



**Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño**



Ingeniería en Software y Tecnologías Emergentes

Materia: Organización de Computadoras

Alumno: Jesus Eduardo Rodriguez Ramirez

Profesor: Jonatan Crespo Ragland

Grupo 932

Taller 3

Ensenada, B.C; a 10 de septiembre del 2024

A. Investiga para responder lo siguiente en tu cuaderno:

1. ¿Por qué se utilizan números hexadecimales en lugar de binarios en muchas aplicaciones informáticas? Los números hexadecimales son más cortos que los binarios. Son más fáciles de leer y escribir. Se utilizan para representar valores de memoria, direcciones y colores.
2. ¿Cuál es la relación entre los números binarios y hexadecimales en términos de longitud? 1 dígito hexadecimal equivale a 4 dígitos binarios. Esto significa que un número hexadecimal es cuatro veces más compacto.
3. ¿Qué es el sistema octal y cómo se relaciona con el binario y el hexadecimal? El sistema octal es un sistema de base 8, utilizando los dígitos del 0 al 7.
 - > Relación con el binario:
 - Cada dígito se puede representar con tres dígitos binarios. Esto facilita la conversión entre binario y octal, ya que solo necesitas agrupar los dígitos binarios en grupos de tres.
 - > Relación con el hexadecimal:
 - Cada dígito hexadecimal se puede representar con cuatro dígitos. Aunque no hay conversión directa entre octal y hexadecimal, ambos sistemas se utilizan para simplificar la representación de números binarios largos.

8/sep/24.

4. ¿Qué ventajas tiene el uso del sistema hexadecimal frente al sistema decimal en programación? Los números hexadecimales son más cortos que los decimales. Es fácil convertir entre binario y hexadecimal. Son más fáciles de leer y entender para los programadores. Permiten representar grandes cantidades de información en menos espacio.

5. ¿Qué es el código Gray y cómo se relaciona con los números binarios? El código Gray es un tipo especial de código binario en el que dos números consecutivos difieren en un solo dígito. Esto lo hace útil para minimizar errores en la transmisión de datos.

6. ¿Cómo se usa el sistema binario para representar texto (ej. ASCII) en una computadora? Cada carácter (letra, número, símbolo) se asigna a un número entero único en la tabla ASCII. Estos valores numéricos se convierten a binario. La computadora almacena y procesa estos valores binarios. Cuando necesitas mostrar el carácter, la computadora convierte el valor binario de vuelta al carácter correspondiente utilizando la tabla ASCII.

8/sep/24.

7. ¿Cómo se interpreta un número hexadecimal en el contexto de direcciones de memoria en ensamblador?

Los números hexadecimales representan ubicaciones específicas en la memoria. Las instrucciones en ensamblador utilizan estas direcciones para acceder a la memoria. En algunas arquitecturas, las direcciones se dividen en segmentos y desplazamientos. Existen varios modos que determinan cómo se calcula la dirección efectiva de memoria.

8. ¿Por qué es más eficiente representar ciertas direcciones de memoria en hexadecimal en lugar de en decimal?

Los números hexadecimales son más cortos que los decimales. Es fácil convertir entre binario y hexadecimal. Son más fáciles de leer y entender. Permiten representar grandes cantidades de información en menos espacio.

9. ¿Por qué se prefieren los números binarios para la lógica de circuitos digitales? Los circuitos digitales utilizan dos estados (0, 1), lo que simplifica su diseño. Los sistemas binarios son menos susceptibles a errores y ruido. El álgebra booleana se implementa fácilmente con números binarios. Los circuitos binarios son rápidos y consumen menos energía.

8/sep/24.

10. ¿Cómo se utilizan los números hexadecimales en la programación de microcontroladores? Los números hexadecimales representan direcciones de memoria de manera compacta y legible. Se utilizan para configurar registros de control y manipular bits específicos. Facilitan el trabajo con máscaras de bits. El código ejecutable de los microcontroladores a menudo se representa en hexadecimal.
11. ¿Que rol juega el sistema binario en la transmisión de datos a través de redes informáticas? Los datos digitales se representan mediante bits (0 y 1). Los bits se codifican y modulan para ser transmitidos a través de medios físicos. Los dispositivos de red están diseñados para procesar datos binarios eficientemente. Facilita la detección y corrección de errores durante la transmisión.
12. ¿Cómo afecta la representación en binario al rango de valores que puede almacenar un tipo de dato en un lenguaje de programación? El rango de valores depende del número de bits asignados al tipo de dato. Almacenan solo valores no negativos. Almacenan tanto valores positivos como negativos, utilizando el bit más significativo como bit de signo. En tipos de datos de punto flotante, la representación binaria afecta la precisión y el rango de valores.

13. ¿Cómo se implementa la aritmética binaria en un procesador y qué papel juega el ALU (Unidad Aritmético Lógica)?

