



**INSTITUT UNIVERSITAIRE
DE TECHNOLOGIE**

UNIVERSITÉ D'ANGERS

IUT Angers

Rapport Mini Projet SIN

Étudiants :

Matéis RAGON

Professeurs :

M. Sylvain PEZERIL

M. Benjamin DUTRUCH

14 janvier 2025

Table des matières

1	Rappel du cahier des charges	2
2	Schéma fonctionnel	2
3	Schéma fonctionnel Quartus®	3
4	Fonctions utilisées	5
5	Le graphe des états	6
6	Conclusion	7
6.1	Ce qui fonctionne	7
6.2	Ce qui diffère du cahier des charges	7

1 Rappel du cahier des charges

Un **Shot Timer** est un dispositif électronique utilisé principalement dans le cadre du tir sportif ou d'entraînement au tir, que ce soit avec des armes à feu, des armes airsoft. Sa fonction principale est de mesurer les temps liés aux tirs, permettant ainsi aux tireurs d'évaluer leur vitesse, leur efficacité et leurs performances.

Ici, nous ne traiterons pas les fonctions d'un Shot Time classique tel que :

- L'affichage des statistiques (Split Time¹ moyen, temps de réaction du premier tir)
- La programmation de scénarios
- Sauvegarde des sessions de tirs
- Démarrage aléatoire de la session de tir

En revanche, nous traiterons les fonctions telles que :

- Lancer une session de tir
- Arrêter une session de tir s'il n'y a pas de tir pendant 10s
- Reset le Shot Timer
- Comptabiliser les tirs
- Afficher des statistiques basiques (meilleur Split Time, temps total, nombre de tirs total)

2 Schéma fonctionnel

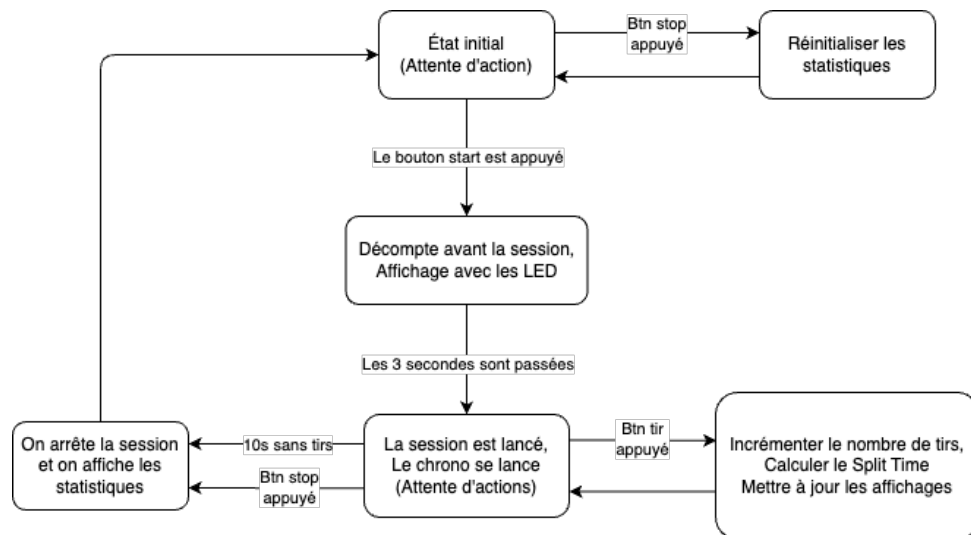


FIGURE 1 – Schéma fonctionnel du Shot Timer

Notre schéma fonctionnel est composé de trois grands blocs. Le premier bloc est constitué d'un état initial et d'une réinitialisation des statistiques. Le second bloc gère le décompte avant la session de tir. Et pour finir, le troisième s'occupe de la session en elle-même.

