

## Octal

El sistema numérico octal, el cual se representa por su base 8, es un sistema numérico que emplea los números de 0 a 7. Este sistema se estructura de manera posicional, donde cada número simboliza una potencia de 8, ejemplo  $3452_2 = 3 \cdot 8^3 + 4 \cdot 8^2 + 5 \cdot 8^1 + 2 \cdot 8^0$ . Esto permite comprender cómo se representan los valores en este sistema, lo que sirve como base para conectar con su uso en programación.

Se relaciona el sistema octal con el binario al observar que cada dígito octal representa exactamente tres bits. Por ejemplo, el número 7 representa 111 en términos binarios. Esta relación posibilita la agrupación de bits de manera ordenada y clara, simplificando las transformaciones entre binario y octal. Esta conversión ágil se emplea para ilustrar datos binarios de manera más comprensible en programación y hardware.

Se aplica el sistema octal en programación y arquitectura de computadores al utilizarlo, por ejemplo, en sistemas Unix para representar permisos de archivos con dígitos octales como 0755. También se emplea en arquitecturas antiguas, como la serie PDP, donde la estructura de palabras (de 18 o 36 bits) se adapta mejor a agrupaciones octales. Además, en lenguajes de bajo nivel, se usa para definir valores binarios agrupados con claridad, lo cual es útil para configurar registros o máscaras de bits en microcontroladores.

Se compara el sistema hexadecimal con otros sistemas numéricos, como el binario, el octal y el decimal, destacando que el hexadecimal permite representar grandes cantidades de datos binarios en menos espacio y con mayor legibilidad. A diferencia del octal (base 8), que agrupa 3 bits por dígito, el hexadecimal agrupa 4 bits por dígito, lo que se ajusta mejor a la representación de bytes (8 bits).

Se compara el sistema octal con el hexadecimal al observar que, aunque el octal agrupa 3 bits por dígito y es completamente numérico, el hexadecimal agrupa 4 bits por dígito, lo que lo hace más compacto. El octal es útil cuando se desea evitar letras en la notación y se prioriza la lectura numérica, mientras que el hexadecimal se prefiere hoy en día por su eficiencia en representar bytes completos, especialmente en arquitecturas de 8, 16 o 32 bits [1].

## Conversiones

### Octal a decimal

El sistema decimal (base 10) es el que usamos comúnmente en la vida diaria. Convertir octal a decimal permite entender valores numéricos más fácilmente. Usamos potencias de 8. Se toma cada dígito del número octal, empezando desde la derecha, y se multiplica por 8 elevado a la posición correspondiente.

- Ejemplo:  $157_8$  a  $X_{10}$   
 $1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0$   
 $= 1 \times 64 + 5 \times 8 + 7 \times 1$   
 $= 64 + 40 + 7 = 111$

Resultado:  $157_8 = 111_{10}$

### Octal a binario

El sistema binario es el lenguaje nativo de las computadoras. Convertir de octal a binario es muy directo, porque 1 dígito octal equivale a 3 bits. Cada dígito octal se reemplaza por su equivalente binario de 3 bits.

- Ejemplo  $157_8$  a  $X_2$   
 $1 \rightarrow 001$   
 $5 \rightarrow 101$   
 $7 \rightarrow 111$

Resultado:  $157_8 = 00110111_2$

### Octal a Hexadecimal

El sistema hexadecimal se usa mucho en programación (direcciones de memoria, colores, etc.). Convertir de octal a hexadecimal normalmente se hace pasando primero a binario.

En este caso se usa un método indirecto para hacer esta conversión:

- Convierte el octal a binario (como vimos antes).
- Se agrupa los bits en grupos de 4 (empezando desde la derecha).
- Convierte cada grupo de 4 bits a su equivalente hexadecimal

- Ejemplo  $157_8$  a  $X_{16}$

- Primero a binario:

- 1 → 001

- 5 → 101

- 7 → 111

Resultado: **001101111<sub>2</sub>**

- Agrupar en grupo de 4 bits:

- 0 0110 1111

(agregamos ceros a la izquierda para que el número de bits sea múltiplo de 4):

- 0000 0110 1111

- Convertimos a hexadecimal

- 0000 → 0

- 0110 → 6

- 1111 → F

Resultado:  $157_8 = \underline{6F}_{16}$

## Hexadecimal

El sistema hexadecimal es un sistema numérico de base 16 que utiliza los dígitos del 0 al 9 y las letras A a F para representar los valores del 10 al 15. Este sistema funciona de manera posicional, donde cada dígito representa una potencia de 16. Por ejemplo, el número  $1A3_{16}$  se interpreta como  $1 \times 16^2 + 10 \times 16^1 + 3 \times 16^0$ . Este tipo de representación permite expresar valores binarios extensos de forma más compacta y legible.

