УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ

Нивес Капроцки

Једно решење модула за виртуализацију горњих канала на платформи са ограниченим ресурсима

МАСТЕР РАД

Нови Сад, 2017.

УНИВЕРЗИТЕТ У НОВОМ САДУ ● **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:			
Идентификациони број, **ИБР**:			
Тип документације, **ТД**:		Монографска документација	
Тип записа, **ТЗ**:		Текстуални штампани материјал	
Врста рада, **ВР**:		Дипломски – мастер рад	
Аутор, **АУ**:		Нивес Капроцки	
Ментор, **МН**:		Др Јелена Ковачевић	
Наслов рада, **НР**:		Једно решење модула за виртуализацију горњих канала на платформи са ограниченим ресурсима	
Језик публикације, **ЈП**:		Српски / латиница	
Језик извода, **ЈИ**:		Српски	
Земља публиковања, **ЗП**:		Република Србија	
Уже географско подручје, **УГП**:		Војводина	
Година, **ГО**:		2017	
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)		7/50/14/2/25/0/0	
Научна област, **НО**:		Електротехника и рачунарство	
Научна дисциплина, **НД**:		Рачунарска техника	
Предметна одредница/Кqучне речи, **ПО**:		ДСП, виртуализација, горњи канали	
УДК			
Чува се, **ЧУ**:		У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, **ВН**:			
Извод, **ИЗ**:		У оквиру овог рада је реализован модул за виртуализацију и креирање горњих канала на платформи са ограниченим ресурсима. Успешна реализација алгоритма се постиже имплементацијом у ниском програмском језику и оптимизацијом урађеног кода. Испитивањем и верификацијом решења је утврђено да је исправна обрада у реалном времену омогућена.	
Датум прихватања теме, **ДП**:			
Датум одбране, **ДО**:		11.07.2017.	
Чланови комисије, **КО**:	Председник:	Др Илија Башичевић	
	Члан:	Др Иван Мезеи	Потпис ментора
	Члан, ментор:	Др Јелена Ковачевић	

UNIVERSITY OF NOVI SAD ● **FACULTY OF TECHNICAL SCIENCES**
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:			
Identification number, **INO**:			
Document type, **DT**:		Monographic publication	
Type of record, **TR**:		Textual printed material	
Contents code, **CC**:		Master Thesis	
Author, **AU**:		Nives Kaprocki	
Mentor, **MN**:		Phd Jelena Kovačević	
Title, **TI**:		One solution of height channels virtualization on DSP platform	
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**:		2017	
Publisher, **PB**:		Author's reprint	
Publication place, **PP**:		Novi Sad, Dositeja Obradovica sq. 6	
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)		7/50/14/2/25/0/0	
Scientific field, **SF**:		Electrical Engineering	
Scientific discipline, **SD**:		Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, **S/KW**:		DSP, virtualization, height channels	
UC			
Holding data, **HD**:		The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, **N**:			
Abstract, **AB**:		This paper describes the module for virtualization and creation of height channels on DSP platform. Successful implementation is achieved by using low level programming languages and optimization of the code. Testing and verification of the solution has showed that correct real-time execution is enabled.	
Accepted by the Scientific Board on, **ASB**:			
Defended on, **DE**:		11.07.2017	
Defended Board, **DB**:	President:	Phd Ilija Bašičević	
	Member:	Phd Ivan Mezei	Menthor's sign
	Member, Mentor:	Phd Jelena Kovačević	

Zahvalnost

Zahvaljujem se svom mentoru Jeleni Kovačević, tehničkom mentoru Sanji Čordašić kao i svom kolegi sa instituta „RT-RK" Branku Đorđeviću za pomoć i saradnju tokom izrade projekta.

Takođe se zahvaljujem svojoj porodici i prijateljima na konstantnoj podršci tokom master studija.

SADRŽAJ

SPISAK SLIKA

SPISAK TABELA

SKRAĆENICE

DSP - *Digital Signal Processor*, Digitalni signal procesor

AVR - *Audio Video Receiver*, Audio/video prijemnik

HRTF - *Head Related Transfer Function*, Prenosna funkcija koja se odnosi na glavu

HDMI - *High-Definition Multimedia Interface*, Multimedijalni sprežni podsistem visoke definicije

SPI - *Serial Peripheral Interface*, Serijski sprežni sistem za periferije

I2C - *Inter-Integrated Circuit*

I2S - *Inter-IC Sound*

SPDIF - *Sony/Philips Digital Interface Format*, Sony Philips digitalni sprežni podsistem

ACCN - *Audio Configuration Change Notification*, Notifikacija vezana za promenu audio konfiguracije

FIFO - *First In First Out*, Prvi ulaz, prvi izlaz

MAC - *Multiply and Accumulate*, Instrukcija za množenje i akumuliranje rezultata

SRS - *Shift Round Saturate*, Jedinica sa aritmetičko pomeranje, zaokruživanje i saturaciju Crystal DSP familije procesora

DMA - *Direct Memory Access*, Direktan pristup memoriji

OS - *Operating System*, Operativni sistem

MIF - *Module Interface*, Sprežni podsistem

MCT - *Module Call Table*, Tabela pozivanja modula

MCV - *Module Control Vector*, Tabela kontrolnih vektora modula

BBT - *Black Box Testing*, Ispitivanje metodom crne kutije

1. Uvod

U okviru ovog rada realizovan je modul za virtualizaciju gornjih kanala na platformi sa ograničenim resursima, odnosno digitalnom signal procesoru DSP (*eng. Digital Signal Processor*). Polazna osnova implementacije je referentni algoritam realizovan u C programskom jeziku. Kako su zahtevi podrazumevali rešenje na jednom procesoru, za implementaciju je izabran četvorojezgarni CS49844 procesor Crystal DSP familije kompanije *Cirrus Logic*. Najveći izazov tokom implementacije na platformi sa ograničenim resursima je dostupna količina memorije i procesorskog vremena. Zato što nova generacija audio dekodera zahteva veliku količinu resursa, završni blok obrade, u okviru koga se nalazi i modul za virtualizaciju gornjih kanala, mora da bude implementiran na samo jednom jezgru procesora. Uspešna realizacija algoritma se postiže implementacijom u niskom programskom jeziku i optimizacijom urađenog koda. Ova tehnologija ima prvenstveno primenu u audio/video prijemnicima AVR (*eng. Audio Video Receiver*), kao i Soundbar sistemima.

Sa razvojem višekanalnih audio sistema je porastao broj kanala preko kojih se reprodukuje zvuk. Veći broj kanala podrazumeva veći broj zvučnika, što nije u skladu sa trendom u potrošačkoj elektronici da uređaji budu kompaktniji i manje glomazni. Rešenje su digitalni signal procesori, koji omogućavaju virtualizaciju kanala okruženja (eng. Surround) i gornjih kanala. Ovom obradom se stvara percepcija da zvuk dolazi iz odgovarajućeg izvora, a ne sa mestu gde se nalazi fizički zvučnik. Na ovom principu su bazirani Soundbar sistemi, koji predstavljaju grupu zvučnika postavljenu direktno ispred slušaoca. Osim toga, moguće je odgovarajućom obradom prilagoditi izlazni broj kanala tako da se visoko kvalitetan zvuk pušta čak i preko klasičnih stereo sistema.

Rad je organizovan u sedam poglavlja. U drugom poglavlju rada date su teorijske osnove koje podrazumevaju kratak istorijski razvoj višekanalnih audio tehnologija, uvod u kodovanje

zasnovano na kanalima i kodovanje zasnovano na audio objektima, opis glavnih tehnika dekorelacije signala i metoda renderovanja virtualnih izvora zvuka. U trećem poglavlju izneto je jedno rešenje modula za virtualizaciju i kreiranje gornjih kanala. U četvrtom poglavlju date su tehničke specifikacije ciljne platforme, detalji realizacije audio sistema koji koristi opisani modul i tehnike optimizacije realizovanog rešenja. Peto poglavlje sadrži rezultate testiranja i utrošak resursa. U šestom poglavlju ukratko je sažeto sve što je u okviru ovog rada urađeno. Poslednje, sedmo poglavlje, sadrži spisak korišćene literature prilikom izrade ovog rada.

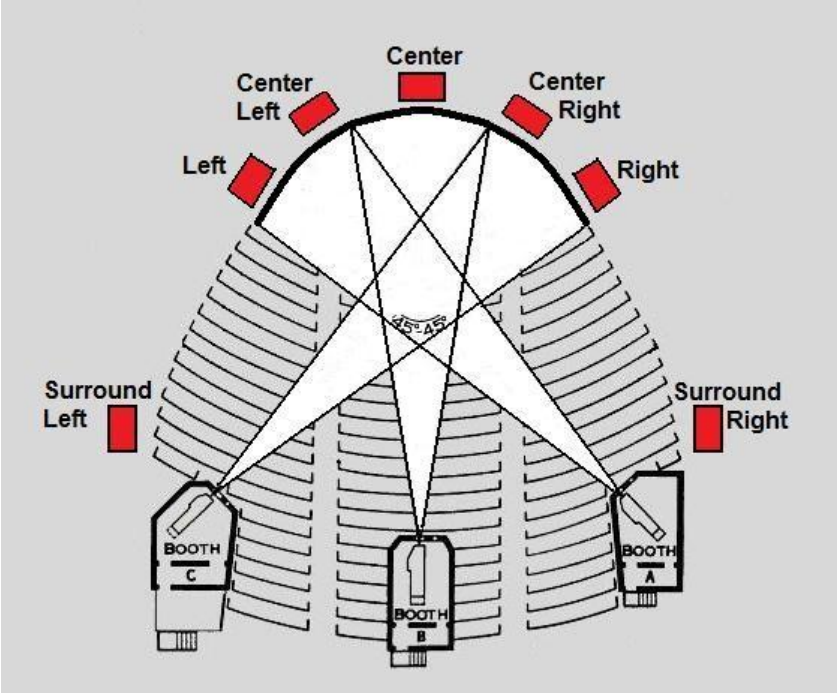
2. Teorijske osnove

Filmski, televizijski zvuk i muzički formati su bili proizvod različitih industrija koje su radile izolovano jedna od druge. Zvuk iz okruženja (surround zvuk) je prvenstveno zamišljen kao način da se gledaocima poboljša iskustvo gledanja filmova u bioskopu i filmska industrija je pokretala njegov razvoj. Popularnost zvuka iz okruženja u kućnim sistemima je dovela do toga da ove industrije bliže sarađuju i izgrade ujednačeniji pristup reprodukciji zvuka.

Pronalazačem zvuka iz okruženja se po više osnova smatra Volt Dizni. Dok je film Fantazija još bio u razvoju, četrdesetih godina dvadesetog veka, Dizni se sastao sa inžinjerima iz Bel laboratorije koji su radili na tehnologiji snimanja stereo zvuka pomoću više mikrofona. Njegova ideja je bila da se tokom segmenta filma „Bumbarov let" zvuk letenja bumbara čuje svuda oko publike, a ne samo ispred nje. Iz toga je proizašla tehnologija koja je dobila ime Fantasound [1] i omogućavala je reprodukciju zvuka sa levim, desnim i centralnim prednjim kanalom. U određenim slučajevima su dodavani i levi i desni zadnji kanali. Koristeći optički zvuk su tri zvučna kanala snimana na posebne filmske trake, koje su puštane paralelno sa slikom. Četvrta traka je korišćena za kontrolisanje pojačanja svakog od tri kanala. Fantasound tehnologija je ugrađena u samo četrnaest bioskopa, jer je neophodna dodatna oprema bila preskupa za široku upotrebu.

Tek čitavu deceniju kasnije je Holivud prihvatio novu, povoljniju višekanalnu zvučnu tehnologiju. Ona se pojavila paralelno sa novim bioskopskim filmskim formatima, Cinerama i CineScope. Ovim formatima ekrana je bio potreban veći broj kanala kako bi se pokrilo čitavo prednje polje filmskog platna. Rešenje je bio zvučni zid koji je obuhvatao četiri do sedam zvučnih kanala. Prethodno je svaki zvučni kanal optički kodovan na filmsku traku, ali kako bi veći broj kanala mogao da bude smešten na jednu traku, više kanala je magnetski kodovano. Cinerama sistem (Slika 2.1) je bio izuzetno kompleksan i skup, a zvuk je reprodukovan

puštanjem magnetne filmske trake od 35 mm sa sedam zvučnih kanala. Iako pomenuta tehnologija nije zaživela, bioskopi su uspeli da nadmaše kvalitet koji je pružala tada sve popularnija televizija. S druge strane, u standardnoj postavci CineScope tehnologije su bila četiri kanala, levi, centralni, desni i kanal okruženja, snimani na magnentnu filmsku traku od 35mm.



Slika 2.1 Raspored zvučnika u Cinerama sistemu

Kanal okruženja se u početku koristio uglavnom za povremene dramatične sekvence pa je stoga dobio ime kanal za efekte. Iako je tokom šezdesetih i sedamdesetih godina opalo interesovanje za tehnologije zvuka iz okruženja zbog visoke cene magnetnih formata i smanjenog broja posetilaca u bioskopima, nastavljeno je eksperimentisanje sa kanalom okruženja. U formatu sa šest kanala snimljenih na filmskoj traci od 70mm je na primer kanal okruženja korišćen za ambijentalne zvukove tokom celog filma. Ovako je započet današnji koncept tehnologije zvuka iz okruženja kojim se stvara difuzno zvučno polje.

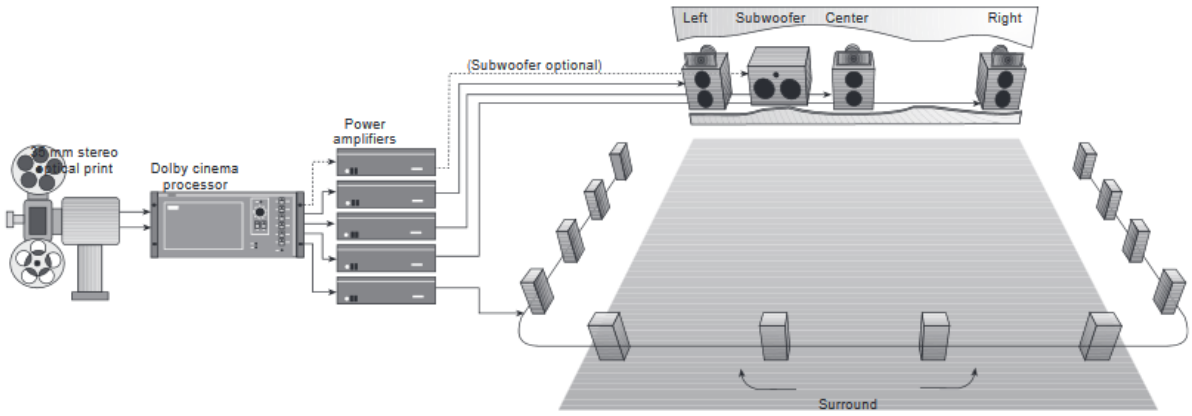
Iako su u bioskopima u to vreme korišćena četiri do sedam zvučnih kanala za stereo zvuk, kućni stereo formati su sadržali samo dva kanala. Oni su predstavljali dramatičan napredak u odnosu na mono zvuk uz uporedivo jednostavnu implementaciju. Dvokanalni zvuk je postao standard u kućnim sistemima, čak i kad je filmska industrija postavila četiri kanala kao minimum da se stvori realno zvučno polje. Kada je predstavljen dvokanalni stereo radio prenos, dalje se ustalilo među potrošačima da stereo zvuk podrazumeva samo dva kanala.

Kako je rasla popularnost kućnih stereo sistema, proizvođači su tražili način da prošire svoje tržište. Tako je nastao kvadrofonski (četvorokanalni) kućni stereo sistem ranih sedamdesetih godina prošlog veka. Zahtevao je još dva dodatna zvučnika u zadnjim uglovima sobe, kako bi se reprodukovali dodatni kanali iz specijalno kodovanih programskih izvora. Zato

što postojeći stereo sistemi nisu jednostavno mogli da rukuju većim brojem kanala, razvijeno je nekoliko šema koje su omogućavale kodovanje dodatnih informacija u dva osnovna kanala. Većina je bila bazirana na matričnim tehnikama, gde su dodatni kanali snimani u osnovne kanale sa različitom relativnom fazom. Kvadrofonski sistemi nisu prihvaćeni od strane velikog dela tržišta prvenstveno jer je mali broj potrošača uvideo značajno poboljšanje u odnosu na stare sisteme.

Kvadrofonski sistemi nisu dovedeni u vezu sa višekanalnim stereo sistemima koji su upotrebljavani u bioskopima, i termin "zvuk iz okruženja" je i dalje bio rezervisan isključivo za filmsku industriju. Glavni razlog za ovo mišljenje je bila i činjenica da je najpopularniji kućni medijum bila televizija, koja je i dalje pružala mono zvuk niskog kvaliteta. Narednu deceniju su se kućni stereo, bioskopski stereo i televizijski zvuk i dalje razvijali izolovano.

