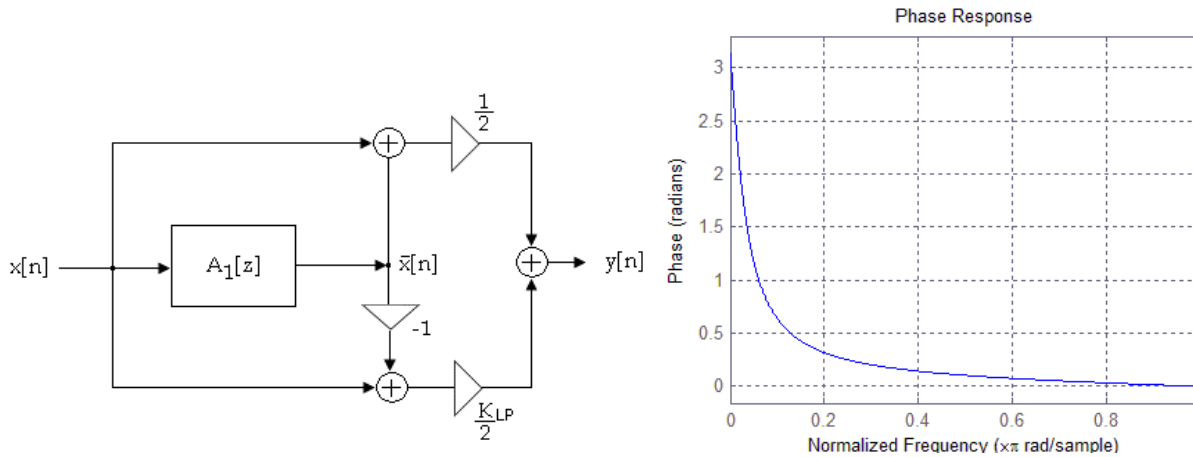


## 1 Kontrola niskih i visokih tonova (bass/treble)

Jednostavno kolo za regulaciju nivoa (pojaćanje ili slabljenje) niskih frekvencija prikazano je na slici 1a. Filter  $A_1$  je allpass filter prvog reda (primer fazne karakteristike dat je na slici 1b). Ulazni signal  $x[n]$  i signal na izlazu allpass filtera  $\bar{x}[n]$  imaju jednaku amplitudsku karakteristiku ali su im faze različite. Za  $\omega = 0$  razlika faza ova dva signala je  $\pi$  (ponišćavaju se), dok za  $\omega > \omega_c$  razlika faza teži nuli (za  $\omega = 1$  razlika je nula). Sabiranjem ova dva signala ponišćiće se niske frekvencije, dok će srednje i visoke ostati nepromenjene (gornja grana kola na slici 1a). Oduzimanjem ova dva signala ponišćiće se srednje i visoke frekvencije, dok će niske ostati nepromenjene (donja grana kola na slici 1a). Koeficijent  $K_{LP}$  određuje nivo niskih frekvencija u krajnjem signalu  $y[n]$  – ako je manji od 1 doći će do slabljenja niskih frekvencija, a ako je veći od jedan doći će do pojaćanja niskih frekvencija. Srednje i visoke frekvencije ostaće nepromenjene. Filter predstavljen na slici 1a nazivamo shelving filter prvog reda. Prenosnu karakteristiku shelving filtera moguće je menjati bez ponovnog projektovanja filtera – menja se samo koeficijent  $K_{LP}$ .



Slika 1 - a) NP shelving filter prvog reda; b) Fazna karakteristika allpass filtera prvog reda za  $\alpha = 0.9$ .

Prenosna funkcija allpass filtera prvog reda  $A_1$  je:

$$A_1(z) = \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}},$$

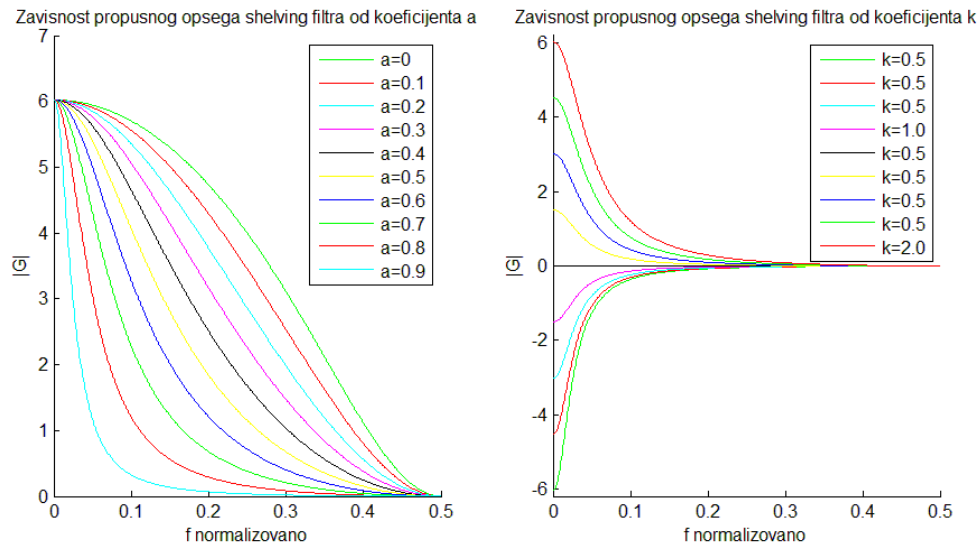
pa je prenosna funkcija shelving filtera:

