

QuestionnaireGroupe3

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

Il est essentiel de commencer par renseigner ces questions avant de passer aux suivantes. Les questions Q2 et Q3 sont automatiquement renseignées par la date courante et l'heure courante respectivement; ainsi, il ne reste que Q1 que vous devez renseigner.

Q1- Nom ou pseudonyme

Dsk

Nom ou pseudonyme

Saisissez votre nom dans l'espace ci-dessus, ou votre pseudonyme si vous voulez rester anonyme.

Q2-Date :

2022-05-22

Date actuelle

Ce champ est automatiquement renseigné par la date courante.

Q3-Heure du début :

23:01

Heure actuelle

Ce champ est automatiquement renseigné par l'heure courante.

C- Compréhension du comportement des instructions vectorielles `_mm512_mask_add_ps` et `_mm_shuffle_epi32`

1- Instruction vectorielle `_mm512_mask_add_ps`

Avant de répondre aux questions Q7 et Q8, observez attentivement la figure ci-dessous qui est une capture d'écran du prototype SIMD Giraffe, disponible en ligne sur <https://github.com/pmntang/SIMDGiraffe>. Cette capture d'écran est divisée en trois cadrans.

Sur le cadran de gauche il y a une description de l'instruction fournie par Intel®.

Sur le cadran d'en bas, il y a la traduction graphique de cette description. Cette traduction consiste à afficher pour chacun de ces vecteurs, ses champs (ou coordonnées). Nous utilisons les lettres indexées de l'alphabet (A0, A1, ... B0, B1, ..., ...) inscrites à l'intérieur de rectangles bleus pour désigner ces champs. Par exemple les champs (coordonnées) du vecteur src sont A0, A1, ... ce qui veut dire que $src0 = A0$, $src1 = A1$... ; ainsi de suite pour les autres vecteurs opérands ; quant au vecteur résultat, les champs (coordonnées) de r sont E0, E1, ... ce qui veut dire que $r0 = E0$, $r1 = E1$... La description graphique est précédée sur la ligne, à gauche du signe de l'égalité (=), par le nom du vecteur en question (src, k, a, b, r) et son type (`_m512`, `_mmask16`, `_m512`, `_m512`, `_m512`).

Sur le cadran de droite, il y a la description visuelle des liens entre chaque champ (ou coordonnée) du vecteur résultat r et les champs (ou coordonnées) des vecteurs opérands utilisés pour calculer ce champ (ou cette coordonnée). Cette description consiste comme on peut le remarquer, à donner pour chaque

champ du vecteur résultat r la formule qui permet de calculer ce champ à partir des champs opérandes utilisés pour effectuer ce calcul. Par exemple on peut voir sur ce cadran que $r_0 = E_0 = (1-B_0) \times A_0 + B_0 \times (C_0 + D_0)$, $r_1 = E_1 = (1-B_1) \times A_1 + B_1 \times (C_1 + D_1)$, ...

Vous ne devez utiliser que les explications fournies (vous pouvez naturellement consulter le site d'Intel® ou le site de [SIMDGiraffe](https://www.intel.fr/content/www/fr/fr/developement/simd-giraffe/)), mais ne faites pas recours à d'autres ressources (par exemple recherche sur Google, autres documents, etc.). Vous pouvez aussi regarder cette courte vidéo où cette instruction est expliquée par un expert du domaine de la programmation vectorielle: <https://youtu.be/omQ0ebeYJBg>

Choose SIMD Instruction

`_mm512_mask_add_ps(_mm512 src, __mmask16 k, _mm512 a, _mm512 b)`

Synopsis

`_mm512_mask_add_ps(_mm512 src, __mmask16 k, _mm512 a, _mm512 b)`

b)

```
#include <immintrin.h>
Instruction: vaddps zmm {k}, zmm, zmm
CPUID Flags: AVX512F/KNCNI
```

