Chrono - Motorsteuerung

Projekt Digitales Design

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Orientierung: Systemtechnik (SYND)
Kurs: Digitales Design (DiD)
Verfasser: Christophe Bianchi, François Corthay, Silvan Zahno, Axel Amand
Datum: 11. April 2023
Version: v2.0

Inhaltsverzeichnis

1 Einführung

Ziel des Projekts ist es, das erworbene Wissen am Ende des Semesters direkt mit Hilfe eines praktisches Beispieles anzuwenden. Es geht darum, einen Schrittmotor anzusteuern, um einen Zeiger auf einem Zifferblatt einer Uhr präzise so zu bewegen, dass er sich wie eine einfache Stoppuhr verhält. Dieses Chronometersystem ist in der Abbildung 1 dargestellt.

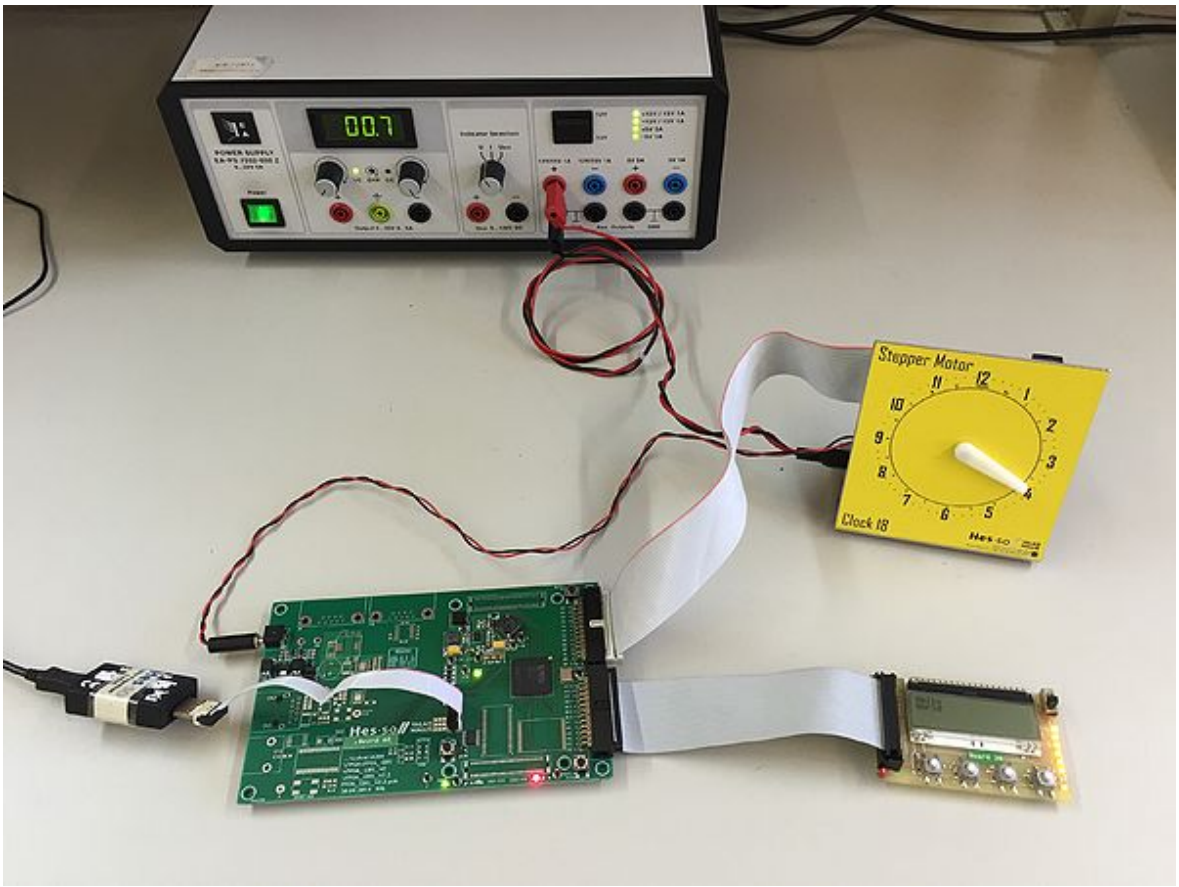


Abbildung 1: Hardwareaufbau Chrono (EBS2)

