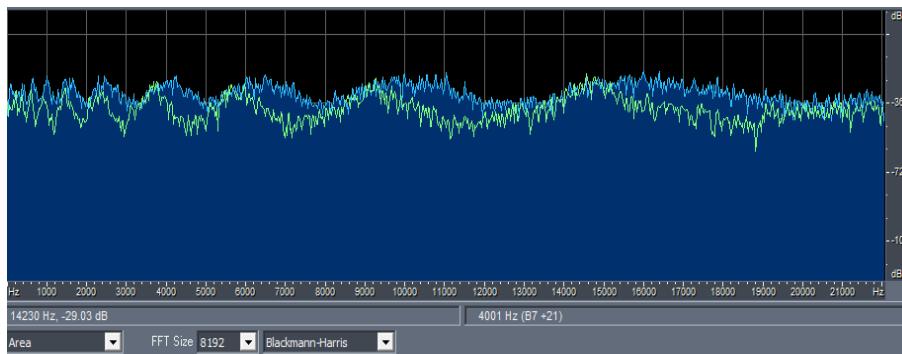
Prilikom smanjivanja jačine audio signala rezultati svih testnih slučajeva imaju zajedničkih karakteristika. Naime, u domenu nižih frekvencija, izlazni signal je smanjen više nego referentni. Međutim, postoji tendencija da se signal, kako frekvencije rastu, izjednačava po jačini sa ulaznim signalom. Na taj način, utišavanje signala sa rastom frekvencija postaje neuspešnije. Ovakav zaključak potrkepljuje Slika 6.6.



ulazni signal: pink noise  
plavi signal: referentni  
zeleni signal: GEQ II

Slika 6.6: test killedAll, GEQ II

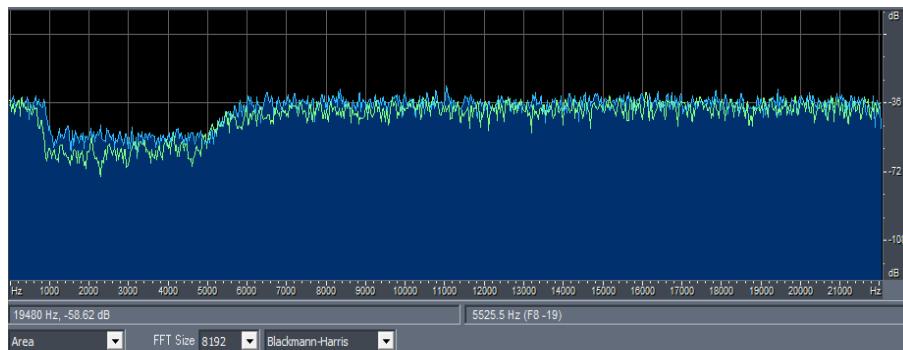
Sa druge strane, u situaciji kada naizmenično pojačavamo i smanjujemo frekvencijske opsege signala, može se primetiti da postoji pomeranje pikova u odnosu na referentni izlaz u istoj testnoj situaciji, što je posebno izraženo u višim frekvencijama. Lokalni minimumi signala propuštenih kroz GEQ II su niži od minimuma referentnih signala, dok su maksimumi amplitude oba signala približni, što Slika 6.7 i ilustruje.



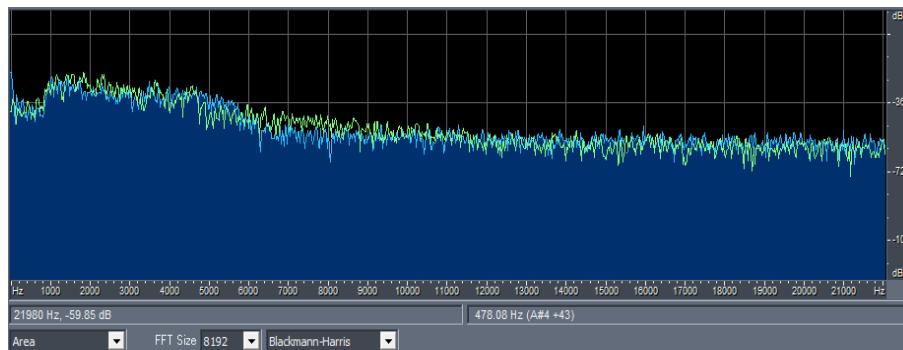
ulazni signal: white noise  
plavi signal: referentni  
zeleni signal: GEQ II

Slika 6.7: test UpDown, GEQ II

Kada se jedan izolovani frekvencijski opseg smanji (Slika 6.8) ili pojača (Slika 6.9), tada su poklapanja u smislu oblika anvelope i amplitude signala propuštenog kroz GEQ II sa referentnim izlazom veća.

