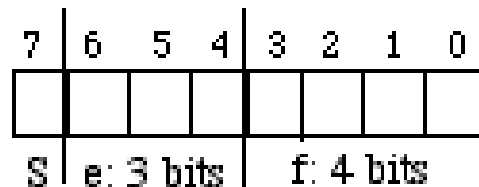


Problemas propuestos

- Pasar de base 2 a base 10: $(1011010)_2$, $(0100111001)_2$
- Pasar de base 10 a base 2: 21, 58, 73
- Pasar de base 10 a base 2, octal y hexadecimal: 35, 97
- Pasar a base 2 y a base 10: $(157)_8$, $(430)_8$
- Pasar a base 2 y a base 10: $(3B)_{16}$, $(DF)_{16}$
- En un sistema se describen números en punto flotante en 8 bits según el esquema de la figura y un formato del tipo:
 $(-1)^s * 2^{e-3} * (1.f)$
 - a). Encontrar el valor del número (en base 10) correspondiente a los siguientes datos en código hexadecimal: $(7D)_{16}$, $(A6)_{16}$
 - b). Indicar en este formato (en código hexadecimal) el valor: $(3.75)_{10}$, $(-0.625)_{10}$



Problemas propuestos

- Realizar las siguientes sumas en binario:
 $75 + 43$, $98 + 87$.
- Realizar el esquema de un circuito “full-adder” utilizando circuitos “half-adder”.
- Realizar las siguientes restas en binario:
 $75 - 43$, $68 - 31$
- Realizar las siguientes multiplicaciones en binario:
 $19 * 13$, $11 * 21$
- Construir un circuito multiplicador de números de 4 bits, utilizando módulos de multiplicación de 1 bit y sumadores de 4 bits para realizar la suma de filas.

- La división A/B de dos números A ($a_3a_2a_1a_0$) y B ($b_3b_2b_1b_0$) de 4 bits, calculando el cociente Q ($q_3q_2q_1q_0$) y el resto R ($r_3r_2r_1r_0$), puede realizarse según el siguiente método:
 - Tomar el dividendo como $(000a_3a_2a_1a_0)$ y el divisor como $(b_3b_2b_1b_0)$.
 - Sea D_1 los 4 bits de la izquierda ($000a_3$) del dividendo, comparar D_1 con el divisor B , si $D_1 \geq B$ el bit del cociente q_3 es 1 y se genera X ($x_3x_2x_1x_0$) = $D_1 - B$; si no el cociente Q_3 es 0 y X ($x_3x_2x_1x_0$) = D_1 .
 - Tomar D_2 como $(x_2x_1x_0a_2)$, si $D_2 \geq B$ el bit del cociente q_2 es 1 y se genera Y ($y_3y_2y_1y_0$) = $D_2 - B$; si no el cociente q_2 es 0 e Y ($y_3y_2y_1y_0$) = D_2 .
 - Tomar D_3 como $(y_2y_1y_0a_1)$, si $D_3 \geq B$ el bit del cociente q_1 es 1 y se genera Z ($z_3z_2z_1z_0$) = $D_3 - B$; si no el cociente q_1 es 0 y Z ($z_3z_2z_1z_0$) = D_3 .
 - Tomar D_4 como $(z_2z_1z_0a_0)$, si $D_4 \geq B$ el bit del cociente q_0 es 1 y se genera el resto R ($r_3r_2r_1r_0$) = $D_4 - B$; si no el cociente Q_0 es 0 y el resto R ($r_3r_2r_1r_0$) = D_4 .
- a) Indicar con ejemplos por qué el cociente y el resto deben tener 4 bits.
- b) Realizar las divisiones $13/5$, $11/3$.

