



**УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**

Одсек за електротехнику и рачунарство

Институт за рачунарство и аутоматику

Катедра за рачунарску технику и рачунарске комуникације

**Једна реализација вишеканалног аудио кодера у
оквиру корисничког пријемника за дигиталну
телевизију**

- Дипломски рад из предмета-
- Архитектуре и алгоритми ДСП II -

Ментор:
доц. др Јелена Ковачевић

Кандидат:
Ђорђе Пешут, е11062

Нови Сад, фебруар 2011.



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 публикација	
Тип записа, ТЗ:	Текстуални штампани материјал	
Врста рада, ВР:	Дипломски рад	
Аутор, АУ:	Ђорђе Пешут	
Ментор, МН:	доц. др Јелена Ковачевић	
Наслов рада, НР:	Једна реализација вишеканалног аудио кодера у оквиру корисничког пријемника за дигиталну телевизију	
Језик публикације, ЈП:	Српски	
Језик извода, ЈИ:	Српски	
Земља публиковања, ЗП:	Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2011	
Издавач, ИЗ:		
Место и адреса, МА:	Нови Сад	
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилог)		
Научна област, НО:	Електротехника и рачунарска техника	
Научна дисциплина, НД:		
Предметна одредница/Кључне речи, ПО:	Аудио кодер, транскодер, кориснички пријемник за дигиталну телевизију	
УДК		
Чува се, ЧУ:	У библиотеци ФТН, Нови Сад, Трг Доситеја Обрадовића 6	
Важна напомена, ВН:		
Извод, ИЗ:	У раду је дат опис реализације вишеканалног аудио кодера у оквиру корисничког пријемника за дигиталну телевизију на дигиталном сигнал процесору	
Датум прихватања теме, ДП:		
Датум одбране, ДО:		
Чланови комисије, КО:	Председник:	
	Члан:	Потпис ментора
	Члан, ментор:	



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

KEY WORDS DOCUMENTATION

Accession number, ANO :		
Identification number, INO :		
Document type, DT :		Monographic's publication
Type of record, TR :		Word printed record
Contents code, CC :		Master thesis
Author, AU :		Đorđe Pešut
Mentor, MN :		Jelena Kovačević, PhD
Title, TI :		One implementation of multichannel audio encoder for set-top box
Language of text, LT :		Serbian
Language of abstract, LA :		Serbian
Country of publication, CP :		Serbia
Locality of publication, LP :		Vojvodina
Publication year, PY :		2010
Publisher, PB :		
Publication place, PP :		Novi Sad
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)		
Scientific field, SF :		Electrical engineering and computer science
Scientific discipline, SD :		
Subject/Key words, S/KW :		Audio encoder, transcoder, set-top box
UC		
Holding data, HD :		Library of FTN, Novi Sad, Trg Dositeja Obradovića 6
Note, N :		
Abstract, AB :		This paper gives one implementation of multichannel audio encoder for set-top box on digital signal processor
Accepted by the Scientific Board on, ASB :		
Defended on, DE :		
Defended Board, DB :	President:	
	Member:	Mentor's sign
	Member, Mentor:	

САДРЖАЈ

1.	Увод.....	7
2.	Теоријске основе рада.....	9
	2.1 Дискретизација континуалних сигнала.....	10
	2.2 Основни принципи компресије аудио сигнала.....	11
	2.2.1 Компресија аудио сигнала заснована на тромости људског уха.....	11
	2.2.2 Компресија аудио сигнала без губитака.....	17
	2.3 Дигитални сигнал процесори.....	18
	2.3.1 Предности дигиталних сигнал процесора у обради аудио сигнала у односу на друга решења.....	19
	2.3.2 Особине дигиталних сигнал процесора.....	20
	2.4 Кориснички пријемник за дигиталну телевизију.....	21
	2.5 Критеријуми за оцену квалитета модификованог алгорита.....	22
3.	Анализа проблема.....	24
	3.1 Опис кодера.....	24
	3.1.1 Филтер банка.....	27
	3.1.2 Анализа тренутних амплитудних промена.....	27
	3.1.3 ADPCM.....	27
	3.1.4 Фактори скалирања и нормализација одбирака.....	29
	3.1.5 Додела бита.....	29
	3.1.6 Скаларна квантизација.....	30
	3.1.7 Канал за репродукцију ниских фреквенција (<i>LFE</i>).....	30
	3.2 Конвертовање аритметике покретног зареза у аритметику непокретног зареза	30
	3.3 Проблеми при писању асемблерског кода.....	32
4.	Реализација.....	33
	4.1 Опис циљне платформе.....	34
	4.1.1 Опис развојних алата.....	36
	4.2 Методологија развоја програмске подршке на наменској платформи.....	38
	4.2.1 Модел 0.....	38
	4.2.2 Модел 1.....	38
	4.2.2.1 Замена приступања елементима низа преко индекса приступањем преко показивача.....	38
	4.2.2.2 Смањивање броја аргумената функције.....	39
	4.2.2.3 Редуковање структура.....	39
	4.2.3 Модел 2.....	39
	4.2.4 Оптимизација програмске подршке.....	39
5.	Испитивање и верификација.....	41
	5.1 Резултати испитивања.....	42
6.	Закључак.....	44
7.	Литература.....	45

СПИСАК СЛИКА

Слика 1.1. Блок дијаграм транскодера.....	8
Слика 2.1 Дискретизација сигнала по времену (одмеравање): (а) континуалан сигнал, (б) сигнал дискретизован по времену.....	10
Слика 2.2 Илустрација квантовања континуалног сигнала.....	11
Слика 2.3 Апсолутна граница чујности у тихом окружењу.....	12
Слика 2.4 Критични опсези: (а) приказани у <i>Bark</i> скали, (б) приказани у фреквентној скали.....	14
Слика 2.5 Пример који илуструје случај када шум маскира тон.....	15
Слика 2.6 Пример који илуструје случај када тон маскира шум.....	16
Слика 2.7 Ширење маскирања на суседне критичне опсеге.....	17
Слика 2.8 Компресија сигнала без губитака у комбинацији са компресијом са губицима.....	18
Слика 2.9 Компресија сигнала без губитака.....	18
Слика 2.10 Харвард архитектура.....	19
Слика 2.11 Упоредни приказ платформи за имплементацију ДСП алгоритама [13]...	19
Слика 2.12 кориснички пријемник за дигиталну телевизију.....	21
Слика 2.13 Референтни дизајн FTA / PayTV корисничког пријемника за дигиталну телевизију.....	21
Слика 3.3 Структура улазно/излазног оквира.....	26
Слика 4.1 Графички приказ фаза реализације.....	33
Слика 4.2 Детаљни приказ имплементације вишеканалног аудио кодера [24].....	34
Слика 4.3 блок шема <i>ZDSP5</i> процесора.....	35
Слика 4.4 спрежни подсистеми <i>ZDSP5</i> процесора.....	36
Слика 4.5 <i>ADP</i> развојно окружење.....	37
Слика 5.1 Статистишке информације разлике сигнала.....	42

СПИСАК ТАБЕЛА

Табела 2.1 Идеализована група филтера за тачке назначене на слици 2.4.....	14
Табела 4.1 Потребан број <i>MIPS</i> -а за сваку од функција (после оптимизације).....	40
Табела 5.1 Субјективна скала заснована на ITU-R BS.1284 стандарду у опсегу од 1.0 до 5.0 [28].....	42
Табела 5.2 Резултати испитивања.....	42

СКРАЋЕНИЦЕ

DSP	- Процесор за дигиталну обраду сигнала (енг. <i>Digital Signal Processor</i>)
STB	- Кориснички пријемник за дигиталну телевизију (енг. <i>Set top box</i>)
MPEG	- Група стручњака за покретне слике (енг. <i>Moving Pictures Experts Group</i>)
CD	- Оптички медиј за складиштење података (енг. <i>Compact Disc</i>)
SNR	- Однос сигнал-шум (енг. <i>Signal to Noise Ratio</i>)
PSNR	- Вршни однос сигнала и шума (енг. <i>Peak Signal to Noise Ratio</i>)
MAC	- Јединица за множење и сабирање са акумулацијом (енг. <i>Multiply And Accumulate</i>)
DVD	- Оптички медијум за складиштење сигнала високог квалитета (енг. <i>Digital Versatile Disc</i>)
SMR	- Однос сигнала и шума који се маскира (енг. <i>Signal to Mask Ratio</i>)
AAC	- Стандард за компресију аудио сигнала (енг. <i>Advanced Audio Coding</i>)
WMA	- Стандард за компресију аудио сигнала (енг. <i>Windows Media Audio</i>)
DTS	- Стандард за компресију аудио сигнала (енг. <i>Digital Theatre System</i>)
HDTV	- Телевизија високе дефиниције (енг. <i>High Definition TV</i>)
ATH	- Апсолутни праг чујности (енг. <i>Absolute Threshold Of Hearing</i>)
PNS	- Замена шума заснована на несавршености људског уха (енг. <i>Perceptual Noise Substitution</i>)
SBR	- Проширење спектралног опсега (енг. <i>Spectral Band Replica</i>)
NS	- Временско обликовање шума (енг. <i>Temporal Noise Shaping</i>)
LTP	- Предвиђање на дуге стазе (енг. <i>Long Term Prediction</i>)
CPU	- Централна процесорска јединица (енг. <i>Central Processor Unit</i>)
RAM	- Меморија са случајним приступом (енг. <i>Random Access Memory</i>)
ROM	- Меморија из које се подаци могу само читати (енг. <i>Read Only Memory</i>)
ALU	- Аритметичко-логичка јединица (енг. <i>Arithmetic-Logic Unit</i>)
IF	- Фаза захватања инструкције (енг. <i>Instruction Fetch Stage</i>)
ID	- Фаза декодовања инструкције (енг. <i>Instruction Decode Stage</i>)
EX	- Фаза извршавања инструкције (енг. <i>Execute Stage</i>)
FFT	- Брза Фуријеова трансформација (енг. <i>Fast Fourier Transform</i>)
MDCT	- Модификована дискретна косинусна трансформација (енг. <i>Modified Discrete Cosine Transformation</i>)
SDG	- Субјективна оцена одступања (енг. <i>Subjective Difference Grade</i>)
PEAQ	- Перцептуално оцењивање аудио квалитета (енг. <i>Perceptual Evaluation of Audio Quality</i>)
MOV	- Излазне променљиве модела (енг. <i>Model Output Variables</i>)
ODG	- Објективна оцена одступања (енг. <i>Objective Difference Grade</i>)
DI	- Индекс изобличења (енг. <i>Distortion Index</i>)
DMA	- Директан приступ меморији (енг. <i>Direct Memory Access</i>)
PCM	- Импулсна кодна модулација (енг. <i>Pulse Code Modulation</i>)
MPM	- Средишњи ниво за обраду (енг. <i>Mid-Processor Module</i>)
PPM	- Последњи ниво за обраду (енг. <i>Post-Processor Module</i>)
MIPS	- Јединица за одређивање перформанси процесора – број милиона извршених инструкција у секунди (енг. <i>Milions of Instructions Per Second</i>)

1. Увод

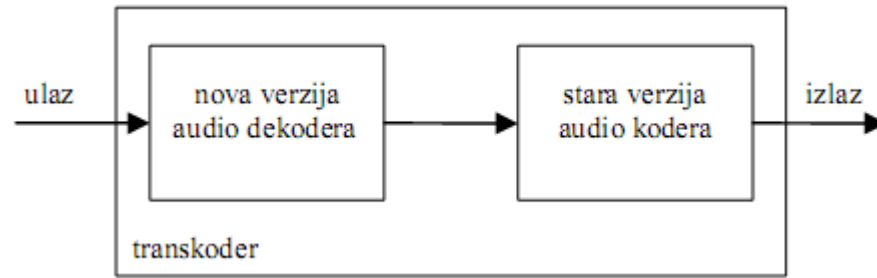
У раду је реализовано једно решење вишеканалног аудио кодера, као саставног дела транскодера, на једној класи процесора за дигиталну обраду сигнала (енг. *Digital Signal Processor, DSP*), уз коришћење минималних ресурса. Описана је структура кодера, његова улога у транскодеру, као и реализација на ДСП процесору фирме *Zoran*. Транскодер се користи у корисничком пријемнику за дигиталну телевизију, који има задатак да аудио сигнале са различитих извора комбинује и прилагођава телевизору.

Дигитална телевизија (*DTV*) је нова телевизијска услуга која представља највећи помак у историји телевизије још од педесетих година прошлог века и увођења телевизије у боји. Кориснику су доступни програми са далеко оштријом сликом, бољом дефиницијом боја, звуком квалитета као на *CD* медијима (5.1), пренос *HDTV* (енг. *High Definition TV*, резолуције 1920x1080, аспекта 16x9) и друга унапређења које дигитална телевизија носи са собом. Осим пријема квалитетнијег сигнала, *DTV* омогућава и рационалније искоришћење спектра (до четвороструког смањења ширине опсега), отвара додатни простор за бежичне апликације и нуди читав спектар додатних услуга од стране оператера, налик онима које данас пружају сателитски и кабловски оператери (нпр. сервиси типа *PayTV*, *Pay-per-view TV*, интерактивна телевизија и други) [1].

Да би се ове услуге користиле, потребан је кориснички пријемник за дигиталну телевизију (енг. *Set-top box, STB*). *STB* је уређај који повезује телевизор и неки од спољних извора сигнала, претварајући сигнал у садржаје који се приказују на екрану, при чему извор сигнала може бити сателитска антена, кабловска ТВ мрежа, телефонска линија, *DSL* (енг. *Digital Subscriber Line*) и други, све до обичних *VHF* (енг. *Very High Frequency*) и *UHF* (енг. *Ultra High Frequency*) антена. Осим уобичајеног видео садржаја пружа и додатни огроман спектар могућности пријема аудио сигнала, интернет страница, интерактивне и комерцијалне телевизије, видео игара и другог [2].

Кориснички пријемник за дигиталну телевизију, приликом повезивања са телевизором, добија информације о уграђеним декодерима, и на основу њих одређује у које формате ће кодовати улазни сигнал који није познат телевизору. У неким случајевима је потребно и трансकोдовање сигнала у старију верзију.

Предмет анализе овог рада је кориснички пријемник (*STB*) са уграђеним транскодером, који аудио формат нове генерације прилагођава старијој верзији истог стандарда. Наиме, проширивањем аудио стандарда долази до некомпатибилности старијих верзија са новијим, па је потребно трансковати сигнале како би се остварила компатибилност. У оквиру овог рада је реализован вишеканални аудио кодер, као део транскодера, приказан на следећој слици:



Слика 1.1. Блок дијаграм транскодера

Процесори за дигиталну обраду сигнала, који се користе у наменским системима, морају да представљају јефтино решење по питању величине процесора, потрошње енергије и цене израде, тако да њихови ресурси (меморија, број регистара, број извршених инструкција у секунди) морају бити сведени на минимум.

Реализација кодера урађена је на основу референтног *C* кода у аритметици са покретним зарезом. Прво је код прилагођен аритметици са непокретним зарезом, а потом је написан и у асемблеру. Циљна платформа је аудио ДСП процесор *ZDSP5*, израелске компаније *Zoran*.

Документација је сачињена од седам поглавља.

