

NOM : B A R A I S  
 PRENOM : O L I V I E R  
 INE : 2 6 0 1 1 8 8 0



NOM DU MODULE

Février 2022

- Durée de l'épreuve : 1h00 heures
- Polycopiés de cours autorisés
- Calculatrices non connectées autorisées.
- Le barème est donné à titre purement indicatif.

Les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.

### Exercice 1 : Jeu d'instructions Nios II (5 points)

On s'intéresse à un processeur NIOS-II qui exécute les instructions d'un programme représenté ci-dessous. L'état de la mémoire au début de l'exécution du programme est représenté dans le tableau ci-après.

```
movia r2, 0x12ABCD
addi r3, zero, 0x8FFF
stb r2, 0x6(zero)
ldh r4, 0x2(zero)
srai r5, r3, 8
```

	00	01	10	11
0x000	0xF1	0x23	0xC3	0x84
0x004	0x40	0x53	0x9D	0x62

Q1 : Complétez les tables ci-dessous en précisant les valeurs des registres ainsi que l'état de la mémoire à l'issue de l'exécution de ce programme.

	00	01	10	11
0x000				
0x004				

  

r2	Scal pas
r3	
r4	
r5	

Q2 : Pouvait-on utiliser l'instruction **movi** à la place de **movia** dans le programme ci-dessus ? Si non, expliquez pourquoi.

Non

Q3 : Que se passe-t-il lors de l'exécution de l'instruction **ldw r4, 0x1(zero)** ? Expliquez pourquoi.

Oui

## Exercice 2 : programmation assembleur (10 points)

On s'intéresse ici à la traduction du code C ci-dessous vers le langage machine NIOS II

```
struct node {
    int* data;
    struct node* next;
};

int tab[] = {1,45,234};
struct node root = { .data=tab, .next=NULL};
struct node* current = &root;
```

**Q1** : Donnez la taille (en octets) de l'espace mémoire occupé par chacune des variables globales **tab**, **root** et **current** (vous justifierez votre réponse).

tab

**Q2** : Complétez les directives d'assemblage NIOS-II permettant d'allouer et d'initialiser les variables globales **tab**, **root** et **current**. Déduisez en les valeurs associées aux identificateurs **\_tab**, **\_root**, **\_current** ainsi que le contenu de la mémoire au début du programme.

.data	Ident	valeur	0x0	0x1	0x2	0x3
_tab:	_tab		0x000			
	_root		0x004			
_root:	_current		0x008			
			0x00C			
_current:			0x010			
			0x014			

**Q3** : Traduisez (en langage d'assemblage NIOS-II) l'instruction ci-dessous.

current=NULL;	mul
---------------	-----

**Q4** : Traduisez (en langage d'assemblage NIOS-II) l'instruction ci-dessous (on fera l'hypothèse que la valeur associée à **\_current** est codée sur 16 bits)

root.next = current;	bar
----------------------	-----