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
```

Novice view

How to compute these fields:

$E_{15} = (1-B_{15}) \times A_{15} + B_{15} \times (C_{15} + D_{15})$	$E_{14} = (1-B_{14}) \times A_{14} + B_{14} \times (C_{14} + D_{14})$
$E_{13} = (1-B_{13}) \times A_{13} + B_{13} \times (C_{13} + D_{13})$	$E_{12} = (1-B_{12}) \times A_{12} + B_{12} \times (C_{12} + D_{12})$
$E_{11} = (1-B_{11}) \times A_{11} + B_{11} \times (C_{11} + D_{11})$	$E_{10} = (1-B_{10}) \times A_{10} + B_{10} \times (C_{10} + D_{10})$
$E_9 = (1-B_9) \times A_9 + B_9 \times (C_9 + D_9)$	$E_8 = (1-B_8) \times A_8 + B_8 \times (C_8 + D_8)$
$E_7 = (1-B_7) \times A_7 + B_7 \times (C_7 + D_7)$	$E_6 = (1-B_6) \times A_6 + B_6 \times (C_6 + D_6)$
$E_5 = (1-B_5) \times A_5 + B_5 \times (C_5 + D_5)$	$E_4 = (1-B_4) \times A_4 + B_4 \times (C_4 + D_4)$
$E_3 = (1-B_3) \times A_3 + B_3 \times (C_3 + D_3)$	$E_2 = (1-B_2) \times A_2 + B_2 \times (C_2 + D_2)$
$E_1 = (1-B_1) \times A_1 + B_1 \times (C_1 + D_1)$	$E_0 = (1-B_0) \times A_0 + B_0 \times (C_0 + D_0)$

[return to expert view](#)

operator = ☐ + ☐ x ☐ - ☐ / ☐ mov ☐ :int ☐ exp ☐ ln ☐ (☐) ☐ idx ☐ inv

`_mm512 src` =

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

`__mmask16 k` =

B ₁₅	B ₁₄	B ₁₃	B ₁₂	B ₁₁	B ₁₀	B ₉	B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

`_mm512 a` =

C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

`_mm512 b` =

D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

`_mm512 r` =

E ₁₅	E ₁₄	E ₁₃	E ₁₂	E ₁₁	E ₁₀	E ₉	E ₈	E ₇	E ₆	E ₅	E ₄	E ₃	E ₂	E ₁	E ₀
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Q7- Après avoir bien observé la figure 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 $r = _mm512_mask_add_ps(src, k, a, b)$

$\bigcirc r = (1, 0, 0, 1, 2, 0, 0, 1, 0, 3, 4, 0, 0, 0, 4, 0)$ $\bigcirc r = (12, 2, 4, 6, 2, 8, 10, 2, 4, 6, 8, 2, 6, 2, 4, 0)$ $\odot r = (12, 3, 4, 6, 2, 5, 4, 2, 2, 6, 8, 1, 1, 3, 4, 1)$ $\bigcirc r = (13, 0, 0, 7, 4, 0, 0, 3, 0, 9, 12, 0, 0, 0, 8, 0)$

Cochez le bouton radio devant la bonne réponse. Vous pouvez vous aider du quadrant de droite de la figure précédente

Q8- Toujours à l'aide de la figure ci-dessus, donnez une formule générale de calcul des coordonnées de $r(r_i)$ en fonction de celles de $src(src_i)$, $k(k_i)$, $a(a_i)$ et $b(b_i)$. $r_i = ?$

$\bigcirc r_i = E_i = C_i + D_i = a_i + b_i$ $\bigcirc r_i = E_i = (1-B_i) \times A_i + C_i + D_i = (1-k_i) \times src_i + a_i + b_i$ $\odot r_i = E_i = (1-B_i) \times A_i + B_i \times (C_i + D_i) = (1-k_i) \times src_i + k_i \times (a_i + b_i)$ $\bigcirc r_i = E_i = B_i \times A_i + (1-B_i) \times (C_i + D_i) = k_i \times src_i + (1-k_i) \times (a_i + b_i)$

Cochez le bouton radio devant la bonne réponse en remarquant que sur la figure précédente $r=(E_i)i=r(r_i)$; $b=(D_i)i=b(b_i)$; $a=(C_i)i=a(a_i)$; $k=(B_i)i=k(k_i)$; $src=(A_i)i=src(src_i)$. i étant l'indice.

2- Instruction vectorielle `_mm_shuffle_epi32`

Avant de répondre aux questions Q9 et Q10, observez attentivement la figure ci-dessous qui est une capture d'écran du prototype SIMD Giraffe, disponible en ligne sur <https://github.com/pmmtang/SIMDGiraffe>. Cette capture d'écran est divisée en trois cadrans.

Sur le cadran de gauche il y a une description de l'instruction fournie par Intel®.

Sur le cadran d'en bas, il y a la traduction graphique de cette description. Cette traduction consiste à afficher pour chacun de ces vecteurs, ses champs (ou coordonnées). Nous utilisons les lettres indexées de l'alphabet (A0, A1, ... B0, B1, ..., ...) inscrites à l'intérieur de rectangles bleus pour désigner ces champs. Ainsi les champs (coordonnées) du vecteur `a` sont A0, A1, A2, A3, ce qui veut dire que $a_0 = A_0$, $a_1 = A_1$, $a_2 = A_2$, $a_3 = A_3$; les champs (coordonnées) du vecteur `imm8` sont B0, B1, B2, B3, ce qui veut dire que $imm_0 = B_0$, $imm_1 = B_1$, $imm_2 = B_2$, $imm_3 = B_3$; quant au vecteur résultat, les champs (coordonnées) de `r` sont C0, C1, C2, C3; ce qui veut dire que $r_0 = C_0$, $r_1 = C_1$, $r_2 = C_2$, $r_3 = C_3$. La description graphique est précédée sur la ligne, à gauche du signe de l'égalité (=), par le nom du vecteur en question (`a`, `imm8`, `r`) et son type (`__m128i`, `int`, `__m128i`).

Sur le cadran de droite, il y a la description visuelle des liens entre chaque champ (ou coordonnée) du vecteur résultat `r` et les champs (ou coordonnées) des vecteurs opérands utilisés pour calculer ce champ (ou cette coordonnée). Cette description consiste comme on peut le remarquer, à donner pour chaque champ du vecteur résultat `r` la formule qui permet de calculer ce champ à partir des champs opérands utilisés pour effectuer ce calcul. On peut ainsi voir sur ce cadran que $r_0 = C_0 = A_0 B_0$, $r_1 = C_1 = A_1 B_1$, $r_2 = C_2 = A_2 B_2$, $r_3 = C_3 = A_3 B_3$.

Vous ne devez utiliser que les explications fournies (vous pouvez naturellement consulter le site d'Intel® ou le site de [SIMD Giraffe](https://github.com/pmmtang/SIMDGiraffe)), mais ne faites pas recours à d'autres ressources (par exemple recherche sur Google, autres documents, etc.). Vous pouvez aussi regarder cette courte vidéo où cette instruction est expliquée par un expert du domaine de la programmation vectorielle: <https://www.youtube.com/watch?v=WVz1jHTIOtY>

Choose SIMD Instruction

_mm_shuffle_epi32

_m128i _mm_shuffle_epi32 (_m128i a, int imm8)

Synopsis

_m128i _mm_shuffle_epi32 (_m128i a, int imm8)

#include <emmintrin.h>

Instruction: pshufd xmm, xmm, imm

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])
    
```