Sredinom sedamdesetih godina se pojavila nova tehnologija, Dolby Stereo, gde je optički sniman zvuk na filmsku traku od 35mm. Kako bi se obezbedila kompatibilnost sa mono bioskopima, bilo je neophodno smestiti stereo trake u isti prostor koji su zauzimale tradicionalne mono trake [2]. Utvrđeno je da se uz Dolby tehnologiju smanjivanja šuma dobija visok kvalitet zvuka kada su dva kanala snimljena na filmsku traku. Međutim, veći broj kanala je povećavao šum na neprihvatljiv nivo. Pošto je u bioskopu standardna postavka podrazumevala prisustvo levog, desnog, centralnog i kanala okruženja, morao je da se nađe način da se veći broj kanala smesti na filmsku traku. Rešenje je nađeno u matričnoj tehnici koja se prethodno koristila u kvadrofonskim kućnim sistemima. Dolby stereo optički format se pokazao izuzetno praktičnim, tako da čak i danas filmovi imaju analognu Dolby traku kako bi mogli da se puštaju u svim bioskopima (Slika 2.2).



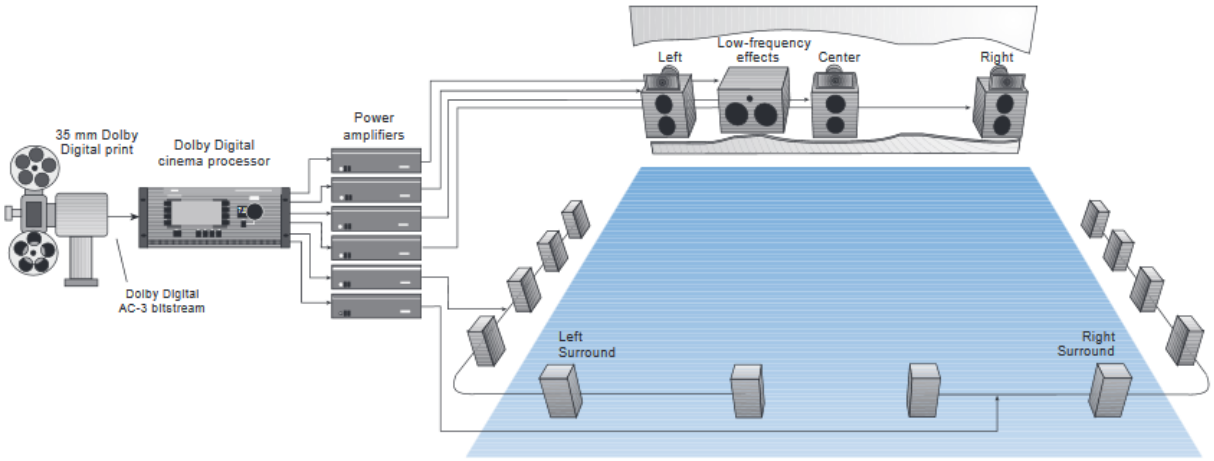
Slika 2.2 Dolby Stereo sistem u bioskopu

Tokom sedamdesetih godina se desila video revolucija sa pojavom video kaseta i kablovske televizije. U to vreme su potrošači već navikli na visoko kvalitetni stereo zvuk iz kućnih muzičkih sistema, ali i višekanalni Dolby Stereo format rasprostranjen u filmskoj

industriji. Prvobitan mono zvuk kod video kaseta je zamenjem HiFi tehnologijom koja je obezbeđivala visoko kvalitetan stereo zvuk. Ubrzo je stereo zvuk uključen i u televizijske sisteme za prenos.

Zvuk iz okruženja je uveden u kućne muzičke sisteme uvođenjem jednostavnog Dolby Surround dekodera koji je dekodovao kanal okruženja. Za razliku od kvadrofonskih sistema, Dolby Surround je prihvaćen od strane tržišta. Prvi razlog je bio taj što se konfiguracija kanala čvrsto uspostavljena u filmskoj industriji mogla samo jednostavno prebaciti u potrošačku elektroniku. Takođe, glavna svrha ove tehnologije je bilo poboljšanje iskustva gledaoca, zbog čega su mnogi potrošači odlučili da ulože u sistem koji podržava pomenutu tehnologiju.

Bioskopski zvuk postaje digitalan uvođenjem Dolby Digital tehnologije početkom devedesetih godina. Ova tehnologija je predstavila 5.1 konfiguraciju kanala [3], koja podrazumeva levi, desni, centralni, levi i desni kanal okruženja, kao i šesti niskofrekventni kanal, koji zahteva desetinu punog opsega. I danas je Dolby Digital vodeći zvučni format u filmskoj industriji (Slika 2.3).



Slika 2.3 Dolby Digital sistem u bioskopu

Dolby Digital u kućnim sistemima predstavlja konačan prelaz višekanalnih formata iz bioskopskih sistema u kućne zvučne sisteme. Kao i njegova bioskopska varjanta, poseduje levi, desni, centralni, levi kanal okruženja, desni kanal okruženja i niskofrekventni kanal. Za razliku od Dolby Surround sistema koji poseduje jedan kanal okruženja ograničenog opsega, kod Dolby Digital sistema postoje dva nezavisna kanala okruženja koji reprodukuju zvuk jednakog kvaliteta kao i tri prednja kanala. Na ovaj način se postigao veći nivo realizma i bolja lokalizacija. Ovi sistemi takođe podržavaju i različite konfiguracije zvučnika i mogu efikasno da višekanalni sadržaj predstave i preko samo dva, ili čak i jednog kanala.

2.1 Vrste kodovanja 3D zvučnih slika

Postoje dva pristupa prenošenja 3D zvučnih slika preko telekomunikacionih kanala. Prvi pristup podrazumeva postojanje najmanje dva zvučna kanala, kod kojih nije moguće pristupiti posebnim zvučnim objektima jer su oni već ukomponovani u zvučnu sliku. S druge strane, kod pristupa zasnovanog na audio objektima su svi objekti posebni tokovi podataka koji se pojedinačno prenose zajedno sa opisom slike. Ovaj opis definiše prostornu strukturu slike, njeno ponašanje u vremenu kao i brojne naprednije karakteristike. Izbor između kodovanja zasnovanog na kanalima i kodovanja zasnovanog na audio objektima zavisi od tipa zvučne slike, mogućnosti sistema i primene.

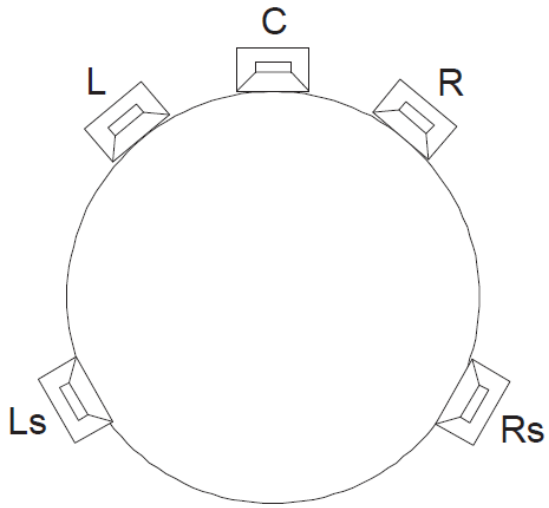
2.1.1 Kodovanje zasnovano na kanalima

Glavna prednost ovog pristupa je zahtev za relativno jednostavnim sistemom. Međutim, njegova skalabilnost je dosta slaba jer je stvoren za specifičnu konfiguraciju zvučnika i reprodukcija na drugačijoj konfiguraciji dovodi do umanjenog kvaliteta zvuka. Takođe, obrađivana zvučna slika je statična i ne može biti modifikovana od strane krajnjeg korisnika.

Snimanje pomoću dva mikrofona omogućava da se celokupna 3D zvučna slika reprodukuje pomoću običnog stereo sistema. Ovaj tip snimanja je zasnovan na psihoakustici i načinu na koji mozak sprovodi lokalizaciju izvora zvuka, veličinu izvora zvuka, njegovu udaljenost i okolinu. Snimci se dobijaju postavljanjem mikrofona u slušne kanale subjekta ili korišćenjem veštačke glave sa mikrofonima [4]. Isti snimci mogu da se kreiraju i veštački izvršavanjem operacije konvolucije nad snimkom sa jednog mikrofona i HRTF (*eng. Head Related Transfer Function*) bazom podataka. Glavni problem u opisanim metodama predstavlja ograničenost okretanja glave slušaoca, jer se sa rotacijom glave menja i lokacija izvora zvuka. Rešenje ovog problema je korišćenje uređaja koji prati orjentaciju glave slušaoca u realnom vremenu. Ipak, precizna lokalizacija zahteva kompleksnu obradu sa malim kašnjenjem kao i veću kontrolu sadržaja nego što dva predefinisana kanala omogućavaju. Dodatni problem predstavlja HRTF, koja se razlikuje od korisnika do korisnika i može da dovede do greške.

Drugi metod predstavlja korišćenje više zvučnih kanala punog opsega za reprodukciju 3D zvučnih slika. U ovom slučaju se svaki kanal pušta na odgovarajućem posebnom zvučniku. Jedan od najčešće korišćenih surround zvučnih formata je 5.1 (Slika 2.4). Kako bi se stvorila iluzija virtualnih izvora zvuka postavljenih oko slušaoca često se koristi tehnika menjanja amplitude nad parovima kanala. Još jedna tehnika koja je u širokoj upotrebi u 5.1 sistemima su ambisionici, kojima je cilj da se poveća tačnost lokalizacije izvora zvuka i proširi prostor u kom se primećuje surround efekat. Glavna mana pristupa gde se svaki kanal reprodukuje na jednom zvučniku je relativno veliki broj kanala, i samim tim i zvučnika. Druga negativna strana je fiksna

konfiguracija zvučnika koja ograničava fleksibilnost i skalabilnost formata. Takođe, kako bi se smanjio bitski tok višekanalnih zvučnih sadržaja, koriste se tradicionalne tehnike kodovanja. Međutim, modeli za kompresiju nisu prilagođeni višekanalnom kontekstu, pa se prilikom puštanja ovakvih sadržaja sa različitih pozicija u prostoru javljaju šumovi.



Slika 2.4 Raspored zvučnika u 5.1 surround sistemu

Ambisionici su 3D višekanalna tehnika nastala sedamdesetih godina prošlog veka [5]. Za razliku od prethodnih tehnika, ambisionici koduju celokupnu 3D zvučnu sliku u konačan broj kanala koji su poznati kao B format. Zvučna slika se snima specijalnim mikrofonom (Soundfield mikrofonom) ili sa jednim omnidirekcionim mikrofonom i tri gradijentna mikrofona, od kojih je svaki postavljen u jedan od tri zamišljena koordinatna sistema (X, Y, Z). Od četiri kanala koja se snimaju, jedan je mono, dobijen preko omnidirekcionalnog mikrofona (ili superpozicijom svih kapsula Soundfield mikrofona), a ostala tri su snimljena kao razlike pritisaka na odgovarajućim osama, na primer kanal X je razlika u gradijentu „ispred“, odnosno „iza“ referentne tačke na x-osi. Analogno je i za kanale Y i Z. Daljim dekodovanjem snimka moguće je za odgovarajuću konfiguraciju zvučnika dobiti reprodukciju koja odgovara zamišljenoj: očuvane su lokalizacija i karakteristike snimka i prostorije. Prednost ovog pristupa je da je kodovan B format zvučne slike nezavisan od posebne konfiguracije zvučnika, i omogućava veću fleksibilnost i raznolikost nego prethodna tehnika. Iako ambisionici većeg stepena zahtevaju veći broj kanala, dobija se detaljniji opis zvučne slike i preciznija lokalizacija izvora zvuka.

2.1.2 Kodovanje zasnovano na audio objektima

Kodovanje zasnovano na audio objektima omogućava interaktivnost i skalabilnost zvučne slike. Ista 3D zvučna slika može da bude reprodukovana na različitim sistemima, od kućnog bioskopa do mobilnog telefona, jer je renderovanjem moguće prilagoditi zvučnu sliku ciljnom uređaju. Druga prednost ovog pristupa je mogućnost modifikacije zvučne slike od strane

krajnjeg korisnika. Iz tog razloga je kodovanje zasnovano na audio objektima pogodno za primene kao što je virtualna realnost ili konferencijski razgovori koji se odvijaju u realnom vremenu. Takođe je veoma korisno što je moguće kodovati zasebne objekte, pa se dobija ušteda u protoku tako što se za svaki objekat koristi optimalna metoda kodovanja. Glavna negativna odlika ovih sistema je velika kompleksnost i visoka cena.

Kod zvučnih sistema baziranih na objektima se kodovani sadržaj prenosi bitskim tokom do dekodera koji iščitava kompresovane audio podatke. Dekoder zatim na osnovu ciljne konfiguracije odlučuje koji sadržaj je neophodno dekodovati. Ukoliko sistem podržava samo pristup zasnovan na kanalima u ovom momentu se odbacuju objekti i stvaraju kanali za predefinisanu konfiguraciju zvučnika. Zajedno sa objektima su u bitskom toku prenešeni i meta podaci, koji bliže opisuju objekte.

Nakon dekodovanja, sistemi zasnovani na audio objektima vrše 3D audio renderovanje. Glavni zadatak ovog bloka obrade je dodavanje ili oduzimanje postojećih audio objekata u ili iz zvučnog zapisa koji se prenosi bitskim tokom. Oduzimanje objekata je neophodno na primer kada se stvara kanalni audio sadržaj od audio objekata, pa ih je potrebno ukloniti iz bitskog toka kako ne bi bili dvostruko uključeni u izlazni reprodukovani zvuk.

Svaki objekat dolazi do renderer bloka sa odgovarajućim meta podacima koji diktiraju njegovu poziciju u prostoru, faktore skaliranja itd. Renderer vrši proračune čiji je krajnji rezultat niz amplituda sa kojima treba primeniti objekat na svaki od zvučnika. Kalkulacije su bazirane na prezentaciji sfernog koordinatnog sistema. U sklopu renderovanja je moguća i interakcija sa korisnikom, gde korisnik može da utiče na određene atribute objekata.

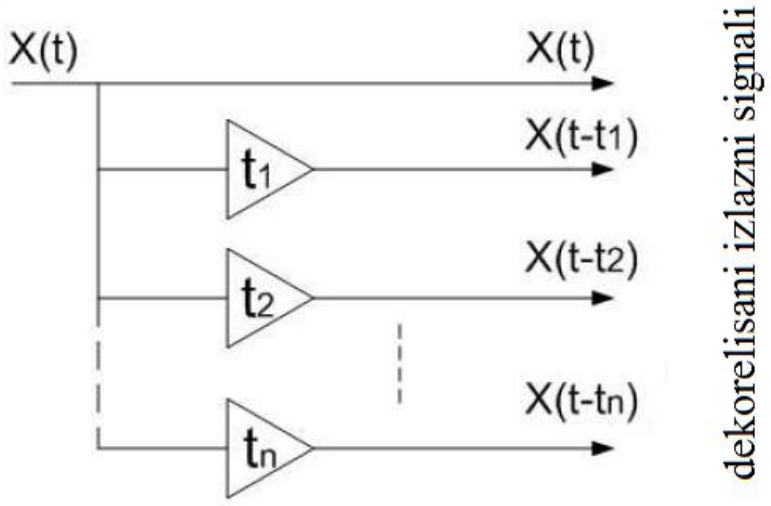
2.2 Tehnike dekorelacije signala

Dekorelacija je proces tokom kog se signali transformišu tako da pojedinačno zvuče isto, ali se njihov talasni oblik menja, pa u kombinaciji sa drugim signalima se stvaraju određeni efekti. Stepen dekorelacije zvuka se pokazao kao značajan predviđač prostornih efekata. Čak i u prirodi se dekorelacija dešava kao proizvod akustičnih ili električnih procesa koji često menjaju zvuk izvora. Dok se u muzičkim studijima vokali nekad snimaju na dve zasebne trake kako bi varijacije u snimcima proizvele dekorelaciju. Dekorelacija utiče na percepciju prostorne slike tako što eliminiše obojenost, stvara se difuzno zvučno polje, eliminiše se rotacija zvučne slike i slično.

Postoji veliki broj tehnika dekorelacije signala [6]. U okviru ovog poglavlja će biti predstavljena dekorelacija vremenskim kašnjenjem, svepropusnim FIR filterom i svepropusnim IIR filterom.

