1. ¿Qué es la base de un sistema de numeración?

Se le llama base de un sistema de numeración a la cantidad de símbolos distintos que hay en dicho sistema.

1. ¿Cuáles son los 4 sistemas de numeración / codificación más utilizados en informática / programación?

Los cuatro sistemas más usados en informática son el hexadecimal, BCD, binario y decimal.

1. ¿Con cuál de los sistemas de numeración / codificación que respondió en la pregunta anterior cree que necesitará más dígitos para representar un valor dado? ¿Con cuál menos? ¿Por qué?

Con el sistema binario, dado que posee una base mas chica y se requieren más dígitos para representar un valor. Y el sistema con menos es el hexadecimal porque posee una base mas grande lo que le permite representar valores mas grandes con menor cantidad de dígitos.

1. ¿Cuáles son las ventajas y desventajas de los sistemas de numeración / codificación que mencionó en las respuestas anteriores? En otras palabras, ¿por qué se usan? Responda por al menos 3 de los 4.

Sistema Binario:

Ventajas:

Simplicidad en la implementación de circuitos digitales.

Facilita el diseño de lógica digital.

Desventajas:

Requiere más dígitos para representar valores grandes.

Sistema Hexadecimal:

Ventajas:

Representa grandes valores con menos dígitos.

Fácil conversión entre binario y hexadecimal.

Desventajas:

Menos intuitivo para las personas acostumbradas al sistema decimal.

Sistema Decimal:

Ventajas:

Familiar y fácil de entender para la mayoría de las personas.

Desventajas:

Menos eficiente para las computadoras que prefieren trabajar en binario.

Sistema BCD (Binary-Coded Decimal):

Ventajas:

Precisión en aplicaciones financieras y calculadoras.

Facilita la conversión entre binario y decimal.

Desventajas:

Menos eficiente en términos de almacenamiento comparado con binario puro.

6. ¿Cuántas magnitudes distintas se pueden representar con…?

a. 8 bits sin tener en cuenta el signo

b. 32 bits sin tener en cuenta el signo

c. 8 bits teniendo en cuenta el signo (se acepta respuesta con ± 1 de

tolerancia)

d. 32 bits teniendo en cuenta el signo (se acepta respuesta con ± 1 de

tolerancia)

e. N bits con signo y sin almacenar signo

f. 2 dígitos hexadecimales (sin tener en cuenta el signo)

g. 8 dígitos hexadecimales (sin tener en cuenta el signo)

h. N dígitos hexadecimales sin signo

i. Con 8 bits usando codificación BCD (sin signo)

a.256

b. 2 ^ 32 = 4294967296

c. Con 8 bits teniendo en cuenta el signo, se pueden representar 2^7 = 128 valores distintos para números enteros con signo. Sin embargo, debido a que uno de los valores se reserva para representar el cero, quedan 128−1=127 valores únicos para representar números enteros positivos y negativos.

d. Con 32 bits teniendo en cuenta el signo, se pueden representar 2^31= 2147483648 valores distintos para números enteros con signo. Sin embargo, al igual que en el caso de 8 bits, uno de los valores se reserva para representar el cero, por lo que quedan 2147483648−1=2147483647 valores únicos para representar números enteros positivos y negativos.

e. 2^(n-1) tomando valores únicos para representar números enteros positivos y negativos.

f. FF=15\*16^1+15\*16^0=255, son 256 valores con dos dígitos 0-255.

g. FFFFFFFF = 15\*16^7+15\*16^6+15\*16^5+15\*16^4+15\*16^3+15\*16^2+15\*16^1+15\*16^0= 4294967295

h. 15\*16^(n-1)+…+15\*16^0

i. 1001 1001 = 99

elegir

7. ¿Cuál es el valor más grande que se puede representar con N bits si...?

a. No se tiene en cuenta el signo

b. Se tiene en cuenta el signo

c. Si N es múltiplo de 4, no se usa signo pero se usa codificación BCD

a. 2^(n-1)+…+2^(0) = 2^n

b. 2^(n-1)

c. (10^(n/4))-1 ej 8 bits 10^2 -1=99

8. a. ¿Qué significa la palabra unsigned en lenguaje C?

b. ¿Cómo se usa?

c. ¿A qué tipo de datos se puede aplicar? Provea 4 ejemplos.

d. ¿Modifica el tamaño que ocupa el tipo de datos?

a. quiere decir que la variable que estoy almacenando no posee signo.

b. se coloca detrás de la variable entera que se va a declarar.

c. solo se le puede aplicar a variables del tipo entero, como int, char, short, y long.

d. No, la palabra clave unsigned no modifica el tamaño que ocupa el tipo de datos. Lo que cambia es el rango de valores que la variable puede representar.