Novice view

How to compute these fields:

$C_3 = A_{B_3}$ $C_2 = A_{B_2}$ $C_1 = A_{B_1}$ $C_0 = A_{B_0}$

[return to expert view](#)

operator =

_m128i a =

int imm8 =

_m128i r =

Q9- Après avoir bien observé la figure 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)

☐ $r = (6, 2, 1, 3)$ ☐ $r = (6, 7, 4, 3)$ ☐ $r = (3, 4, 7, 6)$ ☐ $r = (3, 7, 4, 6)$

Cochez le bouton radio devant la bonne réponse. Vous pouvez vous aider du quadrant de droite de la figure précédente

Q10- Toujours à l'aide de la figure ci-dessus, donnez une formule générale de calcul des coordonnées de $r(r_i)$ en fonction de celles de $a(a_i)$ et $imm8(imm8_i)$. $r_i=?$

☐ $r_i=C_i=A_{ij}=a_{ij}$, avec $j=B_i=imm8_i$ ☐ $r_i=C_i=A_i=a_i$ ☐ $r_i=C_i=A_i \times B_i=a_i \times imm8_i$ ☐ $r_i=C_i=A_j=a_j$, avec $j=B_i=imm8_i$

Cochez le bouton radio devant la bonne réponse en remarquant que sur la figure précédente $r=(C_i)i=r(r_i)$; $a=(A_i)i=a(a_i)$; $imm8=(B_i)i=imm8(imm8_i)$. i étant l'indice.

B- Connaissances préliminaires

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

Considérons l'espace vectoriel réel R^3 . Pour $A, B, C, \text{Res1}, \text{Res2}$, cinq vecteurs de R^3 tels que $A=(a_1, a_2, a_3)$, $B=(b_1, b_2, b_3)$, $C=(c_1, c_2, c_3)$, $\text{Res1}=(x_1, x_2, x_3)$, $\text{Res2}=(y_1, y_2, y_3)$ on définit $\text{vectSum}(A,B,C)=\text{Res1}$ et $\text{vectProd}(A,B,C)=\text{Res2}$ 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=(2,1,0) ; Res2=(1,1,0). ☐ Res1=(0,0,2) ; Res2=(1,0,1). ☐ Res1=(2,2,2) ; Res2=(1,1,1). ☐ Res1=(1,0,2) ; Res2=(0,0,1).

Cochez le bouton radio devant la bonne réponse

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= ?

xi= ai-bi+ci ; yi= bi*(ai-ci) +ci.

Ecrivez xi= ; yi= . Puis, inscrivez l'expression de xi (respectively yi) en fonction des ai, bi et ci dans l'espace devant xi (respectivement yi).

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

Cochez le bouton radio devant la bonne réponse

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

Vous avez presque terminé, un dernier effort!

Q18- Heure de fin:

00:08

Heure de fin

Inscrivez dans ce champ l'heure courante, à la minute près, lorsque vous aurez terminé le remplissage du questionnaire et si vous l'avez rempli de manière ininterrompue. Si vous avez marqué des interruptions, calculez l'heure de fin en déduisant de l'heure courante la durée des totales des interruptions.

Q19- Autres commentaires et remarques:

Vos commentaires

Inscrivez dans ce champ vos remarques, commentaires et observations sur tout sujet d'intérêt en rapport avec l'étude dont le questionnaire, le prototype SIMDGiraffe, etc.