

Verslag spectrogram

Academiejaar 2019 – 2020

Robin Nollet, Sebastian Vantomme, Ine Vanderhaeghe

Verslag: bouwen van een spectrogram op een FPGA met behulp van de VHDL-taal

Inhoudsopgave

1	Probleemstelling: een audiospectrogram	3
1.1	Wat is een spectrogram?	3
1.2	Hoe wordt een spectrogram voorgesteld?	3
1.3	Gebruikte methoden om een spectrogram te implementeren	4
1.4	Afwerkingsgraad	4
2	Oplossingsstructuur	4
2.1	Audio binnentrekken	4
2.2	Fourier nemen	4
2.3	Grafisch weergeven	4
3	Implementatie	4
3.1	Audio binnentrekken	4
3.2	Fourier nemen	5
3.3	Grafisch weergeven	5
4	Besluit	5

1 Probleemstelling: een audiospectrogram

Doorheen het semester hebben we ons met ons groepje bezig gehouden met het bouwen van een audiospectrogram.

1.1 Wat is een spectrogram?

Het kan handig zijn om snel de frequentieinhoud van een signaal te zien, naar analyse toe. Gezien een signaal ook vaak veranderlijk is in de tijd, kan ook de frequentieinhoud veranderen in functie van de tijd. Er bestaat een methode om de frequentieinhoud van een signaal weer te geven in functie van de tijd. Dit heet een *Spectrogram*. Een spectrogram kan goed samengevat worden volgens de volgende definitie: het is een visuele representatie van de energie in elke frequentie uitgezet in de tijd. Een spectrogram heeft verschillende toepassingen. Enkele voorbeelden hiervan is bv. in de audio, of in de analyse van licht. Dit project focust specifiek op het analyseren van audiosignalen, hoorbaar voor het menselijk oor. Hiervoor nemen we standaard een range van 20Hz tot 20kHz . In de audiowereld worden deze frequenties vaak de *pitch* genoemd van het signaal.

Mogelijke toepassingen van een spectrogram die hoorbare audiosignalen worden hieronder opgesomd:

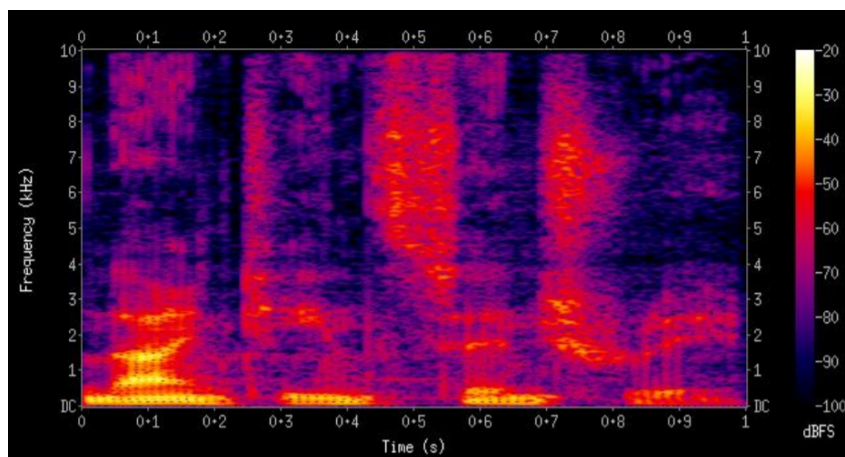
- analyse van Muziek(-instrumenten)
- analyse van Spraak
- analyse van Dierengeluiden
- analyse van RF modulatie technieken

1.2 Hoe wordt een spectrogram voorgesteld?

De grafische voorstelling van een spectrogram zou eigenlijk in 3 dimensies moeten gegeven worden. Dit is echter wat lastig om dan op één oogopslag informatie te kunnen uit aflezen. Daarom worden in de literatuur vaak de volgende afspraken gevolgd:

- De tijd wordt op de horizontale as uitgezet;
- De frequentie wordt op de verticale as uitgezet;
- De sterkte van de frequentie wordt vaak weergegeven met behulp van kleurintensiteit of een kleurgradiënt.

