

Display - Contrôle d'écran

Projet Systèmes numérique



Orientation : [Informatique et systèmes de communication \(ISC\)](#)
Cours : Systèmes numérique (DiD)
Auteur : [Axel Amand](#), [Silvan Zahno](#)
Date : 2 juin 2023
Version : v2.0

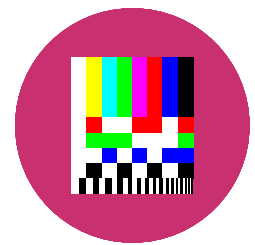
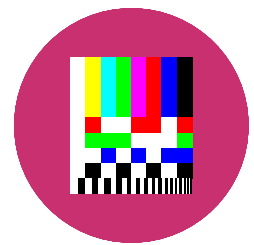


Table des matières

| | | |
|----------|--------------------------------|-----------|
| 1 | Introduction | 2 |
| 2 | Spécifications | 3 |
| 2.1 | Fonctions | 3 |
| 2.2 | Circuit | 3 |
| 2.3 | VGA Timing (exemple) | 5 |
| 2.4 | Projet HDL-Designer | 7 |
| 3 | Composants | 8 |
| 3.1 | Carte FPGA | 8 |
| 3.2 | Boutons et LEDs | 9 |
| 3.3 | Module PMod-DVI | 9 |
| 4 | Evaluation | 11 |
| 5 | Premières étapes | 12 |
| 5.1 | Tips | 12 |
| | Références | 13 |
| | Acronymes | 14 |



1 Introduction

L'objectif du projet est d'appliquer directement les connaissances acquises à la fin du semestre à l'aide d'un exemple pratique. Il s'agit de piloter un écran via une interface **Video Graphics Array (VGA)** afin d'afficher une image prédéfinie. Ce système d'affichage est représenté dans l'illustration 1.

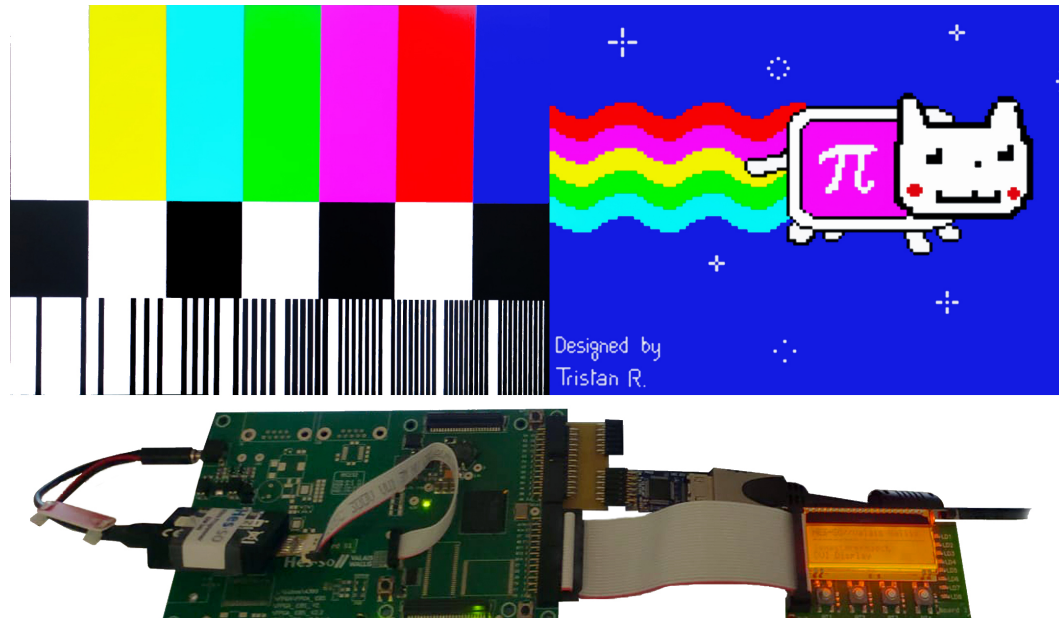
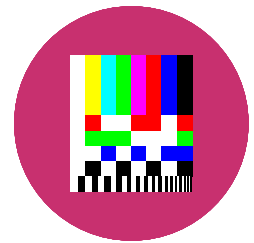


FIGURE 1 – Équipement du display

Le but est de réaliser les **Spécifications** minimales définies au (chapitre 2). Les étudiants peuvent, en option, ajouter des fonctions supplémentaires. Il n'y a pas de limites aux idées, par exemple l'écran **LCD** peut être utilisé pour afficher certaines informations, l'image de test peut être modifiée et étendue ...



