|  |  |
| --- | --- |
|  | **UNIVERZITET U NOVOM SADU**  **FAKULTET TEHNIČKIH NAUKA**  **Odsek za elektrotehniku i računarstvo**  **Institut za računarstvo i automatiku**  **Katedra za računarsku tehniku i računarske komunikacije** |
|  |  |

**Algoritmi i arhitekture DSP**

**Projektni zadatak:**

**Realizacija algoritma kombinovanja kanala na Cirrus Logic DSP platformi**

**Student: Simon Radosavljević RA206/2020**

Novi Sad 2023

Sadržaj

[1. Opis zadatka 1](#_Toc531251795)

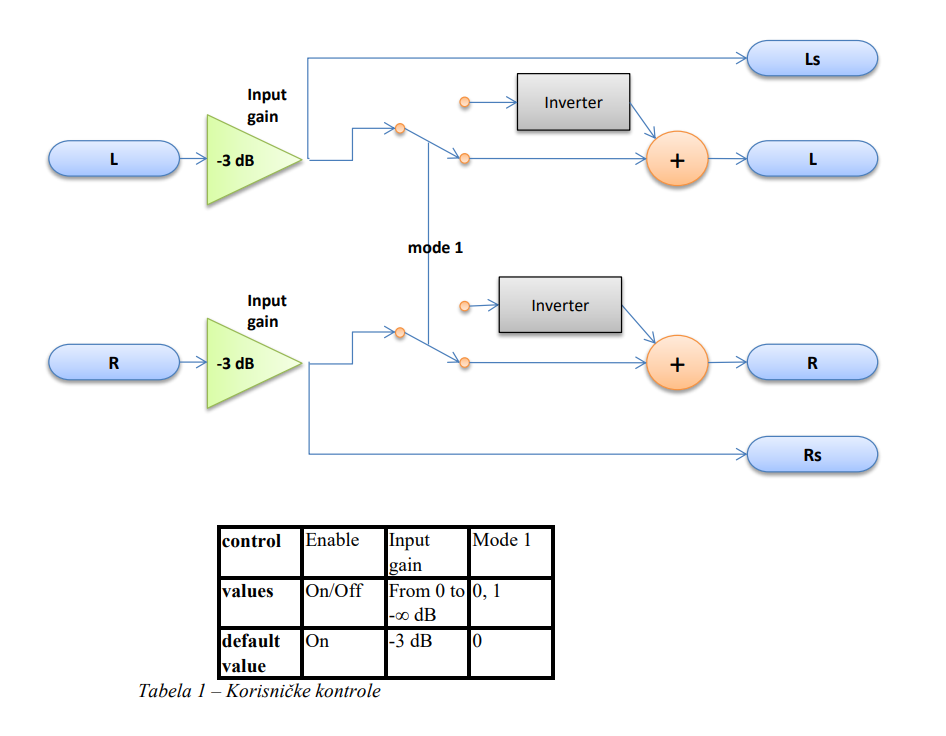
[2. Opis realizacije 2](#_Toc531251795)

[3. Ispitivanje i verifikacija](#_Toc531251798) 3

1. **Opis zadatka**

-Realizacija projektnog zadatka zasnovana je na implementaciji kombinovanja kanala na osnovu date šeme i tabele. Sam zadatak se sastoji iz implementacije 4 modela, modela 0 1 2 i 3. Model 0 1 i 2 je bilo potrebno implementirati u Visual Studi-u dok je model 3 i finalni model bilo potrebno implementirati u Clideu-u.

Model 0 je referentni model koji sadrži osnovnu implementaciju šeme. Model 1 je sličan modelu 0 samo što je malo optimizovaniji. Razlika izlaza iz modela 0 i 1 mora biti 0 bita. Model 2 je realizovan prelazak iz floating point u fix point arhitekturu, gde je dozvoljena razlika u odnosu na ostale modele 1 do 2 bita. Poslednji korak nam je model3 i on predstavlja potpuni prelazak na ciljnu arhitekturu.