- > Implementación de la aritmética binaria: La ALU utiliza puertas lógicas para realizar operaciones aritméticas y lógicas. Los datos se almacenan temporalmente en registros dentro de la ALU. La ALU realiza sumas y restas directamente; operaciones más complejas se descomponen en sumas y restas.
- > Papel de la ALU: La ALU ejecuta las instrucciones aritméticas y lógicas del procesador. Interactúa con la unidad de control y los registros. Realiza operaciones lógicas esenciales para la toma de decisiones.

14. ¿Qué es el "endianness" y cómo afecta la representación de números en binario y hexadecimal? El "endianness" se refiere al orden en que se almacenan los bytes de un número en la memoria de una computadora. Dependiendo del endianness, la interpretación de los datos en la memoria puede variar. Es necesario convertir el orden de los bytes al transferir datos entre sistemas con diferentes endianness. Algunos procesadores están optimizados para un tipo específico de endianness.

8/sep/24.

15. ¿Cómo se relaciona el sistema binario con los algoritmos de compresión de datos? Los algoritmos de compresión reemplazan patrones en los datos binarios con representaciones más cortas. Eliminan la redundancia en los datos binarios. Permite reconstruir exactamente de los datos originales. Reduce el tamaño del archivo a costa de una posible pérdida de calidad.

16. ¿Cuál es el valor máximo que se puede representar con un número binario de 8 bits? ¿Y en hexadecimal? El valor máximo que se puede representar con un número binario de 8 bits es 255 en decimal. Esto se debe a que con 8 bits pueden tener ($2^8 = 256$) combinaciones posibles, que van desde 0 hasta 255. En hexadecimal, el valor máximo de 8 bits es 0xFF. Esto se debe a que cada dígito hexadecimal representa 4 bits, por lo que dos dígitos hexadecimales (4 bits cada uno) pueden representar hasta ($2^8 - 1 = 255$) en decimal.

17. ¿Qué es un "nibble" en el contexto de números binarios y hexadecimales? Un nibble es un grupo de cuatro bits, equivalente a la mitad de un byte. En hexadecimal, un nibble puede representar un solo dígito, de 0 a F (0-15 en decimal).

18. ¿Cómo se codifican los colores en formato RGB utilizando el sistema hexadecimal? En el sistema RGB, los colores se representan mediante la combinación de tres componentes: rojo (R), verde (G), y azul (B). Cada componente puede tener un valor entre 0 y 255. En el sistema hexadecimal, estos valores se expresan con dos dígitos hexadecimales cada uno, formando un código de seis caracteres.
19. ¿Cómo se utilizan los números hexadecimales en la representación de direcciones de hardware, como las MAC addresses (dirección MAC)? Las direcciones MAC son identificadores únicos para dispositivos de red, representados en formato hexadecimal. Constan de 48 bits divididos en seis grupos de dos dígitos hexadecimales.
20. ¿Cómo se utiliza el sistema hexadecimal en la representación de IPs en versiones de IPv6? Las direcciones IPv6 se representan utilizando el sistema hexadecimal y constan de 128 bits, divididos en ocho bloques de 16 bits cada uno. Cada bloque de 16 bits se representa como cuatro dígitos hexadecimales, separados por dos puntos.

8/sep/24.

21. ¿Cómo se utilizan los números binarios en la codificación de señales digitales? Se toman muestras de una señal analógica e intervalos regulares. Se asignan valores binarios a cada muestra. Los valores binarios se almacenan o transmiten en un formato específico. Los datos binarios se envían a través de redes digitales o se almacenan en medios digitales. Los datos binarios se convierten nuevamente en señales analógicas para su reproducción.
22. ¿Qué rol juega el sistema binario en la codificación del audio original (como MP3)? Se toman muestras de la señal analógica a intervalos regulares. Cada muestra se convierte en un valor binario. Los valores binarios se comprimen y codifican en un formato como MP3. Los datos binarios se almacenan o transmiten digitalmente. Los datos se convierten nuevamente en señales analógicas para su reproducción.
23. ¿Qué es la "aritmética de saturación" en operación binarias y cómo se implementa en hardware? La aritmética de saturación es una técnica en operaciones binarias donde los resultados se limitan a un rango fijo. Si el resultado excede el valor máximo o mínimo permitido, se ajusta a esos límites. Esto se implementa en hardware mediante comparaciones adicionales después de cada operación aritmética para ajustar el resultado.

8/sep/21.

24. ¿Cómo se usan los números binarios para representar matrices en gráficos por computadora y cómo se convierten a hexadecimal? En gráficos por computadora, los números binarios se utilizan para representar matrices de píxeles. Cada píxel puede tener valores RGB codificados en binario. Para convertir estos valores a hexadecimal, se agrupan los bits en grupos de cuatro y se convierten a su equivalente hexadecimal.

25. ¿Cómo se utilizan los números hexadecimales en la representación de códigos de error en sistemas operativos? Los números hexadecimales se utilizan en la representación de códigos de error en sistemas operativos debido a su capacidad para ser compactos y legibles.

10/sep/24.