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

Die Basisfunktionen sind wie folgt definiert:

- Falls die Taste *restart* gedrückt wird, kehrt der Zeiger zur Startposition (12 Uhr) zurück, die durch eine Reed-Relais angezeigt wird, die sich in der Nähe des Schrittmotors befindet.
- Falls die Taste *start* gedrückt wird, bewegt sich der Zeiger jede Sekunde um eine 60stel Umdrehung.
- Falls die Taste *stop* gedrückt wird, bleibt der Zeiger stehen und wartet an dieser Position.

Das minimale System geht nicht mit Fällen um, in denen der Benutzer erratisch handelt, z. B. wenn er *restart* drückt, während der Zeiger bereits auf der 12-Uhr-Position steht.
Die Abbildung 2 zeigt diese Verhaltensweisen schematisch.

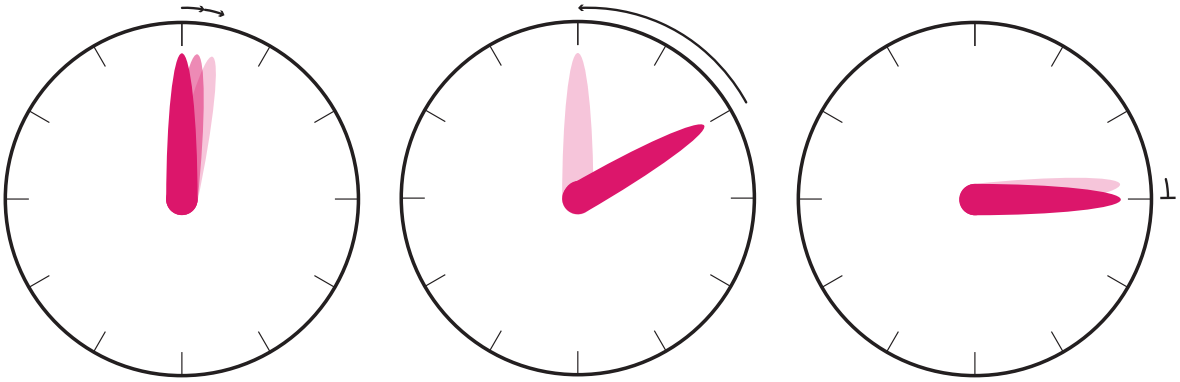


Abbildung 2: Diagramm der grundlegenden Funktionsweisen. Von links nach rechts: *start*, *restart* und *stop*.

2.2 Schaltung

Der Zeiger wird von einem Schrittmotor gesteuert, der in der Abbildung 3 schematisch aufgezeigt wird.

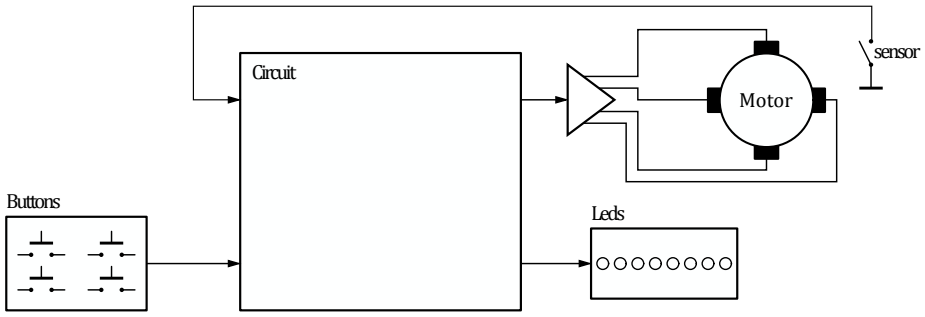


Abbildung 3: Chrono Schaltung

Die Schaltung funktioniert wie folgt:

- Der Schrittmotor wird von den vier Signalen *coil*1, *coil*2, *coil*3 und *coil*4 gesteuert. Der

Motor wird durch aufeinanderfolgende Impulse auf seine vier Spulen gesteuert.

