

Programmierbare Logik

Equalizer

alternative Prüfungsleistung

Christoph Dieck (630033)

Jürgen Döffinger (631551)

27. November 2012

Inhaltsverzeichnis

1. Entwurf	1
2. Implementierung	1
2.1. Clock Generator	3
2.2. Man Machine Interface (MMI)	5
2.2.1. MMI Controller	6
2.2.2. LCD Controller	11
2.3. Audio Unit	12
2.3.1. I^2C Master	13
2.3.2. Audio Input Register	16
2.3.3. Filter Unit	18
2.3.4. Audio Output Register	25
3. Test	26
3.1. Messung des Amplitudengangs mit dem Audio-Analyzers	26
4. Fazit	27
A. Anhang	A1
A.1. Initialisierungswerte - <i>Audio Codec</i>	A1
A.2. I^2C Master - State Machine Diagramm	A4
A.3. Look-Up-Table für Komponente db2linear	A5
A.4. Cutoff-Frequenzen f_{cn} , Mittenfrequenzen f_{0n} , Güte Q_∞	A7
A.5. Messergebnisse	A8
A.5.1. Ergebnis der 1. Messung	A8
A.5.2. Ergebnis 2. Messung	A22
Literatur	B1

Abbildungsverzeichnis

1.	Blockschaltbild - Equalizer	2
2.	Blockschaltbild - <i>Clock Generator</i>	3
3.	Blockschaltbild - <i>Clock Divider</i>	4
4.	Blockschaltbild - MAN MACHINE INTERFACE (MMI)	5
5.	Blockschaltbild - <i>Audio Unit</i>	12
6.	2-Wire MPU Serial Control Mode	13
7.	Master Mode	15
8.	Left Justified Mode	17
9.	Programmablaufplan <i>IIR Filter</i>	24
10.	<i>I²C</i> Master - State Machine Diagramm	A4
11.	Amplitudengang Filter 1	A8
12.	Amplitudengang Filter 2	A9
13.	Amplitudengang Filter 3	A10
14.	Amplitudengang Filter 4	A11
15.	Amplitudengang Filter 5	A12
16.	Amplitudengang Filter 6	A13
17.	Amplitudengang Filter 7	A14
18.	Amplitudengang Filter 8	A15
19.	Amplitudengang Filter 9	A16
20.	Amplitudengang Filter 10	A17
21.	Amplitudengang Filter 11	A18
22.	Amplitudengang Filter 12	A19
23.	Amplitudengang Filter 13	A20
24.	Amplitudengang Filter 14	A21
25.	Amplitudengang Filter 1	A22
26.	Amplitudengang Filter 2	A23
27.	Amplitudengang Filter 3	A24
28.	Amplitudengang Filter 4	A25
29.	Amplitudengang Filter 5	A26
30.	Amplitudengang Filter 6	A27
31.	Amplitudengang Filter 7	A28
32.	Amplitudengang Filter 8	A29
33.	Amplitudengang Filter 9	A30
34.	Amplitudengang Filter 10	A31
35.	Amplitudengang Filter 11	A32
36.	Amplitudengang Filter 12	A33
37.	Amplitudengang Filter 13	A34
38.	Amplitudengang Filter 14	A35

Tabellenverzeichnis

1.	Tastatur- und Schalterbelegungen	6
2.	Filterauswahl und Anzeigen	8
3.	Filterauswahl und Wert des Signals <i>filter</i>	9
4.	Anzeigefunktion der 7-Segment-Anzeigen	10
5.	<i>LCD Controller</i> - Eingänge und die zugewiesenen Wertefunktionen	11
6.	Initialisierungswerte - <i>Audio Codec</i>	A3
8.	Cutoff-Frequenzen f_{cn} , Mittenfrequenzen f_{0n} , Güte Q_{∞}	A7

1. Entwurf

Das Projekt sieht vor einen 14 stufigen graphischen Equalizer in das ALTERA DE2 DEVELOPMENT AND EDUCATION BOARD zu implementieren.

Die 14 Stufen sind durch 14 Peak-Filter 2. Ordnung zu realisieren. Die Verarbeitung der Audiodaten erfolgt getrennt von der Verarbeitung der vom Benutzer eingegebenen Anweisungen. Des Weiteren ist eine Einheit zu bilden, welche sämtliche benötigten Takte erzeugt.

Die Verstärkungsfaktoren für die 14 Peak-Filter und der Lautstärkeregelung sind im Bereich von 20 dB bis -40 dB einstellbar umzusetzen.

Die Audiodaten sind vom AUDIO-CODEC abzuholen, zu Verarbeiten und an diesen wieder zu übergeben. Während der Verarbeitung sind die Daten aus der seriellen Form in eine parallele Form zu bringen und anschließend mit dem Verstärkungsfaktor der Lautstärkeregelung zu multiplizieren. Danach erfolgt die Filterung durch die 14 Filterstufen. Zum Schluss sind die auszugebenden Daten von der parallelen Form in die serielle Form zu bringen, damit diese dem AUDIO-CODEC übermittelt werden können.

Wie oben schon im Absatz beschrieben, sind die Eingaben des Benutzers von einer eigenen Einheit zu verarbeiten. Bei dieser Verarbeitung sind die Verstärkungsfaktoren einzustellen bzw. die Auswahl des Filters umzustellen und über die 7-Segment-Anzeigen und das Display auszugeben.

Weiterhin soll die Möglichkeit bestehen die Ausgabe des Audiosignals mit einem Schalter zu unterbinden (Mute), einen Reset auszuführen, auf Bypass umzuschalten und die 7-Segment-Anzeigen abzuschalten.

2. Implementierung

Alle Komponenten des Equalizers sind als reine VHDL - Dokumente erstellt worden. Es gibt keinerlei andere Formate, wie beispielsweise BLOCK DIAGRAM / SHEMATIC FILE. Dies hat den Vorteil, dass der Equalizer auch unabhängig von der Entwicklungsumgebung QUARTUS II auf andere Systeme leicht portiert werden kann. Lediglich die PIN Zuweisungen sind gegebenenfalls neu zu erstellen, bei Verwendung einer anderen Entwicklungsumgebung. Dem Entwurf ist zu entnehmen, dass der Equalizer aus drei Einheiten besteht, wie es Abbildung 1 zeigt.

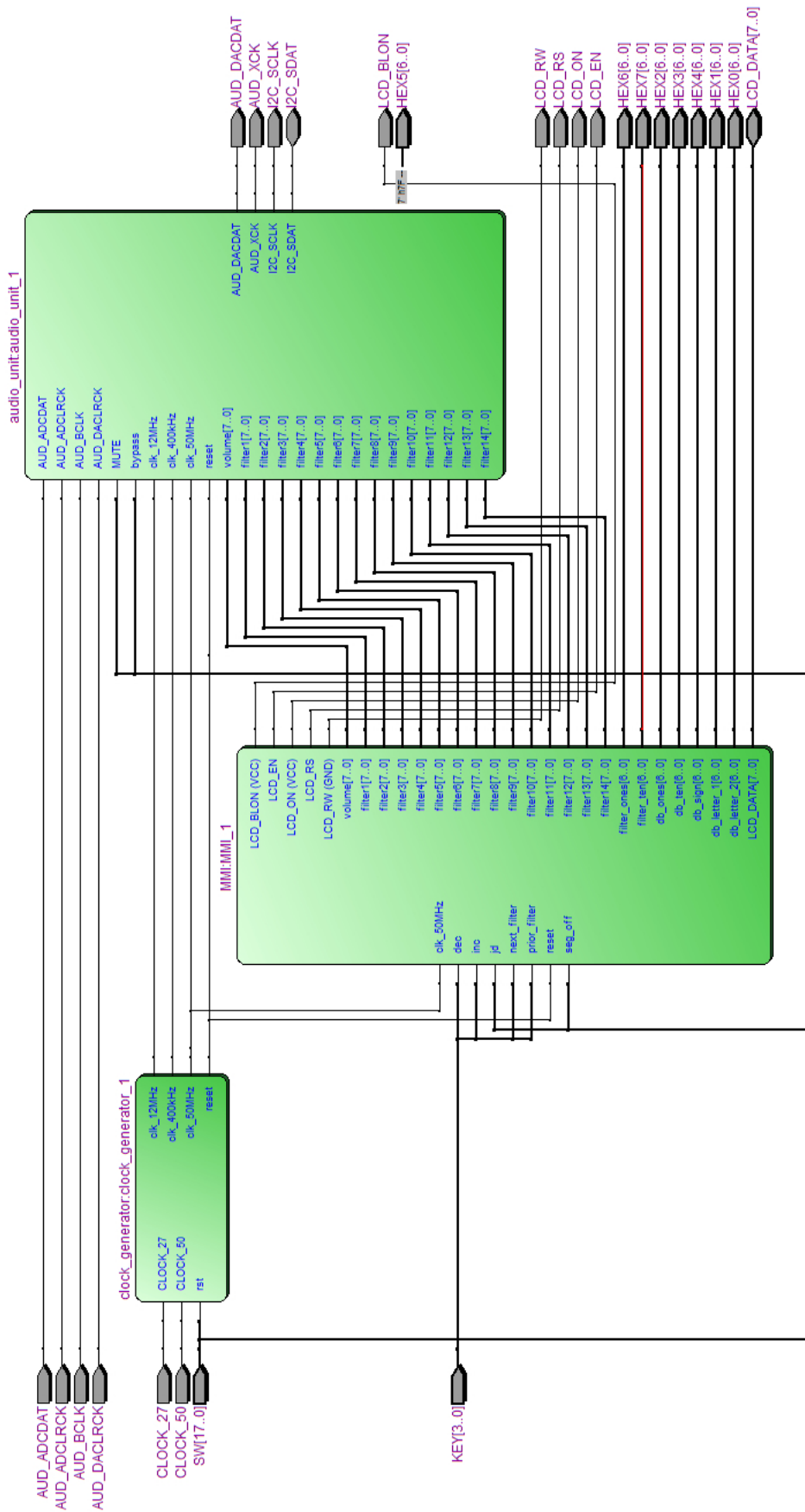


Abbildung 1: Blockschaltbild - Equalizer

Die drei Hauptbestandteile des Equalizer sind:

- Clock Generator
- MMI
- Audio Unit

2.1. Clock Generator

Die Komponente *Clock Generator* stellt den anderen Komponenten die benötigten Takte synchron zur Verfügung. Der Datenaustausch zwischen den Komponenten *Audio Unit* und *MMI* findet taktunabhängig statt.

Der *Clock Generator* stellt folgende Takte zur Verfügung:

- clk_12MHz (12 MHz Takt)
- clk_400kHz (400 kHz Takt)
- clk_50MHz (50 MHz Takt)

Dabei werden die vom Board zur Verfügung gestellten quartzesteuerten Takte 50 MHz und 27 MHz verwendet, um die oben aufgezählten Takte zu bilden. Diese werden den beiden PLL's zugeführt, wie man der Abbildung 2 entnehmen kann.

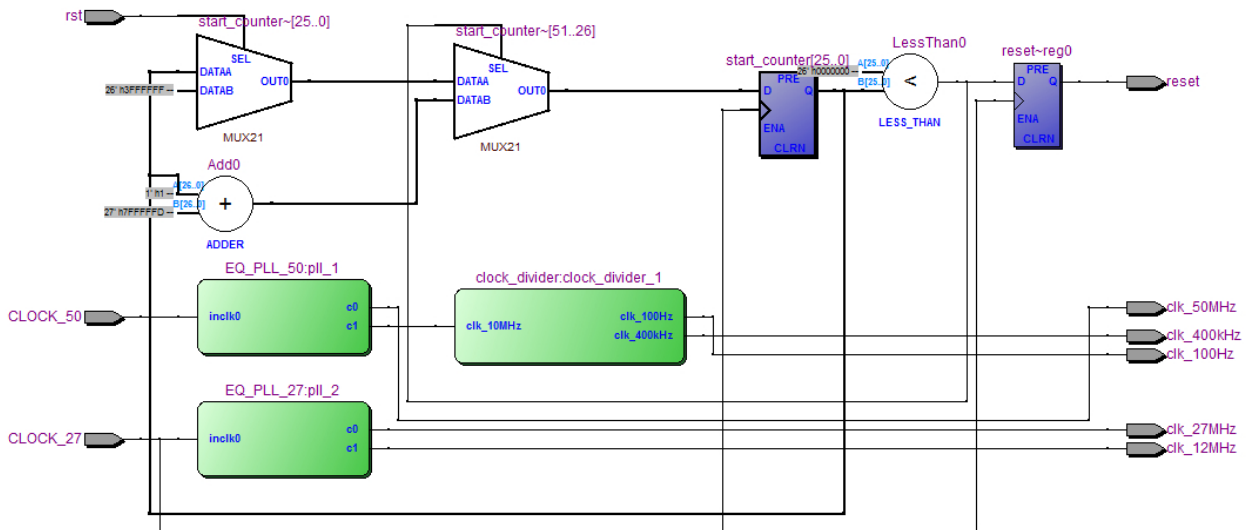


Abbildung 2: Blockschaltbild - *Clock Generator*

Die beiden PLL's wurden mit EQ_PLL_50 und EQ_PLL_27 bezeichnet. Dabei weist die Zahl in der Bezeichnung auf den eingangsseitig verwendeten Takt hin. Somit wird der Komponente EQ_PLL_50 der 50 MHz Takt und der Komponente EQ_PLL_27 der 27 MHz Takt zugeführt.

Für die PLL's kommen die internen PLL's des FPGA *Cyclone II EP2C35F672C6* zur Anwendung. Dazu wurden zwei Komponente $ALTPLL$ aus der Quartus II Bibliothek *Mega-functions* erzeugt und entsprechend konfiguriert.

Die PLL EQ_PLL_50 erzeugt die Takte 50 MHz und 10 MHz. Die PLL EQ_PLL_27 erzeugt die Takte 27 MHz und 12 MHz.

Mit Ausnahme des Taktes 10 MHz werden alle anderen direkt ausgegeben. Der Takt 10 MHz wird der Komponente *Clock Divider* zugeführt. Diese erzeugt aus dem 10 MHz Takt die Takte 100 kHz (derzeit nicht in Verwendung) und 400 kHz.

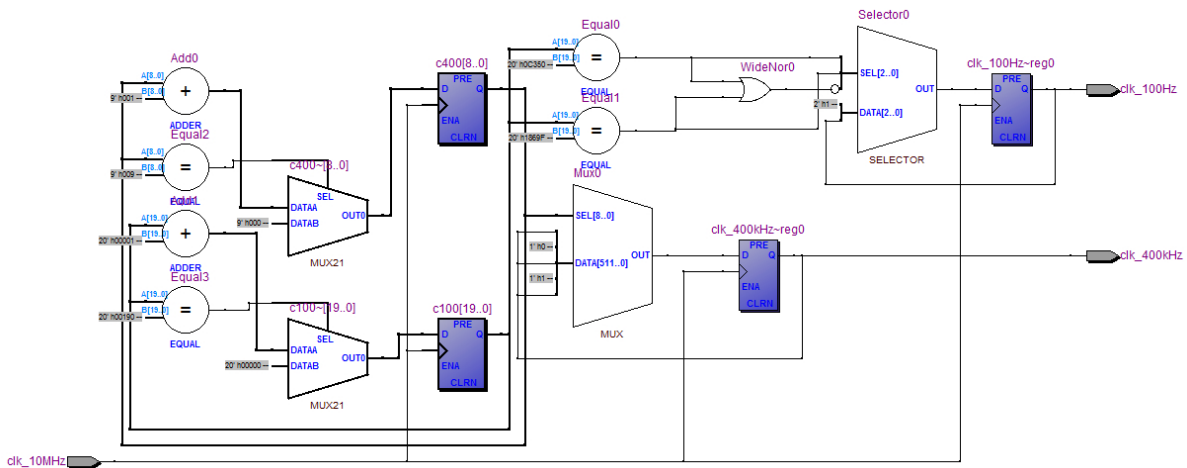


Abbildung 3: Blockschaltbild - *Clock Divider*

Die zweite Aufgabe des *Clock Generators* ist die Verarbeitung des Resets bzw. die Einschaltverzögerung nach dem Einschalten des Boards zur Verzögerung des Beginns der Arbeit der anderen Komponenten. Letzteres erfolgt durch die Unterbrechung der Taktweitergabe. Das heißt die Takte werden nicht erzeugt und somit findet in den anderen Komponenten keine Verarbeitung statt.

Bei Anforderung eines Resets über den Eingang *rst* wird ebenfalls für die Zeit von etwa zwei Sekunden kein Takt ausgegeben und gleichzeitig das Reset-Signal an alle Komponenten, welche das Reset-Signal benötigen, über den Ausgang *reset*, weitergeleitet.

2.2. Man Machine Interface (MMI)

Das MAN MACHINE INTERFACE (MMI) ist übersetzt die Benutzerschnittstelle. Sie soll dem Benutzer auf einfache Weise ermöglichen, die Filterauswahl zu treffen, den Verstärkungsfaktor des Filters zu verändern und die Eingaben mithilfe von Ausgabegeräten zu bestätigen.

Das *MMI* arbeitet unabhängig von den anderen Komponenten des Equalizers. Es gibt lediglich eine Schnittstelle zur *Audio Unit*, bei der das *MMI* die eingestellten Verstärkungsfaktoren an die *Audio Unit* weitergibt.

Weiterhin erhält das *MMI* vom *Clock Generator* einen 50 MHz Takt. Mit diesem werden innerhalb des *MMI* alle weiteren benötigten Takte erzeugt.

Die Eingaben des Benutzers erfolgen über die Taster und Schalter des Development and Education Boards.

Als Ausgabegeräte dienen die 7-Segment-Anzeigen und das LCD (liquid crystal display).

Das *MMI* besteht aus zwei Unterkomponenten.

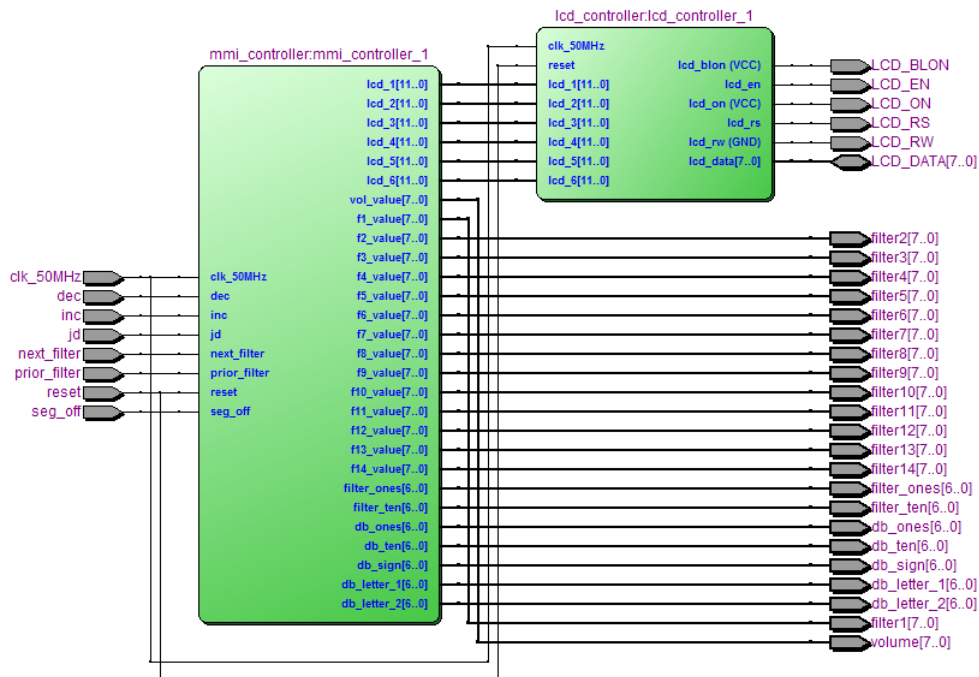


Abbildung 4: Blockschaltbild - MAN MACHINE INTERFACE (MMI)

2.2.1. MMI Controller

Der *MMI Controller* verarbeitet alle Eingaben des Benutzers und steuert sämtliche Ausgaben.

Die Eingaben erfolgen über die 4 Taster *KEY0* bis *KEY3* und die Schalter *SW0* bis *SW3* sowie *SW17*. Dabei werden über die Taster und Schalter folgende Funktionen realisiert.

Schalter / Taster	Funktion
KEY0	Verstärkungsfaktor um 1 dB erhöhen
KEY1	Verstärkungsfaktor um 1 dB verringern
KEY2	nächsten Filter auswählen
KEY3	vorherigen Filter auswählen
SW0	Mute
SW1	7-Segment-Anzeigen ein-/ausschalten
SW2	Bypass
SW17	Reset

Tabelle 1: Tastatur- und Schalterbelegungen

2.2.1.1. Reset

Betätigt der Benutzer den Schalter *SW17* einmal in Stellung 1 und dann wieder zurück in Stellung 0, wird eine Reset-Anforderung am Eingang *rst* der Komponente *Clock Generator* registriert und wie im Abschnitt 2.1 beschrieben verarbeitet. Dabei gibt die Komponente die Anforderung an den Eingang *reset* der Komponente *MMI* weiter. Das *MMI* setzt nun alle Zähler zurück, womit die Verarbeitung auf den Anfangszustand zurückgesetzt wird. Gleichzeitig wird die Filterauswahl auf den Lautstärkeregler gesetzt und alle Verstärkungsfaktoren, auch der des Lautstärkereglers, auf 0dB, also einem linearen Verstärkungsfaktor von 1, gesetzt.

2.2.1.2. Mute

Damit die Stummschaltung (im Englischen kurz *Mute* genannt) aktiv wird, muss der Benutzer den Taster *SW0* in die Position 1 bringen. In der Position 0 ist die Stummschaltung deaktiviert. Die Stummschaltung, wie man dem Namen entnehmen kann, sorgt für eine schnelle Abschaltung des Audiosignals, ohne dabei den Verstärkungsfaktor der Lautstärkeregelung zu beeinflussen.

Das Signal vom Schalter *SW0* wird direkt an den Eingang *mute* der Komponente *Audio Unit* geführt, da es für das *MMI* hier nichts zu Verarbeiten gibt. Die Komponente *Audio Unit* verknüpft das Mute-Signal mit einer UND-Verknüpfung mit dem Audio-Ausgangssignal. Eine ausführliche Erklärung der Verarbeitung des Mute-Signals in der *Audio Unit* erfolgt im Abschnitt [2.3.3](#).

