**Laurent Tremblay**

*Gr.001*

**Projet final**

*Pédale numérique pour effets de guitare*

Travaille présenté à

**M. Jasmin St-Laurent, M. Claude Barbaud et Mme. Chantal Milette**

dans le cadre du cours

*Projet de fin d’études, 243-631-MA*

**Collège Maisonneuve**

2018-05-11

Table des matières

**Aucune entrée de table des matières n'a été trouvée.**

**Introduction**

Dans le cadre du cours de projet de fin d’études, j’ai réalisé un projet électronique ayant comme fonction de modifié un signal audio provenant d’une guitare à l’aide d’un FPGA. Celui-ci possèderait une interface graphique pour afficher à l’utilisateur l’effet actif ainsi que les paramètres modifiables des effets par des potentiomètres externes. L’utilisateur aurait à sa disposition deux boutons poussoirs pour avancer et reculer entre les multiples effets ainsi qu’un bouton pour activer les effets audios ou non. Parmi ces effets, on peut retrouver un effet de distorsion et un effet de trémolo. L’effet de distorsion consiste à limiter le signal audio pour ajouter des harmoniques à celui-ci, et l’effet de trémolo consiste à moduler le signal audio à l’aide d’une onde de fréquence très basse, ce qui donne l’impression qu’on augmente et réduit automatiquement le volume du son.

J’ai choisi ce projet car je joue de la guitare dans mon temps libre et qu’il est important d’avoir des pédales pour modifier le son de la guitare. Il y a aussi le fait que les pédales à effets pour guitare sont très couteuses, car une pédale peut coûter entre 75 et 500 dollars, et celle-ci ne produit qu’un seul effet. On peut donc s’imaginé qu’on peut se retrouver avec une grosse facture seulement en possédant quelques effets.

L’objectif du projet était donc de pouvoir créer tous les effets voulus pour une fraction du prix que l’ensemble des pédales standard pourraient coutés. De plus, ce projet me donnait raison d’utilisé un FPGA, car ceux-ci sont extrêmement puissant pour le traitement de signaux numériques, et par ce fait approfondir mes connaissances au niveau de la composante et du langage VHDL.

**Bilan des travaux**

\*\* bilan a la fin

**Description générale**

Schéma synoptique

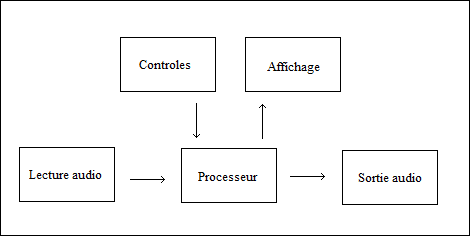
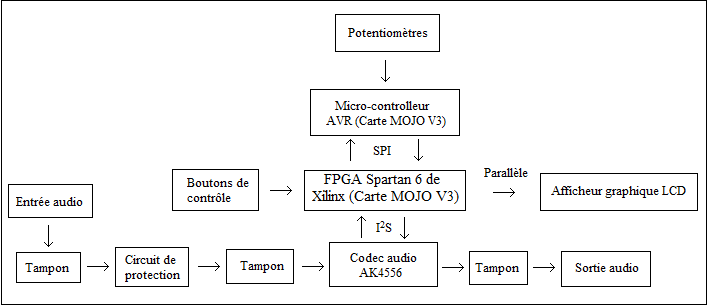


