

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO (5 CFU)
2 Febbraio 2010

NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (9 punti)

Si consideri una rete sequenziale avente un ingresso X e un'uscita Z. L'uscita Z = 1 quando viene riconosciuta la sottosequenza 010011. Negli altri casi l'uscita Z = 0.

1. (4 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 D.
3. (3 punti) Calcolare le forme minime per le variabili di eccitazione dei flip flop e per l'uscita, impiegando le mappe di Karnaugh.

ESERCIZIO 2 (10 punti)

Si consideri una memoria primaria costituita da 32 parole e una memoria cache costituita da 8 parole, con blocchi da 2 parole. Si considerino i tre metodi di indirizzamento diretto, associativo su insiemi a 2 vie e completamente associativo.

1. (3 punti) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nei tre casi.
2. (5 punti) Indicare lo stato finale della cache nel caso il processore acceda in sequenza alle parole di indirizzo da 0 a 3 e da 16 a 19 in questo ordine, e ripeta la sequenza di accesso per due volte, nei tre casi di indirizzamento indicati (gli indirizzi sono in decimale).
3. (2 punti) Nell'ipotesi che il tempo di accesso alla memoria primaria sia 10 volte il tempo di accesso alla memoria cache, indicare di quanto si riduce il tempo di accesso alla memoria con l'uso della cache nei tre casi di indirizzamento considerati nelle domande precedenti.

ESERCIZIO 3 (7 punti)

Indicare in modo chiaro e sintetico qual è la funzione svolta da ognuno dei tre frammenti MIPS seguenti. Si accetta come risposta anche la traduzione in linguaggio C dei frammenti. In ogni caso sia chiaramente indicata la funzione dei registri e delle aree di memoria.

Frammento 1.	Frammento 2.	Frammento 3.
<code>muli \$15, \$16, 4</code> <code>lw \$14, 1024(\$15)</code> <code>add \$14, \$14, \$13</code> <code>subi \$15, \$15, 4</code> <code>sw \$14, 1024(\$15)</code>	<code>move \$16, \$0</code> <code>lui \$16, 60₁₀</code> <code>addi \$16, \$16, 2305₁₀</code>	<code>Loop: lw \$8, save(\$9)</code> <code>add \$9, \$9, \$20</code> <code>bne \$8, \$21, Loop</code>

ESERCIZIO 4 (7 punti)

Si consideri una architettura basata su accumulatore. Il contenuto della memoria a un certo istante è il seguente (per semplicità usiamo la notazione decimale): l'indirizzo 100 contiene il valore 300; l'indirizzo 250 contiene 150; l'indirizzo 300 contiene 500; l'indirizzo 400 contiene 250. Dire quale sarà il contenuto dell'accumulatore dopo le seguenti istruzioni:

Istruzioni	Modalità indirizz. (1)	Modalità indirizz. (2)	Modalità indirizz. (3)
LOAD 100	immediato	diretto	diretto
ADD 400	diretto	diretto	indiretto

nei tre casi riportati nelle tre colonne.

ESERCIZIO 1

Soluzione.

Grafo degli stati

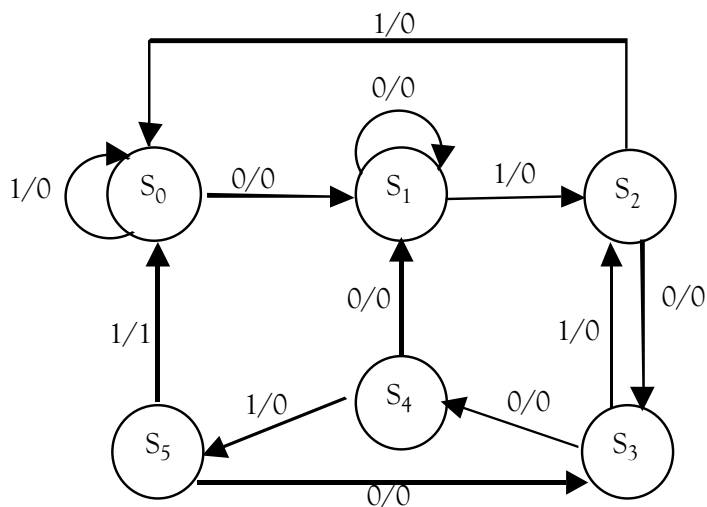


Tabella di flusso

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 ₃ /0	S ₀ /0
S ₃	S ₄ /0	S ₂ /0
S ₄	S ₁ /0	S ₅ /0
S ₅	S ₃ /0	S ₀ /1

