

QUIZZ et questions ouvertes MT-EL 23/12/2016

Remarque : l'ordre des réponses était différent selon les variantes. Donc ne faites pas attention à la lettre correspondant à la réponse correcte mais seulement à la **réponse correcte elle-même** qui est surlignée en jaune. Quelques explications brèves sont ajoutées pour pouvoir retrouver la bonne réponse des questions plus difficiles.

Un processeur 32 bits travaille avec une fréquence d'horloge de 2GHz, c'est à dire qu'il exécute une instruction par cycle d'horloge. Ce processeur dispose de 32 registres, chacun de 32 bits. Le chargement d'un registre avec accès en lecture dans le cache s'effectue en 10ns (lecture de un mot de 4 octets). Le chargement d'un registre demandant un accès à la mémoire centrale s'effectue en 200 ns (la taille du bloc transféré n'a pas d'importance pour cet exercice).

Soit le programme assembleur (incomplet) pour lequel on suppose que les registres **r0**, **r1**, **r2** et **r3** sont déjà initialisés avec l'adresse mémoire des variables **a**, **b**, **c** et **d** (respectivement) :

```
0 : mcharge r4, r0
1 : mcharge r5, r1
2 : mcharge r6, r2
3 : mcharge r7, r3
4 : multiplie r8, r4, r7
5 :
6 : soustrait r10, r8, r9
```

Voir la p2 pour les conventions utilisées pour les instructions.

Question 1 : Comment faut-il compléter le programme à la ligne 5 pour qu'il calcule dans le registre **r10** le déterminant **D** d'une matrice 2x2 selon la formule: $D = a*d - c*b$

- A multiplie r9, r5, r8
- B multiplie r9, r5, r6
- C multiplie r9, r6, r8
- D aucune des autres réponses

Question 2 : Si l'on suppose que les valeurs des 4 variables **a**, **b**, **c** et **d** sont déjà dans le cache au lancement du programme, quel est le temps pris par le processeur pour exécuter ce programme

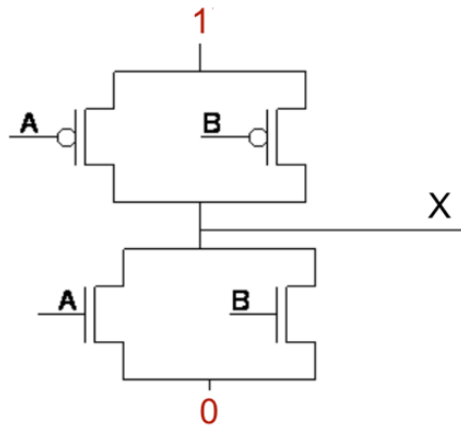
- A 71.5 ns
- B 41.5 ns // 4 accès au cache pour mcharge (x10ns) + 3 opérations (3x 0.5ns)
- C 10.5 ns
- D 70 ns

Question 3 : On suppose que les valeurs de 2 variables sont déjà dans le cache au lancement du programme tandis que les valeurs des deux autres variables sont en mémoire centrale. Plusieurs scénarios d'exécution sont possibles selon 1) la localité spatiale des variables en mémoire centrale et 2) l'éventuel remplacement du bloc contenant une variable dans le cache par un bloc transféré de la mémoire centrale. Malgré cette diversité de scénarios, choisir le seul temps d'exécution correct du programme :

- A 641.5 ns
- B 421.5 ns // 2 accès en mémoire centrale (x200ns) + 2 accès cache (x10ns) + 3 op. (x1.5ns)
- C 821.5 ns
- D 241.5 ns

- A $F = (A \text{ ET } B) \text{ OU } C$
- B $F = (A \text{ ET } C) \text{ OU } B$
- C $F = \text{négation logique de } ((A \text{ ET } B) \text{ OU } C)$ // construire une table de vérité avec les 8 cas
- D $F = \text{négation logique de } ((A \text{ ET } C) \text{ OU } B)$

Question 7 : indiquer la proposition correcte à propos de ce circuit :



