Exercices pour le cours ALPA

P. Quinton, T. Risset

28 septembre 2001

Version Corrigée

Les exercices difficiles sont indiqués par une étoile ($\star$)

1 algorithmique systolique

PRÉREQUIS : ALGORITHMIQUE SYSTOLIQUE. (source: [**?**])

1.1 Convolution généralisée (15 mn)

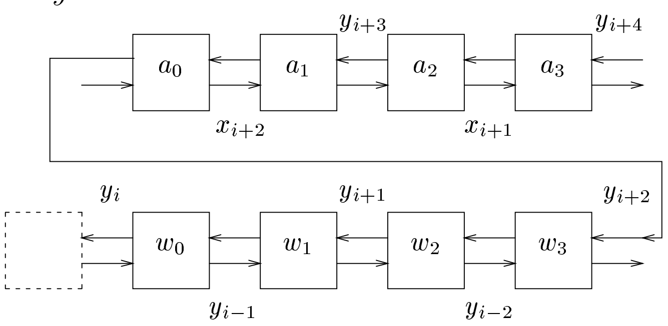
- Décrire l'architecture systolique pour la convolution généralisée

$$y_i = \sum_{k=0}^{3} a_k x_{i-k} + \sum_{k=0}^{3} w_k y_{i-k+1}.$$

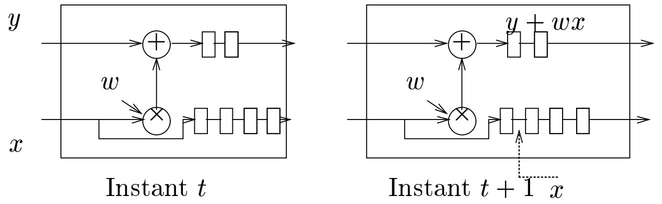
- Expliquer pourquoi on est naturellement conduit à utiliser des réseaux de convolution bidirectionnels.

- Expliquer ce qu'on peut faire pour utiliser, malgré tout, un réseau de type unidirectionnel pour la partie non récursive de la convolution.

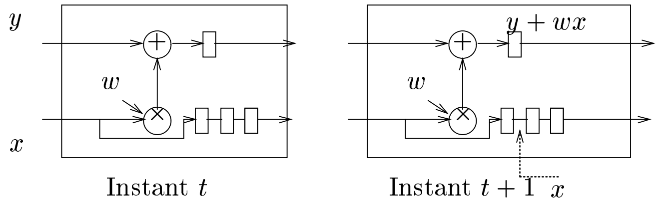
Corrigé: *Le réseau est le suivant, on utilise le réseau de la convolution récursive pour la partie récursive ($\sum_{k=0}^{3} w_k y_{i-k+1}$), on a donc une période de 2 pour les coefficients y. Du coup, on doit utiliser un réseau bidirectionnel aussi pour la partie non récursive car on ne peut pas "accélérer" les y entre les deux réseaux.*



On peut, malgrès tout, utiliser un réseau unidirectionnel (le réseau vu en cours) pour la partie non récursive, mais pour respecter la période des coefficients y, il faut que l'horloge contrôlant les registres de ce réseau soit deux fois plus lente que l'horloge du réseau bidirectionnel. Si on veut conserver une seule horloge, on ralentit le réseau en doublant le nombre de registres.



On s'aperçoit alors que l'on peut optimiser cette architecture. En effet, entre deux cellules successives, les y rencontrent deux registres et les x quatre. Le fait de retirer un registre sur les deux canaux en même temps ne modifiera pas les positions relatives des coefficients x et y (cette notion est formalisée par ce qu'on appelle le "théorème de la coupe"). On peut donc passer là l'architecture suivante qui effectue le même traitement que la précédente mais dont les sorties arrivent plus tôt:



1.2 Produit de matrices (15 mn)

- Décrire l'architecture systolique (réseau carré) pour le produit de matrices carrées $C = AB$.
- Expliquer comment modifier ce réseau et son utilisation, de façon que les coefficients de C soient évacués vers la gauche, afin de reboucler les résultats.
- Examiner aussi le cas ou les résultats C sont rebouclés, lorsqu'ils sortent (normalement) par la droite du réseau. Donner le délai nécessaire entre l'entrée de deux matrices successives, et expliquer quel dispositif de mémorisation doit être utilisé.

