



Examen – Architecture des ordinateurs

(Durée 1h30mn)

Le 30 /03 /2021

Questions de cours (sur 4 points)

1. Expliquer les termes « **Compiler** » et « **Assembler** » dans le cadre de l'architecture des ordinateurs.
2. Le **traitement d'une instruction**, par l'**unité de commande**, peut être décomposé en **trois phases** principales. Expliquer brièvement la **première phase**.
3. Quelle est la différence entre **SRAM** et **SDRAM**

Exercice 1 TD : (Test TD sur 6 points)

I/ Soit un ordinateur à architecture bus système composé d'un processeur travaillant à une **fréquence de 800 MHz** et d'une **mémoire centrale de 256 Méga mot** mémoire de **deux octets**. Son **registre d'instruction** se compose de **16 bits**.

- 1- Calculer la **capacité** maximale de la mémoire (en Octets).
- 2- Trouver la taille (en Bits) de : **RAM** (Registre d'Adresse Mémoire), **RDM** (Registre de Données Mémoire), **CO** (Compteur Ordinal) et **ACC** (Accumulateur).
- 3- Calculer le **taux de transfert** (théorique) d'un bus de donnée d'un processeur.

II/ Soit une mémoire cache à deux niveaux ayant les caractéristiques suivantes :

| Niveaux | Temps d'accès succès (ns) | Taux d'échec | Pénalité d'échec(Cycles) | Taille | Nombre de mots par ligne | Taille d'un mot |
|----------|---------------------------|--------------|--------------------------|--------|--------------------------|-----------------|
| Cache L1 | 2 | 20% | 5 | 128 Ko | 32 mots | 4 Octets |
| Cache L2 | 4 | 10% | 10 | 2 Mo | 32 mots | 4 Octets |

- 1- Calculer la **taille d'une ligne** ? Combien y-a-t-il de **lignes** dans cette mémoire cache ?
- 2- Combien y-a-t-il de blocs de la mémoire **cache L2** par bloc de la mémoire **cache L1** ?
- 3- Calculer le **temps d'exécution moyen d'une instruction** pour chaque niveau sachant que durée d'un cycle horloge est **T**.

Exercice 2 TD : (sur 6 points)

I/ L'ordinateur dont il est question ici possède une architecture dont les instructions machines possèdent un **seul opérande (machine à une adresse)** et dont le mode d'adressage est **immédiat**. Les mnémoniques à considérer, pour les instructions arithmétiques, sont ADD- SUB- MPY et DIV, et pour les instructions de manipulation de données LOAD et STORE.

- 1- Donner le code nécessaire pour évaluer l'expression : $Z = (A+B)/((C-D)*E)$ en utilisant le minimum d'instruction possible.
- 2- En supposant que l'architecture de la machine est à **pile**, donner le code pour évaluer la même expression et ce en utilisant le minimum d'instruction possible.

II/ Parfois, l'optimisation des logiciels peut considérablement améliorer les performances d'un système informatique. Supposons qu'un **processeur** peut effectuer une opération de **multiplication** en **10 ns**, et une opération de **soustraction** en **1 ns**.

- 1- Combien de temps** faut-il au **CPU** pour calculer le résultat de l'opération $d = a \times b - a \times c$?
- 2- Pourriez-vous optimiser l'équation** pour qu'elle prenne moins de temps ?

Exercice TP : (sur 4 points)

Ecrire en assembleur MIPS le code correspondant au fragment du programme suivant :

if t1 < t2 then t3 := t2 / t1 else t3 := t1 / t2

Bon courage

Corrigé Examen – Architecture des ordinateurs

(du 30 /03 / 2021)

Questions de cours (sur 4 points)

1. Expliquer les termes « **Compiler** » et « **Assembler** » dans le cadre de l'architecture des ordinateurs. (1 point)

Le programme en langage de haut niveau n'est donc pas compréhensible par le microprocesseur. Il faut le **compiler** pour le traduire en assembleur puis l'**assembler** pour le convertir en code machine compréhensible par le microprocesseur.

