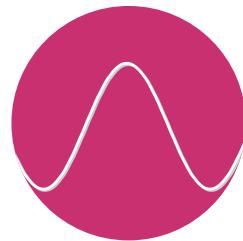


# Synchro - Generatorsynchronisierung

Projekt Digitales Design

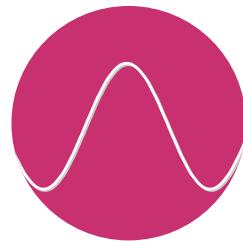


Orientierung: [Energie und Umwelttechnik \(ETE\)](#)  
Kurs: Digitales Design (DiD)  
Verfasser: [Christophe Bianchi](#), [François Corthay](#), [Silvan Zahno](#), [Axel Amand](#)  
Datum: 12. April 2023  
Version: v2.1



## Inhaltsverzeichnis

|                                       |           |
|---------------------------------------|-----------|
| <b>1 Einführung</b>                   | <b>2</b>  |
| <b>2 Spezifikation</b>                | <b>4</b>  |
| 2.1 Funktionen . . . . .              | 4         |
| 2.2 Schaltung . . . . .               | 4         |
| 2.2.1 Signale . . . . .               | 4         |
| 2.2.2 Todo . . . . .                  | 5         |
| 2.3 HDL-Designer Projekt . . . . .    | 5         |
| <b>3 Komponenten</b>                  | <b>6</b>  |
| 3.1 Synchro Schlittenaufbau . . . . . | 6         |
| 3.2 FPGA-Platine . . . . .            | 7         |
| 3.3 Knöpfe und LEDs . . . . .         | 8         |
| <b>4 Bewertung</b>                    | <b>9</b>  |
| <b>5 Erste Schritte</b>               | <b>10</b> |
| 5.1 Tips . . . . .                    | 10        |
| <b>Literatur</b>                      | <b>11</b> |
| <b>Akronyme</b>                       | <b>11</b> |



## 1 Einführung

Das Projekt hat die Aufgabe das erlernte Wissen am Ende des Semesters an einem praktischen Beispiel direkt anzuwenden. Es soll ein Gleichstrommotor so angesteuert werden, damit ein Schlitten auf einer Schraube, vordefinierte Punkte präzise anfahren kann. Das System kann in der Abbildung 2 betrachtet werden.

