



Chronometer (Chrono)

Vorlesung Digitales Design (DiD)



Orientierung: Systemtechnik (Synd)

Spezialisierung: Infotronics (IT)

Kurs: Digitales Design (DiD)

Authoren: Silvan Zahno, Axel Amand, François Corthay, Christophe Bianchi

Datum: 13.03.2025

Version: v3.0



Inhalt

1 Einführung	3
2 Spezifikation	4
2.1 Funktionen	4
2.2 Schaltung	4
2.3 Szenario (Beispiel)	5
2.4 HDL-Designer Projekt	7
3 Komponenten	8
3.1 Uhrenzifferblatt	8
3.2 Motorsteuerungsschaltung	8
3.2.1 Schrittmotor	8
3.3 Reed-Relais	10
3.4 FPGA-Platine	10
3.5 Knöpfe und LED	12
4 Bewertung	13
5 Erste Schritte	14
5.1 Tips	14
Glossar	15



1 | Einführung

Ziel dieses Projekts ist es, am Ende des Semesters theoretisches Wissen in einem praktischen Rahmen anzuwenden. Das Ziel ist es, einen Schrittmotor zu steuern, um einen Uhrzeiger präzise auf einem Zifferblatt zu bewegen und so einen einfachen Chronometer zu simulieren. Das System ist in [Abbildung 1](#) dargestellt.

