

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
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 **3 questions** dans cette partie).

Question 1 – Performance de la mémoire cache (/3)

Cette question s'intéresse à l'impact du type et des caractéristiques d'une cache sur les performances atteinte par un programme. A cette fin, on considère le programme MIPS de la Figure 1. En exécutant pas à pas les instructions de ce programme, on souhaite simuler le comportement de la cache et déterminer notamment le *miss rate*.

```
1 |      addi $t0, $zero, 5
2 | loop:
3 |      beq $t0, $zero, done
4 |      lw $t1, 0x4($zero)
5 |      lw $t2, 0x24($zero)
6 |      addi $t0, $t0, -1
7 |      j loop
8 | done:
```

FIGURE 1 – Programme MIPS.

Deux caches différentes sont considérées dont les caractéristiques sont données ci-dessous. Ces caches sont supposées initialement vides.

- **CACHE 1** : une cache *direct-mapped* de 32 octets organisée en 8 lignes de 4 octets
- **CACHE 2** : une cache *2-way-associative* de 32 octets contenant des lignes de 4 octets.

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 1

NOM : PRENOM : SECTION :

Il vous est demandé de fournir les mesures suivantes pour l'exécution du programme de la Figure 1 sur les deux caches décrites ci-dessus.

1. le nombre d'accès se traduisant par un *hit* ;
2. le nombre d'accès se traduisant par un *miss* – on demande en outre de distinguer entre *compulsory*, *capacity* et *conflict misses* ;
3. le *miss rate*.

Q1a

(CACHE 1 : *direct-mapped*)

| | |
|-------------------------|--|
| Nombre de hits | |
| Nombre de <i>misses</i> | |
| <i>compulsory</i> | |
| <i>conflict</i> | |
| <i>capacity</i> | |
| <i>Miss rate</i> | |

Q1b

(CACHE 2 : *2-way-associative*)

| | |
|-------------------------|--|
| Nombre de hits | |
| Nombre de <i>misses</i> | |
| <i>compulsory</i> | |
| <i>conflict</i> | |
| <i>capacity</i> | |
| <i>Miss rate</i> | |

Examen du cours de Fonctionnement des Ordinateurs

2^{ème} Session, Août 2019

Partie 1

NOM : PRENOM : SECTION :

Question 2 – Timer (/4)

Cette question s'intéresse au périphérique « timer » utilisé lors des travaux pratiques. La Figure 2 montre le schéma-bloc du **TIMER1** d'un microcontrôleur de la famille PIC32MX. Sur ce schéma, il est facile de repérer plusieurs registres qui contrôlent le fonctionnement du timer. Premièrement, **TMR1** est un registre compteur qui s'incrémente au fur et à mesure que le timer reçoit des impulsions d'horloge. Deuxièmement, **PR1** est un registre qui fixe la période du timer. Lorsque **TMR1** et **PR1** sont égaux (ils sont comparés entre eux en permanence), **TMR1** est remis à zéro (*Reset*) et un événement d'expiration du timer est généré (**T1IF**). Les registres **TMR1** et **PR1** sont chacun représentés sur 16 bits.

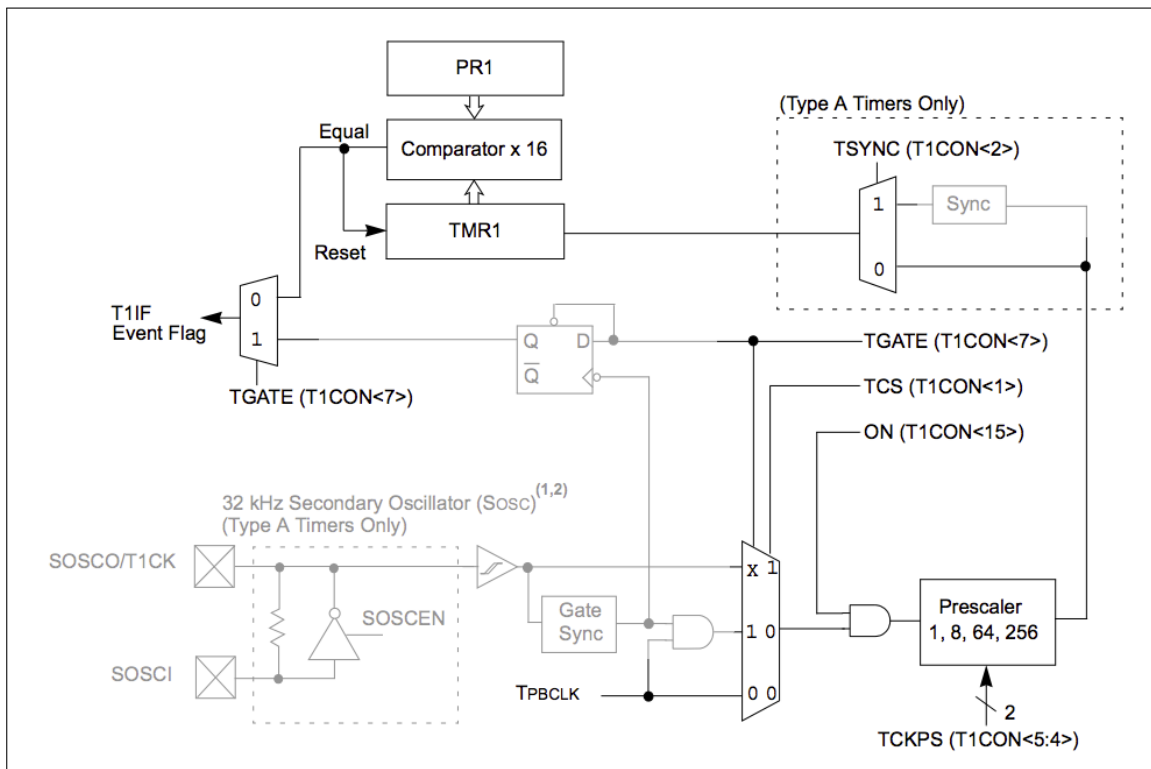


FIGURE 2 – Schéma-bloc du **TIMER1** d'un microcontrôleur de type PIC32MX (© Microchip).

