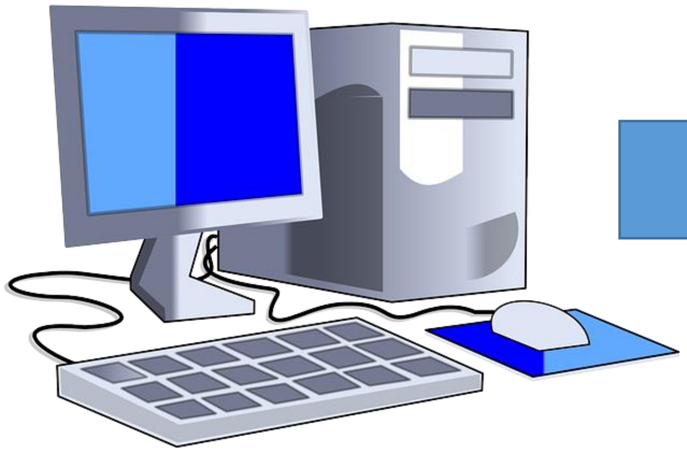
Detección y Corrección de Errores

Dra. Ing. Rocío Andrea Rodríguez

UNLaM– 2020

Ejemplo



0110

0110

0111
0100
0010
1110

¡No lo detecto!

Distancia entre dos elementos de un código:

- Es la cantidad de bits que deben modificarse para transformar una combinación perteneciente al código en otra también perteneciente al código.

- Es la cantidad de bits en la que difieren dichas combinaciones del código

Distancia mínima:

- Es la menor distancia que puede encontrarse entre dos combinaciones del código

¿Que distancia tienen los códigos BCD vistos?

- 8421 → 1
- XS 3 → 1
- Aiken → 1
- Jhonson → 1
- Gray → 1

BCD 8421

	0 -> 0000
	1 -> 0001
XS3	2 -> 0010
0 -	3 -> 0011
1 —	4 -> 0100
2 —	5 -> 0101
3 —	6 -> 0110
4 —	7 -> 0111
5 —	8 -> 1000
6 -	9 -> 1001
⋮	

Bit de Paridad

El agregar un bit de paridad aumenta en 1 la distancia del código

BCD 8421 **con paridad par en los 1 colocado a la izquierda**

0 -> **0** 0 0 0 0
1 -> **1** 0 0 0 1
2 -> **1** 0 0 1 0
3 -> **0** 0 0 1 1
4 -> **1** 0 1 0 0
5 -> **0** 0 1 0 1
6 -> **0** 0 1 1 0
7 -> **1** 0 1 1 1
8 -> **1** 1 0 0 0
9 -> **0** 1 0 0 1

00110 →

- **1**0110
- 0**1**110
- 00**0**10
- 001**0**0
- 0011**1**

¿Cual es la distancia mínima del código?

¡Ya puedo detectar la presencia de un bit erroneo!

BCD 4 → 0100
8421

925 → 1111 0010 1011 Aiken

Ejercicio 1

Se desea enviar en Aiken el número 925 usando paridad impar en los 1 colocados a la derecha de cada cifra codificada

9 2 5

1111**1** 0010**0** 1011**0**

Ejercicio 2

Se desea recibido las siguientes cadenas de bits (utilizando BCD XS3 con paridad impar en los ceros colocados a la izquierda de cada cifra codificada):

a) 110110110010111

894 ← $\underbrace{1\ 1\ 0\ 1\ 1}_{\text{1 cero}}$ $\underbrace{0\ 1\ 1\ 0\ 0}_{\text{3 ceros}}$ $\underbrace{1\ 0\ 1\ 1\ 1}_{\text{1 ceros}}$

b) 00011110111010111011

Error ← $\underbrace{0\ 0\ 0\ 1\ 1}_{\text{3 ceros}}$ $\underbrace{1\ 1\ 0\ 1\ 1}_{\text{1 cero}}$ $\underbrace{1\ 0\ 1\ 0\ 1}_{\text{2 ceros}}$ $\underbrace{1\ 1\ 0\ 1\ 1}$

Se debe volver a transmitir

Detectar y Corregir errores

distancia mínima

$$D > d + k$$

Cantidad de errores a detectar

Cantidad de errores a corregir

$$d \geq k$$

Con un código de distancia 1:
No puedo detectar errores, entonces no puedo corregir

Con un código de distancia 2:
Puedo detectar 1 error pero no corregir

Código detector y corrector de errores de Hamming

- Permite detectar y corregir un único error
- Agrega bits de paridad

Cantidad de bits de paridad

$$d + p < 2^p$$

Cantidad de bits de datos

¿Con **6 bits de paridad** cual es la cantidad máxima de datos que puedo enviar?

