

V H D L

---

# RAPPORT TECHNIQUE

---

Bellini, Chevrier,  
Daloiso, Iniziato,  
Lopez, Zaieter

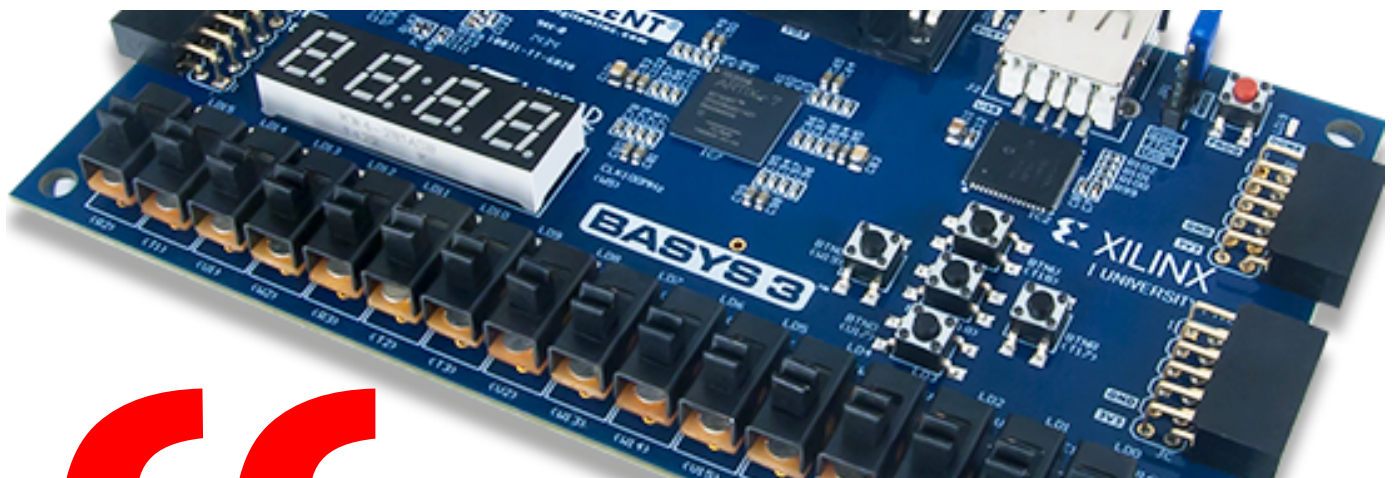
# SOMMAIRE

Introduction	3
Sujet 2	4
Sujet 3	5
Sujets 4 & 5	6
Sujet 6	7
Sujets 7 & 8	8
Sujet 9	9
Sujet 10	10
Conclusion	11

Le présent rapport décrit la réalisation de plusieurs projets VHDL effectués dans le cadre du cours de VHDL, qui constitue la dernière étape pédagogique avant la validation du module. Ces projets ont été développés sur Vivado sur la BASYS3, permettant la mise en œuvre pratique des concepts théoriques abordés durant l'année.

La **répartition** est la suivante :

- Loan et Lana : Sujets 2 (Chenillard à LEDs), 7 (Image fixe monochrome VGA) et 8 (Image fixe 3 couleurs VGA).
- William et Jassime : Sujets 4&5 (Affichage 4 chiffres identiques/différents) , 9 (Le pong), et 10 (Le Snake).
- Pierre-Louis et Gabriel : Sujets 6 (chronomètre) et 3 (Compteur/décompteur incrémental).



