

Rapport du projet de fin de semestre de systèmes logiques

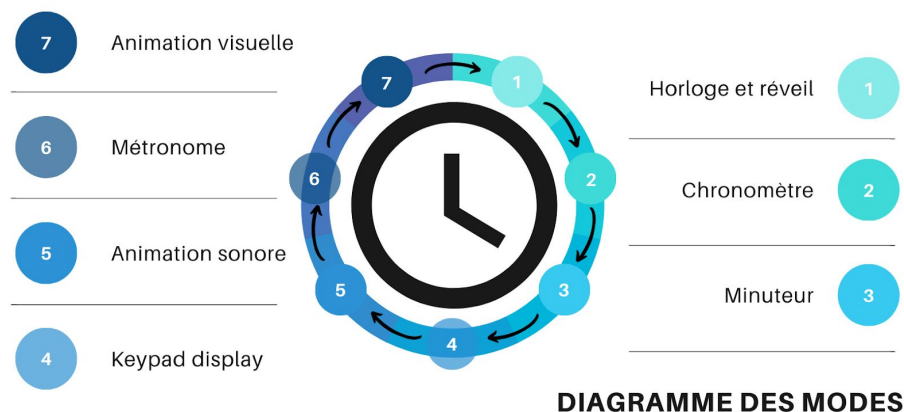
I. Description générale et mode d'emploi

A. Description générale du projet

Le projet consiste en l'élaboration d'une montre digitale conçue sur Logisim-evolution et développée sur une carte DE10-lite. Cette montre doit comporter les fonctions usuelles d'une montre digitale (affichage de l'heure, chronomètre, minuteur, réveil) ainsi que des fonctions plus élaborées (animations sonores, animations visuelles etc.) La montre aura pour périphériques d'entrée un 4x4Keypad, deux boutons poussoirs ainsi que 10 DIP-Switches. Elle aura comme périphériques de sortie des LEDs, six afficheurs 7-segments et un buzzer.

B. Mode d'emploi

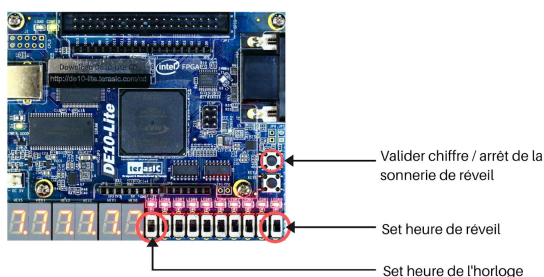
La montre a été conçue dans le souci de rendre son utilisation la plus intuitive possible. Elle propose à l'utilisateur de passer d'un mode au suivant en appuyant sur un bouton poussoir : celui du haut. Les modes de la montre sont dans l'ordre : l'affichage de l'heure et le réveil, le chronomètre, le minuteur, l'affichage des touches du Keypad, l'animation musicale, le métronome et l'animation visuelle des LEDs.



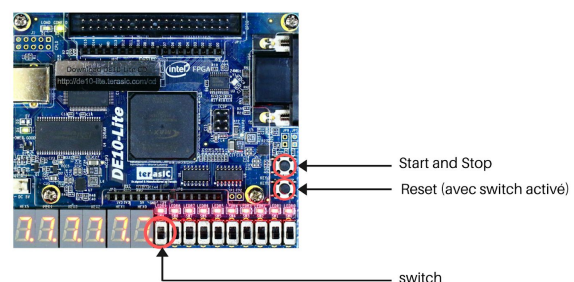
Tous les sets de chiffres sur les afficheurs se font avec un DIP-switch activé. Pour passer à l'afficheur suivant, il faut cliquer sur le bouton du haut.

Dans chaque mode, il est possible de réinitialiser la montre en cliquant sur le bouton du bas.

