Systèmes Logiques

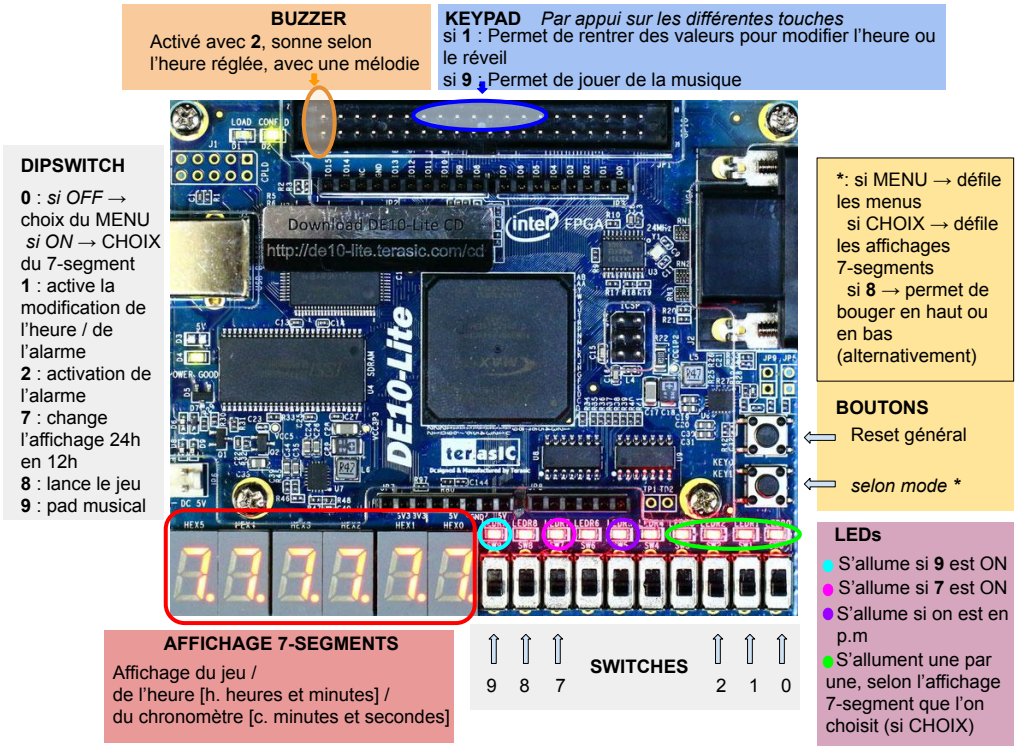
Projet de fin de semestre : Montre digitale

Présentation générale

Pour ce projet de fin de semestre, nous avons dû réaliser une montre analogique à l'aide du logiciel Logisim qui fonctionne sur une carte FPGA (DE-10 lite). En plus de la fonction horloge de base, nous avons décidé d'y ajouter un chronomètre, un mode d'affichage 12 ou 24h, un réveil avec musique et un jeu. Nous avons intégré le Keypad à notre projet pour régler l'heure de la montre, mais aussi dans une fonction additionnelle de Pad musical. Un bouton permet à l'utilisateur de naviguer entre les différents modes de la montre, nous avons ajouté des messages défilants pour chacun d'entre eux, sauf le pad musical et l'affichage 12h, qui sont chacun mis en avant par une LED.
Ces six modes sont en fonctionnement en parallèle, ce qui nous permet de garder le chronomètre actif et retourner à l'affichage de l'heure, ou encore de jouer de la musique en tout temps, une fois le mode activé.