2.2.1.3. Bypass

Der *Bypass* unterbindet die Filterung durch den Equalizer (EQ), sodass am Ausgang das Originalsignal vom Eingang anliegt, ohne dass dabei die Einstellungen des EQ verändert werden.

Der Benutzer muss zur Aktivierung des *Bypass* den Schalter *SW1* in Position 1 bringen. Verbleibt der Schalter in Position 0, so ist der *Bypass* deaktiviert.

Auch das Bypass-Signal wird direkt an die *Audio Unit* weitergeleitet, da auch hier für das *MMI* kein Handlungsbedarf besteht. Ist der *Bypass* aktiviert, so unterdrückt die *Audio Unit* die Filterung. Mehr dazu im Abschnitt [2.3.3](#).

2.2.1.4. Filterauswahl

Damit der Benutzer den Peak-Filtern bzw. der Lautstärkeregelung einen bestimmten Verstärkungsfaktor zuweisen kann, muss er vorher den jeweiligen Filter auswählen. Dazu hat er zwei Möglichkeiten. Er kann den vorherigen Filter mittels Taster *KEY3* und den nächsten Filter mittels Taster *KEY2* auswählen. Lässt er den jeweiligen Taster gedrückt, wird die Auswahl weitergeschaltet. Wird das Ende erreicht, so wird beim weiteren Schalten die Auswahl auf den Anfang gesetzt und umgekehrt. Der Benutzer hat dabei die Möglichkeiten zwischen folgenden Filtern zu wählen und erhält über die entsprechenden Anzeigeräte folgende Anzeigen.

Filterauswahl	Mittenfrequenz in Hz	7-Segment-Anzeige (HEX6 & HEX7)	LCD
Lautstärke		00	Vol
Filter 1	26	01	F01
Filter 2	42	02	F02
Filter 3	68	03	F03
Filter 4	112	04	F04
Filter 5	184	05	F05
Filter 6	302	06	F06
Filter 7	494	07	F07
Filter 8	809	08	F08
Filter 9	1326	09	F09
Filter 10	2171	10	F10
Filter 11	3557	11	F11
Filter 12	5825	12	F12
Filter 13	9541	13	F13
Filter 14	15627	14	F14

Tabelle 2: Filterauswahl und Anzeigen

Eine ausführliche Beschreibung der jeweiligen Filtereigenschaften findet sich im Abschnitt [2.3.3.4](#).

Die Auswahl des Filters wird im Signal *filter* gespeichert. Ändert der Benutzer die Auswahl über die zuvor beschriebenen Prozeduren, so wird dieses Signal auf den entsprechenden Wert gesetzt. Dabei sind den Filtern folgende Werte zugewiesen.

Filterauswahl	Wert des Signals <i>filter</i>
Lautstärke	0000
Filter 1	0001
Filter 2	0010
Filter 3	0011
Filter 4	0100
Filter 5	0101
Filter 6	0110
Filter 7	0111
Filter 8	1000
Filter 9	1001
Filter 10	1010
Filter 11	1011
Filter 12	1100
Filter 13	1101
Filter 14	1110

Tabelle 3: Filterauswahl und Wert des Signals *filter*

2.2.1.5. Einstellung des Verstärkungsfaktors

Hat der Benutzer einen Filter oder die Lautstärke gewählt, so kann er mithilfe der Taster *KEY1* und *KEY0* den entsprechenden Verstärkungsfaktor verändern. Dabei wird der Verstärkungsfaktor durch betätigen des Tasters *KEY1* um 1 dB abgesenkt und bei betätigen des Tasters *KEY0* um 1 dB erhöht.

Auch hier kann der Benutzer die Taster gedrückt halten, um damit ein weiterschalten zu erzwingen. Der Einstellbereich für die Verstärkungsfaktoren liegt im Bereich von 20 dB bis -40 dB.

Die Werte der Verstärkungsfaktoren werden in den Signalen *f1_db* bis *f14_db* hinterlegt und der Verstärkungsfaktor für die Lautstärke im Signal *vol_db*.

2.2.1.6. 7-Segment-Anzeigen

Die 7-Segment-Anzeigen, zeigen den ausgewählten Filter und dessen eingestellten Verstärkungsfaktor an. Dabei werden alle 7-Segment-Anzeigen außer der *HEX5*, welche dunkel geschaltet wird, verwendet. Die Anzeigen sind in folgende Funktionen aufgeteilt.

7-Segment-Anzeige	Funktion
HEX7	10er Stelle der Filterauswahl
HEX6	1er Stelle der Filterauswahl
HEX5	keine Funktion zugewiesen
HEX4	Vorzeichen des angezeigten Verstärkungsfaktor
HEX3	10er Stelle des eingestellten Verstärkungsfaktors
HEX2	1er Stelle des eingestellten Verstärkungsfaktors
HEX1	zeigt ständig das <i>d</i> für die Anzeige dB an.
HEX0	zeigt ständig das <i>b</i> für die Anzeige dB an.

Tabelle 4: Anzeigefunktion der 7-Segment-Anzeigen

Solange der Schalter *SW1* in Position 1 steht, werden alle Anzeigeelemente dunkel geschaltet, also auf logisch 1 gesetzt. Steht jedoch der Schalter *SW1* in Position 0, dann zeigen die 7-Segment-Anzeigen die ihnen zugewiesenen Werte an.

Der Wert der Filterauswahl wird direkt im *MMI Controller* auf das 7-Segment-Anzeigen-Format umcodiert und als *filter_ones* für die 1er Stelle und *filter_ten* für die 10er Stelle ausgegeben.

Das Gleiche gilt für den anzuzeigenden Verstärkungsfaktor. Dieser wird dann als *db_sign* für das Vorzeichen, *db_ten* für die 10er Stelle und *db_ones* für die 1er Stelle ausgegeben.

Die Anzeige db wird über die Ausgänge *db_letter_1* für das d und *db_letter_2* für das b ausgegeben.

Alle Ausgänge sind direkt mit den 7-Segment-Anzeigen verbunden.

2.2.2. LCD Controller

Der *LCD Controller* steuert die Ausgaben auf das LCD (im weiteren Verlauf der Dokumentation als Display bezeichnet).

Nach dem Einschalten des Development and Education Board wird das Display initialisiert. Anschließend werden die Ausgaben auf dem Display kontinuierlich aktualisiert.

Bei der Initialisierung des Displays wird zunächst die Anweisung *Function Set* übermittelt. Dabei wird das Display auf 8 bit, 2 Zeilen und 5x8 Pixel pro Zeichen eingestellt. Als nächstes erfolgt die Anweisung *Display On/Off*. Dabei wird das Display eingeschaltet und der Cursor auf nicht blinkend eingestellt und abgeschaltet. Das Display zeigt also keinen vorangestellten Cursor an. Als nächstes wird das Display gelöscht, womit der Cursor auf die position Zeile 1 Spalte 1 gesetzt wird. Die nächste Anweisung ist *Entry Mode*, welche das Display anweist die Cursorposition nach dem übermitteln eines anzuzeigenden Zeichens um eine Position weiter nach rechts zu springen und ggf. in die nächste Zeile zu springen. Danach folgen Anweisungen, welche dafür sorgen, dass in der ersten Zeile der Schriftzug *Equalizer* erscheint.

Anschließend wird die zweite Zeile befüllt. Dies erfolgt kontinuierlich in einer Schleife. Dabei werden die Eingangswerte *lcd_1* bis *lcd_6*, welche vom *MMI Controller* übermittelt werden, auf dem Display ausgegeben und enthalten folgende Werte.

Eingang	Wertefunktion
lcd_1	1. Zeichen der Filterauswahl
lcd_2	2. Zeichen der Filterauswahl
lcd_3	3. Zeichen der Filterauswahl
lcd_4	Vorzeichen des eingestellten Verstärkungsfaktors
lcd_5	10er Stelle des eingestellten Verstärkungsfaktors
lcd_6	1er Stelle des eingestellten Verstärkungsfaktors

Tabelle 5: *LCD Controller* - Eingänge und die zugewiesenen Wertefunktionen

Die entsprechenden Zeichen der Filterauswahl können dem Abschnitt [2.2.1.4](#) entnommen werden.

Die an den Eingängen *lcd_1* bis *lcd_6* anliegenden Informationen werden vom *MMI Controller* über die gleichnamigen Ausgänge ausgegeben. Der Inhalt der Werte entspricht der Speicherposition im CGRAM des Displays. Die jeweilige Speicherposition im CGRAM enthält ein Zeichen des ASCII-Zeichensatzes. Im *MMI Controller* wurden die Werte der Filterauswahl und des zugehörigen Verstärkungsfaktors zuvor entsprechend umcodiert.

2.3. Audio Unit

Die *Audio Unit* besteht aus vier Komponenten. Diese Komponenten haben folgende Aufgaben.

- Initialisierung des *Audio Codec*
- Daten vom *Audio Codec* lesen und für die weitere Verarbeitung aufbereiten
- Filterung durchführen und Lautstärke anpassen
- Daten dem *Audio Codec* übergeben

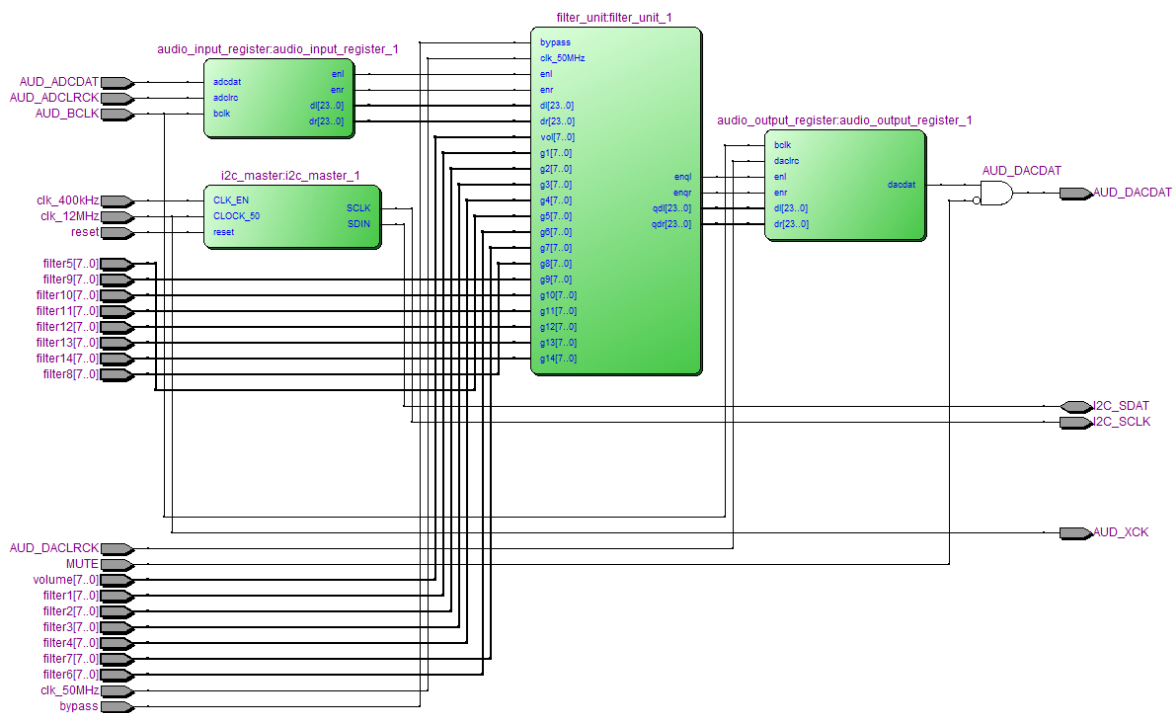


Abbildung 5: Blockschaltbild - *Audio Unit*

Die vier Komponenten sind:

- I^2C Master
- Audio Input Register
- Filter Unit
- Audio Output Register

2.3.1. I²C Master

Der *I²C Master* ist für die Initialisierung des *Audio Codec* zuständig. Hier müssen diverse Einstellungen vorgenommen werden, welche im Folgenden noch beschrieben werden.

Die Anweisungen werden über den *I²C-Bus*¹ übertragen. Die Kommunikation über den *I²C-Bus* kann auf zwei Arten erfolgen. Einmal dem *3-Wire MPU Serial Control Mode* und dem *2-Wire MPU Serial Control Mode*. Da beim verwendeten Development and Education Board die Pins *CSB* und *Mode* des *Audio Codec* fest mit *GND* verbunden sind, wurde somit von Seiten des Herstellers der Mode vorgegeben. Es kommt somit der *2-Wire MPU Serial Control Mode* zur Anwendung.

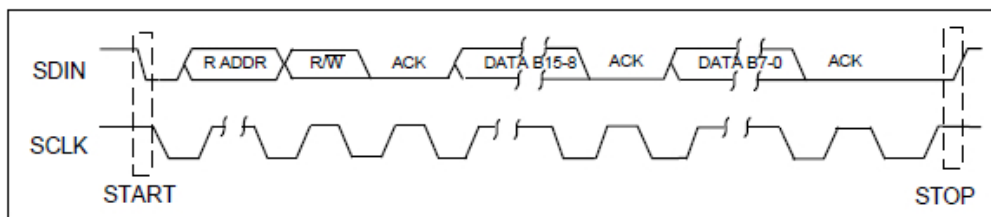


Abbildung 6: 2-Wire MPU Serial Control Mode²

Der *I²C Master* wurde als State Machine konzipiert, wie es die Abbildung 10 im Anhang A.2 zeigt.

Als Erstes muss die Start-Sequenz gesendet werden (siehe Abbildung 6 – START).

Die jeweiligen Daten werden im Format des in Abbildung 6 gezeigten Diagramms übertragen. Da es sich um einen BUS handelt, an dem auch mehr als nur ein IC lauschen kann, muss zunächst die Adresse (R ADDR) gesendet werden, welche den IC eindeutig identifiziert. Diese besteht aus 8 bit, wobei das 8. Bit dem IC übermittelt, ob auf ihn geschrieben oder von ihm gelesen werden soll. Bei dem hier verwendeten IC ist die Adresse 34_{HEX} und somit das 8. Bit auf logisch 0 gesetzt, was dem *Audio Codec* mitteilt, dass ihm Daten übermittelt werden. Nach dieser Übertragung wartet der *I²C Master* auf das *ACK*-Signal vom *Audio Codec*.

Sollte der *Audio Codec* kein *ACK* senden, so wird die Datenübertragung abgebrochen. Wurde das *ACK* gesendet, wird mit der Übertragung des zweiten Datenblocks mit den Datenbits 15 bis 8 fortgefahren. Die Datenbits 15 bis 8 repräsentieren die Register Adresse im *Audio Codec*. Nach Übertragung wartet der *I²C Master* erneut auf ein *ACK* vom *Audio Codec*.

¹http://www.nxp.com/documents/user_manual/UM10204.pdf

²[5]

Sollte der *Audio Codec* kein *ACK* senden, so wird auch hier die Datenübertragung abgebrochen. Wurde jedoch ein *ACK* empfangen, so werden die letzten 8 Bits des Datenblocks übertragen. Diese werden in das Register des *Audio Codec* mit der Adresse welche zuvor gesendet wurde geschrieben. Diese 8 Datenbits repräsentieren den Datenwert, wobei hier für bestimmte Einstellungen teilweise die Informationen in den einzelnen Bits stecken. Auch hier wird auf ein *ACK* gewartet.

Sollte der *Audio Codec* kein *ACK* senden, so wird auch hier die Datenübertragung abgebrochen. Wurde das *ACK* empfangen, so muss der *I²C Master* noch die Stop-Sequenz senden (siehe Abbildung 6 – STOP).

Die Datenübertragung auf dem *I²C*-BUS findet im *FAST Mode* statt, also mit 400 kHz als Taktfrequenz (SCLK).

Dieser Ablauf muss für jede Einstellung durchlaufen werden. Die Tabelle 6 im Anhang A.1 zeigt die Reihenfolge mit der die Einstellungen gesendet werden und auf welche Parameter der *Audio Codec* eingestellt wird.

Einige Parameter möchte ich an dieser Stelle näher erläutern, da diese für die Verarbeitung der Audiodaten bzw. der Datenübertragung der Audiodaten zwischen *Audio Unit* und *Audio Codec* von Bedeutung sind.

Zunächst werden alle Register auf Standardwerte zurückgesetzt und anschließend die Verstärkungsfaktoren der Lautstärkeregelung an den Audioeingängen auf 0 dB gesetzt. Die Verstärkungsfaktoren der Lautstärkeregelung an den Audio-Ausgängen werden ebenfalls auf 0 dB eingestellt.

Des Weiteren wird das Mikrofon und der Bypass (des *Audio Codec*) abgeschaltet. Und die Audioeingänge auf den ADC und die Audioausgänge auf den DAC geroutet.

Der Hochpassfilter zur Rauschunterdrückung wird zugeschaltet und das De-emphasis-Filter abgeschaltet. Weiterhin werden alle nicht benötigten Komponenten des *Audio Codecs* von der Stromversorgung getrennt (Power Down Control).

Die Wortlänge der Audiodaten wird auf 24 bit eingestellt. Die Übertragung findet im *Left Justified Mode* statt, auf dem im Abschnitt 2.3.2 noch näher eingegangen wird. Der *Audio Codec* wird in den Master Mode gebracht. Dies bedeutet, dass er alle Takte vorgibt, wie es die Abbildung 7 zeigt.

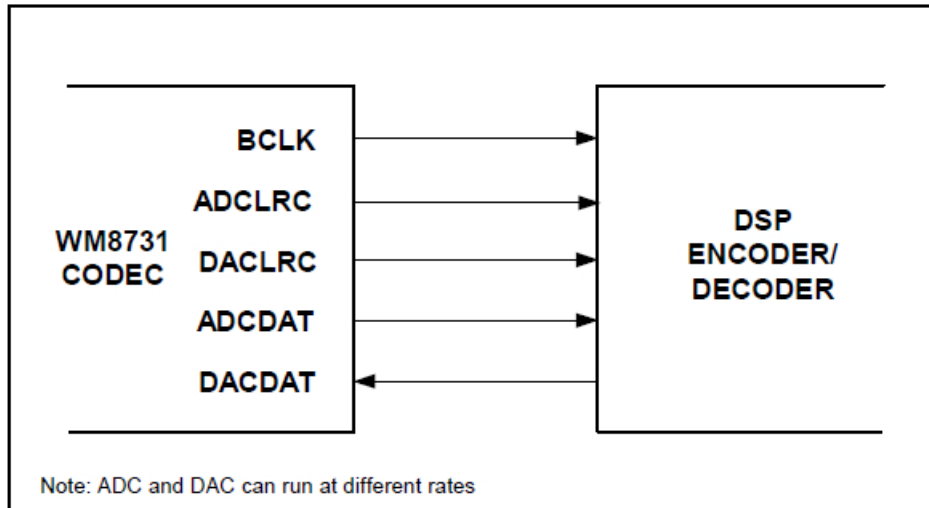


Abbildung 7: Master Mode³

Die Abtastfrequenz (Sample Rate) f_s wird sowohl für den ADC als auch für den DAC auf 48 kHz eingestellt. Gleichzeitig wird der *Audio Codec* in den *USB Mode* versetzt und somit der Takt $BCLK$ auf $250 \cdot f_s$ eingestellt, also auf 12 MHz. Das heißt die Übertragung wird durch den *Left Justified Mode* in zwei Hälften aufgeteilt.

Zunächst werden die Daten des linken Kanals übertragen und nach

$$\frac{250}{2 \cdot 12MHz} \approx 10,42\mu s \quad (1)$$

die des rechten Kanals. Wobei die Übertragung der 24 Bits pro Kanal bereits nach

$$\frac{24}{12MHz} = 2\mu s \quad (2)$$

abgeschlossen ist, womit für die Verarbeitung pro Kanal

$$\frac{250}{2 \cdot 12MHz} - \frac{24}{12MHz} \approx 8,42\mu s \quad (3)$$

³[5]

zur Verfügung stehen. Abzuziehen sind hier noch die Übertragungszeiten der Audiodaten von *Audio Input Register* zu *Filter Unit* und nach der Verarbeitung weiter zum *Audio Output Register*. Dies erfolgt innerhalb von 3 Takten und somit innerhalb einer Zeit von 250ns. Allerdings findet diese Übertragung noch innerhalb der Zeit statt, in der die Daten des jeweils anderen Kanals verarbeitet werden. Das heißt, während die Daten des linken Kanals vom *Audio Codec* empfangen bzw. zum *Audio Codec* gesendet werden, werden die Daten des rechten Kanals in der *Filter Unit* verarbeitet und direkt nach Abschluss der Verarbeitung an das *Audio Output Register* übergeben. Gleichzeitig werden die Audiodaten nach vollständigem Empfang durch das *Audio Input Register* direkt an die *Filter Unit* weitergegeben, sodass die Daten in den entsprechenden Komponenten bereits vorliegen, wenn sie dann benötigt werden. Gleiches gilt natürlich umgekehrt für den Empfang und Sendung der Daten des rechten Kanals und der Verarbeitung der Daten des linken Kanals.

2.3.2. Audio Input Register

Das *Audio Input Register* hat die Aufgabe, die Audiodaten beim *Audio Codec* abzuholen, eine Seriell-Parallel-Wandlung durchzuführen und die Daten der *Filter Unit* bereitzustellen.

Am Eingang des *Audio Input Registers* liegen folgende Signale an:

- AUD_ADCDAT
- AUD_ADCLRCK
- AUD_BCLK

Das Taktsignal *AUD_BCLK* ist der 12 MHz Takt der durch den *Audio Codec* vorgegeben wird. Dies ist der Takt mit dem die einzelnen Bits des Signals *AUD_ADCDAT* vom *Audio Codec* übertragen werden. Das Signal *AUD_ADCLRCK* zeigt an, ob die Daten des linken oder des rechten Kanals übertragen werden. Bei der Übertragung der Daten kommt der *Left Justified Mode* zur Anwendung. Dem Diagramm in Abbildung 8 kann der Ablauf der Übertragung beim *Left Justified Mode* entnommen werden.

