



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: Александар Мартиновић

Број индекса: РА4-2015

Тема рада: Имплементација модула за рендеровање 3D аудио слике на DSP платформи по MPEG-H стандарду

Ментор рада: доц. др Јелена Ковачевић

Нови Сад, 2020.



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:	Александар Мартиновић		
Ментор, МН:	доц. др Јелена Ковачевић		
Наслов рада, НР:	Имплементација модула за рендеровање 3D аудио слике на DSP платформи по MPEG-H стандарду		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2020.		
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)	8/40/15/4/18/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	3D звучна слика, HOA, DSP, обрада звука, рендеровање, MPEG-H		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У оквиру овог рада урађена је имплементација HOA декодера на Cirrus Logic DSP платформи. Имплементација обухвата анализу референтног C++ кода који је написан у складу са MPEG-H стандардом и по узору на њега, писање асемблерског кода за циљну DSP платформу.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	доц. др Небојша Пјевалица	
	Члан:	доц. др Иван Каштелан	Потпис ментора
	Члан, ментор:	доц. др Јелена Ковачевић	



KEY WORDS DOCUMENTATION

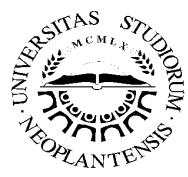
Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Aleksandar Martinović	
Mentor, MN:	PhD Jelena Kovačević	
Title, TI:	The implementation of a 3D audio image rendering module on DSP platform by MPEG-H standard	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2020	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: <small>(chapters/pages/ref./tables/pictures/graphs/appendices)</small>	8/40/15/4/18/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	3D audio, HOA, DSP, sound processing, rendering, MPEG-H	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	This thesis describes the implementation of HOA decoder on the Cirrus Logic platform. The implementation involves the analysis of a C++ reference code written in accordance with MPEG-H standard and writing assembly code, which is based on it, for the target DSP platform.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	PhD Nebojša Pjevalica
	Member:	PhD Ivan Kaštelan
	Member, Mentor:	PhD Jelena Kovačević
		Menthor's sign

Zahvalnost

Zahvaljujem se Institutu RT-RK što mi je omogućio realizaciju ovog rada, kao i mentoru doc. dr Jeleni Kovačević i tehničkom mentoru Andreju Popoviću, koji su pružili svoju nesebičnu pomoć tokom izrade rada.

Takođe bih se zahvalio kolegama Sanji Čordašić, Goranu Kruniću, Radetu Maksimoviću, Srđanu Šuvakovu, Predragu Sladeku i Jeleni Boroji, kao i ostalim kolegama na pruženoj pomoći, strpljenju, savetima i sugestijama.

Najviše od svih sam zahvalan svojoj porodici i prijateljima koji su mi pružali neizmernu podršku i bili uz mene u svakom trenutku.



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



SADRŽAJ

1.	Uvod	1
2.	Teorijske osnove.....	3
2.1	3D Zvučna slika	3
2.1.1	Sveobuhvatni zvuk	3
2.1.1.1	Kanalni sveobuhvatni zvuk	4
2.1.1.2	Objektni sveobuhvatni zvuk.....	4
2.1.1.3	Scenski sveobuhvatni zvuk	4
2.1.2	Interakcija/Personalizacija.....	6
2.1.3	Od distributera do korisnika	6
2.2	MPEG-H Standard	7
2.2.1	Modul za prilagođenje formata	9
2.2.2	Renderer objekta.....	9
2.3	HOA	10
2.3.1	Dekompozicija zvučnog polja u koderu	11
2.3.2	MPEG-H dekoder za HOA renderovanje	12
3.	Algoritmi za HOA obradu	13
3.1	Kontrola inverznog pojačanja	13
3.2	Preraspodela kanala	14
3.3	Sinteza dominantnog zvuka	15
3.3.1	Računanje HOA prezentacije aktivnih direkcijskih signala.....	15
3.3.2	Računanje HOA prezentacije prediktivnih direkcijskih signala.....	17
3.3.3	Računanje HOA prezentacije aktivnih vektorskih signala	19
3.3.4	Prostorno-vremenska interpolacija V-vektora.....	20
3.4	Ambijentalna sinteza.....	22
3.5	HOA kompozicija	23



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



4. Ciljna DSP platforma.....	24
5. Programsko rešenje.....	27
5.1 Analiza referentnog koda	27
5.2 Inicijalizacija.....	27
5.3 Pregled korišćenih polja i metoda klasa.....	28
5.3.1 Klasa HOADecoder	28
5.3.2 Klasa FrameParams	28
5.3.3 Klasa InverseGainControl	29
5.3.4 Klasa AmbienceSynthesis	29
5.3.5 Klasa DirBasedPredomSoundSynthesis	29
5.3.6 Klasa VectorBasedPredomSoundSynthesis	30
5.3.7 Klasa HOAMatrix	31
5.3.8 Klasa HOAMatrixStatic	31
5.4 Optimizacija asemblerorskog koda	31
5.5 Razvojno okruženje	32
6. Rezultati.....	34
6.1 Ispitivanje ispravnosti programa uz pomoć bit-identičnih testova	34
6.2 Potrošnja resursa	35
7. Zaključak	37
8. Literatura	39

SPISAK SLIKA

Slika 1.1 Prikaz 3D zvučne slike [4].....	2
Slika 2.1 Primer kanalnog sveobuhvatnog zvuka sistema 5.1.4	4
Slika 2.2 Sferični harmonici od reda $N = 0$ (gornji red) do reda $N = 3$ (donji red) [9]	6
Slika 2.3 Blok šema dekodera MPEG-H 3D zvučne slike	9
Slika 2.4 Blok dijagram faze MPEG-H renderovanja	10
Slika 2.5 Kodovanje HOA u MPEG-H.....	11
Slika 2.6 Dekodovanje HOA u MPEG-H	12
Slika 3.1 Sinteza dominantnog zvuka.....	15
Slika 3.2 Računanje vektorskih signala	19
Slika 3.3 Interpolacija V-vektora.....	20
Slika 3.4 Primer postepenog pojačavanja (engl. <i>Fade-In</i>) i postepenog smanjivanja (engl. <i>Fade-Out</i>) komponenti	22
Slika 4.1 Blok dijagram procesora <i>DC49844</i>	24
Slika 4.2 Blok dijagram <i>Crystal 32 DSP</i> arhitekture.....	25
Slika 4.3 Blok dijagram <i>Crystal 32 DSP</i> jezgra	26
Slika 5.1 Izgled <i>for</i> petlje u C++ kodu	32
Slika 5.2 Izgled hardverske petlje u asembleru	32
Slika 5.3 Primer kontrolisanog izvršavanja u CLIDE razvojnog okruženju	33
Slika 6.1 Testiranje modula	35

SPISAK TABELA

Tabela 3.1 Vrednosti L_{IP} u zavisnosti od vrednosti <code>CodedSpatialInterpolationTime</code> ..	17
Tabela 4.1 Raspoloživa memorija i procesorsko vreme po jezgrima procesora <i>DC49844</i>	25
Tabela 6.1 Prikaz potrošnje memorije za HOA obradu	35
Tabela 6.2 Prikaz potrošnje procesorskog vremena	36

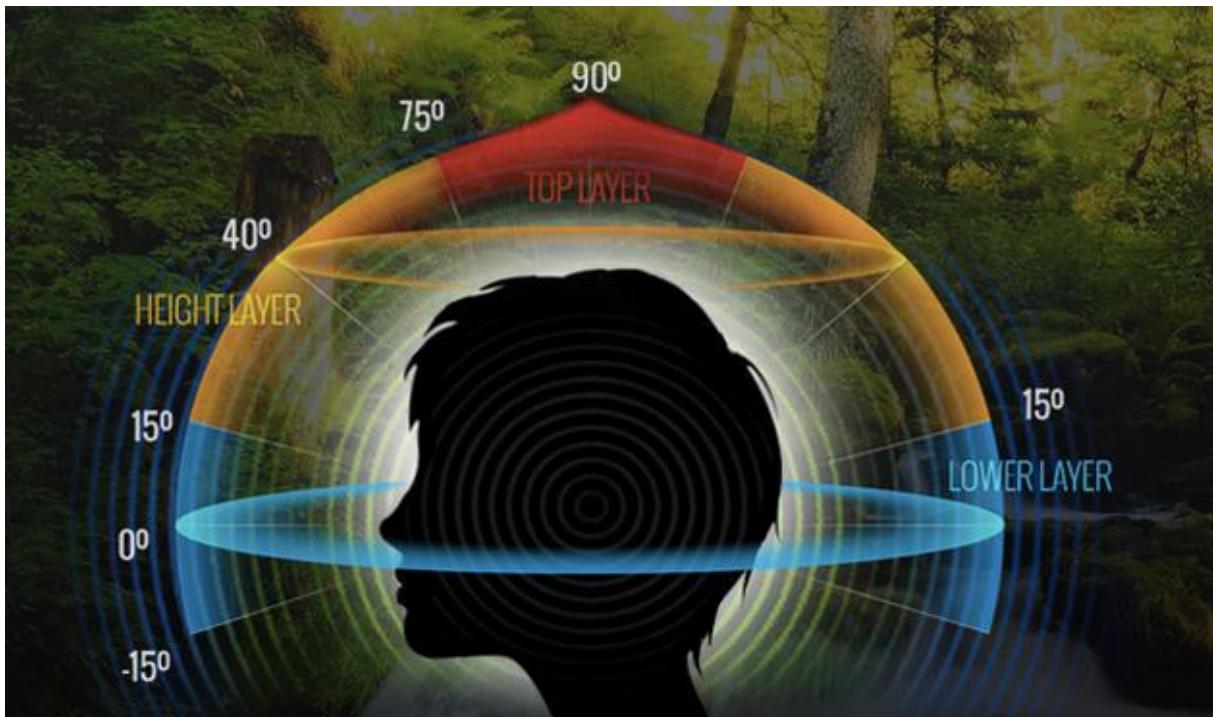
SKRAĆENICE

3D	- <i>Three-dimensional</i> , Trodimenzionalni prostor
DSP	- <i>Digital signal processor</i> , Digitalni signalni procesor
MPEG-H	- <i>Moving Picture Experts Group</i> , Grupa standarda
CLIDE	- <i>Cirrus Logic Integrated Development Environment</i> , Integrисано razvojno okruženje kompanije <i>Cirrus Logic</i>
HOA	- <i>Higher Order Ambisonics</i> , Ambisonici višeg reda
VR	- <i>Virtual Reality</i> , Virtuelna realnost
VRML	- <i>Virtual Reality Modeling Language</i> , Jezik za modelovanje virtuelne realnosti
BIFS	- <i>Binary Format for Scenes</i> , Binarni format scena
AVR	- <i>Audio/Video receiver</i> , Audio/Video prijemnik
USAC	- <i>Unified Speech and Audio Coding</i> , Objedinjeni koder govora i zvuka
SAOC	- <i>Spatial Audio Object Coding</i> , Kodovanje prostornih zvučnih objekata
VBAP	- <i>Vector Base Amplitude Panning</i> , Pomeranje amplitude vektora u 3D prostoru
MIPS	- <i>Milions of Instructions per Second</i> , Miliona instrukcija po sekundi
DMA	- <i>Direct Memory Access</i> , Direktan pristup memoriji
AGU	- <i>Address Generation Unit</i> , Jedinica za generisanje adrese

1. Uvod

Zadatak ovog rada predstavlja realizaciju algoritma ambisonika višeg reda u sklopu završne obrade 3D audio slike na *Cirrus Logic* DSP platformi po MPEG-H standardu. Bilo je neophodno napisati odgovarajući asemblerски kod, po uzoru na referentni kod koji je napisan u C++ programskom jeziku. Razvojno okruženje korišćeno za pisanje asemblerског кода је CLIDE, integrisano okruženje razvijeno kao podrška digitalnim signal procesorima kompanije *Cirrus Logic*. U planu је да celokupna implementacija u asemblerском jeziku буде применјена на četvorojezgarnom digitalnom signalnom procesoru oznake DC49844-64.

Kako se tehnologija vremenom razvijala, tako se stvorila i sve veća potreba za unapređenjem tehnologije reprodukcije zvuka. U današnje vreme, sve popularnije postaje gledanje filmova u kućnim uslovima, nego odlazak na projekcije filmova. Koliko je u projekciji filma bitan kvalitet slike, toliko je bitan i kvalitet zvuka. Vremenom је nastala želja da se zvuk reprodukuje toliko dobro da slušalac ima osećaj da se audio objekti nalaze svuda oko njega. U audio kodovanju se ovakav koncept naziva trodimenzionalna zvučna slika (3D audio). Uz pomoć zvuka kodovanog po 3D principima moguće je dovesti ugodaj gledanja filma na TV uređaju, na nivo koji je jednak gledanju projekcije filma u bioskopskoj sali. Na tržištu postoji nekoliko tehnologija koje daju rešenja 3D zvučne slike, као што су Dolby Atmos, MPEG-H, Auro-3D и DTS:X. Ovaj rad се bazira na tehnologiji MPEG-H standarda, коју је развио Fraunhofer institut.[1]. MPEG-H standard припада групи стандарда отвореног типа.[2] То је, на пример, omogućило компанији Amazon да путем платформе *Amazon Music* и уз помоћ њихових pametnih zvučnika svoјим korisnicima ponude muziku која садржи sveobuhvatни звук. [3]



Slika 1.1 Prikaz 3D zvučne slike [4]

Glavna tema ovog rada je *Higher Order Ambisonics* (skraćeno HOA) algoritam, odnosno algoritam ambisonika višeg reda. HOA predstavlja ambijentalno polje pritiska koje okružuje slušaoca. Pored horizontalne ravni, uz pomoć HOA dobija se izvor zvuka kako ispod slušaoca, tako i iznad. U sklopu MPEG-H standarda, HOA predstavlja jedan od modula za završnu obradu zvuka. Zadatak ovog modula je da obrađuje HOA koeficijente koji sadrže informacije o okolini, odnosno oni predstavljaju upravo opis polja pritiska.[5]

Ovaj rad se sastoji od osam poglavlja. U drugom poglavlju se govori o teorijskim osnovama. Objasnjeno je šta znači 3D zvučna slika, kao i na koji način se ona primenjuje. Zatim se govori i o MPEG-H standardu, u okviru kojeg je opisan HOA dekoder koji je glavna tema ovog poglavlja. U trećem poglavlju govori se o algoritmima uz pomoć kojih se izvršava sama HOA obrada. U četvrtom poglavlju se govori o ciljnoj platformi za koju je napisan asemblerски kod. Spomenute su i neke od specifičnosti arhitekture ciljne platforme. U petom poglavlju je opisana suština rešenja zadatka, tj. implementacija zadatih funkcija u asemblerском jeziku. U šestom poglavlju, govori se o verifikaciji ispravnosti ponuđenog rešenja, tj. o tome na koji način je proverena funkcionalnost implementiranih funkcija i na kraju je dat uvid u potrošnju procesorskog vremena. U sedmom poglavlju nalazi se zaključak ovog rada, dok je u poslednjem, osmom, navedena literatura koja je korišćena pri izradi ovog rada.

2. Teorijske osnove

U okviru ovog poglavlja biće opisan koncept trodimenzionalnog zvuka, nakon kojeg sledi pregled MPEG-H standarda, gde se u okviru standarda nalazi opis tehnologije HOA obrade.

