

# Qwerty- orgel

door:  
**Tim Langens,  
Maarten Van Genechten.**

---

<b>1</b>	<b>Inhoudsopgave</b>	
1	Inhoudsopgave .....	2
2	Omschrijving.....	3
3	De AC'97 standalone codec.....	4
4	De sinusgenerator .....	5
5	Toetsenbordmultiplexer .....	6
5.1	Multiplexer.....	6
5.2	Frequentiedeler .....	6
5.3	Toetsenbordcodec .....	6
6	Computersimulaties.....	7
6.1	Sinusgenerator .....	7
6.2	Multiplexer.....	7
6.3	Frequentiedeler .....	7
6.4	AC97-standalone .....	7
7	Appendix.....	9
A	Bijlagen.....	9
A.1	Documentatie .....	9
A.2	VHDL-Codes en ucf-bestanden.....	9

## 2 Omschrijving

Het orgel wordt geïmplementeerd op een Xilinx XUPV2P bord. Het orgel doorloopt volgende stappen:

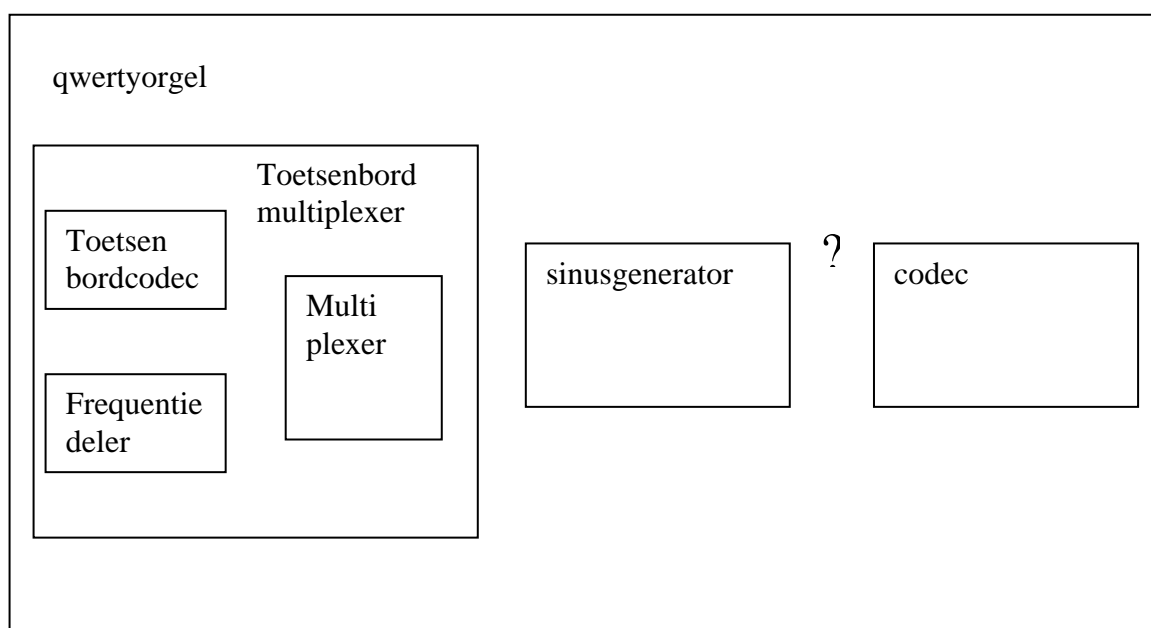
- Een toets wordt aangeslagen,
- Een bepaalde frequentie stemt hiermee overeen,
- Er wordt een digitale sinus met bepaalde frequentie gegenereerd,
- Deze sinus stuurt een PCM-audio-sigitaal,
- Het PCM-sigitaal wordt via een codec naar de AC'97 output gestuurd.

Dankzij meneer Sleutel hebben we een bestaande sinusgenerator en een werkende AC'97 standalone codec gebruikt. Via blackboard hebben we de toetsenbordaansturing. Van deze componenten moet de werking begrepen worden.

