Schaltpläne und Verdrahtungspläne

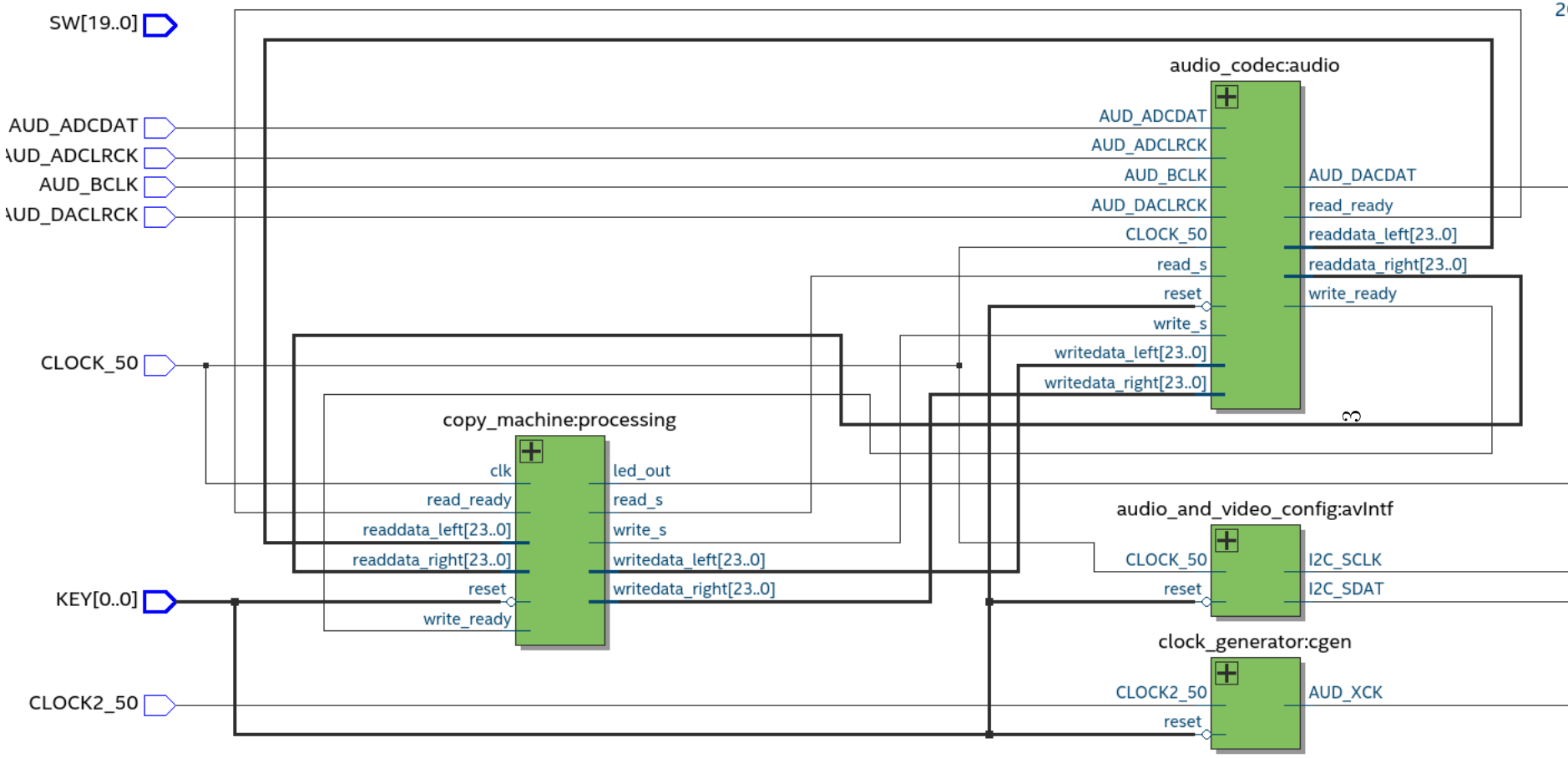
Julian Hatzky

August 8, 2019

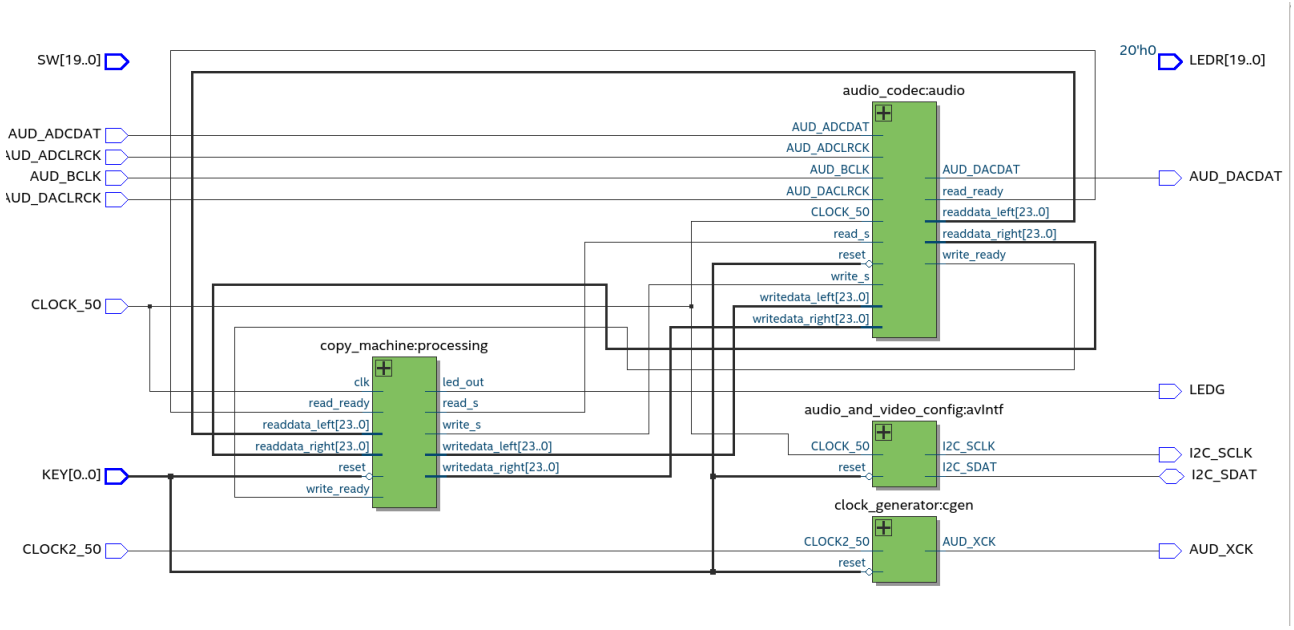
Contents

Part I

The FPGAs architecture


SW[19..0]
AUD_ADCDAT
AUD_ADCLRCK
AUD_BCLK
AUD_DACLRCK
CLOCK_50
KEY[0..0]
CLOCK2_50
audio_codec:audio
AUD_ADCDAT
AUD_ADCLRCK
AUD_BCLK
AUD_DACLRCK
CLOCK_50
read_s
reset
write_s
writedata_left[23..0]
writedata_right[23..0]
AUD_DACDAT
read_ready
readdata_left[23..0]
readdata_right[23..0]
write_ready
copy_machine:processing
clk
read_ready
readdata_left[23..0]
readdata_right[23..0]
reset
write_ready
led_out
read_s
write_s
writedata_left[23..0]
writedata_right[23..0]
audio_and_video_config:avIntf
CLOCK_50
reset
I2C_SCLK
I2C_SDAT
clock_generator:cgen
CLOCK2_50
reset
AUD_XCK

The top-level architecture



The architecture above and all that are following the same design-sheme are automatically created by the Netlist-Viewer of quartus. Hence they are representing exactly the underlying structure which is defined in VHDL. The main part of the project is the copy-machine unit, its inner cirtuits and units and its communication with the other units of the system. The audio-coded is IP (intelllectual property) of Altera and can therefore be used directly for communication with the onboard Wolfson CoDec. The audio-and-video-config unit and the clock-generator are part of the course of Prof. Hartung and therefore already provided. In the following I want to explain all the key components of the system.

The idea as a black box

In the toplevel architecture (picture above) there are two major routes that are used in order to have a full working prototype. The first routes goal is to verify the input signal. Therefore it should be possible to directly listen to the signal that is fed into the system, e.g. on a speaker adapted to the CoDec.Therefore the input signal gets written to the audio-codec and directly written to the output of the CoDec.

The second route branches the incoming signal to the copy-machine. There the detection of the signal is done and in the end the copy machine either writes a 1 to the LEDG port (led on high voltage) if the signal gets detected as the trained noise or a zero if not.

All the other signals and connection are made to support the two routes and enable these two major tasks.

