

Lehrdemonstrationen zur digitalen Filterung auf eingebetteten Systemen

Eduard Funkner

Martin Schwarz

2025-05-28

Inhaltsverzeichnis

| | |
|---|----------|
| Einführung | 3 |
| Zielsetzung und Vorgehen | 4 |
| Signalverarbeitung auf Microcontroller (ESP32) | 5 |
| Signalverarbeitung auf FPGA (PYNQ) | 6 |
| Jupyter Notebooks | 6 |
| VHDL | 7 |

Einführung

In der Welt der Elektrotechnik werden Signale verschiedener Arten gesendet und empfangen. Es wird zwischen kontinuierlichen analogen Signalen und diskretisierten digitalen Signalen differenziert. Ein analoges Signal könnte eine Tonspur, welche von einem Mikrofon aufgenommen wurde sein. Bei digitalen Signalen werden Bitfolgen gesendet die Information enthalten. Diese Bitfolge könnte eine E-Mail sein.

Um diese Signale nutzen zu können müssen diese zuvor gefiltert und verarbeitet werden. Die Filterung dieser Signale erfolgt durch digitale oder analoge Filter. Abhängig von der Anwendung werden spezifischen Filter implementiert. Die Frequenzbereiche der Signale können Mittels dieser Filter verstärkt oder gedämpft werden damit diese dann weiterverarbeitet werden können.

Zielsetzung und Vorgehen

Implementierung digitaler Filter auf DSP

Lerndemonstration von der digitalen Signalverarbeitung auf DSP

Signalverarbeitung auf Microcontroller (ESP32)

Zur Implementierung der digitalen Filter, auf Mikrocontrollern, wird ein ESP32 verwendet.

Spezifisch für Audioanwendungen werden Audioboards, wie das ESP Lyrat Mini oder ESP Lyrat 4.3 genutzt. Diese Boards verfügen über Audiocodecs, welche analoge Signale in ein digitales Format konvertieren können. Die Datenübertragung zwischen den Codecs und dem ESP32 geschieht über I2S. Dieser I2S Datenstrom kann auf dem ESP32 verarbeitet und auf der SD Karte der Board gespeichert werden. Über diese Audioboards soll eine analoges Signal in Form von Sprache aufgenommen werden, von dem ESP32 mit den digitalen Filtern gefiltert werden und anschließend abgespielt, oder auf einer SD Karte gespeichert werden. Außerdem sollen Audio Dateien von einer SD ausgelesen, gefiltert und abgespielt werden.

Der Audiofilter (Biquad) wird aus dem Skript Digital Signal Processing Lecutue von S. Spors entnommen und in Arduino, sowie Micropython auf dem ESP32 implementiert.

Zur Hilfe werden von P.Schatzmann die GitHub Repositorien für die Audio Tools sowie Audio Driver genutzt, um die Implementierung in Arduino durchzuführen.

Signalverarbeitung auf FPGA (PYNQ)

Bevor ein Filter in Hardware implementiert werden kann, muss er zunächst entworfen werden. Dabei ist zu entscheiden, welche Art digitaler Filter zum Einsatz kommen soll. Für Audioanwendungen eignen sich insbesondere linearphasige Filter. Besonders FIR-Filter werden häufig verwendet, da sie eine konstante Gruppenlaufzeit aufweisen.

IIR-Filter bieten zwar eine kürzere Gruppenverzögerung, sind jedoch aufgrund ihrer nicht konstanten Gruppenlaufzeit anfälliger für Verzerrungen. Zudem sind FIR-Filter in der Regel einfacher zu realisieren. Ein FIR-Filter kann entweder mit Tools wie dem `fdatool` in MATLAB entworfen oder aus einem Prototypenfilter abgeleitet werden, häufig aus einem IIR-Filter mit den gewünschten Eigenschaften.

Im nächsten Schritt wird das PYNQ-Z2 vorbereitet. Dazu muss die erforderliche Software auf einem PC installiert werden, einschließlich aller notwendigen Bibliotheken und Module für die Programmierung sowohl mit VHDL.

Anschließend wird das Board in Betrieb genommen, um sich mit seiner Funktionsweise vertraut zu machen. Dazu wird erster Testcode aufgespielt. Dieser Schritt dient dazu, ein grundlegendes Verständnis der Hardware zu erlangen und eine bessere Vorstellung der Umsetzungsmöglichkeiten zu bekommen.

Danach wird zunächst ein Audiosignal in das Board eingespeist und ohne Filterung wieder ausgegeben. Dies dient dazu, sich mit dem Board vertraut zu machen und sicherzustellen, dass die Einspeisung und Ausgabe von Audio grundlegend beherrscht wird.

Anschließend werden die Filter entworfen. (Mit dem `fdatool` in MATLAB lassen sich die Filter Auswahl des Typens und den Koeffizienten direkt für andere Anwendungen wie VHDL umsetzen)

Jupyter Notebooks

Das PYNQ-Z2 verfügt über ein eigenes Betriebssystem, das zunächst eingerichtet werden muss. Dieses enthält eine Jupyter-Notebook-Umgebung, über die das Board konfiguriert und dokumentiert werden kann. Zudem sind Beispielprojekte verfügbar. Der Zugriff auf das Board erfolgt remote über die IP-Adresse. Das Dateisystem ist über denselben Weg im Datei-Explorer erreichbar. Zusätzlich steht ein Jupyter-Terminal zur Verfügung. Um mit dem Laptop arbeiten zu können, sollte das Board entsprechend eingerichtet werden. Es empfiehlt sich, den Setup-Guide vollständig durchzugehen.

VHDL

In VHDL müssen alle erforderlichen Top-Module für das Board entweder geschrieben oder bestehende Module eingebunden werden. Dabei wird unter anderem die Portzuweisung vorgenommen. Anschließend wird der Filter selbst in VHDL implementiert und mithilfe von VIVADO synthetisiert. Nach der Synthese wird der Bitstream für das Board generiert.