

Preguntas de repaso teórico. Se sugiere justificar todas las respuestas para un mejor entendimiento de la teoría

1. ¿Qué es el ISA?
2. ¿Por qué es incorrecto basar el rendimiento de un computador únicamente en MIPS, CPI y FLOPS?
3. ¿Cuál es la función del *program counter*?
4. ¿Por qué se suele separar la memoria caché L1 en datos e instrucciones?
5. Proponer un ejemplo de localidad espacial y uno de temporal.
6. (V o F) Las matrices que no son cuadradas pueden ser almacenadas en row-major o en col-major sin problemas ya que su lectura estará determinada por la lectura row-view y col-view.
7. (V o F) Una arquitectura RISC no tiene sistema operativo; por tanto, no necesita tener instrucciones que accedan a memoria además de load y store.
8. (V o F) Si un procesador tiene un modo de trabajo de 2GHz y un modo turbo de 3.5 GHz, implica que puede cambiar su CPI ante un requerimiento del sistema.
9. (V o F) Según la Jerarquía de memorias, no es posible que en un solo nivel se encuentren dos memorias destinadas para diferentes acciones.
10. (V o F) La latencia inherente del sistema operativo no debería de considerar los accesos a memoria caché debido a que se suele tener hit ratios altos y, estadísticamente hablando, no representan una unidad significativa.

1) El ISA es

Arquitectura del conjunto de instrucciones.
Instruction Set Architecture.

parte de la arquitectura del CPU • Definido como un grupo de comandos implementados en la

$$T_{ejec} = \frac{NI}{MIPS \times 10^6}$$

2)

MIPS: Depende del ISA. No es consistente.
Depende del programa.

CPI: ciclos por instrucción. Depende del ISA, pero influyen más factores en el rendimiento, como la frecuencia del procesador.

FLOPS: Solo se centran en las operaciones de punto flotante.

3) Program counter

↳ Es un registro del CPU, que indica la posición donde está el CPU en su secuencia de instrucciones.
↳ parte de la unidad central

4) Caché L1 datos
instrucciones

I: guarda información sobre la operación que el CPU debe realizar.

D: guarda los datos sobre los que se realizará la operación.

However, most programs will exhibit very different patterns of location for their instructions and their data. This means that it would be unlikely for the instructions and data to be able to share the cache efficiently. Because instructions and data aren't necessarily near each other in memory. A data access would bump instruction from the cache, and loading instructions would bump data from the cache.

Separar Datos e instrucciones en caché, permite que los accesos sean menos aleatorios y más agrupados.

6) $\left[\begin{array}{c} \\ \\ \end{array} \right]$

Verdadero,

5) Localidad

- Temporal: En un bucle, se requiere la lectura de una variable repetidas veces. en un mismo espacio de memoria.

- Espacial: Se requiere recorrer un arreglo

7) RISC sin sistema operativo
no necesita más que load y store.

Verdadero, todas las operaciones se hacen registro - registro

8) Con modo turbo se puede cambiar el CPI? Falso

$$CPI = \frac{\sum_{i=1}^N CPI_i I_i}{I_c} \Rightarrow \text{la frecuencia no influye en el CPI.}$$

9) En un mismo nivel, 2 memorias para diferentes acciones.

Falso, Cache L1 se divide en datos e instrucciones.

10) Se debería considerar el acceso a caché?