Schéma de principe



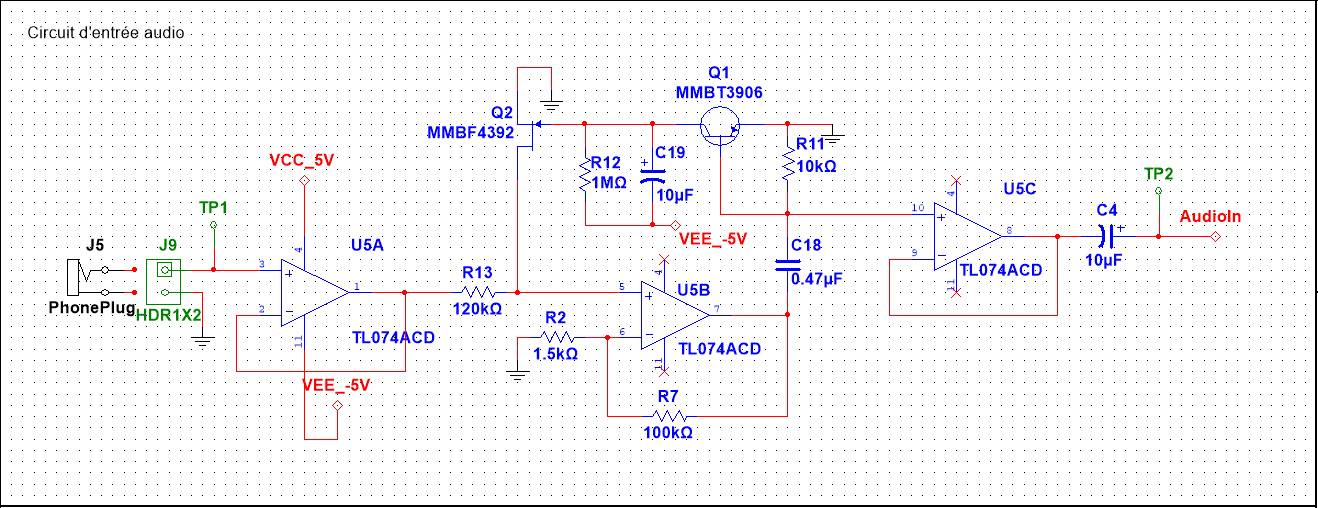
Il y a plusieurs principes utilisés dans ce projet. Ceux-ci peuvent être séparés en deux : soit une partie analogique et une partie numérique. Pour la partie analogique, il y a l’utilisation de l’amplificateur TL04 comme tampon et comme circuit de protection pour transmettre le signal audio au codec audio AK4556. Il est important d’utilisé un circuit tampon pour isoler les impédances, soit celle de l’entrée audio, celle de l’entrée du circuit d’acquisition du codec audio et celle à la sortie du codec audio. De plus, il faut aussi utiliser un circuit de protection pour limiter le niveau de tension de l’entrée audio, car elle pourrait endommager l’entrée du circuit d’acquisition du codec. Dans mon cas, j’ai choisi d’utilisé un circuit de compression audio : ce circuit ajuste automatiquement la tension du signal audio pour qu’elle soit à une tension fixe, prévenant les surtensions. En plus, ce choix permet d’avoir une meilleure résolution de notre signal audio, cependant il vient couper la dynamique du signal audio. Le deuxième aspect de la partie analogique est l’utilisation d’un codec Audio, celui-ci permettant le sur-échantillonnage et le filtrage du signal audio. La communication entre le codec Audio et le FPGA se fait à l’aide du protocole I2S.

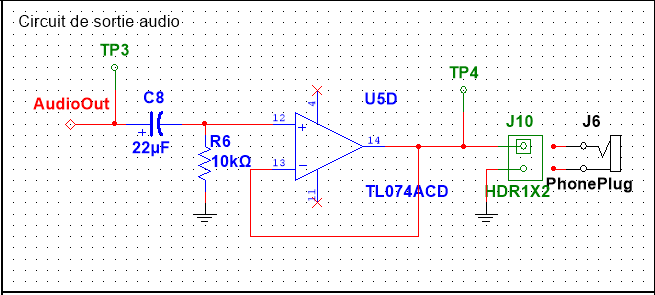
Pour la partie numérique, on peut retrouver une interface pour l’utilisateur ainsi que plusieurs protocoles de communications. L’utilisateur possède trois boutons de contrôles, soit deux pour changer de pages (précédent et prochain) ainsi qu’un bouton activant la pédale. L’utilisateur peut aussi modifier les paramètres des effets à l’aide de trois potentiomètres. Les tensions de ces potentiomètres sont lues à l’aide des convertisseurs analogique à numérique du microcontrôleur AVR de la carte MOJO V3 et sont ensuite transmit via le protocole SPI au FPGA Spartan 6. Le dernier item de l’interface utilisateur est un afficheur graphique LCD. Le FPGA envoi des données en parallèle à l’afficheur étant donné qu’on ne manque pas de broches, que c’est plus simple et que c’est rapide.

