

TAREAS A REALIZAR

1.- IMPLEMENTACIÓN DE CÓDIGO

Implementen el algoritmo de burbuja (*bubble-sort*) en el subconjunto de MIPS que usa SIMDE. Asuman que trabajarán con enteros. Tienen ejemplos y detalles en [Wikipedia](#). El algoritmo debe asumir que se tiene un tamaño de vector en la posición 40 de memoria y, a partir de esa posición, los valores del vector. Tienen un fichero de ejemplo con 16 elementos en “order1.mem”.

2.- EJECUCIÓN TEÓRICA

Estimen teóricamente el tiempo de ejecución en ciclos de su programa para el fichero de ejemplo, asumiendo que disponen de una máquina secuencial cuyas latencias para cada tipo de instrucción son las mismas que las utilizadas por defecto en SIMDE.

3.- EJECUCIÓN DEL CÓDIGO EN LA MÁQUINA SUPERESCALAR

a) Analicen el rendimiento de la máquina con el programa implementado. Usen como datos de prueba los definidos en el fichero “order1.mem”. Empleen la configuración por defecto de la máquina superescalar, asumiendo que no hay fallos de caché. Calculen la aceleración con respecto a la ejecución secuencial.

b) Repitan el análisis, pero probando con los siguientes porcentajes de fallos de caché: 10%, 20%. Recuerden hacer el análisis lanzando varias simulaciones (Entre 20 y 50 debería bastar) y quédense con el promedio. Calculen la aceleración en cada caso frente a la versión sin fallos de caché.

c) Analicen para qué unidades funcionales sería más eficiente tratar de reducir la latencia. Asuman que el porcentaje de fallos de caché se mantendría en torno al 10%. Si es posible, apliquen la ley de Amdahl y la fórmula de rendimiento general de la asignatura.

d) Manteniendo las latencias por defecto, definan un esquema de máquina aumentando el grado de emisión y/o el número de unidades funcionales que sea lo más eficiente posible. Justificar las decisiones tomadas y las descartadas. Para definir la eficiencia, supongan que la arquitectura inicial cuesta 100 € y que incrementar el grado de emisión cuesta 20 € por instrucción emitida adicional. Además, añadir una nueva UF cuesta 5 €, independientemente del tipo. Calcula la eficiencia como el incremento del coste dividido por la aceleración obtenida.

e) Manteniendo las latencias por defecto, definan un esquema de máquina disminuyendo el grado de emisión y/o el número de unidades funcionales que sea lo más eficiente posible. Justificar las decisiones tomadas y las descartadas. Para definir la eficiencia, supongan que la arquitectura inicial cuesta 100 € y que decrementar en una unidad el grado de emisión ahorra 10 €. Además, eliminar una UF ahorra 2 €, independientemente del tipo. Calcula la eficiencia como el incremento del coste dividido por la aceleración obtenida.

ENTREGABLES

Se habilitará una tarea en Moodle donde se entregará un fichero comprimido con todos los códigos desarrollados. En la misma tarea se subirá el fichero de una presentación que habrá que hacer en clase (de unos 10 minutos).

EVALUACIÓN

Se evaluará, por un lado, la calidad de la presentación y del trabajo crítico realizado (profundidad del trabajo, capacidad de análisis, coherencia de las propuestas...). Se realizará también una pequeña competición usando como referencia la tabla siguiente (servirá para poner nota en la parte de “actividades prácticas en laboratorio”). Cada persona debe subir sus resultados a un [documento compartido](#).

Ítem (siempre referido al peor caso)	Puntuación
Código secuencial que tarde menor número de ciclos	1
Menor número de ciclos en la máquina superescalar por defecto	2
Mayor aceleración (superescalar frente a secuencial)	1
Menor penalización al aumentar el porcentaje de fallos de caché	1
Para el apartado d)	
Menor número de ciclos en la máquina adaptada	1
Mayor aceleración (adaptada frente a original)	1
Mejor ratio coste / aceleración adaptación	2
Para el apartado e)	
Menor número de ciclos en la máquina adaptada	1
Mayor aceleración (adaptada frente a original)	1
Mejor ratio coste / aceleración adaptación	2
Otros	Hasta 3 puntos