IO – Introducció als Ordinadors

GRUP A – Sessió 03-04

Tema 1 - Sistemes de representació numèrica

laSalle
UNIVERSIDAD RAMON LLULL

Conceptes fonamentals

BIT

- En els sistemes digitals, un ‘bit’ és la mínima unitat d’informació, la qual pot prendre només dos valors: ‘0’ o ‘1’.
- Un ‘bit’ és un nombre binari d’un dígit.

Sistema binari

- És un sistema de numeració en el qual els nombres es representen utilitzant 2 tipus de dígits (bi = dos) ‘0’ o ‘1’ → bit.

Concepte de “dígit”

- Estrictament, un “dígit” en un nombre binari, és un bit → **110[0]101$_2$**
- Però és més habitual referir-se a un **dígit decimal** representat en **binari.**
- Per exemple, el dígit “9” decimal en binari és → 1001_2

Conceptes fonamentals

Sobre els nombres binaris

- **No comencen mai per ‘0’,** com tampoc passa amb el cas dels decimals: per exemple, diem que els alumnes de la classe són 35, no 0035. Una altra cosa és que ens obliguin a representar un número amb un cert número de bits.
  - **$001\ 0101_2$ =** No és estrictament incorrecte, però els **zeros** al davant no aporten res.

BYTE

- 8 bits
- Exemple: $\overbrace{0110\ 0101}_2 = 101_{10}$

Emmagatzematge de la informació

- El valor d’un bit es guarda mitjançant un valor de voltatge:

  5 volts → 1 lògic (Vcc)
  0 volts → 0 lògic (GND)

Conceptes fonamentals

Capacitat de codificació

- Quantitat de combinacions (valors) que es poden representar amb un **cert nombre de bits** determinat.

Decimal	Binari (natural)
0 →	000
1 →	001
2 →	010
3 →	011
4 →	100
5 →	101
6 →	110
7 →	111

Exemple:

Amb 3 bits puc codificar 8 valors diferents (=> 8 combinacions possibles).

Bit de més pes
MSB (Most Significant Bit)

Bit de menys pes
LSB (Less Significant Bit)

Sistemes numèrics:

- **Decimal** (Base:10)
- **Binari** (Base:2)
- **Octal** (Base:8)
- **Hexadecimal** (Base: 16)

Conversions de valors d'un sistema a un altre:

- **Binari → Decimal (mètode de potencies) - Decimal a Binari (mètode de divisions)**
- Decimal → Octal (mètode de divisions) - Octal a Decimal (mètode de potencies)
- Octal → Binari (igual que Binari a Octal) concatenant / codificació 0-7.
- Hexadecimal → Decimal (mètode de potencies) – Decimal a Hexadecimal (mètode de divisions)
- Hexadecimal → Binari (igual que Binari a Hexadecimal) concatenant / codificació 0-F.

Sistema Octal (base: 8)

- SN en el qual cada dígit pot prendre **8** valors (=símbols) diferents (0-7).
- Els nombres en octal solen indicar-se amb un subíndex ‘8’ o bé 'o'.
- Exemples: 76302_8 31627_o 5410_8

Conversió: Octal a decimal (8 → 10) ?

Sistema Octal (base: 8)

- SN en el qual cada dígit pot prendre **8** valors (=símbols) diferents (0-7).
- Els nombres en octal solen indicar-se amb un subíndex '8' o bé 'o'.
- Exemples: 76302_8 31627_o 5410_8

Conversió: Octal a decimal (8 → 10) ?

Exemple: 76302_8

Observem que no hi haurà cap '8' ni cap '9'

8^4	8^3	8^2	8^1	8^0	
7·(4096)	**6·(512)**	**3·(64)**	**0·(8)**	**2·(1)**	= 28672 + 3072 + 192 + 2 = **31.938_{10}**

$31627_8 = 3\cdot8^4 + 1\cdot8^3 + 6\cdot8^2 + 2\cdot8^1 + 7\cdot8^0 = 12288 + 512 + 384 + 14 + 7 = 13207_{10}$

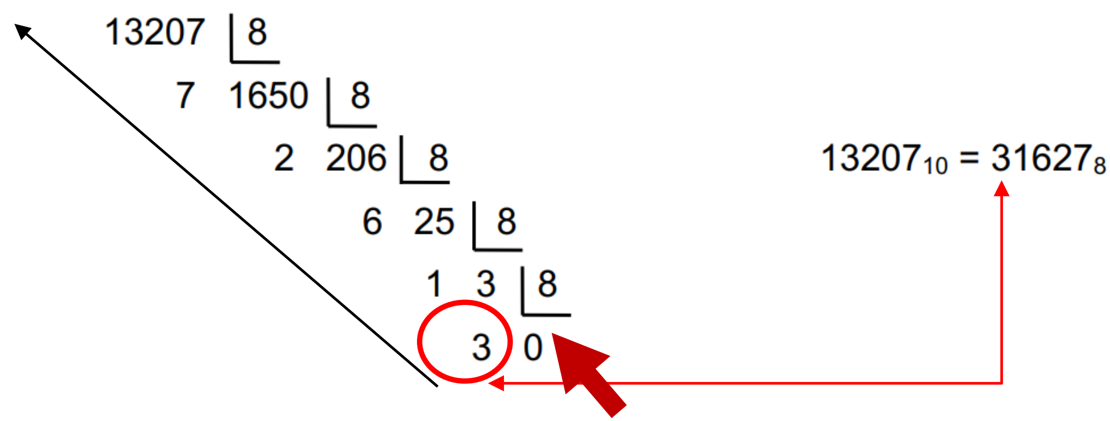
$5410_8 = 5\cdot8^3 + 4\cdot8^2 + 1\cdot8^1 + 0\cdot8^0 = 2560 + 256 + 8 = 2824_{10}$