Le registre **T1CON** permet de contrôler le timer. Le bit **TCS** par exemple permet de sélectionner quel signal d'horloge fait incrémenter le timer. Le bit **ON** permet d'activer ou désactiver le timer. Les bits **TCKPS** permettent de pré-diviser par 1, 8, 64 ou 256 le signal d'horloge sélectionné.

Ce qui vous est demandé : En supposant que le signal d'horloge sélectionné soit T_{PBCLK} , que sa fréquence soit égale à 16 MHz, et en utilisant les paramètres du timer, calculez les durées suivantes :

Q2a T_{max} , la plus longue durée d'expiration du timer.

Q2b T_{min} , la plus petite durée non-nulle d'expiration du timer.

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 1

NOM : PRENOM : SECTION :

Q2a

(T_{\max} , durée la plus longue)

.....

.....

.....

.....

.....

Q2b

(T_{\min} , durée la plus courte, non-nulle)

.....

.....

.....

.....

.....

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 au 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.

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 1

NOM : PRENOM : SECTION :

Q3a

(valeur de x)

.....

.....

.....

.....

.....

Q3b

(valeur de y)

.....

.....

.....

.....

.....

Q3c

(représentation de $x + y$)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
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 **2 questions** dans cette partie).

Question 1 – Mémoire virtuelle (/3)

Soit un système informatique capable d'adresser 4 Go de mémoire virtuelle et doté d'une mémoire physique de 256 Mo. La mémoire est découpée en pages de 16 Ko. Pour rappel, 1 Ko = 1024 octets = 2^{10} octets, 1 Mo = 2^{20} octets et 1 Go = 2^{30} octets.

Ce qui vous est demandé : déterminer les caractéristiques de la mémoire virtuelle.

1. Quelle est la taille d'une adresse virtuelle ?
2. Quelle est la taille (en bits) de l'offset à l'intérieur d'une page ?
3. Combien y a-t-il de pages physiques ?
4. Combien y a-t-il de pages virtuelles ?
5. Quelle est la taille (en bits) d'un PPN (*physical page number*) ?
6. Quelle est la taille (en bits) d'un VPN (*virtual page number*) ?
7. Quelle est la taille (en octets) de la table des pages ? On considère que chaque entrée est stockée sur un nombre entier d'octets.

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 2

NOM : PRENOM : SECTION :

Q1

(Caractéristiques mémoire virtuelle)

| | |
|---|--|
| Taille adresse virtuelle (en bits) | |
| Taille offset (en bits) | |
| Nombre pages physiques | |
| Nombre pages virtuelles | |
| Taille PPN (en bits) | |
| Taille VPN (en bits) | |
| Taille d'une entrée de la table des pages (en octets) | |
| Taille de la table des pages (en octets) | |

Question 2 – Relation de récurrence en MIPS (/7)

L'objectif de cette question est de fournir l'implémentation en langage d'assemblage MIPS d'une fonction **réursive** qui calcule la relation de récurrence $f(n)$ montrée ci-dessous. A titre d'exemple, si n vaut 3, $f(n)$ devrait retourner 16.

$$f(n) = \begin{cases} 2f(n-1) + 3f(n-2) + 1 & , \text{ si } n > 1 \\ 1 & , \text{ si } n = 0 \text{ ou } n = 1 \end{cases}$$

La fonction sera appelée `eval_rec` et suivra la convention d'appel suivante. Le registre `a0` contiendra l'argument n , un entier non-signé représenté sur 32 bits. Le registre `v0` contiendra la valeur de retour de la fonction, un entier non-signé représenté sur 32 bits.

De plus, pour pouvoir évaluer cette fonction sur de nombreuses valeurs, le programme principal (`main`) sera chargé de demander à l'utilisateur un entier, de calculer $f(n)$, d'afficher la valeur obtenue à la console et de recommencer. Si l'entier fourni par l'utilisateur est négatif, alors le programme s'arrête. Pour demander un entier à l'utilisateur, pour afficher un entier à la console voire pour afficher une chaîne de caractères, un appel système sera utilisé (instruction `syscall`). La Table 1 résume les appels systèmes dont vous pourriez avoir besoin.

Il vous est demandé de fournir

Q2a l'implémentation de la fonction `main`.

Q2b l'implémentation de la fonction `eval_rec`

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 2

NOM : PRENOM : SECTION :

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

| Appel système (v0) | Argument | Résultat | Description |
|-----------------------|----------|----------|--|
| 1 | a0 | — | Affiche l’entier donné par a0 |
| 4 | a0 | — | Affiche la chaîne située à l’adresse donnée par a0 |
| 5 | — | v0 | Demande à l’utilisateur un entier et le retourne dans v0 |

TABLE 1 – Résumé des appels systèmes utiles. Le registre `v0` contient le numéro d'appel système. L'instruction `syscall` est utilisée pour effectuer l'appel.

Q2a

```
(main)
```


Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 2

NOM : PRENOM : SECTION :

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

[illegible][illegible]

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 2

NOM : PRENOM : SECTION :

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

[illegible][illegible]

Examen du cours de Fonctionnement des Ordinateurs
2^{ème} Session, Août 2019
Partie 2

NOM : PRENOM : SECTION :

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

[illegible][illegible]