

Sistemi di numerazione e codici

Capitolo 1



Generalità

■ Sistema di numerazione

- Insieme di simboli (cifre) e regole
- stringa di cifre \leftrightarrow valore numerico
- codici posizionali (il valore dipende dalla posizione delle cifre)

In base 10 (la piu' comune)

$$A_n A_{n-1} A_{n-2} \dots \dots \dots A_0 \quad \Rightarrow \quad N = A_n \cdot 10^n + A_{n-1} \cdot 10^{n-1} + A_{n-2} \cdot 10^{n-2} + \dots \dots \dots + A_0 \cdot 10^0$$

Ad esempio

$$1923 \quad \Rightarrow \quad 1 \cdot 10^3 + 9 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$

- Si possono pero' adottate altre basi con $B \neq 10$
(le piu' comuni: $B=2$, $B=8$, $B=16$)
si adottano B cifre diverse
(Ad.es $B=16$: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)



Rappresentazione in varie basi

- Cosa significa numerare (secondo una base predefinita) un insieme di oggetti ?

Numero	resto 10	Numero	resto 7	Numero	resto 2	Numero	resto 8	Numero	resto 16
1765	5	1765	1	1765	1	1765	5	1765	5
176	6	252	0	882	0	220	4	110	14
17	7	36	1	441	1	27	3	6	6
1	1	5	5	220	0	3	3		
				110	0				
				55	1				
				27	1				
				13	1				
				6	0				
				3	1				
				1	1				



Sistema di numerazione su base qualsiasi

- La base 2 e' la piu' piu' piccola possibile (ON/OFF) - Binary Digit
- Le basi 8 e 16 permettono rappresentazioni piu' compatte del numero binario
 - Il passaggio da base 2 a base 8 o 16 e viceversa e' particolarmente facile

$$55_{10} = 110111_2$$

$$110111_2 = 37_{16} = 67_8$$



Conversione tra basi diverse

- Divisione successiva per la Base
 - si divide ripetutamente il numero per la base voluta fino ad ottenere un quoziente nullo e si memorizzano i resti (la seq. dei resti ordinata rappresenta la notazione)
- Per quanto detto il passaggio da basi B a B^n e viceversa risulta particolarmente semplice

Es:

$$157_{10} = 10011101_2 = 235_8 = 9D_{16}$$

Numero	resto 10	Numero	resto 7	Numero	resto 2	Numero	resto 8	Numero	resto 16
157	7	157	3	157	1	157	5	157	13
15	5	22	1	78	0	19	3	9	9
1	1	3	3	39	1	2	2	0	0
0	0	0	0	19	1	0	0		
				9	1				
				4	0				
				2	0				
				1	1				
				0	0				
				0	0				
				0	0				



Conversione di frazioni

- La parte frazionale viene distinta dalla parte intera mediante una “virgola” : “,”

Ad esempio

$$1923,45 \rightarrow \underbrace{1 \cdot 10^3 + 9 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0}_{\text{parte intera}} + \underbrace{4 \cdot 10^{-1} + 5 \cdot 10^{-2}}_{\text{parte frazionale}}$$

- La virgola distingue le cifre che vanno moltiplicate per B con esponente positivo da quelle con esponente negativo
- La conversione avviene in tal caso per moltiplicazioni successive

$$0,375_{10} = 0,011_2$$



Conversione di frazioni

■ NOTA BENE

- Se con una base una notazione frazionaria richiede un numero finito di cifre, potrebbe richiederne infinite con una diversa notazione

$$(1/3)_{10} = 0,333333 \dots_{10} = 0,1_3$$

$$0,6375_{10} = 0,101000\overline{11}_2$$

■ Conversione da binario a decimale

- Parte intera: raddoppio successivo + somma a partire dalla cifra piu' significativa
- Parte frazionaria: idem + successiva divisione per 2^f ove f sono i bit rappresentativi della parte frazionale

$$101,010_2 = (5 + 2/8)_{10} = 5,25_{10}$$



Aritmetica Binaria

■ Addizione

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0 \text{ (+ riporto di 1 al rango superiore)}$$

■ Sottrazione

$$0-0=0$$

$$1-0=1$$

$$1-1=0$$

$$0-1=1 \text{ (+ riporto negativo di 1 al rango superiore)}$$

- La sottrazione puo' pero' avvenire tramite la somma usando una notazione complementata