2.1 3D Zvučna slika

Sa razvojem VR tehnologije, grafike za video igre, 3D grafike u filmovima, sve se više razmišlja i na temu 3D zvučne slike, koji će ići u korak sa razvojem grafike. Postoji težnja da se kod gledaoca 3D grafike, stvari osećaj da se stvarno nalazi u datom okruženju. To je realno ostvarivo kada se 3D grafici doda i 3D zvučna slika.

Ljudski mozak je sposoban da lokalizuje zvuk u prostoru oko nas koristeći kombinaciju binauralnih tonova koje primi uz pomoć čula sluha. Nakon obrade ovih tonova u mozgu, dobijamo percepciju o uglu, nagibu i udaljenosti od izvora zvuka u prostoru. Tehnike za reprodukciju 3D zvučne slike imaju za cilj da rekonstruišu binauralne tonove koji se prostiru od izvora zvuka iz okoline do uha slušaoca. [6]

2.1.1 Sveobuhvatni zvuk

Sistemi nove generacije nam pružaju doživljaj sveobuhvatnog (engl. *immersive*) zvuka, koji pored toga što se prostire po horizontalnoj, prostire i po vertikalnoj osi, i po tome se razlikuje u odnosu na okružujući zvuk. Ovakav zvuk korisniku pruža bolji osećaj realnosti, odnosno korisnik se oseća kao da se i sam nalazi unutar scene ili programa, uprkos tome što je on samo posmatrač.

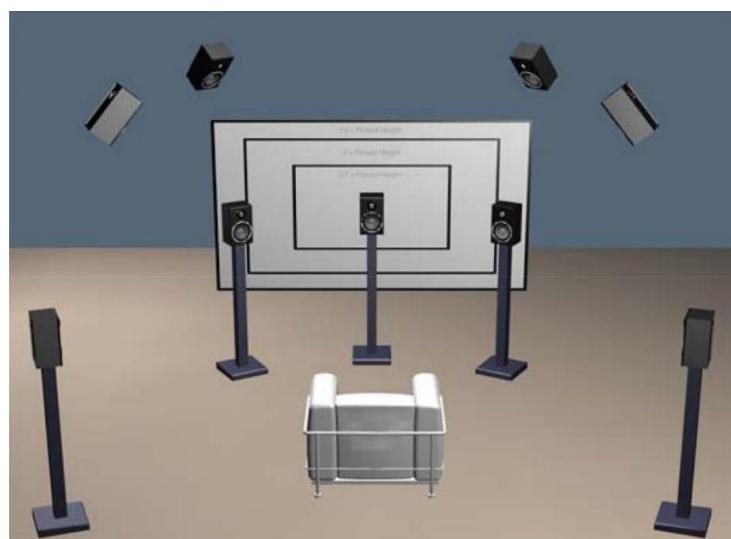
Sveobuhvatni zvuk se može prenositi na tri osnovna načina:

- tradicionalni kanalni zvuk, gde se svaki pojedinačni kanal prenosi do tačno jednog određenog zvučnika;

- prenos zvuka kroz objekte, koji se mogu nalaziti u sve tri dimenzije nezavisno od pozicije zvučnika;
- scenski (ambisonici), gde je zvučna scena predstavljena uz pomoć skupa koeficijenata signala, koji predstavljaju linearnu težinu funkcija prostornih ortogonalnih sfernih harmonika. [7]

2.1.1.1 Kanalni sveobuhvatni zvuk

U prenosu zvuka putem kanala, sveobuhvatni zvuk se reprodukuje nadogradnjom plafonskih zvučnika na klasične 5.1 i 7.1 sisteme, koji se trenutno koriste za TV i Blu-ray multimediju. Kanali za plafonske zvučnike su dodati ovom sistemu zvučnika da bi se napravio zvuk koji se prostire iznad slušaoca. Obično se za to koristi četiri zvučnika (kao što je prikazano na slici 2.1) iako se sličan efekat može postići i sa samo dva prednja plafonska zvučnika. Postoje i sistemi zvučnika koji su mnogo složeniji, kao što su na primer 22.2 i 10.2 sistemi. [8]



Slika 2.1 Primer kanalnog sveobuhvatnog zvuka sistema 5.1.4

2.1.1.2 Objektni sveobuhvatni zvuk

Prethodno je objektni sveobuhvatni zvuk korišćen u video igramu i multimediji kao što su VRML i MPEG-4 BIFS, da bi se locirao izvor zvuka u trodimenzionalnom svetu. U ovim slučajevima, izvori zvuka mogu biti zaklonjeni nekim objektima. Sa druge strane, zvučni objekti za video ili film su obično usko vezani za predefinisani koordinantni sistem usmeren na slušaoca koji se ne pomera, odnosno bez modelovanih prepreka i sobne reverberacije (odbijanja zvuka).

2.1.1.3 Scenski sveobuhvatni zvuk

Svuda oko nas se čuju različiti zvukovi, ali ljudsko uho nije sposobno da izoluje svaki ponaosob, kao na primer cvrkut ptica ili padanje kiše. Uvek će se pored zvukova koje želimo

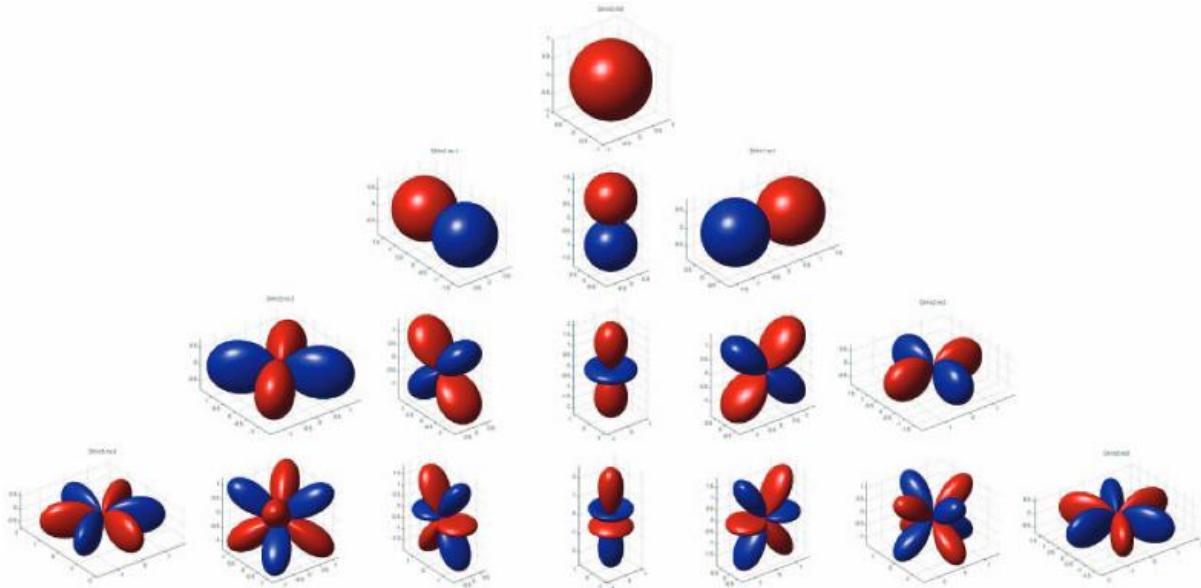
da čujemo, u određenoj meri čuti i neki pozadinski zvukovi. Ovakvi pozadinski zvukovi koji su spojeni ili pomešani sa korisnim zvučnim objektima se nazivaju scenski sveobuhvatni zvukovi.

Zadatak scenskog zvuka je prikazati lokalizovano polje pritiska $p(x, y, z, t)$ što preciznije. Da bi ovo bilo uspešno urađeno koristeći HOA, polje pritiska treba biti prikazano kao rešenje talasne jednačine koristeći osnovne funkcije sferičnih harmonika:

$$p(r, \theta, \phi, \omega, t) = \left[\sum_{n=0}^{\infty} j_n \left(\frac{\omega r}{c} \right) \sum_{m=-n}^n a_n^m(\omega, t) Y_n^m(\theta, \phi) \right] e^{i\omega t} \quad (1)$$

gde c predstavlja brzinu zvuka, ω kružnu frekvenciju, j_n sferičnu Beselovu funkciju n-tog stepena, $Y_n^m(\omega, t)$ - funkcije sferičnog harmonika n-tog reda i m-tog stepena za azimut ϕ ugao nagiba θ . Razloženi koeficijenti $a_n^m(\omega, t)$ u potpunosti opisuju polje zvuka i drugačije se nazivaju koeficijenti sferičnog harmonika, HOA koeficijenti, HOA signali, ili jednostavno kao „koeficijentni signali“. Primer sferičnih harmonika se nalazi na slici 2.2. Za praktične svrhe, beskonačna suma u jednačini (1) je postavljena na $n = N$, što rezultira $(N + 1)^2$ koeficijentnih signala.

S obzirom da $a_n^m(\omega, t)$ prezentacija nema predstavu o poziciji zvučnika, renderer (modul za završnu obradu) je zadužen da pretvara koeficijente u signale poznate zvučniku. Rendereri ovog tipa obično imaju uvid u broj i pozicije dostupnih zvučnika i srazmerno tome proizvode pogodno renderovanje za određeno okruženje. Napredniji rendereri mogu takođe da uračunaju i trenutne uslove akustike, kao što je reverberacija prostorije. Ova metoda „ne vezivanja“ zvuka za pozicije zvučnika, dozvoljava da format bude prilagodljiv praktično bilo kom rasporedu zvučnika. To takođe održava prostornu rezoluciju, koja je neophodna za akustično fokusiranje na prostorne regije, dozvoljavajući slušaocu interakciju sa poljem zvuka, na način koji nije moguć korišćenjem kanalnog zvuka.



Slika 2.2 Sferični harmonici od reda $N = 0$ (gornji red) do reda $N = 3$ (donji red) [9]

2.1.2 Interakcija/Personalizacija

Korišćenje audio objekata, obično u kombinaciji sa kanalnim ili scenskim zvukom, omogućava gledaocu interakciju sa sadržajem za pravljenje personalizovanog doživljaja slušanja. MPEG-H sistem opisuje TV program kao graf objekata i za njih vezanih metapodataka koji daje unapred predefinisane zvučne scene, ponuđene od strane emitera.

U naj složenijem slučaju, sistem podržava slanje programa kao 16 nezavisnih objekata, kojima korisnik može da menja poziciju u tri dimenzije nezavisno i jačinu zvuka, od potpuno utišanog do +12 dB. Međutim, ovoliki stepen slobode je pogodan za korisnike koji se bolje razumenju u tehnologiju od običnog gledaoca, s obzirom da za običnog gledaoca ovo može biti previše složeno.

Najjednostavniji primer predstavlja prenos nekog sportskog događaja uz koji se emituju komentari na dva ili više različitih jezika. Gledalac ima mogućnost da odabere jezik na kojem želi da sluša komentare.

2.1.3 Od distributera do korisnika

Novi tipovi platformi i infrastruktura prenosa su postali veoma značajni i konstantno su u razvoju. U višeplatformskom okruženju, isti MPEG-H sadržaj se dostavlja preko više različitih kanala za prenos i može se reprodukovati na više različitih uređaja (npr. AVR, TV, mobilni telefon) u različitim okruzenjima (npr. tišina dnevne sobe, glasni gradski prevoz,...).

Sa tačke gledišta korisnika, karakteristike zvučnog sadržaja bi trebale odgovarati pojedinačnim uslovima slušanja i prioritetima, bez obzira na poreklo sadržaja i njegovog kanala

za prenos. Kao posledica različitosti okruženja za slušanje, potrebna je prilagodljivost zvučnog sadržaja, da bi se izbegla neugodnost kod korisnika u različitim slučajevima. Ovo su neki od najučestalijih problema koji mogu da izazovu neprijatnost kod slušaoca:

- Korisnik mora da prilagođava jačinu zvuka pri promeni TV programa, zbog toga što jačina zvuka nije ujednačena.
- Uzastopni objekti u programu nemaju istu jačinu, npr. previše glasne reklame nakon filma.
- Razumljivost dijaloga u filmu u nekim tišim delovima, može biti teška za slušanje usled bučnog okruženja.
- Raspon zvučnog sadržaja previše veliki za željeni nivo reprodukcije: Nivo glasnih delova filma su neugodno visoki, kada su neki tiši delovi sasvim dovoljno glasni; ili kada su tiši delovi skoro pa nečujni, dok su glasniji delovi umerene jačine zvuka.
- Raspon zvučnog sadržaja je preveliki za uređaj za reprodukciju, npr. zvučnici lošijeg kvaliteta u uređaju.
- Odsecanje zvučnog signala nakon što se izvrši *downmixing* originalnog sadržaja na manji broj zvučnika za reprodukciju zvuka. [10]

2.2 MPEG-H Standard

Poboljšanjem algoritama za kompresiju videa stvorili su se uslovi za prenos video sadržaja visoke rezolucije kao što su 4K ili 8K, koji mogu da se reprodukuju u kućnim uslovima. Pored ovako kvalitetnog video sadržaja stvorila se potreba i za zvukom visokog kvaliteta. To je dovelo do nastanka 3D zvučne slike. Razvoj algoritama za kodovanje videa je uslovio nastanak novih audio tehnologija i poboljšanje postojećih algoritama za obradu zvuka.

Osnovna ideja novih tehnologija je da se zvuk predstavi na isti način bez obzira na uslove reprodukcije. Slušalac bi trebao da ima potpuno isti doživljaj i u bioskopskoj sali i kraj televizora sa dva ugrađena zvučnika. Konfiguracija zvučnika, karakteristike prostorije i nivo pozadinske buke ne bi trebali da utiču na kvalitet zvuka. Za često korištene konfiguracije zvučnika (kao što su 5.1, 7.1, 9.1,...) producent ili emiter može unapred da podesi *downmix* koeficijente koji obezbeđuju najbolji kvalitet reprodukovanih zvuka. Ovi koeficijenti se prenose kao metapodaci u okviru bitskog toka. Problem oko rasporeda zvučnika, MPEG-H rešava slanjem pozicije svakog objekta rendereru u obliku metapodataka, pri čemu se računaju *downmix* koeficijenti za ciljnu konfiguraciju zvučnika.

MPEG-H standard nudi mogućnost podrške do 128 kanala kao i 128 objekata, koji mogu da se preslikaju na najviše 64 zvučnika. Da bi se preneo bitski tok koji sadrži ovoliki broj kanala

i objekata, potreban je veoma veliki propusni opseg (engl. bandwidth). Međutim, usled ograničenja trenutno dostupnog bitskog protoka, napravljeni su profili koji ograničavaju složenost dekodera na sledeći način (tabela 2.1):

- po broju podržanih kanala ili objekata,
- po skupu podržanih funkcionalnosti i
- po broju mogućih izlaznih konfiguracija zvučnika.