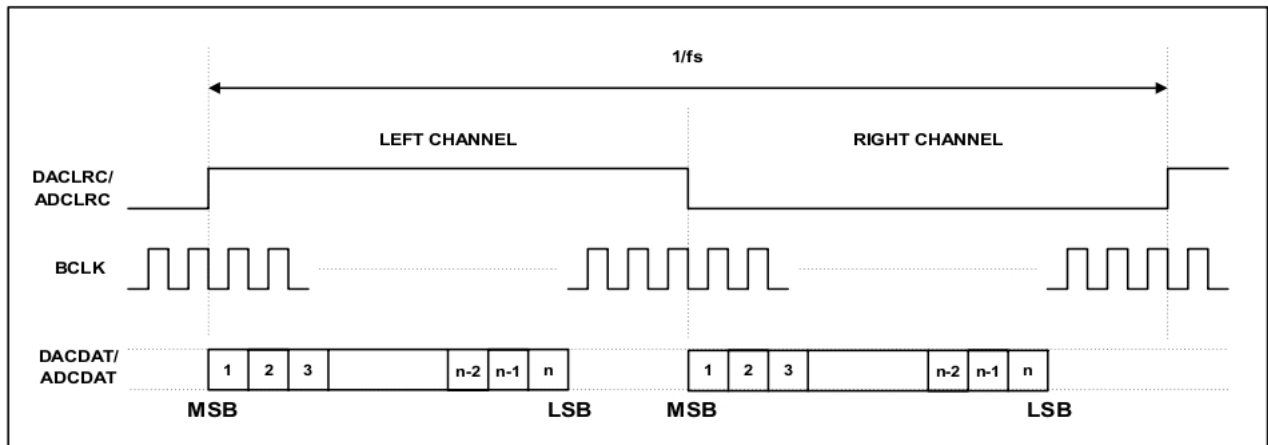


Abbildung 8: Left Justified Mode⁴

Wie dem Diagramm zu entnehmen ist, werden die Daten des rechten und linken Kanals nicht gleichzeitig übertragen. Daher wartet das *Audio Input Register* auf die Änderung des Signals *AUD_ADCLRCK*. Ändert es sich, wird der Zähler des linken bzw. rechten Kanals zurückgesetzt. Die Zähler dienen der Feststellung, welches der 24 bit gerade übertragen wird. Die Bits des Audiosignals kommen am Eingang *AUD_ADCDAT* an und werden mit steigender Flanke des Signals *AUD_BCLK* übernommen und dem Signal *data* zugeführt. Dabei wird das entsprechende Signal einmal einem Linksschieben unterzogen und das übertragene Bit als LSB angehängt.

Nach Empfang der 24 Bit wird der Ausgang *enl* für den linken und *enr* für den rechten Kanal auf logisch 0 gesetzt und das Signal *data* dem Ausgang *dl* für den linken und *dr* für den rechten Kanal übergeben. Nach dieser Übergabe wird der jeweilige Ausgang *enl* bzw. *enr* auf logisch 1 gesetzt. Der Flankenwechsel von 1 auf 0 bzw. dann von 0 auf 1 der Ausgänge *enl* und *enr* signalisiert der folgenden Komponente, dass sich die Daten ändern werden bzw. dass nun neue Daten anliegen.

⁴[5]

2.3.3. Filter Unit

Die *Filter Unit* ist das Herzstück der Audioverarbeitung. Sie hat die Aufgabe die Filterung gemäß den Vorgaben des Benutzers durchzuführen. Die *Filter Unit* besteht aus weiteren vier Komponenten.

- dB zu linear Decoder (db2linear)
- Volume Multiplier
- Coefficients Register
- IIR Filter

Die *Filter Unit* bekommt, wie in Abschnitt 2.3.2 schon beschrieben, die Audiodaten vom *Audio Input Register*. Anschließend werden die Audiodaten mit dem Verstärkungsfaktor der Lautstärkeregelung multipliziert. Da der Verstärkungsfaktor vom *MMI* in dB übergeben wird, muss dieser vorher noch in den linearen Verstärkungsfaktor gewandelt werden. Dazu dient die Komponente *db2linear*. Die Wandlung erfolgt mittels Look-Up-Table. Wurde vom Benutzer Bypass zugeschaltet, so werden die Audiodaten dem Ausgang zugeführt, ansonsten werden Sie dem *IIR Filter* übergeben. Dieser berechnet aus den Vorgaben des Benutzers das gefilterte Audiosignal, welches nach der Filterung an den Ausgang übergeben wird. Bevor die Daten an das *Audio Output Register* übergeben werden, werden sie mit dem Mute-Signal UND-Verknüpft. Dies führt dazu, dass nur logische Nullen ausgegeben werden, wenn der Benutzer Mute zugeschaltet hat. Das Ausgeben von logischen Nullen führt dazu das kein Ton ausgegeben wird.

2.3.3.1. db zu linear Decoder (db2linear)

Der Decoder wandelt die ihm übergebenen logarithmischen Verstärkungsfaktoren in die entsprechenden linearen Verstärkungsfaktoren um. Dazu wird eine Look-Up-Table verwendet. Die LUT ist dem Dokument als Anhang A.3 angehängt. Die LUT wird in der VHDL-Umsetzung als selektive Signalzuweisung realisiert.

2.3.3.2. Volume Multiplier

Der *Volume Multiplier* dient dem Lautstärkeregler zur Einstellung der Lautstärke des Ausgangssignals in Abhängigkeit vom Eingangssignal. Hierzu wird der lineare Verstärkungsfaktor mit dem Eingangssignal multipliziert.

Als Multiplizierer kommt die Komponente *LPM_MULT* aus der Quartus II Bibliothek *Mega-functions* zur Anwendung. Diese wurde so eingestellt, dass sie das 24 bit Audiosignal mit dem 40 bit Verstärkungsfaktor multipliziert und als Ergebnis ein 64 bit Wort ausgibt. Auf die interne Umwandlung der Komponente *LPM_MULT* von 64 bit auf 24 bit kann hier nicht zurückgegriffen werden, da das Ergebnis aus den 64 bit extrahiert werden muss. Daher wird die Umwandlung von 64 bit auf 24 bit durch die *Filter Unit* durchgeführt.

2.3.3.3. Coefficients Register

Das *Coefficients Register* hat die Aufgabe die Koeffizienten für die jeweilige Filterbank zur Verfügung zu stellen. Dabei ist der entsprechende Verstärkungsfaktor mit einzubeziehen. Das heißt, dass hier die Berechnung der Filterkoeffizienten stattfindet und diese dann dem *IIR Filter* übergeben werden.

Die Grundlage für die Berechnung der Koeffizienten liefern folgende Gleichungen⁵ für den verwendeten Peek-Filter:

$$b_0 = \frac{1 + \frac{V_0}{Q_\infty}K + K^2}{1 + \frac{1}{Q_\infty}K + K^2} \quad (4)$$

$$b_1 = \frac{2(K^2 - 1)}{1 + \frac{1}{Q_\infty}K + K^2} \quad (5)$$

$$b_2 = \frac{1 - \frac{V_0}{Q_\infty}K + K^2}{1 + \frac{1}{Q_\infty}K + K^2} \quad (6)$$

$$a_1 = \frac{2(K^2 - 1)}{1 + \frac{1}{Q_\infty}K + K^2} \quad (7)$$

$$a_2 = \frac{1 - \frac{1}{Q_\infty}K + K^2}{1 + \frac{1}{Q_\infty}K + K^2} \quad (8)$$

Da in der Implementierung nur noch der Verstärkungsfaktor V_0 sich ändern kann festgehalten werden, dass die Koeffizienten b_0 und b_2 zur Laufzeit zu berechnen sind. Alle anderen Koeffizienten können berechnet in eine LUT abgelegt werden. Sei nun zu klären wie die

⁵[7]

Konstante K und die Güte Q berechnet wird.

$$K = \tan\left(\pi \frac{f_c}{f_s}\right) \quad (9)$$

Dabei ist f_c die Cutoff-Frequenz und f_s die Sample Rate bzw. die Abtastfrequenz. Zur Berechnung der Cutoff-Frequenzen ist zunächst die Bandbreite zu berechnen. Diese ist logarithmisch zu bilden und ergibt sich aus den $n = 14$ Filterbänken und den Grenzfrequenzen des hörbaren Bereiches $f_h = 20 \text{ kHz}$ und $f_l = 20 \text{ Hz}$.

$$\log(B) = \log(f_h) - \log(f_l) \quad (10)$$

$$\log(B) = \frac{\log\left(\frac{f_h}{f_l}\right)}{n} \quad (11)$$

$$\log(B) = \frac{\log\left(\frac{20000 \text{ Hz}}{20 \text{ Hz}}\right)}{14} \quad (12)$$

$$\log(B) = \frac{3}{14} \approx 0,2143 \quad (13)$$

Mit dem berechneten Wert für die Bandbreite können nun die Cutoff-Frequenzen berechnet werden.

$$\log(f_{cn}) = \log(f_{c_{-1}}) + \log(B) \quad (14)$$

$$f_{cn} = 10^{\log(f_{c_{-1}}) + \log(B)} \quad (15)$$

$$\text{mit } n = \{1, 2, 3 \dots 14\} \text{ und } f_{c1} = 20 \text{ Hz}$$

Aus den berechneten Cutoff-Frequenzen ergeben sich die Mittenfrequenzen wie folgt.

$$f_{0n} = \sqrt{f_{cn} \cdot f_{c_{n+1}}} \quad (16)$$

$$\text{mit } n = \{1, 2, 3 \dots 14\} \text{ und } f_{c14+1} = 20 \text{ kHz}$$

Für die Berechnung der Koeffizienten fehlt noch die Berechnung der Güte Q_∞ .

$$Q = \frac{f_{0n}}{2|(f_{0n} - f_{cn})|} \quad (17)$$

Eine Tabelle mit den berechneten Cuttloff- und Mittenfrequenzen sowie den Güten kann dem Anhang A.4 entnommen werden.

Wie bereits festgestellt wurde, müssen die Koeffizienten b_1 , a_1 und a_2 nicht zur Laufzeit berechnet werden, sondern werden in der Komponente *IIR Filter* mittels LUT abgelegt.

Bei den Koeffizienten b_0 und b_2 fällt auf, dass sie sich lediglich in einem Vorzeichen unterscheiden (siehe Gleichungen (4) und (6)). Dies lässt Raum, die Berechnung zur Laufzeit zu vereinfachen, indem die Koeffizienten zu drei einzelnen Konstanten c_1 , c_2 und c_3 zerlegt werden. Dies führt dazu, dass die Konstanten für die Berechnung beider Koeffizienten verwendet werden können. Die Koeffizienten werden so wie folgt berechnet.

$$b_0 = c_1 + V_0 \cdot c_2 + c_3 \quad (18)$$

$$b_2 = c_1 - V_0 \cdot c_2 + c_3 \quad (19)$$

Die Konstanten ergeben sich aus der Zerlegung der Brüche in drei einzelne Brüche der Gleichungen (4) und (6).

$$b_0 = \frac{1 + \frac{V_0}{Q_\infty} K + K^2}{1 + \frac{1}{Q_\infty} K + K^2} = \frac{1}{1 + \frac{1}{Q_\infty} K + K^2} + \frac{\frac{V_0}{Q_\infty} K}{1 + \frac{1}{Q_\infty} K + K^2} + \frac{K^2}{1 + \frac{1}{Q_\infty} K + K^2} \quad (20)$$

$$b_2 = \frac{1 - \frac{V_0}{Q_\infty} K + K^2}{1 + \frac{1}{Q_\infty} K + K^2} = \frac{1}{1 + \frac{1}{Q_\infty} K + K^2} - \frac{\frac{V_0}{Q_\infty} K}{1 + \frac{1}{Q_\infty} K + K^2} + \frac{K^2}{1 + \frac{1}{Q_\infty} K + K^2} \quad (21)$$

Es ist also deutlich zu sehen, dass die Konstanten für beide Gleichungen eingesetzt werden können und nur bei der Berechnung der Koeffizienten einmal eine Subtraktion und einmal eine Addition durchzuführen ist. Zieht man nun noch den Verstärkungsfaktor V_0 vor den Bruch, so erhält man.

$$b_0 = \frac{1}{1 + \frac{K}{Q_\infty} + K^2} + V_0 \frac{\frac{K}{Q_\infty}}{1 + \frac{K}{Q_\infty} + K^2} + \frac{K^2}{1 + \frac{K}{Q_\infty} + K^2} = c_1 + V_0 \cdot c_2 + c_3 \quad (22)$$

$$b_2 = \frac{1}{1 + \frac{K}{Q_\infty} + K^2} - V_0 \frac{\frac{K}{Q_\infty}}{1 + \frac{K}{Q_\infty} + K^2} + \frac{K^2}{1 + \frac{K}{Q_\infty} + K^2} = c_1 - V_0 \cdot c_2 + c_3 \quad (23)$$

Somit sind nun die Grundlagen zur Berechnung der Koeffizienten b_0 und b_2 gelegt.

Wird also vom *MMI* eine Änderung eines der Verstärkungsfaktoren eines Filters gemeldet, so werden die Koeffizienten b_2 und b_0 neu berechnet und dem *IIR Filter* übergeben. Die LUT's für die Koeffizienten b_1 , a_1 und a_2 sind im *IIR Filter* hinterlegt.

Die Multiplikation wird wieder durch die Komponente *LPM_MULT* der Quartus II Bibliothek *Megafunctions* gebildet. Wobei hier eine Multiplikation mit dem 40 bit großen Verstärkungsfaktor V_0 und der 64 bit großen Konstante c_2 stattfindet. Auch hier wird das 104 bit große Ergebnis dem *Coefficients Register* übergeben, sodass dieser das Ergebnis selber extrahieren kann. Dabei weisen die Koeffizienten eine Wortlänge von 64 bit auf.

Zunächst wird die Multiplikation des Verstärkungsfaktors V_0 mit der Konstanten c_2 durchgeführt. Anschließend werden die Konstanten addiert bzw. subtrahiert und somit der Koeffizient gebildet.

Diese Berechnung findet kontinuierlich statt und wird für die Koeffizienten der jeweiligen Filterbank nacheinander durchgeführt. Sind alle Koeffizienten berechnet, fängt dieser Vorgang wieder von vorne an.

2.3.3.4. IIR Filter

Die Bearbeitung der Audiodaten mit den Vorgaben des Benutzers erfolgt im *IIR Filter*. Wie der Name der Komponente bereits besagt, kommt ein IIR-Filter zur Anwendung. Die Komponente *IIR Filter* besteht im Grunde nur aus einem IIR-Filter 2. Ordnung. Dieser wird über eine Schleife, bei der zu Beginn die Koeffizienten entsprechend ausgetauscht werden, zu einer Kombination aus 14 Filterbänken kombiniert.

Da es möglich sein soll den Filter bzw. die Koeffizienten jederzeit zu tauschen, wurde hier keine spezielle Filterstruktur wie z.B. *transponierte Struktur der Direktform II* o.ä. umgesetzt, sondern rein die Struktur nach der Standardformel.

$$y(n) = \sum_{k=0}^N b_k \cdot x(n - k) - \sum_{l=1}^N a_l \cdot y(n - l) \quad (24)$$

Im Fall des hier eingesetzten IIR Filters beträgt $N = 2$. Daraus ergibt sich.

$$y(n) = \sum_{k=0}^2 b_k \cdot x(n - k) - \sum_{l=1}^2 a_l \cdot y(n - l) \quad (25)$$

Über den Eingang x werden die Audiodaten dem *IIR Filter* zugeführt. Die im *Coefficients Register* berechneten Koeffizienten b_0 und b_2 sind den Eingängen $b0_1$ bis $b0_14$ und $b2_1$ bis $b2_14$ zu entnehmen. Dabei steht die Zahl 1 bis 14 für den jeweiligen Filter. Über den Eingang en wird die Übergabe der Audiodaten gesteuert. Die berechneten Audiodaten werden über den Ausgang y wieder ausgegeben. Die Verarbeitung wird mit 50 MHz durchgeführt. Der Takt liegt am Eingang clk_50MHz an.

Die Koeffizienten b_1 , a_1 und a_2 sind als Konstanten (constant) $b1_1 .. b1_14$, $a1_1 .. a1_14$ und $a2_1 .. a2_14$ definiert. Ihnen werden bei Start die entsprechenden Werte zugewiesen.

Die Werte $x(n - 1)$, $x(n - 2)$, $y(n - 1)$ und $y(n - 2)$ werden in den Signalen $x1_1 .. x1_14$, $x2_1 .. x2_14$, $y1_1 .. y1_14$ und $y2_1 .. y2_14$ gespeichert.

Die Komponente *IIR Filter* berechnet die gefilterten Audiodaten nach folgendem Schema.

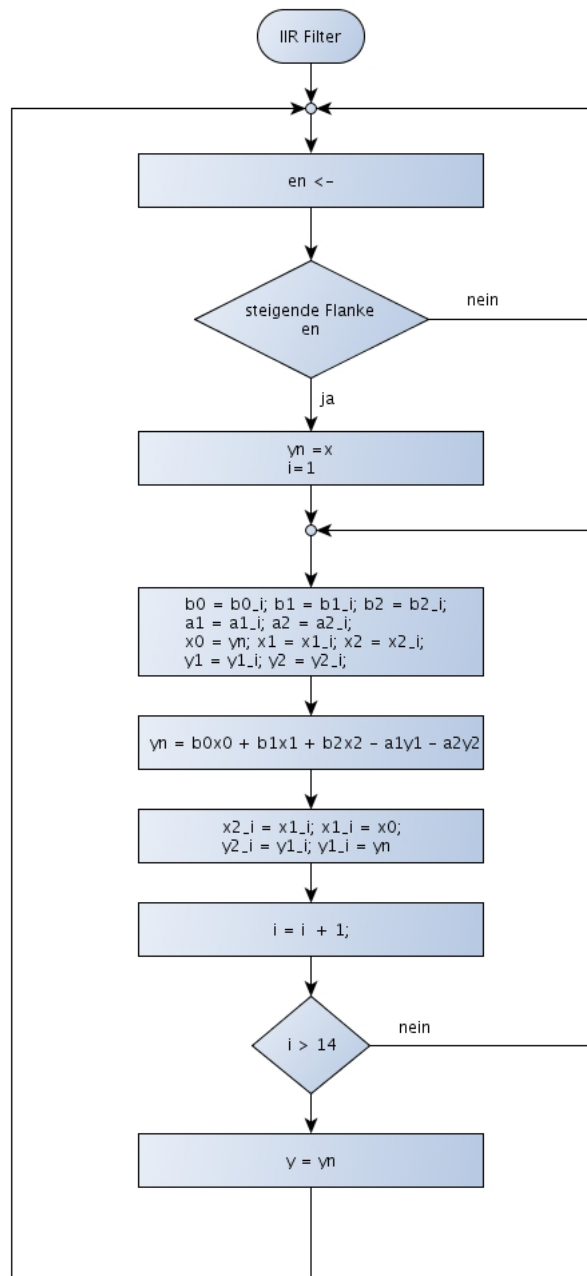


Abbildung 9: Programmablaufplan *IIR Filter*

Nach der Berechnung wird, wie im PAP zu sehen ist, das Ergebnis über den Ausgang y ausgegeben und die *Filter Unit* gibt es weiter an das *Audio Output Register*.

2.3.4. Audio Output Register

Das *Audio Output Register* hat die Aufgabe, die von der *Filter Unit* erzeugten Audiodaten zu übernehmen und zeitgerecht die einzelnen Bits an den *Audio Codec* zu übertragen.

Bei steigender Flanke an den Eingängen *enl* für den linken und *enr* für den rechten Kanal, werden die Daten an den Eingängen *dl* (linker Kanal) und *dr* (rechter Kanal) übernommen und in den Signalen *idl* (linker Kanal) und *idr* (rechter Kanal) gespeichert.

Durch die Änderung des Zustandes von logisch 0 auf logisch 1 und umgekehrt des Eingangssignals *daclrc* wird vom *Audio Codec* mitgeteilt, dass die nächsten Daten übermittelt werden können. Über den Endzustand logisch 1 oder logisch 0 wird bestimmt, ob die Daten des linken oder des rechten Kanals erwartet werden.

Um sicherzustellen, dass keine Daten übertragen werden, wenn neue Daten in die Signale *idl* und *idr* geschrieben werden, werden die Daten dieser Signale immer in die Signale *sdr* und *sdl* übernommen, während keine Datenübertragung für den entsprechenden Kanal zum *Audio Codec* stattfindet.

Die Datenübertragung der einzelnen Bits an den *Audio Codec* findet mit fallender Flanke des Eingangssignals *blk* statt. Dies ist der vom *Audio Codec* erzeugte 12 MHz Takt, wie im Abschnitt 2.3.1 beschrieben. Die einzelnen zu übertragenden Bits stammen vom MSB der Signale *sdr* und *sdl*. Nach der Übertragung wird ein Linksschieben durchgeführt, sodass im MSB immer genau das Bit steht, welches gerade übertragen werden soll.

Damit auch nur genau 24 Bit übertragen werden, laufen die Zähler *counter_left* für den linken und *counter_right* für den rechten Kanal mit. Diese Zähler werden immer dann zurückgesetzt, wenn keine Übertragung für den entsprechenden Kanal stattfindet.

Die Daten werden über den Ausgang *dadat* an den *Audio Codec* übermittelt. Die Übertragung findet ebenfalls im *Left Justified Mode* statt⁶.

⁶siehe Abbildung 8 im Abschnitt 2.3.2 auf Seite 17.

3. Test

Nach der Implementation wurden verschiedene Tests durchgeführt. Zunächst wurde auditiv geprüft, also ob Änderungen zu hören sind. Dies war der Fall und auch im erwarteten Bereich. Da der auditive Test ein sehr subjektiver Test ist und es hier auf die Testperson ankommt, wurde dieser Test als nicht ausreichend eingestuft. Deshalb wurde mit dem Audio-Analyzer der Amplitudengang gemessen und die Ergebnisse mit den Berechneten verglichen.

3.1. Messung des Amplitudengangs mit dem Audio-Analyzers

Die Messung mit dem Audio-Analyzer im Labor für Audiotechnik, wurden für jeden Filter einzeln durchgeführt. So wurde am Equalizer nacheinander für den zu messenden Filter 6dB Verstärkung und für die anderen Filter 0dB Verstärkung eingestellt.

