

SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI CALCOLATORI ELETTRONICI NUOVO E VECCHIO ORDINAMENTO DIDATTICO

15 Luglio 2005

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

ESERCIZIO 1 (NO: 8 punti – VO: 7 punti)

Si consideri una rete sequenziale avente un ingresso x e una uscita z . L'uscita $z = 1$ quando viene riconosciuta la sottosequenza 1110. Negli altri casi l'uscita $z = 0$.

- 1) (NO: 4 punti – VO: 3 punti) Disegnare il diagramma degli stati.
- 2) (2 punti) Codificare gli stati e scrivere la tabella di flusso. Si scriva poi la tabella delle transizioni qualora si usino flip-flop di tipo T.
- 3) (2 punti) Calcolare le forme minime per le variabili di eccitazione dei flip-flop e per l'uscita impiegando le mappe di Karnaugh.

Soluzione.

Diagramma stati:

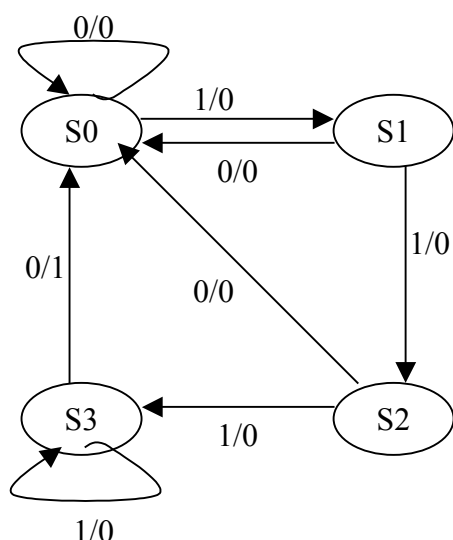


Tabella di flusso		
	Stato successivo/uscita	
	X=0	X=1
S0	S0/0	S1/0
S1	S0/0	S2/0
S2	S0/0	S3/0
S3	S0/1	S3/0

Codifica degli stati: $S0 \rightarrow 00$; $S1 \rightarrow 01$; $S2 \rightarrow 10$; $S3 \rightarrow 11$ (AB)

Tabella di eccitazione del flip flop T:

Q	Q'	T
0	0	0
0	1	1
1	0	1
1	1	0

Tabella delle transizioni							
A	B	X	A'	TA	B'	TB	Z
0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	0	1	0
1	0	0	0	1	0	0	0
1	0	1	1	0	1	1	0
1	1	0	0	1	0	1	1
1	1	1	1	0	1	0	0

AB \ X	00	01	11	10
0			1	1
1		1		

$$TA = A\bar{X} + \bar{A}BX$$

AB \ X	00	01	11	10
0		1	1	
1	1	1		1

$$TB = B\bar{X} + \bar{A}X + \bar{B}X$$

$$Z = AB\bar{X}$$

ESERCIZIO 2 (8 punti)

(a) (NO: 8 punti – VO: 6 punti) Un calcolatore ha un sistema di memoria virtuale a tre livelli costituito da: cache, memoria primaria e disco. La lettura di una parola che si trova già memorizzata nella cache richiede 15 ns. La lettura di una parola dalla memoria primaria e il suo trasferimento in cache richiedono complessivamente 40 ns. La lettura di una parola dal disco e il suo trasferimento in memoria primaria richiedono complessivamente 10 ms. La probabilità che una parola si trovi già in cache è pari a 0.95, mentre la probabilità che una parola si trovi in memoria primaria dato che non è presente nella cache è pari a 0.6. Calcolare il tempo medio di accesso al sistema di memoria.

(b) (solo VO: 2 punti) Enunciare e spiegare in modo chiaro e sintetico il principio di località.

Soluzione:

(a) Si devono considerare tre possibili situazioni:

Posizione della parola richiesta	Probabilità	Tempo di accesso totale
In cache	0.95	15 ns
Non in cache ma in memoria primaria	$(1 - 0.95) * 0.6 = (0.05) * 0.6 = 0.03$	$40ns + 15 ns = 55 ns$
Non in cache né in memoria primaria	$(1 - 0.95) * (1 - 0.6) = 0.05 * 0.4 = 0.02$	$10 ms + 40 ns + 15 ns = 10.000.055 ns$

Pertanto il tempo medio di accesso alla gerarchia di memoria è dato da:

$$T_m = 0.95 * 15 + 0.03 * 55 + 0.02 * 10.000.055 = 200.017 ns$$

(b) Vedi dispense del corso.

