

Rapport de projet

Introduction et description générale

Ce projet de fin de semestre du cours Systèmes Logiques nous a amené à réaliser les circuits logiques d'une montre à affichage numérique incluant de plus les fonctionnalités suivantes : un réveil matin capable de distinguer les jours de la semaine, un chronomètre avec gestion des tours, ainsi qu'un minuteur (compte à rebours).

Le circuit logique est adapté pour la carte *Intel Terasic DE10-Lite* et utilise six afficheurs sept segments, sept diodes lumineuses, deux boutons afin de naviguer entre les modules, ainsi qu'un des interrupteurs pour allumer et éteindre l'ensemble. Une des broches sort également un signal électrique dans le but d'alimenter un buzzer externe pour fournir le son des alarmes.

Les deux boutons différencient la pression courte, d'environ un à deux dixièmes de seconde, et la pression maintenue, reconnue à partir de 0.8 seconde lorsque la pression est relâchée. Cela fournit de fait quatre boutons logiques.

Instructions pour l'installation

Téléverser le fichier *Projet.circ* sur la carte à l'aide de Quartus Prime Lite, via la fenêtre FPGA Commander de Logisim-Evolution, en sélectionnant pour la fréquence de tics de l'horloge 2 kHz (2048 Hz).

Mode d'emploi de l'utilisateur

Pour rendre l'installation fonctionnelle, la première chose à faire est d'activer l'interrupteur à l'extrême droite sur la carte, suite à quoi, les interrupteurs n'auront plus à être touchés.

De manière générale, lorsqu'aucune heure ou durée n'est en train d'être saisie par l'utilisateur, appuyer courtement sur le premier bouton passe à l'état suivant.

Le premier état à s'afficher après l'allumage est le module *heure*, affichant l'heure courante si celle-ci a été réglée. Pour régler l'heure, pressez longuement le premier bouton. Lorsque la pression est relâchée, les deux afficheurs correspondant aux heures se mettent à clignoter pour signifier qu'ils sont sélectionnés. Une pression courte sur le premier bouton incrémente alors les valeurs sélectionnées tandis qu'une pression courte sur le second bouton conduit à sélectionner l'afficheur des dizaines de minutes, puis des minutes. Pour valider et revenir à l'affichage de l'heure, pressez à nouveau longuement le premier bouton. Il est également possible de consulter le jour de la semaine courant en pressant longuement sur le second bouton et à nouveau pour revenir.

L'état suivant est le *réveil*. Cet état permet d'enregistrer une heure de réveil par jour de la semaine. Le réveil des lundis s'affiche en premier, le nombre compris dans le premier afficheur correspondant au jour de la semaine. Appuyez courtement sur le second bouton pour naviguer aux jours de la semaine suivants. Pour modifier une heure, pressez le premier bouton longuement ; la démarche est identique à celle de l'état *heure*. De plus, appuyer longuement sur le second bouton permet d'activer ou de désactiver l'alarme du jour. Afin de consulter quelles alarmes sont actives, référez-vous aux sept premières LEDs de la carte : la première correspondant aux lundis, et ainsi de suite. Ces voyants seront aussi visibles depuis l'état *heure*.

Le troisième état est le *chronomètre*. Dans ce module, le second bouton sollicité courtement permet de démarrer/suspendre/reprendre la mesure du temps. Pendant que le chronomètre tourne, un appui court sur le premier bouton permet de enregistrer des tours (jusqu'à sept, au maximum). Pour chaque tour enregistré, une LED s'allume. Lorsque le chronomètre est en pause, un appui prolongé sur le premier bouton permet de visionner les tours enregistrés (la LED correspondante clignote alors) et un appui court sur le premier bouton permet de visionner la valeur suivante. Un autre appui prolongé sur le premier bouton fait revenir au chronomètre en pause. Appuyer de manière prolongée sur le second bouton réinitialise le module.

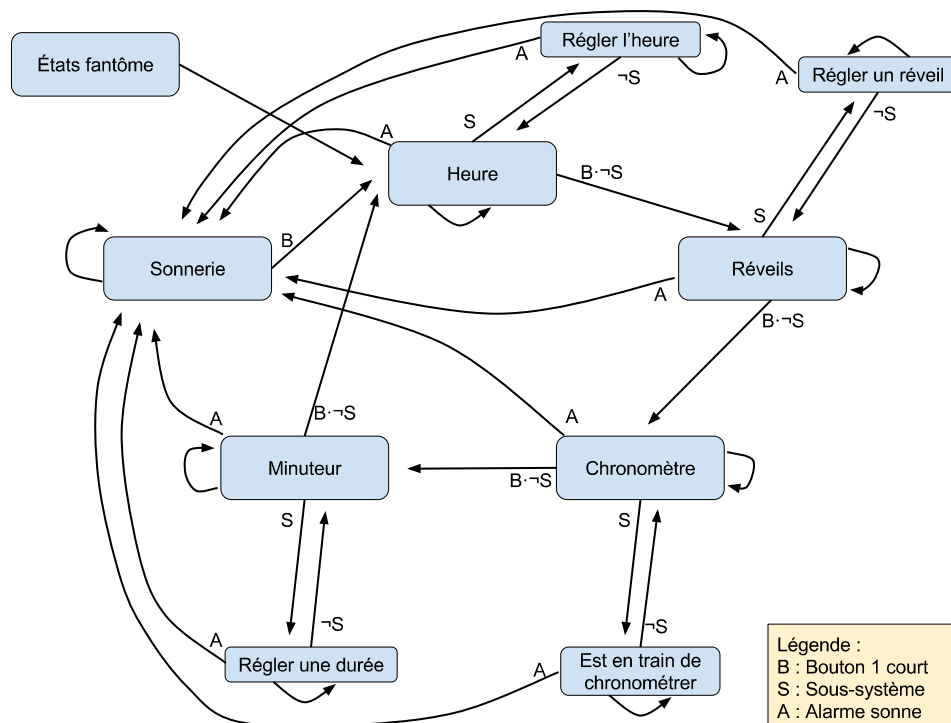
Indication : le chronomètre affiche les dixièmes de seconde tant que l'espace le permet : dès que les heures sont utilisées, les dixièmes ne sont plus affichés. Observez de manière générale que sur les afficheurs, les points au bas des nombres jouent le rôle de séparateurs.

