**Guía de Estudio: Advanced Computer Architectures 2025-2**

**Objetivo:** Preparar a los alumnos para los exámenes teórico-prácticos, cubriendo las 4 unidades del temario con definiciones, fórmulas, ejercicios resueltos y consejos de estudio.

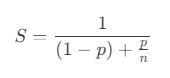
**1. Paralelismo (Unidad I)**

**Conceptos Clave:**

* **Paralelismo de datos (SIMD):** Misma operación aplicada a múltiples datos (ej: GPUs, instrucciones AVX).
* **Paralelismo de control (MIMD):** Diferentes operaciones ejecutadas en paralelo (ej: CPUs multinúcleo).
* **Clasificación de Flynn:**
  + **SISD:** Uniprocesadores tradicionales.
  + **SIMD:** Procesadores vectoriales (ej: Intel SSE).
  + **MISD:** Múltiples operaciones sobre un dato (raro, ej: sistemas tolerantes a fallos).
  + **MIMD:** Multiprocesadores (ej: clusters).

**Fórmulas:**

* **Speed-up (Ley de Amdahl):**



Donde *p* = fracción paralelizable, *n* = número de procesadores.

**Ejercicio tipo:**

**Problema:** Si el 70% de un programa es paralelizable y se usan 8 procesadores, calcula el *speed-up*.

**2. Pipeline y Procesadores Superscalares (Unidad II)**

**Conceptos Clave:**

* **Pipeline:** División de la ejecución en etapas (FETCH, DECODE, EXEC, MEM, WB).
* **Hazards:**
  + **RAW (Read After Write):** Dependencia de datos. Solución: *Forwarding*.
  + **WAR/WAW:** Dependencias por escritura. Solución: Renombrado de registros.
* **Superscalar:** Ejecuta múltiples instrucciones por ciclo (ej: Intel Core i7).

**Fórmulas:**

* **CPI (Ciclos por Instrucción) con branches:**

CPI=1+(Penalización×Tasa de fallos)

**Ejercicio tipo:**

**Problema:** Un pipeline tiene un 20% de branches con penalización de 4 ciclos y predicción con 80% de acierto. Calcula el CPI.

**3. Procesadores Vectoriales y Arrays (Unidad III)**

**Conceptos Clave:**

* **Registros vectoriales:** Almacenan bloques de datos para operaciones SIMD (ej: registros AVX-512).
* **Chaining:** Encadenamiento de operaciones vectoriales para paralelismo.
* **Memoria entrelazada:** Reduce conflictos dividiendo la memoria en bancos.

**Fórmulas:**

* **Tiempo de ejecución vectorial:**

**T=Startup+(Latencia×Elementos)**

**Ejercicio tipo:**

**Problema:** Un vector de 128 elementos con *startup* de 10 ciclos, latencia de 1 ciclo/elemento y *chaining* para 4 operaciones. Calcula el tiempo total.

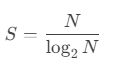
**4. Sistemas Multiprocesador (Unidad IV)**

**Conceptos Clave:**

* **Memoria compartida:** Todos los procesadores acceden a una memoria global (ej: NUMA).
* **Memoria distribuida:** Cada procesador tiene su memoria (ej: clusters con MPI).
* **Redes de interconexión:**
  + **Hipercubo:** Conecta 22*n* nodos con latencia *O*(log*N*).
  + **Malla:** Escalabilidad lineal (ej: sistemas embebidos).

**Fórmulas:**

* **Speed-up en suma paralela:**

 (Para un árbol de reducción)

**Ejercicio tipo:**

**Problema:** Suma de un arreglo de 256 elementos en 16 procesadores con memoria compartida.

**Ejercicios tipo para Práctica**

**Ejercicio tipo (Pipeline):**

Dado el siguiente código en un pipeline de 5 etapas:

asm

ADD R1, R2, R3

SUB R4, R1, R5

MUL R6, R4, R7

Identifica hazards y propone soluciones.

**Ejercicio tipo (Vectorial):**

Calcula el tiempo para procesar 3 operaciones vectoriales de 64 elementos con *startup* de 8 ciclos, latencia 2 ciclos/elemento y *chaining* para 2 operaciones.

**Ejercicio tipo (Multiprocesador):**

Diseña un algoritmo paralelo para multiplicar matrices 8x8 en 4 procesadores.

**Consejos de Estudio:**

1. **Teoría:** Enfócate en clasificaciones (Flynn, hazards), ventajas/desventajas de arquitecturas.
2. **Práctica:** Resuelve ejercicios de CPI, *speed-up*, y tiempo vectorial.
3. **Simulaciones:** Usa herramientas como **Gem5** (pipelines) o **MPI** (multiprocesadores).
4. **Casos de estudio:** Revisa ejemplos de CPUs modernas (ej: Intel Pentium, IBM Power9).

**ESTILO EXAMEN ETS**

* **60 a 70% conocimiento teorico, pero embebido en ejercicios**
* **30 a 40% programación en VHDL ejercicios en código escrito a mano.**

**Referencias Rápidas:**

* **Hwang, K.** (1993): Clasificaciones de paralelismo.
* **Stallings, W.** (2010): Pipeline y hazards.
* **Patterson & Hennessy** (2008): Procesadores vectoriales y multicore.