



FH Bielefeld
University of
Applied Sciences

Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Studiengang Ingenieurinformatik

Titel der Ausarbeitung

Art der Ausarbeitung

Name der/s Autors/in bzw. Autoren/innen inkl. Matrikelnummer

23. Februar 2022

Betreuer:
Prof. Dr. Axel Schneider
Dr. Hanno Gerd Meyer

Eine kurze deutsche Zusammenfassung.

A short abstract in english.

Inhaltsverzeichnis

1	Einleitung	5
1.1	was kann das gerät	6
2	Komponenten	8
2.1	Arithmetik	8
2.1.1	Zähler	8
2.1.2	Teiler	8
2.2	Takterzeugung	8
2.2.1	Clock-Enable	8
2.3	Funktionen	8
2.3.1	Konstante	8
2.3.2	Rechteck	8
2.3.3	Zick-Zack	8
2.3.4	Rampe	8
2.4	Konfigurationsschnittstelle	8
2.4.1	UART-Schnittstelle	8
2.4.2	Instruktionsauswertung	9
2.5	DAC-Konverter	9
2.5.1	DAC-Kanal	9

1 Einleitung¹

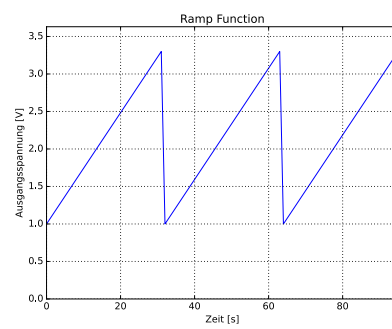
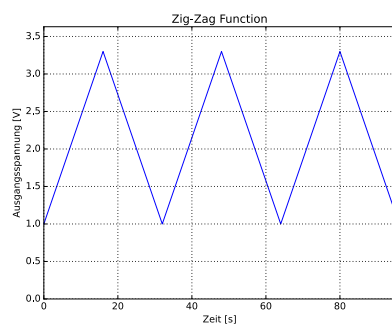
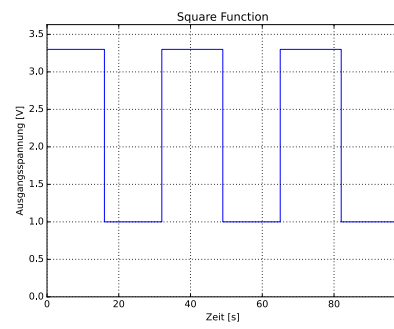
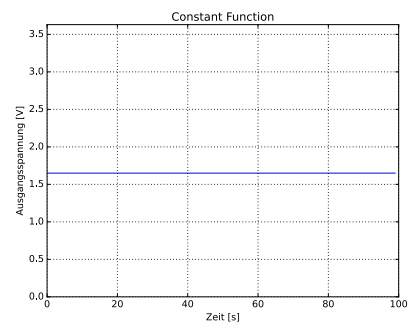
Diese Studienarbeit behandelt die Konzipierung und Implementierung eines digitalen Funktionsgenerators in der Hardwarebeschreibungssprache VHDL. Ein Funktionsgenerator ist ein elektronisches Bauteil, das in der Lage ist, verschiedene Spannungsverläufe an seinem Ausgang auszugeben. Diese Spannungsverläufe entsprechen einer mathematischen Funktion. Z. B. kann ein Funktionsgenerator genutzt werden, um ein Rechteck-Signal mit einer bestimmten Frequenz auszugeben, das dann als Auslöser für eine Kamera fungiert. Im Normalfall würde ein Funktionsgenerator als Digitalschaltung in einen Chip integriert oder auf eine Platine gelötet werden. Einen anderen Ansatz zum Bau von digitalen Schaltungen bieten "Free Programmable Gate Arrays", kurz FPGA. Auf diesen ICs befinden sich verschiedene Bausteine die durch Anlegen einer Programmierspannung miteinander verknüpft werden können. Somit ist es möglich, verschiedenste Schaltungen auf demselben IC zu verwirklichen. Die Schaltungen können mithilfe einer Beschreibungssprache designed werden. Eine dieser Sprachen ist VHDL ("Very Highspeed Hardware Description Language"), welche in dieser Studienarbeit verwendet werden soll.

Im folgenden soll das Konzept des Funktionsgenerators erläutert werden und seine Funktionsweise erklärt werden.