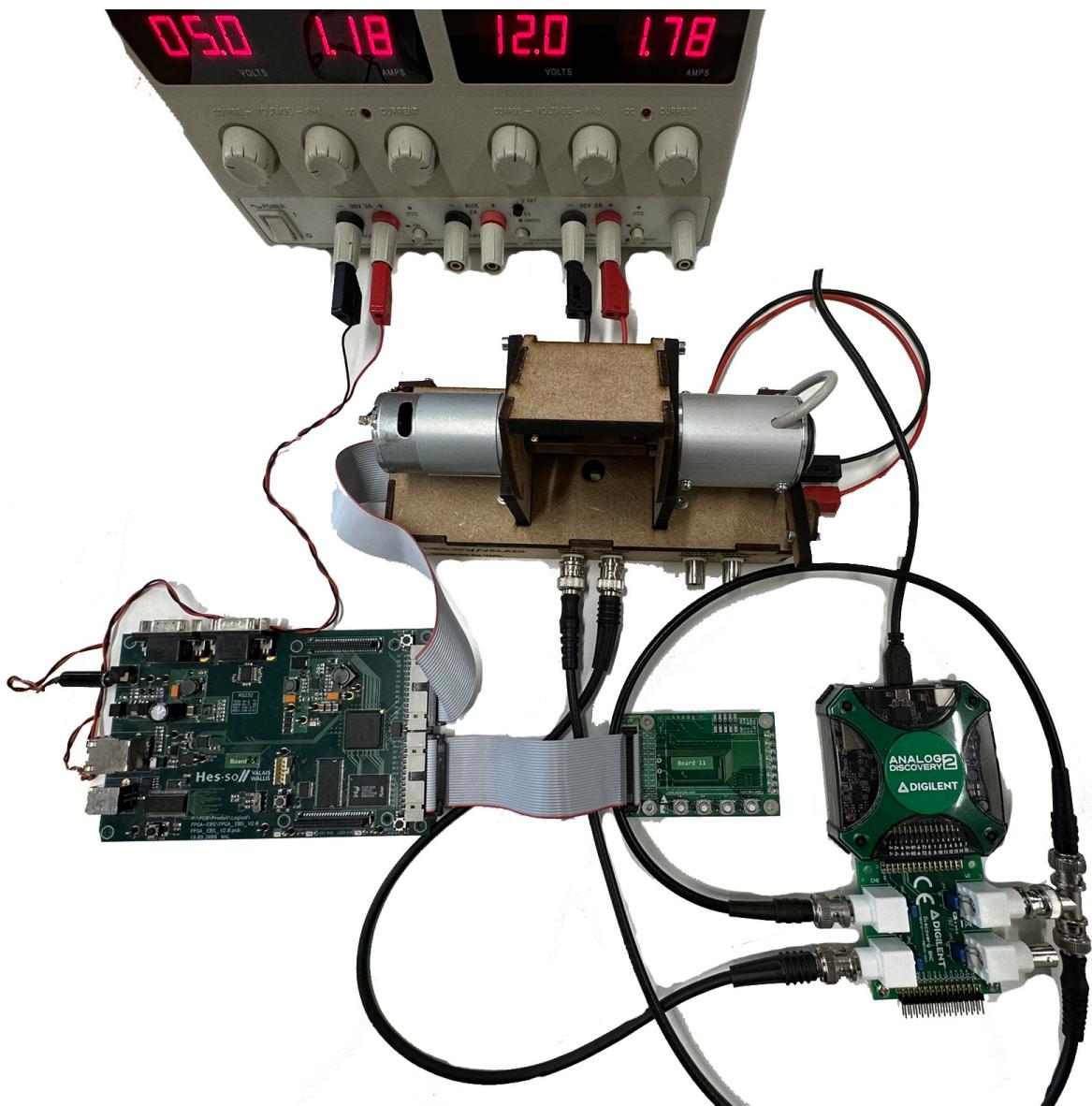
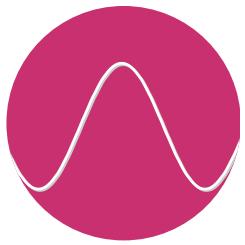
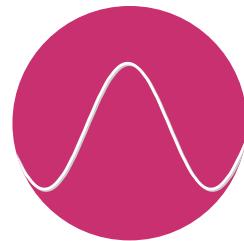


Abbildung 1: Hardwareaufbau Synchro

Die Aufgabe besteht aus einer klar definierten minimalen [Spezifikation](#) (Kapitel 2), welche von der Entwicklungsgruppe mit zusätzlichen Funktionen optional erweitert werden kann. Den Ideen sind hier keine Grenzen gesetzt, als Beispiel kann das [LCD](#) Display benutzt werden um bestimmte Informationen anzuzeigen.



Mithilfe von Zusatzfunktionen können einige Extrapunkte erarbeitet werden.



## 2 Spezifikation

### 2.1 Funktionen

Das Referenzsignal und der Generatorausgang werden mit Hilfe von zwei Komparatoren auf 1 Bit digitalisiert. Die Digitalschaltung empfängt diese Signale und steuert einen mit dem Generator gekoppelten Gleichstrommotor.

### 2.2 Schaltung

Die Schaltung kann über 4 Tasten gesteuert werden. Sie kann Informationen auf einer Reihe von 8 LEDs anzeigen.

Die Schaltung funktioniert wie folgt:

- Die Differenz zwischen der Netzperiode und der Generatorperiode wird berechnet.
- Wenn der Generator zu langsam ist, wird die Spannung des DC-Motors erhöht; wenn der Generator zu schnell ist, wird die Spannung des DC-Motors gesenkt.

