

Display - Bildschirmsteuerung

Projekt Digitales Design

Hes·so  **VALAIS
WALLIS**



Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

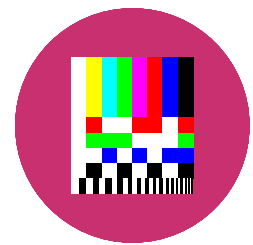
Orientierung: [Informatik und Kommunikationssysteme \(ISC\)](#)

Kurs: Digitales Design (DiD)

Verfasser: [Axel Amand](#), [Silvan Zahno](#)

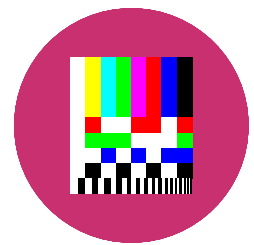
Datum: 2. Juni 2023

Version: v2.0



Inhaltsverzeichnis

| | |
|-------------------------------------|-----------|
| 1 Einführung | 2 |
| 2 Spezifikation | 3 |
| 2.1 Funktionen | 3 |
| 2.2 Schaltung | 3 |
| 2.3 VGA Timing (Beispiel) | 5 |
| 2.4 HDL-Designer Projekt | 7 |
| 3 Komponenten | 8 |
| 3.1 FPGA-Platine | 8 |
| 3.2 Knöpfe und LEDs | 9 |
| 3.3 PMod-DVI Modul | 9 |
| 4 Bewertung | 11 |
| 5 Erste Schritte | 12 |
| 5.1 Tips | 12 |
| Literatur | 13 |
| Akronyme | 14 |



1 Einführung

Ziel des Projekts ist es, das erworbene Wissen am Ende des Semesters direkt mit Hilfe eines praktischen Beispiels anzuwenden. Es geht darum, ein Display über einen **Video Graphics Array (VGA)** Schnittstelle anzusteuern, um ein vordefiniertes Bild anzuzeigen. Dieses Displaysystem ist in der Abbildung 1 dargestellt.

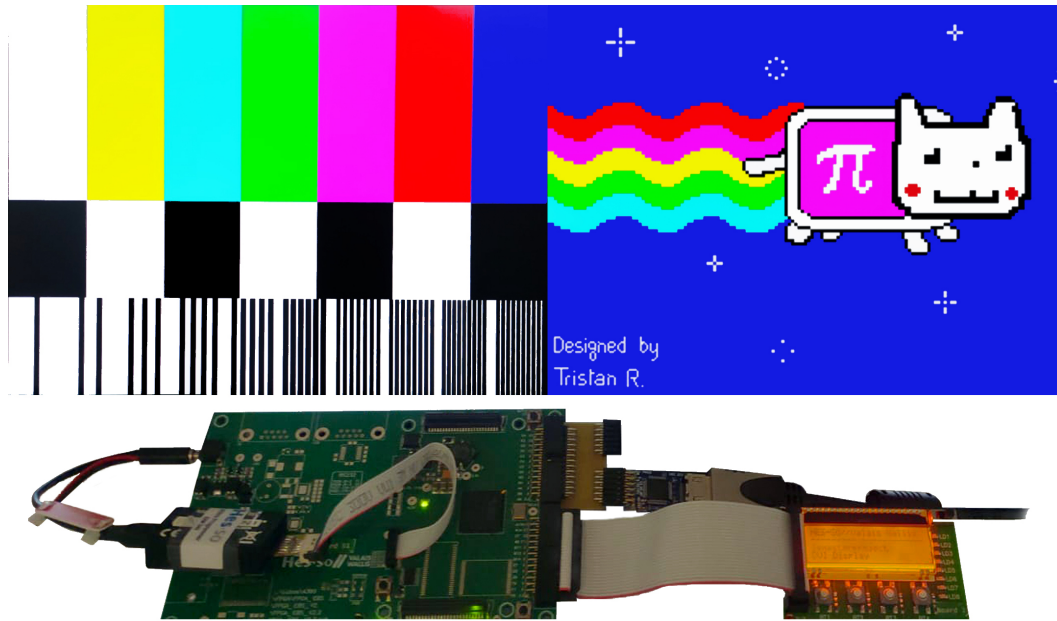
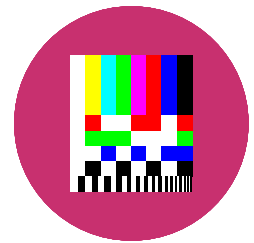


Abbildung 1: Hardwareaufbau Display

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 oder das Testbild kann verändert und erweitert werden.



