Informática Básica: Representación de la información

Departamento de Electrónica y Sistemas

Otoño 2010

Contents

Definición

Sistema de numeración

Es un conjunto de símbolos y reglas de generación que permiten construir todos los números válidos en el sistema. Un sistema de numeración puede representarse como $N = (S, R)$

- `N` *es el sistema de numeración considerado (p.ej. decimal, binario, etc.).*
- `S`, *es el conjunto de símbolos permitidos en el sistema. (p.ej. en decimal S=0,1,...9, en binario S=0,1)*
- `R` *son las reglas que nos indican qué números son válidos en el sistema, y cuáles no.*

Clasificación de los sistemas de numeración

Tipos de sistemas de numeración:

- No posicionales: Los más primitivos. Tienen que ver con la coordinalidad entre conjuntos. Ejemplos: númeración maya, numeración egipcia.
- Semi-posicionales: Ejemplo: números romanos (IV, V, VI, VII).
- Posicionales: La base indica la cardinalidad del conjunto $\mathbb{S}$. Contamos usando todos los símbolos del conjunto $\mathbb{S}$, cuando se agotan se consideran todas las posibles combinaciones de 2 símbolos, luego de 3 ...

Sistemas posicionales: Teorema fundamental de la numeración (aplicable a sistemas posicionales)

- *N, número válido en el sistema de numeración.*
- *b base del sistema de numeración. Número de símbolos permitidos en el sistema.*
- *d_i, un símbolo cualquiera de los permitidos en el sistema de numeración.*
- *n número de dígitos de la parte entera.*
- *, coma fraccionaria. Símbolo utilizado para separar la parte entera de un número de su parte fraccionaria.*
- *k número de dígitos de la parte fraccionaria.*

$$N = \begin{cases} \langle d_{(n-1)} \ldots d_1 d_0, d_{-1} \ldots d_{-k} \rangle = \sum_{i=-k}^{n} d_i b^i \\ N = d_n b^n + \ldots + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + \ldots + d_{-k} b^{-k} \end{cases}$$

Ejemplos

Ejemplos de aplicación del Teorema fundamental de la numeración

- Binario : b=2, $\mathtt{S}= 0, 1$
- Decimal: b=10, $\mathtt{S}= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
- Hexadecimal: b=16,
  $\mathtt{S}= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$ donde
  $A = 10, B = 11, C = 12, D = 13, E = 14$ y $F = 15$

Ejemplos

Ejemplos de aplicación del Teorema fundamental de la numeración

- Binario : b=2, S= 0, 1

$$N = d_n \dots d_1 d_0, d_{-1} \dots d_{-k} =$$

$$d_n \cdot 2^n + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0, + d_{-1} \cdot 2^{-1} + \dots + d_{-k} \cdot 2^{-k} =$$

$$111_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4 + 2 + 1 = 7_{10}$$

- Decimal: b=10, S= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Hexadecimal: b=16,
  S= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F donde
  $A = 10, B = 11, C = 12, D = 13, E = 14$ y $F = 15$

Ejemplos

Ejemplos de aplicación del Teorema fundamental de la numeración

- Binario : b=2, S= 0, 1
- Decimal: b=10, S= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

$$N = d_n \ldots d_1 d_0, d_{-1} \ldots d_{-k} =$$

$$d_n \cdot 10^n + \ldots + d_1 \cdot 10^1 + d_0 \cdot 10^0, + d_{-1} \cdot 10^{-1} + \ldots + d_{-k} \cdot 10^{-k} =$$

$$1492, 36 = 1 \cdot 10^3 + 4 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0, +3 \cdot 10^{-1} + 6 \cdot 10^{-2}$$

- Hexadecimal: b=16,
  S= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F donde
  $A = 10, B = 11, C = 12, D = 13, E = 14$ y $F = 15$

