

# Traccia

Determinare il Codice di Hamming del byte 00000001. Inoltre, indicare cosa succede a destinazione dovesse cambiare il primo bit controllore.

## Soluzione

### Determinazione del Codice di Hamming

Il Codice di Hamming aggiunge bit di controllo (o di parità) a un messaggio per consentire la correzione di errori a un bit. I bit di controllo vengono posizionati in posizioni che sono potenze di 2 (1, 2, 4, ecc.). Per il byte 00000001, il processo è il seguente:

#### 1. Posizionamento dei bit di controllo

Si inseriscono i bit di controllo nelle posizioni potenze di 2:

- p1: posizione 1
- p2: posizione 2
- p4: posizione 4
- p8: posizione 8

La sequenza diventa:

p1, p2, d1, p4, d2, d3, d4, p8, d5, d6, d7, d8,

che, per il messaggio 00000001, diventa:

p1, p2, 0, p4, 0, 0, 0, p8, 0, 0, 0, 1.

#### 2. Calcolo dei bit di controllo

I bit di controllo sono calcolati per garantire la parità pari dei gruppi di bit in cui essi appaiono.

- **Calcolo di p1:** p1 controlla i bit nelle posizioni 1, 3, 5, 7, 9, 11 (p1, d1, d2, d4, d5, d7):  
$$p1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$
- **Calcolo di p2:** p2 controlla i bit nelle posizioni 2, 3, 6, 7, 10, 11 (p2, d1, d3, d4, d6, d7):  
$$p2 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$

- **Calcolo di p4:** p4 controlla i bit nelle posizioni 4, 5, 6, 7, 12 (p4, d2, d3, d4, d8):  
 $p4 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$ .
- **Calcolo di p8:** p8 controlla i bit nelle posizioni 8, 9, 10, 11, 12 (p8, d5, d6, d7, d8):  
 $p8 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$ .

### 3. Codice di Hamming finale

Il codice di Hamming per il messaggio 00000001 è:

111100000001.

### Effetto di un errore sul primo bit controllore

Se il primo bit controllore (p1) cambia, il codice ricevuto avrà un errore in una posizione specifica.

A destinazione, il ricevitore calcola la sindrome di errore, che è il risultato della verifica della parità per ciascun bit di controllo. La sindrome indica la posizione del bit errato (in notazione binaria).

- Se p1 è errato, la sindrome avrà il primo bit impostato a 1.
- Il ricevitore correggerà l'errore invertendo il bit nella posizione indicata dalla sindrome.

Se non ci sono altri errori, il messaggio originale sarà recuperato correttamente.

## Traccia

Determinare l'Internet Checksum dei seguenti bit:

0001 0000 0101 0011 0011 1111 0011 0011

Quindi spiegare come avviene l'individuazione dell'errore se a destinazione dovesse cambiare il primo bit (in grassetto).

# Soluzione

## Calcolo dell'Internet Checksum

L'Internet Checksum è calcolato sommando tutti i segmenti di bit di 16 bit ciascuno e poi complementando il risultato.

I bit forniti vengono divisi in segmenti di 16 bit:

0001 0000 0101 0011 e 0011 1111 0011 0011.

### 1. Somma dei segmenti

Eseguiamo la somma binaria dei due segmenti:

$$\begin{aligned} &0001\ 0000\ 0101\ 0011 + 0011\ 1111\ 0011\ 0011 \\ &= 0101\ 1111\ 1000\ 0110. \end{aligned}$$

Poiché la somma non genera un overflow (cioè non supera i 16 bit), non è necessario aggiungere alcun riporto.

### 2. Complemento a uno

Il risultato della somma 0101 1111 1000 0110 viene complementato (bit a bit):

$$0101\ 1111\ 1000\ 0110 \rightarrow 1010\ 0000\ 0111\ 1001.$$

Pertanto, l'Internet Checksum è:

1010 0000 0111 1001.

## Individuazione dell'errore

A destinazione, il ricevitore effettua la somma del checksum ricevuto e dei dati ricevuti. Se il risultato è 1111 1111 1111 1111 (tutti 1), significa che i dati sono integri. Altrimenti, si è verificato un errore.

Se il primo bit (**in grassetto**) del messaggio originale cambia, il nuovo messaggio diventa:

1001 0000 0101 0011 0011 1111 0011 0011.

Eseguiamo nuovamente il calcolo del checksum:

## 1. Somma dei nuovi segmenti

I segmenti aggiornati sono 1001 0000 0101 0011 e 0011 1111 0011 0011. La somma binaria è:

$$\begin{aligned} &1001\ 0000\ 0101\ 0011 + 0011\ 1111\ 0011\ 0011 \\ &= 1100\ 1111\ 1000\ 0110. \end{aligned}$$

Non c'è overflow.

## 2. Complemento a uno

Il complemento del risultato 1100 1111 1000 0110 è:

$$1100\ 1111\ 1000\ 0110 \rightarrow 0011\ 0000\ 0111\ 1001.$$

A destinazione, il ricevitore rileverà che la somma del checksum calcolato e dei dati ricevuti non dà 1111 1111 1111 1111, indicando un errore nei dati trasmessi.