Sistema Octal (base: 8)

Conversió: Decimal a octal (10 → 8)

- Usem el mètode de les divisions successives, aquest cop dividint entre 8.

Exemple



Sistema Octal (base: 8)

Conversió: Decimal a octal (10 → 8)

- Usem el mètode de les divisions successives, aquest cop dividint entre 8.

Exemple

13207 | 8
7 1650 | 8
2 206 | 8
6 25 | 8
1 3 | 8
3 0

$13207_{10} = 31627_8$

Conversió: Octal a Binari (i viceversa)

- Per representar números del **0-8** només necessitem **3 bits,** per tant es tracta d'anar concatenant. Exemple → **3256_8:**

$3_8 = 011_2$

$2_8 = 010_2$

$5_8 = 101_2$

$6_8 = 110_2$

$3256_8 = 011\ 010\ 101\ 110_2 = 11010101110_2$

Aquest canvi funciona **<u>en les dues direccions</u>**

8 és una potencia de 2^Y → 2^3

Sistema Hexadecimal (base: 16)

- Cada dígit pot prendre **16** valores diferents: { 0 al 9 + A, B, C, D, E, F }
- Les lletres A, B, C, D, E i F equivalen als valors 10, 11, 12, 13, 14 i 15 respectivament.
- La base es pot indicar amb un 16, però és més usual a través de la lletra 'h'.

Conversió: hexadecimal a decimal (16 → 10)

Exemple: **1A340h** → 1A340h → 1(10)340h

16^4	16^3	16^2	16^1	16^0
$1\cdot(16^4)$	$10\cdot(16^3)$	$3\cdot(16^2)$	$4\cdot(16^1)$	$0\cdot(16^0)$

= 65536 + 40960 + 768 + 64 = **107.328_{10}**

Conversió: hexadecimal a binari (16 → 2)

- Per representar números del **0-15** només necessitem 4 **bits,** per tant es tracta d'anar concatenant **(dígit = <u>4 bits</u>)**.

1 0 0 0	0 1 1 0	0 0 0 1
8	6	1

$1000\,0110\,0001_2 = 861_h$

Binario	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

→ el canvi és el mateix en les dues direccions.

Sistema Hexadecimal (base: 16)

Conversió: hexadecimal a binari

- Fan falta **4 bits** per representar un dígit hexadecimal.

Binario	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Binario	Hexadecimal
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- Exemples:

1 0 0 0	0 1 1 0	0 0 0 1
8	6	1

$100001100001_2 = 861_h$

el 1010011010_2 lo convertimos en 001010011010_2

0 0 1 0	1 0 0 1	1 0 1 0
2	9	A

$001010011010_2 = 29A_h$

RESUM

Sistemes numèrics:

- **Decimal** (base: 10)
- **Binari** (base: 2)
- **Octal** (base: 8)
- **Hexadecimal** (base: 16)

Sistemes amb altre bases:

Base 3, Base 4, Base 6...?

Conversions de valors d'un sistema a un altre:

- **Binari a Decimal (mètode de potencies) - Decimal a Binari (mètode de divisions)**
- **Decimal a Octal (mètode de divisions) - Octal a Decimal (mètode de potencies)**
- **Octal a Binari (igual que Binari a Octal) concatenant / codificació 0-7.**
- **Hexadecimal a Decimal (mètode de potencies) – Decimal a Hexadecimal (mètode de divisions)**
- **Hexadecimal a Binari (igual que Binari a Hexadecimal) concatenant / codificació 0-F.**

Transformacions amb números de altres bases diferents a les anteriors

- Usem el **“sistema decimal”** com intermediari.
- Per exemple, per passar un nombre de “base 6” → “base 4”.

***Exemple*:** $\mathbf{3451_6}$

passos: 1) **Base 6 →** **Base 10** (mètode de les potencies)

2) **Base 10 → base 4** (mètode de les divisions)

6^3	6^2	6^1	6^0	
$3\cdot(6^3)$	$4\cdot(6^2)$	$5\cdot(6^1)$	$1\cdot(6^0)$	$= 648 + 144 + 30 + 1 = \mathbf{823_{10}}$

1

823	**4**
3	205 ⌊4
1	51 ⌊4
3	12 ⌊4
0	3 ⌊4
3	0

2

30313_4

Transformacions amb números de altres bases diferents a les anteriors

- Usem el **"sistema decimal"** com intermediari.
- Per exemple, per passar un nombre de "base 6" → "base 4".