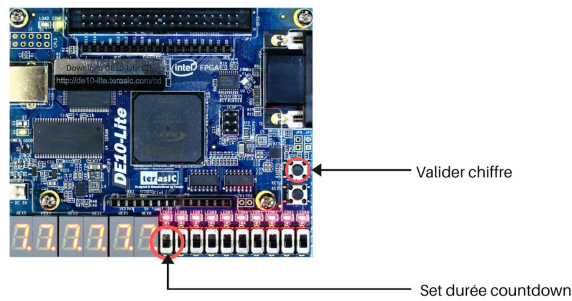
Horloge et réveil : les afficheurs montrent l'heure et le buzzer sonne à l'heure set par l'utilisateur relativement à l'heure à laquelle il a set l'horloge.



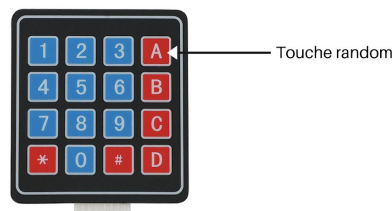
Chronomètre : les afficheurs montrent le temps qui défile entre un *start* et un *stop*.



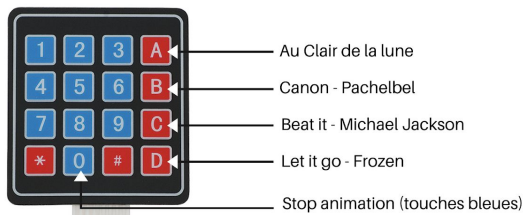
Minuteur : les afficheurs montrent la durée set qui décroît et le buzzer sonne quand le temps est à 0.



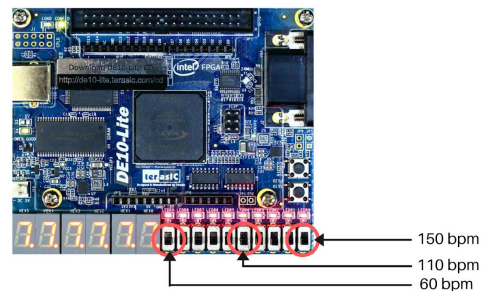
Keypad display : n'importe quelle touche pressée est affichée sur les 6 afficheurs.



Animation sonore : le buzzer émet la musique choisie.

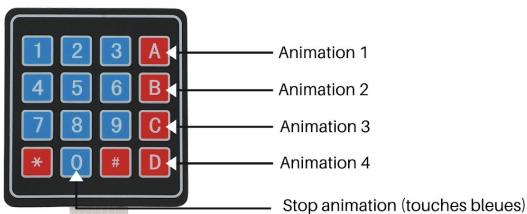


Métronome : le buzzer émet une pulsation à une fréquence choisie.



NB : si plusieurs DIP-switches sont à logic-1, c'est celui relatif à la plus grande fréquence qui est pris en compte.

Animation visuelle : des LEDs clignotent selon un mode déterminé par l'animation choisie.



II. Problèmes rencontrés et solutions apportées

A. Description de la machine d'état générale

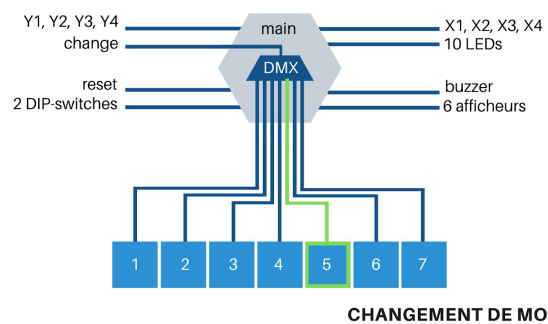
La fonction *main* du projet s'articule autour d'une combinaison de Demultiplexer/Multiplexer alimentés par un compteur 3-bits incrémenté par un bouton poussoir (qui change de mode). Cette combinaison permet aux entrées et sorties d'être exploitées spécifiquement pour la fonction active et non pour les autres. L'autre bouton poussoir est utilisé principalement pour le *reset* général, c'est à dire celui qui permet à l'utilisateur de réinitialiser sa montre complètement.