# Traccia

Determinare il Codice di Hamming del byte 00000111. Inoltre, indicare cosa succede a destinazione dovesse cambiare il terzo bit controllore.

# Soluzione

## Determinazione del Codice di Hamming

Il Codice di Hamming aggiunge bit di controllo (o di parità) a un messaggio per consentire la correzione di errori a un bit. I bit di controllo vengono posizionati in posizioni che sono potenze di 2 (1, 2, 4, ecc.). Per il byte 00000111, il processo è il seguente:

### 1. Posizionamento dei bit di controllo

Si inseriscono i bit di controllo nelle posizioni potenze di 2:

- p1: posizione 1
- p2: posizione 2
- p4: posizione 4

- p8: posizione 8

La sequenza diventa:

p1, p2, d1, p4, d2, d3, d4, p8, d5, d6, d7, d8,

che, per il messaggio 00000111, diventa:

p1, p2, 0, p4, 0, 0, 0, p8, 0, 1, 1, 1.

## 2. Calcolo dei bit di controllo

I bit di controllo sono calcolati per garantire la parità pari dei gruppi di bit in cui essi appaiono.

- **Calcolo di p1:** p1 controlla i bit nelle posizioni 1, 3, 5, 7, 9, 11 (p1, d1, d2, d4, d5, d7):  

$$p1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$
- **Calcolo di p2:** p2 controlla i bit nelle posizioni 2, 3, 6, 7, 10, 11 (p2, d1, d3, d4, d6, d7):  

$$p2 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 0.$$
- **Calcolo di p4:** p4 controlla i bit nelle posizioni 4, 5, 6, 7, 12 (p4, d2, d3, d4, d8):  

$$p4 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$
- **Calcolo di p8:** p8 controlla i bit nelle posizioni 8, 9, 10, 11, 12 (p8, d5, d6, d7, d8):  

$$p8 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1.$$

## 3. Codice di Hamming finale

Il codice di Hamming per il messaggio 00000111 è:

101100011111.

## Effetto di un errore sul terzo bit controllore (p4)

Se il terzo bit controllore (p4) cambia, il codice ricevuto avrà un errore in una posizione specifica.

A destinazione, il ricevitore calcola la sindrome di errore, che è il risultato della verifica della parità per ciascun bit di controllo. La sindrome indica la posizione del bit errato (in notazione binaria):

- Se p4 è errato, la sindrome avrà il terzo bit impostato a 1.

- Il ricevitore correggerà l'errore invertendo il bit nella posizione indicata dalla sindrome.

Se non ci sono altri errori, il messaggio originale sarà recuperato correttamente

## Traccia

Determinare l'Internet Checksum dei seguenti bit:

0001 0000 0101 0011 0011 1111 0011 0011

Quindi spiegare come avviene l'individuazione dell'errore se a destinazione dovesse cambiare il primo bit (in grassetto).

## Soluzione

### Calcolo dell'Internet Checksum

L'Internet Checksum è calcolato sommando tutti i segmenti di bit di 16 bit ciascuno e complementando il risultato.

I bit forniti vengono divisi in segmenti di 16 bit:

0001 0000 0101 0011 e 0011 1111 0011 0011.

#### 1. Somma dei segmenti

Eseguiamo la somma binaria dei due segmenti:

$$\begin{aligned} &0001\ 0000\ 0101\ 0011 + 0011\ 1111\ 0011\ 0011 \\ &= 0101\ 1111\ 1000\ 0110. \end{aligned}$$

Poiché la somma non genera un overflow (cioè non supera i 16 bit), non è necessario aggiungere alcun riporto.

#### 2. Complemento a uno

Il risultato della somma 0101 1111 1000 0110 viene complementato (bit a bit):

$$0101\ 1111\ 1000\ 0110 \rightarrow 1010\ 0000\ 0111\ 1001.$$

Pertanto, l'Internet Checksum è:

1010 0000 0111 1001.

## Individuazione dell'errore

A destinazione, il ricevitore effettua la somma del checksum ricevuto e dei dati ricevuti. Se il risultato è 1111 1111 1111 1111 (tutti 1), significa che i dati sono integri. Altrimenti, si è verificato un errore.

### Caso in cui il primo bit cambia

Se il primo bit (**in grassetto**) del messaggio originale cambia, il nuovo messaggio diventa:

1001 0000 0101 0011 0011 1111 0011 0011.

Eseguiamo nuovamente il calcolo del checksum:

#### 1. Somma dei nuovi segmenti

I segmenti aggiornati sono 1001 0000 0101 0011 e 0011 1111 0011 0011. La somma binaria è:

$$\begin{aligned} &1001\ 0000\ 0101\ 0011 + 0011\ 1111\ 0011\ 0011 \\ &= 1100\ 1111\ 1000\ 0110. \end{aligned}$$

Non c'è overflow.

#### 2. Complemento a uno

Il complemento del risultato 1100 1111 1000 0110 è:

$$1100\ 1111\ 1000\ 0110 \rightarrow 0011\ 0000\ 0111\ 1001.$$

## Rilevazione dell'errore

A destinazione, il ricevitore somma il nuovo checksum calcolato e i dati ricevuti. Il risultato non sarà 1111 1111 1111 1111, indicando che si è verificato un errore.