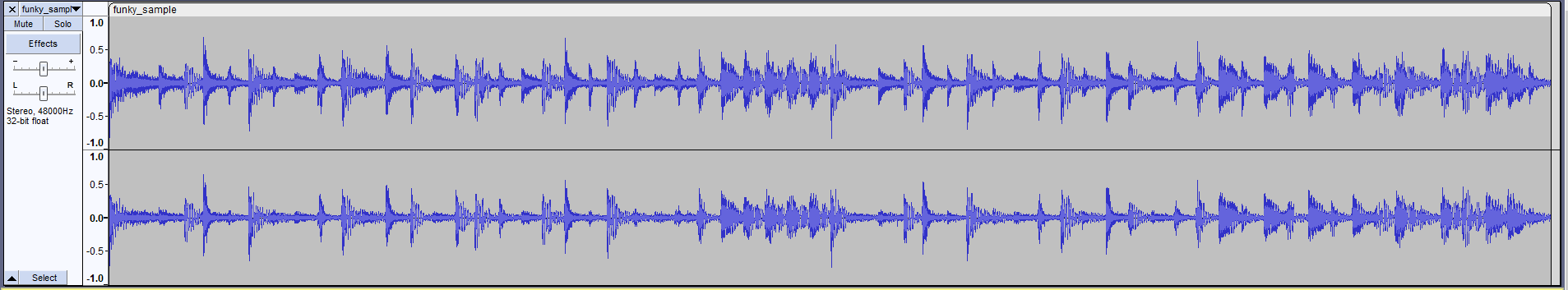


1. **Opis realizacije**

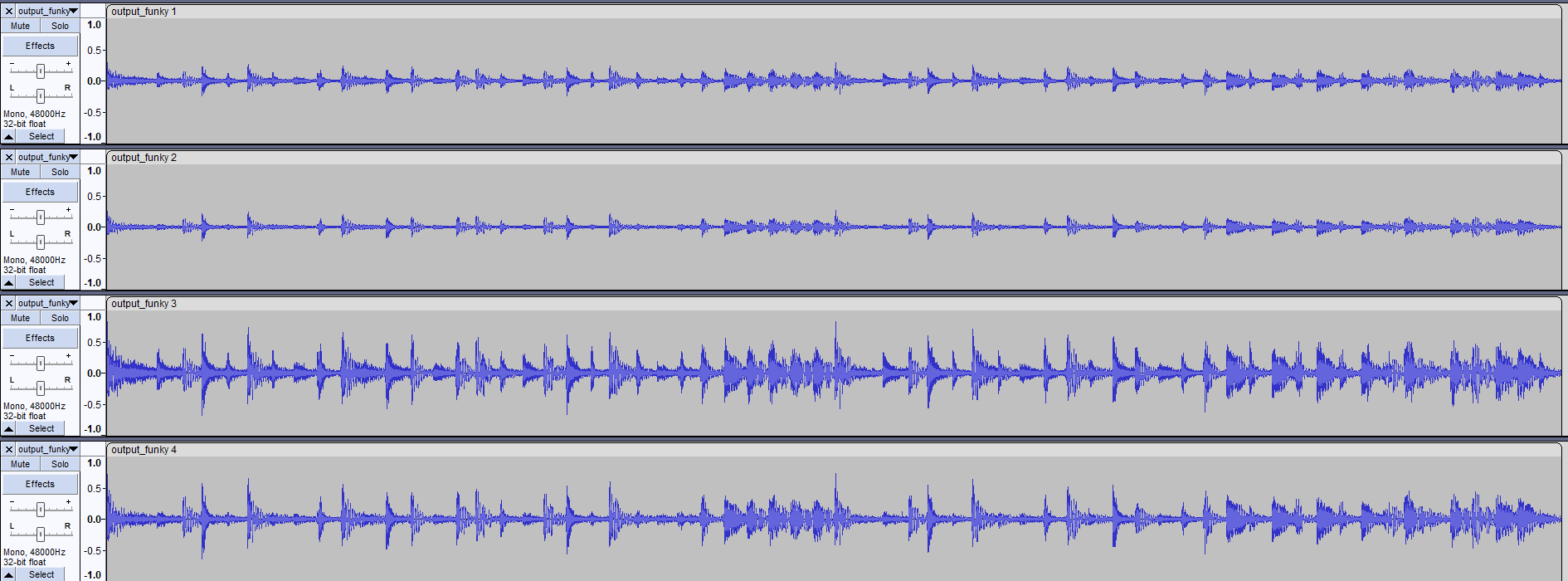
**Model0** je implementiran bez obaziranja na potrošnju memoirje i bilo kakve optimizacije vezane za ciljnu arhitekturu. Inverter efekat je primenjen u odvojenom fajlu inverter.cpp (inverter.h) gde se nalazi funkcija za inicijalizaciju kao i funckija za transformaciju.

Na slici ispod predstavljen je jedan od izlaza modela 0, gde se konkretno radi o funky\_sample.wav.

Taj ulazni signal izgleda ovako:

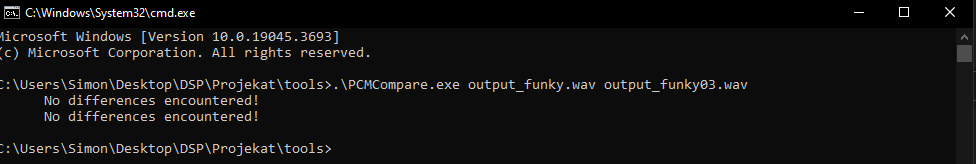


Kada ga provučemo kroz model 0 u slučaju da je enable 1 I mode 0 dobicemo izlaz sa 4 kanala od kojih su prva dva invertovana(L i R su invertovani).



**Model1** u ovom modelu su odrađene funkcionalne optimizacije referentnog koda, tačnije modela 0. Uklonjene su lokalne strukture, kao i C indeksiranje (jer bi C indeksiranje kasnije zauzimalo mnogo instrukcija), smanjen je broj argumenata kod određenih funckija čime je rasterećen stek.

