

digitales Theremin Fachbericht

PROJEKT 6
12. August 2020

Auftraggeber:	Prof. Dr. Hanspeter Schmid
Betreuung:	Prof. Dr. Hanspeter Schmid Herr Prof. Karl Schenk
Team:	Andreas Frei Dennis Aeschbacher
Studiengang:	Elektro- und Informationstechnik
Semester:	Frühlingssemester 2020

Abstract

In this Project a Theremin was built that mainly operates on digital hardware unlike the original device that solely used analog electronics. The device is supposed to be used in presentations for trade fairs by the Institute for Sensors and Electronics ISE. As such the device should be built in a appealing housing. Moreover the device should have other additional functionality such as soundeffects or the ability to record sound. The digital hardware was implemented in VHDL on the developer board DE1-SoC from terasIC with a Cyclone V FPGA from Intel. The sole analog component implemented was the oscillator that controls the pitch. The pitch of the device can be changed well, but the sound itself has a flaw at the moment, because there is an audible crackle. This is due to a communication problem with the codec that was used for the audio output. This problem will not be corrected during this project, because the communication will be implemented differently in the finished product. The work in this project served as a platform for the continuation in project 6. The next steps would be to implement the volume control and redesign the pcb for two antennas and oscillators.

Inhaltsverzeichnis

1	Einleitung	1
2	Technische Grundlagen	2
2.1	analoges Theremin	2
2.2	digitales Theremin	2
3	Realisierung	3
3.1	Antennenoszillator	3
3.2	Clock	3
3.3	CPU	5
3.4	Pitch Generation	6
3.5	Volume Generation	6
4	Validierung	9
4.1	PCB	9
4.2	Frequenzmessung	9
4.3	Glissando Effekt	9
4.4	Ton Display	9
5	Schlusswort	10
6	Ehrlichkeitserklärung	11

1 Einleitung

Das Theremin kennen heutzutage nur wenige Leute, obwohl es das erste elektronische Instrument war. Es wurde 1920 von dem Russen Lev Sergejewitsch Termen, welcher sich später zu Leon Theremin umbenennen liess, erfunden **Theremin_h**. Personen die regelmässig Filme schauen, haben die Musik welche mit einem Theremin gemacht wird bestimmt schon einmal gehört. Ein Beispiel dafür ist Ghostbusters, wo das Theremin oft im Hintergrund zu hören ist. Zudem ist das Theremin in einigen Science-Fiction-Filmen und Horrorfilmen zu hören **Goast_m**. Das Theremin wird ohne es zu berühren gespielt, indem man mit den Händen die Distanz zu zwei Antennen ändert. Dies führt zu Veränderung der Tonhöhe und Lautstärke.

Im Projekt 5 und 6 soll nun ein solches Instrument entwickelt werden. Mit dem Unterschied, dass das sonst analoge Instrument digital aufgebaut werden soll. Dabei soll es auf einem Field Programmable Gate Array (FPGA) implementiert werden. Später soll das Theremin als Messobjekt für das Institut für Sensorik und Elektronik ISE verwendet werden. Im Rahmen des Projekt 5 wurde die Tonhöhenantenne des Theremin realisiert. Dazu wurde die Antenne zusammen mit dem Antennenoszillator analog beibehalten. Die restlichen Komponenten wurden in VHDL realisiert. Das Resultat wurde auf dem DE1-SoC Board von terasIC mit einem Cyclone V FPGA von Intel getestet.

Der folgende Fachbericht beginnt mit dem Kapitel 2 Technische Grundlagen. In der ersten Hälfte des Kapitel wird erklärt wie ein analoges Theremin funktioniert und welche Komponenten ein Theremin ausmachen. In der zweiten Hälfte werden digitale Lösungsansätze besprochen. Anschliessend wird im Kapitel 3 Realisierung beschrieben wie die Komponenten realisiert wurden. Im Kapitel 4 Validierung wird als erstes auf die Inbetriebnahme des Antennenoszillators eingegangen. Als nächstes werden die Simulationen des VHDL Codes erläutert. Im letzten Abschnitt wird auf die Inbetriebnahme des VHDL Codes auf dem DE1-SoC Board Bezug genommen.

2 Technische Grundlagen

In diesem Kapitel wird erklärt wie ein Theremin analog funktioniert. In der zweiten Hälfte des Kapitels wird das Theremin mit digitalen Komponenten vorgestellt.

