

Projektni zadatak

Dodavanje muzičkih efekata u audio signal korištenjem razvojnog okruženja ADSP-21489

Objavljen: 29.12.2023. godine

Rok za predaju: 1. termin - 07.02.2024. godine

2. termin - 21.02.2024. god. - **Osvojeni broj bodova se skalira sa 0,75.**

U slučaju da student ne uradi projektni zadatak koji je pozitivno ocijenjen u navedenim rokovima, nema pravo izlaska na završni ispit.

U sklopu projektnog zadatka je potrebno realizovati sistem za dodavanje muzičkih efekata u audio signal korištenjem razvojnog okruženja ADSP-21489. Pored samog *CrossCore Embedded Studio* programskog paketa, studenti su dužni da relevantne signale na kojim će biti vršeno testiranje generišu i u okviru Pythona, te da rezultate dobijene na samoj ploči uporede sa rezultatima dobijenim u Pythonu. Takođe, od studenata se zahtjeva korištenje Git sistema za verzionisanje koda.

Digitalni audio efekti su osnovi današnje muzičke produkcije. Audio efekti se prema načinu obrade signala najčešće dijele na:

- filtriranje - nisko-visokopropusni filtri, ekvilajzer,
- vremenski promjenjivi filtri - Wah-wah, phaser,
- kašnjenje - vibrato, flanger, chorus, echo, delay,
- modulatori - ring modulacija, tremolo,
- nelinearna obrada - kompresija, limiter, distorzija, noise gate,
- specijalni efekti - panning, reverb, surround, pitch shifter, rotary speaker simulation...

Iako je studentima dozvoljeno da izvrše realizaciju proizvoljnih audio efekata, u nastavku će biti data lista određenih efekata za gitaru, koje je moguće odabrati za implementaciju:

Grupa 1 - Manje zahtjevni filtri

- Delay,
- Compressor,
- Noise Gate,
- Equalizer,
- Volume Pedal,
- Tape Saturation,

- Octave Pedal,
- Envelope Filter.

Grupa 2 - Umjereno zahtjevni efekti

- Chorus,
- Flanger,
- Phaser,
- Tremolo,
- Rotary Speaker Simulator,
- Auto-Wah,
- Bit Crusher.

Grupa 3 - Zahtjevniji efekti

- Wah-Wah,
- Reverb,
- Pitch Shifter,
- Harmonizer.

Projektni zadatak zahtijeva implementaciju **minimalno tri audio efekta**, pri čemu bar jedan efekat mora biti izvan Grupe 1. Efekti iz više grupe (složeniji efekti) nose veći broj bodova. Nakon izbora audio efekata potrebno je izvršiti realizaciju datog efekta na DSP razvojnom okruženju, izvršiti profilisanje koda radi identifikacije bottleneck-ova, dokumentovati iskorišćene optimizacione metode, te predstaviti prednosti i mane implementiranih metoda. Na kraju je potrebno izvršiti poređenje rezultata dobijenih na DSP ploči sa rezultatima dobijenim u Pythonu. Osnovni dijelovi realizacije se mogu sumarizovati u narednih par koraka:

- **Generisanje referentnih signala u Pythonu:** korištenjem odgovarajućih biblioteka (npr. NumPy, Scipy i Matplotlib) generisati ulazne i izlazne signale u Pythonu.
- **Implementiracija istog efekta na ADSP procesoru:** implementirati isti efekat na razvojnom okruženju.
- **Analiza performansi:** testirati brzinu izvršavanja implementiranog algoritma na DSP-ju, kao i zauzeće memorije, te pokušati implementacijom različitih optimizacionih metoda uticati na poboljšanje performansi.
- **Validacija rezultata:** korištenjem profajlera, izvršiti poređenje rezultata dobijenih na DSP-ju, sa očekivanim rezultatima dobijenim u Pythonu. Nacrtati dobijene signale, kao i signal greške.
- **(Opciono) Korekcija implementacije algoritma na DSP-ju:** ako se uvide izražene razlike između signala dobijenog na DSP-ju i signala dobijenog u Pythonu, predložiti načine postprocesiranja signala sa ciljem smanjenja greške odstupanja.

Svi navedeni koraci treba da se izvršavaju na ADSP ploči. Izvršiti optimizaciju svih koraka kroz koje navedeni sistem prolazi. Voditi računa o zauzeću memorijskih resursa i brzini izvršavanja. U izvještaju priložiti rezultate za različite pristupe optimizaciji koda.

Obrazložiti dobijene rezultate.