Pomoću PCMCompare proverena je razlika između modela0 i modela1

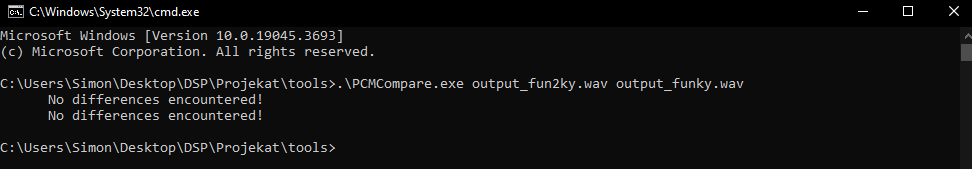


**Model2** je izveden iz modela1 pri čemu je prelazak u fixed point izvšen pomoću dodavanja datoteke koje sadrže emulacione klase za tipove za tipove sa nepokretnim zarezom kao što su fixed\_point\_math.h stdfix\_emu.h

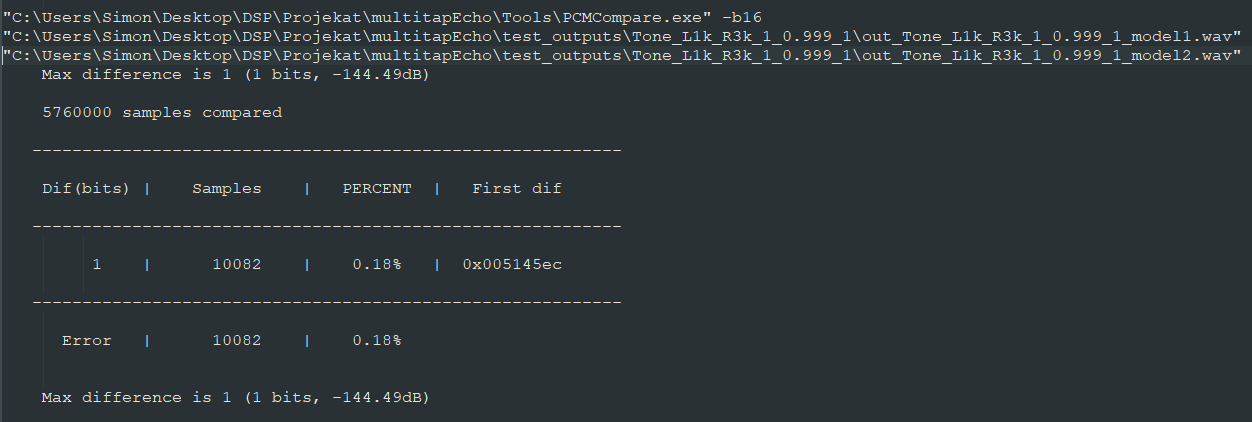
fixed\_point\_math.cpp i stdfix\_emu.cpp.

Izvršena je zamena tipova podataka pokretnom zarezu. Korišćena je dodela vrednosti FRACT\_NUM svakom elementu i pretvaranja konstati u nepokrentom zarezu upotrebom FRACT\_NUM

Razlika izmedju modela 1 I modela2 prikazana je na slici:

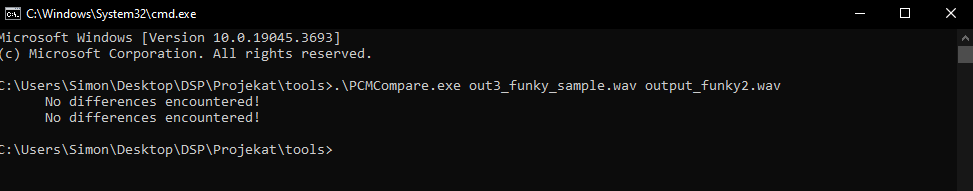


Medjutim, kod nekih ulaznih signala desava se da ima 1 bit razlike(kod Tone\_L1k\_R3k\_1\_ u nekim slucajevima).



**Model 3** je izveden iz modela 2 pri čemu je kod prebačen u CLIDE (u čist C).

Pordenjenjm izlaza iz modela 2 i 3 nema razlike u bitima:



Nakon toga izvršena je priperma C koda za assembler, tako što smo dodali \_\_memY kvalifikatori koji označavaju da će se podaci smeštati u Y memoriju prilikom čega moramo dodeliti taj kvalifikator svakom pokazivaču koji će u nekom trenutku pokazivati na sampleBuffer.

Funckija gainProcessing je implementirana kao I funckije za inicijalizaciju I invertovanje singala.

1. **Ispitivanje I verifikacija**

Ispitivanje je vršeno pomoću skripte koja je data na vežbama. Testirana su 3 signala funky\_sample.wav

speech\_2ch.wav Tone\_L1k\_R3k.wav. Parametri koji su prosleđivani su enable gain i mode. Izlaz skripte dat je u test\_outputs folderu.

-Nažalost provera utroška MIPS-a kao ni finalni mode nisu urađeni.