Mithilfe von Zusatzfunktionen können einige Extrapunkte erarbeitet werden.



2 Spezifikation

2.1 Funktionen

Die Basisfunktionen sind wie folgt definiert:

- Falls die Taste *start* gedrückt wird, wird ein Testbild auf dem Monitor angezeigt.
- Falls die Taste *stop* gedrückt wird, wird das Testbild entfernt und der Monitor wird schwarz.
- Das Testbild besteht aus allen möglichen Farbkombinationen welche mit dem **3bit per pixel (bpp) Digital Visual Interface (DVI)** Modul möglich sind. Die Abbildung 2 zeigt ein mögliches Testbild das angezeigt werden kann.
- Die verwendete Auflösung muss 640px x 480px @ 60 Hz betragen.



Abbildung 2: Mögliches Testbild welches alle Farbkombinationen anzeigt

2.2 Schaltung

Die **FPGA** Entwicklungskarte bildet das Kernstück des Systems. Dort wird ein **Knöpfe und LEDs** angeschlossen sowie ein **PMod-DVI Modul**. Am **HDMI** Ausgang des Modules wird der Bildschirm angeschlossen. Das Gesamtsystem ist schematisch in der Abbildung 3 ersichtlich.

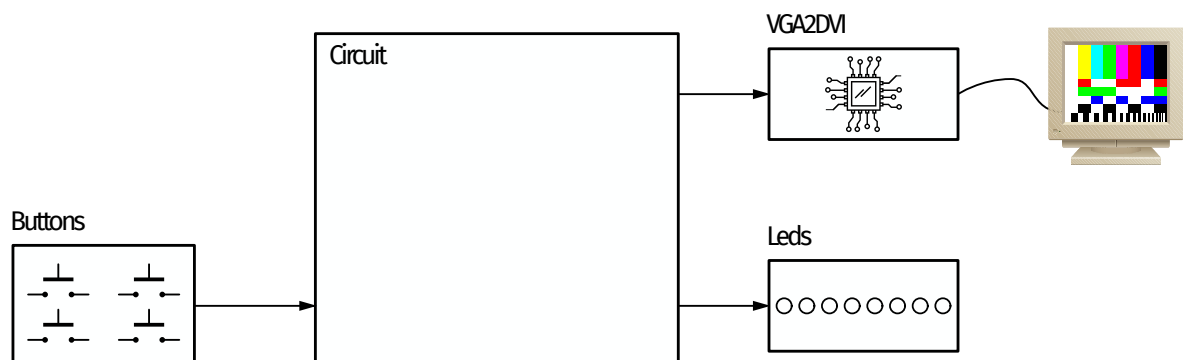
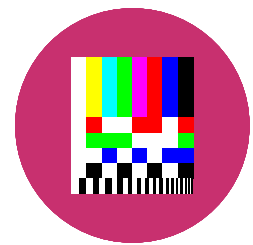


Abbildung 3: Display Schaltung



Die komplette Schaltung funktioniert wie folgt:

- Vier Tasten werden zur Steuerung des Systems verwendet: *start*, *stop* sowie zwei frei verfügbare Knöpfe *button₃* und *button₄*. Diese können für optionale Funktionen benutzt werden.
- Mit dem drücken des *start* Knopfes werden die Bildinformationen, die aus der Position der Pixel berechnet wird (z.B. farbige Balken), über die **VGA** Schnittstelle zum **PMod**-Module kontinuierlich übertragen.
- Durch das Drücken des *stop* Knopfes wird die Übertragung gestoppt.
- Das **PMod**-Modul besitzt Konfigurationsspins sowie die zuvor erwähnte **VGA**-Video-Schnittstelle, um die Bilddaten zu empfangen. Hierzu werden folgende Signale verwendet: *vga_dataEnable*, *vga_pixelClock*, *vga_hsync*, *vga_vsync* sowie *vga_rgb[2 : 0]*.
- Das Modul konvertiert **VGA**-Video Signale zu **Transition Minimized Differential Signaling (TMDS)** Signalen und sendet diese über die **HDMI** Schnittstelle zum Monitor.
- 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 (display-toplevel-empty.pdf) zeigt alle Signale, die an die **Field Programmable Gates Array (FPGA)**-Platine angeschlossen sind, siehe Abbildung 4.

