AP HOGESCHOOL ANTWERPEN

AP.BE

Audio Interface

Digital System Development

Opleiding: Bachelor Elektronica ICT – Deeltraject IoT

Academiejaar: 2022-2023

Jelte Boumans

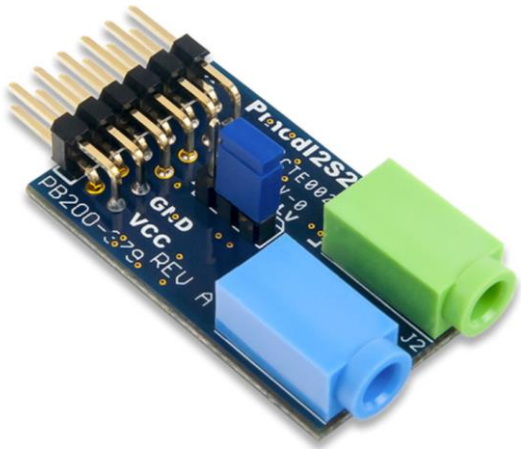
Inhoud

1 Doel

Het doel van deze opdracht is om een project te maken met het FPGA development bordje genaamd “Basys 3 Artix-7”. Enkele voorbeelden van zulke projecten zijn: video DSP met FPGA, audio-interface met FPGA, project met MicroBlaze op een FPGA en je mag ook een eigen uitdagend project verzinnen.

Ik koos er uiteindelijk voor om een soort audio-interface te ontwerpen op een FPGA. Mijn doel is om een systeem te ontwerpen dat via een audio jack break-out board stereo audio data ontvangt. Deze samples gaan op 2 manieren kunnen worden aangepast:

- Het volume/gain zal kunnen aangepast worden via 2 on-board drukknoppen. Het volumeniveau zal getoond worden via de on-board 7 segment display.
- Het output signaal zal gefilterd kunnen worden door 3 verschillende FIR filters: een laagdoorlaatfilter, hoogdoorlaatfilter en banddoorlaatfilter. Deze filters zullen de “bass”, “midrange” en “treble” respectievelijk scheiden van elkaar. Dit zal worden bestuurd met 3 on-board schakelaars om een filter te selecteren.



Audio break-out bordje

2 Project

In dit hoofdstuk van het verslag ga ik alle verschillende onderdelen van het systeem bespreken. Alle verschillende onderdelen zijn te vinden in de blokdiagrammen in het volgende hoofdstuk.

2.1 Audio

Zoals eerder vermeld zullen de audio samples binnen komen via het stereo break-out bordje. Deze worden dan doorgestuurd via het I2S protocol. De blocks die dit deel regelen heb ik niet zelf ontworpen, dus daarovert kan ik zelf niet te veel uitleg geven. Ik heb ze enkel wat aangepast om ze te laten werken met mijn blocks.

De block waar ik het over heb is i2s_rxtx, deze bevindt zich in de txrx_system block. Deze block zorgt voor 24-bit audio samples met een samplefrequentie van 96kHz. Deze samples komen uit de pins out_l en out_r. Als je deze pins rechtstreeks verbindt met de in_l en in_r pins zal de audio rechtstreeks naar de output gaan. Dus door de samples uit de pins out_l en out_r aan te passen kan ik effecten op de audio toepassen.

2.2 Volume

In mijn volume regeling is het alleen maar mogelijk om de audio stiller te maken relatief met het input audio signaal. Het versterken is niet mogelijk zonder de audio helemaal te vervormen en onbeluisterbaar te maken (door “digital clipping”). Er zijn 2 componenten die zorgen voor het regelen van het volume.

2.2.1 Volume controller

Deze component bestaat uit 3 verschillende blocks: een clock divider, een binaire teller en een encoder voor de 7 segment display. Als inputs neemt het de 2 drukknoppen en het 100MHz kloksignaal binnen.

Het kloksignaal zal worden gedeeld naar een +-4Hz en naar de teller gaan. De teller zal dan omhoog tellen als de “increase” knop wordt ingedrukt terwijl er een stijgende flank is van het trage kloksignaal, en vice versa met de “decrease” knop. Deze teller kan tellen van 0 tot en met 10 en zal niet overflowen.

De output van de teller (vol_data) gaat dan naar een encoder. Deze encoder zal de inkomende binaire data omzetten naar het juiste 8-bit signaal om het geselecteerde volumeniveau te tonen op de 7 segment display. De output van de teller gaat ook rechtstreeks naar de uitgang van deze block en rechtstreeks naar de audio_system block. Daarin bevindt zich de volgende block.

