

---

## RAPPORT DE PROJET

### *Projet d'électronique n°3 : FRUIT'ECE*

---

*Auteurs :*

Erwan	PLANET
Hugo	FAIVRE
Eloi	BRICQ
Thibaud	DANSETTE

*Enseignant :*

Mme. GULDNER  
Mr. SCHNEIDER

Le contexte du projet FRUIT'ECE est inspiré du jeu vidéo "Fruit Ninja", où l'utilisateur découpe des fruits pour accumuler des points. L'objectif de ce projet est d'adapter ce jeu sur une FPGA, en tirant parti des capacités matérielles pour gérer l'affichage, la détection des mouvements et l'interaction utilisateur. Ce projet se situe dans le cadre d'une initiation à la conception sur FPGA, mettant en œuvre des composants tels que des capteurs à ultrasons et un encodeur numérique pour interagir avec le jeu.

Les objectifs techniques incluent :

1. **Horloge du système** : Générer un signal d'horloge à 1 Hz à partir d'une entrée de 50 MHz.
2. **Affichage** : Gérer l'affichage du jeu et des menus sur un écran via VGA, avec des fruits apparaissant et se déplaçant de manière aléatoire.
3. **Interaction utilisateur** : Le joueur contrôle la hauteur et l'angle de coupe des fruits à l'aide d'un capteur à ultrasons et d'un encodeur numérique.
4. **Sonorisation** : Différents sons signalent la découpe des fruits ou des bombes, ainsi que la perte de vies.
5. **Tests et validation** : Des simulations et des tests sur oscilloscope valident la précision des signaux.

Ce projet s'inscrit dans un contexte d'apprentissage où les étudiants appliquent leurs connaissances en électronique et programmation pour développer une application ludique complexe sur FPGA, renforçant leurs compétences en conception numérique et en gestion de projet.

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au binôme et qu'il a été rédigé de manière autonome.

PARIS, le 21/10/2024

## Table des matières

<b>I. Objectif</b>	<b>3</b>
<b>II. Glossaire</b>	<b>3</b>
A. Termes	3
B. Acronymes	3
<b>III. L'équipe</b>	<b>3</b>
A. Présentation de l'équipe	3
B. Organisation de l'équipe	3
C. Diagramme de Gantt	3
<b>IV. Contexte et problématique</b>	<b>3</b>
A. Contexte	3
B. Problématique	3
C. Spécifications techniques	4
<b>V. Conception</b>	<b>4</b>
A. Architecture fonctionnelle	4
B. Architecture matérielle.	4
C. Architecture logicielle	4
<b>VI. Développement</b>	<b>4</b>
A. Module FS1 : Générer un signal d'horloge	4
B. Module FS2 : Afficher le menu de jeu	4
C. Module FS3 : Gérer et afficher le jeu	4
D. Module FS4 : Interagir avec le jeu	4
E. Module FS5 : Afficher le temps restant et les points	4
<b>VII. Tests et validation</b>	<b>4</b>
A. Module FS1 : Générer un signal d'horloge	4
B. Module FS2 : Afficher le menu de jeu	4
C. Module FS3 : Gérer et afficher le jeu	4
D. Module FS4 : Interagir avec le jeu	4
E. Module FS5 : Afficher le temps restant et les points	5
<b>VIII. Bilan</b>	<b>5</b>
A. État d'avancement	5
B. Pertinence de la solution technique	5
C. Bilan sur le travail d'équipe	5
<b>IX. Bibliographie</b>	<b>6</b>
<b>X. Annexes</b>	<b>7</b>

## I. Objectif

L'objectif de ce document est de présenter nos avancées sur le projet FRUIT'ECE.

Il comprend des explications techniques, accompagnées de schémas, de résultats, ainsi qu'un état d'avancement du projet et le bilan final.

## II. Glossaire

### A. Termes

Terme	Définition
QUARTUS	Logiciel permettant de développer, simuler et programmer des circuits logiques comme les FPGA, avec un ensemble d'outils de base et avancés.
Encodeur Numérique	Un encodeur numérique électromécanique est un dispositif qui convertit la position ou le mouvement mécanique en signaux numériques.
Machine à état	Une machine à états est un modèle logique, elle change d'état en fonction de certains signaux, ce qui permet de contrôler des comportements complexes dans les systèmes électroniques.
MUX	MUX est l'abréviation du multiplexeur qui est un circuit logique qui permet de combiner plusieurs signaux d'entrée en un seul signal de sortie. Il sélectionne l'une des entrées parmi plusieurs et la transmet à la sortie en fonction des signaux de commande.

### B. Acronymes

Acronyme	Signification	Explication
FPGA	Field-Programmable Gate Array	Un FPGA est un circuit intégré configurable qui permet de créer des circuits numériques avec des portes logiques.
VHDL	VHSIC Hardware Description Language	Langage utilisé pour décrire et simuler des circuits électroniques de manière textuelle.
SMF	State machine file	Fichier qui définit les états et transitions d'une machine à états.
ADC	Analog-to-Digital Converter	Convertisseur qui transforme un signal analogique en signal numérique.
MIF	Memory Initialization File	Fichier utilisé pour initialiser le contenu d'une mémoire ROM dans les FPGA.
VHD	Virtual Hard Disk	Fichier qui simule un disque dur virtuel.
VGA	Video Graphics Array	Norme de connexion vidéo analogique utilisée pour relier des ordinateurs à des moniteurs ou projecteurs.
ROM	Read-Only Memory	Mémoire non volatile utilisée pour stocker des données fixes qui ne changent pas ou rarement.
BDF	Block Design File	Fichier dans Quartus permettant de créer des circuits numériques en assemblant des composants graphiques (schématiques).

### III. L'équipe

#### A. Présentation de l'équipe

Notre équipe est composée de 4 étudiants de deuxième année : Erwan PLANET, Eloi BRICQ, Hugo FAIVRE ainsi que Thibaud DANSETTE.

Chacun de nous apporte des qualités et des compétences, ce qui nous a permis de bien nous compléter pour aboutir à un produit fini.