2.1 analoges Theremin

bla bla

2.2 digitales Theremin

bla bla

3 Realisierung

Das digitale Theremin ist auf dem Entwicklungsboard DE1-SOC von terasIC aufgebaut. Dieses enthält ein Cyclone V 5CSEMA5 FPGA von Intel. Weiter befindet sich auf dem Board der Audio Codec WM8731 von Wolfson für die Ausgabe an einem Lautsprecher. In Abbildung 3.1 ist der Aufbau des Digitalen Theremin aufgezeigt inklusive der Peripherie ausserhalb des FPGA.

Das Theremin, welches im FPGA aufgebaut ist, besteht aus zwei Bereichen. Einerseits der Signalverarbeitung und Übermittlung an den Codec. Dieser besteht aus den Komponenten *Volume* und *Pitch Generation*, *DC-FIFO* und dem *Audio Serializer*. Der zweite Bereich ist Das Nios System. Dieses besteht aus dem Prozessor und diversen IP Cores, welche die Kommunikation mit den Peripherien ermöglicht. Ausserhalb des FPGA ist zudem das entwickelte PCB, welches die beiden Antennenoszillatoren enthält, mit welchen das Theremin gespielt werden kann.

Die Kommunikation zwischen dem Nios Prozessor und den anderen Komponenten geschieht über das *Avalon Memory Mapped Interface*. Der Prozessor ist in dieser Kommunikation Master und die restlichen Komponenten Slaves. Die Übertragung der Audioinformation in der Signalverarbeitung geschieht über das *Avalon Streaming Interface*. Wobei Sender als Streaming Source und Empfänger als Streaming Sink deklariert sind. Das Streaming Interface ist notwendig für den Einsatz des Dual-Clock-FIFO (DC-FIFO). Dieses übernimmt den Übergang verschiedener Clockregionen zwischen den Komponenten *Pitch Generation* und *Audio Serializer*.

Die Clocks, welche zu den verschiedenen Komponenten gehen sind in Abbildung 3.1 für eine bessere Übersichtlichkeit weggelassen worden. Für eine Liste aller Clock Frequenzen und deren Ziel siehe Kapitel 3.2.

3.1 Antennenoszillator

Text über-
arbeiten
Andy?

3.2 Clock

Die verschiednen Clocks für die Hardwarekomponenten und die CPU werden in zwei PLL Blöcken generiert. Ein Block für die Signalverarbeitung und einer für das Nios System. In Tabelle 3.1 sind alle Frequenzen aufgelistet.

Alle Frequenzen welche nicht 50MHz sind ergaben sich daraus, dass die externen Peripherien, welche mit den entsprechenden IP Cores verbunden sind, diese Frequenzen als Vorgabe haben. Weiter benötigen die Komponenten Pitch- und Volume Generation die Frequenz 54Mhz, da deren Frequenz ein vielfaches von 48kHz sein muss.

