

## PRÁCTICA 2:

### *Implementación y optimización de un cálculo en ensamblador DLX*

#### 1. Descripción de la práctica

El objetivo de la práctica es el desarrollo y optimización de un código que realice el siguiente cálculo:

$$M = (A \times B) \cdot \frac{1}{|A + B|} + C \cdot \alpha$$

siendo:

- $A, B, C$  y  $M$  matrices 3x3
- $\alpha$  número real
- $|A|$  determinante de la matriz
- $\times, +$  producto y suma de matrices
- $\bullet$  producto de matriz por escalar

Datos de entrada y salida en DLX:

```

;;;;;;;;;;;;;;;; NO MODIFICAR ORDEN

A:      .float      1.500000,      2.750000,      3.257000
        .float      21.002000,      2.658000,      2.157000
        .float      56.251000,      3.154000,      3.255000

B:      .float      3.500000,      3.500000,      2.500000
        .float      4.500000,      4.500000,      6.500000
        .float      5.500000,      1.500000,      2.500000

C:      .float      15.000000,      17.000000,      14.000000
        .float      17.000000,      15.000000,      17.000000
        .float      14.000000,      17.000000,      15.000000

Alfa:   .float      1.235

M:      .float      0.0,0.0,0.0
        .float      0.0,0.0,0.0
        .float      0.0,0.0,0.0

;;;;;;;;;;;;;;;; FIN NO MODIFICAR

```

#### 2. Se pide

- a) Realizar una versión no optimizada que realice el cálculo pedido
- b) Optimizar el cálculo realizado en a) empleando las técnicas habituales de uso de registros adicionales, reordenación de código, desenrollamiento de bucles (si los hay), etc.
- c) Se debe mantener el orden de las variables de entrada y salida en memoria. Se deben comprobar las posibles divisiones por 0. En ambos casos el resultado debe ser almacenado en la matriz M.
- d) Los valores de entrada se pueden cambiar.

#### 3. Se deberá entregar

- a) Las dos versiones del programa (normal y optimizada), comentadas.
- b) Se entregará un breve documento explicando las mejoras realizadas y comparación de resultados obtenidos

**Las pruebas a realizar se harán con la siguiente configuración:**

CONFIGURACIÓN	
Memory size:	0x8000
faddEX-Stages:	1
faddEX-Cycles:	2
fmulEX-Stages:	1
fmulEX-Cycles:	5
fdivEX-Stages:	1
fdivEX-Cycles:	19
Forwarding:	enabled

ESTADÍSTICAS	
Total	
Nº de ciclos:	
Nº de instrucciones ejecutadas (IDs):	
Stalls	
RAW stalls:	
LD stalls:	
Branch/Jump stalls:	
Floating point stalls:	
WAW stalls:	
Structural stalls:	
Control stalls:	
Trap stalls:	
Total	
Conditional Branches	
Total:	
Tomados:	
No tomados:	
Instrucciones Load/Store	
Total:	
Loads:	
Stores:	
Instrucciones de punto flotante	
Total:	
Sumas:	
Multiplicaciones:	
Divisiones:	
Traps	
Traps:	

#### 4. Lugar de entrega

La entrega se realizará en Studium en las fechas indicadas. Se subirá un único archivo (.zip, .rar, etc.) que contenga lo contemplado en el punto 3.

#### 5. Evaluación de la práctica

Para aprobar la práctica se deberán de entregar las dos versiones y que el resultado sea correcto, para cualquier valor de entrada. A partir de ahí según lo entregado se obtendrá mayor o menor calificación (número de ciclos empleados para la ejecución, documentación entregada,...). La nota final obtenida por cada persona en las prácticas vendrá corregida por un factor real comprendido entre 0 y 1 según la defensa realizada de las mismas.

La detección de copia parcial o total de la práctica conllevará la suspensión de las prácticas, y por tanto de la asignatura.

Cualquier modificación de la práctica se notificará en Studium.