

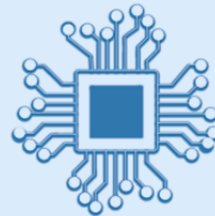
Taller de Sistemas Digitales (E1225) - Arquitectura de Computadores I (E0225)

Curso: 2025

TRABAJO PRÁCTICO N°:1

Introducción

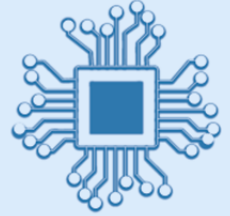
1. Definir la Ley de Amdahl y graficar para distintos valores de fracción de código paralelizado y número de procesadores. ¿De qué manera se puede generalizar las aplicaciones de esta ley?
2. Definir y graficar la Ley de Gustafson-Barsis. Graficar para distintos valores de fracción de código paralelizado y número de procesadores y analizar. Comparar resultados con el punto anterior y sacar conclusiones.
3. Supongamos que estamos considerando una mejora del procesador, de un sistema servidor que se utiliza para un servidor web. La nueva CPU es 10 veces más rápida en el cálculo de la aplicación web que el procesador original. Suponiendo que la CPU está inactiva durante 20% del tiempo, haciendo cálculos durante el 30% del tiempo y a la espera de E/S durante el 50% del tiempo, ¿cuál es la mejora de desempeño (Speedup) en este caso?
4. Supongamos que hacemos una actualización a un equipo que mejora el modo de ejecución en un factor de 15. Este nuevo “modo rápido” se utiliza durante el 40% del tiempo, medido como porcentaje del tiempo de ejecución cuando el “modo rápido” está en uso. ¿Cuál es la mejora de desempeño (Speedup) que podemos lograr?
5. Un programa cuenta con el 10% de instrucciones de división. Todas las instrucciones que no dividen llevan un ciclo de ejecución. Todas las instrucciones de división llevan 50 ciclos.
 - a. ¿Cuál es el CPI de este programa en este procesador?
 - b. ¿Qué porcentaje de tiempo se gasta sólo haciendo divisiones?
 - c. ¿Cuál debería ser el Speedup si al aumentar la velocidad se divide el tiempo de ejecución de la instrucción de división por 2x?
 - d. ¿Cuál debería ser el Speedup si al aumentar la velocidad se divide el tiempo de ejecución de la instrucción de división por 5x?
 - e. ¿Cuál debería ser el Speedup si al aumentar la velocidad se divide el tiempo de ejecución de la instrucción de división por 10x?
 - f. ¿Cuál debería ser el Speedup si al aumentar la velocidad se divide el tiempo de ejecución de la instrucción de división por 50x?
 - g. ¿Cuál debería ser el Speedup si las instrucciones de división son infinitamente rápidas (cero ciclos)?



Taller de Sistemas Digitales (E1225) - Arquitectura de Computadores I (E0225)

6. Se ejecutan dos programas P1 y P2 en dos computadoras distintas.
La computadora A tiene un procesador 68000 con una frecuencia de reloj de 2,5 Ghz. La computadora B un x86 y tiene una frecuencia de reloj de 3GHz. En promedio, los programas compilados generan **1,5 veces** más instrucciones en la computadora A que en la B.
 - a. Para el Programa P1, el equipo A tiene un CPI de 2 y el equipo B tiene un CPI de 3. ¿Qué equipo es más rápido para P1?
 - b. Para el Programa P2, el equipo A tiene un CPI de 1 y el equipo B tiene un CPI de 2. ¿Qué equipo es más rápido para P2?
7. Realizar un gráfico de Tareas en función del tiempo de una línea de montaje de una lámpara de pie, teniendo en cuenta:
 - a. Un solo obrero tornea la base, el pie, la lámpara, hace el cableado y pone la pantalla.
 - b. Distintos obreros hacen cada uno las bases, agregan pies a las bases, instalan lámparas y cablean, independientemente y uno más va agregando las pantallas.
 - c. Calcular el tiempo que llevará hacer 10 lámparas en cada uno de los casos. ¿Qué sucede si el cableado tarda mucho más que el resto de las tareas?
(Organización del trabajo – Taylor – 1911)
8. La ejecución de una instrucción en un procesador se compone de:
 - Búsqueda de la instrucción (FI)
 - Decodificación de la Instrucción (DI)
 - Ejecución del Código de Operación (CO)
 - Búsqueda de Operandos (FO)
 - Ejecución de la Instrucción (EI)
 - a. Hacer el esquema de un pipeline de ejecución de instrucciones.
Suponiendo que las distintas partes tarden:
FI = 6 ns, DI = 5 ns, CO = 8 ns, FO = 9 ns, EI = 6 ns. Retardo de latch = 1 ns
 - b. Realizar un diagrama de tareas vs. Tiempo para la ejecución de 5 instrucciones del mismo tipo.
 - c. Calcular el Speedup, frecuencia máxima de funcionamiento del pipeline, rendimiento y ancho de banda.
 - d. ¿Qué sucede cuando hay accesos a memoria para instrucciones y datos en forma concurrente.
9. Defina riesgos en un pipeline: RAW, WAW y WAR
 - a. Dado el siguiente programa decir en qué instrucciones habrá tales riesgos.