Complemento decimale

- ES:

$$123 - 73 = 123 + \text{comp}(73) = 123 + 927 = 1\overline{)050} = 50$$

- In questo caso si usa un complemento a $10^3=1000$ ovvero $\text{comp}(73)=1000-73$ che e' facile da calcolare basta adottare la corrispondenza

0 → 9	9 → 0
1 → 8	8 → 1
2 → 7	7 → 2
3 → 6	6 → 3
4 → 5	5 → 4

e poi sommare 1

$$073 \rightarrow 926 + 1 = 927$$



Complementi a B e B-1

- Analogamente in altre Basi (ad esempio base 2)

Si definiscono:

$$C_B = B^n - N \quad \text{e} \quad C_{B-1} = B^n - 1 - N$$

Da cui si desume che:

$$C_B = C_{B-1} + 1$$

Attenzione : viene definito come (Complemento a B-1) il (Complemento a B)-1

- Il complemento a B-1 è semplice da calcolare
 - basta una tabella di equivalenza (come prima)

$$C_{B-1} = B^{n-1} - N = B'B'B'B'B'B'B'B' - N \quad \text{ove} \quad B' = B-1$$

- Il complemento a B si ottiene dal precedente sommandovi 1

$$N_1 - N_2 \rightarrow N_1 + C_B(N_2) = N_1 + (B^n - N_2) = B^n + (N_1 - N_2)$$



Numeri negativi

- Dalla differenza di N_1 ed N_2 vi possono essere due casi:
 - $N_1 \geq N_2$: il risultato risulta maggiore o uguale a B^n , che pertanto va eliminato dal risultato finale (eliminazione dell'1 più significativo oltre il range del numero stesso)
 - $N_1 < N_2$: il risultato risulta minore di B^n , e deve essere inteso come complemento a B (pertanto rappresentante di un numero negativo) del risultato

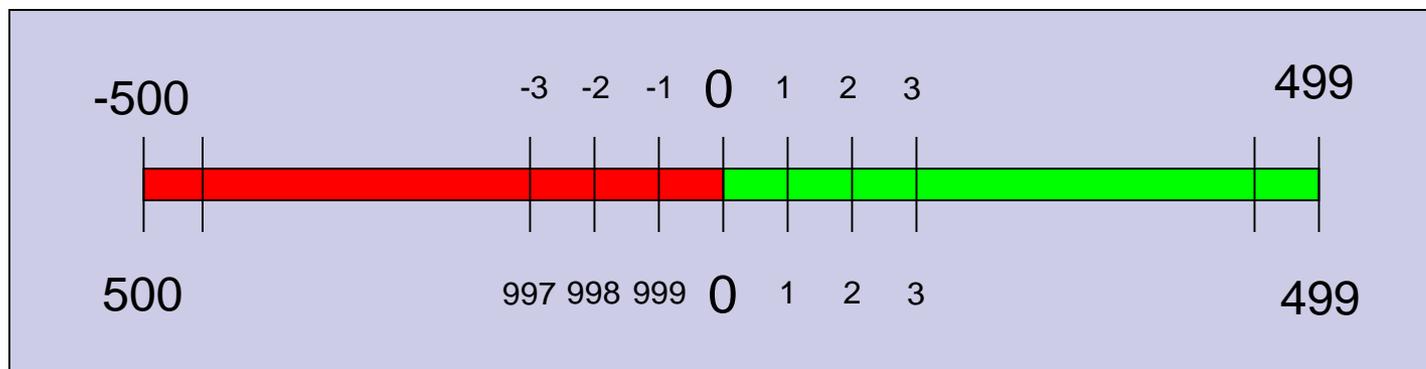


Numeri negativi

- I numeri negativi possono pertanto essere rappresentati in base al loro complemento a B

$$-143 \rightarrow C_{10}(143) = 999 - 143 + 1 = 856 + 1 = 857$$

- Si può notare che il range dei numeri risulta modificato:
 - $0 < n < 499$: range dei numeri positivi
 - $500 < n < 999$: range dei numeri negativi