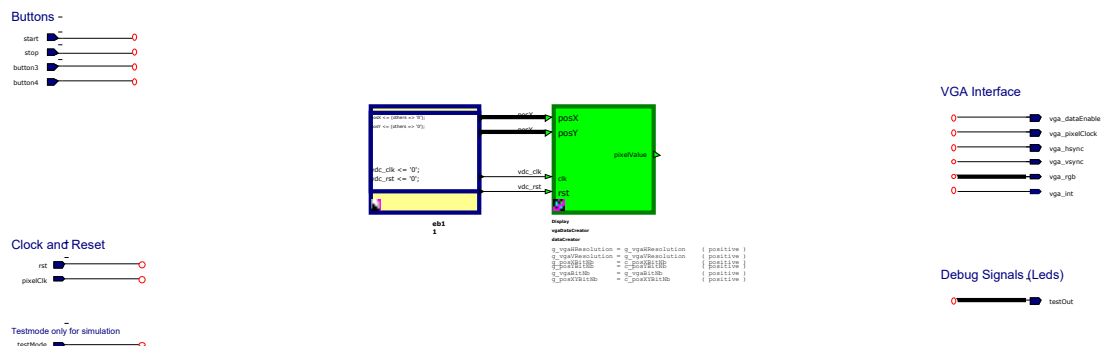
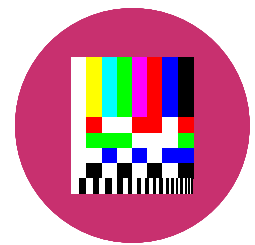


Abbildung 4: Leere Toplevel Schaltung

Das Signal **testOut** ermöglicht es, die auf dem **Knöpfe und LEDs** vorhandenen LEDs ein- und auszuschalten.



Der bereits vorhandene Block **vgaDataCreator** ermöglicht es, eine Pixelfarbe entsprechend der vorgegebenen X- und Y-Positionen (von 0 bis 639 für X, 0 bis 479 für Y) zu liefern, um ein Bild anzuzeigen. Er ermöglicht die grobe Erkennung eines Synchronisationsfehlers (d. h. das Bild sieht verzerrt, wellig, abgeschnitten ... aus). **Dieser Block wird zu Testzwecken zur Verfügung gestellt und ersetzt nicht die eigentliche Anforderung, ein benutzerdefiniertes Bild anzuzeigen.**



2.3 VGA Timing (Beispiel)

In den Abbildungen 6 und 5 wird das VGA Timing für die Auflösung 640px x 480px @ 60Hz dargestellt.

Das Bild wird Linie für Linie aufgebaut beginnend mit der oberen linken Ecke. Das Signal *vga_rgb* enthält die Daten eines Pixels, bei jedem Takt des *vga_pixelClock* kann ein neues Pixel übertragen werden. *vga_int* kann bei RGB-Signalen aktiviert werden, wodurch hellere Farben angezeigt werden können. *vga_dataEnable* zeigt an ob das Datensignal aktiv ist und diese gelesen werden dürfen. Die restlichen zwei Signale *vga_hsync* sowie *vga_vsync* geben an, ob eine neue Linie (*hsync*) oder eine neue Seite (*vsync*) beginnt.



Die zeitlichen Bedingungen für die Signale *vga_hsync* sowie *vga_vsync* sind genau zu verstehen und zu beachten. Mehr Infos können in der [Video Electronics Standards Association \(VESA\)](#) Spezifikation [13] sowie auf der TinyVGA Webseite [12] sowie gefunden werden.

