Práctico 1 – Detección y corrección de errores. Representación Interna de Datos

Objetivos:

- Familiarizarse con los distintos mecanismos de representación interna de datos utilizados por los computadores modernos
- Conocer y practicar con mecanismos de detección y corrección de errores en la transmisión de información

---

Preguntas teóricas:

(a) Explique (y ejemplifique) por qué agregar un único bit de paridad a un sistema de codificación binario permite detectar errores pero no corregirlos.

(b) ¿Por qué los códigos de Hamming no son adecuados para transmisiones?

(c) Se desea definir una nueva tabla de codificación de caracteres adaptada a la realidad de nuestro país. Usando como base la tabla ASCII presentada en las notas de teórico, discuta qué elementos deben agregarse y cuáles podrían eliminarse.

(d) Indique de qué manera se marca el fin de un string en los lenguajes de programación que han utilizado en las asignaturas de ha cursado en Facultad.

(e) Indique cuáles de las representaciones de enteros con signo tienen diferente cantidad de números positivos que negativos.

(f) ¿Por qué en punto flotante IEEE 754 se utiliza la representación con desplazamiento para el exponente?

(g) Discuta sobre las ventajas y desventajas de disponer de números desnormalizados.

(h) En punto flotante IEEE 754 de simple precisión, indique cuotas superiores en el error de representación de números en los rangos [0,1], [2,4], [128,256] y $[2^{20},2^{21}]$.

Ejercicio 1 ★★

1. Explicar si los siguientes sistemas son capaces de corregir errores:
   - Código de Hamming
   - Paridad
   - Entrelazado 2 de 5
   - Paridad horizontal+vertical

Codificar en el sistema de Hamming la tira 1001

Ejercicio 2 ★

Se desea transmitir dígitos decimales en código BCD a través de un canal con ruido. Con ese objetivo se genera a partir del código BCD un código de Hamming de 7 bits. Decodificar el siguiente mensaje asumiendo que a lo sumo ha ocurrido un único error en cada palabra del código.

1001000 – 0000000 – 1110100 – 0001111

- Orden de los bits: $m_4 m_3 m_2 p_3 m_1 p_2 p_1$
- Digito BCD: $m_4 m_3 m_2 m_1$

Ejercicio 3 ★

Escribir los números (decimales) 1023, 45216 y 71822 en representación de enteros sin signo binario de 16 bits.

Ejercicio 4 ★ (en OpenFing)

Representar los números (decimales) 524 y -3264 en los siguientes formatos de enteros con signo de 16 bits:

a) Valor y signo
b) Desplazamiento
c) Complemento a 1
d) Complemento a 2

Ejercicio 5 ★★

Explicar en qué orden se pueden realizar las sumas de los siguientes cuatro dígitos hexadecimales, que representan números en notación complemento a 2 de 16 bits, para que en ningún momento se produzca overflow.

7744 + 5499 + 6788 + AB68 + 88BD + 9879 = 0003

Ejercicio 6 ★★

Realizar las siguientes operaciones en complemento a 2, indicando el valor de los bits de condición Z (cero), N (negativo), C (acarreo) y V (overflow).

a) 0x2977 + 0x5689
b) 0xCAFE + 0xB007
c) 0xF21C + 0x0DE4
d) 0x5789 – 0x021F

Ejercicio 7 ★

Asumiendo que se disponen de 16 bits en total y 6 bits para la parte fraccional.

a) Indique el mayor número representable.
b) Represente 5.125 y -8.75

Ejercicio 8 ★★

Sumar los números 3EE00000h y 3D800000h, representados en punto flotante, formato IEEE de precisión simple (1 bit para el signo, 8 bits para el exponente y 23 bits para la mantisa) y expresar el resultado normalizado en hexadecimal.

Ejercicio 9 ★★

Representar los números $23x2^{24}$, $23x2^{-24}$, 1049318 y 104891 en los siguientes formatos de punto flotante:

a) 4 bits para el exponente y 20 para la mantisa
b) 8 bits para el exponente y 10 para la mantisa

Ejercicio 10 ★★★

Representar los números $67x2^{-7}$ y $37x2^{7}$ en punto flotante de 16 bits (signo, exponente de 5 bits y mantisa de 10 bits). Sumar ambos números. Dividir el primer numero entre el segundo.

Ejercicio 11 ★★★

Se desea hacer una modificación a los números de punto flotante de precisión simple del estándar de IEEE (un bit de signo, 8 bits de exponente y 23 de mantisa) para transformarlos a números de punto flotante sin signo.

1. Indicar cuál es el número más grande y el menor número mayor que cero en la representación de precisión simple de IEEE.
2. Plantear dos posibles formas de modificar el estándar para representar números sin signo (y seguir usando 32 bits), aprovechando el bit que queda disponible. Para cada una de ellas indique el mayor número y el menor número mayor que cero que se puede representar.

Ejercicio 12 ★★

Dadas las dos tiras de bits:

1. 1000 1111 1110 1111 1100 0000 0000 0000
2. 0000 0000 0000 0000 0000 0000 0000 0000

Indicar qué representan en las siguientes representaciones:

a) Entero en complemento a 2
b) Entero sin signo
c) Decimal empaquetado de 4 bits
d) Punto flotante simple precisión

Ejercicio 13 ★★

1. Comparar los rangos de información que pueden almacenarse en dos bytes, si el formato es:
   a) Decimal empaquetado
   b) Desempaquetado (1 byte por dígito)
   c) Binario sin signo
   d) Binario con signo
2. Representar el número 564143415 en formato decimal empaquetado. ¿Cuántos bytes ocupa?