

**UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**



**LAB1: SIMD – PARALELISMO A NIVEL DE
INSTRUCCIÓN
TALLER DE PROGRAMACIÓN PARALELA**

Datos del alumno:

Nombre completo:	Marcial Natalio Hernández Sánchez
RUT:	17386509-0
Número de teléfono:	569-72825633
E-mail:	marcial.hernandez@usach.cl
Profesor:	Fernando Rannou

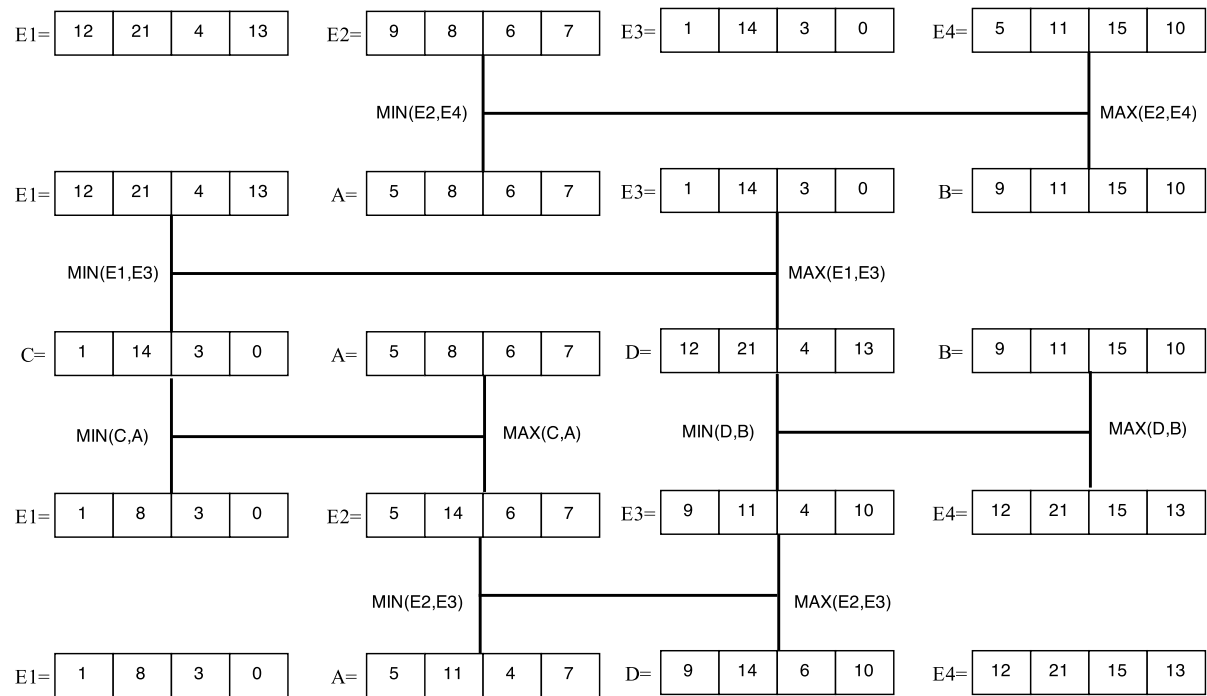
Santiago - Chile
2015

RESUMEN EJECUTIVO

El presente informe tiene como objetivo mostrar la forma de implementación de la parte SIMD de la presente experiencia. Compuesto de cuatro diagramas, que detallan la traza de los algoritmos utilizados. Se optó por dibujar y detallar en su mayoría solo trazas, debido a que el algoritmo en detalle se puede encontrar en la investigación que se menciona en la bibliografía. Los algoritmos que se grafican son "Ordenamiento in Register", "Bitonic Merge Network" y "Merge SIMD de secuencias".