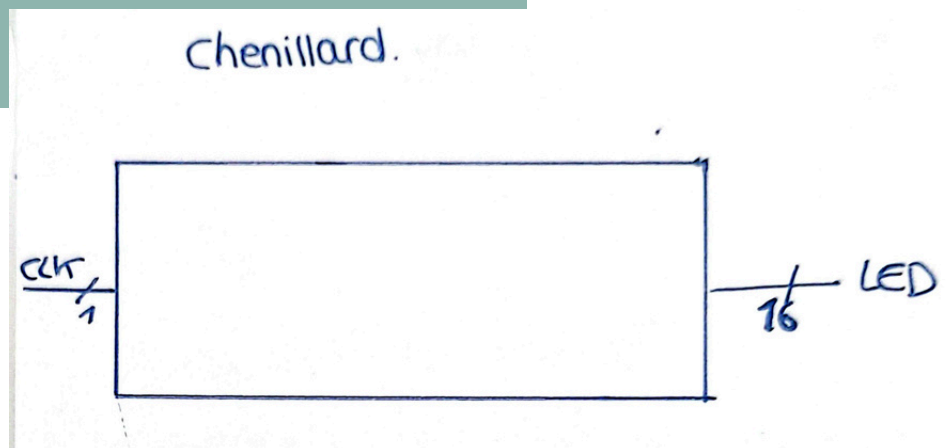
“

Cette organisation a favorisé les échanges techniques entre groupes. Nous avons tenu des réunions régulières pour partager les avancées, harmoniser nos méthodes et intégrer les modules. L'utilisation d'outils collaboratifs pour la gestion du code et des contraintes a assuré la cohésion du projet.

Le chenillard consiste à allumer successivement les LEDs de la BASYS3 dans un ordre cyclique, avec une temporisation entre chaque changement.

## ARCHITECTURE

Horloge et temporisation  
Registre à décalage circulaire  
Sortie vers LEDs



## CONSTRUCTION

Pour ce faire, un registre à décalage circulaire est utilisé afin de faire circuler un bit actif le long des différentes sorties. Ce bit contrôle directement l'activation des LEDs. Une temporisation a été mise en place pour rendre le défilement visible à l'œil nu. Le comportement est cyclique : lorsqu'on atteint la dernière LED, le cycle recommence. Une remise à zéro permet de réinitialiser le chenillard à son état initial.

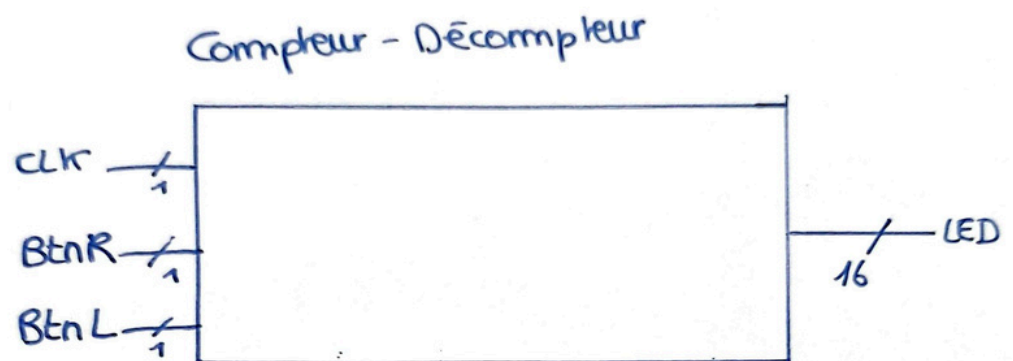
## DIFFICULTÉS

- Assurer une temporisation stable pour garantir une vitesse d'animation constante et visible.
- Gérer correctement le comportement du circuit lors d'un reset, notamment en s'assurant que le bit actif revient bien à sa position de départ sans erreurs de synchronisation.

Ce projet implémente un compteur simple contrôlé par deux boutons poussoirs. L'un incrémente le nombre de LEDs allumées, l'autre le décrémente.

## ARCHITECTURE

Entrées boutons gauche/droite  
Compteur binaire saturé  
Sortie LEDs



## CONSTRUCTION

Ce projet implémente un compteur contrôlé par deux boutons-poussoirs. Le premier bouton permet d'incrémenter la valeur affichée via les LEDs, tandis que le second permet de la décrémente. Le système repose sur la détection des fronts montants des signaux des boutons, afin d'éviter des répétitions indésirables. Une saturation est également mise en place : le compteur ne peut ni dépasser la valeur maximale, ni descendre en dessous de zéro. La synchronisation entre les entrées boutons (asynchrones) et l'horloge de la carte FPGA est essentielle pour garantir un fonctionnement fiable. La valeur du compteur est directement traduite sur l'état des LEDs.

