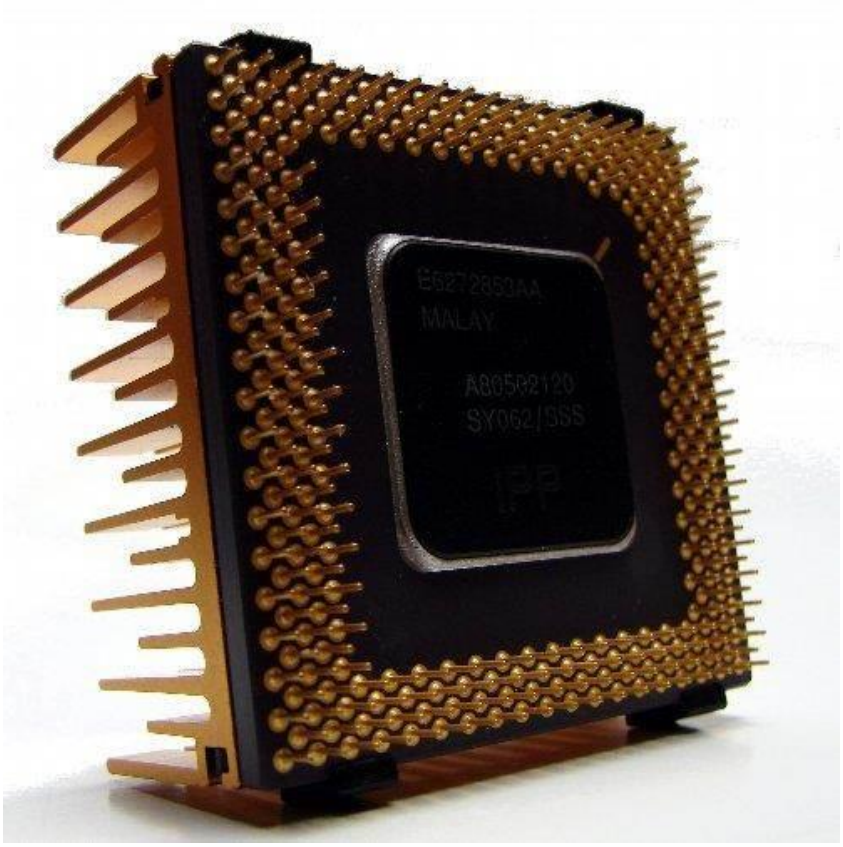
MATERIAL ADICIONAL F-III

- **Ejercicios para desarrollar individualmente**
- **Ejercicio en grupo**

PROYECTO



2017

Proyecto de evaluación del rendimiento

F-III. Implementación de una rutina para comparación de arquitecturas SISD y SIMD

Arquitectura de los Computadores
Grado en Ingeniería Informática
Dpto. Tecnología Informática y Computación
Universidad de Alicante

MATERIAL ADICIONAL F-III

PROYECTO DE EVALUACIÓN DEL RENDIMIENTO

I. CONTENIDOS

Con el objetivo de comparar dos tipos de arquitecturas, en concreto SISD y SIMD, según la taxonomía de Flynn, y para conocer con detalle el repertorio de instrucciones de una arquitectura CISC; la fase III del proyecto consistirá en desarrollar en grupo una rutina empleando el repertorio de instrucciones de los últimos procesadores de Intel y AMD en el que existe una serie de instrucciones específicas para tratamiento de arrays de datos enteros denominadas SSE, junto con unos registros asociados.

Previo a la realización de la rutina en grupo, se pretende que individualmente el alumno conozca las capacidades de paralelización de cálculo que incorporan los procesadores de propósito general. Esto es, comprender cómo se realiza la programación con las extensiones SIMD de las arquitecturas superescalares de Intel. Para ello, se utilizará MICROSOFT VISUAL STUDIO .NET C++ en entorno WINDOWS.

II. DOCUMENTACIÓN EXTENSIONES SIMD (MMX, SSE)

Para la documentación referente a las extensiones SIMD se utilizará la página web a la que se hace referencia abajo como guía de documentación. El enlace contiene información referente a programación con tecnología MMX y SSE, descripción de instrucciones, modelo de trabajo, ejemplos, etc…

http://www.tommesani.com/Docs.html

Adicionalmente, es muy interesante la consulta de documento de Intel 'Intel® 64 and IA-32 Architectures Developer's Manual: Vol. 1'. En el capítulo 5 se incluye una descripción del juego de instrucciones de IA-32 incluido el juego de instrucciones MMX, SSE, SSE2 y SSE3. Además, los capítulos 9 y 10 proporcionan información sobre el desarrollo de programas empleando instrucciones MMX y SSE, respectivamente.

http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-vol-1-manual.html

III. ENTREGA Y EVALUACIÓN

La evaluación se realizará sobre el desarrollo de la práctica en clase (individual y en grupo). Las entregas serán:

- **Individual**: Memoria explicativa de los ejercicios del apartado IV
- **Grupo**: Código de la rutina elegida y memoria explicativa de la rutina desarrollada incluyendo comparativa de rendimiento.