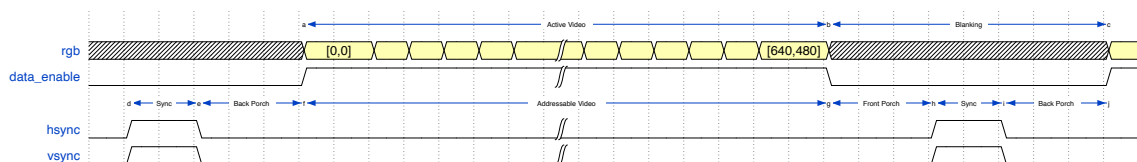
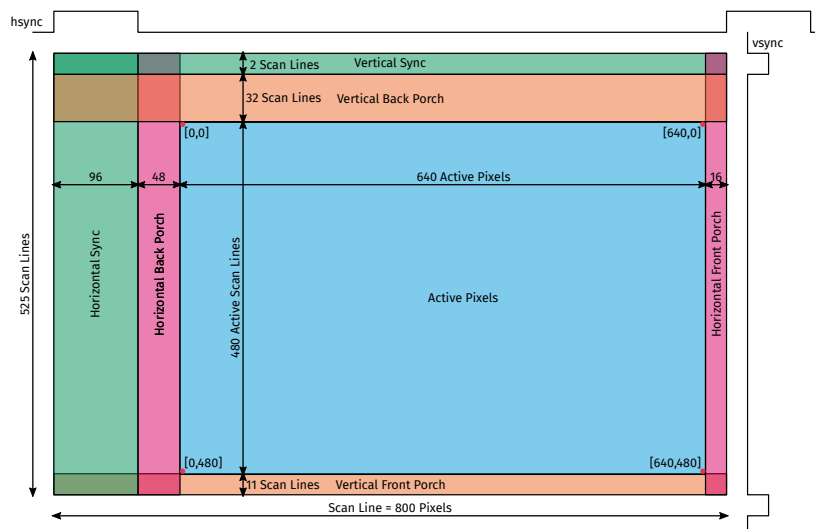


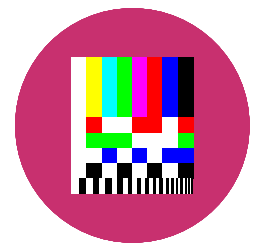
Abbildung 5: Zeitliche Abfolge der VGA Signale (*hsync* und *vsync* vermischt)



Außerhalb des Bereichs **Active Pixels** muss der RGB-Wert zu jeder Zeit 0 sein.

Bei einem Bild von 640px x 480px werden rein theoretisch 800px x 525px übertragen, wobei die zusätzlichen Pixel für das vertikale und horizontale "front porch" und "back porch" benötigt werden. Diese sind dazu da, einem alten CRT Monitor genug Zeit einzuräumen damit der Elektronenstrahl seine Position zwischen den Linien und Seiten neu positionieren kann.



Abbildung 6: **VGA** Video Dekodierung

Die Timings der Signale *vga_hsync* sowie *vga_vsync* sind genau vorgegeben.
Ein Beispiel für die Auflösung 1920px x 1440px @ 60Hz ist in der Listing 1 ersichtlich:

Name 1920x1440 @ 60Hz
Aspect Ratio 4:3

Pixel Clock 234 MHz
Pixel Time 4.27 ns
Horizontal freq. 90 kHz
Line Time 11.11 µs
Vertical freq. 60 Hz
Frame Time 16.66 ms

Horizontal Timings

Visible Area 1920
Front Porch 128
Sync Width 208
Back Porch 344
Total (blanks) 672
Total (all) 2600
Sync Polarity neg

Vertical Timings

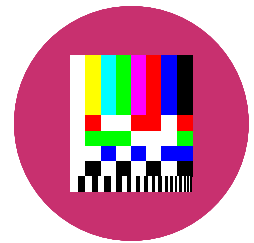
Visible Area 1440
Front Porch 1
Sync Width 3
Back Porch 56
Total (blanks) 60
Total (all) 1500
Sync Polarity pos

Active Pixels

Visible Area 2,764,800

Listing 1: **VGA** Konfiguration für 1920px x 1440px @ 60Hz

Die obigen Abbildungen sind Beispiele, es liegt an den Studenten diese zu verstehen und an die eigenen Bedürfnisse anzupassen.