Exemple: **3451_6**

passos: 1) **Base 6 →** **Base 10** (mètode de les potencies)

ERRORS TIPICS:

1. Arribar fins al final → el quocient ha de quedar sempre "0".
2. Repassar els càlculs (divisions, potencies, etc.) → $X^0 = 1$.
3. Indicar el sistema numèric amb el subíndex → 111 podria ser base 2 o 10 o ....

823 | **4**
3 205 | 4
1 51 | 4
3 12 | 4
0 3 | 4
3 0

2

30313_4

Codis Binaris
BASE 2,

BASE 3, BASE 4, BASE 5, BASE 6, BASE 7, BASE 8, ...

Codis Binaris

- **Codi Binari Natural (BN)**
- **Codi GRAY**
- **Codi Johnson**
- **Codi BCD**
  - **BCD Natural**
  - ~~**BCD Exces 3**~~
  - ~~**BCD Aiken**~~

El que ja hem vist

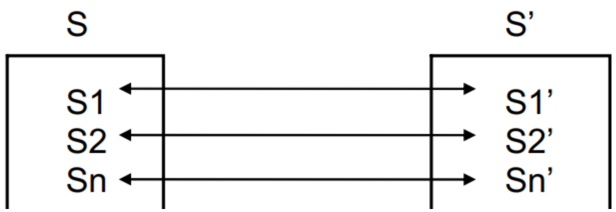
Binari NO Natural (BNN)
(segueixen unes regles)

Codis Binaris

(conceptes bàsics)

Codis binaris: Concepte de codificació

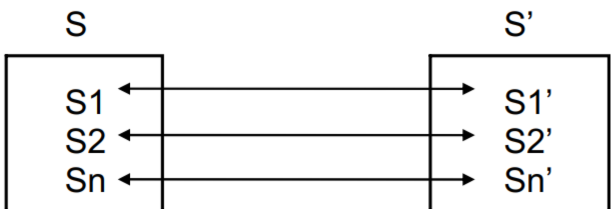
- **Representació** d'una **combinació de símbols (S)** a través d'una altra **combinació de símbols (S')**.
- **Aplicació** entre un **conjunt de dades a representar** i un conjunt de **possibles representacions**



$$\underbrace{1\ 0\ 0\ 0}_{8}\ \underbrace{0\ 1\ 1\ 0}_{6}\ \underbrace{0\ 0\ 0\ 1}_{1}{}_{2} = 861_h$$

Codis binaris: Concepte de codificació

- **Representació** d'una **combinació de símbols (S)** a través d'una altra **combinació de símbols (S')**.
- **Aplicació** entre un **conjunt de dades a representar** i un conjunt de **possibles representacions**



$$\underbrace{1\ 0\ 0\ 0}_{8}\ \underbrace{0\ 1\ 1\ 0}_{6}\ \underbrace{0\ 0\ 0\ 1_2}_{1} = 861_h$$

Codis binaris: Codi Binari Natural (BN) - El que hem vist anteriorment

- El codi BN és un sistema de **numeració ponderat** amb **base fixa.**
- És a dir, un nombre N a representar segueix la fórmula:

$$N = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_i b^i + \dots a_0 b^0 + a_{-1} b^{-1} + \dots a_{-p} b^{-p}$$

$$\begin{array}{cccccc} & 2^3 & 2^2 & 2^1 & 2^0 & \\ 1000_2 \rightarrow & 1 & 0 & 0 & 0 & = 8_{10} \end{array}$$

Codis binaris: Concepte de codificació

Decimal	Binari Natural
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

El codi **Binari Natural (BN)** es correspon amb el sistema de numeració binari ponderat. És a dir, cada posició d'una seqüència de dígits (zeros i uns) té associat un pes. Aquest fet permet extreure una relació directa amb el valor corresponent en el sistema numèric decimal.

← Exemple de codi binari de 4 bits

Un codi BN pot ser de tants bits com es necessitin per a poder representar (codificar) una valor màxim en decimal.

Codis binaris: Concepte de codificació:

Codis: (1) ADJACENTS, (2) CONTINUS, i (3) CICLICS

Decimal	Binari Natural
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
....	....

1. **Combinacions adjacents →** aquelles que difereixen en un sol bit. Per exemple la combinació 1010 i el 1000 són adjacents, 1100 i 1101 són també adjacents, mentre que 1001 i 1100 no ho són.

2. **Codi continu → quan totes les** combinacions corresponent al **codi decimal** son sempre adjacents de forma consecutiva.
   Per exemple → **Codi BN <u>NO és continu</u>**

→ combinacions $3 = 0011_2$

$4 = 0100_2$

veiem que hi ha <u>3 bits diferents</u>

Codis binaris: Concepte de codificació:

Codis: (1) ADJACENTS, (2) CONTINUS, i (3) CICLICS

1. **Combinacions adjacents →** Les que difereixen només en un sol bit.
2. **Codi continu →** Combinacions corresponent al **codi decimal** a on aquestes són sempre adjacents de forma consecutiva.