Bei einer ersten Messung musste festgestellt werden, dass die Filter ein starkes Tiefpassverhalten aufzeigten. Die Ursache konnte im eingeschalteten De-Emphasis-Filter des *Audio Codec* ermittelt werden. Nach Abschalten des De-Emphasis-Filter konnte, bei einer weiteren Messung, der Amplitudenverlauf eines Peak-Filters ermittelt werden. Allerdings musste festgestellt werden, dass die berechneten Mittenfrequenzen mit den Gemessenen nicht übereinstimmten. Des Weiteren wurde festgestellt, dass der eingestellte Verstärkungsfaktor nicht mit dem Gemessenen übereinstimmt. Weiterhin konnte erkannt werden, dass die Bandbreiten zu groß waren. Die Ursachen für die Fehler konnten in der Güte Q und der Konstante K ermittelt werden. Hier wurde zur Berechnung der Güte Q die logarithmische statt der linearen Bandbreite und zur Berechnung der Konstante K statt der Mittenfrequenzen die Cutoff-Frequenzen verwendet. Einen Vergleich der Messergebnisse zwischen den berechneten und gemessenen Amplitudengängen kann dem Anhang [A.5.1](#) auf Seite [A8](#) entnommen werden.

Nach der Neuberechnung der Koeffizienten und der Integration in die Logikstruktur wurde eine zweite Messung durchgeführt. Bei dieser wurde im Gegensatz zur ersten Messung mit 20 dB Verstärkungsfaktor gemessen. Bei dieser Messung stimmten die Werte mit den Berechneten nahezu überein. Die Diagramme der zweiten Messung sind im Anhang [A.5.2](#) auf Seite [A22](#) einsehbar.

4. Fazit

Bei dem Projekt Equalizer wurde eine Logikstruktur mit getrennten Einheiten geschaffen, welche Unabhängig voneinander arbeiten. Es wurde eine Anzeige über 7-Segment-Anzeigen und einem LCD realisiert. Über einen Audio-Codec konnte ein Austausch von Audio-Daten implementiert werden, welche in einer weiteren Einheit einer Filterung und Lautstärke-regelung unterzogen werden.

Die Filterlogik besteht aus einem IIR-Filter welcher mit Koeffizienten, zur Realisierung eines Peak-Filters, gespeist wird. Es konnte mithilfe der im Test durchgeführten Messung nachgewiesen werden, dass eine prinzipielle Funktionalität der Peak-Filter vorhanden ist. Es musste aber auch die Erkenntnis gewonnen werden, dass an diesen Koeffizienten noch weiter gearbeitet werden muss, da nicht gänzlich die berechneten Werte erreicht wurden. Des Weiteren wurde nachgewiesen, dass die Kaskadierung von 14 Filterelementen voll funktions-fähig ist.

Da die Auslastung des FPGA bei etwa 20% der Logikelemente liegt, sind hier noch genügend Reserven vorhanden, um weitere Funktionen dem Equalizer hinzuzufügen. Allerdings ist darauf zu achten, dass die Anzahl der mitgelieferten Multiplizierer eine weitere Nutzung dieser nicht erlaubt. Sind bei einer Erweiterung weitere Multiplizierer gefordert, so müssen diese aus den Logikelementen des FPGA gebildet werden.

Zudem wäre zu prüfen, ob die hohen Wortlängen der Koeffizienten notwendig ist, oder ob diese verringert werden kann, um hier die Anzahl der Multiplizierer zu verringern.

Es ist vorstellbar mit einer weiteren Funktion die Nutzung des integrierten Flash-Speichers zu implementieren, um bestimmte Einstellwerte des Equalizers abzuspeichern. Diese Einstellwerte können zu einem späteren Zeitpunkt schnell wieder abgerufen werden um den Equalizer auf diese einzustellen.

Eine weitere Möglichkeit besteht in der Nutzung des SD-Kartenlesers, so könnten die Koeffizienten aus einer SD-Karte vom SD-Kartenleser ausgelesen werden, um die Filterstruktur zu ändern. Auch könnten hier Konfigurationsdaten hinterlegt werden, um beispielsweise die Filterstruktur in einem begrenzten Maße dynamisch änderbar zu gestalten.

Allerdings sollte zunächst darüber nachgedacht werden, den Equalizer in einer erweiterten Version mit einem Lautstärkebegrenzer zu versehen, welcher beispielsweise einfach nur den Überlauf in den Berechnungen der Filterung verhindert.

A. Anhang

A.1. Initialisierungswerte - *Audio Codec*

Register Address	Datenbits 15 bis 9	Datenbits 8 bis 0	Beschreibung
Reset	0001111	000000000	Alle Register werden auf Standardwerte zurückgesetzt.
Left Line In	0000000	000010111	<ul style="list-style-type: none">• Disable Simultaneous Load• Disable Mute• Volume Control - 0dB
Right Line In	0000001	000010111	<ul style="list-style-type: none">• Disable Simultaneous Load• Disable Mute• Volume Control - 0dB
Left Headphone Out	0000010	001111001	<ul style="list-style-type: none">• Disable Simultaneous Load• Left Channel Zero Cross detect Disable• Volume Control - 0dB
Right Headphone Out	0000011	001111001	<ul style="list-style-type: none">• Disable Simultaneous Load• Right Channel Zero Cross detect Disable• Volume Control - 0dB

Analogue Audio Path Control	0000100	000010010	<ul style="list-style-type: none"> • Side Tone Attenuation - -6dB • Disable Side Tone • Select DAC • Disable Bypass • Line Input Select to ADC • Mic Input Mute Enable • Microphone Input Level Boost Disable
Digital Audio Path Control	0000101	000010010	<ul style="list-style-type: none"> • clear offset • Disable DAC Soft Mute • De-emphasis Disable • ADC High Pass Filter Enable
Power Down Control	0000110	000010010	<ul style="list-style-type: none"> • Power Off Disable • CLKOUT Power Down Disable • Oscillator Power Down Disable • Outputs Power Down Disable • DAC Power Down Disable • ADC Power Down Disable • Microphone Input an Bias Power Down Enable • Line Input Power Down Disable

Digital Audio Interface Format	0000111	000010010	<ul style="list-style-type: none"> • Don't invert BCLK • Enable Master Mode • DAC Left Right Clock Swap to Right Channel DAC Data Right • Right Channel DAC data when DACLRC low • Input Audio Data Bit Length Select to 24 bits • Audio Data Format Select to MSB-First; left justified
Sampling Control	0001000	000000001	<ul style="list-style-type: none"> • CLOCKOUT is Core Clock • Core Clock is MCLK • ADC and DAC sample rate 48 kHz; MCLK = 12 MHz • Base Over-Sampling Rate = $250f_s$ • USB mode select
Active Control	0001001	000000001	<ul style="list-style-type: none"> • Activate Interface

Tabelle 6: Initialisierungswerte - *Audio Codec*

A.2. I²C Master - State Machine Diagramm

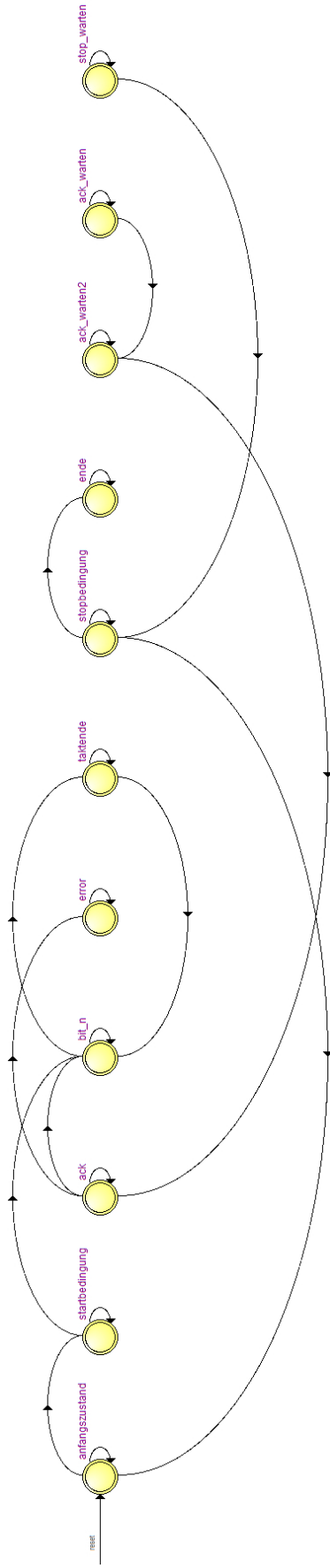


Abbildung 10: I²C Master - State Machine Diagramm

A.3. Look-Up-Table für Komponente db2linear

Das Modul db2linear wandelt über eine LUT den logarithmischen Verstärkungsfaktor in den linearen Verstärkungsfaktor um. Die vorberechneten Werte ergeben sich aus folgender Gleichung:

$$V_0|_{linear} = 10^{\frac{V_0|_{dB}}{20}} \quad (26)$$

Die Werte sind in der entsprechenden VHDL-Datei als Hexadezimalwerte abgelegt. Die Wortbreite beträgt 40 bit. Das Zahlenformat ist das vorzeichenbehaftete Zweierkomplement als Festkommawert mit fünf Vorkommastellen und entsprechend 34 Nachkommastellen.

$V_0 _{dB}$	$V_0 _{dB_{HEX}}$	$V_0 _{linear}$	$V_0 _{linear_{HEX}}$
20	14	10,0000	5000000000
19	13	8,9125	474CD1B7F1
18	12	7,9433	3F8BD79D82
17	11	7,0795	38A2BACB45
16	10	6,3096	327A01A469
15	0F	5,6234	2CFCC01646
14	0E	5,0119	2818508600
13	0D	4,4668	23BC1478BB
12	0C	3,9811	1FD93C1F52
11	0B	3,5481	1C629405AB
10	0A	3,1623	194C583ADA
9	09	2,8184	168C0C59AB
8	08	2,5119	141857E9D4
7	07	2,2387	11E8E6A0D8
6	06	1,9952	0FF64C16AD
5	05	1,7783	0E39EA8E2A
4	04	1,5849	0CADDC7B6A
3	03	1,4125	0B4CE07BF4
2	02	1,2589	0A12477C7E
1	01	1,1220	08F9E4CFB5
0	00	1,0000	0800000000
-1	FF	0,8913	0721482BFE
-2	FE	0,7943	065AC8C2F3
-3	FD	0,7079	05A9DF7ABA
-4	FC	0,6310	050C335D3D
-5	FB	0,5623	047FACCF07

-6	FA	0,5012	04026E73CC
-7	F9	0,4467	0392CED8DF
-8	F8	0,3981	032F52CFEE
-9	F7	0,3548	02D6A866F7
-10	F6	0,3162	0287A26C49
-11	F5	0,2818	0241346F5D
-12	F4	0,2512	02026F30FB
-13	F3	0,2239	01CA7D767C
-14	F2	0,1995	0198A13577
-15	F1	0,1778	016C310E37
-16	F0	0,1585	0144960C57
-17	EF	0,1413	012149A5FE
-18	EE	0,1259	0101D3F2D9
-19	ED	0,1122	00E5CA14C5
-20	EC	0,1000	00CCCCCCCC
-21	EB	0,0891	00B6873799
-22	EA	0,0794	00A2ADAD18
-23	E9	0,0708	0090FCBF79
-24	E8	0,0631	008138561F
-25	E7	0,0562	00732AE180
-26	E6	0,0501	0066A4A52E
-27	E5	0,0447	005B7B15AF
-28	E4	0,0398	00518847FE
-29	E3	0,0355	0048AA70B2
-30	E2	0,0316	0040C3713A
-31	E1	0,0282	0039B87189
-32	E0	0,0251	00337184E5
-33	DF	0,0224	002DD958A6
-34	DE	0,0200	0028DCEBBF
-35	DD	0,0178	00246B4E38
-36	DC	0,0158	00207567A2
-37	DB	0,0141	001CEDC3CC
-38	DA	0,0126	0019C86515
-39	D9	0,0112	0016FA9BAD
-40	D8	0,0100	00147AE147

A.4. Cutoff-Frequenzen f_{cn} , Mittenfrequenzen f_{0n} , Güte Q_∞

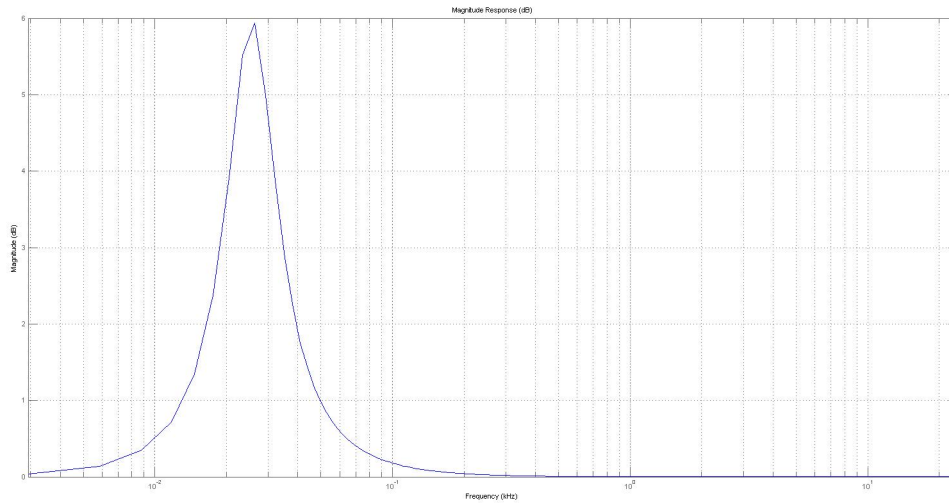
n	f_{cn} in Hz	f_{0n} in Hz	Güte Q
1	20	26	2,17
2	33	42	2,34
3	54	69	2,30
4	88	112	2,34
5	144	184	2,3
6	236	302	2,29
7	386	494	2,29
8	632	809	2,29
9	1036	1326	2,29
10	1697	2171	2,29
11	2779	3557	2,29
12	4552	5825	2,29
13	7455	9541	2,29
14	12211	15627	2,29

Tabelle 8: Cutoff-Frequenzen f_{cn} , Mittenfrequenzen f_{0n} , Güte Q_∞

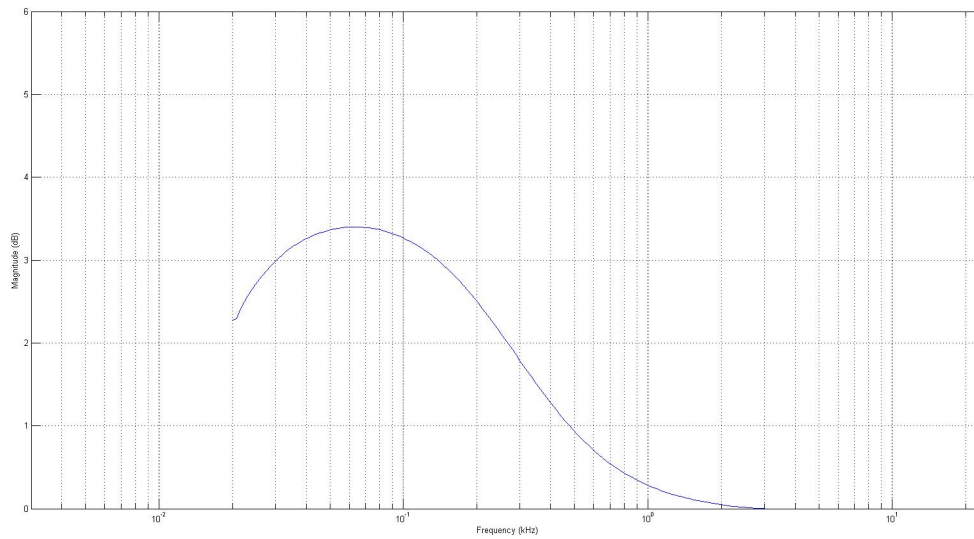
Zur Anwendung kamen die Gleichungen (15) auf Seite 20 zur Berechnung der Cutoff-Frequenzen, (16) auf Seite 20 zur Berechnung der Mittenfrequenzen und die Gleichung (17) auf Seite 20 zur Berechnung der Güte.

