

INFORMÁTICA

Trabajo Práctico N°1 – Introducción a la arquitectura de computadoras y sistemas operativos.

Parte 1: Introducción a la Arquitectura de Computadoras

1. **Explique el concepto de arquitectura de una computadora y la importancia de su organización en capas.**

Se refiere al diseño y la estructura de los componentes fundamentales de una computadora, incluyendo el hardware y cómo interactúan entre sí. Esto abarca la CPU, memoria, dispositivos de entrada/salida y la interconexión entre ellos. Es un enfoque que divide el sistema en diferentes niveles (hardware, firmware, sistema operativo, aplicaciones) para facilitar el diseño, desarrollo y mantenimiento. Cada capa tiene funciones específicas y se comunica con las adyacentes. Esto mejora la modularidad, permite abstraer la complejidad y optimiza el desempeño al permitir ajustes específicos sin afectar todo el sistema. Facilita la comprensión y diseño de sistemas complejos, permite la compatibilidad y mejora la eficiencia en la gestión de recursos.

2. **Verdadero/Falso (Justificar las falsas):**

a) **La memoria caché es más lenta que la memoria principal.**

Falso. La memoria caché es más rápida que la memoria principal. Está diseñada para almacenar temporalmente datos a los que la CPU accede con frecuencia, reduciendo los tiempos de acceso y mejorando el rendimiento.

b) **El microprograma define las instrucciones que puede ejecutar la CPU.**

Verdadero

3. **Realice un diagrama que muestre la interconexión entre la CPU, la memoria principal y los dispositivos de E/S.**

4. **Describa los tres pasos principales que realiza la CPU para ejecutar una instrucción.**

Los tres pasos principales que realiza la CPU para ejecutar una instrucción son:

Fetch (Captura): La CPU obtiene la instrucción desde la memoria principal. El Contador de Programa (PC) señala la dirección de la próxima instrucción, que es luego cargada en el Registro de Instrucciones (IR).

Decode (Decodificación): La instrucción capturada es interpretada por la Unidad de Control. En este paso, se identifica qué operación se debe realizar y qué recursos se necesitan (registros, ALU, etc.).

Execute (Ejecución): La operación decodificada es ejecutada. Puede implicar operaciones aritméticas/lógicas en la Unidad Aritmética-Lógica (ALU), transferencias de datos entre registros o comunicación con dispositivos de E/S.

5. **Compare las arquitecturas RISC y CISC, proporcionando ejemplos de dispositivos que usan cada una.**

Arquitectura RISC (Reduced Instruction Set Computer):

- **Características:** Utiliza un conjunto reducido de instrucciones simples y rápidas. Las instrucciones son uniformes y se ejecutan en un solo ciclo de reloj. Enfocada en la eficiencia y el rendimiento.
- **Ventajas:** Mayor velocidad y eficiencia en la ejecución de instrucciones, simplicidad en el diseño de hardware.
- **Ejemplos de dispositivos:** Procesadores ARM utilizados en smartphones, tabletas y dispositivos IoT.

Arquitectura CISC (Complex Instruction Set Computer):

- Características: Utiliza un conjunto más amplio y complejo de instrucciones, algunas de las cuales pueden realizar tareas complejas en un solo paso. Las instrucciones suelen ser más variadas y pueden requerir varios ciclos de reloj para ejecutarse.
 - Ventajas: Menor cantidad de código para tareas complejas, lo que reduce el tamaño de los programas.
 - Ejemplos de dispositivos: Procesadores Intel y AMD utilizados en computadoras personales y servidores.
- Comparación clave: RISC se centra en la simplicidad y rapidez de ejecución de instrucciones, ideal para dispositivos móviles y sistemas embebidos. CISC, por otro lado, busca realizar operaciones más complejas con menos instrucciones, optimizando la programación para sistemas de propósito general como PC y estaciones de trabajo.