2.4 HDL-Designer Projekt

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

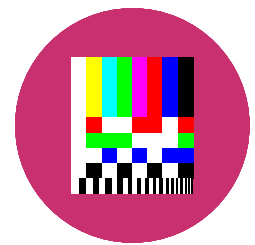
```
did_display
+--Board/          # Project and files for programming the FPGA
|   +--concat/     # Complete VHDL file including PIN-UCF file
|   +--ise/        # Xilinx ISE project
+--Display/        # Library for the components of the student solution
+--Display_test/   # Library for the simulation 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
+--Tools/         # Specific tools, like a picture to BRAM translator
```



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.

- Ein Entwicklungsboard **FPGA**, siehe Abbildung 8.
- Eine Steuerkarte mit 4 Tasten und 8 **LEDs**, siehe Abbildung 9.
- Ein **PMod-DVI** Modul um den Bildschirm über eine **HDMI** Schnittstelle an das System anzuschliessen, siehe Abbildung 10.

3.1 **FPGA-Platine**

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [9]. Diese beherbergt eine **Xilinx Spartan xc3s500e FPGA** [**Spartan3FPGAFamily**] [14] 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 $f_{clk} = 66MHz$ [3].

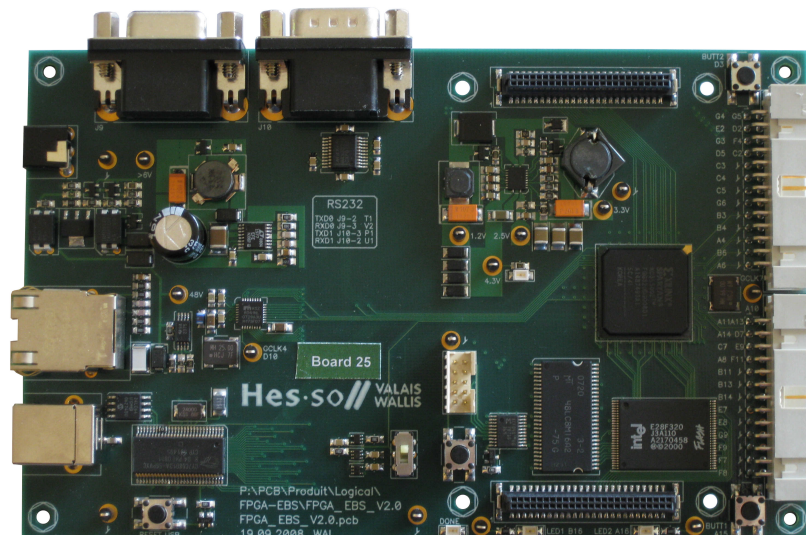


Abbildung 7: **FPGA** Platine [9]

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.

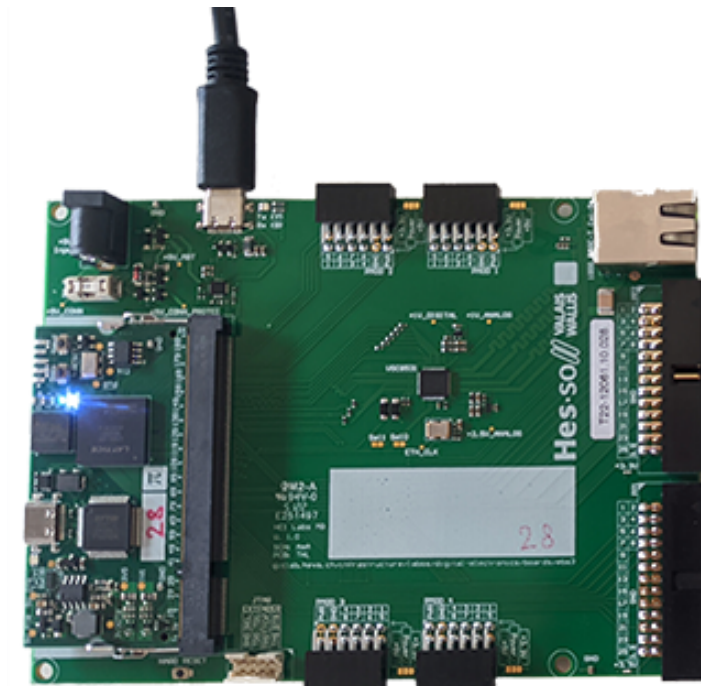
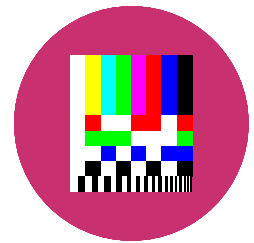