Rta: 57 bits de datos

Relación entre los Bits de Datos y los de Paridad

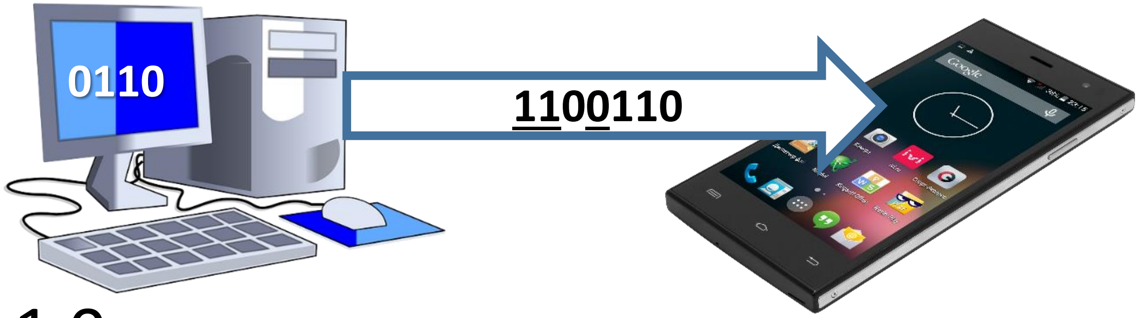
- Para 1 digito en BCD Natural n=4 => p=3 => Un total de 7 bits

Cantidad de Combinaciones

$$\underbrace{d + p}_{\text{Total de bits a enviar}} < 2^p$$

000 $\rightarrow$ No hay error
001 $\rightarrow$ Error en bit 1
010 $\rightarrow$ Error en bit 2
011 $\rightarrow$ Error en bit 3
100 $\rightarrow$ Error en bit 4
101 $\rightarrow$ Error en bit 5
110 $\rightarrow$ Error en bit 6
111 $\rightarrow$ Error en bit 7

