

Arhitekture i algoritmi DSP 2



Auditorne vežbe AU-7[0]
Ispitivanje rada i verifikacija DSP modela



Odsek za računarsku tehniku i računarske komunikacije

Metodologija razvoja i verifikacija



Model	Radno okruženje	Verifikacija
0	Visual Studio	Subjektivna – Audacity, analiza (mora raditi prema specifikaciji algoritma)
1	Visual Studio	Objektivna – PCMCompare Bitska razlika: 0 bita
2	Visual Studio	Objektivna – PCMCompare Bitska razlika: 1-2 bita
3	CLIDE	Objektivna – PCMCompare Bitska razlika: 1-2 bita (0 bita u odnosu na model 2)
Finalni model	CLIDE/CS497xx	Objektivna razlika: 1-2 bita (0 bita u odnosu na model 3)

Kontrole DSP modula



- ❑ Kontrole DSP modula koje se mogu menjati u realnom vremenu – MCV kontrole
- ❑ U modelima 0, 1, 2 se prosleđuju kao argumenti programa (main funkcije)
- ❑ U modelu 3 se najčešće testiraju samo „zabetonirane“ difolt vrednosti
- ❑ U finalnom modelu se testiraju preko MCV kontrola u realnom vremenu

Kontrole za gainProc primer



- ❑ Enable
 - On – obrada se izvršava
 - Off – obrada se ne izvršava
- ❑ Mute
 - On – tišina na izlazu
 - Off – trenutna obrada na izlazu
- ❑ G1, G2 – promenljivi koeficijenti pojačanja
- ❑ Output mode – izlazni kanali
 - 2_0_0 – u wav fajlu/na izlazu izgenerisani samo L i R kanali
 - 3_0_0 – u wav fajlu/na izlazu izgenerisani L, R i C kanali
 - 1_0_0 – u wav fajlu/na izlazu izgenerisan samo C kanal

control	Enable	G1	G2	Mute	Output Mode
values	On/Off	From 0 to $-\infty$ dB	From 0 to $-\infty$ dB	On/Off	2_0_0, 3_0_0, 1_0_0,
default value	On	minus 4 dB	minus 2 dB	Off	2_0_0

Podešavanja za izlazne kanale (output mode)



- ❑ X_Y_Z
 - X – broj L, R i C kanala
 - Y – broj Ls, Rs kanala
 - Z – broj LFE/SUB kanala
- ❑ U slučaju 2_0_1 očekuje se da se u wav fajlu nalaze 3 izgenerisana kanala: L, R i LFE
- ❑ U slučaju 0_2_0 očekuje se da se u wav fajlu nalaze 2 izgenerisana kanala: Ls i Rs
- ❑ Ostale kanale ne generisati na izlazu (kao tišinu)!
- ❑ Redosled kanala u wav fajlu nije bitan dokle god znate sami koji je koji kanal i dokle god se raspored ne menja od modela do modela

Arhitekture i algoritmi DSP 2



Auditorne vežbe AU-7[1]
Automatizacija ispitivanja



Odsek za računarsku tehniku i računarske komunikacije

Automatizacija ispitivanja



- ❑ Potrebno ispitati svaki model za svaku kombinaciju konfiguracija
- ❑ Za svaku kombinaciju i za svaki model potrebno ispitati više različitih testnih ulaza
- ❑ Veliki broj konfiguracija, veliki broj testova!
- ❑ Kako olakšati testiranje?
- ❑ Automatizacijom – python, batch, bash i drugm skript jezicima.

Način odabira parametara

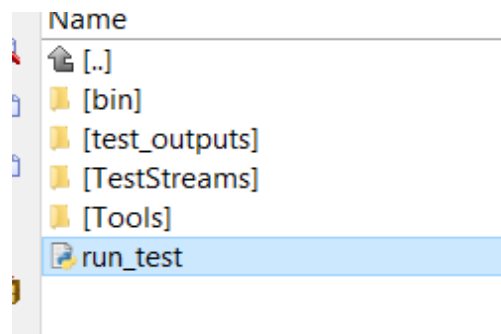


- ☐ Promenljivi gain
- ☐ Testirati za sve moguće vrednosti od -1.000 do 0.999?
- ☐ Ne, testirati difolt i granične slučajeve (bar 2 različite vrednosti po gainu)
- ☐ U modelu 2 se može desiti da se greška javlja samo pri određenim graničnim vrednostima gain-ova, obratiti pažnju na takve slučajeve!
- ☐ Koliko ulaznih strimova? Bar 2 različita strima.

Automatizacija na vežbama



- ❑ Python skripta `run_test.py` sa sledećom strukturom foldera:
 - Bin – izvršni exe fajlovi modela (nalaze se u debug ili release folderu VS projekta)
 - TestStreams – svi ulazni test vektori/streamovi koji se testiraju
 - Tools – izvršni exe fajlovi alata koje skripta koristi (PCMCompare)
 - test_outputs – automatski izgenerisan folder sa izlazima, streamovi i logovi iz PCMCompare-a za svaku konfiguraciju
 - run_test.py sama skripta koja se pokreće i izvršava testiranje



- ❑ Koraci za menjanje run_test.py:
 - 1. podešavanje putanja i imena izvršnih fajlova
 - 2. podešavanje inputa
 - 3. podešavanje parametara

Korak 1 (putanje i imena)



- ❑ Sve putanje relativne u odnosu na run_test.py

```
import os
import subprocess
import itertools
import shutil

# input_streams_root == "/path/to/script/Teststreams"
input_streams_root = os.path.join(os.path.dirname(__file__), "TestStreams")
# base name of project
target = "multitapEcho"
# executables_root = "/path/to/script/Debug"
executables_root = os.path.join(os.path.dirname(__file__), "bin")
# compare_tool = "/path/to/script/Tools/PCMCompare.exe"
compare_tool = os.path.join(os.path.dirname(__file__), "Tools", "PCMCompare.exe")

# name of models for testing == ["model0", "model1", "model2"]
models = [f'model{x}' for x in range(3)]
# *****
# This tool expects the executables provited to be of the following format:
#   executables = [ f'{target}_{m}' for m in models ]
#   i.e. multitapEcho_model0
# *****
```

Korak 2 (ulazni strimovi)



- ❑ Navesti imena ulaznih strimova, moraju se nalaziti na „TestStreams“ putanji prethodno definisanoj

```
|  
# returns list of paths to input streams  
def get_inputs():  
    # Get all .wav files from input_streams_root  
    # inputs = []  
    # for root, dirs, files in os.walk(input_streams_root):  
    #     for file in files:  
    #         if file.endswith(".wav"):  
    #             inputs.append(os.path.join(root, file))  
  
    # inputs == ["/path/to/input_streams/funky_sample.wav", "/path/to/input_streams/WhiteNoise.wav"]  
    inputs = [os.path.join(input_streams_root, x) for x in ['funky_sample.wav', 'WhiteNoise.wav']]  
  
    return inputs
```

Korak 3 (parametri)



- ❑ Podesiti sve moguće vrednosti po jednom parametru, skripta će odraditi kombinaciju svih parametara

```
# returns list of lists of desired params
def get_params():

    params = [
        [0, 1],                # enable
        # [0, 1],              # mute
        # ["0.999", "-0.999"], # gain
        # [0, 1, 2]            # output mode
    ]

    return params
```

Projekat i skripta



- ❑ Skriptu i strukturu foldera koristiti po projektu
- ❑ Za svaki novi projekat prekopirati strukturu foldera
- ❑ I izmeniti skriptu prema novom projektu