Clase 2

Práctico de Diseño Lógico

Clase 2 – temas

- Representación de números negativos
  - Complemento a 2
- Representación de números fraccionarios
  - Punto fijo
  - Punto flotante
- Códigos
  - Distancia
  - Corrección y detección de errores

Representación de números negativos

- Magnitud y signo
- Desplazamiento
- Complemento a 1
- **Complemento a 2**

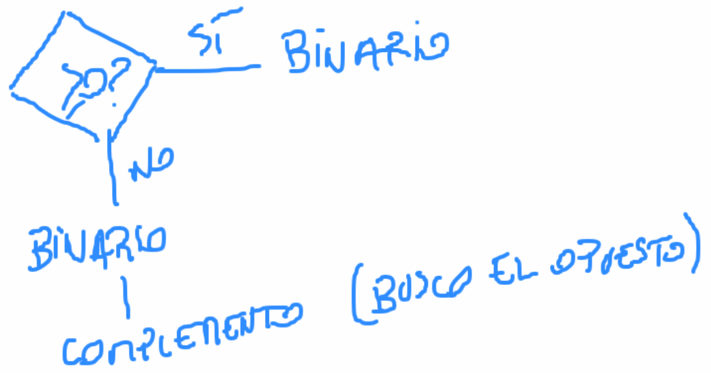
Complemento a 2 (repaso)

n bits $\longrightarrow 2^n$

$\frac{2^n}{2} = 2^{n-1}$

- Definición de complemento a 2: $-A(2c)= 2^n - A(2c)$
- Pesos de complemento a 2: -64, 32, 16, 8, 4, 2, 1
- Regla para calcular el complemento de un número en C2: **A(2c)= !A +1**
- Rango de representación: **[$-2^{(n-1)}$; $2^{(n-1)}-1$]** (n cantidad de bits)
- Los positivos coinciden con la representación de los binarios.
- El complemento a 2 de un número es su opuesto: si estaba representando un número negativo en C2, al complementarlo obtengo un positivo.

número $\longrightarrow$ C2



Complemento a 2 (repaso)

$1 \oplus 0 = 1$

OV

$$
\begin{array}{r}
{\scriptstyle 0\,1} \\
0\,1\,0\,0 \\
+\ 0\,1\,0\,1 \\
\hline
1\,0\,0\,1
\end{array}
$$

- Definición de complemento a 2: $-A(2c)= 2^n - A(2c)$
- Pesos de complemento a 2: -64, 32, 16, 8, 4, 2, 1
- Regla para calcular el complemento de un número en C2: **A(2c)= !A +1**
- Rango de representación: **[$-2^{(n-1)}$; $2^{(n-1)}-1$]** (n cantidad de bits)
- Los positivos coinciden con la representación de los binarios.
- El complemento a 2 de un número es su opuesto: si estaba representando un número negativo en C2, al complementarlo obtengo un positivo.
- El resultado de una op con dos n° en C2 es válido si no hay OVERFLOW.
  - De dos operandos de igual signo obtengo uno de signo opuesto
  - **Cin xor Cout = OV**

Ejercicio 8

Ejercicio 8. Se desean representar en complemento a 2 los siguientes números: 1, -5, 29, -47 y 64.

a) ¿Cuántos bits son necesarios?
b) Halle la representación de dichos números en complemento a 2, con la cantidad de bits hallada en a).

Rango de representación: **$[\ -2^{(n-1)}\ ;\ 2^{(n-1)}-1\ ]$**
(n cantidad de bits)

- Parte a)
  - Para representar **1** necesito n =
  - Para representar **-5** necesito n =
  - Para representar **29** necesito n =
  - Para representar **-47** necesito n =
  - Para representar **64** necesito n =

  Entonces para representarlos a todos necesito n = bits

Rango de representación Complemento a 2

N	Desde $-2^{(n-1)}$	Hasta $2^{(n-1)}-1$
1	-1	0
2	-2	1
3	-4	3
4	-8	7
5	-16	15
6	-32	31
7	-64	63
8	-128	127

Rango de representación: **$[\ -2^{(n-1)}\ ;\ 2^{(n-1)}-1\ ]$**
(n cantidad de bits)

Ejercicio 8

Ejercicio 8. Se desean representar en complemento a 2 los siguientes números: 1, -5, 29, -47 y 64.

a) ¿Cuántos bits son necesarios?
b) Halle la representación de dichos números en complemento a 2, con la cantidad de bits hallada en a).