Decimal	Binari No Natural (BNN)
0	000
1	100
2	101
3	001
4	011
5	010
6	110
7	111

Codi continu: adjacents de forma consecutiva

Decimal	Binari No Natural (BNN)
0	000
1	010
2	011
3	111
4	110
5	100
6	101
7	001

Codi continu: adjacents de forma consecutiva

- **BNN:** Codis binaris que no es corresponen amb el sistema de numeració binari ponderada.
- Aquests 2 exemples són continus: sempre hi ha adjacència consecutiva en tots els termes.
- El segon, a més a més, és cíclic.

Codis binaris: Concepte de codificació:

Codis: (1) ADJACENTS, (2) CONTINUS, i (3) CICLICS

1. **Combinacions adjacents →** Les que difereixen només en un sol bit.
2. **Codi continu →** Combinacions corresponent al **codi decimal** a on aquestes són sempre adjacents de forma consecutiva.

Decimal	Binari No Natural (BNN)
0	000
1	100
2	101
3	001
4	011
5	010
6	110
7	111

Decimal	Binari No Natural (BNN)
0	000
1	010
2	011
3	111
4	110
5	100
6	101
7	001

Ha de funcionar amb totes les combinacions per N bits

- **BNN:** Codis binaris que no es corresponen amb el sistema de numeració binari ponderada.
- Aquests 2 exemples són continus: sempre hi ha adjacència consecutiva en tots els termes.
- El segon, a més a més, és cíclic.

Codis binaris: Concepte de codificació:

Codis: (1) ADJACENTS, (2) CONTINUS, i (3) CICLICS

3. Codis cíclic → codi continu en el que la **primera i última combinació** són també **adjacents**.

Decimal	Binari No Natural (BNN)
0	000
1	100
2	101
3	001
4	011
5	010
6	110
7	111

Decimal	Binari No Natural (BNN)
0	000
1	010
2	011
3	111
4	110
5	100
6	101
7	001

Ha de funcionar amb totes les combinacions per N bits

- **Per què?** faciliten la detecció d'errors.
- **Exemple codi taula 1:** és un codi **continu**, però **no cíclic**: la primera combinació (000) no és adjacent amb l'última (111).
- **Exemple codi taula 2:** és un codi **continu**, i **cíclic**: la primera combinació (000) és adjacent amb l'última (001). Això no pot dependre del número de bits. És a dir per 4 bits o més s'ha de mantenir.

Codis Binaris

- **Codi Binari Natural (BN)**
- **Codi GRAY**
- **Codi Johnson**
- **Codi BCD**
  - **BCD Natural**
  - ~~**BCD Exces 3**~~
  - ~~**BCD Aiken**~~

Binari NO Natural (BNN)
(segueixen unes regles)

Codis binaris: Codi GRAY (Binari No Natural)

- Codi "**reflectit**" → Per formar el codi Gray de "N" bits, es fan servir els valors del codi de "N-1" bits i es reflecteixen, de manera que queden $2 \cdot 2^{N-1}$ combinacions.
- A les primeres 2^{N-1} se'ls afegeix un "0" per l'esquerra, mentre que a les següents 2^{N-1} valen un 1.

N=1	N=2	N=3
0	00	000
1	01	001
	11	011
	10	010
		110
		111
		101
		100

Decimal	Binari GRAY
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

Codi GRAY de 3 dígits

Codis binaris: Codi GRAY (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{345_{10}}$,

Com obtenir el cadi GRAY equivalent?

Codis binaris: Codi GRAY (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{345_{10}}$,

Com obtenir el cadi GRAY equivalent?

Hi ha una operació en lògica binaria que es diu X-OR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Ho podem veure com una suma a on 1 + 1 = 0 (2 → 10)

Codis binaris: Codi GRAY (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{345_{10}}$,

Com obtenir el cadi GRAY equivalent?

IDEA

1. Convertir el valor a **binari natural** (mètode de les divisions).

$$\mathbf{345_{10}} = 1\ 0101\ 1001_b$$

Codis binaris: Codi GRAY (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{345_{10}}$,

Com obtenir el cadi GRAY equivalent?

IDEA

1. Convertir el valor a **binari natural** (mètode de les divisions).

   $\mathbf{345_{10}} = 1\ 0101\ 1001_b$

2. Desplaçar 1 bit a la dreta ("shiftar") i aplicar la operació **X-OR** bit a bit

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

$\mathbf{345_{10}} = 1\ 0101\ 1001$

→ 1010 1100 ~~1~~

$1\ 1111\ 0101_b$

Decimal → GRAY

Eliminem el dígit de menys pes

Codis binaris: Codi GRAY (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor **345_{10}** ,

Com obtenir el cadi GRAY equivalent?

Exemple senzill amb el '7'

Decimal	GRAY	Binari (natural)
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

```
 111
 111 →
------
 100
```

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

Codis Binaris

- **Codi Binari Natural (BN)**
- **Codi GRAY**
- **Codi Johnson**
- **Codi BCD**
  - **BCD Natural**
  - ~~**BCD Exces 3**~~
  - ~~**BCD Aiken**~~

Binari NO Natural (BNN)
(segueixen unes regles)

Codis binaris: Codi JOHNSON (Binari No Natural)

- Codi binari **continu** i **cíclic.**
- **Capacitat de codificació** menor que els anteriors → Es necessiten més bits:
  - Codi "**Gray**" i "**BN**" codifiquen els valors amb 2^N combinacions.
  - Codi "**Johnson**" de N bits els valors es codifiquen amb **2N** combinacions.