- Ovviamente in base 10 questa non è una pratica usata

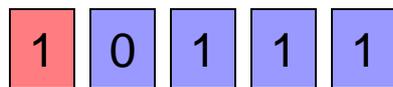


Numeri binari negativi

- Risulta invece estremamente diffusa nel caso di numeri binari ove i numeri negativi vengono rappresentati in base al loro complemento a 2

Es:

$$-9 = -01001 = C_2(01001) = 10110 + 1 = 10111$$



Bit di segno:

$\left\{ \begin{array}{l} 0: \text{numero positivo} \\ 1: \text{numero negativo} \end{array} \right.$

Numero positivo: i restanti numeri rappresentano il numero stesso

Numero negativo: i restanti numeri rappresentano il numero complementato

$$C_2(0111) = 1000 + 1 = 1001 = 9_2$$



Errori nei risultati

- Il risultato di un'operazione somma/sottrazione è coerente solo se il risultato non esce dal range dei numeri rappresentabili
 - Ovvero
 - o non si è avuto alcun riporto né nel bit di segno né fuori dalla parola
 - o si sono avuti riporti in entrambi
 - se si è avuto un solo riporto il risultato è errato

$0011 \rightarrow 3+$	$0011 \rightarrow 3+$	$1011 \rightarrow -5+$	$1111 \rightarrow -1+$
$\frac{0010 \rightarrow 2=}{0101 \rightarrow 5}$	$\frac{0110 \rightarrow 6=}{1001 \rightarrow -7}$	$\frac{1010 \rightarrow -6=}{10101 \rightarrow 5}$	$\frac{1110 \rightarrow -2=}{11101 \rightarrow -3}$
nessun riporto	riporto sul segno	riporto fuori dalla parola	riporto sia sul segno che fuori dalla parola

■ : cifre “out of range” e pertanto non vengono calcolate

■ : bit di segno



Moltiplicazione e Divisione

- Moltiplicazione binaria

$$0*0=0$$

$$0*1=0$$

$$1*0=0$$

$$1*1=1$$

nel caso di piu' cifre si procede come nel caso decimale

- Divisione

- vengono di solito eseguite per sottrazioni successive



Modificare il numero di BIT

ESERCIZI:

- Aumentare o diminuire
- Per notazioni “signed” ed “unsigned”
- Da unsigned verso signed
- Considerazioni su arrotondamento e troncamento per numeri sia signed che unsigned



Range di un numero in Fixed Point

- **Unsigned** a N bit dove $N = I + F$
 - Parte Intera : I bit
 - Parte Frazionaria: F bit
 - Numero minimo: 0 (000000,0000)
 - Numero massimo $2^I - 2^{(-F)}$ (111111,1111)
 - Step $2^{(-F)}$ (000000,0001)
- **Signed** a N bit dove $N = I + F$
 - Parte Intera : I bit (di cui uno è il bit di segno)
 - Parte Frazionaria: F bit
 - Numero massimo positive $2^{(I-1)} - 2^{(-F)}$ (011111,1111)
 - Numero minimo negativo $2^{(I-1)}$ (100000,0000)
 - Zero (000000,0000)
 - Minimo step positivo (000000,0001)
 - Minimo step negativo (111111,1111)



Codici

- **Codice:**
Insieme di parole $[C]$ adottato per rappresentare gli elementi di un insieme $[C^*]$
- **Simboli:**
elementi costituenti le parole di codice
- **Codificazione:**
associazione parola di $[C] \leftrightarrow$ elemento di $[C^*]$
- **Codice non ambiguo:** se la corrispondenza e' univoca,
- **Codice ambiguo:** se almeno una parola di $[C]$ rappresenta 2 o piu' elementi di $[C^*]$;

- se vi sono K simboli ed N elementi e le parole sono di lunghezza l :
usando n simboli $\rightarrow K^n$ combinazioni per non avere ambiguita' $N < K^n$

$$n = \lceil \log_K N \rceil$$

- **Codice efficiente / ridondante / ambiguo:** se $l = n$, $l > n$, $l < n$



Codici Efficienti

- Rappresentazioni cifre decimali
4 bits = 16 configurazioni → 6 configurazioni sono inutilizzate
- Codice BCD
codice ponderato (detto anche codice 8421)
- Codice eccesso tre
(binario +3) e' un codice autocomplementante
- Codice Aiken (o 2421)
autocomplementante e ponderato