**Description précise**

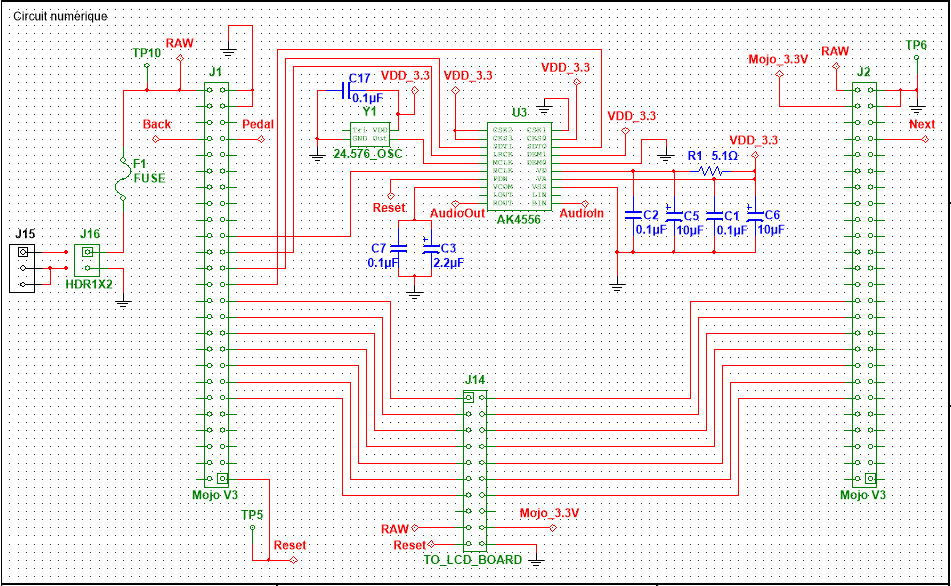
**Schéma de montage**

Plaquette principale

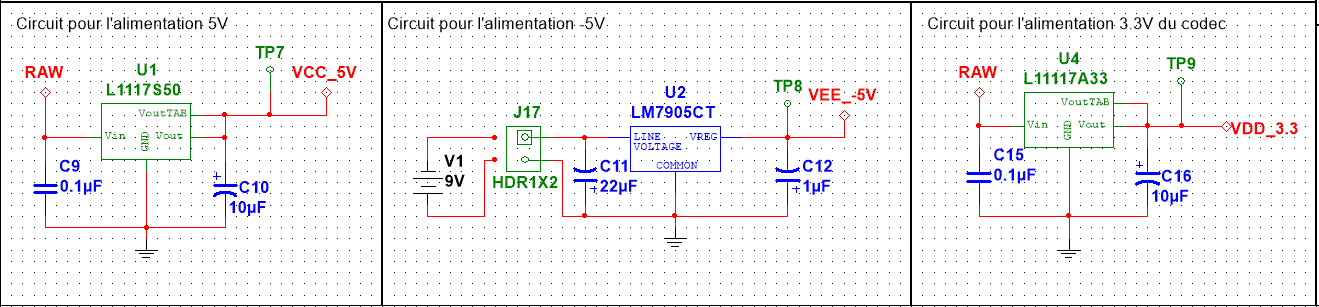
Le circuit d’entrée audio comporte trois étages. Le premier étage est un suiveur qui isole l’entrée l’impédance de la guitare et celle du circuit. Le deuxième étage est un circuit de compression audio, régulant le signal d’entré pour qu’il soit toujours à 1.31 volts. Ce circuit est important pour protéger l’entrée du convertisseur analogique à numérique du codec audio AK4556. De plus, il permet d’utiliser une plus grande résolution du convertisseur analogique à numérique. Le troisième étage est un autre suiveur qui isole l’impédance du circuit de compression et l’impédance du convertisseur analogique à numérique du codec audio. Il possède un condensateur à sa sortie pour couper le décalage courant continue.



