Ocho grandes ideas en arquitectura de computadoras:

Diseñar teniendo en cuenta la Ley de Moore

Usar la abstracción para simplificar el diseño

Hacer rápido el caso común

Desempeño por paralelismo

Desempeño por pipelining

Desempeño por predicción

Jerarquía de memorias

Confiabilidad por redundancia

Cinco realidades en la computación:

Los ints no son enteros, y los floats no son reales.

Se tiene que saber Assembly

La memoria importa

Importa mas el desempeño que la complejidad

Las computadoras hacen más que ejecutar programas

Debajo de un programa:

Aplication software: Escrito en un lenguaje de alto nivel.

System software: Compilador, traduce el código a código máquina. Sistema operativo: Maneja la entrada y salida, gestiona la memoria, programa tareas.

Hardware: Procesador, memoria, controladores, I/O.

Rendimiento:

Tiempo de respuesta: Tiempo que toma realizar una tarea (tiempo de ejecución).

Throught: Cantidad de tareas por unidad de tiempo.

Rendimiento = 1 / tiempo de respuesta.

Clocking de la CPU: La operación del hardware digital se rige por un reloj (clock) de frecuencia fija.

Periodo del reloj: Duración de un ciclo del reloj, se mide en segundos.

Frecuencia del reloj (rate): Se miden en ciclos por segundos (hertz).

**Tiempo de CPU** = (Ciclos de reloj) x (Periodo) = (Ciclos de reloj) / (periodo). Es el tiempo utilizado en procesar una tarea.

El rendimiento mejora al aumentar la frecuencia del reloj, o al reducir los ciclos de reloj, es una relación de compromiso (trade off) entre la frecuencia del reloj, y la cantidad de ciclos.

**Ciclos de reloj** = (recuento de instrucciones) x (ciclos por instrucción)

Entonces al tiempo de CPU lo podemos calcular como:

Tiempo de CPU = **cant instrucciones** x **ciclos por instrucc** x **periodo**

**Tiempo de ejecución** = (recuento de instrucciones) x CPI x (periodo) = (recuento de instrucciones) x CPI / (frecuencia del reloj)

CPI: Ciclos de reloj por instrucción (promedio), determinado por el hardware del CPU.

**Principio de localidad**: Los programas tienden a usar los datos e instrucciones que se usaron recientemente. Se dice que un programa pasa la mayor parte del tiempo solo en un 10% de su código. **Localidad espacial:** Los elementos cuyas direcciones son cercanas suelen ser referenciados. **Localidad temporal:** Elementos utilizados recientemente es probable que sean reutilizados.

**Ley de Amdahl**: El aumento del rendimiento debido a una mejora en un componente del sistema está limitado por la fracción de uso de ese componente. Cuantifica la regla del caso común.