BCD

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

Ecc. 3

0	0011	9	1100
1	0100	8	1011
2	0101	7	1010
3	0110	6	1001
4	0111	5	1000

Aiken

0	0000	9	1111
1	0001	8	1110
2	0010	7	1101
3	0011	6	1100
4	0100	5	1011



Codici Ridondanti

- Utili ad evidenziare/correggere eventuali errori (si usano k bit per il controllo)
 - $m = n + k$
- **Ridondanza**: rapporto tra i bit usati ed i bit strettamente necessari
 - R (ridondanza) = $m / n = 1 + k / n$
- **Peso**: numero di bits diversi da 0
- **Distanza**: numero di bits per cui 2 configurazioni differiscono
- **Molteplicita' d'errore**: Distanza tra la configurazione trasmessa e quella (non significativa) ricevuta – errori singoli, doppi, tripli ...
- **Distanza di Hamming (h)**: la minima distanza tra tutte le possibili coppie di parole di un codice: Sono individuabili gli errori con molteplicita' minore di h . Se h e' grande si puo' operare una correzione dell'errore (codici autocorrettori)



Probabilità di errore non rilevato

Sia p : la probabilità di errore di ogni singolo bit

la probabilità che una parola si trasformi in un'altra a distanza esattamente r e'

$$P_r = p^r \cdot (1-p)^{m-r} \binom{m}{r}$$

r : cifre errate
 $m-r$: cifre esatte
Combinazioni possibili

la probabilità che l'errore non sia rilevato dipende da quante configurazioni significative N_r si trovano a distanza "r" dalla parola

$$P_{tr} = P_{sr} \cdot p^r \cdot (1-p)^{m-r} \cdot \binom{m}{r} \quad \text{ove} \quad P_{sr} = \frac{N_r}{\binom{m}{r}}$$

conf. significative
conf. possibili

La prob. di errore non rilevato e la sommatoria per ogni r

$$P_t = \sum_h^m N_r p^r \cdot (1-p)^{m-r} \cong N_h p^h$$

tipicamente $p \ll 1$



Codice a controllo di parita'

- Ai vari bit che compongono la parola si aggiunge un ulteriore bit (ridondante)
 - detto bit e' 0 se il peso della parola e' pari
 - e' 1 se il peso e' dispari
- La parola risultante sara' a peso pari
- La distanza di Hamming e' 2
- E' in grado di rilevare tutti gli errori di molteplicita' dispari



Esempio cod. a controllo di parita'

- Un codice a 7 bit (128 parole)
- ha una ridondanza $R=7/6=1,16$
- Vi sono 64 parole ed altrettante config. non significative
- Per ogni parola il numero di parole che distano 2 sono:

$$N_h = \binom{7}{2} = 21$$

Per avere un'altra parola del codice si devono commutare 2 bit su 7

- Supponendo $p=0,01$
- La prob. di errore non rilevato e'

$$P_t = N_h p^h = 21 \cdot 0,01^2 = 0,21\%$$

In pratica coincide con la probabilita' che vi sia un errore di molteplicita' 2 (solo perche' tutte le configurazioni a distanza 2 sono significative)



Codici a peso costante

- Tutte le parole hanno lo stesso peso w
- la distanza di Hamming e' $h=2$

se ' m ' e' la lunghezza

- le parole significative saranno $\binom{m}{w}$
- mentre le config. non signif. saranno $2^m - \binom{m}{w}$
- Per ogni parola il numero di parole a distanza 2 e'

$$N_2 = w(m-w)$$

Bisogna commutare uno $1 \rightarrow 0$
(in w modi)
ed uno $0 \rightarrow 1$
(in $m-w$ modi)

pertanto

$$P_t = N_2 p^2 = w(m-w) p^2$$

Non basta che vi sia un errore doppio, ma questo deve portare anche in un'altra configurazione significativa



Codice 2 da 5

- Usa tutte le configurazioni possibili
infatti:

$$n = \binom{5}{2} = 10$$

- Ridondanza

$$R = \frac{5}{4} = 1,25$$

0	01100	5	00110
1	11000	6	10001
2	10100	7	01001
3	10010	8	00101
4	01010	9	00011

- Prob. di errore non rilevato con $p=0,01$

$$P_t = 2(5 - 2)p^2 = 0,06\%$$



Codice biquinario

- Doppio controllo di parità sui primi 2 e sugli ultimi 5 bits
- Ridondanza

$$R = \frac{7}{4} = 1,75$$

0	10 10000	5	01 10000
1	10 01000	6	01 01000
2	10 00100	7	01 00100
3	10 00010	8	01 00010
4	10 00001	9	01 00001