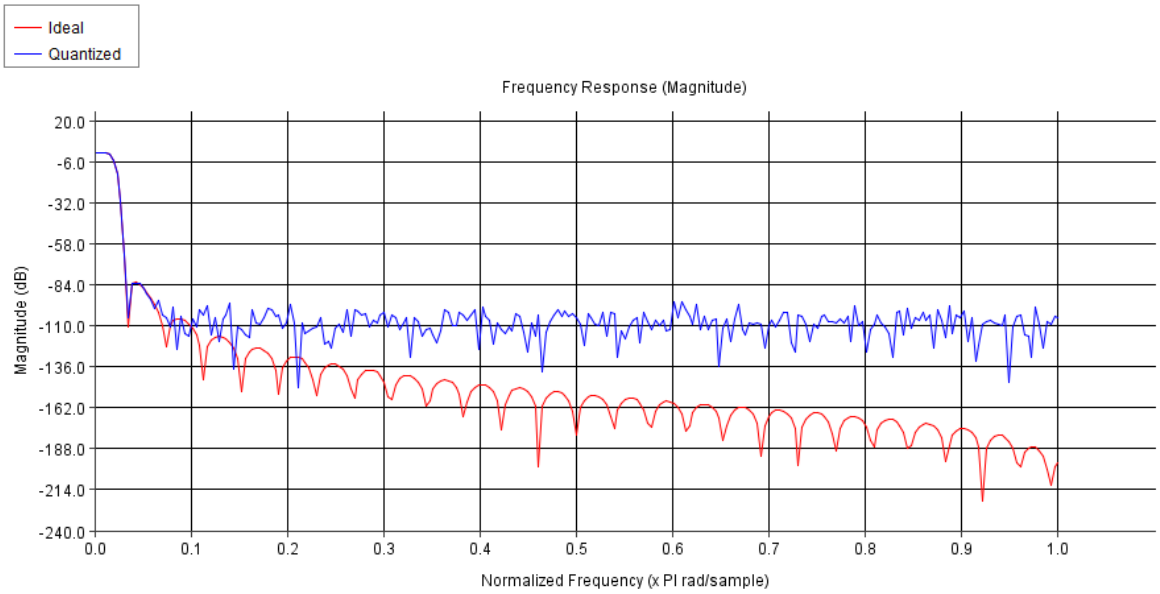
2.2.2 Volume divider

Deze block bevindt zich in de audio_system block. Er zijn er ook 2 van, 1 voor elk audio channel (L en R). Deze block is zeer simpel, het krijgt de data vanuit de teller en de audio samples van de I2S block. Met deze data kan je het volume regelen met een simpele formule: out = in * (volume/10). Hierdoor als bijvoorbeeld de teller op 4 staat, zal deze block het zelfde audio signaal uitsturen, alleen met maar 40% volume. Dit output signaal gaat dan rechtsreeks naar de volgende block, de filters.

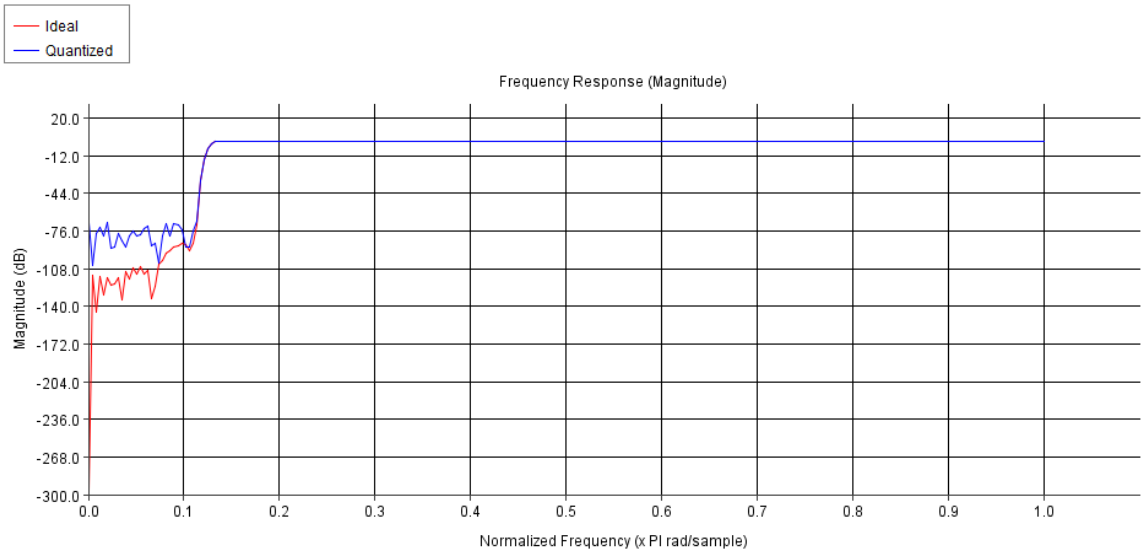
2.3 Filter system

Deze block bevindt zich ook in de audio_system block. Er zijn er ook 2 van, 1 voor elk audio channel (L en R). Het bestaat uit 4 verschillende componenten: een laagdoorlaatfilter, hoogdoorlaatfilter, banddoorlaatfilter en een multiplexer. De werking van de filters kunnen getest worden door een “Online Tone Generator”.

