**SESION #2**

Grupo de: Luna Jaimes, Samuel Ortiz, Miguel Urrea

**Consulta y explica con un ejemplo la Aritmética de punto fijo.**

La aritmética de punto fijo es una técnica utilizada en la computación y la electrónica para realizar operaciones matemáticas en números representados de manera fija, es decir, con una cantidad predefinida de bits para la parte entera y la parte fraccionaria de un número. A diferencia de la aritmética de punto flotante, en la que se asigna una cantidad variable de bits a la parte entera y fraccionaria, la aritmética de punto fijo utiliza una representación fija de los números, lo que puede resultar en limitaciones en el rango y la precisión de los valores.

En la aritmética de punto fijo, los valores se representan con una cierta cantidad de bits para la parte entera y una cierta cantidad de bits para la parte fraccionaria. Por ejemplo, en una representación de punto fijo de 8 bits con 4 bits para la parte entera y 4 bits para la parte fraccionaria, el rango de valores posibles sería desde -8 hasta 7.9375, con una resolución de 0.0625.

Supongamos que tenemos dos números representados en aritmética de punto fijo de 8 bits: A = 3.75 y B = 1.5. Ambos números tienen 4 bits para la parte entera y 4 bits para la parte fraccionaria.

Representación de A: 0011.1100

Representación de B: 0001.1000

Ahora, vamos a realizar una suma utilizando la aritmética de punto fijo:

0011.1100 (A)

+ 0001.1000 (B)

-------------

0101.0100

El resultado de la suma es 0101.0100, que en decimal es 5.25. Sin embargo, es importante tener en cuenta que, debido a la representación de punto fijo, el resultado no puede almacenar la parte decimal 0.25 con precisión completa, ya que solo tenemos 4 bits para la parte fraccionaria. Por lo tanto, el resultado se redondea a 5.25.

La aritmética de punto fijo es útil en aplicaciones donde se necesita realizar cálculos con recursos limitados, como en microcontroladores o sistemas embebidos, ya que suele requerir menos recursos computacionales que la aritmética de punto flotante. Sin embargo, es importante considerar las limitaciones de rango y precisión al utilizar esta técnica.

**Consulta y explica con un ejemplo la Aritmética de punto flotante**

La aritmética de punto flotante es un método utilizado en sistemas computacionales para representar y realizar operaciones matemáticas en números reales con decimales. Esta técnica permite una mayor flexibilidad y precisión que la aritmética de punto fijo, ya que asigna bits de manera variable tanto a la parte entera como a la parte fraccionaria, lo que permite un rango más amplio de valores y una mayor precisión en las operaciones.

En la aritmética de punto flotante, la representación se basa en dos componentes principales: la mantisa y el exponente. La mantisa almacena la parte significativa del número, mientras que el exponente determina la escala o magnitud del número. Además, la representación se realiza a través de niveles de bits para el signo, exponente y mantisa. Por lo general, se utiliza una base de 2 (sistema binario) en la aritmética de punto flotante, aunque en algunos casos se utiliza la base 10 (decimal).

Es importante tener en cuenta que, aunque permite una mayor precisión en comparación con la aritmética de punto fijo, la aritmética de punto flotante todavía presenta limitaciones en términos de precisión debido a la representación finita de los números en binario.

**Convertir a binario, octal y hexadecimal cada uno de los siguientes decimales:**

a)325

Binario

325 / 2 = 162 residuo 1

162 / 2 = 81 residuo 0

81 / 2 = 40 residuo 1

40 / 2 = 20 residuo 0

20 / 2 = 10 residuo 0

10 / 2 = 5 residuo 1

5 / 2 = 2 residuo 1

2 / 2 = 1 residuo 0

1 / 2 = 0 residuo 1

La representación binaria de 325 es 101000101.

Conversión a Octal

325 / 8 = 40, residuo 5

40 / 8 = 5, residuo 0

5 / 8 = 0, residuo 5

La representación octal de 325 es 505.

Conversión a Hexadecimal (Base 16):

325 / 16 = 20, residuo 5 (5 en hexadecimal)

20 / 16 = 1, residuo 4 (4 en hexadecimal)

1 / 16 = 0, residuo 1 (1 en hexadecimal)

b) 954

Conversión a Binario:

954 / 2 = 477 residuo 0

477 / 2 = 238 residuo 1

238 / 2 = 119 residuo 0

119 / 2 = 59 residuo 1

59 / 2 = 29 residuo 1

29 / 2 = 14 residuo 1

14 / 2 = 7 residuo 1

7 / 2 = 3 residuo 1

3 / 2 = 1 residuo 1

1 / 2 = 0 residuo 1

La representación binaria de 954 es 1110110110.

Conversión a Octal:

954 / 8 = 119 residuo 2

119 / 8 = 14 residuo 7

14 / 8 = 1 residuo 6

1 / 8 = 0 residuo 1

La representación octal de 954 es 1666.

Conversión a Hexadecimal:

954 / 16 = 59 residuo B

59 / 16 = 3 residuo 11 (B en hexadecimal)

3 / 16 = 0 residuo 3

La representación hexadecimal de 954 es 3BA.

c) 1562

Conversión a Binario:

1562 / 2 = 781 residuo 0

781 / 2 = 390 residuo 1

390 / 2 = 195 residuo 0

195 / 2 = 97 residuo 1

97 / 2 = 48 residuo 1

48 / 2 = 24 residuo 0

24 / 2 = 12 residuo 0

12 / 2 = 6 residuo 0

6 / 2 = 3 residuo 0

3 / 2 = 1 residuo 1

1 / 2 = 0 residuo 1

La representación binaria de 1562 es 11000001110.