- $N_h=5$
- Prob. di errore non rilevato

$$P_t = N_h p^h = 5 \cdot (0,01)^2 = 0,05\%$$

Le configurazioni significative a distanza 2 da ogni parola sono **SOLO 5**

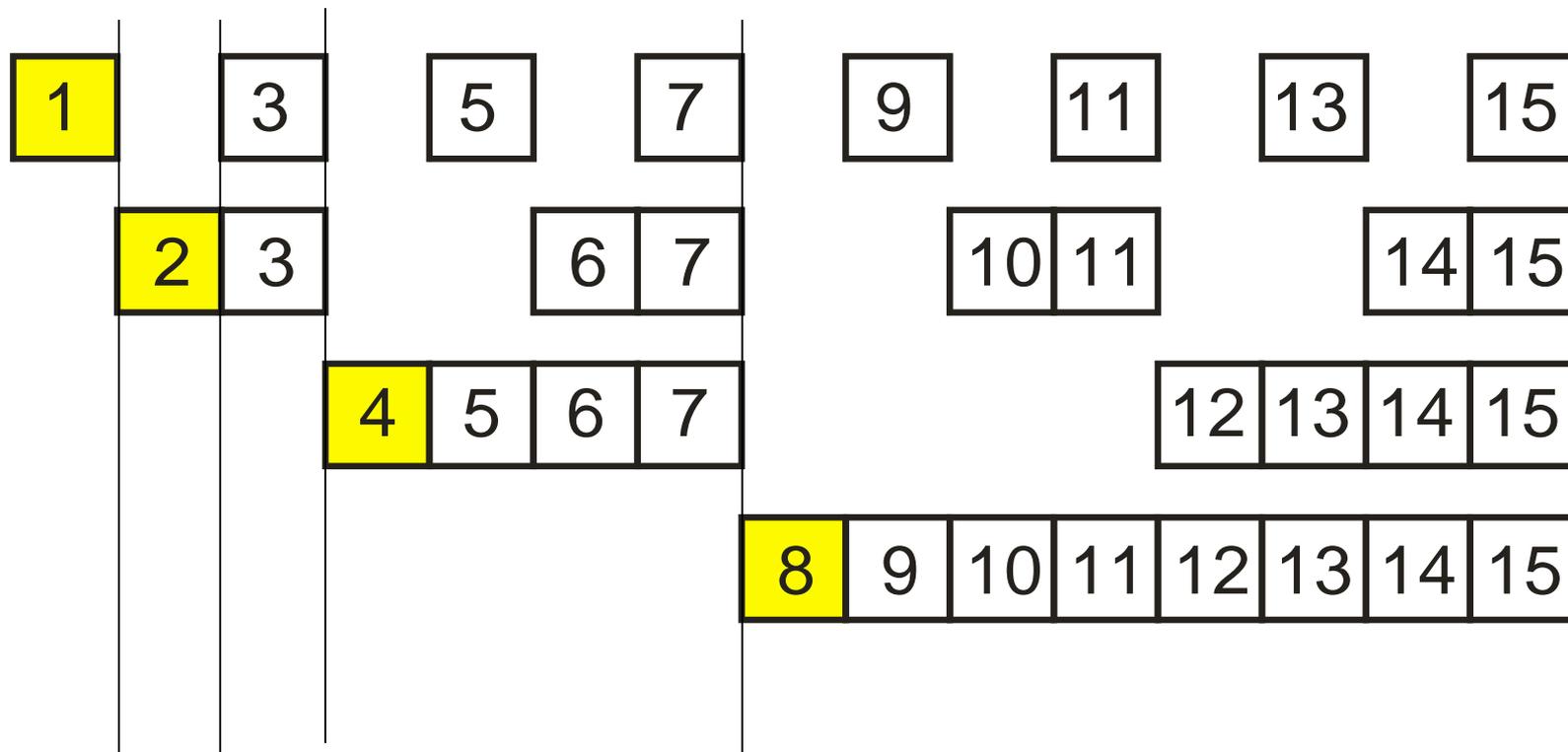


Codici di Hamming

- Sono codici con $h=3$ o $h=4$ usati come rilevatori/autocorrettori di errore
- molteplicità di errore rilevabile $r < h-1$
- molteplicità di errore correggibile $c < h/2$
- Dato un codice efficiente ad n bits vi si aggiungono k bits di controllo che controllano la parità di gruppi di bits i bits aggiunti si posizionano alla posizione 2^b
 - bit 1 : controllo di parità per 1,3,5,7,9,11,13,15,17,
 - bit 2 : controllo di parità per 2,3,6,7,10,11,14,15,
 - bit 4 : controllo di parità per 4,5,6,7,12,13,14,15,
 - bit 8 : controllo di parità per 8,9,10,11,12,13,14,15,
- In ricezione si verifica la parità per ogni gruppo e si scrive 0 se verificata, 1 se non verificata. Il risultato (letto in binario) darà la posizione del bit errato



Codice di Hamming (schema)



 : Bits di controllo (controllori di parita')

Nota: la commutazione di un bit della parola comporta la commutazione di almeno due bits di parita' pertanto la distanza minima tra le parole e' 3
N.B. NON tutte le parole sono distanti 3 tra loro ma tutte le parole sono sicuramente distanti ALMENO 3 altre distano molto di piu'



Cod. di Hamming (esempio)

Si voglia trasmettere : [0 1 0 1]

Si trasmettera': [b₁ b₂ 0 b₄ 1 0 1]

ove b₁=0 , b₂ = 1 , b₄ = 0

Si trasmettera' pertanto [0 1 0 0 1 0 1]

Supponendo di ricevere [0 1 0 1 1 0 1]

parita' dei bit 1,3,5,7: 0

parita' dei bit 2,3,6,7: 0

parita' dei bit 4,5,6,7: 1

Risultato : errore in posizione 1 0 0 → ovvero 4



Cod. di Hamming

- Per il corretto funzionamento

$$m \leq 2^k - 1$$

m: dimensione della parola totale = n+k

k: bit di controllo