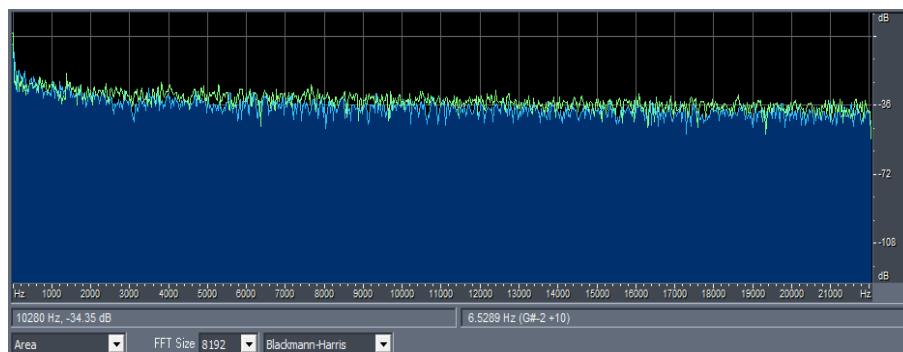


Slika 6.8: test BP3, GEQ II



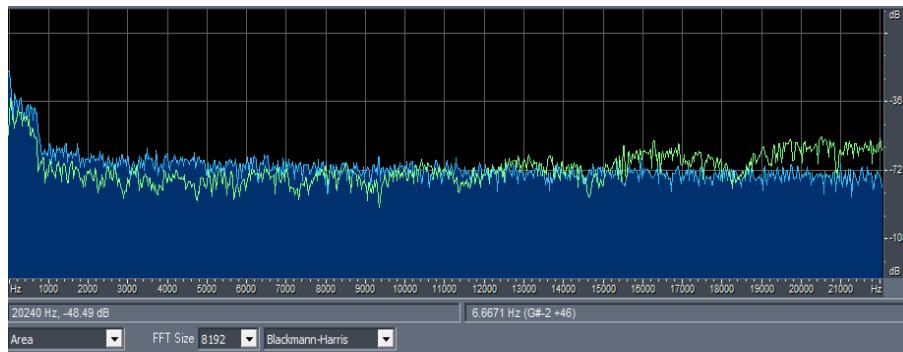
Slika 6.9: test BP4, GEQ II

Odstupanja izlaznih signala iz GEQ II u odnosu na referentne signale su mala kada je u pitanju pojačavanje ukupnog signala. Ovakav zaključak potkrepljuje Slika 6.10.



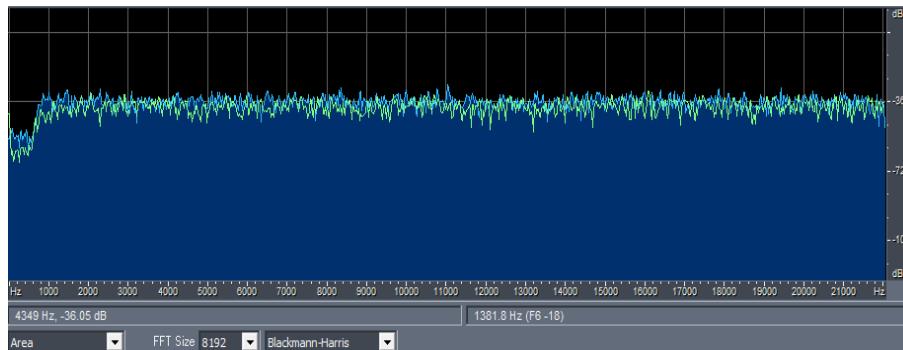
Slika 6.10: test AllUp, GEQ II

Kada je u pitanju propuštanje frekvencija, tu se može zaključiti sledeće: u propusnim opsezima se izlazni signali oba modela u značajnoj meri podudaraju. Sa druge strane, u opsezima koji se smanjuju važe iste, već utvrđene, ponašajne karakteristike realizovanog GEQ II filtera: sa rastom frekvencija se smanjuje uspešnost potiskivanja. Slika 6.11 i Slika 6.12 ilustruju prethodno navedeno.