Les fonctions supplémentaires permettent d'obtenir des points supplémentaires.



2 Spécifications

2.1 Fonctions

Les fonctions de base sont définies comme suit :

- Si la touche *start* est appuyée, une image test s'affiche sur le moniteur.
- Si la touche *stop* est appuyée, l'image test est supprimée et le moniteur devient noir.
- L'image test est composée de toutes les combinaisons de couleurs possibles qui peuvent être combinées avec le 3bit per pixel (bpp) Digital Visual Interface (DVI) module. L'illustration 2 montre une image test possible qui peut être affichée.
- La résolution utilisée est de 640px x 480px @ 60 Hz.



FIGURE 2 – Mire de test montrant toutes les combinaisons de couleurs

2.2 Circuit

La carte de développement FPGA constitue le cœur du système. On y connecte une carte Boutons et LEDs ainsi qu'un Module PMod-DVI. L'écran est connecté à la sortie du module par HDMI. L'ensemble du système est schématisé dans l'illustration 3.

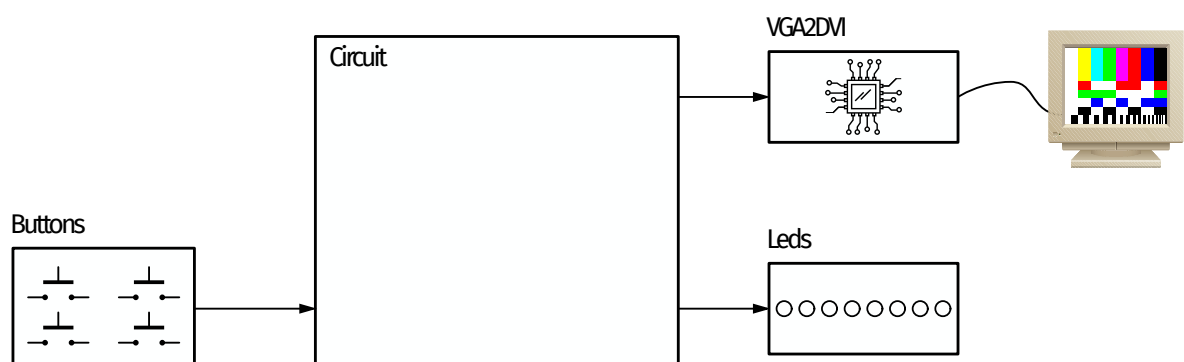
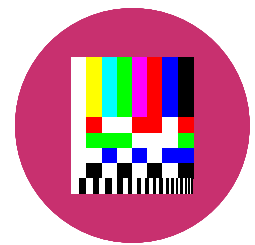


FIGURE 3 – Circuit d'affichage



Le circuit complet fonctionne comme suit :

- Quatre boutons sont utilisés pour contrôler le système : *start*, *stop* ainsi que deux boutons librement disponibles *button₃* et *button₄*. Ceux-ci peuvent être utilisés pour des fonctions optionnelles.
- En appuyant sur le bouton *start*, une image, calculée selon la position des pixels (ex. barres de couleur), est transmise en continu au module **PMod** via l'interface **VGA**.
- En appuyant sur le bouton *stop*, la transmission s'arrête.
- Le module **PMod** possède des broches de configuration ainsi que l'interface vidéo **VGA** mentionnée précédemment pour recevoir les données d'image. Les signaux suivants sont utilisés à cet effet : *vga_dataEnable*, *vga_pixelClock*, *vga_hsync*, *vga_vsync* ainsi que *vga_rgb[2 : 0]*.
- Le module convertit les signaux vidéo **VGA** en **Transition Minimized Differential Signaling (TMDS)** et les envoie au moniteur via l'interface **HDMI**.
- Les broches *testOut* peuvent être utilisées pour fournir des informations supplémentaires sur le système, par exemple pour le débogage ou pour contrôler les **LEDs**.