- Ein Reed-Relais wird an der Mitternachts-/Mittagsposition des Zifferblatts der platziert [10]. Er erkennt, dass sich der Zeiger in der Startposition befindet (*sensor*).
- Drei Tasten werden zur Steuerung des Systems verwendet: *restart*, *start* und *stop*. Eine weitere Taste, $button_4$, kann für optionale Funktionen verwendet werden.
- Die *testOut*-Pins können verwendet werden, um zusätzliche Informationen über das System auszugeben, z. B. für Debuggingzwecke oder zur Kontrolle von LEDs.

Der leere Design-Toplevel (chrono-toplevel-empty.pdf) zeigt alle Signale, die an die **F**ield **P**rogrammable **G**ates **A**rray (FPGA)-Platine angeschlossen sind, siehe Abbildung 4.

Abbildung 4: Leere Toplevel Schaltung

2.3 Szenario (Beispiel)

In den Abbildungen 5 und 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.

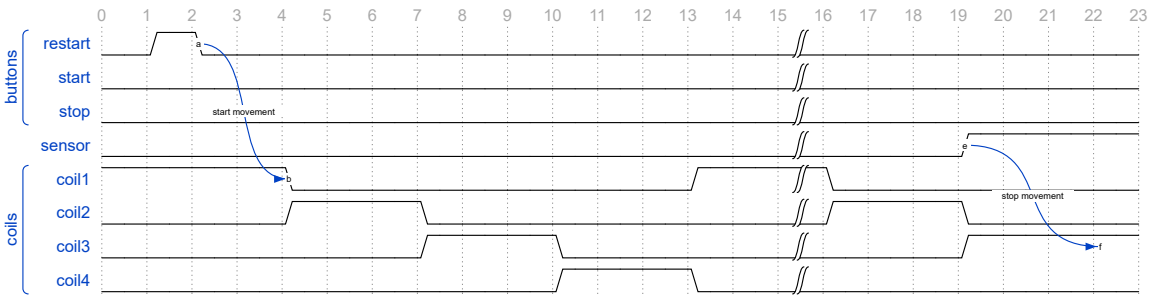


Abbildung 5: Chrono Szenario - Restart

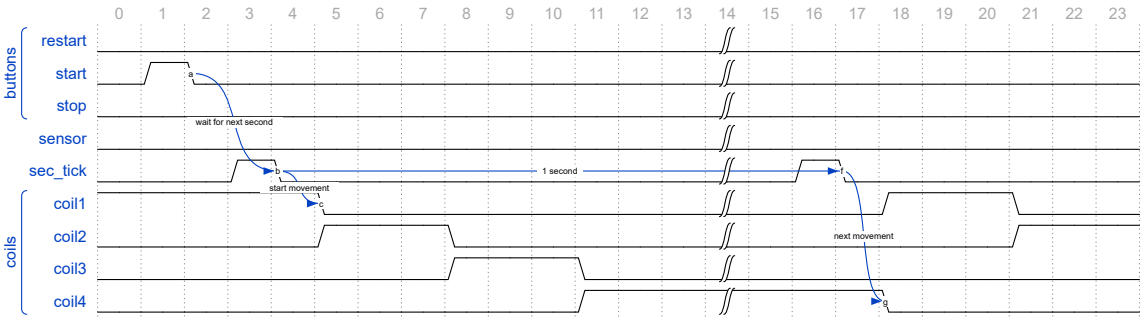


Abbildung 6: Chrono Szenario - Start

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

2.4 HDL-Designer Projekt

Ein vordefiniertes HDL-Designer Projekt kann im Cyberlearn 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 Hardwareplatinen, die in der Abbildung 1 zu sehen sind.