Equipe composé de :	Compétences	Qualités
Erwan PLANET	Gestion de projet	Bonne logique, esprit analytique
Hugo FAIVRE	Simulation	Autonomie
Thibaud DANSETTE	Maîtrise du logiciel	Réfléchi
Eloi BRICQ	Maîtrise du logiciel	Rigoureux

**Figure 1** : Présentations de l'équipe

#### B. Organisation de l'équipe

Pour assurer le bon développement de ce projet, nous avons élaboré un diagramme de Gantt que nous avons essayé de suivre rigoureusement.

Nous avons travaillé majoritairement en présentiel à l'école, en utilisant nos heures libres et certaines soirées. Chacun d'entre nous a contribué au projet sur Quartus, que ce soit pour la création de machines à états ou la réalisation de schémas.

Voici comment nous avons réparti les exigences fonctionnelles entre nous pour optimiser notre efficacité :

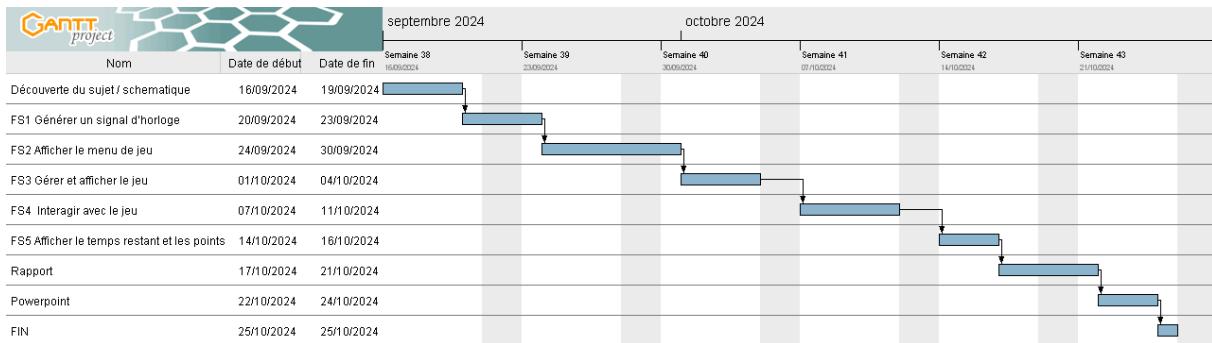
- Erwan a pris en charge la fonction FS2, qui consiste à afficher le menu.
- Eloi s'est occupé de la fonction FS4, qui permet d'interagir avec le jeu.
- Hugo a travaillé sur la fonction FS5, l'affichage du temps restant et des points.
- Thibaud a géré les fonctions FS1 (génération d'un signal d'horloge) et FS3 (génération et affichage du jeu).

A noter que certaines tâches plus complexes ont été réalisées par plusieurs membres de notre équipe. Pour ce qui est de notre rapport de projet, nous avons tous contribué à son élaboration ainsi que pour le PowerPoint.

Ce partage des responsabilités nous a permis de progresser efficacement tout en respectant les délais du projet.

## C. Diagramme de Gantt

Comme pour chaque projet il est important de concevoir un diagramme de gantt pour prendre connaissance de son avancé dans le planning et ainsi pouvoir voir plus facilement si l'on a du retard ou si nous sommes dans les temps.



**Figure 2 : Diagramme de GANTT**

## IV. Contexte et problématique

### A. Contexte

Il s'agit d'une adaptation du jeu vidéo "Fruit Ninja". Ce projet se concentre sur une implémentation de ce jeu sur FPGA, un environnement utilisé dans l'industrie pour des applications nécessitant des systèmes embarqués performants. Le contexte sociétal ici peut être lié à notre apprentissage et au développement de compétences techniques, contribuant ainsi à nous former pour devenir des ingénieurs capables de répondre aux défis industriels modernes.

Le jeu Fruit Ninja a été inventé par Halfbrick Studios, et sa technologie a évolué vers des versions VR et diverses adaptations.

Le projet utilise un FPGA pour gérer l'affichage et les interactions du jeu, une technologie qui permet de concevoir des circuits numériques programmables, évoluant depuis des années pour offrir des solutions de plus en plus complexes et puissantes dans les systèmes embarqués.

### B. Problématique

Le projet répond à la problématique de recréer un jeu vidéo interactif en utilisant FPGA, avec des contraintes techniques de gestion de signaux, d'affichage vidéo et d'interfaces homme-machine. L'objectif est de simuler un environnement ludique tout en respectant des exigences techniques spécifiques.

## C. Spécifications techniques

Le projet comprend les spécifications suivantes :

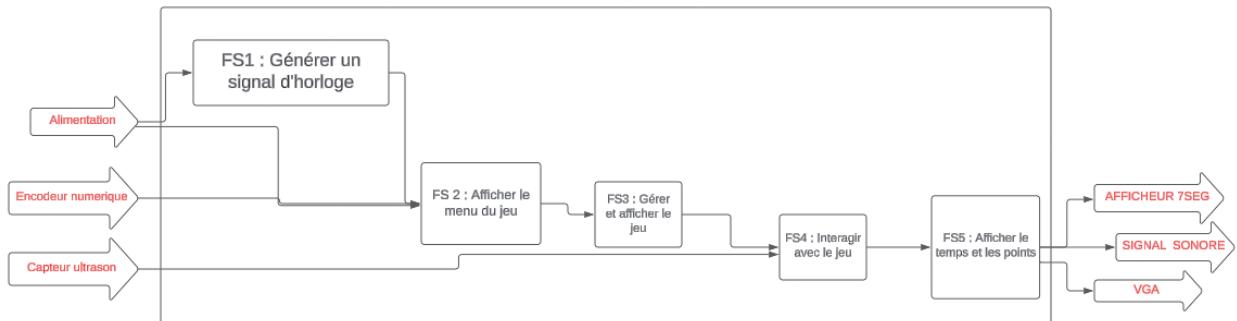
- Utilisation d'une machine à états pour gérer un jeu.
  - Affichage sur un moniteur via un port VGA, avec un score et un compteur de vies visibles.
  - Capteur à ultrasons et encodeur numérique pour contrôler les actions du joueur.
  - Des modes de jeu inspirés de "Fruit Ninja", avec des bombes, des fruits, et des combos

## V. Conception

## A. Architecture fonctionnelle

L'architecture fonctionnelle du projet repose sur les fonctionnalités suivantes :

- **Générer** des signaux vidéo pour afficher le jeu sur un moniteur.
  - **Afficher** le score et le nombre de vies du joueur en temps réel.
  - **Déetecter** les actions du joueur via un capteur à ultrasons pour déterminer la hauteur de coupe et un encodeur numérique pour l'angle de coupe.
  - **Interagir** avec le joueur à travers des sons produits par un buzzer, qui change selon l'action (fruits coupés, bombes, etc.).
  - **Gérer** un menu de sélection des modes de jeu (classique, zen, arcade).
  - **Générer et déplacer** des fruits et des bombes à l'écran, en ajustant leur vitesse et leur trajectoire.