У другом поглављу изложене су теоријске основе рада, кроз објашњење основних принципа компресије аудио сигнала. Дат је кратак преглед основних класа алгоритама за компресију података и начин њиховог рада. Дат је кратак опис дигиталних сигналних процесора и њихове основне карактеристике, као и опис корисничких пријемника за дигиталну телевизију.

У трећем поглављу је представљена анализа проблема реализације вишеканалног аудио кодера на ДСП платформи са ограниченим ресурсима. Дат је детаљан опис кодера и анализиран је поступак конвертовања аритметике покретног зареза у аритметику непокретног зареза. Анализирани су критеријуми за оцену квалитета модификованог алгорита и проблеми при писању асемблерског кода.

У четвртном поглављу је описана реализација кодера на *ZDSP5* платформи. Приказана је потрошња ресурса на наменској платформи кроз преглед утрошене меморије и потрошње процесорског времена.

У петом поглављу су дати резултати испитивања и верификације реализованог вишеканалног аудио кодера.

У шестом поглављу је, у оквиру закључка, дат преглед свега што је урађено у раду, анализирани су резултати испитивања и представљени даљи правци развоја.

Садржај седмог поглавља је листа литературе која је коришћена приликом израде овог рада.

2. Теоријске основе рада

Компресија звучног сигнала има улогу да обезбеди компресовану дигиталну репрезентацију звучног сигнала у циљу ефикасног складиштења и преноса. Основна идеја је да се оригинални сигнал представи што је могуће мањим бројем бита, уз очување транспарентности са оригиналом.

Увођење оптичких дискова (енг. *Compact Disc, CD*) у раним осамдесетим годинама прошлог века довело је у први план све предности дигиталне репрезентације звучних сигнала: високи квалитет репродукције, робусност, реализацију функција за обраду сигнала (филтрирање, уклањање еха, ...). Са друге стране, ове предности су са собом донеле захтев за великим пропусним опсегом мултимедијалних система. Наиме, на типичном аудио диску налази се сигнал одмерен фреквенцијом одабирања 44.1/48 КHz са резолуцијом одмерка од 16 бит-а. Резултат овога је проток података од 705.6/768 Kbit/sec за један канал, односно 1.41/1.54 Mbit/sec за два канала, респективно. Даљи развој мултимедијалних апликација и бежичних система преноса података захтевао је што квалитетнију репрезентацију звучног сигнала (већи број канала, већа фреквенција одабирања, већа резолуција одмерка), што доводи до знатног ширења пропусног опсега система за пренос и захтева за повећањем капацитета медија на који се звучни сигнал складишти. Како развој медија за складиштење и повећање пропусног опсега система не могу да прате захтеве за што бољом репродукцијом, јавила се потреба за компресијом сигнала.

Основна идеја компресије је да се из сигнала уклоне сувишне информације и произведе много компактнија репрезентација сигнала, у циљу ефикасног складиштења и преноса. Алгоритми којима се компресија постиже могу се поделити у две групе: алгоритми код којих се сигнал може комплетно реконструисати и алгоритми код којих долази до трајног губитка информација. Сама чињеница да алгоритми код којих се сигнал може комплетно реконструисати након реконструкције дају тачну копију оригиналног сигнала, резултује slabим степеном компресије, од 2 до 4. Са друге стране алгоритми код којих долази до трајног губитка информација се одликују високим степеном компресије, од 8 до 40. Употреба алгоритама којима се део информација трајно губи је тренутно веома заступљена. Она покрива широк спектар апликација као што су на пример пренос звука путем интернета у виду радио станица, звучни запис *DVD* филмова, репродукција звука на персоналним рачунарима, мобилним телефонима и разним преносивим уређајима. Алгоритми који омогућавају потпуну реконструкцију звука представљају новији тренд у компресији сигнала и углавном се користе за студијско архивирање снимака и представљају део спецификације *DVD* формата следеће генерације.

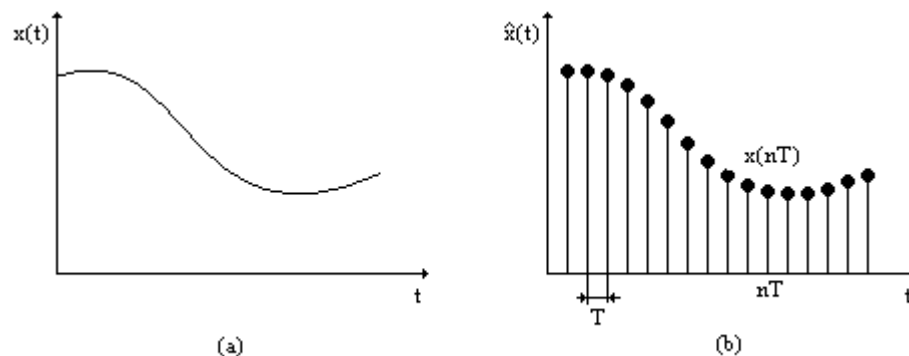
2.1 Дискретизација континуалних сигнала

Како је звучни сигнал по својој природи континуалан, да би се могао обрађивати коришћењем процесора за дигиталну обраду сигнала, он се мора прво конвертовати у дигитални облик, то јест у низ бројева. При томе се обавља процес двоструке дискретизације: *дискретизација по времену (одабирање)* и *дискретизација по амплитуди (квантовање)*.

Процес одабирања [3] [4] представља конверзију континуалног сигнала у сигнал дискретан у времену, односно:

$$x[n] = x_a[nT],$$

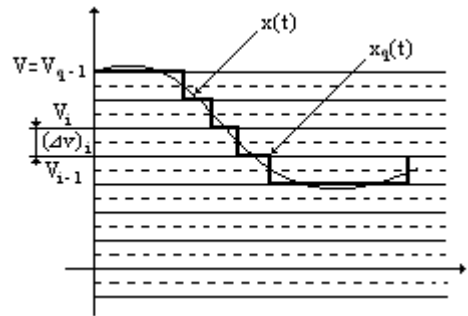
где је $x_a(nT)$ вредност континуалног сигнала у тренутку одабирања, T је периода одабирања, док је $F_s=1/T$ учесталост одабирања. Другим речима, у поступку дискретизовања континуалног сигнала по времену меримо његове тренутне вредности у тачно дефинисаним временским интервалима, као што је приказано на слици 2.1.



Слика 2.2 Дискретизација сигнала по времену (одмеравање): (а) континуалан сигнал, (б) сигнал дискретизован по времену

Приликом одабирања континуалног сигнала, основни циљ који треба да буде остварен је да се све информације из оригиналног сигнала задрже у одбирцима. Ако је могуће тачно реконструисати оригинални сигнал из одбирака, сматра се да није наступио губитак информација. Да би реконструкција била успешна мора бити задовољена *Никвистова (Nyquist) теорема* или *теорема одабирања* [4] која гласи: *Временски континуални сигнал може се потпуно реконструисати из својих одбирака ако је, и само ако је, учесталост одабирања бар два пута већа од највише учесталости у спектру сигнала.*

Поступак квантовања [3] [4] представља дискретизацију тренутних вредности континуалних сигнала у амплитудском домену. Квантовањем се тренутна вредност сигнала, која у општем случају припада неизбројивом скупу тачака из континуалног опсега, трансформише у најближу дозвољену вредност из коначног, дискретног скупа амплитуда.



Слика 2.3 Илустрација квантовања континуалног сигнала

Сам процес квантовања одвија се онако како је то приказано на слици 2.2. Вероватни опсег тренутних вредности улазног сигнала подељен је на q зона. Разлика између два нивоа квантовања назива се интервал квантовања:

$$(\Delta v)_i = V_i - V_{i-1} \quad (i = 1, 2, 3, \dots, q-1)$$

Улазни, континуални сигнал (танко извучена крива), $x(t)$, током времена пролази кроз зоне различитих амплитудских кванта. Границе зона назначене су испрекиданим хоризонталним линијама. Док се сигнал налази унутар зоне једног кванта, на излазу квантизера одржава се константни ниво који обично одговара средњој аритметичкој вредности граница зона. У тренутку када улазни сигнал прелази у зону суседног кванта, на излазу квантизера ниво се скоковито мења и узима вредност следећег дозвољеног нивоа. Тако, током времена на излазу квантизера настаје квантовани сигнал, $x_q(t)$, који је представљен степеничастом кривом. Број квантизационих нивоа се најчешће узима као степен броја 2, да би се у потпуности искористио број бита којима се они представљају.

Дигитални подаци добијени одабирањем и квантовањем, који се састоје од низа бинарних вредности које представљају број квантизационог нивоа у једнаким временским интервалима, другачије се називају *PCM* сигнали.

2.2 Основни принципи компресије аудио сигнала

Алгоритми за компресију аудио сигнала могу се поделити у две основне фамилије: компресија без губитака и компресија са губицима, односно компресија заснована на тромости људског уха. Другим речима, реконструкцијом сигнала, компресованог алгоритмом којим се губи део информација, добија се модификована верзија оригиналног сигнала, док алгоритми који компресују без губитака поштују интегритет оригиналног сигнала, и након реконструкције дају сигнал који је једнак оригиналном сигналу.

2.2.1 Компресија аудио сигнала заснована на тромости људског уха

Основна идеја ове компресије заснована је на тромости људског уха. Чуло слуха је веома сложен апарат, чији се рад заснива на могућности да осети и разликује огроман број карактеристика звука које потичу од извора, система преноса и акустичног амбијента на месту генерисања и месту преноса.

Да би чуло слуха регистровало акустичне вибрације (промене звучног притиска) као звук потребно је да оне буду довољног интензитета и да се мењају у чујном опсегу фреквенција. Основне карактеристике звука које ухо разликује су:

1. јачина звука,

2. висина тона и
3. боја звука.

Осећај јачине звука директно зависи од јачине физичке побуде која стиже до уха. Људско ухо реагује на промене јачине звучног притиска. Доња граница за јачину звучног сигнала назива се праг чујности и износи $p_0 = 2 \cdot 10^{-5} \text{ Pa}$ на (1000 Hz) [5]. Горња граница је одређена појавом бола до које долази услед великих помераја у органу чула слуха. Граница бола (на 1000 Hz) одговара преко милион пута већем притиску од прага чујности:

$$20 \log \frac{P_{\max}}{p_0} = 20 \log 10^6 = 120 \text{ dB}$$

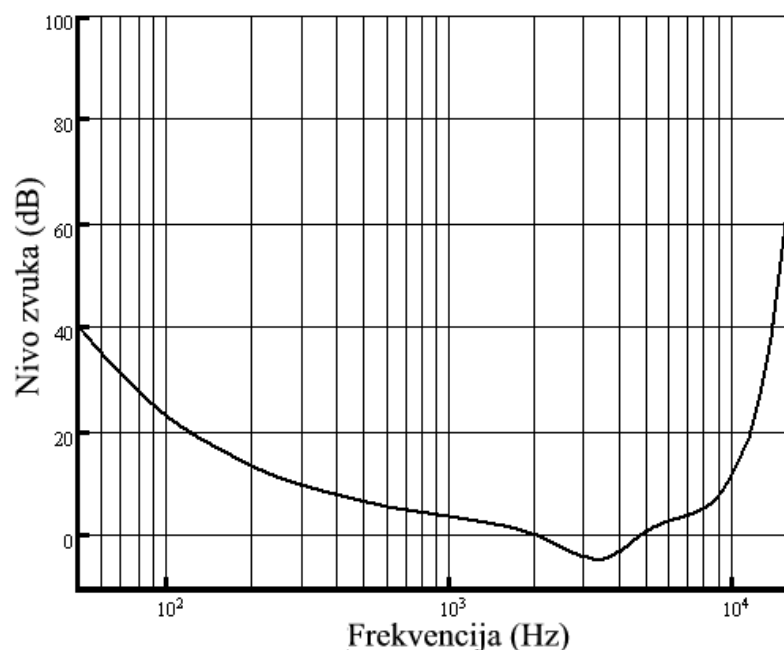
Због великог динамичког опсега чула слуха није пожељно узети притисак као мерило јачине звука – бројне вредности за поједине чујне звукове разликовале би се за 10^6 пута. Зато је уведен појам нивоа звука [5], чија је јединица децибел (dB). Њиме се даје логаритамски однос две величине, а као референтна величина узима се праг чујности p_0 и у односу на њега се посматра ниво звука:

$$L[\text{dB}] = 20 \log \frac{P}{p_0}.$$

Апсолутна граница чујности [5] представља минималну количину енергије, коју треба да садржи звук, да би га слушалац могао регистровати у тихом окружењу. Фреквенцијску зависност апсолутне границе чујности утврдио је *Fletcher* 1940. године и њена функција се може добро апроксимирати нелинеарном функцијом:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 [\text{dB}]$$

Када се примени на компресију звучног сигнала, $T_q(f)$ се може посматрати као максимални дозвољени ниво енергије за кодовање сигнала у фреквенцијском домену, слика 2.3. Другим речима сваки сигнал чији је ниво звука испод криве, $T_q(f)$, се може одстранити, јер не садржи довољну количину енергије да би га чуло слуха препознало.



Слика 2.4 Апсолутна граница чујности у тихом окружењу

Употреба апсолутне границе чујности има ограничену употребу приликом компресије, из разлога што је за њену апроксимацију као извор узет чист тон. Наиме, детекција границе чујности спектрално комплекснијег извора представља модификовану верзију апсолутне границе чујности, чији је облик одређен извором у било ком тренутку времена. Како извор звука, генерално гледано, зависи од времена, следи да је и детекција границе чујности временски зависна функција. Да би се оваква граница чујности правилно проценила, прво треба анализирати како људско ухо врши спектралну анализу .

Трансформација фреквенције у положај врши се у унутрашњем уху [6]. Звучни талас, кроз спољашње и средње ухо, долази у унутрашње ухо, где се простире дуж мембране на којој се налазе нервни завршетци. У зависности од фреквенције таласа, максимални одзив догодиће се на месту специфичном за дату фреквенцију, то јест различити нервни завршетци ће бити побуђени различитим фреквенцијама, у зависности од њиховог положаја. Уколико је извор звука синусоидалног облика, талас ће да пролази кроз унутрашње ухо све док не нађе регион са резонантном фреквенцијом најближом резонантној фреквенцији таласа. Ова појава се у дигиталној обради сигнала може представити као група појасних филтера, чији се опсежи преклапају. Амплитуда одзива групе појасних филтера је несиметрична и нелинеарна, а ширина пропусног опсега расте са порастом фреквенције. Овако дефинисана група појасних филтера, која је у функцији фреквенције сигнала, назива се још и критични опсег (енг. *Critical bandwidth*). На пример, за шум са уским фреквентним опсегом и константним нивоом звука, јачина опаженог звука остаје константна чак и када се фреквентни опсег шума помери на саму границу критичног опсега. Уколико се опсег шума повећава даље јачина опаженог звука расте. Ово може објаснити на следећи начин: јачина опаженог звука остаје константна све док је фреквентни опсег шума у границама једног критичног опсега, а повећава се када се фреквентни опсег шума повећа толико да енергија шума пређе у суседни критични опсег.