1. Split Time : Temps entre deux tirs

3 Schéma fonctionnel Quartus®

Quartus® est un logiciel édité par Intel qui permet de développer sur des cartes telles que la "Altera Cyclone 2 Education Board" qu'on utilise en cours. Ce logiciel permet de développer des programmes sous forme de blocs logiques, et le logiciel compile ensuite les logigrammes en langage VHDL.

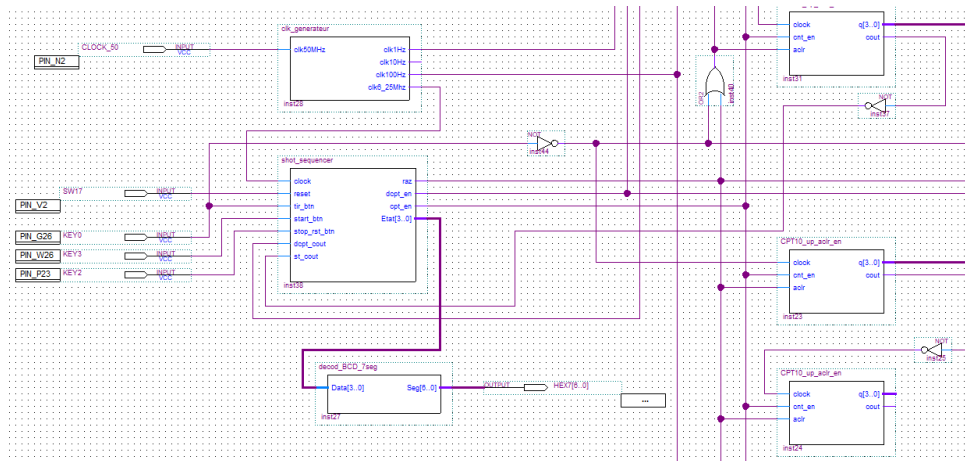


FIGURE 2 – Schéma Quartus® - Séquenceur et compteurs de tirs

Sur la figure ci-dessus est mis le séquenceur, le générateur d'horloge ainsi que les compteurs de tir. Il y a deux compteurs, car à l'origine, il est possible de tir jusqu'à 99 fois, cependant à défaut d'avoir assez d'afficheurs sept segments, il a été choisi que l'afficheur des dizaines servirait pour afficher les états du séquenceur.

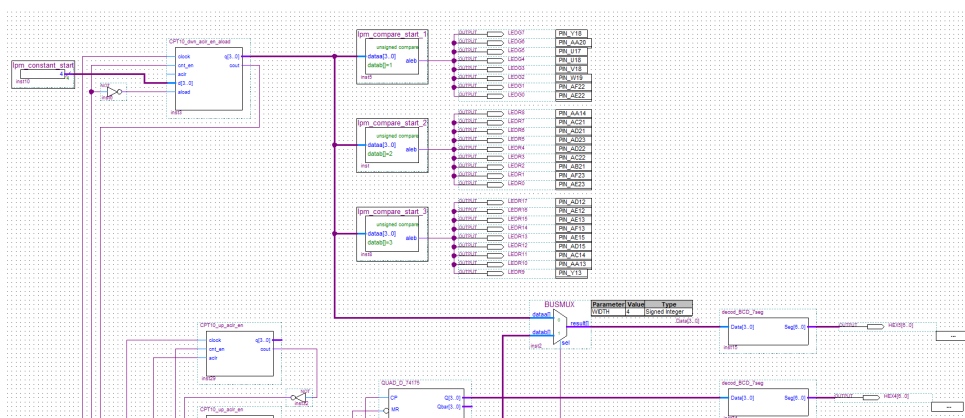


FIGURE 3 – Schéma Quartus® - Décompte avant la session de tir

La partie de décompte avant la session de tir est composée d'un compteur, qui est initialisé à la valeur 4. Elle-même mise dans le chargement parallèle avec une constante. Nous avons ensuite un comparateur pour chaque valeur du compteur pour aller à zéro (trois, deux et un). Puis ces comparateurs délivrent un niveau logique haut pour allumer les leds.