Ejercicios de operaciones lógicas a nivel bit

11. Explique el significado de los siguientes operadores en lenguaje C (en el contexto de operaciones a nivel bit): &, |, ~

11. En el lenguaje C, los operadores &, | y ~ se utilizan para realizar operaciones a nivel de bits en valores enteros. Aquí te explico el significado de cada uno de estos operadores:

El operador & realiza una operación AND bit a bit entre dos operandos. El resultado tiene un bit en 1 solo si ambos bits correspondientes en los operandos son 1.

El operador | realiza una operación OR bit a bit entre dos operandos. El resultado tiene un bit en 1 si al menos uno de los bits correspondientes en los operandos es 1.

El operador ~ realiza una operación NOT bit a bit en un solo operando. Invierte todos los bits del operando, cambiando los bits de 1 a 0 y de 0 a 1.

Ejercicios de arquitecturas de hardware

14. ¿Cuál es la diferencia entre hardware y software?

14. La diferencia entre hardware y software radica en que el hardware es la parte física y tangible de un sistema informático, como procesadores, memoria y dispositivos de entrada/salida, mientras que el software es la parte intangible y abstracta, compuesta por programas e instrucciones que dirigen al hardware para realizar tareas específicas. El hardware necesita el software para ser funcional, y el software requiere hardware para ejecutarse.

15. ¿Qué es un dispositivo periférico?

15. Un dispositivo periférico es un hardware externo que se conecta a una computadora para ampliar sus capacidades y funcionalidades, permitiendo la entrada, salida, almacenamiento o comunicación de datos. Ejemplos comunes de periféricos incluyen teclados, ratones, impresoras, monitores, discos duros externos y dispositivos USB.

16. ¿Cuál es la característica fundamental de la arquitectura de Von Neumann?

16. La característica fundamental de la arquitectura de Von Neumann es que utiliza una única memoria para almacenar tanto las instrucciones del programa como los datos que el programa manipula. Esto significa que la CPU accede a la memoria de manera secuencial para recuperar instrucciones y datos, utilizando el mismo bus para ambas tareas, lo que simplifica el diseño del hardware, pero puede limitar el rendimiento debido al "cuello de botella" de la memoria.

17. ¿Qué es un bus y por qué se utiliza?

17. Un bus en el contexto de la informática es un conjunto de líneas de comunicación que se utilizan para transferir datos y señales de control entre los diferentes componentes de un sistema informático, como la CPU, la memoria, los dispositivos de entrada/salida y otros dispositivos. Se utiliza para permitir la comunicación y la transferencia de información de manera eficiente entre estos componentes, facilitando la interacción y el funcionamiento del sistema en su conjunto.

18. ¿Cuáles son los buses que utiliza la arquitectura de Von Neumann?

18. La arquitectura de Von Neumann utiliza tres tipos principales de buses:

Bus de datos: Se utiliza para transferir datos entre la CPU, la memoria y los dispositivos de entrada/salida. Permite la comunicación de los datos que la CPU necesita para ejecutar instrucciones y procesar información.

Bus de direcciones: Este bus se utiliza para enviar direcciones de memoria desde la CPU a la memoria principal. Permite a la CPU acceder a ubicaciones específicas en la memoria para leer o escribir datos.

Bus de control: El bus de control se utiliza para enviar señales de control y comandos entre la CPU y otros componentes del sistema, como la memoria y los dispositivos de entrada/salida. Ayuda a coordinar las operaciones del sistema y asegura que se realicen en el momento adecuado y en el orden correcto.

19.a ¿Qué es el período T de una señal y en qué unidades se mide?

19.a El período T de una señal periódica es el tiempo que tarda la señal en completar un ciclo completo de su forma de onda. Se mide en segundos (s) y representa la duración de un ciclo de la señal, es decir, el tiempo entre dos puntos equivalentes en la forma de onda, como dos picos consecutivos o dos cruces por cero. El período es el inverso de la frecuencia de la señal, por lo que se puede calcular como T = 1/f, donde f es la frecuencia de la señal en hertz (Hz).

19.b ¿Qué es la frecuencia f de una señal y en qué unidades se mide?

19.b La frecuencia f de una señal periódica es el número de ciclos completos que la señal completa en un segundo. Se mide en hercios (Hz) y representa la cantidad de veces que la forma de onda se repite en un segundo. Por ejemplo, una señal con una frecuencia de 100 Hz completa 100 ciclos en un segundo. La frecuencia está relacionada con el período T de la señal a través de la fórmula f = 1/T, donde T es el período en segundos.

19.c ¿Qué relación matemática existe entre f y T y qué significa?