**Figure 3 : Architecture fonctionnelle**

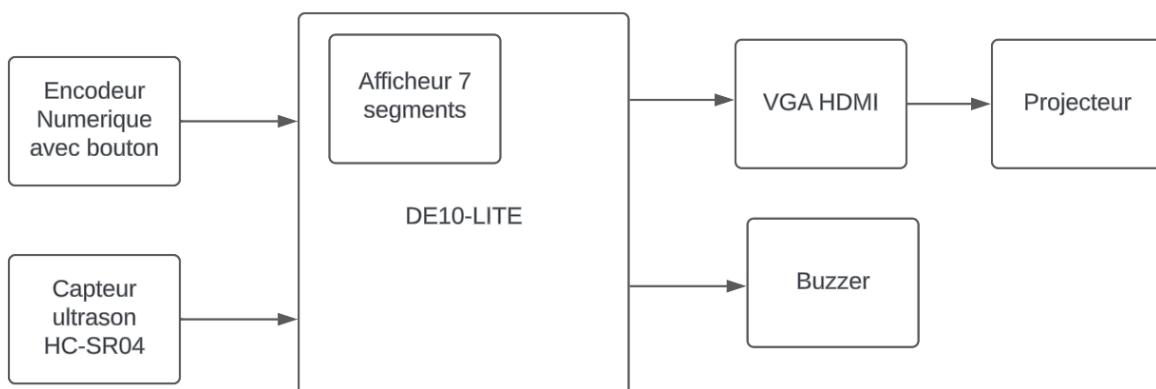
## B. Architecture matérielle

Le matériel utilisé comprend :

- **FPGA (DE10-Lite)** : Sert à contrôler l'ensemble du système, gérer les signaux vidéo, les entrées/sorties, et implémenter la logique du jeu.
- **Capteur à ultrasons** : Mesure la distance (la hauteur de coupe) à laquelle le joueur effectue l'action.
- **Encodeur numérique avec bouton poussoir** : Permet de se déplacer dans le menu et de sélectionner l'angle de coupe et d'activer la découpe des fruits.
- **Buzzer** : Génère des sons différents selon les actions (coupe de fruit, bombe touchée, etc.).
- **Afficheurs 7 segments** : Utilisés pour afficher le score et les informations de jeu.
- **Port VGA/HDMI** : relie le FPGA à un écran pour afficher le jeu.

Les différentes briques sont connectées comme suit :

- Le **FPGA** contrôle l'ensemble des composants. Le capteur à ultrasons et l'encodeur sont reliés en entrée au FPGA, qui reçoit les informations sur les actions du joueur. Les afficheurs, le buzzer et le port VGA sont en sortie, où le FPGA envoie des informations pour l'affichage et le retour sonore



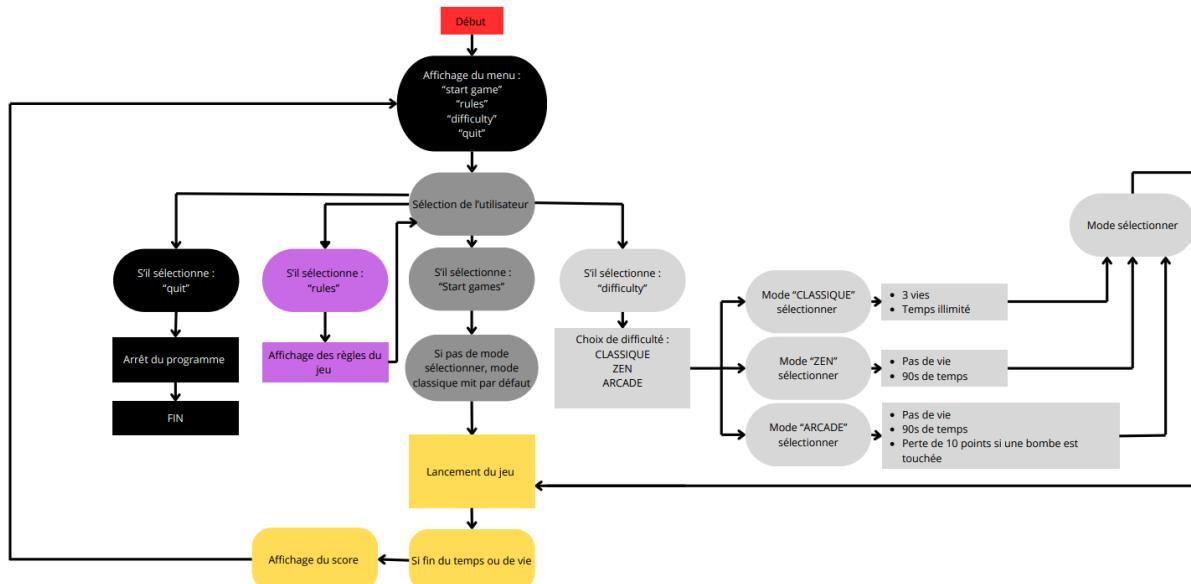
Architecture matérielle

**Figure 4** : Architecture matérielle

## C. Architecture logicielle

Le programme fonctionne sur une machine à états implémentée sur le FPGA:

1. **État initial** : Affichage du menu du jeu avec options (démarrer le jeu, sélectionner le mode, etc.).
2. **État de jeu** : Le FPGA génère les fruits de manière aléatoire, les fait apparaître sur l'écran, et détecte les actions du joueur (coupe, angle).
3. **Calcul des points** : En fonction des actions (fruits coupés, combos, etc...), les points sont calculés et mis à jour.
4. **Retour sonore** : Le buzzer réagit selon l'action (fruit coupé, bombe, échec).
5. **Fin de partie** : Le jeu se termine lorsque le joueur n'a plus de vies ou que le temps est écoulé (selon le mode choisi).