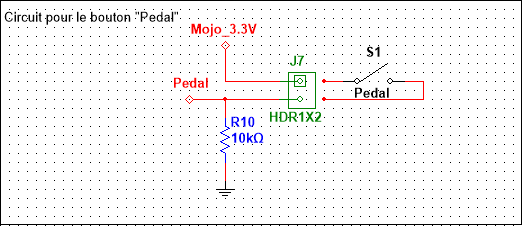
En contrepartie, le circuit audio comporte un seul étage, soit un suiveur qui vient isoler la sortie du convertisseur numérique à analogique du codec audio AK4556. Il possède aussi un filtre passe haut à son entrée pour couper le décalage courant continue.



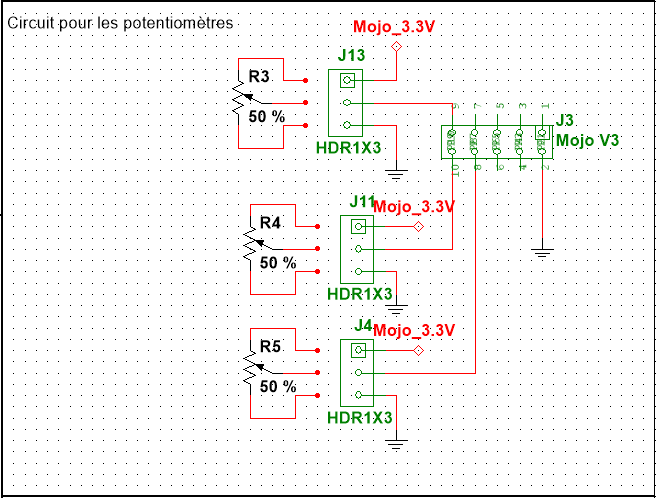
Le circuit principal du projet se retrouve ici. Le connecteur J15 est une entrée pour le bloc d’alimentation 9V. Le connecteur J14 permet le lien entre la plaquette principale et la plaquette pour l’écran graphique. Les connecteurs J1 et J2 font le lien entre la carte Mojo et les circuits du projet. Dans ce circuit, on retrouve le codec AK4556, qui lui est responsable pour l’échantillonnage du signal audio.



La plaquette utilise trois alimentations, soit 5 volts et -5 volts pour l’alimentation de l’amplificateur opérationnel TL074 utilisé dans les circuits audios, ainsi que 3.3V pour alimenter le codec audio. Il est important de noter que la carte Mojo V3 utilise l’alimentation fourni à la carte pour produire ses propres tensions.



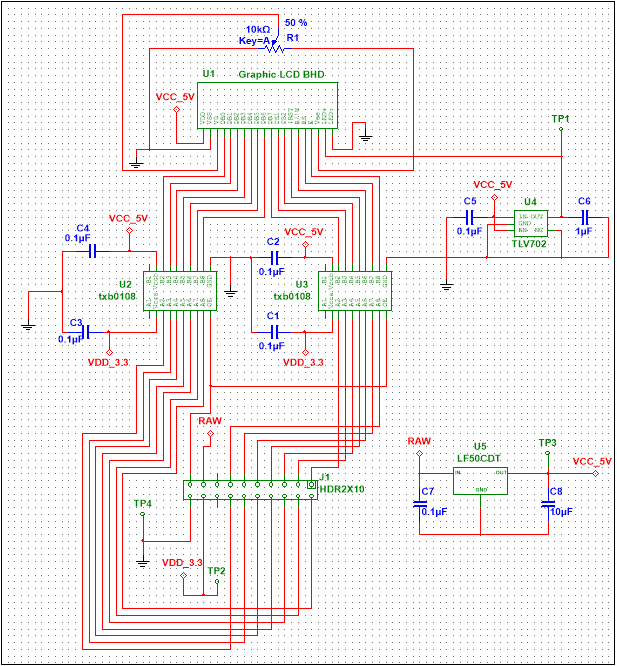
Le même circuit est utilisé pour le les boutons « Pédale », « Suivant » et « Précédent ».