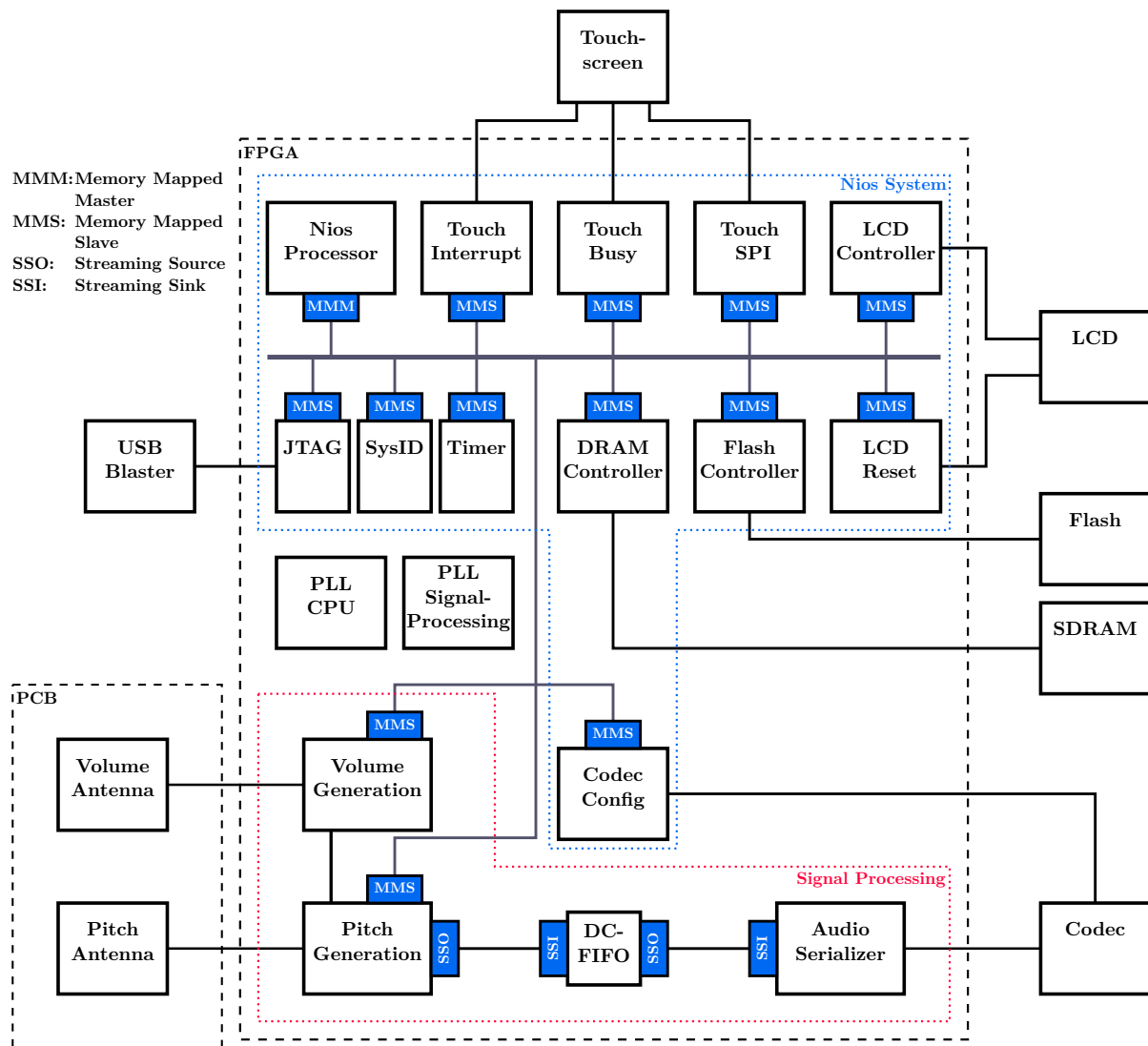


Abbildung 3.1: Blockschaltbild gesamtes Theremin

Tabelle 3.1: Clockfrequenzen der verschiedenen Komponenten

Komponente	Frequenz	PLL Core
Nios Processor	50 MHz	PLL CPU
JTAG Controller	50 MHz	PLL CPU
Timer	50 MHz	PLL CPU
SysID	50 MHz	PLL CPU
DRAM Controller	50 MHz	PLL CPU
SDRAM	50 MHz	PLL CPU
LCD Controller	15 MHz	PLL CPU
LCD Reset	15 MHz	PLL CPU
Touch Interrupt	15 MHz	PLL CPU
Touch Busy	15 MHz	PLL CPU
Touch SPI	15 MHz	PLL CPU
Audio Config	12 MHz	PLL CPU
Flash Controller	25 MHz	PLL CPU
Pitch Generation	54 MHz	PLL Signal-Processing
Volume Generation	54 MHz	PLL Signal-Processing
DC-FIFO Input	54 MHz	PLL Signal-Processing
DC-FIFO Output	24 MHz	PLL Signal-Processing
Audio Serializer	24 MHz	PLL Signal-Processing

3.3 CPU

Der eingesetzte Nios Prozessor ist für die Bedienung des Theremin und die Steuerung der Signalverarbeitungshardware zuständig. Die diversen eingesetzten IP Cores sind in den unten stehenden Kapiteln beschrieben.

JTAG, Timer und System ID

Der JTAG IP Core ermöglicht das flüchtige Programmieren des Nios wie auch das Kommunizieren mit selbem für Debugging Zwecke. Durch den Einsatz des Timer IP Cores erhält der Nios einen Interval Timer um beispielsweise periodisch Interrupts zu generieren. In dem System ID IP Core ist die Systemidentifikationsnummer gespeichert. Diese wird benötigt um beim laden der Software sicherzustellen, dass das passende Hardware Image vorhanden ist. Alle drei Komponenten sind mit Standardeinstellungen in das Nios System eingefügt worden.

