

# M2R - NSI - Optimisation et compilation

Rattrapage, 10 mai 2011

*Barème indicatif: problème 1: 50%, problème 2: 50%. Les questions n'ont pas forcément une réponse unique. Les analyses critiques seront appréciées.*

## 1 Modèle polyédrique

On considère la boucle suivante:

```
for (i=0; i<n; i++)
  for (j=0; j<n-i-1; j++)
    S(i, j) = 3*S(i, j+i) + S(j, i) + j + 2*i;
```

- 1) Ecrire l'espace d'itération sous forme polyédrique (i.e.  $Ax \leq b$ ).  
Expliciter mathématiquement  $A$  et  $b$ , et écrire les 3 fonctions d'accès.
- 2) Montrer que les fonctions d'accès sont toutes valides en tout point  $(i, j)$ .
- 3) Calculer le volume de l'espace d'itération.
- 4) Peut-on simplement intervertir les deux boucles sur  $i$  et  $j$  ? Pourquoi ?
- 5) Ecrire une version équivalente du code avec les boucles en  $i$  et en  $j$  inversées.
- 6) Définir et expliquer trois types de transformations de boucles au choix.

## 2 Optimisation de code VLIW

Le processeur ST200 est un processeur VLIW. Il comporte entre autres 4 unités de calcul entier, une unité de load/store, une unité de branchement.

Deux versions compilées, avec les options -O2 et -O3 respectivement du compilateur st200cc, du même kernel (fonction C) sont présentées ci dessous.

1. Que fait cette fonction? (décrivez la ou écrire l'algorithme)
2. Quelles sont les optimisations appliquées aux deux versions? Justifiez.
3. Quelle est la meilleure version? Justifiez.
4. Question de cours: l'ordonnancement par software pipelining (pipeline logiciel) est-il optimal? Justifiez.

Nota Bene:

- ldw est l'acronyme de l'instruction assembleur load word.

- add est l’acronyme de l’instruction assembleur addition.
- cmpne est l’acronyme de l’instruction assembleur compare not equal.
- br est l’acronyme de l’instruction assembleur branch.
- L?\_BB10\_main: est un label de branchement.
- \$r représente un registre entier. \$b représente un registre de condition.
- un “bundle” est un ensemble d’opérations entre deux paires de points-virgule “;”. Les opérations d’un “bundle” sont lancées en parallèle.
- Une addition prend 1 cycle. Un chargement mémoire a une latence de 3 cycles.
- La partie finale du code d’écriture du ou des résultats en mémoire n’est pas donnée ici.
- Le code comporte aussi des épilogues de boucle qui ne sont pas donnés ici.
- On négligera l’instruction “stw 420[r12] =r63”.

| loop with -O2   | loop with -O3   |
|---|---|
| <i>L?_BB10_main</i> :<br>ldw \$r20 = 0[\$r15];;<br>ldw \$r19 = 4[\$r15];;<br>ldw \$r18 = 8[\$r15]<br>cmpne \$b0 = \$r16, 96;;<br>ldw \$r8 = 12[\$r15]<br>add \$r17 = \$r20, \$r17;;<br>add \$r17 = \$r19, \$r17;;<br>add \$r17 = \$r18, \$r17<br>add \$r15 = \$r15, 16<br>add \$r16 = \$r16, 4;;<br>add \$r17 = \$r8, \$r17<br>br \$b0, <i>L?_BB10_main</i> ; ; | ldw \$r17 = 0[\$r15]<br>add \$r15 = \$r15, 4;;<br>add \$r17 = \$r17, \$r18;;<br>ldw \$r20 = 0[\$r15];;<br>ldw \$r19 = 4[\$r15];;<br>ldw \$r18 = 8[\$r15];;<br>stw 420[\$r12] = \$r63;;<br><i>L?_BB10_main</i> :<br>add \$r23 = \$r20, \$r17<br>ldw \$r22 = 12[\$r15]<br>cmpne \$b0 = \$r16, 100<br>add \$r16 = \$r16, 9;;<br>add \$r23 = \$r19, \$r23<br>ldw \$r21 = 16[\$r15];;<br>add \$r23 = \$r18, \$r23<br>ldw \$r20 = 20[\$r15];;<br>add \$r22 = \$r22, \$r23<br>ldw \$r19 = 24[\$r15];;<br>add \$r21 = \$r21, \$r22<br>ldw \$r18 = 28[\$r15];;<br>add \$r20 = \$r20, \$r21<br>ldw \$r17 = 32[\$r15]<br>add \$r15 = \$r15, 36;;<br>add \$r19 = \$r19, \$r20<br>ldw \$r20 = 0[\$r15];;<br>add \$r18 = \$r18, \$r19<br>ldw \$r19 = 4[\$r15];;<br>add \$r17 = \$r17, \$r18<br>ldw \$r18 = 8[\$r15]<br>br \$b0, <i>L?_BB10_main</i> ; ; |