

## MONTRE DIGITALE

Nous avons créé une montre digitale avec plusieurs fonctions. La fonction principale est d'afficher le jour, l'heure, et la minute. Il permet aussi de changer l'heure et la date, et inclut un chronomètre, un minuteur, un changement de fuseau horaire (Washington DC, Wellington, Port Louis, Brasilia, Avarua District), un message défilant, une calculatrice, 3 chansons (Zelda, Ode to Joy et Harry Potter), et une alarme. Elle fonctionne avec l'utilisation des deux boutons et le commutateur DIP.

### MODE D'EMPLOI

Pour parcourir les modes de fonctionnement, on utilise toujours le bouton A (celui du haut). Le bouton B (celui du bas) sert à sélectionner le mode que l'on désire et, à la fin d'un sous-menu, sert à revenir à l'heure actuelle. On active l'option LONGPRESS d'un bouton en le maintenant enfoncé pour plus d'une seconde. À la première utilisation, il est impératif de réinitialiser la montre en appuyant sur LONGPRESS B. On peut à tout moment effectuer un reset total de la montre avec un LONGPRESS B.

Horloge : Ce mode affiche tout simplement l'heure.

Changer l'heure: Dans le mode nommée "set", l'heure actuelle sera affichée. Il faut utiliser le commutateur DIP pour sélectionner le chiffre qui sera changé. Pour changer le chiffre, il faut maintenir le bouton A enfoncé. C'est un changement asynchrone avec l'horloge.

Chronomètre: Dans le mode nommé "chrono", les minutes, secondes, et déca secondes commencent tous par 0. Pour le démarrer, il faut appuyer sur le bouton A, puis pour l'arrêter, réutiliser le même bouton. Pour recommencer à 0, il faut faire un LONGPRESS A.

Minuteur: À l'intérieur de "Timer", l'affichage clignote tant que l'utilisateur n'a pas choisi le temps désiré en utilisant les DIP-switches et le bouton A. On valide avec B et pour débiter le compte à rebours on appuie sur A. Un autre appui du bouton A arrêtera le timer. Un LONGPRESS A recommencera le minuteur à la valeur entrée précédemment. Le buzzer s'active quand le compte est terminé et pour l'arrêter, on peut soit effectuer un LONGPRESS A ou activer le DIP-switch 9.

Fuseaux horaires: Dans le mode "zones", on peut parcourir les capitales données avec le bouton A, et sélectionner capitale voulue avec le bouton B. L'heure de ce fuseau horaire est maintenant affiché dans le mode "horloge"

Alarme: Dans le mode "Alarm", l'utilisateur peut utiliser les DIP-switches et le bouton A pour choisir l'heure désirée. On valide l'heure en appuyant sur B. Ensuite, l'utilisateur est invité à choisir la mélodie désirée en utilisant le bouton A pour faire défiler les trois chansons et le bouton B pour valider. Finalement l'alarme peut être activé en activant le DIP-switch 10. Quand l'alarme sonne, on l'arrête avec le DIP-Switch 9

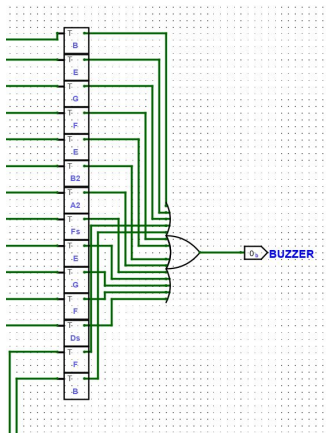
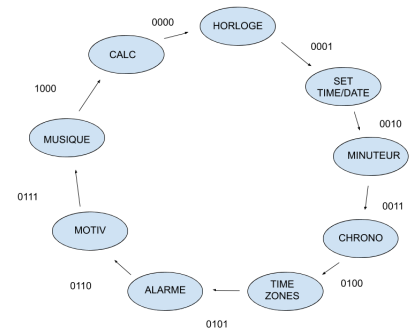
Message: En sélectionnant le mode "Motiv", le message "You can do it" va défiler à l'écran en boucle.