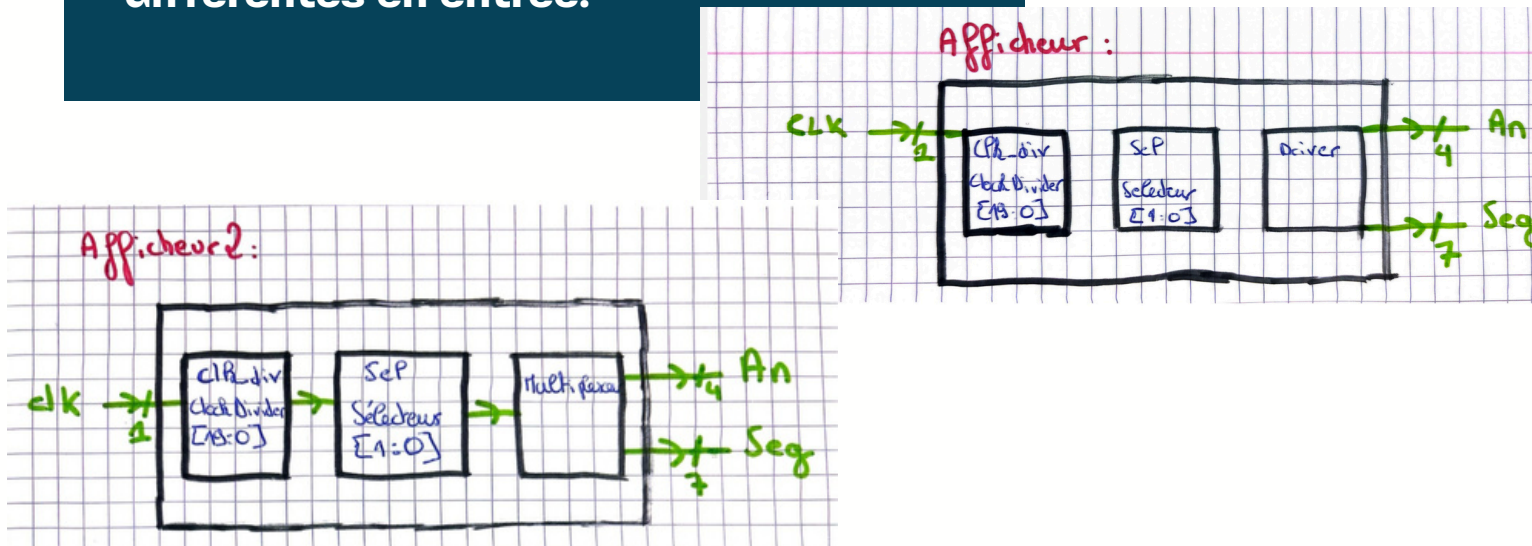
## DIFFICULTÉS

La difficulté principale a été de gérer correctement la remise à zéro lors des dépassements, c'est-à-dire d'éviter tout comportement indéfini quand les limites haute ou basse du compteur sont atteintes. La gestion des entrées asynchrones a également nécessité une attention particulière pour éviter les erreurs de déclenchement.

Ces deux projets visent à afficher des chiffres sur les quatre afficheurs 7 segments de la carte BASYS3. Le premier consiste à afficher un même chiffre sur les quatre afficheurs, tandis que le second étend cette fonctionnalité pour afficher quatre chiffres différents simultanément.

## ARCHITECTURE

Identique au sujet 4, mais avec 4 valeurs différentes en entrée.



## CONSTRUCTION

Chaque chiffre est stocké dans un registre indépendant, évitant tout mélange de valeurs entre les digits. Le système utilise un multiplexage dynamique pour rafraîchir chaque afficheur de manière cyclique à haute fréquence, de sorte que l'affichage semble stable à l'œil nu. Un décodeur unique est réutilisé pour tous les digits afin d'optimiser la logique.

## DIFFICULTÉS

- Il a fallu respecter scrupuleusement le timing de multiplexage, pour éviter tout scintillement ou affichage erroné.
- Il était crucial de garantir une cohérence de l'affichage final, notamment en assurant que les données de chaque registre soient bien affichées sur le digit correspondant.
- Il fallait aussi s'assurer qu'il n'y ait aucun mélange de valeurs entre les différents registres.
- Enfin, le rafraîchissement devait être fluide, ce qui impliquait un ajustement précis de la fréquence du multiplexage.





Ces projets consistent à afficher des images fixes sur un écran via le port VGA de la BASYS3. L'un une couleur unie, l'autre le drapeau français.