2.2.1 Vremensko kašnjenje

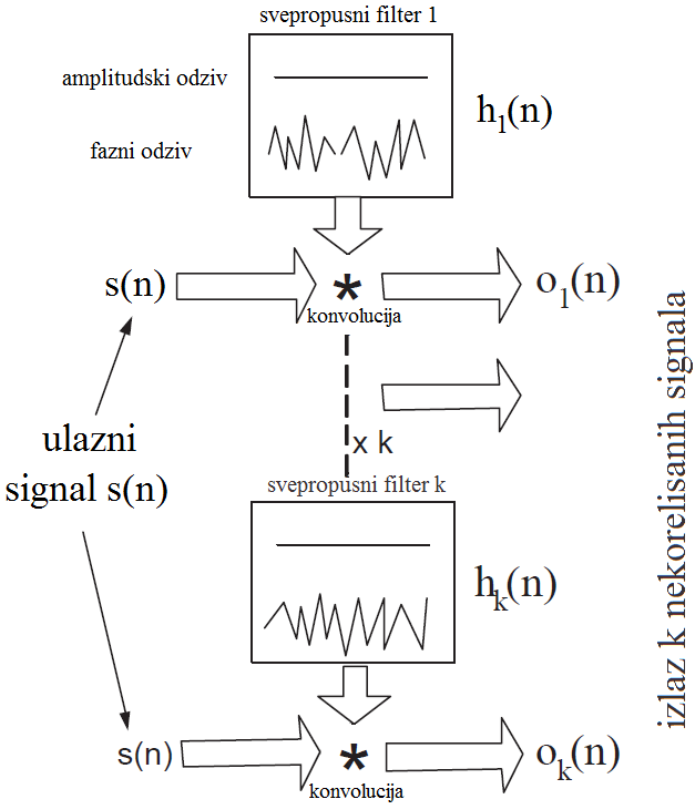
Najjednostavnija forma dekorelacije signala se dobija uvođenjem malog kašnjenja između dve kopije ulaznog signala (Slika 2.5). Iako je jednostavna i jeftina za implementaciju, ova tehnika ima nekoliko nedostataka u slučaju kada se koristi za određivanje opsega izvora zvuka. Kašnjenja koja se uvode između izlaznih signala stvaraju efekat koji menja boju zvuka izvora signala. Ovaj efekat umanjuje delove spektralnog sadržaja izvora pa se zbog manje energije spektra signala opseg izvora zvuka čini manjim. Procena opsega je pogrešna i jer obojenost proizvodi neprirodne i minijaturne izvore zvuka. Tehnika dekorelacije bazirana na kašnjenju je pogodna za kreiranje prostorno proširenih izvora zvuka kada se koriste velika kašnjenja. Kašnjenje je tada dovoljno dugačko da se izbegne eho ili identifikacija replika signala od strane slušalaca.



Slika 2.5 Dekorelacija signala kašnjenjem ulaznog signala

2.2.2 Fiksni svepropusni FIR filteri

Naprednija tehnika dekorelacije signala je pomoću konvolucije ulaznog signala sa svepropusnim FIR filterima koji imaju nasumične šumeće fazne odzive. Rezultujuće faze izlaznih signala proizvode statistički ortogonalne signale (Slika 2.6). Zbog neosetljivosti slušnog sistema na fazne varijacije signala i zbog očuvanosti spektra ulaznog signala, izlazni signali zvuče isto, ali su zbog izmenjene faze dekorelisani.



Slika 2.6 Filter banka za dekorelaciju

Kako bi se dobio FIR filter za dekorelaciju sa odgovarajućim faznim i frekvencijskim odzivima, može da se koristi metoda odabiranja frekvencijskog odziva. Metoda se sastoji iz kreiranja vektora A koji predstavlja odziv amplitude filtera i popunjen je jedinicima jer se koristi svepropusni filter, i vektora B koji predstavlja odziv faze filtera i popunjen je slučajnim signalom koji se kreće izmedju -180 i 180 stepeni. Frekvencijski odziv FIR filtera u kompleksnoj vektorskoj formi se opisuje vektorom H_f:

$$H_f = A.(\cos(B) + j * \sin(B)) \quad (1)$$

Koeficijenti svepropusnog filtera se dobijaju transformacijom H_f iz frekvencijskog u vremenski domen pomoću inverzne FFT:

$$H_n = FFT^{-1}(H_f) \quad (2)$$

Ipak postoji nekoliko problema sa fiksnom FIR tehnikom dekorelacije. Prvo, kod filtera malog reda se ne dobija veliki stepen dekorelacije zbog nedovoljnog mešanja faza. Zatim, odabiranje niže preciznosti i amplitudski odzivi dobijenih filtera za dekorelaciju nisu savršeno ravni što dovodi do obojenosti dekorelisanih signala. S druge strane, ukoliko je red filtera previše veliki, signal je zamućen u vremenskom domenu, što moze biti problematično za signale sa brzim promenama. Kako bi se ograničio ovaj efekat preporučuje se korišćenje filtera dužine kraće od 10ms. Red filtera stoga treba da se izabere tako da se dobije što veća dekorelacija, ali da se ograniči efekat vremenskog zamućenja.

2.2.3 Fiksni svepropusni IIR filteri

Svepropusni IIR filteri za dekorelaciju se izvode iz dobro poznate osobine IIR svepropusnih filtera:

$$H(z) = \frac{a_N + a_{N-1}z^{-1} + a_{N-2}z^{-2} + \cdots + z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \quad (3)$$

Kako bi se kreirali dekorelacioni IIR svepropusni filteri sa slučajnim faznim odzivima, nasumično se postavljaju polovi unutar jedinične kružnice. Zatim se pronalaze polinomijalni koeficijenti prema jednačini za nasumično postavljene polove i uzimaju se a_k koeficijenti filtera. Prednost u odnosu na FIR dekorelacionu tehniku je postizanje jednakog stepena dekorelacija uz pomoć filtera nižeg reda.

2.3 Metode renderovanja virtualnih izvora zvuka

Percepcija zvuka je bazirana na mnoštvu replika koje se razlikuju po vremenu i nivou, i na efektima frekvencijskog odziva nezavisnog od pravca koji prouzrokuje refleksija zvuka u ušnoj školjci, koji su sveukupno opisani pomoću HRTF. Ušna školjka može da se modeluje kao linearan vremenski konstantan sistem koga karakteriše HRTF u frekvencijskom domenu. Koristeći različite audio tehnike je moguće renderovati virtualne zvučne izvore u prostoru pomoću zvučnika ili slušalica. Cilj je da takav sistem reprodukuje isti nivo zvučnog pritiska na bubne opne slušaoca kao da je pravi izvor zvuka na lokaciji gde je smešten virtualni izvor. Kako bi se to postiglo moraju se uzeti u obzir glavne karakteristike lokalizacije zvuka od strane čoveka, koji su bazirani na spektralnim informacijama predstavljenim sa HRTF [7].

Spektralne informacije dobijene iz HRTF mogu da se koriste za implementaciju grupe filtera koji menjaju zvuk na isti način kao HRTF. Rani pokušaji ove metode su bili bazirani na analitičkim proračunima slabljenja i kašnjenja koje glava uvodi u zvučno polje, gde se posmatra pojednostavljen sferni model glave. Novije metode su bazirane na merenjima pojedinačnih ili prosečnih HRTF za svaki željeni pravac virtualnog izvora zvuka. Prednost novijih metoda u odnosu na analitičke je u tome što se uzima u obzir nepravilnost oblika čovekove glave i refleksija od gornjeg dela tela.

U slučaju audio renderovanja kod slušalica se javlja neželjena iskrivljenost zvuka usled anomalija u frekvencijskom odzivu slušalica, koji se uključuje u frekvencijski odziv signala koji stiže do bubnih opni slušaoca. Rešenje je da se ne izvrši samo konvolucija mono signala sa HRTF virtualnog izvora, već i sa filterom koji invertuje frekvencijski odziv slušalica. Ako je frekvencijski odziv slušalica H_p, a HRTF za odgovarajući pravac i uho H_x, inverzija u frekvencijskom domenu može da se izvrši na dva načina, u zavisnosti od toga da li je zvuk mono

ili stereo [8]. U slučaju mono ulaza, odziv inverznog filtera je $H_{inv} = H_x / H_p$. Mono signal (S) se obrađuje ovim filterom zatim prenosnom funkcijom slušalica, tako da je odziv S_e na bubne opne:

$$S_e = H_P H_{inv} S = H_p \frac{H_x}{H_p} S = H_x S \quad (4)$$

što je upravo traženi frekvencijski odziv. U drugom slučaju ulaz je stereo signal S_b koji već sadrži sve neophodne HRTF informacije ($S_b = H_x S$), i tada je neophodno samo invertovati odziv slušalica $H_{inv} = 1 / H_p$. Signal S_e na bubnoj opni je jednak:

$$S_e = H_P H_{inv} S_b = H_p \frac{1}{H_p} S_b = H_x S \quad (5)$$

Zvučnici se mogu takođe koristiti za renderovanje stereo zvuka ili mono zvuka obrađenog sa HRTF. Ipak, kako bi se reprodukovao odgovarajući zvuk do svakog uha, potrebno je da se eliminiše preklapanje zvuka koje se javlja u svakom sistemu zvučnika. Ovo se dešava jer svaki zvučnik ne šalje zvuk samo do uha sa te strane (ipsilateralnog), već i do suprotnog (kontralateralnog) uha. Ukidanje preklapanja se postiže eliminacijom H_{LR} i H_{RL} komponenti, tako da se čini kao da svaki zvučnik reprodukuje zvuk samo za odgovarajuće ipsilateralno uho. Treba uzeti u obzir da su kontralateralne komponente H_{LR} i H_{RL}, kao i ipsilateralne komponente H_{LL} i H_{RR} izvedene iz HRTF i zavise od pozicije zvučnika u odnosu na uši slušaoca, tako da sa promenom pozicije slušaoca se menjaju i vrednosti ovih komponenti. Glavno ograničenje sistema za eliminaciju preklapanja je činjenica da svako pomeranje slušaoca koje je veće od 75-100 mm potpuno narušava željeni prostorni efekat. Ovo ograničenje se može prevazići prilagođavanjem HRTF filtera svakoj novoj poziciji slušaoca.

Postoji nekoliko različitih pristupa eliminaciji preklapanja, kod kojih se uglavnom javlja problem da su neefikasni u slučaju kada se slušalac pomera ili kad nije pozicionaran simetrično u odnosu na zvučnike. Predstavljeno rešenje koristi algoritam za praćenje pozicije glave, gde se filteri računaju u realnom vremenu u zavisnosti od trenutne pozicije slušaoca. Koristimo matričnu prezentaciju za prikaz sistema zvučnika i ušiju, predstavljeni kao sistem sa dva ulaza koji se simultano obrađuju. U frekventnom domenu je sa H_i označena ipsilateralna komponenta, sa H_c kontralateralna komponenta, H_L je HRTF virtalnog zvučnog izvora za levo uho, H_R je HRTF virtalnog zvučnog izvora za desno uho, dok je S mono ulazni signal. Signali E_L i E_R na levoj i desnoj bubnoj opni su opisani narednim izrazom:

$$\begin{bmatrix} E_L \\ E_R \end{bmatrix} = \begin{bmatrix} H_L & 0 \\ 0 & H_R \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \quad (6)$$

Uvođenje kontralateralnih i ipsilateralnih komponenti iz zvučnika uvodi dodatnu prenosnu matricu:

$$\begin{bmatrix} E_L \\ E_R \end{bmatrix} = \begin{bmatrix} H_i & H_c \\ H_c & H_i \end{bmatrix} \begin{bmatrix} H_L & 0 \\ 0 & H_R \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \quad (7)$$

Kako bi se dobili signali opisani izrazom (6) sa sistemom koji daje izlaz opisan izrazom (7), potrebno je izvršiti obradu nad ulazom S koja uvodi inverznu matricu fizičkog sistema zvučnika:

$$\begin{bmatrix} E_L \\ E_R \end{bmatrix} = \begin{bmatrix} H_i & H_c \\ H_c & H_i \end{bmatrix} \begin{bmatrix} H_i & H_c \\ H_c & H_i \end{bmatrix}^{-1} \begin{bmatrix} H_L & 0 \\ 0 & H_R \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \tag{8}$$

Može se primetiti da su (6) i (8) zapravo jednaki. Daljim rešavanjem se dobija:

$$\begin{bmatrix} E_L \\ E_R \end{bmatrix} = \begin{bmatrix} H_i & H_c \\ H_c & H_i \end{bmatrix} \frac{1}{Hi^2} \frac{1}{(1-\frac{H_c^2}{H_i^2})} \begin{bmatrix} H_i & -H_c \\ -H_c & H_i \end{bmatrix} \begin{bmatrix} H_L & 0 \\ 0 & H_R \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \tag{9}$$

Što se na kraju može zapisati kao:

$$\begin{bmatrix} E_L \\ E_R \end{bmatrix} = \begin{bmatrix} H_i & H_c \\ H_c & H_i \end{bmatrix} \begin{bmatrix} 1 & -\frac{H_c}{H_i} \\ -\frac{H_c}{H_i} & 1 \end{bmatrix} \begin{bmatrix} \frac{H_L}{H_i} & 0 \\ 0 & \frac{H_R}{H_i} \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \tag{10}$$

Ako se pretpostavi da je:

$$\frac{1}{(1-\frac{H_c^2}{H_i^2})} \cong 1$$

Ove pretpostavka je bazirana na činjenici da kontralateralna komponenta ima značajno manju energiju od ipsilateralne. Komponente H_L/H_i i H_R/H_i odgovaraju inverziji zvučnika. To jeste, HRTF koji odgovaraju stvarnoj poziciji zvučnika su invertovane jer dodaju spektralne informacije koje nisu u stereo signalu virtualnog izvora. Matrica

$$\begin{bmatrix} 1 & -\frac{H_c}{H_i} \\ -\frac{H_c}{H_i} & 1 \end{bmatrix}$$

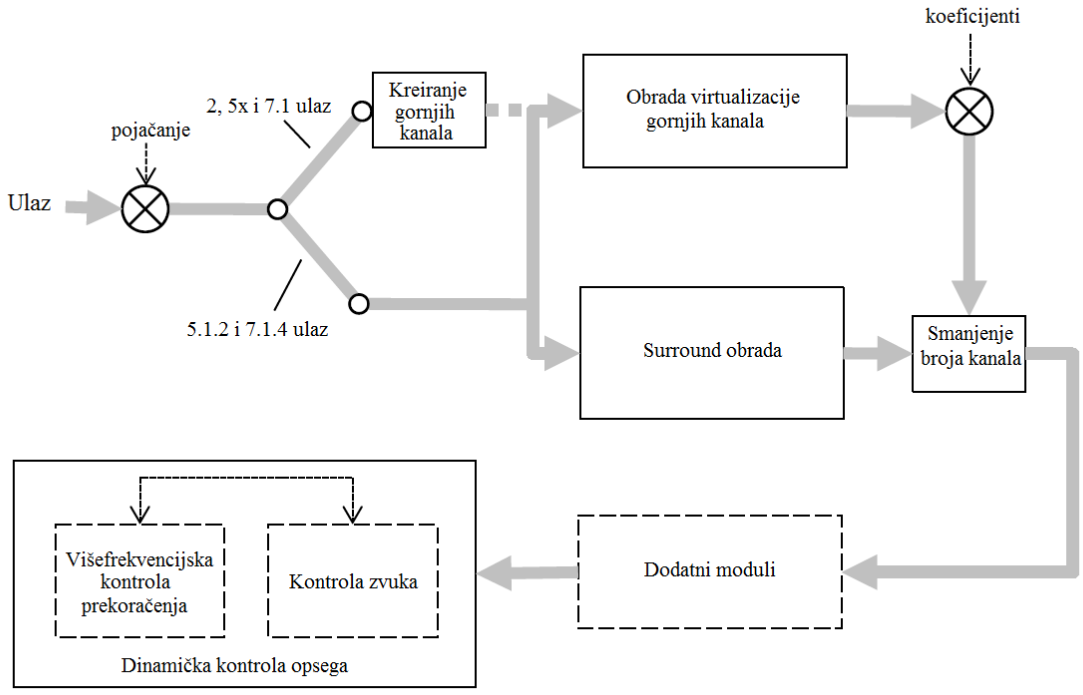
odgovara eliminaciji preklapanja. U ovoj implementaciji su inverzija odziva zvučnika i eliminacija preklapanja usko povezani, ali moraju da se posmatraju kao odvojene komponente. Na kraju, signali X_L i X_R koji se reprodukuju na zvučnicima kako bi se dobio virtualni izvor na željenoj lokaciji su predstavljeni sledećim izrazom:

$$\begin{bmatrix} X_L \\ X_R \end{bmatrix} = \begin{bmatrix} \frac{H_L}{H_i} & -\frac{H_c}{H_i}\frac{H_R}{H_i} \\ -\frac{H_c}{H_i}\frac{H_L}{H_i} & \frac{H_R}{H_i} \end{bmatrix} \begin{bmatrix} S \\ S \end{bmatrix} \tag{11}$$

Mono signal S prolazi kroz ove filtere, nakon čega se svaki kanal prosleđuje na odgovarajući zvučnik. Slično kao kod slušalica je moguće dizajnirati filter tako da prihvata stereo ulaz S_b umesto mono ulaza S. Tada konvolucija sa HRTF parom komponenti H_L i H_R nije potrebna jer signal već sadrži potrebne HRTF informacije.

3. Koncept rešenja

Modul za završnu obradu pruža sveobuhvatno i fleksibilno rešenje koje ima široku primenu. Ovaj sistem omogućava percepciju gornjih kanala bez obzira na odsustvo gornjih zvučnika ili zvučnika koji su usmereni naviše. Takođe pored virtualizacije gornjih kanala, vrši se i horizontalna virtualizaciju u sklopu bloka obrade zvuka iz okruženja. Komponente za kreiranje i smanjivanje broja kanala čine ovaj sistem kompatibilan sa različitim konfiguracijama zvučnika, dok krajnja komponenta za kontrolu opsega na izlazu daje čist zvuk u širokom rasponu frekvencija.