Ejemplo - Emisor



Enviar el 6 en BCD Natural: 0 1 1 0

P_1	P_2	X_3	P_4	X_5	X_6	X_7	…
		0		1	1	0	

CARATERÍSTICAS

Paridad PAR EN 1, colocada en posiciones potencias de 2

$$\mathbf{d + p < 2^p}$$

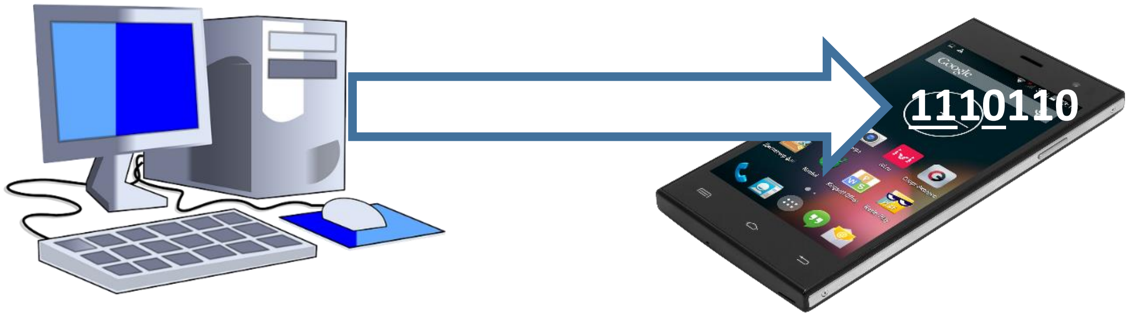
Datos -> X
Paridad -> P

$P_4 = (\quad X_5, \; X_6, \; X_7) = (1,1,0) =$

$P_2 = (X_3, \quad X_6, \; X_7) = (0,1,0) =$

$P_1 = (X_3, \; X_5, \quad X_7) = (0,1,0) =$

Ejemplo - Receptor



		0				
1	1	~~1~~	0	1	1	0
P_1	P_2	X_3	P_4	X_5	X_6	X_7

$\Rightarrow$ 0 1 1 0

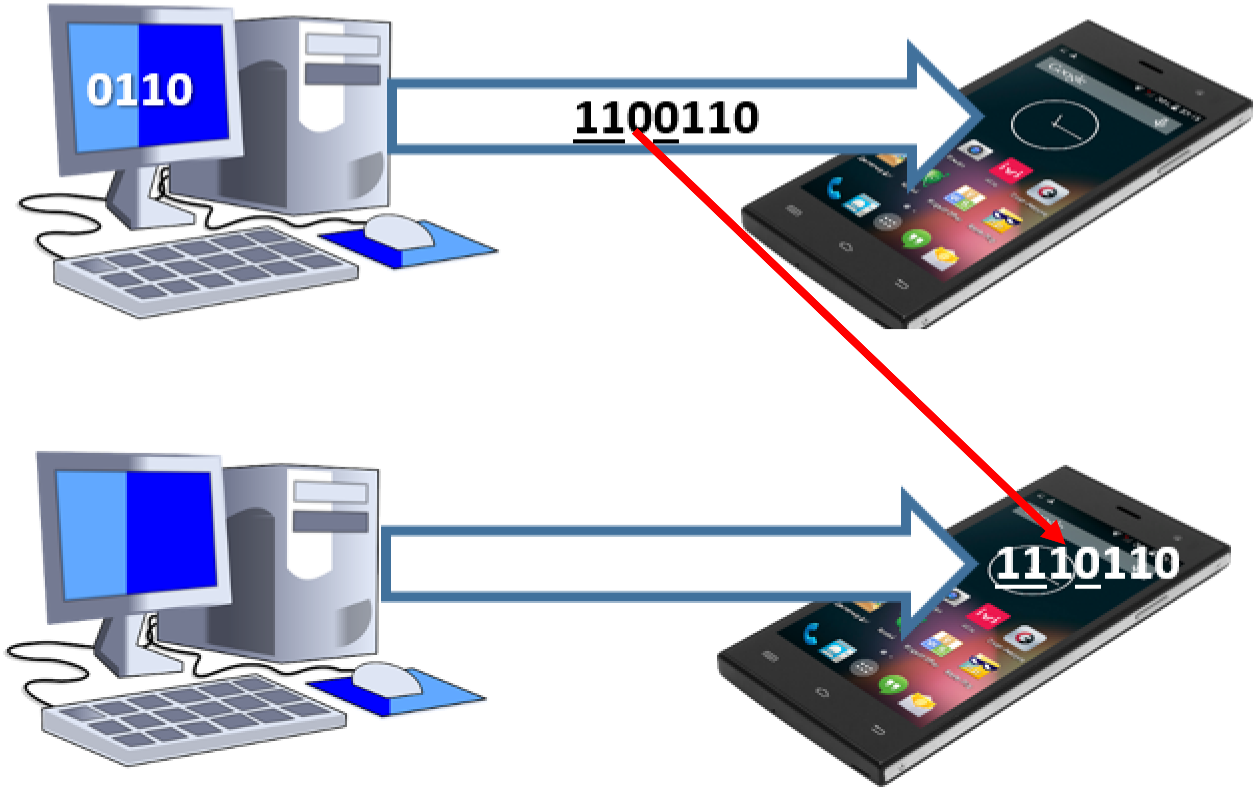
$E_4 = (P_4, \quad X_5, X_6, X_7) = (0,1,1,0) =$ 0 ✓ Cumple paridad par en 1

$E_2 = (P_2, X_3, \quad X_6, X_7) = (1,1,1,0) =$ 1 🚫 Hay un error

$E_1 = (P_1, X_3, X_5, \quad X_7) = (1,1,1,0) =$ 1 🚫 Hay un error

¡ERROR EN BIT 3!

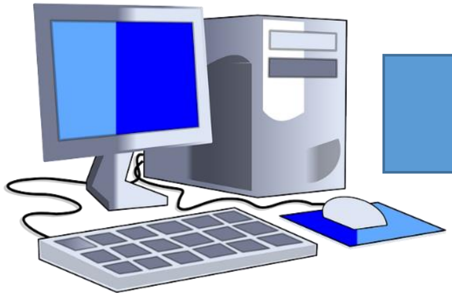
Ejemplo - Emisor



Ejemplo - Receptor

En Resumen

Al trasmitir



Al recibir

Se conoce la **cadena de bits original**

Se calculan los bits de paridad

Se incorporan los bits de paridad a la cadena original →

Se controla si hay errores
(con el bit de paridad incluido todas las ecuaciones)

Si hay error se corrige

Se descartan los bits de paridad

Se obtiene la **cadena de bits original**

Ejercicio

Se ha recibido la palabra de doce bits (código Aiken) 010101111100. Se desea determinar cuál fue la palabra originalmente generada, si la misma se planteó de acuerdo con los criterios de Hamming.

a) 73 b) 97 c) 37 d) 79 e) 14

~~0~~	~~1~~	0	~~1~~	0	1	1	~~1~~	1	1	0	~~0~~ 1
~~P_1~~	~~P_2~~	X_3	~~P_4~~	X_5	X_6	X_7	~~P_8~~	X_9	X_{10}	X_{11}	X_{12}

$\rightarrow \underbrace{0\,0\,1\,1}_{3}\ \underbrace{1\,1\,0\,1}_{7}$

$E_8 = (P_8 \quad X_9 \quad X_{10} \quad X_{11} \quad X_{12}) = (1,1,1,0,0) = 1$

$E_4 = (P_4 \quad X_5 \quad X_6 \quad X_7 \quad X_{12}) = (1,0,1,1,0) = 1$

$E_2 = (P_2 \quad X_3 \quad X_6 \quad X_7 \quad X_{10} \quad X_{11}) = (1,0,1,1,1,0) = 0$

$E_1 = (P_1 \quad X_3 \quad X_5 \quad X_7 \quad X_9 \quad X_{11}) = (0,0,0,1,1,0) = 0$

¡ERROR EN BIT 12!

¡A practicar!
Ya pueden hacer todos que tengan los ejercicios
que aún no hayan hecho de la Unidad 2