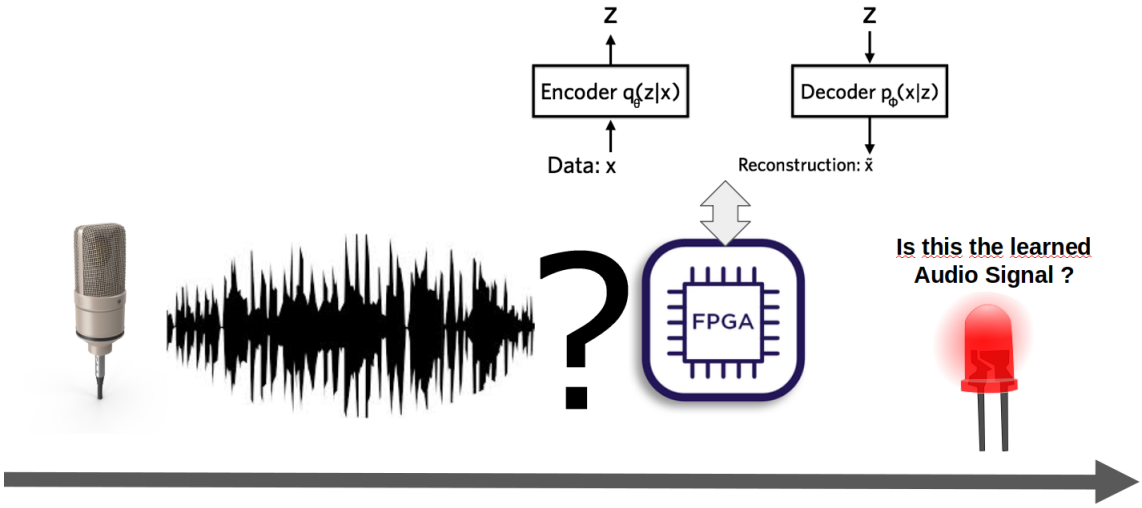
Modelle des Entwurfs in grafischer Form

Julian Hatzky

August 11, 2019



Use a Neural Network implemented on a FPGA to detect a specific Audio Signal.

The schemata of a model of the architecture as a whole. Audio Data is fed into the system via a microphone. After that a Neural Network is used to filter the incoming data and forwards it to a compare unit that decides whether it is the the right signal or not and based on that decision turn on a LED.

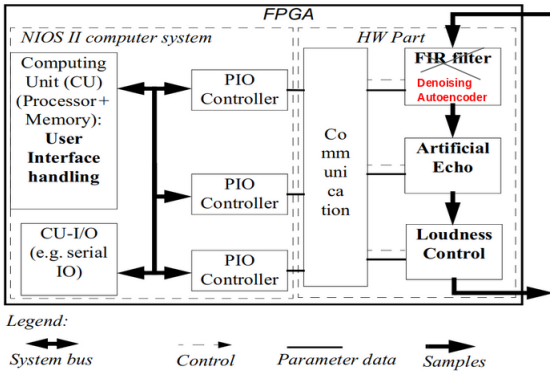
The Idea

Change the FIR Filter with a denoising variational autoencoder (deep neural network)

System Design 1
SW-HW partioning

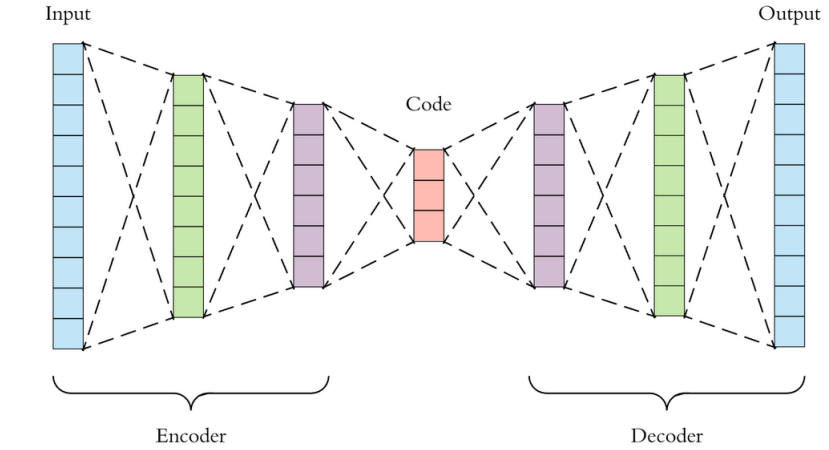
- User interface classical software task
- FIR filter, echo and loudness control: computing intensive tasks, simple functions
  => Hardware modules

In other systems, this might be not as easy as here!