Ce circuit est utilisé pour pouvoir lire la tension des potentiomètres utilisés pour modifier les paramètres des effets. Ceux-ci sont connectés au connecteur J3, qui lui est directement lié aux convertisseurs analogiques à numériques du microcontrôleur Atmel de la carte Mojo.

\*\*Le circuit principal complet peut être trouvé à l’annexe.

Circuit pour l’écran graphique

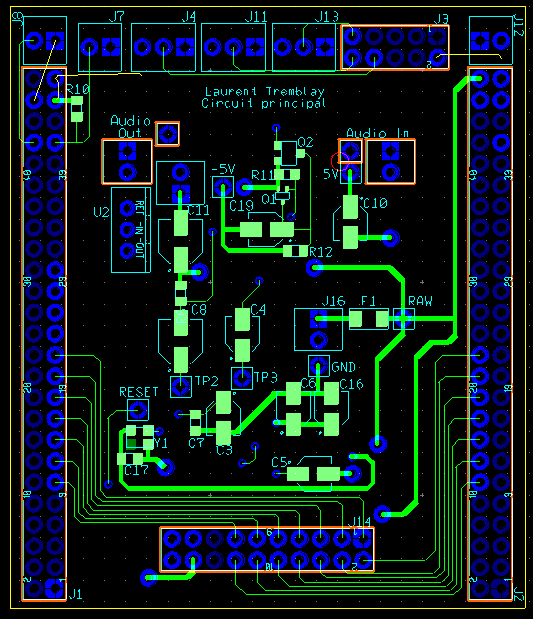


Ce circuit est le circuit complet pour interfacer l’afficheur LCD avec le FPGA. Le connecteur J1 permet la liaison entre les deux plaquettes. Le circuit possède deux régulateurs : un LF50CDT régule la tension à 5V, soit l’alimentation nécessaire pour la logique de l’écran, et un TLV02 qui régule la tension à 3.5V, soit l’alimentation nécessaire pour alimenter le rétro-éclairage de l’écran. Le circuit possède aussi deux TXB0108, qui eux convertissent la tension provenant des tampons de sortie du FPGA, soit 3.3V, à la tension nécessaire pour la logique de l’écran, soit 5V.