Ширина критичних опсега до фреквенције 500Hz износи 100Hz, док за фреквенције преко 500Hz расте. За просечног слушаоца ширина критичних опсега, слика 2.4б, се може апроксимирати формулом [7]:

$$BW_c(f) = 25 + 75 \left[1 + 1.4 \left(\frac{f}{1000} \right)^2 \right]^{0.69}$$

Идеализована група филтера, која одговара тачкама назначеним на кривама на слици 2.4, дата је у табели 2.1. Удаљеност критичног бенда се назива *Bark* [7], и функција:

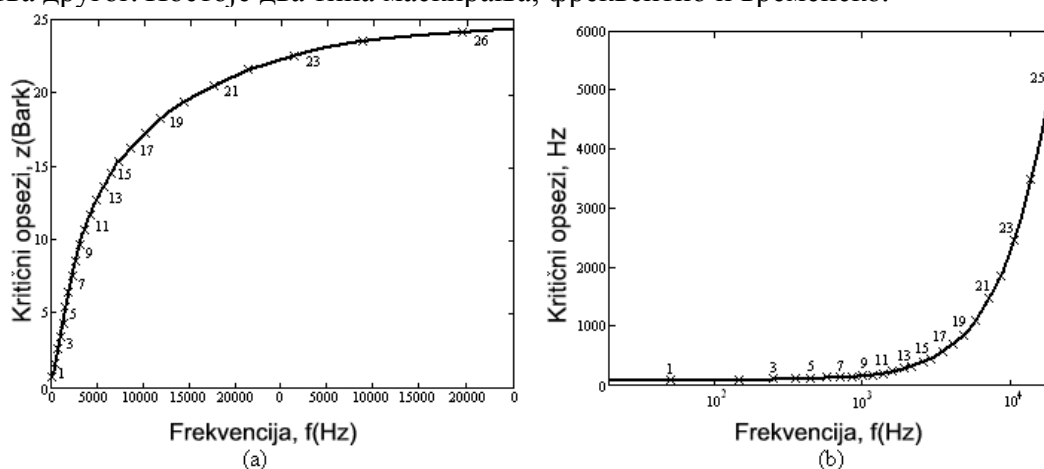
$$z(f) = 13 \arctan(0.00076f) + 3.5 \arctan \left[\left(\frac{f}{7500} \right)^2 \right]$$

се најчешће користи за трансформацију из фреквентне у *Bark* скалу, слика 2.4а.

Табела 2.1 Идеализована група филтера за тачке назначене на слици 2.4

Број опсега	Централна фреквенција (Hz)	Ширина опсега (Hz)	Број опсега	Централна фреквенција (Hz)	Ширина опсега (Hz)	Број опсега	Централна фреквенција (Hz)	Ширина опсега (Hz)
1	50	-100	10	1175	1080-1270	19	4800	4400-5300
2	150	100-200	11	1370	1270-1480	20	5800	5300-6400
3	250	200-300	12	1600	1480-1720	21	7000	6400-7700
4	350	300-400	13	1850	1720-2000	22	8500	7700-9500
5	450	400-510	14	2150	2000-2320	23	10500	9500-12000
6	570	510-630	15	2500	2320-2700	24	13500	12000-15500
7	700	630-770	16	2900	2700-3150	25	19500	15500-
8	840	770-920	17	3400	3150-3700			
9	1000	920-1080	18	4000	3700-4400			

Фреквентна резолуција групе филтера, односно критични опсежи, у великој мери одређују који су делови сигнала занемарљиви, односно који делови сигнала се из њега могу одстранити, а да се то не осети. Спектрална анализа коју врши група филтера доводи до појаве маскирања. Маскирање представља процес када један звук постаје нечујан због присуства другог. Постоје два типа маскирања, фреквентно и временско.

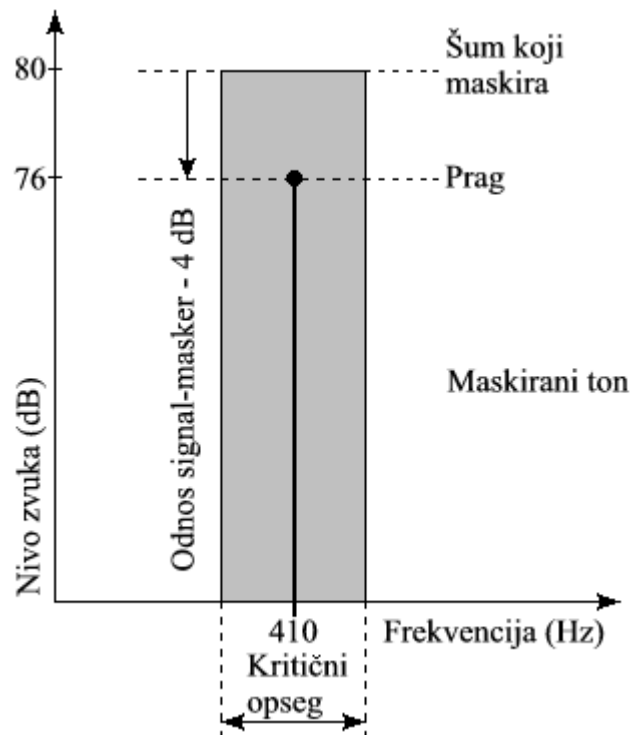
Слика 2.5 Критични опсежи: (а) приказани у *Bark* скали, (б) приказани у фреквентној скали

Фреквентно маскирање настаје у присуству два или више извора у систему. Оно се јавља као последица чињенице да након побуђивања чула слуха тоном одређене фреквенције, побуђивање тоном исте фреквенције али мање амплитуде, неће изазвати никакве реакције у органу чула слуха. Иако произвољан спектар звучног сигнала може да садржи много комплексних сценарија фреквентног маскирања, за сврху компресије довољно је разликовати три:

1. шум маскира тон,
2. тон маскира шум и
3. шум маскира шум.

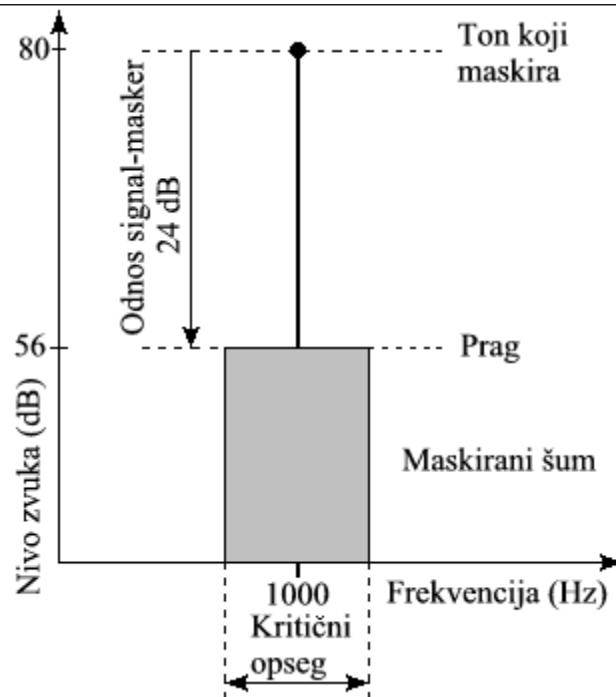
Први сценарио представља случај када шум, уског фреквентног опсега, маскира тон у границама истог критичног опсега [8]. Бројне студије које описују овај случај, изведене

за случајан шум и чист тон, довеле су до сазнања да на прагу детекције за маскирани тон, минимални однос сигнала и шума којим се сигнал маскира (*SMR*), настаје када је фреквенција маскираног тона близу централној фреквенцији шума. Ова се вредност, за већину случајева, налази између -5 и $+5$ dB. На слици 2.5 дат је пример где шум центриран на 410KHz са нивоом звука од 80dB маскира тон такође центриран на 410KHz, са односом сигнала и шума (*SMR*) на граници чујности од 4dB.



Слика 2.6 Пример који илуструје случај када шум маскира тон

Други сценарио, односно случај када тон маскира шум, настаје када чист тон, центриран у једном критичном опсегу, маскира шум било каквог облика [9]. Маскирање ће се догодити само у случају када је спектар шума испод очекиваног прага детекције. На граници детекције, за шум маскиран чистим тоном, најмања разлика између интензитета тона који маскира и шума који се маскира, настаје када је фреквенција тона близу централној фреквенцији шума. Тада се однос сигнала који маскира и сигнала који се маскира (*SMR*) налази између 21-28dB. На слици 2.6 дат је пример где се шум центриран на 1KHz маскира тоном чија је фреквенција 1KHz, а ниво звука 80dB. Резултујући *SMR*, на граници чујности је 24dB.



Слика 2.7 Пример који илуструје случај када тон маскира шум

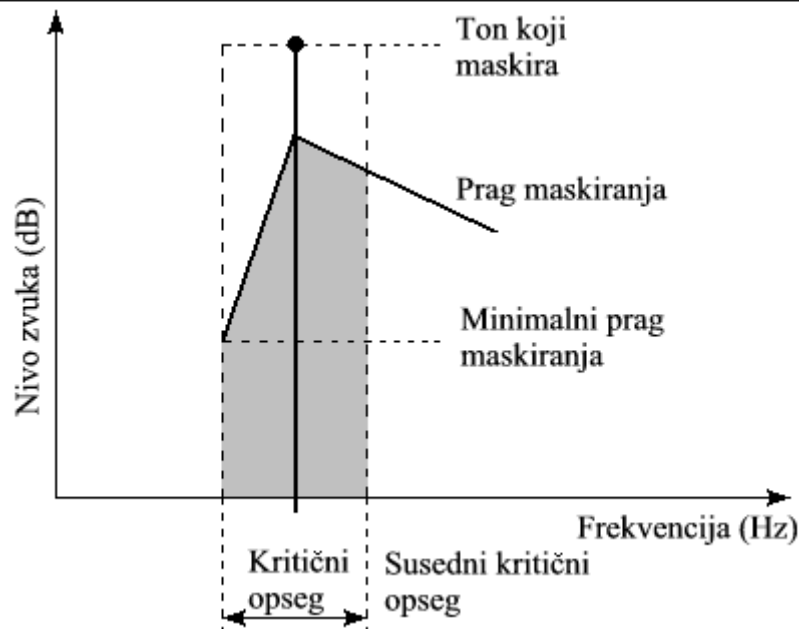
Трећи сценарио, тј. случај када се шум уског опсега маскира такође шумом уског опсега [5], је теже описати него претходна два сценарија. Разлог за то је велики утицај повезаности шума који маскира и шума који се маскира. Заправо, различите везе између њих, резултују различитим вредностима односа сигнала који маскира и сигнала који се маскира (*SMR*) на граници чујности.

Сигнал који маскира, центриран у једном критичном опсегу, има предвидив утицај на почетак детекције у осталим критичним опсезима. Ова појава, позната и као ширење маскирања, се у апликацијама најчешће моделује као функција са нагибом од +25dB и –10dB по једном *Barku*. Аналитички, она се може записати на следећи начин [10]:

$$SF_{dB}(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1 + (x + 0.474)^2} \text{ dB},$$

где је x јединица изражена као *Bark*. Ако у једном критичном опсегу постоји тон, његово маскирање ће се продужити и на суседни критичан опсег, слика 2.6. Уколико се у суседном критичном опсегу налази тон, чији је ниво звука испод прага маскирања, он ће бити маскиран и као такав се може избацити из сигнала, а да то људско ухо не примети.

Временско маскирање произилази из чињенице, да након што људско ухо чује гласан звук, треба да прође одређено време да би могло да чује тих звук. Наиме, нервни завршеци у унутрашњем уху вибрирају интензитетом који зависи од јачине побуде. Уколико је побуда јака, нервни завршеци ће бити у стању велике побуде и биће им потребно време да се смире.



Слика 2.8 Ширење маскирања на суседне критичне опсеге

2.2.2 Компресија аудио сигнала без губитака

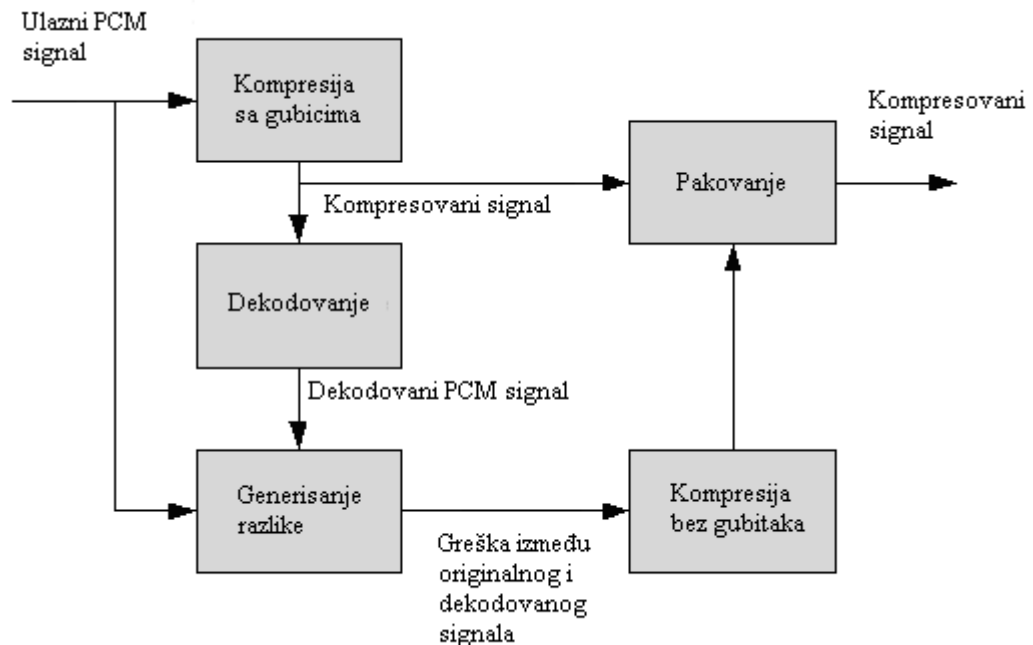
За разлику од компресија засноване на тромости људског уха, где се компресовањем уклањају компоненте чији се недостатак неће приметити, овај тип компресије не уклања ни један део оригиналног сигнала, односно реконструкцијом се добија сигнал који је идентичан са оригиналним сигналом. Дакле, овакав начин компресије се може посматрати као преправљени начин записивања стандардног записа импулсне кодне модулације (енг. *Pulse Code Modulation, PCM*), са циљем да се смањи захтевани проток података за пренос сигнала. Њоме се минимизирају појаве сувишних информација у *PCM* запису и тиме се ствара нови формат дигиталног записа звучних сигнала.

Основна мана ових алгоритама представља низак степен компресије, чија се уобичајена вредност налази између 2 и 3, што је веома мало у поређењу са степеном компресија који одликују алгоритме засноване на тромости људског уха, који се крећу између 2.7 и 24. Очито је да проток података који је потребан за пренос записа, добијеног компресијом без губитака, није константан, већ зависи од количине сувишних информација које се налазе у *PCM* запису. Другим речима различити сигнали ће бити компресовани различитим степеном компресије, што за собом повлачи чињеницу да ће ова компресија произвести запис за чији је пренос потребан проток података који није константан.