## ARCHITECTURE

**Contrôleur VGA**  
**Mémoire image**  
**Génération des signaux RGB**

Image Fixe Monochrome :

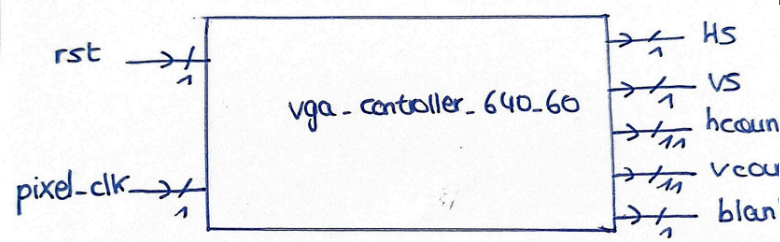
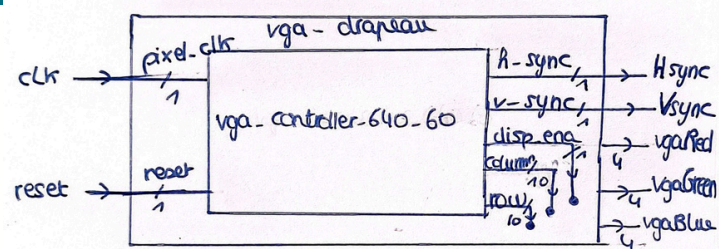


Image fixe 3 couleurs différentes : (drapeau)



## CONSTRUCTION

Le système repose sur un contrôleur VGA respectant les timings horizontaux et verticaux du standard VGA à 25 MHz. Une mémoire image est parcourue pour générer le contenu visuel pixel par pixel, en fonction des coordonnées fournies par le contrôleur.

Pour le projet monochrome, chaque pixel est soit allumé, soit éteint, selon un bit unique.

Pour le projet trois couleurs, l'image représente un drapeau français, et la mémoire utilise 2 bits par pixel pour encoder les couleurs. Un décodage RGB transforme ces deux bits en un signal couleur exploitable. Les signaux RGB sont générés dynamiquement et synchronisés avec la position actuelle des pixels.

## DIFFICULTÉS

- Le respect strict des timings VGA, indispensable pour assurer une image stable à l'écran sans décalage ni scintillement.
- La gestion correcte de la mémoire image selon la profondeur couleur.
- La synchronisation des signaux RGB avec les coordonnées VGA, en particulier pour le projet couleurs, où le décodage des 2 bits a requis une adaptation du contrôleur pour produire les couleurs appropriées.



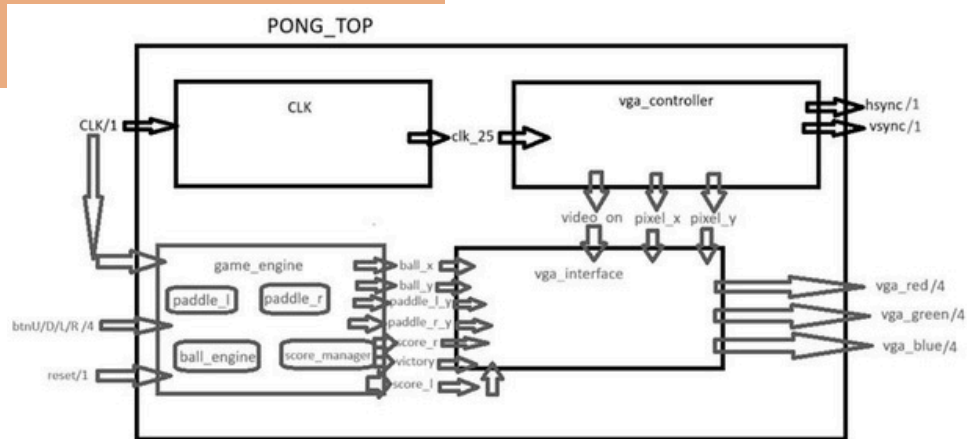
Le jeu Pong est un jeu interactif classique où deux joueurs contrôlent chacun une raquette verticale pour renvoyer une balle qui rebondit dans une aire de jeu. Le score est affiché, et la vitesse de la balle augmente avec le temps.

## ARCHITECTURE

Gestion des raquettes

Gestion de la balle

Gestion des collisions Affichage VGA



## CONSTRUCTION

L'affichage du score peut se faire sur les afficheurs 7 segments ou directement via VGA.

L'architecture est modulaire, avec des blocs distincts pour la gestion des raquettes, de la balle, des collisions et du score. Les positions de chaque élément sont calculées dynamiquement à chaque cycle d'horloge pour assurer un mouvement fluide.

