Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Tarea 2: RISC vs CISC

Curso: Arquitectura de Computadores I

Profesor: Dr.-Ing. Jeferson González Gómez

Asistente: Carlos Eduardo Rodriguez Segura

Estudiante: Jose Eduardo Campos Salazar

Fecha de entrega: 30 de agosto

Índice

1. Pregunta teórica

Explique brevemente: ¿Por qué las arquitecturas RISC suelen ser más predecibles en sus tiempos de ejecución que las arquitecturas CISC? ¿En qué situaciones es importante esta predictibilidad?

Las arquitecturas **RISC** suelen ser más predecibles en sus tiempos de ejecución porque cada instrucción es simple y realiza una sola operación elemental (por ejemplo, cargar, sumar o guardar). Esto permite que el tiempo de ejecución de cada instrucción sea uniforme y constante.

En cambio, las arquitecturas **CISC** tienen instrucciones más complejas que realizan múltiples operaciones en una sola instrucción, lo que hace que su tiempo de ejecución sea más variable y, por tanto, menos predecible.

Esta predictibilidad es muy importante en **sistemas que requieren ejecución determinista**, como en sistemas de **tiempo real**, o en arquitecturas que implementan **paralelismo y pipeline**, donde es necesario optimizar el flujo de instrucciones y minimizar riesgos de dependencia.

2. Resultados de la simulación

Se modelaron las arquitecturas **CISC** y **RISC** en Python para realizar la suma de dos vectores de 10 elementos.

2.1. CISC

La arquitectura CISC utiliza una instrucción `SUMMEM`, la cual en un solo paso:

- Trae dos datos contiguos de memoria.
- Los suma.
- Guarda el resultado en memoria.

Ejecución (fragmento):

```
SUMMEM 1 + 2 = 3 → 21
SUMMEM 2 + 4 = 6 → 22
...
SUMMEM 10 + 20 = 30 → 30
```

Vector resultante (Mem[21-30]):

$$3, 6, 9, 12, 15, 18, 21, 24, 27, 30$$

Estadísticas:

- Cantidad de instrucciones ejecutadas: 10
- Cantidad de ciclos: 30

2.2. RISC

La arquitectura RISC emplea instrucciones simples: `TRAER`, `SUMAR`, `ESCRIBIR`.

Ejecución (fragmento):

```
TRAER 1 → R1 = 1
TRAER 11 → R2 = 2
SUMAR R1(1)+R2(2) → R3 = 3
ESCRIBIR R3 (3) → 21
...
TRAER 10 → R1 = 10
TRAER 20 → R2 = 20
SUMAR R1(10)+R2(20) → R3 = 30
ESCRIBIR R3 (30) → 30
```

Vector resultante (Mem[21-30]):

$$3, 6, 9, 12, 15, 18, 21, 24, 27, 30$$

Estadísticas:

- Cantidad de instrucciones ejecutadas: 40
- Cantidad de ciclos: 40

3. Análisis comparativo

Característica	CISC	RISC
Instrucciones ejecutadas	10	40
Ciclos totales	30	40
Complejidad de instrucciones	Alta	Baja
Predictibilidad	Baja	Alta

En esta simulación:

- CISC fue más eficiente en ciclos totales (30 vs 40).
- RISC ofrece mayor regularidad, lo que facilita paralelismo y optimizaciones.

4. Conclusión

La práctica muestra que **CISC puede ser más eficiente en ciertos casos**, ya que agrupa varias operaciones en una sola instrucción. Sin embargo, **RISC ofrece mayor predictibilidad y uniformidad**, cualidades muy valiosas en arquitecturas modernas donde se aplican técnicas avanzadas como **pipeline, paralelismo a nivel de instrucción y ejecución fuera de orden**.

En resumen, **CISC optimiza el número de instrucciones, mientras que RISC optimiza la simplicidad y regularidad del hardware**, lo que repercute en la escalabilidad del diseño.