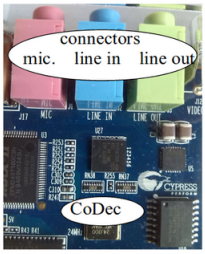
Audio Codec

The Audio Codec is IP of Altera. Because of that, it is not necessary to change the code. However, I want to give a fex explanations about the CoDec relying of the Altera Documentation and the explanations given in the lab of Prof. Hartung.

2 The Audio Subsystem of the DE2 SoPC

2.1 *Understanding the Audio Subsystem*

The foto on the right side shows the audio part on the lab experimentation board DE2-115. The three coloured connectors allow you to connect a microphone (red), a line-in cable (green) and a line-out cable (yellow). On the board they are connected to the audio CoDec chip (a Wolfson WM 8731) which is situated in the bottom of the picture..

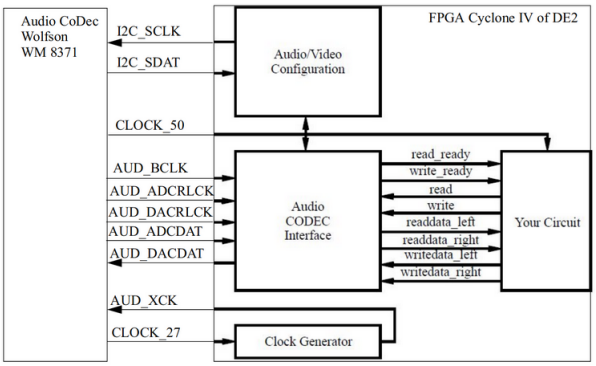


The handling of the CoDec is done by the *Audio Port* entity (a digital system hardware block provided by Altera as a *intellectual property*) which is synthesized into the FPGA (see figure in introduction). At its interface to the NIOS II processor it provides FIFO buffers for reading and writing which relieves the program from handling a sample before the next appears. Instead, it may process the samples in a block-oriented way.

The register interface of the Audio Port is given in the following figure :

Address	31 ··· 24	23 ··· 16	15 ··· 10	9	8	7 ··· 3	2	1	0	
0x10003040	Unused			WI	RI		CW	CR	WE	RE	Control register
0x10003044	WSLC	WSRC	RALC			RARC					Fifospace register
0x10003048	Left data										Leftdata register
0x1000303C	Right data										Rightdata register

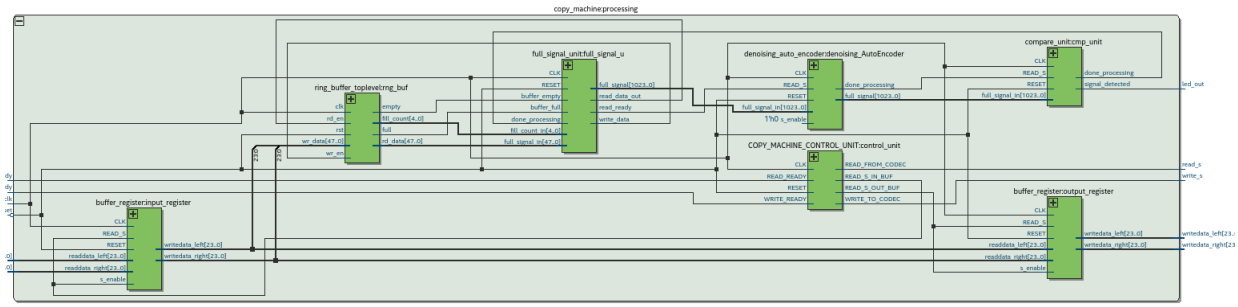
The following figure shows the connection of the CoDec to the application ('Your Circuit') via Altera's IP:



Source: Intel FPGA university prog. Educational material: Digital Logic Lab 12
Added by Author: port names between Wolfson CoDec and ACDI

Copy-Machine

4.1 Top-Level Architecture



The copy-machines primary aim is to detect whether the incoming audio-signal is the signal that its trained on (in our case the screeching noise of the train when driving through a curve) or if its an unfamiliar signal. To solve the problem the architecture above shows the nested units. The first component is a buffer-register that temporarily stores the left and right channel inputs of the audio signal. Each channel is 24 bit of data.