1.1 was kann das gerät

der generator kann 4 verschiedene Funktionen ausgeben,

1. Konstante Ein konstanter Wert liegt am Ausgang an.
2. Rechteck-Funktion Der Wert wechselt zwischen High- und Low-Pegel in der Frequenz f . Der Anteil der Zykluszeit T , in dem der Ausgang auf High ist, wird über den dutycycle eingestellt.
3. Zick-Zack-Funktion Der Analogwert steigt vom Low-Pegel bis zum High-Pegel linear an, erreicht er den High-Pegel, fällt der Analogwert wieder kontinuierlich auf Low ab. Somit schwankt der Pegel mit der Frequenz f .
4. Rampen-Funktion Der Analogwert wächst, wie bei der Zick-Zack-Funktion, linear bis auf High an, dann fällt er aber auf Low zurück. Alternativ kann die Rampe auch vom High-Pegel her abfallen und bei Erreichen von Low wieder auf High zurück springen.



2 Komponenten

In diesem Kapitel soll der Aufbau des Funktionsgenerators anhand seiner digital-technischen Komponenten erläutert werden.

2.1 Arithmetik

2.1.1 Zähler

2.1.2 Teiler

Während beim Multiplizieren, Addieren und Subtrahieren auf Standardfunktionen zurückgegriffen wird, wird für die Division zweier Binärzahlen eine eigene Komponente entworfen. Die Gründe hierfür werden in Abschnitt 2.3.3 erläutert.

2.2 Takterzeugung

2.2.1 Clock-Enable

2.3 Funktionen

2.3.1 Konstante

2.3.2 Rechteck

2.3.3 Zick-Zack

2.3.4 Rampe

2.4 Konfigurationsschnittstelle

Die Konfigurationsschnittstelle `CONFIG_INTERFACE` besteht aus einer UART-Schnittstelle, über die der Datenaustausch zwischen Benutzer und Funktionsgenerator erfolgt, sowie der Instruktionsauswertung, die die empfangenen UART-Signale in Konfigurationsbefehle übersetzt.

2.4.1 UART-Schnittstelle

Die UART-Schnittstelle beruht auf dem **U**niversal-**A**synchronous-**R**ceiver-**T**ransmission-Protokoll. Das Protokoll ermöglicht es, byteweise serielle Daten zu verschicken und zu empfangen. Hierfür reichen zwei Drähte aus, die jeweils eins der beiden Signale

RX (Receive) und TX (Transmit) transportieren. Zum Start der Kommunikation wird die RX Leitung vom Sender von high auf low gezogen, sodass der Empfänger anfängt die nachfolgenden acht Bits zu einem Byte zusammenzusetzen. Anschließend muss mindestens ein Stop-Bit folgen, bei dem die Receive Leitung des Empfängers auf High liegt. Darauf kann je, nach Implementierung, noch ein Stop-Bit sowie ein Paritätsbit folgen. Da es zwischen dem Sender und Empfänger kein synchrones Taktsignal gibt, ist es wichtig, dass ihre Sende- und Empfangsfrequenz gleich ist. Diese Frequenz ist die sogenannte Baudrate. Im Funktionsgenerator ist sie auf 115200 Bits / s festgelegt. Die im Generator verwendete Schnittstelle wurde, um den Arbeitsaufwand zu verringern, aus einer Vorlage übernommen (Quelle). Sie beinhaltet sowohl eine Empfänger- als auch eine Sender-Komponente. Es gibt folgende Eingangssignale:

1. **CLK**: Eingang, gibt die Taktfrequenz der Komponente vor
2. **CE**: Eingang, "chip-enable"-Signal, aktiviert die Komponente wenn es auf low gesetzt wird
3. **reset**: Eingang, die Schnittstelle wird auf den Initialisierungszustand zurückgesetzt und die aktuelle Übertragung bzw. aktuelle Empfangsprozesse werden abgebrochen.
4. **tx**: Ausgang, Das von der Schnittstelle versendete TX-Signal
5. **tx_start**: Eingang, wenn **tx_start** auf high gesetzt wird, wird mit der Übertragung von **data_in** begonnen
6. **data_in**: Eingang, ein 8-Bit breites Signal, dass das zu versendende Byte enthält.
7. **rx**: Eingang, das von der Schnittstelle empfangene RX-Signal
8. **data_out**: Ausgang, ein 8-Bit breites Signal, dass das zuletzt von der Schnittstelle empfangene Byte beinhaltet.
9. **rx_uart_rdy**: Ausgang, dieses Signal zeigt an, wenn ein komplettes Byte empfangen wurde und bereit ist, gelesen zu werden.

2.4.2 Instruktionsauswertung

2.5 DAC-Konverter

2.5.1 DAC-Kanal