Taller 4 Adrian Balderas Rosas

A. Investiga para responder lo siguiente en tu cuaderno.

1.- Se realizan moviendo los bits de un numero binario hacia la izquerda o hacia la derecha.
USO: Esta operacion equivale a dividir el numero por una potencia de dos. Desplazar un numero a la derecha por una posición equivale a dividir el numero por 2 y descartar el residuo. Desplazar por n posiciones es equivalente a dividir el numero por 2^n, y para numeros enteros, el residuo se descarta.

2. Es fundamentar por estas razones clave:
1. Simplicidad del hardware
2. Representacion de datos
3. Operaciones logicas y Aritmeticas
4. Optimización de recursos
5. Codificación y Decodificacion
6. Interoperabilidad y Estandares

3. El "Overflow" Ocurre cuando el resultado de una operacion aritmetica excede la capacidad maxima que puede ser representada con el numero de bits disponibles en el sistema. Este fenomeno puede dar lugar a resultados incorrectos o inesperados debido a que el numero "desborda" la representacion limitada.

4. Mientras que el proceso basico de multiplicacion es similar en ambos sistemas, la multiplicacion binaria se basa en operaciones mas sencillas debido a la base 2, mientras que la multiplicacion decimal involucra mas pasos y manipulacion de digitos debido a la base 10.

5.
1. Alineación: coloca el divisor y el dividendo en la configuracion adecuada, similar a la division larga en decimal
2. Division paso a paso:
- Comienza desde el bit mas significativo del dividendo y compara con el divisor.
- Si el divisor cabe en el grupo de bits del dividendo, escribe un 1 en el cociente y resta el divisor del grupo de bits.

Taller 4

- Si el divisor no cabe, escribe un 0 en el cociente y desplaza el siguiente bit del dividendo hacia abajo.
- Repite el proceso para todos los bits del dividendo

3. Resto: El numero que queda despues de todas las restas es el resto de la division.

Aunque la division binaria y decimal siguen principios matematicos similares, sus diferencias estan en como se manejan los digitos y las bases numericas.

6. Es una tecnica utilizada en la aritmetica binaria para representar numeros enteros con signo y realizar operaciones aritmeticas, como suma y resta de manera eficiente.

- Uso en la aritmetica Binaria

1. Representacion de numeros negativos.
2. Suma y Resta.
3. Deteccion de overflow

7.

complemento a uno: Utiliza la inversion de todos los bits del numero (Cambiando 0s por 1s y 1s por 0s)

complemento a dos: se obtiene el complemento a dos tomando el complemento a uno del numero y luego sumando 1 al resultado.

Diferencias

- Representacion del cero
- Suma y resta

8. razones

1. Unica Representacion para cero
2. simplificacion en la Aritmetica:
3. Propiedad del Overflow
4. Unicidad del bit mas significativo (MSB).
5. Facil negacion.

9. El tamaño del registro en complemento a dos afecta directamente el rango de numeros representables, A medida que aumenta el numero de bits en el registro, el rango de numeros que se pueden representar tambien aumenta.

Taller 4

10. Ventajas:

1. Simplificacion de la suma y resta
2. Unica representacion del cero
3. Facil deteccion de desbordamiento
4. Algoritmos mas eficientes
5. Amplia adopcion

11. Razones:

1. Complejidad en las Operaciones aritmeticas
2. Doble representacion del cero
3. Dificultad en la deteccion de desbordamiento
4. Ineficiencia en la representacion de numeros negativos.

12. Consideremos una representacion de 4 bits:

- Sin signo: Los numeros representables van de 0 (0000) a 15 (1111)
- Con signo y magnitud: Los numeros representables van de -7 (1111) a 7 (0111).

Hay dos representaciones para el cero +0 (0000) y -0 (1000).

13. Pasos generales

1. Alinear los numeros: Coloca los numeros uno debajo del otro, alineando los digitos de acuerdo a su valor posicional.
2. Restar digito a digito: Comienza por la derecha, y resta los digitos correspondientes.
3. Pedir Prestado: si el digito de arriba es menor que el digito de abajo, pide prestado 1 al siguiente digito de la izquierda. Esto equivale a sumar 2 al digito actual

Ejemplo:

$$\begin{array}{r} 1101 \\ -\ 101 \\ \hline 1000 \end{array}$$

el sobrante lo bajamos directamente

Taller 4

14.
$$\begin{array}{r} 11110 \\ +\ 1000001 \\ \hline 1011111 \end{array}$$

15.
$$\begin{array}{r} 1000000 \\ +\ 1100110 \\ \hline 10100110 \end{array}$$

16.
$$\begin{array}{r} 1111001 \\ +\ 101101 \\ \hline 10100110 \end{array}$$

17.
$$\begin{array}{r} 101100 \\ +\ 11100110 \\ \hline 100010010 \end{array}$$

18.
$$\begin{array}{r} 1100101 \\ +\ 1100110 \\ \hline 11001011 \end{array}$$

19.
$$\begin{array}{r} 1100100 \\ -\ 10101 \\ \hline 1001111 \end{array}$$

20.
$$\begin{array}{r} 1100011 \\ -\ 100000 \\ \hline 1000011 \end{array}$$

21.
$$\begin{array}{r} 1001110 \\ -\ 10101 \\ \hline 111001 \end{array}$$

22.
$$\begin{array}{r} 110011 \\ -\ 101101 \\ \hline 000110 \end{array}$$

23.
$$\begin{array}{r} 110010 \\ -\ 100000 \\ \hline 010010 \end{array}$$

24. 32 en su complemento a 2

$$00100000 = \begin{array}{r} 11011111 \\ +\ 00000001 \\ \hline 11100000 \end{array}$$

25. 21 en su complemento a 2

$$00010101 = \begin{array}{r} 11101010 \\ +\ 00000001 \\ \hline 11101011 \end{array}$$

26. 12 en su complemento a 2

$$00001100 = \begin{array}{r} 11110011 \\ +\ 00000001 \\ \hline 11110100 \end{array}$$