Lab1

Questions Set #1 :

- On fait entrer deux pics en entrée dans le DSP et on ressort un signal avec un seul pic. Il y a donc eu un changement grâce à un filtre qui donne en sortie des transformées de Fourier.

- Dans ce spectre, le signal bleu est le signal d’entrée (IN) et le signal rose est le signal de sortie (OUT). Les deux signaux sont des signaux numériques.

- On remarque que le signal d’entrée possède 2 pics de même amplitude a contrario du signal de sortie qui n’en possède qu’un seul (le premier). Le filtre est donc un filtre passe-bas qui va annuler les fréquences supérieures trop fortes.

Lab2

Question Set #1 (0x00FFFF00)

• what is the meaning of this mask? (0x00FFFF00)

Ce masque signifie que on modifie tous les signaux reçus par la carte : le signal est composé de 4 octets. Dans ce masque, on supprime le premier octet (octet de poids fort) et le dernier octet (octet de poids faible), on les force à prendre la valeur 0.

• what do you notice (in audio)?

Dans le casque, on observe que les sons forts ainsi que les basses fréquences sont « salis », elles perdent en qualité et semblent très parasités, tandis que les bruits (interférences, légères erreurs) disparaissent.

• can you explain this result?

En supprimant les bit de poids fort, on perd de l’information, dans le cas présent, on supprime l’octet de poids le plus fort, on perd donc toutes les informations importantes comme les sons fort, ceux qui ont une grande amplitude. C’est pour cela que les sons forts et de basse fréquence (les sons de grands amplitude) perdent en qualité, parce qu’ils sont parasités, modifiés.

En supprimant l’octet de poids faible, on perd également des informations, mais cette fois ci ces informations sont des informations de moindre importance, ce sont des informations négligeables. C’est dans ces bits que se trouve généralement le bruit et les parasites, c’est pour ça que ces bruits disparaissent.

Question Set #2 (0x000FFFFF)

• what is the meaning of this mask? (0x000FFFFF)

Ce masque signifie que on modifie tous les signaux reçus par la carte : le signal est composé de 4 octets. Dans ce masque, on supprime le premier octet et demi (octet de poids fort), on le force à prendre la valeur 0.

• what do you notice (in audio)?

Dans le casque, on observe que les sons forts ainsi que les basses fréquences sont « salis », c’est similaire à ce que l’on entendait avant, mais cette fois ci, le son est beaucoup plus parasité.

• can you explain this result?

En supprimant l’octet et demi de poids fort, on perd de l’information, dans le cas présent, on supprime les bits de poids les plus fort, on perd donc toutes les informations importantes comme les sons fort, ceux qui ont une grande amplitude. C’est pour cela que les sons forts et de basse fréquence (les sons de grands amplitude) perdent en qualité, parce qu’ils sont parasités, modifiés. Comme pour le masque précédent, mais cette fois ci, comme on masque plus que le bit de poids le plus fort, on perd beaucoup plus d’information que précédemment. Quand dans l’exercice précédent, on perdait que les informations de plus grande importance (donc assez rare) cette fois ci on supprime aussi celles de plus moyenne importance, ce qui fait que on supprime beaucoup plus d’informations, et donc le signal est beaucoup plus parasité.

Question Set #3

• what happens if we try to make it even louder?

Si on essaie de le mettre encore plus fort, le son se retrouve parasité

• can you explain why?

Cela s’explique par le fait que certains bits dépassent la limite imposée par la taille du mot. Les informations les plus importantes sont naturellement situés sur les bits de poids fort (à gauche du mot ; au début du mot). En effectuant des décalages vers là gauche, on rapproche ces bits de la limite du mot, et si on décale trop, les bits sortent du mot et donc se perdent.

Je donne un exemple :

On utilise le mot (0x14CD) = (0b 0001 0100 1100 1101)

Si on fait un double décalage à gauche on obtient le mot suivant :

(0x5334) = (0b 0101 0011 0011 0100) : ce mot est plus fort que le mot précédent, et n’est pas modifié : on observe bien qu’on ne perd pas de 1 dans le mot binaire.

Si on fait un triple décalage à gauche on obtient le mot suivant :

(0xA668) = (0b 1010 0110 0110 1000) : ce mot est plus fort que le mot précédent, et n’est pas modifié : on observe bien qu’on ne perd toujours pas de 1 dans le mot binaire.

Si on fait un quadruple décalage à gauche on obtient le mot suivant :

(0x4CD0) = (0b 0100 1100 1101 0000) : ce mot est moins fort que le mot précédent, et est modifié : on observe bien qu’on a perdu un 1 dans le mot binaire. Cela donne une information complétement différente de la précédente : c’est cet effet de débordement qui parasite le signal.

Question Set #4

• compare the values of iChannel0LeftOut and iChannel0LeftIn. Is this what you expect?

Les valeurs sont bien celles attendues. La valeur de LeftOut est bien égale à celle de LeftIn, à laquelle on à appliqué un décalage d’un octet à gauche, puis un masque sur le premier et dernier octet.

• do the same thing with iChannel0RightOut and iChannel0RightIn

Les valeurs sont bien celles attendues, on obtient la même chose que pour le canal de gauche. La valeur de rightOut est bien égale à celle de RightIn, a laquelle on à appliqué un décalage d’un octet à gauche, puis un masque sur le premier et dernier octet.

• can you explain now why the signal got distorted?

Le signal est déformé parce que on applique un masque sur les bits de poids fort après les avoir décalés vers la gauche, ce qui revient à faire un masque (dans le cas présent) sur les 12 premiers bits. C’est comme dans les questions 2, on perd beaucoup d’information. On peut conserver le masque ainsi que les décalages à gauche tout en perdant moins d’information en appliquant le masque sur le signal d’entrée avant d’effectuer le décalage. En faisant comme ça, on applique le masque seulement sur les 8 bits de poids fort, on perds donc moins d’information importantes.

Talk Through with TDM

Question: Can you give two differences between TDM and I2S from the source code of the two projects?

TDM et I2S sont 2 méthodes de transport de données différentes.

Le TDM est une méthode qui implique du multiplexage et du démultiplexage, ce qui permet de communiquer avec un plus grand nombre de canaux à la fois.

L’I2S est une autre méthode, qui fonctionne par adressage, mais qui est plus complexe et qui dans le cas présent ne permet pas une communication avec toutes les sorties audios.

Dans le code on peut voir que pour le traitement des données, qu’on utilise un troisième buffer pour la troisième sortie audio, ce qu’on ne fait pas pour l’I2S.

On voit aussi que le sous-programme Sport0 est différent : pour le TDM, on fonctionne sur 32 bits, tandis qu’en I2S on ne fonctionne que sur 24 bits