Problèmes rencontrés et solutions apportées :

1. Mode on

Au début du projet, nous avons essayé d'utiliser une variable que chaque fonction changerait en son sein et qu'elle modifierait pour activer le mode suivant.

On aurait donc eu un `logic_1` qui se serait répandu d'une fonction à la suivante. Nous avons plutôt opté pour un *main* comme décrit ci-dessus. Cette option réduit drastiquement le risque de bugs potentiels. Ainsi, chaque mode est indépendant et le système est donc plus robuste.

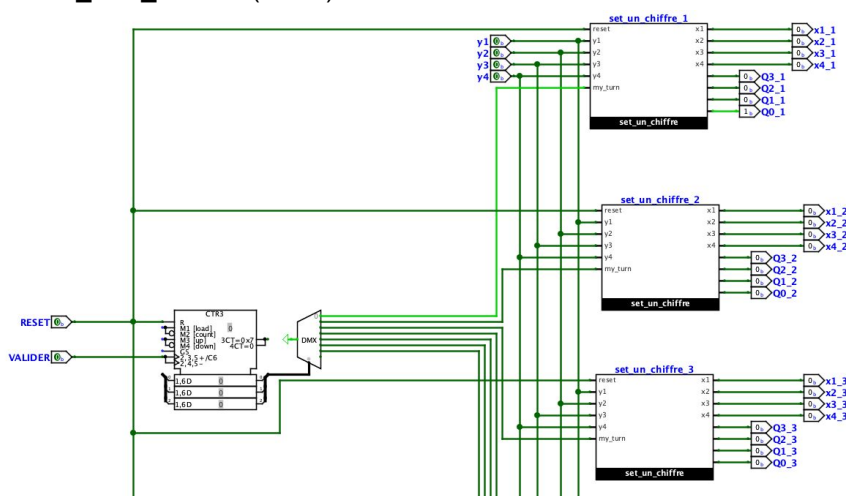


2. Entrées et sorties *hex_dec*

Nous avons eu des difficultés certaines avec l'utilisation du Keypad, à commencer par comprendre et compléter le *bubble-counter* fourni par le professeur. Nous sommes restés longtemps bloqués sans pouvoir utiliser le Keypad puisque nous n'avions pas trouvé la bonne fréquence d'utilisation (nous avons finalement pris 256kHz).

Aussi, dans la phase de développement du set de chiffres sur le keypad, nous testions dans un premier temps la fonction *set_un_chiffre* dont les entrées et les sorties étaient Y et X respectivement.

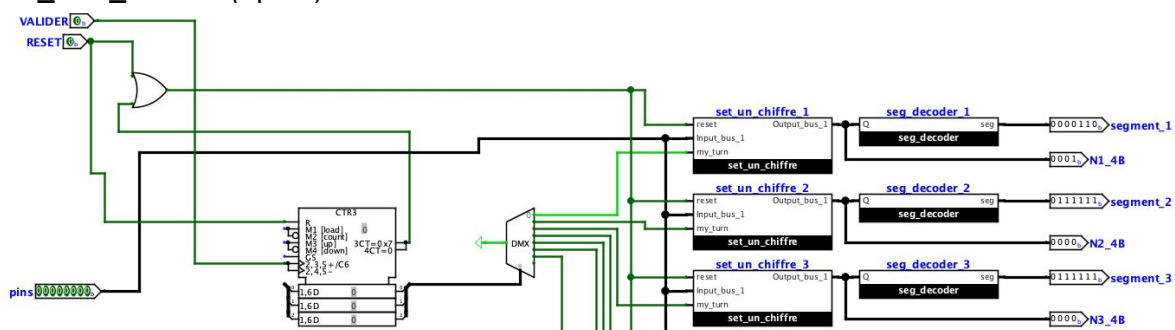
set_tous_chiffres (avant) :



Ensuite, il a fallu utiliser la fonction *set_tous_chiffres* et est apparu le problème des X. Il y avait autant de paquets de (X1, X2, X3, X4) qu'il n'y avait d'afficheurs comme le montre la figure ci-contre. Il fallait traiter les X de telle sorte que les seules sorties désirées étaient celles intrinsèques au bon afficheur.