ESERCIZIO 3 (solo NO: 9 punti)

Implementare in Assembler MIPS una funzione che calcoli il prodotto scalare di due vettori v e w , dati i rispettivi indirizzi iniziali in $\$4$ e in $\$5$ e la relativa dimensione N in $\$6$ ($N > 0$). Il risultato sia posto in $\$7$. Vincolo: si usi il minimo numero di registri e il minimo numero di salti (condizionati e incondizionati).

Soluzione.

$\$9 \leftarrow v[i]; \$10 \leftarrow w[i]$

```

prodotto_scalare: addi $29, $29, -8
                  sw $9, 0($29)
                  sw $10, 4($29)
                  move $7, $0
loop:             lw $9, 0($4)
                  lw $10, 0($5)
                  mul $9, $9, $10
                  add $7, $7, $9
                  addi $6, $6, -1
                  addi $4, $4, 4
                  addi $5, $5, 4
                  bne $6, $0, loop
exit:             lw $9, 0($29)
                  lw $10, 4($29)
                  addi $29, $29, 8
                  jr $31

```

ESERCIZIO 3 (solo VO: 6 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 101001101110 (il bit meno significativo è a sinistra), risultato della codifica di una parola di N bit secondo il codice di Hamming.

- 1) (1 punto) calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessari.
- 2) (1 punto) scrivere la parola di N bit a partire dalla stringa data;
- 3) (4 punti) indicare eventuali errori nella stringa codificata, specificando quale dei bit è stato alterato.

Soluzione.

- 1) Deve venire 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
1	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' (ricordiamo che somma e differenza tra bit producono lo stesso risultato):

$$e_0 = c_0 \oplus c'_0 = 1$$

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

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

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

Poiché il vettore risultante 1011 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 l'undicesimo (b_6), e la parola corretta è 10111100.

ESERCIZIO 4 (NO: 8 punti – VO: 7 punti)

- 1) (NO: 5 punti – VO: 4 punti) Il sistema operativo di un calcolatore partiziona dinamicamente la memoria disponibile. Si supponga che ad un dato istante lo stato della memoria sia quello in figura, coi processi P1, ..., P4. Si supponga di voler allocare i seguenti processi: P5 da 75KB, P6 da 40KB, P7 da 15KB. Spiegare che cosa il sistema operativo deve fare per rendere possibile tale allocazione.

Memoria	
P1 (100KB)	
Vuoto (50KB)	
P2 (200KB)	
P3 (100KB)	
Vuoto (30KB)	
P4 (50 KB)	
Vuoto (50KB)	

- 2) (3 punti) La memoria di un calcolatore è gestita con una tecnica di 'paginazione su richiesta'. Si consideri la seguente richiesta di pagine: 2, 3, 4, 3, 2, 4, 3, 2, 4, 5, 6, 7, 5, 4, 2, 4, 5, 2, 6, 7, 2. Se la memoria contiene complessivamente quattro pagine calcolare il numero di 'page faults' nel caso di strategia di rimpiazzamento delle pagine LRU.

Soluzione:

1) La memoria è evidentemente *frammentata*. Poiché viene utilizzato un partizionamento dinamico, che dobbiamo considerare *rilocabile* in assenza di altre informazioni provenienti dal testo, per gestire efficientemente la memoria è necessario provvedere a *rilocare* le partizioni esistenti e *compattare* le lacune presenti in memoria. Come si può notare sono disponibili complessivamente 130KB, quindi tutti e tre i processi citati nel testo P5...P7 possono essere allocati senza problemi.

- 2) Page trace nel caso di strategia di rimpiazzamento delle pagine LRU (x = hit)

Tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Richieste	2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7	2
Pagine	2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7	2
		2	3	4	3	2	4	3	2	4	5	6	7	5	4	2	4	5	2	6	7
			2	2	4	3	2	4	3	2	4	5	6	7	5	5	2	4	5	2	6
										3	2	4	4	6	7	7	7	7	4	5	5
Hit				x	x	x	x	x	x				x	x		x	x	x			x

Hit ratio = 12/21; Page faults = 9

ESERCIZIO 5 (solo VO: 5 punti)

Illustrare le misure di "speedup" e "efficienza" usate per valutare le prestazioni di un'architettura multiprocessore.

Soluzione

Vedi dispense del corso.