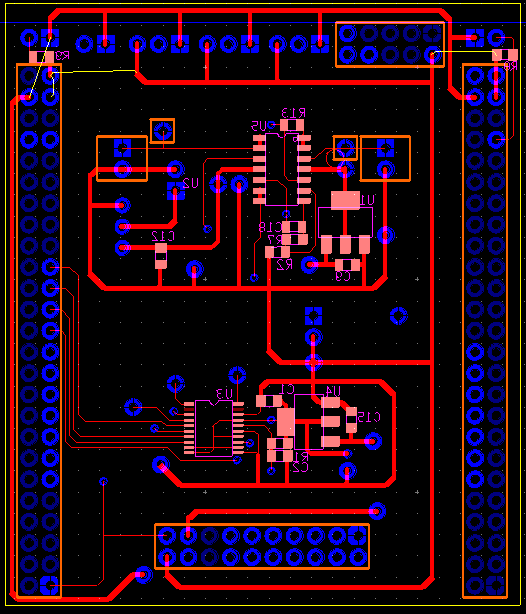
**Circuit imprimé**

Circuit imprimé pour le circuit principal

*Copper Top*

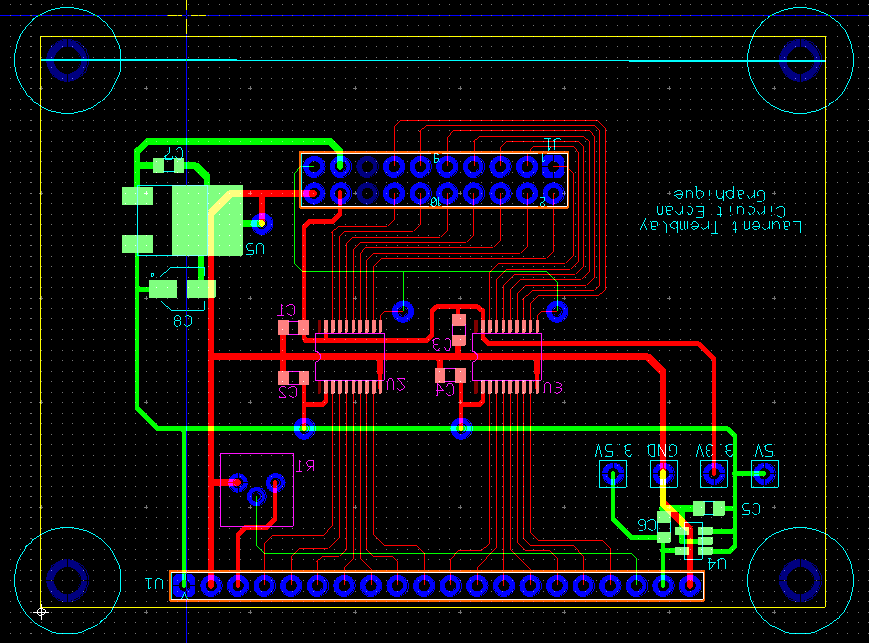


*Copper Bottom*



Les images ci-hauts sont les côtés *top* et *bottom* du circuit principal. Les connections non reliés en jaune sont effectués lorsque le circuit est inséré dans la carte Mojo V3. De plus, l’erreur indiqué par le point de test de l’entrée audio et de l’alimentation 5V étant trop rapproché n’est pas réellement une erreur étant donné que ceux-ci sont à 100 millièmes de pouce de distance.

Circuit imprimé pour l’écran LCD



Ce circuit imprimé est utilisé pour alimenter l’écran graphique et pour transformer la tension 3.3V provenant du processeur (dans notre cas le FPGA) pour qu’il soit converti à 5V. La ligne bleue vers le bas de la plaquette indique où il faut couper pour enlever l’excès.

**Câbles**

Ribbon cable -> 20, AWG,

Crimped cables -> AWG, lenght

**Programmes**

**Boîtier**

**Caractéristiques et limites de l’appareil**

Consommation d’énergie : Alimentation positive : 1,35 Watts (9v x 150mA)

Alimentation négative : 0,045 Watts (9v x 5mA)

Autonomie : 100 heures avant de changer la batterie (~550mAh /5mA)

Type d’alimentation : Murale et batterie 9V

Milieu :

Température :

Plage de fonctionnement :

Vitesse d’acquisition du signal audio : 48kHz

**Liste de matériel**

Partie 1



Partie 2



\*\* Ajouter : Régulateur 5V DPAK pour écran LCD

**Évaluation des coûts de production**

Les coûts de production se séparent en quatre catégories : le processeur, les composantes passives, les composantes actives et les acessoires.

**Procédure d’assemblage**

Souder composantes,

Percer boitier,

Crimp les fils,

Cable ruban,

Souder les fils aux périphériques externes,

Installer la carte mère dans le Mojo V3,

Mettre tous les connecteurs, dans les sockets

Installer le programme

**Procédure de calibration**

La seule calibration nécessaire pour le projet est d’ajuster le potentiomètre « R1 », sur la carte de l’écran LCD, pour ajuster le rétro-éclairage de l’écran et ainsi avoir la meilleure image.