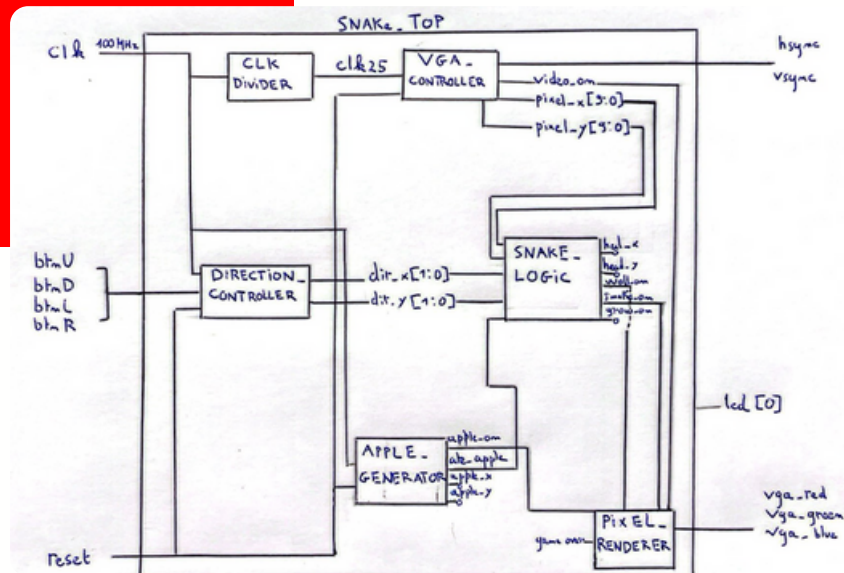
## DIFFICULTÉS

- La gestion précise des collisions, nécessitant des calculs fiables et une prise en compte du timing pour réagir au bon moment.
- La synchronisation entre les mouvements des raquettes, de la balle et l'affichage pour éviter tout effet de décalage.
- L'optimisation de la vitesse de la balle, qui devait s'adapter à la difficulté croissante tout en restant jouable.
- La gestion fiable des entrées utilisateurs, en particulier les boutons, pour garantir une réactivité immédiate et sans erreur.

Le jeu Snake consiste à contrôler un serpent qui grandit à chaque fois qu'il mange une « pastille », tout en évitant les murs et son propre corps. Le jeu est affiché sur le port VGA.

## ARCHITECTURE

**Gestion du serpent**  
**Gestion des déplacements**  
**Gestion des collisions**  
**Affichage VGA**  
**Gestion de la nourriture**



## CONSTRUCTION

Implémentation de structures dynamiques (tableaux ou listes).  
 Algorithme de déplacement par mise à jour des positions des segments.  
 Détection des collisions.  
 Gestion des temps pour rendre le déplacement fluide.

## DIFFICULTÉS

Programmation modulaire

- Code structuré en blocs : déplacement, collisions, affichage, croissance
- Détection de collisions
- Gestion des collisions avec les murs (fin de partie)
- Croissance du serpent
- Allongement du corps après chaque fruit mangé
-

# CONCLUSION

La réalisation de ces projets VHDL sur la carte BASYS3 a été une expérience très formatrice pour notre apprentissage de la conception matérielle. Chaque projet nous a aidés à renforcer nos connaissances, des bases du langage VHDL jusqu'à des applications plus avancées comme le traitement vidéo en temps réel et la programmation de jeux interactifs.

Nous avons rencontré des difficultés classiques, notamment la gestion précise des temporisations, la prise en compte fiable des entrées boutons, la synchronisation des différents modules, ainsi que le multiplexage des afficheurs 7 segments et la gestion des signaux VGA. Ces défis nous ont permis de développer des compétences importantes en débogage matériel, simulation et organisation modulaire du code.

Le travail en équipe a favorisé le partage des compétences et une bonne répartition des tâches. Cette organisation, associée à la diversité des projets, a permis une progression solide et complète de nos savoir-faire.

**Les projets menés couvrent un large spectre d'applications pratiques en VHDL, offrant une base solide pour des développements futurs plus avancés.**

”

V H D L

---

# FIN DU RAPPORT

---

**Bellini, Chevrier,  
Daloiso, Iniziato,  
Lopez, Zaieter**