- Eine Chrono-Baugruppe mit einer "**P**rinted **C**ircuit **B**oard (PCB)Platine, die den Motor steuert und den Sensor ausliest, siehe Abbildung 7.
- Ein Entwicklungsboard FPGA, siehe Abbildung 13.
- Eine Steuerkarte mit 4 Tasten und 8 LEDs, siehe Abbildung 14.

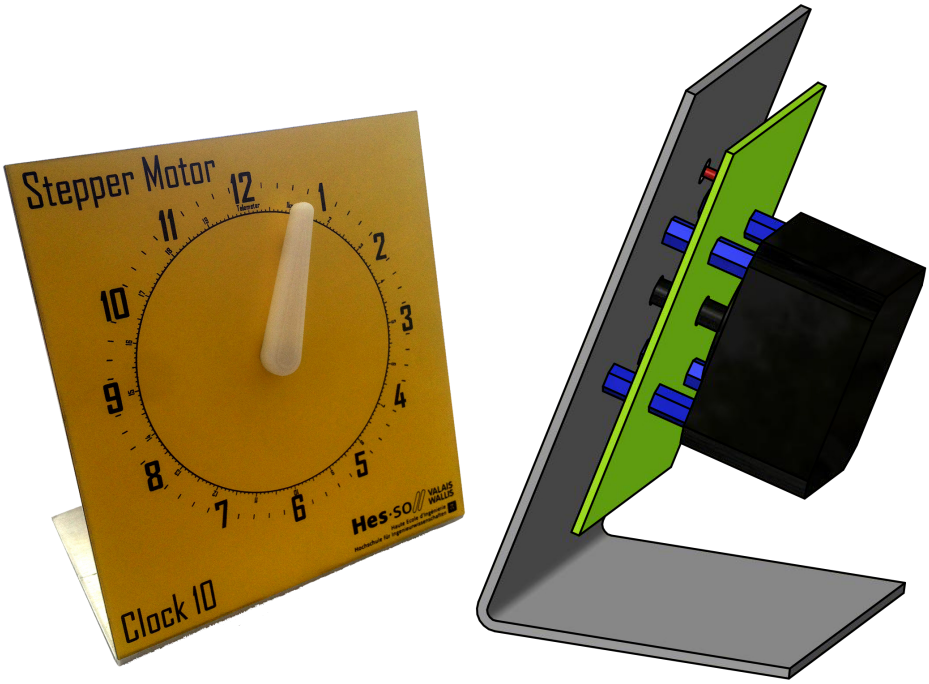


Abbildung 7: Chrono Schlittenaufbau

3.1 Uhrenzifferblatt

Der Aufbau der Uhr besteht aus dem Schrittmotor, dem Reed-Relais 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 [13].

3.2.1 Schrittmotor

Der Schrittmotor hat die folgenden Eigenschaften, die im Datenblatt [12] nachgelesen werden können:

- 200 Schritte pro Umdrehung
- 8-12V

- 4 Phasen (Spulen)

Der Schrittmotor wird von einem H-Brücken-Treiber L6207 [17] gesteuert, siehe Abbildung 8. Die maximale Schaltfrequenz der H-Brücke liegt bei $100kHz$. Dies muss bei der Erzeugung des Signals **P**ulse **W**idth **M**odulation (PWM) berücksichtigt werden.