Zelf moet er een multiplexer en frequentiedeler aangemaakt worden.

Ook moet er uitgezocht worden hoe het PCM-sigitaal is opgebouwd. Zodat indien nodig een tussenvorm moet worden aangemaakt om de sinus naar een PCM-sigitaal om te vormen.

De verschillende componenten worden top-down besproken.



**Figuur 1: schema van de verschillende componenten**

### 3 De AC'97 standalone codec

Normaal gezien wordt het geluid afgespeeld via de processor die aanwezig is op het XUP-bord. Aangezien de processor niet nodig is wordt er gebruik gemaakt van een standalone codec. De codec, die wij gebruiken, opb\_ac97\_v2\_00\_a, is fysisch niet meer dan een link van de audio-in poort naar de audio-out poort. Op de in-poort komt een digitaal PCM-sigitaal binnen en dit wordt via een vertraging doorgelinkt naar de uitgang.

De codec moet dus worden aangepast zodat deze niet het ingangssignaal van de audio-in poort gebruikt maar het sinussignaal dat aangemaakt wordt door de sinusgenerator. De codec wordt ook ineens gebruikt als bovenste element om alles aan elkaar te linken.

Bij de simulatieresultaten kan je zien dat er daadwerkelijk een sinus naar de pcm-in poort van de AC'97-codec wordt gestuurd. Indien dit geïmplementeerd wordt moet men de enable van de standalone component hoog houden. Wanneer de frequentie van de sinus te laag ligt zal je elke klokpuls een tik horen die qua volume een sinus vormt. De frequentie van de sinus is te hoog wanneer de frequentie boven de gehoorsfrequentie komt.

Om het volume te regelen moeten we de opbouw van een PCM-sigitaal kennen. Een PCM-sigitaal is opgebouwd uit een 2's complement. Het sigitaal wordt opgebouwd door op een bepaald tijdstip een sample te nemen van de golf. Het standaardsigitaal waarmee audio gedigitaliseerd wordt is 44KHz. De voorstelling in VHDL gebeurt door middel van een `std_logic_vector(15 downto 0)`. Door deze om te zetten naar een `bit_vector` en dan de vector te verschuiven naar links of rechts. Wordt het geluid versterkt. Deze verschuiving wordt gerealiseerd met het commando `SLA/SRA(shift left/right arithmetic)`.

## 4 De sinusgenerator

De sinusgenerator maakt gebruik van een apart bestand met daarin een tabel. Deze tabel bevat 128 waarden die samen een digitale sinusgolf genereren. Daardoor zal de klokfrequentie die het element aanstuurt 256x groter moeten zijn dan de te bekomen frequentie van de sinus. Met andere woorden zullen we om een sinus met frequentie 1KHz te bekomen de ingang met een klok van 256 KHz moeten sturen.

## 5 Toetsenbordmultiplexer

De toetsenbordmultiplexer voegt drie componenten samen:

1. multiplexer
2. frequentiedeler
3. toetsenbordcodec

Het ingangssignaal van de component is een klok. Deze klok is de standaardklok aanwezig op het XUP-bord, maw. een klok van 100MHz. Het uitgangssignaal van deze component is een klok met frequentie afhankelijk van de toets die is ingedrukt.

### 5.1 Multiplexer

De multiplexer krijgt 8 verschillende frequenties binnen en een 8-bit signaal binnen die de ascii-code voorstelt van het toetsenbord. De multiplexer heeft als uitgang een klok met één van de inkomende frequenties.

### 5.2 Frequentiedeler

De frequentiedeler deelt de inkomende klokfrequentie van 100 MHz. En geeft de ingangsfrequentie samen met de 7 volgende frequenties via een bus door. Dit omdat er een sinus gaat worden gegenereerd die een hoorbare frequentie heeft. De laatste frequentie is net te laag om een degelijk hoorbare sinus te genereren. Het delen van de frequentie gebeurt met behulp van JK-flipflops.

