Examen S4-CLM Mai 2011

Tous documents autorisés. Calculatrices autorisées.

Durée 2h

Pour toutes les questions, une justification **concise** est demandée. Pour tout l'examen on considère le jeu d'instructions DIDE.

Jeu d'instructions DIDE [6 pts]

Donner les résultats en **hexadécimal**. NB : le binaire ne sera pas accepté.

Q1. Donner le codage des instructions :

a) ADD R15, R1, R4 b) ADDI R15, R1, 3 c) STORE R7, 8(R1) d) LUI R2, 0x1289 e) BC 8

Q2. L'état initial des registres est :

R2 = 0x81234567 R3 = 0x0000000F

Quel est l'état des registres après l'exécution des instructions

a) ADD R1, R2, R3 b) ADDI R1, R2, 0XFFFF c) XOR R1, R2, R3 d) SLL R1, 4, R2

d) SLR R1, 8, R2 e) SAR R1, 12, R2

Q3. Si R1 contient initialement 3, quelle est la valeur (notation décimale) de R2 après l'exécution du programme :

```
SLL R2, 7, R1
ADD R2, R2, R1
SLL R2, 1, R2
ADD R2, R2, R1
```

Conditionnelles et Boucles [8 pts]

Le tableau d'entiers X est implanté à partir de l'adresse 0x20000000, et contient {9, 7, 1, 4, 3, 5, 8} ; le tableau d'entiers Y est implanté à partir de l'adresse 0x10000000. L'état initial est :

R9=0x20000018 R10= 0x20000000 R11= 0x10000000

Soit le programme *Prog1*

```
      LOAD  R1, 0(R10)
Deb   STORE R1, 0(R11)
      LOAD  R2, 4(R10)
      ADD   R1, R1, R2
      ADDI  R10, R10, 4
      ADDI  R11, R11, 4
      CMP   R9, R10
      BGT   Deb
      STORE R1, 0(R11)
```

Q4. Donner la valeur (notation **décimale**) du registre R1 après la fin de l'exécution du programme.

Q5. Quel est le contenu (notation **décimale**) du tableau Y après la fin de l'exécution du programme ?

Q6. Donner le contenu (notation **hexadécimale**) des octets mémoire aux adresses comprises entre 0x10000000 et 0x1000001B. On utilisera une table, sur le modèle :

Adresse	Contenu
0x10000000	00
0x10000001	…

Q7. Soit le fragment de programme C.

```
int i, r ;
int X[N];
i = 0;
r = 1;
do {
if (X[i] > X[i+1])
        r = 0;
 i++;
 } while ((i < 99) && (r ==1));
```

a) Ecrire le programme assembleur correspondant à ce fragment. X est implanté à partir de l'adresse 0x10008000, et r à l'adresse 0x00004000. Il est vivement conseillé de donner une première version non optimisée, puis une ou plusieurs versions optimisées. NB : vos programmes doivent être commentés.

b) Comment faut-il modifier vos programmes si X est déclaré comme un tableau d'unsigned int ?

Q8. On considère la déclaration en C :

```
char c[5] ;
double x[3] ;
short k ;
int tab[7] ;
```

En supposant que l'allocation mémoire se fait dans l'ordre des déclarations et à partir de l'adresse 0, donner l'adresse de x[0], tab[0] et tab[6], en notation **décimale**.

Format des instructions [2 pts]

Q9. On veut utiliser 64 registres, et non plus 32, pour le jeu d'instructions DIDE, et conserver les mêmes codes opérations. Quelle est la taille maximale de l'immédiat dans les instructions arithmétiques et logiques ? NB : on demande le **nombre de bits**, pas la valeur de l'immédiat. La réponse est très simple et peut être justifiée en une ligne.

Micro-architecture [4 pts]

On considère le chemin de données de la figure 1, où l'UAL peut effectuer les opérations

S = Ea + Eb , notée + ; S = Ea + 4, notée +4 ; S = Ea, notée +0

BR est un banc de registres. EXT est un extenseur de signe, 16 vers 32 bits.

On augmente le jeu d'instructions DIDE avec les deux instructions BL et JMP décrites table 1.

Instruction	Codop	Action	Commentaire
BL Rd, Imm16	10010	RD <- PC, puis PC <- PC + ES(Imm16)	Sauvegarde de PC dans Rd, suivie d'un branchement inconditionnel.
JMP Rd	10110	PC <- Rd	Saut à l'adresse contenue dans Rd

Table 1 : Instructions supplémentaires

Q10. Compléter la table 2 pour décrire la logique de commandes pour

a) la lecture de l'instruction b) l'exécution de l'instruction ADD c) l'exécution de l'instruction BL
d) l'exécution de l'instruction JMP

Attention, 1 ligne = 1 cycle, vous devez ajouter des lignes.

Q11. En ne considérant que les quatre actions de la Q10., donner une représentation graphique et la table de transition de l'automate de séquencement.

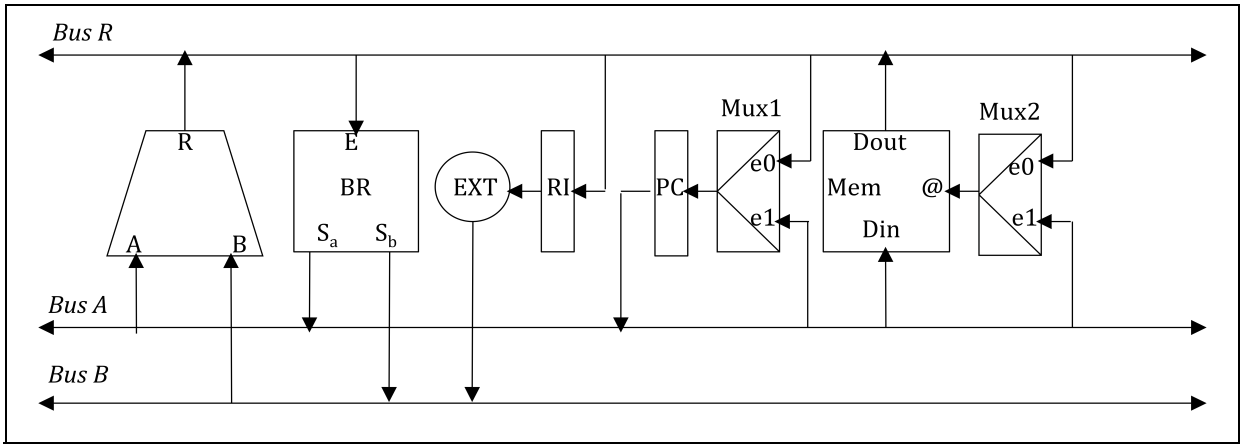

Figure 1 : Chemin de données

	UAL	Bus A	Bus B	Bus R	Ecr Regs	Mux1	Mux2	Mem

Table 2 : logique de commandes