B. Desarrolla los siguientes ejercicios y sus procedimientos.

1. Convierte 1000 en decimal, a binario y hexadecimal

Binario:

$$1000/2 = 500; \text{ residuo } 0.$$

$$500/2 = 250; \text{ residuo } 0.$$

$$250/2 = 125; \text{ residuo } 0.$$

$$125/2 = 62; \text{ residuo } 1.$$

$$62/2 = 31; \text{ residuo } 0.$$

$$31/2 = 15; \text{ residuo } 1.$$

$$15/2 = 7; \text{ residuo } 1.$$

$$7/2 = 3; \text{ residuo } 1.$$

$$3/2 = 1; \text{ residuo } 1.$$

$$1/2 = 0; \text{ residuo } 1.$$

$$= 1111101000$$

Hexadecimal:

$$1000/16 = 62; \text{ residuo } 8$$

$$62/16 = 3; \text{ residuo } 14 = E$$

$$3/16 = 0; \text{ residuo } 3$$

$$= 3E8$$

10/sep/24.

2. Convierte 3451 decimal, a binario y hexadecimal.

Binario:

$$3451 / 2 = 1725; \text{ residuo } 1$$

$$1725 / 2 = 862; \text{ residuo } 1$$

$$862 / 2 = 431; \text{ residuo } 0$$

$$431 / 2 = 215; \text{ residuo } 1$$

$$215 / 2 = 107; \text{ residuo } 1$$

$$107 / 2 = 53; \text{ residuo } 1$$

$$53 / 2 = 26; \text{ residuo } 1$$

$$26 / 2 = 13; \text{ residuo } 0$$

$$13 / 2 = 6; \text{ residuo } 1$$

$$6 / 2 = 3; \text{ residuo } 0$$

$$3 / 2 = 1; \text{ residuo } 1$$

$$1 / 2 = 0; \text{ residuo } 1$$

$$= 1101101111011$$

Hexadecimal:

$$3451 / 16 = 215; \text{ residuo } 11 = B$$

$$215 / 16 = 13; \text{ residuo } 7 = 7$$

$$13 / 16 = 0; \text{ residuo } 13 = D$$

$$= DB7$$

10/sep/24.

3. Convierte 11110101011 en binario, a decimal y hexadecimal.

Binario {
$$1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0$$
$$= 1024 + 512 + 256 + 128 + 32 + 8 + 2 + 1$$
$$= 1963$$

Hexadecimal:

$$1963/16 = 122; \text{ residuo } 11 = B$$

$$122/16 = 7; \text{ residuo } 10 = A$$

$$7/16 = 0; \text{ residuo } 7$$

$$= 7AB$$

4. Convierte 101011000011 en binario, a decimal y hexadecimal.

Binario:

$$1 \times 2^{11} + 1 \times 2^9 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^1 + 1 \times 2^0$$
$$= 2048 + 512 + 128 + 64 + 2 + 1$$
$$= 2755$$

Hexadecimal:

$$2755/16 = 172; \text{ residuo } 3$$

$$172/16 = 10; \text{ residuo } 12 = C$$

$$10/16 = 0; \text{ residuo } 10 = A$$

$$= AC3$$

10/sep/24.

5. Convierte AFF en hexadecimal, a decimal y binario.

Decimal:

$$\begin{aligned} & 10 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 \\ & = 2560 + 240 + 15 \\ & = 2815 \end{aligned}$$

Binario

$$2815 / 2 = 1407; \text{ residuo } 1$$

$$1407 / 2 = 703; \text{ residuo } 1$$

$$703 / 2 = 351; \text{ residuo } 1$$

$$351 / 2 = 175; \text{ residuo } 1$$

$$175 / 2 = 87; \text{ residuo } 1$$

$$87 / 2 = 43; \text{ residuo } 1$$

$$43 / 2 = 21; \text{ residuo } 1$$

$$21 / 2 = 10; \text{ residuo } 1$$

$$10 / 2 = 5; \text{ residuo } 0$$

$$5 / 2 = 2; \text{ residuo } 1$$

$$2 / 2 = 1; \text{ residuo } 0$$

$$1 / 2 = 0; \text{ residuo } 1$$

$$= 101011111111$$

10/sep/24.

6. Convierte D1B en hexadecimal, a decimal y binario.

Decimal:

$$\begin{aligned} & 13 \times 16^2 + 1 \times 16^1 + 11 \times 16^0 \\ & = 3328 + 16 + 11 \\ & = 3355 \end{aligned}$$

Binario:

$$\begin{aligned} 3355 / 2 &= 1677; \text{ residuo } 1 \\ 1677 / 2 &= 838; \text{ residuo } 1 \\ 838 / 2 &= 419; \text{ residuo } 0 \\ 419 / 2 &= 209; \text{ residuo } 1 \\ 209 / 2 &= 104; \text{ residuo } 1 \\ 104 / 2 &= 52; \text{ residuo } 0 \\ 52 / 2 &= 26; \text{ residuo } 0 \\ 26 / 2 &= 13; \text{ residuo } 0 \\ 13 / 2 &= 6; \text{ residuo } 1 \\ 6 / 2 &= 3; \text{ residuo } 0 \\ 3 / 2 &= 1; \text{ residuo } 1 \\ 1 / 2 &= 0; \text{ residuo } 1 \end{aligned}$$

$$= 110100011011$$