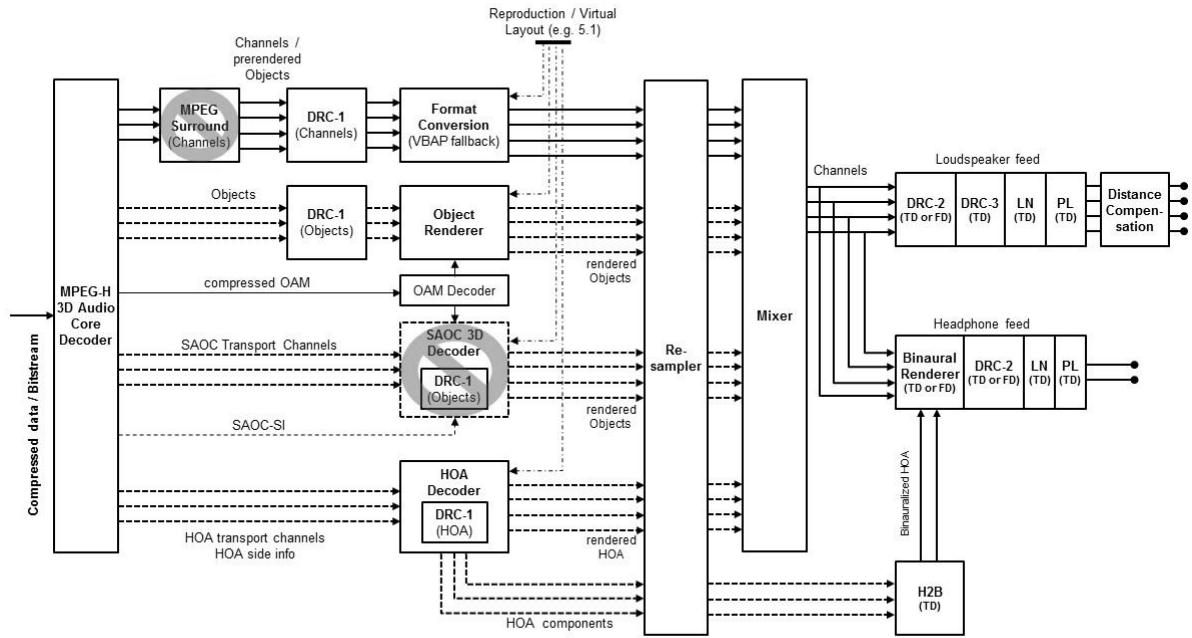
Nivo profila	1	2	3	4	5
Najveća učestanost odabiranja[kHz]	48	48	48	48	96
Najveći broj kanala kodera u bitskom toku	10	18	32	56	56
Najveći broj istovremeno dekodovanih kanala u koderu	5	9	16	28	28
Najviše zvučnika na kojima se emituje izlaz	2	8	12	24	24
Primer postavke zvučnika	2.0	7.1	7.1+4H	22.2	22.2
Najveći broj dekodovanih objekata	5	9	16	28	28

Tabela 2.1 MPEG-H 3D Audio nivoi profila niske složenosti

Na slici 2.3 dat je šematski prikaz MPEG-H sistema, na kom su predstavljeni svi veći moduli za obradu signala (precrtani delovi nisu deo sistema):

- Na početku, kompresovani bitski tok koji sadrži sve signale zvuka, bili oni kanali, objekti ili HOA komponente, se dekoduju uz pomoć USAC dekodera (USAC-3D).
- Ulazni kanalni signali su pomoću konvertora formata prilagođeni reprodukciji na ciljnoj konfiguraciji zvučnika
- Uz pomoć renderera objekata se renderuju signali objekata za određenu konfiguraciju zvučnika.

HOA sadržaj se renderuje za ciljni sistem zvučnika koristeći određene HOA metapodatke.



Slika 2.3 Blok šema dekodera MPEG-H 3D zvučne slike

2.2.1 Modul za prilagođenje formata

MPEG-H renderer sadrži modul za prilagođenje formata koji prilagođava dekodovani ulazni signal kanala za različite formate izlaza. Da bi se proizveo izlazni signal visokog kvaliteta, modul za prilagođenje formata u MPEG-H 3D zvučnoj slici poseduje sledeće karakteristike:

- Automatsko generisanje optimizovanih *downmix* matrica, uzimajući u obzir nestandardne pozicije zvučnika.
- Podrška za opciono prenosive *downmix* matrice da bi se sačuvala stilizovana namera producenta ili emitera.
- Primena ekvilajzera da bi se očuvali kvalitet i boja zvuka.
- Unapređeni aktivni *downmix* algoritam za izbegavanje *downmixing* artefakata.

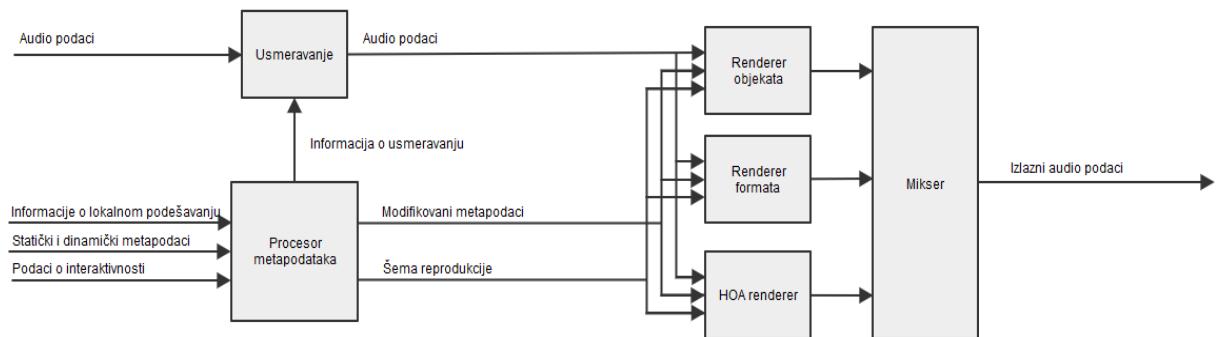
2.2.2 Renderer objekta

U MPEG-H sistemu, metapodaci između ostalog prenose i prostornu poziciju objekata zvuka. Na osnovu tih podataka renderer može da pripremi izlaze za ciljnu konfiguraciju zvučnika. Takođe, vremenski promenljiva pojačanja se mogu signalizirati pojedinačno za svaki zvučni objekat.

Renderer objekata primenjuje VBAP (engl. *Vector Base Amplitude Panning*) ili pomeranje amplitude vektora u 3D prostoru, da bi renderovao preneseni zvučni objekat za zadatu konfiguraciju izlaznih kanala. Kao ulaze u sistem, renderer očekuje:

- Geometrijske podatke ciljne postavke za renderovanje.
- Jedan dekodovani tok podataka zvuka po prenesenom zvučnom objektu.
- Dekodovan objektni metapodatak povezan sa prenesenim objektima, npr. vremenski promenljive pozicije podataka i pojačanja.

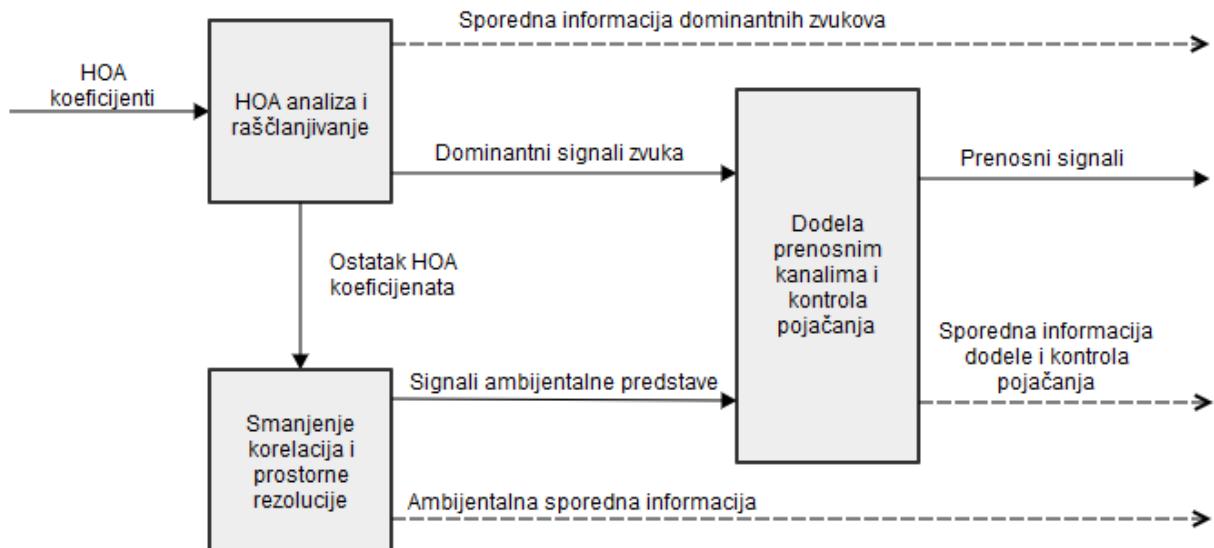
VBAP se oslanja na triangulaciju 3D površine koja okružuje slušaoca. U okviru renderera objekata MPEG-H 3D zvučne slike je implementiran algoritam automatske triangulacije za proizvoljnu izlaznu konfiguraciju zvučnika. S obzirom da nisu sve ciljne konfiguracije zvučnika u potpunosti 3D konfiguracije, npr. mnogim konfiguracijama nedostaje zvučnik ispod horizontalne ravni, triangulacija uvodi imaginarne zvučnike da bi se obezbedile celokupne mreže 3D trouglova bilo koje konfiguracije za VBAP algoritam.



Slika 2.4 Blok dijagram faze MPEG-H renderovanja

2.3 HOA

HOA (engl. *Higher Order Ambisonics*) ili Ambisonici višeg reda su matematička predstava akustičnog talasnog polja koje okružuje slušaoca. Talasno polje se predstavlja pomoću sfernih harmonika, koji određuju količinu akustičnog talasnog polja unutar određenog slobodnog regiona koji okružuje poziciju slušaoca do gornje granice frekvencije iznad koje nastaje prostorni alijsing. Vremenski promenljivi koeficijenti sfernog proširenja harmonika se nazivaju HOA koeficijenti i oni su nosioci informacije talasnog polja koji se treba preneti i reprodukovati.



Slika 2.5 Kodovanje HOA u MPEG-H

Kako bi se očuvao visok stepen kompresije MPEG-H kodeka, HOA koeficijenti se ne upisuju direktno u bitski tok, već se primeni proces kodovanja u dve faze. Primjenjuje se prostorno kodovanje HOA komponenti i višekanalno perceptivno kodovanje.

Prostorni blok za kodovanje HOA koeficijenata primjenjuje dva osnovna principa: dekompozicija (razlaganje) ulaznog polja i dekorelacija signala pre prenosa u jezgro kodera.

2.3.1 Dekompozicija zvučnog polja u koderu

U HOA koderu zvučno polje koje je predstavljeno HOA koeficijentima je razloženo na glavne i ambijentalne komponente zvuka. U isto vreme, generišu se parametarske sporedne informacije, što dekoderu signalizira vremenski promenljivu aktivnost komponenti različitog zvučnog polja.

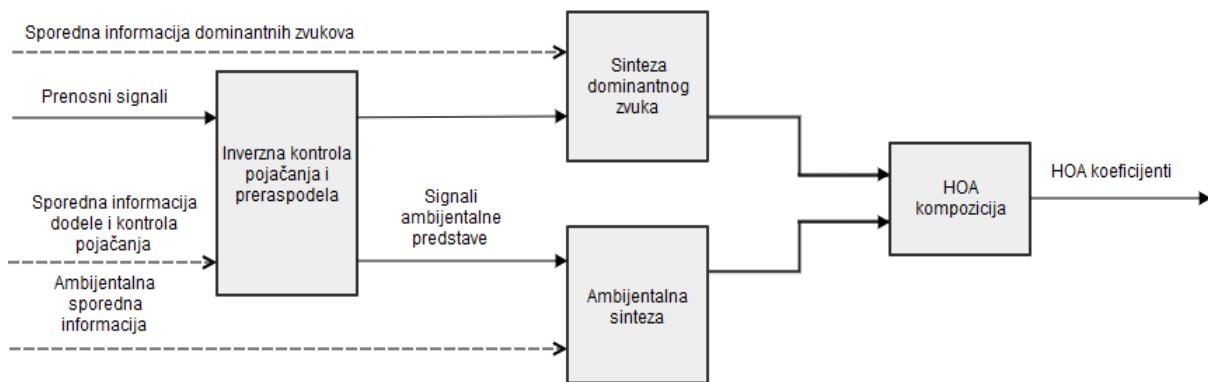
Glavne komponente uglavnom sadrže usmerene zvukove i kodirane su kao dodatak ravnom talasu koji putuje kroz korisno talasno polje u određenim pravcima. Broj glavnih komponenti može varirati kroz vreme kao i njihovi pravci. Prenose se kao zvučni tok zajedno sa pridruženim vremenski promenljivim parametarskim informacijama kao što su pravac i aktivnost usmerenih komponenti zvuka u polju.

Ostatak HOA ulaza, koji nije obuhvaćen glavnim komponentama, predstavlja ambijentalnu komponentu zvučnog polja. Ona uglavnom sadrži neusmerene komponente zvuka. Detalji prostornih svojstava ovog dela polja se smatraju manje bitnim. Stoga, prostorna rezolucija ambijentalne komponente je tipično umanjena ograničavanjem HOA reda da bi se poboljšala efikasnost kodovanja.

2.3.2 MPEG-H dekoder za HOA renderovanje

HOA sadržaj se u MPEG-H dekoderu dekoduje u HOA prezentaciju po sledećim koracima (slika 2.6):

- Dekodovanje višekanalnog USAC 3D jezgra.
- Inverzna dekorelacija ambijentalnog zvuka, npr. transformacija iz dekorelisane predstave u predstavu HOA koeficijenata.
- Sinteza prezentacije HOA koeficijenata glavnih komponenti zvuka.
- Kompozicija HOA (superpozicija HOA prezentacije glavnih i ambijentalnih komponenti).



Slika 2.6 Dekodovanje HOA u MPEG-H

U narednom koraku obrade dekodovana HOA prezentacija se renderuje za određeni sistem zvučnika koristeći opšti HOA renderer. HOA renderovanje se sastoji od jednostavnih matričnih množenja višekanalnih HOA prezentacija i renderovanih matrica.

Renderovana HOA matrica mora biti generisana u toku inicijalizacije ili kada se HOA red ili postavke reprodukcije promene. U matrici se nalaze pojačanja HOA komponenti koja određuju njihov uticaj u izlaznoj konfiguraciji zvučnika. Množenjem matrice i HOA koeficijenata dobija se najbolja aproksimacija akustičnog polja u prostoru oko slušaoca. Jedna od glavnih karakteristika dizajna renderovane HOA matrice je očuvanje energije, što znači da je jačina zvuka očuvana nezavisno od konfiguracije zvučnika. [11]

3. Algoritmi za HOA obradu

U ovom poglavlju naglasak će biti na algoritmima koji omogućavaju uspešnu realizaciju HOA obrade. To su blokovi:

- kontrola inverznog pojačanja,
- preraspodela kanala,
- sinteza dominantnog zvuka i
- ambijentalna sinteza.

3.1 Kontrola inverznog pojačanja

Cilj bloka za kontrolu inverznog pojačanja - IGC (engl. *Inverse Gain Control*) jeste invertovanje modifikacije pojačanja koja se izvršava nad signalima pre perceptualnog kodovanja u fazi HOA kodovanja kako bi se ponovo stvorio njihov početni raspon vrednosti.