A.5. Messergebnisse

A.5.1. Ergebnis der 1. Messung

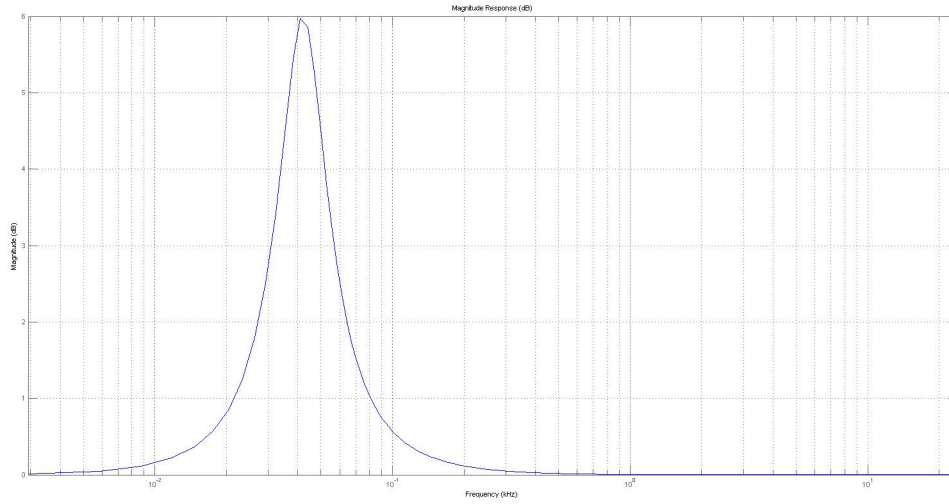


(a) berechnet

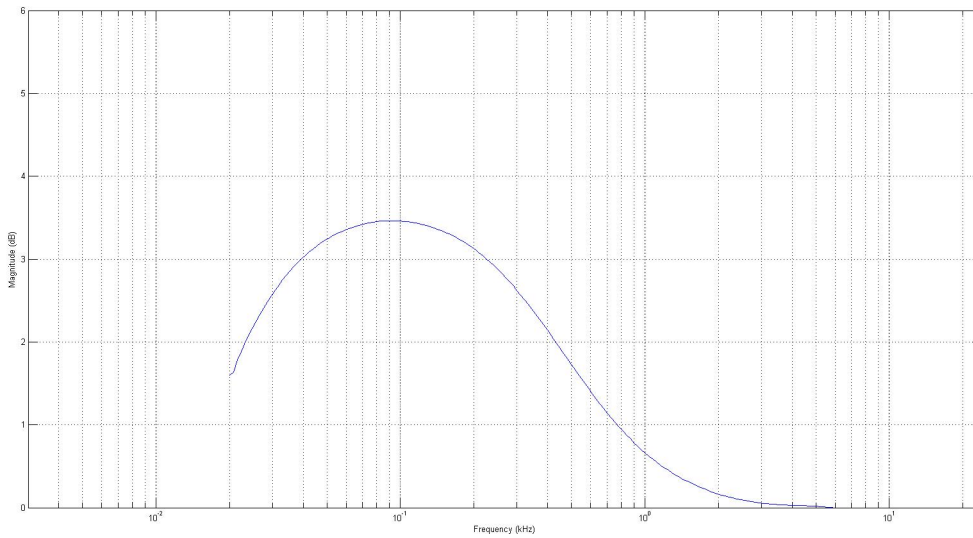


(b) gemessen

Abbildung 11: Amplitudengang Filter 1

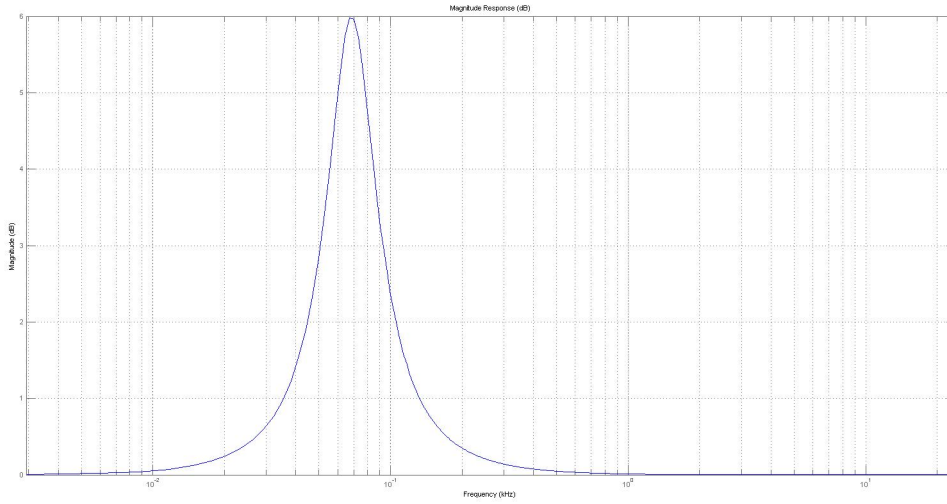


(a) berechnet

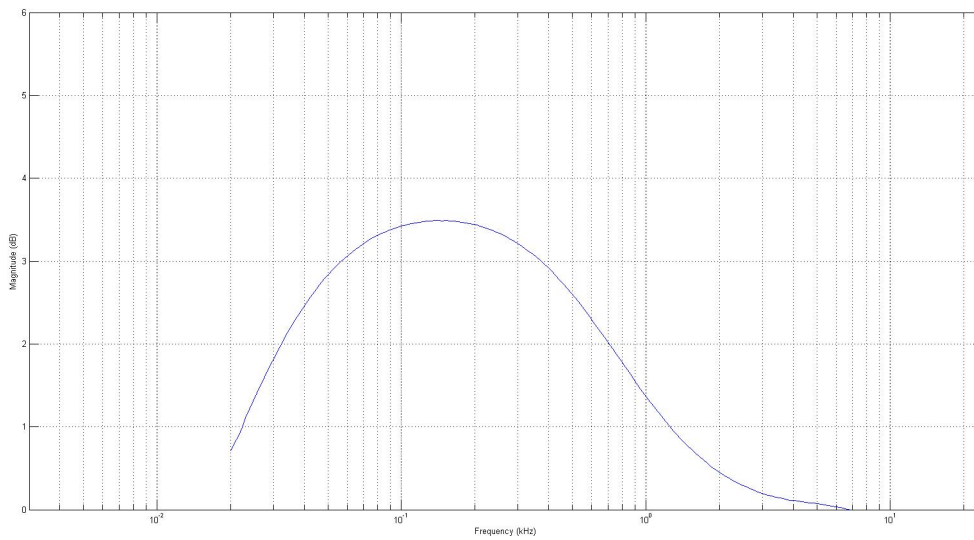


(b) gemessen

Abbildung 12: Amplitudengang Filter 2

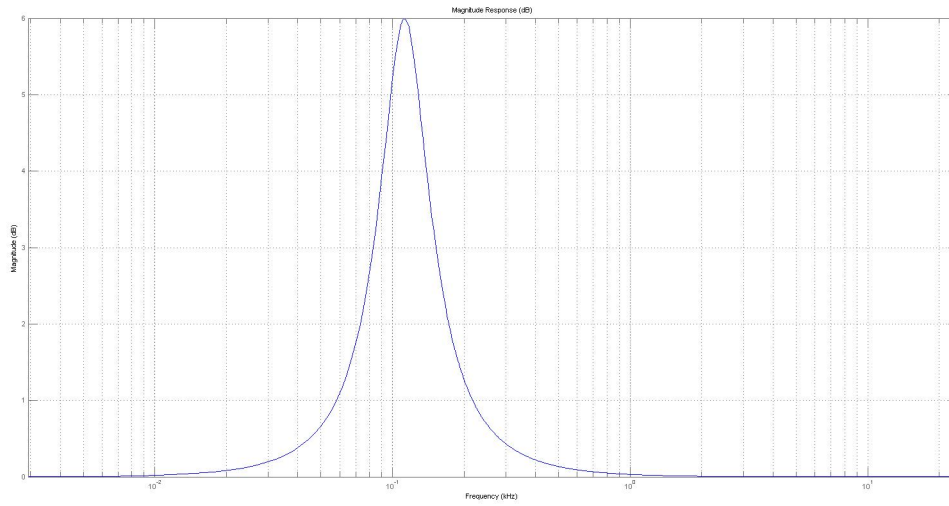


(a) berechnet

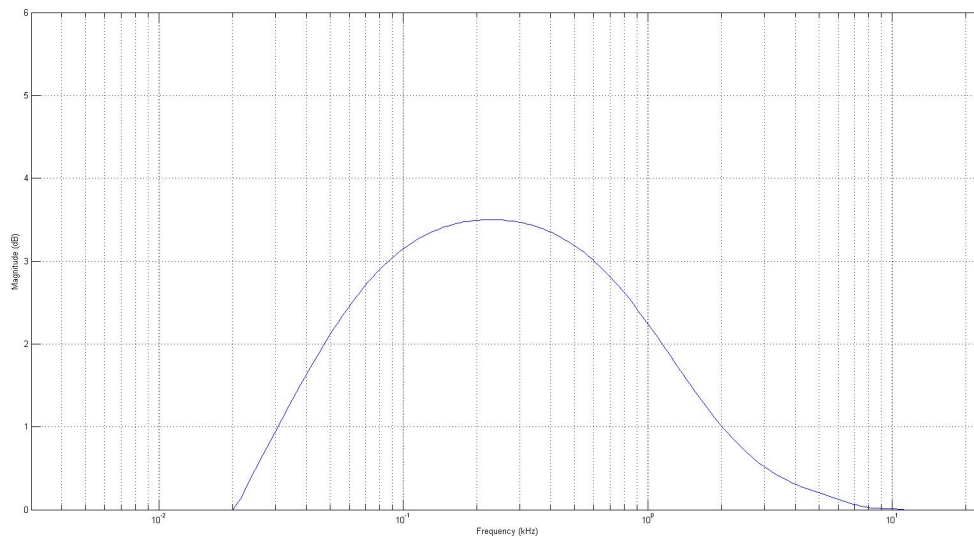


(b) gemessen

Abbildung 13: Amplitudengang Filter 3

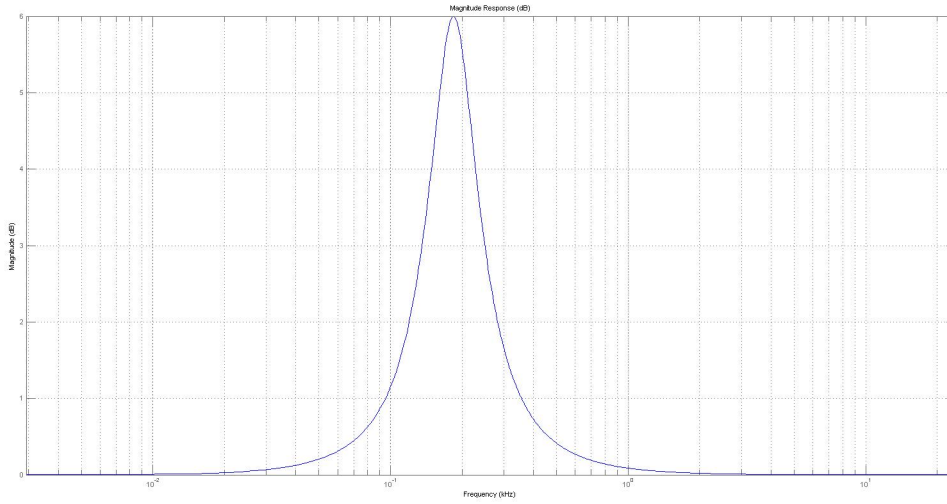


(a) berechnet

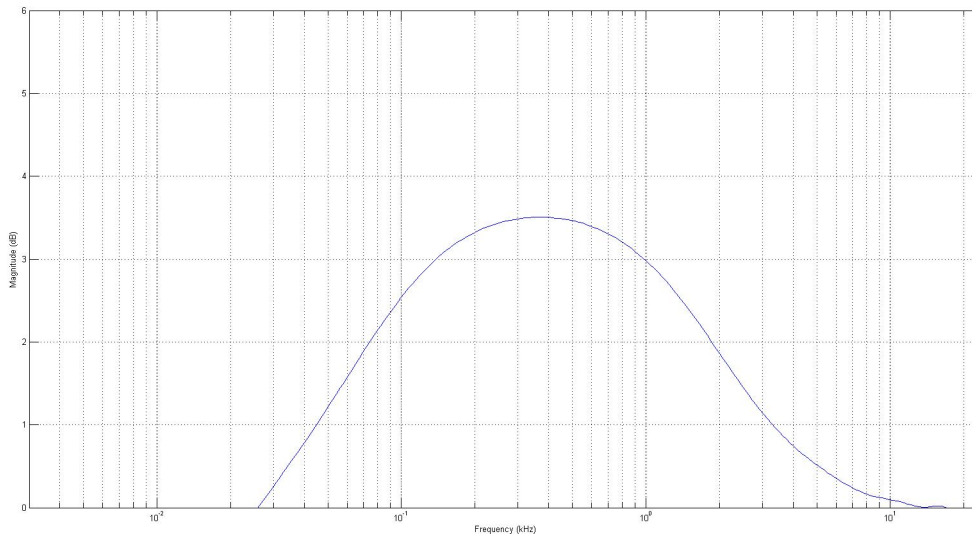


(b) gemessen

Abbildung 14: Amplitudengang Filter 4

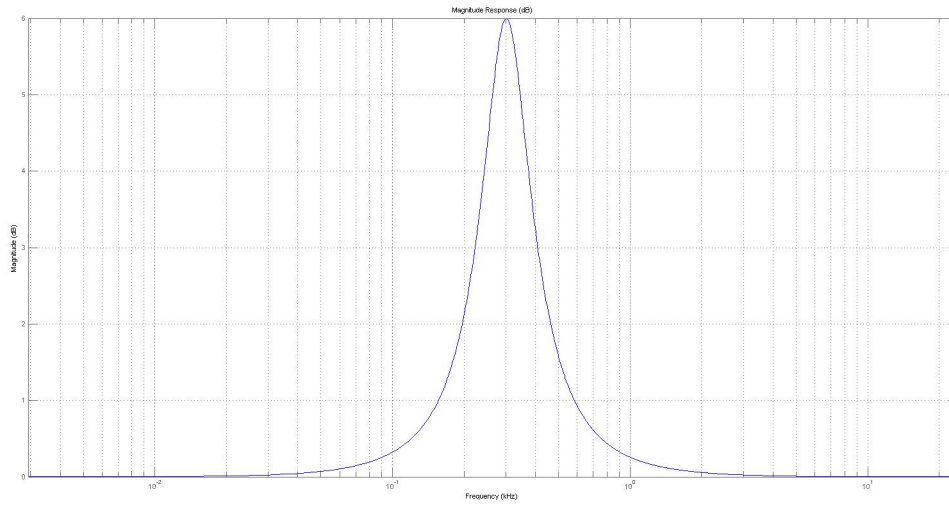


(a) berechnet

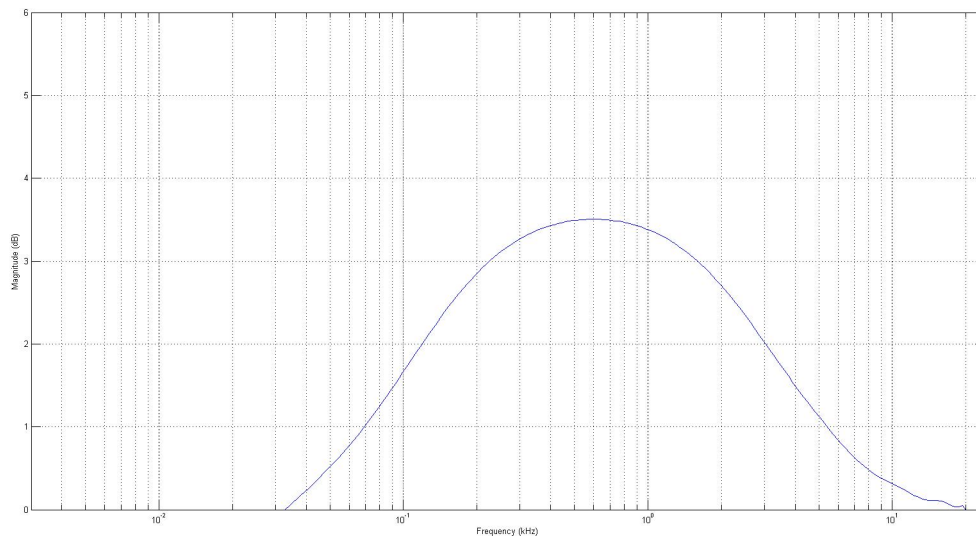


(b) gemessen

Abbildung 15: Amplitudengang Filter 5

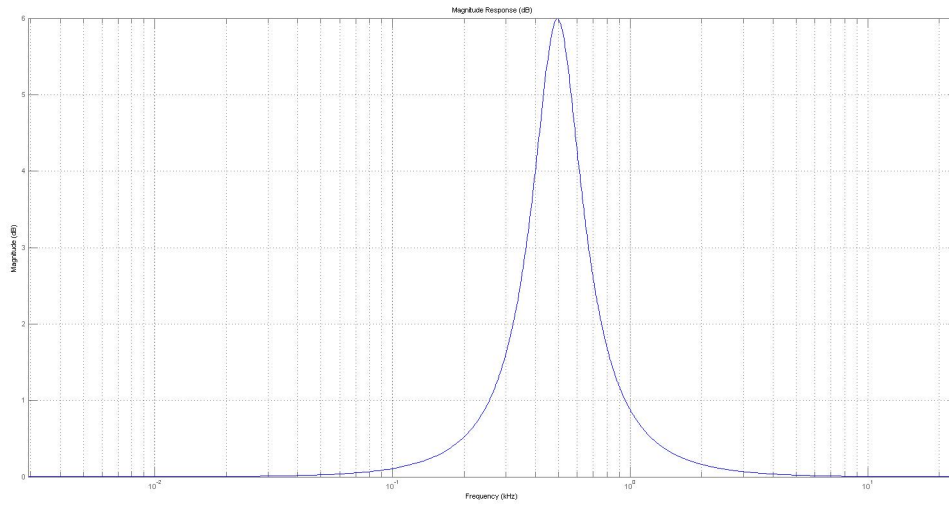


(a) berechnet

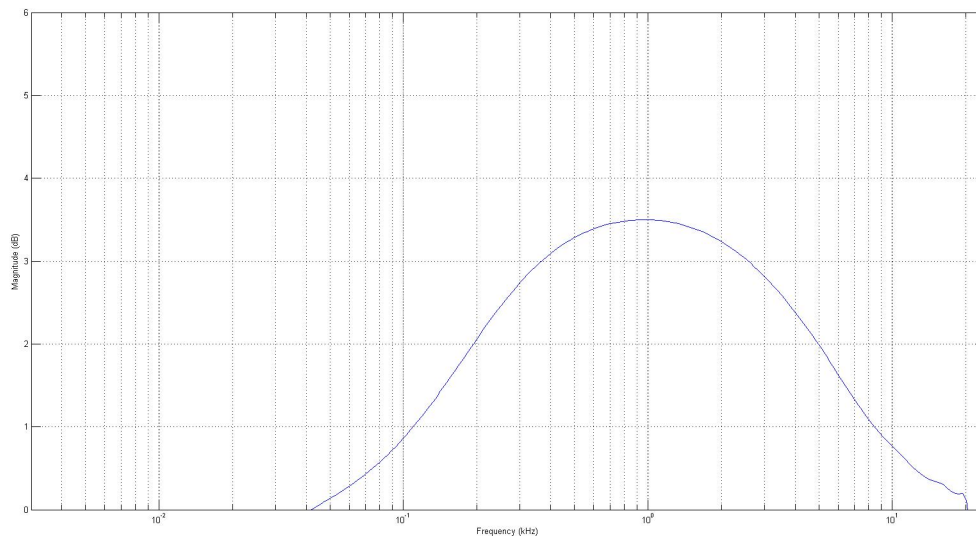


(b) gemessen

Abbildung 16: Amplitudengang Filter 6

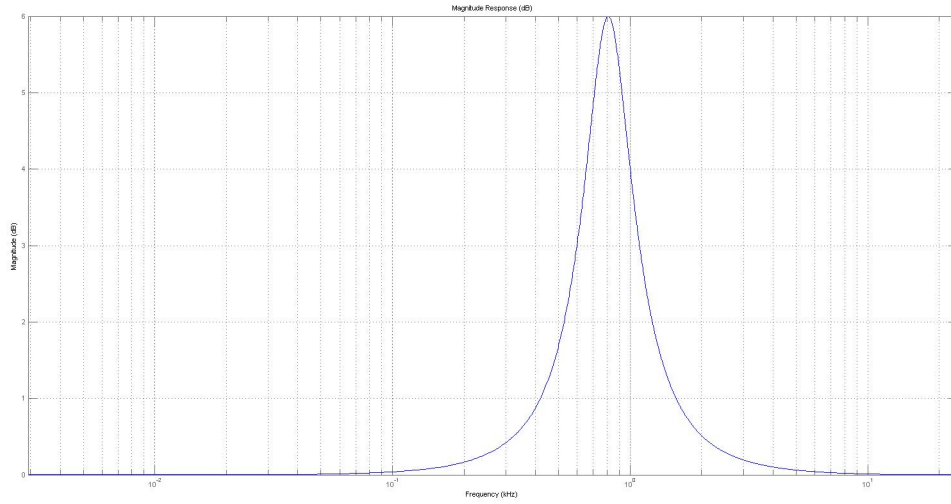


(a) berechnet

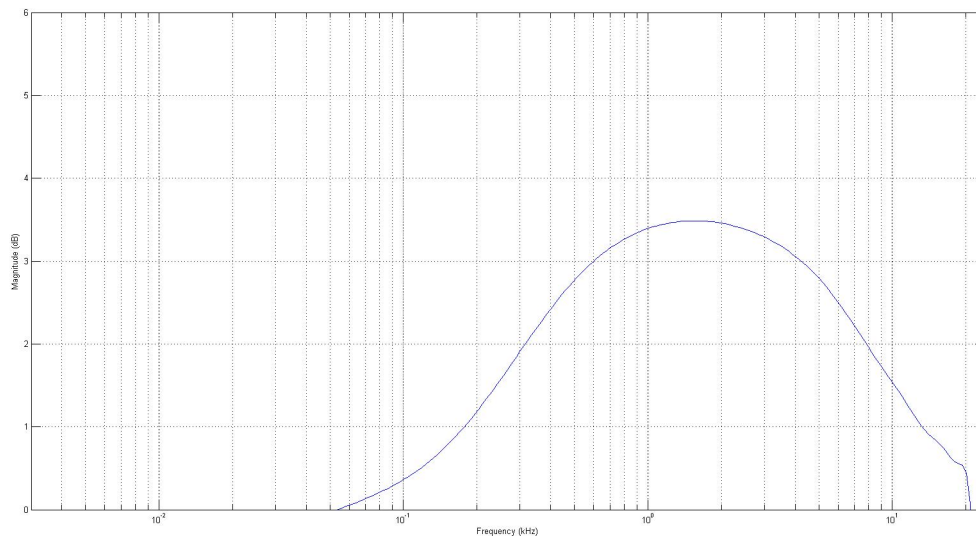


(b) gemessen

Abbildung 17: Amplitudengang Filter 7

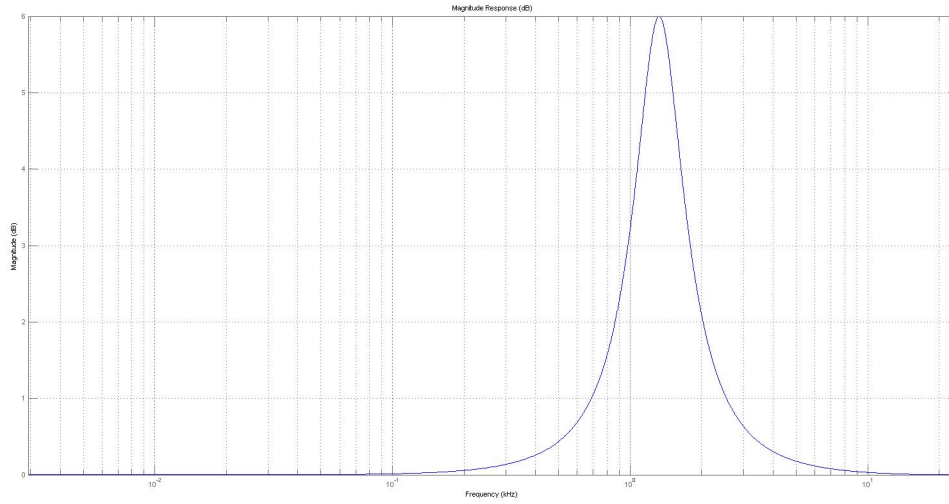


(a) berechnet

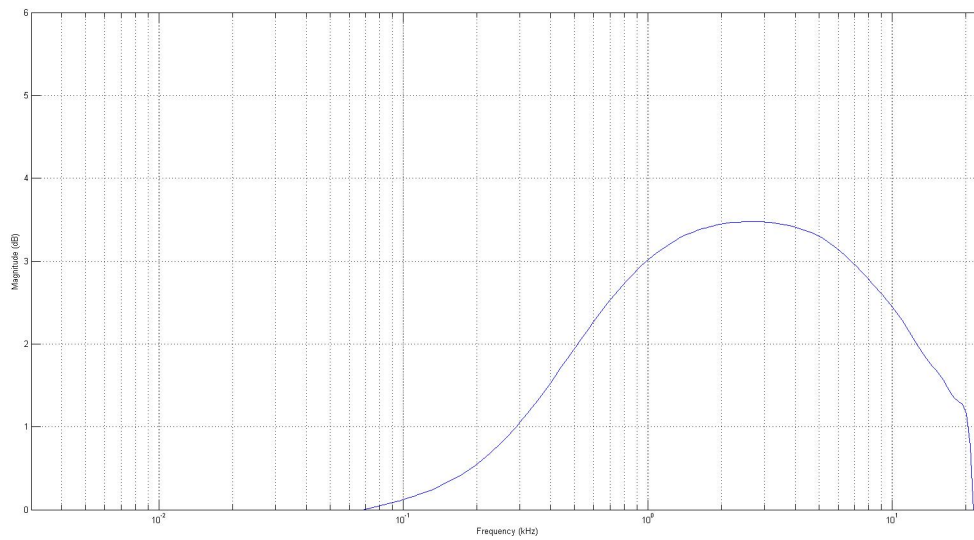


(b) gemessen

Abbildung 18: Amplitudengang Filter 8

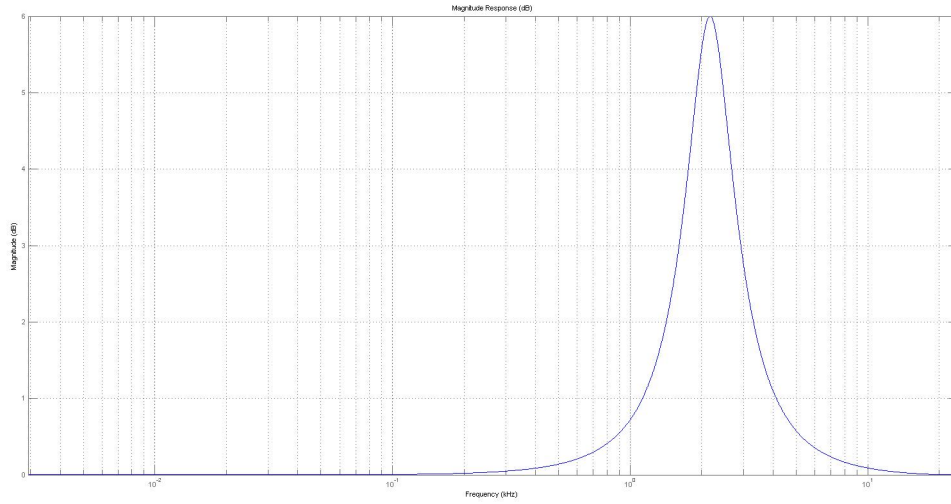


(a) berechnet

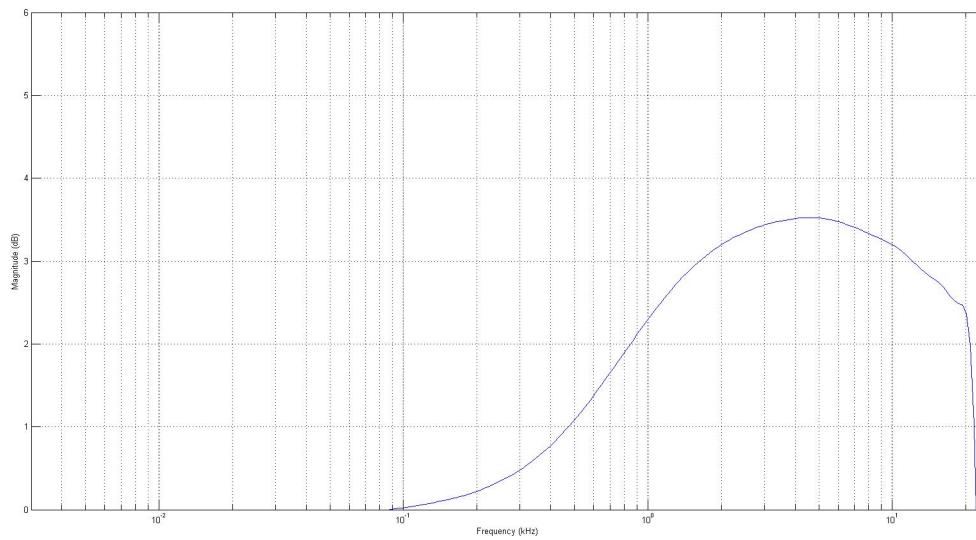