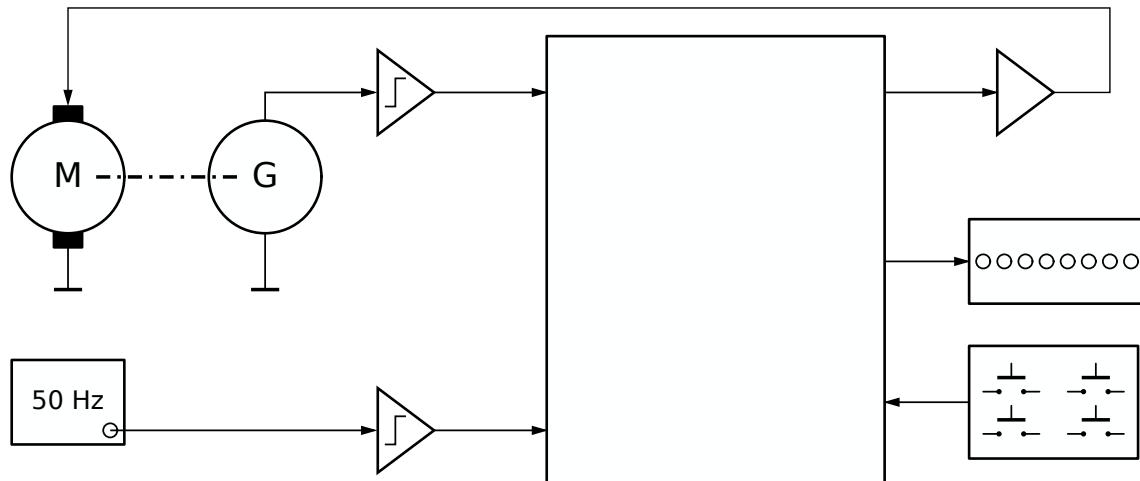
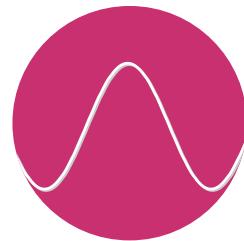


Abbildung 2: Schema des Systems

#### 2.2.1 Signale

Die Signale des Systems sind:

- **mains** und **generator** erzeugen bei jeder Umdrehung des Motors bzw. des Generators einen Impuls bei '1'. Sie ermöglichen durch Zählen und Vorkenntnis der Taktfrequenz die Bestimmung der beiden Drehfrequenzen.
- Der Vektor **buttons** steht für die Schaltflächen auf der Karte [Knöpfe und LEDs](#).
- Der Vektor **leds** steht für die [LEDs](#) auf der Karte [Knöpfe und LEDs](#).
- Das **PWM**-Signal steuert den Motor. Die maximale **PWM**-Frequenz beträgt 100kHz.
- Mit dem Vektor **testOut** können Sie Debug-Signale ausgeben, um sie zu messen.



### 2.2.2 Todo

Das System lässt sich in vier Hauptblöcke unterteilen:

- Zwei Periodenzähler für den Motor und den Generator
- Ein Komparator, der bestimmt, ob der Motor gebremst, beschleunigt oder unverändert gelassen werden soll.
- Ein Regler, mit dem man den Duty-Cycle der **PWM** einstellen kann.
  - Wenn der Generator langsamer ist, erhöhe D+
  - Wenn der Motor langsamer läuft, erhöhe D-
  - Ansonsten den Duty-Cycle nicht ändern
- Ein **PWM**-Modulator, der ein Ausgangssignal von maximal 100 kHz mit einem Duty-Cycle gemäß der gegebenen Amplitude erzeugt.

## 2.3 HDL-Designer Projekt

Ein vordefiniertes HDL-Designer Projekt kann im [Cyberlearn](#) heruntergeladen oder geklont werden. Die Dateistruktur des Projektes sieht folgendermassen aus:

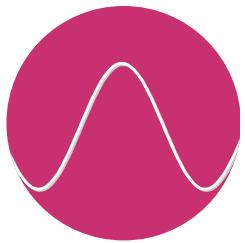
```
eln synchro
+--Board/                                # Project and files for programming the FPGA
|   +-concat/                             # Complete VHDL file including PIN-UCF file
|   +-ise/                                # Xilinx ISE project
+--GeneratorControl/                      # Library for the components of the student solution
+--GeneratorControl_test/                 # Library for the simulation testbenches
+--img/                                    # Pictures
+--Libs/                                   # External libraries which can be used e.g. gates, io, sequenti
+--Prefs/                                  # HDL-Designer settings
+--Scripts/                               # HDL-Designer scripts
+--Simulation/                            # Modelsim simulation files
```



Der Pfad des Projektordners darf keine Leerzeichen enthalten.