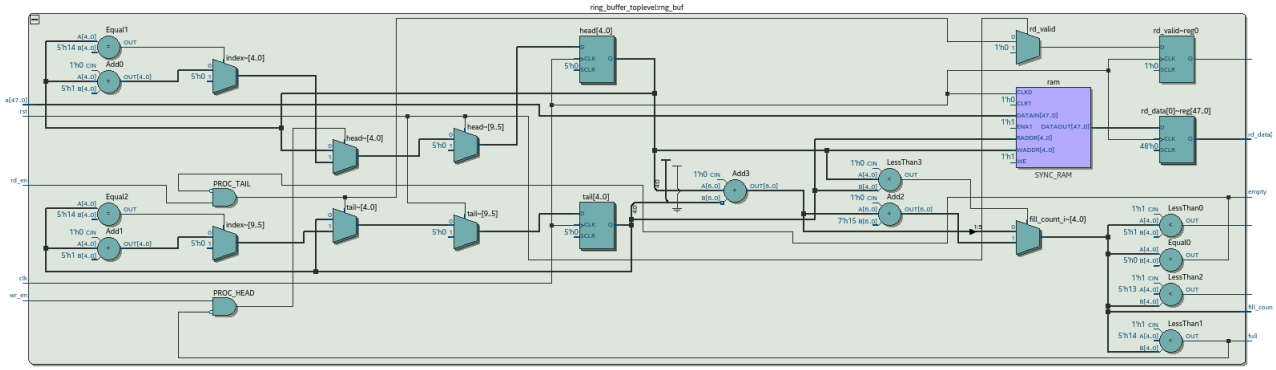
For the first route (first route described in "Idea is the black box") the left and right channel data is directly connected to an output buffer which forwarts it to the audio-codec component of the top-level architecture. The audio-codec than writes it to the output.

For the second route the right and left channel signals are connected to a ring buffer. When the ring-buffer (here 1024 bit storage capacity) is full, the full-signal-unit component uses all the data currently stored in the ring-buffer and creates a big 1024 bit vector of it. The vector than gets feeded into the denoising-auto-encoder component where the signal is filtered due to the predefined weights of the Auto Encoder Neural Network. After that the filtered signal is compared to a ideal signal within the compare-unit component and based on the result the LEDG is triggered by the output signal

of the compare-unit component.

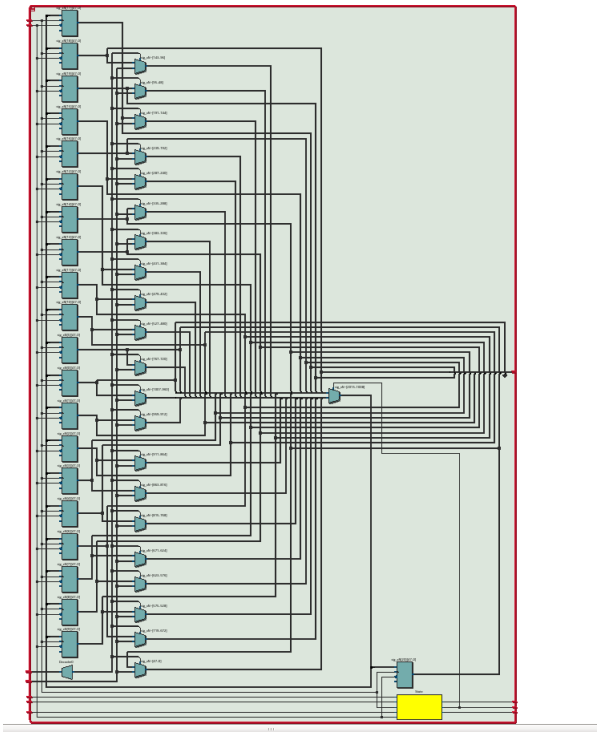
The copy-machine's components are described in the following sections.

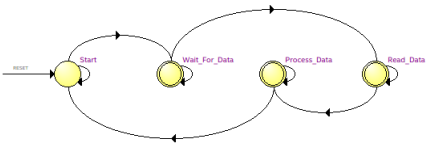
4.2 Ringbuffer component



The Ringbuffer internally has a RAM than can store up to 1024 bits of data. Data is shifted in a circular fifo manner like in known ring-buffer implementations. The ringbuffer is used to enable a sliding window technique, used by the denoising auto-encoder, which can be used to slide over the input signal and detect patterns of the known audio signal, even if they come in different shapes and to different times. In summary it is just a storing unit with additional flags like for example: is-full, is-empty, etc.