n: bit di informazione

$$n + k = 2^k - 1$$

$$n = 2^k - (k + 1)$$

- Si dicono ottimi i codici in cui per la relazione di cui sopra è verificata con il segno uguale

k=1	n = 0
k=2	n = 1
k=3	n = 4
k=4	n = 11
k=5	n = 26



Cod. di Hamming

- Nel caso di un cod. di Hamming a 7 bits

$$P_t \leq 15 \cdot p^3 \quad \Rightarrow \quad \text{se } p = 0,01 \quad \Rightarrow \quad P_t \leq 15 \cdot 0.01^3 = 0,0015\%$$

Ammettendo che le parole a distanza 3 siano tutte parole del codice e tutte le altre parole del codice

- Esistono anche cod. di Hamming con $h=4$ a 8 bit (vi e' un ulteriore bit di parita` globale: si rilevano errori doppi e tripli e si correggono quelli singoli)

$$P_t \leq 15 \cdot p^4 \quad \Rightarrow \quad \text{se } p = 0,01 \quad \Rightarrow \quad P_t \leq 15 \cdot 0.01^4 = 1,5 \cdot 10^{-7}$$



Cod. di Hamming (h=4)

■ Esempio

- si voglia trasmettere [1 1 0 0]
- si crea la parola [p b₁ b₂ 1 b₄ 1 0 0]
- ovvero [p 0 1 1 1 1 0 0]
- quindi [0 0 1 1 1 1 0 0]
- supponiamo vi sia un errore doppio [0 0 1 0 1 1 1 0]
- che l'errore sia doppio lo si rileva perché il bit di parità è corretto, mentre le parità parziali non lo sono
 - parità complessiva : 0
 - parità bit 1,3,5,7 : 1
 - parità bit 2,3,6,7 : 0
 - parità bit 4,5,6,7 : 1
- Vi sono 2 possibilità:
 - o sono sbagliati p e b₅ (prob. 2/N)
 - oppure sono sbagliati 2 bit interni (prob 1-2/N) (più elevata)
 - però vi sono (2^{k-1}-1) possibili coppie di bit (k: bit di controllo)



Codici riflessi

- Sono ciclici ed ogni config. differisce dalla precedente per un bit
 - producono poco “rumore”
 - non sono ponderati
 - Si dicono completi se hanno tutte le 2^n combinazioni
- Il piu' noto e il codice di Gray

n=1
0
1

n=2
00
01
11
10

n=3
000
001
011
010
110
111
101
100

n=4
0000
0001
0011
0010
0110
0111
0101
0100
1100
...



