

Prima Parte

Da non compilare se si è superato il primo compito con un voto soddisfacente.
Non si può consegnare solo la prima parte.

Domande a risposta multipla

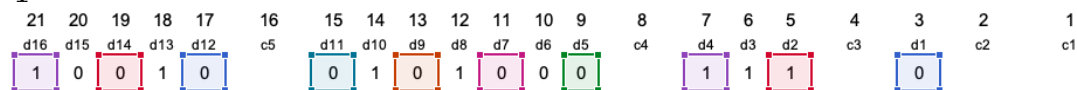
es2

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 1001001010001110 di dati:

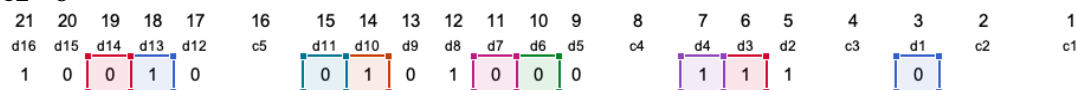
- | | | | |
|---------------------------------------|--|----------------------------|-------------------------|
| <input type="checkbox"/> a | 0000110010010100011110 | <input type="checkbox"/> b | 11010101010100001111001 |
| <input checked="" type="checkbox"/> c | 100100010100001110001 | <input type="checkbox"/> d | 100100101000111000001 |
| <input type="checkbox"/> e | nessuna delle risposte precedenti è corretta | | |

Spiegazione:

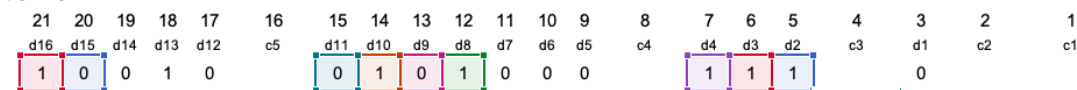
c1=1



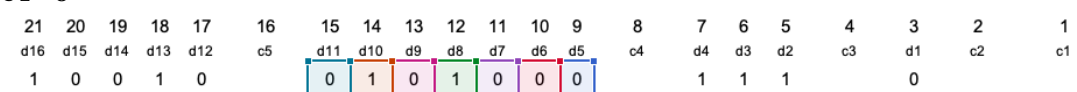
c2=0



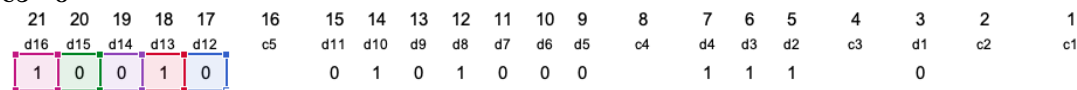
c3=0



c4=0



c5=0



es3

Sia dato un disco rigido con le seguenti caratteristiche:

- capacità di 1TB;
- 2 piatti (4 facce);
- 65536 tracce per faccia e 8192 settori per traccia;
- velocità di rotazione di 10000 rpm;
- tempo medio di posizionamento della testina di 6,5 ms.

Il tempo totale medio per trasferire (tempo di accesso totale medio, secondo il libro) 256KB memorizzati in una stessa traccia è di circa

☐ a 9,6875

☒ b 9,875 ms

☐ c 9,59375 ms

☐ d 10,25 ms

☐ e nessuna delle risposte precedenti è corretta

Spiegazione:

Sappiamo che

$$T_S = 6,5 \text{ ms e } T_L = (1000 / (10000/60)) / 2 \approx 3,0 \text{ ms}$$

e che il tempo totale di trasferimento è dato da

$$T = T_S + T_L + T_t$$

dove il tempo di trasferimento (in millisecondi) è dato dalla formula

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

b	#byte da trasferire
N	#byte per traccia
r	velocità rotazione (in rotazioni per sec.)

Il numero di byte per faccia sarà dato dalla capacità totale del disco diviso il numero di facce

$$1\text{TB} / 4 = 2^{40} / 2^2 = 2^{38}$$

Il numero di byte per traccia N sarà dato dalla capacità totale di una faccia diviso il numero di tracce ($65536 = 2^{16}$)

$$N = 2^{38} / 2^{16} = 2^{22}$$

Quindi

$$\begin{aligned} T_t &= [1000 \times 256\text{KB}] / [(10000/60) \times 2^{22}] \\ &= [1000 \times 2^{18}] / [(10000/60) \times 2^{22}] \\ &= 0,375 \text{ ms} \end{aligned}$$

