Université de Paris Sud
———
Master IST - M2Pro

DSP

Fonctions mathématiques

Compte rendu de TP à rendre pour la semaine suivante dernier délai

1 Calcul du PGCD

Le calcul du PGCD (Plus Grand Commun Diviseur) est un algorithme classique. Son implentation efficace sur DSP est utilise pour la simplification de fraction. Si l'algorithme initial d'Euclide est efficace lorsqu'une instruction de division est disponible, elle peut être problématique sur une architecture ne disposant pas de division. Les Algorithmes proviennent de *Semi-numerical algorithm : the art of computer programming* de D. Knuth.

1.1 Algorithme d'Euclide

Soient u et v deux entiers positifs, l'algorithme suivant calcule le pgcd de u et v :

1. si $v = 0$, l'algorithme se termine et v est la réponse u,
2. $r \leftarrow u \ mod \ v$, $u \leftarrow v$, $v \leftarrow r$. Retournez en 1. (Note : ces opérations réduisent la valeur de v mais laissent le pgcd inchangé)

A faire :

1. Portez l'algorithme en C sur DSP. Validez votre code en calculant pgcd(40902,24140).
2. Analysez le code asm généré. Que peut on dire ?

1.2 Algorithme binaire

Grâce aux propriétés mathématiques du PGCD, il est possible d'adapter l'algorithme aux architectures sans division. En effet :
- si u et v sont tous les deux pairs, alors $pgcd(u, v) = 2pgcd(u/2, v/2)$,
- si u est pair et v impair, alors $pgcd(u, v) = pgcd(u/2, v)$,
- $pgcd(u, v) = pgcd(u - v, v)$,
- si u et v sont tous les deux impairs, alors $u - v$ est pair, et $|u - v| < max(u, v)$.

On obtient l'algorithme suivant :

1. initialisez $k \leftarrow 0$ et répétez la séquence suivante : $k \leftarrow k + 1$, $u \leftarrow u/2$, $v \leftarrow v/2$ jusqu'à ce que u et v ne soient plus tous les deux pairs.
2. (maintenant les valeurs initiales de u et v ont été divisées par 2^k et l'une des deux valeurs est impaire).
   Si u est impair alors $t \leftarrow -v$ et allez en 4. Sinon $t \leftarrow u$
3. (A ce moment t est pair et non nul) $t \leftarrow t/2$
4. si t est pair alors allez en 3
5. si $t > 0$, alors $u \leftarrow t$, sinon $v \leftarrow -t$ (la plus grande valeur de $|u|$ et $|u|$ a été remplacé par $|t|$)
6. $t \leftarrow u - v$. Si $t \neq 0$ allez en 3 sinon l'algorithme se termine avec comme réponse $u \cdot 2^k$

A faire :

1. Portez l'algorithme en C sur DSP. Validez votre code en calculant pgcd(40902,24140).
2. Analysez le code asm généré. Que peut on dire ?
3. Portez l'algorithme en assembleur linéaire.

2 Fonction réciproque : calcul de l'inverse

1. Implantez l'algorithme de Newton en C pour le calcul de $1/d$ avec $d > 1$
2. Implantez l'algorithme de Newton en assembleur linéaire
3. comparez les performances en C et en assembleur linéaire

3 chronométrie

Utilisez la méthode de chronométrie du TP précédent.