Codice di Gray

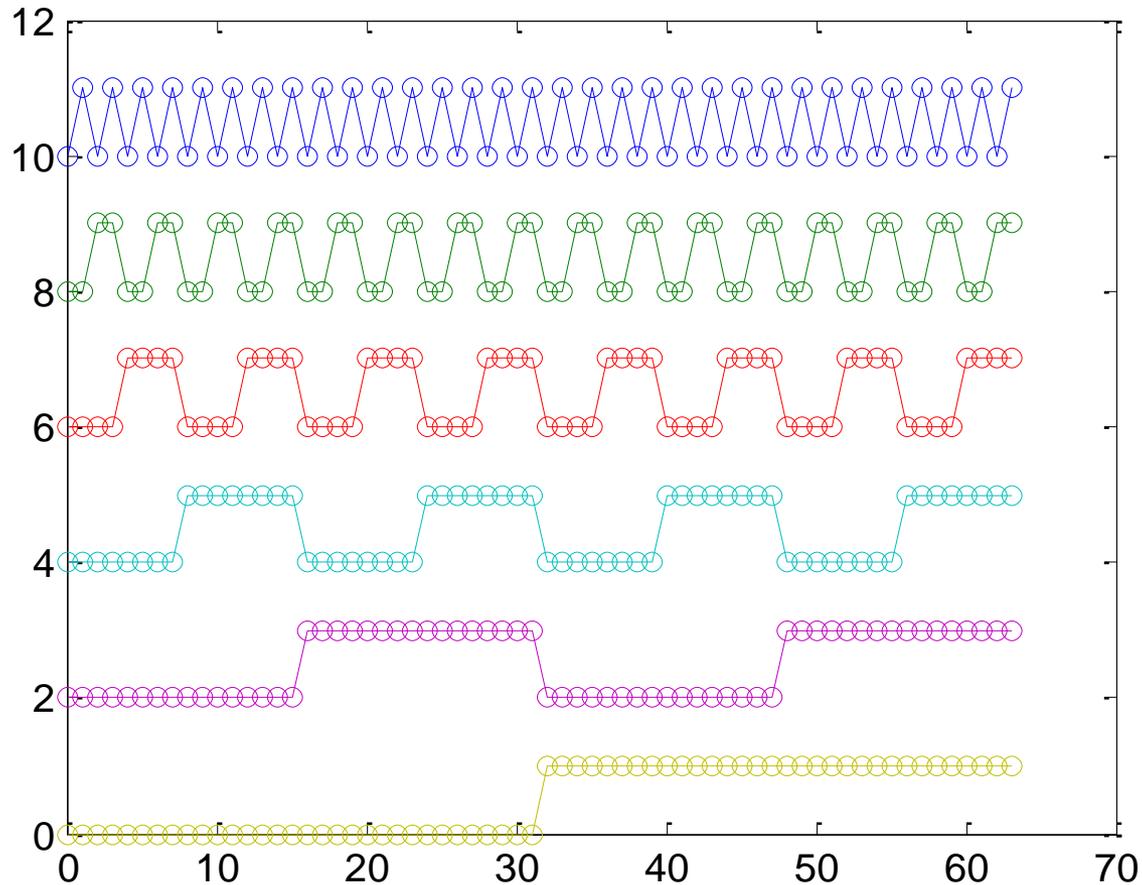
- La conversione Gray \leftrightarrow binario e' particolarmente semplice

Gray:	1 0 1 1 1 1 0 1 0 0 1
Binario:	1 1 0 1 0 1 1 0 0 0 1

- Da gray a binario procedendo da sx a dx
 - ad ogni 1 si opera un'inversione del bit
 - ad ogni 0 si copia inalterato
- Da binario a Gray procedendo da sx a dx
 - quando i 2 bit sono uguali si scrive uno 0
 - quando sono diversi si scrive un 1



Rappresentazione grafica Cod. binario



Rappresentazione grafica Cod. Gray