**Figure 5** : Architecture logicielle

## VI. Développement

### A. Module FS1 : Générer un signal d'horloge

Pour la mise en œuvre de ce FS1, nous avons voulu recycler notre TD2 pour avoir ce fameux signal de 1 Hz, or après renseignement nous avons dû par obligation utiliser des bascules JK et ça ne nous a pas posé de problème.

Pour la fonction FS1, notre objectif était de ralentir le signal d'horloge principal de 50 MHz afin de générer un signal de sortie de 1 Hz utilisable pour diverses fonctionnalités du jeu. Cela a été réalisé en utilisant des bascules JK configurées en mode diviseur d'horloge.

Pour les bascules JK nous avons dû en utiliser 26. Afin de connaître ce nombre de bascules nous avons effectué ce calcul  $2^n$ , avec n qui représente le nombre de bascule pour avoir le plus approximativement 1 Hz. Dans notre cas, il nous faut 26 bascules JK pour obtenir cette fréquence.

Chaque bascule JK divise la clock par 2.

De plus pour la fréquence nous avons eu pour consigne d'avoir moins de 40% de marge d'erreur, et nous avons eu que 35% ce qui est beaucoup mais nous sommes en dessous de la limite.

### B. Module FS2 : Afficher le menu du jeu

Pour notre menu, nous avons mis des images (start game, difficulty, rules, quit) et quand nous tournons notre encodeur numérique, nous passons à la l'image suivante, et lorsque que l'on veut sélectionner une image il faut cliquer avec l'encodeur numérique pour entrer dans le mode sélectionner.

- L'affichage VGA

Utilisation d'un port VGA pour l'affichage du menu sur un écran externe. La résolution et la fréquence de rafraîchissement ont été ajustées pour assurer une compatibilité avec la majorité des moniteurs et projecteurs. Des tampons de frame ont été utilisés pour éviter le scintillement et améliorer la fluidité des transitions entre les écrans.

- La lecture de l'encodeur numérique

Nous avons utilisé l'encodeur numérique pour naviguer dans le menu. Une attention particulière a été portée à la sensibilité de l'encodeur pour permettre une navigation précise sans problème de saturations.

- La lecture du bouton poussoir

Le bouton poussoir de l'encodeur numérique permet de sélectionner les options du menu et de sortir des options une fois rentré dedans.

- L'affichage du bouton start qui clignote à 1 Hz.

Un timer FPGA a été configuré pour faire clignoter le texte "Start" à une fréquence de 1 Hz, attirant l'attention des joueurs et indiquant le début possible d'une session de jeu.

- La machine à états du menu, et le moyen de revenir sur les pages précédentes.

Nous allons gérer le menu avec une machine à état qui sera contrôlée en entrée par un encodeur.

Un machine à état permet de remplacer du code.

## C. Module FS3 : Gérer et afficher le jeu

Ici nous allons voir comment nous pourrions gérer et afficher le jeu.

- Préparation des bitmaps, conversion et approche pour les afficher, les déplacer, etc.

Les images des fruits, des bombes, et des éléments d'interface utilisateur ont été conçues en utilisant des logiciels de graphisme puis converties en bitmaps. Ces images ont ensuite été transformées en formats de données adaptés pour leur utilisation dans un FPGA.

- Comment est géré l'aléatoire ?

Pour générer l'apparition aléatoire des fruits et de leur position, un générateur de nombres aléatoires basé sur un module SBPA donné dans les ressources. Cela assure une distribution équitable et imprévisible des fruits sur l'écran, augmentant ainsi le défi pour le joueur.

En sortie du SBPA nous avons un nombre binaire aléatoire basé sur 8 bits. Que l'on va devoir ensuite convertir en un entier et traiter cette donnée afin d'afficher aléatoirement le fruit.

Le fonctionnement est similaire pour la position d'apparition des fruits sur l'axe x.

- Comment est gérée la trajectoire ?

Nous allons essayer d'avoir une trajectoire parabolique pour essayer de représenter le mouvement en balistique.

$$y = -x^2$$

- Comment est géré le nombre de fruits à afficher ?

Le jeu limite le nombre maximum de fruits et de bombes affichés simultanément à l'écran pour éviter la saturation visuelle et limiter la charge sur le FPGA. Ce nombre est géré par un compteur qui suit les fruits tranchés et ceux qui sortent de l'écran.

Nous pensons relier la pin de la clock à celle de 1 Hz de manière à avoir l'apparition d'un fruit par seconde pour le niveau facile. Le fait d'avoir une fréquence d'apparition à 1 Hz limiterait le nombre de fruits.

- A-t-on une limitation liée à la taille de la ROM pour le stockage des images ?

La taille de la ROM disponible sur le FPGA impose certaines limitations sur le nombre et la résolution des images stockées(64 x 32 pour les fruits). Pour optimiser l'espace de stockage, les images des fruits ont été conçues en utilisant un nombre limité de couleurs et en partageant des palettes de couleurs communes quand cela était possible.

## D. Module FS4 : Interagir avec le jeu

Pour interagir avec le jeu nous avons un encodeur numérique et un capteur ultrason.

- Affichage de la lame

Pour afficher la lame nous allons afficher un pixel tous les 16 pixels, donc cela nous donne à la fin une ligne pointillée sur l'écran.