- A Il réalise l'opération logique A OU-EXCLUSIF B
- B Il réalise la négation de l'opération logique A OU-EXCLUSIF B
- C Il produit un court-circuit quand A=1 et B=1
- D Il produit un court-circuit dans 2 cas : (A=1 et B=0) ou (A=0 et B=1)

Un processeur 32 bits travaille avec une fréquence d'horloge de 2GHz, c'est à dire qu'il exécute une instruction par cycle d'horloge. On désire calculer la somme de tous les octets, un par un, d'un fichier de taille 8 MiB (voir p2 pour les unités) stocké sur un disque dur externe.

Pour cet exercice, on suppose que l'instruction d'addition qui ajoute la valeur d'un octet à la somme est effectuée en un seul cycle d'horloge lorsque l'octet est présent dans la mémoire cache. On ignore également la gestion d'une éventuelle variable de boucle pour calculer la somme.

La mémoire cache est constituée d'un seul bloc de 4KiB. En cas de **défaut de cache** un bloc de 4KiB est transféré avec une latence de 2^7 cycles d'horloge. La mémoire centrale est organisée en pages de 4KiB et, si la donnée recherchée n'est pas en mémoire centrale (**défaut de page**), la mémoire centrale obtient du système d'exploitation un bloc de 4 pages à la fois avec une latence de 2^{13} cycles d'horloge.

Question 8 : Combien y a-t-il de défaut de cache pour pouvoir traiter l'ensemble des données à partir de la mémoire cache ?

- A 1024
- B 2048 // $2^{23}/2^{12}$
- C 4096
- D 8192

Question 9 : Combien y a-t-il de défaut de page pour obtenir l'ensemble des données en mémoire centrale et quel est (en binaire) le nombre total de cycles pour calculer cette somme

- A 256 et 101010000000000000000000
- B 256 et 100000000010000010000000
- C 512 et 110001000000000000000000 // $2^{23}/2^{14}$ et $2^{23} + 2^{11}2^7 + 2^92^{13} = 2^{23} + 2^{18} + 2^{22}$
- D 512 et 100000000010000010000000

Le programme suivant doit calculer le terme $F(n)$ de la suite de Fibonacci qui est définie comme suit: $F(0)=0$, $F(1)=1$, $F(n) = F(n-1) + F(n-2)$ La valeur de n se trouve déjà dans le registre $r1$. Le résultat $F(n)$ doit être fourni dans $r2$:

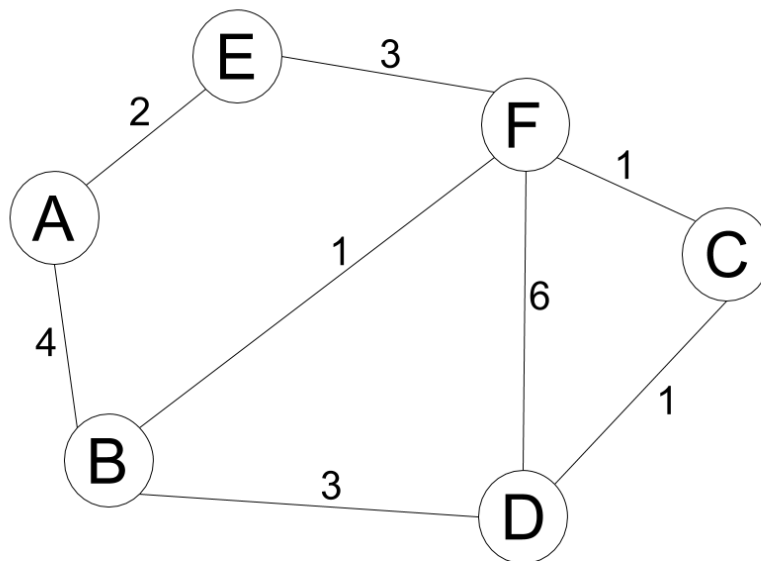
1. `continue_diff r1, 0, 4`
2. `charge r2, r1`
3. `stop`
4. `charge r3, 0`
5. `charge r4, 1`
6. `soustrait r1, r1, 1`
7. `continue_egal r1, 0, 13`
8. `somme r5, r3, r4`
9. `charge r2, r3`
10. `charge r4, r5`
11. `soustrait r1, r1, 1`
12. `continue 7`
13. `charge r2, r4`
14. `stop`

Voir la p2 pour les conventions utilisées pour les instructions.