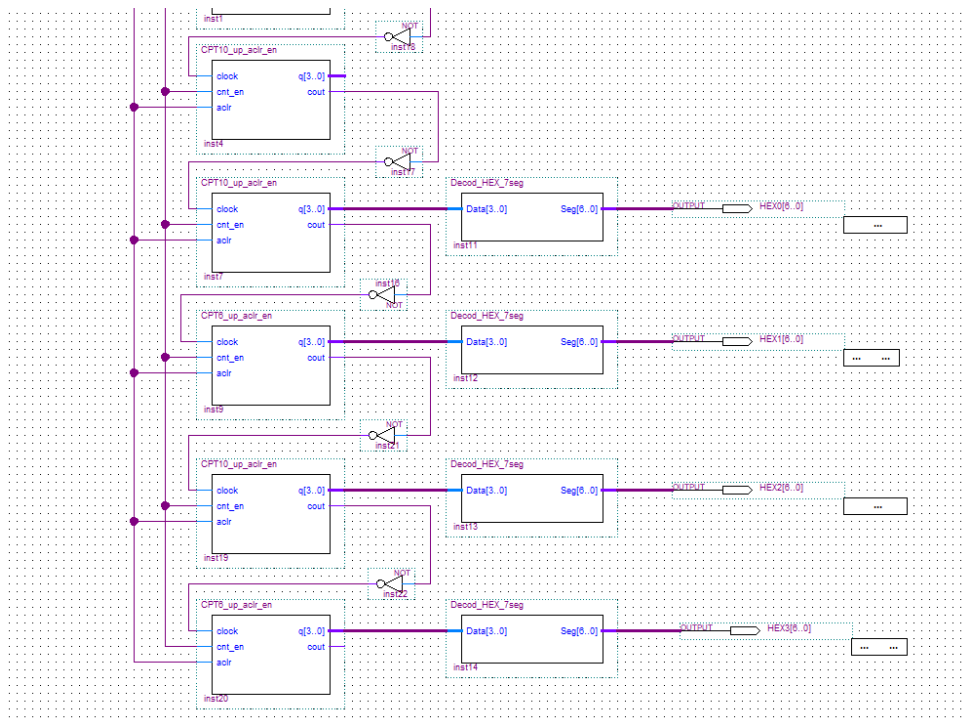


FIGURE 4 – Schéma **Quartus®** - Compteurs du temps

La partie des compteurs de temps est assez simple puisqu'elle ne contient uniquement des compteurs qui s'enchainent pour former un chronomètre, minutes, secondes et dixième de secondes. Ici, il a été fait le choix de n'afficher que les minutes et secondes, encore une fois par limitation matérielle.