IV. EJERCICIOS INDIVIDUALES

En esta parte se pretende testear varios algoritmos utilizando las tecnologías MMX/SSE para ello se propone compilar y ejecutar los siguientes proyectos, y analizar los resultados obtenidos:

NET Bubble Sort SEE (comparando assembler + C)

Las cuestiones a resolver son:

- ¿Qué hace el programa?
- ¿Se utilizan extensiones MMX o SSE?
- Comenta el funcionamiento de la función: *void sortDoublesSSE(Int32 byteCount, double* values)*
- ¿Qué ganancia obtenemos con el algoritmo MMX/SSE con respecto al algoritmo secuencial? Realiza una batería de pruebas y muéstralo utilizando gráficas explicativas.

NET Data Transfer (comparando assembler + C)

Las cuestiones a resolver son:

- ¿Qué hace el programa?
- Explica las siguientes instrucciones: shufpd, cmpltpd, movmskpd
- Explica el funcionamiento de la siguiente función: int DataTransferOptimised(int* piDst, int* piSrc, unsigned long SizeInBytes)
- ¿Qué ganancia obtenemos con el algoritmo optimizado mediante extensiones SIMD con respecto al algoritmo secuencial? Realiza una batería de pruebas y muéstralo utilizando gráficas explicativas.

NET Inner Product (comparando assebler + C)

Las cuestiones a resolver son:

- ¿Qué hace el programa?
- Comenta la siguiente función: float sse3_inner(const float* a, const float* b, unsigned int size)
- ¿Qué ganancia obtenemos con el algoritmo optimizado mediante extensiones SIMD con respecto al algoritmo secuencial? Realiza una batería de pruebas y muéstralo utilizando gráficas explicativas.

NET MatrixVectorMult (comparando assembler + C)

Las cuestiones a resolver son:

- ¿Qué hace el programa?
- ¿Qué ganancia obtenemos con el algoritmo optimizado mediante extensiones SIMD con respecto al algoritmo secuencial? Realiza una batería de pruebas y muéstralo utilizando gráficas explicativas.

V. EJERCICIO EN GRUPO

Se pretende la realización de una rutina en grupo, en primer lugar con lenguaje C, en segundo lugar en lenguaje ensamblador 8086 genérico (esta implementación es **optativa**) y, finalmente, nuevamente en ensamblador pero incorporando instrucciones SSE. El grupo deberá realizar distintas pruebas que proporcionen unos resultados en forma de gráficas sobre el tiempo de ejecución, en las que se observará la optimización realizada por el compilador utilizado así como la ganancia que supone la utilización de la tecnología SSE basada en el paradigma SIMD.

La rutina a implementar es libre. El grupo puede proponer un algoritmo o elegir uno de los que se comentan más adelante. Si el grupo propone un algoritmo, este debe desarrollar de forma intensiva una tarea que sea propicia para ser implementada con las tecnologías MMX y SSE (por ejemplo, tareas de procesamiento de imagen). Esto es que intervengan un flujo múltiple de datos sobre la misma operación (SIMD). El algoritmo se propondrá al profesor correspondiente del turno de prácticas para su validación.

El grupo puede considerar desarrollar una rutina entre las siguientes:

- Obtención del vector normal a un punto en función de los vecinos más cercanos (anterior y siguiente). Se considerará un conjunto amplio de puntos.

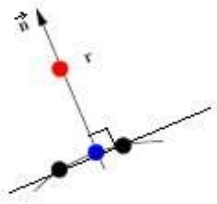


Figura 1 - Normal a un punto

- Cálculo de la recta de regresión a partir de una dispersión de puntos.

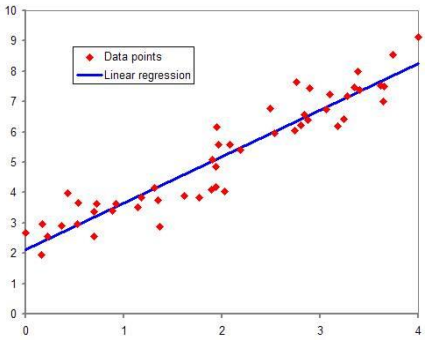


Figura 2 – Recta regresión

- Métodos de interpolación polinomial: Lagrange, Newton, Neville, etc. Seleccionar uno.

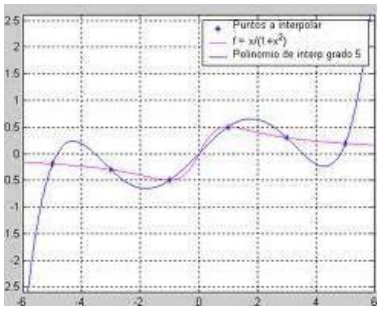


Figura 3 – Polinomios interpolación