

## **Schaltkreisentwurf mit VHDL**

Ansteuerung Leuchte mittels PWM

**Carolina da Rocha Nobre**

EI18WI-b

Abschlussbeleg für das Lehrfach "GSSE/Schaltkreisentwurf"

Fakultät Computer- & Biowissenschaften  
Prof. Dr.-Ing. Th. Beierlein  
Hochschule Mittweida  
WS2020

# Gliederung

<b>1</b>	<b>Aufgabenstellung</b>	<b>2</b>
<b>2</b>	<b>System Parameter</b>	<b>2</b>
<b>3</b>	<b>Theoretische Grundlagen</b>	<b>2</b>
<b>4</b>	<b>Funktionsbeschreibung</b>	<b>3</b>
<b>5</b>	<b>Blockschaltbild</b>	<b>4</b>
<b>6</b>	<b>Entwurfsdurchführung</b>	<b>5</b>
6.1	Vorteiler . . . . .	5
6.2	Flankenerkennung . . . . .	5
6.3	Zähler . . . . .	6
6.4	Schwelle . . . . .	6
6.5	Comparator . . . . .	6
<b>7</b>	<b>Simulation</b>	<b>7</b>
7.1	Vorteiler . . . . .	7
7.2	Zähler . . . . .	7
7.3	BTNFLANKE . . . . .	7
7.4	Schwelle . . . . .	8
7.5	Comparator . . . . .	8
<b>8</b>	<b>Implementierung</b>	<b>9</b>
<b>9</b>	<b>Quelle</b>	<b>10</b>

# 1 Aufgabenstellung

Ziel der Aufgabe ist die Steuerung von der Helligkeit eines LED mittels Puls Weite Modulation. Die LED soll 10 Helligkeitsstufen Vorweisen. Die Steuerung wird mittels zwei Pushbuttons erfolgen.

Benötigt wird:

- Basys 3
- micro-USB to USB Kabel
- Computer mit dem Software VIVADO 2019.1 instalirt (Vivado kann auf Linux oder Windows Installiert werden)

Für die Aufgabe wird den Basys 3™ FPGA Board von dem Hersteller Digilent verwendet. Der Basys 3 bietet eine umfangreiche Auswahl an Ports und Peripheriegeräten (Datenblatt s.2). Davon wird an diesen Projekt das Clocksignal, eine LED und zwei Pushbuttons verwendet. Keine äußere Peripherie ist für diese Entwurf nötig. Für das Programmierung und Stromversorgung wird nur einem USB zu micro-USB Kabel benötigt. Das Kabel verbindet das Computer zu dem JTAG USB Port auf dem Board.

Für den Entwurf, Simulieren und Implementierung wurde das Software Vivado in der version 2019.1 von der Hersteller Xilinx verwendet. Das ausgewählte Programmiersprache ist VHDL.

## 2 System Parameter

Betriebsspannung	5V
Clock (W5)	100MHz
Pushbutton BTNR (T17)	Eingang
Pushbutton BTNL (W19)	Eingang
Led LD0 (U16)	Ausgang

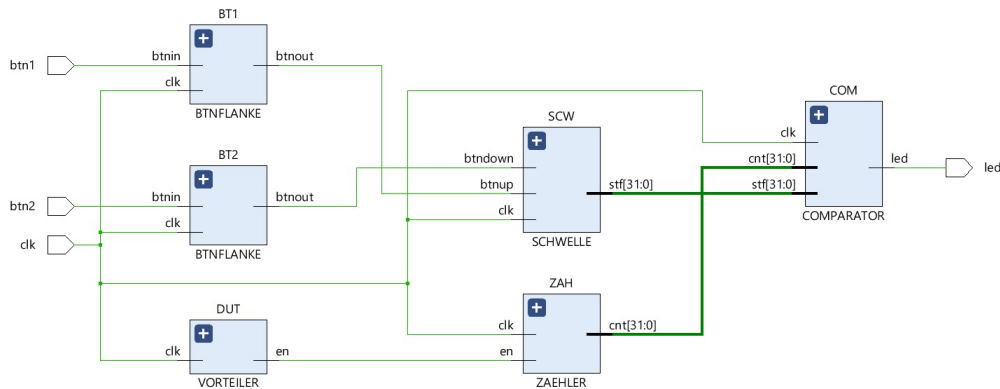
## 3 Theoretische Grundlagen

Die Ein- und Ausblenden der LED wird durch Pulsweitmodulation gesteuert. Die Lichtstärke wird geändert, indem das verhalten von der An- und Ausschaltphasen in einem Zeitfenster geändert wird. Diese Zeitfenster und damit die an und Ausschaltphasen sind so schnell, dass die Menschliche Augen es nicht wahrnehmen kann. Wir haben die Illusion, dass der LED dunkler oder heller aussieht.