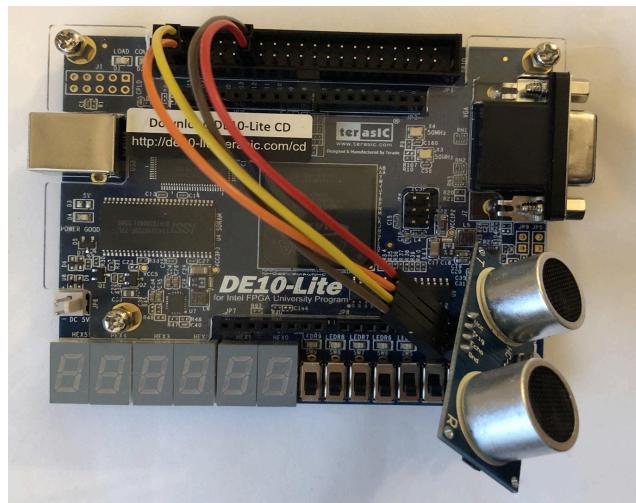
Nous allons changer la couleur de chaque pixel choisis en blanc qui se trouve sur l'axe des x pour faire la lame.

- Conversion de la combinaison de l'encodeur en un angle

Pour tourner la lame de découpe nous allons utiliser l'encodeur numérique, qui en fonction de son incrémentation pourra tourner la lame de découpe à l'aide de fonctions cosinus et sinus.

- Lecture de la distance avec le capteur à ultrasons

Pour cette partie nous avons récupéré une partie du TP3 sur le capteur à ultrason. Il va nous permettre de déterminer la distance à laquelle se trouve l'obstacle. Cette partie va nous permettre de diminuer ou augmenter la hauteur de la lame. Si l'obstacle est prêt du capteur, la lame sera en bas de l'écran, dans le cas contraire la lame sera en haut.



**Figure 6 :** Branchement du capteur à ultrason.

- Pilotage du buzzer et modulation du son

Le buzzer sera utilisé pour avertir le joueur quand il a découpé un fruit et fera un son différent lorsqu'une bombe est coupée. On utilisera donc une fréquence différente pour les 2 sons différents.

## E. Module FS5 : Afficher le temps restant et les points

Pour ce module nous allons utiliser l'afficheur 7 segments pour afficher le décompte du temps restant, nous utiliserons les deux afficheurs de gauche pour les secondes et pour les points sur les afficheurs de droite.

Pour le temps restant nous allons faire un décompte qui commence à 10 à une fréquence de 1Hz ce qui équivaut à 10 secondes.

## VII. Tests et validation

### A. Module FS1 : Générer un signal d'horloge

L'objectif ici était de passer d'une clock de 50 MHz à une clock de 1Mhz pour cela nous avons utilisé 26 bascule JK.

De plus nous avons dû respecter des exigences technique qui sont :

- L'horloge d'entrée doit être de 50 MHz.
- La fréquence d'horloge de sortie doit avoir au maximum  $\pm 40\%$  d'erreurs.
- Le diviseur d'horloge doit être conçu à l'aide de bascules JK



Figure 7 : RTL du diviseur d'horloge

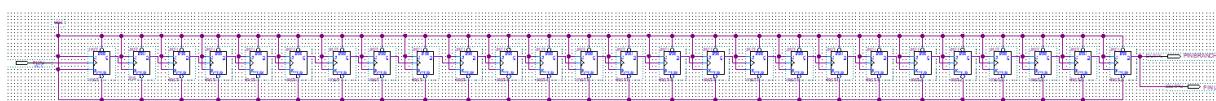
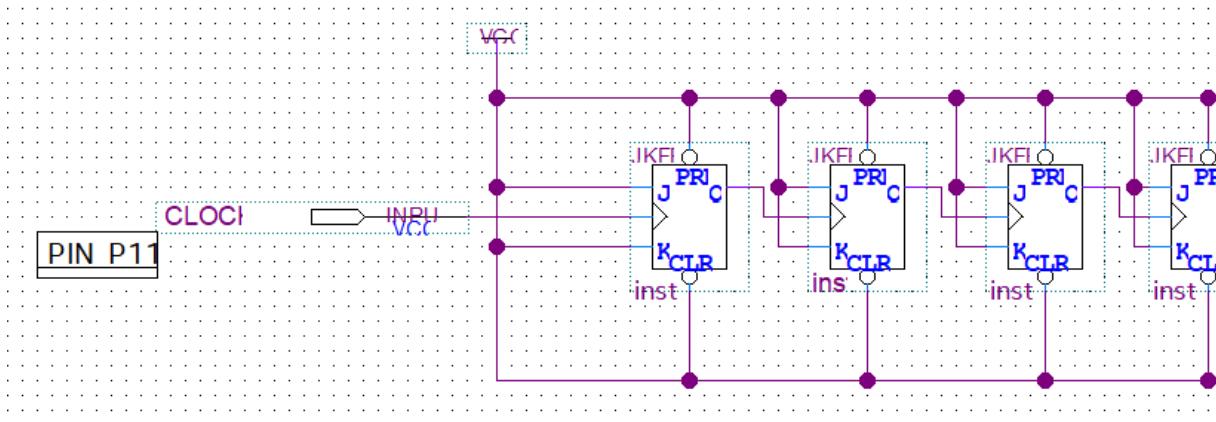
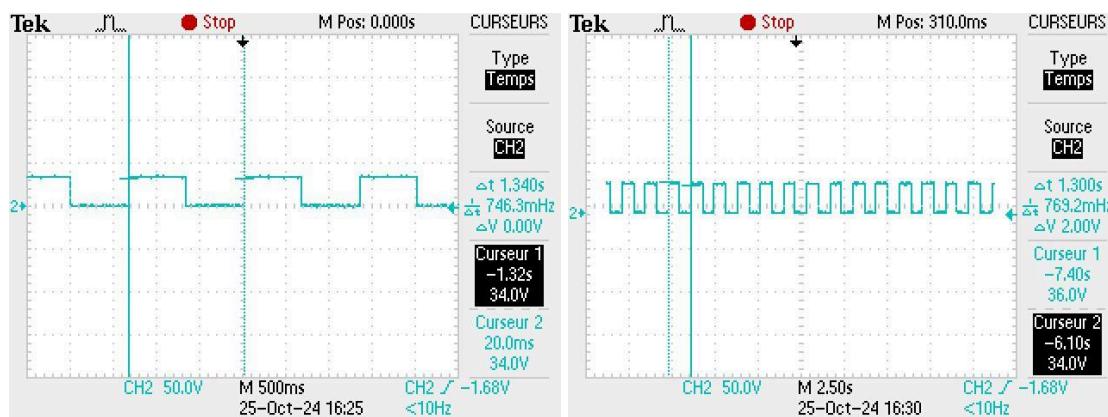


