

La codifica dell'informazione

I sistemi di elaborazione operano al loro interno soltanto con segnali a due valori (binari).

I sistemi di elaborazione devono tuttavia scambiare informazioni con il mondo esterno in ingresso e in uscita.

CODIFICA DELL'INFORMAZIONE E CODICI BINARI

Tali informazioni devono essere comprensibili ad un operatore umano, e assumono prevalentemente la forma di caratteri alfanumerici (numeri decimali, lettere dell'alfabeto, simboli di punteggiatura, simboli matematici etc..).

Tale apparente incommunicabilità fra i due linguaggi, viene ricompota mediante l'adozione di opportune convenzioni (o *codici*), mediante le quali è possibile rappresentare in modo univoco un certo numero di simboli con configurazioni di bit prestabilite.

Andrea Bobbio

Anno Accademico 1996-1997

Unicità e lunghezza di un codice

Per rappresentare M simboli diversi (*parole*), mutuamente esclusivi, con un codice binario, la lunghezza m della sequenza di bit del codice deve soddisfare la seguente relazione:

$$m \geq \lceil \log_2 M \rceil$$

Codici decimali pesati a 4 bit

ZZ: Command not found. Per rappresentare le 10 cifre distinte dei numeri decimali, occorrono codici binari a 4 bit.

Un codice decimale si dice *pesato* se la relazione che intercorre tra la cifra decimale D da rappresentare e l'insieme delle cifre binarie ad esso associato è del tipo:

$$D = \sum_{i=0}^{m-1} b_i \cdot p_i$$

dove:

- m : numero di bit del codice
- b_i : generico bit del codice
- p_i : peso corrispondente.

Codice decimale BCD

Il codice decimale *BCD* (*Binary Coded Decimal*) è un codice pesato con pesi $(8 - 4 - 2 - 1)$.

Con questa codifica, ogni cifra decimale è rappresentata dal binario puro corrispondente, secondo la seguente tabella:

<i>Decimale</i>	<i>Codice BCD</i>			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Con questa codifica, le posizioni del codice da (1010) a (1111) non sono utilizzate.

Conversione decimale → binario BCD

Si converte ogni cifra decimale separatamente nel corrispondente codice *BCD* su 4 bit.

ESEMPIO:

Convertire il numero decimale $(5902)_{10}$ in codice binario *BCD*.

5	9	0	2
↓	↓	↓	↓
0101	1001	0000	0010

Conversione binario BCD → decimale

Si raggruppano le cifre binarie a 4 a 4 e si converte ciascun raggruppamento nella corrispondente cifra decimale secondo il codice *BCD*.

ESEMPIO:

Ricavare il valore decimale della sequenza di bit

0001100001110011

sapendo che corrisponde a una codifica *BCD*.

0	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
1		8		7		3									

Somme in binario BCD

L'operazione di somma aritmetica in codice binario *BCD* si esegue sommando, in binario, le cifre di ugual peso, con opportuni aggiustamenti dovuti alle peculiarità della codifica *BCD*.

ESEMPIO:

00100011+2310+
000100101210
001101013510

Il risultato è corretto perchè non c'è riporto su nessuna cifra.

Regola generale per somme in BCD

L'operazione di somma aritmetica in codice binario *BCD* si esegue sommando, in binario, le cifre di ugual peso. Se la somma fornisce una configurazione non permessa, oppure si ha riporto, occorre ancora sommare il valore 0110 (= 6₁₀). L'eventuale riporto si somma alle cifre di peso superiore.

ESEMPIO:

$$\begin{array}{r} 0010\ 1001\ +\ 29_{10}\ + \\ 0001\ 0010\ \\ \hline 0011\ 1011\ \\ \hline \end{array}$$

La cifra calcolata 1011 non fa parte delle cifre ammesse dal codice *BCD*. Il codice *BCD* salta dalla cifra 1001 = (9₁₀) alla cifra 0000 = (0₁₀) con un salto di 6 unità nella corrispondente codifica binaria.