Speicher

Der Arbeitsspeicher ist ein externer 64MB SDRAM Chip IS42S16320D von ISSI. Für die Kommunikation mit dem Nios Prozessor ist der SDRAM Controller IP Core zuständig. Der Nios Prozessor kann über das Memory Mapped Interface mit dem Core Kommunizieren und so auf das SDRAM zugreifen. Da dieser Chip auf dem Entwicklungsboard sowieso vorhanden ist, haben wir uns gegen Onchip Speicher entschieden um Ressourcen zu sparen.

Das Hardware Image und der Programmcode ist auf dem Board enthaltenen Flash Speicher gespeichert. Dabei wird anders als bei dem nicht flüchtigen Programmieren nicht das SRAM Object File (.sof) geladen sondern ein JTAG Indirect Configuration File (.jic). Dieses kann in Quartus aus dem SRAM Object File und dem in Eclipse generierten HEX File erstellt werden. Anschliessend kann es über ein Serial Flash Loader Image über den USB Blaster auf den Flash geladen werden. Beim Einschalten des Gerätes wird zuerst das Hardwareimage ins FPGA geladen und anschliessend der Programmcode geholt. Auf Empfehlung von Dokumentationen von Intel haben wir uns dafür entschieden den Programmcode durch einen Bootcopier ins SDRAM zu kopieren. Abbildung 3.2 zeigt das Layout des Flash Speichers nach dem Programmieren.

non_volatile

LCD Controller & Reset

Für das beschreiben des LCD ist die von terasIC bereitgestellte VHDL Komponente LT24_Controller zuständig. Dieser kann über das Memory Mapped Interface von dem Nios Prozessor gesteuert werden. Das verwendete Display LT24 von terasIC enthält für das Schreiben des LCD den LCD Treiber ILI9341 von ILITEK. Dieser Chip wird durch den LT24_Controller über das parallele 16 Bit Interface gesteuert. Weiter kann der LCD Chip über den PIO Core LCD Reset zurückgesetzt werden. Wie diese beiden Komponenten in Software angesteuert werden ist in Kapitel ?? genauer beschrieben. **LCD_Chip**

Touchscreen

Der Touch Screen Digitizer AD7843 von Analog Devices misst den resistiven Touchscreen des LCD aus und übermittelt die digitalisierten Koordinaten über SPI an den Prozessor. Der Nios

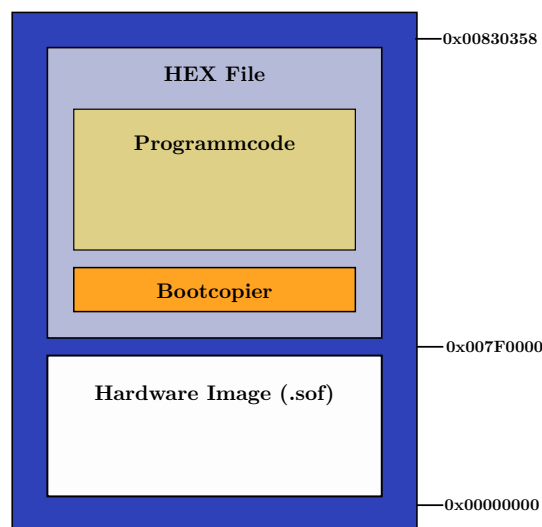


Abbildung 3.2: Layout des Flash Speichers

Kommuniziert dabei über drei verschiedene IP Cores mit diesem Chip. Der SPI Core *Touch SPI* für die Datenübertragung, der PIO Core *Touch Busy* um den Beschäftigungsstatus des Chips zu wissen und den zweiten PIO Core *Touch Interrupt*, welcher den Nios über eine Betätigung des Touchscreens informiert. Bei einer Berührung des Touchscreens löst *Touch Interrupt* beim Nios Prozessor einen Interrupt aus, welcher sofort die Koordinaten über SPI anfordert. **Touch_ADC**

3.4 Pitch Generation

Die Hauptaufgabe der Komponente Pitch Generation ist es das Audiosignal aus dem Rechtecksignal der Tonhöhenantenne (Pitch Antenna) zu generieren.