## **4 Funktionsbeschreibung**

Das LED wird mit 10 Helligkeitsstufen programmiert. Bei einmal pressen von der Pushbutton "BTNR (T17)" soll die LED "LD0" um einem Stufe heller werden. Bei pressen der "BTNL (W19)" soll der LED um einem Stufe dunkler werden. Bei erreichen des Maximal- bzw. Minimal- Helligkeitsstufen, reagiert das LED nicht mehr auf das Signal von der jeweiligen Pushbutton.

## 5 Blockschaltbild



Blockschaltbild

Die Schaltung hat 3 Eingänge: Das Clocksignal und Zwei Buttons. Die zwei Buttons kontrollieren das Dimmen von der LED. Als Ausgang hat die Schaltung nur den LED selbst. Außerdem hat man 3 Zwischensignale: en (enabled), cnt(counter) und stf(stufe).

Den Taktfrequenz generiert bei dem Oszillator ist für diese Anwendung zu hoch. Deshalb geht den von dem Oszillator generierte Clocksignal in einem VORTEILER. Somit wird den Takt die gesamte Schaltung langsamer.

Der ZAEHLER bestimmt der Anfang des Ausgangsimpuls und zählt bis auf die gesamte Pulsbreite. Die gesamte Pulsbreite dauert mehrfach von dem Signal "en".

Weil ein Mensch viel langsamer ist als den Clock und bei einmal pressen des Buttons viele aufsteigende Flanke von Clock dauern kann, braucht man ein Modul vor dem Schwelle zur Flankenerkennung. Dieses BTN-FLANKE Modul verkürzt den Button Signal in nur eine Periode von Clock. Damit werden die Befehle in der Schwelle nur einmal geführt und nicht mehrmals solange der Button gepresst ist.

Der SCHWELLE bekommt das Signal von den Buttons auf dem Board und gibt aus in welchen Stufe das LED sein soll.

Der COMPARATOR vergleicht den wert aus dem ZAEHLER und SCHWELLE und setzt die Pulsbreite

## 6 Entwurfsdurchführung

Die Schaltung wird Synchron betrieben.

„Ein synchroner Schaltkreis ist eine Digitalschaltung bestehend aus taktf flankengesteuerten Flipflops welche alle von einem zentralen Taktsignal aus getaktet werden“. (Wikipedia)

Der Basys 3 enthält eine 100 MHz Oscillator verbunden zu dem Pin W5. (Datenblatt s.7). Diese Oscillator bestimmt das Arbeitstakt (sog. Clock) von der Schaltung.

Das design hat 3 Inputs und einem Output.

Inputs

- Clock
- BTNR
- BTNL

Output

- LED

Hier wird die Funktionsweise jedes Modul Ausführlicher beschrieben.

### 6.1 Vorteiler

Die Funktion eines Vorteiler ist die Schaltung in einem gewünschte Frequenz zu takten. Der Clocksignal von der Oszillator ist mit 100MHz zu Hoch. Um die Schaltung langsamer zu takten zählt man die aufsteigende Flanken von der Clocksignal. In diese Projekte will man der Schaltung mit etwa 1MHz takten. Deshalb werden Hunderte aufsteigende Flanke gezählt. Jede Hunderte Flanke wird der Signal am Ausgang für die Dauer eine Periode '1' und dann wieder '0'. Diese Signal wird "en" (enabled) genannt. Jetzt sollen alle andere Flankenwechseln nur bei einem aufsteigende Flanke von clk und en = '1' passieren. So wird die Schaltung Synchron.

### 6.2 Flankenerkennung

Hier werden zwei Flip-flops verwendet um die aufsteigende Flanke beide Buttons zu erkennen. Dafür wird das erste Flip-flop das Signal von der Button speichern. In den nächsten Clock aufsteigende Flanke wird diesen

wert am nächsten FlipFlop übertragen. Somit hat man die aktuelle und die letzte Zustand des Buttons. Nur wenn die aktuelle Zustand HIGH ist und die letzte Zustand LOW ist, hat man eine aufsteigende Flanke. Dabei ist der Ausgang HIGH. Sonst ist der Ausgang LOW. Somit kriegt man ein zu Clock synchrones Signal, das genau einen Takt dauert.