Za tu svrhu, zadužena je funkcija prozoriranja nepromenljivog šablonu (engl. *fixed template transition window function*):

$$f_{IGC} = [f_{IGC}(1) \quad f_{IGC}(2) \quad \cdots \quad f_{IGC}(L)]$$

čiji elementi su definisani sa:

$$f_{IGC}(l) = \frac{1}{4} \cos\left(\frac{\pi(i-1)}{L-1}\right) + \frac{3}{4}, \quad l = 1, \dots, L.$$

Sa prepostavkom da će ulazni okvir (engl. *frame*) $z_i(k)$ i izlazni okvir $y_i(k)$ biti predstavljeni njihovim odbircima:

$$z_i(k) = [z_i(k, 1) \quad z_i(k, 2) \quad \cdots \quad z_i(k, L)]$$

$$y_i(k) = [y_i(k, 1) \quad y_i(k, 2) \quad \cdots \quad y_i(k, L)]$$

računanje odbiraka izlaznog okvira je dato sa:

$$y_i(k, l) = \begin{cases} \frac{z_i(k, l)}{g_{IGC,i}(k-1)} \cdot [f_{IGC}(l)]^{-e_i(k)} & , \beta_i(k) = 0 \\ \frac{z_i(k, l)}{g_{IGC,i}(k-1)} \cdot 2^{e_i(k)} & , \beta_i(k) = 1 \end{cases}, \quad i = 1, \dots, I, l = 1, \dots, L.$$

Faktor $g_{IGC,i}(k-1)$ je inicijalizovan vrednostima:

$$g_{IGC,i}(0) = 1, \quad i = 1, \dots, I.$$

i rekurzivno se ažurira u k-tom okviru sa:

$$g_{IGC,i}(k) = g_{IGC,i}(k-1) \cdot 2^{-e_i(k)}, \quad i = 1, \dots, I.$$

3.2 Preraspodela kanala

Svrha bloka za preraspodelu kanala (engl. *Channel Reassignment*) je da se napravi:

- okvir $X_{ps}(k)$ od svih dominantnih signala zvuka,
- okvir $C_{I,AMB}(k)$ iz prezentacije ambijentalnih HOA komponenti iz okvira signala sa korigovanim pojačanjem $y_i(k), i \in \{1, \dots, I\}$ i
- vektor dodele $v_{AMB,ASSIGN}(k)$, koji za svaki prenosni kanal ukazuje na indeks sa mogućnošću sadržaja sekvene koeficijenta ambijentalne HOA komponente.

Pored toga, koriste se skupovi $\mathcal{J}_{DIR,ACT}(k)$ i $\mathcal{J}_{VEC,ACT}(k)$, koji bi trebali da sadrže prve elemente svih n-torki od matrica $\mathcal{M}_{DIR}(k)$ i $\mathcal{M}_{VEC}(k)$. Bitno je naglasiti da se prepostavlja da ova dva skupa ne sadrže zajedničke elemente.

Za raspodelu koriste se sledeći koraci:

1. Prepostavlja se da je okvir $X_{ps}(k)$ okvir svih dominantnih signala zvuka sastavljen od pojedinačnih okvira dominantnih signala zvuka kao:

$$X_{ps}(k) = \begin{bmatrix} X_{ps,1}(k) \\ X_{ps,2}(k) \\ \vdots \\ X_{ps,D}(k) \end{bmatrix},$$

gde svaki okvir treba da se sastoji od njegovih odbiraka prema:

$$x_{ps_i}(k) = [x_{ps_i}(k, 1) \ x_{ps_i}(k, 2) \ \dots \ x_{ps_i}(k, L)], \quad i = 1, \dots, J,$$

gde je $J = I - O_{MIN}$

Vrednosti odbiraka okvira usmerenog signala se računaju po formuli:

$$x_{ps_i}(k, l) = \begin{cases} y_i(k, l) & , i \in \mathcal{J}_{DIR}(k) \cup \mathcal{J}_{VEC}(k) \\ 0 & , \text{inače} \end{cases}, \quad i = 1, \dots, J, \quad l = 1, \dots, L.$$

2. Okvir $C_{I,AMB}(k)$ prezentacije međuzavisnosti ambijentalnih HOA komponenti je sastavljen prema:

$$C_{I,AMB}(k) = \begin{bmatrix} C_{I,AMB,1}(k) \\ C_{I,AMB,2}(k) \\ \vdots \\ C_{I,AMB,o}(k) \end{bmatrix}$$

gde su:

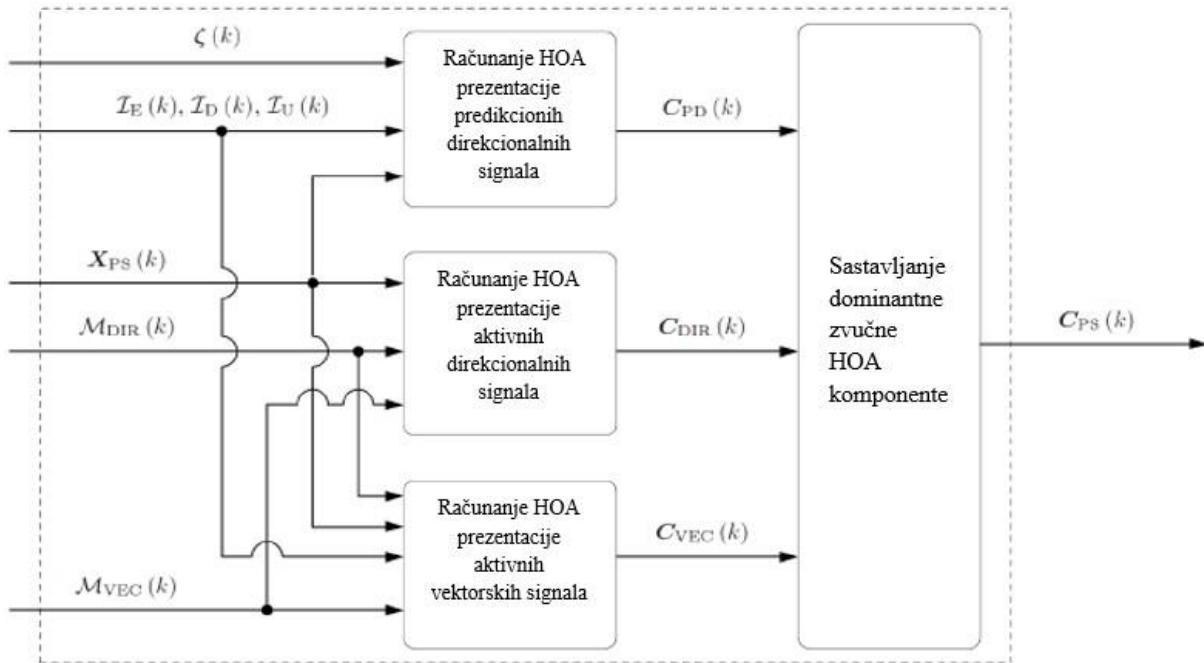
$$C_{I,AMB,n}(k) = [C_{I,AMB,n}(k, 1) \quad C_{I,AMB,n}(k, 2) \quad \dots \quad C_{I,AMB,n}(k, L)] \quad , \quad n = 1, \dots, O.$$

Vrednosti odbiraka prezentacije međuzavisnosti ambijentalnih HOA komponenti su dobijene na sledeći način:

$$C_{I,AMB,n}(k, l) = \begin{cases} y_i(k, l) & , \exists i \in \{1, \dots, I\} , v_{AMB,ASSIGN_i}(k) = n \\ 0 & , \text{inače} \end{cases}$$

3.3 Sinteza dominantnog zvuka

Sinteza dominantnog zvuka predstavlja kreiranje okvira $C_{ps}(k)$ HOA prezentacije od komponente dominantnog zvuka iz okvira $X_{ps}(k)$ svih signala dominantnog zvuka koristeći skup n-torki $\mathcal{M}_{DIR}(k)$ i $\mathcal{M}_{VEC}(k)$, skup predikcionih parametara $\zeta(k)$, i skupove $\mathcal{I}_E(k)$ $\mathcal{I}_D(k)$ i $\mathcal{I}_U(k)$. Na slici 3.1 je prikazano kako se u četiri koraka obrade dolazi do sinteze.



Slika 3.1 Sinteza dominantnog zvuka

3.3.1 Računanje HOA prezentacije aktivnih direkcijskih signalata

Da bi se izbegli artefakti usled promena smerova između uzastopnih okvira, računanje HOA prezentacije iz direkcijskih signalata se zasniva na konceptu preklopljenog dodavanja (engl. *overlap add*). Stoga, HOA prezentacija $C_{DIR}(k)$ aktivnih direkcijskih signalata se računa

kao suma komponenti postepenog stišavanja (engl. *fade out*) i postepenog pojačavanja (engl. *fade in*):

$$C_{DIR}(k) = C_{DIR,OUT}(k) + C_{DIR,IN}(k)$$

Da bi se dve individualne komponente sračunale, u prvom koraku okviri trenutnog signala za indikaciju direkcionog signala $d \in \mathcal{I}_{DIR}(k_1)$ i okvir indeksa direkcionog signala k_2 su definisani kao:

$$C_{DIR,I}^{(d)}(k_1; k_2) = \Psi^{(N,29)} \Big|_{\Omega_{TEST,QUANT,d}(k_1)} \chi_{DIR,d}(k_2),$$

gde $\Psi^{(N,29)}$ predstavlja matricu reda N sa unapred definisanim smerovima $\Omega_n^{(29)}$, $n = 1, \dots, Q = (29 + 1)^2 = 900$, i $\Psi^{(N,29)}|_q$ označava q-ti vektor kolone od $\Psi^{(N,29)}$.

Vrednosti odbiraka postepeno stišavajućih i pojačavajućih HOA komponenti su predstavljeni preko:

$$c_{DIR,OUT,i}(k, l) = \sum_{d \in \mathcal{I}_{DIR,NZ}(k-1)} c_{DIR,I,i}^{(d)}(k-1; k, l) \cdot \begin{cases} w_{DIR}(L+l) & , d \in \mathcal{I}_{DIR}(k) \\ w_{VEC}(L+l) & , d \in \mathcal{I}_{VEC}(k) \\ 1 & , \text{inače} \end{cases}$$

$$c_{DIR,OUT,i}(k, l) = \sum_{d \in \mathcal{I}_{DIR,NZ}(k)} c_{DIR,I,i}^{(d)}(k; k, l) \cdot \begin{cases} w_{DIR}(l) & , d \in \mathcal{I}_{DIR}(k-1) \cup \mathcal{I}_{VEC}(k-1) \\ 1 & , \text{inače} \end{cases},$$

gde $\mathcal{I}_{DIR,NZ}(k)$ označava skup prvih elemenata od $\mathcal{M}_{DIR}(k)$ matrice gde je korespondentni drugi element različit od nule. *Fading* trenutnih HOA prezentacija za preklopljeno dodavanje operacija je izvršen uz pomoć dva različita *fading* prozora:

$$w_{DIR} = [w_{DIR}(1) \quad w_{DIR}(2) \quad \dots \quad w_{DIR}(2L)]$$

$$w_{VEC} = [w_{VEC}(1) \quad w_{VEC}(2) \quad \dots \quad w_{VEC}(2L)]$$

čiji su elementi definisani uz pomoć:

$$w_{DIR}(l) = \frac{1}{2} \left[1 - \cos \left(2\pi \frac{l-1}{2L} \right) \right]$$

$$w_{VEC}(l) = \begin{cases} \frac{l-1}{L_{IP}-1} & , 1 \leq l \leq L_{IP} \wedge \text{SpatialInterpolationMethod} = 0 \\ \frac{1}{2} \left[1 - \cos \left(2\pi \frac{l-1}{2L_{IP}} \right) \right] & , 1 \leq l \leq L_{IP} \wedge \text{SpatialInterpolationMethod} = 1 \\ \frac{1}{L_{IP}-1} & , L_{IP}+1 \leq l \leq L \\ 1 - \frac{l-L-1}{L_{IP}-1} & , L+1 \leq l \leq L+L_{IP} \wedge \text{SpatialInterpolationMethod} = 0 \\ 1 - \frac{1}{2} \left[1 - \cos \left(2\pi \frac{l-L-1}{2L_{IP}} \right) \right] & , L+1 \leq l \leq L+L_{IP} \wedge \text{SpatialInterpolationMethod} = 1 \\ 0 & , L+L_{IP}+1 \leq l \leq 2L \end{cases}$$

gde su vrednosti L_{IP} date u tabeli 3.1.

L	CodedSpatialInterpolationTime							
	0	1	2	3	4	5	6	7
768	0	32	64	128	256	384	512	768
1024	0	64	128	256	384	512	768	1024
2048	0	128	256	512	768	1024	1536	2048
4096	0	256	512	1024	1536	2048	3072	4096

Tabela 3.1 Vrednosti L_{IP} u zavisnosti od vrednosti CodedSpatialInterpolationTime

3.3.2 Računanje HOA prezentacije predikcionih direkcijskih signala

Skup parametara $\zeta(k) = \{p_{TYPE}(k), P_{IND}(k), P_{Q,F}(k)\}$ vezan je za prostornu predikciju direkcijskih signala koji se sastoje od sledećih komponenti:

- Vektor $p_{TYPE}(k)$, čiji elementi $p_{TYPE,n}(k), n = 1, \dots, O$ ukazuju da li je za n-ti smer $\Omega_n^{(N)}$ i k-ti i $(k + 1)$ -i okvir izvršena predikcija ili ne. Ukoliko jeste, onda oni takođe ukazuju na vrstu predikcije. Značenje elemenata je sledeće:
- Matrica $P_{IND}(k)$, čiji elementi $p_{IND,d,n}(k), d = 1, \dots, D_{PRED}, n = 1, \dots, O$ predstavljaju indekse iz kojih direkcijski signali, predikcija smera $\Omega_n^{(N)}$ i k-ti i $(k + 1)$ -i okvir treba da budu izvedeni. Ukoliko nema izvedbe predikcije smera $\Omega_n^{(N)}$, odgovarajuća kolona matrice $P_{IND}(k)$ se onda sastoji od nula. Ukoliko se manje od D_{PRED} direkcijskih signala koristi za predikciju smera $\Omega_n^{(N)}$, elementi manjeg značaja u n-toj koloni $P_{IND}(k)$ matrice su takođe nule.
- Matrica $P_{IND}(k)$, je matrica koja poseduje odgovarajuće kvantizovane faktore predikcije $p_{Q,F,d,n}(k), d = 1, \dots, D_{PRED}, n = 1, \dots, O$.

