École de technologie supérieure Professeur : Marcel Gabrea
Département de génie électrique Session: Automne 2017

SYS-835 : Processeur numérique du signal et ses applications

Laboratoire 2

Introduction à la carte DSP et à sa programmation.

2.1 Objectifs

Ce laboratoire a pour but de familiariser l'étudiant(e) avec l'utilisation du logiciel Code Composer Studio (CCS) et avec l'utilisation des cartes numériques.

2.2 Matériel requis

Traitement de texte (Word)
Carte DSP (DSK TMS320C6713)
Code Composer Studio (CCS 5.2.1)

2.3 Introduction à la carte DSK TMS320C6713

Voir le paragraphe 4.5 du Chapitre 4 2017 architecture TMS320C6x.pdf et Code Composer Studio v5 - Ex01a.pdf sur le site du cours.

Écrire un programme en C pour calculer $y = \sum_{i=0}^{39} x(i)h(i)$ où $x(i) = 2i - 70$ et $h(i) = 28 - 3i$ à l'aide de CCS en mode **simulation** et par la suite sur la carte **DSK TMS320C6713**.

Faire les captures d'écran de toutes les étapes et les copier dans le fichier Word lab2.doc ou lab2.docx avec une description de deux – trois lignes de chaque étape.

Vérifier le résultat final (*y*).

2.4 Optimisation. Mesures de performances.

Utiliser CCS 5.2.1.00018 en mode **simulation** afin de pouvoir mesurer (en utilisant le profiler) le nombre de cycles de chaque fonction.

Utiliser les configurations Debug et Release afin de mesurer les performances du programme écrit au point 2.3 en utilisant le Profiler de CCS.

Faire les captures d'écran de toutes les étapes et les copier dans le fichier Word lab2.doc ou lab2.docx avec une description de deux – trois lignes de chaque étape.

2.5 Programmation en assembleur

Voir le paragraphe 5.1 du Chapitre 5 2017 optimisation TMS320C6x.pdf sur le site du cours.

Soit le code suivant :

Main.c

```
short   child (short *, short *, short );

short   z;
short   w[3] = {1, 2, 3};
short   x[3] = {10, 20, 30};
short   y[3] = {40, 50, 60};

void    main(void ) {
        short   n=3;

        z = child(x, y, n);

}
```

Child.c

```
short   child(short *a, short *b, short n){
        int     i;
        short c=0;

        for(i=0; i<n; i++)
        {
                c += w[i] * a[i] - b[i];
        }
        return c;
}
```

Coder la fonction child en assembleur (child1.asm) et en assembleur linéaire (child2.sa).

Vérifier le résultat final (*z*) en appelant les fonctions child, child1 et child2 à l'aide de **CCS** et de la carte **DSK TMS320C6713**.

Faire les captures d'écran de toutes les étapes et les copier dans le fichier Word lab2.doc ou lab2.docx avec une description de deux – trois lignes de chaque étape.