Slika 3.1 Blok dijagram sistema za završnu obradu

Slika 3.1 prikazuje sistem za završnu obradu dekodovanog ulaznog sadržaja raspoređenog po kanalima. Podržane su različite ulazne konfiguracije i u zavisnosti od prisutnosti kanala se vrši odgovarajuća obrada. Centralni modul sistema, čija implementacija na platformi sa ograničenim resursima će biti opisana u sklopu narednog poglavlja, vrši virtualizaciju gornjih kanala. Ukoliko gornji kanali nisu prisutni u ulaznom sadržaju, ovom modulu prethodi modul za kreiranje gornjih kanala na osnovu sadržaja u horizontalnim kanalima.

3.1 Modul za virtualizaciju gornjih kanala

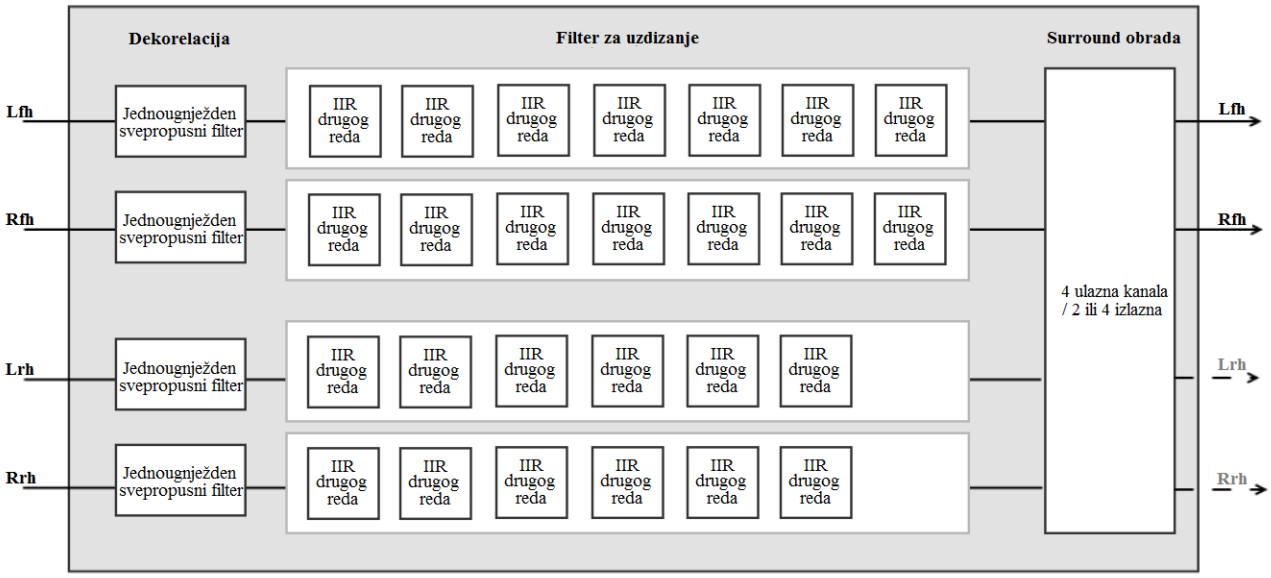
Modul za virtualizacije gornjih kanala omogućava reprodukovanje uzdignutih tonova bez fizičkih gornjih zvucnika ili zvučnika koji su usmereni naviše. Iz tog razloga je ovaj modul kompatibilan sa uobičajenim stereo ili surround sistemima. Virtualizacija gornjih kanala proizvodi traženi efekat nezavisno od horizontalne pozicije. Modul se sastoji iz tri komponente:

• Dekorelacija pomoću jednougnježdenih svepropusnih filtera – unapređuje degradaciju usled pozicije, jačine zvuka ili boje zvuka fantomskog izvora.

• Filter za uzdizanje – izvodi se iz HRTF veličina ili odnosa snaga za virtualizaciju gornjih kanala. Ovaj filter zavisi od azimuta virtualnog izvora zvuka relativnog u odnosu na pravac u kom slušalac gleda.

• Obrada okruženja nad gornjim kanalima – virtualizuje segment okruženja tonova virtualno uzdignutog izvora zvuka.

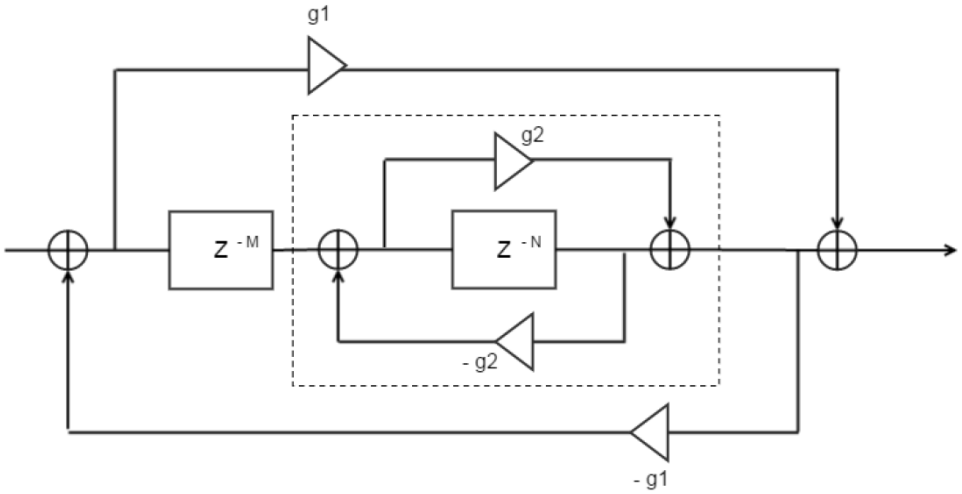
Slika 3.2 prikazuje komponente modula za virtualizaciju gornjih kanala i njihov tok signala. Jednougnježdeni svepropusni filteri se primenjuju nad ulaznim signalima pre virtualizacije. Filteri koji zavise od azimuta se primenjuju kako bi se stvorio virtuelno uzdignut zvuk za date azimute. Na kraju se vrši obrada virtualizacije okruženja nad signalima koji su virtualno uzdignuti.



Slika 3.2 Komponente modula za virtualizaciju gornjih kanala

3.1.1 Obrada međukanalne dekorelacije

Virtualizacija gornjih kanala koristi jednougnježdene svepropusne filtere za međukanalnu dekorelaciju. Kašnjenje i vrednosti koeficijenata su pažljivo usklađeni kako bi se dobio optimalan rezultat. Slika 3.3 prikazuje blok dijagram jednougnježdenog svepropusnog filtera.

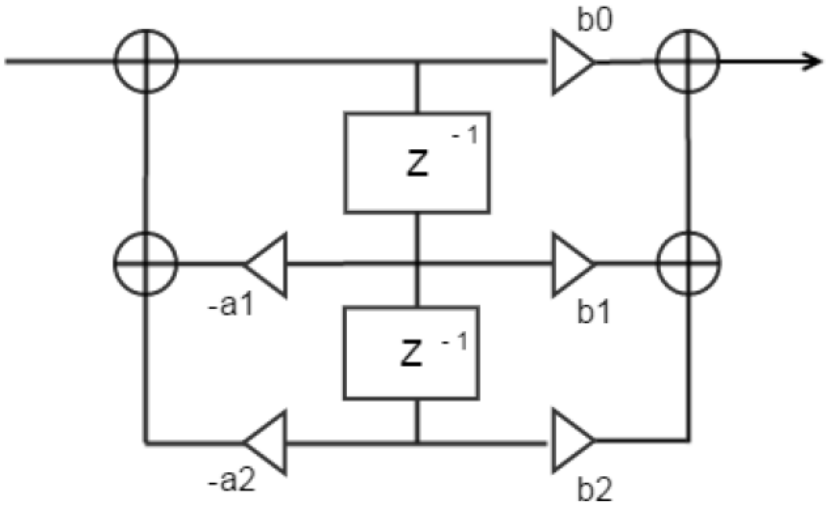


Slika 3.3 Blok dijagram jednougnježdenog svepropusnog filtera

Pošto se filteri koriste za dekorelaciju između dva kanala, parovi kanala (prednji gornji levi i prednji gornji desni kanal, zadnji gornji levi i zadnji gornji desni kanal) imaju različite polarnosti u svojim izlaznim svepropusnim pojačanjima. Različite polarnosti izlaznih pojačanja generišu naizmenične fazne odzive između svakog para kanala kroz ceo frekvencijski spektar. Levi i desni zadnji gornji kanali imaju takođe i Np i Nn filtere, koji su isti kao filteri levog i desnog prednjeg gornjeg kanala.

3.1.2 Filter za virtualno uzdizanje gornjih kanala

Filteri za virtualno uzdizanje gornjih kanala su realizovani pomoću sedam IIR filtera drugog reda za prednje gornje kanale i šest filtera drugog reda za zadnje gornje kanale. I prednji i zadnji filteri gornjih kanala su dizajnirani koristeći HRTF odnose spektralnih veličina za izvor lociran pod uglom od 45 stepeni. Slika 3.4 prikazuje blok dijagram direktne forme IIR filtera drugog reda.



Slika 3.4 Blok dijagram direktne forme IIR filtera drugog reda

Filteri imaju određene segmente koji unose pozitivno pojačanje, gde jedan od pojaseva unosi pojačanje od približno 20dB, ali je vrednost maksimalnog kombinovanog pojačanja 12dB. Kako bi se sprečilo aritmetičko prekoračenje, filteri koji unose negativno pojačanje se primenjuju pre onih koji unose pozitivno pojačanje. Pošto filteri za virtualizaciju gornjih kanala ne unose velika negativna pojačanja ili niska odsecanja, novi redosled neće unositi nikakve dodatne probleme.

Filteri za virtualizaciju gornjih kanala predstavljaju replike za uzdizanje za date azimute, stoga menjanje odziva filtera može dovesti do degradacije efekta uzdizanja. Ukoliko sistem i zvučnici imaju nejednak frekvencijski balans izmedju niskih i visokih frekvencija, pogotovo ako sistem ima više energiju visoke frekvencije nego niske, poželjno je da se smanji energija visoke frekvencije filtera za virtualizaciju. Kontrola zvuka filtera za virtualizaciju gornjih kanala omogućava da se umanji oštrina nastala kombinacijom odziva sistema i filtera za virtualizaciju gornjih kanala.

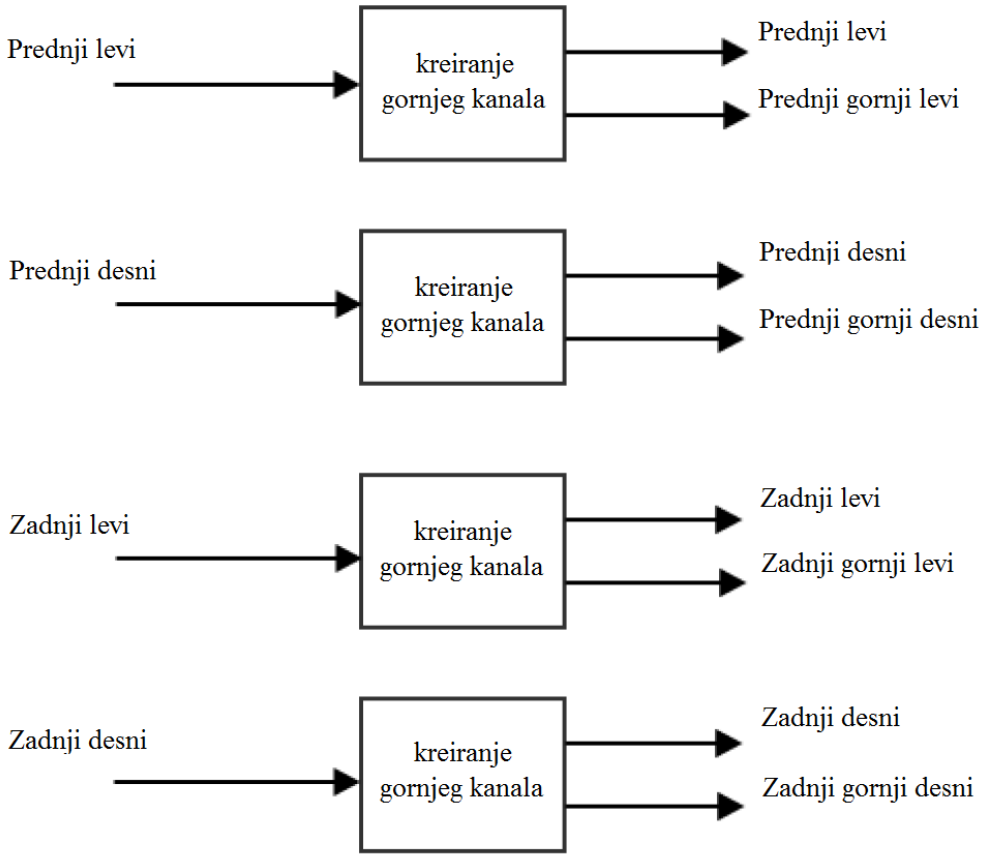
3.1.3 Obrada okruženja nad gornjim kanalima

Uzdignuti signal koji je rezultat dekorelacije i filtriranja uzdiže izvor samo vertikalno. Kako su HRTF odnosi veličina bazirani na datim azimutima, očekuje se da postoje fizicki zvučnici na datim azimutima. Na primer za 7.1.4 ulazni signal, četiri gornja kanala su prednji

levi i desni, i zadnji levi i desni. Filter za virtualizaciju gornjih kanala daje na izlazu četiri kanala koji su virtualno uzdignuti. Tako da ako sistem ima 5.1 izlaznu konfiguraciju, četiri kanala treba da se pošalju na levi, desni, levi izlaz okruženja i desni izlaz okruženja. Obrada okruženja nad gornjim kanalima se koristi kada ne postoje fizički zadnji zvučnici ili su zadnji zvučnici postavljeni napred, tako da uzdignuti signali moraju da budu i horizontalno virtualizovani.

3.2 Modul za kreiranje gornjih kanala

U situacijama kada gornji kanali nisu prisutni u izvornom materiju, a onemogućeno je korišćenje kompleksnijih dekoderskih rešenja za kreiranje potrebnih kanala, koristi se jednostavniji modul u okviru završne obrade koji stvara veštački sadržaj u gornjim kanalima koristeći horizontalne ulazne kanale. Ova obrada se izvršava pre virtualizacije gornjih kanala. Slika 3.5 prikazuje blok dijagram kreiranja gornjih kanala za sve pojedinačne donje kanale prisutne na ulazu.



Slika 3.5 Blok dijagram obrade kreiranja gornjih kanala za sve pojedinačne donje kanale

Za obradu kreiranja gornjih kanala, pri čemu se povećava broj kanala sa četiri na osam, mapa kanala je sledeća:

1. Četiri ulazna kanala:

a) Prednji levi

b) Prednji desni

c) Zadnji levi

d) Zadnji desni

2. Osam izlaznih kanala:

a) Prednji levi

b) Prednji desni

c) Zadnji levi

d) Zadnji desni

e) Prednji gornji levi

f) Prednji gornji desni

g) Zadnji gornji levi

h) Zadnji gornji desni

Za obradu kreiranja gornjih kanala pri čemu se povećava broj kanala sa dva na četiri, mapa kanala je sledeća:

1. Dva ulazna kanala:

a) Prednji levi

b) Prednji desni

2. Četiri izlazna kanala:

a) Prednji levi

b) Prednji desni

c) Prednji gornji levi

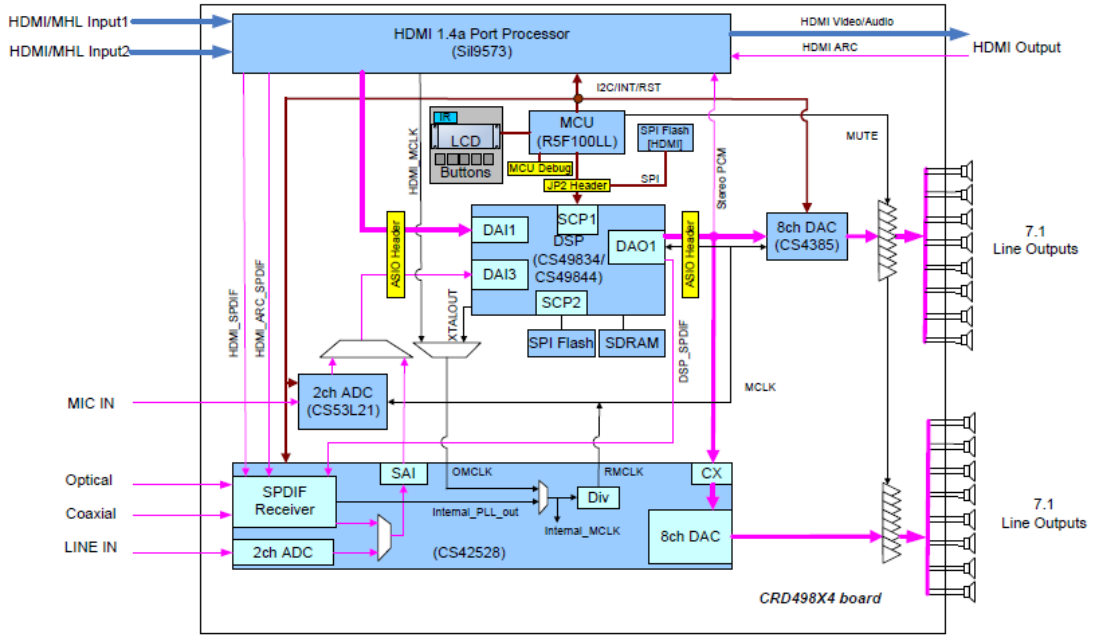
d) Prednji gornji desni

Prvi korak pri kreiranju gornjih kanala je par kvazikomplementarnih niskofrekventnih šelving filtera za slabljenje i pojačavanje koji se primenjuju nad kreiranim i horizontalnim signalima. Filteri su neophodni kako bi se izbegla izobličenja koja bi nastala kada bi se zakašnjen kreiran signal sabrao sa horizontalnim izlazom u modulu za virtualizaciju gornjih kanala. Smanjivanjem energije signala ispod čujne frekvencije (npr. 2kHz) u modulu za kreiranje gornjih kanala, spektralni zarezi, koji su uzrok izobličenja, se uspešno smanjuju. Kvazikomplementarni šelving filter se primenjuje nad horizontalnom komponentom kako bi se kompenzovalo smanjenje niskofrekventne energije prilikom filtriranja nad sabranim izlazom u modulu za virtualizaciju gornjih kanala.

