

Série TD : Architecture et programmation des DSP C6x

Exercice 1

Supposons un paquet FP qui comprend une opération MAC répartie sur plusieurs paquets EP:

```
LDH .D1 A0*,A1
|| NOP
|| LDH .D1 A0*,A2
|| NOP
NOP
|| MPY .M1 A1,A2,A3
|| ADD .L1 A4,A3,A4
NOP
```

1. Identifier chacun des paquets EP.
2. Voir si le paquet FP s'exécute correctement sachant que MAC comprend deux instructions (MPY → ADD).
3. Proposer un arrangement correct pour le paquet FP afin que l'opération MAC s'effectue correctement. Justifier.

Exercice 2

Soit programme assembleur TMS320C6713 suivant :

```
LDH .D1 *A5,A4
NOP 4
MVK .S1 A4,A1
SUB .L1 A1,1,A1
LOOP : MPY .M1 A4,A1,A4
NOP
SUB .L1 A1,1,A1
[A1] B .S1 LOOP
NOP 5
STH .D1 A4,*A6
```

- a) Que fait cette fonction
- b) Quelle est la fonction associée au registre A1 dans ce programme
- c) Justifier la raison de l'instruction NOP à la 2ème, la 5ème et à la 8ème ligne du programme
- d) Décrire les opérations effectuées lors de l'exécution du programme

e) Trouver la valeur maximale que peut avoir x pour que le résultat de l'exécution de ce programme soit significatif

Exercice 3 : Produit scalaire

1. Ecrire en C la fonction réalisant : $y = \sum_{i=1}^{100} a_i \cdot x_i$
2. Récrire la même fonction en assembleur
3. Calculer le nombre de cycles d'exécution dans ce cas.
4. Proposer une optimisation sur votre code de sorte à réduire le nombre de cycles.

Exercice 4

Nous souhaitons générer deux vecteurs $X^{\rightarrow} = \{x_0, x_1, \dots, x_{N-1}\}$ et $Y^{\rightarrow} = \{y_0, y_1, \dots, y_{N-1}\}$ par l'utilisation d'une suite définie par :

$$\begin{cases} x_{n+1} = 1 - y_n + |x_n| \\ y_{n+1} = x_n \end{cases}$$

Où $x_0 = -0.25$, $y_0 = 0.25$, $x_n \in [-1, +1]$, $y_n \in [-1, +1]$, $n \in (0, 1, 2, \dots, N-1)$ et $N=6$

Coder un programme assembleur C6x qui réalise le calcul précédent sur un DSP C6x en utilisant une arithmétique à virgule fixe format Q1.31 signé.

Exercice 5

Supposons qu'on aura besoin de générer une suite $X = (x_0, x_1, \dots, x_{N-1})$ de $N=20$ réels définis par :

$$\begin{cases} \mu x_n, & x_n = 0.5 \\ \mu - \mu x_n, & x_n \neq 0.5 \end{cases}$$

avec $\mu = 0.85$ et $x_0 = 0.75$

a) Coder une fonction assembleur `_gen_suite` qui permet de générer la suite X en utilisant une arithmétique Q1.15.

Linker Command File (à ajouter à un projet en assembleur pour mapper les sections à la mémoire)

extension **.cmd** (nom_du_fichier.cmd)

```
MEMORY
{
  PROGRAM      : origin = 0x0,      len = 0x10000
  DATARAM      : origin = 0x80000000, len = 0x10000
}

SECTIONS
{
  .vectors > PROGRAM
  .text > PROGRAM

  .bss > DATARAM
  .data > DATARAM
  .stack > DATARAM
  .systemem > DATARAM
}
```