

# Realizacija algoritma kombinovanja kanala na Cirrus Logic DSP platformi

---

## Cilj

Realizovati kombinovanje kanala na osnovu date šeme i tabele. Za implementaciju koristiti razvojno okruženje dato na vežbama.

## Radno Okruženje

Windows, Visual Studio 2015, Cirrus Logic SDK.

## Postavka i opis zadatka

1. Na osnovu šeme kombinovanja kanala prikazane na slici 1 i tabele kontrola za dato kombinovanje kanala realizovati referentni C kod u aritmetici pokretnog zareza.
2. Obrada se vrši po blokovima. Jedan blok predstavlja 16 odbiraka signala. Na ulaz dolazi 2 kanala, a izlaz predstavlja do 8 kanala.
3. Omogućiti prosleđivanje naziva ulaznih i izlaznih datoteka kao i vrednosti kontrola preko parametara komandne linije.
4. Na osnovu referentnog C koda realizovati C kod prilagođen aritmetici Crystal DSP procesora, poštujući metodologiju izrade modula (od modela 1 do modela 3).
5. 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.
6. Uraditi procenu utroška resursa (MIPS, memorija).
7. Sprovesti CCC specifične optimizacije.
8. MIPS kritične sekcije implementirati upotrebom asemblerskog jezika. Mogu se koristiti ugrađeni (*inline*) asemblerski iskazi ili implementacija čitavih funkcija u asemblerskom jeziku.
9. Korake 6, 7 i 8 izvršavati iterativno. Poželjno je voditi evidenciju o sprovedenim optimizacijama, i broju ušteđenih ciklusa za svaku optimizaciju.
10. Integrisati prethodno dobijeni modul u programsko okruženje. Napraviti Overlay i Application projekte za razvojnu ploču i simulator.
11. Testirati izlaze iz simulatora sa rezultatima modela 3.
12. Pokrenuti projekat na razvojnoj ploči. Snimiti izlazni audio koristeći Audacity program. Porediti snimljene datoteke sa izlazima iz modela 0 (spektralno poređenje).
13. Ponavljati korake 6, 7 i 8 dokle god se postiže ubrzanje.
14. 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 08.03.2018. 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.