1 **MOV R1 <- R2**



Taller de Sistemas Digitales (E1225) - Arquitectura de Computadores I (E0225)

```
2  MOV M <- R1
3  MOV R2 <- R3
4  MOV M <- R2
5  MOV R2 <- R4
```

10. Para cada uno de los tipos de riesgos implemente una pequeña porción de código en la que aparezcan, con una estructura de pipeline de 5 etapas. Efectúe un diagrama de tiempos y simule.

11. Defina riesgos por dependencia de control en un pipeline

- ¿Qué efecto produce un salto condicional al decidirse por sí?
- Explique los diversos tipos de soluciones que se implementan:
 - *Flujos Múltiples
 - *Prefetch del destino del salto
 - *Branch Prediction
 - *Salto retardado
- Ejemplifique con una porción de código y haga el diagrama de tiempos. Simule.

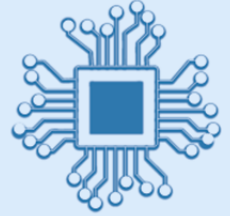
12. Calcular CPI promedio y Speedup en caso de ejecutar el siguiente programa con un pipeline o sin él:

```
program Ex1
for n = 1 to 20
p = p + 1
next
end
```

13. Analizar el siguiente programa, suponiendo que se ejecuta en un procesador con pipeline de 5 etapas (FI,DI,FO,EX,WR) y determinar CPI promedio, Speedup y riesgos en cada ítem:

```
MOV #1, R01
MOV #5, R03
MOV #3, R01
ADD R01, R03
HLT
```

- Observar el valor del registro R03 ejecutando las instrucciones como si el pipeline funcionara de forma ideal (sin burbujas de detención).
- Observar el valor del registro R03 ejecutando las instrucciones como si el pipeline funcionara de forma ideal (sin burbujas de detención), pero agregando una instrucción NOP a continuación de la tercer instrucción.
- Observar el valor del registro R03 ejecutando las instrucciones,



Taller de Sistemas Digitales (E1225) - Arquitectura de Computadores I (E0225)

teniendo en cuenta las burbujas de detención debido a riesgos, pero agregando una instrucción NOP a continuación de la tercera instrucción. ¿Qué tipo de riesgo se produce?

d. Incluir la utilización de “operand forwarding”

e. **Utilizar el simulador CPU-SO para verificar los resultados.**

14. Calcular para el siguiente programa CPI prom. y speedup utilizando “Jump Prediction”:

```
program Ex6
i = 0
for p = 1 to 40
i = i + 1
if i = 10 then
i = 0
r = i
end if
next
end
```

15. Analizar la técnica de optimización de código “Loop Unroll” para reducir los riesgos de control.

a. Calcular CPI prom., Speedup y ver el número de líneas que se ejecutan, utilizando esta técnica en el siguiente código:

```
program Ex4
for n = 1 to 8
t = t + 1
next
end
```

a. Contar los riesgos de control que se producen con y sin “Loop Unrolling”.

b. Aplicar las optimizaciones posibles y verificar Speedup y riesgos de cada tipo.

Referencias:

-Mostafa Abd-El-Barr- Fundamentals of Computer Organization and Architecture (Wiley 2005) (Capítulos 6 y 7)

-Hennessy-Patterson - Computer Architecture: A Quantitative Approach (4th Ed Elsevier 2007) (Apéndice C)

-William Stallings-Computer Organization and Architecture, Designing for Performance – 8th Edition