Le circuit TopLevel vide (display-toplevel-empty.pdf) montre tous les signaux connectés à la platine **Field Programmable Gates Array (FPGA)**, voir illustration 4.

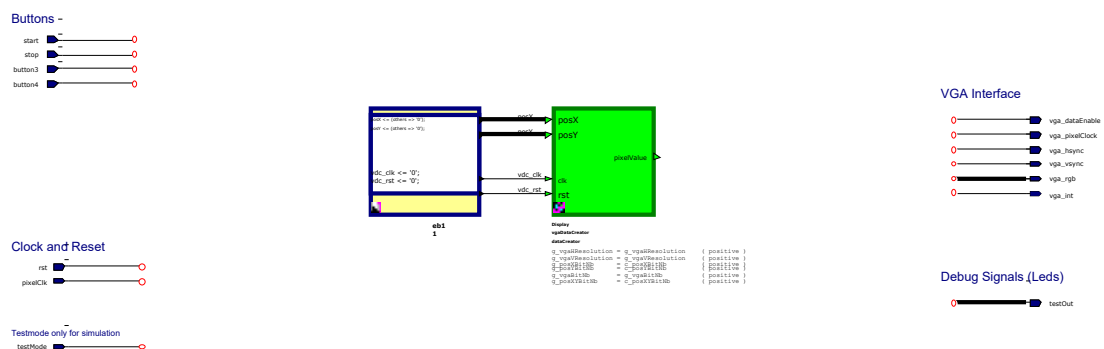


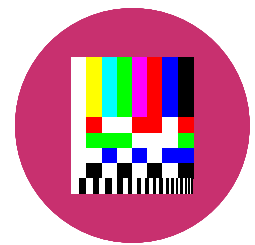
FIGURE 4 – Circuit Toplevel vide



Le signal **testOut** permet d'allumer et éteindre les leds présentes sur la plaque **Boutons et LEDs**.



Le bloc **vgaDataCreator** déjà présent permet de fournir une couleur de pixel selon les positions X et Y données (de 0 à 639 pour X, 0 à 479 pour Y) afin d'afficher une image. Il permet de détecter grossièrement une erreur de synchronisation (c.-à-d. l'image semble déformée, ondulée, coupée ...). **Ce bloc est fourni à des fins de test et ne remplace pas la demande première qui est d'afficher une image personnalisée.**



2.3 VGA Timing (exemple)

Les figures 6 et 5 montrent le VGA timing pour la résolution 640px x 480px @ 60Hz.

L'image est construite ligne par ligne, en commençant par le coin supérieur gauche. Les signaux *vga_hsync* et *vga_vsync* indiquent si une nouvelle ligne (*hsync*) ou une nouvelle page (*vsync*) commence. *vga_dataEnable* indique si le signal de données est actif (c.-à-d. dans la zone *ActivePixels*). Le signal *vga_rgb* contient les données d'un pixel, transmis à chaque cycle de *vga_pixelClock* lorsque nécessaire. *vga_int* peut être activé avec les signaux RGB, permettant d'afficher des couleurs plus claires.



Les conditions temporelles pour les signaux *vga_hsync* et *vga_vsync* doivent être parfaitement comprises et respectées. Plus d'informations peuvent être trouvées dans la spécification **Video Electronics Standards Association (VESA)** [13] ainsi que sur le site web de TinyVGA [12]