6. **Convertir los siguientes números binarios a decimal, hexadecimal y octal:**

a) **10101010 11010010 10011010**

Decimal: 11,195,034

Hexadecimal: AAD29A

Octal: 52551232

b) **11001100 00111001 00101110**

Decimal: 13,383,982

Hexadecimal: CC392E

Octal: 63034456

7. **¿Qué son los registros en una CPU y cuál es la función del Program Counter?**

Registros en la CPU: Son pequeñas unidades de almacenamiento dentro del procesador que guardan temporalmente datos e instrucciones durante la ejecución de programas. Son mucho más rápidos que la memoria RAM y se utilizan para operaciones inmediatas, como cálculos, direcciones de memoria y control de ejecución.

Función del Program Counter (Contador de Programa): Es un registro que guarda la dirección de la siguiente instrucción que la CPU debe ejecutar. Después de capturar una instrucción, el Program Counter se incrementa para apuntar a la siguiente, permitiendo la ejecución secuencial de un programa. Es esencial para el control del flujo en un sistema de cómputo.

8. **Explique la diferencia entre multiprocesadores y multicomputadoras.**

Los multiprocesadores operan en un sistema unificado con memoria compartida, mientras que las multicomputadoras consisten en sistemas independientes que cooperan para realizar tareas a través de redes.

9. **Verdadero/Falso (Justificar las falsas):**

a) **En una arquitectura de computadoras, los datos siempre se almacenan en formato big endian.**

Falso. En una arquitectura de computadoras, los datos no siempre se almacenan en formato big endian. El formato de almacenamiento depende de la arquitectura. En el formato big endian, el byte más significativo se almacena primero (en la dirección de memoria más baja). Sin embargo, en little endian, el byte menos significativo se almacena primero. Algunas arquitecturas usan big endian, otras little endian, o incluso pueden ser configurables.

b) **Los errores de redondeo en punto flotante son iguales para todos los números.**

Los errores de redondeo en punto flotante no son iguales para todos los números. Los errores dependen de la magnitud del número y de la precisión con la que se representa. En general, los números más grandes o pequeños pueden sufrir más errores de redondeo debido a la limitación en la cantidad de dígitos significativos que pueden almacenarse en una representación de punto flotante. Además, algunas operaciones matemáticas pueden introducir más errores que otras.

10. **¿Se puede transmitir datos entre una máquina que usa big endian y otra que usa Little endian? ¿Cómo?**

Sí, es posible transmitir datos entre una máquina que usa big endian y otra que usa little endian. La conversión de datos se realiza intercambiando el orden de los bytes antes de la transmisión y luego devolviendo el orden original al recibir los datos. Esto se puede hacer manualmente o utilizando funciones de biblioteca específicas para la conversión de endianness.

11. Describa cómo se representa un número con punto flotante en una computadora.

Un número con punto flotante en una computadora se representa utilizando el estándar IEEE 754, que divide el número en tres partes:

Signo (1 bit): Indica si el número es positivo (0) o negativo (1).

Exponente (8 o 11 bits, según precisión): Representa la potencia a la que se eleva la base (generalmente 2). Está sesgado para permitir exponentes negativos.

Mantisa o significando (23 o 52 bits, según precisión): Contiene la parte fraccionaria del número, normalizada para que su primer dígito sea 1 (implícito).

La fórmula general es:

$$(-1)^{\text{signo}} \times 1.\text{mantisa} \times 2^{(\text{exponente}-\text{sesgo})}$$

12. Investigue un microcontrolador usado en la industria mecatrónica y describa sus características principales. (si utiliza alguna IA para la investigación, añada el enlace al chat).

Un microcontrolador ampliamente utilizado en la industria mecatrónica es el PIC32 de Microchip. Este microcontrolador es un dispositivo de 32 bits que ofrece un rendimiento robusto y una amplia compatibilidad con aplicaciones avanzadas, como sistemas de control industrial, automatización y robótica. Algunas de sus características principales son:

Procesamiento: Su arquitectura de 32 bits permite manejar operaciones complejas y ejecutar múltiples tareas en tiempo real, ideal para aplicaciones que requieren procesamiento intensivo.