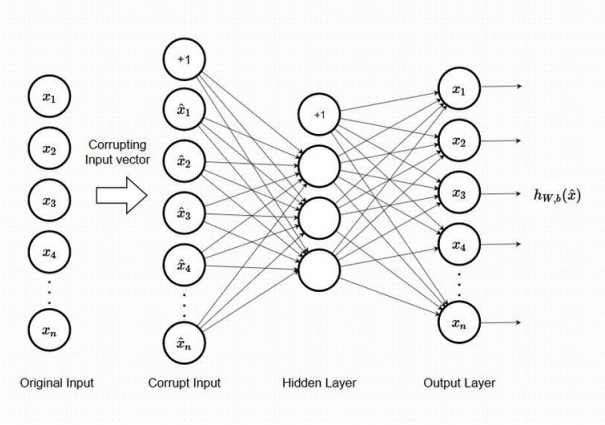


This is a top-level view of the architecture. The first idea was to use the architecture provided in the SW-HW lab and change the FIR filter with an Denoising Autoencoder. However some more changes had to be made.

Deep AutoEncoder (Unsupervised Learning)



Denoising Autoencoder (better generalization)



The AutoEncoder uses dimensionality reduction to create a latent representation of the input vector. This latent representation can be used to learn certain patterns out of the signals that represent the key components of the signal. The denoising AutoEncoder is extended in a way that noise is added to the input vector and so the AutoEncoder learns to ignore background noise and filter them out. With this architecture the denoising autoencoder can be used as a filter component.

Exponential functions and LUTs

Example for an Sigmoidal activation function.

“Journal of Applied Research and Technology “

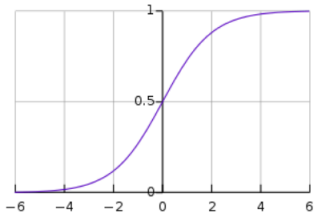
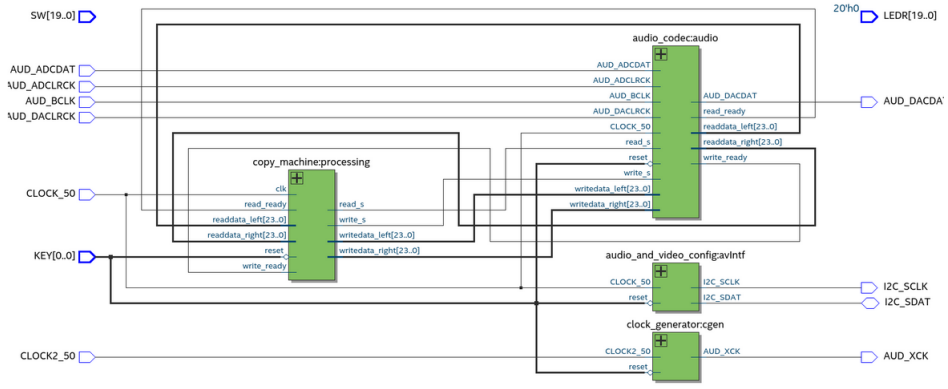