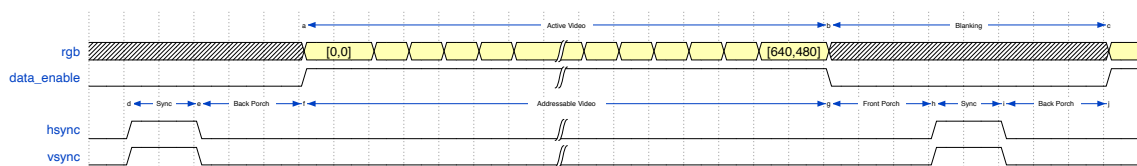
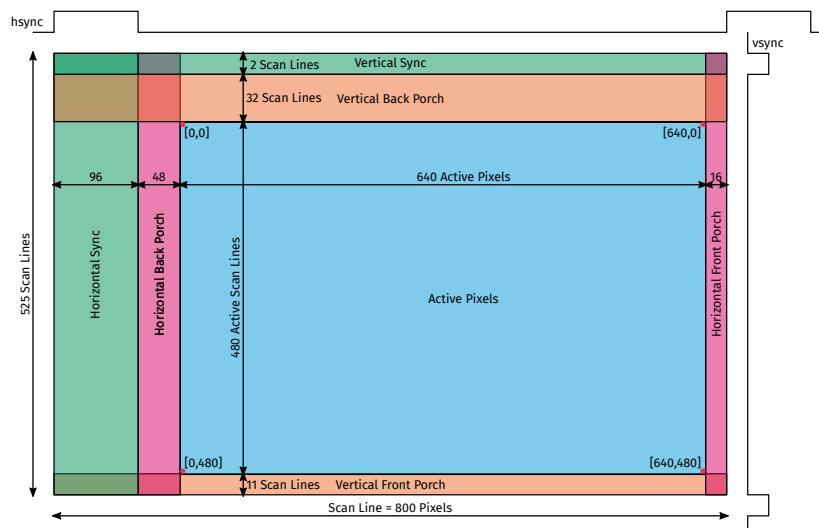


FIGURE 5 – Séquence temporelle des signaux VGA (*hsync* et *vsync* mélangés)



Hors de la zone **Active Pixels**, la valeur du RGB doit en tout temps être 0.

Pour une image de 640px x 480px, 800px x 525px sont théoriquement transmis, les pixels supplémentaires étant nécessaires pour le "front porch" et le "back porch" verticaux et horizontaux. Ceux-ci sont destinés à donner à un vieux moniteur CRT suffisamment de temps pour que le faisceau d'électrons puisse se repositionner entre les lignes et les pages.



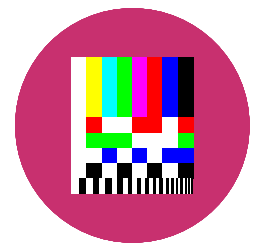


FIGURE 6 – VGA décodage de l'affichage

Les timings des signaux *vga_hsync* et *vga_vsync* sont prédéfinis avec précision.
Un exemple pour la résolution 1920px x 1440px @ 60Hz est donné dans le listing 1 :

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

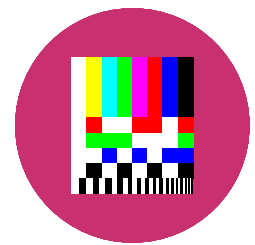
Visible Area 2,764,800

Listing 1 – VGA configuration pour 1920px x 1440px @ 60Hz



Les figures ci-dessus sont des exemples. L'étudiant se doit de les comprendre et les adapter pour ses propres besoins.

.



2.4 Projet HDL-Designer

Un projet HDL-Designer prédéfini peut être téléchargé ou cloné dans [Cyberlearn](#). La structure de fichier du projet se présente comme suit :

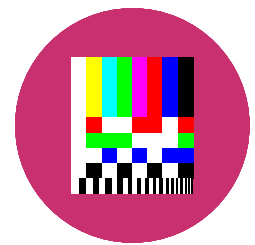
```
did_display
+--Board/          # Project and files for programming the FPGA
|   +--concat/     # Complete VHDL file including PIN-UCF file
|   +--ise/        # Xilinx ISE project
+--Display/        # Library for the components of the student solution
+--Display_test/   # Library for the simulation 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
```



Le chemin d'accès au dossier du projet ne doit pas contenir d'espaces



Le dossier de projet *doc/* contient de nombreuses informations précieuses : fiches techniques, évaluation de projet et documents d'aide pour HDL-Designer, pour n'en citer que quelques-uns



3 Composants

Le système se compose de 3 platines différentes, visibles dans la figure 1.

- Une carte de développement **FPGA**, voir figure 8.
- Une carte de contrôle à 4 boutons et 8 **LEDs**, voir figure 9.
- Un module **PMod** vers **DVI** pour l'affichage de l'image sur un écran par HDMI, voir figure 10.

