

Redes de datos

Numeración binaria (repaso)

Tecnólogo en Telecomunicaciones

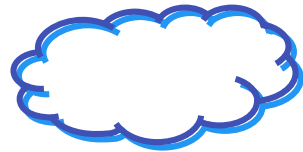
Centro Universitario Regional del Este – Sede Rocha

Facultad de Ingeniería

Universidad de la República

Numeración decimal

- El sistema de numeración más usual para nosotros es es sistema decimal
- Se llama decimal porque utiliza 10 símbolos
- Símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Con esos símbolos podemos contar



0



1



2



3

Numeración decimal

- Seguimos así hasta que se nos terminan los símbolos disponibles



Numeración decimal

- Cuando se nos terminan los símbolos le damos un valor posicional
- Si está en la primer columna de la derecha, son unidades ($10^0=1$)
- En la segunda, son decenas (multiplicamos su valor por $10=10^1$)
- En la tercera, centenas (multiplicamos por $100=10^2$)

millares	centenas	decenas	unidades
x 1000	x 100	x 10	x 1
10^3	10^2	10^1	10^0

- Entonces el número 8215 lo podemos expresar como:
 - $8 \times 10^3 + 2 \times 10^2 + 1 \times 10^1 + 5 \times 10^0$
 - $8000 + 200 + 10 + 5$

Numeración binaria

- Mismo concepto, pero solo contamos con dos símbolos: 0 y 1
- Entonces para contar más usamos valor posicional
- Ahora el valor posicional es de 2^n
- Contamos.....

cuenta	2^2	2^1	2^0
0			0
1			1
2		1	0
3		1	1
4	1	0	0
...			

- Entonces el decimal 4 en binario es 100 = $1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4 + 0 + 0$

Ejemplos

- Ejemplos:
 - 00001 = 1
 - 00010 = 2
 - 00101 = 5
 - 01000 = 8
 - 01111 = 15
 - 10000 = 16
- Con 4 bits, puedo contar desde
 - 0000 a 1111
 - o sea del 0 a 15
- Con 8 bits, puedo contar desde
 - 00000000 a 11111111
 - O sea de 0 a 255
- Con 16 bits, puedo contar de 0 a 65535
- En general con N bits tengo 2^N valores posibles, del 0 al $2^N - 1$

Operaciones con bits

- Suma (resta)
 - $0+0 = 0$
 - $0+1 = 1$
 - $1+0 = 1$
 - $1+1 = 10$ (0 y me llevo el 1)
 - Si la suma es sin acarreo, entonces $1+1=0$
- Multiplicación
 - $0 \times 0 = 0$
 - $0 \times 1 = 0$
 - $1 \times 0 = 0$
 - $1 \times 1 = 1$

Operaciones lógicas

- AND (“y”) (equivalente a multiplicación, resultado 1 solo si ambos en 1)
 - $0 \text{ AND } 0 = 0$
 - $0 \text{ AND } 1 = 0$
 - $1 \text{ AND } 0 = 0$
 - $1 \text{ AND } 1 = 1$
- OR (“o”) (uno u otro, resultado 0 solo si ambos son 0)
 - $0 \text{ OR } 0 = 0$
 - $0 \text{ OR } 1 = 1$
 - $1 \text{ OR } 0 = 1$
 - $1 \text{ OR } 1 = 1$
- XOR (OR exclusivo) (uno u otro pero no los dos, resultado 0 cuando son iguales, resultado 1 si son diferentes, suma sin acarreo)
 - $0 \text{ XOR } 0 = 0$
 - $0 \text{ XOR } 1 = 1$
 - $1 \text{ XOR } 0 = 1$
 - $1 \text{ XOR } 1 = 0$

Numeración hexadecimal

- Lo mismo pero en base 16 (16 símbolos)
- Se usan los símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
- Se usa muchas veces para escribir de forma más corta cadenas de bits
- Ejemplos:
 - 11111111 en binario = 255 decimal = FF en hexadecimal
 - Podemos expresar 16 bits en 1 se expresan como FFFF en hexa