2. Le **traitement d'une instruction**, par l'**unité de commande**, peut être décomposé en **trois phases** principales. Expliquer brièvement la **première phase**. (2 points = 0.5 * 4)

Phase 1 : Recherche de l'instruction à traiter

1.1 Mettre le contenu du **CO** dans le **registre d'adresse mémoire (R@M)**. // CO qui contient l'adresse de l'instruction suivante. (CO) → R@M

1.2 Commande de **lecture** à partir de la mémoire

1.3 **Sélection de l'instruction** (Adresse d'instruction) et **son contenu** est transféré vers le **registre de Données Mémoire (RDM)**. // au bout d'un certain temps (temps d'accès à la mémoire) (@inst) → RDM

1.4 **Transfert du contenu du RDM** dans le **registre instruction (RI)** du processeur. (RDM) → RI

3. Quelle est la différence entre **SRAM** et **SDRAM** (1 point = 0.5 * 2)

SRAM (Static Random Access Memory) est une RAM statique utilisée dans les mémoires caches et **SDRAM (Synchrone Dynamic Random Access Memory)** est une RAM dynamique utilisé dans la mémoire centrale.

Exercice 1 TD : (Test TD sur 6 points)

I/ Soit un ordinateur à architecture bus système composé d'un processeur travaillant à une **fréquence** de **800 MHz** et d'une **mémoire centrale** de **256 Méga mot** mémoire de **deux octets**. Son **registre d'instruction** se compose de **16 bits**.

- 1- Calculer la **capacité** maximale de la mémoire (en Octets). (0.75 point)

Capacité=Nombre de mots*taille du mot(0.25)=256*2=512 MOct=2⁹*2²⁰ Octets=2²⁹ Octets ou 2³² bits (0.5)

- 2- Trouver la taille (en Bits) du **RAM** (Registre d'Adresse Mémoire), **RDM** (Registre de Données Mémoire), **CO** (Compteur Ordinal) et **ACC** (Accumulateur). (1.5 point)

RAM= Nbre de ligne d'adresse

→ **Nombre de mots=2^{nombre de lignes d'adresse} (0.25) = 256 Méga = 2⁸*2²⁰ = 2²⁸ donc RAM = 28 bits (0.5)**

RDM =taille mot = 2 octets = 16 bits (0.25)

CO=Nbre de ligne d'adresse= 28 bits (0.25)

ACC= Nbre de ligne de données = 16 bits (0.25)

- 3- Calculer le **taux de transfert** (théorique) d'un bus de donnée d'un processeur. (1 point)

Taux de transfert (Mo/s) = Fréquence (en MHz) × Largeur du bus (en octets) (0.25)

Largeur du bus = Taille d'un bus de données = taille d'un mot= 2 octets (0.25)

Taux de transfert (Mo/s) = 800 * 2 = 1600 Mo/s (0.5)

Ou Largeur du bus = 2 octets = 16 bits

Taux de transfert (Mbits/s) = 800 * 16 = 12800 Mbits/s

II/ Soit une mémoire cache à deux niveaux ayant les caractéristiques suivantes :

| Niveau | Temps d'accès succès (ns) | Taux d'échec(ns) | Pénalité d'échec(Cycles) | Taille | Nombre de mots par ligne | Taille d'un mot |
|----------|---------------------------|------------------|--------------------------|--------|--------------------------|-----------------|
| Cache L1 | 2 | 20% | 5 | 128 Ko | 32 mots | 4 Octets |
| Cache L2 | 4 | 10% | 10 | 2 Mo | 32 mots | 4 Octets |

1- Calculer la **taille d'une ligne** ? Combien y-a-t-il de **lignes** dans cette mémoire cache ? **(1.5point)**

Taille d'une ligne (L1 et L2) = Taille mot* Nombre de mots par ligne (0.25)

$$= 32 * 4 = 2^5 * 2^2 = 2^7 \text{ Octets ou } 2^{10} \text{ bits (0.5)}$$

Nombre de lignes = Taille cache / Taille d'une ligne (0.25)

Nombre de lignes (L1) = 128 KOct / 2^7 Oct = 2^7 * 2^{10} / 2^7 = 2^{10} Lignes (0.25)

Nombre de lignes (L2) = 2 MOct / 2^7 Oct = 2 * 2^{20} / 2^7 = 2^{14} Lignes (0.25)

2- Combien y-a-t-il de blocs de la mémoire **cache L2** par bloc de la mémoire **cache L1** ? **(0.5 point)**