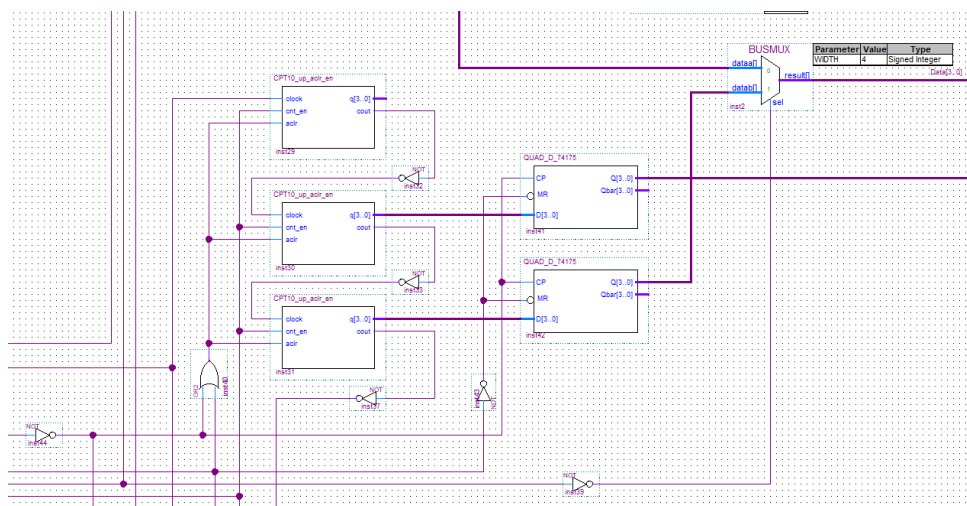


FIGURE 5 – Schéma **Quartus®** - Compteurs et bascules pour le Split Time

Pour la partie Split Time, on retrouve trois compteurs, deux servent aux dixièmes de secondes et un est pour les secondes. Ensuite, nous retrouvons deux bascules D 4 bits "74175". Elles servent à bloquer le temps du Split Time après chaque tir jusqu'au suivant. On retrouve également un multiplexeur 4 bits qui permet d'alterner entre l'affichage du Split Time ou du décompte.

4 Fonctions utilisées

Dans ce projet, nous avons utilisé diverses fonctions de base et certaines fonctions que nous avons modifiées pour des besoins spécifiques. (nous verrons les fonctions utilisées dans l'ordre des figures ci-dessus)

- Le générateur d'horloge "*clk_generateur*" :
Nous avons utilisé ce générateur d'horloge avec comme horloge 1Hz pour le décompte, 100Hz pour les compteurs de temps du Split Time et du chronomètre, et 25MHz pour l'horloge du séquenceur.
- Le séquenceur "*shot_sequencer*" :
Nous avons fait un séquenceur sur mesure que nous détaillerons dans la partie du séquenceur prévu dans le document à cet effet.
- Les compteurs "*CPT10_up_aclr_en*" :
Nous avons utilisé ces compteurs spécifiquement parce qu'ils nous permettent de compter en base 10. Et il possède une entrée enable qui permet d'activer ou non le compteur.
- Les compteurs "*CPT10_down_aclr_en_aload*" :
Nous avons utilisé ces compteurs spécifiquement parce qu'ils nous permettent de décompter en base 10. Il possède une entrée enable qui permet d'activer ou non le compteur. Ainsi qu'une entrée pour le chargement parallèle.
- La constante "*lpm_constant_start*" :
Cette constante, que nous mettons dans le chargement parallèle, nous permet de fixer le départ du décompte avant la session de tir à la valeur 4.
- Les comparateurs "*lpm_compare_start_\$*" :
Ces comparateurs permettent d'allumer des leds misent en aval de ces compteurs.
- Le multiplexeur 4 bits "*BUSMUX*" :
Ce multiplexeur répond à la problématique qu'il n'y avait pas assez d'afficheurs sept segments sur la maquette Altera®. Cette solution nous permet en entrant deux bus 4 bits et une entrée de sélection de pouvoir permuter entre les deux bus.
- Les bascules D 4 bits "*QUAD_D_74175*" :
Ces bascules D de modèle 74175 nous permettent de pouvoir figer la sortie des compteurs qui prennent le Split Time pour pouvoir l'envoyer dans les afficheurs sept segments.