**Manuel de l’utilisateur**

**Conclusion**

Évaluation du choix matériel

S’il était question de recommencer le même projet, je crois qu’il serait plus intéressant d’utiliser un processeur adapté au traitement des signaux numériques à la place d’un FPGA. Bien sûr, le FPGA permet d’avoir une latence très faible entre l’entrée et la sortie du signal audio grâce à la parallélisation des calculs, cependant, un processeur adapté pour le traitement de signaux numérique permet d’obtenir des résultats presque équivalents beaucoup facilement. En effet, étant donné que ceux-ci sont conçu dans l’optique de traiter les signaux, ils possèdent plusieurs périphériques internes permettant d’effectuer des calculs intenses comme des transformations de fourrier et des divisions très rapidement. Ces périphériques sont déjà optimisés et très performant, ce qui qui un avantage comparé au FPGA, où il faut tout concevoir de A à Z, et où le temps d’optimisation est plus important que le temps de développement. De plus, le FPGA a besoin de plusieurs périphériques pour fonctionner, comme de la mémoire, un programmeur, etc. Comparer au processeur de signaux numériques, qui lui peut fonctionner seul, où avec quelques petites composantes externes comme des oscillateurs. Toutes ces composantes viennent augmenter la taille du projet, et étant donné que les pédales vendues sur le marché sont relativement petites, il serait beaucoup plus intéressant d’utiliser un processeur de signaux numériques pour réduire la taille de la plaquette.

Modifications éventuelles à apporter

Pour améliorer le projet, il y a plusieurs modifications à apporter. Premièrement, il faudrait utiliser un écran LCD plus petit et adapté aux tensions du processeur. En effet, l’écran prend énormément de place, et à cause de cela il faut lui faire sa propre plaquette pour pouvoir faire les connections entre ses broches espacées. De plus, il requiert une tension de 5V pour les niveaux logiques hauts, ce que le processeur ne peut pas fournir. Tous ces facteurs viennent grandement augmenter le coût de production, alors comme amélioration, il serait question de trouver un écran ayant un connecteur intégré qui fonctionne à 3.3V, ou la tension des sorties du processeur. Le connecteur permettrait de connecter l’écran directement au circuit principal, et étant donné que l’on n’aura pas besoin d’adapteurs de tensions entre le processeur et l’écran, il sera possible de les connecter directement. Deuxièmement, le circuit de compression audio devrait être améliorer. En effet, ce circuit devrait être modifié pour qu’il utilise la plage entière du convertisseur 24 bits du codec audio, et il devrait avoir un temps de réponse plus rapide. De plus, celui-ci semble avoir un problème avec les tensions trop hautes, alors il serait question d’ajouter un circuit de pré-amplification pour automatiquement régler la tension provenant de la guitare à un niveau relativement faible pour que le compresseur fonctionne correctement. On pourrait aussi choisir d’améliorer les composantes utilisées pour générer le moins de bruit possible. Dernièrement, une amélioration très intéressante serait d’ajouter un circuit qui permettrais d’avoir une alimentation bipolaire à l’aide d’une alimentation unipolaire. Pour obtenir le moins de bruits possible, il est important d’utiliser un amplificateur nécessitant une alimentation bipolaire, ce qui fait en sorte qu’on se retrouve avec deux alimentations à la place d’une. Pour éliminer cela, il faudrait ajouter un circuit, soit un convertisseur courant continue à courant continue (DC-DC) pour obtenir une tension négative à l’aide de notre tension positive.