Treba primetiti da su parametri predikcije $\zeta(k)$ povezani sa okvirima k i $(k + 1)$. Uz to, uvedena je sledeća zavisna jednačina:

$$b_{ACT}(k) = \begin{cases} 1 & , \exists n \text{ tako da je } p_{TYPE,n}(k) = 0 \\ 0 & , \text{ inače} \end{cases}$$

koja ukazuje na to da li se treba izvršiti predviđanje vezano za okvire k i $k + 1$. Kvantizovani faktori predikcije $p_{Q,F,d,n}(k), d = 1, \dots, D_{PRED}, n = 1, \dots, O$, su dekvantizovani da bi se obezbedili pravi faktori predikcije:

$$p_{F,d,n}(k) = \left(p_{Q,F,d,n}(k) + \frac{1}{2} \right) \cdot 2^{-Bsc+1}$$

Računanje predikcije HOA komponenti se prikazuje preko direkcijskih signala (npr. *general plane wave functions*) koji potiču iz predefinisanih smerova $\Omega_n^{(N)}, n = 1, \dots, O$ i transformacije ove prezentacije u HOA prezentaciju. Ovi direkcijski signali se predviđaju na osnovu dominantnih signala zvuka $X_{ps}(k)$ koristeći skup parametara $\zeta(k)$. Računanje predviđenih direkcijskih signala se zasniva na konceptu preklopljene dodavanja da bi se izbegli artefakti usled promene parametara predikcije između uzastopnih okvira. Stoga, k-ti okvir predviđenog direkcijskog signala, predstavljen preko $X_{PD}(k)$, je računat kao sumu komponenti postepenog stišavanja i postepenog pojačavanja:

$$X_{PD}(k) = X_{PD,OUT}(k) + X_{PD,IN}(k).$$

Vrednosti odbiraka $x_{PD,OUT,n}(k, l)$ i $x_{PR,IN,n}(k, l)$, $n = 1, \dots, O, l = 1, \dots, L$, postepeno stišanih i postepeno pojačanih predviđenih direkcijskih signala se računaju uz pomoć sledećih formula:

$$x_{PD,OUT,n}(k, l) = w_{DIR}(L + l) \cdot \begin{cases} 0 & , p_{TYPE,n}(k - 1) = 0 \\ \sum_{d=1}^{D_{PRE}} p_{F,d,n}(k - 1) \cdot x_{PS,p_{IND,d,n}(k-1)}(k, l) & , p_{TYPE,n}(k - 1) = 1 \end{cases}$$

$$x_{PD,IN,n}(k, l) = w_{DIR}(l) \cdot \begin{cases} 0 & , p_{TYPE,n}(k) = 0 \\ \sum_{d=1}^{D_{PRE}} p_{F,d,n}(k) \cdot x_{PS,p_{IND,d,n}(k)}(k, l) & , p_{TYPE,n}(k) = 1 \end{cases}$$

U narednom koraku, predviđeni direkcijski kanali se prebacuju u HOA domen uz pomoć:

$$C_{PD,I}(k) = \Psi^{(N,N)} \cdot X_{PD}(k),$$

pri čemu $\Psi^{(N,N)}$ predstavlja matricu reda N, uzimajući u obzir predefinisane smerove $\Omega_n^{(N)}, n = 1, \dots, O$. Sa pretpostavkom da će preliminarna HOA prezentacija $C_{PD,I}(k)$ predviđenih direkcijskih signala biti predstavljena pomoću odbiraka:

$$C_{PD,I}(k) = \begin{bmatrix} c_{PD,I_1}(k, 1) & \cdots & c_{PD,I_1}(k, L) \\ \vdots & \ddots & \vdots \\ c_{PD,I_O}(k, 1) & \cdots & c_{PD,I_O}(k, L) \end{bmatrix},$$

odbirci krajnjeg izlaza HOA prezentacije

$$C_{PD}(k) = \begin{bmatrix} c_{PD,1}(k, 1) & \cdots & c_{PD,1}(k, L) \\ \vdots & \ddots & \vdots \\ c_{PD,O}(k, 1) & \cdots & c_{PD,O}(k, L) \end{bmatrix},$$

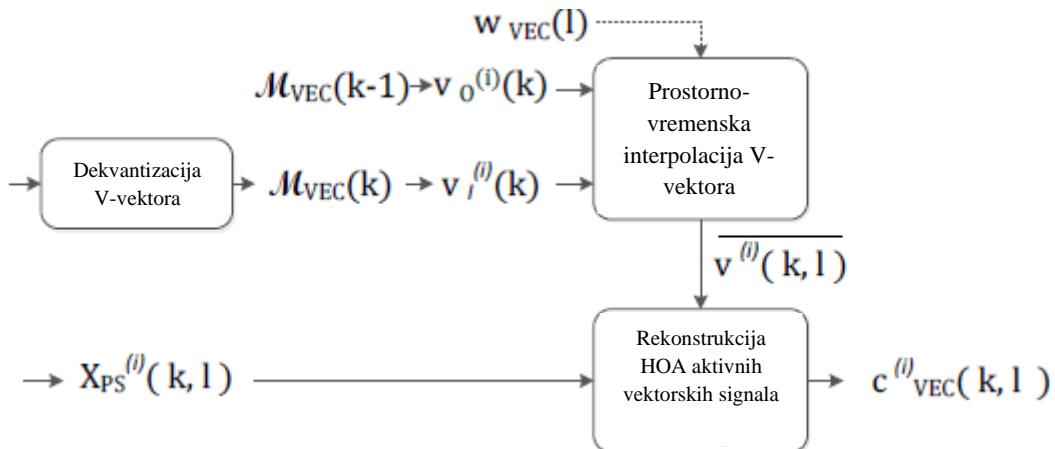
predviđenog direkcijskog signala se računaju na sledeći način:

$$c_{PD,n}(k, l) = \begin{cases} 0 & , n \in \mathcal{I}_U(k) \\ c_{PD,I,n}(k, l) \cdot w_{DIR}(l) & , n \in \mathcal{I}_D(k) \wedge b_{ACT}(k-1) = 1 \\ c_{PD,I,n}(k, l) \cdot w_{DIR}(L+l) & , n \in \mathcal{I}_E(k) \wedge b_{ACT}(k) = 1 \\ c_{PD,I,n}(k, l) & , \text{ inače} \end{cases}, \quad n = 1, \dots, O, l = 1, \dots, L.$$

3.3.3 Računanje HOA prezentacije aktivnih vektorskih signala

Računanje HOA prezentacije vektorskog signala se zasniva na konceptu interpolacije V-vektora koji opisuje distribuciju komponenti dominantnog zvuka u prostoru (npr. smerovi, oblici i širine). S obzirom da V-vektor već sadrži sve bitne prostorne informacije o komponentama dominantnog zvuka, prostrona predikcija se ne koristi za vektorske signale. Interpolacija V-vektora se sprovodi tokom vremena. Ova prostorno-vremenska interpolacija osigurava kontinuiranu i nesmetanu evoluciju zvučnog polja izvan granica okvira, što pruža perceptualno transparentan kvalitet. Na ovaj način, HOA signal se rekomponuje iz originalne dekompozicije koja se sastoji iz interpoliranih vektora i odgovarajućih dominantnih signala.

Kao što je prikazano na slici 3.2, HOA koeficijenti su preformulisani iz vektorskog signala. Skalarna dekvantizacija se prvo vrši u odnosu na svaki V-vektor da bi se generisala $\mathcal{M}_{VEC}(k)$ matrica, gde se svaki i-ti vektor trenutnog okvira može predstaviti kao $v_I^{(i)}(k)$. V-vektori se izdvajaju iz HOA koeficijenata inverzne linearne transformacije. Prostorno-vremenska interpolacija se vrši u odnosu na matrice $\mathcal{M}_{VEC}(k)$ i $\mathcal{M}_{VEC}(k-1)$ (što predstavlja V-vektore iz prethodnog okvira sa individualnim vektorima iz $\mathcal{M}_{VEC}(k-1)$ predstavljenim kao $v_O^{(i)}(k)$). Metoda interpolacije se kontroliše od strane $w_{VEC}(l)$. Nakon interpolacije, i-ti interpolirani V-vektor $(\overline{v^{(i)}(k, l)})$ se množi sa i-tim dominantnim signalom $x_{PS}^{(I)}(k, l)$, da bi se dobila i-ta kolona HOA prezentacije $(c_{VEC}^{(i)}(k, l))$. Vektori kolone se zatim sumiraju da bi stvorili HOA prezentaciju vektorskog signala. Na ovaj način se HOA koeficijenti rekonstruišu za okvir.



Slika 3.2 Računanje vektorskog signala

3.3.4 Prostorno-vremenska interpolacija V-vektora

Interpolaciona matrica $\overline{V^{(i)}(k)} = [\overline{v^{(i)}(k, 1)}, \overline{v^{(i)}(k, 2)}, \dots, \overline{v^{(i)}(k, L)}]$ se prvo računa za svaki indeks $i \in \mathcal{I}_{VEC}(k) \cup \mathcal{I}_{VEC}(k - 1)$ vektorskog signala koji je aktivan u k-tom ili (k - 1)-om okviru. Njene kolone $\overline{v^{(i)}(k, l)}$ za svaki odbirak okvira $1 \leq l \leq L$ predstavljaju interpolirani vektor koji je definisan kao:

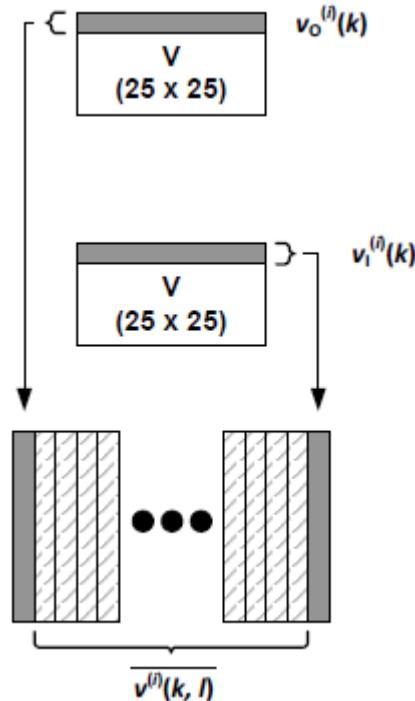
$$\overline{v^{(i)}(k, l)} = w_I^{(i)}(l)v_I^{(i)}(k) + (1 - w_I^{(i)}(l))v_O^{(i)}(k).$$

U jednačini iznad, $v_I^{(i)}(k)$ i $v_O^{(i)}(k)$ predstavljaju vektore između kojih se odvija interpolacija i definisani su kao:

$$v_I^{(i)}(k) = \begin{cases} v^{(i)}(k) & , i \in \mathcal{I}_{VEC}(k) \\ 0 & , \text{ inače} \end{cases}$$

$$v_O^{(i)}(k) = \begin{cases} v^{(i)}(k - 1) & , i \in \mathcal{I}_{VEC}(k - 1) \\ 0 & , \text{ inače} \end{cases}$$

gde 0 predstavlja nulti vektor. Ovo se može videti na slici 3.3.



Slika 3.3 Interpolacija V-vektora

Funkcija interpolacije $w_I^{(i)}(l)$ je definisana preko:

$$w_I^{(i)}(l) = \begin{cases} w_{DIR}(l) & , i \in \mathcal{I}_{VEC}(k - 1) \cap \mathcal{I}_{DIR}(k) \\ w_{VEC}(l) & , \text{ inače} \end{cases}$$

gde je $w_{VEC}(l)$:

$$w_{VEC}(l) = \begin{cases} \frac{l-1}{L_{IP}-1}, & 1 \leq l \leq L_{IP} \text{ if } \text{SpatialInterpolationMethod} = 0 \\ \frac{1}{2} \left[1 - \cos \left(2\pi \frac{l-1}{2L_{IP}} \right) \right], & 1 \leq l \leq L_{IP} \text{ if } \text{SpatialInterpolationMethod} = 1, \\ 1, & L_{IP} + 1 \leq l \leq L \end{cases}$$

a L_{IP} se čita iz Tabele 3.1.

HOA prezentacija $C_{VEC}^{(i)}(k) = [c_{VEC}^{(i)}(k, 1) \ c_{VEC}^{(i)}(k, 2) \ \dots \ c_{VEC}^{(i)}(k, L)]$ za svaki i-ti vektorski signal $x_{PS,i}(k)$, $i \in \mathcal{I}_{VEC}(k) \cup \mathcal{I}_{VEC}(k-1)$, je matrica dimenzija $(N+1)^2 \times L$, čije kolone su date sa:

$$c_{VEC}^{(i)}(k, l) = \overline{\nu^{(i)}(k, l)} x_{PS,i}(k, l).$$

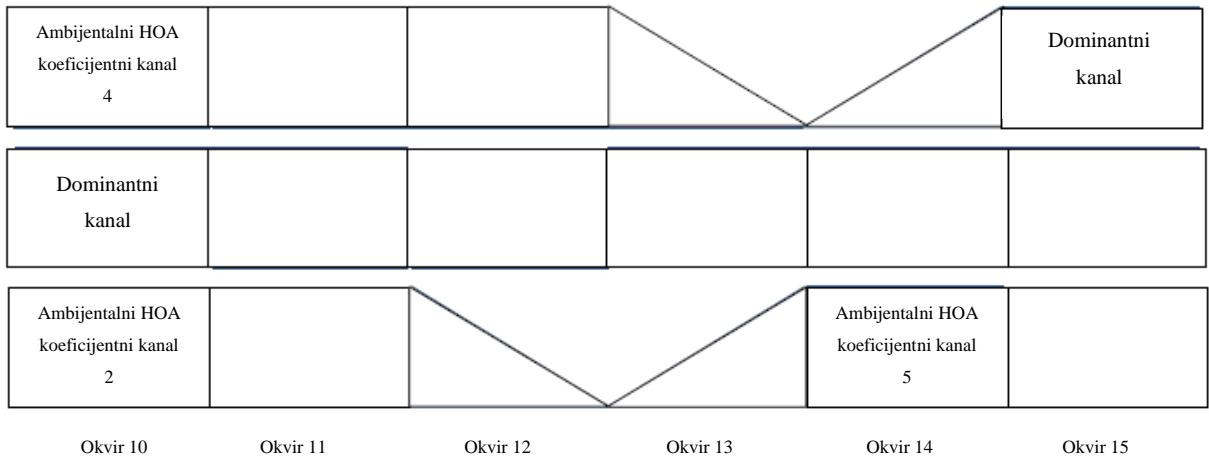
Celokupna HOA prezentacija vektorskog signala se može izračunati sabiranjem HOA udela iz svakog zasebnog vektorskog signala:

$$\tilde{C}_{VEC}(k) = \sum_{i \in \mathcal{I}_{VEC}(k) \cup \mathcal{I}_{VEC}(k-1)} C_{VEC}^{(i)}(k).$$

U slučaju da ova suma ima vrednost 1, izvršiće se naredne operacije:

- Ukoliko postoje koeficijenti sekvene ambijentalne HOA komponente koje se eksplisitno dodatno prenose i postepeno pojačavaju tokom k-tog okvira (od kojih su indeksi sadržani u skupu $\mathcal{I}_E(k)$), odgovarajući koeficijenti sekvenci HOA prezentacije $\tilde{C}_{VEC}(k)$ se moraju postepeno smanjiti koristeći deo prozora za postepeno smanjivanje w_{DIR} .
- Ukoliko postoje sekvene ambijentalne HOA komponente koje se eksplisitno dodatno prenose i postepeno smanjuju tokom k-tog okvira (od kojih su indeksi sadržani u skupu $\mathcal{I}_D(k)$), odgovarajući koeficijenti sekvenci HOA prezentacije $\tilde{C}_{VEC}(k)$ se moraju postepeno pojačati koristeći deo prozora za postepeno pojačavanje w_{DIR} .
- Konačna HOA prezentacija vektorskog signala se dobija uz pomoć:

$$c_{VEC,n}(k, L) = \begin{cases} \tilde{c}_{VEC,n}(k, L) \cdot w_{DIR}, & n \in \mathcal{I}_D(k) \wedge \Sigma = 1 \\ \tilde{c}_{VEC,n}(k, L) \cdot w_{DIR}(L+l), & n \in \mathcal{I}_E(k) \wedge \Sigma = 1 \\ \tilde{c}_{VEC,n}(k, L), & \text{inače} \end{cases}, \quad 1 \leq l \leq L.$$



Slika 3.4 Primer postepenog pojačavanja (engl. *Fade-In*) i postepenog smanjivanja (engl. *Fade-Out*) komponenti

Na slici 3.4 prikazano je kako se ambijentalni HOA koeficijentni kanali 4, 2, 5 i dominantni signalni kanal postepeno pojačavaju i postepeno smanjuju. Ambijentalni HOA koeficijentni kanal 4 se nalazi u periodu tranzicije tokom okvira 13, sa dominantnim kanalom koji se nalazi u tranziciji tokom okvira 14, da bi izmenio ambijentalni HOA koeficinetni kanal 4 u bitskom toku. Drugi dominantni kanal ostaje nepromenjen od 10. do 15. okvira. Ambijentalni HOA koeficijentni kanal 2 se nalazi u tranziciji tokom okvira 12, sa ambijentalnim HOA koeficijentnim kanalom 5 koji je u tranziciji tokom okvira 13, da bi izmenio ambijentalni HOA koeficijentni kanal 2 u bitskom toku.