Prilikom optimizacije detektovati **“bottle neck”** u kodu, tj. koji dio čitavog sistema zahtijeva najviše procesorskih ciklusa i njemu posvetiti posebnu pažnju tokom optimizacije. Izvršiti mjerenje vremena izvršavanja značajnijih funkcija unutar projekta.

Napisani kod treba da bude organizovan i jasan. Voditi računa o imenima varijabli, pisanju komentara, konzistentnosti u notaciji i uopšteno voditi računa da se poštuju principi **“Čistog koda”** detaljnije o principima čistog koda na linku: gist.github.com/wojteku/73c6914cc446146b8b533c0988cf8d29).

Tokom izrade projektnog zadatka **obavezno je** koristiti git sistem verzionisanja. Ovo podrazumijeva izradu **javnog git** repozitorijuma za projekat i ažuriranje svake izmjene napravljene u projektu na taj git repozitorijum. Voditi računa o README.md fajlu na git repozitorijumu, on treba da sadrži osnovne stavke projektnog zadatka (šta se nalazi u projektu, kako se kompajlira, kako se pokreće, koji hardver se koristi, itd). Predaja projektnog zadatka je sada samo obezbjeđivanje pristupa git repozitorijum za vaš projekat tako što ćete obavijestiti zadužene asistente putem mejla (mejl je obavezno poslati **svim** zaduženim asistentima {vedran.jovanovic, damjan.prerad}@etf.unibl.org).

Svatom studentu je dozvoljeno da samostalno proširi obim projektnog zadatka, dodavajući nove zanimljive funkcionalnosti. Svaka dodatna funkcionalnost, ako se ispravno implementira, će biti dodatno bodovana, ali ukupan broj bodova osvojen na projektnom zadatku ne može biti veći od maksimalnog broja bodova predviđenih za projektni zadatak.

Primjeri dodatnih funkcionalnost su:

- implementacija *ping-pong* DMA pristupa SDRAM-u,
- implementacija projekta u skladu sa *MISRA-C:2004* standardom,
- implementacija algoritma u real-time-u, te demonstracija na nekom audio zapisu,
- prikaz progressa obrade korištenjem dioda na ploči (npr. u toku obrade, uključivanjem i isključivanjem dioda vizuelno prikazati postotak obrade, slično loading baru pri instalaciji programa),
- korištenje dodatnih tastera za izbor i kontrolu parametara audio efekta,
- parametarske promjene - real-time podešavanje parametara efekta,
- kaskadno vezivanje efekata - kombinovanje više efekata istovremeno.

Detalji predaje radova: Nakon završetka kompletnog projekta, najkasnije do navedenog roka, predaju se sledeći rezultati rada:

- kod korišten za izradu projektnog zadatka,
- izvještaj, ne duži od 12 stranica napisan prema šablonu priloženom uz tekst projektnog zadatka koji sadrži sve važne detalje rješenja:
 - o detalje realizacije,
 - o tipove korištenih parametara,
 - o dokumentaciju koda,
 - o dobijene rezultate (odstupanja od očekivanih rezultata, brzinu izvršavanja, zauzeće memorije...),
 - o analizu dobrih strana i nedostataka,

- o prijedloge poboljšanja,
- o uputstvo za upotrebu,
- o ovaj dokument treba da bude obrađen na računaru, i predaje se u **.PDF** formatu u sklopu repozitorijuma zajedno sa kodom.

NAPOMENA: U izvještaj se ne stavlja sav izvorni kod, nego samo njegovi isječci koji se odnose na dio koji se objašnjava.

Način rada i ocjenjivanje: Zadaci se rade samostalno. Neki od koraka nisu strogo definisani tako da se studentima ostavlja mogućnost izbora samog rješenja. Pri ocjenjivanju će se gledati razumijevanje urađenog, a na konačnu ocjenu će uticati i obim testiranja same aplikacije, a to podrazumijeva korištenje različitih tehnika rada sa ugrađenim računarskim sistemima koje su obrađene na predavanjima i vježbama (sa naglaskom na tehnike optimizacije i brzinu izvršavanja programa), kao i inicijativa da se dodaju nove funkcionalnosti, ili da se istraže dodatne mogućnosti hardvera u svrhu poboljšavanja performansi rješenja, bilo u pogledu brzine izvršavanja, zauzeća memorije ili nekog trećeg parametra. Konsultacije između studenata su dozvoljene, ali svaki student treba da preda **ORIGINALNO** rješenje. U slučaju prepisivanja odgovarajući radovi će biti ocijenjeni sa nula poena.

Oprema: Za rad na ovom zadatku studentima je na raspolaganju računarska oprema u Laboratoriji za digitalnu obradu signala, kao i razvojno okruženje ADSP-21489 potrebno za izradu projektnog zadatka.

Predmetni nastavnici i asistenti