Regola generale per somme in BCD

ESEMPIO (segue):

$$\begin{array}{r} 0010\ 1001\ +\ 29_{10}\ + \\ 0001\ 0010\ \\ \hline 0011\ 1011\ \\ \hline \end{array}$$

Alla cifra calcolata 1011, che non fa parte delle cifre ammesse dal codice *BCD*, si devono aggiungere le 6 unità mancanti, e sommare l'eventuale riporto alla cifra di peso superiore.

$$\begin{array}{r} \begin{array}{c} (1) \\ 0011\ 1011\ + \\ 0000\ 0110\ \\ \hline 0100\ 0001\ \end{array} \longrightarrow 41_{10} \end{array}$$

Regola generale per somme in BCD

ESEMPIO:

(1)

0 0 1 0

0 0 0 1

0 1 0 0

1 0 0 1

1 0 0 0

0 0 0 1

+

29₁₀

18₁₀

?₁₀

La cifra calcolata 0001, fa parte delle cifre ammesse dal codice *BCD*, ma il risultato è errato. Avendo ottenuto un riporto si devono aggiungere le 6 unità mancanti da 1001 = (9₁₀) alla cifra 0000 = (0₁₀), e sommare l'eventuale riporto alla cifra di peso superiore.

0 1 0 0

0 0 0 0

0 1 0 0

0 0 0 1

0 1 1 0

0 1 1 1

+

→

47₁₀

Codifica dell'informazione

12

Codice riflesso (Gray)

I codici *riflessi* sono codici in cui le rappresentazioni di due numeri consecutivi differiscono per un solo bit.

Trovano applicazione nelle conversioni analogico-digitale, in quanto, nel passaggio da un numero rappresentato al successivo, si evita di passare attraverso possibili configurazioni spurie (dovute al non simultaneo cambiamento di tutti i bit del codice).

Il codice *Gray* viene ottenuto seguendo questo procedimento:

1. Si scrivono i primi due bit {0, 1};

2. Si riflettono rispetto ad un ipotetico asse;

3. Si aggiunge uno 0 nella parte alta e un 1 nella parte bassa;

4. Si torna a 2) fino al raggiungimento della dimensione prefissata del codice.

Generazione codice Gray

0	0		00
1	1	<i>riflessione</i>	01
	1		11
	0		10

00	00		000
01	01		001
11	11	<i>aggiungo 0</i>	011
10	10		010
	10		110
	11	<i>aggiungo 1</i>	111
	01		101
	00		100

Generazione codice Gray (segue)

000	000	aggiungo 0	0000
001	001		0001
011	011		0011
010	010		0010
110	110		0110
111	111		0111
101	101		0101
100	100		0010
	100		1100
	101	aggiungo 1	1101
	111		1111
	110		1110
	010		1010
	011		1011
	001		1001
	000		1000

Codice Gray a 4 bit

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

Codici a barre

Codici alfanumerici

Quando, oltre alle cifre decimali, si vogliono anche codificare caratteri (maiuscoli e minuscoli), punteggiatura, simboli matematici etc... occorre estendere il numero di bit m del codice.

Se M è il numero totale di caratteri per cui si vuole definire un codice binario, occorre prevedere una lunghezza di codice m data dalla relazione:

$$m \geq \lceil \log_2 M \rceil$$

Codice ASCII

Il codice ASCII è di gran lunga il codice alfanumerico più diffuso per lo scambio di informazioni fra sistemi di elaborazione.

La sigla ASCII significa American Standard Code for Information Interchange.

Il codice ASCII costituisce di fatto uno standard per la codifica dell'informazione nei sistemi di elaborazione.

