

Bloque 0

La información digital

Prácticas de
Introducción a los Computadores
Curso 2007-2008

SESIÓN 1

Sistemas de codificación numéricos

Objetivos

Esta práctica tiene como objetivo aplicar los conceptos básicos de códigos numéricos. Estos conceptos se pondrán en práctica con el programa ICB, que puede ser muy útil para que el alumno compruebe si ha aprendido los conceptos correspondientes.

Conocimientos y materiales necesarios

Para el adecuado aprovechamiento de esta sesión de prácticas, el alumno necesita:

- Conocer los sistemas de codificación numéricos, en particular los de los números naturales y enteros (binario natural, hexadecimal, signo-magnitud, complemento a 2 y exceso a Z).
- El programa ICB que se suministrará durante la sesión de prácticas.

Desarrollo de la práctica

1. Introducción al programa de códigos numéricos

Para practicar conceptos relacionados con códigos numéricos se va a utilizar el programa ICB (Intérprete de Códigos Binarios). Ejecútalo siguiendo las instrucciones que te indique tu profesor. El programa tiene varias secciones para practicar distintos conceptos. Vamos a explorarlas en los siguientes apartados.

2. Bases de representación

Escoge la opción de menú *Ver*→*Bases*. En la pantalla que aparece puedes escribir números naturales en base binaria, decimal, hexadecimal u octal, y ver cuál sería la representación en otra base.

- ☐ Calcula «a mano» cuál es la representación en binario, hexadecimal y octal del número 35. Anota el resultado de la conversión en la siguiente tabla.

Decimal	Hexadecimal	Octal
35		

- ☐ A continuación escribe ese número en el campo reservado para números decimales y comprueba si has realizado bien la transformación de bases.

3. Representación de enteros

Escoge la opción de menú *Ver→Enteros*. En la pantalla que aparece puedes escribir números enteros y ver su representación en decimal, en signo-magnitud, en complemento a 2 y en exceso a Z.

- ☐ Vamos a trabajar con 5 bits. Para ello, escribe el valor 5 en la caja de arriba a la izquierda. ¿Qué rango hay en signo-magnitud?^[1] ¿Qué rango hay en complemento a 2?^[2] ¿Qué rango hay en exceso central?^[3] ¿Por qué se puede representar un número más en C2 y en exceso central que en signo-magnitud?
- ☐ Codifica «a mano» los números 10 y -10 en signo-magnitud, complemento a 2 y exceso a Z y comprueba que los has codificado bien con el programa.
- ☐ Piensa qué secuencia binaria habría que introducir en la caja de exceso a Z para que la secuencia que aparezca en complemento a 2 sea 11001. Comprueba tu respuesta con el programa. ¿Cuál era la secuencia?^[4]

1

2

3

4

4. Comparación de formatos

Escoge la opción de menú *Ver→Comparación de formatos*. Esta pantalla permite comprobar cómo una secuencia de bits tiene distintos significados según con qué formato la interpretemos.

- ☐ Deduce «a mano» con qué valor decimal se corresponde la secuencia 10111 si la interpretamos como binario natural, como signo-magnitud, como complemento a 2 y como exceso a 9. Anota los resultados en la siguiente tabla.

Secuencia	B. natural	Sig-Mag	C. a 2	Ex-9
10111				

- ☐ Comprueba con el programa que has realizado bien todas las transformaciones.

5. Aritmética

Escoge la opción de menú *Ver→Aritmética*, o pulsa el último botón de la barra de botones. Esta pantalla permite estudiar paso a paso cómo se realizan sumas o restas en binario o en hexadecimal.

5.1. Sumas

Comenzaremos realizando una suma en estas bases, interpretando de diferente modo el resultado.

- ❑ Vamos a realizar la suma de los números 137 y 43. En primer lugar haremos la suma en hexadecimal. Comienza por escribir estos números en base 16. Haz la conversión en papel, por el método de las divisiones sucesivas ¿Cuál es la representación en hexadecimal del 137?^[5] ¿Y del 43?^[6] Utiliza la pantalla de cambio de bases del programa para verificar tus respuestas.
- ❑ Realiza la suma en papel de estas cantidades hexadecimales. Recuerda que el método consiste en sumar las cantidades representadas por las cifras, escribiendo debajo lo que salga si es menor que la base. Si excede a la base se escribe en cuánto la excede y “se lleva una”. ¿Qué te sale como resultado?^[7]
- ❑ Ahora vamos a usar el programa ICB para verificar si lo hemos realizado bien. Vuelve a la pantalla de aritmética y pulsa sobre el triángulo que tiene la punta hacia arriba. Esto abre un diálogo en el que podrás introducir los operandos y la operación (suma o resta) a realizar con ellos. Asegúrate de que tienes elegida la base hexadecimal. Cuando hayas introducido los datos, aparecerá la suma planteada en la pantalla. Pulsando el triángulo que señala a la derecha, podremos realizar esta suma “paso a paso” (es decir, columna a columna). Mira cómo el programa resuelve cada paso y cuándo “se lleva una”. Verifica si te había salido lo mismo.

5

6

7

Seguidamente realizaremos la misma suma, pero en base dos.