3.1 Carte **FPGA**

La carte principale est la carte de développement de laboratoire FPGA-EBS 2 de l'école [9]. Elle héberge une puce **Xilinx Spartan xc3s500e FPGA** [**Spartan3FPGAFamily**] [14] et dispose de nombreuses interfaces différentes (**Universal Asynchronous Receiver Transmitter (UART)**, **Universal Serial Bus (USB)**, Ethernet, etc.). L'oscillateur utilisé produit un signal d'horloge (*clock*) avec une fréquence de $f_{clk} = 66\text{MHz}$ [3].

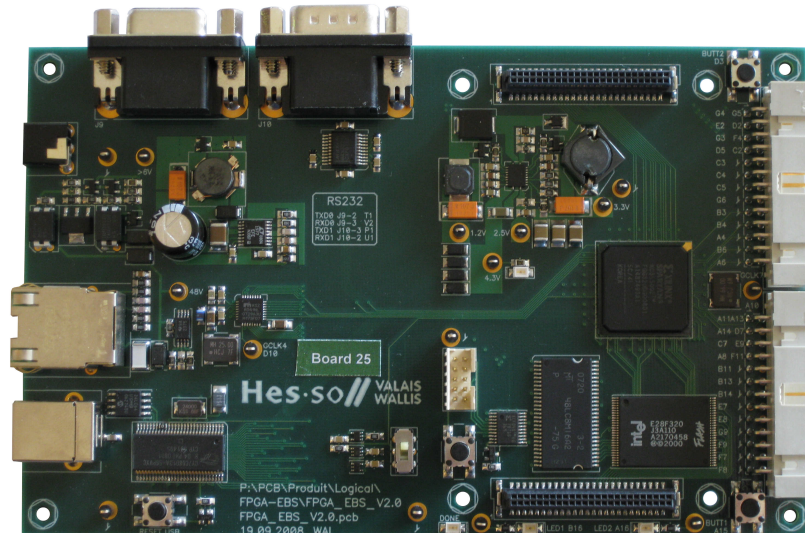


FIGURE 7 – Carte électronique **FPGA** [9]

Sur la carte EBS3, l'oscillateur utilisé produit un signal d'horloge (*clock*) avec une fréquence de $f_{clk} = 100\text{MHz}$, réduit par PLL à $f_{clk} = 60\text{MHz}$.

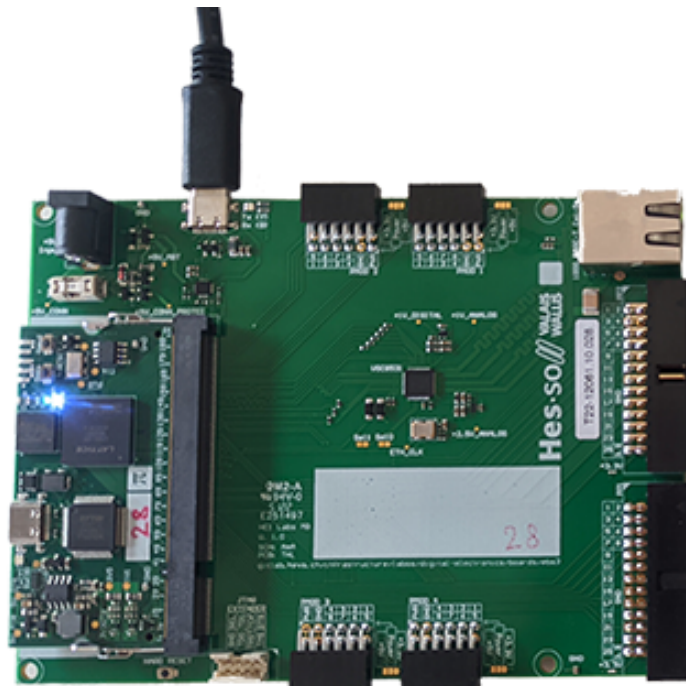
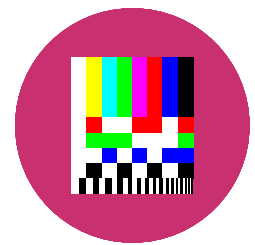


FIGURE 8 – Carte électronique **FPGA** EBS3 [amandaxelSchematicFPGAEB3V12023]



A moins de vouloir simuler le bloc LCD, aucune manipulation n'est nécessaire pour simuler le système EBS2 ou EBS3. Sinon, ouvrir le testbench **display_tb** et double-cliquez sur les déclarations **Pre-User** (en haut à gauche de la page) pour modifier la variable **clockFrequency** selon la valeur de clock souhaitée.