Taille cache L2 / Taille cache L1 (0.25) = 2 MOct/128 KOct = 2^{21}/2^{17} = 2^4 = 16 (0.25)

3- Calculer le **temps d'exécution moyen d'une instruction** pour chaque niveau sachant que durée d'un cycle horloge est **T**. **(0.75 point)**

Temps d'accès mémoire moyen = Temps d'accès succès + Taux d'échec x Pénalité d'échec (0.25)

Temps d'accès mémoire moyen Cache L1 = 2 + (20%) x 5 = 2 + (0.2) x 5 = 3T (0.25)

Temps d'accès mémoire moyen Cache L2 = 4 + (10%) x 10 = 4 + (0.1) x 10 = 5T (0.25)

Exercice 2 TD : (sur 6 points)

I/ L'ordinateur dont il est question ici possède une architecture dont les instructions machines possèdent un **seul opérande (machine à une adresse)** et dont le mode d'adressage est **immédiat**. Les mnémoniques à considérer, pour les instructions arithmétiques, sont ADD- SUB- MUL et DIV, et pour les instructions de manipulation de données LOAD et STORE.

1- Donner le code nécessaire pour évaluer l'expression : $Z = (A+B)/((C-D)*E)$ en utilisant le minimum d'instruction possible. **(2 points = 0.25 * 8)**

LOAD C; [Acc] ← C
SUB D; [Acc] ← C-D
MPY E; [Acc] ← (C-D)*E
STORE Z; [Z] ← (C-D)*E
LOAD A; [Acc] ← A
ADD B; [X] ← A+B
DIV Z; [Acc] ← (A+B)/((C-D)*E)
STROE Z; [Z] ← [Acc]
[Z] ← (A+B)/((C-D)*E)

2- En supposant que l'architecture de la machine est à **pile**, donner le code pour évaluer la même expression et ce en utilisant le minimum d'instruction possible. **(2.5 points = 0.25 * 10)**

PUSH A; Pile = {A}
PUSH B; Pile = {A; B}

ADD; Pile = { A+B }
 PUSH C; Pile = { (A+B); C }
 PUSH D; Pile = { (A+B); C; D }
 SUB Pile = { (A+B); C-D }
 PUSH E; Pile = { (A+B); (C-D); E }
 MPY; Pile = { (A+B); (C-D)*E }
 DIV; Pile = { (A+B) / ((C-D)*E) }
 POP Z; Pile = { } [Z] ← (A+B)/((C-D)*E)

Ou

PUSH A; Pile = { A }
 PUSH B; Pile = { A; B }
 ADD; Pile = { A+B }
 PUSH E; Pile = { (A+B); E }
 PUSH C; Pile = { (A+B); E; C }
 PUSH D; Pile = { (A+B); E; C; D }
 SUB Pile = { (A+B); E; C-D }
 MPY; Pile = { (A+B); E*(C-D) }
 DIV; Pile = { (A+B) / (E*(C-D)) }
 POP Z; Pile = { } [Z] ← (A+B)/(E*(C-D))

II/ Parfois, l'optimisation des logiciels peut considérablement améliorer les performances d'un système informatique. Supposons qu'un **processeur** peut effectuer une opération de **multiplication** en **10 ns**, et une opération de **soustraction** en **1 ns**.

1- **Combien de temps** faut-il au **CPU** pour calculer le résultat de l'opération $d = a \times b - a \times c$? **(0.75 point)**

$d = a \times b - a \times c$: **2 multiplication + une soustraction** donc 20 ns + 1 ns au **total 21 ns**

2- Pourriez-vous **optimiser l'équation** pour qu'elle prenne moins de temps ? **(0.75 point)**

Après optimisation : $d = a \times (b - c)$: une multiplication + une soustraction donc **11 ns**

Exercice TP : (sur 4 points)

Ecrire en assembleur MIPS le code correspondant au fragment du programme suivant :

if t1 < t2 then t3 := t2 / t1 else t3 := t1 / t2

blt \$t1, \$t2, Then (1)

div \$t3, \$t1, \$t2 (1)

j End (0.5)

Then : (0.25) div \$t3, \$t2, \$t1 (1)

End : (0.25)

.....