Le dernier état est le *minuteur*. Ce module requiert que l'utilisateur entre une durée qui sera comptée à rebours. Cela se fait de la même manière que pour changer l'heure dans le module *heure*. Après avoir validé et être revenu à l'affichage du minuteur, une première pression courte sur le second bouton permet de démarrer le décompte. Une seconde pression annule le décompte. La sonnerie se déclenche à la fin du temps imparti.

Sonnerie : Cet état supplémentaire montre le texte « sonne » sur l'affichage et permet soit d'éteindre la sonnerie et de retourner au premier état (premier bouton), soit de mettre en place un rappel (anglais : snooze) cinq minutes plus tard (second bouton). Dans cette seconde possibilité, le texte « rappel » s'affiche et l'alarme est reportée.

Description de la machine d'états finis générale

La machine d'états finis générale se compose principalement des cinq états décrits dans le mode d'emploi : quatre pour les fonctionnalités, et une pour la sonnerie. Un bouton (B) permet de cycloser parmi les quatre états principaux. Afin que l'utilisateur puisse procéder à des réglages (régler l'heure, par exemple), ces quatre états disposent chacun d'un sous-système (S) que l'on peut considérer comme une machine d'états finis sous-jacente à deux états chacun (booléen), mais comme la machine d'état finis générale prend en compte ces sous-états, on considérera plutôt une unique machine d'états. Chacun des états mène vers l'état *sonnerie* lorsqu'une alarme (A) retentit. Le schéma de la machine peut être consulté ci-dessous :



Remarque : si la sonnerie se déclenche alors qu'un sous-état est actif, celui-ci le restera en arrière-plan. Ainsi par exemple, le chronomètre peut continuer à chronométrer s'il était actif lorsque l'alarme s'est déclenchée.

Bien entendu, le projet comprend également d'autres machines d'états finis plus spécifiques, comme par exemple pour régler une heure en sélectionnant successivement les heures, puis les dizaines de minutes, et finalement les minutes. Une autre machine d'état finis intéressante est celle des modules alarme et chronomètre respectivement, permettant de cycloser en lecture ou en écriture parmi les sept registres enregistrant respectivement une alarme ou un tour du chronomètre.

Solutions techniques apportées

Un des points essentiels de notre approche pour la conception d'une montre numérique est que nous gérons le temps non pas en enregistrant séparément les heures, minutes et secondes du temps courant, mais en stockant uniquement le nombre dixièmes de secondes écoulés depuis le début de la semaine courante, sur 23 bits. Cela apporte l'avantage de ne pas avoir à se préoccuper de la retenue lors d'additions ou de soustractions de durées. Néanmoins, cela nous a contraints à créer notre propre module diviseur car le module fourni par le logiciel Logisim-Evolution n'est pas compatible avec la carte.

Afin d'éviter que registres et mémoires flipflop ne démarrent avec des valeurs imprévisibles, nous avons introduit un interrupteur. Jusqu'à ce que celui-ci soit commuté vers le haut, toutes ces mémoires se font réinitialiser. La logique de notre circuit est ainsi déterministe.

Problèmes rencontrés lors du développements

Lors du développement, nous n'avons rencontré que très peu de problèmes. Néanmoins, Logisim-Evolution nous a joué des tours car le module diviseur que nous avons créé, bien que complètement fonctionnel sur la carte, lui fait faire trop de calculs par pas de la simulation. Or c'est justement là le critère que choisit ce logiciel pour détecter les phénomènes d'oscillations ; la simulation s'arrête alors et une erreur – qui ne devrait pas apparaître – apparaît. Cependant, en remplaçant notre diviseur par celui de Logisim, notre projet est simulé sans problèmes, si bien que nous fournissons deux fichiers .circ : un pour la consultation dans Logisim, et un pour mettre sur la carte.

De plus, comme nous avons besoin de deux fréquences d'horloge très différentes, une pour faire vibrer le buzzer, et l'autre pour le reste du projet, nous avons dû convertir un signal 2048 Hz en 10 Hz afin de disposer des deux. Le module qui se charge de cela ralentit la simulation, si bien que nous ne le connectons pas non plus sur le fichier destiné à la simulation dans Logisim. Il sera néanmoins visualisable.

Notez que les deux fichiers .circ que nous fournissons ne diffèrent qu'en ces deux points, et hormis ceux-ci, ils se veulent identiques.