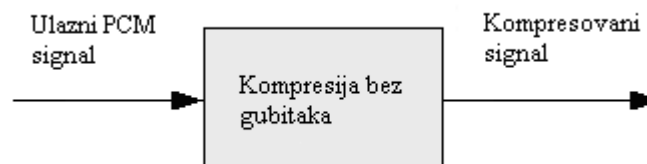
Ипак, у последње време, развијени су алгоритми за компресију без губитака који за пренос захтевају константан проток података. Овим алгоритмима, на пример, 16-битни одмерци са фреквенцијом одабирања 96КHz, могу бити компресовани тако да се за њихову представу користи 8 бита, док се, на пример, 16-битни одмерци са фреквенцијом одабирања 48КHz могу компресовати тако да се за њихову представу може користити 12 бита. Као резултат добијеног константног протока података, смањен је степен компресије. Из наведених примера може се закључити да је степен компресије алгоритмима за компресију без губитака бољи када је фреквенција одабирања већа, што значи да употреба ових алгоритама добија оправдан разлог за употребу са порастом резолуције представљања сигнала у дигиталном облику.

Тренутно развијени алгоритми за компресију без губитака могу се поделити у две групе [11]:

1. алгоритми који користе комбинацију компресије са губицима и компресије без губитака, слика 2.8 и
2. алгоритми који за компресију сигнала користе само алгоритам за компресију без губитака, слика 2.9.



Слика 2.9 Компресија сигнала без губитака у комбинацији са компресијом са губицима



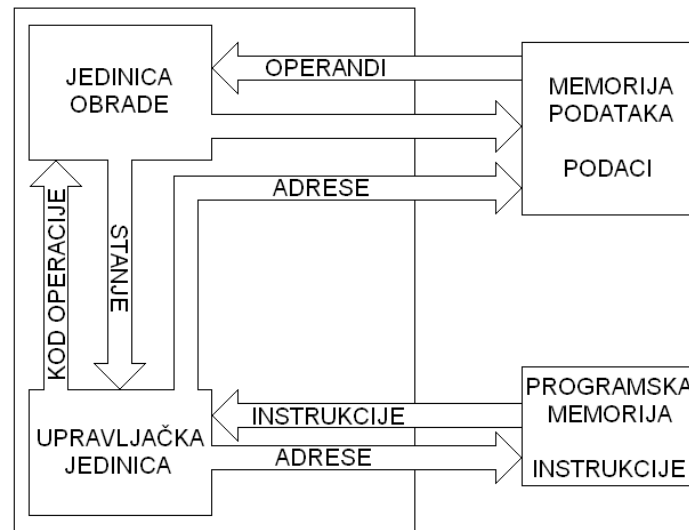
Слика 2.10 Компресија сигнала без губитака

2.3 Дигитални сигнал процесори

Дигитални сигнални процесори – ДСП су специјална врста процесора који имају многе сличности са процесорима опште намене. Као и процесори опште намене ДСП садрже централну процесорску јединицу (енг. *Central Processor Unit, CPU*), меморију, инструкцијски сет, сабирнице, итд. Основна разлика је у томе што су све наведене компоненте делимично модификоване, у односу на процесор опште намене, како би одређене операције могле да се обављају знатно ефикасније. Генерално, ДСП процесори су оптимизовани за брзе нумеричке прорачуне и брзу обраду дигиталних сигнала у реалном времену.

Данас су алгоритми за дигиталну обраду података свеprisутни, од мобилних телефона, преко дигиталних камера, до ДВД уређаја. Код ДСП процесора је веома важно окружење у ком раде, а не само алгоритми који су на њима имплементирани. Већина ДСП

апликација захтева интеракцију са спољним светом, у реалном времену, због чега ДСП процесори морају бити дизајнирани тако да што брже реагују на спољне утицаје, да брзо прихватају улазне податке, брзо врше обраду и дају излазне податке. Брзи серијски улазно/излазни портови и друге периферије специфичне за ДСП процесоре омогућавају прихватање и пренос велике количине података у реалном времену. За обраду сигнала, у циљу избегавања већег броја читања и уписа у спољну меморију као и секвенцијалног читања инструкција и података уведена је архитектура која раздваја меморију на меморију за податке и меморију за инструкције. Та архитектура названа је *Харвард* архитектуром и приказана је на слици 2.10 [12]:

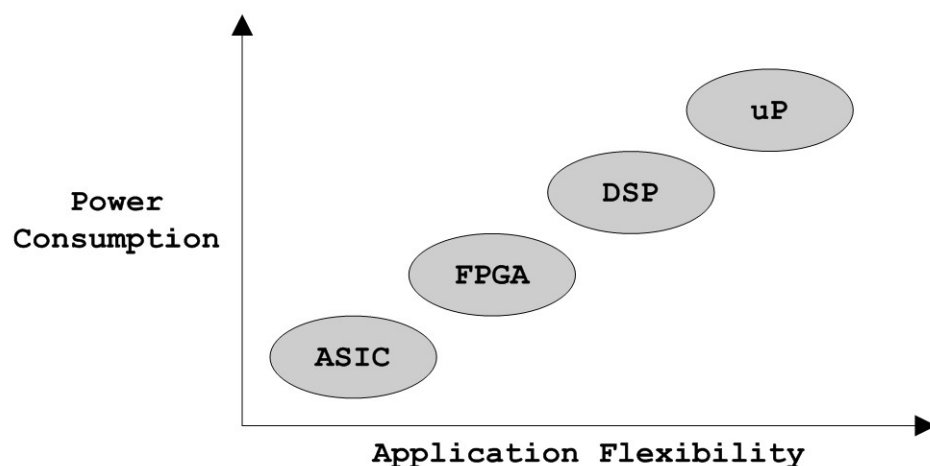


Слика 2.11 Харвард архитектура

2.3.1 Предности дигиталних сигнал процесора у обради аудио сигнала у односу на друга решења

Алгоритми за дигиталну обраду сигнала, сем на ДСП процесорима, могу да се имплементирају и на следећим платформама (слика 2.11):

- ASIC (енг. *Application-Specific Integrated Circuit*)
- FPGA (енг. *Field-Programmable Gate Arrays*)
- Микропроцесори (енг. *Microprocessors*)



Слика 2.12 Упоредни приказ платформи за имплементацију ДСП алгоритама [13]

ASIC је хардверска имплементација без коришћења програмске подршке. Овакве врсте имплементација раде веома брзо у поређењу са програмским решењима. Проблем је што је оваква имплементација веома компликована јер је свака реализација неког алгорита у суштини процесор за себе. Додатна потешкоћа је што се оваква решења веома тешко модификују, а чест случај у пракси су потребе за честим модификацијама одређених алгорита и потреба за додавањем одређене подршке на већ постојећи алгорита, за шта је у овој ситуацији потребно много труда и времена, што је веома скупо. Уколико је потребно да се неки алгорита извршава веома брзо и уколико је сигурно да се неки алгорита неће у будућности мењати, што је веома редак случај, тада се користи ово решење.

FPGA имплементација је такође хардверска имплементација, са већим степеном флексибилности од *ASIC* решења, јер се *FPGA* може репрограмирати за разлику од *ASIC*-а. *FPGA* се састоји од великог броја логичких кола, која се „програмирањем” повезују на одређени начин и тиме добијамо одређену функционалност. Репрограмабилност је већа него код *ASIC* решења, али мања него код ДСП решења. Додатно, велик је степен неискористивости логичких кола, па се и цена изградње оваквог решења додатно повећава услед недовољног искоришћења самог *FPGA* чипа.

Из горе наведеног се виде предности програмског у односу на хардверска решења. Коришћење микропроцесора опште намене за имплементацију алгорита за дигиталну обраду сигнала због своје неоптимизованости за постојећу проблематику, такође не представља добро решење. Из свега се лако могу уочити предности реализације ових алгорита на ДСП платформама.

2.3.2 Особине дигиталних сигнал процесора

Због комплексности ДСП апликација и потребе за извршавањем великог броја операција, основни циљ ДСП дизајна је да се архитектури омогући извршавање једне аритметичке операције у једном машинском циклусу. Да би се ово постигло, ДСП архитектуре се пажљиво дизајнирају тако да инструкцијски циклус буде једнак са машинским циклусом хардвера за већину инструкција. У идеалном случају, ово подразумева да ДСП процесор обавља следеће четири операције у сваком машинском циклусу:

- Захватање инструкције (енг. *Instruction Fetch*)
- Израчунавање адресе операнда инструкције
- Захватање операнда инструкције
- Извршавање операције

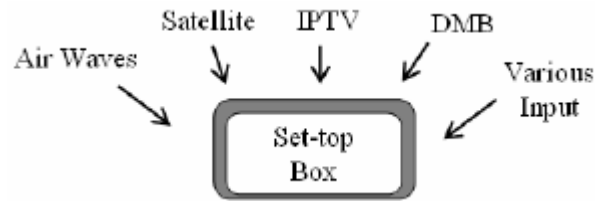
Ово може да се постигне увођењем паралелизма на нивоу извршавања инструкција (енг. *Pipelining*) у комбинацији са специјализованим хардверским јединицама и посебном организацијом меморије.

ДСП архитектуре се могу поделити према типу података који обрађују на:

- ДСП са аритметиком у непокретном зарезу (енг. *Fixed point*)
- ДСП са аритметиком у покретном зарезу (енг. *Floating point*)

Код архитектура са аритметиком у непокретном зарезу, мора се водити рачуна о скалирању резултата целобројних операција. Код процесора са покретним зарезом ово се аутоматски извршава. Јединице за рад са покретним зарезом су изузетно захтевне по питању величине силикона и брзине, па се у системима који захтевају малу потрошњу енергије и високе перформансе чешће користе ДСП процесори са аритметиком у непокретном зарезу [14].

2.4 Кориснички пријемник за дигиталну телевизију

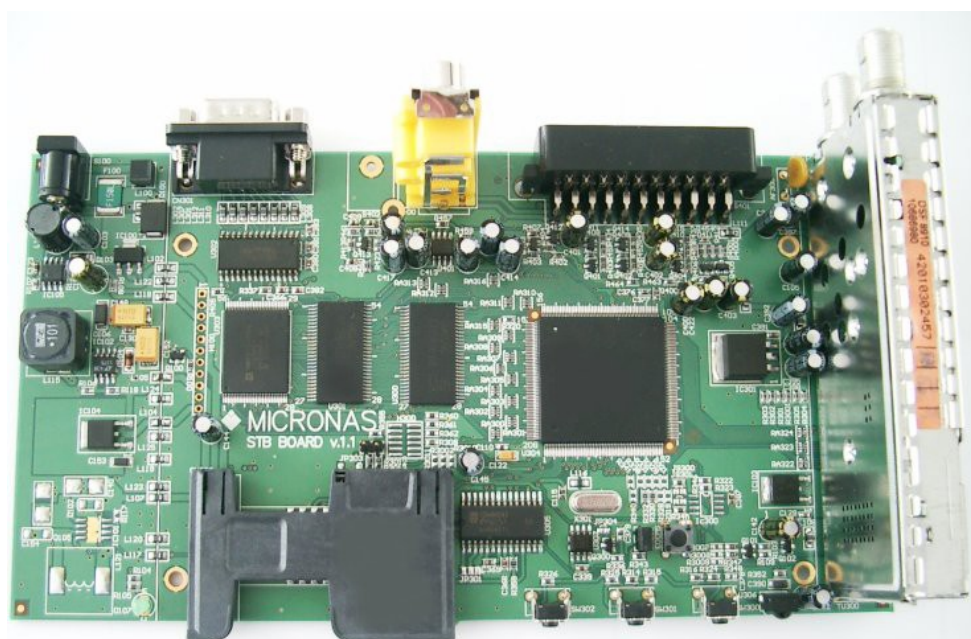


Слика 2.13 кориснички пријемник за дигиталну телевизију

Кориснички пријемник за дигиталну телевизију је уређај који повезује телевизор и неки од спољних извора сигнала, претварајући сигнал у садржаје који се приказују на екрану, при чему извор сигнала може бити сателитска антена, кабловска ТВ мрежа, телефонска линија, *DSL* (енг. *Digital Subscriber Line*) и други, све до обичних *VHF* (енг. *Very High Frequency*) и *UHF* (енг. *Ultra High Frequency*) антена. Осим уобичајеног видео садржаја пружа и додатни огроман спектар могућности пријема аудио сигнала, интернет страница, интерактивне и комерцијалне телевизије, видео игара и другог [2].

Кориснички пријемник за дигиталну телевизију, приликом повезивања са телевизором, добија информације о уграђеним декодерима, и на основу њих одређује у које формате ће кодovati улазни сигнал који није познат телевизору. У неким случајевима је потребно и трансковање сигнала у старију верзију.

На слици 2. је приказан такозвани *low-cost* модел референтног дизајна корисничког пријемника за дигиталну телевизију *MDE 9515B*, произведеног од стране компаније *Micronas*. Реализован је у техници 2-слоја *PCB* на плочи величине 100mm x 160mm (европски формат), где су на једној страни смештене компоненте: *DRX 8872C* (*DVB-T* демодулатор), *MDE 9515B* (*CPU*, аналогни видео, *MPEG-2*, 32Mbit *SDRAM*) и *DAC 3550A* (аудио *D/A* конвертор). [27]



Слика 2.14 Референтни дизајн FTA / PayTV корисничког пријемника за дигиталну телевизију

2.5 Критеријуми за оцену квалитета модификованог алгоритма

И поред тога што квалитет аудио сигнала представља фундаментални концепт аудио кодовања, тешко га је на објективан начин описати. Једноставне објективне мере квалитета аудио сигнала као што су однос сигнала и шума - SNR и вршни однос сигнала и шума - $PSNR$, дефинисане су на следећи начин:

$$P = \frac{\sum_{n=0}^N (x(n) - r(n))^2}{N} \quad (2.1)$$

$$PSNR = 10 \log \left(\frac{2^{b-1}}{P} \right) dB \quad (2.2)$$

$$SNR = 10 \log \left(\frac{\sum_{n=1}^N r(n)^2}{P} \right) dB \quad (2.3)$$

где је r референтни сигнал, x сигнал који се испитује, N величина оквира у одбирцима, а b број бита по одбирку.

Ове мере квалитета у потпуности занемарују психоакустичке ефекте аудио кодовања. Неадекватност коришћења ових мера код перцептуалних кодера објаснили су истраживачи *Johnston* и *Branderburg* крајем осамдесетих година прошлог века. Они су извели следећи експеримент [19]:

У оригинални сигнал унели су две врсте шума:

а) шум који је прилагођен, односно обликован, према психоакустичном моделу маскирања и налази се испод прага маскирања,

б) бели шум који је равномерно распоређен у фреквентом домену, значи углавном изнад прага маскирања.

Оба сигнала су имала исти SNR од 13 dB, али другачији перцептуални аудио квалитет: сигнал са перцептуално прилагонећим шумом имао је једва чујна изобличења за разлику од сигнала са белим шумом код кога су чујна изобличења била велика. Овај ефекат је назван ‘чудо од 13 децибела’ (енг. ‘13 dB miracle’) [20]. Међутим, SNR и $PSNR$ дају добру квантитативну меру па се често користе у оцени квалитета аудио сигнала нарочито у фази развоја аудио технологије.