4. Realizacija

U ovom poglavlju dat je opis implementacije sistema za virtualizaciju gornjih kanala čiji je algoritam opisan u prethodnom poglavlju. Zbog svoje arhitekture i resursa, za ciljnu platformu je odabran četvorojezgarni CS49844 procesor Crystal DSP familije kompanije Cirrus Logic [9]. Dat je opis fizičke arhitekture celokupnog sistema i komunikacija mikrokontrolera sa DSP-om, zatim glavne tehničke specifikacije ciljne DSP platforme i metode implementacije modula za virtualizaciju gornjih kanala. Na kraju je dat pregled glavnih modula sistema i njihova raspodela po jezgrima.

4.1 Arhitektura sistema prijemnika audio i video sadržaja



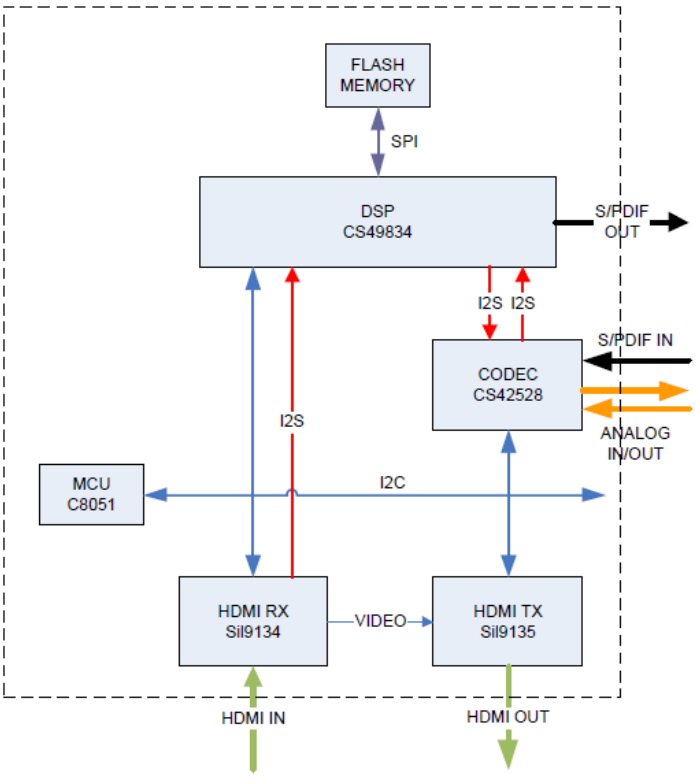
Slika 4.1 Blok dijagram prijemnika audio i video signala zasnovanog na CS49844 procesoru

Prijemnik audio i video sadržaja ima ulogu da primi analogni ili digitalni signal na ulazu, obradi ga (po potrebi dekoduje), pa tako obrađen signal da pretvori u analognu veličinu. Ovaj signal se dalje šalje na audio pojačavač i na kraju se reprodukuje na zvučnicima. Blok šema sistema je data na slici 4.1.

Fizička arhitektura sistema se sastoji iz sledećih glavnih komponenti:

- Četvorojezgarni 32-bitni digitalni signal procesor
- Mikrokontroler
- Audio digitalno-analogni i analogno-digitalni konvertor
- Fleš memorija
- SDRAM memorija

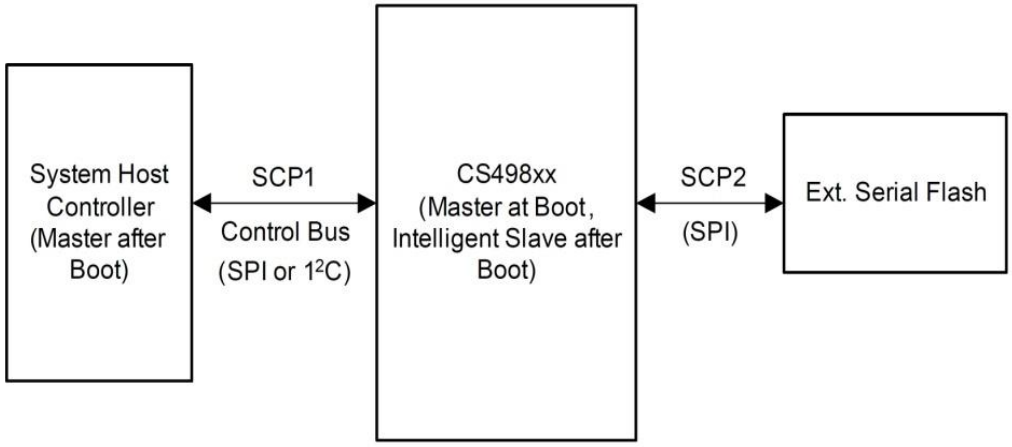
Sistem je povezan sa spoljnim uređajima preko HDMI (*eng. High Definition Multimedia Interface*) (maksimalna brzina 24Mbps) ili SPDIF (*eng. Sony/Philips Digital Interface Format*) (maksimalna brzina 3Mbps) sprege (Slika 4.2), dok se primljeni sadržaj prenosi do i od digitalnog signal procesora preko I2S (*eng. Inter-IC Sound*) (maksimalna brzina 24Mbps) magistrale. DSP i fleš memorija su povezani preko SPI (*eng. Serial peripheral Interface*) (6Mbps) sprege, koja se koristi za učitavanje firmver modula i čuvanje konfiguracionih parametara sistema. Na kraju, I2C (*eng. Inter-Integrated Circuit*) sprega se koristi za konfigurisanje i upravljanje sistemom.



Slika 4.2 Sprege u audio sistemu baziranom na CS49834 DSP

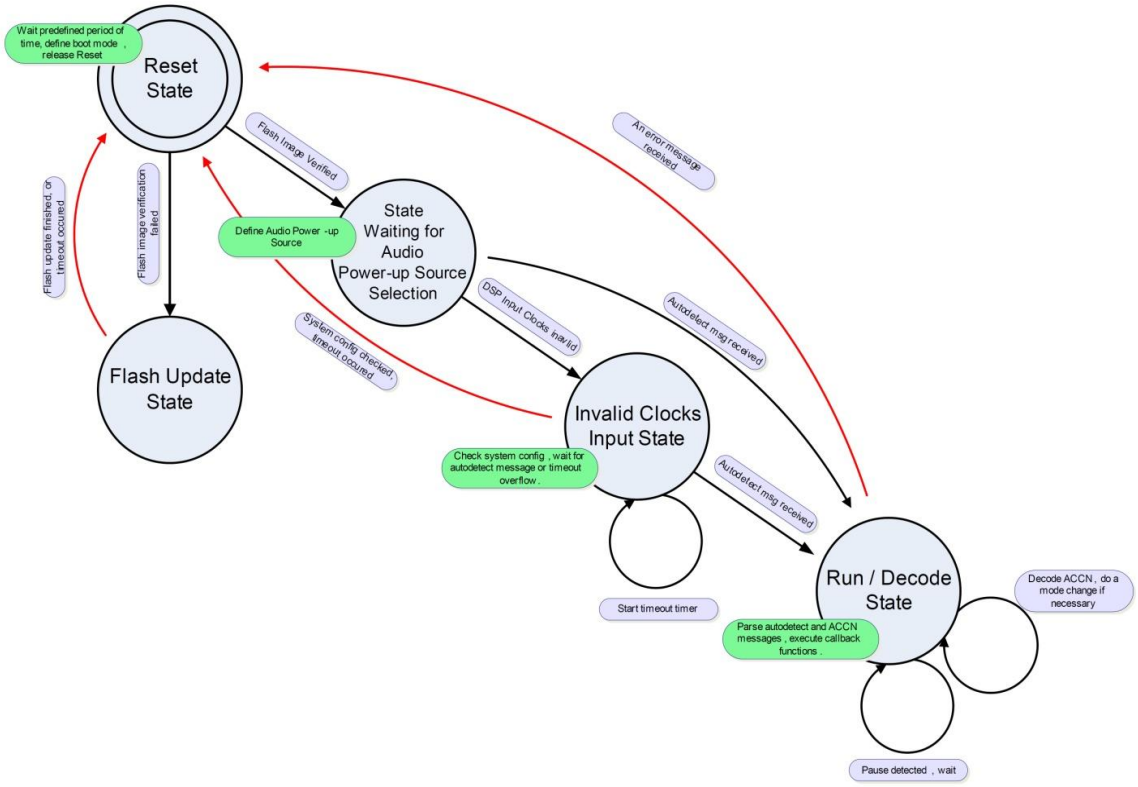
Centralna jedinica obrade audio signala celog sistema predstavlja procesor za digitalnu obradu signala, koga nadgleda zaseban sistemski mikrokontroler (host). DSP je sistem sa ograničenim resursima pa zbog efikasnijeg rada upravljanje, rukovanje korisničkim događajima, podešavanje sistema i sl. obavlja mikrokontroler [10].

U AVR sistemu DSP radi u tzv. master boot modu. Blok dijagram rada DSP-a je prikazan na slici 4.3. Preko prve SPI magistrale je povezan sa mikrokontrolerom, a drugom SPI magistralom je povezan sa eksternom fleš memorijom. DSP je konfigurisan da učita podatke sa eksterne SPI fleš memorije kada izađe iz stanja reseta. Posle inicijalizacije, u toku koje ima glavnu ulogu (Master), DSP se prebacuje u stanje tzv. inteligentnog podređenog uređaja (Intelligent Slave).



Slika 4.3 Blok dijagram rada DSP-a u master-boot modu

Pošto se uključio i inicijalizovao, DSP šalje poruku vezanu za stanje DSP SPI fleš memorije, tj. da li su podaci u njoj ispravni. Ako je sadržaj fleš memorije oštećen, očekuje se ažuriranje. Ako je sve ispravno, na osnovu statusa audio sadržaja, DSP šalje poruku o statusu takta ulaznog signala ili tzv. autodetect i ACCN (*eng. Audio Configuration Change Notification)* poruke (autodetect je poruka na osnovu koje se utvrđuje tip audio toka, u skladu sa standardom). U zavisnosti od ulaznog toka, DSP se prebacuje između stanja dekodovanja i stanja čekanja. Autodetect i opcionalna ACCN poruka se prijavljuju u stanju dekodovanja, dok se poruka za tišinu dobija kada se DSP vrati u stanje za čekanje signala. Slika 4.4 prikazuje stanja DSP-a iz perspektive mikrokontrolera.



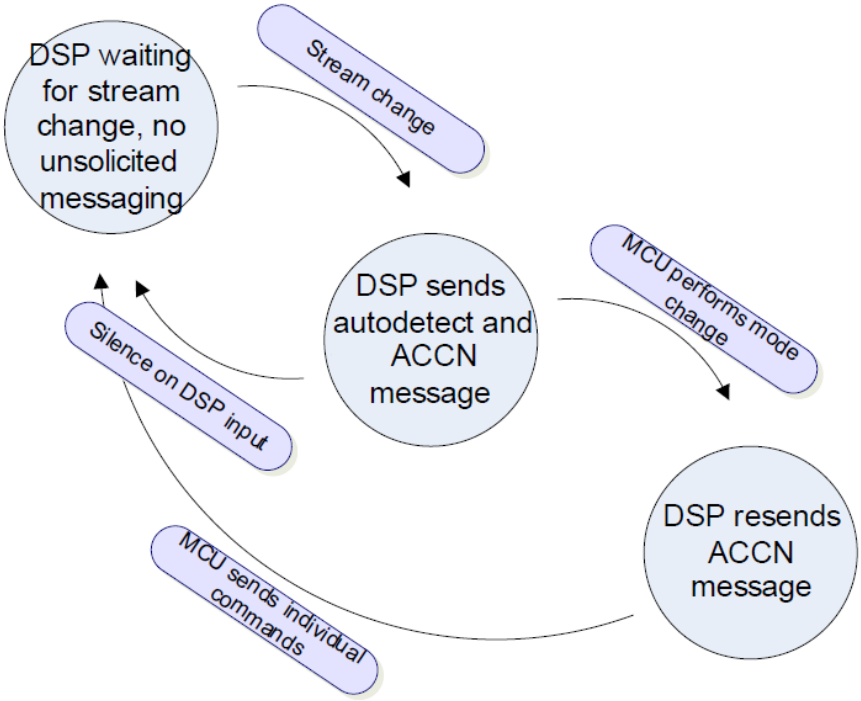
Slika 4.4 Stanja DSP-a sa gledišta mikrokontrolera

Gledano sa strane mikrokontrolera, aktivacijom DSP IRQ signala, DSP poruke se primaju koristeći SPI magistralu i za vreme trajanja DSP IRQ prekidne rutine se pamte u FIFO bafer. Odatle se uzimaju za raščlanjivanje u glavnoj petlji DSP aplikacije. Raščlanivanje DSP poruka je tabelarno. Tabela poruke sadrži kolonu sa identifikacijom tipa poruke (Message ID), kolonu sa očekivanom veličinom poruke (dužina je u 32-bitnim rečima), kao i kolonu sa imenom procedure koja se poziva. Posle pronalaženja rezultata po identifikacionoj oznaci, izvršava se data Callback procedura, koristeći parametar dužine poruke i pokazivač na memorijsku lokaciju koja sadrži podatke prikupljene iz FIFO (*eng. First In First Out*) bafera, kao parametre procedure. Callback procedura analizira njene ulazne parametre i odgovara na DSP poruku koristeći procedure za upis u bafer.

Procedura za raščlanjivanje poruka može uvek da proveri da li ima još pridošlih DSP poruka u FIFO baferu izvršavajući za to predviđenu proceduru. Pošto je ACCN poruka primljena i razložena, mikrokontroler šalje odgovor u dva koraka:

- Izvršavaju se promene DSP režima rada. Nakon što su poslate poruke za promenu režima rada DSP-a, DSP šalje obaveštenje o prijemu ovih poruka slanjem nove ACCN poruke
- Nova ACCN poruka se opet razlaže i šalju se individualne konfiguracione poruke

Dijagram toka komunikacije pri slanju poruka između DSP-a i mikrokontrolera je prikazan na slici 4.5.



Slika 4.5 Komunikacija između DSP-a i mikrokontrolera

Za PCM signal se šalje samo tzv. autodetect poruka, dok se za kompresovan signal, pored autodetect poruke, šalje i ACCN poruka. Dok autodetect poruka sadrži samo informacije o tipu ulaznog audio signala, ACCN poruka sadrži specifične podatke o signalu. ACCN poruke su specifične za same dekodere, od broja reči, do značenja svake reči (ne postoji nikakav standard za njih).

4.2 Arhitektura digitalnog signal procesora

Platforma sa ograničenim resursima na kojoj je vršena implementacija je DSP procesor CS49844 kompanije Cirrus Logic (Slika 4.6), koji poseduje četiri 32-bitna DSP jezgra, koji rade sa aritmetikom u nepokretnom zarezu. Svako jezgro se zasniva na unapređenoj Harvard arhitekturi [11] sa dve memorijske zone za podatke i jednom memorijskom zonom za instrukcije. Ovakva organizacija omogućava paralelan pristup instrukcijama i podacima, čime se postiže velika iskorišćenost resursa datog procesora.

Svako od četiri jezgra obezbeđuje 300 miliona instrukcija u sekundi (MIPS) procesne moći i radi na taktu od 300MHz - može da izvrši dve pomnoži i saberi MAC (*eng. Multiply and Accumulate*) instrukcije, dva čitanja ili upisa iz/u memoriju i dva povećanja/umanjenja adresnih registara u jednom taktu. Jedno jezgro poseduje 60 kilo reči memorije u svakoj od memorijskih zona, gde je reč dužine 32 bita. U tabeli 4.1 je dat pregled resursa.

