

# QuestionnaireGroupe1

Merci de prendre part à cette étude qui vise à comprendre l'influence que peut avoir la visualisation sur la compréhension du comportement de logiciel. Vous avez à répondre à 12 questions dont les réponses vont de remplissages automatiques à quelques calculs que vous pouvez faire mentalement. Vous pouvez aussi vous servir d'une calculatrice ou d'une feuille de calcul Excel. Si vous le souhaitez, vous pouvez répondre de façon anonyme à ce questionnaire en fournissant un pseudonyme à la place de votre nom. Cette étude a obtenu une certification éthique du Comité d'éthique de la recherche avec les êtres humains de l'université TELUQ (CER-TELUQ) numéro 2022-08 du 12 avril 2022.

## A- Renseignements personnels, date et heure du début de remplissage du questionnaire

**Q1- Nom ou pseudonyme**

Shinybuss

**Q2-Date :**

2022-05-24

**Q3-Heure du début :**

09:18

## C- Compréhension du comportement des instructions vectorielles \_mm512\_mask\_add\_ps et \_mm\_shuffle\_epi32

### 1- Instruction vectorielle \_mm512\_mask\_add\_ps

Lisez attentivement l'explication de cette instruction donnée sur la figure ci-dessous que vous pouvez retrouver sur le site [web d'Intel©](#) avant de répondre aux questions Q7 et Q8. Vous pouvez aussi regarder cette [courte vidéo](#) où cette instruction est expliquée.

Vous ne devez utiliser que les explications fournies (vous pouvez naturellement consulter le site [web d'Intel©](#) ), mais ne faites pas recours à d'autres ressources (par exemple recherche sur Google, autres documents, etc.)

```
__m512 __mm512_mask_add_ps (__m512 src, __mmask16 k, __m512 a, __m512 b)
```

vaddps

### Synopsis

```
__m512 __mm512_mask_add_ps (__m512 src, __mmask16 k, __m512 a, __m512 b)
#include <immintrin.h>
Instruction: vaddps zmm {k}, zmm, zmm
CPUID Flags: AVX512F
```

### Description

Add packed single-precision (32-bit) floating-point elements in `a` and `b`, and store the results in `dst` using writemask `k` (elements are copied from `src` when the corresponding mask bit is not set).

### Operation

```
FOR j := 0 to 15
    i := j*32
    IF k[j]
        dst[i+31:i] := a[i+31:i] + b[i+31:i]
    ELSE
        dst[i+31:i] := src[i+31:i]
    FI
ENDFOR
dst[MAX:512] := 0
```

### Performance

Architecture	Latency	Throughput (CPI)
Icelake	4	1
Skylake	4	0.5

**Q7- Après avoir bien lu la description et les explications ci-dessus, dites ce que fait l'instruction `_mm512_mask_add_ps` en effectuant le calcul suivant: étant donnés `src=(1, 3, 4, 1, 2, 5, 4, 1, 2, 3, 4, 1, 1, 3, 4, 1)` ; `k=(1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0)` ; `a=(6, 1, 2, 3, 1, 4, 5, 1, 2, 3, 4, 1, 3, 1, 2, 1)` ; `b=(6, 1, 2, 3, 1, 4, 5, 1, 2, 3, 4, 1, 3, 1, 2, 1)` . Calculer `r = _mm512_mask_add_ps(src, k, a, b)`  
`r = (13, 0, 0, 7, 4, 0, 0, 3, 0, 9, 12, 0, 0, 0, 8, 0)`**

**Q8- Toujours à l'aide de la description et des explications ci-dessus, donnez une formule générale de calcul des coordonnées de `r(ri)` en fonction de celles de `src(srci)`, `k(ki)`, `a(ai)` et `b(bi)`. `ri= ?`  
`ri=ki x srci+(1-ki) x (ai+bi)`**

## 2- Instruction vectorielle `_mm_shuffle_epi32`

Lisez attentivement l'explication de cette instruction sur la capture d'écran ci-deessous. Vous pouvez retrouver cette explication sur [Intel © web site](#). Assurez-vous autant que possible d'avoir bien lu l'explication avant de répondre aux questions Q9 et Q10. Vous pouvez aussi regarder [cette courte vidéo](#) où cette instruction est expliquée ; cette [cette courte vidéo](#) est une explication faite par un expert du domaine de la programmation vectorielle. Vous ne devez utiliser que les explications fournies (vous pouvez naturellement consulter le [site web d'Intel ©](#), mais ne faites pas recours à d'autres ressources (par exemple recherche sur Google, autres documents, etc.).

```
__m128i _mm_shuffle_epi32 (__m128i a, int imm8)
```

psrshufd

### Synopsis

```
__m128i _mm_shuffle_epi32 (__m128i a, int imm8)
#include <emmintrin.h>
Instruction: psrshufd xmm, xmm, imm8
CPUID Flags: SSE2
```

### Description

Shuffle 32-bit integers in `a` using the control in `imm8`, and store the results in `dst`.