Abbildung 8: EBS3 **FPGA** Platine [amandaxelSchematicFPGAEB3V12023]



Sofern Sie nicht den LCD-Block simulieren wollen, ist keine Manipulation erforderlich, um das EBS2- oder EBS3-System zu simulieren. Ansonsten öffnen Sie die Testbench **display_tb** und doppelklicken Sie auf die **Pre-User**-Deklarationen (oben links auf der Seite), um die Variable **clockFrequency** auf den gewünschten Wert für clock zu ändern.

3.2 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].

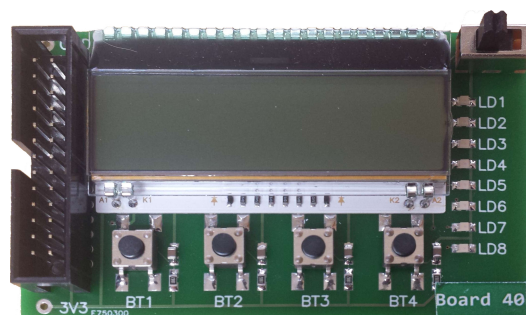


Abbildung 9: Knöpfe-**LED**-**LCD** Platine [10]

3.3 **PMod-DVI** Modul

Das **PMod** modul **VGA** nach **DVI** Konvertiert die **VGA** Signale to **TMDS** Signalen. Diese erlaubt es einen **HDMI** Monitor an das System anzuschliessen.

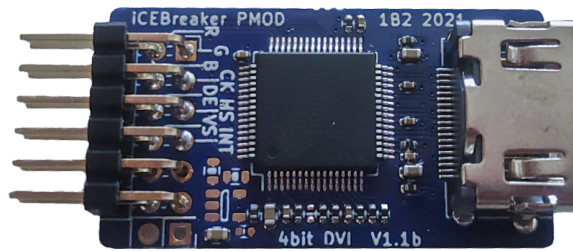
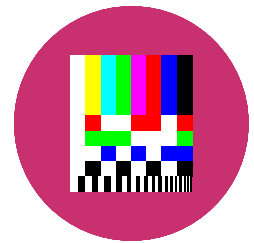


Abbildung 10: PMod-DVI Modul

Das Blockschaltbild des Chips Texas Instrument TFP410 [7] kann in der Abbildung 11 entnommen werden.



Studiert das Datasheet [7] sowie das Schema des PMod-Moduls genau [8].

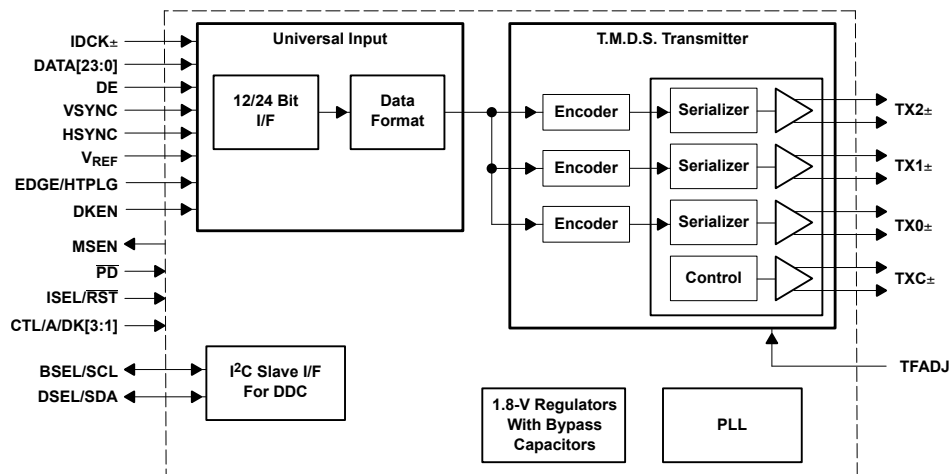
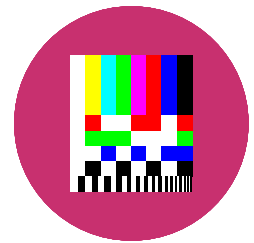