Los positivos coinciden con la representación de los binarios.
A(2c)= !A +1

- Parte b)
  - Representación en complemento a 2 de 8 bits de -47 d
    - 47 d = 32 + 15
    - 47 d = 0010 1111
    - -47 d = 1101 0001
  - Representación en complemento a 2 de 8 bits de 64 d = 0100 0000

Representación de números fraccionarios en Punto Fijo

$$\underbrace{b_n.2^n + ... + b_1.2^1 + b_0.2^0}_{[I]} \underset{,}{+} \underbrace{b_{-1}.2^{-1} + ... + b_{-m}.2^{-m}}_{[F]}$$

====> $I = b_n \ldots\ldots\ldots\ldots b_1 b_0$

$F = 0, b_{-1} b_{-2} b_{-3} \ldots\ldots\ldots\ldots\ldots\ldots b_{-m}$

La parte entera se representa en Magnitud y Signo

Pasaje de número decimal a punto fijo

Ejercicio 9. Representar los siguientes números decimales en la notación de punto fijo (8 bits para la parte entera y 8 bits para la parte fraccionaria). Determine el error introducido en la conversión.

a) 5,1
b) 72,0625
c) 49,53
d) 0,28125

a) 5,1

Parte entera: I = 0000 0101

Parte fraccionaria:

0,1 x 2 = 0 + 0,2
0,2 x 2 = 0 + 0,4
0,4 x 2 = 0 + 0,8
0,8 x 2 = 1 + 0,6
0,6 x 2 = 1 + 0,2

00011 0011 0011 0011 0011 0011 0011… como solo tengo 8 bits: F = 0001 1001

Error: $0,6 \times 2^{-8}$

Pasaje de número decimal a punto fijo

Ejercicio 9. Representar los siguientes números decimales en la notación de punto fijo (8 bits para la parte entera y 8 bits para la parte fraccionaria). Determine el error introducido en la conversión.

a) 5,1
b) 72,0625
c) 49,53
d) 0,28125

b) 72,0625
Parte entera: 72 = 64 + 8 => I = 0100 1000
Parte fraccionaria:
0,0625 x 2 = 0 + 0,125
0,125 x 2 = 0 + 0,25
0,25 x 2 = 0 + 0,5
0,5 x 2 = 1 + 0

F = 0001 0000
Error: 0

O por método de los pesos:

½	¼	1/8	1/16
0,5	0,25	0,125	0,0625

Representación de números fraccionarios en Punto Flotante

- Ejemplo base 10: 51680.012
  - $51680.012 = 5.1680012 \times 10^4$
- Ejemplo base 2 : 100110.11
  - $100110.11 = 1.0011011 \times 2^5$
- El primer bit en el caso de base 2 es siempre uno ¿qué pasa con el 0? Por convención, el 0 se representa con todos los bits de s, f y e en 0.
- Simple precisión: [s(1)][e(8)][f(23)] con **$N=(-1)^s \times 1,f \times 2^{(e-127)}$**

Ejemplo - ejercicio 9-b

- I,F= 0100 1000, 0001 0000
- I,F= 0100 1000, 0001 0000 = 1,00 1000 0001 x 2^6
- s=0
- 6 = e – 127 => e=133 = 85h
- f=00100000010000000000000
- error = 0

$N=(-1)^s \times 1,f \times 2^{(e-127)}$

Ejercicio 10

Ejercicio 10.

i) Representar los siguientes números decimales en la notación de punto flotante de simple precisión, según el estándar IEEE 754.

a) 10
b) 1/256
c) 128 Mega
d) 128 Mega +1
e) 1,5
f) 4,25

ii) Determine el mayor y el menor número positivo representables en esta notación.