Im Projektordner *doc/* können viele wichtige Informationen gefunden werden. Datenblätter, Projektbewertung sowie Hilfsdokumente für HDL-Designer um nur einige zu nennen.



### 3 Komponenten

Das System besteht aus 3 verschiedenen Hardwareplatten, welche in der Abbildung ?? ersichtlich sind:

- Eine Baugruppe..., siehe Abbildung 3
- einer **FPGA** Entwicklungsplatine, siehe Abbildung 6
- einer Steuerplatine mit 4 Tasten und 8 **LEDs**, siehe Abbildung 7

#### 3.1 Synchro Schlittenaufbau

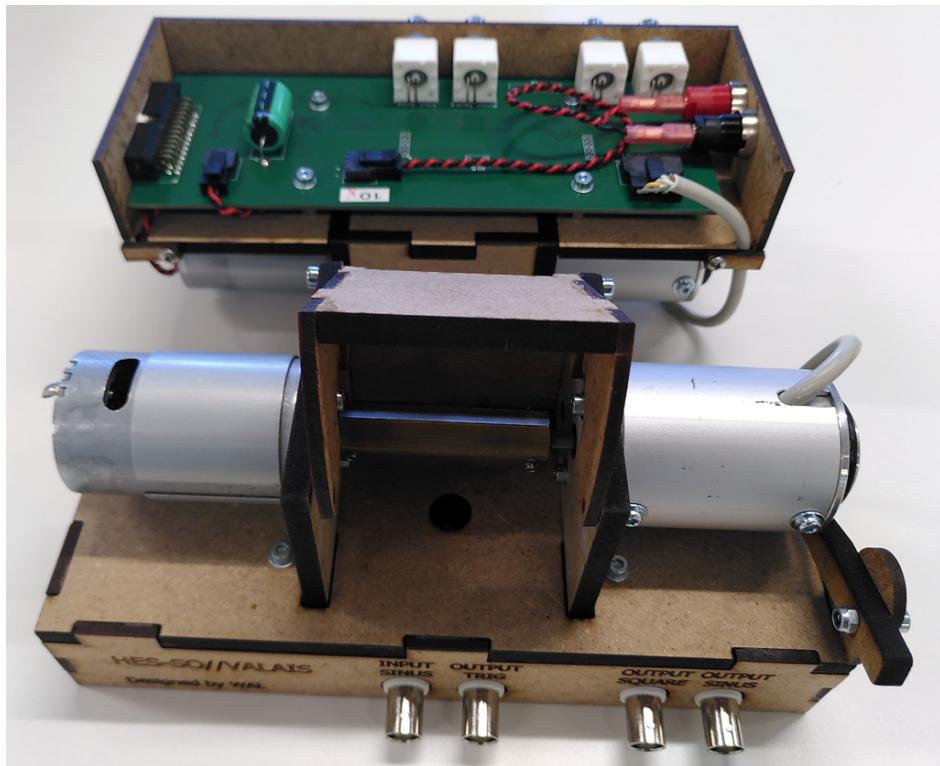


Abbildung 3: Synchro Schlittenaufbau

Um eine korrekte Regelung des Systems zu beobachten, kann ein 50-Hz-Sinus in den **Input sinus**-Eingang des Systems eingespeist werden. Dieser steuert die unter der Motor-Generator-Kopplungsachse befindliche **LED**.

Bei einer Schwingung mit 50 Hz und einem System, das mit der gleichen Frequenz geregelt wird, erscheint die Achse statisch.

Es ist möglich, das digitale Oszilloskop **Analog Discovery 2** mit seinem Ausgang **W1** zu verwenden, um dieses Signal einzuspeisen. Öffnen Sie einfach die Datei **Board/synchro.dwf3work** und klicken Sie auf **Run** in der Registerkarte **Wavegen 1**:

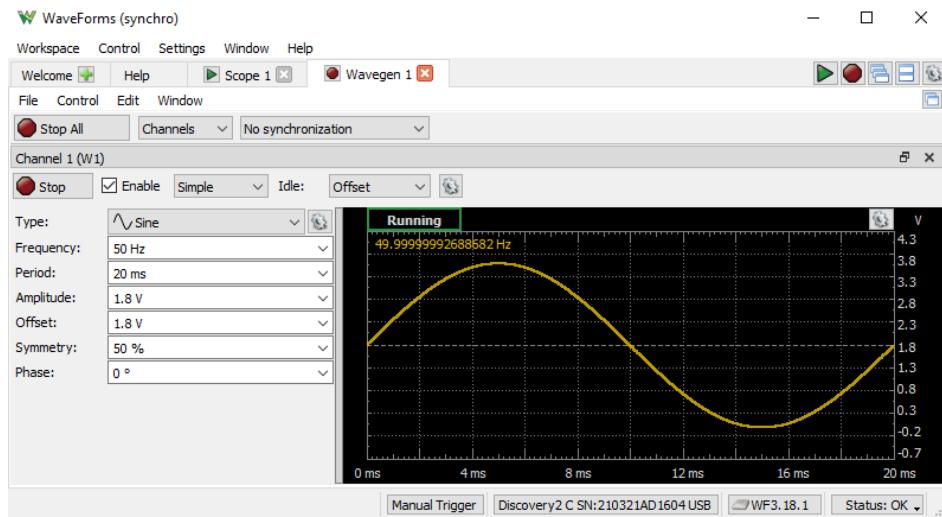
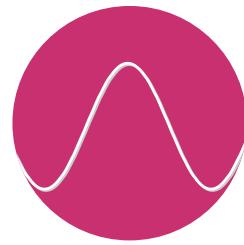


Abbildung 4: Sinus-Einstellungen - Analog Discovery 2

### 3.2 FPGA-Platine

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [9]. Diese beherbergt eine **Xilinx Spartan xc3s500e FPGA** [**Spartan3FPGAFamily**] [13] und verfügt über viele verschiedene Schnittstellen (**Universal Asynchronous Receiver Transmitter (UART)**, **Universal Serial Bus (USB)**, Ethernet, etc.). Der benutzte Oszillatior erstellt ein Taktsignal (*clock*) mit einer Frequenz von  $f_{clk} = 66MHz$  [4].

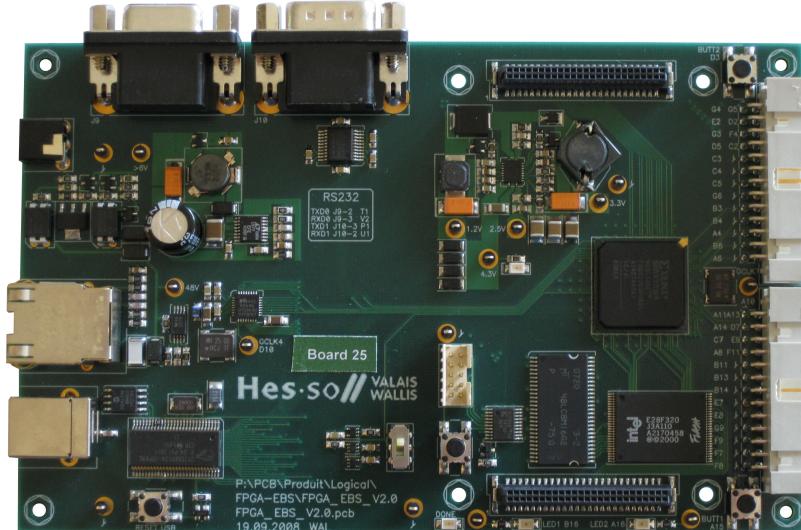
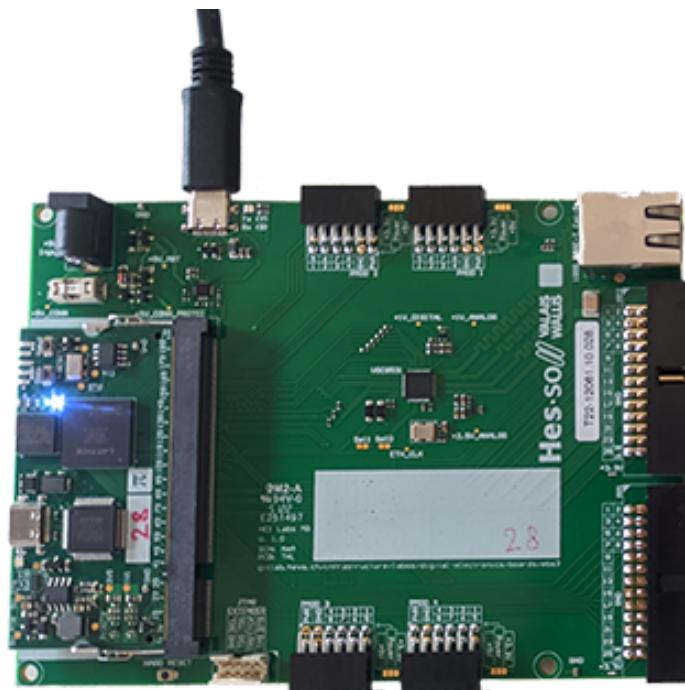
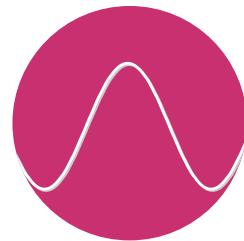