3.2 Boutons et **LEDs**

La platine boutons et les **LEDs** [10] est connectée à la platine **FPGA**. Elle possède 4 boutons et 8 **LEDs** qui peuvent être utilisés dans le design, ainsi qu'un affichage **LCD** [11] [5].

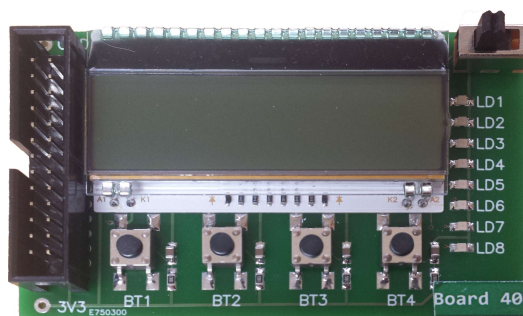


FIGURE 9 – Carte électronique boutons-**LED-LCD** [10]

3.3 Module **PMod-DVI**

Le module **PMod VGA** vers **DVI** convertit les signaux **VGA** en signaux **TMDS**. Cela permet de connecter un moniteur **HDMI** au système.

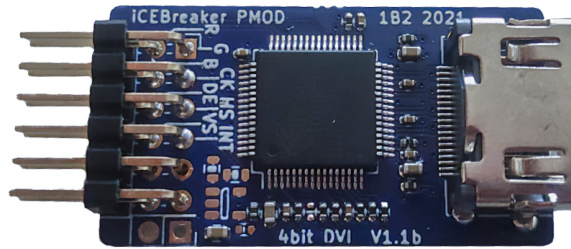
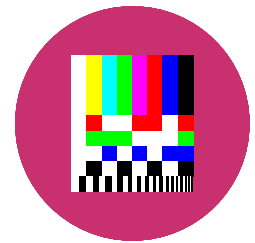


FIGURE 10 – PMod-DVI module

Le schéma fonctionnel de la puce Texas Instrument TFP410 [7] peut être consulté dans le diagramme 11.



Etudiez attentivement le datasheet [7] ainsi que le schéma du module PMod [8].

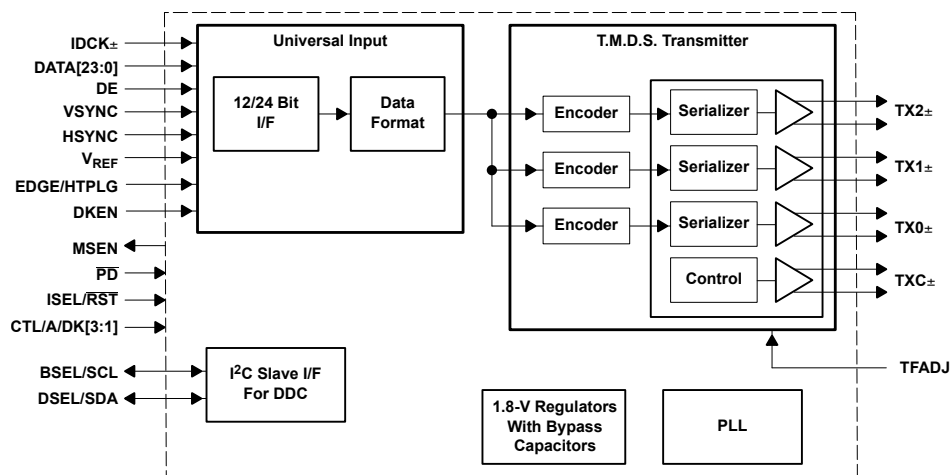
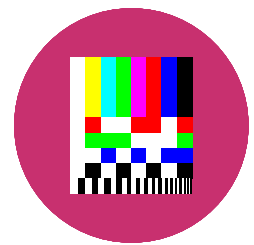


FIGURE 11 – Schéma fonctionnel de la puce du PMod TI TFP410 [7]



4 Evaluation

Dans le dossier *doc/*, le fichier **evaluation-bewertung-display.pdf** montre le schéma d'évaluation détaillé, tableau 1.