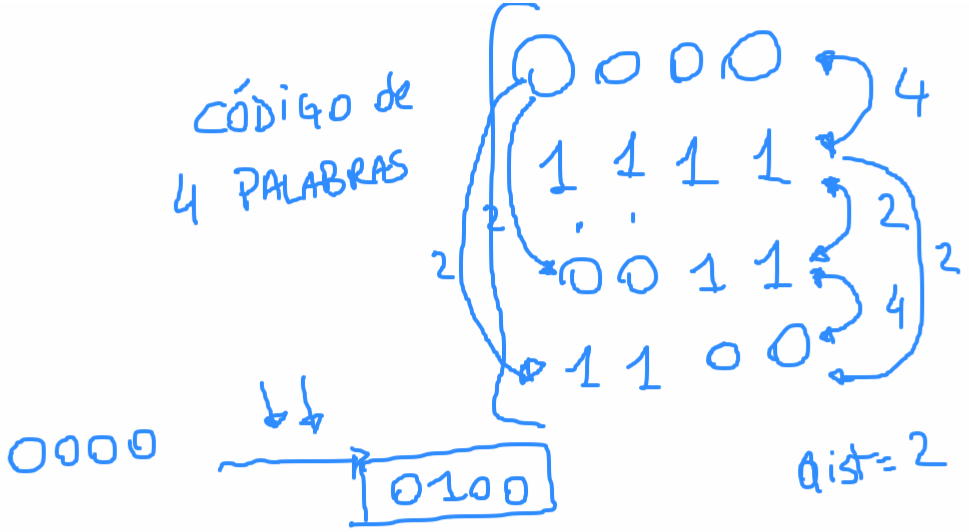
- Partes c) y d)
- Kilo = 2^{10}
- Mega = 2^{20}
- 128 Mega = $128 \times 2^{20} = 2^{7} \times 2^{20} = 2^{27} = 1 \times 2^{27}$
- En simple precisión: s = 0, f = 0000…0000 , 27=e-127 => e = 154 = 9Ah
- Error: 0
- 128 Mega + 1 =
- $(1 \times 2^{27} + 1) \times 2^{-27} \times 2^{27} = 1{,}000000000000000000000000001 \times 2^{27}$
- s = 0, f = 0000…0000 , e = 1001 1010 , ¿error?
- Error = 1

$N=(-1)^{s} \times 1{,}f \times 2^{(e-127)}$

Códigos para detección y corrección de errores.

- Distancia entre palabras: cantidad de bits que difieren entre sí.
- Distancia del código: mínima distancia entre palabras válidas.
- Error simple: cambia un bit.
- Error doble: cambian dos bits.
- Error triple: cambian tres bits.
- Etc.

Ejemplo



Ejercicio 2.6

Ejercicio 6. Se consideran los cuatro códigos siguientes:

código A	código B	código C	código D
0001	código Gray	01011	000000
0010	de 3 bits	01100	001111
0100		10010	110011
1000		10101	

a) ¿Cuál de las siguientes propiedades son satisfechas por cada uno de ellos?

- Detectar errores simples.
- Detectar errores dobles.
- Detectar errores triples.
- Corregir errores simples.
- Corregir errores dobles.
- Corregir errores simples y detectar errores dobles.

b) ¿Cuántas palabras pueden agregarse al código A sin cambiar sus propiedades de detección y corrección de errores?. Dé un posible conjunto de dichas palabras. ¿Es este conjunto único?

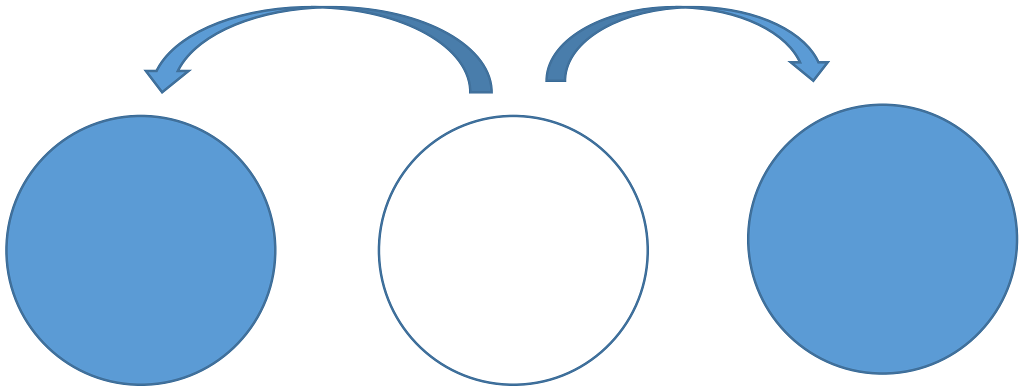
Código B

000
001
011
010
110
111
101
100

dist = 1

Código A

código A
0001
0010
0100
1000



Puedo detectar errores simples, pero no puedo corregirlos.