Abbildung 5: **FPGA Platine** [9]

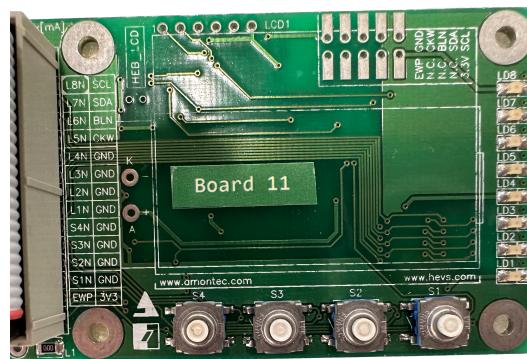
Auf der EBS3-Karte erzeugt der verwendete Oszillatior ein Taktsignal (*clock*) mit einer Frequenz von  $f_{clk} = 100MHz$ , die durch PLL auf  $f_{clk} = 60MHz$  reduziert wird.

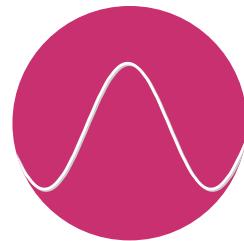
Abbildung 6: EBS3 **FPGA** Platine [2]

Die Simulators sind standardmäßig für die EBS3 boards eingestellt. Um sie zu ändern, öffnen Sie einen Block von testbench **xxx\_tb** und doppelklicken Sie auf die **Pre-User**-Deklarationen (oben links auf der Seite), um die Variable **clockFrequency** auf den gewünschten clock-Wert zu ändern.

### 3.3 Knöpfe und **LEDs**

Die Platine mit den Knöpfen und **LEDs** [10] wird an die **FPGA** Platine angeschlossen. Sie hat 4 Tasten und 8 **LEDs**, die im Design verwendet werden können. Falls gewünscht kann diese Platine mit einer **LCD** Anzeige ausgestattet werden [11] [5].

Abbildung 7: Knöpfe-**LED-LCD** Platine [10]



## 4 Bewertung

Im Ordner `doc/` zeigt die Datei `evaluation-bewertung-cursor.pdf` das detaillierte Bewertungsschema, Tabelle 1.

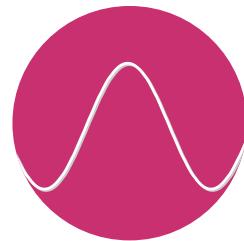
Die Schlussnote beinhaltet den Bericht, den Code sowie eine Präsentation eurerseits des Systems.

| Evaluierter Aspekt                  | Punkte     |
|-------------------------------------|------------|
| <b>Bericht</b>                      | <b>55</b>  |
| Einleitung                          | 3          |
| Spezifikation                       | 5          |
| Entwurf                             | 20         |
| Verifizierung und Validation        | 10         |
| Integration                         | 9          |
| Schlussfolgerung                    | 3          |
| Formale Aspekte des Berichtes       | 5          |
| <b>Funktionalität der Schaltung</b> | <b>30</b>  |
| <b>Qualität der Lösung</b>          | <b>10</b>  |
| <b>Präsentation</b>                 | <b>10</b>  |
| <b>Total</b>                        | <b>105</b> |