(b) gemessen

Abbildung 19: Amplitudengang Filter 9

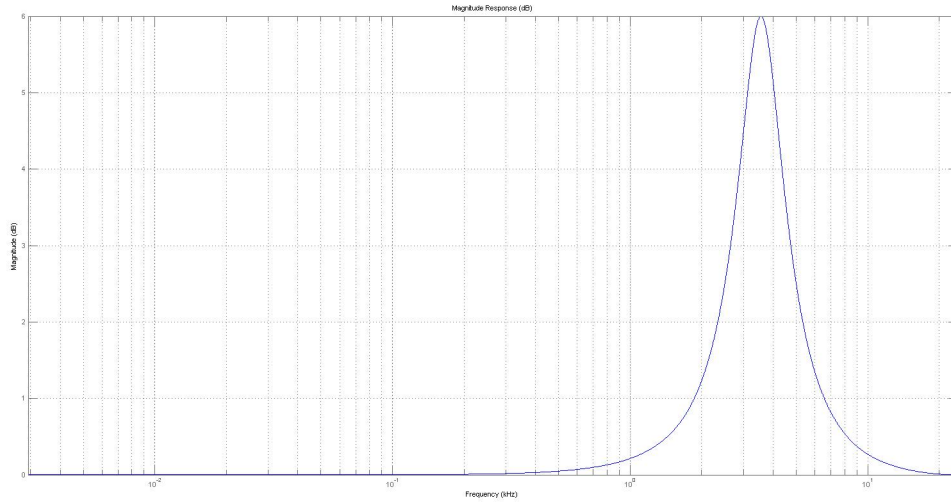


(a) berechnet

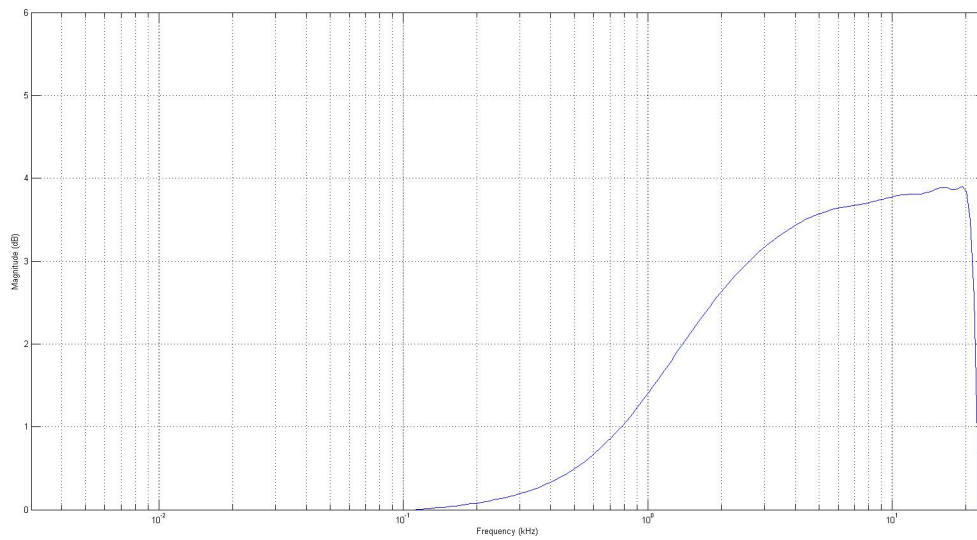


(b) gemessen

Abbildung 20: Amplitudengang Filter 10

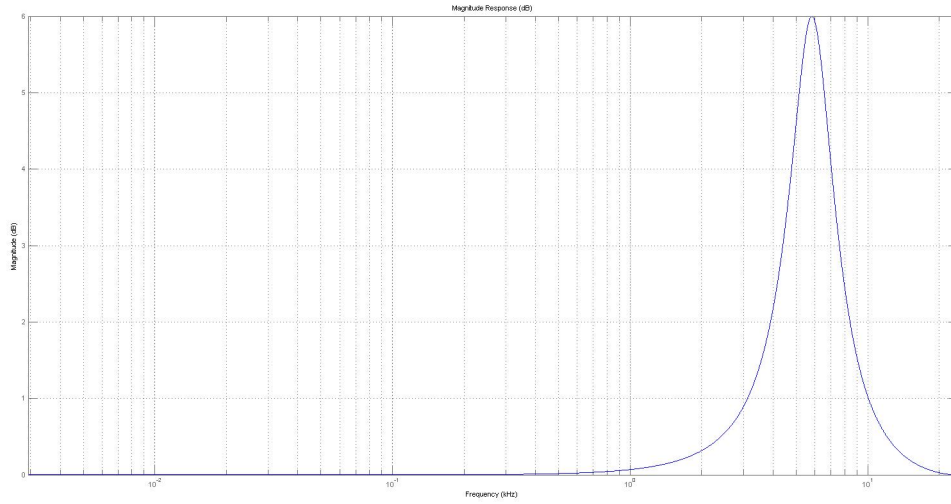


(a) berechnet

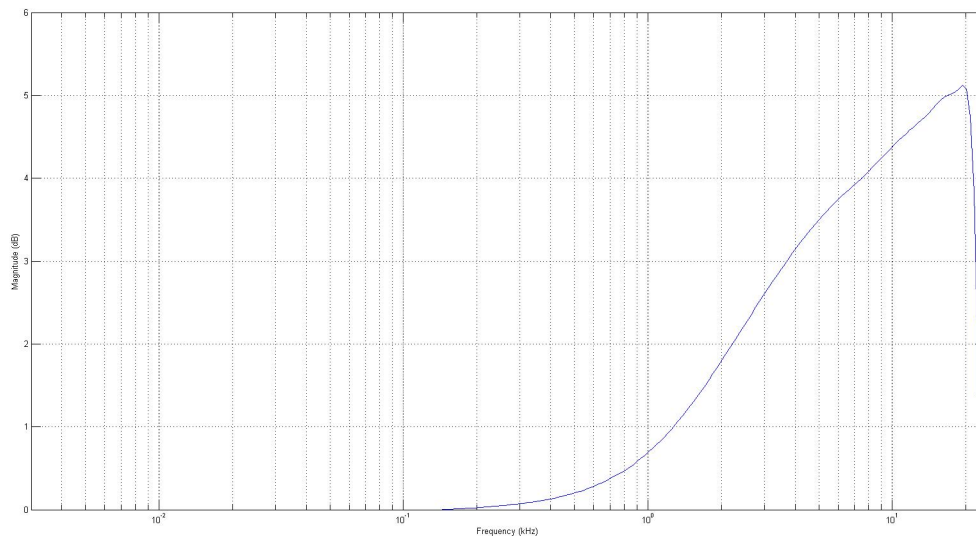


(b) gemessen

Abbildung 21: Amplitudengang Filter 11

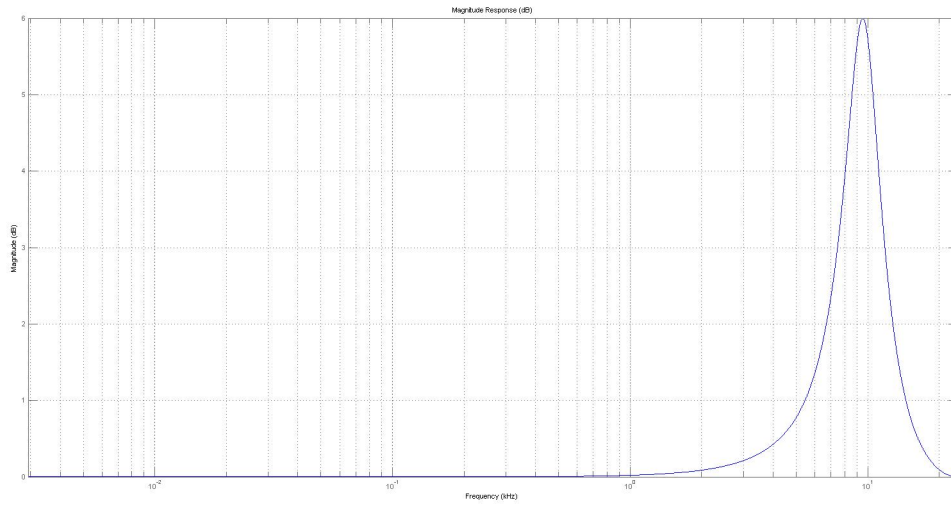


(a) berechnet

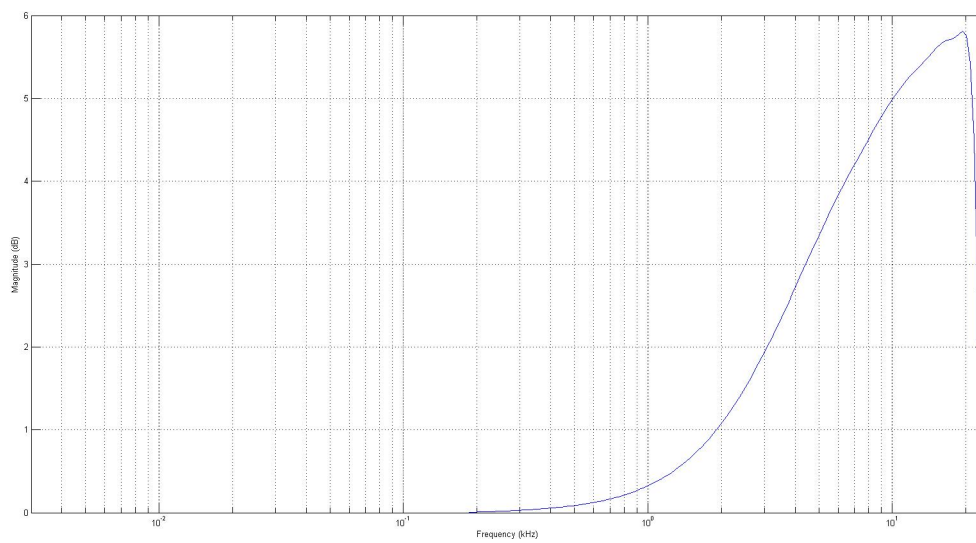


(b) gemessen

Abbildung 22: Amplitudengang Filter 12

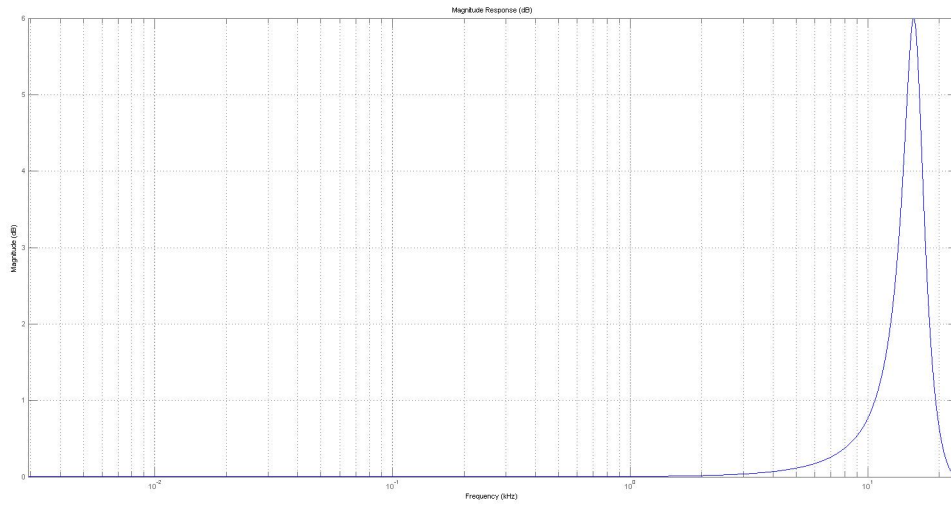


(a) berechnet

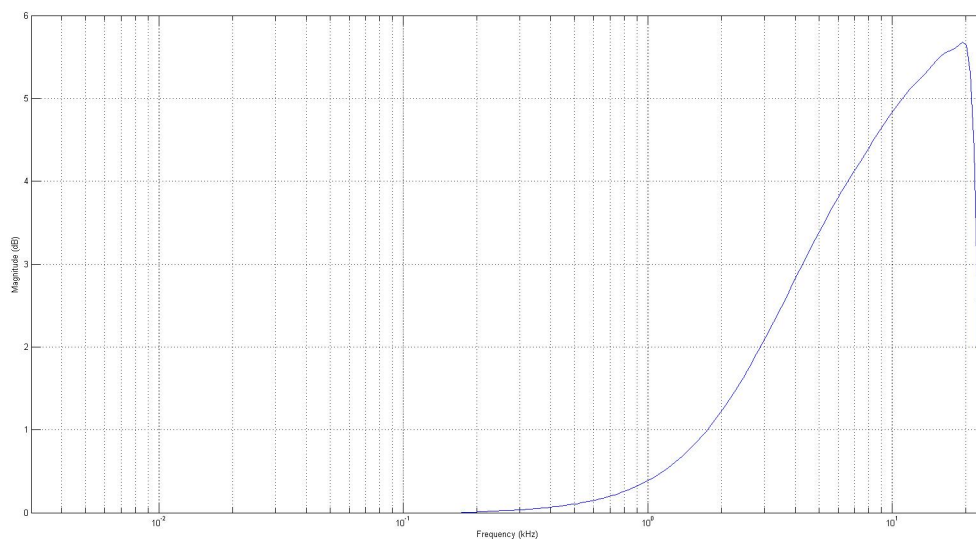


(b) gemessen

Abbildung 23: Amplitudengang Filter 13



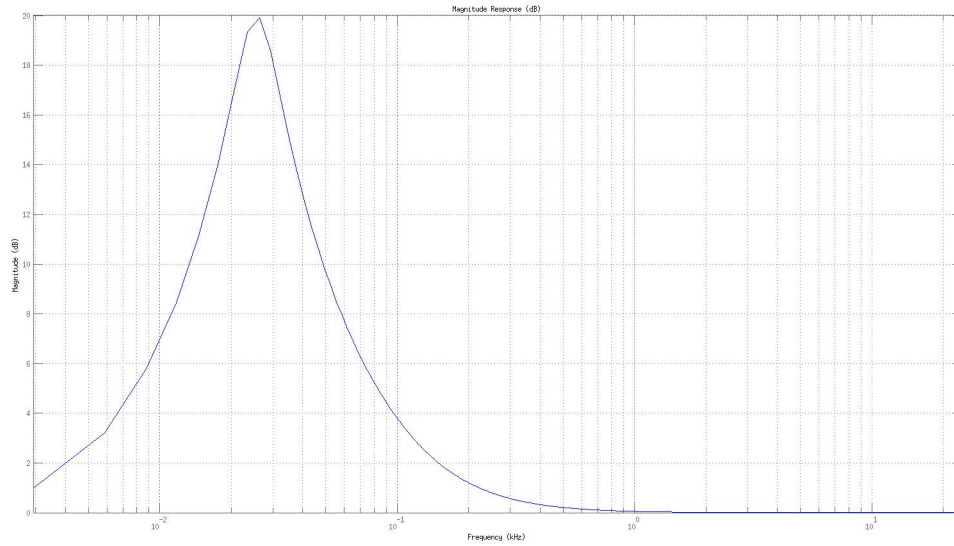
(a) berechnet



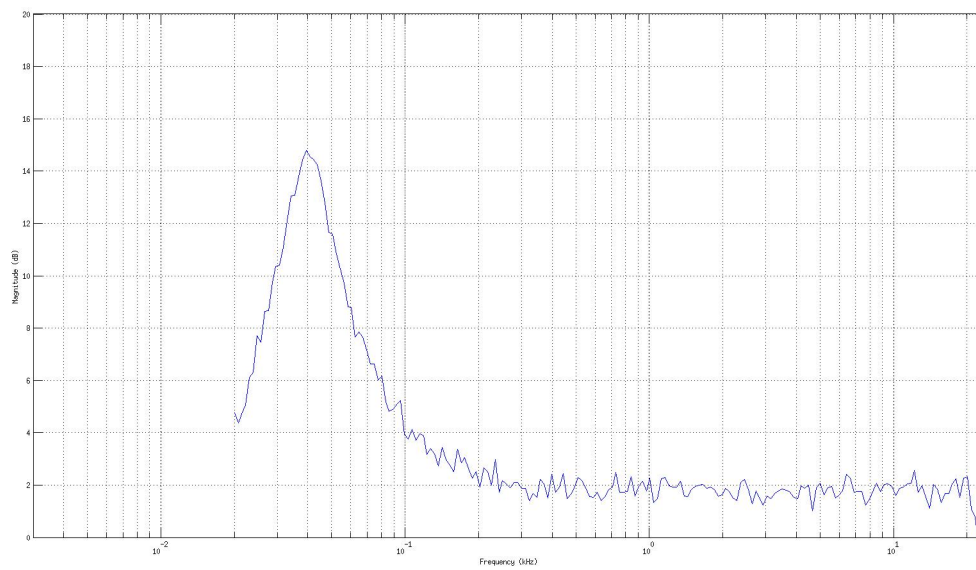
(b) gemessen

Abbildung 24: Amplitudengang Filter 14

A.5.2. Ergebnis 2. Messung

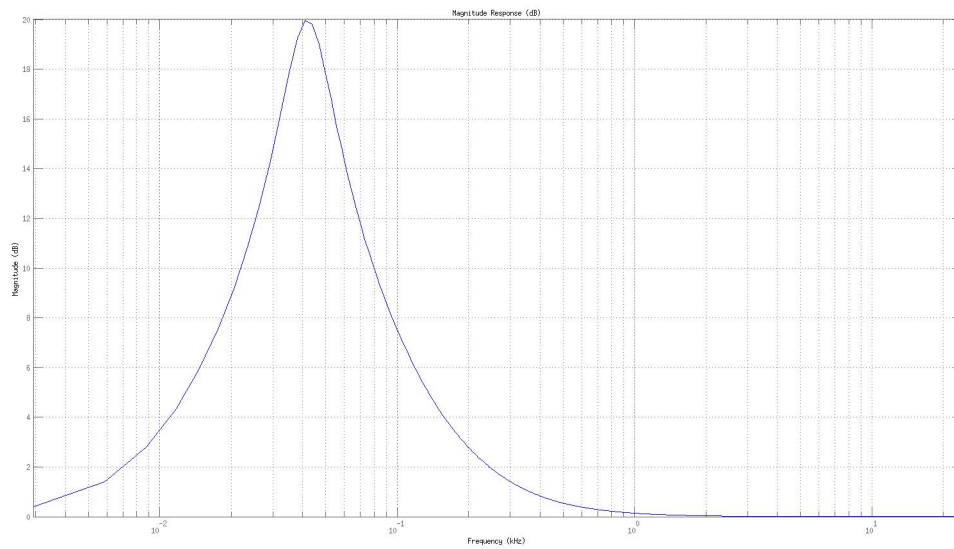


(a) berechnet

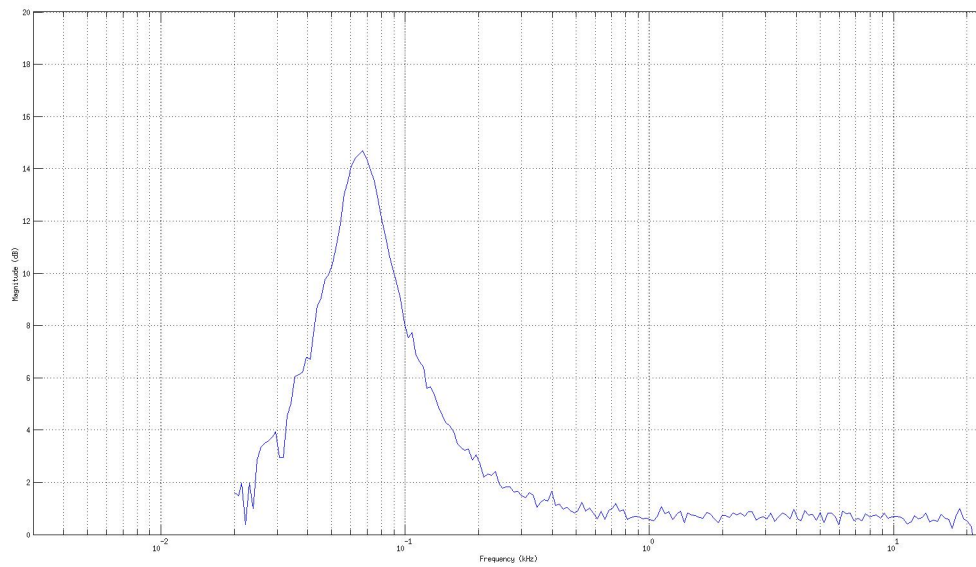


(b) gemessen

Abbildung 25: Amplitudengang Filter 1

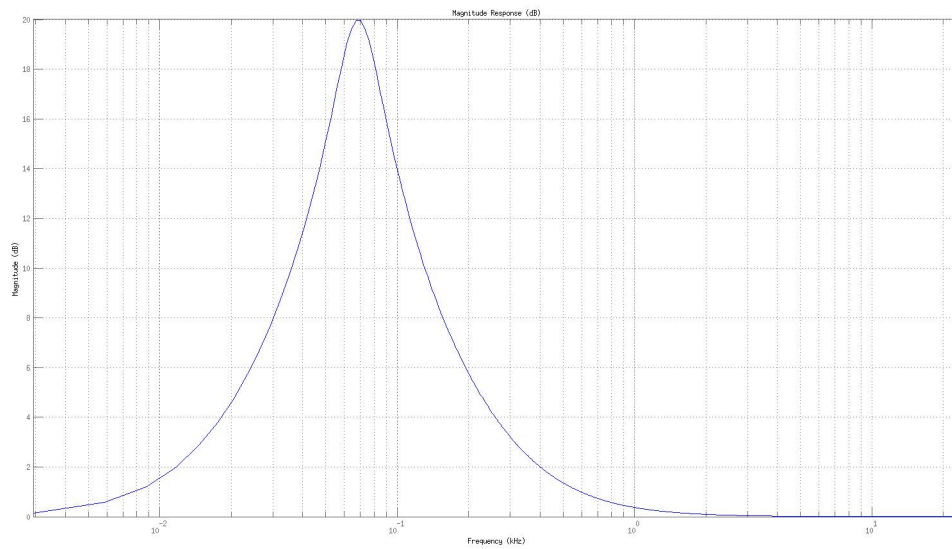


(a) berechnet

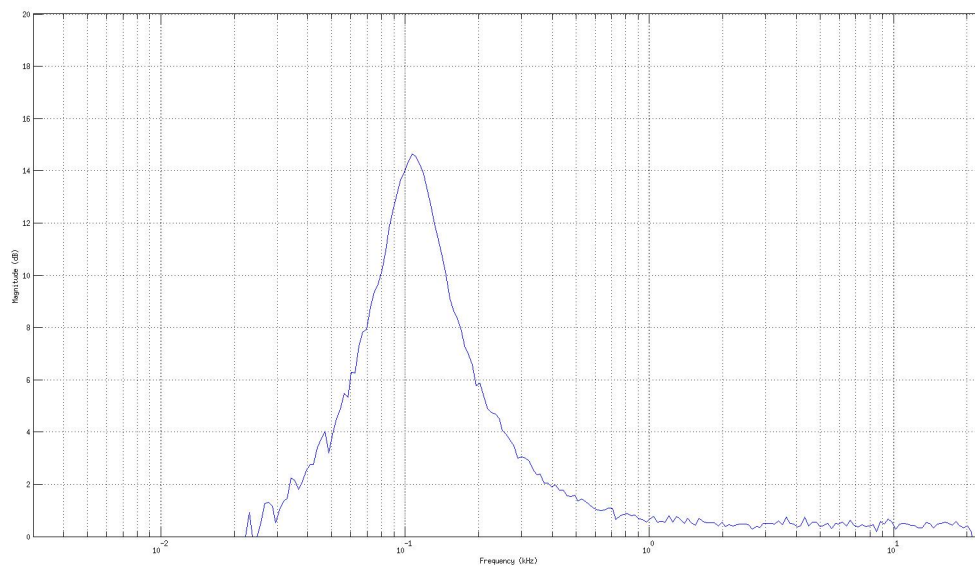


(b) gemessen

Abbildung 26: Amplitudengang Filter 2

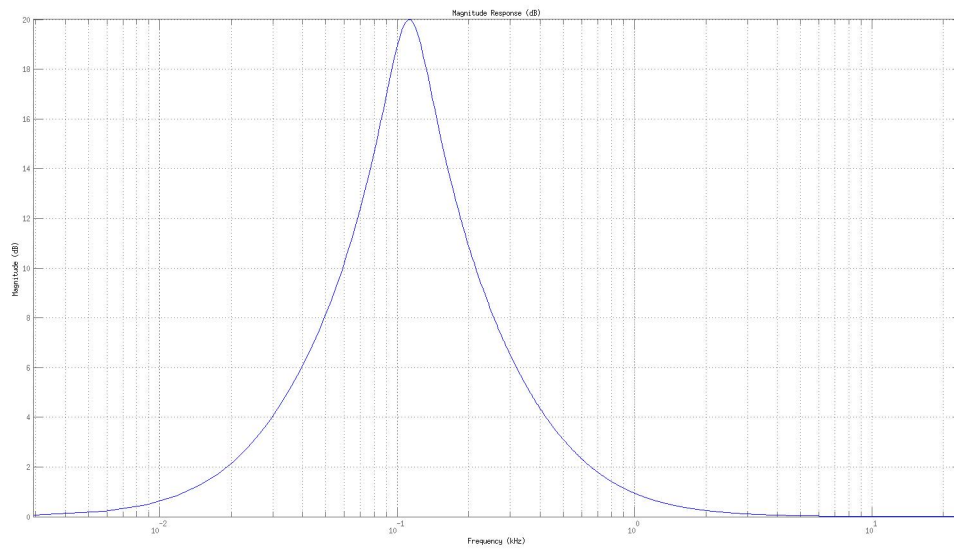


(a) berechnet

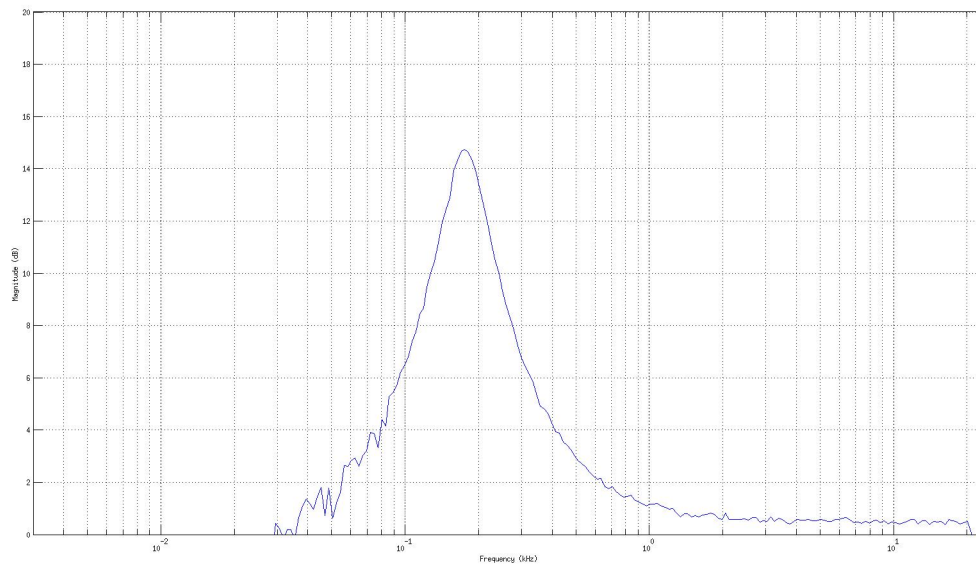


(b) gemessen

Abbildung 27: Amplitudengang Filter 3

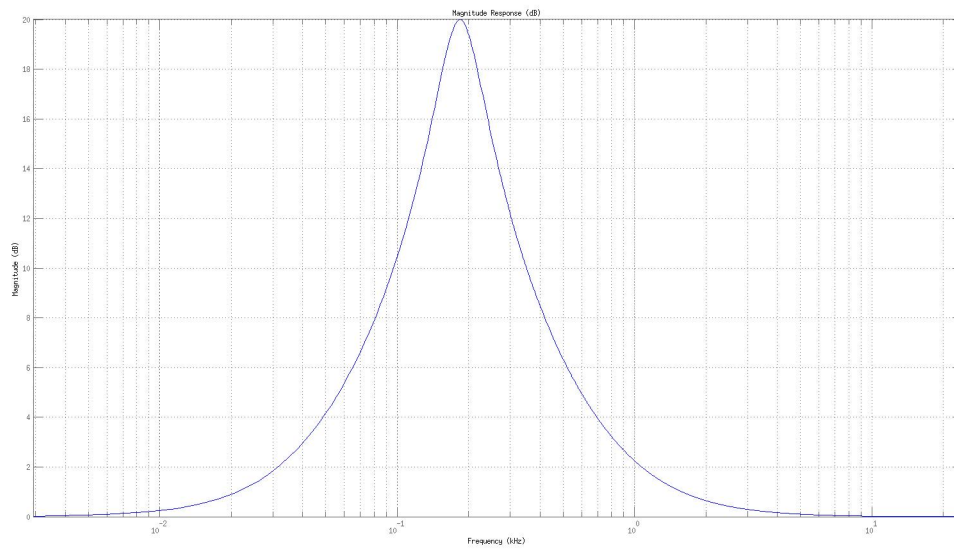


(a) berechnet

