UNIVERSITÉ DE PARIS-DAUPHINE

CYCLE

ANNÉE : 2023 SESSION : 2

MATIÈRE : Architecture des Ordinateurs.

UV =

Il est interdit aussi bien de signer à la fin de la composition que d'indiquer son nom ou son numéro sur les feuilles intercalaires.

Le candidat inscrit ici très lisiblement son

Nom : PHICHITH

Prénoms : Navannapha

N° GROUPE : 4

Numéro de convocation :

N° de groupe : TD4

Nombre d'intercalaires : 2

	Note	Signature	Note finale	APPRÉCIATIONS EXPLIQUANT LA NOTE
1er correcteur			17,5	
2e correcteur				

Ne pas écrire dans cette marge

5 Sujet : Exercice 1 : Nombres flottants

1) $0,625 = 1,25 \times 2^{-1} = 0.0110.0100$

$10 = 1,25 \times 2^{3} = 0.1010.0100$

1 $3,125 = 1,5625 \times 2^{1} = 0.1000.10010$

2) $10 \times 3,125 = 31,25$ et $10 \times 0,625 = 6,25$ exactement pour ces 2 multiplications.

1 On multiplie les mantisses entre elles : on a alors pour :

• $10 \times 3,125 = 1,01 \times 2^{3} \times 1,001 \times 2^{1}$

$$\begin{array}{r} 1,1001 \\ \times \quad 1,01 \\ \hline 11001 \\ +11001.. \\ \hline 1,111101 \times 2^4 \end{array}$$

$= 1,111101 \times 2^{4}$ qui s'arrondit à $1,11110 \times 2^{4} = \boxed{31}$

$10 \times 0{,}625$

$= 1{,}01 \times 2^3 \times 1{,}01 \times 2^{-1}$

$$\begin{array}{r} 1{,}01 \\ \times\ 1{,}01 \\ \hline 101 \\ +\ 101\ \cdot\ \cdot \\ \hline 1{,}1001 \times 2^2 \end{array}$$

d'où $1{,}01 \times 2^3 \times 1{,}01 \times 2^{-1}$

$= 1{,}1001 \times 2^2$

$= 6{,}25$ soit le résultat correct.

1/2 3a) Si les calculs sont exacts, on a : $x = 10 \times 3{,}125 - 10 \times 0{,}625 = 25$

3b) On a effectué les produits dans la question 2. Il ne nous reste plus qu'à les soustraire : on a alors avec les mantisses respectives,

$10 \times 3{,}125 - 10 \times 0{,}625 = 1{,}1110 \times 2^4 - 1{,}1001 \times 2^2$

$= 1{,}1110 \times 2^4 - 0{,}011001 \times 2^4$

$= 1{,}100011 \times 2^4$

qui s'arrondit à $1{,}10001 \times 2^4$

$\boxed{= 24{,}5 \text{ en décimal}}$

1

$$\begin{array}{r} 1{,}111100 \\ -\ 0{,}011001 \\ \hline 1{,}100011 \times 2^4 \end{array}$$

3c) On fait $f - g = 3{,}125 - 0{,}625$ ~~avec les mantisses respectives~~ : on a alors on obtient $2{,}5 = 1{,}25 \times 2^1 = 0.1000.01000 - 1{,}01 \times 2^1$

On multiplie ensuite les mantisses entre elles de 2,5 et 10 :

donc $10 \times 2{,}5 = 1{,}01 \times 2^3 \times 1{,}01 \times 2^1$

$= 1{,}1001 \times 2^4$

$\boxed{= 25}$

1

$$\begin{array}{r} 1{,}01 \\ \times\ 1{,}01 \\ \hline 101 \\ +\ 101\ \cdot\ \cdot \\ \hline 1{,}1001 \times 2^4 \end{array}$$

et on obtient bien 25 le résultat exact.

1/2

3d) Selon les calculs réalisés, rester dans les registres peut changer le résultat final. Les arrondis n'intervenant pas au même endroit, on peut obtenir des résultats différents: les calculs en virgule flottante ne sont pas toujours reproductibles

3,5

Exercice 2: Circuits Logiques

1)

F	a	b	s_0	s_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

1

2) $s_0 = \bar{F}(a \oplus b) + F(a+b)$

$\boxed{s_1} = \bar{F}ab + Fab = ab(\bar{F}+F) = \boxed{ab}$

1

3) On peut écrire le OU de (a ou b) comme $a + b = a \oplus b + ab$.