Codice ASCII

ASCII	simbolo	ASCII	simbolo	ASCII	simbolo
0000000	NUL	0101010	*	1010101	U
0000001	SOH	0101011	+	1010110	V
0000010	STX	0101100	,	1010111	W
0000011	ETX	0101101	-	1011000	X
0000100	EOT	0101110	.	1011001	Y
0000101	ENQ	0101111	/	1011010	Z
0000110	ACK	0110000	0	1011011	[
0000111	BEL	0110001	1	1011100	\
0001000	BS	0110010	2	1011101	]
0001001	HT	0110011	3	1011110	^
0001010	NL	0110100	4	1011111	_
0001011	VT	0110101	5	1100000	a
0001100	NP	0110110	6	1100001	b
0001101	CR	0110111	7	1100010	c
0001110	SO	0111000	8	1100011	d
0001111	SI	0111001	9	1100100	e
0010000	DLE	0111010	:	1100101	f
0010001	DC1	0111011	;	1100110	g
0010010	DC2	0111100	<	1100111	h
0010011	DC3	0111101	=	1101000	i
0010011	DC4	0111110	>	1101001	j
0010101	NAK	0111111	?	1101010	k
0010110	SYN	1000000	@	1101011	l
0010111	ETB	1000001	A	1101100	m
0011000	CAN	1000010	B	1101101	n
0011001	EM	1000011	C	1101110	o
0011010	SUB	1000100	D	1101111	p
0011011	ESC	1000101	E	1110000	q
0011100	FS	1000110	F	1110001	r
0011101	GS	1000111	G	1110010	s
0011110	RS	1001000	H	1110011	t
0011111	US	1001001	I	1110100	u
0100000	SP	1001010	J	1110101	v
0100001	!	1001011	K	1110110	w
0100010	"	1001100	L	1110111	x
0100011	#	1001101	M	1111000	y
0100100	\$	1001110	N	1111001	z
0100101	%	1001111	O	1111010	{
0100110	&	1010000	P	1111011	
0100111	,	1010001	Q	1111100	]
0101000	(	1010010	R	1111101	~
0101001	)	1010011	S	1111110	
		1010100	T	1111111	DEL

Codice ASCII esteso

Esiste una versione del codice ASCII che usa un ottavo bit (*1 byte*). Tale codice viene detto *Codice ASCII esteso*, ed è, ad esempio, usato nei personal computer IBM MS/DOS.

Codice ASCII esteso

Codice ASCII esteso

Codici Ridondanti

Per codificare M simboli distinti (*parole*) con un codice binario, occorrono $m \geq \lceil \log_2 M \rceil$ bit.

Un codice si dice ridondante, quando codifica gli M simboli distinti con $n = m + r$ bit, cioè usando r bit aggiuntivi rispetto agli m bit strettamente richiesti dalla codifica binaria.

L'aggiunta di bit di ridondanza permette di costruire codici che consentono di controllare eventuali errori di trasmissione.

Si hanno due tipi di codici ridondanti:

- ◇ Codici a rivelazione di errore - Consentono di individuare la presenza di un errore;
- ◇ Codici a correzione di errore - Consentono non solo di individuare la presenza di un errore, ma anche di identificarne la posizione in modo da poterlo correggere.

Codice BCD con bit di parità

Per costruire un codice *BCD* a rivelazione di errore, si può aggiungere ad ogni parola codice un bit ridondante ($r = 1$) detto di *parità*.

Il bit di *parità* viene posto a 0 o a 1, in modo tale che la somma degli uni nella parola codice sia pari:

<i>Decimale</i>	<i>Codice BCD</i>				<i>Bit parità</i>
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0

Conversione decimale → binario BCD con bit di parità

Si converte ogni cifra decimale separatamente nel corrispondente codice *BCD* e si aggiunge il quinto bit di parità come da tabella precedente.

ESEMPIO:

Convertire il numero decimale $(4765)_{10}$ in codice binario *BCD* con bit di parità.

4	7	6	5
↓	↓	↓	↓
0 1 0 0 1	0 1 1 1 1	0 1 1 0 0	0 1 0 1 0

Codici Ridondanti - Distanza di Hamming

La *distanza di Hamming* fra due parole codice, si ottiene contando il numero di bit diversi in posizioni corrispondenti:

1	0	1	0	0	1
0	0	1	1	1	1
<i>Distanza di Hamming=3</i>					

In un codice a rivelazione di errore la distanza di Hamming fra due parole codice deve essere ≥ 2 , in quanto un errore singolo deve produrre una sequenza di bit che non appartiene a nessuna parola codice.

In un codice a correzione di errore singolo la distanza di Hamming fra due parole codice deve essere ≥ 3 , in quanto un errore singolo deve produrre una sequenza di bit che ha distanza di Hamming 1 dalla parola originale e distanza di Hamming almeno 2 rispetto a ogni altra parola codice, in modo da identificare univocamente la parola originaria.

Codici di Hamming

Il *codice di Hamming* fornisce una procedura sistematica per generare codici ridondanti correttivi, tali che sia palese l'indicazione degli eventuali bit errati nella parola codice.