Difficultés rencontrées

Durant la conception du projet, je suis tombé face à face avec plusieurs problèmes dont j’ai heureusement réussi à surmonter. Premièrement, je me suis rendu compte assez rapidement qu’on ne pouvait pas simplement générer une tension négative en inversant les broches d’un régulateur. En effet, pour que cela fonctionne, il faut utiliser un convertisseur CC-CC, soit un détail que je n’avais aucunement prévu. Comme solution rapide, j’ai dû ajouter une batterie 9V ainsi qu’un régulateur négatif pour obtenir une tension négative régulé pour le circuit analogique nécessitant une alimentation bipolaire. Deuxièmement, lorsque je testais mes effets, je me suis rendu compte qu’il y avait des pics de tensions aléatoires qui affectait le signal. Ce problème venait du fait que le contrôleur ATMEL de la carte Mojo V3 échantillonne beaucoup trop vite, et par ce fait ne laisse pas assez de temps à ses convertisseurs analogues à numériques pour se décharger, ce qui fait en sorte qu’ils s’influencent tous. Troisièmement, je n’avais pas prévu de circuits de protection pour l’entrée du convertisseur analogique à numérique du codec audio. Lors de mes tests, je me suis rendu compte que le signal provenant de la guitare dépassait la tension maximale du convertisseur analogique à numérique du codec audio, et il fallait donc que je trouve une solution rapide et efficace. Celle-ci était d’implémenté un circuit de compression audio, qui celui-ci s’occupe de normaliser le signal à une tension fixe. Le circuit fonctionne bien pour protéger, cependant il n’est pas parfait et vient causer d’autres problèmes non voulus si la tension est trop élevée, ce qui le rend sujet à une future amélioration. Quatrièmement, j’ai eu de la difficulté à implémenté un filtre FIR pour le paramètre « Tone » de l’effet de distorsion. Pour créer ce paramètre, il fallait que j’entre 23 fois 15 coéficients flottants différents pour effectivement filtrer le signal, ce qui demande beaucoups de temps étant donné tous les autres problèmes survenus qui ont ralenti le projet. Finalement, lors de la conception de mes circuits imprimés, je me suis rendu compte que les connecteurs de l’écran graphique LCD était trop large pour simplement mettre y attacher un connecteur et que la tension du niveau logique haut était plus élevée que celle de mon FPGA. Cela a fait en sorte que je me suis retrouvé à devoir créer une deuxième plaquette spécialement pour l’écran, ainsi que d’imaginer une façon de connecter cette plaquette avec la plaquette principale.

Connaissances acquises pendant le projet

Durant la conception du projet, j’ai acquis de nombreuses connaissances par rapport à plusieurs aspects. Premièrement j’ai beaucoup appris sur le langage VHDL ainsi que sur les FPGA. Il y avait énormément de connaissances qui n’avaient pas été montré dans les cours que nous avions eu, comme l’utilisation des variables en VHDL, qui elles sont uniquement disponibles à l’intérieur des processus, l’utilisation des « for generate » et des « if generate » en VHDL pour générer plusieurs instances d’un élément logique rapidement, l’utilisation des « integer » et des nombres naturels en VHDL, la gestion de ressources dans les FPGA, soit les blocs RAM, les blocs DSP et tous les autres modules disponibles à l’intérieur du FPGA, les broches dédiées dans les FPGA, comme par exemple les broches dédiés aux signaux des horloges, l’utilisation plus extensive des bancs de tests pour tester plusieurs modules entre eux, et finalement cela m’a permis d’avoir une meilleur conceptualisation de la façon dont la description matérielle est implémenté au niveau des portes logiques du FPGA à l’aide du synthétiseur. Deuxièmement, j’ai appris à utiliser un codec audio. Ceux-ci sur-échantillonnent le signal audio dans le but d’améliorer la résolution du signal audio, de réduire le bruit et de réduire la taille des filtres d’anti-repliement. Aussi, ceux-ci utilisent quelques protocoles de communications dédiés au transfert de données audios modulés par impulsions codées (PCM). Le protocole que j’ai dû apprendre était le I2S, soit un standard développé par Philips Semiconductor permettant la communication de données modulés par impulsions codées entre deux composantes. Troisièmement, j’ai appris à concevoir des circuits imprimés dans l’optique de réduire le plus possible le bruit généré par les composantes numériques dans les circuits analogiques. Pour ce faire, j’ai utilisé un document mis par Normand Provencher sur le wiki TGE (voir annexe) pour comprendre plusieurs concepts importants lors de la conception de circuit imprimés comportant des circuits analogiques et numériques comme le placement des composantes par rapport au type de circuits et par rapport à la fréquence, la façon de créer un point neutre pour les retours de courants ainsi que la façon d’utiliser des condensateurs de découplage et de dérivation. Finalement, j’ai appris à utiliser un écran graphique de type LCD pour pouvoir afficher des images programmables.

**Annexes**

Document sur la conception des pcb sur wiki tge