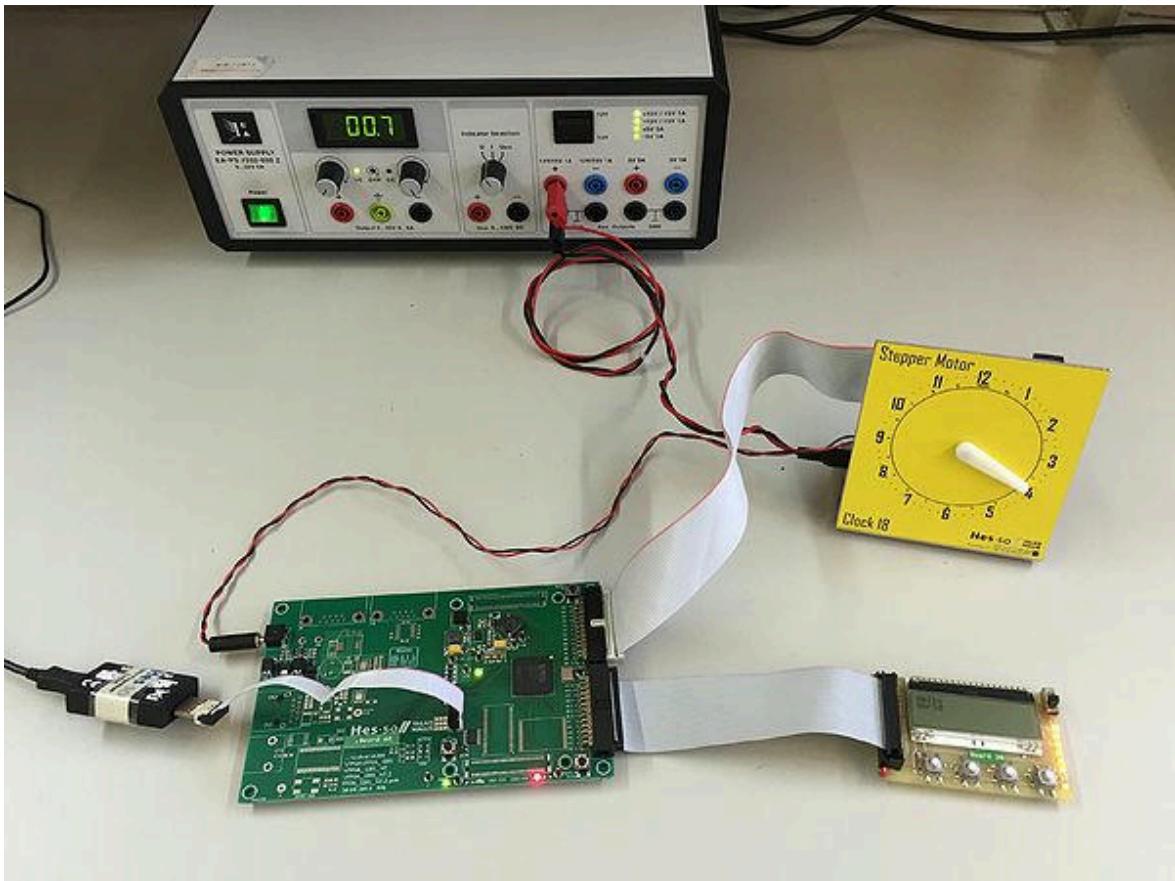


Abbildung 1 - Hardwareaufbau Chrono (EBS2)

Die Mindestspezifikationen (siehe [Abschnitt 2](#)) müssen erfüllt sein, aber die Studierenden werden ermutigt, zusätzliche Funktionen zu implementieren. Beispielsweise kann ein [Liquid Crystal Display \(LCD\)](#)-Bildschirm verwendet werden, um verschiedene Informationen anzuzeigen.



Durch die Implementierung zusätzlicher Funktionen können zusätzliche Punkte in der abschliessenden Bewertung erzielt werden.



2 | Spezifikation

2.1 Funktionen

Die grundlegenden Funktionen des Systems sind:

- **Restart:** Wenn die Restart-Taste gedrückt wird, kehrt der Zeiger zur 12-Uhr-Position zurück, die von einem Reed-Relais ([Abschnitt 3.3](#)) in der Nähe des Schrittmotors ([Abschnitt 3.2.1](#)) erkannt wird.
- **Start:** Durch Drücken der Start-Taste bewegt sich der Zeiger mit 1/60 einer Umdrehung pro Sekunde.
- **Stop:** Durch Drücken der Stopp-Taste bleibt der Zeiger in seiner aktuellen Position stehen.

Das Grundsystem behandelt keine fehlerhaften Benutzereingaben, wie das Drücken von **restart**, wenn der Zeiger bereits auf 12 Uhr steht.

Das Diagramm der grundlegenden Funktionalitäten ist in der [Abbildung 2](#) dargestellt.

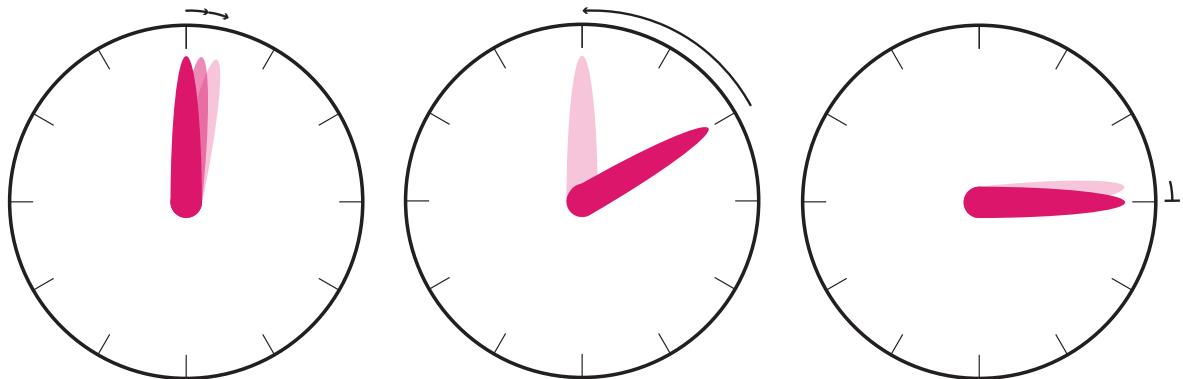


Abbildung 2 - Diagramm der grundlegenden Funktionsweisen. Von links nach rechts: **start**, **restart** und **stop**.