Traza Ordenamiento In Register

Ordenamiento por columna (lane)



Traspuesta del resultado anterior

Para realizar la traspuesta de:

1	8	3	0
5	11	4	7
9	14	6	10
12	21	15	13

Se aplicó la función

`_MM_TRANSPOSE4_PS` (`__m128 row0`, `__m128 row1`, `__m128 row2`, `__m128 row3`)

Que realiza la operación
`__m128 tmp3, tmp2, tmp1, tmp0;`
`tmp0 = _mm_unpacklo_ps(row0, row1);`
`tmp2 = _mm_unpacklo_ps(row2, row3);`
`tmp1 = _mm_unpackhi_ps(row0, row1);`
`tmp3 = _mm_unpackhi_ps(row2, row3);`
`row0 = _mm_movelh_ps(tmp0, tmp2);`
`row1 = _mm_movehl_ps(tmp2, tmp0);`
`row2 = _mm_movelh_ps(tmp1, tmp3);`
`row3 = _mm_movehl_ps(tmp3, tmp1);`

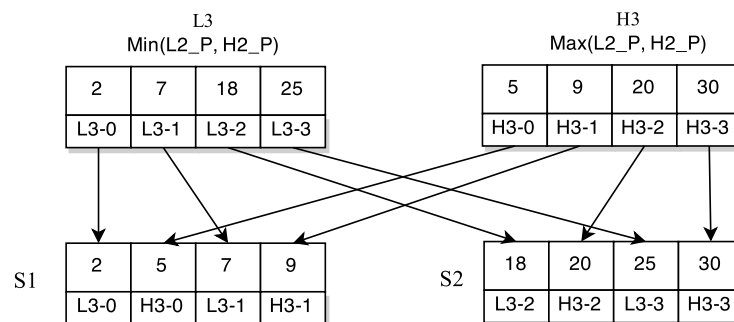
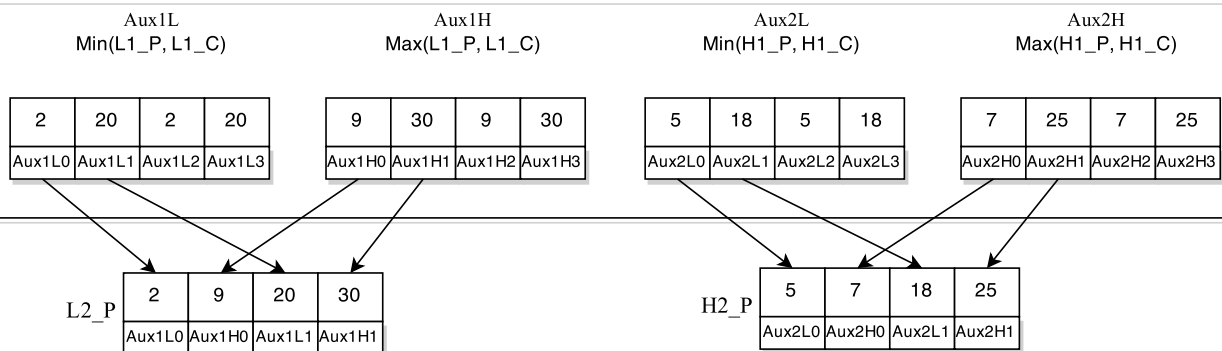
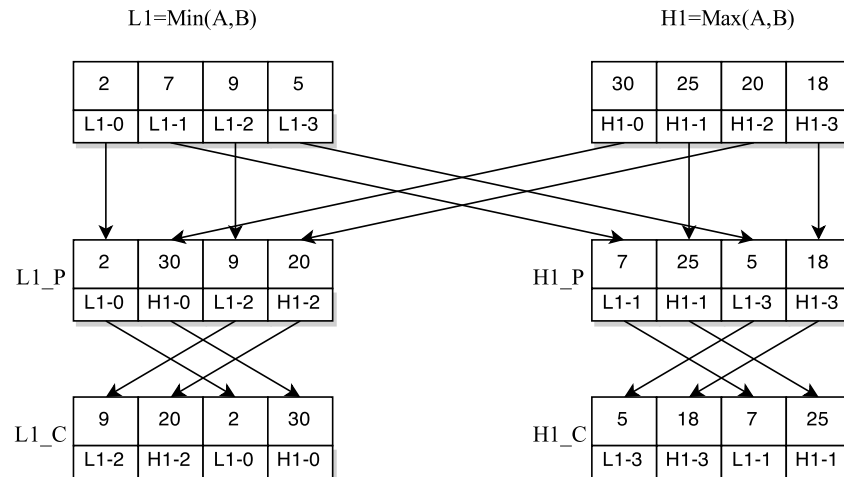
Retornando:

1	5	9	12
8	11	14	21
3	4	6	15
0	7	10	13

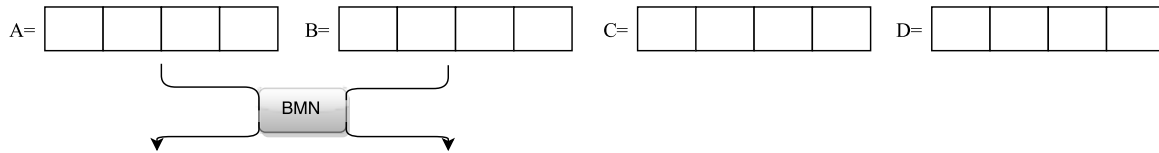
Traza algoritmo de Bitonic merge Network

*B tiene que ser bitonica
decreciente, para ello se
aplicaba un Shuffle sobre
si mismo cambiando las
posiciones de los valores
3-2-1-0 a 0-1-2-3*

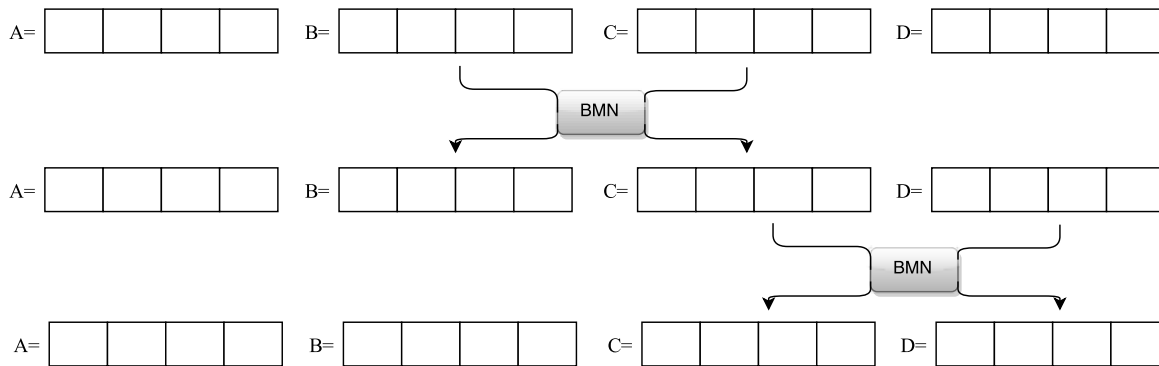
$$A = \begin{bmatrix} 2 & 7 & 9 & 18 \end{bmatrix}$$