Табела 2.2 Субјективна скала заснована на ITU-R BS.1284 стандарду у опсегу од 1.0 до 5.0

Субјективни квалитет сигнала	Оцена	SDG	Изобличење
Веома добар	5.0	0	Нечујно
Добар	4.0	-1.0	Чујно али не смета
Средњи	3.0	-2.0	Мало смета
Слаб	2.0	-3.0	Смета
Лош	1.0	-4.0	Веома смета

Пошто објективне мере не дају задовољавајуће резултате у потпуности, уобичајено се приступа субјективној процени квалитета. Ова врста испитивања назива се и слушно

испитивање (енг. *Listening Tests*) и подразумева велики број обучених слушалаца у контролисаном окружењу. Скала оцењивања коришћена у *BS.1116* слушном испитивању, дефинисана у [21] приказана је у табели 3.1. У оквиру оваквог испитивања, чујна разлика између оригиналног сигнала и сигнала који се испитује посматра се као оштећење и оцењује се у складу са скалом приказаном у табели. Често се, ради поједностављене анализе резултата, уводи субјективна оцена одступања (енг. *Subjective Difference Grade - SDG*) дефинисана на следећи начин:

$$SDG = Ocena\ Ispitanog\ Signala - Ocena\ Referentnog\ Signala \quad (2.4)$$

Негативна вредност *SGD* значи да слушалац успешно разликује оригинални сигнал од испитног сигнала. *SDG* је нула када је квалитет транспарентан и када су изобличења нечујна.

3. Анализа проблема

Полазна основа за реализацију кодера на наменској архитектури био је *C* код у аритметици са покретним зарезом. С обзиром на то да циљна платформа подржава само аритметику непокретног зареза, било је потребно модификовати полазни алгоритам тако да подржава аритметику непокретног зареза. Услед ограничених ресурса циљне платформе, неопходно је било извршити оптимизацију меморије и брзине извршавања.

Као и већина сличних алгоритама који се користе за компресију звука попут *DTS* и *Dolby Digital* вишеканалних кодека, *MPEG-2 Advanced Audio Coding (AAC)*, *Windows Media Audio (WMA)* и других, тако је и описани кодер веома сложен, имплементиран је у аритметици покретног зареза и захтева значајан број израчунавања. Процесорска снага (енг. *Processing power*) потребна да би се компресовао звук је веома важна, поготово код апликација за рад у реалном времену. У многим апликацијама, извршавање алгоритма на ДСП процесорима, може представљати ефикасно решење које омогућава брзо извршавање са смањеном ценом. Међутим, како већина ДСП процесора подржава аритметику у непокретном зарезу потребно је конвертовати оригинални алгоритам у аритметику с непокретним зарезом уз уношење прихватљиве грешке која је последица ограничене дужине операнада. Поред модификације самог алгорита, потребно је и прилагодити меморијске захтеве ресурсима којима располаже циљна платформа.

Реализација вишеканалног кодера на циљној платформи подразумевала је следеће кораке:

- Анализу полазног *C* кода у аритметици са покретним зарезом
- Модификацију оригиналног алгоритма тако да се користи аритметика непокретног зареза
- Реализацију асемблерског кода за циљни ДСП процесор

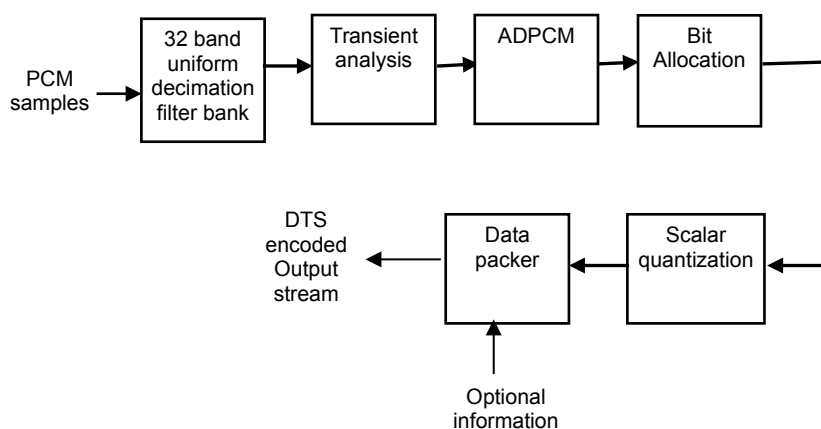
3.1 Опис кодера

Кодер је реализован у складу са следећим захтевима:

- излазни битски проток има фиксну битску брзину од 1.5 Mbps
- 512 одбирака (енг. *sample*) по оквиру (енг. *frame*)
- максималан број канала 6 (5.1 – 5 примарних аудио канала (фреквенције до 24KHz) и један канал за ниске фреквенције)
- фреквенција одабирања, $F_s = 48\text{KHz}$

- 7 – битни линеарни фактори скалирања

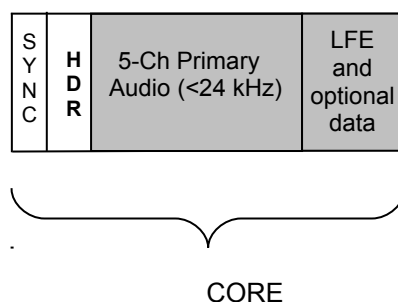
Анализом референтног *C* кода издвојени су основни функционални блокови кодера који су приказани на следећој слици:



Слика 3.15 Блок дијаграм кодера

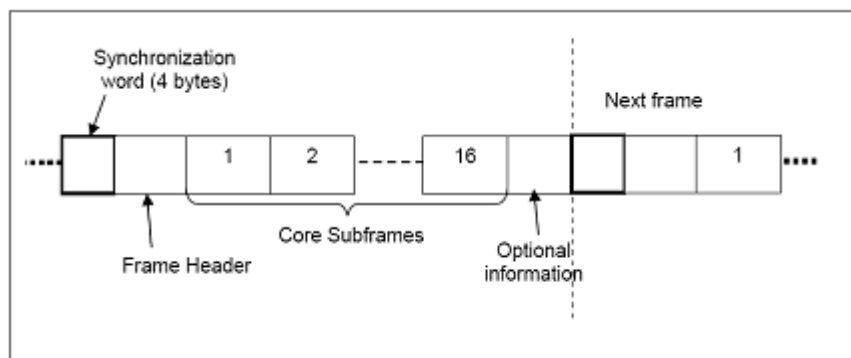
- Филтер банка (са несавршеном реконструкцијом) – улазни оквир у филтер банку представљају 512 *PCM* одбирака сваког од примарних канала, који се дели вишефазном филтер банком на 32 подопсега. Након тога се одбирци сваког подопсега групишу у анализирајуће прозоре.
- Анализа тренутних амплитудних промена, односно брзопроменљивих сигнала (енг. *Transient analysis*) – овај блок врши откривање и локализацију брзопроменљивих сигнала у текућем прозору одговарајућег подопсега (енг. *Subband analysis window*).
- ADPCM* – у одсуству појаве тренутних амплитудних промена у текућем прозору (један подоквир), односно у стационарном стању, одбирци подопсега се користе за оптимизацију скупа од четири *ADPCM* коефицијента по подопсегу. *ADPCM* кодовање се динамички укључује када је могуће представити одбирке у текућем прозору са значајно мањим бројем бита.
- Додела бита (енг. *Bit allocation*) – функције за доделу бита одређују на који начин се пуни излазни битски ток (енг. *Bit stream*) и колико бита могу да користе функције за кодирање.
- Скаларна квантизација (енг. *Scalar quantization*) – улазни подаци, то јест, одбирци се квантизују коришћењем скаларне квантизације.
- Паковање података – кодирани сигнал се пакује у излазни оквир [15].

На основу захтева, улазни сигнал у транскодер се може представити следећом сликом:



Слика 3.16 Пример улазног сигнала у транскодер

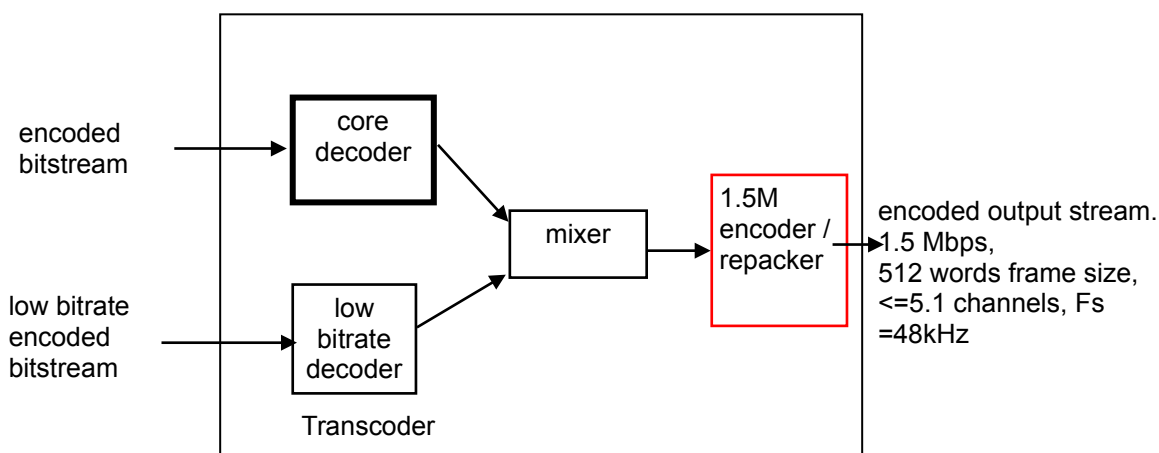
Структура улазног и излазног оквира је дата на следећој слици:



Слика 3.17 Структура улазно/излазног оквира

- Синхронизацијска реч – помаже декодеру да се синхронизује унутар битског тока
- Заглавље оквира – садржи информације о конфигурацији кодера, конструкцији оквира и слично
- Подоквири – садрже податке аудио сигнала. У овом раду, сваки оквир има 2 подоквира.

Описани кодер је саставни део транскодера, што се може видети на следећој слици.



Слика 3.18 блок шема транскодера

На слици 3.4 су приказани блокови транскодера, при чему 1.5M кодер представља последњу фазу обраде сигнала. Кодован ток података се доводи на улаз транскодера (нпр. са ДВД-а) и као такав се декодује у *core decoder* блоку. Кодован ток података нижих битских брзина по секунди (енг. *Low bit rate, LBR*) се такође доводи на улаз транскодера са другог извора (нпр. преко интернета) и као такав се декодује у *LBR* декодер блоку. Ова два декодована тока података се затим комбинују у сагласности са подешавањима транскодера и достављеним мета-подацима, и добијени комбиновани сигнал се доводи на улаз 1.5M кодера. Добијени излаз је записан у *SPDIF* формату и као такав се доводи на *SPDIF* улаз уређаја за декодовање.

3.1.1 Филтер банка

Улазни *PCM* одбирци примарних канала се деле, вишефазном филтер банком, на 32 подопсега једнаке ширине. Ширина сваког подопсега се израчунава по формули:

$$bandwidth = Fs / (32*2) [Hz],$$

где је Fs фреквенција одабирања и она у овом раду износи 48КHz, па се за ширину подопсега добија цифра од 750Hz.

Одбирци сваког подопсега се групишу у подопсежне прозоре за анализу, од којих сваки садржи два подпрозора (који се називају и подподоквири (енг. *subsubframes*)) са по осам подопсежних одбирака у сваком од њих.

Постоје два типа филтер банки, перфектна (енг. *perfect, PR*), која има савршену реконструкцију, и неперфектна (енг. *Non-perfect, NPR*) која нема савршену реконструкцију. С обзиром на то да је ово кодер са губицима и да је перцептуални квалитет важнији од савршене реконструкције сигнала, користи се филтер банка са несавршеном реконструкцијом.

3.1.2 Анализа тренутних амплитудних промена

Тренутна амплитудна промена (енг. *Transient*) се дефинише као оштар и брз прелаз између ниске и високе амплитудне фазе. У случају коришћења само једног фактора скалирања у текућем прозору, он може бити превелик у односу на одбирке нижег нивоа, који претходе тренутној амплитудној промени и на тај начин проузроковати пред-одјек.

У циљу ублажавања овог проблема, алгоритам је унапређен. Када се прелаз лоцира, онда се рачунају два фактора скалирања, један за одбирке који претходе тренутној амплитудној промени и други за одбирке који следе. Прецизност локализације тренутне амплитудне промене је на нивоу подподоквира (енг. *subframe*), који се састоји од осам одбирака, тако да је за два подподоквира могућа једна амплитудна промена (на пример, у другом подподоквиру). Након анализе тренутне промене амплитуде, *TMODE* добија вредност 0 или 1, и пакује се у енкодирани ток података (енг. *bitstream*). Вредност 0 означава да нема тренутне амплитудне промене у подоквиру, а 1 ако је промена лоцирана.

3.1.3 ADPCM

На основу резултата добијених анализом тренутних амплитудних промена зависи хоће ли се извршавати предвиђање одбирака помоћу *ADPCM*-а или не. Наиме, *ADPCM* се примењује само у случају да не постоји тренутна амплитудна промена.

У идеалном случају, *ADPCM* користи корелацију између одбирака подопсега како би направио различите, међусобно неповезане, одбирке који ће имати мања варирања него оригинални одбирци. Уместо представљања сваког одбирка независно, као у *PCM* формату, *ADPCM* рачуна разлику између сваког аудио одбирка и његове предвиђене вредности и производи *ADPCM* вредности. За сваки прозор над којим се врши анализа (подоквир), одбирци подопсега се користе за оптимизацију скупа од четири коефицијента за предвиђање. Ови коефицијенти се оптимизују по оквиру, тако да је грешка предвиђања минимизирана по критеријуму најмањих квадрата по оквиру. Скуп од четири генерисана коефицијента се затим квантизује као вектор од четири елемента користећи стратегију претраге стабла векторске квантизације (*VQ*). Кодна књига *VQ* се састоји од 4096 вектора и сваки од елемената је 16-битни, тако да је величина *VQ* кодне књиге 32КВ.

Ове векторски кванитоване групе од 4 коефицијента за предвиђање се касније користе за добијање сигнала разлике одбирака подопсега у анализирајућем прозору. Неслагања ових сигнала разлике се пореде са неслагањима одбирака подопсега како би се

одредила корист предвиђања. Уколико је корист предвиђања већа од унапред дефинисаног прага, сигнал разлике се чува за касније кодовање. У супротном, сигнал разлике се одбацује, а за касније кодовање ће бити коришћени одбирци подопсега.

Употреба *ADPCM* кодовања је назначена индикатором методе предвиђања (*PMODE*) у компресованом току података. На овај начин, *ADPCM* се динамички укључује кад год је могуће презентовати одбираке текућег прозора са значајно мањим бројем бита.