Pertanto il tempo totale di accesso è

$$T = 6,5 + 3,0 + 0,375 = 9,875 \text{ ms}$$

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	000100001000	s	D4	100	08	101	0A	102	D7	103	02
2	000100001111	l		104	1F	105	00	106	80	107	E0
3	000100001110	s	DC	108	AE	109	73	10A	AF	10B	23
4	000100011101	s	9F	10C	A1	10D	42	10E	90	10F	75
5	000100011001	l		110	B9	111	16	112	FD	113	D0
6	000100011111	l		114	0A	115	07	116	03	117	71
7	000100000001	s	FF	118	3E	119	D3	11A	71	11B	23
8	000100000010	l		11C	A1	11D	8A	11E	90	11F	15
				120	F9	121	86	122	A0	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 4B, 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: 8 bit, set: 2 bit, word: 2

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

8 linee suddivise in 4 set

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 $M[ind.] = contenuto$
0001 0000 1000 1 0 8	m	SET 10 [AE 73 AF 23] linea 0 tag:00010000 w.a. v [D4 73 AF 23]*	
0001 0000 1111 1 0 F	m	SET 11 [A1 42 90 75] linea 0 tag:00010000	

continuare nella pagina seguente

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)	Modifica memoria $M[ind.] = contenuto$
0001 0000 1110 1 0 E	h	SET 11 [A1 42 DC 75]* linea 0 tag:00010000	
0001 0001 1101 1 1 D	m	SET 11 [A1 8A 90 15] linea 1 tag:00010001 l w.a. v [A1 9F 90 15]*	
0001 0001 1001 1 1 9	m	SET 10 [3E D3 71 23] linea 1 tag:00010001	
0001 0001 1110 1 1 F	h	SET 11 [A1 9F 90 15]* linea 1 tag:00010001	
0001 0000 0001 1 0 1	m	SET 00 [08 0A D7 02] linea 0 tag:00010000 l w.a. v [08 FF D7 02]*	
0001 0000 0010 1 0 2	h	SET 00 [08 FF D7 02]* linea 0 tag:00010000	

es8bis

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	parola scritta (HEX)	ind	byte	ind	byte	ind	byte	ind	byte
1	000100001000	s	D4FF032A	100	08	101	0A	102	D7	103	02
2	000100001100	l		104	1F	105	00	106	80	107	E0
3	000100011100	s	DC3E1189	108	AE	109	73	10A	AF	10B	23
4	000100010100	s	9F9E9B9C	10C	A1	10D	42	10E	90	10F	75
5	000100011000	l		110	B9	111	16	112	FD	113	D0
6	000100010000	l		114	0A	115	07	116	03	117	71
7	000100000000	s	FF112233	118	3E	119	D3	11A	71	11B	23
8	000100100100	l		11C	A1	11D	8A	11E	90	11F	15
				120	F9	121	86	122	A0	123	00
				124	E9	125	16	126	05	127	00

Si assuma che la dimensione di parola coincida con 4 byte, e la presenza di una cache di ampiezza 32B, dimensione di blocco di 2 parole, 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: 8 bit, set: 1 bit, word: 3 (di cui ultimi 2 indicano la posizione del byte nella parola)

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

4 linee suddivise in 2 set

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 $M[ind.] = \text{contenuto}$
0001 0000 1000 1 0 8	m	SET 1 [AE 73 AF 23 A1 42 90 75] linea 0 tag:00010000 l w.a. v [D4 FF 03 2A A1 42 90 75]*	
0001 0000 1100 1 0 C	h	SET 1 [D4 FF 03 2A A1 42 90 75]* linea 0 tag:00010000	

continuare nella pagina seguente

Indirizzo	hit/ miss	Cache (per ogni linea di cache indicare il contenuto del campo tag)	Modifica memoria $M[ind.] = contenuto$
0001 0001 1100 1 1 C	m	SET 1 [3E D3 71 23 A1 8A 90 15] linea 1 tag:00010001 I w.a. V [3E D3 71 23 DC 3E 11 89]*	
0001 0001 0100 1 1 4	m	SET 0 [B9 16 FD D0 0A 07 03 71] linea 0 tag:00010001 I w.a. V [B9 16 FD D0 9F 9E 9B 9C]*	
0001 0001 1000 1 1 8	h	SET 1 [3E D3 71 23 DC 3E 11 89]* linea 1 tag:00010001	
0001 0001 0000 1 1 0	h	SET 0 [B9 16 FD D0 9F 9E 9B 9C]* linea 0 tag:00010001	
0001 0000 0000 1 0 0	m	SET 0 [08 0A D7 02 1F 00 80 E0] linea 1 tag:00010000 I w.a. V [FF 11 22 33 1F 00 80 E0]*	
0001 0010 0100 1 2 4	m	SET 0 [F9 86 A0 00 E9 16 05 00] linea 0 (LRU) tag:00010010	$M[100-107] =$ [B9 16 FD D0 9F 9E 9B 9C]