Exemple

- Codificar 10 números **decimals** en **GRAY** fan falta **4** bits.
- Codificar 10 números **decimals** en **Johnson** fan falta **5** bits.

- Amb **5 bits** codi **GRAY** → 2^5 = **32 combinacions** (diferents).
- Amb **5 bits** codi **Johnson** → 2 · 5 bits = **10 combinacions**.

MECANICA: el codi Johnson de N bits parteix de la combinació formada per tots els bits a zero i va posant a 1 els bits de la dreta de manera seqüencial, fins que obtenim la combinació formada per tots els bits a 1. A partir d'aquesta combinació es tornen a posar a zero els bits començant per la dreta fins a trobar de nou la combinació de partida (00000).

Decimal	Binari JOHNSON
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

Codis binaris: Codi JOHNSON (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{555_{10}}$

Com obtenir el cadi Johnson equivalent?

Codis binaris: Codi JOHNSON (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor **555_{10}**

Com obtenir el cadi Johnson equivalent?

1. Determinar el **rang [0...X]** on:
   - **X = 2·N – 1**
   - **N = número de bits mínim necessari**

Codi **Johnson** de 3 bits

Quants bits necessito per definir combinacions fins arribar al decimal 555? (màxim valor)

0	**0**	**00**	**000**	**0000**	
1	**1**	**01**	**001**	**0001**	
2		**11**	**011**	**0011**	
3		**10**	**111**	**0111**	**...**
4			**110**	**1111**	
5			**100**	**1110**	
6				**1100**	
7				**1000**	
...					

Recordem que en Johnson:

- **Número de combinacions → Parell (2, 4, 6...∞)**
- **Últim valor sempre senar (1, 3, 5, ...∞)**

Rang → [0 .. 2·N-1]

Codis binaris: Codi JOHNSON (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{555_{10}}$

Com obtenir el cadi Johnson equivalent?

1. Determinar el **rang [0...X]** on:
   - X = 2·N – 1
   - N = número de bits mínim necessari

Quants bits necessito per definir combinacions fins arribar al decimal 555? (màxim valor)

555 = [2·N – 1] ← Rang [0...555]

(555 + 1) / 2 = N

N = **278** → **2·(N=278) – 1 = 555**

Número de dígits necessari i el número o valor de la combinació del mig de la taula, coincideixen

278 dígits

0 = 0.................0

1 = 0...............01

...

278 = 1.................1

...

555 = 1 0.........000

Codis binaris: Codi JOHNSON (Binari No Natural)

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{555_{10}}$

Com obtenir el cadi Johnson equivalent?

1. Determinar el **rang [0...X]** on:
   - X = 2·N – 1
   - N = número de bits mínim necessari

Número de dígits necessari i el número o valor de la combinació del mig de la taula, coincideixen

Quants bits necessito per definir combinacions fins arribar al decimal 555? (màxim valor)

555 = [2·N – 1] ← Rang [0...555]

(555 + 1) / 2 = N

N = **278** → **2·(N=278) – 1 = 555**

Si hagués estat parell, seria el penúltim valor (ex: 556 → hauríem de incloure fins a 557)

$\overbrace{}^{\textbf{278 dígits}}$

0 = 0................0

1 = 0..............01

...

278 = 1................1

...

555 = 1 0.........000

Codis binaris: Codi JOHNSON (Binari No Natural)

Exemple. Si tinc 10 valors necessito 5 bits en Johnson

Què passa quan tractem valors binaris grans?

Per exemple, donat el valor $\mathbf{555_{10}}$

Com obtenir el cadi Johnson equivalent?

1. Determinar el **rang [0...X]** on:
   - X = 2·N – 1
   - N = número de bits mínim necessari

Decimal	Binari JOHNSON
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

Exemple:
N=5 bits
Rang?

[0...2·N-1]
[0...2·5-1]
[0.........9]

Quants bits necessito per definir combinacions fins arribar al decimal 555? (màxim valor)

555 = [2·N – 1] ← Rang [0...555]
(555 + 1) / 2 = N
N = **278** → **2·(N=278) – 1 = 555**

Si hagués estat parell, seria el penúltim valor (ex: 556 → hauríem de incloure fins a 557)

0 = 0................0
1 = 0..............01
...
278 = 1................1
...
555 = 1 0.........000

Codis binaris: Codi JOHNSON (Binari No Natural)

Què passa ara si ens diuen que donat el valor **555_{10}** com a màxim, ens demanen la combinació del número **300_{10}**?

Com la obtenim?

Codis binaris: Codi JOHNSON (Binari No Natural)

***Què passa ara si ens diuen que donat el valor* 555$_{10}$** com a màxim, ens demanen la combinació del número **300$_{10}$**?

Decimal	Binari JOHNSON
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

PAS 1:

Si màxim = 555, el rang és [0...555], la meitat és **278**

Per representar 555 (última combinació) → un 1 + 277 ze

o sigui → $555 = \underbrace{10000000000.............000000}_{\textbf{278 dígits}}$

Exemple pel cas de 10 combinacions

Codis binaris: Codi JOHNSON (Binari No Natural)

***Què passa ara si ens diuen que donat el valor* $\mathbf{555_{10}}$** com a màxim, ens demanen la combinació del número $\mathbf{300_{10}}$?