Kompletan dominantni zvučni HOA komponenta $C_{ps}(k)$ se dobija kao zbir HOA komponenti $C_{DIR}(k)$ direkcionih signala, HOA komponenti $C_{PD}(k)$ predviđenih direkcionih signala i HOA komponenti vektorskih signala $C_{VEC}(k)$:

$$C_{ps}(k) = C_{DIR}(k) + C_{PD}(k) + C_{VEC}(k).$$

Potrebito je napomenuti da su HOA komponente $C_{DIR}(k)$, $C_{PD}(k)$ i $C_{VEC}(k)$ jednake nuli, u slučaju da nisu obrađene za k-ti okvir.

3.4 Ambijentalna sinteza

Ambijentalni HOA komponentni okvir $C_{AMB}(k)$ je sastavljen na sledeći način:

$$C_{AMB}(k) = \begin{bmatrix} c_{AMB,1}(k) \\ c_{AMB,2}(k) \\ \vdots \\ c_{AMB,o}(k) \end{bmatrix}$$

gde je:

$$c_{AMB,n}(k) = [c_{AMB,n}(k, 1) \ c_{AMB,n}(k, 2) \ \dots \ c_{AMB,n}(k, L)] \quad , \ n = 1, \dots, O.$$

Prvi O_{MIN} koeficijenti ambijentalne HOA komponente su sastavljeni od:

$$\begin{bmatrix} c_{AMB,1}(k) \\ c_{AMB,2}(k) \\ \vdots \\ c_{AMB,O_{MIN}}(k) \end{bmatrix} = \Psi^{(N_{MIN}, N_{MIN})} \cdot \begin{bmatrix} c_{I,AMB,1}(k) \\ c_{I,AMB,2}(k) \\ \vdots \\ c_{I,AMB,O_{MIN}}(k) \end{bmatrix},$$

gde $\Psi^{(N_{MIN}, N_{MIN})}$ predstavlja matricu reda N_{MIN} s obzirom na predefinisane smerove $\Omega_n^{(N_{MIN})}, n = 1, \dots, O_{MIN}$. Množenje po modulu matrice $\Psi^{(N_{MIN}, N_{MIN})}$ predstavlja inverznu prostornu transformaciju koja je namenjena za invertovanje prostorne transformacije primenjene u koderu, za dekorelaciju prvog O_{MIN} koeficijenta ambijentalne HOA komponente. Vrednosti odbiraka preostalih koeficijenata ambijentalne HOA komponente su dobijaju iz:

$$c_{AMB,n}(k, l) = c_{I,AMB,n}(k, l) \quad , \quad O_{MIN} < n \leq O.$$

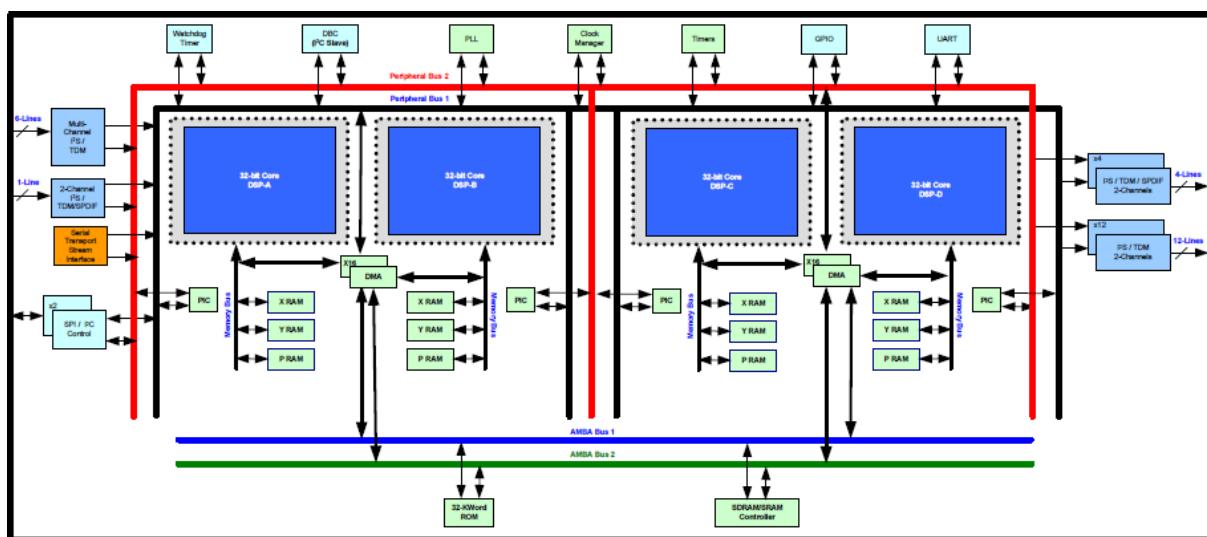
3.5 HOA kompozicija

Na samom kraju, dekodovani HOA okvir $C(k)$ se računa kao:

$$C(k) = C_{AMB}(k) + C_{DIR}(k) + C_{PD}(k) + C_{VEC}(k) = C_{AMB}(k) + C_{PS}(k). [2]$$

4. Ciljna DSP platforma

Za realizaciju ovog projekta bilo je neophodno implementirati asemblerski kod po ugledu na dati referentni C++ kod. Cilj zadatka je bio da se omogući da se realizovani asemblerski kod izvršava na zadatoj ciljnoj platformi. Za ovaj projekat korišćena je razvojna ploča sa oznakom *CDB49x* kompanije *Cirrus Logic* na kojoj se nalazi DSP *DC49844-64*. [12]



Slika 4.1 Blok dijagram procesora DC49844

Kao što je prikazano na slici 4.1, ovaj DSP se sastoji od 4 32-bitna jezgra (A, B, C i D) sa aritmetikom u nepokretnom zarezu. Njegova arhitektura se zasniva na Harvard arhitekturi, što samim tim znači da postoji odvojena programska memorija i odvojena memorija za podatke. Svako od ova 4 jezgra sadrže 32-bitne memorijske zone za podatke X i Y. Unutar jezgara postoji i L memorija, koja predstavlja 64-bitnu prezentaciju X i Y memorijskih zona. Takođe, postoji i programska memorija koja je označena simbolom P. Ovakvom organizacijom memorije omogućeno je da se istovremeno pristupa i instrukcijama i podacima iz memorije. Time se

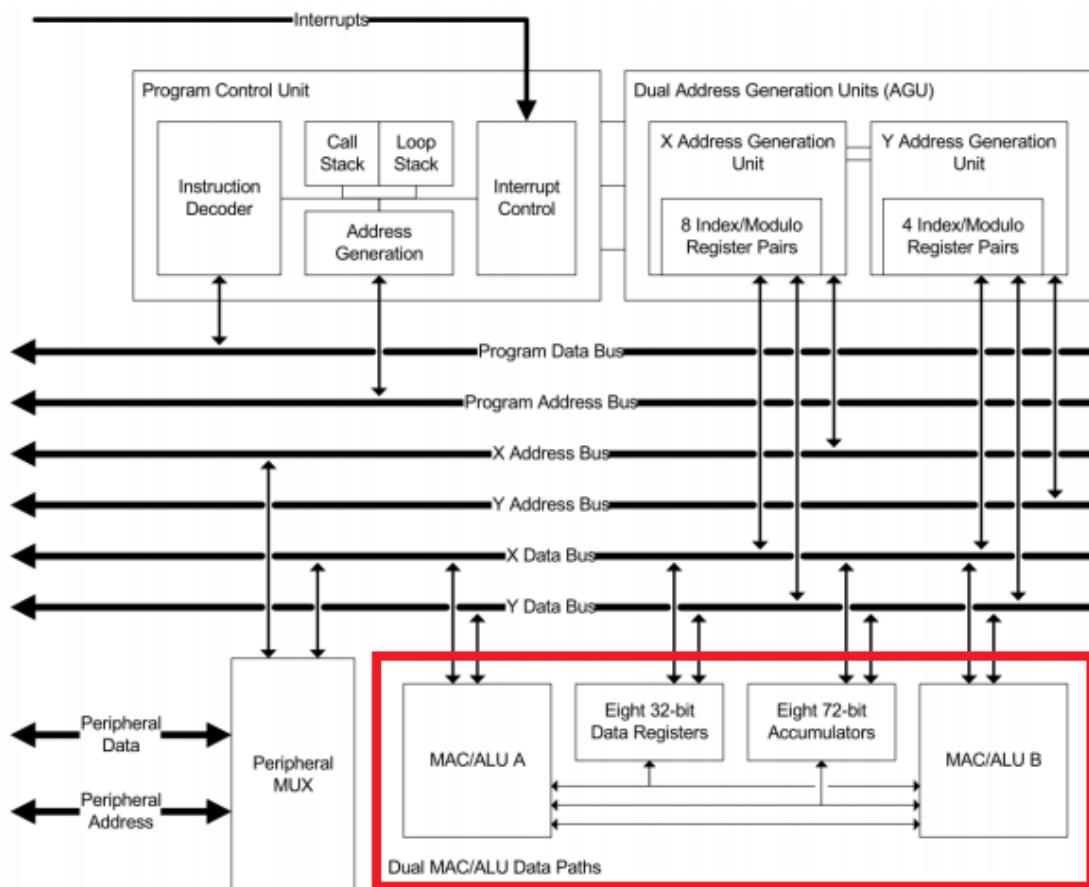
doprinosi znatnom ubrzavanju rada procesora, kao i velikoj iskorišćenosti resursa datog procesora, koji može da izvršava čak 6 instrukcija istovremeno.

	X memorija [kWord]	Y memorija [kWord]	P memorija [kWord]	Procesorsko vreme [MIPS]
Jezgra A,B,C,D	60	60	60	300

Tabela 4.1 Raspoloživa memorija i procesorsko vreme po jezgrima procesora *DC49844*

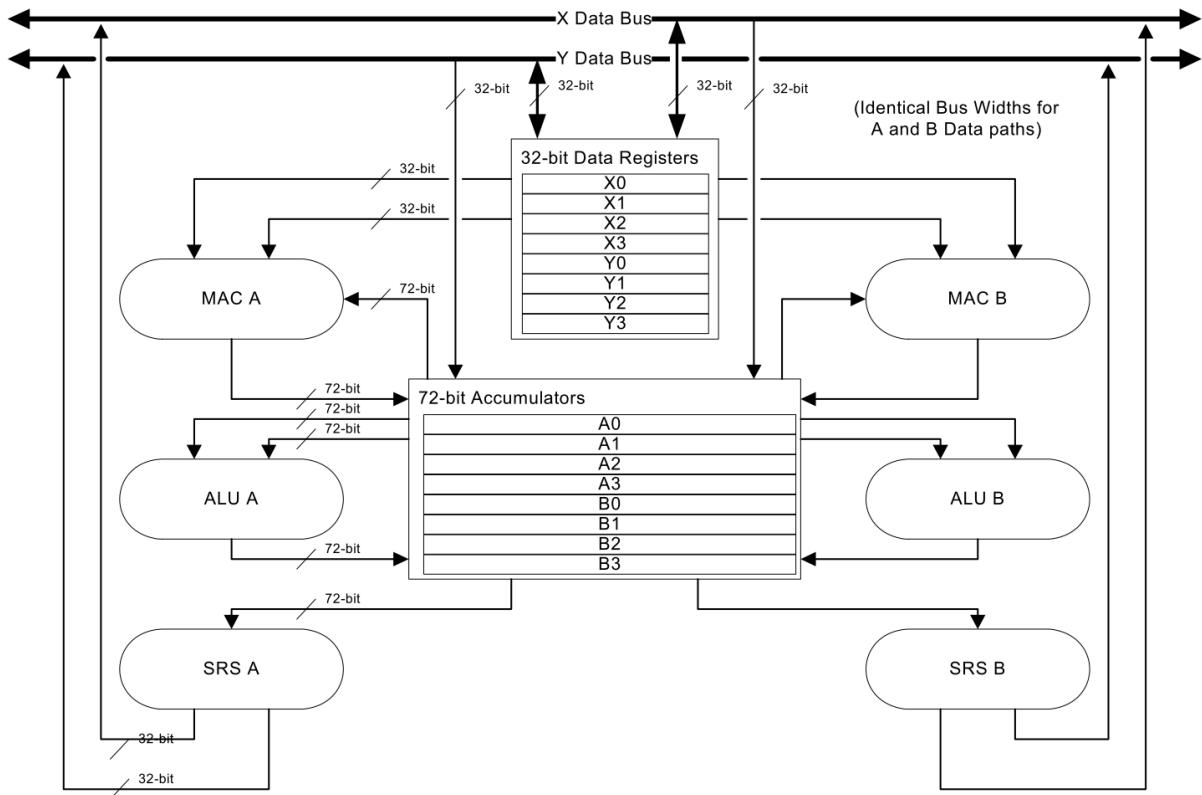
Kako je prikazano u tabeli 4.1, svako jezgro procesora ima raspoloživu memoriju od 60k 32-bitnih reči, kao i procesorsku moć od 300 miliona instrukcija u jednoj sekundi(engl. *MIPS*).

Na slici 4.2 dat je blok dijagram arhitekture *Crystal* procesora, čije je jezgro zasnovano na aritmetici u nepokretnom zarezu i čije su visoke performanse dobijene postizanjem visokog stepena paralelizma.



Slika 4.2 Blok dijagram *Crystal 32 DSP* arhitekture

Na slici 4.3 je prikazan blok dijagram *Crystal DSP* jezgra. To jezgro sadrži jedinicu za kontrolu toka programa, kao i paralelne tokove podataka (A, B) i paralelne jedinice za generisanje adresa (AGU).



