

PRIMA PROVA INTERMEDIA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO
12 Novembre 2009

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (8 punti)

Progettare una rete sequenziale che presenti un ingresso X e una uscita Y posta ad 1 ogni volta che viene riconosciuta la sequenza 011010. Si richiede:

1. (4 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni usando flip flop di tipo D;
2. (4 punti) il calcolo delle forme minime per le variabili di eccitazione dei flip flop e per l'uscita, impiegando le mappe di Karnaugh.

ESERCIZIO 2 (13 punti)

1. (2 punti) Si consideri una memoria primaria costituita da 1024 parole e una memoria cache costituita da 256 parole. Si mostri il formato degli indirizzi nei casi in cui il metodo di indirizzamento sia rispettivamente:

- a. associativo su insiemi, con insiemi da 64 blocchi e blocchi di 2 parole;
- b. diretto, con blocchi di 64 parole.

Si utilizzi una strategia di rimpiazzamento FIFO qualora fosse necessario.

2. (6 punti) Si indichino lo stato finale della cache e l'hit rate per ognuno dei metodi precedenti, sapendo che il processore effettua le chiamate da 0 a 75 (compresi) per due volte (0, 1, ... 75, 0, 1, ... 74, 75).
3. (3 punti) Si confrontino gli hit rate ottenuti al punto 2 per le memorie cache di cui al punto 1.a e 1.b. Spiegare e motivare il confronto, anche con riferimento al principio di località.
4. (2 punti) Si illustrino brevemente le motivazioni che giustificano l'uso di una gerarchia di memoria.

ESERCIZIO 3 (6 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificati utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 001001101110 (il bit meno significativo è a sinistra), risultato della codifica di una parola di N bit secondo il codice di Hamming.

1. (2 punti) calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessario per una stringa di 12 bit;
2. (1 punto) scrivere la parola di N bit a partire dalla stringa data;
3. (3 punti) indicare eventuali errori nella stringa codificata, specificando quale dei bit è stato alterato.

ESERCIZIO 4 (6 punti)

Si supponga di disporre di tre macchine: a pila, a uno e a due indirizzi. Per ognuna di queste si abbiano le seguenti istruzioni:

A pila		A un indirizzo		A due indirizzi	
Istruzione	Semantica	Istruzione	Semantica	Istruzione	Semantica
PUSH X	$M[X] \rightarrow \text{push}$	STORE X	$\text{ACC} \rightarrow M[X]$	MOV X1,X2	$M[X1] \rightarrow M[X2]$
POP X	$\text{pop} \rightarrow M[X]$	LOAD X	$M[X] \rightarrow \text{ACC}$	ADD X1,X2	$M[X1] + M[X2] \rightarrow M[X2]$
ADD	$\text{pop} + \text{pop} \rightarrow \text{push}$	ADD X	$\text{ACC} + M[X] \rightarrow \text{ACC}$	DIV X1,X2	$M[X1] / M[X2] \rightarrow M[X2]$
DIV	$\text{pop} / \text{pop} \rightarrow \text{push}$	DIV X	$\text{ACC} / M[X] \rightarrow \text{ACC}$	MUL X1,X2	$M[X1] * M[X2] \rightarrow M[X2]$
MUL	$\text{pop} * \text{pop} \rightarrow \text{push}$	MUL X	$\text{ACC} * M[X] \rightarrow \text{ACC}$		

dove ACC indica il registro accumulatore e $M[X]$ indica il dato nella locazione di memoria X.

Facendo attenzione a non sovrascrivere i contenuti iniziali della memoria, si scriva, per ognuna delle tre macchine, la sequenza delle istruzioni necessarie per realizzare la seguente operazione:

$$Z = A * (B + C) / D$$

Suggerimento: nel caso "a due indirizzi" si utilizzi un registro P dove depositare i risultati parziali.

ESERCIZIO 1

Soluzione

1.