ulazni signal: pink noise  
plavi signal: referentni  
zeleni signal: GEQ II

Slika 6.11: test Lowpass, GEQ II



ulazni signal: white noise  
plavi signal: referentni  
zeleni signal: GEQ II

Slika 6.12: test Highpass, GEQ II

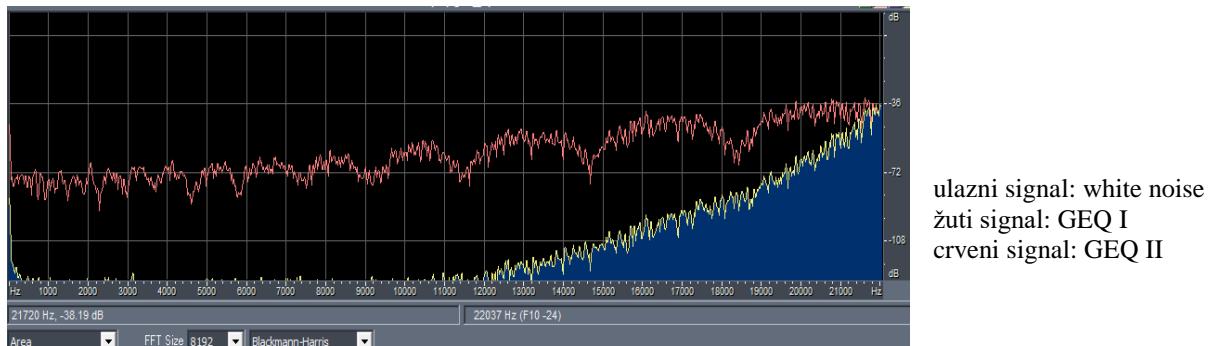
Sve ove razlike su, može biti, posledica različite implementacije grafičkog ekvilizatora. Nemamo uvid u to koji je dizajn korišćen prilikom konstruisanja grafičkog ekvilizatora u programskom paketu Adobe Audition: paralelni ili kaskadni, kao ni koliki je Q korišćen. Sve to može doprineti razlikama u izlaznim signalima.

Ipak, opšti utisak je da je stepen podudaranja izlaznog signala sa referentnim u konkretnoj testnoj situaciji u pogledu amplitude i oblika envelope konstanta samog realizovanog algoritma, te da ne zavisi od karakteristika korišćenih ulaznih signala.

### 6.1.3 Poređenje rezultata dobijenih primenom GEQ I i GEQ II algoritma

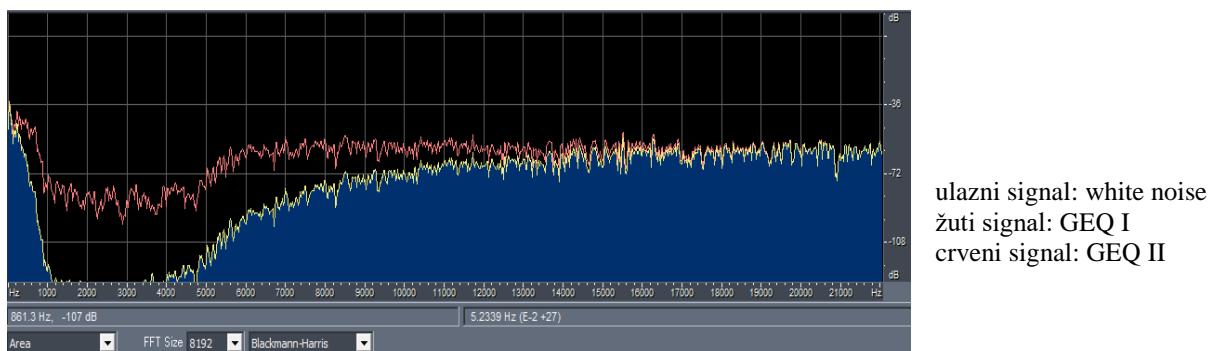
Poredeći dva konstruisana algoritma, GEQ I i GEQ II, mogu se uočiti brojne razlike u performansama ova dva modela grafičkog ekvilizatora. Testne situacije su iste kao i za analizu rezultata modela GEQ I.

Kada je u pitanju potiskivanje signala, uočljivo je da je u tome definitivno uspešniji GEQ I algoritam. Frekvencije bivaju potisnute i za 30 dB više. O tome svedoči Slika 6.13.