PAS 1:

Si màxim = 555, el rang és [0...555], la meitat és **278**

Per representar 555 (última combinació) → un 1 + 277 zeros

o sigui → 555 = $\underbrace{\text{10000000000.............000000}}_{\text{278 dígits}}$

PAS 2:

300 està per sobre o sota de la meitat (278)? → (300 > 278)

Una mica per sota → Per tant tindrà alguns **1's** i molts **0's**

300 - 278 = **22** → 22 posicions per sota = **256** uns + **22** zeros

11111..........1110000......0

0000..0 ← 0

...

0111..1 ← 277 } 278

1111..1 ← 278

1111..0 ← 279

...

1000..0 ← 555 } 278

Codis binaris: Codi JOHNSON (Binari No Natural)

Exemple 2: Ens demanen la codificació del $\mathbf{555_{10}}$ en codi Johnson on **N=300** (o sigui, on cada una de les combinacions és de 300 bits)

Codis binaris: Codi JOHNSON (Binari No Natural)

Exemple 2: Ens demanen la codificació del $\mathbf{555_{10}}$ en codi Johnson on **N=300** (o sigui, on cada una de les combinacions és de 300 bits)

555 ja no és el valor màxim de la meva taula

Número de dígits

Codis binaris: Codi JOHNSON (Binari No Natural)

Exemple 2: Ens demanen la codificació del $\mathbf{555_{10}}$ en codi Johnson on N=300 (o sigui, on cada una de les combinacions és de 300 bits)

Pas 1: Calcular el valor màxim: $2 \cdot N - 1 \rightarrow 2 \cdot (300) - 1 = 599 \rightarrow [0..599]$

Valor màxim

Codis binaris: Codi JOHNSON (Binari No Natural)

Exemple 2: Ens demanen la codificació del $\mathbf{555_{10}}$ en codi Johnson on N=300 (o sigui, on cada una de les combinacions és de 300 bits)

Pas 1: Calcular el valor màxim: $2 \cdot N - 1 \rightarrow 2 \cdot (300) - 1 = 599 \rightarrow [0..599]$

Pas 2: Si la combinació del 599 = 10000.....0000 (o sigui un 1 i 299 zeros)

Aleshores 599 - 555 = 45 + 1 uns, i la resta zeros

$$\underbrace{\underbrace{\mathbf{11111.....111}}_{\mathbf{45}}\underbrace{\mathbf{0000000......0000000}}_{\mathbf{255}}}_{\mathbf{300\ dígits}}$$

0000..0 ← 0

...

0111..1 ← 299

300

1111..1 ← 300

1111..0 ← 301

555 ...

1000..0 ← 599

300

Codis Binaris

- Codi Binari Natural (BN)
- Codi GRAY
- Codi Johnson
- Codi BCD
  - BCD Natural
  - ~~BCD Exces 3~~
  - ~~BCD Aiken~~

Binari NO Natural (BNN)
(segueixen unes regles)

Codis binaris: Codi BCD (Binari No Natural)

- BCD (Binary Coded Decimal).
- Codi que només contindrà valors codificats de **valors decimals**.
- Cada dígit "decimal" es codificarà amb **4 bits**, ja que amb tres ($2^3 = 8$) només es podrien representar del 0 al 7.
- Com s'utilitzen 4 bits ($2^4 = 16$) per representar els dígits del **0 al 9**, es perden 6 combinacions.

0_{10}	→	0000_2
1_{10}	→	0001_2
2_{10}	→	0010_2
3_{10}	→	0011_2
4_{10}	→	0100_2
5_{10}	→	0101_2
6_{10}	→	0110_2
7_{10}	→	0111_2
8_{10}	→	1000_2
9_{10}	→	1001_2

Codis binaris: Codi BCD (Binari No Natural)

- BCD (Binary Coded Decimal).
- Codi que només contindrà valors codificats de **valors decimals**.
- Cada dígit "decimal" es codificarà amb **4 bits**, ja que amb tres ($2^3 = 8$) només es podrien representar del 0 al 7.
- Com s'utilitzen 4 bits ($2^4 = 16$) per representar els dígits del **0 al 9**, es perden 6 combinacions.

$0_{10} \rightarrow 0000_2$
$1_{10} \rightarrow 0001_2$
$2_{10} \rightarrow 0010_2$
$3_{10} \rightarrow 0011_2$
$4_{10} \rightarrow 0100_2$
$5_{10} \rightarrow 0101_2$
$6_{10} \rightarrow 0110_2$
$7_{10} \rightarrow 0111_2$
$8_{10} \rightarrow 1000_2$
$9_{10} \rightarrow 1001_2$

Exemple

- **9.999** en BCD → **4 dígits x 4 bits / dígit = 16 bits XXXX-XXXX-XXXX-XXXX**

No confondre amb el codi **Binari Natural (BN) → 14 bits ← 2^{14}** (0 - 16.384)

9999 = 10 0111 0000 1111

Si es representés aquesta xifra en BN serien suficients 14 bits, ja que $2^{14} = 16384$, que és més gran que 9999, i $2^{13} = 8.192$, que és menor que 9999, així que realment s'estan utilitzant, en aquest cas, dos bits més dels necessaris en la representació binària natural.

