

Nombre: Daniel Martínez Bordes

Grupo: 21

Nombre: \_\_\_\_\_

## Hoja de respuesta al Estudio Previo

1. El número -23,75 en formato IEEE se expresa:

En simple precisión:

1 1 000000 11011111 00000000  $\equiv -1,011111 \cdot 2^{17}$

En doble precisión:

1 1 000000 00 11011111 00000000  $\equiv -1,011111 \cdot 2^{1023}$

2. Dado el número 1048576,2:

Se codifica exacto en simple precisión (S/N): N

Error en simple precisión: 0,075

Se codifica exacto en doble precisión (S/N): N

3. Las instrucciones ensamblador: flds, fmul, fadd y fstps sirven para:

Operaciones  
como float  
registro  
FPU  
simple  
o doble  
precisión

flds: Carga el valor del 1º operando en el registro FPU como float num.  
fmul: Multiplica los operandos como float y almacena el resultado en el reg. dest.  
fadd: Suma los operandos y almacena el resultado en el reg. dest.  
fstps: Copia ST(0) a el operando dest. (reg. pro. mem.) en format como float y se un pop de stack

4. El primer código en ensamblador se ejecuta:

MIPS: 1280,77 Mipo

MFLOPS: 196,92 mflops

El segundo código en ensamblador se ejecuta:

MIPS: 815,909 mips

MFLOPS: 232,727 mflops

Speedup con respecto al primer código: 18,18%

Comenta de forma crítica los resultados anteriores:

Podemos observar que aunque el 1º código ejecuta más instrucciones/s su tiempo de ejecución es mayor y el rendimiento disminuye en consecuencia esto es debido a que se debe ejecutar más instrucciones para realizar la misma operación (menos óptimo)

Dado que el procesador es el mismo el tiempo de ejecución se debe a la optimización del código. Realiza más op. de como floatante.

Nombre: -10

4+123 = 131  
Grupo: 1000

Nombre: \_\_\_\_\_

### Hoja de respuestas de la práctica

- El resultado de la operación  $z = x^4 - 4y^4 - 4y^2$  con  $x = 665857$  y  $y = 470832$  en doble precisión es:
- El resultado de la operación  $z = x^4 - 4y^4 - 4y^2$  con  $x = 665857$  y  $y = 470832$  en simple precisión es:
- El resultado de la operación  $z = x^4 - 4y^4 - 4y^2$  con  $x = 665857$  y  $y = 470832$  en enteros es:
- Calculado con medios (bc, Google, Calculadora, etc.) externos es:

bc:

Nombre Medio 2:

Resultado Medio 2:

$$\begin{aligned} \text{error} &= |15.1 \text{epr} - 15.1 \text{epr}| \\ &= 1048576,2 - 1048576,125 \end{aligned}$$

Explica cuál y por qué es el resultado correcto de los anteriores. ¿Qué problemas genera lo que habéis descubierto y cuáles son las posibles soluciones?

$$\text{MIPS} = \frac{\text{millions instr}}{\text{seg}} = \frac{\# \text{instr}}{10^{-6} \text{ seg}} = \frac{3330}{10^{-6} \cdot 2,6 \cdot 10^6} = 1280,77$$

$2 + 256 \cdot 13 =$

$$\begin{aligned} \text{MIPS} &= \frac{1395}{10^{-6} \cdot 2,2 \cdot 10^6} = 815,909 \end{aligned}$$

$3 + 256 \cdot 7 = 1795$

5. El programa en C de la práctica ejecuta:

Instrucciones:

Segundos:

Operaciones de Coma Flotante:

MIPS:

MFLOPS:

6. Optimizado el programa ejecuta:

Instrucciones:

Segundos:

Operaciones de Coma Flotante:

MIPS:

MFLOPS:

Speedup:

Nombre: \_\_\_\_\_

Nombre: \_\_\_\_\_

Grupo: \_\_\_\_\_