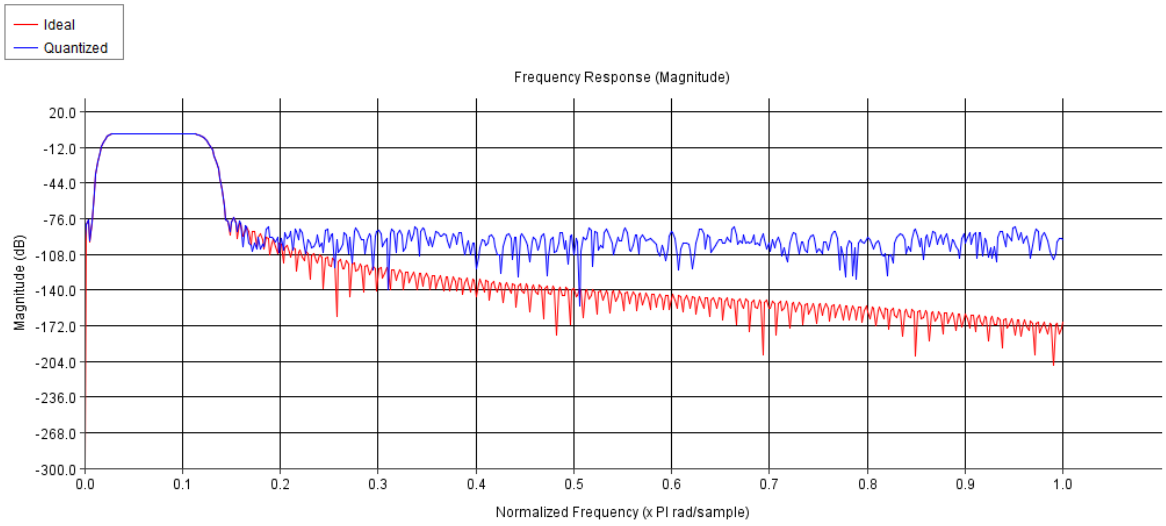
De laagdoorlaatfilter laat alleen de “bass” door:



De hoogdoorlaatfilter laat alleen de “treble” door:



Tenslotte laat de banddoorlaatfilter alleen de “midrange” door:

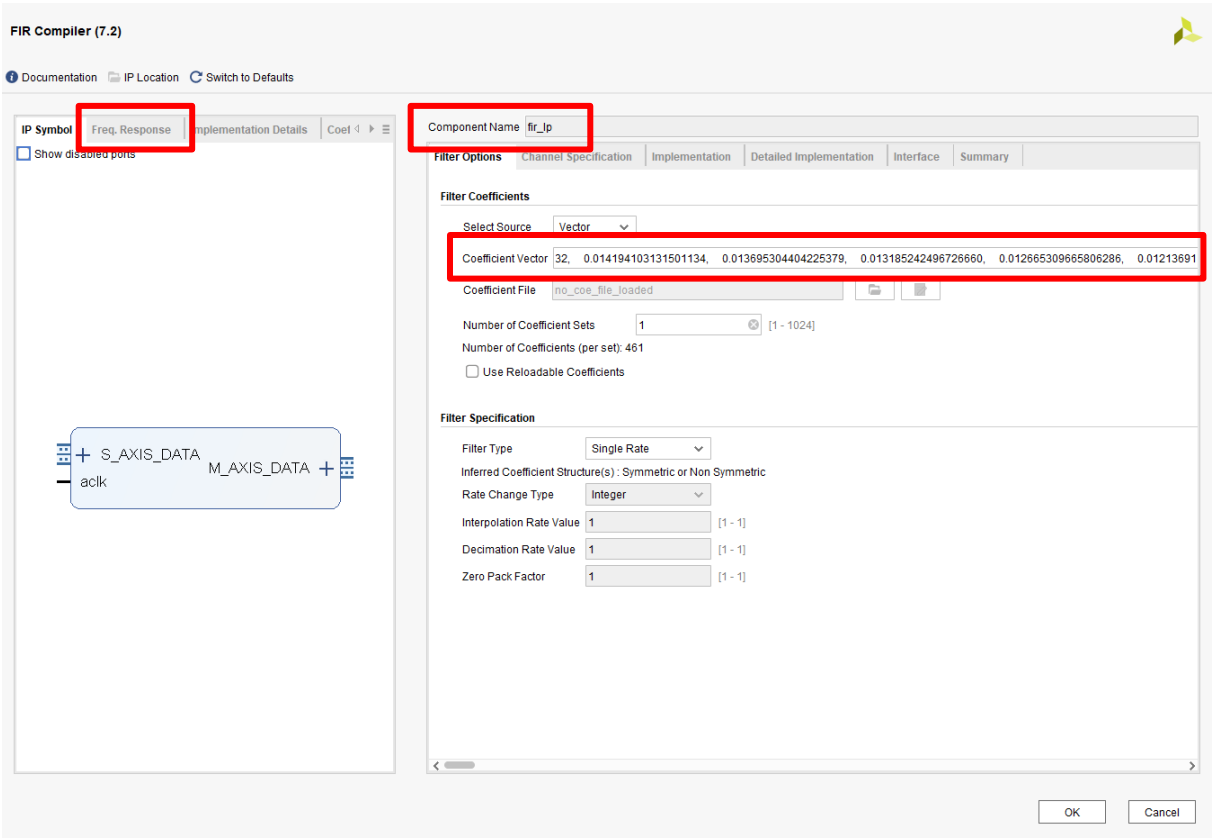


De filters hebben een sample frequentie van 96kHz (hetzelfde als de I2S block). Dus 1.0 op de X-as is gelijk aan 96kHz en dan is 0.5 gelijk aan de helft van 96kHz. De waardes voor de cutoff frequenties heb ik bepaald door deze website. De nodige waardes van de taps heb ik berekend via deze website.

