**Arhitekture I algoritmi DSP**

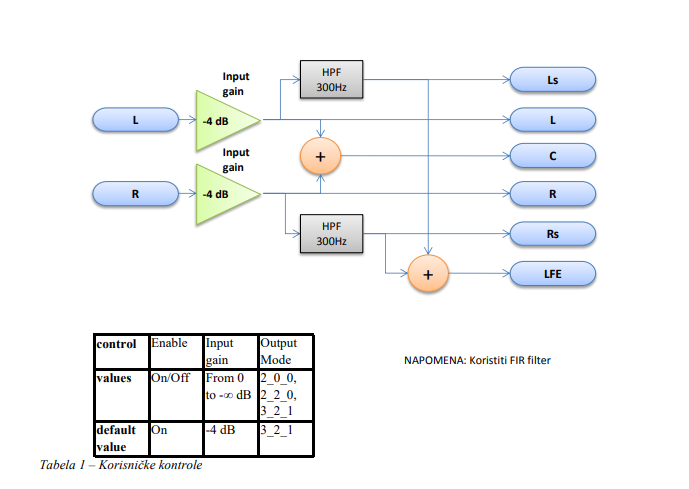
**Projektni zadatak:**

Realizacija algoritma kombinovanja kanala na Cirrus Logic DSP platformi br 46.

Mihailo Babić RA52/2020

# **Opis zadatka**

Zadatak je bio implementacija algoritma (zadatak46) datog sa slike:



Bitni parametri date obrade su:

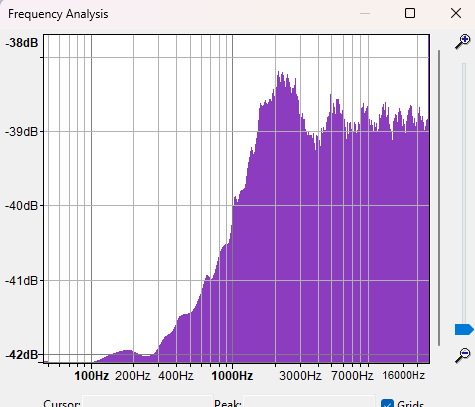
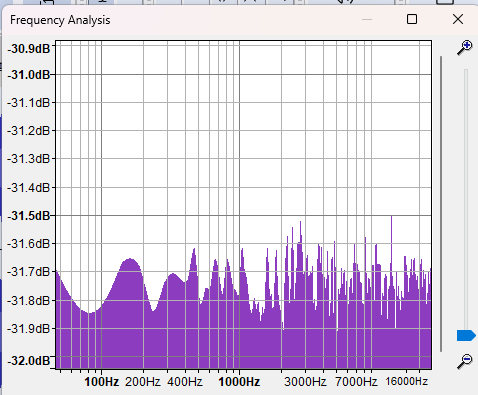
* Gain (-inf, 0)
* HPF (High Pass Filter) granične frekvencije 300Hz
* Kanalni mod od kojeg zavisi broj izlaznih kanala
* Enable, koji nam određuje da li se signal obrađuje

# **Opis realizacije**

## Model 0

Zadužen je za referentni izlaz pomoću kojeg verifikujemo sve ostale modele

Radno okruženje je Visual Studio, C++ programski jezik, Razvija se isključivo kako bi procesing algoritam proradio. Izlazi iz svih ostalih modela se porede sa izlazima iz modela 0.



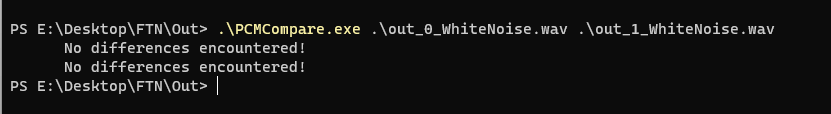
Analiziramo izlaz iz referentnog modela, na osnovu slušanja, spektralnih analiza kanala i očekivanih rezultata, proveravamo referentni izlaz. Na osnovu ove slike vidimo da nas filter filtrira signal na 300Hz.

Sve bitne parametre smo unosili preko argumenata main funkcije programa (enable, gain, mode)

## Model 1

Svrha modela 1 su funkcionalne optimizacije C++ koda, smanjivanjem broja argumenata funkcija obrade, uklanjanje lokalnih struktura i promenljivih, prilagođavanje operacija pristupa podacima, umesto indeksiranja koristimo pristup preko pokazivača na element, kao i optimizacija programskih petlji izbacivanjem nezavisnih delova koda u petlji

Izlaz iz modela 1 mora biti u potpunosti isti kao i model 0, što znači da bitska razlika mora biti 0.



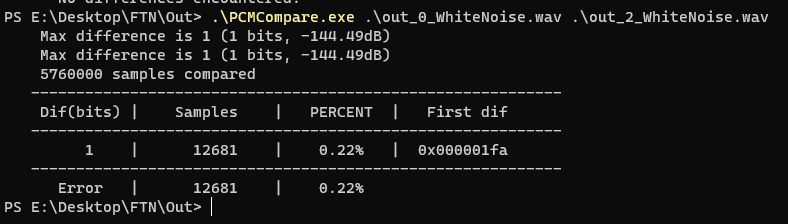
Primer poređenja izlaza signala iz modela 0 i modela 1

## Model 2

U modelu 2 glavni zadatak nam je prelaženje sa aritmetike pokretnog zareza u aritmetiku fiksnog zareza koja se koristi na ciljanoj platformi.

Pomoću emulacionih biblioteka simuliramo takve tipove podataka.

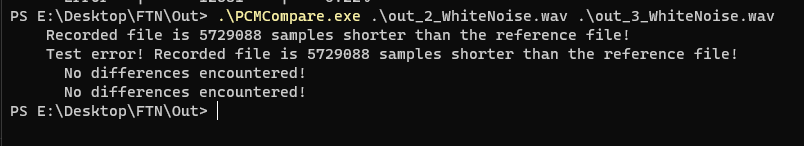
Cilj poređenja modela 0/1 i modela 2 je da imamo najviše 1-2 bita razlike



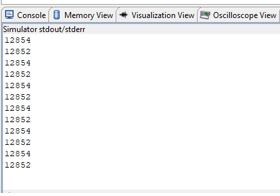
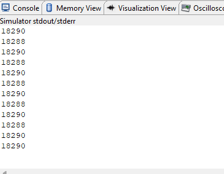
## Model 3

Prebacujemo model 2 iz VS u CLIDE okruženje, i prevodimo kod iz C++ u C jezik. Umesto emulacionih fajlova koristimo ugrađenu biblioteku. Vrednosti ne unosimo preko argumenata već ih hardkodujemo gain

Razlika izlaza između modela 2 i modela 3 mora biti 0 bita.



Naredni korak je razvijanje naše funkcije obrade u asembleru, radi optimizacije koda i smanjena broja ciklusa potrebnih za izvršavanje



Poređenje broja ciklusa između C funkcije (ugašen debug) i ASM funkcije. ASM funkciju je dalje moguće optimizovati. Naredna faza modela 3 je portovanje modela da radi kao DSP Aplikacija gde korisnik preko MCV kontrola kontroliše ulazne parametre.