Figure 8 : Schématique du diviseur d'horloge



**Figure 9 :** Branchement des bascules JK entre elles.

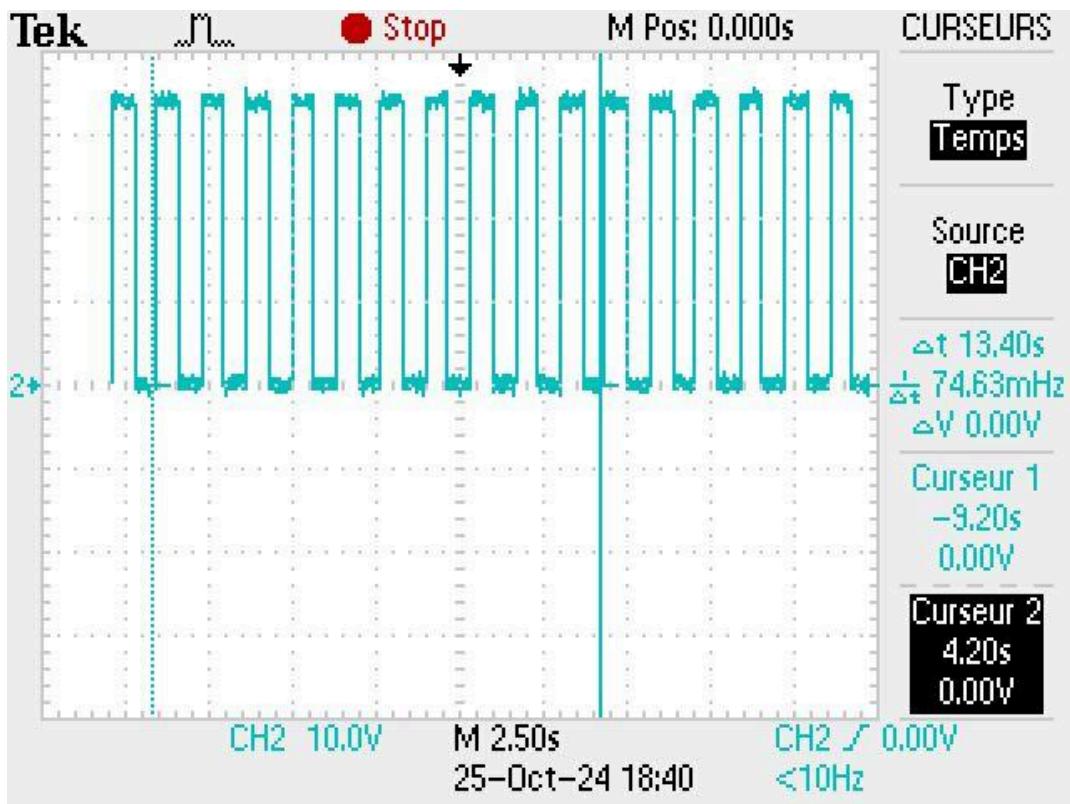
Nous avons ci dessus (figure 9) le branchement des 4 première bascules JK, en entrée la pin P11 va générer une fréquence de 50 MHz au diviseur d'horloge. Etant donné que chaque bascule divise la fréquence par deux nous aurons une fréquence de sortie d'environ 1,3Hz.



**Figure 10 :** Signal en sortie du diviseur d'horloge.

En sortie de notre diviseur d'horloge la clock est bien réduite, elle passe de 50 Mhz à environ 1 Hz.

Comme nous pouvons le voir sur les oscilloscopes ci-dessus nous avons un écart entre 2 fronts montant d'horloge de 1,3 secondes au lieu de 1 seconde soit environ 30% d'erreur et on voit que même sur beaucoup de périodes ce signal ne change pas on a donc bien un signal stable.



**Figure 11 :** Signal en sortie du diviseur d'horloge.

Nous pouvons voir que sur 10 périodes  $t = 13.40$  secondes au lieu de 10 secondes, les 40% d'erreurs sont validés soit 34% d'erreurs dans notre cas (cette valeur diffère d'au dessus car les curseurs de l'oscilloscope sont un peu déplacés par rapport au test sur une période mais l'erreur ne change pas).

Ce module est donc validé.

## B. Module FS2 : Afficher le menu du jeu



Figure 12 : Menu

les 2er images nous pouvons voir le "START" avec une image noir à droite, c'est l'image qui clignote sur le signal 1Hz

ensuite nous avons l'image "RULES" avec l'image des règles qui s'affiche lorsque l'on clique sur l'encodeur numérique

dans un troisième temps, nous avons les images "QUIT" et "START GAME"

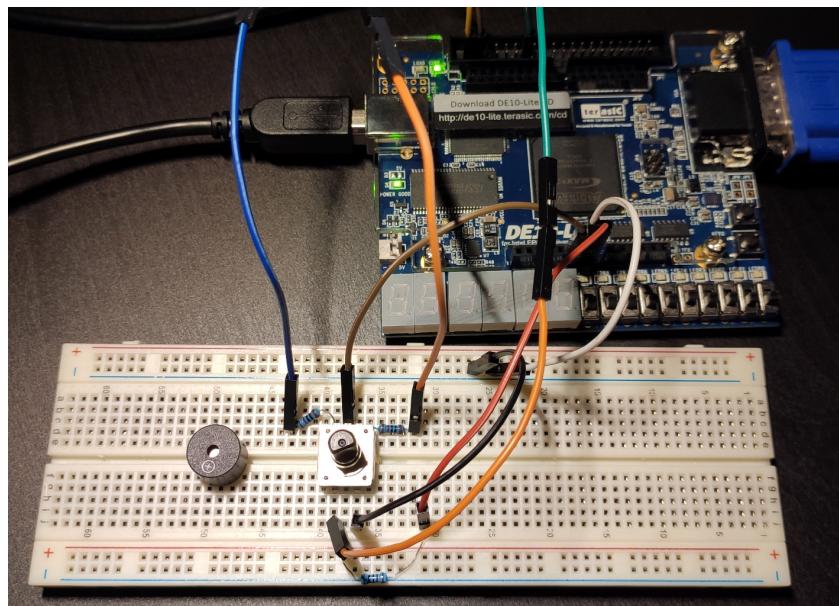


Figure 12 : Branchement

et pour finir l'image "DIFFICULTY" et quand nous cliquons dessus un message apparaît