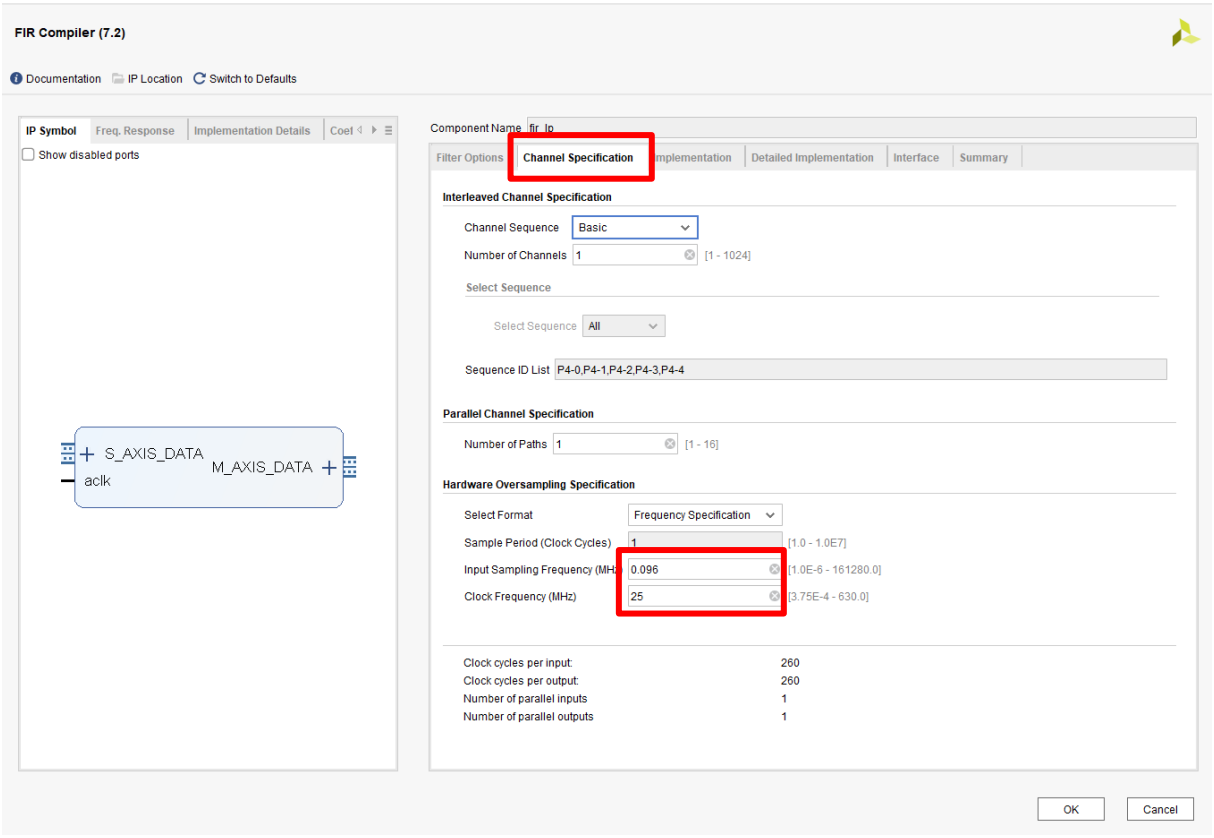
Elke output van de filters is verbonden aan een multiplexer. Deze multiplexer wordt bestuurd door 3 on-board schakelaars. Enkel wanneer maar 1 van de 3 schakelaars aan is zal de bijhorende filter aan de output worden aangesloten. Als er meer dan 1 of geen schakelaars aan zijn gaat de audio langs geen enkele filter.

2.4 Filters instellen

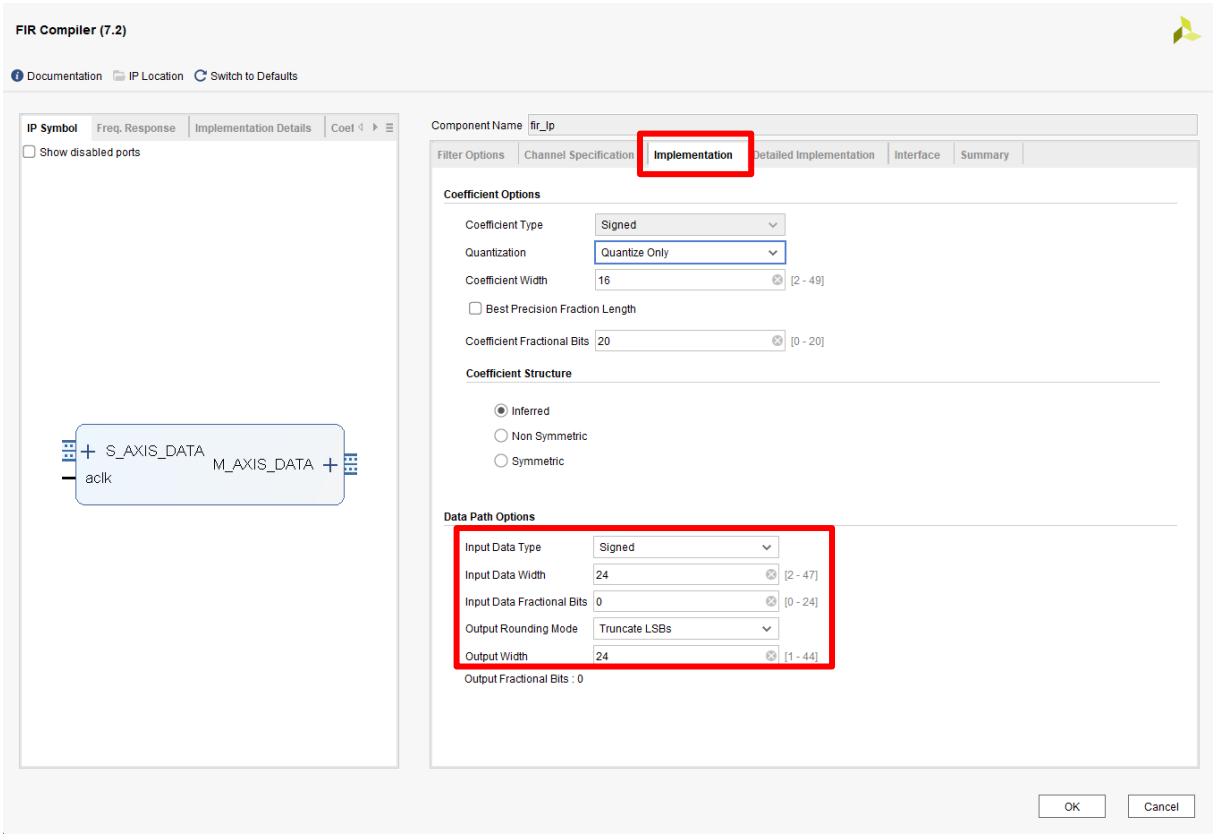
De filters kunnen natuurlijk niet zo maar werken, enkele instellingen moeten juist geconfigureerd worden. Een FIR filter kan niet werken zonder coëfficiënten, aangezien de filters honderden coëfficiënten kunnen hebben is deze zelf berekenen is een nutteloze inspanning. Maar dit is geen probleem voor ons aangezien er veel websites zijn die deze waardes kunnen berekenen voor ons in een aantal seconden. Dit is de beste website die ik kon vinden: FIIIR.com.



