

Projektni Zadatak 2021/2

Arhitekture i algoritmi DSP-a 2

Autor: Aleksa Heler

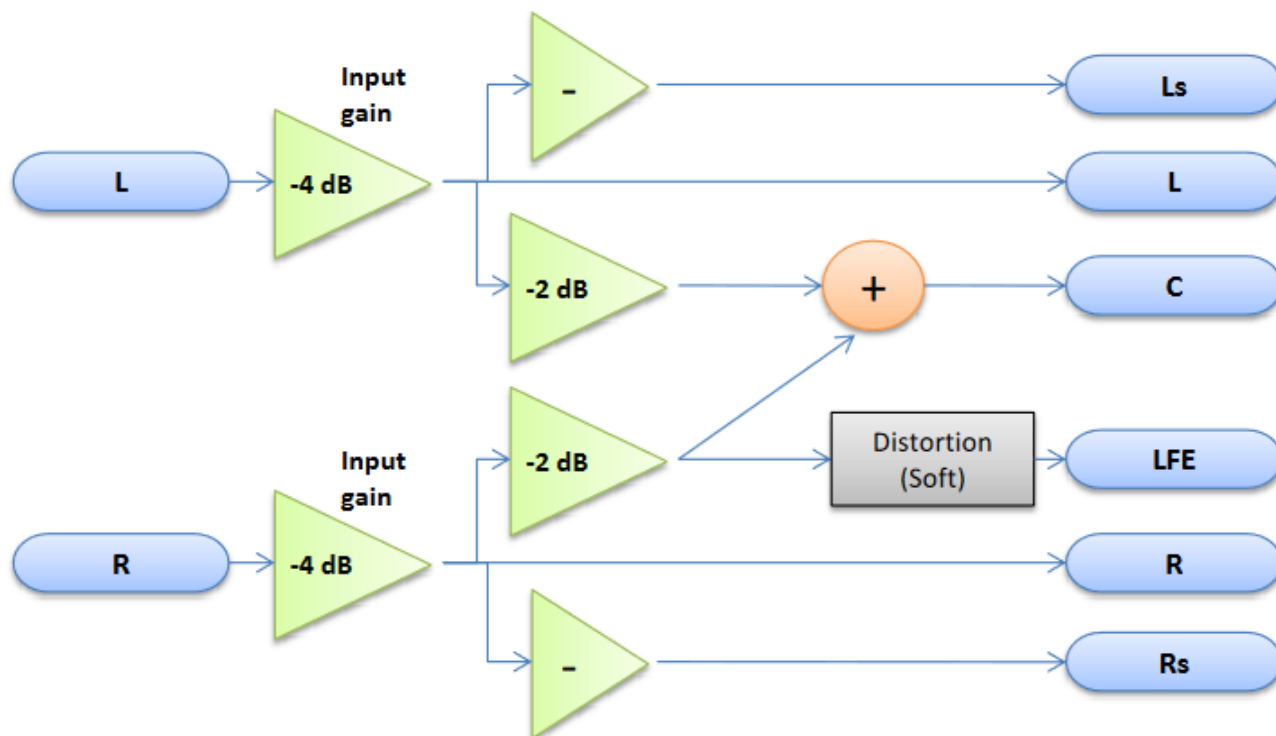
Fakultet tehničkih nauka, Novembar 2021.

Sadržaj

Opis zadatka.....	3
Opis realizacije.....	4
Model 0.....	4
Model 1.....	4
Model 2.....	4
Optimizacija distorzije.....	4
Model 3.....	5
Profilisanje.....	5
Simulator.....	6
Ispitivanje i verifikacija.....	7
Model 0 / model 1.....	7
Model 1 / model 2.....	7
Model 2 / model 3.....	7

Opis zadatka

Realizovati kombinovanje kanala na osnovu date šeme i tabele. Za implementaciju koristiti razvojno okruženje Cirrus Logic IDE (CLIDE).



control	Enable	Input gain	Output Mode
values	On/Off	From 0 to $-\infty$ dB	2_2_0, 2_2_1, 3_2_0, 3_2_1
default value	On	-4 dB	2_2_0

Opis realizacije

Model 0

Prvi korak je realizovati **referentni C kod u aritmetici pokretnog zareza** (*floating point*). Za ovo je korišćeno okruženje *Visual Studio 2019 Community*. Obrada se vrši po blokovima. Jedan blok predstavlja 16 odbiraka signala. **Na ulaz dolazi 2 kanala** (L, R), a **izlaz predstavlja do 6 kanala** (L, R, C, LFE, Ls, Rs). Omogućiti prosleđivanje naziva ulaznih i izlaznih datoteka kao i vrednosti kontrola (*gain*, *output mode*) preko parametara komandne linije. Ovde se ne vodi računa o optimizaciji, već direktnoj implementaciji algoritma, što će biti korišćeno kao referentni sistem za dalju optimizaciju i verifikaciju.

Model 1

U ovom koraku vrši se **optimizacija referentnog C koda**. Primer je indeksiranje u *for* petljama. Dok se u referentnom kodu koristi oblik *array[i]* to nije optimalno, već se koriste pokazivači i svaku iteraciju se oni povećavaju.

Model 2

Ovaj korak predstavlja **prilagođavanje koda da koristi nepokretni zarez** (*fixed point*). Kako C ne podržava nepokretni zarez kao primitivan tip, koristi se biblioteka za emuliranje pomoću klase.

Optimizacija distorzije

U ovom koraku je primena distorzije dodatno optimizovana. Počevši od referentne jednačine za računanje *positive soft knee* oblasti distorzije, koja izlazi iz opsega [0, 1) potrebnog za *fixed point*:

$$output = \frac{(3 - (2 - 3 \cdot input)^2)}{3}$$

Dolazi se do pojednostavljenog oblika koji ulazi u opseg I jednostavnija je za izračunati:

$$output = -0.08333333 + input - 0.75 \cdot input^2;$$

Analogno ovome pojednostavljena je i jednačina za *negative soft knee* oblast.

Model 3

Sledeći korak je implementacija koda u **CLIDE** okruženju kao *Standalone (ULD)* projekat. U ovom koraku bilo je potrebno svim globalnim promenljivama dodeliti memorijsku zonu, tj. da li će biti čuvane u X ili Y memoriji.

Profilisanje

Nakon prebacivanja modela u CLIDE okruženje, bilo je potrebno odrediti koliko instrukcija je potrebno za određene zahtevne delove obrade, kao i pregled iskorišćene memorije.

Upotrebom biblioteke *dsplib/timers.h* otkriveno je da funkcija *processing()* u fajlu *processing.c* koristi 2135 ciklusa prilikom izvršavanja. Pritom je iskorištena memorija data u sledećoj tabeli:

Broj lokacija	Memorija
2.576	X
257	Y
1.120	CODE

Simulator

Ispitivanje i verifikacija

Za ispitivanje i verifikaciju izlaznih datoteka svih modela korišćena je alatka **PCMCompare.exe** koja se nalazi u folderu *Audio*. Radi automatizacije napravljena je *batch* skripta **test.bat**. Koja pokrene sve modele koji su kompajlovani i koristeći prethodno pomenutu alatku poredi izlazne datoteke kako bi se pronašle sve razlike u izlazu. Detaljni izveštaju se nalaze u fajlovima *Docs/compare_modelX_vs_modelY.txt* gde *X* i *Y* predstavljaju dva modela koja se poredi.

Model 0 / model 1

Ni jedna razlika između izlaza modela 0 i modela 1 nije pronađena. Izlazni fajlovi su identični.

Model 1 / model 2

Na 8.896.608 uzoraka (*sample*) upoređenih, samo 24 se razlikuju za 1 bit. Što prestavlja približno 0% grešku, koja se može objasniti prelaskom na nepokretni zarez (*fixed point*) koji može uvesti minimalne greške usled načina na koji se brojevi beleže. Ovaj rezultat je očekivan i zadovoljavajuć.

Model 2 / model 3

Na istom broju uzoraka kao i prošli primer (izlazni fajlovi su slični), pronađeno je čak 2021 uzorak koji se razlikuje za 1 bit. I ova razlika se svodi na isti problem prelaska na nepokretni zarez kao i prošli primer. Takođe je rezultat očekivan i zadovoljavajuć.