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

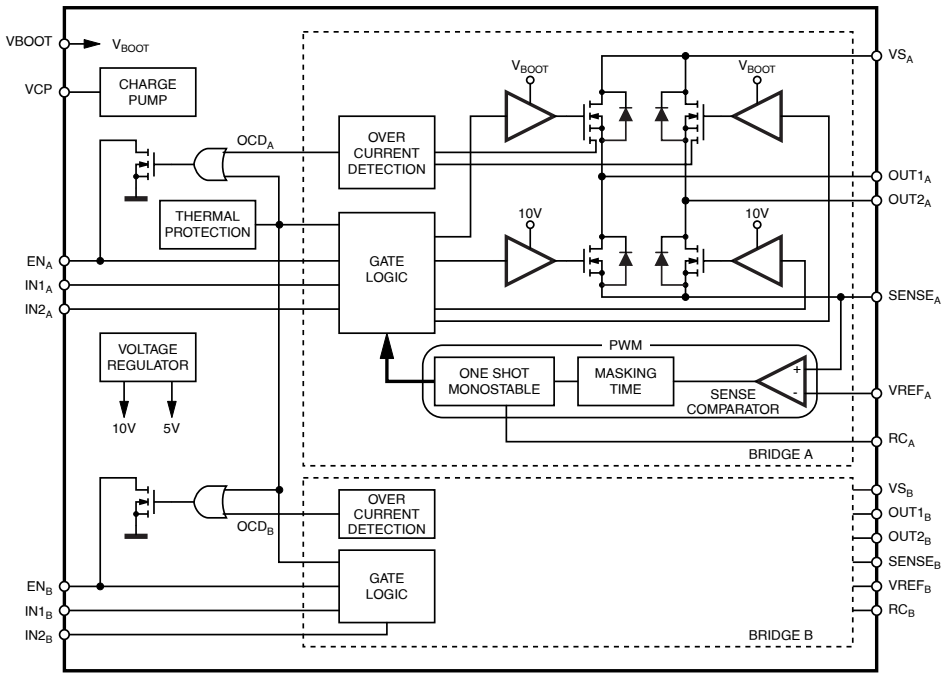


Abbildung 8: H-Brücke L6207N Schaltung [17]

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).

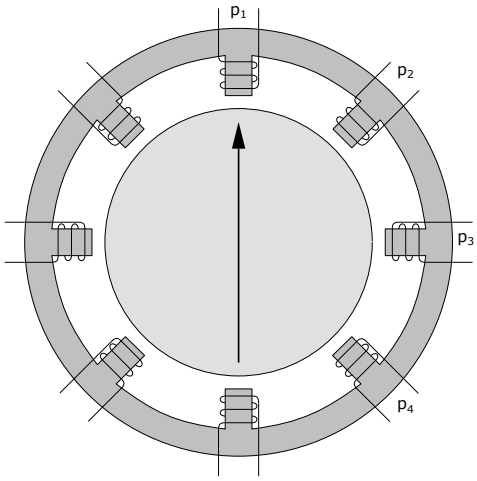


Abbildung 9: Schaltplan des Schrittmotors.

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 *coil*1, *coil*2, *coil*3 und *coil*4 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 erhitzen und verbrennen.

3.3 Reed-Relais

Das Reed-Relais ist ein Schalter, der mithilfe von Elektromagneten geschaltet werden kann [10] [6]. Wenn sich ein Magnet in der Nähe des Sensors befindet, schliesst sich der Kontakt, siehe 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 11 hervorgehoben, während der Magnet in Magenta hervorgehoben wurde.

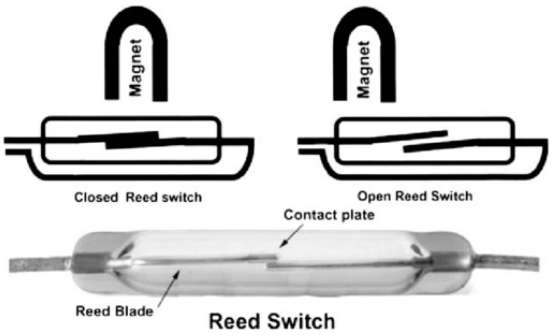


Abbildung 10: Reed Schalter [8]