In figuur 1 is een typische grafische voorstelling gegeven.



Figuur 1: Een typische grafische voorstelling van een spectrogram

1.3 Gebruikte methoden om een spectrogram te implementeren

In de literatuur kunnen 2 veelgebruikte methoden voor de implementatie van een Spectrogram gevonden worden.

1. Met behulp van de FFT: van het ingenomen signaal wordt de *Fast Fourier Transform (FFT)* genomen, wat een versnelde versie is van de *Discrete Fourier Transform (DFT)*. Dit toont de frequentieinhoud van het signaal. Een belangrijke eigenschap van deze transformatie is dat ze een discrete ingang (de audio) en een discrete uitgang (de frequentieinhoud van dit signaal) heeft.
2. Met behulp van een filterbank: hier wordt een reeks banddoorlaatfilters geïmplementeerd, die allemaal op hetzelfde signaal werken. Door na iedere filter dan te gaan kijken wat de energie van het signaal is, kan dit als een maat gebruikt worden voor de frequentieinhoud van ons ingangssignaal.

Naar implementatie toe heeft men dan een keuze, dit hangt af welke resolutie men wilt in de frequentie. Wil men een grote resolutie in de frequentie, gaat men eerder kiezen voor de aanpak met de FFT. Als men met een lage frequentieresolutie zich tevreden kan nemen, wordt er vaak geopteerd voor de implementatie met die meerdere filters.

Wij hebben hier gekozen voor de implementatie met een FFT, omdat deze een stuk flexibeler is in implementatie.

1.4 Afwerkingsgraad

2 Oplossingsstructuur

2.1 Audio binnentrekken

Om de audio binnen te lezen, zijn er 2 belangrijke bestanden: *adau1761* en *audio_if*. Deze bestanden lezen de data binnen en genereren daar de juiste clock signalen voor. In het top level wordt daar dan een audio loopback van gemaakt om de data die binnengelezen werd ook weer naar buiten te sturen. De data die binnengelezen werd wordt ook naar een FIFO gestuurd, zodat deze op de juiste momenten binnengelezen kan worden in de fft controller.

2.2 Fourier nemen

2.3 Grafisch weergeven

3 Implementatie

3.1 Audio binnentrekken

Er kan gebruik gemaakt worden van de audiocodec die aanwezig is op het Zedboard. Dit is de *adau1761*. Hiervoor moest een driver geschreven worden. Voor ons project hebben we een driver kunnen gebruiken die we gekregen hadden van Dhr. Verslype. Deze driver krijgt zowel serieel als parallelle data binnen. De samples van het linker en rechter kanaal zijn 24 bit breed. Deze worden verwerkt in de driver en er wordt een linker en rechter output gegenereerd die 24 bit breed zijn en een seriële data output.

Daarnaast moest een interface geschreven worden die de clock genereert en de data die de driver binnenleest op een juiste manier door geeft naar het top level. Ook hier hebben we gebruik gemaakt van een interface die we gekregen hadden van Dhr. Verslype. Deze interface heeft een kloksignaal nodig van 100 MHz en 12,288 MHz. Deze genereren 3 nieuwe kloksignalen: de *master clock*, de *digital audio bit clock* en de *digital audio left-right clock*. Deze worden gebruikt in de *adau1761* codec.

De configuratie van de audio interface werkt met het I^2C protocol, het doorgeven van audio data gebeurt via het I^2S protocol.

In het top level wordt daarna een audio loopback gemaakt. De data wordt ook doorgegeven aan een FIFO memory, zodat deze uitgelezen kan worden door de fft controller. De data wordt serieel doorgegeven,

eerst moet deze dus geparalleliseerd worden en per 24 bits opgeslagen worden in de memory. Het memory is 2048 waarden groot, dus wanneer deze vol is wordt een flag hoog gezet zodat de fft controller weet dat alle waarden die in de memory staan geldig zijn.

3.2 Fourier nemen

3.3 Grafisch weergeven

4 Besluit

Afsluitende tekst.