2.2 Schaltung

Der Schrittmotor steuert die Handbewegung, wie in [Abbildung 3](#) gezeigt.

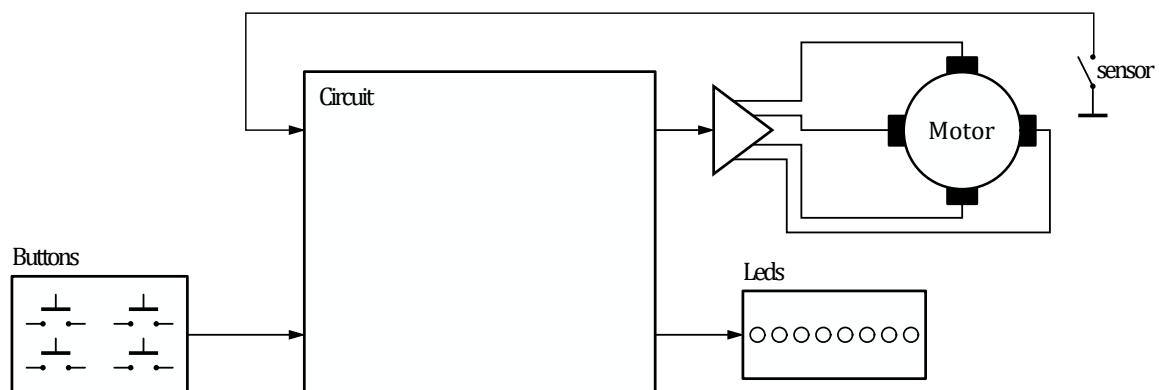


Abbildung 3 - Chrono Schaltung

Der Schaltkreis funktioniert wie folgt:



- Der Schrittmotor ([Abschnitt 3.2.1](#)) wird von vier Steuersignalen gesteuert: **coil1**, **coil2**, **coil3** und **coil4**.
- Ein Reed-Relais ([Abschnitt 3.3](#)) in der 12-Uhr-Position erkennt, wenn der Zeiger zum Start zurückkehrt.
- Drei Tasten (**restart**, **start**, **stop**) steuern das System, mit einer optionalen vierten Taste (`button_4`) für zusätzliche Funktionen [1].
- TestOut**-Pins können zum Debuggen oder Steuern von **Light Emitting Diode (LED)** verwendet werden.

[Abbildung 4](#) zeigt das Top-Level-Design des Schaltkreises, einschliesslich aller Verbindungen zum **Field Programmable Gate Array (FPGA)**-Board. Sie finden die Datei auch im Repository ([chrono-toplevel-empty.pdf](#)).

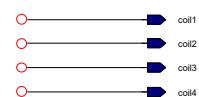
Buttons



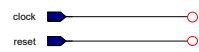
12 o'clock Sensor



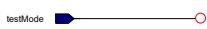
Stepper Motor Coils



Clock & Reset



Testmode only for simulation



Debug Signal (Leds)



Abbildung 4 - Leere Toplevel Schaltung

2.3 Szenario (Beispiel)

In der [Abbildung 5](#) und [Abbildung 6](#) werden zwei verschiedene Szenarien dargestellt. Zunächst wird die Taste **restart** gedrückt und der Zeiger bewegt sich mit voller Geschwindigkeit in die Ausgangsposition (**sensor**). Wenn man dann die Taste **start** drückt, beginnt sich der Zeiger im Uhrzeigersinn zu drehen. Sobald sich der Zeiger bewegt hat, wird er durch Drücken der Taste **stop** an seiner aktuellen Position angehalten. Durch Drücken von **start** oder **restart** kann der Timer dann neu gestartet bzw. auf Null zurückgesetzt werden.