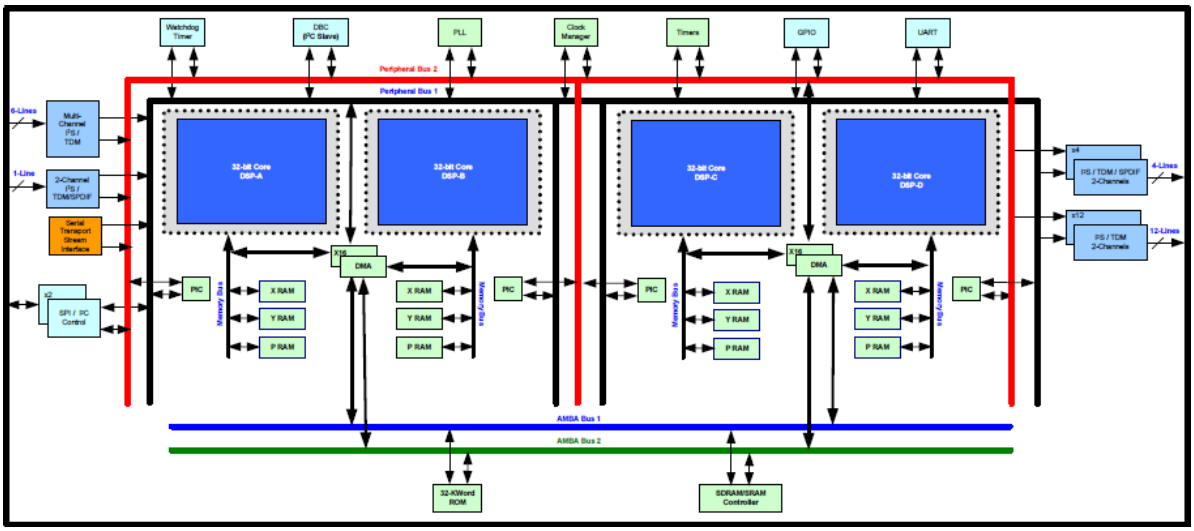
Takođe, svako jezgro sadrži osam 32-bitnih registara opšte namene (x0-x3 i y0-y3), osam 72-bitnih registara (tzv. akumulatora, a0-a3 i b0-b3) za smeštanje međurezultata, dvostruku

preciznost i dr., dvanaest indeksnih registara (i0-i11), veličine 16 bita, namenjenih za adresiranje kod čitanja i upisa iz/u memoriju, i njima odgovarajuće modulo registre (nm0, nm1, ..., nm11), koji omogućava da odgovarajući registar radi u modulo režimu (korisno kod kružnih bafera) kao i bit-reverznom režimu (vrlo korisno kod algoritama koji zahtevaju izuzetno brzo izvršenje, kao što su brze furijeove transformacije).

Svako od jezgara sadrži i SRS jedinicu (*eng. Shift Round Saturate*) koja upravlja aritmetičkim pomeranjem, zaokruživanjem i saturacijom između akumulatora i registara. Sprega između jezgara i spoljašnje memorije ostvarena je upotrebom kontrolera za direktan pristup memoriji (*eng. Direct Memory Access, DMA*). DMA ima osobinu da može da vrši prenos podataka između perifernih uređaja, spoljašnje memorije ili unutrašnje memorije jezgra, bez intervencije jezgara, što oslobađa više resursa za samu obradu signala.

resursi	DSP A	DSP B	DSP C	DSP D
Procesorsko vreme	300 MIPS	300 MIPS	300 MIPS	300 MIPS
Memorija za podatke (X memorija)	60 kilo reči	60 kilo reči	60 kilo reči	60 kilo reči
Memorija za podatke (Y memorija)	60 kilo reči	60 kilo reči	60 kilo reči	60 kilo reči
Programska memorija (P memorija)	60 kilo reči	60 kilo reči	60 kilo reči	60 kilo reči

Tabela 4.1 Količina raspoloživih resursa ciljne platforme



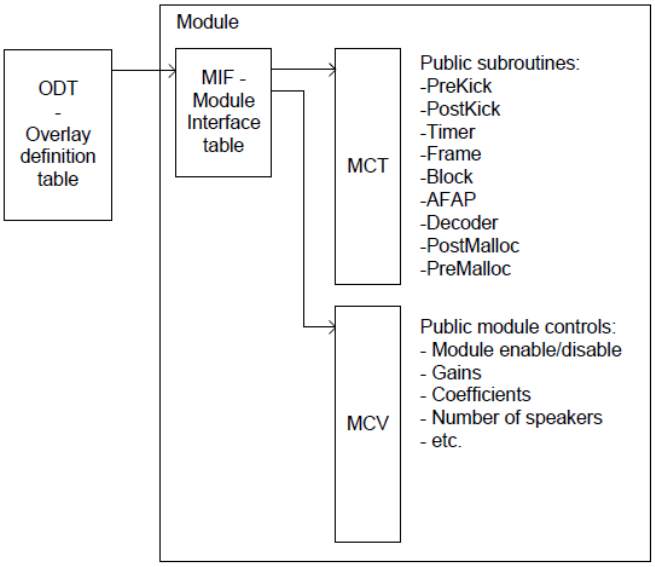
Slika 4.6 Blok dijagram CS49844 DSP procesora

4.3 Radni okvir procesora

Cirrus Logic radni okvir (framework) predstavlja sistemski softver procesora koji skraćuje vreme za razvoj aplikacije, uvodeći neke od ideja i metodologija iz objektno orijentisanog programiranja u svet asemblerskog koda.

Jezgro radnog okvira se sastoji od jednostavnog operativnog sistema (OS), čija je glavna uloga da bude raspoređivač za određen broj procesnih entiteta (modula). Uslovno rečeno, OS predstavlja monitorsku petlju koja poziva rutine odgovarajućih modula po unapred definisanom redosledu.

Moduli su definisani kao objekti sastavljeni od rutina i podataka, u skladu sa radnim okruženjem. Svaki modul ima svoj jedinstveni sprežni podsistem (*Module Interface* – MIF), kojim je modul povezan sa OS-om. Njega čini MIF tabela koja sadrži pokazivače na tabele sa ostalim sprežnim informacijama. Dve najvažnije tabele su MCT tabela (*Module Call Table*), i MCV tabela (*Module Control Vector*). MCT tabela je niz od devet elemenata – pokazivača na osnovne javne (public) rutine. Redosled elemenata u tabeli je unapred definisan, a ukoliko neka od rutina nije definisana za dati modul, na mestu njenog pokazivača se nalazi nula. Ove rutine poziva OS kao odgovor na pojavu odgovarajućih događaja u sistemu (eng. event handles). MCV tabela predstavlja niz javno dostupnih konfiguracionih parametara datog modula, i ona omogućava konfigurisanje modula od strane glavnog kontrolera uređaja (host). Struktura ove tabele nema neku unapred definisanu formu i programeru je prepušteno da formira njen sadržaj i strukturu. Sa OS strane, sprega ka modulima se sastoji od ODT tabele (Overlay Definition Table) koja sadrži pokazivače na MIF tabele svih učitanih modula. Slika 4.7 prikazuje spregu modula sa operativnim sistemom.



Slika 4.7 Blok dijagram sprege modula sa operativnim sistemom

Rutine koje se nalaze u MCT tabeli su jedine funkcije modula kojima OS pristupa. Svaka od njih ima svoju specifičnu namenu i njihovim pozivanjem od strane OS-a, modul odgovara na događaje u sistemu. Prva rutina je Pre-Kickstart rutina, koju OS poziva samo nakon prijema inicijalizacione poruke (reset) i pre uspostavljanja komunikacije sa sistemskim kontrolerom. Ona omogućava inicijalizaciju modula, prvenstveno elemenata MCV tabele na njihove podrazumevane vrednosti. Nakon što su izvršene Pre-Kickstart rutine svih učitanih modula, OS po prijemu kickstart poruke, nastavlja se sa pozivima Post-Kickstart rutina. Post-Kickstart rutina omogućava obradu konfiguracionih podataka prosleđenih modulu od strane kontrolera. Nakon što su izvršene Post-Kickstart rutine svih modula, završena je inicijalizaciona faza i OS prelazi na izvršavanje normalnog ciklusa izvršavanja modula.

Pre-Malloc rutina se poziva prvi put nakon završetka inicijalizacionih rutina ukoliko je bilo koji modul u sistemu tokom inicijalizacije od OS-a zatražio inicijalizaciju dinamički dodeljene memorije. Takođe, ona može biti inicirana slanjem zahteva za reinicijalizaciju OS-a. S druge strane, Post-Malloc rutina se poziva nakon završetka alokacije dinamičke memorije. Ova rutina omogućava inicijalizaciju dinamički dodeljene memorije modula. Pre/Post-Malloc rutine se po potrebi izvršavaju pre prvog poziva Block i Frame rutina. Timer rutina se poziva periodično na svakih N milisekundi (podrazumevano je 1ms) kao odgovor na prekid (interrupt) generisan od strane brojača realnog vremena. Izvršavanje Block i Frame rutine je upravljano ulaznim tokom podataka pri čemu se Block rutina izvršava pri prijemu svakih 16 PCM odbiraka u U/I nizu, dok se Frame rutina izvršava na svakih N blokova. Ovaj broj blokova zavisi od dekoderskog modula u sistemu, odnosno od njegove jedinice dekodovanja (decoding frame). AFAP je skraćenica od "As Fast As Possible". Ova rutina se poziva kada god se desi neki događaj u sistemu, naravno, uz uslov da ne prekida druge rutine istog prioriteta. AFAP, Timer, Frame, i Block rutine čine takozvanu Foreground thread (nit). Background rutina se izvršava periodično u pozadinskoj niti (Background thread), koja ima niži prioritet od Foreground niti i izvršava se kao pozadinski proces.

Pored dela za rukovanje događajima, važan deo radnog okvira čini sistemski ulazno/izlazni memorijski niz koji služi za smeštanje audio podataka koji ulaze u sistem, nad kojima moduli vrše obradu i koji potom izlaze iz sistema.

4.4 Metodologija razvoja sistema na DSP platformi

Sistem na ciljnoj platformi se razvija na osnovu komercijalnog algoritma implementiranog u programskom jeziku C. Glavni koraci tokom razvoja sistema su analiza referentnog koda, modifikacija algoritma u skladu sa ciljnom platformom i optimizacija kritičnih delova koda.

Prvi korak u razvoju sistema, analiza referentnog koda, podrazumeva razumevanje uloge implementirane funkcije ili modula, kao i uočavanje glavnih struktura podataka i memorijskih skladišta. Prilikom pisanja referentnog koda se najviše obraća pažnja na ispravnost algoritma, tako da se tokom analize referentnog koda uočavaju segmenti koji će se funkcionalno optimizovati i prilagoditi ciljnoj platformi.

Nakon analize se prelazi na funkcionalnu optimizaciju u asemblerskom programskom jeziku. Ove optimizacije podrazumevaju organizaciju podataka, prilagođenje pristupa podacima i optimizaciju programskih petlji.

Organizacija podataka na ciljnoj platformi podrazumeva korišćenje akumulatora i registara za privremeno skladištenje podataka i prosleđivanje parametara funkcije. Na ovaj način se štede memorijske lokacije koje bi zauzimale dodatne promenljive, ali i instrukcije koje bi bile neophodne za smeštanje vrednosti u promenljive pre poziva funkcije i njihovo preuzimanje na početku funkcije. U referentnom kodu se često koriste velike strukture u kojima se opisuje trenutno stanje aplikacije ili pojedinačnih procesnih modula. Poželjno je uočiti takve strukture i napraviti njihove globalne instance u asemblerskom kodu. Kada se funkciji prosleđuje pokazivač na strukturu, potrebno je obratiti pažnju da li je u kodu realizovana jedna ili više struktura. Samo ukoliko se koristi ista struktura, moguće joj je neposredno pristupati u telu funkcije.

Jedno od hardverskih proširenja digitalnih signal procesora je i jedinica za generisanje adresa podataka. Adresni generator je u stanju da obavi izvestan broj aritmetičkih operacija i da generiše adresu na kojoj se nalazi podatak. Po pravilu podržava kružno ili modulo adresiranje, kao i režim bit-obrnutog pristupa koji se koristi kod FFT-a. Iz ovog razloga se posredno pristupanje elementima niza preko indeksa menja pristupom preko pokazivača na element.

Modul za virtualizaciju gornjih kanala sadrži veliki broj digitalnih filtera, gde se najveći deo procesorskih ciklusa tokom izvršavanja DSP aplikacije troši se na izvršavanje programskih petlji. Ušteda jednog instrukcionog ciklusa unutar tela petlje dovodi do uštede N instrukcijskih ciklusa, gde N predstavlja broj iteracija petlje po jedinici obrade.

Deo koda unutar programskih petlji ponekad je nezavisan od trenutne iteracije ili čitave petlje, pa nema potrebe da se izvršava više puta. Ovaj kod naziva se nezavisni, odnosno invarijantni kod. Primer ovakvog koda jeste dodeljivanje konstantne vrednosti nekom resursu. Umesto da se ta dodela obavlja u svakoj iteraciji petlje, može se izvršiti samo jednom, pre ulaska u petlju. Ovaj pristup produžava životni vek promenljivih i dodatno opterećuje resurse, ali ubrzava izvršenje celokupnog koda jer skraćuje telo petlji.

Ciljna platforma je 32-bitna platforma sa aritmetikom u nepokretnom zarezu, pri čemu je opseg vrednosti tipova u nepokretnim zarezom [-1, 1). Konstante koje su van ovog opsega je potrebno skalirati, da bi se omogućile aritmetičke operacije nad ispravnim vrednostima.

Skaliranje se vrši promenom prezentacije brojeva u nepokretnom zarezu. Konstante koje se nalaze u intervalu od [1,2) treba da podelimo sa 2, a konstante koje se nalaze u intervalu od [2,3) treba da podelimo sa 4.

Akumulatorski registri čuvaju međurezultate MAC instrukcija, kao i drugih aritmetičkih operacija. Ovi registri su po pravilu veći od dužine reči množača (2 x n) za m zaštitnih bita, ali pored toga može da dođe do prekoračenja maksimalne vrednosti koja može da se smesti u akumulator. Kako bi se ovo izbeglo, moguće je uvesti bezbednosni opseg (headroom), odnosno, skalirati vrednosti signala na ulazu u blok odbrade (na opseg manji od maksimalne veličine tipova za podatke), a na izlazu iz bloka izvršiti inverzno skaliranje [12].

Nakon implementacije u asemblerskom programskom jeziku i provere ispravnosti rešenja se prelazi na dodatnu optimizaciju kritičnih delova koda, kao što su programske petlje ili funkcije koje se pozivaju veliki broj puta. Ove optimizacije podrazumevaju korišćenje pogodnosti arhitekture digitalnog signal procesora i njegovih hardverskih proširenja.

4.5 Metode optimizacije na ciljnoj platformi

Prva metoda optimizacije kojom se štedi procesorsko vreme predstavlja paralelizacija instrukcija, koju omogućava postojanje dve odvojene memorije za podatke i zasebne memorije za instrukcije. Instrukcije koju mogu da se pišu u paraleli su množenje, MAC instrukcije, instrukcije na nivou bita i kombinacija istih.

U paraleli može da se izvrši jedno pomeranje iz X memorije i jedno iz Y memorije, ograničenja pri tome su da X memorija može da se adresira samo korišćenjem i0 i i1 registara, Y memorija može da se adresira pomoću registara i4 i i5, za pomeranje iz X memorije se koriste samo X registri i A akumulatori, dok za pomeranje iz Y memorije se koriste Y registri i B akumulatori. Akumulatori mogu da budu samo izvori, a registri podataka mogu da budu samo destinacija. Primer takvih instrukcija:

```
x0 = xmem[i0]; i0+=n; y0 = ymem[i4]; i4+=n
xmem[i1] = a3; i1+=n; y3 = ymem[i4]
```

Takođe, pomeranje iz jednog u drugi registar podataka i između registara podataka i akumulatora može da se izvršava u paraleli. Ako su obe destinacije registri podataka, jedan mora biti X registar, a drugi Y registar, takođe ako su oba izvora akumulatori, jedan mora biti A akumulator, a drugi B. Ukoliko su i izvor i destinacija akumulator ili registar podataka, moraju da imaju podudarne indekse.

```
ymem[i4]=b2; b2=a2
x0=y0; b3=a3
```

Aritmetičke operacije i MAC instrukcije mogu da se podese tako da se više operacija izvršava u paraleli:

a0=x2*x3;b0=-y2*x3

a1=a0+=x0*y2;b1=b0-=y0*y2

a1=a0-x2*x1; b1=b0-y2*x1

a1=a0=a0+x1*y1; b1=b0=b0+y1*y1

Dve operacije na nivou bita mogu da se izvršavaju u paraleli ukoliko se u jednoj instrukciji koristi A akumulator kao destinacija, a u drugoj B akumulator i ukoliko su indeksi ispravno podešeni:

a0 =~ a1;b0 =~ b1

a0 = a0 | b3;b0 = b0 | a3

a0 = a0 ^ a3;b0 = b0 ^ b3

a1 = a1 & a3;b1 = b1 & b3

a0 = a0 << 8;b0 = b0 << 8

Moguće je koristiti logexp operaciju za aproksimaciju normalizacije, logaritmovanja, deljenja, stepenovanja i sl. Te operacije su uklopljene u pipeline, koji daje rezultat nakon dva ciklusa. Bez korišćenja logexp, gorepomenute operacije troše i nekoliko desetina puta više instrukcija.