5 Le graphe des états

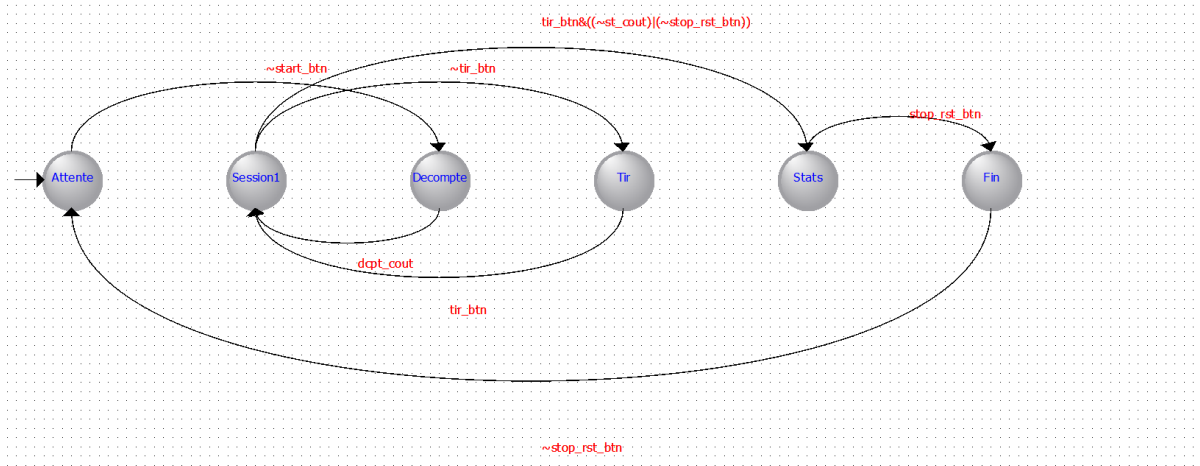


FIGURE 6 – Séquenceur Quartus®

État	Sorties	Transition	Description
Attente	$RAZ = 1$ $dcpt_en = 0$ $cpt_en = 0$ $Etat[3 : 0] = 1$	Not Bouton_Start	Si le bouton de start est pressé alors, on passe à l'état de Décompte.
Decompte	$RAZ = 0$ $dcpt_en = 1$ $cpt_en = 0$ $Etat[3 : 0] = 2$	Compteur_Decompte = 0	Quand le compteur arrive à 0 on passe à l'état de la Session1.
Session1	$RAZ = 0$ $dcpt_en = 0$ $cpt_en = 0$ $Etat[3 : 0] = 3$	Not Bouton_Tir OU Bouton_Tir & (Compteur_Split_Time Not Bouton_Stop_Reset)	Si le bouton de tir est pressé alors, on passe à l'état de Tir. Sinon on passe à l'état de Stats
Tir	$RAZ = 0$ $dcpt_en = 0$ $cpt_en = 0$ $Etat[3 : 0] = 4$	Bouton_Tir	Si le bouton de Tir est relâché alors, on passe à l'état de la Session1.
Stats	$RAZ = 0$ $dcpt_en = 0$ $cpt_en = 0$ $Etat[3 : 0] = 5$	Bouton_Stop_Reset	Si le bouton de Stop Reset est pressé alors, on passe à l'état de Fin.
Fin	$RAZ = 0$ $dcpt_en = 0$ $cpt_en = 0$ $Etat[3 : 0] = 6$	Not Bouton_Stop_Reset	Si le bouton de Stop Reset est relâché alors, on passe à l'état d'Attente.

6 Conclusion

6.1 Ce qui fonctionne

Attendu	Résultat	Commentaire
Lancer une session de tir	OK	N/A
Arrêter une session de tir s'il n'y a pas de tir pendant 15 s	OK	La session de tir, s'arrête au bout de 9 s finalement
Comptabiliser les tirs	OK	N/A
Afficher des statistiques basiques	OK	N/A
Meilleur Split Time	PAS OK	L'affichage du meilleur Split Time impliquais d'utiliser des fonctionnalités avancées de Quartus® et ce n'était pas possible dans le temps alouée.
Temps total	OK	N/A
Nombre total de tirs	OK	N/A

6.2 Ce qui diffère du cahier des charges

Comme vu dans la section précédente, nous avons changé l'exigence que la session de tir s'arrête au bout de 15 secondes, nous avons modifié ça pour que ce soit au bout de 9 secondes.

Par manque de temps et sûrement avec trop d'ambition, je n'ai pas pu implémenter l'affichage du meilleur Split Time, celui-ci nécessitait des fonctions avancées de **Quartus®** et je n'ai pas pu les mettre en place.