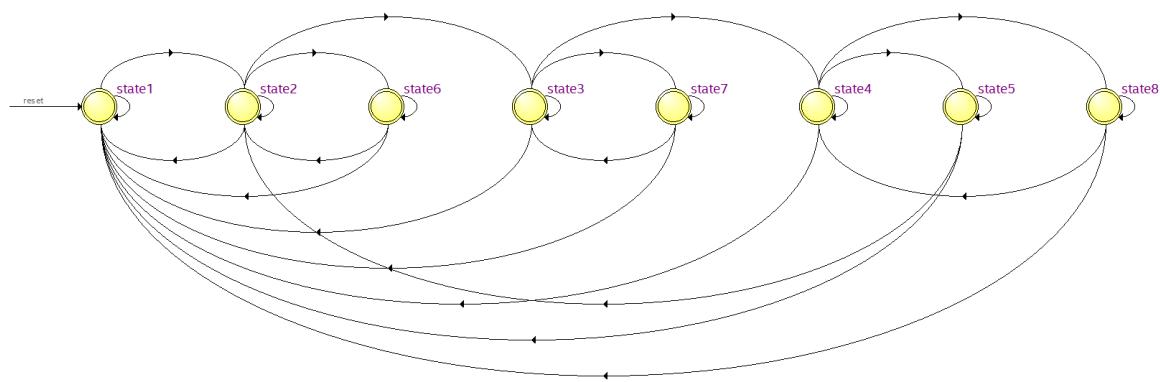


Figure 13 : Machine à état.

Sur cette machine à états, on peut voir que les états 2, 3, 4 et 5 correspondent aux 4 options principales du menu et chacune de ces options renvoie sur l'option lancée dans un nouveau state si on clique sur le bouton de l'encodeur.

Ce module est donc validé.

### C. Module FS3 : Gérer et afficher le jeu

Pour la génération aléatoire des fruits nous avons repris notre signal de 1 Hz pour le mettre dans le module SBPA ci-dessous.

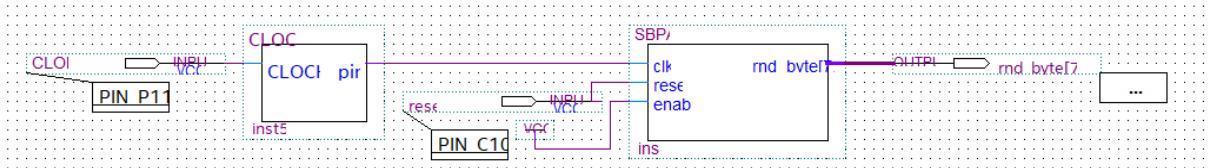


Figure 14 : Generateur de nombre aléatoire.

## D. Module FS4 : Interagir avec le jeu

Afin de pouvoir couper les fruits nous allons utiliser un encodeur qui permettra de pencher la lame ainsi qu'un capteur à ultrason pour régler la hauteur de la lame.

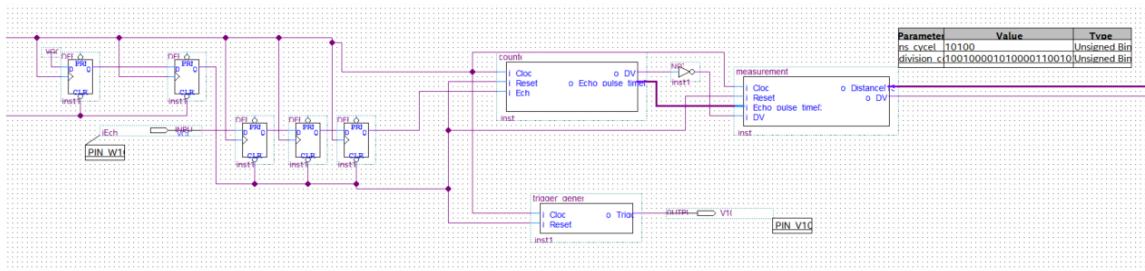


Figure 15 : Schématique du capteur à ultrason.

Nous avons donc récupéré ces blocs qui faisaient partie du TP3.

Le groupe a essayé de brancher au VGA contrôleur les sorties du bloc du capteur à ultrason. Pour cela nous avons dû modifier le VHDL du VGA controller. Il a aussi fallu traiter le signal du capteur qui est composé d'un nombre binaire sur 14 bits.

Et ensuite régler la hauteur de la lame en fonction de l'entier converti.

Lors de la phase test nous avons eu la lame qui bougeait mais sans succès, ce n'était pas du tout fluide et nous n'avons pas réussi à lui faire parcourir tout l'écran.

## **E. Module FS5 : Afficher le temps restants et les points**

Ce module n'a pas été validé à temps.

# **VIII. Bilan**

## **A. État d'avancement**

Notre projet a atteint les % des objectifs, mais il a son thème, son menu, et nous avons nos RTL et une machine à état.

Les modules qui restent à finaliser sont le jeu principal qui a été avancé (FS3), les interactions dans le jeu principal avec l'encodeur numérique et le capteur ultrason (FS4) et enfin l'affichage du temps et du nombre de points gagnés durant la partie (FS5).

## **B. Pertinence de la solution technique**

Nous étions limité par la faible mémoire de la FPGA, 230 KB au total alors qu'une image de 640\*480 fait 110 KB, pour tout un projet donc il fallait faire attention à la taille des nos images, que nous avons redimensionnées au maximum à 200\*100. Nous avons également rencontré des problèmes d'interférence en fonction de l'environnement dans lequel nous étions car la carte FPGA est très sensible aux divers champs électrostatiques et électromagnétiques qui nous entourent.

