

Examen Compilation Avancée – 2nde partie (10 points)

A traiter sur une copie séparée

Tous documents autorisés – 9 questions – 2 pages

Considérer le code assembleur de référence suivant (les commentaires sont donnés en *italique*), les instructions sont codées sur 32 bits :

Loop : (I1) LD R2, R3, R20 ; *chargement dans le registre R2 du contenu mémoire situé à l'adresse R3+ R20*
(I2) LD R4, R5, R20
(I3) ADD R2, R2, R4 ; *R2 = R2 + R4*
(I4) MUL R6, R7, R8 ; *R6 = R7 * R8*
(I5) SUB R6, R6, R11 ; *R6 = R6 - R11*
(I6) LD R12, R13, R20
(I7) OR R4, R4, R5 ; *R4 = R4 OU R5*
(I8) SUB R12, R12, R1
(I9) ADD R6, R6, R4
(I10) ADDI R20, R20, 4 ; *R20 = R20 + 4*
(I11) BNE R20, R1, Loop ; *si R20 <> R1, alors PC = Loop*

1. Donner un exemple de dépendances de données de chaque type : RAW : *Read After Write*, WAW : *Write After Write*, WAR : *Write After Read*, pour le code de référence.
2. Donner le graphe de dépendances (DAG) du code de référence (en spécifiant les fausses dépendances si besoin).
3. La latence des opérations mémoire est de 3 cycles, celle des opérations de multiplication de 2 cycles et la latence de toutes les autres opérations est de 0 cycle. Ce qui signifie que si LD R1, R2, R20 s'exécute à t , ADDI R1, R1, 2 s'exécutera à $t + 4$. Toutes les opérations sont pipelinées et tous les accès à la mémoire sont des succès. Calculer le chemin critique de ce graphe.
4. Donner le temps d'exécution du code de référence sans aucun réordonnement (insérer des NOP pour marquer les délais dans le code).
5. Utiliser l'algorithme d'ordonnement par liste afin d'ordonner le code pour un processeur scalaire (1 instruction exécutée par cycle). Les priorités considérées pour cet algorithme sont les suivantes : 1) instruction sur le chemin critique, sinon 2) instruction ayant la plus grande latence, sinon 3) instruction située d'abord dans l'ordre séquentiel du code. Donner le code correspondant et le temps d'exécution. On pourra utiliser la même représentation qu'en TD :

Temps	Instructions candidates	Instruction ordonnée
1		
2		
3		

...		
-----	--	--

6. Même question que précédemment en considérant l'ordonnancement par liste pour un processeur pouvant exécuter 2 instructions quelconques simultanément (VLIW). Donner le code résultat et le temps d'exécution. On pourra utiliser la même représentation qu'en TD :

Temps	Instructions candidates	Instructions ordonnées
1		
2		
3		
...		

7. Donner le nombre d'échecs associés à ce code pour N itérations (N tours de boucle) en considérant un cache à correspondance directe contenant 4 lignes (blocs) de 32 octets. A l'initialisation (en hexadécimal), R3 = 0x40000, R5 = 0x80020, R13 = 0xA00A0 et R20 = 0. Les éléments traités sont sur 4 octets (LD - chargement en mémoire de 4 octets).
8. Considérer le temps d'exécution du code = temps calcul + temps accès mémoire où temps accès mémoire = nombre d'échecs * latence d'accès mémoire. Donner le temps d'exécution sur les codes des questions 4 et 6 pour une latence d'accès mémoire de L cycles. Commenter.
9. Comment modifier le code de référence pour réduire le chemin critique ?