

Domande a risposta multipla

es2

Si consideri un codice di correzione di Hamming su 16 bit. Dire quale sequenza di bit è memorizzata in memoria se si devono memorizzare i seguenti 16 bit 0100111000101011 di dati:

- | | | | |
|----------------------------|--|----------------------------|-----------------------|
| <input type="checkbox"/> a | 010011100010101111110 | <input type="checkbox"/> b | 111110110100011010010 |
| <input type="checkbox"/> c | 110101000111001001111 | <input type="checkbox"/> d | 010010110001011011111 |
| <input type="checkbox"/> e | nessuna delle risposte precedenti è corretta | | |

Soluzione:

Di seguito si riporta la collocazione dei bit dati all'interno della tabella che permette di calcolare i bit di controllo per il codice di correzione di Hamming su 16 bit.

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
d16	d15	d14	d13	d12	c5	d11	d10	d9	d8	d7	d6	d5	c4	d4	d3	d2	c3	d1	c2	c1
0	1	0	0	1		1	1	0	0	0	1	0		1	0	1	1	1	1	1

Si ricorda che i bit di controllo necessari per $M = 16$ bit dati sono $K = 5$, in accordo con il valore minimo di K nell'equazione $2^K - 1 \geq M + K$. Inoltre, i bit di controllo sono collocati nelle posizioni che corrispondono a potenze di 2.

Di seguito si riporta il valore di ogni bit di controllo calcolato nel seguente modo:

- si rappresenta in binario (su 5 bit) la posizione di ogni bit dati;
- per il bit di controllo corrispondente alla posizione 2^i ($i \in \{0, 1, 2, 3, 4\}$) si selezionano i bit dati che hanno il bit i della rappresentazione binaria della posizione uguale a 1;
- il valore del bit di controllo è calcolato come il risultato dello XOR dei bit dati per lui selezionati.

21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
d16	d15	d14	d13	d12	c5	d11	d10	d9	d8	d7	d6	d5	c4	d4	d3	d2	c3	d1	c2	c1
0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	1	1	1	1	1

Pertanto la risposta corretta è d).

es3

Sia dato un disco rigido con le seguenti caratteristiche:

- capacità di 128GB;
- 1 piatto (2 facce);
- 16384 tracce per faccia e 8192 settori per traccia;
- velocità di rotazione di 10000 rpm;
- tempo medio di posizionamento della testina di 9,5 ms.

Il tempo totale medio per accedere a 128KB memorizzati in settori contigui su uno stesso cilindro è di circa

☐ a 12,59375 ms

☐ b 15,6875 ms

☐ c 12,5875 ms

☐ d 12,6875 ms

☐ e nessuna delle risposte precedenti è corretta
Soluzione:

Sappiamo che

$$T_S = 9,5 \text{ ms e } T_L = (1000/(10000/60))/2 = 3,0 \text{ ms}$$

e che il tempo totale di trasferimento è dato da

$$T = T_S + T_L + T_t,$$

con tempo di trasferimento (in millisecondi) dato dalla formula

$$T_t = \frac{b}{rN} \times 1000,$$

dove b è il numero di byte da trasferire, r è la velocità di rotazione al secondo del disco, N è il numero di byte contenuto in una traccia.

Il numero di byte per faccia sar dato dalla capacit totale del disco diviso il numero di facce

$$128GB/2 = 2^{37}/2^1 = 2^{36}.$$

Il numero di byte per traccia N sarà dato dalla capacità totale di una faccia diviso il numero di tracce ($16384 = 2^{14}$)

$$N = 2^{36}/2^{14} = 2^{22}.$$

Quindi

$$\begin{aligned} T_t &= [1000 \times 128KB]/[(10000/60) \times 2^{22}] \\ &= [1000 \times 2^{17}]/[(10000/60) \times 2^{22}] \\ &= 0,1875 \text{ ms}. \end{aligned}$$

Essendo i settori memorizzati in un cilindro, si possono leggere simultaneamente i settori posti su tracce collocate nella medesima posizione di facce diverse. Pertanto il tempo di trasferimento dei 128KB deve essere diviso per 2 (numero facce):

$$T = 9,5 + 3,0 + 0,1875/2 \approx 12,59375 \text{ ms}.$$

Pertanto la risposta corretta è a).

Esercizio

es8

Sia data la seguente sequenza di indirizzi in lettura (l) o scrittura (s) emessi dalla CPU e che la memoria abbia il contenuto esadecimale mostrato di seguito:

#	indirizzo (binario)	l/s	byte scritto (HEX)	ind	byte	ind	byte	ind	byte	ind	byte
1	000100001001	l		100	08	101	00	102	07	103	02
2	000100001101	s	AB	104	00	105	00	106	00	107	00
3	000100001110	s	39	108	AE	109	59	10A	AD	10B	23
4	000100011100	l		10C	A1	10D	42	10E	90	10F	75
5	000100001000	s	D4	110	B9	111	16	112	00	113	00
6	000100011110	l		114	0A	115	07	116	03	117	71
7	000100001010	s	98	118	3E	119	13	11A	71	11B	23
8	000100100001	l		11C	A1	11D	82	11E	90	11F	15
				120	FF	121	C6	122	AD	123	00
				124	E9	125	16	126	05	127	00

Si assuma che la dimensione di parola coincida con un byte, e la presenza di una cache di ampiezza 32B, dimensione di blocco 2B, inizialmente vuota, e ad associazione a 2 vie (politica di rimpiazzo LRU, politica di scrittura write-back e gestione dei miss in scrittura con la politica write allocate).

Si mostri come sia il contenuto della cache che il contenuto della memoria cambia.

Soluzione (da compilare)

- Indicare di seguito in quali campi (e la loro dimensione) gli indirizzi emessi dalla CPU sono suddivisi:

tag da 8 bit, set da 3 bit, word da 1 bit

- Indicare di seguito in quante linee/set la cache è suddivisa:

La cache è suddivisa in 4 set, ognuno di 2 linee da 2B

Indicare l'evoluzione della cache e della modifica della memoria nello schema sottostante:

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)	Modifica memoria $M[ind.] = contenuto$
109hex 000100001001	miss	set 100 linea 0 [AE 59] tag: 10hex	
10Dhex 000100001101	miss	set 110 linea 0 [A1 42] write allocate tag: 10hex ↓ [A1 AB]* tag: 10hex	
10Ehex 000100001110	miss	set 111 linea 0 [90 75] write allocate tag: 10hex ↓ [39 75]* tag: 10hex	

continuare nella pagina seguente

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)	Modifica memoria $M[ind.] = contenuto$
11Chex 000100011100	miss	set 110 linea 1 [A1 82] tag: 11hex	
108hex 000100001000	hit	set 100 linea 0 [D4 59]* tag: 10hex	
11Ehex 000100011110	miss	set 111 linea 1 [90 15] tag: 11hex	
10Ahex 000100001010	miss	set 101 linea 0 [AD 23] write allocate tag: 10hex ↓ [98 23]* tag: 10hex	
121hex 000100100001	miss	set 000 linea 0 [FF C6] tag: 12hex	