Hier zie je de instellingen van een FIR filter IP. Het eerste dat je kan doen is de naam aanpassen (indien nodig) naar iets meer passend. Daarna kan je al jouw coefficienten copy-pasten in het “Coefficient Vector” vlak. Je kan dan ook links boven op “Freq. Response” drukken om de bode diagram te zien (zoals op pagina 5 en 6).



Er moeten ook enkele aanpassingen gemaakt worden in een andere tab, de “Channel Specification”. Hier moet je de sample frequentie (in MHz) specificeren en ook de frequentie van het kloksignaal dat via de “aclk” pin zal worden aangesloten.

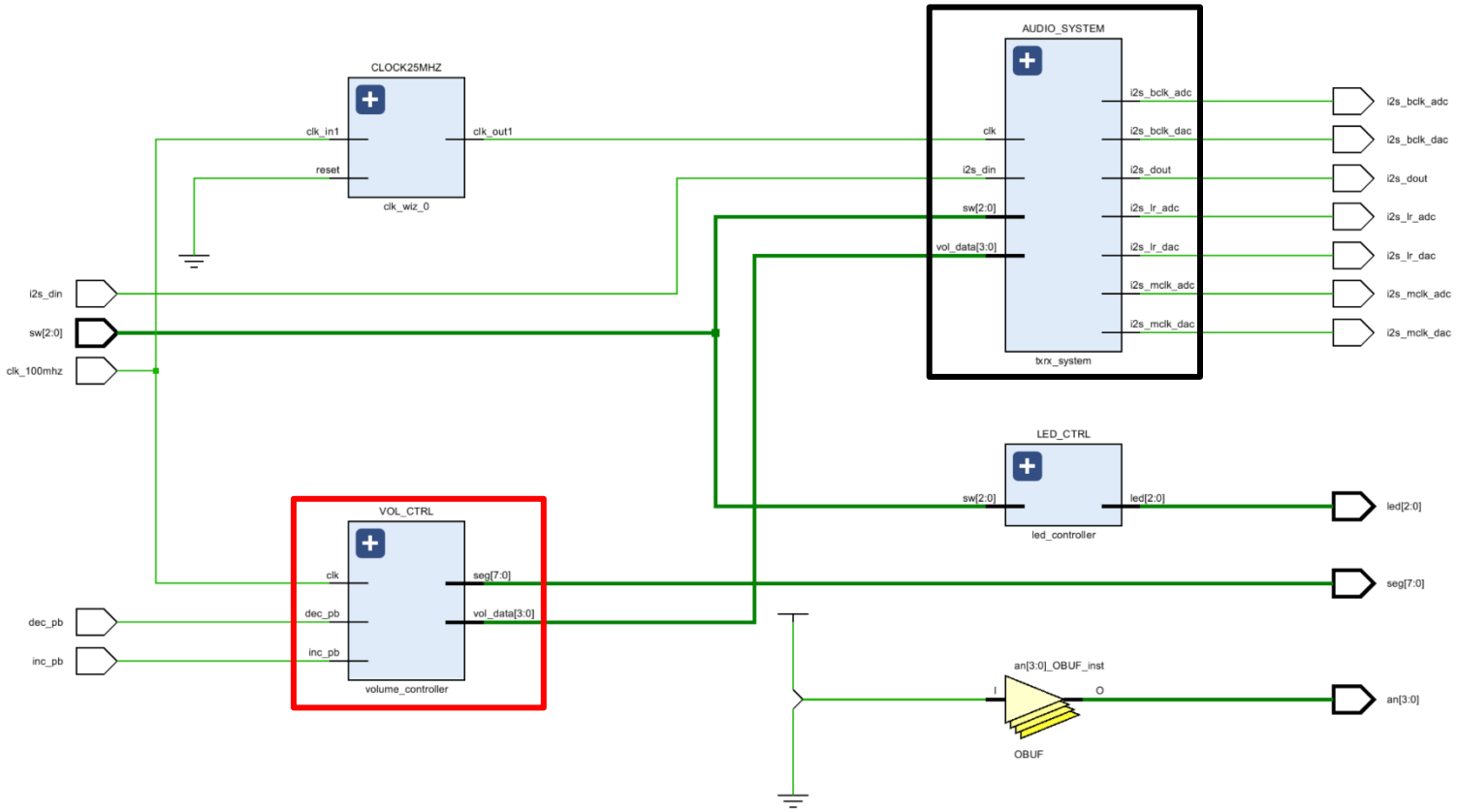


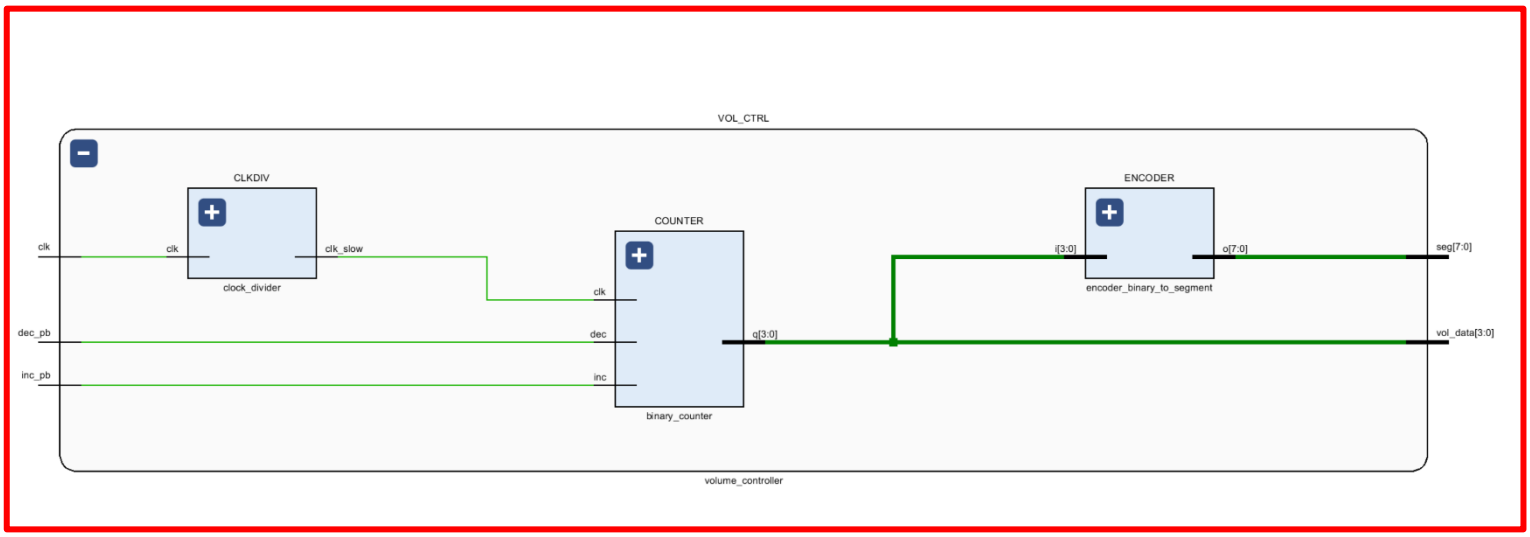
Tenslotte moeten er nog aanpassing gemaakt worden in de tab “Implementation”. Eerst ga je bij “Input Data Width” specificeren hoe veel bits jouw input signaal heeft. Deze volgende stap is niet nodig als het aantal bits van jouw output signaal niet uitmaakt. Stel “Output Rounding Mode” in op “Truncate LSBs”. Hierdoor kan je de grootte van je output signaal aanpassen naar wat de gebruiker wenst (in dit geval moest het 24 zijn, net zoals de input). Let er ook op dat “Input Data Type” staat op “Signed” indien het input signaal ook negatieve samples heeft.