Se relaciona el sistema hexadecimal con el binario al observar que cada dígito hexadecimal equivale a exactamente cuatro bits binarios. Por ejemplo, el número hexadecimal 2F se traduce como 0010 1111 en binario. Esta correspondencia directa permite realizar conversiones rápidas y exactas entre ambos sistemas, lo que agiliza la interpretación de datos binarios en ámbitos de programación, diseño digital y arquitectura de computadores.

El sistema hexadecimal se aplica en programación y arquitectura de computadores al utilizarlo para representar direcciones de memoria, códigos de colores (como en HTML y CSS), valores en registros, y errores o códigos en depuración. En lenguajes como C o ensamblador, se emplea la notación 0x para indicar que un valor está expresado en hexadecimal, por ejemplo, 0x1A3F. Esta notación es ampliamente utilizada por su claridad y brevedad en la representación de datos binarios.

Se compara el sistema hexadecimal con otros sistemas numéricos, destacando que el hexadecimal permite representar grandes cantidades de datos binarios en menos espacio y con mayor legibilidad. A diferencia del octal (base 8), que agrupa 3 bits por dígito, el hexadecimal agrupa 4 bits por dígito, lo que se ajusta mejor a la representación de bytes (8 bits) [2].

## Conversiones

### Hexadecimal a Binario

El sistema binario es el más cercano al hardware. Convertir de hexadecimal a binario es muy simple, ya que 1 dígito hexadecimal equivale a 4 bits. Solo hay que reemplazar cada dígito hexadecimal por su equivalente binario de 4 bits.

- Ejemplo:  $3F_{16}$  a  $X_2$

$$3 \rightarrow 0011$$

$$F \rightarrow 1111$$

$$\underline{\text{Resultado: } 3F_{16} = 00111111_2}$$

### Hexadecimal a Decimal

El sistema decimal es el que usamos cotidianamente. Para convertir de hexadecimal a decimal, se multiplica cada dígito por potencias de 16 según su posición (empezando desde la derecha con  $16^0$ ), y luego se suman los resultados.

- Ejemplo:  $3F_{16}$  a  $X_{10}$

$$3 \times 16^1 = 48$$

$$F = 15 \rightarrow 15 \times 16^0 = 15$$

$$48 + 15 = 63$$

$$\underline{\text{Resultado: } 3F_{16} = 63_{10}}$$

### Hexadecimal a Octal

El sistema octal se usa en configuraciones de bajo nivel como permisos de archivos o registros. Para convertir de hexadecimal a octal, primero se pasa a binario (4 bits por cada dígito hexadecimal) y luego se agrupan los bits en bloques de 3 desde la derecha para convertir a octal.

- Ejemplo:  $3F_{16}$  a  $X_8$

$$3 \rightarrow 0011$$

$$F \rightarrow 1111$$

$$\rightarrow \text{Binario: } 00111111$$

(Agrupamos de 3 en 3 desde la derecha)

00 111 111

→ agregamos un cero para completar bloques: 000 111 111

000 → 0

111 → 7

111 → 7

Resultado:  $3F_{16} = 077_8$

- [1] S. Rawat, B. Nautiyal, and A. Sah, "A Different and Realistic Approach to Inter Base Conversion for Number System," *Int. J. Comput. Appl.*, vol. 52, no. 18, pp. 37–44, Aug. 2012, doi: 10.5120/8306-1941.
- [2] A. C. Jha, "Positional Number System," *NUTA J.*, vol. 7, no. 1-2 SE-Articles, pp. 1–9, Dec. 2020, doi: 10.3126/nutaj.v7i1-2.39924.