

ARCHITECTURE DES ORDINATEURS

PARTIEL Octobre 2011

Tous documents autorisés – 2H

Pour toutes les questions, on utilise le jeu d'instructions NIOS-II. Les différentes parties sont indépendantes

PARTIE 1 : Implantation mémoire

Soit la déclaration de variables suivante pour un programme C

```
short A,B,C ;  
double D[2];  
char toto[3] ;  
float F,G,H;  
unsigned char W ;  
int X,Y,Z ;
```

Q 1) En supposant que A soit à l'adresse A000 0000_H, donnez les adresses (hexadécimales) de C, D[1], toto[2], W et Z. Quelle est l'occupation mémoire en octets ?

Q 2) Proposer une déclaration qui minimise l'occupation mémoire et donnez le nombre d'octets correspondant à cette nouvelle déclaration.

PARTIE 2 : Exécution d'instructions

Les registres du processeur NIOS contiennent les valeurs suivantes en hexadécimal. (R0=0).

R1	1234 5678
R2	A000 4000
R3	FFFF FFFE
R4	ABEF CD01

Q 3) Donner le contenu des registres R5 à R10 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes.

- a) ADD R5, R2, R1
- b) SUB R6, R1, R2
- c) SLLI R7, R1, 4
- d) SRAI R8, R2, 8
- e) SRLI R9, R1, 1
- f) MUL R10, R1, R3

PARTIE 3 : Comparaison des jeux d'instructions ARM et NIOS II

Les modes d'adressage ARM sont rappelés en annexe.

Q 4) Donner l'instruction ou la suite d'instructions NIOS II correspondant à l'instruction ou la suite d'instructions ARM suivantes :

Instructions ARM :

- a) LDR R3, [R1, #4]
- b) LDR R3, [R1], #4
- c) LDR R3, [R1,#4] !
- d) ADD R3, R1, R2 LSL # 3
- e) CMP R1, R2
 BGT Boucle
- f) CMP R1, #4
 BEQ Boucle

PARTIE 4 : Désassemblage

Soit le programme assembleur P1 qui travaille sur un tableau d'entiers T[1000] dont l'adresse de début est contenue dans le registre R1 et range le résultat dans la case mémoire RES d'adresse 00003000_H

	ADDI R2,R0, 1000
	SLLI R2,R2,2
	ADD R2,R1,R2
	ADD R3,R0,R0
	ADD R4,R0,R0
Boucle	LDW R5, 0(R1)
	LDW R6,4(R1)
	ADD R3,R3,R5
	ADD R4,R4,R6
	ADDI R1,R1,8
	BNE R1,R2, Boucle
	ADD R3,R3,R4
	STW R3, 3000 _H (R0)

Q 5) Donner le programme C correspondant au programme P1

Q 6) Que fait le programme P1 ?

PARTIE 5 : Programmation assembleur

Q 7)Ecrire un programme en assembleur NIOS qui compte le nombre de valeurs d'un tableau d'entiers T[1000] qui sont supérieures ou égales à 10 et inférieures ou égales à 15.

L'adresse du tableau T est initialement contenue dans R1. On rangera le résultat dans la case mémoire d'adresse 0000 0100_H

Q 8) Quelles modifications faut il apporter au programme de la question Q7) pour le transformer en une fonction, avec passage des paramètres par R1 (adresse de T[0]) et R2 (nombre N) et retour du résultat dans R1 ?

PARTIE 6 : PREDICTION DE BRANCHEMENTS

Soit un branchement qui a le comportement donné ci-dessous, P signifiant un branchement Pris et N un branchement non pris.

Branchement	P	P	N	N	P	N	P	N	P	P	N	N
-------------	---	---	---	---	---	---	---	---	---	---	---	---

Q 9) Donner le nombre de bonnes prédictions de branchement en supposant

- a) un prédicteur statique « toujours Pris »
- b) un prédicteur dynamique 1 bit (initialisé à Pris)
- c) un prédicteur dynamique 2 bits (initialisé à Fortement Pris)

ANNEXE : Modes d’adressage ARM

Mode	Assembleur	Action
Déplacement 12 bits, Pré-indexé	[Rn, #deplacement]	Adresse = Rn + déplacement
Déplacement 12 bits, Pré-indexé avec mise à jour	[Rn, #deplacement] !	Adresse = Rn + déplacement Rn ← Adresse
Déplacement 12 bits, Post-indexé	[Rn], #deplacement	Adresse = Rn Rn ← Rn + déplacement