Slika 4.3 Blok dijagram *Crystala 32 DSP* jezgra

U svakom od jezgara DSP procesora nalazi se četiri X (x_0, x_1, x_2, x_3) i četiri Y (y_0, y_1, y_2, y_3) 32-bitna regista za podatke, osam ($a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$) 72-bitnih akumulatora koji se sastoje od 3 pod regista sa oznakama: *Guard* (8-bitni), *High* (32-bitni) i *Low* (32-bitni), pri čemu možemo pristupiti svakom od njih posebno. Takođe jezgro sadrži i dvanaest ($i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}, i_{11}$) indeksnih regista, gde svaki indeksni registar ima odgovarajući modulo registar ($nm_0, nm_1, nm_2, nm_3, nm_4, nm_5, nm_6, nm_7, nm_8, nm_9, nm_{10}, nm_{11}$) koji služe za različite režime adresiranja. Svaka putanja podataka ima i MAC (engl. *Multiply and Accumulate*) – jedinica za množenje i akumuliranje, ALU (engl. *Arithmetic and Logical Unit*) – aritmetičko-logička jedinica i SRS (engl. *Shift, Round, Saturate*) – jedinica za pomeranje, zaokruživanje i ograničavanje opsega. MAC jedinica se koristi za množenje i sabiranje, pri čemu se rezultati ovih operacija smeštaju u akumulatorske registe. ALU jedinica služi za operacije sabiranja i oduzimanja, kao i za logičke operacije. SRS jedinica se koristi da bi se 72-bitni podatak koji se nalazi u akumulatorskim registrima mogao upisati u 32-bitne registre za podatke ili poslati na 32-bitnu X ili Y magistralu. Ova jedinica je dobila ime po samom redosledu operacija koji obavlja – Pomeranje (engl. *Shift*), Zaokruživanje (engl. *Round*) i Zasićenje, odnosno ograničavanje opsega (engl. *Saturate*). [13]

5. Programsко rešenje

U ovom poglavlju će biti opisan postupak realizacije HOA renderera, što je bio zadatak ovog rada. Na početku poglavlja dat je opis referentnog koda, a nakon toga sledi postupak implementacije funkcija. Na kraju se govori o razvojnom okruženju, koje se koristi za proveru rezultata rada.

5.1 Analiza referentnog koda

Na početku implementacije, neophodno je bilo upoznati se sa delom referentnog koda, koji je napravljen od strane nemačkog instituta za unapređenje primenjenih istraživanja *Fraunhofer IIS*. Referentna implementacija je realizovana upotrebom C++ programskog jezika i predstavlja jednu od implementacija MPEG-H standarda. Pre početka realizacije pojedinačnih funkcija, izvršena je detaljna analiza svake funkcije, sa naglaskom na razumevanje algoritma. To je bilo neophodno, da bi se mogao uspešno napisati asemblerски kod koji ima istu funkcionalnost kao i referentni C++ kod.

5.2 Inicijalizacija

Pre početka izvršavanja obrade potrebno je da svi parametri koji su značajni za HOA obradu budu inicijalizovani na početne vrednosti.

Pri pokretanju programa, vrši se inicijalizacija celokupnog sistema, pri čemu je najbitnija priprema polja svake od klase za obradu. Ulazni i izlazni baferi se prosleđuju svakoj od funkcija za obradu i predstavljaju dvostruki pokazivač na podatak koji je tipa `FIXP_DB`.

Za polja klase u asembleru je zauzet deo memorije i njima može da se pristupa preko instanci klase. Svakom polju se dodeljuje određena veličina, koja je definisana u referentnom kodu, a kada želimo da pristupimo nekom polju, zapravo pristupamo početnoj adresi polja.

Takođe, postoji odvojeni deo memorije i za lokalne promenljive, odnosno za promenljive koje se nalaze u memoriji sve dok se ne završi određeni blok za obradu. Taj deo memorije je predstavljen simbolom `X_BX_mpegh_dec_TMP_ScratchMemory`, ukoliko se lokalne promenljive smeštaju u X memoriju, odnosno `X_BY_mpegh_dec_TMP_ScratchMemory`, ukoliko se lokalne promenljive smeštaju u Y memoriju. Ova memorija se deli na podnivoe, pri čemu funkcije koje se ne odvijaju u istom vremenskom periodu dele ovaj memorijski prostor, tj. njihove lokalne promenljive se smeštaju na iste memoriske lokacije.

5.3 Pregled korišćenih polja i metoda klasa

U ovom delu poglavlja biće izložena polja i metode klasa koji su uspešno implementirani tokom izrade projekta.

5.3.1 Klasa HOADecoder

Najvažnija klasa koja je implementirana tokom izrade ovog projekta je HOADecoder, pa su samim tim i njena polja i metode najviše korišćene u ovom projektu.

Polja ove klase predstavljaju glavne parametre za HOA obradu, od kojih su najčešće korišćena:

- ❖ `m_numOfHoaCoeffs` – broj HOA koeficijenata;
- ❖ `m_numHOATransportChannels` – broj HOA kanala za transport;
- ❖ `m_frameSize` – veličina okvira;
- ❖ `m_frameParams` – instanca klase FrameParams;
- ❖ i instance klase koje služe za glavni deo HOA obrade.

Klasa HOADecoder takođe sadrži i metodu `processSamples` unutar koje se pozivaju sve metode klasa koje su bitne za HOA obradu (`m_invGainControl.process`, `m_ambSynth.process`, `m_dirBasedPredomSoundSynth.process`, `m_vecBasedPredomSoundSynth.process`) i na kraju svake metode se proverava da li je došlo do neke greške.

5.3.2 Klasa FrameParams

Polja klase `FrameParams` se sastoje od nekoliko instanci klase `HOAIndexSetStatic`, jednog niza pokazivača i nekoliko nizova, od kojih su nam najvažniji:

- ❖ `m_exceptions` – izuzeci;
- ❖ `m_exponents` – eksponenti ili stepeni;
- ❖ `m_ambAssignVec` – vektor ambijentalne dodele;
- ❖ `m_activeDirs` – aktivni pravci.

Klasa `FrameParams` sadrži samo jednu metodu i to je metoda `init` kojom se inicijalizuju sva polja ove klase.

5.3.3 Klasa InverseGainControl

Klasa `InverseGainControl` od polja sadrži `m_numHOATransportChannels` sadrži broj prenosnih HOA kanala, `m_prevGainExp` koji predstavlja niz u kom se nalaze prethodni eksponenti pojačanja i `m_window` koji pokazuje na niz podataka dobijenih prozorskom funkcijom.

Sa druge strane, od metoda poseduje `init` metodu, uz pomoć koje se inicijalizuju potrebni parametri. Zatim poseduje glavnu `process` metodu, u kojoj se za svaki prenosni kanal na početku računa pojačanje u zavisnosti od eksponenta ili od toga da li postoji neki izuzetak i onda se to pojačanje primenjuje na ulazne odbirke podataka. Nakon toga se nad ulaznim odbircima primenjuje prozorska funkcija. Implementirane su i metode `update`, u kojoj se vrši ažuriranje eksponenata i `setPrevGain` u kojoj se postavlja određena vrednost na neki element u nizu prethodnih eksponenata.

5.3.4 Klasa AmbienceSynthesis

U klasi `AmbienceSynthesis` se nalaze polja `m_minNumOfCoeffsForAmbHOA` koja služe za čuvanje minimalnog broja koeficijenata za ambijentalnu HOA sintezu, `m_numOfHoaCoeffs` predstavlja promenljivu koja sadrži informaciju o broju HOA koeficijenata i `m_numHOATransportChannels` sadrži broj prenosnih HOA kanala. Takođe, tu se nalazi i matrica za renderovanje koja predstavlja pokazivač na instancu klase `HOAMatrix`, kao i ambijentalnu matricu koja predstavlja instancu klase `HOAMatrixStatic`.

U ovoj klasi se nalazi implementirana metoda `init`, koja služi za inicijalizaciju podataka za obradu, kao i metoda `process`. U metodi `process` se na početku kreira mapa uz pomoć vektora dodele. Nakon toga se u zavisnosti od broja ambijentalnih koeficijenata prave ambijentalni signali od ulaznih odbiraka, koji se kasnije skaliraju brojem koji predstavlja razliku između eksponenta matrice za renderovanje i eksponenta ambijentalne matrice. Ambijentalni signali se nakon skaliranja šalju na dalju obradu u metodi `multSig`, koja pripada klasi `HOAMatrix`.

5.3.5 Klasa DirBasedPredomSoundSynthesis

Klasa `DirBasedPredomSoundSynthesis` sadrži polja `m_fadeInWinDir`, `m_fadeOutWinDir`, `m_fadeOutWinVec`, koji su definisani kao pokazivači na definisan tip `FIXP_SGL`, koji predstavlja tip veličine `short` u nepokretnom zarezu. Pored toga, još jedno od polja je i `m_renderFineModeMatrix`, koje je instanca klase `HOAMatrixStatic`, kao i `m_prevActiveDirs`, niz prethodno aktivnih pravaca.

Implementirane su metode `init` i `process`. U `init` metodi se podaci inicijalizuju na početne vrednosti. U metodi `process` se vrši obrada u zavisnosti od aktivnih pravaca. U prvom

delu se obrada vrši u zavisnosti od prethodno aktivnih pravaca. U slučaju da pravac zadovoljava uslov prave se signali od ulaznih odbiraka. Nakon toga se postavlja niz vrednosti za prozor postepenog stišavanja u zavisnosti od toga da li postoji trenutno aktivni pravac ili u zavisnosti od dominantnih indeksa. Zatim se od matrice za renderovanje u finom režimu pravi nova matrica unutar koje se nalaze informacije o smerovima prostiranja. Ova matrica se zajedno sa signalima i prozorskom funkcijom prosleđuje na dalju obradu u metodu `multSigWin` klase `HOAMatrix`. U drugom delu, obrada se vrši u zavisnosti od trenutno aktivnih pravaca. Najveća razlika u ova dva dela je ta da se u prvom delu pravi prozor postepenog stišavanja u zavisnosti od trenutno aktivnih pravaca, dok se u drugom delu pravi prozor postepenog pojačavanja u zavisnosti od prethodno aktivnih pravaca. Pored `init` i `process` metoda implementirana je i metoda `update` u kojoj se ažuriraju prethodno aktivni pravci.

5.3.6 Klasa VectorBasedPredomSoundSynthesis

Polja klase `VectorBasedPredomSoundSynthesis` koja su od značaja u ovom projektu su:

- ❖ `m_numOfHoaCoeffs` – broj HOA koeficijenata;
- ❖ `m_fadeInWinDir` – prozor za postepeno pojačavanje pravaca;
- ❖ `m_fadeOutWinDir` – prozor za postepeno stišavanje pravaca;
- ❖ `m_fadeInWinVec` – prozor za postepeno pojačavanje vektora;
- ❖ `m_fadeOutWinVec` – prozor za postepeno stišavanje vektora;
- ❖ `m_numVecs` – trenutni broj vektora;
- ❖ `m_prevNumVecs` – prethodni broj vektora;
- ❖ `m_vecIndices` – trenutni vektor indeksa;
- ❖ `m_prevVecIndices` – prethodni vektor indeksa;
- ❖ `m_prevMatrixNonFade` – prethodna matrica bez postepenog stišavanja i pojačavanja;
- ❖ `m_prevMatrixFadeOut` – prethodna matrica postepenog stišavanja;
- ❖ `m_prevMatrixFadeIn` – prethodna matrica postepenog pojačavanja;
- ❖ `m_matrixNonFade` – trenutna matrica bez postepenog stišavanja i pojačavanja;
- ❖ `m_matrixFadeIn` – trenutna matrica postepenog pojačavanja;
- ❖ `m_matrixFadeOut` – trenutna matrica postepenog stišavanja;
- ❖ `m_ambCoeffIndNoFade` – indeksi ne postepeno pojačanih i stišanih ambijentalnih koeficijenata;
- ❖ `m_ambCoeffIndFadeIn` – indeksi postepeno pojačanih ambijentalnih koeficijenata;
- ❖ `i m_ambCoeffIndFadeOut` – indeksi postepeno stišanih ambijentalnih koeficijenata.

Ova klasa sadrži implementirane metode `init` i `process`. Pored njih su implementirane i metode `prepare`, u kojoj se vrši priprema podataka za obradu, i `update` u kojoj se trenutni broj vektora prebacuje u prethodni broj vektora, trenutni vektor indeksa se prebacuje u prethodni vektor indeksa. Metoda `process` je podeljena na dva veća dela i to na doprinos postepenog stišavanja i na doprinos postepenog pojačavanja.

Prvi deo se deli na tri manje celine i izvršava samo ako je prethodni broj vektora veći od nule. U prvoj celini se vrši obrada ambijentalnih koeficijenata na koje se ne primenjuje postepeno stišavanje (engl. fade out) ili pojačavanje (engl. fade in). Ako je uslov za dominantne indekse ispunjen, formira se bafer od ulaznih odbiraka i prozor za obradu. Na kraju se svi obrađeni podaci prosleđuju u metodu `multSigWin` klase `HOAMatrix` na dalju obradu. U sledeće dve celine na sličan način se obrađuju ambijentalni koeficijenti na koje se primenjuju postepeno pojačavanje ili stišavanje.

Drugi deo metode `process` se takođe deli na tri manje celine i on se izvršava jedino u slučaju da je trenutan broj vektora veći od nule. I ovde kao u prvom delu se formiraju bafer ulaznih odbiraka i prozor za obradu i prosleđuju metodi `multSigWin` klase `HOAMatrix`.

5.3.7 Klasa `HOAMatrix`

Klasa `HOAMatrix` predstavlja model matrice za HOA obradu. Kao polja sadrži osnovne elemente matrice, kao što su broj redova, broj kolona, korak pri prolazeњu kroz matricu, podatke i eksponent. Kao metode sadrži neke osnovne metode za rad sa matricama, a pored njih sadrži i dve veoma važne metode za ovaj projekat, a to su metode `multSig` i `multSigWin`. U metodi `multSig` se u zavisnosti od eksponenta matrice svaki ulazni odbirak množi sa određenim elementom matrice i pritom se skalira eksponentom. Metoda `multSigWin` sadrži istu obradu kao i prethodna, sa dodatnom primenom prozorske funkcije.

5.3.8 Klasa `HOAMatrixStatic`

Klasa `HOAMatrixStatic` je preklopljena i jedina razlika u odnosu na prethodnu klasu joj je to što se veličina matrice zadaje prilikom definisanja instance matrice.

5.4 Optimizacija asemblerskog koda

Asemblerski jezik je blizak hardveru i omogućava bolje iskorišćenje resursa namenskih sistema sa ograničenim resursima.[14] DSP za koji je napisan kod ovog projekta ima određena hardverska proširenja. Jedno od takvih proširenja predstavlja hardverska petlja. Hardverska petlja je specijalna hardverska upravljačka konstrukcija koja ponavlja jednu ili više instrukcija određen broj puta. Broj ponavljanja može biti ili neki 16-bitni indeksni registar ili 10-bitna nenegativna konstantna vrednost koja je različita od nule. Glavna prednost zbog koje se

hardverska petlja koristi je brže izvršavanje u odnosu na softversku petlju.[15] Primer hardverske petlje je prikazan na slici 5.2.