Процедура предвиђања је следећа:

1. Обезбедити меморију (*sample[m]*) за смештање 4 одбирка, израчуната у претходном пролазу, и свих одбирака из текућег подоквира (16 одбирака)
2. Направити 5 аутокорелационих услова од одбирака подопсега и одбирака израчунатих у претходном пролазу, према формули:

$$AC_i = \sum_{m=0}^{N+4-i} (sample[m] * sample[m+i]), i = 0,1,2,3,4$$

где је N број одбирака подоквира у текућем подосегу (у овом случају 20)

3. Користећи ове услове, уз помоћ *Durbin*-овог алгоритма направити четири коефицијента за предвиђање $Alpha_1$ до $Alpha_4$.

$$rE(0) := AC(0)$$

$$rk_1 = \frac{AC(1)}{rE(0)}$$

$$rk_i = \frac{\left[AC(i) - \sum_{j=1}^{i-1} Alpha_j(i-1) \times AC(i-j) \right]}{rE(i-1)}, i = 2,3,\dots,p$$

$$Alpha_i(i) = rk_i, i = 1,2,\dots,p$$

$$Alpha_j(i) = Alpha_j(i-1) - rk_i Alpha_{i-j}(i-1), \begin{cases} i = 2,3,\dots,p \\ j = 1,2,\dots,i-1 \end{cases}$$

$$rE(i) = (1 - rk_i^2) \times rE(i-1), i = 1,2,\dots,p$$

где је $Alpha(i)$ i -ти коефицијент предвиђања за оптимални предвиђач. Резултати се рачунају рекурзивно на основу горе наведених једначина.

$Alpha_j(i-1)$ (у коду означена као *arOld_j*) је верзија $Alpha$ -е из претходне итерације једначина.

$AC[i]$ је i -ти аутокорелациони коефицијент израчунат према формулама из корака 2.

$rE[i]$ је међурезултат грешке који се користи за модификацију коефицијената у следећој итерацији једначина.

4. Коришћењем кодне књииге *VQ* проналази се вектор чији коефицијенти највише одговарају $Alpha$ коефицијентима, $Alpha'_i$, $i = 1, 2, 3, 4$, где је $Alpha'_i$ најсличнији $Alpha$ коефицијентима из расположиве кодне књииге. Најбоље поклапање коефицијената се добија двофазном претрагом стабла, и израчунава се коришћењем минималне квадратне грешке (ϵ_{VQ}).

$$\epsilon_{VQ} = \sum_{i=1}^4 (Alpha_i - VQelement)^2$$

5. Предвиђа се одбирак помоћу 4 коефицијента из кодне књииге, $Alpha'_i$, $i = 1, 2, 3, 4$

$$sample'[n] = \sum_{i=1}^4 Alpha'_i \times sample[n - i]$$

где је $sample'[n]$ нови предвиђени одбирак, $Alpha'$ најбоље поклапање $Alpha$ коефицијената из расположиве кодне књиге VQ и $sample[n-1]$ је низ одбирака који се рачунају, којима претходе $ADPCM$ одбирци израчунати у прошлом пролазу.

6. Вредност грешке се добија као разлика оригиналног одбирка и њему одговарајућег предвиђеног одбирка.

$$\epsilon_n = sample'[n] - sample[n]$$

7. Сабирају се квадратне вредности грешака и рачуна се средња квадратна грешка (енг. *mean square error, MSE*)

$$J = \frac{1}{2N} \sum_{n=1}^N \epsilon_n^2$$

8. Уколико се догоди да је средња квадрата вредност подоквира већа од производа $P_{gain_threshold} \cdot a$ и средње квадратне грешке, онда је:

$Pflag[ch][sb][subfrm] = 1$ означава да је метода предвиђања активирана

$SubfrmMSV[ch][sb][subfrm]$ = средња квадратна грешка подоквира, односно користи се вредност грешке.

$Subfrmrms[ch][sb][subfrm]$ = корен средње квадратне грешке подоквира, односно користи се *rms* вредност грешке.

Уколико је $\frac{\sum_{n=1}^N sample[n]^2}{N} > P_{gain_threshold} \times J$ онда следи,

$P_{flag}[ch][cb][subfrm] = 1$, што означава активирани метод предвиђања;

$$SubfrmMSV[ch][sb][subfrm] = J$$

$$Subfrmrms[ch][sb][subfrm] = \sqrt{J}$$

Стварни одбирци се затим замењују са сигналом грешке између оригиналног и предвиђеног одбирка, спремног за даље кодовање.

$$sample[n] = sample[n] - sample'[n] = \epsilon_n$$

3.1.4 Фактори скалирања и нормализација одбирака

Уколико је сигнал разлике (резултат $ADPCM$ -а) предодређен за кодовање, корен средњег квадрата (енг. *Root mean square, RMS*) вредности одбирака сигнала разлике се користи као фактор скалирања за подоквир. Ако је сигнал разлике био одбачен, онда се кодују одбирци подоквира и као фактор скалирања се користи RMS вредности одбирака подопсега. Пре саме квантизације, одбирци подопсега или грешке предвиђања сваког подоквира се деле са фактором скалирања који одговара том подоквиру.

Након тога, фактори скалирања се линеарно квантизују 7-битном линеарном табелом са 128 нивоа са кораком од 1.1 dB, која дозвољава динамички опсег од преко 140 dB за кодовање одбирака подопсега. Овај избор квантизационе табеле (нпр. индекс табеле) се такође уписује у излазни ток података за сваки анализирајући прозор.

3.1.5 Додела бита

Циљ алгоритма за доделу бита је расподела бита на основу резултата психоакустичке анализе сигнала, односно задржавање елемената потребних за очување највеће могуће верности аудио сигнала на декодерској страни. Метод доделе бита коришћен у 1.5Mbps кодеру је метод минималне средње квадратне грешке, чији је принцип рада следећи :

На почетку се сваком подопсегу, на основу конфигурације кодера и броја бита у секунди (енг. *Bit rate*), додели минималан број бита. Сви нераспоређени бити се деле између подопсега помоћу технике попуњавања (енг. *waterfill*). Ова техника функционише тако што итеративно додељује слободне бите опсезима са већом *RMS* енергијом.

Начин доделе бита се уписује у излазни ток података као 5-битна реч.

3.1.6 Скаларна квантизација

Након одређивања величине квантизационог корака и додељивања бита, *PCM* одбирци подопсега или *ADPCM* сигнал разлике могу бити квантизовани коришћењем једног од 26 квантизатора средњег нивоа (квантизатори 1 до 10 су кодовани у блоковима, а 11 до 26 су линеарно кодовани).

3.1.7 Канал за репродукцију ниских фреквенција (*LFE*)

Канал за репродукцију ниских фреквенција (који се такође назива и '.1' канал) се обрађује другачије од основних канала пуног фреквентног опсега. *LFE* канал проистиче из директног процеса пропуштања ниских фреквенција и децимације пуног фреквентног опсега *PCM* тока података коришћењем 64х децимационог филтера. Дискретизовани *PCM* одбирци се групишу у сагласности са анализирајућим прозором подопсега. Дискретизовани одбирци се нормализују фактором скалирања (*PCM* вредност текућег анализирајућег прозора), а затим квантизују као 8-битне речи. Квантизовани одбирци и фактор скалирања *LFE* канала се затим уписују у излазни ток података.

3.2 Конвертовање аритметике покретног зареза у аритметику непокретног зареза

Већина алгоритама за обраду звука су имплементирани у аритметици покретног зареза [16]. Аритметика покретног зареза се заснива на одвојеном израчунавању мантисе и експонента. Иако се израчунавања извршавају над различитим подацима, она нису потпуно независна. У случају сабирања у аритметици покретног зареза, једна мантиса мора бити померена како би се добили исти експоненти. Након сабирања, резултат је потребно нормализовати што утиче и на мантису и на експонент резултата. Аритметика непокретног зареза је слична целобројној аритметици. Сабирање два броја у непокретном зарезу се може извршити целобројним сабирањем, док множење захтева једно додатно померање резултата. Очигледно је да сложеније израчунавање у аритметици покретног зареза захтева више простора на процесору у односу на аритметику непокретног зареза [17].

Приликом имплементације алгоритама за обраду звука неопходно је водити рачуна о цени добијеног решења, па је употреба процесора са аритметиком покретног зареза неисплатива. Овај проблем се решава конверзијом оригиналног алгоритма у аритметику непокретног зареза са прихватљивим нарушавањем квалитета која је резултат ограничене дужине података.

Процес конвертовања аритметике покретног зареза у аритметику непокретног зареза се може поделити у две фазе [16]:

I. ФАЗА: Одређивање динамичког опсега сигнала (енг. *Dynamic range estimation*)

Приликом обраде сигнала потребно је одредити максималну $|x_{max}|$ и минималну $|x_{min}|$ апсолутну вредност променљивих које се користе приликом израчунавања што представља опсег вредности дате променљиве. Такође је потребно утврдити и за колико се две суседне вредности променљиве морају разликовати што представља резолуцију којом је потребно представити дати динамички опсег x_{res} . Однос највеће и најмање могуће вредности променљиве представља њен динамички опсег. Динамички опсег се може представити као однос опсега и резолуције (формула 3.1), бројем бита потребних за представљање дате променљиве (формула 3.2) или у децибелима [dB] (формула 3.3) [18].

$$Dynamic\ Range\ (Ratio) = \frac{x_{max}}{x_{res}} \quad (3.1)$$

$$Dynamic\ Range\ (Bits) = \log_2 \left(\frac{x_{max}}{x_{res}} \right) = N\ Bits \quad (3.2)$$

$$Dynamic\ Range\ (dB) = 20 \log_{10} \left(\frac{x_{max}}{x_{res}} \right) = 6.02 * N + 1.76 \quad (3.3)$$

Динамички опсег сигнала се одређује експериментално, обрадом ограниченог броја испитних сигнала. Овакав начин зависи од тога да ли је одабрани скуп сигнала довољно добар да предвиди све могуће улазне сигнале алгоритма који се имплементира. Проблем се јавља када неки неочекивани улазни сигнал проузрокује да поједине вредности унутар алгоритма прекораче задати опсег.

II. ФАЗА: Скалирање у непокретном зарезу (енг. *Fixed point scaling*)

На основу предвиђеног динамичког опсега, у свакој рутини имплементираних алгоритма одређује се колико улазни сигнал може бити скалиран, са прихватљивом грешком заокруживања (енг. *Rounding error*). Понекад се може десити да акумулиране грешке заокруживања доведу до прекорачења опсега иако се то није десило приликом испитивања оригиналног алгоритма. Из тог разлога, врши се додатно испитивање како би се утврдио утицај грешке заокруживања на динамички опсег.

3.3 Проблеми при писању асемблерског кода

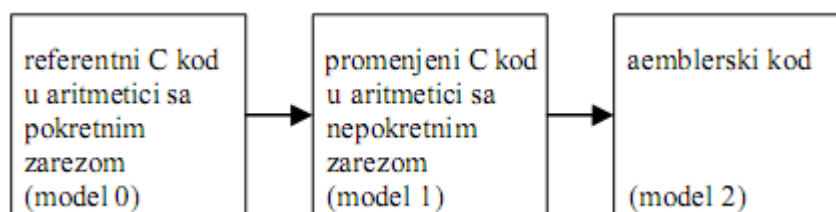
Приликом писања асемблерског кода неопходно је добро познавати архитектуру, инструкцијски сет и меморијске ресурсе циљног ДСП процесора. Услед ограничене процесорске снаге и расположиве меморије потребно је користити различите методе оптимизације за убразавање извршавања и величине кода. Често су ове технике међусобно искључиве, у смислу да се убрзање извршавања постиже на рачун повећавања кода, и обрнуто. Из наведеног разлога, методе оптимизације је потребно прилагодити циљном процесору и алгоритму који је потребно имплементирати.

Неке од метода оптимизације које се користе приликом писања асемблерског кода су [22]:

- *Оптимальна употреба регистара* - Ово је ефикасна метода оптимизације када се привременим или локалним променљивама доделе одрђени регистри уместо да се користе меморијске локације за њихово смештање. На тај начин, може се избећи често приступање меморији и уштедети значајан број циклуса. Због ограниченог броја опште наменских регистара (енг. *General purpose registers*) у функцијама треба смањити број локалних променљивих на минимум.
- *Користићење инструкције гранања* (енг. *Branch delay line*) - Код проточних архитектура, за инструкције гранања потребно је један или два циклуса пре извршавања скока на одредишну инструкцију. Због тога, у асемблерском коду потребно је инструкцију гранања позвати једну или две инструкције пре него што је треба извршити скок. На тај начин се може смањити вишак инструкција потребан за скок (енг. *Jump overhead*).
- *Користићење хардверске петље* (енг. *Zero overhead loop*) - Хардверске петље са регистрима бројача, почетне и крајње адресе, уклањају потребу за коришћењем додатног регистра као бројача и додатних инструкција за смањење бројача, проверу услова завршетка петље и скока на почетак петље.
- *Користићење специјализованог адресирања* - Специјализовано адресирање уклања потребу за додатним инструкцијама потребним за израчунавање адресе наредне меморијске локације којој се приступа.
- *Распетљавање петљи* (енг. *Loop unrolling*) - Ова метода омогућава да се искористе слободне инструкције унутар петље које су последица великог броја циклуса потребних да резултат множења постане доступан за смештање у меморију (енг. *MAC delay*). Ова техника доводи до повећања величине кода, па се не користи потуно рапетљавање, него само једног нивоа.
- *Спајање петљи* (енг. *Loop merging*) - У неким случајевима у алгоритму који се имплементира, постоје независне петље које имају исти број итерација. Комбиновањем таквих петљи у једну доводи до уклањања вишка инструкција потребних за припремање петљи (енг. *Loop overhead*).
- *Користићење паралелних инструкција* - Ако инструкцијски сет подржава извршавање више операција у паралели, потребно је приликом писања асемблерског кода посебно обратити пажњу на ову могућност и користити посебно у угњежденим петљама и петљама са великим бројем итерација.

4. Реализација

Полазна тачка је био код реализован у програмском језику *C*, у аритметици са покретним зарезом. До асемблерског кода, полазни код је прошао кроз неколико фаза, као што је приказано на слици 1.



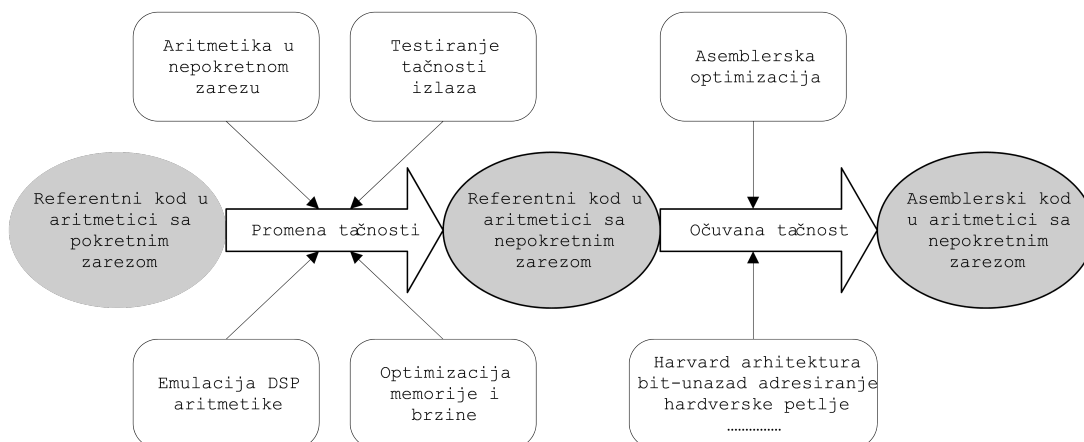
Слика 4.19 Графички приказ фаза реализације

- a) Модел 0 представља референтни *C* код у аритметици са покретним зарезом
- b) Модел 1 је модификовани *C* код у аритметици са непокретним зарезом, који даје занемарљиво лошије резултате у односу на референтни *C* код (због великих динамичких опсега и непрецизности аритметике са непокретним зарезом [23]).
- c) Модел 2 представља асемблерски код који даје идентичне резултате као модел 1.