Normalizacija:

logexp X=nop(norm64(x0)) Y=nop(norm32(y0))

nop

x0, y0 = logexp

Logaritamska funkcija:

logexp X = log(norm32(x0)) Y = log(norm32(y0))

nop

x0, y0 = logexp

Eksponencijalna funkcija:

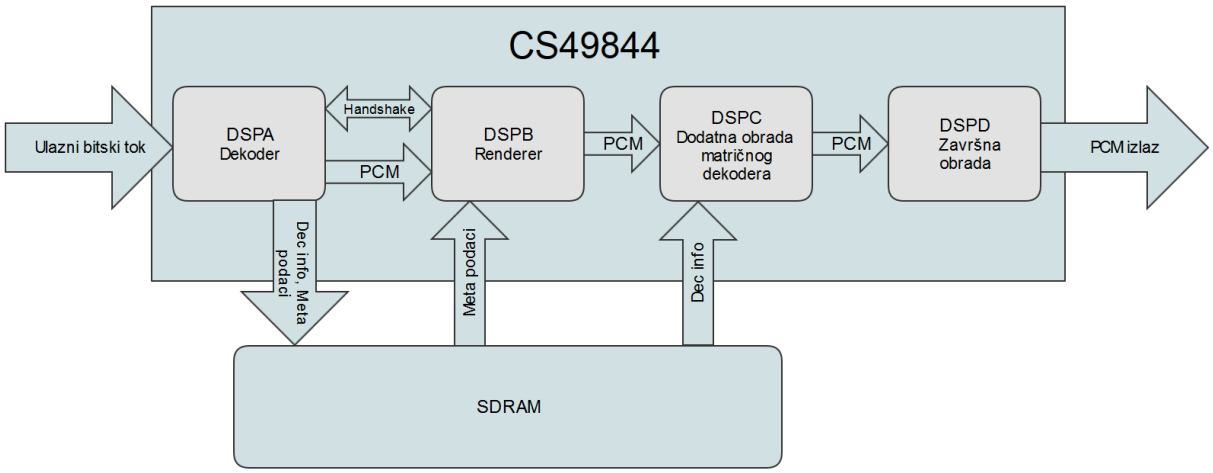
logexp X = exp(x0) Y = nop(x0)

nop

x1, y1 = logexp

4.6 Moduli implementiranog sistema

Celokupni implementirani sistem sastoji se iz tri modula: dekoder, matrični dekoder i završna obrada. Vrlo je složen te je i pored optimizovanog koda i pažljivog trošenja resursa vrlo zahtevan, kako memorijski tako i u pogledu procesorskog vremena. Iz tog razloga zauzima sva četiri jezgra CS48944 procesora. Na slici 4.8 je prikazana raspodela modula po jezgrima. Dekoder kao prvi nivo obrade smešten je na DSPA jezgru koje prima ulazni kompresovani tok podataka. Na DSPB i DSPC jezgru smešten je matrični dekoder koji vrši renderovanje i podešavanje konfiguracije zvučnika i mapiranje kanala. Na DSPD jezgru izvršavaju se različite završne obrade signala kao što je kontrola jačine zvuka, kontrola prekoračenja i obrada okruženja. U okviru modula na DSPD jezgru se nalazi i blok za kreiranje i virtualizaciju gornjih kanala. Pored datih modula, za svako od jezgara postoji i zaseban operativni sistem.



Slika 4.8 Raspodela modula implementiranog sistema na jezgra ciljne platforme

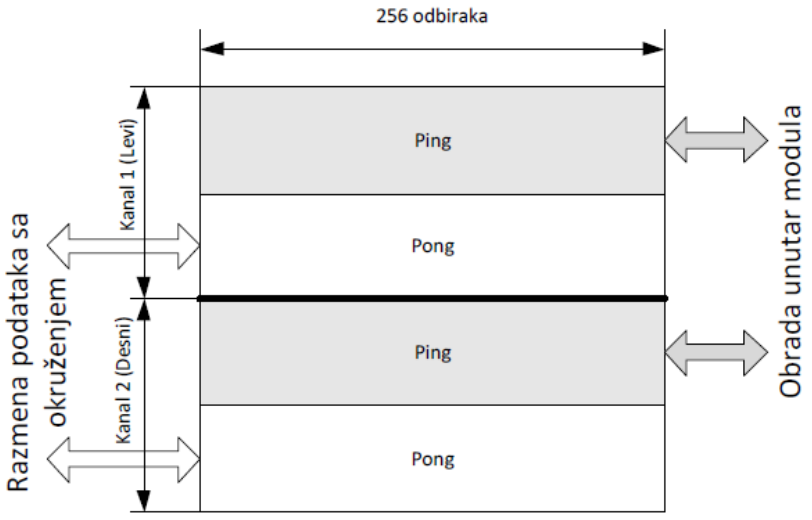
- **DSPA jezgro**

Jezgro DSPA prima ulazni bitski tok koji nosi kodovani sadržaj sa HDMI ili SPDIF serijske sprege. On se smešta u internu Y memoriju ovog jezgra - FIFO memoriju, veličine 16k reči i dodatnih 24k reči eksterne memorije. Kako ulazni bitski tok neprekidno pristiže, u slučaju da je interna FIFO memorija puna, ulazni podaci se smeštaju u eksternu memoriju, zatim se prebacuju u internu kada je ona oslobođena - za ovo se brine OS.

Prvi nivo obrade ulaznog signala jeste dekodovanje koje obavlja dekoder. Zavisno od ulaznog bitskog toka jedan od dostupnih dekodera će se učitati na DSPA jezgru. Ovim upravlja autodetekcija u OS-u i mikrokontroler. Za smeštanje i prenos PCM podataka odgovoran je sistemski ulazno/izlazni niz (*eng. I/O buffer*) alociran od strane operativnog sistema u Y memoriji, izdeljen na 32 segmenta, po 128 lokacija, za svaki od podržanih kanala. Svi moduli mogu pristupiti ovom nizu, putem pokazivača na svaki od kanala koje OS redovno ažurira.

Međutim, dekodovani podaci se od strane dekodera prvo smeštaju u interni PCM memorijski niz, odakle se kopiraju u ulazno/izlazni niz. Razlozi za ovo su da IO memorijski niz može da uskladišti samo 128 PCM odbiraka po kanalu dok se dekodovanjem minimalne jedinice jednog ulaznog bitskog toka može dobiti više odbiraka, kao i da se izbegne čekanje da dekoder završi sa dekodovanjem jedne minimalne jedinice dekodovanja pre nego što se pozovu moduli iz druga dva nivoa obrade.

Da bi se zadovoljila ova dva uslova, interni PCM memorijski niz ima posebnu strukturu. Ukoliko bi dužina internog PCM memorijskog niza bila jednaka jedinici dekodovanja, pre nego što se počne sa dekodovanjem druge celine, sistem bi morao da čeka da se svi PCM-ovi oba kanala prebace u IO memorijski niz i da se nad njima završi sva obrada definisana u nivou međuobrade i nivou završne obrade. Da bi se čekanje izbeglo, koristi se pristup koji se naziva dvostruko skladištenje podataka (eng. double buffering). Ovaj pristup podrazumeva da se istovremeno vrši učitavanje novih podataka i obrada nad već učitanim podacima, što podrazumeva da je veličina internog PCM memorijskog niza dvostruko veća od jedinice obrade dekodera, po kanalu. Ovakva struktura se naziva Ping-Pong memorijski niz i prikazana je na slici 4.9. Svaki kanal ima dva dela, Ping i Pong. Dok dekoder dekoduje i upisuje sadržaj u Ping deo kanala, Pong deo kanala se prosleđuje na ulazno-izlazni memorijski niz na dalju obradu drugim modulima. Kada dekoder završi i popuni Ping delove kanala, Ping i Pong menjaju uloge i sada se dekodovani sadržaj upisuje u Pong deo, dok se upisani PCM podaci iz Ping dela upisuju u ulazno/izlazni memorijski niz i na taj način prosleđuju drugim modulima.



Slika 4.9 Ping-Pong struktura memorijskog niza

- **DSPB i DSPC jezgro**

Matrični dekoder je smešten na DSPB i DSPC jezgru. Blok renderer smešten je na DSPB jezgru. Ovaj blok inicijalno, pri uključenju uređaja, u odnosu na izlaznu konfiguraciju zvučnika koju dobija preko MCV tabele OS-a a čiju vrednost prijavljuje dekoder, preračunava skup mogućih tačaka u prostoru za smeštanje objekata, nakon čega se taj isti skup zajedno sa meta podacima koriste za raspoređivanje objekata na zadatu izlaznu konfiguraciju zvučnika. Ova inicijalizacija traje približno 180 000 000 instrukcija, odnosno oko 0,6 sekundi. Kako se obrada u bloku renderer može započeti tek nakon ove inicijalizacije, DSPB mora obavestiti DSPA jezgro kada je inicijalizacija završena. To je znak da je modul spreman za primanje podataka i obradu, nakon čega DSPA započinje slanje meta podataka u SDRAM memoriju. Ovaj handshake između DSPA i DSPB jezgra kontroliše OS preko odgovarajućih polja svoje MCV tabele i dešava se samo nakon pomenute inicijalizacije renderera – jednom nakon pokretanja sistema.

Nakon što primi meta podatke od strane DSPA jezgra, jezgro DSPB započinje obradu - uklanjanje objekata iz ulaznog kanalnog sadržaja i njihovo dodavanje izlaznom sadržaju zvučnika. Na DSPC jezgru nalazi se obrada signala kao što je spuštanje ili povećanje izlaznog broja kanala u skladu sa ciljnom konfiguracijom zvučnika, prilagođavanje zvučnog sadržaja različitom prostornom rasporedu zvučnika, normalizovanje nivoa signala. Ovaj modul takođe zahteva određene informacije od strane dekodera kao što su izlazna konfiguracija zvučnika, frekvencija signala, zatim opseg izlaznog signala, i slično. Prenos ovih informacija obavlja se na isti način kao i prenos podataka između DSPA i DSPB jezgra, preko SDRAM memorije.

- **DSPD jezgro**

Na poslednjem DSP jezgru se vrši završna obrada. Na ulazu su dobijeni PCM odbirci raspoređeni po kanalima. U zavisnosti od prosleđenih MCV kontrola se vrše odgovarajuće obrade. Ovaj modul omogućava stvaranje gornjih kanala ukoliko oni nisu napravljeni tokom obrade u okviru matričnog dekodera. Takođe, blokovi za obradu okruženja i virtualizaciju gornjih kanala vrše horizontalnu i vertikalnu virtualizaciju. Kontrola opsega s druge strane na izlazu daje čist zvuk u širokom rasponu frekvencija. Obrađeni odbirci na izlazu iz modula završne obrade su spremni za reprodukovanje na zvučnicima.

5. Testiranje i verifikacija

Testiranje i verifikacija sistema predstavljaju krajnji korak u implementaciji sistema. Svrha ovog procesa je da se pokaže tačnost implementacije algoritma na ciljnoj platformi, tako što se utvrdi da se generisani izlazi iz implementiranog sistema slažu sa izlazima generisanim iz referentnog koda - referentnim izlazima. U toku ovog procesa osim detektovanja grešaka u softveru proverava se i kvalitet softvera.

U okviru ovog rada urađeno je funkcionalno ispitivanje metodom crne kutije – BBT [13] (*eng. Black Box Testing*). BBT alat je razvijen na Odseku za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu i predstavlja moćan alat za ispitivanje i verifikaciju. Generalno rečeno, BBT alat je automatizovana i daleko naprednija verzija batch skripti. Prednosti BBT alata nad batch skriptama su brojne, od relativno jednostavne sintakse za pisanje ispitnih slučajeva, samim tim i bržeg razvoja istih, do lakog korišćenja, prenosivosti, mnoštva ugrađenih alata koji pojednostavljuju proces ispitivanja, sve do preglednih izveštaja o rezultatima. Ovaj alat dakle omogućava brz razvoj ispitnih slučajeva, a ugrađeni alati omogućavaju izvršavanje širokog spektra operacija. Jednom razvijeni ispitni slučajevi mogu se ponoviti neograničeni broj puta, kao i na proizvoljnom računaru (što je ponekad problem kod običnih batch skripti).

Od proizvođača tehnologije koja je bila predmet ispitivanja dobijena je dokumentacija koja predstavlja ispitnu proceduru. Ova dokumentacija sadrži specifikacije testnih slučajeva, ulazne podatke i ostale dokumente potrebne za verifikaciju implementacije testirane audio tehnologije. Ova dokumentacija je dostupna samo softverskim kompanijama koje žele da verifikuju svoju implementaciju i da dobiju sertifikat od kompanije koja je licencirala tu tehnologiju.

Početni korak u fazi razvoja testova podrazumeva organizaciju direktorijuma radnog okruženja. Pošto i testni okvir i BBT alat imaju određene specifičnosti i unose određene zahteve, radno okruženje mora biti organizovano tako da odgovori sve zahteve.

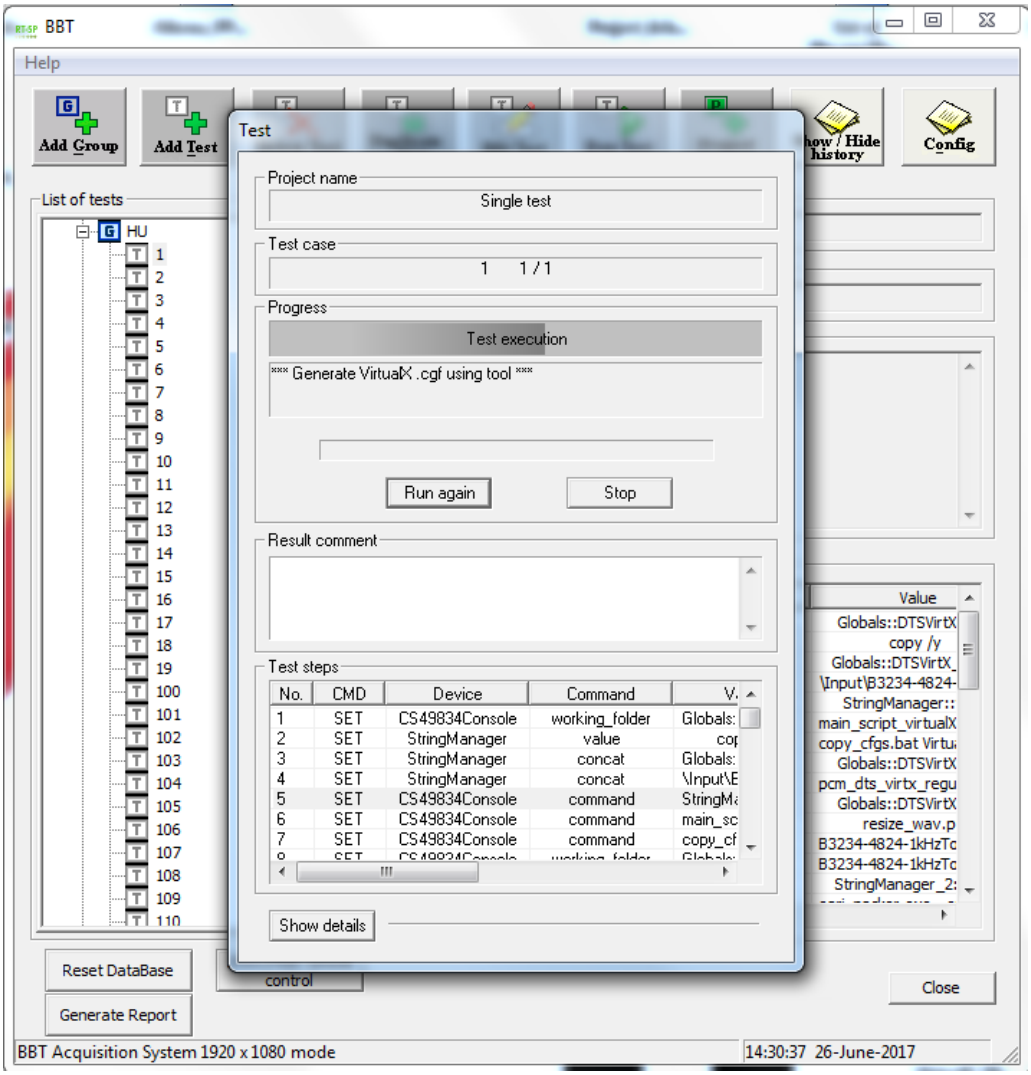
Uz svaki testni slučaj date su i tekstualne datoteke u kojima se nalaze sve informacije po kojima je određeni testni slučaj karakterističan. U ovim datotekama navedeni su ulazni testni vektori, vrednosti konfiguracionih parametara i drugo. Kako bi se sve ove informacije, za sve testne slučajeve, nalazile sumirane na jednom mestu, napisana je Python skripta koja parsira prethodno pomenute datoteke i kreira Excel tabelu. U ovoj tabeli, u prvoj koloni izlistani su svi testni slučajevi, dok su u narednim kolonama upisane vrednosti svih parametara neophodnih za odgovarajući testni slučaj. Dalje se ova Excel tabela prosleđuje skripti u kojoj je implementirana logika pomoću koje se generišu BBT testovi. Uzima se jedna po jedna linija, odnosno željeni testni slučaj sa svim njegovim parametrima i generišu se testni slučajevi za BBT alat (.tst datoteke) čiji je deo prikazan na slici 5.1, kao i konfiguracioni fajlovi (.cfg datoteke).

```
[test]
ID = 1
name = 1
description = 1 test file
log_file = 1.log
////////////////////////////////////////////////////////////
//
//   Copy input stream to tools folder
//
////////////////////////////////////////////////////////////

[step]
description = go to the Tools folder
device = CS49834Console
command = working_folder
value = Globals::DTSVirtX_tools
delay = 0
option = [SET]

[step]
description = Copy VirtualX input streams to Tools folder
device = StringManager
command = value
value = copy /y
delay = 0
option = [SET]

[step]
description = Setup path to the input stream
device = StringManager
command = concat
value = Globals::DTSVirtX_Inputs
delay = 0
option = [SET]

[step]
description =
device = StringManager
command = concat
value = \Input\B3234-4824-1kHzTone-20dB-LR0deg.wav
delay = 0
option = [SET]

[step]
description = Copy test stream temporary to tools folder
device = CS49834Console
command = command
value = StringManager::value
delay = 2000
option = [SET]

[step]
```

Slika 5.1 Primer generisanog testnog slučaja za BBT alat (deo .tst datoteke)

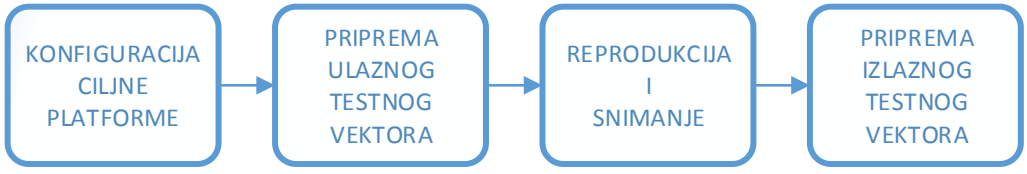
Ukupno je korišćen 1076 test, od kojih je sa 126 testova testiran modul za virtualizaciju gornjih kanala, dok se sa 110 testova verifikovala funkcionalnost modula za kreiranje gornjih kanala. Testovi su podeljeni u 8 grupa, gde svaka grupa odgovara određenoj funkcionalnosti sistema.

