

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 1

NOM : PRENOM : SECTION :

Consignes à lire impérativement !

L'examen est composé de **2 parties**. Chaque partie dure **2 heures**. Il vous est demandé de respecter les consignes suivantes.

- Commencez par écrire vos **nom, prénom et section** (math, info, ...) sur chaque feuille, y compris les feuilles de brouillon.
- Laissez vos calculatrice, téléphone portable et notes de cours dans votre sac. Leur usage n'est **pas autorisé**. Pensez à éteindre votre téléphone portable !
- Faites attention à la clarté et à l'organisation de vos réponses. Respectez les règles grammaticales et orthographiques.
- Utilisez pour vos réponses les **cadres** prévus à cet effet. Si davantage d'espace est nécessaire, utilisez le dos de la feuille ou une feuille supplémentaire et indiquez clairement où se situe le restant de la réponse.
- Vous devez terminer cette partie de l'examen avant de pouvoir sortir de la salle (pour aller à la toilette par exemple).
- Toutes les feuilles (énoncé et brouillon) doivent être remises en fin d'examen.
- Vérifiez que vous avez répondu à toutes les questions (il y a **2 questions** dans cette partie).

Question 1 – Comparateur (/4)

Cette question vise à implémenter un circuit logique permettant de comparer entre eux deux nombres entiers a et b . Le comparateur dispose à cet effet de $2N$ entrées booléennes correspondant aux deux nombres représentés chacun sur N bits en complément à deux. Ces entrées sont nommées $a_{N-1} \dots a_0$ pour les bits du nombre a et $b_{N-1} \dots b_0$ pour les bits du nombre b . Le comparateur dispose également de deux sorties booléennes définies comme suit :

- eq indique si les deux nombres sont égaux, i.e. $eq \Leftrightarrow (a = b)$
- $less$ indique si a est strictement inférieur à b , i.e. $less \Leftrightarrow (a < b)$

Afin de limiter le nombre d'entrées et la complexité du circuit logique du comparateur, la question considère des nombres représentés sur $N = 2$ bits. Il dispose donc de 4 entrées a_1, a_0, b_1 et b_0 .

Ce qui vous est demandé :

- Q1a Etablir la table de vérité de $less$ et eq en fonction des entrées a_1, a_0, b_1 et b_0 . La même table est utilisée pour les 2 sorties ($less$ et eq). Dans la colonne intitulée « valeur », indiquez la valeur des nombres représentés en binaire.
- Q1b Dérivez une expression logique pour $less$ et eq . Il vous est conseillé d'utiliser pour cela les sommes ou produits canoniques. A vous de choisir entre somme et produit, de façon à minimiser le nombre de termes de l'expression.

Partie 1

SECTION :

Q1c Fournissez le schéma du circuit logique correspondant au test d'égalité. Ce circuit logique possèdera 4 entrées a_1 , a_0 , b_1 et b_0 ainsi qu'une sortie unique eq . Veuillez à utiliser les symboles standards pour exprimer les portes logiques AND, OR et NOT. Veuillez également à la propreté et la clarté de votre schéma.

(table de vérité)

Nombre a			Nombre b			$less$ ($a < b$)	eq ($a = b$)
valeur	a_1	a_0	valeur	b_1	b_0		

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 1

NOM : PRENOM : SECTION :

Q1b

(expression logique pour *less* et *eq*)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q1c

(circuit logique pour *eq*)

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 1

NOM : PRENOM : SECTION :

Question 2 – Suite de Fibonacci en MIPS (/6)

L'objectif de cette question est le calcul des nombres de Fibonacci. Pour rappel, les nombres de Fibonacci sont définis par la relation récursive $\forall n > 1, F_n = F_{n-1} + F_{n-2}$ où F_n est appelé $n^{\text{ième}}$ nombre de Fibonacci. Les valeurs de base sont $F_0 = 0$ et $F_1 = 1$. A titre d'exemple, la séquence suivante contient les nombres de Fibonacci de 0 à 9 : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Un programme calculant le $n^{\text{ième}}$ nombre de Fibonacci de manière récursive peut être facilement implémenté dans un langage de haut-niveau. La Figure 1 donne un exemple d'un tel programme en langage C.

```
1 | int fib(int n)
2 | {
3 |     if (n <= 1)
4 |         return n;
5 |     return fib(n - 1) + fib(n - 2);
6 | }
```

FIGURE 1 – Fonction en langage C calculant de façon récursive le $n^{\text{ième}}$ nombre de Fibonacci.