Question 10: Il y a un bug dans ce programme, comment peut-on le réparer ?

- A Remplacer la ligne 9 par: `charge r3, r4`
- B Enlever la ligne 6
- C Remplace la ligne 7 par: `continue_diff r1, 0, 13`
- D Ajouter cette ligne après la ligne 4: `charge r2, 0`

Ce graphe de routeurs Internet montre la distance entre chaque nœud à côté de chaque lien. Par exemple, la distance entre A et B est de 4.



Question 11

Indiquer quelle est la **table de routage du nœud A** parmi les choix suivants. Pour chaque **destination** (colonne de gauche) on indique la **direction** du prochain nœud (colonne du milieu) et la **distance** du plus court chemin (colonne droite)

A		
dest.	dir.	dist.
B	B	4
C	C	6
D	E	7
E	E	2
F	E	5

A		
dest.	dir.	dist.
B	A	4
C	E	5
D	B	7
E	E	2
F	B	5

A		
dest.	dir.	dist.
B	B	3
C	B	5
D	B	6
E	E	2
F	B	4

A		
dest.	dir.	dist.
B	B	4
C	B	6
D	B	7
E	E	2
F	B	5

Réponse : A.

B.

C.

D.

Question 12

La table du routeur F est:

F		
dest.	dir.	dist.
A	B	5
B	B	1
C	C	1
D	C	2
E	E	3

Quelle est sa nouvelle table si le lien est coupé entre B et F ?

F		
dest.	dir.	dist.
A	E	5
B	C	5
C	C	1
D	C	2
E	E	3

F		
dest.	dir.	dist.
A	E	5
B	D	6
C	C	1
D	C	2
E	E	3

F		
dest.	dir.	dist.
A	E	5
B	C	6
C	C	1
D	D	6
E	E	3

F		
dest.	dir.	dist.
A	E	5
B	E	9
C	C	1
D	C	2
E	E	3

Réponse : **A.**

B.

C.

D.

Questions Ouvertes

Question 1 : Assembleur (4 points)

1.1) Le programme suivant en langage Assembleur utilise les entiers positifs stockés dans les registres **r1** par **r2** pour calculer $r1^{r2}$ ($r1$ puissance $r2$) qui doit être rangé dans le registre **r3**. Cependant, ce code présente un bug. Quelle ligne doit-on modifier pour supprimer le bug ?

```
1: charge r3, 0
2: continue_pp r2, 1, 6 // va à la ligne 6 si r2 < 1
3: multiple r3, r1, r3
4: soustrait r2, r2, 1
5: continue 2
6: stop
```

le bug se situe à la ligne 1. L'accumulateur r3 doit être initialisé avec la valeur 1 car une multiplication est effectuée à chaque passage dans la boucle principale (lignes 2-5).

Attention : ne pas initialiser avec r1 car on calculerait $r1^{r2+1}$ à la place.

1.2) Après correction du bug, le programme est exécuté sur un processeur 32 bits qui travaille avec une fréquence d'horloge de 2GHz, c'est à dire qu'il exécute une instruction par cycle d'horloge. Quel est le temps d'exécution de ce programme en ns pour $r1 = 2$ et $r2 = 3$?