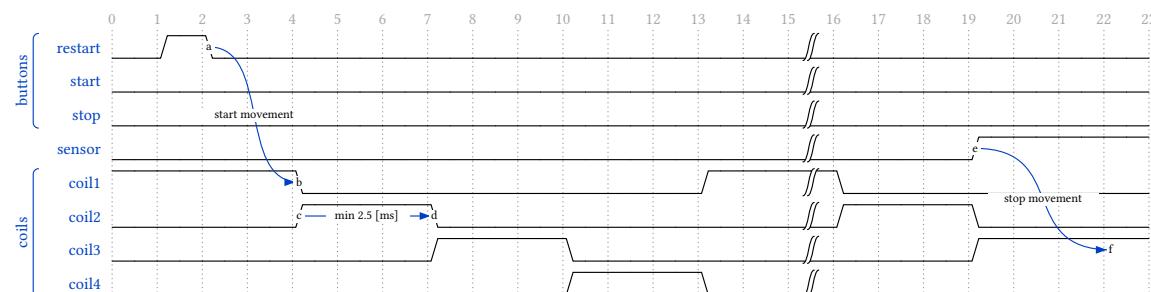


Abbildung 5 - Chrono Szenario - Restart

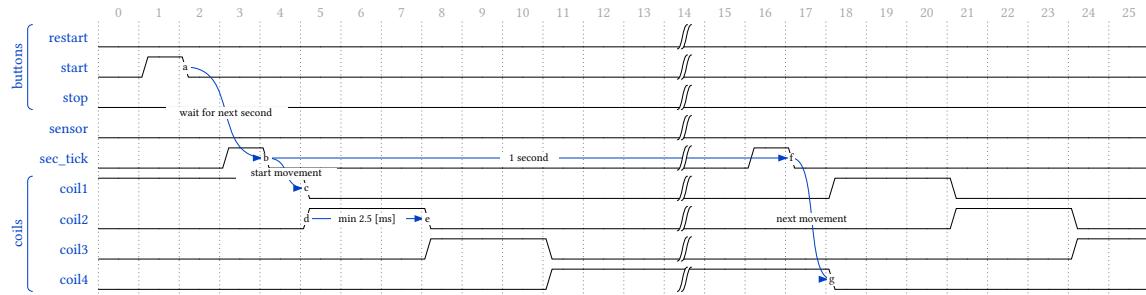


Abbildung 6 - Chrono Szenario - Start



Die obigen Szenarien sind Beispiele, es liegt an den Studenten diese zu kompletieren.



