

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO
26 Settembre 2007

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

ESERCIZIO 1 (NO: 10 punti – VO: 8 punti)

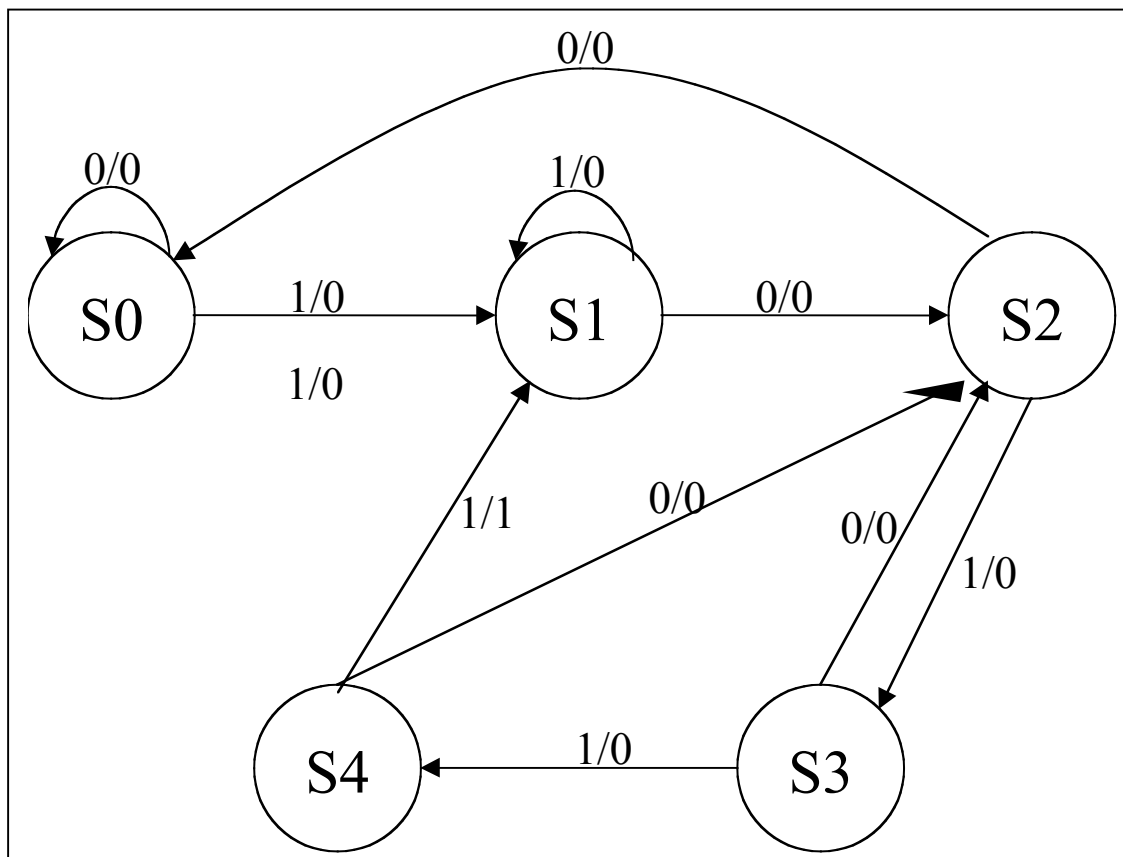
Progettare una rete sequenziale che presenti un ingresso X e un'uscita Z posta a 1 ogni volta che viene riconosciuta la sequenza 10111.

Si richiede:

- (NO: 6 punti – VO: 4 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
- (NO: 4 punti – VO: 2 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop JK. Calcolare anche la rete combinatoria per l'uscita Z.
- (solo VO: 2 punti) Realizzare un flip flop T a partire da un flip flop JK.

Soluzione.

Il diagramma degli stati è il seguente:



La tabella di flusso è data da:

Stato presente	Stato successivo/Uscita	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S3/0
S3	S2/0	S4/0
S4	S2/0	S1/1

Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente:

S0 → 0 0 0; ...; S4 → 1 0 0. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

A partire dalla tabella di eccitazione del flip flop JK:

Q	Q'	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

A	B	C	X	A'	Ja	Ka	B'	Jb	Kb	C'	Jc	Kc	Z
0	0	0	0	0	0	D	0	0	D	0	0	D	0
0	0	0	1	0	0	D	0	0	D	1	1	D	0
0	0	1	0	0	0	D	1	1	D	0	D	1	0
0	0	1	1	0	0	D	0	0	D	1	D	0	0
0	1	0	0	0	0	D	0	D	1	0	0	D	0
0	1	0	1	0	0	D	1	D	0	1	1	D	0
0	1	1	0	0	0	D	1	D	0	0	D	1	0
0	1	1	1	1	1	D	0	D	1	0	D	1	0
1	0	0	0	0	D	1	1	1	D	0	0	D	0
1	0	0	1	0	D	1	0	0	D	1	1	D	1
1	0	1	0	D	D	D	D	D	D	D	D	D	D
1	0	1	1	D	D	D	D	D	D	D	D	D	D
1	1	0	0	D	D	D	D	D	D	D	D	D	D
1	1	0	1	D	D	D	D	D	D	D	D	D	D
1	1	1	0	D	D	D	D	D	D	D	D	D	D
1	1	1	1	D	D	D	D	D	D	D	D	D	D

Ora possiamo disegnare le mappe di Karnaugh

AB		CX			
		00	01	11	10
CX	00			d	d
	01			d	d
	11		1	d	d
	10			d	d
$J_A = BCX$					

AB		CX			
		00	01	11	10
CX	00	d	d	d	1
	01	d	d	d	1
	11	d	d	d	d
	10	d	d	d	d
$K_A = 1$					

		AB			
		00	01	11	10
CX	00		d	d	1
	01		d	d	
	11		d	d	d
	10	1	d	d	d

$$J_B = C\bar{X} + A\bar{X}$$

		AB			
		00	01	11	10
CX	00	d	1	d	d
	01	d		d	d
	11	d	1	d	d
	10	d		d	d

$$K_B = \bar{C} \cdot \bar{X} + CX$$

		AB			
		00	01	11	10
CX	00			d	
	01	1	1	d	1
	11	d	d	d	d
	10	d	d	d	d

$$J_C = X$$

		AB			
		00	01	11	10
CX	00	d	d	d	d
	01	d	d	d	d
	11		1	d	d
	10	1	1	d	d

$$K_C = B + \bar{X}$$

Infine, per quanto riguarda l'uscita Z:

$$Z = A \cdot \bar{B} \cdot \bar{C} \cdot X$$

Volendo utilizzare anche i don't care:

		AB			
		00	01	11	10
CX	00			d	
	01			d	1
	11			d	d
	10			d	d

$$Z = AX$$

c)

Per realizzare un flip flop T da un JK, è sufficiente connettere fra loro gli ingressi J e K, come si può vedere confrontando la tabella delle transizioni del JK con quella del T:

	J	K	Q	Q'	T	Q	Q'
{	0	0	0	0	0	0	0
	0	0	1	1	0	1	1
	0	1	0	0	1	0	1
	0	1	1	0	1	1	0
	1	0	0	1			
	1	0	1	1			
{	1	1	0	1			
	1	1	1	0			

ESERCIZIO 2 (NO: 8 punti – VO: 7 punti)

Si consideri una gerarchia di memoria a tre livelli: cache, primaria, disco. Per mezzo di alcune prove vengono effettuate le seguenti misurazioni:

- numero totale di accessi alla memoria: 50
- numero di hit in cache: 45
- numero di hit in primaria: 49
- tempo medio di accesso in cache: 4 ns
- tempo medio di accesso in primaria: 100 ns
- tempo medio di accesso a disco: 50 ms

- a) (solo VO: 2 punti) Qual è lo scopo di un'architettura gerarchica di memoria? Come funziona?
- b) (NO: 4 punti – VO: 3 punti) Scrivere e spiegare la formula per il calcolo del tempo medio di accesso alla gerarchia;
- c) (NO: 4 punti – VO: 2 punti) Calcolare il tempo medio di accesso alla gerarchia.

Soluzione.

- a) Lo scopo di un'architettura gerarchica di memorie è quello di far intendere all'utilizzatore di disporre di una memoria di grandi dimensioni e veloce. Tale concetto è noto come "virtualizzazione della memoria". Per questo motivo il livello più basso della gerarchia è costituito dalla memoria più piccola e veloce (la cache), e a salire di livello troviamo memorie di dimensioni superiori ma più lente (il disco). Naturalmente, per massimizzare le prestazioni di una simile architettura è molto importante massimizzare il numero di successi al primo livello della gerarchia.

Il funzionamento segue il concetto che si vuole realizzare: quando viene richiesto un accesso alla memoria, il dato viene cercato nella memoria del livello più basso della gerarchia. Ciò richiede un tempo pari al tempo medio di accesso a tale memoria, nel nostro caso la cache. Se non viene trovato nessun dato in cache, si passa al livello successivo, che richiede un ulteriore tempo pari al tempo di accesso alla primaria. Infine, se il dato non viene trovato nemmeno in primaria, si passa alla memoria disco che richiede un tempo aggiuntivo pari appunto al tempo medio di accesso al disco.

