



Devoir surveillé de DSP (Processeur de Traitement de Signal) I3 Electronique

Benoît Decoux – vendredi 7 janvier 2005

Durée : 2 heures - Tout document et calculatrice autorisés

Le corrigé de cet examen sera disponible dans la semaine du 14 janvier
sur le site www-dsp.efrei.fr

Attention : avant de commencer, lire l'erratum à la fin de ce sujet.

Exercice 1 : Arithmétique des DSP (4 points)

- 1) Quelle est la plage de valeurs du format virgule fixe [1,24] ?
- 2) Donner la valeur décimale correspondant à la valeur hexadécimale \$111111 exprimée au format virgule fixe [1,24].
- 3) Donner la valeur hexadécimale, exprimée dans le format virgule fixe [1,24], correspondant à la valeur décimale 0,125.
- 4) Quelle est la plus grande valeur positive que l'on peut coder dans le format mixte [9, 47] ? (Donner la valeur hexadécimale et la valeur décimale)

Exercice 2 : Programmation de base en assembleur DSP56303 (4 points)

- 1) Ecrire un programme qui recopie en mémoire Y le contenu d'une zone de 16 valeurs de la mémoire X. Le commenter.
- 2) Compléter le programme pour qu'il multiplie chacune de ces valeurs par 2 lors du transfert.
- 3) En utilisant l'adressage modulo, copier cette même zone de la mémoire X 10 fois dans la mémoire Y (ces copies étant concaténées).
- 4) Ecrire un programme qui génère un tableau de 256 valeurs de la fonction sinus sur une période, à l'adresse \$100 dans la mémoire Y.

Exercice 3 : Filtrage en C++ (4 points)

- 1) Ecrire une classe "Filtre1o" permettant de programmer des filtres numériques du 1^{er} ordre. Cette classe devra posséder comme membres : 1) des fonctions (= méthodes) : constructeur, destructeur, filtrage d'un échantillon, filtrage d'un tableau d'échantillons, et 2) des variables adéquates.
- 2) Ecrire la fonction de filtrage d'un échantillon citée ci-dessus, dont le seul argument est l'échantillon d'entrée de type float nommé "e" et la valeur retournée l'échantillon de sortie.
- 3) Ecrire la fonction de filtrage d'un tableau d'échantillons de type float nommé "tab_e", dont les arguments sont un pointeur sur ce tableau et sa taille, et la valeur retournée un pointeur sur un tableau des échantillons résultant du traitement.
- 4) Donner le programme principal permettant d'utiliser la fonction précédente et d'appliquer à un tableau d'échantillons "tab_e" deux filtrages en parallèle, la sortie de chaque filtre étant passée par un gain fixe paramétrable et les sorties des deux filtres étant additionnées pour générer un tableau "tab_s". On supposera que l'on dispose d'une fonction réalisant la lecture d'un bloc d'échantillons de signal à partir d'une entrée audio et d'une fonction d'envoi d'un bloc d'échantillon à la sortie audio, dont les prototypes sont respectivement :

```
float *entree_bloc_audio();  
void sortie_bloc_audio(float *tab_s);
```

Exercice 4 : Filtrage en assembleur DSP56303 (4 points)

- 1) Ecrire une routine en assembleur, appliquant l'opération de dérivation suivante à un échantillon d'entrée :

$$s(n) = e(n) - e(n-1)$$

$s(n)$ est l'échantillon de sortie et $e(n)$ l'échantillon d'entrée, n est le pas d'échantillonnage $k \times T_e$ avec k entier et T_e période d'échantillonnage.

On supposera cet échantillon présent dans l'accumulateur A à l'entrée dans la routine, et il faudra mettre l'échantillon résultat avant la sortie dans ce même accumulateur.

Préciser les autres parties à ajouter éventuellement au programme de départ (programme réalisant l'opération entrée-sortie directe "pass.asm", par exemple).

- 2) Même question pour l'opération d'intégration suivante :

$$s(n) = e(n) + s(n-1)$$

- 3) Même question pour l'effet audio appliquant un écho répétitif à un signal, décrit par l'équation de récurrence suivante :

$$s(n) = e(n) + a \times s(n-4800), \text{ avec } a < 1$$

- 4) Dans le cas du traitement précédent, indiquer la durée du décalage temporel entre les répétitions successives du son à la fréquence maximale d'échantillonnage de la carte à DSP utilisée en travaux pratiques, et préciser si ce traitement peut être implanté sur cette même carte (en justifiant la réponse).

Exercice 5 : FFT (transformée de Fourier rapide) en assembleur DSP56303 (4 points)

- 1) Décrire en quelques phrases ce que fait le programme suivant :

```
nb_val equ 8
tab1 equ $20
tab2 equ $40
genetab macro points
org x:tab1
cmpt set 0
dup nb_val
dc cmpt
cmpt set cmpt+1
endm
endm
org x:tab1
ds nb_val
org x:tab2
ds nb_val
genetab nb_val
org p:$40
jsr rout
jmp *
rout
move #tab1,r0
move #tab2,r1
move #0,m0
move #-1,m1
move #nb_val/2,n0
do #nb_val,fin
move x:(r0)+n0,x0
move x0,x:(r1)+
fin
rts
end
```

- 2) Donner la valeur des indices des éléments d'un tableau en contenant 16, après renversement des bits des valeurs initiales (0, 1, ..., 15) de ces indices.
- 3) Donner l'état des sorties de la TFR (Transformée de Fourier Rapide, ou FFT) appliquées à 256 échantillons d'un signal sinusoïdal composé de 2 périodes, d'amplitude égale à 1.

Erratum

Dans le document "Résumé de cours – 2^e partie" (fft.pdf), il est indiqué :

"registre Ni chargé avec le nombre de bits à inverser"

Il faut remplacer cette phrase par celle-ci :

"registre Ni chargé avec 2^{k-1} , si la taille du tableau est 2^k "