$$B = \begin{bmatrix} 30 & 25 & 20 & 5 \end{bmatrix}$$


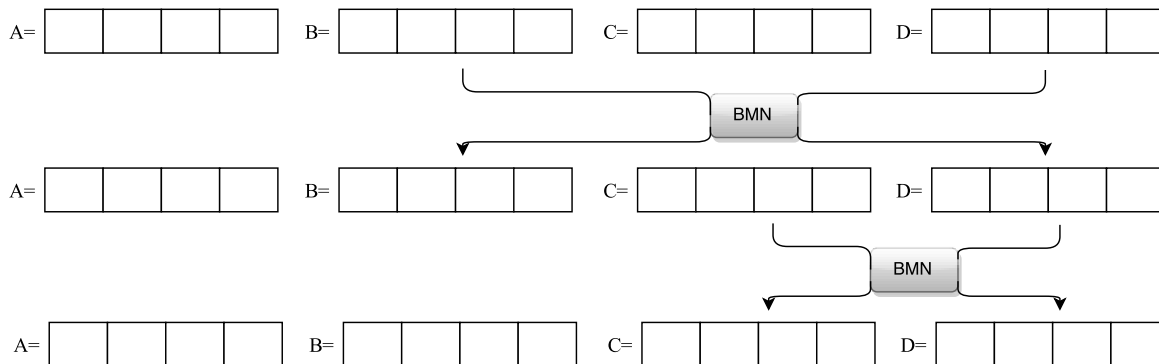
Algoritmo ordenamiento Merge SIMD



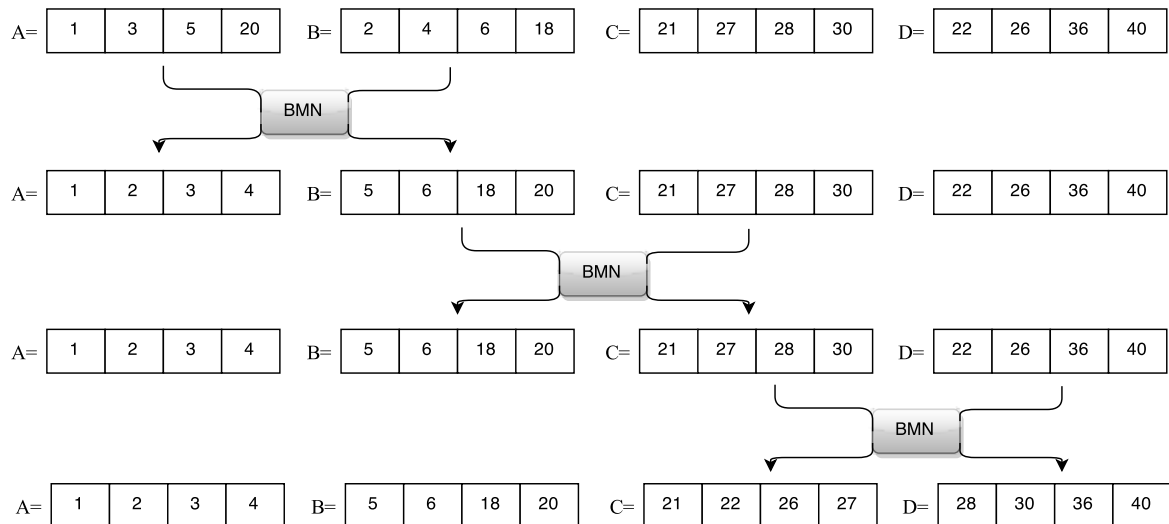
Si la posición 0 de C es menor que la posición 0 de D (`_mm_ucomile_ss(*C,*D)==1`)



Caso contrario



Traza ordenamiento Merge SIMD



Algoritmo obtenido de la investigación de Jatin Chhugani, Anthony D. Nguyen, Victor W. Lee, William Macy, Mostafa Hagog, Yen-Kuang Chen, Akram Baransi, Sanjeev Kumar, and Pradeep Dubey. (2008)

BIBLIOGRAFÍA:

Jatin Chhugani, Anthony D. Nguyen, Victor W. Lee, William Macy, Mostafa Hagog, Yen-Kuang Chen, Akram Baransi, Sanjeev Kumar, and Pradeep Dubey. 2008. Efficient implementation of sorting on multi-core SIMD CPU architecture. *Proc. VLDB Endow.* 1, 2 (August 2008), 1313-1324. DOI=10.14778/1454159.1454171 <http://dx.doi.org/10.14778/1454159.1454171>