Ainsi, réécrivons s_0 comme $s_0 = \bar{F}(a \oplus b) + F(a+b)$

$= \bar{F}(a \oplus b) + F(a \oplus b + ab)$

$= (a \oplus b)(\bar{F}+F) + Fab$

$\boxed{s_0 = a \oplus b + Fab}$

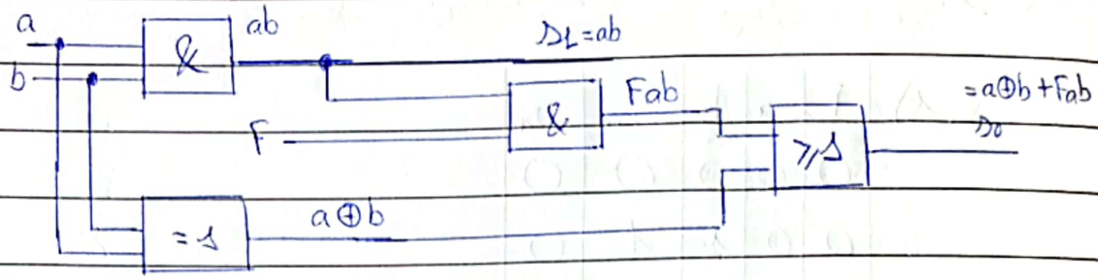
On peut construire le circuit suivant avec

$S_0 = (a \oplus b) + Fab$

$S_1 = ab.$

Circuit logique:

1,5



1,5

V) Mémoire cache

1) Cache à accès direct de 16 blocs de 1 octet

1

2	M	2
3	M	3
9	M	9
11	M	11
14	M	14
16	M	16
13	M	13
62	M	14
4	M	4
10	M	10
16	C	16
45	M	13
8	M	8
10	C	10
3	C	3
13	M	13

Donc $T = 13M + 3C$.

3,5

Exercice 3. Assembleur

```
        ~~LDR  r0,(r0)~~               ~~récupérer la valeur de r0~~
loop:   LDB  r2,(r0)          - récupérer le caractère dans r0
        JZ   r2, fin
        SUB  r31, r2, #' '    - si c'est un espace
        JZ   r31, suivant
        ADD  r2, r2, r1
        SUB  r31, r2, #'Z'    - est-ce que c'est plus grand que Z?
        JLE  r31, stocker     - pas de changement à faire
        SUB  r2, r2, #26      - pour revenir vers A
stocker: STB (r0), r2
suivant: ADD r0, r0, #1       décaler r0 pour le caractère suivant
        JMP  loop
fin:
```

4

Exercice 4:

```
            MVI  r31, r1       - conserver le nombre d'éléments
            SUB  r5, r31, #1
grandeloop: MOV  r31, r5       - conserver le nombre d'échange à faire à chaque passage dans la boucle
            MOV  r2, r0
            MOV  r3, r0
            JZ   r1, fin       - s'il n'y a pas d'éléments
            ADD  r3, r3, #1
            SUB  r1, r1, #1    ← 1 passage dans la boucle
loop:       LDB  r20, (r2)     } récupérer les valeurs à i et i+1
            LDB  r30, (r3)     }
            SUB  r4, r20, r30
            JLE  r4, rien
            STB  (r2), r30     - sinon échanger les 2 valeurs
            STB  (r3), r20
rien:       ADD  r2, r2, #1
            ADD  r3, r3, #1
            SUB  r31, r31, #1  - 1 échange a été fait
            JNZ  r31, loop     - on a tout permuté ?
```

JZ n1, fin - a-t-on fait les N passages dans la boucle ?
JMP grandeLoop - on y retourne sinon !

fin:

suite mémoire cache:

3) Cache associatif à 4 blocs de 4 octets (LRU)

$\boxed{T = \cancel{7}^{8} m + \cancel{3}^{8} c}$

2	1-4	m	1
3		c	1
9	9-12	m	2
11		c	2
14	13-16	m	3
16		c	3
13		c	3
62	61-64	m	4
4		c	1
10		c	2
16		c	3
45	45-48	m	4
8	5-8	m	1
10		c	2
3	1-4	~~m~~	1
13	13-16	m	1

1/2