Problemas propuestos

- Indicar el valor numérico entero al que corresponde cada valor de un código de 4 bits en notación con bit de signo, complemento-2 ó complemento-1.
- Realizar las siguientes operaciones en c-a-2 utilizando el número mínimo de bits necesario para que no haya desbordamiento: $-3 + 7$, $5 - 7$, $-13 - 8$, $10 - 6$.
- Realizar las siguientes operaciones en c-a-2 y c-a-1 para 8 bits: $75 - 43$, $68 - 31$, $-28 - 80$, $-60 - 96$.
- Construir un circuito restador en c-a-2 en base a un módulo cambiador de bits (a definir) y un sumador. Modificar el circuito para que se pueda sumar ó restar según el valor de una señal de control S/R (0 Suma, 1 Resta) sustituyendo el módulo cambiador de bits por el módulo adecuado.

Problemas propuestos

- Realizar los códigos BCD con peso:
(6, 2, 2, 1), (2, 4, 2, 1), (7, 3, 1, -2), (8, 7, -2, -4).
Indicar si alguno de los códigos BCD es autocomplementario. Buscar esta condición en el caso de poder asignar varios códigos a un mismo dígito.
- Demostrar que la suma de los pesos de un código autocomplementario es igual a 9.
- Convertir las siguientes palabras de código binario a código Gray.
(01110110010), (100110101)
- Convertir las siguientes palabras de código Gray a código binario.
(01110110010), (100110101)

- El código Hamming es un código de corrección de error simple. Al código original (M0M1M2) se le añaden 3 bits de paridad P0, P1 y P2. El valor de P0, P1 y P2 se calcula según la siguiente tabla que indica en cada fila una condición de paridad par. Así cada fila F0 (P0, M0, M1), F1 (P1, M0, M2), F2 (P2, M1, M2) debe tener paridad par, con lo que del original (M0M1M2) se forma (P0P1M0P2M1M2)

1	2	3	4	5	6	
P0	P1	M0	P2	M1	M2	
×		×		×		F0
	×	×			×	F1
			×	×	×	F2

Al recibir datos se comprueba si las paridades establecidas son correctas o no, en cada fila si lo están se asocia un 0, si no un 1. El código binario resultante (F2F1F0) indica la columna errónea, si el valor es 000 es que no hay error. Si hay error debe cambiarse el valor del bit de la columna afectada.

a) Calcular el código de Hamming para los tres bits M0M1M2. Comprobar que la DH entre las palabras es al menos 3.

b) Determinar si hay algún error en los siguientes datos recibidos en formato (P0P1M0P2M1M2) y decodificar el valor correcto de (M0M1M2) .

(101110), (111010), (001110)

- Existen códigos binarios en los que los elementos del alfabeto se codifican con distintos números de bits. Estos códigos son útiles cuando cada elemento tiene distinta probabilidad de aparición. El código de Huffman se genera mediante el siguiente procedimiento:

Paso 1. Ordenar los elementos por probabilidad de aparición (tanto por 1) de mayor a menor.

Paso 2. Tomar los dos últimos elementos y asociar al último un bit a 0 y al penúltimo un bit a 1.

Paso 3. Sustituir los dos elementos del paso 3 por un nuevo elemento (X1, X2, ...) cuya probabilidad de aparición sea la suma de la probabilidad de esos dos elementos. Mientras que haya más de 1 elemento volver al paso 1, si no ir al paso 4.

Paso 4. Generar la codificación binaria para cada elemento del alfabeto, asociándole de forma ordenada los bits que se hayan generado desde el elemento de probabilidad 1 (bit más significativo) hasta el elemento inicial (bit menos significativo), pasando por los Xi intermedios generados entre ellos (bits intermedios).

a) Obtener las codificaciones binarias mediante el código de Huffman para los elementos del alfabeto formado por A (0.3), B (0.25), C (0.20), D (0.15) y E (0.1). Entre paréntesis se incluye la probabilidad de aparición Pr de cada elemento.

b) Calcular el promedio P de bits por dato del código de Huffman.

$$P = \sum_i \text{Pr}(i) \cdot (n^\circ \text{ bits})_i$$