Conversión a Octal:

1562 / 8 = 195 residuo 2

195 / 8 = 24 residuo 3

24 / 8 = 3 residuo 0

3 / 8 = 0 residuo 3

La representación octal de 1562 es 303.

Conversión a Hexadecimal:

1562 ÷ 16 = Cociente: 97, Residuo: 10 (A en hexadecimal)

97 ÷ 16 = Cociente: 6, Residuo: 1

6 ÷ 16 = Cociente: 0, Residuo: 6

La representación hexadecimal de 1562 es 61A.

d) 2463

Conversión a Binario:

2463 / 2 = 1231 residuo 1

1231 / 2 = 615 residuo 1

615 / 2 = 307 residuo 1

307 / 2 = 153 residuo 1

153 / 2 = 76 residuo 1

76 / 2 = 38 residuo 0

38 / 2 = 19 residuo 0

19 / 2 = 9 residuo 1

9 / 2 = 4 residuo 1

4 / 2 = 2 residuo 0

2 / 2 = 1 residuo 0

1 / 2 = 0 residuo 1

La representación binaria de 2463 es 100110001111.

Conversión a Octal:

2463 / 8 = 307 residuo 7

307 / 8 = 38 residuo 3

38 / 8 = 4 residuo 6

4 / 8 = 0 residuo 4

La representación octal de 2463 es 7463.

Conversión a Hexadecimal:

2463 / 16 = 153 residuo 15 (F en hexadecimal)

153 / 16 = 9 residuo 9

9 / 16 = 0 residuo 9

La representación hexadecimal de 2463 es 9F9.

**Convertir a decimal los siguientes binarios.**

111001

5 = 1

4 = 1

3 = 1

2 = 0

1 = 0

0 = 1

Decimal = 1 \* 2^5 + 1 \* 2^4 + 1 \* 2^3 + 0 \* 2^2 + 0 \* 2^1 + 1 \* 2^0

Decimal = 32 + 16 + 8 + 0 + 0 + 1

Decimal = 57

Entonces, la representación decimal del binario 111001 es 57.

1010101

6 = 1 (el dígito más a la izquierda)

5 = 0

4 = 1

3 = 0

2 = 1

1 = 0

0 = 1

Decimal = 1 \* 2^6 + 0 \* 2^5 + 1 \* 2^4 + 0 \* 2^3 + 1 \* 2^2 + 0 \* 2^1 + 1 \* 2^0

Decimal = 64 + 0 + 16 + 0 + 4 + 0 + 1

Decimal = 85

Entonces, la representación decimal del binario 1010101 es 85.

11100101

7 = 1 (el dígito más a la izquierda)

6 = 1

5 = 1

4 = 0

3 = 0

2 = 1

1 = 0

0 = 1

Decimal = 1 \* 2^7 + 1 \* 2^6 + 1 \* 2^5 + 0 \* 2^4 + 0 \* 2^3 + 1 \* 2^2 + 0 \* 2^1 + 1 \* 2^0

Decimal = 128 + 64 + 32 + 0 + 0 + 4 + 0 + 1

Decimal = 229

Entonces, la representación decimal del binario 11100101 es 229.

101011110101

11 = 1

10 = 0

9 = 1

8 = 0

7 = 1

6 = 1

5 = 1

4 = 1

3 = 0

2 = 1

1 = 0

0 = 1

Decimal = 1 \* 2^11 + 0 \* 2^10 + 1 \* 2^9 + 0 \* 2^8 + 1 \* 2^7 + 1 \* 2^6 + 1 \* 2^5 + 1 \* 2^4 + 0 \* 2^3 + 1 \* 2^2 + 0 \* 2^1 + 1 \* 2^0

Decimal = 2048 + 0 + 512 + 0 + 128 + 64 + 32 + 16 + 0 + 4 + 0 + 1

Decimal = 2821

Entonces, la representación decimal del binario 101011110101 es 2821.

**Convertir a decimal los siguientes octales.**

1. 65

6 \* 8^1 + 5 \* 8^0 = 48 + 5 = 53

1. 327

3 \* 8^2 + 2 \* 8^1 + 7 \* 8^0 = 192 + 16 + 7 = 215

1. 2586

2 \* 8^3 + 5 \* 8^2 + 8 \* 8^1 + 6 \* 8^0 = 1024 + 320 + 64 + 6 = 1414

1. 4050

4 \* 8^3 + 0 \* 8^2 + 5 \* 8^1 + 0 \* 8^0 = 2048 + 0 + 40 + 0 = 2088

**Convertir a decimal los siguientes hexadecimales.**

Los dígitos A, B, C, D, E y F en hexadecimal representan los números 10, 11, 12, 13, 14 y 15 en decimal, respectivamente.

1. 15A

1 \* 16^2 + 5 \* 16^1 + 10 \* 16^0 = 256 + 80 + 10 = 346

1. 25BD

2 \* 16^3 + 5 \* 16^2 + 11 \* 16^1 + 13 \* 16^0 = 8192 + 1280 + 176 + 13 = 9653

1. CFF2

12 \* 16^3 + 15 \* 16^2 + 15 \* 16^1 + 2 \* 16^0 = 12288 + 3840 + 240 + 2 = 16470

1. 15CF2

1 \* 16^4 + 5 \* 16^3 + 12 \* 16^2 + 15 \* 16^1 + 2 \* 16^0 = 65536 + 20480 + 3072 + 240 + 2 = 89230

**Realiza el procedimiento para las siguientes sumas binarias.**

1. 1111100000 + 111110

1. 01010101010 + 111

1010110001

1. 10011100 + 00001

**Realiza el procedimiento para las siguientes restas binarias**

1. 1111111 – 10101

1. 11100011111 – 1010110100110

Realiza el procedimiento para las siguientes restas binarias

1. 1111011 \* 111100

1. 1111111111\* 110