Table III. VHDL code of an approximated sigmoidal activation function.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity Sigmoid1 is
   port (
      Z: in INTEGER range 0 to 511;
      SAL: out INTEGER range 0 to 255
   );
end Sigmoid1;

architecture Sigmoid1_arch of Sigmoid1 is
signal TEMP: integer range 0 to 255;
signal B1: integer range 0 to 511;
signal A1: integer range 0 to 511;
signal ZTHETA: integer range 0 to 65535;
signal ZZ : integer range 0 to 262144;
constant L : integer range 0 to 255:= 255;
constant M : integer range 0 to 1023:= 512;
begin
A1<=Z;
B1<=M-A1;
ZZ<=B1*A1;
ZTHETA<=ZZ/256;
TEMP<=ZTHETA;
SAL<=TEMP when A1<L else L;

end Sigmoid1_arch;
```

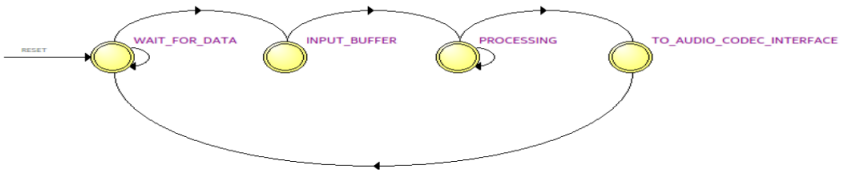
The above picture shows an approach on how to represent the exponential state of the AutoEncoder’s Neurons in VHDL.

The architecture (RTL Viewer)



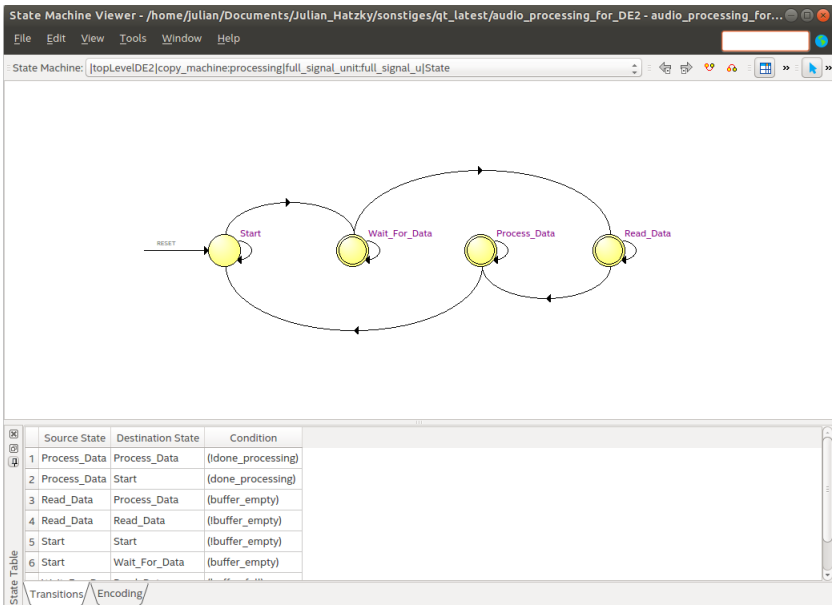
The top-level architecture created in VHDL and demonstrated by a Netlist-Viewer graph. The copy-machine is the key part of the project.

Copy Machine Control Unit (State Machine Viewer)



	Source State	Destination State	Condition
1	INPUT_BUFFER	PROCESSING	
2	PROCESSING	TO_AUDIO_CODEC_INTERFACE	(WRITE_READY)
3	PROCESSING	PROCESSING	(IWRITE_READY)
4	TO_AUDIO_CODEC_INTERFACE	WAIT_FOR_DATA	
5	WAIT_FOR_DATA	INPUT_BUFFER	(READ_READY)
6	WAIT_FOR_DATA	WAIT_FOR_DATA	(IREAD_READY)

	Name	TO_AUDIO_CODEC_INTERFACE	PROCESSING	INPUT_BUFFER	WAIT_FOR_DATA
1	WAIT_FOR_DATA	0	0	0	0
2	INPUT_BUFFER	0	0	1	1
3	PROCESSING	0	1	0	1
4	TO_AUDIO_CODEC_INTERFACE	1	0	0	1



	Source State	Destination State	Condition
1	Process_Data	Process_Data	(!done_processing)
2	Process_Data	Start	(done_processing)
3	Read_Data	Process_Data	(buffer_empty)
4	Read_Data	Read_Data	(!buffer_empty)
5	Start	Start	(!buffer_empty)
6	Start	Wait_For_Data	(buffer_empty)

The above pictures showing up the state-machines that were used within the architecture. In the documentation of the signals a more detailed explanation of the states is given.