

# Partiel Architecture des Ordinateurs - 20 Octobre 2015

---

**Tous documents autorisés. Calculatrices interdites.**  
**Durée 2h**

Une justification **concise** sera donnée pour chaque question. Lorsqu'un résultat est demandé en notation hexadécimale, la notation binaire ne sera pas acceptée.

## 1. Représentation des entiers [2 pts]

**Q1.** On note # l'opération effectuée par un additionneur sur 8 bits. Pour les opérations suivantes, donner le résultat en notation **hexadécimale**, donner la retenue, et indiquer si le résultat est égal à celui de l'opération arithmétique d'addition lorsque les opérandes sont interprétés en naturels et en relatifs (complément à deux), suivant le format de la table.

Opération	Résultat	Retenue (0/1)	Correct en naturels (Oui/Non)	Correct en relatifs (Oui/Non)
0x45 # 0x74				
0xF0 # 0x75				
0xF1 # 0xA4				
0x34 # 0x45				

## 2. Jeu d'instructions MIPS [6 pts]

Pour Q2, Q3, Q5, les résultats seront donnés en notation **hexadécimale**.

**Q2.** Donner le codage des instructions :

a) ADDIU R10, R1, 9    b) SUB R3, R1, R2    c) LW R2, 4(R1)

**Q3.** L'état initial des registres est : R2 = 0xA7654321 et R3 = 0x0000000F. Donner le contenu du registre R1 après l'exécution des instructions :

a) ADD R1, R2, R3    b) ADDIU R1, R2, 0xFFFF    c) XORI R1, R2, 0xF0F0  
d) SRA R1, R2, 8    e) SLT R1, R2, R3    f) SLTU R1, R2, R3

Remarque : ces instructions ne modifient pas R2 et R3.

**Q4.** Initialement R2 = 0x81234567. Donner un exemple de contenu de R3 tel que l'exécution des instructions ADD R1, R2, R3 et ADDU R1, R2, R3 fournisse des résultats différents.

**Q5.** La mémoire est organisée en Big Endian. L'état de la mémoire est donné par la table 1. Le registre R1 contient 0x10000008. Donner le contenu du registre R2 après l'exécution des instructions :

a) LW R2, -4(R1)    b) LW R2, -8(R1)    c) LB R2, -7(R1)    d) LHU R2, -2(R1)

Adresse	Valeur
0x10000000	0x85
0x10000001	0xB0
0x10000002	0x12
0x10000003	0x03
0x10000004	0xAA
0x10000005	0xBB
0x10000006	0xCC
0x10000007	0xDD

Table 1 – Plan mémoire

### 3. Programmation [10 pts]

On utilise le jeu d'instruction MIPS. On rappelle que le registre R0 est câblé à 0.

**Q6.** Si R1 contient initialement 0x00000003, quelle est la valeur (notation **décimale**) de R2 après l'exécution de Prog1 ? Quelle est l'opération arithmétique réalisée par Prog1 ?

Prog1
SLL R2, R1, 4
ADDU R2, R2, R1
SLL R2, R2, 2
ADDU R2, R2, R1

**Q7.** Le tableau d'entiers X est implanté à partir de l'adresse 0x10000000, et contient {1,2,3,4,5,6,7,8}. Le tableau d'entiers Y est implanté à partir de l'adresse 0x20000000. L'état initial est : R10 = 0x10000000, R11 = 0x20000000.

Pour cette question et la suivante, les entiers sont codés sur 32 bits.

Donner le contenu (notation **décimale**) du tableau Y après l'exécution de Prog2

**Q8.** Le tableau d'entiers X est implanté à partir de l'adresse 0x10000000, et contient {3,3,7,4,8,8,8,2}. L'état initial est : R10 = 0x10000000, R3 = R4 = 0x00000001

	Prog2		Prog3
	ADDI R12, R10, 32		ADDI R12, R10, 28
	ADDI R2, R0, 0		LW R1, 0(R10)
bcl :	LW R1, 0(R10)	Deb :	LW R2, 4(R10)
	ADD R2, R2, R1		ADDI R4, R4, 1
	SW R2, 0(R11)		BEQ R1, R2, Choix
	ADDI R10, R10, 4		ADDI R4, R0, 1
	ADDI R11, R11, 4	Choix:	SLT R5, R4, R3
	BNE R10, R12,bcl		MOVZ R3, R4, R5
			ADDI R1, R2, 0
			ADDI R10, R10, 4
			BNE R10, R12, bcl

**a)** Donner la valeur (notation **décimale**) des registres R1, R2, R3 et R4 après la première itération de Prog3. Même question après la deuxième itération.

**c)** Donner la valeur (notation **décimale**) du registre R3 après l'exécution de Prog3.

**d)** Que fait ce programme ?

### 4. Mémoire [2 pts]

**Q9.** On considère la déclaration C suivante :

```
char c[3];
double y[2] ;
short x ;
int* z ;
```

Le placement est aligné et en big endian. Les variables sont allouées dans l'ordre à partir de l'adresse 0x00001000. Donner l'adresse (notation **hexadécimale**) de la variable c[2], de la variable x et de la variable z.