2.4 HDL-Designer Projekt

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

```
did_chrono
+--Board/          # Project and files for programming the FPGA
|   +-concat/      # Complete VHDL file including PIN-UCF file
|   +-ise/          # Xilinx ISE project
+--Chrono/         # Library for the components of the student solution
+--Chrono_test/    # Library for the simulation testbenches
+--doc/            # Folder with additional documents relevant to the project
|   +-Board/        # All schematics of the hardware boards
|   +-Components/  # All data sheets of hardware components
+--img/             # Pictures
+--Libs/            # External libraries which can be used e.g. gates, io, sequential
+--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 drei verschiedenen Hardwareplatten, die in der [Abbildung 1](#) zu sehen sind.

- Eine Chrono-Baugruppe mit einer „[Printed Circuit Board \(PCB\)](#)“-Platine, die den Motor steuert und den Sensor ausliest, siehe [Abbildung 7](#).
- Ein Entwicklungsboard [FPGA](#) ([Abbildung 12](#) oder [Abbildung 13](#)).
- Eine Steuerkarte mit 4 Tasten und 8 [LEDs](#) ([Abbildung 14](#)).

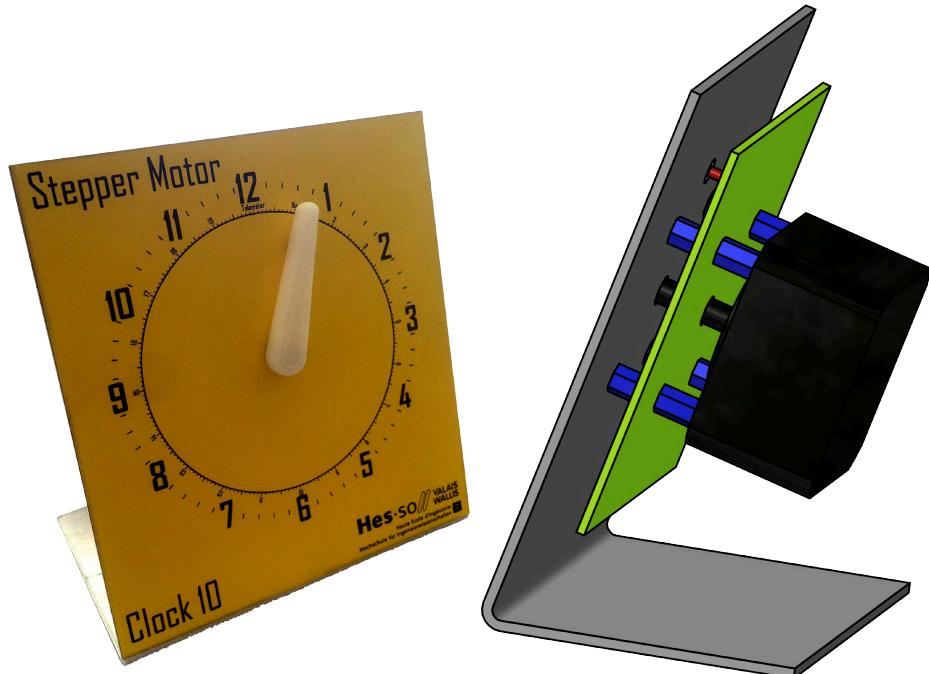


Abbildung 7 - Zusammenbau des Zifferblatts des Chronographen

3.1 Uhrenzifferblatt

Der Aufbau der Uhr besteht aus dem Schrittmotor ([Abschnitt 3.2.1](#)), dem Reed-Relais ([Abschnitt 3.3](#)) sowie dem Uhrzeiger.

3.2 Motorsteuerungsschaltung

Der Schrittmotor des Chronometers wird mit 12 V versorgt. Die Stromversorgungsplatine besitzt eine H-Brücke, die durch digitale Signale gesteuert wird. Auf der Stromversorgungsplatine erzeugt ein 5-V-Regler die Spannung, die die [FPGA](#) Platine versorgt [2].

3.2.1 Schrittmotor

Der Schrittmotor hat die folgenden Eigenschaften, die im Datenblatt [3] nachgelesen werden können:

- 200 Schritte pro Umdrehung
- 8-12V
- 4 Phasen (Spulen)