### **6.3 Zähler**

Der Zähler wird der Anfang jedes Puls und auch die gesamte Pulsbreite für die PWM setzen.

Als Eingang bekommt der Ausgang "en" von der Vorteiler und der Clocksignal. Bei jedem Auftreten der "en" Signal zählt der Zähler auf. Wie viele mal "en" auftritt wird in das Signal "cnt" am Ausgang geliefert. Es wird aber nur bis 1023 gezählt, dann fängt der Zähler wieder von null an. 1024 ist der gesamte Pulsbreite der PWM.

### **6.4 Schwelle**

Funktion der Schwelle ist den Signal beide Buttons aufnehmen die entsprechenden Helligkeitsstufe am Ausgang ausgeben. An der Schaltung wollen wir die LED mit 10 Helligkeit Stufe steuern. Deshalb wird die gesamte Pulsbreite 1024 durch 10 geteilt. Das ergibt 102 und ist der Schrittweite zwischen den Stufen. Bei jedem Pressen von "BTNR" wird der Ausgang mit 102 summiert. Bei jedem Pressen von "BTNL" wird der Ausgang um 102 subtrahiert. Die Grenzen 102 und 921 werden gesetzt, damit bei mehr als 10 mal Infolge Pressen von einem der Buttons, das LED nicht plötzlich wieder dunkler oder heller wird ("zahl" macht ein Loop), sondern aufhört am maximale bzw. minimale Stufe.

### **6.5 Comparator**

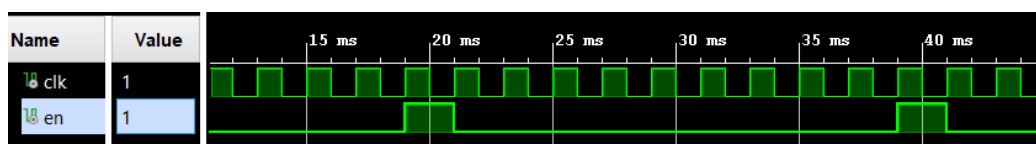
Der COMPARATOR hat eigentlich zwei Komparatoren. Eine vergleicht die zwei Werte "cnt" von der Zähler und "stf" von der Schwelle. Falls diese Werte gleich sind, wird zu "led" '1' zugewiesen. Der andere Komparator vergleicht "cnt" mit 0. Falls "cnt" = '0' ist, wird zu "led" '0' zugewiesen. So macht man die Pulsweitenmodulation.

## 7 Simulation

Hier In der Simulation werden Signale kleiner Skaliert. Das wird aus zwei Grunde gemacht: Damit die benötigte Rechenleistung nicht so hoch ist und damit die Pegel wechseln an der Simulation übersichtlicher wird.

### 7.1 Vorteiler

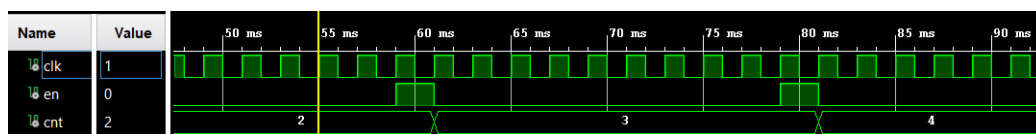
In das Testbench wird einem CLK Signal Simuliert. Das simulierte clk Signal hat eine Dauer von 1ms. Davon 500ns ist das Signal HIGH und 500ns ist das Signal LOW. Aus dem Simulation ist es zu sehen, dass bei jeder 10. aufsteigende Flanke des "clk" Signals der Ausgangssignal "en" aktiviert wird. Die richtige Schaltung soll das Gleiche verhalten vorweisen aber bei jeder 100. aufsteigende Flanke.



VORTEILER

### 7.2 Zähler

Hier sieht man, wie bei jeden Aufsteigende Flanke von "clk" und "en" ist HIGH, das Signal "cnt" am Ausgang Hochgezählt wird.