Comptons tout d'abord le nombre de cycles (le décompte est requis pour avoir tous les points) :

L'instruction de la ligne 1 est exécutée une seule fois = 1 cycle

L'instruction de la ligne 2 est exécutée (3 + 1) fois = 4 cycles : 3 fois FAUX et 1 fois VRAI

Les 3 instructions des lignes 3, 4 et 5 sont exécutées 3 fois = $3 \times 3 = 9$ cycles

L'instruction stop de la ligne 6 est exécutée une fois = 1 cycle

Total = $1 + 4 + 9 + 1 = 15$ cycles

Temps en ns : la durée d'un cycle est donnée par l'inverse de la fréquence d'horloge

Variante Blanche et Jaune : $1/2\text{GHz} = 0.5\text{ns}$ temps total = 7.5 ns

Variante Bleue : $1/1\text{GHz} = 1\text{ns}$ temps total = 15 ns

Variante Verte : $1/4\text{GHz} = 0.25\text{ns}$ temps total = 3.75 ns

1.3) Justifier l'ordre de complexité de ce programme en fonction de **r2** :

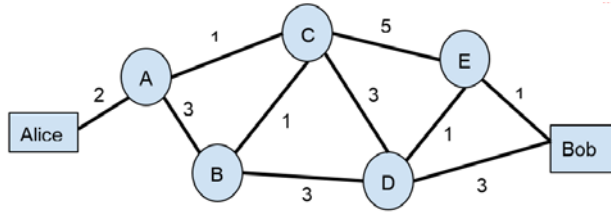
La boucle est contrôlée par la décrémentation de une unité de la variable **r2** à chaque passage. Le nombre de passages est proportionnel à **r2**.

Le coût calcul étant constant à chaque passage, on a seulement un coût linéaire en **r2**, d'où **O(r2)**. La justification et la syntaxe précise **O(r2)** sont requises pour avoir ce point.

Question 2 : Routage multi-critères (4 pts)

Dans cet exercice la valeur associée à chaque lien indique la performance d'un critère qui est différent selon la question.

2.1) Critère de latence exprimé en ms, c'est-à-dire la durée de transmission d'un paquet entre les deux nœuds reliés par le lien. Compléter les tables de routage pour le réseau ci-dessous en retenant le chemin avec la latence la plus faible (colonne de droite notée **lat**).



2.1.1)

B		
dest.	dir.	lat.
Alice	C	4
Bob	D	5

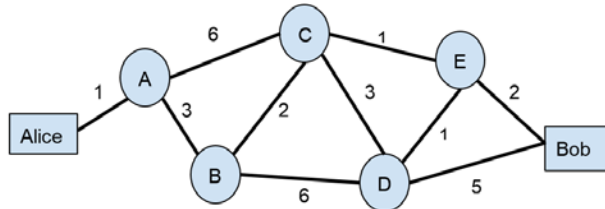
C		
dest.	dir.	lat.
Alice	A	3
Bob	D	5

D		
dest.	dir.	lat.
Alice	C	6
Bob	E	2

2.1.2) Quel est le chemin entre Alice et Bob avec la plus faible latence : **A C D E**

2.1.3) Quelle est cette latence (ms) : **2 + 1 + 3 + 1 + 1 = 8ms**

2.2) Voici maintenant le même réseau avec un critère différent: le coût d'un transfert de 1 GB de données (peu importe la direction), exprimé en CHF. Compléter les mêmes tables, cette fois en minimisant le **coût** (colonne de droite)



2.2.1)

B		
dest.	dir.	coût
Alice	A	4
Bob	C	5

C		
dest.	dir.	coût
Alice	B	6
Bob	E	3

D		
dest.	dir.	coût
Alice	E	8
Bob	E	3

2.2.2) Quel est le chemin entre Alice et Bob avec le plus faible coût : **A B C E**

2.2.3) Quel est son coût : **1 + 3 + 2 + 1 + 2 = 9 CHF**

2.2.4) Quel est le surcoût en CHF du chemin à plus faible latence par rapport à celui-ci : **4 CHF**

2.2.5) Quelle est la latence supplémentaire du chemin le moins cher par rapport au chemin le plus rapide ?

2 + 3 + 1 + 5 + 1 = 12 ms ; 12 - 8 = 4 ms de latence supplémentaire.