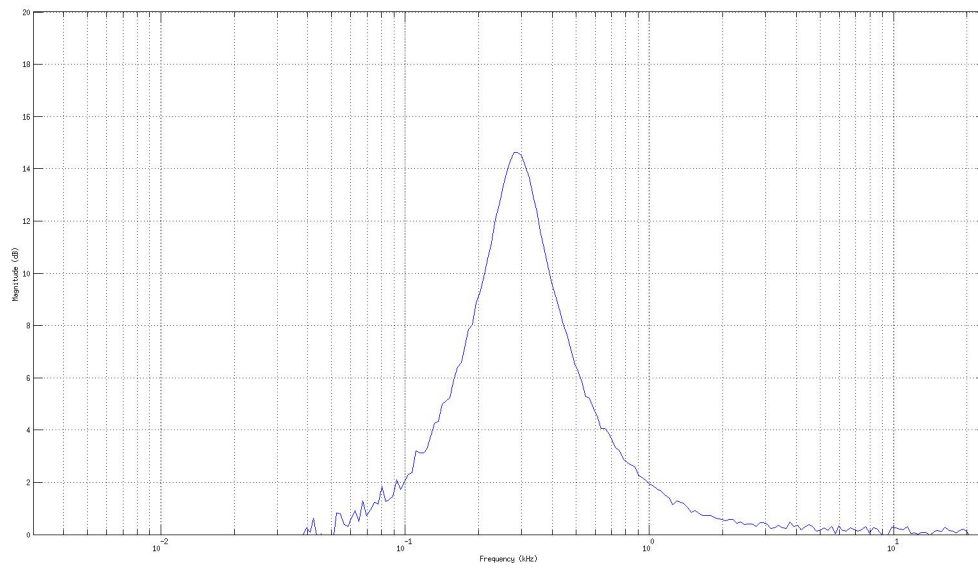


(b) gemessen

Abbildung 28: Amplitudengang Filter 4

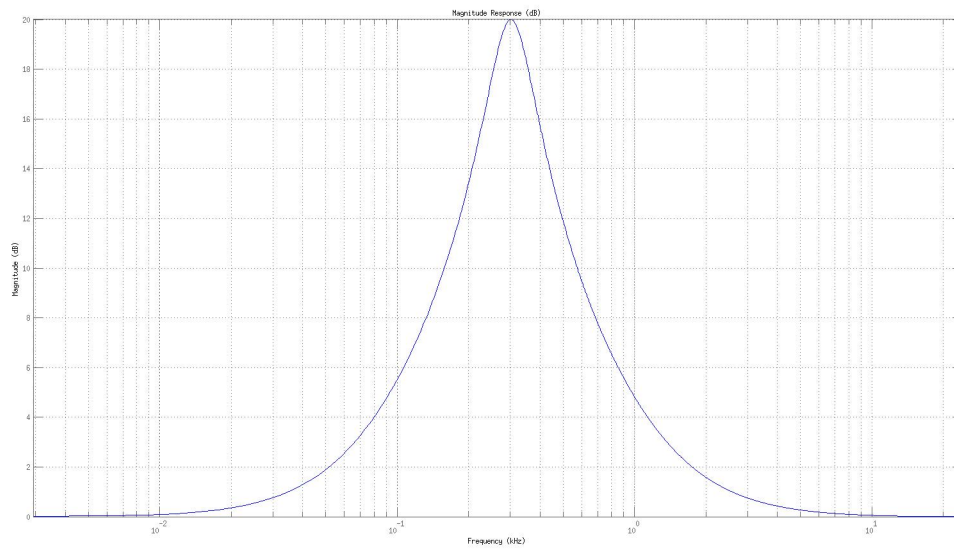


(a) berechnet

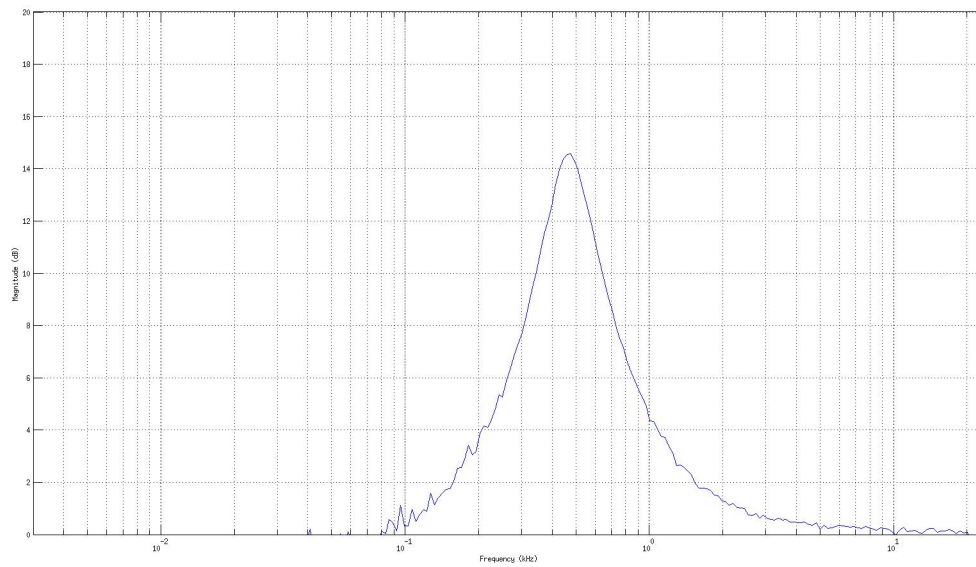


(b) gemessen

Abbildung 29: Amplitudengang Filter 5

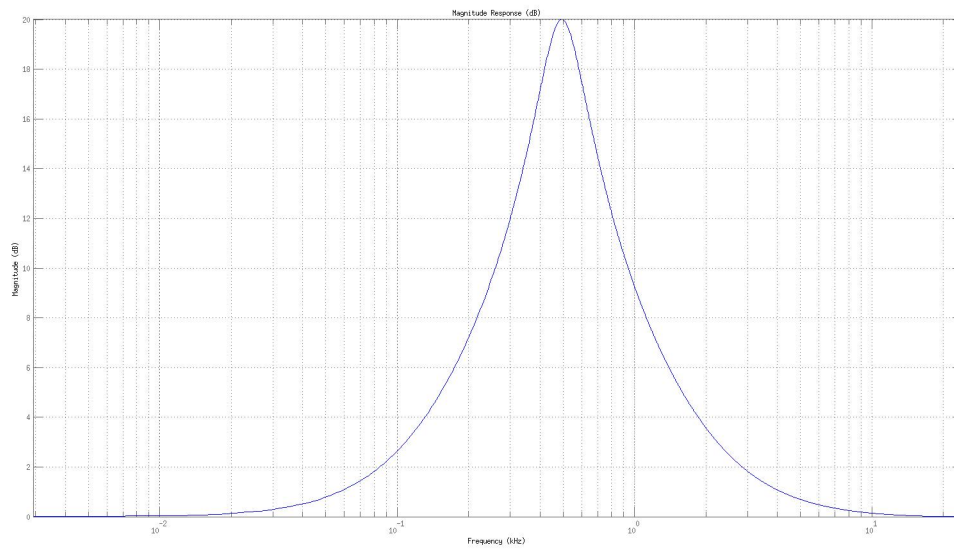


(a) berechnet

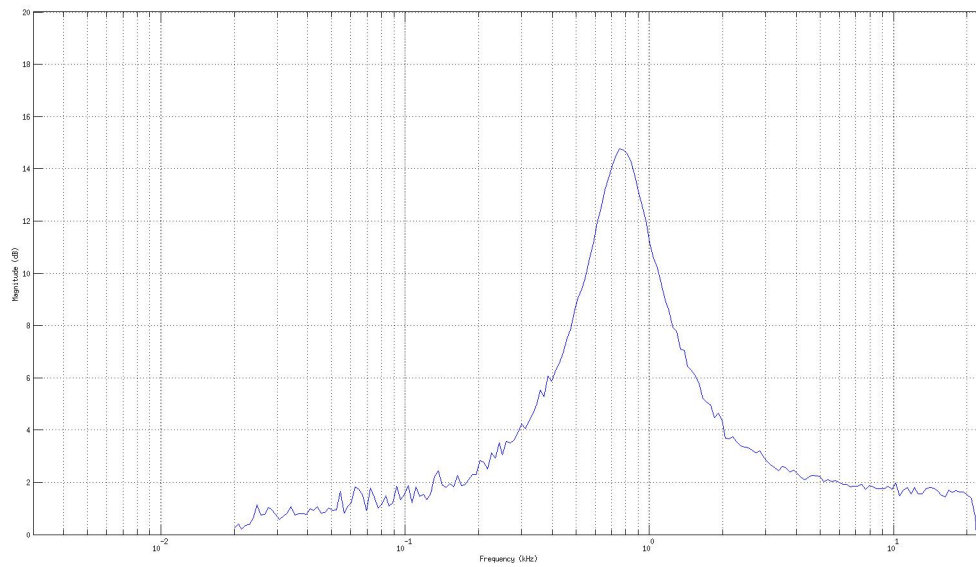


(b) gemessen

Abbildung 30: Amplitudengang Filter 6

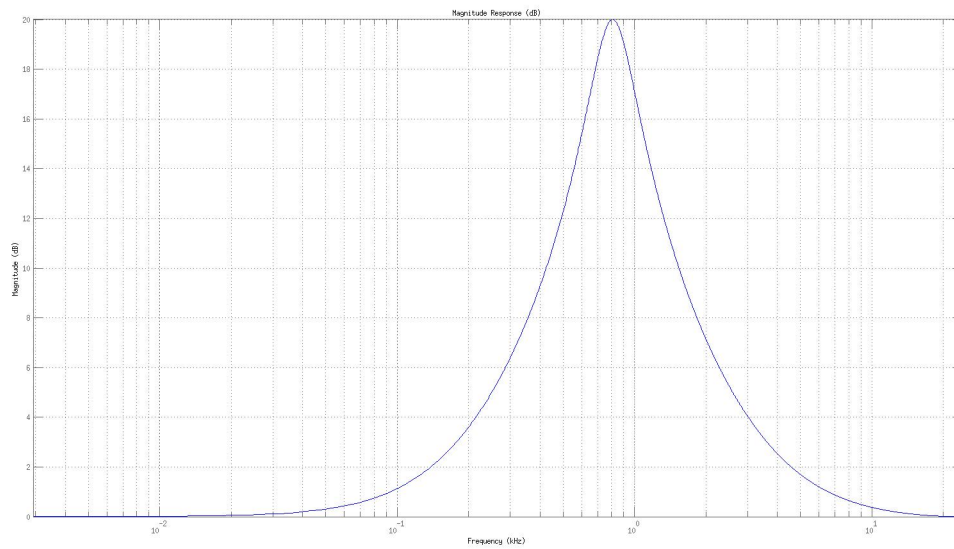


(a) berechnet

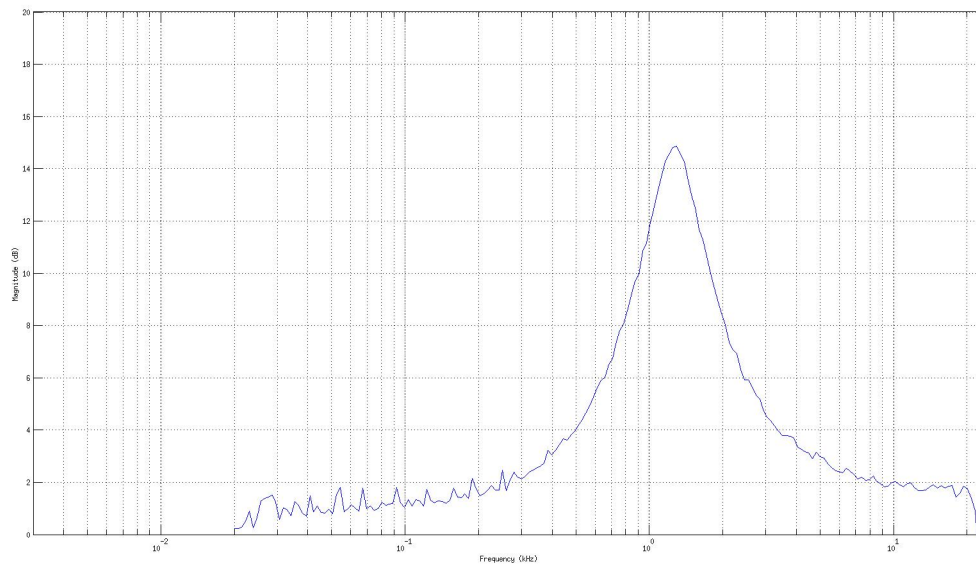


(b) gemessen

Abbildung 31: Amplitudengang Filter 7

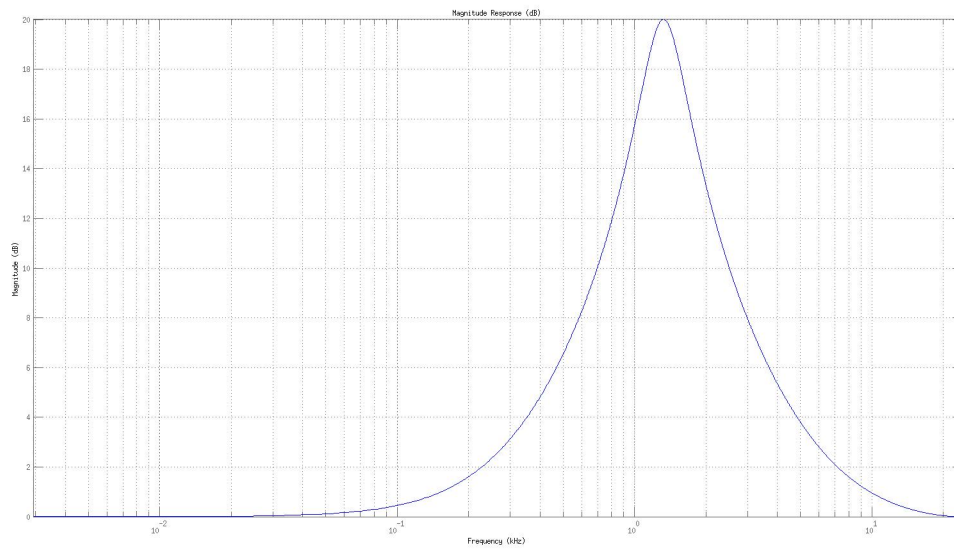


(a) berechnet

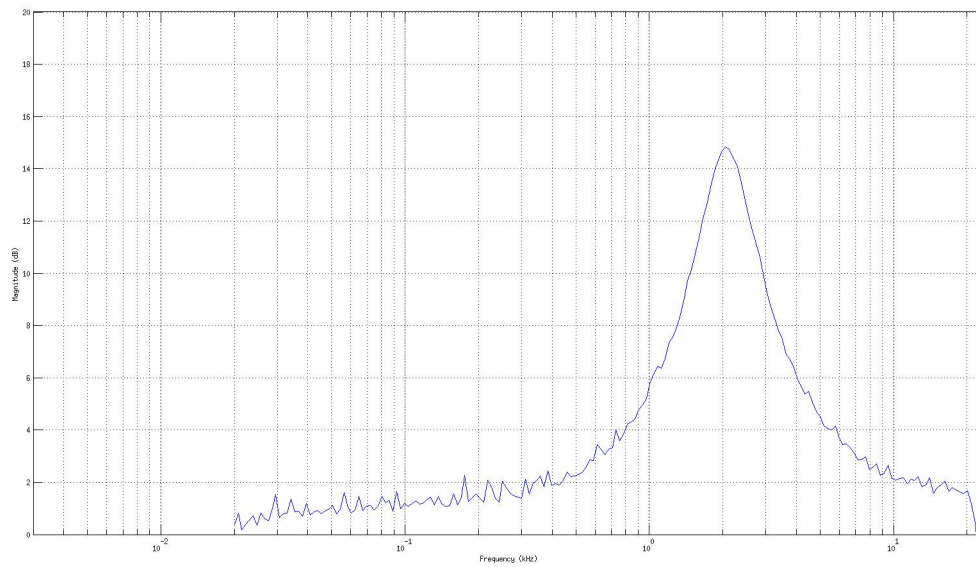


(b) gemessen

Abbildung 32: Amplitudengang Filter 8

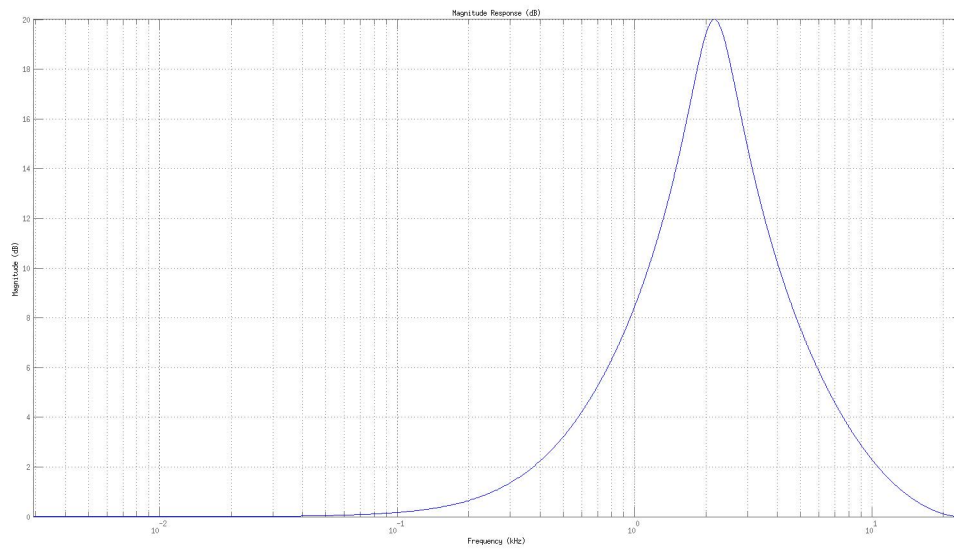


(a) berechnet

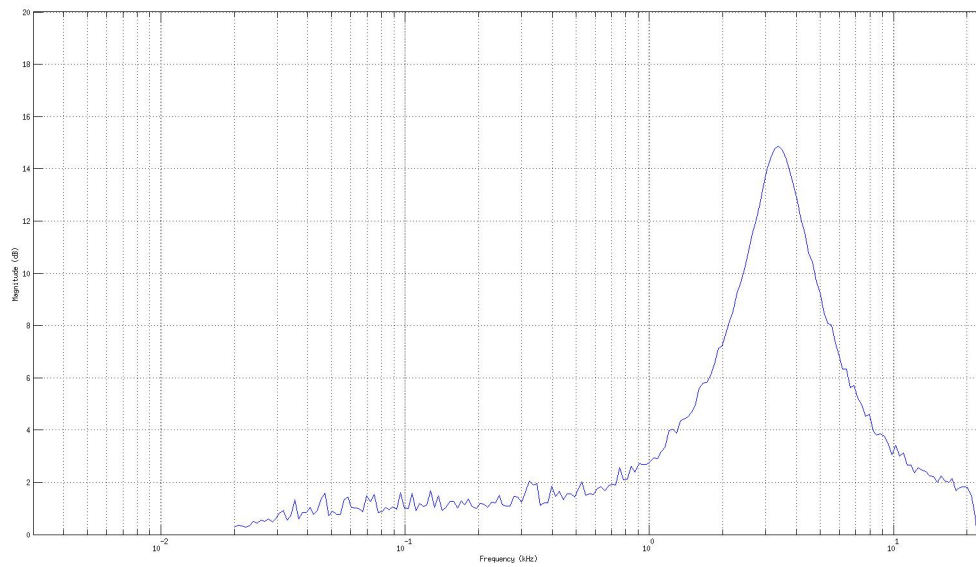


(b) gemessen

Abbildung 33: Amplitudengang Filter 9

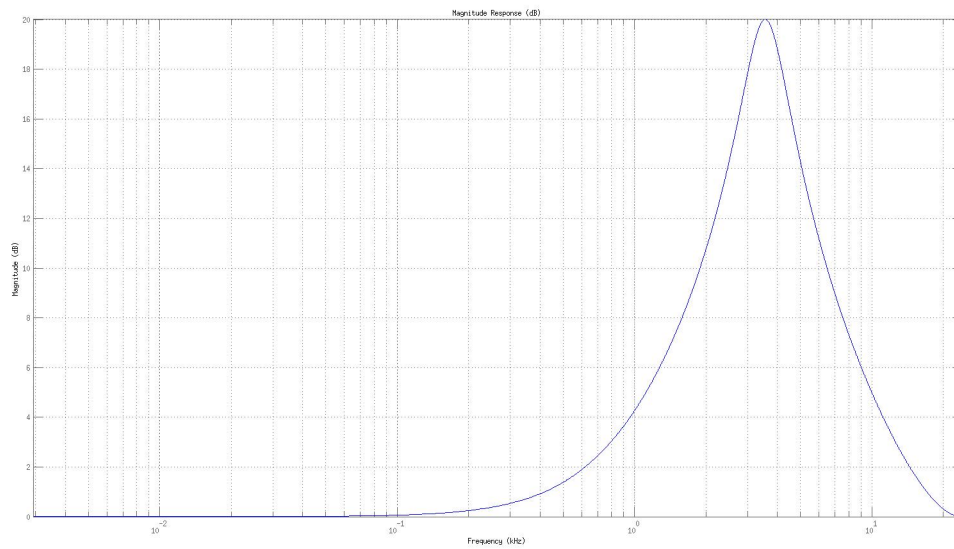


(a) berechnet

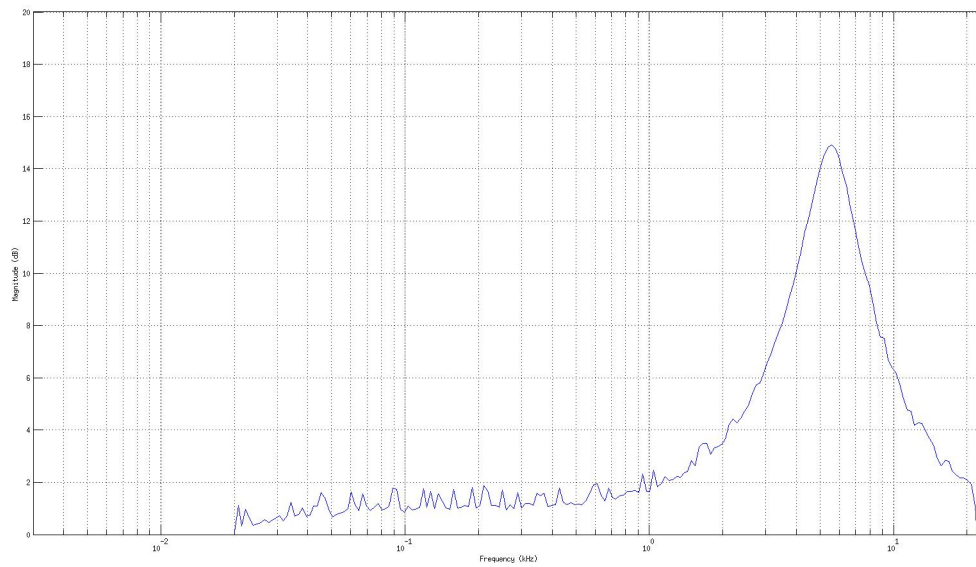


(b) gemessen

Abbildung 34: Amplitudengang Filter 10

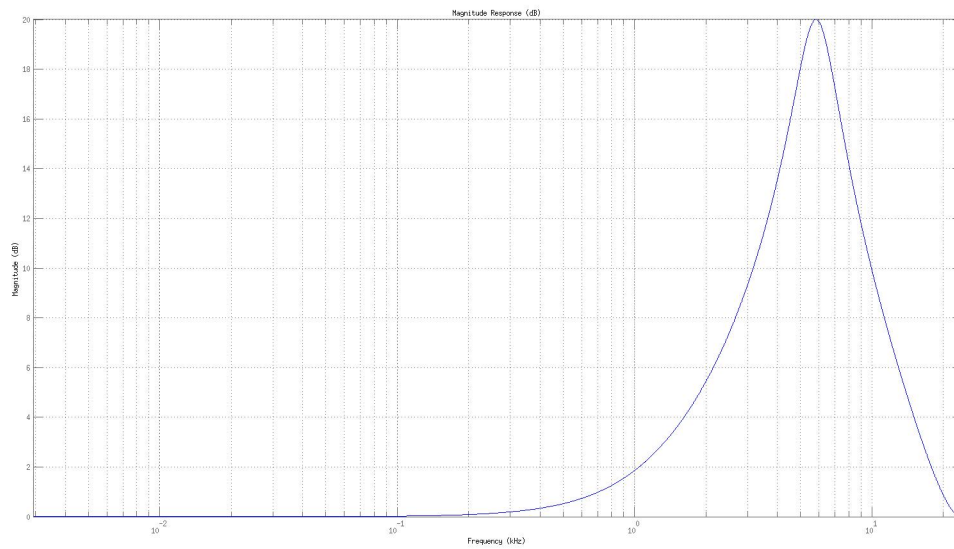


(a) berechnet

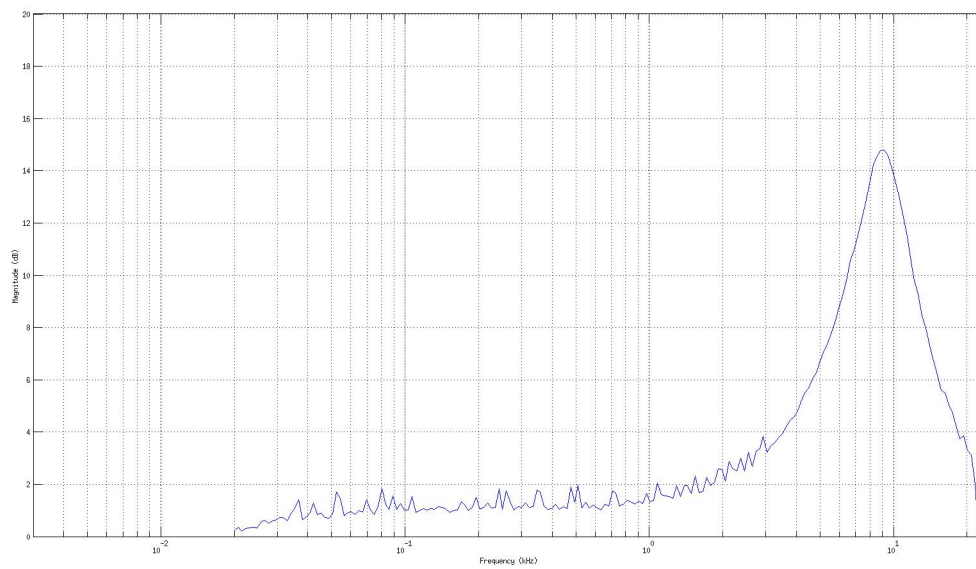


(b) gemessen

Abbildung 35: Amplitudengang Filter 11

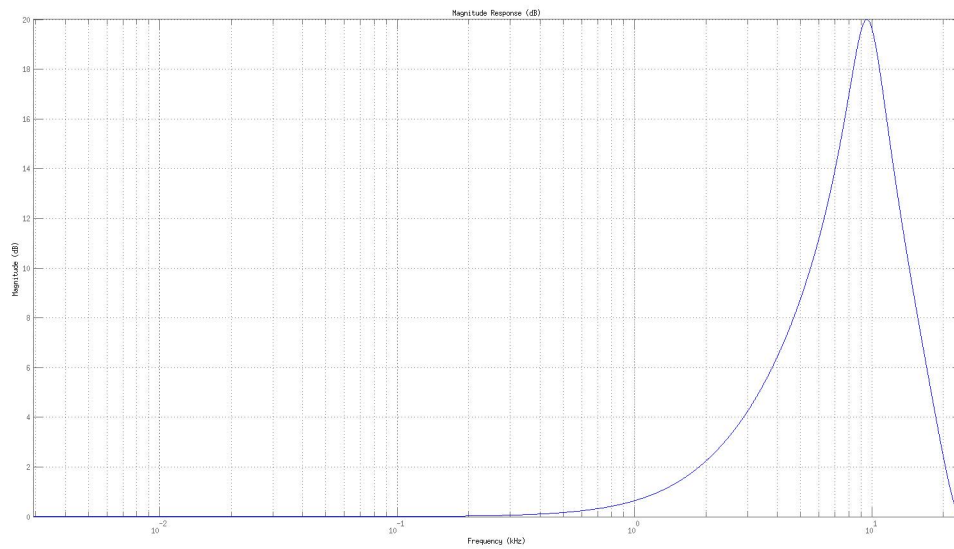


(a) berechnet

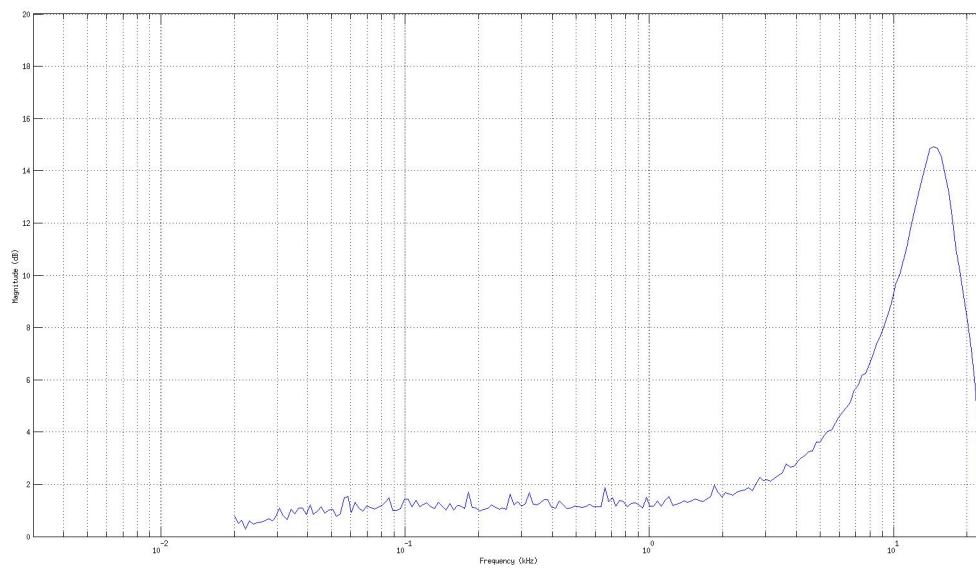


(b) gemessen

Abbildung 36: Amplitudengang Filter 12

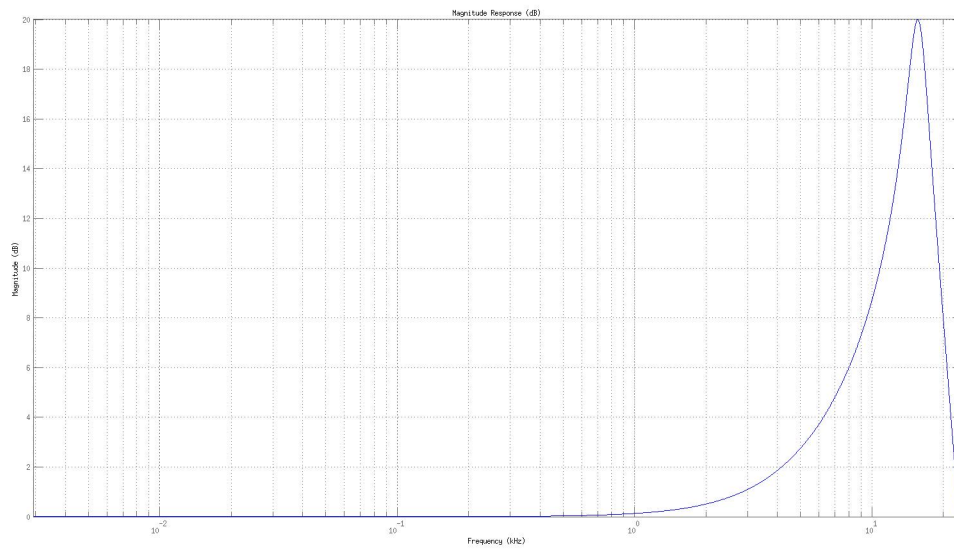


(a) berechnet