Memoria: Cuenta con una memoria flash que varía entre 64 KB y 512 KB, lo que permite almacenar programas más grandes y manejar grandes volúmenes de datos.

Interfaces de comunicación: Ofrece soporte para múltiples protocolos, como I2C, SPI, UART y USB, permitiendo una fácil integración con otros dispositivos y sensores en sistemas mecatrónicos.

Seguridad: Incluye características de seguridad avanzadas, como arranque seguro y autenticación basada en hardware, esenciales para aplicaciones que se conectan a internet y requieren protección frente a ciberataques.

El PIC32 se usa en aplicaciones como controladores de automatización, sistemas embebidos para vehículos y dispositivos médicos debido a su equilibrio entre rendimiento, costo y versatilidad.

13. ¿Cómo se diferencian las arquitecturas Von Neumann y Harvard? Proporcione ejemplos de su uso en la industria.

Las arquitecturas Von Neumann y Harvard se diferencian principalmente en la forma en que gestionan la memoria y los datos:

1. Arquitectura Von Neumann:

- Memoria Unificada: Utiliza una única memoria para almacenar tanto instrucciones como datos.
- Bus Compartido: Emplea un bus único para acceder a la memoria, lo que significa que no puede acceder simultáneamente a instrucciones y datos, generando un cuello de botella conocido como el "Von Neumann bottleneck".
- Ejemplo de Uso: La mayoría de los computadores personales y sistemas embebidos generales utilizan esta arquitectura debido a su simplicidad y menor costo de implementación.

2. Arquitectura Harvard:

- Memorias Separadas: Tiene memorias independientes para instrucciones y datos.

- Buses Independientes: Al tener buses separados, permite acceder a instrucciones y datos simultáneamente, resultando en una mayor eficiencia.
- Ejemplo de Uso: Es común en microcontroladores y sistemas embebidos como los de la familia PIC y AVR, que requieren alta velocidad y eficiencia en el procesamiento, como en aplicaciones de control industrial y dispositivos de tiempo real.

14. ¿Cuál es la importancia de la memoria caché en el rendimiento de la CPU?

La memoria caché es crucial para el rendimiento de la CPU porque actúa como un almacenamiento intermedio de alta velocidad entre el procesador y la memoria principal (RAM). Su importancia se debe a varios factores:

1. Reducción de la Latencia: La caché almacena los datos y las instrucciones más frecuentemente utilizados por la CPU. Al tenerlos accesibles más rápidamente que desde la RAM, se reduce significativamente el tiempo de espera del procesador, lo que mejora el rendimiento.
2. Minimización del Cuello de Botella: La CPU opera a velocidades mucho mayores que la RAM. Sin la caché, la CPU tendría que esperar continuamente a que la RAM suministre datos, lo que ralentizaría las operaciones. La caché mitiga este problema al proporcionar datos casi instantáneamente.
3. Jerarquía de Caché: Las CPU modernas suelen tener varios niveles de caché (L1, L2, L3). La caché L1 es la más pequeña y rápida, mientras que L3 es más grande pero más lenta. Esta jerarquía optimiza el acceso a los datos, priorizando la velocidad cuando es más necesario.
4. Localidad de Referencia: Los programas suelen acceder repetidamente a los mismos datos (localidad temporal) o a datos cercanos (localidad espacial). La caché aprovecha estas características almacenando estos datos cercanos, mejorando la eficiencia.

15. Describa cómo se implementa la técnica de pipelining en las CPU modernas.

La técnica de pipelining en las CPU modernas permite mejorar el rendimiento al superponer la ejecución de múltiples instrucciones. El pipeline divide la ejecución de una instrucción en varias etapas, donde cada una realiza una parte específica del proceso.

<https://kjdelectronics.com/blogs/news/top-microcontrollers-in-2023>

<https://www.embedic.com/technology/details/choosing-the-best-microcontroller-for-robotics-2023>

<https://mechatronai.com/understanding-microcontrollers-in-mechatronic-systems/>

<https://www.engineersgarage.com/microcontroller-trends-2023/>