Ejemplos

Ejemplos de aplicación del Teorema fundamental de la numeración

- Binario : b=2, $\mathtt{S}= 0, 1$
- Decimal: b=10, $\mathtt{S}= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
- Hexadecimal: b=16,
  $\mathtt{S}= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$ donde
  $A = 10, B = 11, C = 12, D = 13, E = 14$ y $F = 15$

$$N = d_n \dots d_1 d_0, d_{-1} \dots d_{-k} =$$

$$d_n \cdot 16^n + \dots + d_1 \cdot 16^1 + d_0 \cdot 16^0, + d_{-1} \cdot 16^{-1} + \dots + d_{-k} \cdot 16^{-k} =$$

$$3E0,A_{16} = 3 \cdot 16^2 + E \cdot 16^1 + 0 \cdot 16^0 + A \cdot 16^{-1} =$$
$$3 \cdot 256 + 14 \cdot 16 + 0 \cdot 1 + 10 \cdot 0,0625 = 992,625_{10}$$

Conversión entre sistemas numéricos

- Transformación de un número M de decimal a base p
- Transformación de un número $i = d_n d_{n-1} ... d_1 d_0$ base p a decimal
- Transformación de base p a base q

Conversión entre sistemas numéricos

- Transformación de un número M de decimal a base p

Sea el entero $i = 0$:

1. Se divide el número M entre p.
2. La división del punto 1 genera un resto que llamaremos a_i y un cociente C_i
3. Si $p > 10$ y la magnitud del resto es $a_i \geq 10$, este debe convertirse al correspondiente dígito en base p)
4. Si el cociente C_i es distinto de cero, se hace M$=C_i$, se incrementa i y se repite desde el punto 1.
5. Si el cociente C_i es igual a cero, el proceso finaliza. El número en base p esta formado por el conjunto de los bits a_i donde el subíndice i indica la posición que ocupa cada bit en el número binario, esto es, el primer resto que se obtuvo (para i=0, a_0) es el bit menos significativo y, el último, el más significativo.

Conversión entre sistemas numéricos

- Transformación de un número M de decimal a base p
- Transformación de un número $i = d_n d_{n-1} ... d_1 d_0$ base p a decimal

Se usa la siguiente fórmula $N = d_n p^n + \ldots + d_1 p^1 + d_0 p^0$

- Transformación de base p a base q

Conversión entre sistemas numéricos

- Transformación de un número M de decimal a base p
- Transformación de un número $i = d_n d_{n-1} ... d_1 d_0$ base p a decimal
- Transformación de base p a base q

1. Se transforma previamente M a base decimal. Llamemos R al número resultado de dicha transformación.
2. Se transforma R, expresado en decimal, a base q.

Conversión entre sistemas numéricos

- Transformación de un número M de decimal a base p
- Transformación de un número $i = d_n d_{n-1} ... d_1 d_0$ base p a decimal
- Transformación de base p a base q
  CASO ESPECIAL: Transformación de base p a base q cuando p es potencia entera de q o viceversa

Códigos binarios

- Códigos binarios para la representación de dígitos decimales
  - BCD: Código de 4 bits utilizado para la codificación de los 10 dígitos decimales.
  - Código Exceso a $2^{n-1}-1$
  - Código 2-de-5
  - Código 7-segmentos
- Código Gray: Código en el que dos elementos consecutivos solo difieren en un bit
- Códigos detectores de errores: El método más común es el uso de un bit de paridad

Códigos binarios

- Códigos binarios para la representación de dígitos decimales
  - BCD: Código de 4 bits utilizado para la codificación de los 10 dígitos decimales.

Número decimal	Código BCD
0	0000
1	0001
2	0010
3	0011
...	...
8	1000
9	1001

Ejemplo: $12_{10} = (0001\ 0010)_{BCD}$

  - Código Exceso a $2^{n-1} - 1$
  - Código 2-de-5
  - Código 7-segmentos
- Código Gray: Código en el que dos elementos consecutivos solo difieren en un bit
- Códigos detectores de errores: El método más común es el uso de un bit de paridad

Códigos binarios

- Códigos binarios para la representación de dígitos decimales
  - BCD: Código de 4 bits utilizado para la codificación de los 10 dígitos decimales.
  - Código Exceso a $2^{n-1} - 1$

> - n es el número de bits del número representado
> - El resultado de sumar el exceso debe ser un entero positivo.