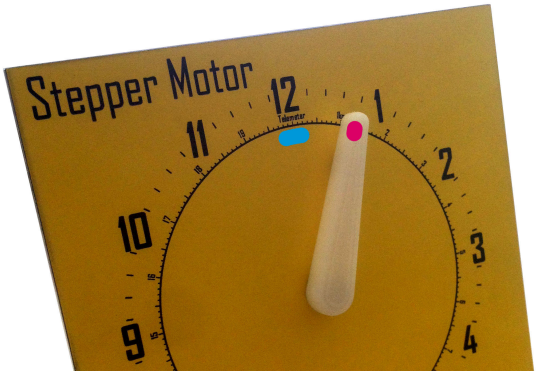


Abbildung 11: Position des Reed-Relais und des Magneten.

3.4 FPGA-Platine

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [14]. Diese beherbergt eine Xilinx Spartan xc3s500e FPGA [**Spartan3FPGAFamily**] [18] und verfügt über viele verschie-

dene Schnittstellen (**U**niversal **A**synchronous **R**eceiver **T**ransmitter (UART), **U**niversal **S**erial **B**us (USB), Ethernet, etc.). Der benutzte Oszillator erstellt ein Taktsignal ($clock$) mit einer Frequenz von $f_{clk} = 66MHz$ [4].

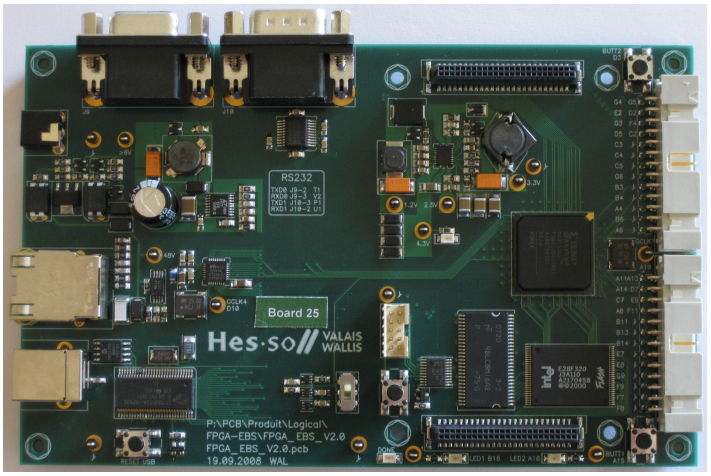


Abbildung 12: EBS2 FPGA Platine [14]

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

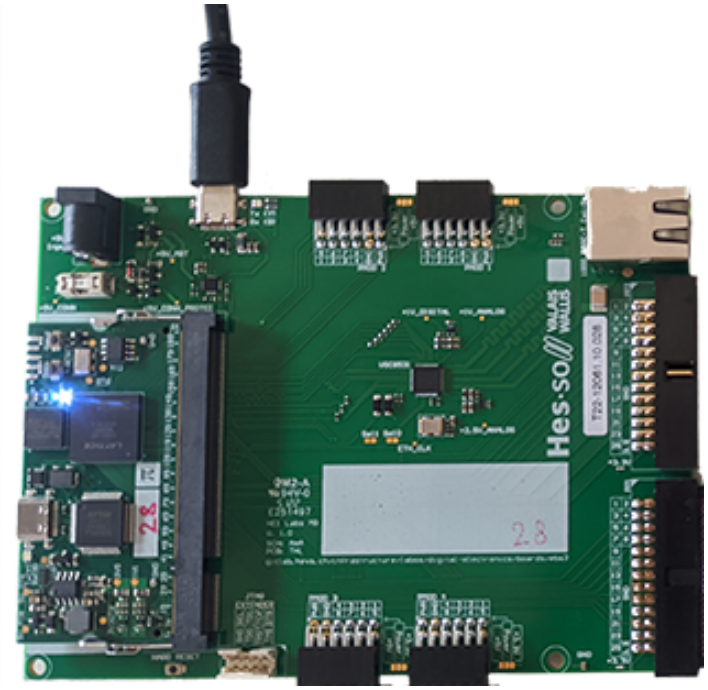


Abbildung 13: EBS3 FPGA Platine [2]

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.