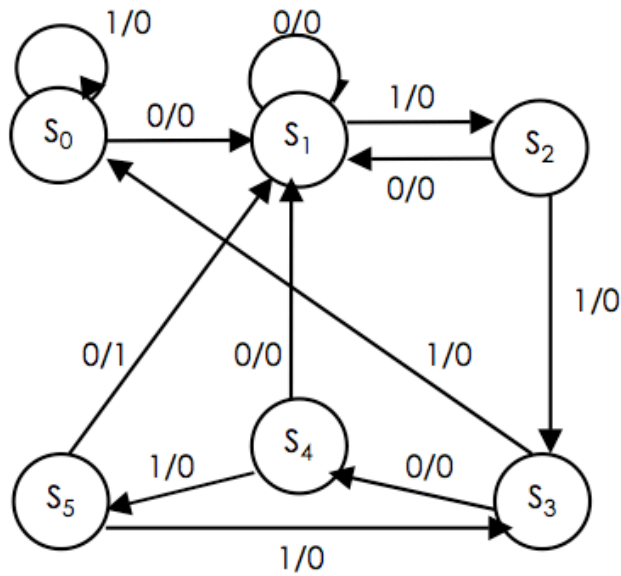


Tabella di flusso

Stato iniziale	Stato finale/uscita	
	x = 0	x = 1
S ₀	S ₁ /0	S ₀ /0
S ₁	S ₁ /0	S ₂ /0
S ₂	S ₁ /0	S ₃ /0
S ₃	S ₄ /0	S ₀ /0
S ₄	S ₁ /0	S ₅ /0
S ₅	S ₁ /1	S ₃ /0

Sono dunque sufficienti tre FF-D. La tabella delle transizioni è:

Stato presente	Stato futuro		Uscita	
	X=0	X=1	X=0	X=1
0 0 0	0 0 1	0 0 0	0	0
0 0 1	0 0 1	0 1 0	0	0
0 1 0	0 0 1	0 1 1	0	0
0 1 1	1 0 0	0 0 0	0	0
1 0 0	0 0 1	1 0 1	0	0
1 0 1	0 0 1	0 1 1	1	0
1 1 0	d d d	d d d	d	d
1 1 1	d d d	d d d	d	d

Indicando le uscite dei tre FF-D con A, B e C, scriviamo le mappe di Karnaugh relative ai valori futuri A', B', C' e all'uscita Y:

BC					BC					BC				
XA	00	01	11	10	XA	00	01	11	10	XA	00	01	11	10
00			1		00					00	1	1		1
01			d	d	01			d	d	01	1	1	d	d
11	1		d	d	11		1	d	d	11	1	1	d	d
10					10		1		1	10				1

$$A' = \overline{X}BC + X\overline{A}\overline{C}$$

$$B' = X\overline{B}\overline{C} + XBC$$

$$C' = A + \overline{X} \cdot \overline{B} + \overline{B}\overline{C}$$

<table> <tr> <td rowspan="2">BC XA</td><td colspan="4"></td></tr> <tr> <td>00</td><td>01</td><td>11</td><td>10</td></tr> <tr> <td>00</td><td></td><td></td><td></td><td></td></tr> <tr> <td>01</td><td></td><td>1</td><td>d</td><td>d</td></tr> <tr> <td>11</td><td></td><td></td><td>d</td><td>d</td></tr> <tr> <td>10</td><td></td><td></td><td></td><td></td></tr> </table>					BC XA					00	01	11	10	00					01		1	d	d	11			d	d	10				
BC XA																																	
	00	01	11	10																													
00																																	
01		1	d	d																													
11			d	d																													
10																																	

$$Y = \overline{X}A\overline{C}$$

ESERCIZIO 2

Soluzione

1. Il campo indirizzi avrà ampiezza 10 bit (devo indirizzare 1024 parole).
 - a. Avremo 2 insiemi da 64 blocchi ciascuno (la cache contiene 256 parole, quindi 128 blocchi, visto che ogni blocco contiene 2 parole):
< TAG 8 bit > < Set Index 1 bit > < Offset 1 bit >
 - b. Avremo 256 parole / (64 parole/blocco) = 4 blocchi, quindi:
< TAG 2 bit > < Cache Index 2 bit > < Offset 6 bit >

2.

a.

	0	1	2	3	...	74	75	0	1	...	75
BF	0	0	1	1		37	37	0	0		37
S.I.	0	0	1	1		1	1	0	0		1
Hit		X		X			X	X	X	X	X

Si noti come per la metà delle prime 76 chiamate (0...75) si abbia un hit.