Musique: Dans le mode "Music", on peut parcourir les trois chansons disponibles avec le bouton A.

Calculatrice: Dans le mode “Calc”, l'utilisateur peut utiliser les DIP-switches et le bouton A pour choisir le premier numéro désiré. Le 4ème interrupteur sert à inclure un signe négatif pour le premier numéro. Le bouton B entre dans le menu d'opérateurs: “add”, “sub”, “mul”, et “div” correspondant à l'additionneur/soustracteur de 8 bits (résultat de -255 à 255), le multiplieur de 4 bits (résultat de -255 à 255), et le diviseur de 4 bits (quotient de -15 à 15, reste de 0 à 15). L'utilisateur pourra les parcourir avec le bouton A, sélectionner avec le bouton B, et choisir le deuxième numéro de la même manière que le premier, avec le 5ème DIP pour ajouter un négatif. La réponse du calcul s'affiche en appuyant sur B.

### MACHINE D'ÉTATS FINIS GENERALE

La machine d'états finis générale se trouve dans le sous-circuit FSM\_Final. Elle fonctionne à l'aide d'un bouton qui ne peut seulement être appuyé qu'une seule fois pour envoyer la valeur de 1 pour la durée d'un tic. Après, le bouton doit être relâché et de nouveau appuyé pour pouvoir envoyer le prochain logique-1. La sortie du bouton est ensuite branché à un additionneur en LSB. De cette manière, à chaque fois qu'un 1 logique est reçu, on augmente la valeur de notre code. La sortie de cette additionneur est branché à l'entrée select d'un démultiplexeur qui décide quel mode est le mode actuel. Les sorties sont aussi branchées aux entrées de l'additionneur pour qu'on puisse l'incrémenter de 0 à 8. Chaque code de 0000 à 1000 correspond à un mode. Des conditions ont été ajoutées pour qu'on puisse passer de 1000 à 0000 en boucle.



### PÉRIPHÉRIQUE

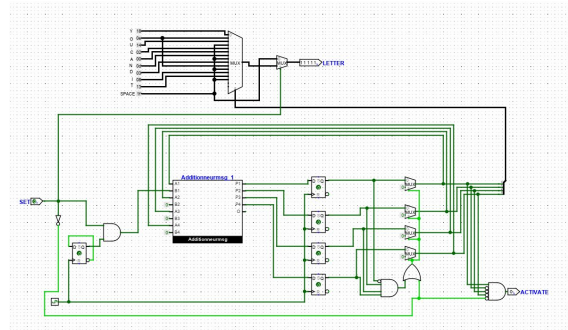
Le buzzer a été implémenté en utilisant 2 blocs principaux: un bloc NoteLength et un bloc Note. On a créé un bloc pour chaque note dans la gamme donnée avec la fréquence appropriée en utilisant un D-Flip-Flop et une horloge. Les tics des horloges ont été réglés en fonction de la fréquence voulue. Nous avons choisi de faire fonctionner notre horloge à 64 kHz pour garder l'erreur sur les fréquences en dessous de 15 hz. La formule utilisée pour calculer les tics hauts/bas est  $\frac{16000}{f}$  avec  $f$  la fréquence désirée.

Chacun de ces blocs ont une entrée enable que lorsqu'elle est active, cette note est jouée par le buzzer. En sortie, ces notes sont branchées à une porte OR qui est branchée au buzzer. En entrée, les blocs Notes sont tous branchés à un bloc NoteLength qui indique la durée de temps d'une note.

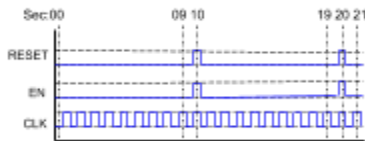
Après qu'une note ai fini de jouer, le bloc NoteLength active le prochain bloc NoteLength de la séquence qui active le “enable” de la prochaine note. À la fin de la partition, un signal est envoyé au bloc “MusicLoop” qui tout d'abord envoie un reset à la mélodie, puis envoie un set pour que la chanson joue en boucle. La chanson se termine une fois qu'on reçoit un signal Reset global qui désactive toutes les notes. Nous avons préféré cette solution à une FSM car nous avons voulu créer trois chansons et nous trouvons que grâce à cette solution, il était plus facile de modifier la partition avec d'autres notes.