Les possibilités d'évolution ou de poursuite de ce projet sont tout d'abord de terminer dans son entiereté le jeu et d'incorporer d'éventuels autres niveaux de difficultés, d'autres fruits, d'autres types d'objets à ne pas découper ou encore des sons variés lorsqu'on coupe des fruits et bombes.

## C. Bilan sur le travail d'équipe

Prénom/nom	Apprentissage	Compétences
Hugo Faivre	Ce que j'ai retenu de ce projet c'est l'utilisation du nouveau logiciel quartus qui était plutôt complexe à comprendre avec des states machines, du VHDL, des MUX etc...	Pour la recherche de mon stage je pourrais maintenant mettre dans mon CV que je sais utiliser le logiciel QUARTUS
Eloi Bricq	J'ai pu développer ma logique combinatoire et appris à comprendre le VHDL qui n'est pas si différent des langages de code vu en cours. J'ai également appris et compris l'utilisation et le fonctionnement de l'affichage de sprites via les ROM ainsi que des machines à états.	Les compétences pouvant être mises en avant lors d'une prochaine recherche de stage sont l'utilisation du logiciel quartus et une bonne logique combinatoire.
Thibaud Dansette	J'ai pu approfondir mes connaissances avec des notions plus avancées que celles abordées en TD comme les machines à état. J'ai également remarqué que le VHDL est un langage descriptif permettant de modéliser différents blocs logiques.	Ce projet m'a permis de renforcer l'esprit d'équipe, d'améliorer mes compétences organisationnelles et de maîtriser l'utilisation de Quartus. Ces compétences seraient un atout à mentionner sur mon CV.
Erwan Planet	Lors de la réalisation de notre projet j'ai appris à bien maîtriser le logiciel Quartus, développer ma logique et appliquer les acquis que l'on a vu pendant les TP et cours.	Les compétences que je vais pouvoir mettre en avant sont la maîtrise de Quartus, la réflexion pour trouver comment créer un système uniquement avec des composants logiques et un bon management pour le travail en équipe.

Durant toute la durée de ce projet notre équipe était bien organisée étant donné que l'on s'est lancés directement dans le projet dès que le sujet était disponible mais étant donné la difficulté assez élevée du projet nous avons vite été bloqués et n'avons pas pu avancer jusqu'à 5 jours avant la deadline. Pour un prochain projet, on pourrait éventuellement demander plus de questions aux professeurs en décrivant mieux nos questions pour avoir la réponse qui nous aidera vraiment.

## IX. Bibliographie

- [1] Dr Schneider, Maxime, «La Toolbox,» [En ligne]. Available:  
[https://boostcamp.omneseducation.com/pluginfile.php/3538704/mod\\_resource/content/1/RAPPORT.pdf](https://boostcamp.omneseducation.com/pluginfile.php/3538704/mod_resource/content/1/RAPPORT.pdf). [Accès le 14 juin 2024].
- [2] IEEE, «IEEE Editorial Style Manual,» 29 février 2024. [En ligne]. Available:  
<https://journals.ieeeauthorcenter.ieee.org/your-role-in-article-production/ieee-editorial-style-manual/>. [Accès le 14 juin 2024].

Cette section doit contenir tous les documents utilisés et sites internet consultés pour développer le projet. La syntaxe à utiliser (le standard bibliographique IEEE) est détaillée dans le document « Comment rédiger un rapport » [1].

Lorsqu'ils sont référencés dans le corps de texte du rapport, un renvoi numéroté doit apparaître, conformément au standard bibliographique IEEE [2].

- <https://boostcamp.omneseducation.com/course/view.php?id=359858&section=21#tabs-tree-start>
- [https://boostcamp.omneseducation.com/pluginfile.php/4601949/mod\\_resource/content/3/NINJAFRUIT\\_aideTechnique.pdf](https://boostcamp.omneseducation.com/pluginfile.php/4601949/mod_resource/content/3/NINJAFRUIT_aideTechnique.pdf)
- [https://mil.ufl.edu/3701/docs/quartus/rom\\_creation.pdf](https://mil.ufl.edu/3701/docs/quartus/rom_creation.pdf)
- <https://passionelectronique.fr/encodeur-rotatif-incremental-mecanique/>
- [https://ftp.intel.com/Public/Pub/fpgaup/pub/Intel\\_Material/Boards/DE10-Lite/DE10\\_Lite\\_User\\_Manual.pdf](https://ftp.intel.com/Public/Pub/fpgaup/pub/Intel_Material/Boards/DE10-Lite/DE10_Lite_User_Manual.pdf)
- <https://cse.google.fr/cse?cx=partner-pub-8110924381468607:6449306375&ie=ISO-8859-1&q=machine+a+%C3%A9tat&sa=Rechercher>
- [https://boostcamp.omneseducation.com/pluginfile.php/4129451/mod\\_resource/content/2/TP4.pdf](https://boostcamp.omneseducation.com/pluginfile.php/4129451/mod_resource/content/2/TP4.pdf)
- <https://www.intel.fr/content/www/fr/fr/content-details/649278/how-to-create-a-state-machine-with-the-quartus-state-machine-wizard.html>
- <https://www.bing.com/videos/riverview/relatedvideo?q=state+machine+quartus&mid=7811BF8A0CF94294F1667811BF8A0CF94294F166&FORM=VIRE>
- <https://www.youtube.com/watch?v=XN3QAjWEREM>
- <https://boostcamp.omneseducation.com/course/view.php?id=344636>
- [https://boostcamp.omneseducation.com/pluginfile.php/4660149/mod\\_resource/content/3/FRUITECE\\_QA1.pdf](https://boostcamp.omneseducation.com/pluginfile.php/4660149/mod_resource/content/3/FRUITECE_QA1.pdf)
- [https://boostcamp.omneseducation.com/pluginfile.php/4679465/mod\\_resource/content/5/FRUITECE\\_QA2.pdf](https://boostcamp.omneseducation.com/pluginfile.php/4679465/mod_resource/content/5/FRUITECE_QA2.pdf)

## X. Annexes

Documents annexes, éventuels codes (**pas de code dans le rapport**).