Corrigé: *à faire*

1.3 Synthèse par équations: transformée en ondelettes (45 mn) ⋆

Pré-requis : Synthèse automatique de réseau systoliques

La transformée en ondelettes à récemment beaucoup été étudiée. Elle permet de faire un passage dans le domaine des fréquences de manière hiérarchique. Le calcul se fait en plusieurs *niveaux*, chaque niveau rendant successivement une information *de moins en moins précise*. À chaque niveau, on peut modéliser le calcul effectué lors d'une transformée en ondelettes unidimensionnelle (à N coefficients et par un facteur de deux) par la formule suivante:

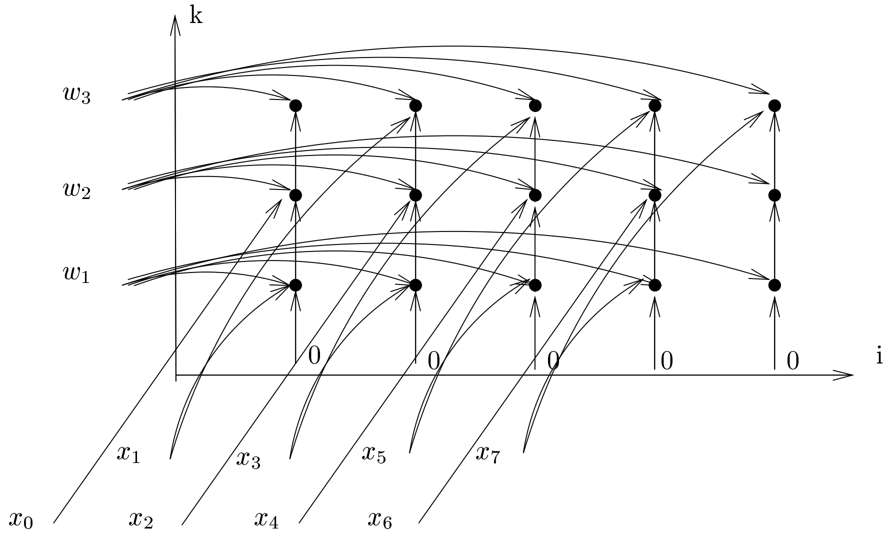
$$y(i) = \sum_{k=1}^{N} w(k) * x(2i - k) \tag{1}$$

dans laquelle, les $x(j)$ sont des coefficients donnés en entrée, les $y(i)$ sont les résultats, les $w(k)$ sont N coefficients fixés. On va rechercher une architecture systolique pour ce problème. On ne vous demandera pas d'écrire les équations récurrentes au cours de la dérivation de l'architecture, mais d'expliquer informellement sur le graphe de dépendance.

1. De quelle manière peut-on sérialiser la réduction de l'équation (1). Choisissez une de ces possibilités et représentez le graphe de dépendance étendu pour $N = 3$, et $M = 5$ (M est le nombre de coefficient y que l'on calcule).

   Corrigé*:*

   On peut sérialiser dans l'ordre croissant des Y ou dans l'ordre décroissant. Dans l'ordre croissant, cela donne:

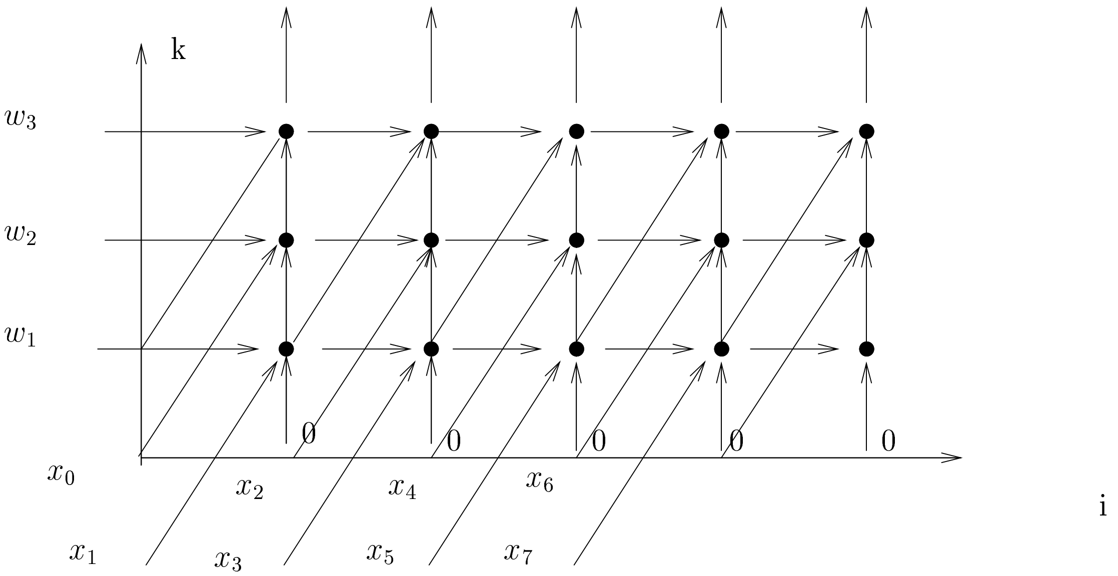


2. énumérez les possibilités pour pipeliner les variables x et w dans ce graphe. Quelles sont, selon vous, les cas à éviter. Choisissez une de ces possibilité et représenter le graphe de dépendance étendu uniformisé pour $N = 3$, et $M = 5$.

   corrigé: *On doit pipeliner les $w(k)$ dans la direction $(0, 1)$ (deux sens possibles), et les $x(j)$ dans la direction $(1, 2)$ (deux sens possibles). Il faut éviter de pipeliner en $(0, 1)$ et $(-1, -2)$ ni en $(0, -1)$ et $(1, 2)$ car on ne pourra pas trouver d'ordonnancement affine*

   On choisit (Par exemple) de pipeliner les $w(k)$ selon la direction $(1, 0)$ (de gauche à droite) et les $x(j)$ selon la direction $(1, 2)$ (de bas en haut). Le nouveau système *d'équations récurrentes obtenu est le suivant:*

$$
\begin{array}{ll}
\{i,k \mid 1 \leq i \leq M; k=0\}: & y(i,k)=0 \\
\{i,k \mid 1 \leq i \leq M; 1 \leq k \leq N\}: & y(i,k)=y(i,k-1)+W(i,k)*X(i,k) \\
\{i,k \mid i=0; 1 \leq k \leq N\}: & W(i,k)=w(k) \\
\{i,k \mid 1 \leq i \leq M; 1 \leq k \leq N\}: & W(i,k)=W(i-1,k) \\
\{i,k \mid 1 \leq i \leq M; 1 \leq k \leq N\}: & X(i,k)=X(i-1,k-2) \\
\{i,k \mid -1 \leq k \leq 0; -1 \leq i \leq M\}: & X(i,k)=x(2i-k) \\
\{i,k \mid -1 \leq i \leq 0; 1 \leq k \leq N\}: & X(i,k)=x(2i-k)
\end{array} \tag{2}
$$



3. Donnez les contraintes que doit vérifier une fonction de temps pour le graphe trouvé à la question 2. Choisissez une fonction de temps respectant ces contraintes. Donnez une fonction d'allocation des calculs compatible avec la fonction de temps choisie.

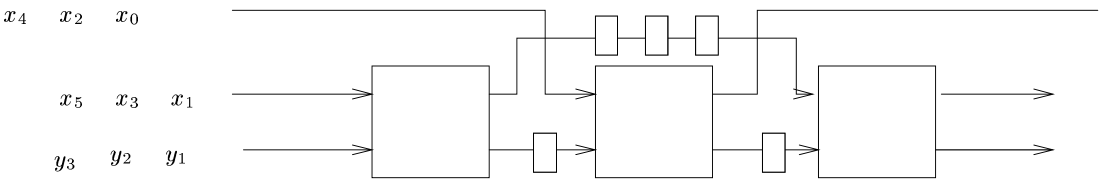
   Corrigé:

   $\tau_i + 2\tau_k \geq 1$
   $\tau_k \geq 1$
   $\tau_i \geq 1$

   la fonction de temps: $T(i,k) = i + k$ *est valide car* $(1,1)$ *respecte les trois vecteurs de dépendance:* $(1,2),(1,0)et(0,1)$. *la fonction d'allocation* $A(i,k) = k$ *est valide car* $(1,1).(0,1) = 1 \neq 0$.

4. Décrivez l'architecture obtenue avec les fonctions de temps et d'allocation trouvées à la question précédente. On pourra faire un schéma de l'architecture.

   Corrigé:



5. Quelle sont: la période du flot de y, la latence (temps total pour l'exécution d'une instance du problème), la macro période (temps entre l'exécution de deux instances successives de problème)?

   corrigé: *période: 1 (un résultat tous les tops.)*

   latence: $N + M$ *(*`a voir`*)*

   efficacité $\frac{NM}{(N+M)*N} = \frac{M}{N+M}$ *(*`a voir`*)*

   macro période: $M + 1$