Tabelle 1: Bewertungsraster



Das Bewertungsraster gibt bereits Hinweise über die Struktur des Berichtes. Für einen guten Bericht konsultieren Sie das Dokument "Wie verfasst man einen Projektbericht?" [3]



## 5 Erste Schritte

Um mit dem Projekt zu beginnen, kann folgendermassen vorgehen werden:

- Lest die obigen Spezifikationen und Informationen genau durch.
- Schaut euch die Hardware und testet das vorprogrammierte Programm.
- Stöbert durch die Dokumente im Ordner *doc/* eures Projektes.
- Entwickelt ein detailliertes Blockdiagramm. Die Signale und deren Funktionen solltet Ihr erklären können.
- Implementierung und Simulation der verschiedenen Blöcken.
- Testen der Lösung auf der Platine und finden etwaiger Fehler .

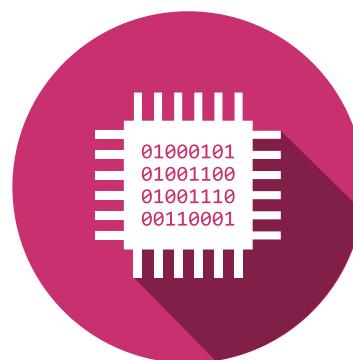
### 5.1 Tips

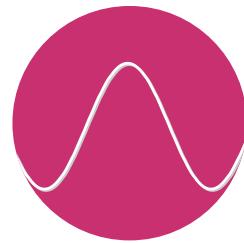
Anbei noch einige zusätzlichen Tips um Probleme und Zeitverlust zu vermeiden:

- Teilt das Problem in verschiedene Blöcke auf, benutzt hierzu das leere Toplevel Dokument (*synchro-toplevel-empty.pdf*). Es ist ein ausgeglichener Mix zwischen Anzahl Komponenten und Komponentengrösse empfohlen.
- Analysiert die verschiedenen Ein- sowie Ausgangssignale, hierzu sollten teilweise die Datenblätter zu Hilfe genommen werden.
- Beachtet bei der Erstellung des Systems das DiD Kapitel “Methodologie für die Entwicklung von digitalen Schaltungen (MET)” [6].



Vergesst nicht Spass zu haben 😊.





## Literatur

- [1] Agilent Technologies. *Datasheet Agilent AEDB-9140 Series Three Channel Optical Incremental Encoder Modules with Codewheel, 100 CPR to 500 CPR*. 2005.
- [2] Amand Axel. *Schematic: FPGA-EBS3 v1.0*. 2023.
- [3] Christophe Bianchi, François Corthay und Silvan Zahno. *Wie Verfasst Man Einen Projektbericht?* 2021.
- [4] CTS. *Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator*. 2006.
- [5] Electronic Assembly. *Datasheet: DOGM Graphics Series 132x32 Dots*. 2005.
- [6] François Corthay, Silvan Zahno und Christophe Bianchi. *Methodologie Für Die Entwicklung von Digitalen Schaltungen*. 2021.
- [7] Olivier Walpen. *Schematic: Cursor Chariot Power Circuit*. 2009.
- [8] *Rotary Encoder*. In: Wikipedia. 23. Aug. 2021. URL: [https://en.wikipedia.org/w/index.php?title=Rotary\\_encoder&oldid=1040238329](https://en.wikipedia.org/w/index.php?title=Rotary_encoder&oldid=1040238329) (besucht am 20.11.2021).
- [9] Silvan Zahno. *Schematic: FPGA-EBS v2.2*. 2014.
- [10] Silvan Zahno. *Schematic: Parallelport HEB LCD V2*. 2014.
- [11] Sitronix. *Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver*. 2006.
- [12] STMicroelectronics. *Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller*. 2003.
- [13] Xilinx. *Datasheet Spartan-3E FPGA Family*. 2008.

## Akronyme

**FPGA** Field Programmable Gates Array. 6–8

**LCD** Liquid Crystal Display. 2, 8

**LED** Light Emitting Diodes. 4, 6, 8

**PWM** Pulse Width Modulation. 4, 5

**UART** Universal Asynchronous Receiver Transmitter. 7

**USB** Universal Serial Bus. 7