Основни захтеви које је било потребно испунити приликом имплементације вишеканалног аудио кодера су:

- тачност (енг. *fidelity*) – добро поклапање излаза у непокретном зарезу са оригиналним излазима
- робустност (енг. *robustness*) – задовољавајући резултати обраде за све очекиване улазе
- брзина извршавања (енг. *performance*) – имплементација у непокретном зарезу мора бити довољно брза како би оправдала деградацију тачности излаза

Како би се испунили сви горе наведени захтеви, имплементација се одвијала у неколико фаза као што је приказано на слици 4.2.



Слика 4.20 Детаљни приказ имплементације вишеканалног аудио кодера [24]

Први корак у имплементацији је био да се аритметика у референтном алгоритму замени аритметиком подржаном од стране циљне платформе.

Након тога, следило је исцрпно испитивање модификованог алгоритма, како би се задовољила прва два захтева: тачност и робустност. Током испитивања није се водило рачуна о ефикасности алгоритма. Испитивање је извршено помоћу слушног испитивања, али је спроведено и на *PC* платформи, која нуди велики број апликација које су потребне током прилагођења аритметике (алати за обраду звука, алати за аутоматско одређивање субјективног квалитета звука,...).

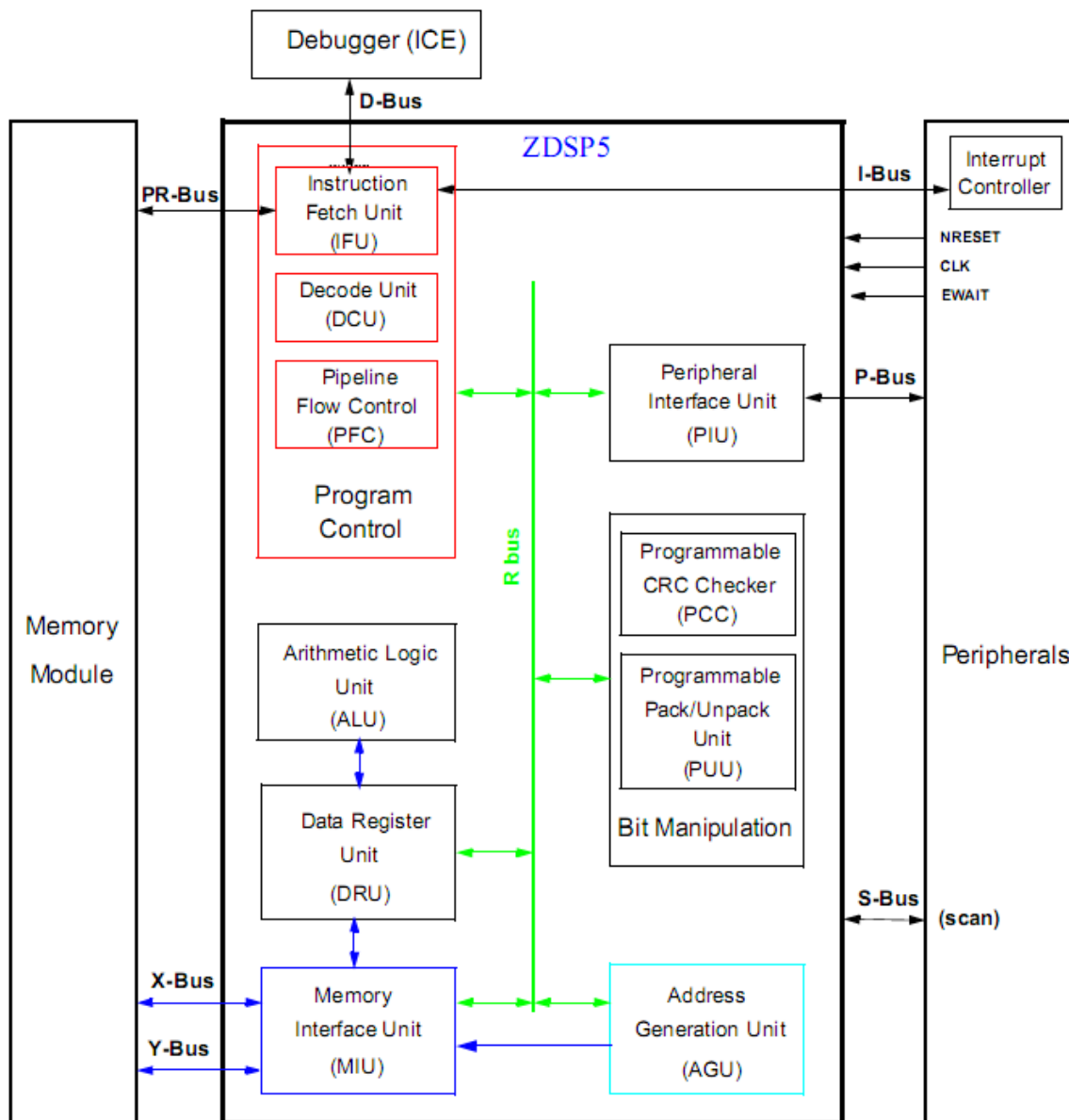
Поред замене аритметике, референтни алгоритам је модификован тако да величина података одговара меморијским ресурсима циљног ДСП процесора.

После модификација полазног алгоритма и испитивања добијен је алгоритам који користи аритметику циљног процесора и обезбеђује тражену тачност. Овај алгоритам представља полазну тачку за реализацију асемблерског програма. Како асемблерски програм и модификовани алгоритам користе исту аритметику и типове података, сви резултати и међурезултати се морају поклапати.

Уз детаљно познавање карактеристика циљног ДСП процесора и оптимизација на нивоу асемблерског програма задовољен је и трећи захтев – ефикасност извршавања.

4.1 Опис циљне платформе

Вишеканални аудио кодер је реализован на аудио ДСП-у, израелске компаније *Zoran*, са кодним именом *ZDSP5*. Овај процесор има проточну структуру (енг. *pipeline*) од 7 нивоа и 32-битну аритметику са непокретним зарезом, такт процесора је 108 MHz, има 102 ефективна *MIPS*-а и 14K речи меморије. Подржава и 20-битну аритметику са непокретним зарезом. Компатибилан је са *ZR38XXX* фамилијом процесора.



Слика 4.22 спрежни подсистеми ZDSP5 процесора

4.1.1 Опис развојних алата

Развојно окружење ZDSP5 процесора представља скуп развојних алата: асемблер, повезивач и ADP симулатор. Развој апликације пролази кроз следеће кораке:

1. Развој алгорита
2. Програмирање алгорита у асемблерском језику у виду модула
3. Креирање дефиниција нивоа (енг. *overlay*) који ће у себи садржати написане модуле
4. Превођење модула и нивоа, и њихово повезивање са осталим модулима и нивоима
5. Покретање кода на процесору или симулатору

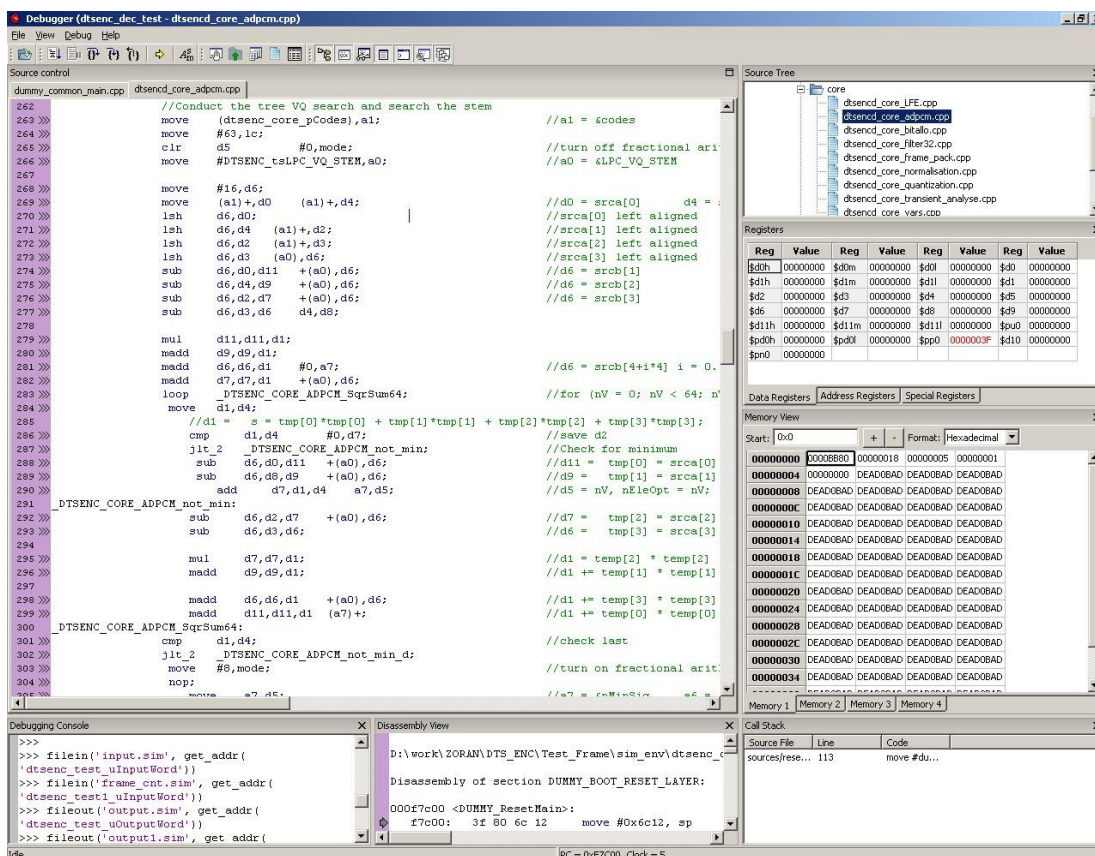
ADP симулатор представља развојно окружење које омогућава тражење и откривање грешака у асемблерском коду, на PC платформи. Састоји се од једног прозора, слика 4.5, у којем се опционо могу налазити следеће компоненте:

1. Приказ кода
2. Приказ меморије
3. Приказ датотека у којима се налази изворни код
4. Приказ регистрара
5. Приказ променљивих чија се вредност прати

Симулатор подржава два режима рада програма:

1. Корак по корак режим, у коме се може пратити целокупно стање процесора, кроз стања меморије и регистрара
2. Нормални режим рада, у којем се програм извршава у својој пуној брзини

За ефикасније отклањање грешака и олакшано пројектовање апликација, симулатор подржава постављање прекидних тачака (енг. *breakpoints*) које се могу поставити на више начина: избором опције *debug/breakpoints* из главног менија програма или двоструким кликом миша на жељену линију кода.



Слика 4.23 ADP развојно окружење

4.2 Методологија развоја програмске подршке на наменској платформи

Алгоритам је реализован у три фазе, при чему свака фаза има одговарајући модел [25].

- Модел 0 представља референтни *C* код у аритметици са покретним зарезом
- Модел 1 представља модификовани *C* код који подржава аритметику непокретног зареза и даје занемарљиво лошије резултате у односу на модел 0
- Модел 2 представља асемблерски код који даје идентичне резултате као модел 1.

4.2.1 Модел 0

Анализа улазних тест вектора и анализа референтног кода. Модел нула представља оригинални код (референтни *C* код).

4.2.2 Модел 1

Када се реализује неки пројекат много је битније питање саме функционалности алгоритма на ком се заснива пројекат него питање имплементације на циљној платформи. За то постоје два разлога:

- Референтни *C* код је написан за потребе развоја, тестирања и верификације алгоритма, у овој фази развоја алгоритма ово су много битнији аспекти кода у односу на саму ефикасност
- Референтни *C* код није посебно прилагођен ни једној циљној платформи.

Сходно томе референтни *C* код није погодан за писање кода прилагођеног циљној платформи, па постоји низ оптимизација и модификација комплетног референтног кода које би требало применити у овој фази имплементације, као што је потребно написати и библиотеку математичких функција које се користе у референтном коду, користећи нумеричке алгоритме. Приликом писања ових функција потребно је обратити пажњу на тачност као и на њихову комплексност (потребно је да буду што једноставније како би имале што мању потрошњу). Проблематичне математичке функције у овом раду су биле:

- Дељење
- Множење 64 бита са 32
- Корен

Дељење је реализовано на самом процесору, кореновање је избачено у потпуности изменом алгоритма (без нарушавања тачности), а множење је реализовано као скуп више инструкција.

4.2.2.1 Замена приступања елементима низа преко индекса приступањем преко показивача

Ово је вероватно техника оптимизације која највише смањује број линија кода у односу на било који други тип оптимизације. Неке процене указују на то да овај тип оптимизације доноси од 10% до 30% смањења броја линија кода, и чак 60% побољшања код брзине извршавања кода. Овај тип оптимизације представља трансформацију стандардног приступања елементима низа преко индекса у *C* језику у индиректно приступањем елементима низа.

4.2.2.2 Смањивање броја аргумената функције

Како би се *C* код што је могуће више приближио асемблерском, аргументи функције се дефинишу као глобалне променљиве или чак константе, вредност променљиве се тада директно користи на месту где је потребна.

4.2.2.3 Редуковање структура

Пољима структуре у референтном *C* коду се обично приступа преко показивача на структуру због модуларности кода., али у функционалном делу кода већина показивача на структуру показује само на једно одређено поље структуре.

Имајући у виду да је приступ пољима структуре преко показивача на структуру, мање ефикасан од директног присуца, може се закључити да ако се приступ пољима структуре замени директним приступом, величина кода се може значајно смањити.

4.2.3 Модел 2

Како је асемблерски код чврсто везан за архитектуру циљног процесора, он није подржан *C* преводиоцем на *PC* рачунару, нити може бити емулиран на једноставан начин. Као последица тога код у моделу 2 се не може извршавати и испитивати у развојном окружењу *Visual C++*. Уместо тога асемблерски код се мора извршавати на симулатору циљне платформе или на циљној платформи намењеној за развој и испитивање асемблерског кода, и тако испитивати у циљу отклањања грешака у режиму контролисаног извршавања програма.

Модел 2 представља код који даје идентичне резултате као *C* код у моделу 1. Асемблерски код је реализован у *ADP* развојном окружењу, описаном у овом поглављу, који је даље испитан и на циљној платформи.

Развој апликације за циљну платформу представља итеративни поступак. То значи да ако се проценом утрошка ресурса процени да решење не задовољава унапред познате критеријуме, приступа се модификовању решења полазећи од модела 1.