Resum codis binaris no naturals (BNN) que hem vist fins ara: Exemples

Codi GRAY:

Decimal	Binari GRAY
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

continu + cíclic

Codi JOHNSON:

Decimal	Binari JOHNSON
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

continu + cíclic

Codi BCD (natural):

Decimal	Binari BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

NO continu
NO cíclic

Codis alfanumèrics

- Els codis estudiats anteriorment només permeten representar informació numèrica.
- A vegades es fa imprescindible representar informació alfabètica, incloent símbols especials → codis alfanumèrics.
- El codi alfanumèric més utilitzat és el codi ASCII (American Standard Code for Information Interchange).
- Per poder representar 26 caràcters alfabètics i els 10 numèrics són necessaris 6 bits.
- Però, com sobren 28 combinacions → $2^6 – (26 + 10) = 64-36$, aquestes s'aprofiten per poder codificar els símbols especials necessaris.
- La taula del codi ASCII original era de 6 bits, però estès en dues ocasions per permetre altres caràcters i símbols, com són 'ç', 'à', 'ñ', ... que en l'original no estaven inclosos.
- La taula ASCII més utilitzada és la de 8 bits, amb la qual es poden representar 256 caràcters diferents.

Codis alfanumèrics

DEC	HEX	OCT	CHAR	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH
0	0	000	NUL	32	20	040		64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	80	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
29	1D	035	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Codis Binaris

Detecció d'ERRORS

Detecció d'errors en codis binaris

- **Transmissions de dades entre sistemes digitals:**

  Hi ha errors deguts a vàries causes:

  - Sorolls.
  - Interferències electromagnètiques.
  - Mal funcionament d'algun component electrònic.
  - etc...

- **Detecció d'errors no es senzilla:**

  Si s'altera un valor binari el codi pot seguir sent vàlid en funció del sistema de detecció que s'utilitzi.

  Exemple: *error en una codificació en* ***BCD Natural***

  0111 → 1111 (7 → 16) = incorrecte (detectable) → 16 $\notin$ { 0-9 }

  0111 → 0011 (7 → 3) = correcte (no detectable)

Detecció d'errors en codis binaris

Conceptes relacionats amb la detecció d'errors

1. **"Distància" entre 2 combinacions binàries:** nombre de bits diferents.

   Exemple: *distància entre la combinació en BCD Natural del 2 i el 7*

   0010 → 0111 (2 → 7) la distancia és **2,** ja que **canvien 2 bits**.

2. **"Distància mínima d'un codi (dm)"** és **la menor** de les distàncies entre 2 combinacions qualssevol del codi (és a dir, no cal que siguin consecutives).
   - Tots els codis que hem vist tenen una dm = 1 → Sempre hi ha una combinació que respecte d'una altre "qualsevol "només es diferencien amb 1 bit.
   - dm = 1 suposa un problema per poder detectar errors quan hi ha un error d'1 bit en una combinació → Una "combinació mutada" es pot convertir en una altra combinació vàlida que pertany al codi.
   - Per poder detectar errors d'1 bit en una combinació, la dm té que se superior a 1 bit.

Detecció d'errors en codis binaris

Conceptes relacionats amb la detecció d'errors

DETECCIÓ

- La solució són els "**codis detectors d'errors**", una eina que permet a un sistema digital receptor determinar si la informació que ha rebut és **correcta** o **no**.
- Si (ERROR) detectat → Es pot enviar al sistema transmissor una **petició de reenviament**.

SOLUCIÓ

→ Usar **Codis de detecció d'errors**

Codis de detecció d'errors

- Els **"codis detectors d'errors"** detecten errors **en 1 sol bit** de la informació.
- Existeixen un gran nombre de codis detectors d'errors.
- En aquest curs només estudiarem els **codis detectors "de paritat"**.

Codis de detecció d'errors "de paritat"

- Afegeixen un bit més al codi com a nou bit LSB o MSB.
- El valor del bit dependrà de si s'ha escollit paritat **parell** o **imparell**:
    - **Paritat parell** → Nombre d'1s de cada combinació es parell (ex: 00110)
    - **Paritat imparell** → Nombre d'1s de cada combinació es imparell (ex: 00100)
- Els codis inicials estudiats fins ara (BN, Gray, Johnson, BCD, ...) presenten distància mínima (dm) = 1. En afegir 1 bit més de paritat, obtenim codis amb distància mínima (dm) = 2.
- La detecció d'errors s'encarrega de comptar el nombre d'1s dels bits rebuts.
- **Sabent quina paritat s'utilitza** sabrem si el valor és "correcte" o "erroni".

Codis de detecció d'errors

- Exemple: codi **BCD natural** amb 1 **bit de paritat imparell**.

Decimal	BCD natural
0	000$\boxed{0}$
1	000$\boxed{1}$
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

1 canvi

Codis de detecció d'errors

- Exemple: codi **BCD natural** amb 1 **bit de paritat imparell**.