### 5.3 Toetsenbordcodec

Deze codec stuurt het toetsenbord aan en zorgt ervoor dat er een ascii-waarde wordt doorgegeven telkens een toets ingedrukt en losgelaten wordt. Hiervoor zijn 2 kloksignalen nodig waarvan één 25 MHz signaal. aan welke pinnen de data- en klokingangen moeten worden gehangen staat in de documentatie.

De reset van het toetsenbord kan aan dezelfde pin bevestigd worden als de reset van de AC'97 codec. De readingang kan de vaste waarde nul gegeven worden. De scan-ready uitgang mag open gelaten worden.

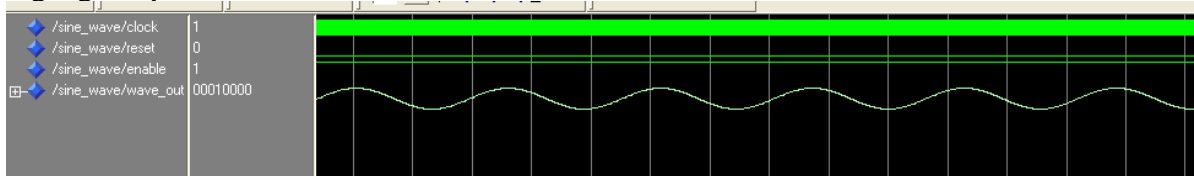
## 6 Computersimulaties

Ook het simuleren van de toetsenbordcodec is complex. Hierdoor zullen de componenten waarvoor deze nodig is moeilijk te simuleren zijn.

Wel zijn er resultaten van de overige componenten.

### 6.1 Sinusgenerator

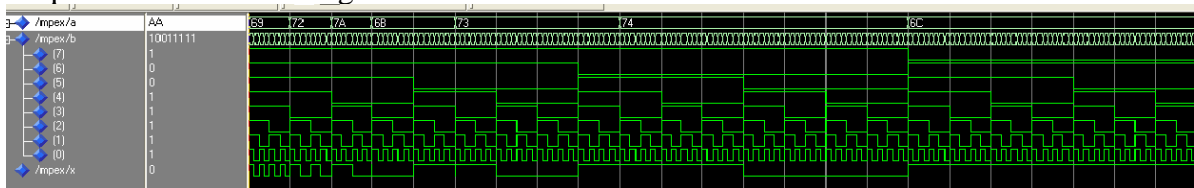
Om de sinusgenerator te simuleren wordt het uitgangssignaal, `wave_out`, voorgesteld als een analogoog signaal, hierdoor kan je zien dat dit signaal een sinusgolf is met als frequentie de ingangsfrequentie 128x kleiner.



Figuur 2: De sinusgolf

### 6.2 Multiplexer

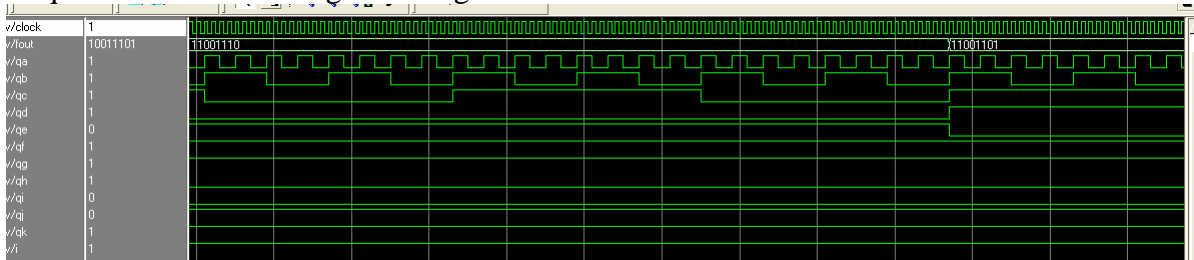
De multiplexer wordt gesimuleerd door aan bus A de gewenste keyboardcode te hangen en door aan elke lijn van bus B een klok toe te wijzen met verschillende frequentie. Deze frequentie is steeds 2x zo groot.