Distancia 2

Código C

01011 3
01100

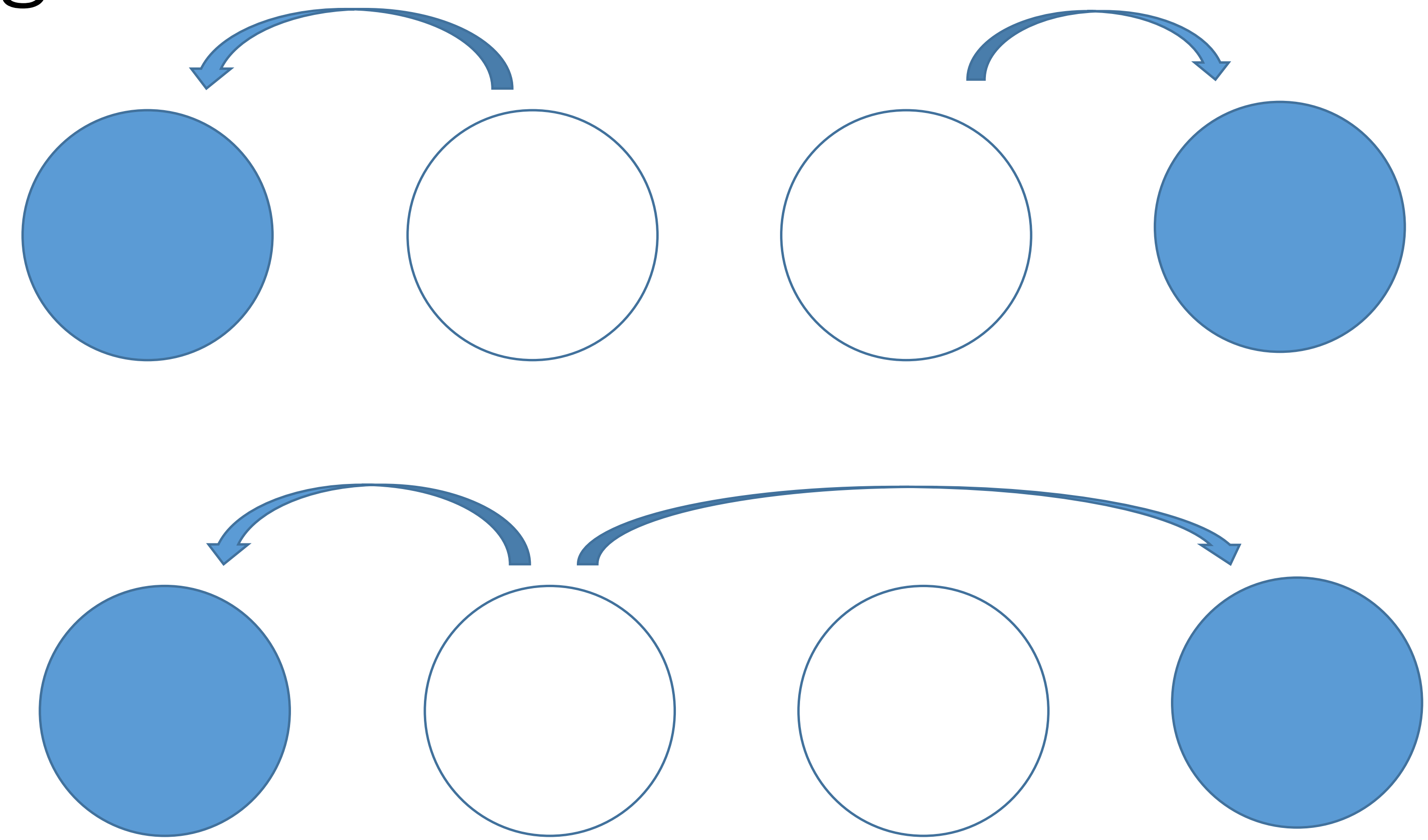
01011 3
10010

01011 4
10101

01100 4
10010

01100 3
10101

10010 3
10101



Opción 1: asumo que solo hay errores simples, entonces puedo corregirlos

Opción 2: asumo que hay errores dobles, entonces no puedo corregirlos porque no sabría cuál fue la palabra original, pero puedo detectarlos

Distancia 3