3.5 Knöpfe und LEDs

Die Platine mit den Knöpfen und LEDs [15] 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 [16] [5].

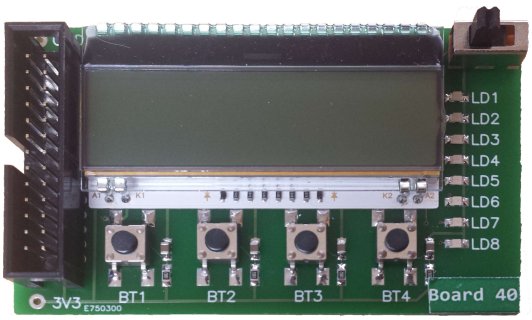


Abbildung 14: Knöpfe-LED-LCD Platine [15]

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?" [3]

.

5 Erste Schritte

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

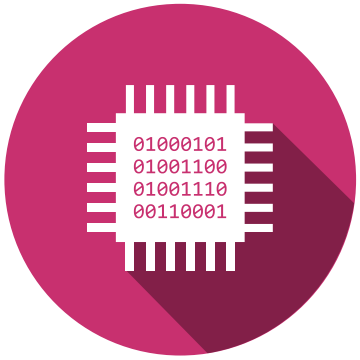
- Lest die obigen Spezifikationen und Informationen genau durch.
- Schaut euch die Hardware und testet das vorprogrammierte Programm.
- Stöbert durch die Dokumente im Ordner *doc/* eures Projektes.
- Entwickelt ein detailliertes Blockdiagramm. Die Signale und deren Funktionen solltet Ihr erklären können.
- Implementierung und Simulation der verschieden 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 (cursor-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)” [7]
- 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 .



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] Standex Electronics. *Datasheet Reed Sensor ORD213*. 2001.

[7] François Corthay, Silvan Zahno und Christophe Bianchi. *Methodologie Für Die Entwicklung von Digitalen Schaltungen*. 2021.

[8] *Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850×345)*. URL: 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 (besucht am 24. 11. 2021).

[9] Olivier Walpen. *Schematic: Cursor Chariot Power Circuit*. 2009.

[10] *Reed Relay*. In: *Wikipedia*. 5. Dez. 2020. URL: https://en.wikipedia.org/w/index.php?title=Reed_relay&oldid=992433034 (besucht am 24. 11. 2021).

[11] *Rotary Encoder*. In: *Wikipedia*. 23. Aug. 2021. URL: https://en.wikipedia.org/w/index.php?title=Rotary_encoder&oldid=1040238329 (besucht am 20. 11. 2021).

[12] Pascal Sartoretti. *Stepper Motor-Module*. 2008.

[13] Silvan Zahno. *Schematic: FPGA-EBS Motor v2.1*. 2009.

[14] Silvan Zahno. *Schematic: FPGA-EBS v2.2*. 2014.

[15] Silvan Zahno. *Schematic: Parallelport HEB LCD V2*. 2014.

[16] Sitronix. *Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver*. 2006.

[17] STMicroelectronics. *Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller*. 2003.

[18] Xilinx. *Datasheet Spartan-3E FPGA Family*. 2008.

Akronyme

FPGA **F**ield **P**rogrammable **G**ates **A**rray. 4, 7, 10, 11

LCD **L**iquid **C**rystal **D**isplay. 2, 11

LED **L**ight **E**mitting **D**iodes. 4, 7, 11

PCB **P**rinted **C**ircuit **B**oard. 7

PWM **P**ulse **W**idth **M**odulation. 8, 9

UART **U**niversal **A**synchronous **R**eceiver **T**ransmitter. 10

USB **U**niversal **S**erial **B**us. 10