

Contents

0.0.1	Cache	1
0.1	Conversion unidades de tiempo	2
0.2	Conversión de unidades de frecuencia	2
0.2.1	P (A) 2023	2
0.2.2	Ejercicio 9	2
0.2.3	9d	3

0.0.1 Cache

- Tiempo de ciclo:

$$T_{clk} = \frac{1}{frec}$$

- Tag:

$$tag = \frac{MPSize}{CacheSize}$$

- Tamaño total datos:

$$capTotal = lineas * anchoBloque$$

- Offset byte:

$$cantBytes = \frac{tamPalabra}{w_m}$$

- AMAT:

$$AMAT = hitTime + missRate * missPenalty$$

- Ciclos de reloj:

$$ClockCycles = CPI + \frac{memAccesses}{instructionCount} * missRate * missPenalty$$

- CPI promedio:

$$CPI_{prom} = CPI + memStallData + memStallInst$$

- Ciclos de espera de memoria:

$$memStallClockCycles = \frac{memAccesses}{instructionCount} * missRate * missPenalty$$

- Latencia de etapa:

$$LatEtapa = tiempoCiclo(Tc) = \frac{tiempoTotal}{N} + penalidadRegistroPipeline$$

- CPI:

$$CPI = 1.23 + 0.1 * (N - 5)$$

- **Tiempo por instrucción:**

$$tiempoPorInstruccion(Ti) = Tc * CPI$$

- **Camino correcto:**

$$cantCaminoCorrecto + fetchPenalty * missedFetches$$

- **Precisión:**

$$precision = \frac{aciertos}{total}$$

0.1 Conversion unidades de tiempo

Unidad	Equivalente en segundos
1 nanosegundo	$1 * 10^{-9}$
1 picosegundo	$1 * 10^{-12}$

0.2 Conversión de unidades de frecuencia

Unidad	Equivalente en hertzios
1 hertzio	1
1 kilohertzio	$1 * 10^3$
1 megahertzio	$1 * 10^6$
1 gigahertzio	$1 * 10^9$
1 terahertzio	$1 * 10^{12}$

0.2.1 P (A) 2023

- **Líneas por vía:**

$$lineasporvia = \frac{lineasCache}{cantVias}$$

0.2.2 Ejercicio 9

1. **AMAT (Average Memory Access Time):**

$$AMAT = hitTime + missRate * missPenalty$$

- **hitTime:** Tiempo que toma acceder a un dato en caché cuando se encuentra (hit).

- **missRate:** Proporción de accesos a memoria que resultan en un fallo (miss).
- **missPenalty:** Tiempo adicional que se requiere para acceder a la memoria principal cuando ocurre un fallo.

2. ClockCycles:

$$ClockCycles = CPI + \frac{memAccesses}{instructionCount} * missRate * missPenalty$$

- **CPI (Cycles Per Instruction):** Número promedio de ciclos de reloj que toma ejecutar una instrucción.
- **memAccesses:** Número total de accesos a memoria realizados.
- **instructionCount:** Número total de instrucciones ejecutadas.

0.2.3 9d

3. CPI promedio con stalls:

$$CPI_{prom} = CPI + memStallData + memStallInst$$

- **memStallData:** Tiempo de espera (stall) debido a accesos a datos en memoria.
- **memStallInst:** Tiempo de espera (stall) debido a accesos a instrucciones en memoria.

4. Cálculo de ciclos de reloj por stalls de memoria:

$$memStallClockCycles = \frac{memAccesses}{instructionCount} * missRate * missPenalty$$

- Esta fórmula calcula el número de ciclos de reloj perdidos debido a los stalls de memoria, considerando el número de accesos a memoria y la tasa de fallos.