Durante le 76 chiamate successive si avrà sempre hit, perché tutte le pagine sono già in cache (non c'è mai rimpiazzamento).

$$\text{hit rate} = (76/2 + 76)/(2*76) = 3/4 = 0.75$$

Stato finale della cache:

Set 0						Set 1					
Blocco 0	Blocco 1	...	18	...	63	0	1	...	18	...	63
Ind. 0, 1	Ind. 4, 5		72, 73			2, 3	6, 7		74, 75		

Occuperò 36 blocchi in tutto (perché sto chiamando 76 parole diverse), equamente ripartiti in 18 per insieme. I restanti saranno vuoti.

b.

	0	1	...	63	64	65		75	0	...	75
BF	0	0	0	0	1	1	1	1	0		4
C.I.	0	0	...	0	1	1	1	1	0		4
Hit		X	X	X		X	X	X	X	X	X

Per contare gli hit dobbiamo analizzare quello che succede nella due metà delle chiamate. Nella prima metà, dopo il primo miss si verificano 63 hit (una volta che il blocco è portato in cache, le 63 chiamate successive alla prima danno un hit), e questo succede 1 volta (fino a 63). Nell'ultimo caso si verifica un cache miss su 64 e successivamente 11 hit (da 65 a 75 inclusi).

Durante le 76 chiamate successive si avrà sempre hit, perché tutte le pagine sono già in cache (non c'è mai rimpiazzamento).

$$\text{hit rate} = (63 + 11 + 76)/(2*76) = 150/152 = 0.99$$

Stato finale della cache:

Blocco 0	1	...
0...63	64...75	

Occuperò completamente il blocco 0 e parte del blocco 1.

3. Si ottiene un hit rate maggiore utilizzando la cache in modalità diretta. Questo è sostanzialmente dovuto al fatto che i blocchi utilizzati al punto 1.b contengono più parole rispetto quelli usati dalla memoria al punto 1.a (64 contro 2) e, unitamente al fatto che le chiamate sono praticamente tutte sequenziali, questo permette di sfruttare meglio il principio di località.
Inoltre, in entrambi i casi le cache sono abbastanza capienti da contenere tutte le parole richieste, per cui non essendo necessario rimpiazzare alcun blocco, il vantaggio intrinseco della cache set-associativa rispetto alla diretta (vale a dire il fatto di poter allocare liberamente i blocchi dentro l'insieme dato) non viene sfruttato.
4. *Vedi dispense del corso.*

ESERCIZIO 3

Soluzione

1. Deve essere rispettata la condizione:

$$2^K \geq N + K + 1 \quad (1),$$

dove K è il numero di bit di controllo inseriti. Essendo $N + K = 12$, si evince dalla (1) che il numero minimo di bit di controllo richiesto è 4. Da cui $N = 8$.

2. Nella codifica di Hamming, la sequenza in ingresso presenta la seguente struttura:

c_0	c_1	b_0	c_2	b_1	b_2	b_3	c_3	b_4	b_5	b_6	b_7
0	0	1	0	0	1	1	0	1	1	1	0

Dove $c_0 \dots c_3$ sono i quattro bit costituenti il vettore di controllo, e $b_0 \dots b_7$ gli otto bit trasmessi. La sequenza ricevuta è 10111110.

3. Per verificare la presenza di un errore, dobbiamo ricalcolare il vettore di controllo a partire dalla sequenza ricevuta. Si ha:

$$c'_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 0$$

$$c'_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1$$

$$c'_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 0$$

$$c'_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 1$$

Il passo successivo è calcolare il vettore di errore dato dalla differenza dei vettori di controllo c e c' :

$$e_0 = c_0 \oplus c'_0 = 0$$

$$e_1 = c_1 \oplus c'_1 = 1$$

$$e_2 = c_2 \oplus c'_2 = 0$$

$$e_3 = c_3 \oplus c'_3 = 1$$

Poiché il vettore risultante 1010 non è nullo, vi è un errore nella stringa di 12 bit data e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato è quindi il decimo (b_5), e la parola corretta è 10111010.

ESERCIZIO 4
Soluzione

A pila	A un indirizzo	A due indirizzi
PUSH D PUSH C PUSH B ADD DIV PUSH A MUL POP Z	LOAD B ADD C DIV D MUL A STORE Z	MOV B,P ADD C,P MUL A,P MOV D,Z DIV P,Z