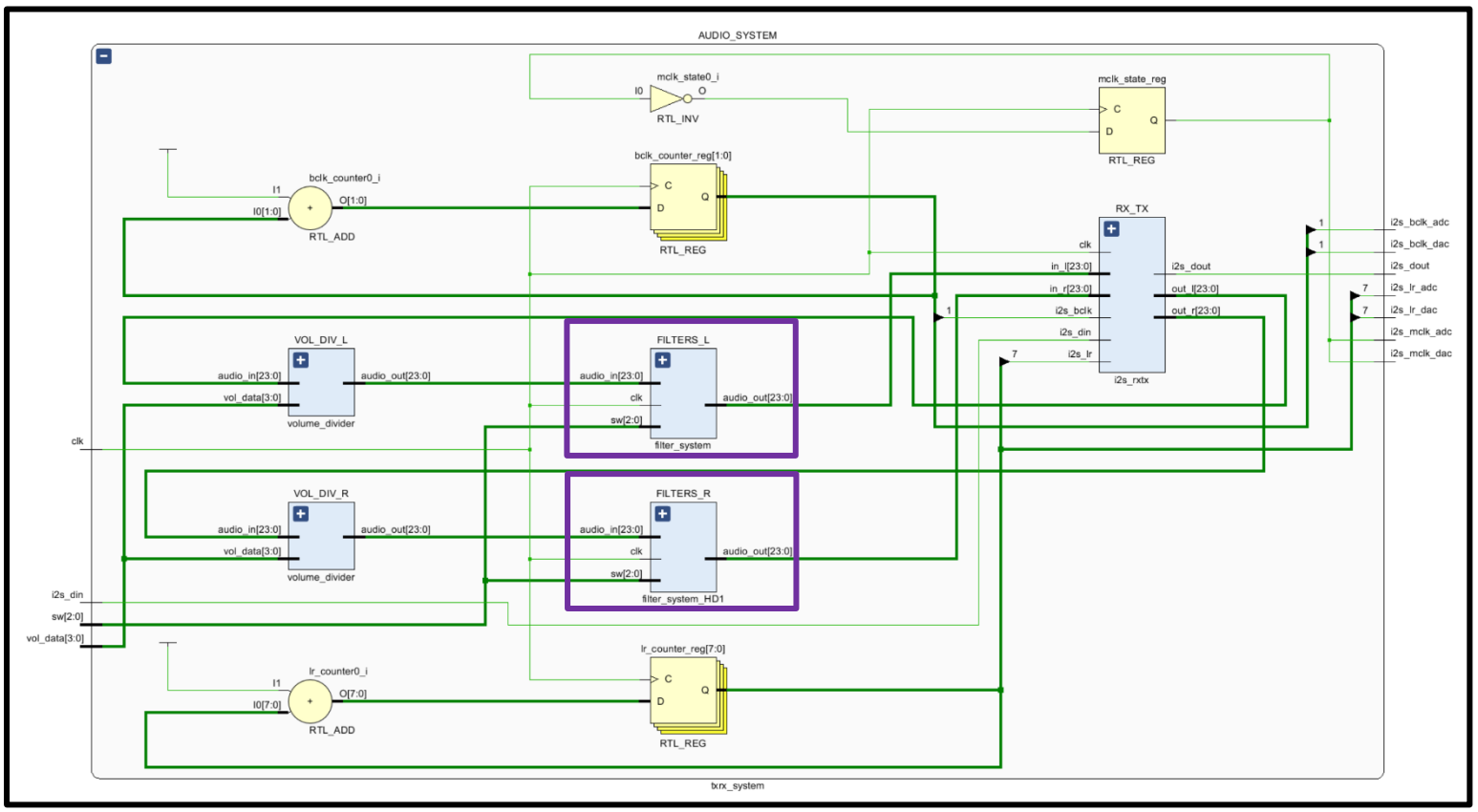
2.5 LED controller

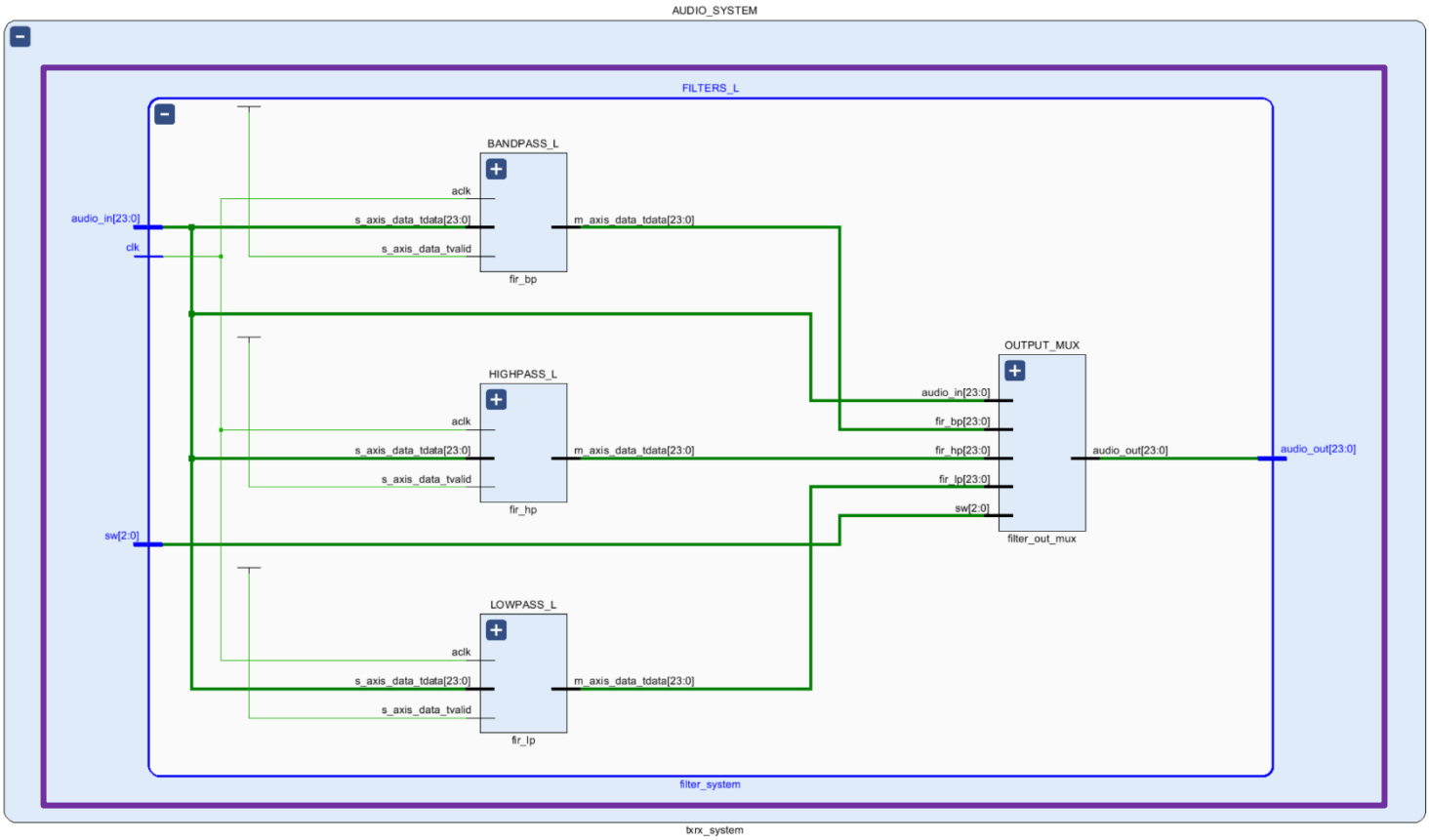
Deze block is zeer simpel, het bestuurt de on-board LEDs boven de on-board schakelaars. Wanneer er 1 schakelaar aan wordt gezet dan zal de bijhorende LED aan gaan. Wanneer er meerdere of geen schakelaars aan zijn zullen er geen LEDs aan staan. Dit zorgt voor een goede visualisatie over welke filter nu actief is.