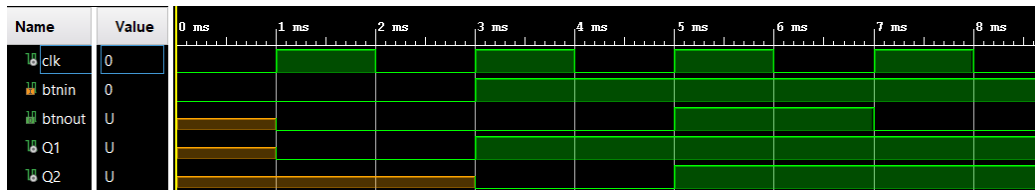
ZAEHLER

### 7.3 BTNFLANKE

In der Simulation habe ich der Button Pressen mit dem Signal "btnin" simuliert. Q1 ist der Ausgang von dem ersten Flipflop und ist das gleiche wie "btnin" (aktuelle Zustand). Q2 ist am selber Zeitpunkt noch LOW (letzte Zustand). Bei 5ms ist Q1 = '1', und Q2 = '0'. Damit wird die Bedingung



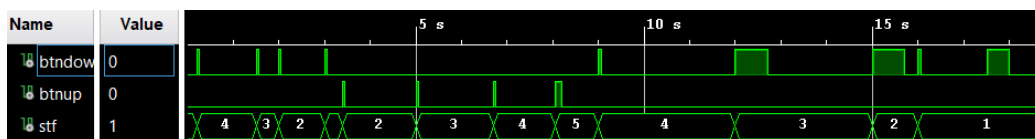
erfüllt und der Ausgang "btnout" ist 1. In der nächste aufsteigende Flanke von Clock ist die Bedingung nicht mehr erfüllt und der Ausgang wird wieder LOW.



Flanken Erkennung

## 7.4 Schwelle

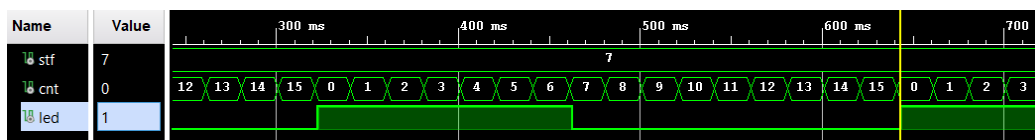
Für die Simulation der Schwelle wurde den wert von 1024 auf 16 verringert und die Schrittweite ist gleich ein. Ein Signal in der Testbench wird das Drucken des Buttons simulieren. Man sieht dass bei drücken von "BTNR" wird "stf" um 1 erhöht und bei drücken von "BTNL" wird "stf" um 1 verringert. Man sieht auch dass bei viele mal drucken den wert nicht mehr sinkt bzw steigt.



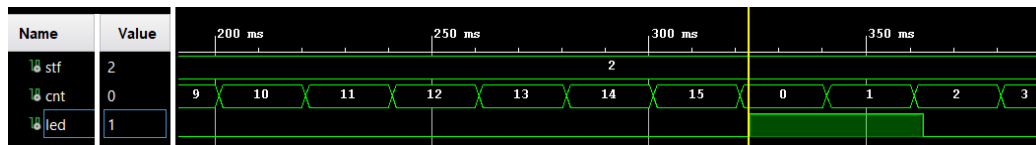
SCHWELLE

## 7.5 Comparator

Hier kann man sehen, wie für unterschiedlichen werten von "stf" den Pulsverhältnis sich ändert.



stf = 7, die Anschaltzeit dauert 7 mal cnt



stf = 2, Anhaltzeit dauert 2 cnt

## 8 Implementierung

Zum Schluss wird das Programm auf dem Board hochgeladen und getestet. Die zwei Buttons reagieren korrekt aufs drucken und das LED wird sanft ein- und Ausgeblendet. Wird allerdings niemals komplett Dunkeln oder Hell. Das ist so, weil in die schwelle die grenzen 122 und 921 gesetzt worden ist.

## 9 Quelle

- Wikipedia - Synchroner Schaltkreis: <https://bit.ly/2QYVtSx>.
- Digitaltechnik - Jürgen Reichardt - 4. Auflage ISBN 978-3-11-047800-6
- Basys 3™ FPGA Board Reference Manual
- [https://www.mikrocontroller.net/articles/VHDL\\_Flankenerkennung](https://www.mikrocontroller.net/articles/VHDL_Flankenerkennung)
- Schaltungsentwurf mit VHDL, Prof. Dr.-Ing. Thomas Beierlein, Hochschule Mittweida, Fak. CB, 2019.

Für den Blockschaltbild wurde den Web App <https://app.diagrams.net/> verwendet.