$$G_1(z) = \frac{K_{LP}}{2} [(1 - A_1(z))] + \frac{1}{2} [(1 + A_1(z))]$$

Granična učestanost shelving filtera zavisi samo od koeficijenta  $\alpha$  i data je sledećim izrazom:

$$\omega_c = \cos^{-1}\left(\frac{2\alpha}{1+\alpha^2}\right)$$

Zavisnost prenosne funkcije shelving filtra od koeficijenta  $\alpha$  (definiše se u procesu dizajna filtra) i koeficijenta  $K$  (zadaje se u procesu upotrebe filtra) prikazana je na slici 2.



Slika 2 - Zavisnost propusnog opsega NP shelving filtra od koeficijenata  $\alpha$  i  $K$

Implementacija niskopropusnog shelving filtra u prgoramskom jeziku C, po uzoru na blok dijagram prikazan na slici 1, data je sa:

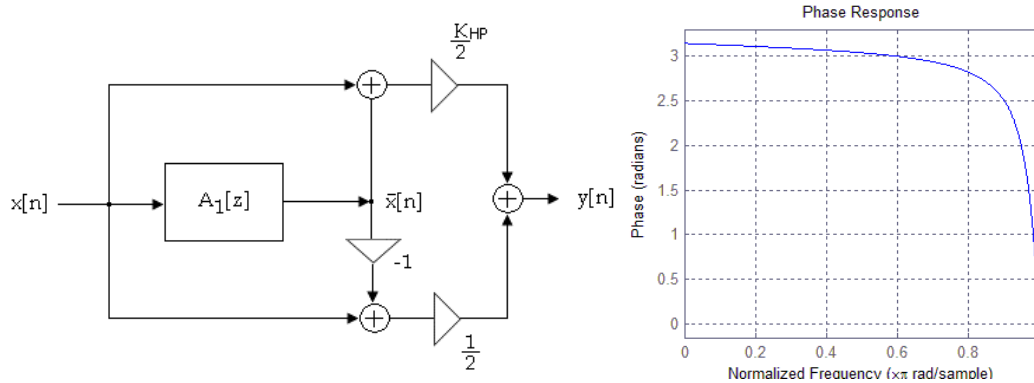
```
double shelvingLP(double input, double* coeff, double* z_x, double* z_y, double k)
{
    double filtered_input, output;
    double accum;

    filtered_input = first_order_IIR(input, coeff, z_x, z_y);
    accum = (input + filtered_input)/2.0;
    accum += ((input - filtered_input)/2.0)*k;
    output = CLIP(accum);

    return output;
}
```

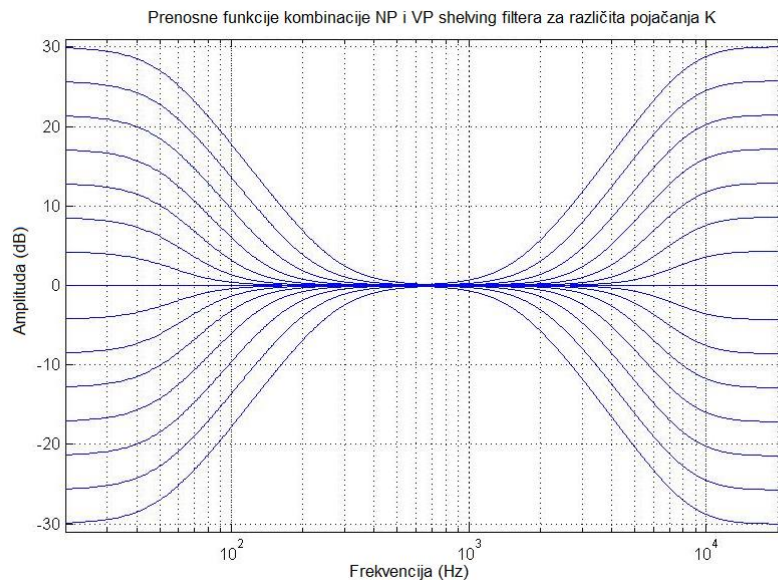
Na prethodnom primeru pomenljiva *input* predstavlja ulazni odbrak, *coeff* niz sa koeficijentima all-pass filtra prvog reda, *z\_x* i *z\_y* promenljive za pamćenje prethodnog ulaza i izlaza iz IIR filtra i *k* koeficijent pojačanja.