Takođe, zbog svoje specifične arhitekture DSP na kojem je rađen projekat podržava i paralelizaciju izvršavanja instrukcija, što znači da se više operacija može izvršavati istovremeno. Ovaj primer paralelizacije se može videti na slici 5.2 u vidu istovremenog smeštanja vrednosti iz akumulatorskog registra u X memoriju na lokaciju koja se nalazi u i2 indeksnom registru i uvećavanje istog indeksnog registra.

```
for (unsigned int idx = 0; idx < m_numHOATransportChannels; idx++)
    m_prevGainExp[idx] -= p_iExponent[idx];
```

Slika 5.1 Izgled *for* petlje u C++ kodu

```
i1 = (HOADecoder_instance.m_frameParams.m_exponents)
i0 = xmem[HOADecoder_instance.m_invGainControl.m_numHOATransportChannels]
i2 = (HOADecoder_instance.m_invGainControl.m_prevGainExp)

a0 = xmem[i1]; i1 += 1

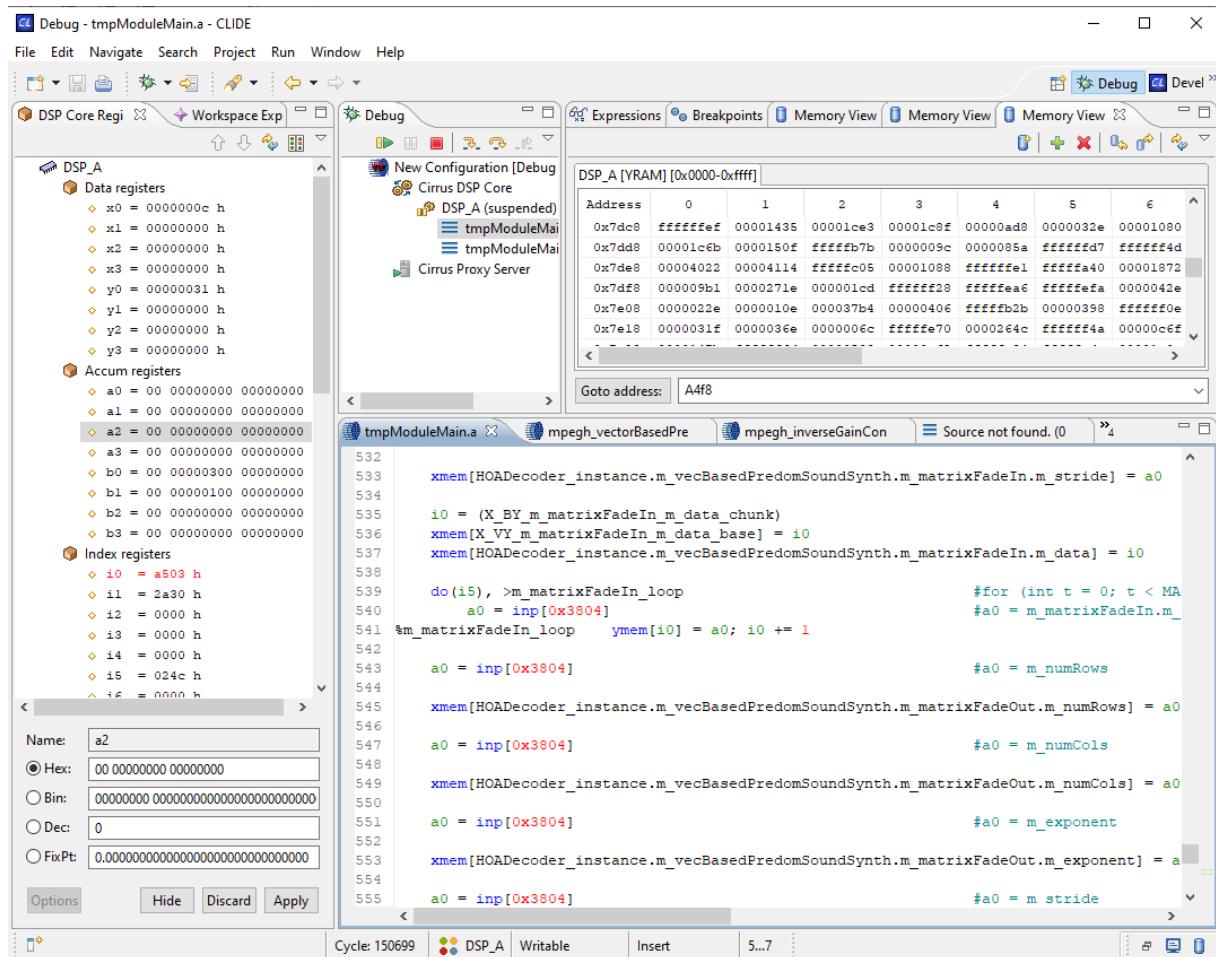
do (i0), >update_Loop
    a1 = xmem[i2]
    a1 = a1 - a0; a0 = xmem[i1]; i1 += 1
%update Loop xmem[i2] = a1; i2 += 1
```

Slika 5.2 Izgled hardverske petlje u asembleru

5.5 Razvojno okruženje

Za potrebe ovog projekta, korišćeno je razvojno okruženje kompanije *Cirrus Logic* pod imenom CLIDE. Ovo razvojno okruženje je zasnovano na *Eclipse* platformi.

CLIDE omogućava razvoj koda u asemblerskom i u C programscom jeziku. Ovo okruženje podržava izvršavanje DSP aplikacije, na ciljnoj platformi. Pored toga, CLIDE sadrži i alat za pregled memorije, kao i alat za kontrolisano izvršavanje programa (engl. *Debugger*) koji se može videti na slici 5.3. Pored toga, ovo razvojno okruženje ima i omogućen pregled trenutnog stanja svih registara i konstanti u više različitih predstava.



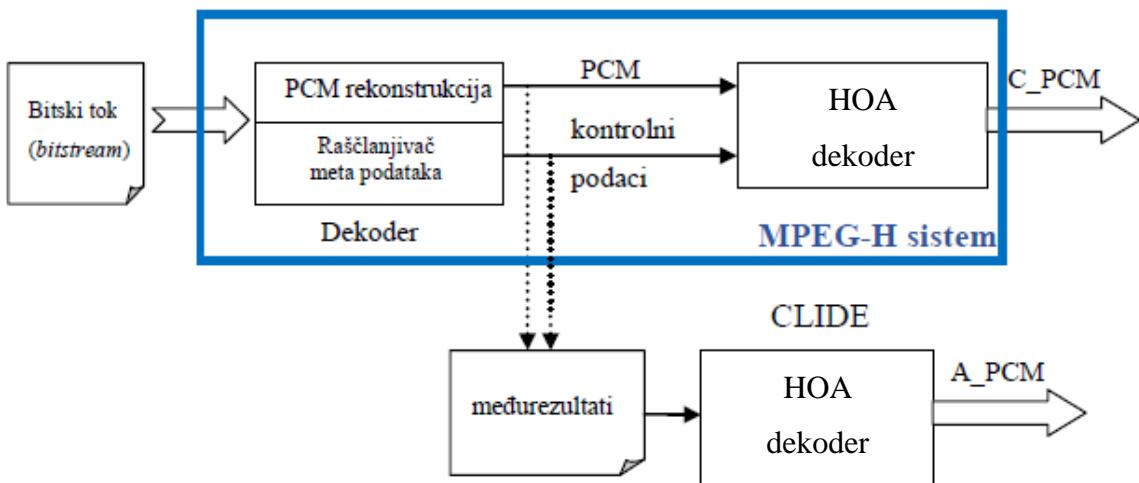
Slika 5.3 Primer kontrolisanog izvršavanja u CLIDE razvojnom okruženju

6. Rezultati

Radi provere ispravnosti rešenja, bilo je neophodno ispitati rad implementiranih funkcija. Ispitivanje je vršeno upoređivanjem izlaznih odbiraka referentne implementacije sa izlaznim odbircima implementacije za ciljnu platformu na nivou bita.

6.1 Ispitivanje ispravnosti programa uz pomoć bit-identičnih testova

Nakon svake implementirane funkcije je vršena provera ispravnosti. Da bi se ispitala ispravnost, korišćeno je 8 ulaznih datoteka sa frekvencijom od 48 kHz. Testiranje je izvršavano tako što bi se jedna od ulaznih datoteka propuštala kroz referentni kod, pri čemu bi se u dve odvojene datoteke generisali podaci koji su potrebni kodu u asembleru. U drugu datoteku bi se generisali obrađeni podaci koji se dobijaju na samom izlazu iz sistema. Podaci iz prve datoteke su se postavljali kao ulazni podaci u asemblerskom kodu i nakon toga su propuštani kroz sistem. Na izlazu iz sistema u asembleru su se takođe generisali podaci koji su služili za upoređivanje. Ispitivanje se obavljalo na način gde se ispravnost utvrđivala upoređivanjem sadržaja referentne datoteke, odnosno druge datoteke i datoteke u koju su se upisivali izlazni podaci implementirane funkcije u asembleru. Ovi bit-identični testovi su se izvršavali pomoću alata *Compare By Content* u okviru programa *Total Commander*. Rezultat ovog testiranja je da se, za svih 8 datoteka koje su korišćene, utrvdilo da su sadržaji iz datoteka sa referentnim izlaznim podacima i sadržaji iz datoteka koji su generisani na izlazu implementiranog sistema, identični na nivou bita.



Slika 6.1 Testiranje modula

S obzirom da je implementacija modula koji prethode ovom projektu i dalje u izradi, nije bilo moguće proveriti ispravnost implementiranih funkcija u okviru celokupnog sistema.

6.2 Potrošnja resursa

Potrošnja resursa ciljne platforme se može videti u tabeli 6.1 u kojoj je prikazana zauzetost programske i memorije podataka izražene u vidu 32-bitnih i 64-bitnih reči, odnosno memorijskih lokacija.

X memorija [32-bitnih reči]	Y memorija [32-bitnih reči]	L memorija [64-bitnih reči]	P memorija [32-bitnih reči]
8760	38883	2	1775

Tabela 6.1 Prikaz potrošnje memorije za HOA obradu

U nastavku je data formula za računanje utroška procesorskog vremena u milionama instrukcija po sekundi:

$$MIPS = \frac{\text{broj_ciklusa} * \frac{F_s}{\text{BLOCK_SIZE}}}{1000000}$$

pri čemu parametar F_s predstavlja učestalost odabiranja, odnosno broj odbiraka u jednoj sekundi. $BLOCK_SIZE$ predstavlja veličinu bloka za obradu. Uz pomoć kontrolisanog izvršavanja programa utvrđen je tačan broj ciklusa za izvršavanje programa. Utrošak procesorskog vremena za HOA Algoritam je direktno proporcionalan broju transportnih kanala koji se nalaze u okviru dekodovanog audio toka. Povećanje broja transportnih kanala dovodi do povećanja utroška procesorskog vremena. U najgorem slučaju broj transportnih kanala je 16, koji je ujedno i najveći broj transportnih kanala u sistemu. Vrednost ukupnog utroška procesorskog vremena za najgori slučaj je prikazana u tabeli 6.2.

Frekvencija odabiranja	48 kHz
Ukupno procesorsko vreme (najgori slučaj)	35,78 MIPS

Tabela 6.2 Prikaz potrošnje procesorskog vremena

7. Zaključak

Zadatak ovog rada je implementacija modula za renderovanje 3D audio slike po MPEG-H standardu na DSP DC49844 sa četiri jezgra, kompanije *Cirrus Logic*. Na početku izrade ovog projekta bilo je neophodno upoznati se i detljivo analizirati referentni kod napisan u programskom jeziku C++ u okviru instituta *Fraunhofer IIS* iz Nemačke, da bi se moglo pristupiti realizaciji projekta.

Izradom ovog rada, stečena su nova saznanja o primeni DSP procesora u široko rasprostranjenoj potrošačkoj elektronici. Bolje je shvaćen koncept funkcionisanja obrade zvuka, kao i 3D zvuka, uz razumevanje MPEG-H standarda. Prilikom implementacije koda u asemblerskom jeziku, unapređeno je znanje iz polja programiranja na niskom, odnosno hardverskom nivou, kao i o ciljnoj platformi na kojoj se primenjuje ovaj modul. Takođe, stečena su nova i unapređena su neka prethodna znanja vezana za rad sa registrima i optimalnu upotrebu resursa.

Na kraju projekta neophodno je bilo ispitati funkcionalnost napisanog koda. Provera ispravnosti se vršila uz pomoć testova na nivou bita, odnosno upoređivanjem izlaznih signala referentne aplikacije sa izlaznim signalima implementirane aplikacije. Za proveru je korišćeno 8 ulaznih datoteka sa frekvencijom odabiranja od 48 kHz. Rezultat testiranja nakon upoređivanja izlaznih podataka iz referente aplikacije i implementirane aplikacije je da su podaci identični na nivou bita. Time je utvrđeno da funkcionalnost implementirane aplikacije zadovoljava uslove rada i potvrđen je završetak projekta. Nakon toga je izvršena procena utroška resursa. Utvrđeno je da je za najgori slučaj utrošeno procesorsko vreme iznosilo 35,78 MIPS, pri čemu je broj transportnih kanala 16, koji je ujedno i najveći broj transportnih kanala u sistemu. Sa manjim brojem transportnih kanala, smanjuje se i potrošnja procesorskog vremena.

Provera ispravnosti programa izvršena je u simulatorskim uslovima, usled nemogućnosti testiranja u realnim uslovima, zbog toga što određeni moduli koji prethode ovom modulu nisu u potpunosti implementirani, tj. nalaze se u procesu implementacije. U daljem radu se planira integracija implementiranog HOA modula u ostatak sistema, implementacija preostalih modula sistema i realizacija celokupnog sistema na datoj ciljnoj platformi.

8. Literatura

- [1] Fraunhofer IIS, <https://www.iis.fraunhofer.de/en/ff/amm/broadcast-streaming/mpegh.html>, oktobar 2019
- [2] Information Technology – High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 3: 3D Audio, ISO/IEC 23008-3:2015, 2015
- [3] AES-LA Webinar: *MPEG-H for Broadcast, Streaming and Music*, <https://www.mpegh.com/en/webinar>, jul 2020
- [4] Audioholics, *Three Methods for Immersive 3D Audio*, <https://www.audioholics.com/audio-technologies/three-methods-for-immersive-3d-audio>, oktobar 2019
- [5] SSA Plugins, *What Is... Higher Order Ambisonics*, <https://www.ssa-plugins.com/blog/2017/07/18/what-is-higher-order-ambisonics>, jul 2020
- [6] Andrea F. Genovese, Charles Craig Jr, Simon Calle: „*3ME – A 3D Music Experience*“, 2016
- [7] P. Coleman et al., „*An Audio-Visual System for Object-Based Audio: From Recording to Listening*“ in IEEE Transactions on Multimedia, vol. 20, no. 8, pp. 1919-1931, Aug. 2018, doi: 10.1109/TMM.2018.2794780.
- [8] Electronic Design, *MPEG-H Audio Brings New Features to TV and Streaming Sound*, <https://www.electronicdesign.com/technologies/communications/article/21800854/mpegh-audio-brings-new-features-to-tv-and-streaming-sound>, jul 2020
- [9] Nils Peters, Deep Sen, Moo-Young Kim, Oliver Wuebbolt, S. Merrill Weiss: „*Scene-based Audio Implemented with Higher Order Ambisonics (HOA)*“, 2015
- [10] Robert L. Bleidt et al.: „*Development of the MPEG-H TV Audio Systems for ATSC 3.0*“, 2017

- [11] Jürgen Herre, Johannes Hilpert, Achim Kuntz, and Jan Plogsties: „*MPEG-H Audio – The New Standard for Universal Spatial / 3D Audio Coding*“, 2014
- [12] Nives Kaprocki: „*Izrada bloka audio obrade za virtuelizaciju zvučnika na DSP platformi*“, Novi Sad, 2016
- [13] Dr Jelena Kovačević, Dejan Bokan: „*Arhitekture i algoritmi digitalnih signal procesora – Zbirka zadatka i laboratorijski priručnik*“, FTN izdavaštvo, Novi Sad, 2016
- [14] Jelena Boroja: „*Implementacija MPEG-H 3D audio modula za završnu obradu objekata na DSP platformi*“, Novi Sad, 2019
- [15] Vladimir Kovačević, Miroslav Popović, Miodrag Temerinac, Nikola Teslić: „*Arhitekture i algoritmi digitalnih signal procesora I*“, FTN izdavaštvo, Novi Sad, 2005