- ❑ Escribe el 137 en base 2 (quizás te resulte más fácil obtenerlo a partir de la versión hexadecimal que tienes del ejercicio anterior)^[8]. Escribe también el 43 en base 2^[9].
- ❑ Plantea la suma y resuélvela en papel. ¿Qué obtienes?^[10]
- ❑ Utiliza la pantalla de aritmética del programa para verificar si lo has hecho bien (asegúrate de usar ahora la base 2).

8

9

10

Fíjate en la suma binaria que acabas de realizar. Estamos trabajando con números naturales de 8 bits ¿es el resultado correcto?^[11] Para saberlo debemos simplemente observar el bit de acarreo final.

11

Si estuviéramos trabajando con números enteros de 8 bits, representados en complemento a 2, las cantidades sumadas ya no serían probablemente 137 y 43. ¿Cuáles serían?^{[12][13]}. En este caso para saber si el resultado es correcto no hay que mirar el acarreo final, sino comparar los signos de operandos y resultado para ver si ha habido *overflow* ¿Cuál sería el bit de *overflow* para esta suma?^[14]. ¿Qué cantidad representa el código que hemos obtenido como resultado?^[15]. Comprueba si es lo que cabía esperar a la vista de las cantidades que hemos sumado.

12

13

14

15

5.2. Restas

La resta es un poco más engorrosa que la suma, ya que sólo se puede restar una cifra de otra, si la de arriba es mayor. Si no lo es, se toma un 1 prestado de la columna de la izquierda, para añadir una “cifra de las decenas” a la cifra que teníamos y así hacer posible esa resta. Por ejemplo, si las cifras a restar son 5 y 7, siendo 5 la de arriba, para hacer posible la resta antepone un 1 al 5, y así realizamos la resta $15-7$. Ese 1 lo hemos tomado prestado de la columna siguiente, y por tanto hay que restarle 1 a esa columna cuando vayamos a calcularla después. Es la forma de “llevarse una” cuando se resta.

Con esta idea en mente, realiza sobre el papel la resta de las cantidades del ejercicio anterior, 137 y 43. Haz primero la resta en base 10, a la que estás acostumbrado, para observar cuándo “te llevas una” interpretándolo en los términos del “1 prestado” del párrafo anterior.

- ☐ Ahora haz la resta en base 16. Ten en cuenta que cuando tomes un 1 prestado, y lo antepongas a otra cifra, el nuevo número de dos cifras estará escrito en hexadecimal. Por ejemplo, al restar $2-3$, tomas prestado un 1 para convertir el 2 en 12, pero este 12 es 12h y por tanto vale 18d, por lo que la resta sale 15d (Fh). Poniendo cuidado en este detalle ¿qué obtienes?^[16]
- ☐ Utiliza ICB para verificar tu respuesta. Ten en cuenta que ICB no “explica” bien cómo se hace la resta de una columna. Así, por ejemplo, en el primer paso de la resta, aunque produce el resultado correcto, la “explicación” que aparece en la línea inferior, que dice “ $9-B=1E$ ”, debería más bien decir “ $19-B=E$ ”

16

Repite la resta anterior pero ahora en binario. Inténtalo primero en papel. Cada vez que el bit de arriba sea un 0 y el de abajo un 1, deberás tomar un “bit prestado” a la columna siguiente, para convertir el 0 en 10 y así poder hacer la resta $10-1=1$. Recuerda descontar en la columna siguiente el bit que hayas tomado prestado. Esto puede volverse farragoso. Si te atascas, no intentes continuar y usa ICB para ver la solución. ¿Cuál es el resultado que obtiene el programa ICB?^[17] Si lo habías conseguido hacer tú solo en el papel ¡enhorabuena! Restar en binario con tanto bit de préstamo no es nada sencillo.

17

Si no lo habías conseguido, no te desanimes, por suerte no tendrás que hacer nunca estas restas, ya que las evitaremos con un truco. Restar es lo mismo que sumar el opuesto, de modo que la resta $137-43$ la transformamos en la suma $137+(-43)$. Sumar es mucho más sencillo. Todo lo que tienes que hacer es sumar el código binario del 137 con el código binario del -43 (el complemento a 2 del 43). ¿Cuál es éste código?^[18]

18

- ☐ Haz la suma binaria en papel del código del 137 y el código del -43. ¿Qué te sale en base 2?^[19]
- ☐ Comprueba que te sale lo mismo que lo que salía en ICB cuando hiciste la resta del 137 menos el 43.
- ☐ ¿Hay overflow en esta operación?^[20]

19

20

Como ves, gracias al complemento a 2, no necesitamos saber cómo realizar las restas. Basta con saber cómo realizar las sumas. Esto tiene importantes implicaciones en el diseño del hardware que hace las operaciones en un computador.

6. Ejercicios adicionales

- ⇒ Interpretar la secuencia 11001 en formato complemento a dos, binario y exceso a 4. Verifica los resultados con el programa de codificación.
- ⇒ Haz una suma de dos números negativos, codificados en 8 bits en complemento a 2. Comprueba si el resultado es correcto. Si eliges números no muy grandes, debería serlo.
- ⇒ Haz una suma de dos números negativos, codificados en 8 bits en complemento a 2, pero elegidos de forma que el resultado se salga del rango. Comprueba que al hacer esta suma el resultado es incorrecto. ¿Cómo se detecta el desbordamiento en este caso?
- ⇒ Busca dos números de seis bits cuya suma genere carry y overflow simultáneamente. Verifica los resultados con el programa de codificación.