Mode d'emploi

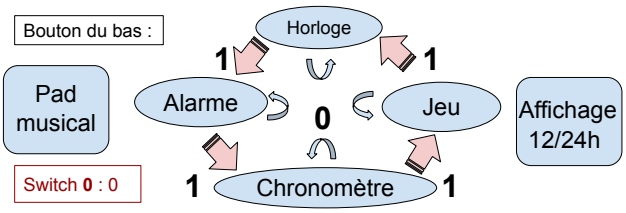
Quand le switch **0** est OFF, on peut changer de mode en appuyant sur le bouton du bas, le nom du mode défile puis l'affichage correspondant apparaît sur les 7-segments.
Pour régler l'heure ou l'alarme, il faut être dans le menu correspondant, choisir le segment que l'on souhaite changer (switch **0**, puis bouton), puis rentrer la nouvelle valeur (Keypad). Si l'utilisateur rentre une valeur impossible, nous la faisons passer à la plus grande possible.
La fonctionnalité de base est l'horloge, qui se lance par défaut à 00h00.



Solutions techniques

Machine d'états générale

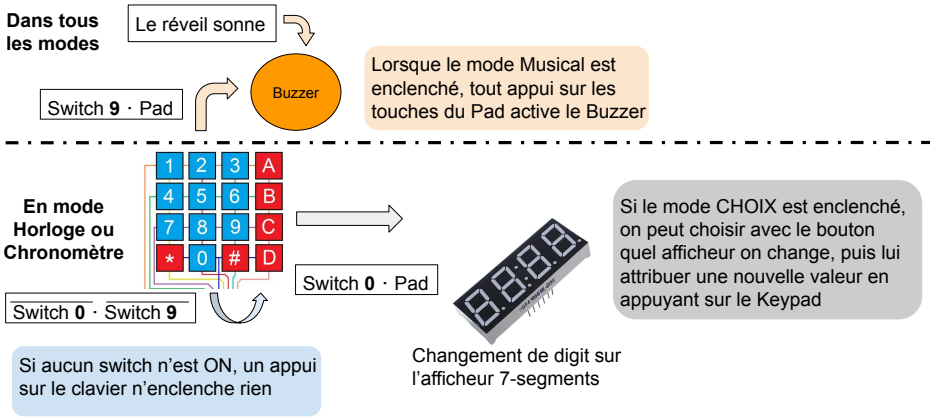
Notre machine d'états générale permet de changer de mode en appuyant sur le bouton du bas selon un cycle défini illustré ci-dessous, sachant que l'affichage de départ est l'horloge.
Le Pad musical et le mode 12h sont liés au Dipswitch et peuvent être activés en parallèle de n'importe quel autre mode.



Machines d'états secondaires

Carte périphérique

Pour utiliser la carte périphérique, nous avons mis en place une machine d'états illustrée ci-dessous. Il faut activer la modification (switch **1**) puis choisir quelle valeur on souhaite modifier (bouton). Nous utilisons ensuite le Keypad pour entrer la nouvelle valeur à l'horloge ou l'alarme. Si l'utilisateur rentre une valeur impossible, nous la faisons passer à la plus grande possible (28h deviens 18h si l'utilisateur a rentré le 2, mais 24h s'il a rentré le 8, équivalent en format 12h). Finalement, le buzzer est activé pour signaler que le réveil sonne, ou quand le pad est utilisé en mode musical.



Jeu

Cette machine d'état permet de lancer l'affichage du jeu, et à l'utilisateur d'y jouer. Elle est assez complexe, puisqu'une fois lancé, l'affichage change continuellement : à chaque flanc montant de l'horloge en entré avec une composante "aléatoire", mais aussi à chaque appui sur le bouton du bas. On peut lancer le jeu depuis le menu correspondant en enclenchant le switch **8**. Des carrés situé en haut ou en bas vont défiler de la droite vers la gauche à chaque tic d'horloge. Pendant cet état actif de jeu, l'utilisateur peut modifier la position de son "personnage" dans le but d'éviter les obstacles. Chaque appui change sa position (en haut ou en bas). S'il entre en collision avec l'obstacle, on passe dans l'état Game Over, qui réinitialise le jeu avec la position par défaut du personnage (en bas) et aucun obstacle.

Jeu actif Etat initial

Problèmes de développement et résolutions

Pendant la réalisation de ce projet, nous avons fait face à de nombreux problèmes, en particulier liés à l'utilisation de la carte DE-10 lite. En effet, celle ci ne prenant pas en compte les aléas des délais, la compilation faisait apparaître des problèmes de fonctionnement alors que le logiciel ne montrait aucun dysfonctionnement.