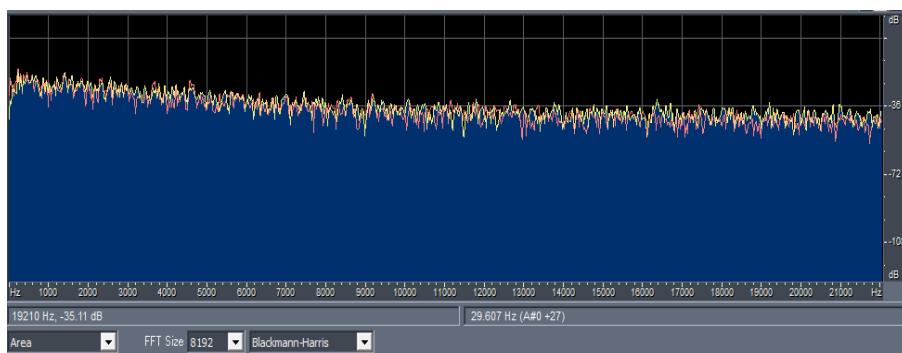
Slika 6.13: test Highpass

Prilikom smanjivanja audio signala u izolovanom frekvencijskom opsegu primećuje se da algoritam GEQ II slabije potiskuje odabrani opseg frekvencija, ali zato, s druge strane ima mnogo oštiju zakošenost (Slika 6.14).



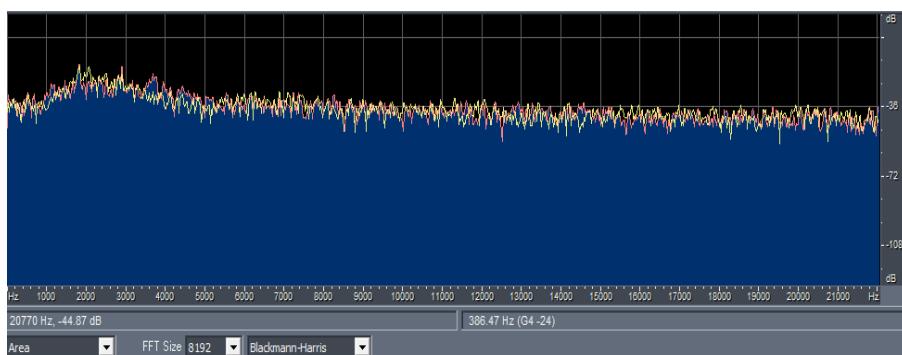
Slika 6.14: test Highpass

Performanse ova dva algoritma su ujednačena kada je u pitanju pojačanje signala (Slika 6.15).



Slika 6.15: test AllUp

Kada je reč o pojačavanju određenog frekvencijskog opsega izlazi dva algoritma se u pogledu anvelope podudaraju u većem delu spektra, što ilustruje Slika 6.16.



Slika 6.16: test BP4

U slučaju naizmenične primene pojačanja i smanjenja amplitude susednih frekvencijskih opsega, može se uočiti da GEQ II algoritam daje anvelopu izraženijih pikova sa oštrijom zakošenosti (Slika 6.17).



Slika 6.17: test UpDown

Prethodno opisane razlike dve različite implementacije grafičkog ekvilizatora se i subjektivno mogu osetiti prilikom slušanja muzike obrađene ovim algoritmima.

## 6.2 Faza 3 – verifikacija rezultata u simulatoru

S obzirom da i ciljna platforma i računar opšte namene na kome je razvijan modul ekvilizatora podržavaju aritmetiku u pokretnom zarezu, pretpostavljeno je da se izlazi pokrenutih modula na razvojnoj mašini i u simulatorskom okruženju neće razlikovati na nivou bita. To je i potvrđeno primenom alata PCMCompare, ručnog bit-identičnog testa, koji poredi dve datoteke na nivou bita.

Testiranje je vršeno nad ulaznim signalima pink noise i white noise, a korišćene su sledeće testne situacije:

- Highpass – fc = 800Hz, Gain = -24dB
- Lowpass – fc = 800Hz, Gain = -24dB
- Highpass – fc = 800Hz, Gain = -18dB
- Lowpass - fc = 800Hz, Gain = -18dB

U slučaju oba načina računanja bikvad koeficijenata, GEQ I i GEQ II, postoji apsolutno poklapanje datoteka formiranih na računaru opšte namene i u simulatorskom okruženju. Jedan od rezultata testova je ilustrovan Slikom 6.18.

```
PS D:\test_scripts\tools> .\PCMCompare_modded.exe D:\test_out\out_peq2_PC_43.wav D:\test_ref\output_31Biquad_LP.wav
No differences encountered!
No differences encountered!
```

Slika 6.18: Ručno bit poređenje izlaza GEQ I i simulatorskog projekta

## 7. Zaključak

U ovde prezentovanom radu je realizovan modul za ekvalizaciju audio signala za ciljnu platformu zasnovanu na ARM Cortex-A53 i Cortex-M4 procesorkim jezgrima i.MX 8M Mini proizvođača NXP. Ovaj zadatak je ispunjen kroz implementaciju aplikacije parametarskog ekvilizatora, koja se odvijala kroz nekoliko faza.