### Operation

```
DEFINE SELECT4(src, control) {
    CASE(control[1:0]) OF
        0:    tmp[31:0] := src[31:0]
        1:    tmp[31:0] := src[63:32]
        2:    tmp[31:0] := src[95:64]
        3:    tmp[31:0] := src[127:96]
    ESAC
    RETURN tmp[31:0]
}

dst[31:0] := SELECT4(a[127:0], imm8[1:0])
dst[63:32] := SELECT4(a[127:0], imm8[3:2])
dst[95:64] := SELECT4(a[127:0], imm8[5:4])
dst[127:96] := SELECT4(a[127:0], imm8[7:6])
```

### Performance

Architecture	Latency	Throughput (CPI)
Skylake	1	1
Broadwell	1	1
Haswell	1	1
Ivy Bridge	1	0.5

**Q9- Après avoir bien lu la description et les explications ci-dessus, dites ce que fait l'instruction `_mm_shuffle_epi32` en effectuant le calcul suivant: étant donnés  $a=(6, 7, 4, 3)$  ;  $imm8=(0, 1, 2, 3)$  . Calculez  $r = \_mm\_shuffle\_epi32(a, imm8)$   
 $r = (6, 2, 1, 3)$**

**Q10- Toujours à l'aide de la description et des explications ci-dessus, donnez une formule générale de calcul des coordonnées de  $r(ri)$  en fonction de celles de  $a(ai)$  et  $imm8(imm8i)$ .  $ri=?$   
 $ri=a_{ij}$ , avec  $j=imm8i$**

## B- Connaissances préliminaires

### I- Connaissance de l'algèbre et de l'espace vectoriel

Considérons l'espace vectoriel réel  $R^3$ . Pour  $A, B, C, Res1, Res2$ , cinq vecteurs de  $R^3$  tels que  $A=(a_1, a_2, a_3)$ ,  $B=(b_1, b_2, b_3)$ ,  $C=(c_1, c_2, c_3)$ ,  $Res1=(x_1, x_2, x_3)$ ,  $Res2=(y_1, y_2, y_3)$  on définit  $vectSum(A,B,C)=Res1$  et  $vectProd(A,B,C)=Res2$  par

$$\begin{cases} x_1 = a_1 - b_1 + c_1 \\ x_2 = a_2 - b_2 + c_2 \\ x_3 = a_3 - b_3 + c_3 \end{cases} \text{ and } \begin{cases} y_1 = b_1 \times (a_1 - c_1) + c_1 \\ y_2 = b_2 \times (a_2 - c_2) + c_2 \\ y_3 = b_3 \times (a_3 - c_3) + c_3 \end{cases}$$

On suppose maintenant que  $A=(1,0,1)$ ;  $B=(1,1,0)$  ;  $C=(0,1,1)$ .

**Q4- Calculez chacun des vecteurs Res1 et Res2: Res1= ?  
Res2= ?**

Res1=(0,0,2); Res2=(1,0,1).

**Q5- Donnez une formule générale de calcul des coordonnées de Res1(xi) et de Res2(yi) en fonction de celles de A (ai), B (bi) et C (ci). xi= ? yi= ?**

$x_i = a_i - b_i + c_i$  ;  $y_i = b_i \times (a_i - c_i) + c_i$  .

## **II- Connaissance du langage C**

Considérons la fonction f suivante en C: `int f (int x, int y) {return x-y;}`.

**Q6- Déterminez en C deux instructions (soit instruction1 et instruction2) qui permettent de déclarer trois variables entiers a, b, c et de placer dans c la différence de a et b à l'aide de la fonction f.**

**Instruction1: ? Instruction2: ?**

Instruction1: `int c, a, b;` Instruction2: `{return c=f(a-b);}`

## **D- Heure de la fin de remplissage du questionnaire et commentaires**

**Q11- Heure de fin:**

11:03

**Q12- Autres commentaires et remarques:**

Aucune compréhension du langage des instructions vectorielles.