- b) In accordo con quanto espresso al punto precedente, la formula per il calcolo del tempo medio di accesso alla gerarchia è la seguente:

$$\bar{T} = H_C T_C + (H_P - H_C)(T_P + T_C) + (H_D - H_P)(T_D + T_P + T_C)$$

T_C , T_P , T_D , sono i tempi medi di accesso a cache, primaria e disco.

H_C , H_P , H_D , sono gli *hit ratio* di cache, primaria e disco. Di solito $H_D = 1$, in quanto si assume che tutti i dati siano contenuti nella memoria del livello più alto della gerarchia.

H_i ($i=C,P,D$) è la probabilità di trovare un dato nella memoria di livello i della gerarchia.

La differenza $(H_P - H_C)$ è la probabilità di trovare un dato in primaria e di non trovarlo in cache.

Stesso significato, associato al livello superiore della gerarchia, ha la differenza $(H_D - H_P)$.

- c) Gli *hit ratio* di primaria e di cache si calcolano immediatamente dai dati forniti: $H_C = 45/50$; $H_P = 49/50$.

Sostituendo questi e gli altri valori nella formula indicata al punto precedente, si ottiene:

$$\bar{T} = \frac{45}{50} 4 + \frac{4}{50} 104 + \frac{1}{50} 50000104 = 1000014 ns$$

ESERCIZIO 3 (solo NO: 7 punti)

Scrivere il codice Assembler MIPS di una funzione che, dati l'indirizzo iniziale di un vettore v e la sua dimensione N (in $\$4$ e $\$5$ rispettivamente), scambi i valori $v[i]$ con $v[N-1-i]$, per $i=0, \dots, (N/2-1)$. E' dato il codice C della funzione corrispondente:

```
void inverti(int *v, int N)
{
    int i, Nmezzi, temp ;

    Nmezzi=div(N,2) ;
    for (i=0 ; i<Nmezzi; i++)
    {
        temp=v[i];
        v[i]=v[N-1-i];
        v[N-1-i]=temp;
    }
}
```

Nello scrivere la funzione, si faccia uso di una funzione esistente $\text{div}(a,b)$ che ricevendo a in $\$4$ e b in $\$5$, scrive in $\$6$ il valore a/b .

Soluzione.

$I \rightarrow \$8$; $\&v[0]+i*4 \rightarrow \$9$; $\&v[0]+(N-1-i)*4 \rightarrow \10

$v[i] \rightarrow \$11$; $v[N-1-i] \rightarrow \$12$

Copia di $\&v[0] \rightarrow \$13$; copia di $N \rightarrow \$14$

<pre>Inverti: addi \$29, \$29, -32 sw \$8, 0(\$29) sw \$9, 4(\$29) sw \$10, 8(\$29) sw \$11, 12(\$29) sw \$12, 16(\$29) sw \$13, 20(\$29) sw \$14, 24(\$29) sw \$31, 28(\$29) move \$13, \$4 move \$14, \$5 move \$4, \$5 addi \$5, \$0, 2 move \$5, \$14 subi \$14, \$14, 1 move \$4, \$13 jal div move \$8, \$0</pre>	<pre>for: beq \$8, \$6, exit subi \$10, \$14, \$8 muli \$9, \$8, 4 muli \$10, \$10, 4 add \$9, \$9, \$13 add \$10, \$10, \$13 lw \$11, 0(\$9) lw \$12, 0(\$10) sw \$11, 0(\$10) sw \$12, 0(\$9) addi \$8, \$8, 1 j for exit: lw \$8, 0(\$29) lw \$9, 4(\$29) lw \$10, 8(\$29) lw \$11, 12(\$29) lw \$12, 16(\$29) lw \$13, 20(\$29) lw \$14, 24(\$29) lw \$31, 28(\$29) addi \$29, \$29, 32 jr \$31</pre>
---	--

ESERCIZIO 3 (solo VO: 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	$\text{ACC} / M[X] \rightarrow \text{ACC}$		

ACC è il registro accumulatore della macchina a un indirizzo.

$M\{X\}$ indica il dato nella locazione di memoria X.

Si scriva, per ognuna delle tre macchine, la sequenza delle istruzioni necessarie per realizzare la seguente operazione:

$$Z = (A + B) / C.$$

Le lettere indicano le locazioni di memoria dove si trovano i dati.

Nella macchina a due indirizzi si faccia uso di una ulteriore locazione P dove introdurre i risultati parziali, pena la perdita dei dati iniziali nelle locazioni A, B o C.