Der Schrittmotor wird von einem H-Brücken-Treiber L6207 [4] gesteuert, siehe Abbildung 8. Die maximale Schaltfrequenz der H-Brücke liegt bei 100kHz. Dies muss bei der Erzeugung des Signals **Pulse Width Modulation (PWM)** berücksichtigt werden.



Die Erfahrung hat gezeigt, dass der Motor eine Geschwindigkeit von 1-2 Umdrehungen pro Sekunde erreichen kann.

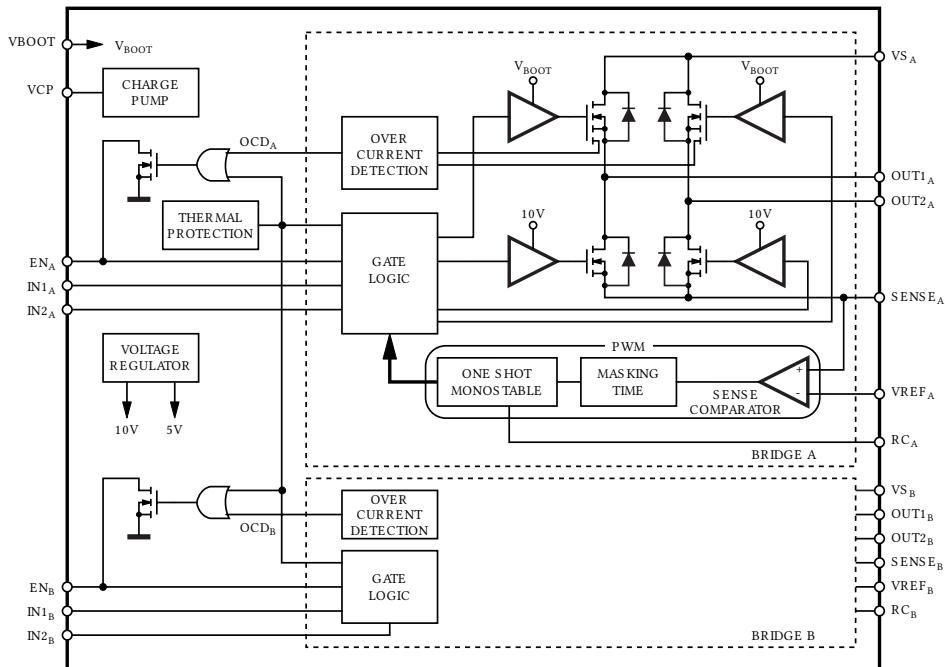


Abbildung 8 - H-Brücke L6207N Schaltung [4]

In dem in Abbildung 9 gezeigten Beispiel wird die Wicklung p_1 (gesteuert durch das Signal **coil1**) mit Strom versorgt, wodurch der Rotor in diese Richtung ausgerichtet wird (angezeigt durch den schwarzen Pfeil).

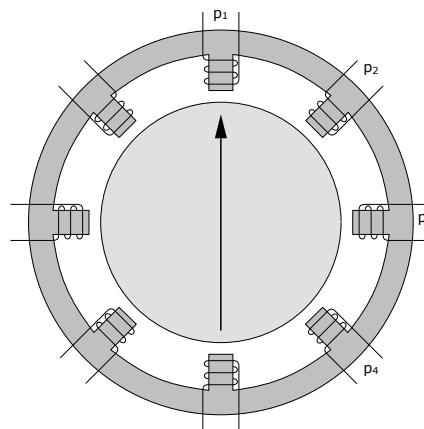


Abbildung 9 - Schaltplan des Schrittmotors.



Um die Position und die Geschwindigkeit des Schrittmotors zu steuern, werden die vier Phasen zur Erzeugung eines Magnetfelds verwendet. Der Rotor kann als einfacher Magnet modelliert werden und richtet sich so nach der Position und der Polarität des erzeugten Magnetfelds aus.

Durch das Anlegen aufeinanderfolgender Impulse an die Signale **coil1**, **coil2**, **coil3** und **coil4** kann so ein Drehfeld erzeugt werden und der Rotor wird dieser Drehung folgen.

Die Veränderung dieser Signale steuert die Position des Motors. Die Haltekraft und der Energieverbrauch können mithilfe eines **PWM**-Signals gesteuert werden (optionale Aufgabe).



Verbrennungs-/Feuergefahr! Es ist wichtig, dass Sie den Motor nicht anhalten, indem Sie 1 (oder mehr) Spule dauerhaft mit Strom versorgen! Der Motor könnte sich erhöhen und verbrennen.

3.3 Reed-Relais

Das Reed-Relais ist ein Schalter, der mithilfe von Elektromagneten geschaltet werden kann [1], [5]. Wenn sich ein Magnet in der Nähe des Sensors befindet, schliesst sich der Kontakt, siehe Abbildung Abbildung 10. Auf dem Zifferblatt der Uhr wird ein Relais (**sensor**) verwendet, um die Startposition des Zeigers am Mittag zu detektieren. Seine Position wird durch die Farbe **blau** in der Abbildung Abbildung 11 hervorgehoben, während der Magnet in **Magenta** hervorgehoben wurde.

