ICC MT & EL Test N° 3 vendredi 23 décembre 2016

Ne PAS retourner ces feuilles avant d'en être autorisé!

Merci de poser votre carte CAMIPRO en évidence sur la table.

Vous pouvez déjà compléter et lire les informations ci-dessous:

NOM _____

Prénom _____

Numéro SCIPER _____

Signature _____

BROUILLON : Ecrivez aussi votre NOM-Prénom sur la feuille de brouillon fournie. Toutes vos réponses doivent être sur cette copie d'examen. Les feuilles de brouillon sont ramassées pour être immédiatement détruites.

Le test écrit commence à: **14h15**

Retourner les feuilles avec la page 8 face à vous à : **15h30**

les contrôles écrits ICC sont SANS document autorisé, ni appareil électronique

Total sur 20 points = 12 points pour la partie Quizz et 8 points pour les questions ouvertes

Vous pouvez utiliser un crayon à papier et une gomme

La partie Quizz (QCM) comporte 12 questions : chaque question n'a qu'une seule réponse correcte parmi les 4 réponses proposées. Chaque réponse correcte donne 1 point. Aucun point n'est donné en cas de réponses multiples, de rature, ou de réponse incorrecte.

Indiquez vos réponses à la partie Quizz dans ***le tableau en bas de cette page***.

La partie « question ouverte » comporte 2 questions = 4 points + 4 points.

	Questions du Quizz												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
A													**A**
B													**B**
C													**C**
D													**D**

Unités de taille utilisées dans cet examen

Les puissances de dix utilisent une seule lettre majuscule comme par exemple :
10^3 = K (kilo), 10^6 = M (méga), 10^9 = G (giga)

Les puissances de deux qui sont proches de certaines puissances de dix utilisent la même lettre majuscule suivie par la lettre minuscule 'i'. Voici les exemples proches des puissances de dix indiquées plus haut:
2^{10} = 1024 = Ki (kibi), 2^{20} = 1024 *1024 = Mi (mébi), 2^{30} = 1024*1024*1024 = Gi (gibi)

Dans cet examen, nous utilisons la lettre '**B**', comme ***Byte***, pour désigner un ***octet***. Donc les unités **KiB, MiB et GiB** désignent respectivement **2^{10}, 2^{20} et 2^{30} octets.**

Syntaxe des instructions assembleur

Attention : certaines questions utilisent seulement un sous-ensemble de ces instructions

Chargement d'un registre : Il est possible de remplacer **r2** par une constante :

`charge r1, r2`	range dans **r1** la valeur stockée dans **r2**
`mcharge r1, r2`	range dans **r1** la valeur stockée en mémoire à l'adresse **r2**

Calcul : Il est possible de remplacer **r3** par une constante :

`somme r1, r2, r3`	calcule **r2 + r3** et range le résultat dans **r1**
`soustrait r1, r2, r3`	calcule **r2 - r3** et range le résultat dans **r1**
`multiplie r1, r2, r3`	calcule **r2 * r3** et range le résultat dans **r1**

Branchement : Il est possible de remplacer **r2** par une constante :

`continue_diff r1, r2, i`	Si **r1** est différent de **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue_egal r1, r2, i`	Si **r1** est égal à **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue_ppe r1, r2, i`	Si **r1** $\leq$ **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue_pp r1, r2, i`	Si **r1** $<$ **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue_pge r1, r2, i`	Si **r1** $\geq$ **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue_pg r1, r2, i`	Si **r1** $>$ **r2** alors continue à l'adresse **i** Sinon passe à l'instruction suivante.
`continue i`	branchement inconditionnel à l'adresse **i**

Divers : arrêt et fin du programme avec l'instruction **stop**

QUIZZ

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
C 10.5 ns
D 70 ns

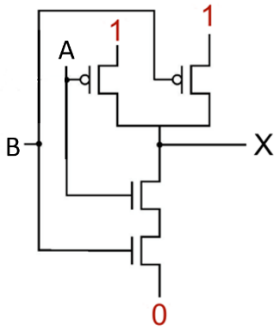
Question 3 : On suppose que les valeurs de 2 variables sont déjà dans le cache au lancement du programme tandis que les valeurs des deux autres variables sont en mémoire centrale. Plusieurs scénarios d'exécution sont possibles selon 1) la localité spatiale des variables en mémoire centrale et 2) l'éventuel 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
C 821.5 ns
D 241.5 ns

Question 4 : pour cette question on utilise l'algorithme LRU (Least Recently Used) pour décider quel est le bloc à remplacer lorsque le cache est plein.
En supposant que le cache a une taille de **5 blocs** et que l'on commence avec un cache vide, indiquer quels sont les blocs présents dans le cache après cette séquence de chargement de blocs (chaque lettre désigne un bloc) :

(premier bloc chargé) A, B, C, D, E, A, F, A, D, F, G (dernier bloc chargé) :

A A, B, C, D, E
B A, F, G, D, E
C A, F, C, D, G
D A, B, D, F, G



Question 5 : Quelle est la table de vérité de la sortie X du circuit ci-dessus ?