Nakon generisanja testova, pokretanjem BBT aplikacije, Slika 5.2, počinje i samo izvršavanje testova.



Slika 5.2 BBT aplikacija nakon pokretanja

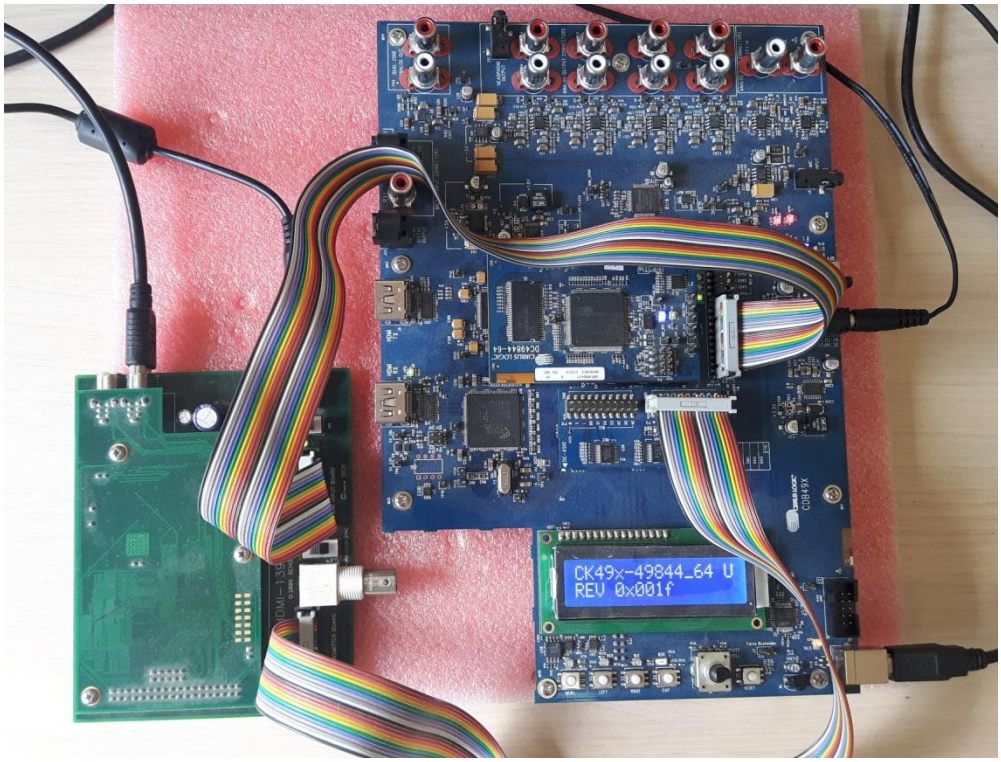
Na sledećoj slici je prikazan blok dijagram koji ilustruje osnovne korake jednog BBT testnog slučaja, koji redom prate korake date u .tst datotekama.



Slika 5.3 Blok dijagram izvršavanja jednog BBT testa

Prvi korak tiče se konfiguracije ciljne platforme, odnosno „spuštanja“ svih izvršnih datoteka koje su potrebne (firmware svih komponenti koje se paralelno izvršavaju) na procesor. U ovom koraku se takođe vrši i pozivanje konfiguracionih datoteka, odnosno komandi koje su u njima navedene, a pomoću kojih se uključuju odnosno isključuju pojedine funkcionalnosti, šalju vrednosti pojedinih parametara, kao što su razna pojačanja, vrednosti graničnih frekvencija i tako dalje.

Na slici 5.4 dat je primer testnog okruženja. Kao ciljni procesor koristi se već pomenuti CS48944, četvorojezgarni digitalni signal procesor iz Crystal DSP familije procesora firme *Cirrus Logic*, a za reprodukciju i snimanje signala korišćena je zvučna kartica Echo Audio. Na desnoj strani slike je razvojna ploča CDB49X, u čijem centralnom delu je pozicioniran digitalni signal procesor, a sa leve strane se nalazi Echo Audio zvučna kartica.



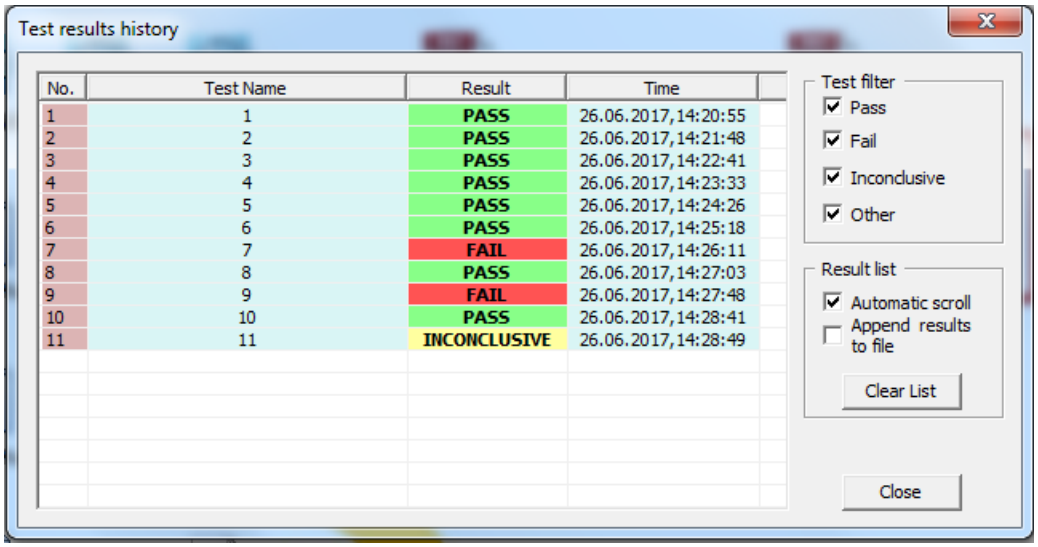
Slika 5.4 Ispitno okruženje

Sledeći korak podrazumeva sve aktivnosti vezane za pripremu ulaznog testnog vektora. U nekim slučajevima, ulazni testni vektor se nalazi u formatu koji se ne može reprodukovati a da se prethodno ne dekoduje. Ponekad se dešava da je neophodno povećati frekvenciju odabiranja kod ulaznog testnog vektora.

U narednom koraku vrše se podešavanja Echo Audio zvučne kartice preko koje se obavlja reprodukcija i snimanje. Zvučna kartica koja je korišćena u ovom radu ograničena je da može reprodukovati testne vektore koji sadrže maksimalno osam kanala. Ukoliko ulazni testni vektor sadrži više od osam kanala, reprodukcija i snimanje se mora izvoditi u nekoliko iteracija.

Poslednji korak odnosi se na aktivnosti nakon snimanja izlaznog testnog vektora. Najčešće su to vremensko poravnanje snimljenog izlaza koje se vrši umetanjem sihronizacione reči, kombinovanje testnih vektora snimljenih kroz nekoliko iteracija u jedan izlazni testni vektor i slično.

Mogući ishodi izvršavanja BBT testova su *PASS*, *FAIL* i *INCOLCLUSIVE*. Ukoliko se svi koraci predviđeni testom uspešno izvrše, ishod testa biće *PASS*, sa druge strane, ukoliko se desi da prilikom izvršavanja nekog od koraka predviđenih testom dođe do greške ishod testa biće *FAIL*. Ukoliko se, pak, desi da ishod izvršavanja testa bude *INCOLCUSIVE* to znači da je izvršavanje testa prekinuto pre nego što su izvršeni svi predviđeni koraci. Primer prikaza ishoda izvršavanja testova generisanog od strane BBT alata dat je na slici 5.5.



Slika 5.5 Prikaz ishoda izvršavanja testova

Da bi se omogućilo potpuno automatizovano ispitivanje prihvatljivosti implementiranog rešenja, proizvođač tehnologije je obezbedio i posebnu softversku komponentu, odnosno testni okvir, koja omogućuje spregu pomoću komandne linije (eng. Command-Line Interface - CLI) za izvršavanje ispitivanja. Osim za ispitivanje prihvatljivosti, testni okvir se koristi i za generisanje referentnih signala, koji će biti upoređivani sa snimljenim signalima. Testni okvir je potpuno automatizovan i ne zahteva dalju interakciju sa korisnikom nakon pokretanja.

Ispitivanje rezultata završava se pokretanjem glavne skripte koja generiše izveštaj na unapred definisanoj lokaciji, u obliku HTML datoteke. Izveštaj ima formu tabele, a najbitnija polja su: jedinstveni identifikator testnog slučaja, metoda evaluacije koja je upotrebljena u konkretnom testnom slučaju i status - *PASSED*, *FAILED*, *UNRESOLVED*. Ukoliko je status testa FAILED ili UNRESOLVED, potrebno je preduzeti određene koraka, i ispitati razlog zbog koga test ne prolazi.

Dobijeni rezultati nakon izvršene opisane procedure testiranja zadovoljili su unapred definisane kriterijume za verifikaciju, status svakog od 1076 testnog slučaja je *PASSED*, i na taj način pokazano je da sistem odgovara zahtevima.

Nakon završetka implementacije i provere ispravnosti rešenja je urađena procena utroška memorije DSP procesora kao i procesorskog vremena potrebnog za obradu u modulu za virtualizaciju gornjih kanala i modulu za kreiranje gornjih kanala. Utrošak memorije je izražen kroz broj zauzetih reči, odnosno 32-bitnih memorijskih lokacija. S druge strane, iako je u radnom okruženju moguće dobiti samo broj utrošenih ciklusa, za svaki od testnih slučajeva je pomoću sledeće formule izmeren utrošak procesorskog vremena u milionima instrukcija po sekundi:

$$MIPS = \frac{brojCiklusa * \frac{Fs}{BLOCK\_SIZE}}{1000000} \quad (12)$$

Parametar Fs predstavlja frekvenciju odabiranja, broj odbiraka u jednoj sekundi ulaznog signala, koja u testovima iznosi 48kHz. BLOCK_SIZE predstavlja veličinu bloka obrade, koja iznosi 256 odbiraka.U tabeli 5.1 je prikazan utrošak X i Y memorije podataka i P programske memorije za implementirane module za virtualizaciju i kreiranje gornjih kanal, kao i za celokupan modul za završnu obradu. U poslednjem redu su date vrednosti maksimalnog utroška procesorskog vremena u milionima instrukcija po sekundi za pomenute module. Sve vrednosti su u okviru grainica dostupnih resursa za jedno jezgro digitalnog signal procesora.

	Modul za virtualizaciju gornjih kanala	**Modul za kreiranje gornjih kanala**	**Celokupan modul za završnu obradu**
X memorija	1366	223	18774
Y memorija	1076	4108	19605
P memorija	458	370	14605
Max MIPS	24,65	14,70	290,65

Tabela 5.1 Utrošak procesorskog vremena i memorije za implementirane module i celu završnu obradu

6. Zaključak

U ovom radu realizovan je modul za virtualizaciju i kreiranje gornjih kanala na višejezgarnoj platformi sa ograničenim resursima, CS49844 DSP procesorom firme Cirrus Logic.

Prvi korak u implementaciji je bio analiza referentnog C koda, razumevanje algoritma korišćenog u obradi i uočavanje glavnih struktura podataka. Nakon toga se prešlo na implementaciju u asembleru, koja je podrazumevala funkcionalne optimizacije kao što je promena pristupa podacima i smanjivanje broja instrukcija u programskim petljama. Modul implementiran u asembleru je testiran i posle uklanjanja grešaka se radilo na daljoj optimizaciji kritičnih delova koda. Poslednji korak je bio završno testiranje i verifikacija.

Rešenje je verifikovano koristeći namenski alat za automatsko izvršavanje testova (BBT) koji radi na principu crne kutije. Nakon uspešno izvršene unapred definisane testne procedure koja se sastoji od 1076 testnih slučajeva, a potom i analize dobijenih rezultata, zaključeno je da sistem daje iste rezultate kao i referentni kod. Ispravno implementiran sistem je zatim dalje analiziran računanjem ukupne potrošnje procesorskog vremena i memorije. Dobijene vrednosti odgovaraju dostupnim resursima DSP-a tako da je obrada u realnom vremenu omogućena.

Dalja unapređenja se odnose na proširenje sistema. Kako bi se omogućila obrada većeg broja različitih formata ulaznih tokova podataka, potrebno je proširiti dekoderski modul i modul za renderovanje. S druge strane, podrška za reprodukovanje zvuka na sistemima sa različitim konfiguracijama zvučnika se omogućava daljim razvojem modula za završnu obradu, pre svega unapređenjem komponenti za podešavanje izlaznog broja kanala i komponente za virtualizaciju.

7. Literatura

[1] Dion Hanson: *The History of Sound in the Cinema*, Cinema Technology Magazine, July/August 1998, pp. 8-13.

[2] Dolby, *Dolby digital – the sound of the future here today*, 1998.

[3] Dolby Laboratories Inc: *Surround Sound Past, Present, and Future*, 1999.

[4] Guillaume Potard: *3D-audio object oriented coding*, PhD thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2006. http://ro.uow.edu.au/theses/539

[5] Florian Hollerweger: *An Introduction to Higher Order Ambisonic*, 2008.

[6] Guillaume Potard, Ian Burnett: *Decorrelation techniques for the rendering of apparent sound source width in 3d audio displays*, Proceedings of the 7th Int. Conference on Digital Audio Effects, October 2004, pp. 280-284.

[7] Jakes Bejoy: *Virtual surround sound implementation using deccorrelation filters and HRTF*, Center for computer research in music and acoustics, Stanford University, 2009.

[8] A. Mouchtaris, P. Reveliotis, C. Kyriakakis: *Inverse Filter Design for Immersive Audio Rendering Over Loudspeakers*, IEEE Transactions on Multimedia, Vol. 2, No. 2, June 2000, pp. 77-87.

[9] Cirrus Logic Inc: *CS49844 Data Sheet*, May 2012.

[10] Savo Gavrilović: *Jedno rešenje softvera za prijemnik audio i video sadržaja*, Zbornik radova Fakulteta tehničkih nauka, godina XXXI, broj 5/2016, pp. 801-804.

[11] V. Kovačević, M. Temerinac, M. Popović, N. Teslić, *Arhitekture i algoritmi DSP-a I*, Novi Sad, FTN, 2004.

[12] J. Kovačević, D. Bokan, *Arhitekture i algoritmi digitalnih signal procesora, Zbirka zadataka i laboratorijski priručnik*, Novi Sad, 2016.

[13] M. Popović, J. Kovačević, *A Statistical Approach to Model-Based Robustness Testing*, ECBS '07 Proceedings of the 14th Annual IEEE International Conference and Workshop on Engineering of Computer Based Systems, March 2007, pp. 485-494.

[14] Owner's Manual Version 2.2 for Windows, *Echo Digital Audio Corporation*, September 2009.