### MESSAGE DÉFILANT "YOU CAN DO IT"

Nous avons utilisé une technique similaire à la FSM générale pour créer notre message défilant. On a branché un additionneur à l'entrée SELECT d'un multiplexer. Lorsque le mode "Motiv" n'est pas activé, le MUX du premier 7segment reçoit un 0000 qui est attribué au code pour la lettre Y. Quand on active le mode, l'additionneur est enclenché et l'additionneur additionne 0001 à la valeur du nombre A à chaque 7000 tics. Ce qui fait en sorte que le MUX parcourt ses différentes options, qui sont toutes attribuées à un caractère. Quand l'additionneur est à la deuxième valeur (0001), un signal ACTIVATE est envoyé au prochain 7 segment qui commence le même processus.



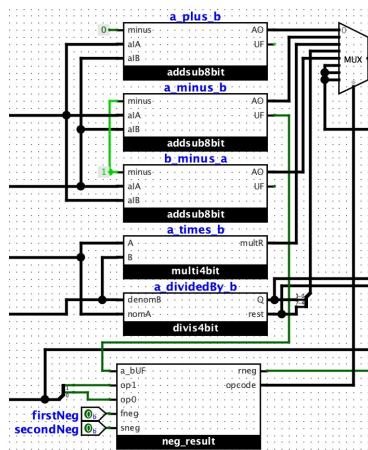
### PROBLÈMES RENCONTRÉS DANS DEVELOPPEMENT → SOLUTIONS TECHNIQUES



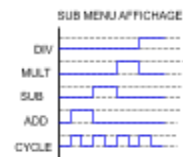
Un des problèmes rencontrés dans le développement était la représentation de nombres par rapport à l'utilisateur et le circuit. L'utilisateur lit des nombres en termes de chiffres de 0-9 rassemblés, qui se traduisent alors en de valeurs 4 bit pour chacun. La valeur maximale de 4 bit est 15, alors qu'on veut s'arrêter à 9, puis augmenter le prochain chiffre de 1. Le schéma temporel à gauche montre comment

la machine d'état parvient à compter les secondes, avec l'output EN qui augmente la valeur grande des secondes de 1, CLK qui augmente la valeur petite des secondes de 1, et RESET qui remet la valeur petite des secondes à 0. On a pu créer les machines nécessaires pour l'horloge, la minuterie, et le chronomètre, simplement en changeant les conditions pour les outputs EN et RESET pour chaque valeur.

Le circuit qui manipule les fuseaux horaires utilise le même système de lecture de nombres, en ajoutant 4 bit à la fois. Par exemple, si on veut augmenter 14h de 8 heures, on va d'abord ajouter 8 et 4, et ajouter 0 et 1. Puisque  $8+4$  est plus grand que 9, il a fallu un sous-circuit qui filtre les valeurs de 4 bit entre 10 et 15, qui va donc soustraire 9, et donner un surplus. Elle fonctionne de la même manière dans l'autre direction, avec des changements d'heures négatifs. A la fin, un filtre est ajouté pour vérifier que l'heure ne dépasse pas 23, dans quel cas il faut recommencer l'heure à 0 et ajouter 1 aux jours.



L'image de droite montre un schéma temporel du submenu des opérations proposées à l'utilisateur, qui est une machine d'état utilisé pour plusieurs d'autres sous-menus comme pour les opérateurs de la calculatrice et le choix de musique.



La calculatrice s'est avérée plus compliquée car l'additionneur, soustracteur et multiplicateur manipulaient des valeurs de 8 bit. Il a fallu alors un convertisseur de deux valeurs de 4 bits en une valeur 8 bit, et vice versa. Le circuit va alors faire toutes les calculations ( $a+b$ ,  $a-b$ ,  $b-a$ ,  $a \cdot b$ ,  $a/b$ ) en même temps, et un multiplexeur choisit le résultat dépendant du code d'opérateur et du signe des deux valeurs, comme vu à droite.