Referenzoszillator

Filter

Frequenzmessung, Kalibration & Glissandoeffekt

3.5 Volume Generation

Das digitale Theremin ist auf dem Entwicklungsboard DE1-SOC von terasIC aufgebaut. Dieses enthält ein Cyclone V 5CSEMA5 FPGA von Intel. Weiter befindet sich auf dem Board der Audio Codec WM8731 von Wolfson für die Ausgabe an einem Lautsprecher. In Abbildung ... ist der Aufbau des Digitalen Theremin aufgezeigt inklusive der Peripherie ausserhalb des FPGA.

Referenz
auf Block-
schaltbild

Filter

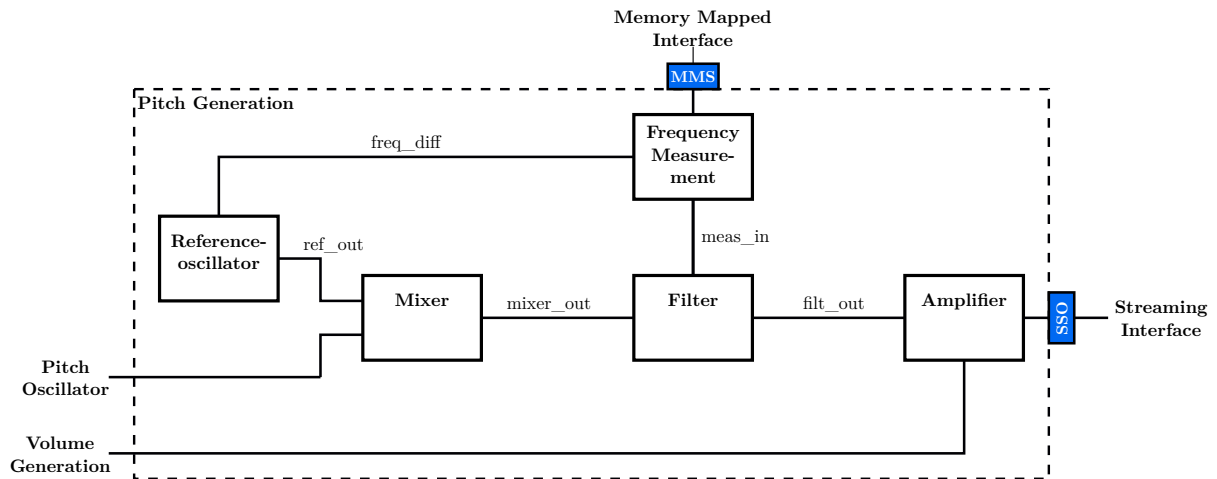


Abbildung 3.3: Blockschaltbild der Custom IP Pitch Generation

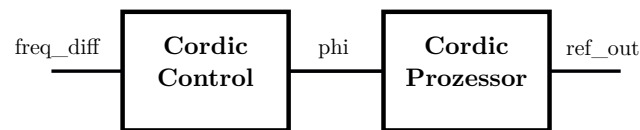


Abbildung 3.4: Aufbau des Referenzoszillators

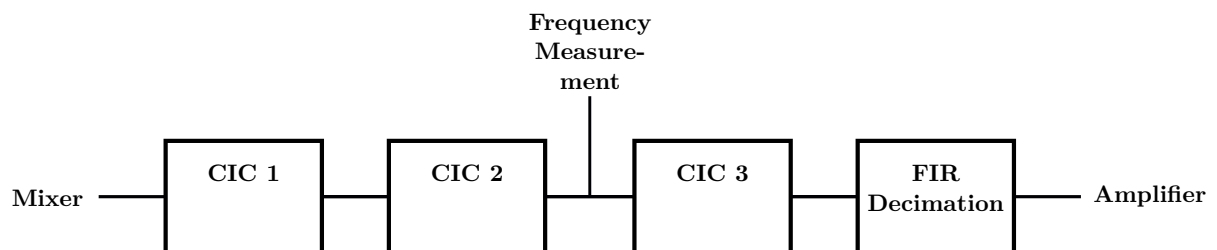


Abbildung 3.5: Aufbau des Filters in der Komponente Pitch Generation

Frequenzmessung & Kalibration

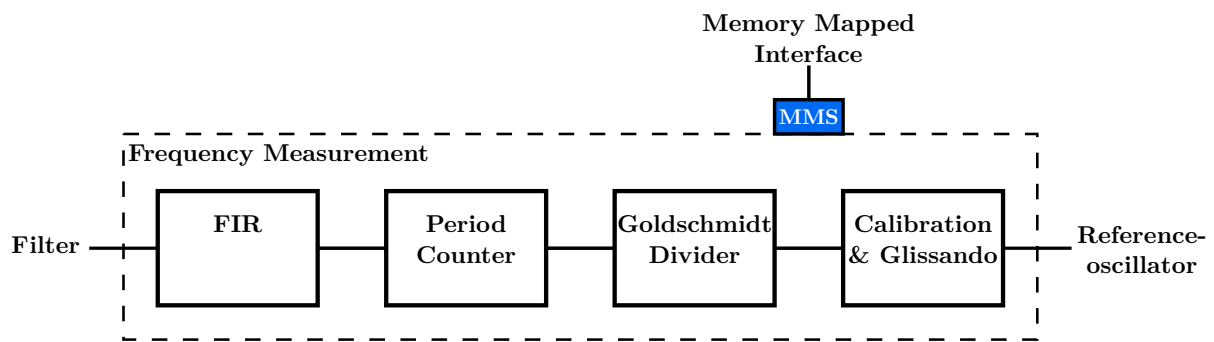


Abbildung 3.6: Aufbau der Frequenzmessung, Kalibration und Glissandoeffekt in der Komponente Pitch Generation