Analogno niskopropusnom shelving filteru pravimo i shelving filter za regulisanje visokih frekvencija. Razlika je da koeficijent  $\alpha$  za VP shelving filter ima negativnu vrednost u opsegu (0, -1) i da se regulisanje nivoa promene visokih frekvencija vrši u gornjoj grani (slika 3a). Fazna karakteristika allpass filtera prvog reda za negativnu vrednost koeficijenta  $\alpha$  ( $\alpha = -0.9$ ) predstavljena je na slici 3b.



Slika 3 - a) VP shelving filter prvog reda; b) Fazna karakteristika allpass filtera prvog reda za  $\alpha = -0.9$

Kombinacijom NP i VP shelving filtera moguće je nezavisno regulisati niske i visoke frekvencije. Na slici 4 prikazan je primer kombinacija prenosnih funkcija koje mogu da se ostvare takvom vezom.



Slika 4 - Primer prenosne funkcije kaskadno vezanih NP i VP shelving filtera za različite vrednosti pojačanja  $K$ .

## Postavka i opis zadatka

1. Realizovati sistem za kontrolu niskih i visokih tonova na osnovu prethodnog opisa. Realizovati referentni program koristeći tipove sa pokretnim zarezom (double, float). Iskoristiti šablon sa vežbi.
2. Obrada se vrši po blokovima. Jedan blok predstavlja 16 odbiraka signala. Na ulaz dolazi  $N \in [1, 8]$  kanala, a izlaz predstavlja isti broj kanala kao na ulazu. Potrebno je primeniti filtriranje na sve ulazne kanale.
3. Kontrolne promenljive (MCV) treba da omoguće uključivanje/isključivanje modula, promenu parametra  $K$  (jedna vrednost za sve kanale), i odabir granične frekvencije  $f$ .
4. Omogućiti prosleđivanje naziva ulaznih i izlaznih datoteka kao i vrednosti kontrola preko parametara komandne linije.
5. Na osnovu referentnog C koda realizovati C kod prilagođen aritmetici Crystal DSP procesora, poštujući metodologiju izrade modula (od modela 1 do modela 3).
6. Testirati izlaze modela 0, 1, 2 i 3 koristeći date test vektore i PCMCompare.exe alat za poređenje datoteka. Voditi računa da se pokriju sve karakteristične vrednosti korisničkih kontrola. Automatizovati proces testiranja upotrebom *batch* skripte.
7. Uraditi procenu utroška resursa (MIPS, memorija).
8. Sprovesti CCC specifične optimizacije.
9. MIPS kritične sekcije implementirati upotrebom asemblerskog jezika. Mogu se koristiti ugrađeni (*inline*) asemblerski iskazi ili implementacija čitavih funkcija u asemblerskom jeziku.
10. Korake 6, 7 i 8 izvršavati iterativno. Poželjno je voditi evidenciju o sprovedenim optimizacijama, i broju ušteđenih ciklusa za svaku optimizaciju.
11. Integrisati prethodno dobijeni modul u programsko okruženje. Napraviti Overlay i Application projekte za razvojnu ploču i simulator.
12. Testirati izlaze iz simulatora sa rezultatima modela 3.
13. Pokrenuti projekat na razvojnoj ploči. Snimiti izlazni audio koristeći Audacity program. Porediti snimljene datoteke sa izlazima iz modela 0 (spektralno poređenje).
14. Ponavljati korake 6, 7 i 8 dokle god se postiže ubrzanje.
15. Na kraju uraditi detaljnu procenu utroška resursa i napisati prateću dokumentaciju. Rešenje testirati na osnovu priloženih test vektora. Smatra se da je test vektor prošao proveru ukoliko su izlazi iz referentnog koda bit-identični sa izlazima implementacije koda prilagođenog za prevođenje odnosno da se spektralne slike izlaza iz referentnog koda poklapaju sa spektralnim slikama izlaza implementacije na ciljnoj platformi.

## Rezultat rada

Nakon implementacije i testiranja realizovane programske podrške potrebno je isporučiti kod programa u elektronskoj formi. Rok za predaju projekta je 01.07.2019. godine. Pored predaje projekta potrebno je održati usmenu prezentaciju rešenja zadatka. Predati projekat je kompletan ukoliko sadrži:

- *Visual Studio* projekte za Model 0, 1 i 2

- *Standalone ULD* projekat sa kodom Modela 3
- *Code Module* projekat koji sadrži integraciju Modela 3 u programsko okruženje
- *Application* i *Overlay* projekte za pokretanje rešenja na ploči i simulatoru
- Skriptu za automatizaciju testiranja
- Dokumentaciju

Dokumentacija o radu sadrži 1-5 stranica. Prilikom pisanja dokumentacije naglasak staviti na rezultate rada:

- potrošnju programske memorije,
- potrošnju memorije za podatke i
- potrošnju MIPS-a za konačno rešenje
- rezultate poređenja izlaznih datoteka sa referentnim

Dokumentacija takođe treba da sadrži detalje o testiranim delovima sa rezultatima i opisom eventualnih problema koji su postojali u radu. Šablon za projektnu dokumentaciju dat je u okviru zadatka.