Verrà considerato solo il caso di *codice di Hamming* autocorrettivo per bit singolo (in grado cioè di correggere un eventuale errore su un solo bit).

Siano:	m	bit di parola
	r	bit di ridondanza
	$n = m + r$	bit di parola codice

Ognuna dell' 2^m parole legali, ha n parole codice errate a distanza di Hamming 1, ottenute cambiando un bit alla volta nella parola originaria.

Ognuna dell' 2^m parole legali richiede $(n + 1)$ configurazioni di bit dedicate. Per cui deve essere:

$$\begin{aligned} 2^n &\geq 2^m(n+1) \\ 2^m \cdot 2^r &\geq 2^m(m+r+1) \\ 2^r &\geq (m+r+1) \end{aligned}$$

Codice di Hamming Autocorrettivo (4+3)

Ogni bit di controllo deve garantire la parità sulle sequenze di bit della tabella:

r_1	r_2	m_1	r_3	m_2	m_3	m_4
1	2	3	4	5	6	7
☐	☐	☐	☐	☐	☐	☐
☐	☐	☐	☐	☐	☐	☐
☐	☐	☐	☐	☐	☐	☐

bit 1 → controlla la parità sui bit 1, 3, 5 e 7.

bit 2 → controlla la parità sui bit 2, 3, 6 e 7.

bit 4 → controlla la parità sui bit 4, 5, 6 e 7.

Il generico bit b è controllato dai bit r_i secondo la seguente tavola della verità:

r_3	r_2	r_1	b
X	X	X	1
			2
	X	X	3
X	X		4
X		X	5
X	X		6
X	X	X	7

Codice di Hamming Autocorrettivo (4+3)

La rilevazione e correzione di un eventuale bit errato avviene controllando il valore di parità dei bit di controllo.

Se il valore di parità del bit di controllo è corretto si pone a zero il valore nella tabella precedente; se non è corretto si pone a 1 il valore.

ESEMPIO:

Se si riscontra errata la parità nei bit r_1 e r_3 , il bit errato è $b = 5$. Infatti:

r_3	r_2	r_1	b
1	0	1	5

Se si riscontra errata la parità nei bit r_1 e r_2 , il bit errato è $b = 3$. Infatti:

r_3	r_2	r_1	b
0	1	1	3

Codice di Hamming Autocorrettivo (4+3)

ESEMPIO: Si determini il codice di Hamming per la parola su $m = 4$ bit:

1 0 1 1

Si mettano i bit assegnati nelle posizioni corrette:

r_1 r_2 m_1 r_3 m_2 m_3 m_4

1 2 3 4 5 6 7

1 0 0 1 1

Deve essere:

- $r_1 = 0$ perchè sia pari la sequenza 1, 3, 5, 7.
- $r_2 = 1$ perchè sia pari la sequenza 2, 3, 6, 7.
- $r_3 = 0$ perchè sia pari la sequenza 4, 5, 6, 7.

Il codice di Hamming completo per la parola data risulta:

r_1 r_2 m_1 r_3 m_2 m_3 m_4

1 2 3 4 5 6 7

0 1 1 0 0 1 1

Codice di Hamming Autocorrettivo (4+3)

ESEMPIO: Si supponga di avere la seguente parola codice, e di voler controllare se è corretta:

r_1 r_2 m_1 r_3 m_2 m_3 m_4

1 2 3 4 5 6 7

0 1 0 0 0 1 1

Si verifica la correttezza di parità dei bit di controllo, e si individua univocamente l'eventuale bit errato:

- La sequenza 1, 3, 5, 7 è dispari (r_1 errato).
- La sequenza 2, 3, 6, 7 è dispari (r_2 errato).
- La sequenza 4, 5, 6, 7 è pari (r_3 corretto).

La parola codice data contiene un bit errato che è il bit $b = 3$. Infatti:

$$\begin{array}{c|c|c} r_3 & r_2 & r_1 \\ \hline 0 & 1 & 1 \end{array} \bigg| \begin{array}{c} b \\ 3 \end{array}$$

La parola codice corretta risulta quindi essere:

1	2	3	4	5	6	7
0	1	1	0	0	1	1