3 Blokdiagrammen






AUDIO_SYSTEM
mclk_state0_i
RTL_INV
mclk_state_reg
RTL_REG
bclk_counter0_i
RTL_ADD
bclk_counter_reg[1:0]
RTL_REG
RX_TX
i2s_rxtx
i2s_bclk_adc
i2s_bclk_dac
i2s_dout
i2s_lr_adc
i2s_lr_dac
i2s_mclk_adc
i2s_mclk_dac
VOL_DIV_L
volume_divider
FILTERS_L
filter_system
VOL_DIV_R
volume_divider
FILTERS_R
filter_system_HD1
clk
i2s_din
sw[2:0]
vol_data[3:0]
lr_counter0_i
RTL_ADD
lr_counter_reg[7:0]
RTL_REG
txrx_system


AUDIO_SYSTEM
FILTERS_L
BANDPASS_L
fir_bp
HIGHPASS_L
fir_hp
LOWPASS_L
fir_lp
OUTPUT_MUX
filter_out_mux
filter_system
audio_in[23:0]
clk
sw[2:0]
audio_out[23:0]
txrx_system

4 Resultaat

Het resultaat is redelijk goed naar mijn mening. Er zijn wel enkele aspecten van het systeem die kunnen verbeterd worden. Door een afwezigheid van meerdere weken heb ik niet alles perfect kunnen realiseren.

4.1 Mogelijke verbeteringen

Het regelen van het volume gaat zeer goed, er is wel een nadeel. Als je de knoppen ingedrukt houdt is er geen probleem, maar als je 1 niveau omhoog wil gaan is het niet even gemakkelijk. Dit komt omdat de knop moet ingedrukt zijn als er een stijgende flank is van het kloksignaal. Dus als je de knop indrukt en loslaat terwijl er geen stijgende flank is geweest van de 4Hz klok dan zal er niets gebeuren.

De filters hebben ook wat aspecten die verbeterd moeten worden. De laagdoorlaatfilter laat nog net iets te veel door in de “midrange”, hier kon ik geen oplossing voor vinden door gebrek aan tijd. De hoogdoorlaatfilter en bandpasfilter hebben ook moeilijkheden wanneer het audiosignaal een hoge frequentie heeft. De hoogdoorlaatfilter spert perfect de laag frequentie signalen maar de hoge frequentie signalen klinken vervormd. De bandpass filter heeft hetzelfde probleem maar dan omgekeerd. De hoge frequentie signalen worden niet perfect gesperd en je hoort ze nog zeer licht, maar ook vervormd.

4.2 Moeilijkheden

Er waren naar mijn mening 2 moeilijkheden in dit project:

- De I2S block was nogal moeilijk te begrijpen omdat het een onbekend concept is voor mij. Het feit dat het door iemand anders was gemaakt maakt het ook niet altijd gemakklijk om de code te begrijpen na het lezen. Hierdoor was deze aanpassen om de samples te kunnen manipuleren niet altijd zo gemakklijk, maar een goede uitdaging.
- De FIR filters waren ook een grote uitdaging voor mij. Ik vond al dat DSP geen gemakkelijk veld was, dus 3 verschillende FIR filters toepassen was zeker niet simpel. Het correct instellen van deze filters in de IP catalogue was ook niet simpel in het begin. Hiervoor moest ik wat onderzoek doen op het internet om bepaalde problemen op te lossen. Zoals bijvoorbeeld de grootte van het output signaal aanpasbaar maken. Maar door dit project is mijn begrip van FIR filters verbeterd.

4.3 Mogelijke uitbreiding

Enkele uitbreidingen die bijvoorbeeld een volgende leerling zou kunnen doen is:

- Het implementeren van IIR filters en de gebruiker de keuze geven om beide IIR en FIR filters te gebruiken om het verschil tussen de 2 duidelijk te kunnen zien.
- De manier waarop volume aangepast wordt vernieuwen. Mijn eerste idee was met een potentiometer en ADC, maar hier had ik niet genoeg tijd voor om te realiseren. Dus dit zou perfect zijn als uitbreiding.