Soluzione.

A pila	A un indirizzo	A due indirizzi	
PUSH C	LOAD A	ADD A,B	MOV B,P
PUSH A	ADD B	DIV B,C	ADD A,P
PUSH B	DIV C	MOV C,Z	MOV C,Z
ADD	STORE Z		DIV P,Z
DIV			
POP Z			
I contenuti delle locazioni A, B e C rimangono intatti.	I contenuti delle locazioni A, B e C rimangono intatti.	Vengono perduti i contenuti delle locazioni B e C.	I contenuti delle locazioni A, B e C rimangono intatti.

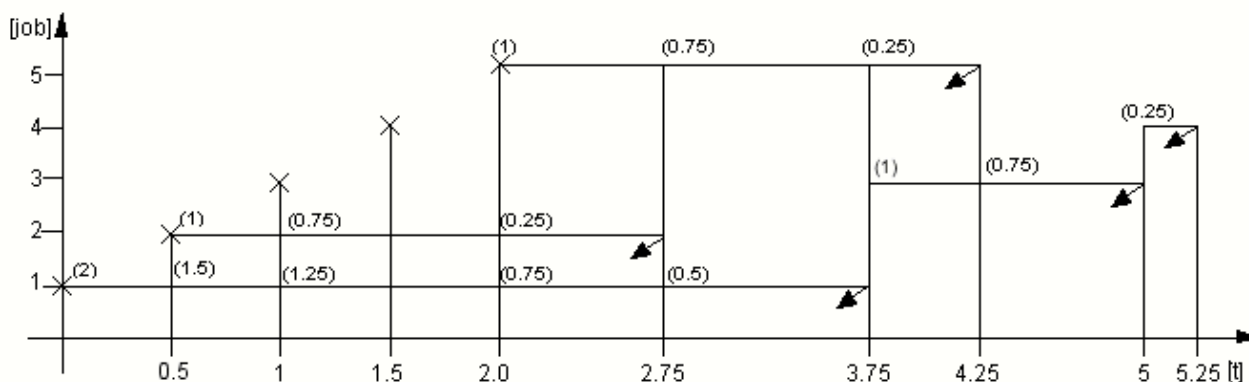
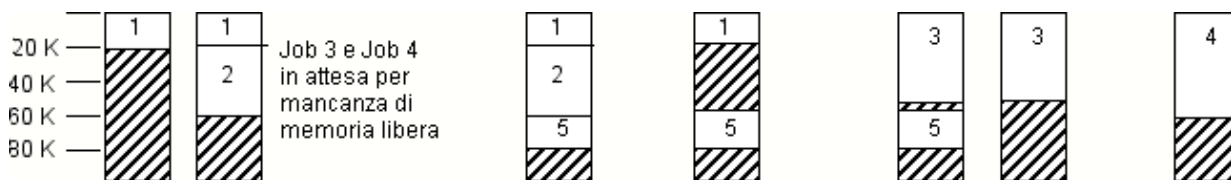
ESERCIZIO 4 (NO: 8 punti – VO: 7 punti)

Sia data la seguente lista di processi (si supponga che l'istante iniziale sia 0):

Job	Tempo di Arrivo	Tempo di CPU Richiesto	Richiesta di Memoria
1	0.0	2.0	20K
2	0.5	1.0	40K
3	1.0	1.0	50K
4	1.5	0.25	60K
5	2.0	1.0	20K

Il sistema ha 100K di memoria disponibile che viene gestita con partizioni dinamiche non rilocabili.

- (NO: 4 punti – VO: 3 punti) Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei job qualora si impieghi la politica di scheduling FIFO multiprogrammata "round robin".
- (NO: 4 punti – VO: 2 punti) Indicare, negli istanti in cui un job va in esecuzione e termina, la partizioni di memoria occupate da ciascun job e le partizioni libere, giustificando l'eventuale "messa in attesa" dei job.
- (solo VO: 2 punti) Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio.

Soluzione:

Job	t_{arrivo}	t_{start}	t_{finish}	Turnaround time	Weighted Turnaround time
1	0	0	3.75	3.75	1.875
2	0.5	0.5	2.75	2.25	2.25
3	1	3.75	5	4	4
4	1.5	5	5.25	3.75	15
5	2	2	4.25	2.25	2.25
Media				3.2	5.075

ESERCIZIO 5 (solo VO: 5 punti)

Descutere in modo chiaro e sintetico vantaggi e svantaggi delle architetture CISC rispetto alle RISC.

(Per la soluzione leggi dispense del corso)