

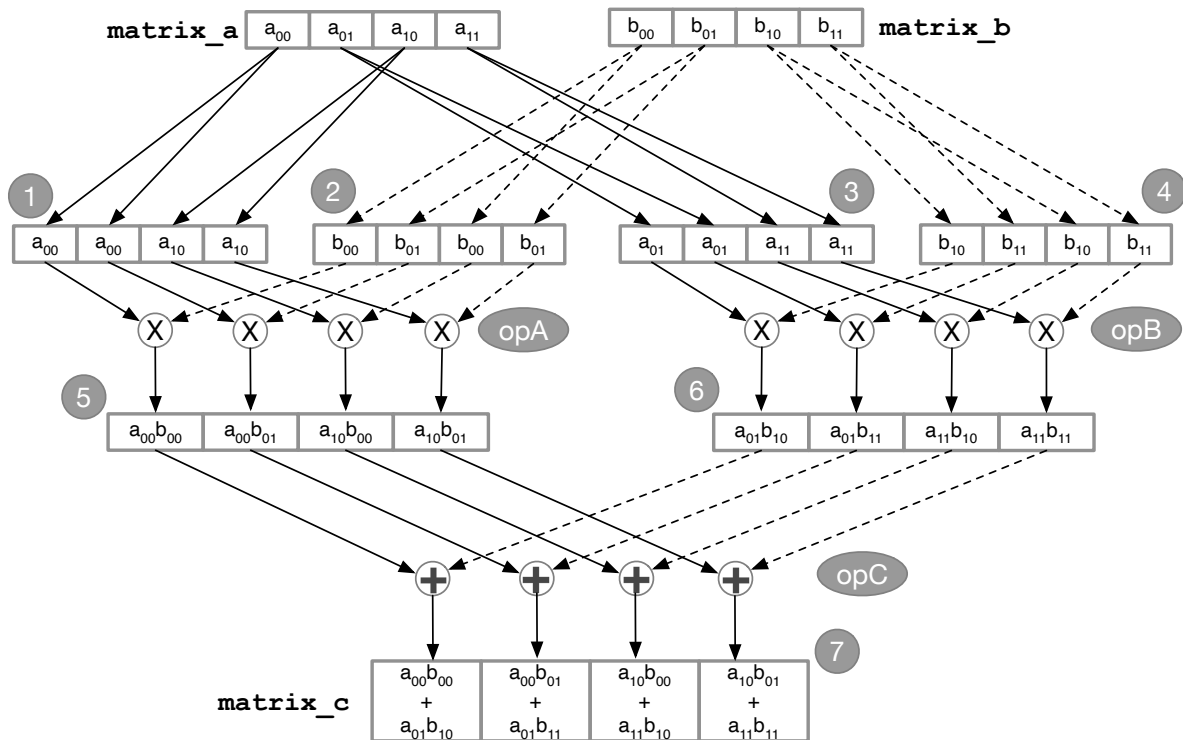
ICI517 Programación Paralela, Control #5  
Escuela de Ingeniería Civil Informática,  
Universidad de Valparaíso  
2022, Semestre 1

Nombre Completo: \_\_\_\_\_

La multiplicación de matrices de 2x2 se define de la siguiente forma:

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \times \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} = \begin{pmatrix} a_{00}b_{00} + a_{01}b_{10} & a_{00}b_{01} + a_{01}b_{11} \\ a_{10}b_{00} + a_{11}b_{10} & a_{10}b_{01} + a_{11}b_{11} \end{pmatrix}$$

Una estrategia vectorial para solucionar esta operación es la que se vió en clases, cuyo diseño es:



En base a lo anterior, responda en forma justificada.

a) Implemente la creación y carga del registro (1) en base a la intrinsics `_mm_shuffle_ps()`. Asuma que el registro `matrix_a` ya está creado y cargado. (4pts)

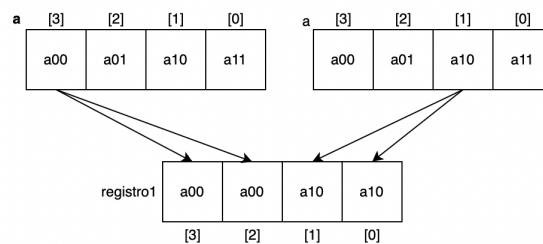
*Creación:*

Se debe declarar el tipo de datos del registro de salida. Debido a que la instrucción es del tipo **ps** (**p**acket **s**ingle **p**recision), el tipo de datos debe ser `__128`.

```
__128 registro1;
```

*Carga:*

Debido a que la intrinsics `_mm_shuffle_ps()` necesita dos registros, el diseño del movimiento de los datos debe ser:



Se debe realizar las siguientes operaciones:

```
c[0] = a[1]
c[1] = a[1]
c[2] = a[3]
c[3] = a[3]
```

Por lo tanto, la macro a utilizar es: `_MM_SHUFFLE(1,1,3,3)`

Finalmente, la carga es:

```
registro1 = _mm_shuffle_ps(a, a, _MM_SHUFFLE(1,1,3,3))
```

b) Implemente la creación y carga del registro (5). Asuma que el registro (1) y (2) están creados y cargados. (3pts)

*Creación:*

Se debe declarar el tipo de datos del registro de salida. Debido a que necesita almacenar 4 floats, el registro debe ser tipo **ps** (**p**acket **s**ingle precision). Luego el tipo de datos es **\_\_128**.

```
__128 registro5;
```

*Carga:*

Según la documentación de Intel, la multiplicación vectorial de números en precisión simple se implementa con la intrínseca **\_mm\_mul\_ps()**. Luego, la carga del **registro5** es:

```
registro5 = _mm_mul_ps(registro1, registro2)
```

c) Implemente la creación y carga del registro (7). Asuma que el registro (5) y (6) están creados y cargados. (3pts)

*Creación:*

Se debe declarar el tipo de datos del registro de salida. Debido a que necesita almacenar 4 floats, el registro debe ser tipo **ps** (packet single precision). Luego el tipo de datos es **\_\_128**.

```
__128 registro7;
```

*Carga:*

Según la documentación de Intel, la suma vectorial de números en precisión simple se implementa con la instrinsic **\_mm\_add\_ps()**. Luego, la carga del **registro7** es:

```
registro7 = _mm_add_ps(registro5, registro6)
```