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

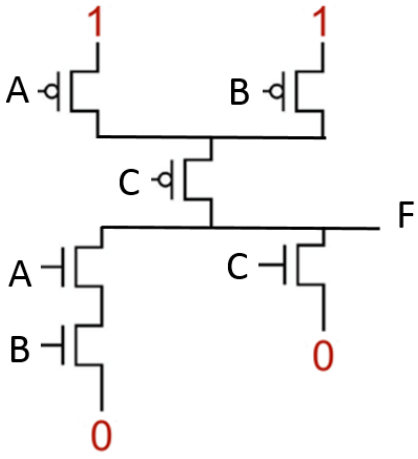
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

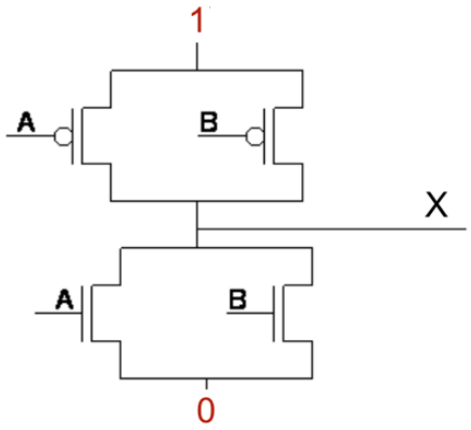
Réponse : A B C D

Question 6 : Quelle la fonction logique réalisée par la sortie F de ce circuit à partir des trois entrées A, B et C ?



A F = (A ET B) OU C
B F = (A ET C) OU B
C F = négation logique de ((A ET B) OU C)
D F = négation logique de ((A ET C) 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 $\mathbf{2^7}$ 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 $\mathbf{2^{13}}$ 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
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 10101000000000000000000
B 256 et 10000000001000001000000
C 512 et 11000100000000000000000
D 512 et 10000000001000001000000

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**:

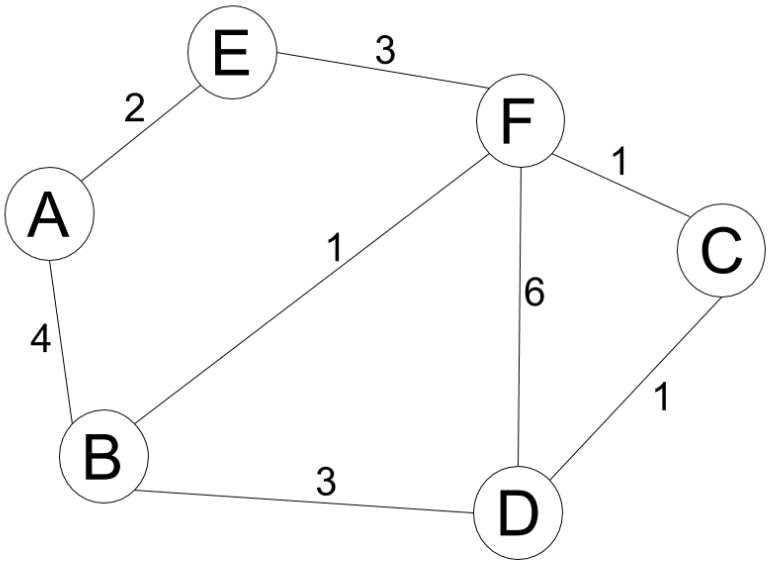
```
1.   continue_diff r1, 0, 4
2.   charge r2, r1
3.   stop
4.   charge r3, 0
5.   charge r4, 1
6.   soustrait r1, r1, 1
7.   continue_egal r1, 0, 13
8.   somme r5, r3, r4
9.   charge r2, r3
10.  charge r4, r5
11.  soustrait r1, r1, 1
12.  continue 7
13.  charge r2, r4
14.  stop
```

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 coté 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 $\mathbf{r1}^{\mathbf{r2}}$ (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
3: multiple r3, r1, r3
4: soustrait r2, r2, 1
5: continue 2
6: stop
```

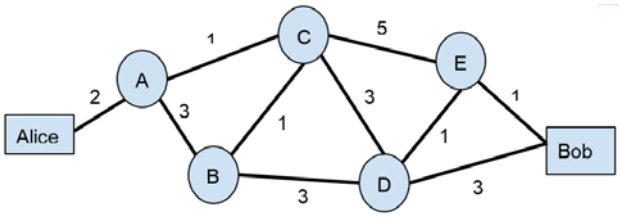
1.2) Après correction du bug, le programme est exécuté sur un processeur 32 bits qui travaille avec une fréquence d'horloge de 2GHz, c'est à dire qu'il exécute une instruction par cycle d'horloge. Quel est le temps d'exécution de ce programme en **ns** pour **`r1 = 2`** et **`r2 = 3`** ?

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

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		
Bob		

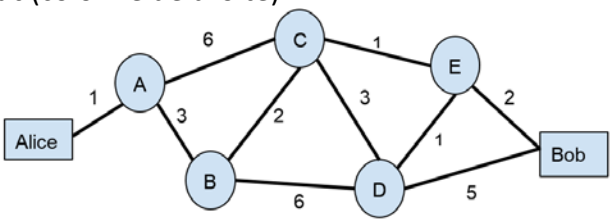
C		
dest.	**dir.**	**lat.**
Alice		
Bob		

D		
dest.	**dir.**	**lat.**
Alice		
Bob		

2.1.2) Quel est le chemin entre Alice et Bob avec la plus faible latence : ……

2.1.3) Quelle est cette latence (ms) :……

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		
Bob		

C		
dest.	**dir.**	**coût**
Alice		
Bob		

D		
dest.	**dir.**	**coût**
Alice		
Bob		

2.2.2) Quel est le chemin entre Alice et Bob avec le plus faible coût : ……

2.2.3) Quel est son coût :……

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

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

……..

Ne rien écrire sur cette page,

Rappel : avez-vous complété le tableau en p1 ?

Présenter cette page sur le dessus dans les 2 cas suivants :

1) *vous avez fini avant 15h30*
2) *les copies sont ramassées*