Abbildung 11: Blockschaltbild des PMod-Chips TI TFP410 [7]



4 Bewertung

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

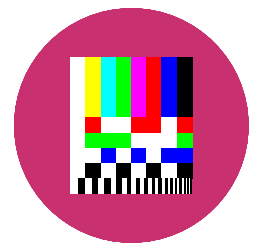
Die Schlussnote beinhaltet den Bericht, den Code sowie eine Präsentation eurerseits des Systems.

| 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?" [2]



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 🐛.

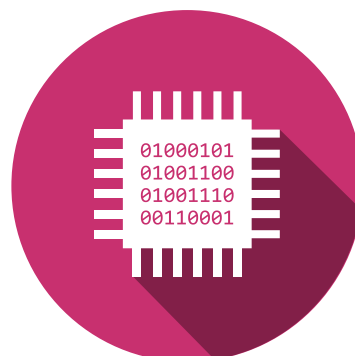
5.1 Tips

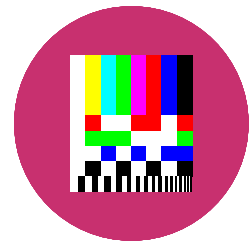
Anbei noch einige zusätzlichen Tips um Probleme und Zeitverlust zu vermeiden:

- Teilt das Problem in verschiedene Blöcke auf, benutzt hierzu das leere Toplevel Dokument (**display-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]
- Es wird empfohlen das System in zwei Schritten zu realisieren.
 - Beginnen Sie damit, ein einfaches statisches Bild anzuzeigen.
 - Integrieren das komplexe Testbild und zusätzliche Funktionen.



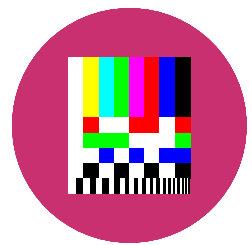
Vergesst nicht Spass zu haben 😊.





Literatur

- [1] Agilent Technologies. *Datasheet Agilent AEDB-9140 Series Three Channel Optical Incremental Encoder Modules with Codewheel, 100 CPR to 500 CPR*. 2005.
- [2] Christophe Bianchi, François Corthay und Silvan Zahno. *Wie Verfasst Man Einen Projektbericht?* 2021.
- [3] CTS. *Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator*. 2006.
- [4] Digilent. *Digilent Pmod Interface Specification*. 2011. URL: https://digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf.
- [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] Texas Instrument. *Datasheet Digital Transmitter Texas Instrumment TFP410*. 2014.
- [8] Esden Piotr. *Schematic: iCEBreaker PMOD 4bit DVI*. 2018.
- [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] TinyVGA. *VGA Signal Timing*. URL: <http://www.tinyvga.com/vga-timing> (besucht am 07.07.2022).
- [13] VESA. *VESA and Industry Standards and Guidelines for Computer Display Monitor Timing (DMT)*. 2013. URL: <https://glenwing.github.io/docs/VESA-DMT-1.13.pdf>.
- [14] Xilinx. *Datasheet Spartan-3E FPGA Family*. 2008.



Akronyme

bpp bit per pixel. [3](#)

DVI Digital Visual Interface. [3](#), [8](#), [10](#)

FPGA Field Programmable Gates Array. [3](#), [4](#), [8](#), [9](#)

HDMI High Definition Multimedia Interface. [3](#), [4](#), [8](#), [10](#)

LCD Liquid Crystal Display. [2](#), [9](#)

LED Light Emitting Diodes. [4](#), [8](#), [9](#)

PMod Peripheral Module. [4](#), [8](#), [10](#)

TMDS Transition Minimized Differential Signaling. [4](#), [10](#)

UART Universal Asynchronous Receiver Transmitter. [8](#)

USB Universal Serial Bus. [8](#)

VESA Video Electronics Standards Association. [5](#)

VGA Video Graphics Array. [2](#), [4–6](#), [10](#)