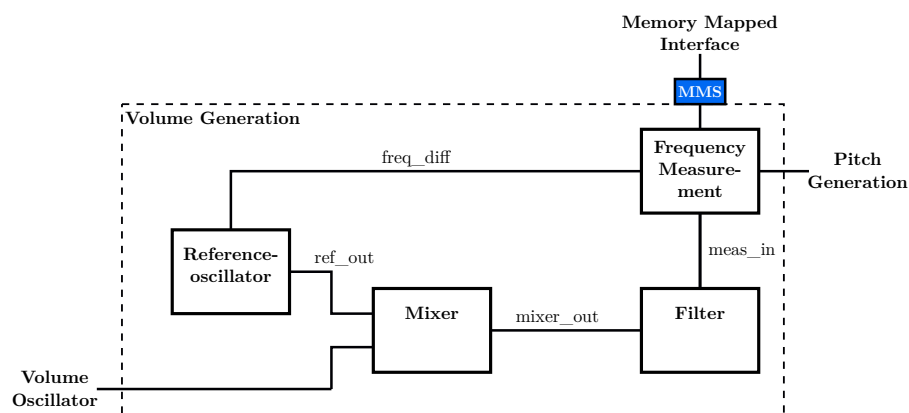


Abbildung 3.7: Blockschaltbild der Custom IP Volume Generation

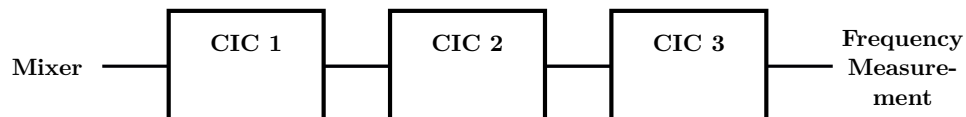


Abbildung 3.8: Aufbau des Filters in der Komponente Volume Generation

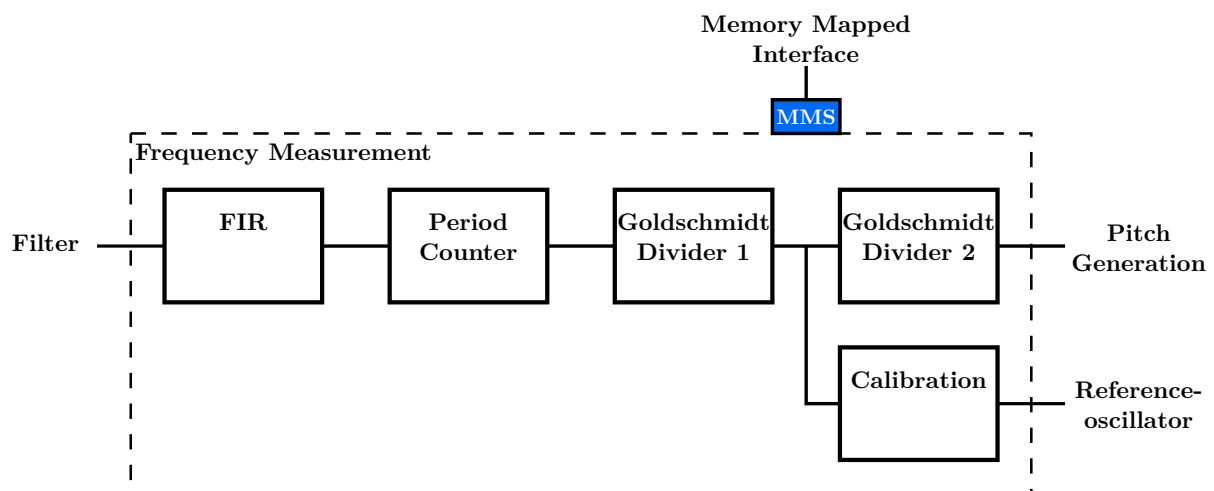


Abbildung 3.9: Aufbau der Frequenzmessung und Kalibration in der Komponente Volume Generation

4 Validierung

In diesem Kapitel wird zuerst das Antennenoszillator PCB getestet. Anschliessend wird mit Simulationen des VHDL Codes, durch berechnen der jeweiligen Spektren der Zwischenresultate, deren gesamte Funktion getestet. Zum Schluss wird auf die Inbetriebnahme und das Debugging Bezug genommen.

4.1 PCB

bla bla

4.2 Frequenzmessung

bla bla

4.3 Glissando Effekt

bla bla

4.4 Ton Display

bla bla

5 Schlusswort

Im Rahmen des Projekt 5 wurde eine digitale Plattform für die Verarbeitung von Signalen einer Thereminantenne entwickelt. Alle Komponenten ausser der Antennenoszillator wurden in VHDL realisiert. Die VHDL Komponenten wurden so realisiert, dass diese im Projekt 6 weiter gebraucht werden können. Momentan lässt sich das Theremin ohne Lautstärkeantenne spielen. Über zwei Taster kann der digitale Referenzoszillator manuell auf die Frequenz des Tonhöhenoszillator abgestimmt werden. Sobald das Theremin kalibriert ist kann es Töne von ca. 100-2000Hz spielen. Die Ziele welche in der Projektklärung definiert wurden konnten erfüllt werden. Bei der kontinuierlichen Tongenerierung gibt es noch eine Unschönheit bei der Ansteuerung des Codec. Es ist im generierten Ton ein Knacken zu hören, welches auf einen Fehler in der Ansteuerung des verwendeten Codec zurückzuführen ist. Dieser Fehler besteht nach wie vor. Jedoch wird diese Ansteuerung in Projekt 6 sowieso anders realisiert.