Najpre je analiziran referentni model aplikacije grafičkog ekvilizatora implementiran u asemblerском језику за Cirrus Logic CS498xx familiju digitalnih signal procesora. Nakon što se uvidelo da ovaj polazni model ne daje очekivane izlazne vektore, kao referenca koristili su se izlazni vektori parametarskog ekvilizatora realizovanog za istu DSP platformu. Grafički i parametarski ekvilizator su strukturno jednaki. Razlika je u tome što su parametri centralne frekvencije i zakošenosti kod grafičkog ekvilizatora fiksni, dok kod parametarskog ekvilizatora njihovu vrednost određuje korisnik. Zbog toga je prilikom kreiranja referentnih signala korišćen parametarski ekvilizator, ali sa unetim parametrima jedna trećina oktave dizajna grafičkog ekvilizatora.

Zatim se po ugledu na referentni algoritam na računaru opšte namene u programskom jeziku C razvila aplikacija za parametarski ekvilizator. Izlazni signali su poređeni sa referentnim signalima kreiranim u prethodnoj fazi. Primećeno je da postoji bitska razlika među izlaznim datotekama kao posledica razlike u podržanim tipovima aritmetike. Uporednom vizuelnom analizom spektra referentnog i izlaznog signala je utvrđeno da se isti ne podudaraju u pogledu oblika anvelope u svim testnim situacijama.

Pored gore navedenog, implementiran je još jedan algoritam ekvilizatora po ugledu na parametarski ekvilizator razvijen za ciljnu platformu TMS320C6000<sup>TM</sup> DSP proizvođača Texas

---

Instruments. Kao referentni je korišćen grafički ekvilizator iz programskog paketa Adobe Audition.

Kao krajnji rezultat dobijena je aplikacija za parametarski ekvilizator sa dva moguća načina računanja koeficijenata bikvad filtera, kao osnovne gradivne jedinice ekvilizatora.

Naposletku, kreiran je simulatorski projekat kao verifikacija rezultata. Pokazalo se, kako je bilo i očekivano, da se izlazi ovog projekta u potpunosti podudaraju sa izlazima aplikacije kreirane na računaru opšte namene, što je utvrđeno objektivnim bit-identičnim testovima.

Dalja istraživanja bi mogla ići u smeru poboljšanja postojećih algoritma ovde prezentovanog rešenja u cilju poboljšanja efikasnosti, ali i brzine. Moguće modifikacije algoritma bi se mogle ticati kobilovanja kaskadne i paralelne organizacije bikvada, a u svrhu smanjenja preklapanja frekvencijskih opsega. Takođe, potrebno je vršiti dodatna poređenja rešenja zasnovanim na digitalnom signal procesoru, sa jedne, i platformi koja sadrži ARM procesorska jezgra, sa druge strane, u smislu performansi, brzine izvršavanja tipičnih DSP operacija, pristupa memoriji, kontrolisanom izvršavanju u realnom vremenu itd.

## 8. Literatura

- [1] Vesa Valimaki, Joshua D. Reiss: “All about audio Equalization: Solutions and Frontiers”, Applied Sciences, 2016
- [2] Juho Liski, Vesa Valimaki, “*The quest for the best graphic equalizer*”, Proceedings of the 20<sup>th</sup> International Conference on Digital Audio Effects, 2017
- [3] Robert Oshana: “*DSP for embedded and real-time systems*”, Elsevier Inc, 2012
- [4] Kovačević dr Jelena, Bokan Dejan: “*Arhitekture i algoritmi digitalnih signal procesora – Zbirka zadataka i laboratorijski priručnik*”, FTN Izdavaštvo, 2016
- [5] Đukić Miodrag, Četić Nenad, Kovčević Jelena, Popović Miroslav: “*A C compiler based methodology for implementing audio DSP applications on a class of embedded systems*”, IEEE International Symposium on Consumer Electronics, 2008
- [6] Remi Payan: “*Parametric Equalization on TMS320C6000 DSP*”, Texas Instruments Inc, 2002
- [7] “*i.MX 8M Mini applications Processor Datasheet for consumer products*”, NXP, 2020