Ce qui vous est demandé :

Il vous est demandé de fournir en langage d'assemblage MIPS les implémentations des fonctions suivantes.

- Q2a Une implémentation en langage d'assemblage MIPS d'un programme appelant la fonction `fib` avec l'argument n valant 6. Le point d'entrée de ce programme est `main`. La valeur retournée par la fonction `fib` doit être affichée avec un appel système effectué avec l'instruction `syscall`. Le numéro d'appel système (4) doit être passé dans le registre `v0` tandis que l'entier à afficher doit être passé dans le registre `a0`.
- Q2b Une implémentation en langage d'assemblage MIPS de la fonction `fib` donnant le $n^{\text{ième}}$ nombre de Fibonacci.
Attention ! Cette fonction doit absolument être récursive !

Il vous est demandé de respecter la convention de passage de paramètres suivante. L'argument n de votre fonction est passé dans le registre `a0`. La valeur de retour de votre fonction est retournée dans le registre `v0`.

Une liste d'instructions MIPS pouvant vous être utile est fournie en annexe.

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 1

NOM : PRENOM : SECTION :

NOM :	PRENOM :	SECTION :
-------------	----------------	-----------------

[illegible][illegible]

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 1

NOM : PRENOM : SECTION :

NOM :	PRENOM :	SECTION :
-------------	----------------	-----------------

[illegible][illegible]

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 2

NOM : PRENOM : SECTION :

Consignes à lire impérativement !

L'examen est composé de **2 parties**. Chaque partie dure **2 heures**. Il vous est demandé de respecter les consignes suivantes.

- Commencez par écrire vos **nom, prénom et section** (math, info, ...) sur chaque feuille, y compris les feuilles de brouillon.
- Laissez vos calculatrice, téléphone portable et notes de cours dans votre sac. Leur usage n'est **pas autorisé**. Pensez à éteindre votre téléphone portable !
- Faites attention à la clarté et à l'organisation de vos réponses. Respectez les règles grammaticales et orthographiques.
- Utilisez pour vos réponses les **cadres** prévus à cet effet. Si davantage d'espace est nécessaire, utilisez le dos de la feuille ou une feuille supplémentaire et indiquez clairement où se situe le restant de la réponse.
- Vous devez terminer cette partie de l'examen avant de pouvoir sortir de la salle (pour aller à la toilette par exemple).
- Toutes les feuilles (énoncé et brouillon) doivent être remises en fin d'examen.
- Vérifiez que vous avez répondu à toutes les questions (il y a **3 questions** dans cette partie).

Question 1 – Cache *set-associative* (/5)

Soit un système informatique composé d'une mémoire RAM et d'une cache.

- La mémoire RAM a une taille de 65536 octets organisés en 65536 mots de 8 bits.
- La cache a une taille de 16 octets organisés en 4 lignes de 4 octets. La cache est de type *set-associative*. Son degré d'associativité est 2 (elle contient par conséquent 2 sets). La stratégie de remplacement de la cache est *least recently used* (LRU) : la ligne remplacée est celle dont le dernier usage est le plus éloigné dans le temps.

L'objectif de cette question est de déterminer le *hit ratio* de cette cache pour la séquence de lectures aux adresses suivantes : 0xA3C8, 0xA3C9, 0xA3CA, 0xB5E0, 0xB5E4, 0xB5E8, 0xB5EC, 0xA3C8. On considère que la cache est initialement vide.

Il vous est demandé de :

Q1a Indiquer la correspondance entre une adresse mémoire (16 bits) et un emplacement en cache, en identifiant les bits de l'adresse qui correspondent au *tag*, au *set* et à l'*offset*. A cet effet, remplissez chaque case (bit) de l'adresse par la lettre T (*tag*) , S (*set*) ou O (*offset*).