Множење 64 бита са 32 у асемблеру је реализовано на следећи начин:

```
mulsu d4,d3,d1;
ashi #-32,d1;
madd d2,d4,d1;
```

64-битни број је у регистрима *d2* и *d3*, у *d2* су горња 32 бита, а у *d3* су доња 32 бита. У *d4* је 32-битни број. *d1* је акумулатор од 72 бита у који се смешта 64-битни резултат множења. Пошто су у *d3* доњи бити 64-битног множиоца, он се мора помножити као неозначен број (инструкцијом *mulsu*). Производ доњих 32 бита са 32-битним множиоцем се смешта у доњи део акумулатора и зато се њихов производ помера за 32 у десно (инструкцијом *ashi*). Након тога се множе горњих 32 бита и 32-битни множилац, и њихов производ се сабира са претходним резултатом и тиме се добија 64-битни коначни производ у акумулатору *d1*.

4.2.4 Оптимизација програмске подршке

Реализација алгоритма на процесору за наменске уређаје захтева: малу потрошњу, односно што мање процесорског времена (*MIPS*-а), и мали утрошак меморије, односно што мању цену крајњег уређаја.

Процес кодовања се одвија у 8 функција, табела 4.1. Због недостатка *C* преводиоца, све функције су ручно написане и након тога се додатно оптимизовале оне које су трошиле највише процесорског времена. Пре оптимизације функција, потрошња је била око 130 *MIPS* -а. Употребљено је 8К речи меморије. Због мале унутрашње меморије, кодна књига је морала да иде у спољну меморију и приликом њеног добављања су се дешавали застоји у раду процесора, док се чекало на завршетак пребацивања меморије из спољне у унутрашњу. Због тога су убачени асинхрони *DMA* трансфери како би се подаци добављали из меморије за време обраде и тиме што мање чекало на завршетак пребацивања.

Користиле су се још неке технике за оптимизацију [26]:

- Издвајање провера појединих услова и условних скокова изван програмских петљи, као и
- скраћивање програмских петљи већим коришћењем паралелног извршавања инструкција

Након оптимизације потрошња се смањила на око 70 *MIPS*-а, за 6-канални (5.1) аудио сигнал чији су одбирци ширине 24 бита.

Табела 4.2 Потребан број *MIPS* -а за сваку од функција (после оптимизације)

Функција	Потрошња <i>MIPS</i> -а
LFE	3.18
Sample quantization	7.34
Frame packing	1.22
Sample prediction with ADPCM	7.86
Filter bank	11.04
Pre-echo prevention and transient	0.62
Scale factors and samples normalization	15.12
Bit allocation	21.46

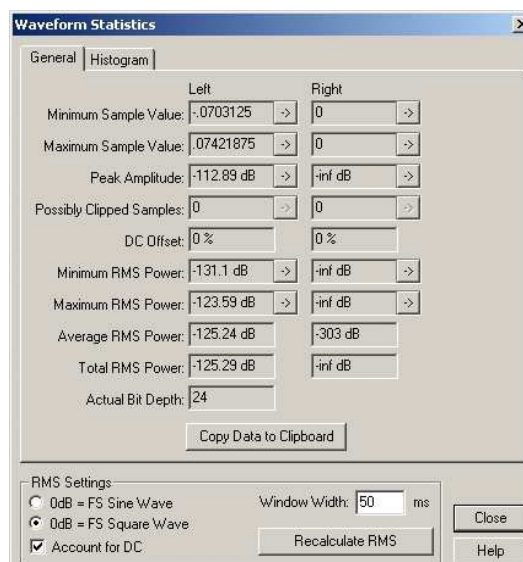
Овај процесор има уграђену функцију за паковање бита у излазни битски ток, која је веома добро имплементирана, тако да је паковање бита заузело тек нешто више од једног *MIPS*-а (око 1.22).

5. Испитивање и верификација

Верификација исправности рада програмске подршке често је врло сложен проблем. При пројектовању скупова испитних случајева се морају узети у обзир сва могућа стања програмске подршке што је практично немогуће због сложености алгорита. Код аудио алгорита, испитни случајеви се углавном заснивају на испитивању одзива система на синусоидалне сигнале. Ипак, најважнији испитни случајеви су они који oponaшају реалну примену програмске подршке, то јест испитивање слушањем.

Код испитивања програмске подршке циљ је да асемблерски код не уноси грешку у односу на *C* код у аритметици са непокретним зарезом. Приликом модификације референтног кода и замене аритметике, потребно је спровести испитивање квалитета аудио сигнала.

Реализовани кодер је испитиван у неколико фаза. Модел 1(*C* код у аритметици са непокретним зарезом) се због великих динамичких опсега који су повећали непрецизност у аритметици са непокретним зарезом није могао поредити на бит идентичност са моделом 0. Одступања у битима су била велика (у неким случајевима чак 15 бита од 24 су се разликовала за поједине одбирке), али су те разлике испод нивоа шума пристуног у сигналу (у овом случају испод -90dB). Поређење је извршено тако што се инвертовани испитни сигнал додао, са преклапањем, на референтни сигнал (одузимање сигнала) и добијени сигнал се анализирао. На слици 5.1 је пример статистичких података о сигналу.



Слика 5.24 Статистишке информације разлике сигнала

Због тога је спроведена и субјективна метода мерења квалитета аудио сигнала. Једна од најчешће коришћених субјективних метода мерења квалитета обрађеног аудио сигнала је метода која се заснива на испитивању слушањем, код које већи број слушалаца оцењује квалитет приказаних испитних сигнала, користећи оцене на скали од 1 до 5, где 5 означава одличан квалитет, док 1 означава незадовољавајући, односно лош квалитет испитног сигнала, табела 5.1. Крајња оцена квалитета добија се рачунањем средње вредности оцене свих слушалаца. Овако добијена оцена обично се назива „средња оцена мишљења“ (*Mean Opinion Score, MOS*). Препоруке за спровођење *MOS* испитивања дефинисане су у оквиру стандарда ITU-R BS 1284-1 [28]. Испитивање модела 1 је спроведено над 26 субјеката, и просечна оцена квалитета аудио сигнала је 4.923.

Табела 5.3 Субјективна скала заснована на ITU-R BS.1284 стандарду у опсегу од 1.0 до 5.0 [28]

субјективни квалитет сигнала	оцена	изобличење
веома добар	5	нечујно
добар	4	чујно, али не смета
средњи	3	мало смета
слаб	2	смета
лош	1	веома смета

Испитивање асемблерског кода и испитаног *C* кода се вршило поређењем на бит идентичност, са дозвољеним одступањем од 1 бита (због различитог заокруживања и сличног), јер људско ухо не може да региструје ту разлику. Испитивање је вршено и у току реализације асемблерских функција. Са испитаним *C* кодом су се правили међурезултати који би асемблеру били представљени као улаз у функцију. На тај начин се испитивала одређена функција написана у асемблеру. Испитивање се вршило како у симулатору, тако и у радном окружењу у коме су се могла пратити стања свих регистара и меморије. Када су све функције биле написане, испитивање се спровело над неколико различитих аудио сигнала. Резултати су се поредили са резултатима над моделом 1 и сви су били бит идентични, чиме је установљено да решење задовољава задате критеријуме референтног модела, као и ограничења циљне платформе.

5.1 Резултати испитивања

Табела 5.4 Резултати испитивања

Назив испитног сигнала	Број тестираних оквира	Максимална разлика у битима	Битска ширина одбирака	Провера статистике сигнала	Конфигурација канала
4824-1kHz-15dBClip.wav	93	7	24		
4824-1kHz1750Hz-0dB-120dB.wav	187	6	24		
4824-1kHz-30Hz-20dB.wav	93	4	24		
4824-200Hz-1kHz-20dB.wav	93	4	24		
4824-200Hz-30Hz-20dB.wav	93	4	24		
4824-200Hz-4kHz-20dB.wav	93	4	24		
4824-200Hz8kHz-	93	5	24		

4kHz8kHz-20dB.wav					
4824-24tone.wav	93	9	24	Spectral diff max is: -92.57 dB	
4824-QuadTone-12dB.wav	468	7	24		
Назив испитног сигнала	Број тестираних оквира	Максимална разлика у битима	Битска ширина одбирака	Провера статистике сигнала	Конфигурација канала
4824-sw-0dB-20db.wav	6187	7	24		
4824-sw-linear-0- 100dB.wav	2812	7	24		
4824-1kHz-30Hz- 120dB0dB.wav	3375	15	24	Spectral diff max is: -103.35 dB	
4824-24tone-30Hz.wav	93	5	24		
4824-30Hz-20dB.wav	93	4	24		
4824-32tone.wav	93	7	24		
4824-32tone-30Hz.wav	93	5	24		
4824-3tone-20dB.wav	93	4	24		
4824-3tone-80dB.wav	93	3	24		
4824-500Hz-2kHz- 20dB.wav	93	4	24		
4824-500Hz-2kHz- 500HzPLUS2kHz- 20dB.wav	93	5	24		
4824-512fr-Impulse.wav	93	8	24		
4824-8kHz-20dB.wav	93	5	24		
4824-MIPS-Intensive- FrntSurr.wav	6744	9	24	Spectral diff max is: -98.73 dB	
4824-MIPS-Intensive.wav	93	5	24		1.0.0
4825-MIPS-Intensive.wav	93	5	24		1.0.1
4826-MIPS-Intensive.wav	93	5	24		2.0.0
4827-MIPS-Intensive.wav	93	5	24		2.0.1
4828-MIPS-Intensive.wav	93	9	24	Spectral diff max is: -89.13 dB	3.0.0
4829-MIPS-Intensive.wav	93	9	24	Spectral diff max is: -89.37 dB	3.0.1
4830-MIPS-Intensive.wav	93	11	24	Spectral diff max is: -72.58 dB	2.2.0
4831-MIPS-Intensive.wav	93	11	24	Spectral diff max is: -74.95 dB	2.2.1
4832-MIPS-Intensive.wav	93	13	24	Spectral diff max is: -60.59 dB	3.2.0
4833-MIPS-Intensive.wav	93	13	24	Spectral diff max is: -62.38 dB	3.2.1

6. Закључак

У оквиру овог рада је приказана једна реализација вишеканалног кодера са губицима на платформи са ограниченим ресурсима (*ZDSP5*, израелске компаније *Zoran*) у саставу транскодера који ће бити уграђиван у корисничке пријемнике дигиталне телевизије. Као полазни код коришћен је *C* код у аритметици са покретним зарезом. Основни циљ је био да се направи кодер који ће радити у реалном времену на овом наменском процесору (потрошња изражена у *MIPS*-има и количина заузете меморије да не буду већи од граничних вредности, за циљну платформу) без нарушавања квалитета аудио сигнала.

Квалитет предложеног решења је утврђен на основу слушног испитивања и поређења на бит идентичност.

Ресурси потребни за реализацију уклопили су се у оквире циљне платформе, то јест захтеви за ресурсима које намеће реализовани алгоритам су мањи од оних које нуди циљна платформа, остављајући места за реализацију додатних компоненти. Укупно процесорско време обраде је смањено на око 70 *MIPS*-а. Искоришћеност програмске меморије је око 8000 речи.

Даљи правци развоја и могућности унапређења се огледају у оптимизацији реализованог решења, како у погледу *MIPS* потрошње, тако и у величини меморијског заузећа и прецизности.

7. Литература

- [1] N. Teslić, V. Mihić, G. Miljković, I. Rešetar, V. Kovačević: *Pregled arhitektura za prijem digitalne televizije*, 2005.
- [2] N. Horowitz: *NRDC Study of Set Top Box and Game Console Power Use*, 2007.
- [3] G. Lukatela, D. Drajić, G. Petrović, R. Petrović: *Digitalne Telekomunikacije*, Građevinska knjiga, Beograd, 1990.
- [4] Dr. Miodrag V. Popović: *Digitalna Obrada Signala*, Akademska misao, Beograd, 2003.
- [5] V. Madisetti, D. Williams: *Digital Signal Processing Handbook*, 1998.
- [6] D. D. Greenwood: *Critical bandwidth and the frequency coordinates of the Basilar membrane*, 1961.
- [7] E. Zwicker, H. Fastl: *Psychoacoustics Facts and Models*, Berlin, 1990.
- [8] B. Scharf: *Critical bands*, New York, 1970.
- [9] R. Hellman: *Asymmetry of masking between noise and tone*, Percep. Psychphys., vol. 11, pp. 241–246, 1972.
- [10] M. Schroeder, B. S. Atal, J. L. Hall: *Optimizing digital speech coders by exploiting masking properties of the human ear*, 1979.
- [11] S. D. Stearns: *Arithmetic coding in lossless waveform compression*, IEEE Trans. Signal Proc., 1995.
- [12] V. Kovačević, M. Popović, M. Temerinac, N. Teslić: *Arhitekture i algoritmi digitalnih signal procesora I*. FTN Izdavaštvo, Novi Sad, 2005.
- [13] S. Samir: *Field Programmable Gate Arrays FPGAs*, [Internet] Version 18, Knol. 2010.
- [14] Guido Costa Souza de Araujo: *Code Generation Algorithms For Digital Signal Processors*, Princeton University, 1997.
- [15] S. Perović, J. Kovačević: *Jedno rešenje radnog okruženja audio kodeka na platformi sa ograničenim resursima*, 2007.

- [16] T. Aamodt: *Floating-Point to Fixed-Point Compilation and Embedded Architectural Support*, Applied Science Graduate Department of Electrical and Computer Engineering, University of Toronto, 2001.
- [17] P. Farrett: *The MPEG Audio Standard*, <http://peterfarrett.com/mpeg-article.html>
- [18] R. Sharma: *Floating Point to Fixed Point Conversion*, Audio DSP Ksharing, <http://rajeshsharma.co.in/docs/FixedPointConversion.pdf>
- [19] J. D. Johnston, K. Brandenburg: *Wideband Coding – Perceptual Considerations For Speech And Music*, Advances in Speech Signal Processing, Furui and Sondhi (Ed.), Marcel Dekker, 1991, Preprint 3011 A-4
- [20] *Method For Objective Measurements Of Perceived Audio Quality*, Recommendation ITU-R BS.1387-1, Geneva, Switzerland, 1998-2001
- [21] *Methods For The Subjective Assessment Of Small Impairments In Audio Systems Including Multichannel Sound Systems*, International Telecommunications Union, Radiocommunication Sector BS.1116, Geneva 1997.
- [22] M. Arora, N. C. Lahane, P. Srinivasan: *Assembly Code Optimization Techniques for Real Time DSP Implementation of Speech Codecs*, Proc. of National Communication Conference, India, 2002.
- [23] T. Aamodt, „Floating-Point to Fixed-Point Compilation and Embedded Architectural Support“, 2001.
- [24] M. Gayer, M. Lohwasser, M. Lutzky: *Implementing MPEG Advanced Audio Coding And Layer-3 Encoders On 32-Bit And 16-Bit Fixed-Point Processors*, Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany
- [25] M. Đukić, N. Četić, J. Kovačević, M. Popović: *A C Compiler Based Methodology For Implementing Audio DSP Applications on a Class of Embedded Systems*, ISCE, IEEE, ISBN: 978-1-4244-2422-1, 2008.
- [26] D. Mišković, M. Popović, Z. Jovanović, „Jedno rešenje optimizacije koda pomeranjem instrukcija bezuslovnih skokova i arhitekturom podržanih petlji“, 2004.
- [27] B. Milovanović, M. Temerinac, N. Teslić, J. Joković, *Digitalizacija zemaljskih TV sistema*
- [28] ITU-R BS. 1284-1: *General methods for the subjective assessment of sound quality*, 2003.