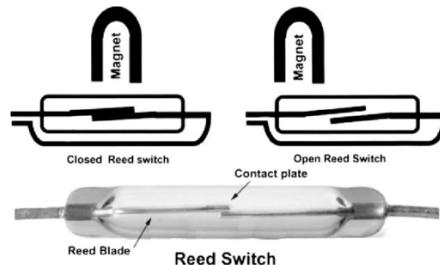


Abbildung 10 - Reed Schalter [6]

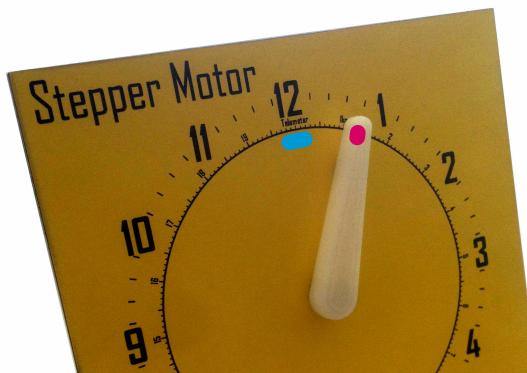


Abbildung 11 - Position des **Reed-Relais** und des **Magneten**.

3.4 FPGA-Platine

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [7]. Diese beherbergt eine **Xilinx Spartan xc3s500e FPGA** [8], [9] und verfügt über viele verschiedene Schnittstellen



(Universal Asynchronous Receiver Transmitter (UART), Universal Serial Bus (USB), Ethernet, etc.). Der benutzte Oszillator erstellt ein Taktsignal (**clock**) mit einer Frequenz von $f_{\text{clk}} = 66\text{MHz}$ [10].

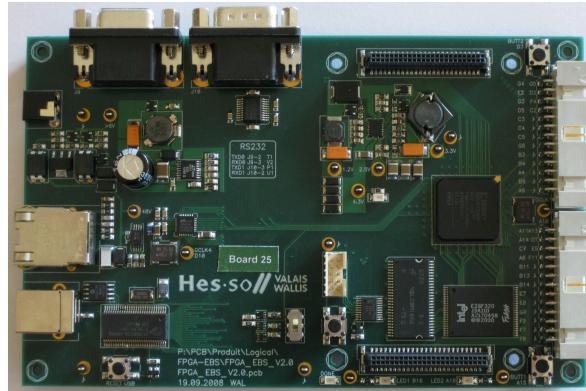


Abbildung 12 - EBS2 FPGA Platine [7]

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

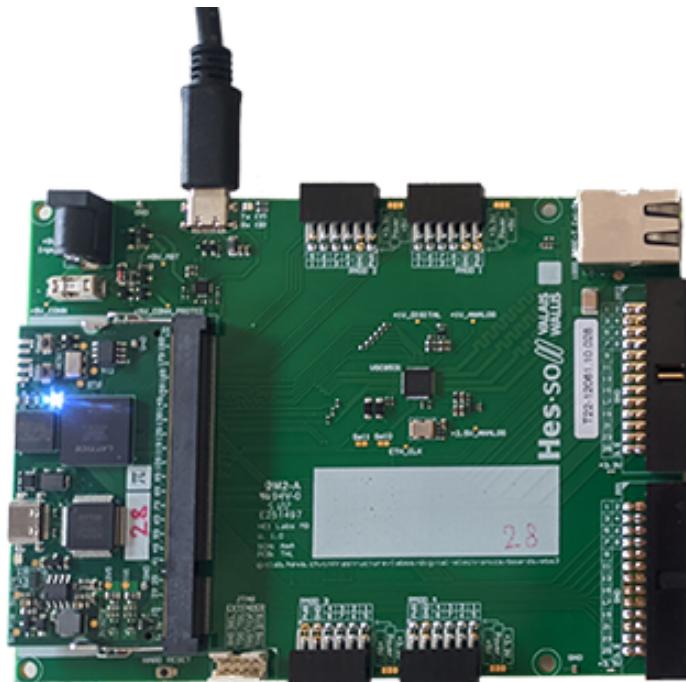


Abbildung 13 - EBS3 FPGA Platine [11]



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.



3.5 Knöpfe und LED

Die Platine mit den Knöpfen und LEDs [12] 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 [13], [14].

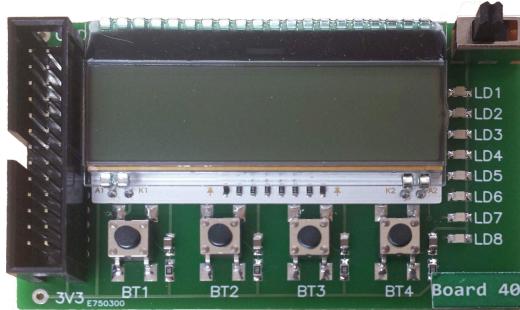


Abbildung 14 - Knöpfe-LEDs-LCD Platine [12]



4 | Bewertung

Im Ordner **doc/** zeigt die Datei **evaluation-bewertung-chrono.pdf** das detaillierte Bewertungsschema, [Tabelle 1](#).

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

Evaluierte Aspekte	Punkte
Bericht	55
Einleitung	3
Spezifikation	5
Entwurf	20
Verifizierung und Validation	10
Integration	9
Schlussfolgerung	3
Formale Aspekte des Berichtes	5
Funktionalität der Schaltung	30
Qualität der Lösung	10
Präsentation	10
Total	105

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?“ [\[15\]](#).



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 (**chrono-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)“ [16].
- Es wird empfohlen das System in zwei Schritten zu realisieren.
 - Reagieren Sie zunächst auf die Tasten und bewegen Sie den Zeiger.
 - Integrieren Sie die Bewegung auf die Sekunde und richten Sie sie auf das Zifferblatt aus.