Figuur 3: De multiplexer

### 6.3 Frequentiedeler

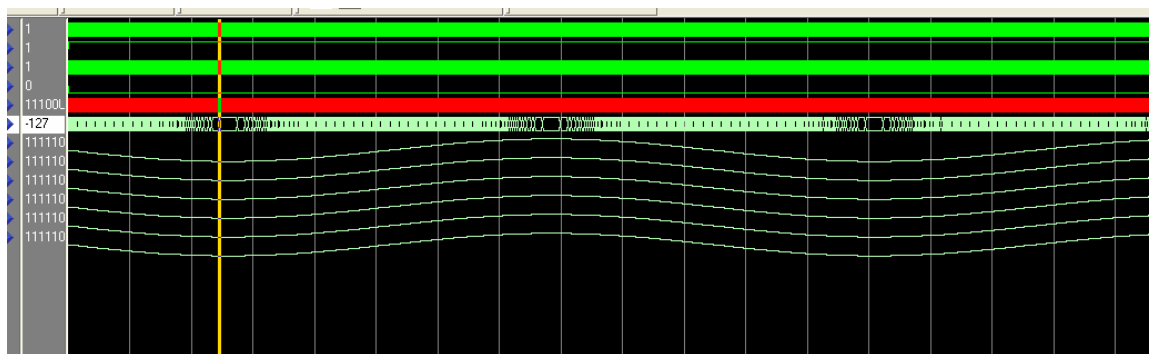
Door aan de ingang van de frequentiedeler een 100MHz klok toe te wijzen kan de frequentiedeler eenvoudig worden gesimuleerd.



Figuur 4: De frequentiedeler

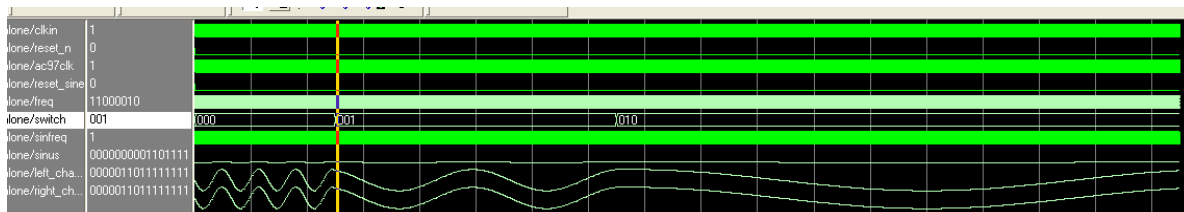
### 6.4 AC97-standalone

De bedoeling is dat de sinus in de ac97-standalone wordt doorgestuurd naar de uitgangskanalen. Dit kan je zien op Figuur 5



**Figuur 5: De sinussen komen aan bij het uitgangssignaal.**

Na toevoeging van een multiplexer, kan je zien dat de sinus varieert van frequentie, naar gelang welke frequentie gekozen wordt.



**Figuur 6: De sinusfrequentie varieert**

## 7 Appendix

### A Bijlagen

#### A.1 Documentatie

- XUPV2P\_User\_Guide.pdf
- IP module toetsenbord.doc

#### A.2 VHDL-Codes en ucf-bestanden

- standalone.vhd
  - ac97\_if.vhd
    - ac97\_command\_rom.vhd
    - ac97\_core.vhd
    - ac97\_if\_pkg.vhd
    - ac97\_timing.vhd
  - sinusgenerator.vhd
    - sine\_package.vhd
    - sine\_wave\_cf.vhd
  - multiplexer.vhd
  - Keyboard.vhd
  - Freqdiv.vhd
    - Jkff.vhd
  - Standalone.ucf