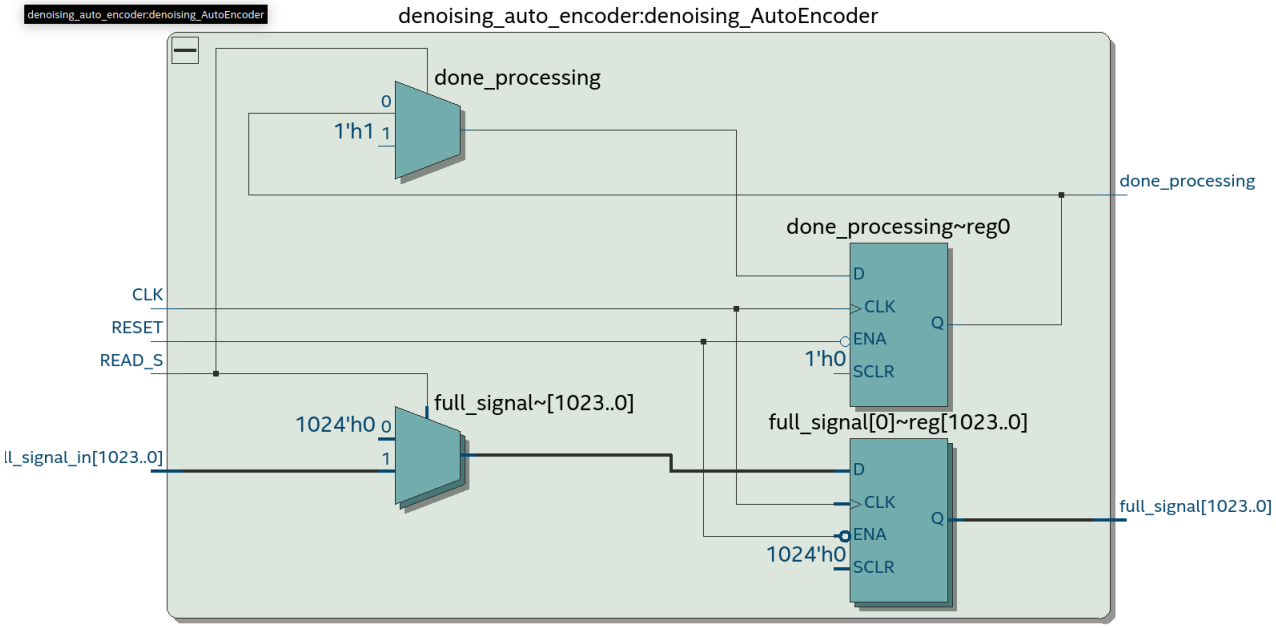
4.3 Full Signal Unit component





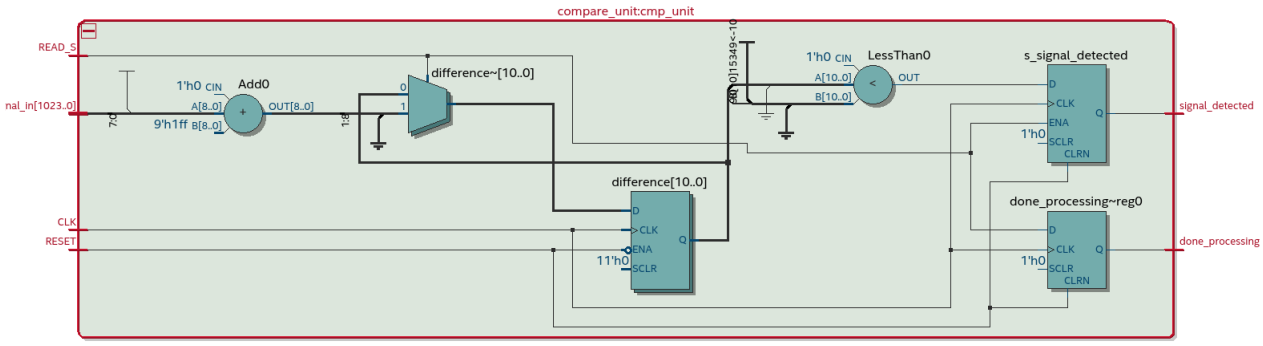
This component receives all the signal units of the ring-buffer (20 buffers with each 48 bit = 960 bit). Hence the output vector is of length 1024 the 64 bit that are left are set to zero as default. Furthermore it contains a state machine that controls the correct shifting and timing.

4.4 Denoising Autoencoder component



Unfortenitely there was not enough time to create the Auto Encoder in VHDL and because of that the signal gets just forwarded in this component.

4.5 Compare Unit component



The compare-unit compares the incoming signal to a hard coded, predefined vector which is the ideal signal and based on a maximum delta value decides whether the LEDG gets a high or low signal. (output: signal-detected) The done-processing flag is used to start a new cycle of processing.

Outlook

A more detailed explanation of the singal processing, flags and functionalities can be found in the documents "elektrischer Aufbau der Schnittstellen" and "Dokumentation von Tests und Verhalten".