

Implementación y optimización de un algoritmo en ensamblador DLX

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

$$a_1, a_2, a_3, a_4, a_5$$

siendo:

- $$V(a_1, a_2, a_3, a_4) = \begin{bmatrix} 1 & a_1 & a_1^2 & a_1^3 \\ 1 & a_2 & a_2^2 & a_2^3 \\ 1 & a_3 & a_3^2 & a_3^3 \\ 1 & a_4 & a_4^2 & a_4^3 \end{bmatrix}$$

[illegible]

tamano: .word 20

```

;;;;; VARIABLES DE SALIDA: NO MODIFICAR ORDEN
; m11, m12, m13, m14
; m21, m22, m23, m24
; m31, m32, m33, m34
; m41, m42, m43, m44

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

checkM: .float    0.0

; Medias
a1:     .float    0.0
a2:     .float    0.0
a3:     .float    0.0
a4:     .float    0.0
a5:     .float    0.0
checkA: .float    0.0

;;;;; FIN NO MODIFICAR ORDEN

```

2. Se pide

- Realizar una versión no optimizada que realice el cálculo pedido
- Optimizar el cálculo realizado en a) empleando las técnicas habituales de uso de registros adicionales, reordenación de código, des-enrollamiento de bucles (si los hay), etc.
- Se debe mantener el orden de las variables de entrada y salida en memoria. Se deben comprobar las posibles divisiones por 0 (y salir sin que salga el mensaje de error (directamente ir al trap 0).
- En ambas versiones el resultado debe ser almacenado en M , $checkM$, $a1, a2, a3, a4, a5$ y $checkA$
- Los valores de entrada se pueden cambiar.

3. Se deberá entregar

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

Las pruebas a realizar se harán con la siguiente configuración, y para la comparación de ciclos se empleará el tamaño 20:

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 en la versión optimizada en relación a la mejor práctica (la que haya conseguido un menor número de ciclos de manera correcta, 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.

Cualquier modificación de la práctica se notificará en Studium.