Université BORDEAUX 1 Samedi 15 juin 2013

Département Informatique

I.U.T. Bordeaux 1

**ASR2 - ARCHITECTURE**

**S2**

durée 1,5 h (3 problèmes)

(Feuilles jeu d’instructions du POWER PC 604 autorisées)

**1 Passage du langage C à l’assembleur PowerPC (temps estimé : 20’)**

Considérons le source de la fonction C suivante soleil.c :

1. int soleil(int tab[], int n)
2. {
3. int soit;
4. soit=tab[0];
5. int k;
6. for (k=1;k<n;k++)
7. if (tab[k]>soit) soit=tab[k];
8. return soit;
9. }

Répondre aux questions suivantes en justifiant votre réponse :

Question 1 : Quelle est la valeur retournée par cette fonction ?

Question 2 : Dans quel registre du processeur seront situés les arguments « n » et « tab » avant l’exécution du code de cette fonction ?

Question 3 : Dans quel registre du processeur doit être placé la valeur de retour ?

Question 4 : Donner la commande permettant d’obtenir le code assembleur correspondant à cette fonction soleil.c

Question 5 : Fournir un code assembleur traduisant cette fonction. On optimisera autant que possible et on expliquera chaque instruction.

**2 Un premier source assembleur mystérieux (temps estimé : 30’)**

Le but est d’optimiser le code assembleur ci-dessous :

1. stw r30,-8(r1)
2. stwu r1,-64(r1)
3. mr r30,r1
4. stw r3,88(r30)
5. stw r4,92(r30)
6. li r0,1
7. stw r0,40(r30)
8. li r0,0
9. stw r0,36(r30)
10. li r0,0
11. stw r0,32(r30)
12. L2:
13. lwz r0,32(r30)
14. lwz r2,92(r30)
15. cmpw cr7,r0,r2
16. blt cr7,L5
17. b L3
18. L5:
19. lwz r2,36(r30)
20. lwz r0,40(r30)
21. add r0,r2,r0
22. stw r0,36(r30)
23. lwz r2,40(r30)
24. lwz r0,88(r30)
25. mullw r0,r2,r0
26. stw r0,40(r30)
27. lwz r2,32(r30)
28. addi r0,r2,1
29. stw r0,32(r30)
30. b L2
31. L3:
32. lwz r0,36(r30)
33. mr r3,r0
34. lwz r1,0(r1)
35. lwz r30,-8(r1)
36. blr

Répondre aux questions suivantes en justifiant votre réponse :

Question 6 : Rappelez pourquoi il s’agit d’un sous-programme ou fonction et pourquoi il est dit *non optimisé*.

Question 7 : Combien de paramètres comporte cette fonction ? Quels sont leur type et leur nature (entrée ou sortie) ? Où sont-ils rangés en mémoire ?

Question 8 : Repérer précisément la boucle en notant les lignes correspondantes. Quelle variable (registre ou case mémoire) contrôle son déroulement ? Comment est-elle initialisée ? Quelle est la condition d’arrêt ? Comment évolue-t-elle ?

Question 9 : La boucle utilise-t-elle des registres de travail (ou case mémoire) ? Si oui, lesquels et quels sont leurs rôles ?

Question 10 : Quelle est la valeur retournée en sortie ?

Question 11 : Donner l’équivalent C de cette fonction.

**3 Un deuxième programme mystérieux (temps estimé 40’)**

Le but est de fournir un équivalent en langage C de la fonction assembleur suivante :

1. mr r9,r3
2. li r2,1
3. li r3,0
4. addi. r4,r4,-1
5. bltlr- cr0
6. L6:
7. slwi r0,r4,2
8. lwzx r0,r9,r0
9. mullw r0,r2,r0
10. add r3,r3,r0
11. slwi r0,r2,2
12. add r0,r0,r2
13. slwi r2,r0,1
14. addi. r4,r4,-1
15. bge+ cr0,L6
16. blr

Cette fonction comporte deux paramètres d’entrée un tableau (tab) et sa dimension (10 par exemple). Le tableau ne comporte que des cellules dont le contenu est inférieur à 10.

Répondre aux questions suivantes en justifiant votre réponse :

Question 12 : Pourquoi ce code est-il dit *optimisé* ?

Question 13 : Trouver un équivalent des lignes 11,12 et 13 en une instruction.

Question 14 : Repérer précisément la boucle en notant les lignes correspondantes. Quelle variable (registre ou case mémoire) contrôle son déroulement ? Comment est-elle initialisée ? Quelle est la condition d’arrêt ? Comment évolue-t-elle ?

Question 15 : La boucle utilise-t-elle des registres de travail ? Si oui, lesquels et quels sont leurs rôles ?

Question 16 : Détailler pour tab = {1,2,3,4,5,6,7,8,9,0} l’exécution pas à pas de cette fonction. Quelle est la valeur retournée en sortie ?

Question 17 : Quelle est la valeur retournée par cette fonction dans le cas général ?

Question 18 : Donner l’équivalent de cette fonction en langage C.