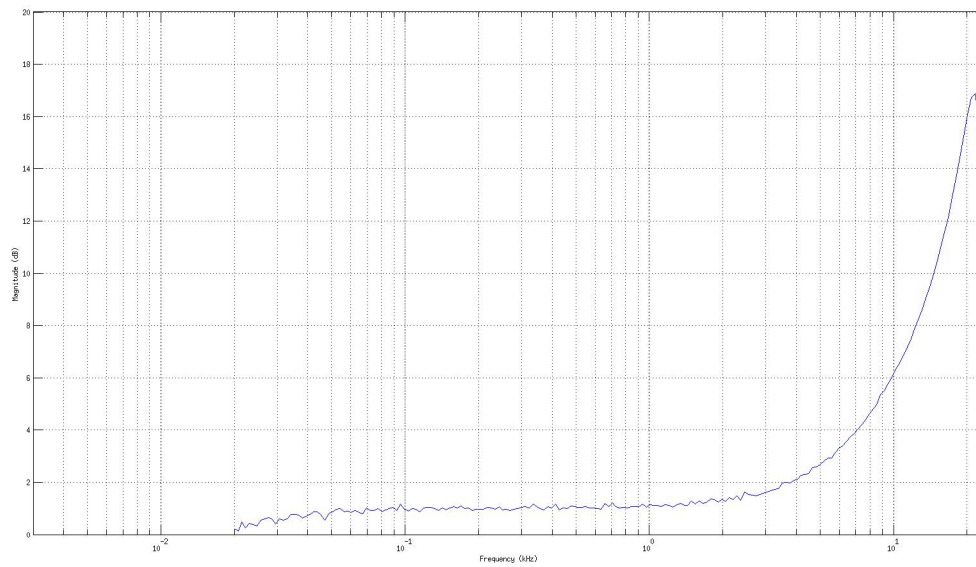


(b) gemessen

Abbildung 37: Amplitudengang Filter 13



(a) berechnet



(b) gemessen

Abbildung 38: Amplitudengang Filter 14

Literatur

- [1] Dr-Ing. Seyed Ali Azizi. *Entwurf und Realisierung digitaler Filter*. Oldenbourg Verlag GmbH, 5., verb u. erw. aufl. edition, 1990.
- [2] Prof. Dr.-Ing. Josef Hoffmann. *Spektrale Analyse mit MatLab und Simulink*. Oldenbourg Wissenschaftsverlag GmbH, 2011.
- [3] Bernd Schwarz Jürgen Reichardt. *VHDL Synthese - Entwurf digitaler Schaltungen und Systeme*. Oldenbourg Verlag GmbH, 2009.
- [4] Prof. Dr.-Ing. Otto Mildenerger. *Entwurf analoger und digitaler Filter*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1992.
- [5] Wolfson Microelectronics plc. Datasheet wm8731. http://www.wolfsonmicro.com/documents/uploads/data_sheets/en/WM8731.pdf, April 2009.
- [6] Prof. Dr. Franz Quint Prof. Dr.-Ing. Josef Hoffmann. *Signalverarbeitung mit MATLAB und Simulink*. Oldenbourg Wissenschaftsverlag GmbH, 2007.
- [7] Prof. Dr. rer. nat. habil. Hanno Kahnt. Praktikum - audioverarbeitung mit matlab - 2. <http://www.et.fh-jena.de/kahnt/praktikum/da-ml2-1.pdf>, 2012.
- [8] Prof. Dipl.-Ing Herbert Wagner. Programmierbeispiele. <http://www.et.fh-jena.de/wagner/ProLog6.htm>, 2012.
- [9] Prof. Dipl.-Ing Herbert Wagner. Vorlesungsscripte. <http://www.et.fh-jena.de/wagner/ProLog4.htm>, 2012.