  - Código 2-de-5
  - Código 7-segmentos
- Código Gray: Código en el que dos elementos consecutivos solo difieren en un bit
- Códigos detectores de errores: El método más común es el uso de un bit de paridad

Códigos binarios

- Códigos binarios para la representación de dígitos decimales
  - BCD: Código de 4 bits utilizado para la codificación de los 10 dígitos decimales.
  - Código Exceso a $2^{n-1} - 1$
  - Código 2-de-5
  - Código 7-segmentos
- Código Gray: Código en el que dos elementos consecutivos solo difieren en un bit

Número decimal	Código Gray (3 bits)
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

- Códigos detectores de errores: El método más común es el uso de un bit de paridad

Códigos binarios

- Códigos binarios para la representación de dígitos decimales
  - BCD: Código de 4 bits utilizado para la codificación de los 10 dígitos decimales.
  - Código Exceso a $2^{n-1} - 1$
  - Código 2-de-5
  - Código 7-segmentos
- Código Gray: Código en el que dos elementos consecutivos solo difieren en un bit
- Códigos detectores de errores: El método más común es el uso de un bit de paridad

- Paridad par: el bit de paridad se pone a uno si el número de 1's es impar (lo que hace que el número de 1's sea par)
- Paridad impar: el bit de paridad se pone a uno si el número de 1's es par

Datos (3 bits)	Paridad par	Paridad impar
000	0000	1000
111	1111	0111
101	0101	1101

Representación binaria de números enteros con signo

Posibles representaciones:

- Signo-magnitud
- Complemento a 1
- Complemento a 2

Representación binaria de números enteros con signo

Posibles representaciones:

- Signo-magnitud

S	M_3	M_2	M_1	M_0

- S: Bit de signo (0: positivo, 1: negativo)
- $M_3 \ldots M_0$: Magnitud

Ejemplos: $(0001)_{SM} \rightarrow (+1)_{10}$, $(1011)_{SM} \rightarrow (\text{-}3)_{10}$

- Complemento a 1
- Complemento a 2

Representación binaria de números enteros con signo

Posibles representaciones:

- Signo-magnitud
- Complemento a 1

- Números positivos: como en S-M
- Números negativos: se invierten todos los bits del número positivo en S-M

	Ca1
+3	011
+2	010
+1	001
+0	000
-0	111
-1	110
-2	101
-3	100

Hay 2 representaciones para el 0.

- Complemento a 2

Representación binaria de números enteros con signo

Posibles representaciones:

- Signo-magnitud
- Complemento a 1
- Complemento a 2

- Números positivos: como en S-M
- Números negativos: se invierten todos los bits del número positivo en S-M y se le suma 1

	Ca1
+3	011
+2	010
+1	001
+0	000
-1	111
-2	110
-3	101
-4	100

Notación binaria de números reales

Posibles representaciones:

- Representación en coma fija
- Representación en coma flotante

Notación binaria de números reales

Posibles representaciones:

- Representación en coma fija

  - El número de bits de la parte fraccionaria es fijo
  - Solo unas pocas fracciones pueden ser representadas → problemas de precisión

$$\boxed{\quad\text{PE}\quad} \; , \; \boxed{\quad\text{PF}\quad}$$

  - PE: Parte entera
  - PF: Parte fraccionaria

  Ejemplos: $(101{,}1)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \rightarrow (5{,}5)_{10}$

- Representación en coma flotante

Notación binaria de números reales

Posibles representaciones:

- Representación en coma fija
- Representación en coma flotante

- Premisa: cualquier número real N en base b puede ser expresado como $N = m \times b^e$
- Ejemplo: $(2,45)_{10} = 245 \times 10^{-2}$. Donde la base $b = 10$, la mantisa $m = 245$ y el exponente $e = -2$
- Para representar un número real binario solo necesitamos almacenar (en formato S-M) la mantisa y el exponente, la base se da por conocida.
- Problema: existen múltiples representaciones para un mismo número. Solución?
  - Normalización por mantisa entera: la mantisa es entera

S	M(1)	M	,	S	E

  - Normalización por mantisa fraccionaria: la mantisa es fraccionaria

S	,	M(1)	M	S	E

Notación binaria de números reales

Posibles representaciones:

- Representación en coma fija
- Representación en coma flotante

Formato IEEE-754

S	E	M

  - S: Signo
  - E: Exponente desplazado (n bits)
  - M: Mantisa
  - $N = (-1)^S \times 2^{E-(2^{n-1}-1)} \times 1.M$
  - Ejemplo: 0 10000010 01101100...0 $\rightarrow (-1)^0 \times 2^{130-127} \times 1{,}01101\ldots$

Notación binaria de números reales

Formato IEEE 754

- Precisión simple: 32 bits

Signo (1bit)	Exponente (8 bits)	Mantisa (23 bits)

- Precisión doble: 64 bits

Signo (1bit)	Exponente (11 bits)	Mantisa (52 bits)

Notación binaria de números reales
Formato IEEE 754

EJEMPLO: Transformar un número decimal a formato IEEE 754 (simple precisión)

Transformar el número $(101110,01010111010000111110000111110001001 1)_2$ a formato IEEE 754 (simple precisión)

1. Se normaliza el número (dejando un único bit a la izquierda de la coma.
   $(1,0111001010111010000111110000111110001001 1)_2 \times 2^5$
2. El exponente en exceso a $(2^{n-1} - 1) = 2^7 - 1 = 127$ es $5 + 127 = 132 = (10000100)_2$
3. De la mantisa se cogen los 23 bits más significativos 1,01110010101110100001111
4. Cuando la mantisa se normaliza, situando la coma decimal a la derecha del bit más significativo, dicho bit siempre vale 1
   - Podemos coger un bit más en la mantisa prescindiendo del 1. $\rightarrow$ 01110010101110100001111
5. El número obtenido es

Signo (1bit)	Exponente (8 bits)	Mantisa (23 bits)
0	10000100	01110010101110100001111

Notación binaria de números reales

Formato IEEE 754

EJEMPLO: Transformar un número en formato IEEE 754 (simple precisión) a decimal

Transformar el número $(3E400000)_{\mathrm{hex}}$ el formato IEEE 754 (simple precisión) a decimal

1. Transformar el número a binario y descomponerlo en los campos que lo componen

Signo (1bit)	Exponente (8 bits)	Mantisa (23 bits)
0	01111100	10000000000000000000000

2. Transformar el componente a base 10
$(01111100)_2 - (2^7 - 1) = (124)_10 - (127)_10 = -3$

3. Escribir el número en notación científica (añadiendo 1, a la mantisa). También despreciamos los 0's a la derecha
$1{,}1 \times 2^{-3}$

4. Expresar el número en base 10
$1{,}1 \times 2^{-3} = (0,0011)_2 = (2^{-3} + 2^{-4})_{10} = 0,125 + 0,00625 = 0,1875$

Código ASCII

ASCII (American Standard Code for Information Interchange): es un esquema binario de codificación de caracteres basado en el orden del alfabeto en inglés.

Binario	Caracter
...	...
100 0001	A
100 0010	B
100 0011	C
100 0100	D
100 0101	E
...	...
110 0001	a
110 0010	b
110 0011	c
110 0100	d
...	...

Otros esquemas: ISO-8859

- ISO-8859: es un esquema de codificación de caracteres. Utiliza 8 bits para codificar cada carácter.
  - El esquema ASCII permitía la codificación de textos en inglés. El esquema ISO-8859 mejor el soporte para otros idiomas, se definen 15 partes. Por ejemplo: ISO-8859-1 permite la codificación de textos que usen el alfabeto latino.

Otros esquemas:UTF-8

- UTF-8: es un esquema de codificación de caracteres basado en símbolos de longitud variable (1-4 bytes por caracter). Los bits más significativos del primer byte de una secuencia multi-byte determinan la longitud de la secuencia. (110=dos bytes, 1110=tres bytes...). La distribución de caracteres es la siguiente:
  - Caracteres codificados con un byte: Los incluidos en US-ASCII, un total de 128 caracteres.
  - Caracteres codificados con dos bytes: Caracteres romances más signos diacríticos, y los alfabetos griego, cirílico, copto, armenio, hebreo, árabe, siríaco y Thaana.
  - Caracteres codificados con tres bytes: Grupo CJK: Chino, japonés y coreano.
  - Caracteres codificados con cuatro bytes: Símbolos matemáticos y alfabetos clásicos para uso principalmente académico.