Decimal	BCD natural	Bit paridad
0	0000	1
1	0001	0
2	0010	0
3	0011	1
4	0100	0
5	0101	1
6	0110	1
7	0111	0
8	1000	0
9	1001	1

2 canvis

- Si revisem totes les combinacions veurem que la distància mínima d'aquest codi és 2.
- Per tant, permetrà detectar canvis d'un bit a on abans no es podia.

Codis de detecció d'errors

- Exemple: codi **BCD natural** amb 1 **bit de paritat imparell**.

Decimal	BCD natural	Bit paridad
0	0000	1
1	0001	0
2	0010	0
3	0011	1
4	0100	0
5	0101	1
6	0110	1
7	0111	0
8	1000	0
9	1001	1

2 canvis

- Si revisem totes les combinacions veurem que la distància mínima d'aquest codi és 2.
- Per tant, permetrà detectar canvis d'un bit a on abans no es podia.

Paritat imparell

- Enviem 97_{10} = 10011 0111 1 ← 7 **1's** (imparell, ok)
- Rebem 10001 0111 1 ← 6 **1's** (parell, error)

ERROR   Bit de paritat

Detecció d'errors en codis binaris: RESUM

- **"Distància" entre dues combinacions binàries:** nombre de bits que cal canviar d'una de les dues combinacions per formar l'altra.
- La **"distància mínima d'un codi**" **(dm)** és **la menor** de les distàncies entre dues combinacions qualssevol del codi. 1 → (4,3,2,1).
- **Codis de detecció d'errors → Codis de detecció d'errors "de paritat"**
  - **Paritat parell →** Nombre d'1s de cada combinació es parell.
  - **Paritat imparell →** Nombre d'1s de cada combinació es imparell.

EXERCICI 2

Canvis de base i codificació

1. Donat el nombre **3Bh** expressar-lo en **base** 3, **base 5** i base **binària**.
2. Donat el nombre **367** expressar-lo en base **octal**, **hexadecimal** i **binària**.
3. Donat el nombre **133.610** expressar-lo en base **octal**, **hexadecimal** i **base 7**.
4. Quants bits fan falta per codificar 20 números **decimals** en codi **GRAY?**
5. Quan bits necessito per codificar 64 combinacions amb codi **GRAY**?
6. Quan bits necessito per codificar 32 combinacions amb codi **JOHNSON**?

Solució

1. Donat el nombre **3Bh** expressar-lo en **base** 3, **base 5** i base **binària**.

$3 \cdot 16^1 + B(11) \cdot 16^0 = 59_{10}$ (decimal)

59 | **3**
2 19 | 3
1 6 | 3
0 2 | 3
2 0

59 | **5**
4 11 | 5
1 2 | 5
2 0

59 | **2**
1 29 | 2
1 14 | 2
0 7 | 2
1 3 | 2
1 1 | 2
1 0

Base 3 = 2012_3

Base 5 = 214_5

Base 2 = $11\ 1011_2$

Alternativa per passar de **hexadecimal** directament a **binari**:
3Bh = 0011 1011 b = 111011_2

2. Donat el nombre **367** expressar-lo en base **octal**, **hexadecimal** i **binària**.

367_{10} (decimal)

367	**8**	45	7
45	8	5	5
5	8	0	5

367	**16**	22	15
22	16	1	6
1	16	0	1

367	**2**	183	1
183	2	91	1
91	2	45	1
45	2	22	1
22	2	11	0
11	2	5	1
5	2	2	1
2	2	1	0
1	2	0	1

Alternativa per passar d'**octal** a **binari**
5 = 101b , 7 = 111b →
557_8 = 101 101 111

Base 8 = 557_8

Base 16 = $16F_{16}$

Base 2 = $1\ 0110\ 1111_2$

3. Donat el nombre **133.610** expressar-lo en base **octal**, **hexadecimal** i **base 7**.

133.610 | **8**
2 16701 | 8
5 2087 | 8
7 260 | 8
4 32 | 8
0 4 | 8
4 0

133.610 | **7**
1 19.087 | 7
5 2.726 | 7
3 389 | 7
4 55 | 7
6 7 | 7
0 1 | 7
1 0

Base 8 = 404752_8

Base 16 = $209EA_{16}$

Base 7 = 1064351_7

133.610 | **16**
10 8350 | 16
14 521 | 16
9 32 | 16
0 2 | 16
2 0

4. Quants bits fan falta per codificar 20 números **decimals** en codi **GRAY?**

   Igual que en codi binari natural → $2^5 = 32$ (32 > **20** > 16 → 2^4)

   5 bits.

   20 → [0..31] → 10100 (BN) → 10100

$$\begin{array}{r} \mathbf{10100} \\ \hline \mathbf{11110_2} \end{array}$$

5. Quan bits necessito per codificar 64 combinacions amb codi **GRAY**?

   Igual que en codi BN (combinacions 0..63): $2^N = 64 \rightarrow 2^N = 2^6$

   N = 6 bits.

6. Quan bits necessito per codificar 32 combinacions amb codi **JOHNSON**?

   Número de combinacions = 2·N on N és el número de bits necessaris.

   N = 32/2 = **16 bits**