Finalement, nous avons court-circuité le problème en plaçant la fonction *hex_dec* seulement dans le *main*. Cela a aussi eu pour conséquence de clarifier notre programme.

set_tous_chiffres (après) :



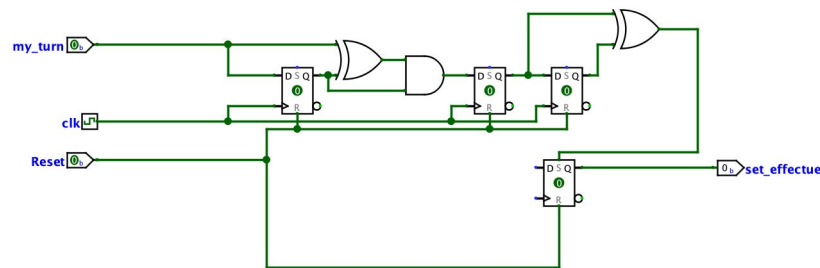
B. Description des machines d'états secondaires

Le choix des modes secondaires a été régi par notre volonté de rendre la montre la plus utile possible c'est à dire la plus complète possible.

Problèmes rencontrés et solutions apportées :

1. Set des chiffres

Nous avons pris le parti de considérer que l'utilisateur, lors des sets des chiffres (set de l'heure de l'horloge, set de l'heure du réveil, set de la durée de countdown) avait le droit d'appuyer autant qu'il le souhaite sur le keypad pour corriger son entrée. C'est pourquoi il doit presser un bouton pour signifier qu'il confirme le dernier chiffre tapé et qu'il souhaite entrer le suivant. L'enjeu est donc de passer d'un chiffre à un autre sans perdre l'information des derniers chiffres entrés. Nous avons décidé de créer une variable *set_effectué* qui est à logic-0 tant que le set du chiffre n'est pas effectué et qui passe à logic-1 quand le set est effectué. Il s'agit donc pour *set_effectué* d'être sensible au flanc descendant de *my_turn* (la variable *my_turn* est celle qui détermine quel afficheur est concerné par le set du chiffre tapé par l'utilisateur).



2. Fréquences

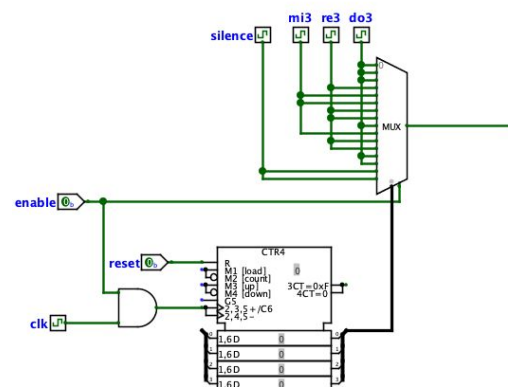
Pour les fonctions additionnelles, nous avons rapidement pensé à un module *music* qui jouerait des musiques de 4 différents genres suivant la lettre (A, B, C ou D) rentrée par l'utilisateur. Pour créer des notes de musique, il fallait diviser la fréquence de notre carte (256 kHz) par deux fois la fréquence souhaitée pour obtenir le nombre de tics de la clock en question. Par exemple, le mi3 a une fréquence de 330Hz, et $256\ 000/(2 \times 330) = 388$.

formule du nombre de tics et propriétés du mi3 :

$$N_{tics} = \frac{f_{carte}}{2f_{voulue}}$$

Properties	Registers
Sélection: Horloge	
VHDL	Verilog
Orientation	Sud
Durée haute	388 Tics
Durée basse	388 Tics
Label	mi3
Position du ...	Nord
Police du La...	SansSerif Gr...

circuit de *Au Clair de la lune* :



De même pour faire varier la rapidité d'incréméntation entre l'horloge et le chronomètre par exemple.