



Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
CE4302 Arquitectura de Computadores 2

Ing. M.Sc Jeferson González Gómez

Taller 2
Extensión SIMD: SSEx

Estudiantes:
Andrés Vargas Rivera 2015097412

Semestre 2
2018

Taller 2

1. 1. ¿Qué es el set SSE, cuál es su utilidad y qué aplicaciones tiene?

- Streaming SIMD Extensions(SSE) es un set de instrucciones que permite realizar operaciones bajo una taxonomía SIMD en arquitecturas x86, fue diseñado por Intel e introducido por primera vez en el procesador Pentium 3, pueden aumentar de manera importante la eficiencia en múltiples datos que van a ser procesados por la misma instrucción.
- Su utilidad es mejorar el rendimiento de sistemas en los cuales se realiza la misma operación sobre diferentes datos de manera que la instrucción en su proceso de ejecución logra adquirir los diferentes datos que van a ser procesados.
- Aplicaciones:
 - Procesamiento gráfico.
 - Procesamiento digital de señales.
 - Operaciones Vectoriales.

2. ¿Cómo realiza la compilación de un código c (.c) que utilice el set SSEx de Intel?

```
gcc -o -mssex test test.c //Donde x representa la versión de SSE
```

3. ¿Qué importancia tienen la definición de variables y el alineamiento de memoria al trabajar con un set SIMD vectorial, como SSE?

- El alineamiento de memoria tiene importancia ya que SSE define paquetes de 16 byte para tipos de datos de 4 a 32 bit por lo que para aumentar la eficiencia en el acceso de los datos este se debe dar en paquetes alineados en 16 byte para que no sea necesario desperdiciar tiempos del clock para acceder a los datos.
- Es importante la definición de variables para asignar el espacio de memoria que va a ser utilizado para poder acceder al dato de manera más eficiente por ejemplo los tipos de variables `__m128` que corresponden a vectores los cuales tienen un tamaño de 128 bits de punto flotante o en la variación `__m128i` para enteros lo cual permite empaquetar una serie de datos para ser accedidos de manera más rápida en forma de un solo conjunto de bits.

Ejercicios Prácticos

1. Con base en el código `hello simd.c`:
 - a. ¿Que realiza el código?

Realiza una sumatoria de dos vectores de 4 posiciones de forma que la posición 0 se suma con la 0, la 1 con la 1 y de esta forma con las demás, dando como

resultado un vector el cual se llama result. Esto se realiza de forma simultánea la suma de cada una de las posiciones de forma simultánea como si fuera solo la suma de un dato con otro, debido a esto es que para este tipo de operaciones se puede optimizar la ejecución de las instrucciones.

b. ¿Qué instrucciones SIMD se están utilizando y con qué fin?

`_mm_set_epi32()`: Conformar un vector de enteros en sus correspondientes posiciones con los valores establecidos.

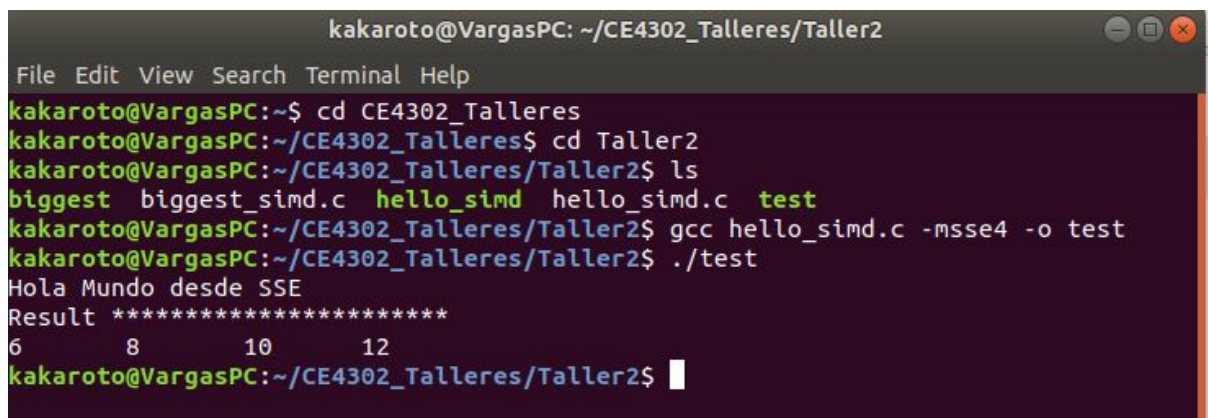
`_mm_add_epi32()`: Suma los enteros de la misma posición teniendo dos vectores de entrada.

`_mm_extract_epi32()`: Extrae un entero de 32 bit situado en el vector en la posición establecida por el segundo argumento que corresponde al inmediato.

c. Versión de SSE

Utiliza la versión SSE2 para las instrucciones de add y set y la versión 4.1 para la instrucción de extract.

d. Screenshot de la compilación del código.



```
kakaroto@VargasPC: ~/CE4302_Talleres/Taller2
File Edit View Search Terminal Help
kakaroto@VargasPC:~$ cd CE4302_Talleres
kakaroto@VargasPC:~/CE4302_Talleres$ cd Taller2
kakaroto@VargasPC:~/CE4302_Talleres/Taller2$ ls
biggest biggest_simd.c hello_simd hello_simd.c test
kakaroto@VargasPC:~/CE4302_Talleres/Taller2$ gcc hello_simd.c -msse4 -o test
kakaroto@VargasPC:~/CE4302_Talleres/Taller2$ ./test
Hola Mundo desde SSE
Result *****
6      8      10     12
kakaroto@VargasPC:~/CE4302_Talleres/Taller2$
```

Link repositorio: https://github.com/avargas10/CE4302_Talleres.git

Referencias

Openmp.org. (2018). [online] Available at:

<https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf> [Accessed 23 Aug. 2018].

Pages.tacc.utexas.edu. (2018). [online] Available at:

<http://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-reduction.html> [Accessed 23 Aug. 2018].

Software.intel.com. (2018). *OpenMP Critical Sections* | Intel® Software. [online] Available at: <https://software.intel.com/en-us/node/608158> [Accessed 23 Aug. 2018].

Yliluoma, J. (2018). *Guide into OpenMP: Easy multithreading programming for C++*. [online] bisqwit. Available at: <https://bisqwit.iki.fi/story/howto/openmp/> [Accessed 23 Aug. 2018].