19.c La relación matemática entre la frecuencia f y el período T de una señal periódica es la siguiente:

𝑓=1/𝑇

Esto significa que la frecuencia de una señal es el inverso de su período, y viceversa. En otras palabras, la frecuencia indica cuántos ciclos de la señal ocurren en un segundo, mientras que el período indica cuánto tiempo dura un ciclo completo de la señal. Por lo tanto, si conoces la frecuencia de una señal, puedes calcular su período utilizando la fórmula anterior, y si conoces el período, puedes calcular la frecuencia utilizando la fórmula inversa.

19.d ¿Cuál es la frecuencia de una señal que tiene 2ns de período?

19.d f=1/(2×10−9 s)= (½)​×10^9 Hz=500MHz

20. ¿Qué es el clock de un sistema digital y para qué sirve?

20. El "clock" en un sistema digital es una señal periódica que oscila entre dos niveles de voltaje, representando un ciclo de trabajo regular. Este ciclo de trabajo se repite a intervalos regulares y se utiliza para sincronizar las operaciones de los componentes del sistema, asegurando que las operaciones se realicen en el momento correcto y a una velocidad adecuada.

El clock es fundamental en los sistemas digitales porque controla la velocidad a la que se ejecutan las operaciones, la transferencia de datos y la sincronización de todos los componentes. Sin un clock adecuado, los componentes del sistema podrían funcionar de forma desincronizada, lo que podría provocar errores en las operaciones y el procesamiento de datos.

21. ¿Para qué sirve el procesador o CPU?

21. El procesador, también conocido como Unidad Central de Procesamiento (CPU), es el componente principal de un sistema informático. Sirve para realizar las operaciones y cálculos necesarios para ejecutar programas y procesar datos. La CPU interpreta las instrucciones de los programas, realiza operaciones aritméticas y lógicas, gestiona la memoria y coordina el funcionamiento de todos los componentes del sistema. En resumen, el procesador es el cerebro de la computadora y es fundamental para su funcionamiento.

22. ¿Cuáles son los tres componentes principales de un procesador y para qué sirve cada uno?

22. Los tres componentes principales de un procesador son la Unidad de Control (CU), la Unidad Aritmético-Lógica (ALU) y los Registros.

Unidad de Control (CU): La CU se encarga de coordinar y controlar las operaciones de la CPU. Interpreta las instrucciones del programa y envía las señales adecuadas a otros componentes para ejecutarlas. También controla el flujo de datos entre la CPU y la memoria, así como entre otros dispositivos.

Unidad Aritmético-Lógica (ALU): La ALU realiza las operaciones aritméticas (suma, resta, multiplicación, división) y lógicas (AND, OR, NOT) necesarias para procesar los datos. Es responsable de realizar cálculos y tomar decisiones basadas en las instrucciones del programa.

Registros: Los registros son pequeñas unidades de almacenamiento de alta velocidad que se encuentran dentro de la CPU. Se utilizan para almacenar datos temporales, direcciones de memoria y resultados de operaciones aritméticas. Los registros son muy rápidos y permiten a la CPU acceder rápidamente a los datos necesarios para ejecutar las instrucciones.

23. ¿Qué es la RAM y qué significa que es de acceso aleatorio?

23. La RAM (Random Access Memory, Memoria de Acceso Aleatorio) es un tipo de memoria que se utiliza en computadoras y otros dispositivos para almacenar datos y programas de forma temporal mientras se están utilizando. La RAM es "de acceso aleatorio" porque se puede acceder a cualquier ubicación de memoria de forma directa, sin necesidad de recorrerla secuencialmente. Esto significa que la CPU puede leer o escribir en cualquier dirección de memoria de la RAM de manera casi instantánea, lo que la hace muy rápida y eficiente para realizar operaciones temporales.

24. ¿Qué es el tamaño de palabra de un procesador?

24. El tamaño de palabra de un procesador se refiere a la cantidad de bits que puede procesar en una única operación. Por ejemplo, un procesador de 32 bits puede procesar instrucciones y datos de 32 bits de longitud en una sola operación, mientras que un procesador de 64 bits puede procesar 64 bits en una sola operación. El tamaño de palabra de un procesador influye en varios aspectos de su rendimiento, como la cantidad de memoria que puede direccionar y la precisión de los cálculos que puede realizar.

25. En el contexto de hardware: ¿Qué es un núcleo?

25. En el contexto de hardware, un núcleo se refiere a una unidad de procesamiento independiente dentro de un procesador que puede ejecutar instrucciones y realizar cálculos de forma autónoma. Los procesadores modernos suelen tener múltiples núcleos, lo que les permite realizar varias tareas simultáneamente, lo que se conoce como procesamiento multinúcleo. Cada núcleo tiene su propia unidad de control, su propia unidad aritmético-lógica y sus propios registros, lo que le permite funcionar como un procesador completo.