Vergesst nicht Spass zu haben.





Glossar

FPGA – Field Programmable Gate Array [5, 8, 10, 11, 12](#)

LCD – Liquid Crystal Display [3, 12](#)

LED – Light Emitting Diode [5, 8, 12](#)

PCB – Printed Circuit Board [8](#)

PWM – Pulse Width Modulation [9, 10](#)

UART – Universal Asynchronous Receiver Transmitter [11](#)

USB – Universal Serial Bus [11](#)



Literatur

- [1] „Reed Relay“. 5. Dezember 2020. Zugegriffen: 24. November 2021. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=Reed_relay&oldid=992433034
- [2] Silvan Zahno, „Schematic: FPGA-EBS Motor v2.1“. 2009.
- [3] P. Sartoretti, „Stepper Motor-Module“. 2008.
- [4] STMicroelectronics, „Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller“. 2003.
- [5] S. Electronics, „Datasheet Reed Sensor ORD213“. 2001.
- [6] „Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850×345)“. Zugegriffen: 24. November 2021. [Online]. Verfügbar unter: <https://www.researchgate.net/profile/Sidakpal-Panaich-2/publication/51169357/figure/fig1/AS:394204346896388@1470997048549/Magnetic-reed-switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.png>
- [7] Silvan Zahno, „Schematic: FPGA-EBS v2.2“. 2014.
- [8] Xilinx, „Spartan-3 FPGA Family“. Zugegriffen: 20. November 2021. [Online]. Verfügbar unter: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>
- [9] Xilinx, „Datasheet Spartan-3E FPGA Family“. 2008.
- [10] CTS, „Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator“. 2006.
- [11] A. Amand und S. Zahno, „FPGA-EBS3 Electronic Technical Documentation“. 2022.
- [12] Silvan Zahno, „Schematic: Parallelport HEB LCD V2“. 2014.
- [13] Sitronix, „Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver“. 2006.
- [14] Electronic Assembly, „Datasheet: DOGM Graphics Series 132x32 Dots“. 2005.
- [15] Christophe Bianchi, François Corthay, und Silvan Zahno, „Wie Verfasst Man Einen Projektbericht?“. 2021.
- [16] François Corthay, Silvan Zahno, und Christophe Bianchi, „Méthodologie de Conception de Circuits Numériques“. 2021.