Une fois le projet exporté sur la carte, nous avons rencontré des problèmes avec les boutons dont les rebonds n'étaient pas pris en compte par Logisim et entraînaient de nombreux bugs sur la carte. Nous avons ainsi ajouté une fonction qui prend en compte le délais d'appui sur le bouton et évite les rebonds (voir Figure 1)

Nos modules les plus complexes nous ont aussi donné du fil à retordre, comme le module Jeu ou les messages défilants. En effet, il était difficile de gérer la coordination des éléments. Nous avons pu régler ceci à l'aide d'étages de compteurs décalés par le biais d'associations d'un différents nombres de D-Flip Flops (voir Figure 2). Particulièrement, dans le module Jeu, pour créer un aspect "aléatoire", nous avons mis en place différents "générateurs" du monde extérieurs : des obstacles (haut ou bas) ou des points (correspondant à aucun obstacle). Ceux ci les génèrent chacun à leur tour, tout en évitant de créer un chemin impossible, c'est-à-dire un obstacle en bas suivant directement un obstacle en haut ou inversement (voir Figure 3).

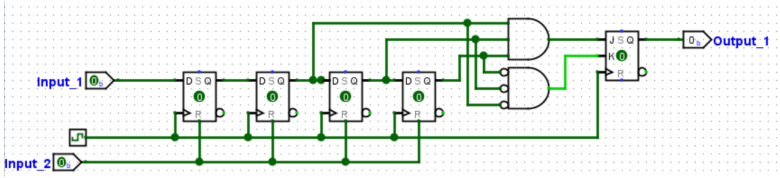


Figure 1: Fonction anti-rebonds

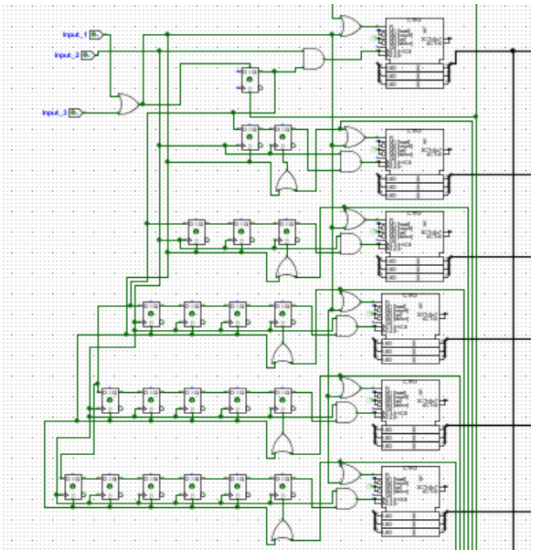


Figure 2: Fonction de défilement d'affichage

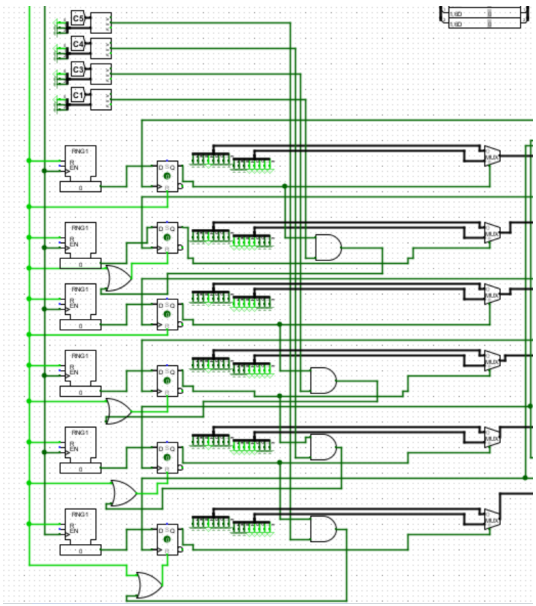


Figure 3: Générateurs aléatoires & gestion des chemins possibles