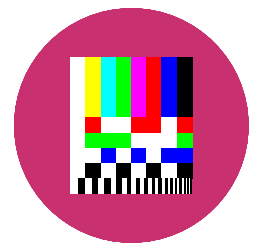
La note finale contient le rapport, le code ainsi qu'une présentation du système.

| Aspects évalués | points |
|----------------------------------|------------|
| Rapport | 55 |
| Introduction | 3 |
| Spécification | 5 |
| Projet | 20 |
| Vérification et validation | 10 |
| Intégration | 9 |
| Conclusion | 3 |
| Aspects formels du rapport | 5 |
| Fonctionnalité du circuit | 30 |
| Qualité de la solution | 10 |
| Présentation | 10 |
| Total | 105 |

TABLE 1 – Grille d'évaluation



La grille d'évaluation donne des indications sur la structure du rapport. Pour un bon rapport, consultez le document "Comment rédiger un rapport de projet" [2]



5 Premières étapes

Pour bien démarrer le projet :

- Lisez attentivement les spécifications et les informations ci-dessus.
- Examinez le matériel et testez le programme déjà présent sur la board [FPGA](#).
- Parcourez les documents dans le dossier *doc/* de votre projet.
- Développez un schéma fonctionnel détaillé. Vous devez pouvoir expliquer les signaux et leurs fonctions.
- Implémentez et simulez les différents blocs.
- Testez la solution sur circuit et trouvez les éventuelles erreurs 🐛.

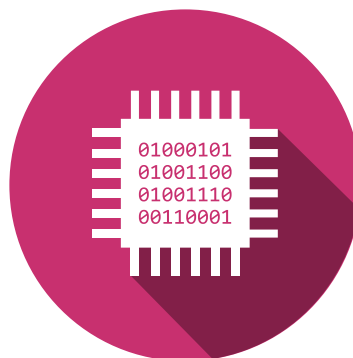
5.1 Tips

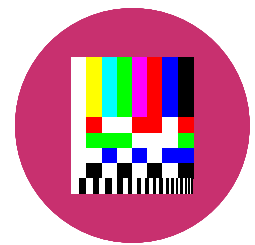
Voici quelques conseils supplémentaires pour éviter les problèmes et les pertes de temps :

- Divisez le problème en différents blocs. Utilisez pour cela le document Toplevel vide (**display-toplevel-empty.pdf**). Il est recommandé d'avoir un mélange équilibré entre le nombre de composants et la taille/complexité des composants.
- Analysez les différents signaux d'entrée et de sortie. Aidez-vous particulièrement des fiches techniques fournies.
- Respectez le chapitre DiD "Méthodologie de conception de circuits numériques (MET)" lors de la création du système. [6].
- Il est recommandé de réaliser le système de façon incrémentale, par exemple :
 - Commencer par afficher une image statique simple
 - Intégrer une mire de test plus complexe
 - Ajouter des fonctions supplémentaires



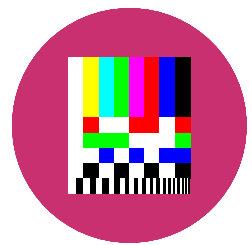
N'oubliez pas de vous amuser 😊.





Références

- [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 et SILVAN ZAHNO. *Comment Rédiger Un Rapport de Projet ?* 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 et CHRISTOPHE BIANCHI. *Méthodologie de Conception de Cicuits Numériques*. 2021.
- [7] Texas INSTRUMENT. *Datasheet Digital Transmitter Texas Insturment 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> (visité le 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.



Acronymes

bpp bit per pixel. [3](#)

DVI Digital Visual Interface. [1](#), [3](#), [8–10](#)

FPGA Field Programmable Gates Array. [1](#), [3](#), [4](#), [8](#), [9](#), [12](#)

HDMI High Definition Multimedia Interface. [3](#), [4](#), [9](#)

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

LED Light Emitting Diodes. [1](#), [3](#), [4](#), [8](#), [9](#)

PMod Peripheral Module. [1](#), [3](#), [4](#), [8–10](#)

TMDS Transition Minimized Differential Signaling. [4](#), [9](#)

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

USB Universal Serial Bus. [8](#)

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

VGA Video Graphics Array. [1](#), [2](#), [4–6](#), [9](#)