Im Projekt 6 wird die zweite Antenne implementiert, um gleichzeitig die Lautstärke einstellen zu können. Des weiteren soll es möglich sein diskrete Töne zu spielen. Dieser Modus soll es Anfängern ermöglichen bekannte Melodien nachspielen zu können.

Damit das theoretische Wissen aus dem Fach digitale Schaltungstechnik (dst) in die Praxis umgesetzt werden kann, wird ein Nios Soft Core Prozessor implementiert. Dieser übernimmt die Ansteuerung des Codec und die Modus Verwaltung. Beim starten des Theremin soll ein automatisches Tuning des Referenzoszillators stattfinden. Dazu wird der digitale Referenzoszillator auf die Frequenz des Antennenoszillator abgestimmt. Um das Theremin für Messen zu verwenden wird das DE1-SoC Board und die Antennenoszillatoren mit den Antennen in ein ansprechendes Gehäuse verbaut. Die Antennen sollen abgeschraubt werden können um einen komfortablen Transport zu ermöglichen.

Als erstes wird im Projekt 6 mit der Implementierung der Lautstärkeantenne auf dem FPGA und dem Redesign des PCB begonnen. Zudem muss Recherche in das Thema Nios Soft Core Prozessor angestellt werden, um diesen später implementieren zu können.

6 Ehrlichkeitserklärung

Mit der Unterschrift bestätigen die Unterzeichnenden Teammitglieder, dass die vorliegende Projektdokumentation selbstständig im Team und ohne Verwendung anderer, als der angegebenen Hilfsmittel verfasst wurde, sämtliche verwendeten Quellen erwähnt und die gängigen Zitierregeln eingehalten wurden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Unterschrift:

Ort, Datum:

Unterschrift:

Ort, Datum:
