Codici di Correzione degli Errori

Architettura degli elaboratori

Laurea in Informatica

Docente: Nicolò Navarin

ECC – Error Correction Code

- Implementato nella memoria RAM di server e workstation
- Implementato in Cache in molte CPU (e.g. AMD ZEN, intel Xeon/Core)
- Single-Error-Correcting (SEC) code
  - Hamming Error Correcting Code
- SEC-DED: Single-Error-Correcting, Double Error Detecting

Codice di Hamming

a) 4 bit originali da salvare

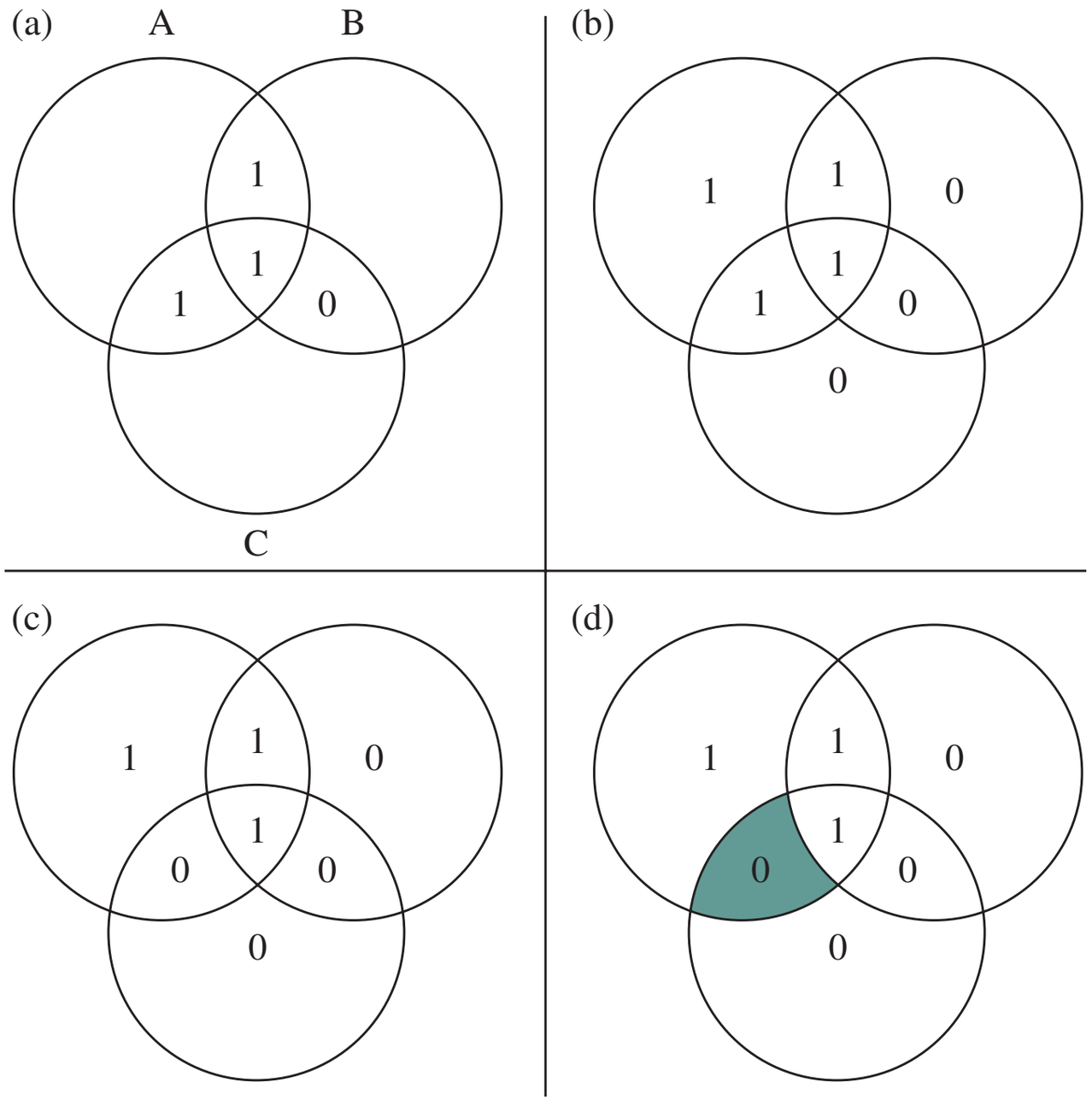

Figure 5.8 Hamming Error-Correcting Code

b) **Bit di controllo** per la correzione errori
Parity bit: la somma degli elementi di ogni insieme deve essere pari

c) Quando i bit vengono letti, può capitare un errore (switch da 0 a 1 o vice-versa) **sia nei bit del messaggio originale che nei bit di controllo**

c) Algoritmo di correzione errori permette di individuare quale bit è affetto da errore e correggerlo

Codice di Hamming

- Permette di **correggere** errori di un singolo bit
- Quanti **bit di controllo** servono per un input di M bit? Chiamiamo tale numero K
- Idea:
  - Ognuno degli M+K bit (dati o di controllo) totale possono essere affetti da errore
  - per la correzione, ho bisogno di poter indicare quale degli M+k bit è affetto da errore, + il caso in cui non ci sono errori
  - Totale: M+K+1 combinazioni
- Ci serve il minimo K tale che $2^K \geq M + K + 1$

Codice di Hamming

- Ci serve il minimo K tale che $2^K \geq M + K + 1$
- Ad esempio, per M=8
  - Servono K=4 bit, infatti $2^4 \geq 8 + 4 + 1$

Codice di Hamming: come posizionare I bit di controllo?

- Sindrome: XOR (bit a bit) tra i bit di controllo di un messsaggio letti dalla RAM e quelli calcolati a partire dal corpo del messaggio

- Desiderata:

  - se la sindrome contiene tutti 0, non ci sono stati errori

  - Se contiene un singolo bit a 1, l'errore è in un bit di controllo, quindi il messaggio rimane corretto

  - Se contiene più bit a 1, indicano la posizione del bit con errore (di cui deve essere eseguito lo switch)

- Soluzione: bit di controllo in posizioni potenza di 2

Codice di Hamming: come posizionare I bit di controllo?

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1
Word stored as	0	0	1	1	0	1	0	0	1	1	1	1
Word fetched as	0	0	1	1	0	1	1	0	1	1	1	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Check bit					0				0		0	1

XOR

Figure 5.10 Check Bit Calculation

0 1 1 0

- Ogni bit di controllo calcola la parità (corrispondente allo XOR) tra i bit il cui position number contiene un 1 nella stessa posizione del bit di controllo.

- Esempio: store e fetch della parola, c'è un errore al bit 6

ESERCIZIO 1 Codice di correzione di Hamming

Si supponga che una parola di dati da 8 bit memorizzata sia

11001010

Adottando l'algoritmo di Hamming, determinare quanti e quali bit di controllo verrebbero immagazzinati in memoria insieme alla parola di dati, ed in quale posizione.

ESERCIZIO 2 Codice di correzione di Hamming

Per la parola

00111001

i bit di controllo memorizzati sono 0111.

Si supponga che, quando la parola viene letta dalla memoria, i bit di controllo siano calcolati per essere 1101.

Quale parola di dati è letta dalla memoria?

ESERCIZIO 3 Codice di correzione di Hamming

Quanti bit di controllo sono necessari se il codice a correzione di errore di Hamming viene usato per rilevare errori di bit singoli in una parola di dati a 1024 bit?

ESERCIZIO 4 Codice di correzione di Hamming

Sviluppare un codice SEC per una parola di dati a 16 bit.
Generate il codice per la parola dati

0101000000111001