Q1b Exécuter manuellement la séquence de lectures. Pour chaque lecture

- donnez les valeurs de *tag*, *set* et *offset*, en hexadécimal.

Examen du cours de Fonctionnement des Ordinateurs

1^{ère} Session, Juin 2016

Partie 2

NOM : PRENOM : SECTION :

- indiquez s'il s'agit d'un *hit* ou d'un *miss*. Si une ligne est remplacée, indiquez laquelle (identifiez la avec *set* et *tag* actuel).

Q1c Calculer le *hit ratio* de la séquence de lectures.

Q1d Calculer le temps maximum nécessaire pour effectuer une lecture en mémoire cache pour que le système RAM/-cache soit efficace. On vous indique à cet effet que la lecture en RAM d'une ligne de cache (4 octets) nécessite 80ns. Indiquez votre raisonnement !!!

Q1a (structure d'une adresse)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Q1b (rejeu de la trace d'accès)

Adresse	Tag	Set	Offset	Hit / Miss
0xA3C8				
0xA3C9				
0xA3CA				
0xB5E0				
0xB5E4				
0xB5E8				
0xB5EC				
0xA3C8				

Q1c (*hit ratio*)

.....

Q1d (temps maximum de lecture en cache)

.....

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 2

NOM : PRENOM : SECTION :

Question 2 – Offrez une pause à votre processeur (/2)

L'objectif de cette question est de produire une suite d'instructions MIPS attendant le plus précisément possible un délai donné en millisecondes. Ce délai est égal au temps d'exécution par le processeur de ces instructions. Cette suite d'instructions sera exécutée sur un processeur de fréquence $f = 100\text{MHz}$. On considère que le processeur exécute une instruction par cycle d'horloge. Toutes les instructions nécessitent un seul cycle.

Ce qui vous est demandé :

La séquence d'instructions permettant d'attendre un délai prescrit. La suite d'instructions fournie ne peut utiliser ni timer ni interruption. Le délai à attendre sera placé auparavant dans le registre `a0`.

Veillez à écrire votre séquence d'instructions le plus clairement et concisement possible. Un code simple sera mieux noté qu'un code (inutilement) compliqué.

Q2

(séquence d'instructions permettant d'attendre un délai prescrit)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 2

NOM : PRENOM : SECTION :

Question 3 – Représentation IEEE 754 (/3)

L'objectif de cette question est de déterminer si vous avez bien compris le fonctionnement de la représentation de nombres flottants suivant le format standard IEEE 754. La question se focalise sur la représentation en simple précision. Pour rappel, dans ce format, un nombre flottant est représenté sur 32 bits : 1 bit de signe s , un exposant e de 8 bits et une mantisse m de 23 bits. La représentation utilise un biais B égal à 127 pour les nombres normalisés et un exposant fixe égal à -126 pour les nombres dénormalisés (lorsque e vaut 0).

On considère les deux nombres suivants donnés selon la représentation IEEE 754 simple précision.

	s	e	m
x	1	10000000	00000000000000000000001
y	1	01101000	00000000000000000000000

Il vous est demandé de :

- Q3a Donner la valeur du nombre x sous forme d'une fraction dont le numérateur est une somme d'exposants de 2.
- Q3b Donner la valeur du nombre y sous forme d'une fraction dont le numérateur est une somme d'exposants de 2.
Utiliser le même dénominateur que pour x .
- Q3c Effectuer l'addition de x et y et donner la représentation IEEE 754 du résultat.

Veuillez justifier vos réponses. Une réponse sans justification sera considérée **incorrecte**.

Q3a

(valeur de x)

.....

.....

.....

Q3b

(valeur de y)

.....

.....

.....

Examen du cours de Fonctionnement des Ordinateurs
1^{ère} Session, Juin 2016
Partie 2

NOM : PRENOM : SECTION :

Q3c

(représentation de $x + y$)

.....

.....

.....

.....

.....

.....

.....

.....