Codifica degli stati (3 bit - Y₂Y₁Y₀)

S₀ → 000 S₁ → 001
S₂ → 010 S₃ → 011
S₄ → 100 S₅ → 101

Tabella delle transizioni (FF-D) Y' indica lo stato futuro

Y ₂	Y ₁	Y ₀	x	Y' ₂	D ₂	Y' ₁	D ₁	Y' ₀	D ₀	z
0	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1	0
0	0	1	1	0	0	1	1	0	0	0
0	1	0	0	0	0	1	1	1	1	0
0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1	0	0	0
1	0	0	0	0	0	0	0	1	1	0
1	0	0	1	1	1	0	0	1	1	0
1	0	1	0	0	0	1	1	1	1	0
1	0	1	1	0	0	0	0	0	0	1
1	1	0	0	d	d	d	d	d	d	d
1	1	1	1	d	d	d	d	d	d	d

Y ₂ Y ₁	Y ₀ x			
	00	01	11	10
00			d	
01			d	1
11			d	
10		1	d	

$D_2 = Y_2 \bar{Y}_0 x + Y_1 Y_0 \bar{x}$

Y ₂ Y ₁	Y ₀ x			
	00	01	11	10
00		1	d	
01			d	
11	1	1	d	
10			d	1

$D_1 = Y_1 Y_0 x + Y_2 Y_0 x + Y_2 Y_0 \bar{x}$

Y ₂ Y ₁	Y ₀ x			
	00	01	11	10
00	1	1	d	1
01			d	1
11			d	
10	1		d	1

$D_0 = Y_1 x + Y_0 x + Y_2 Y_0$

Y ₂ Y ₁	Y ₀ x			
	00	01	11	10
00			d	
01			d	
11			d	1
10			d	

$z = Y_2 Y_0 x$

ESERCIZIO 2

Soluzione:

1.

Metodo diretto: < TAG 2 bit > < Index 2 bit > < Offset 1 bit >
Metodo set-associativo: < TAG 3 bit > < Index 1 bit > < Offset 1 bit >
Metodo associativo: < Block Frame (TAG) 4 bit > < Offset 1 bit >

2.

Stato finale della cache nei tre casi:

Linea	Diretto	Insieme	Set-Associativo	Associativo
0	16	0	0	0
	17		1	1
1	18	1	16	2
	19		17	3
2			2	16
			3	17
3			18	18
			19	19

Le parole 0-1 e 2-3 presentano index pari a 0 e 1 rispettivamente. Le parole 16-17 e 18-19 presentano lo stesso valore di index delle precedenti. Col metodo diretto vengono pertanto mappate nelle stesse linee di cache (la prima e la seconda, come mostrato in figura). Col metodo set-associativo, invece, pur presentando lo stesso index, possono venire mappate in blocchi eventualmente liberi dello stesso insieme. Infine, col metodo associativo l'index non è presente (NB è l'analogo del metodo associativo su insiemi a quattro vie) quindi i blocchi di primaria vengono allocati nei corrispondenti blocchi liberi di cache, nell'ordine di chiamata.

3.

Per calcolare l'entità della riduzione bisogna valutare il rapporto fra tempo medio di accesso alla gerarchia cache-primaria (quindi il tempo medio richiesto con uso della cache) e tempo medio di accesso della sola primaria (senza uso della cache).

Indicando con T il tempo medio di accesso alla gerarchia cache-primaria della domanda precedente e con T_p il tempo di accesso in primaria, l'entità della riduzione è data da:

$$1 - T/T_p$$

Il tempo medio di accesso alla gerarchia cache-primaria è pari a:

$$T = T_c + (1 - H) * T_p$$

Con T_c : tempo di accesso cache e H = hit ratio cache.

Poiché $T_p = 10 * T_c$, si ha

$$T = T_c + (1 - H) * 10 * T_c$$

Il rapporto T/T_p è dunque funzione di H ed è pari a:

$$1 - (1 + (1 - H) * 10)/10$$

Poiché si ha:

Metodo diretto: $H = 8/16 = 0.5$

Metodo set-associativo e associativo: $H = (4 + 8) / 16 = 0.75$

Si ha che con l'indirizzamento diretto la riduzione è pari a 0.4 e negli altri due casi la riduzione è pari a 0.65. In altri termini la riduzione è del 40% nel primo caso e del 65% negli altri due.

ESERCIZIO 3

Soluzione.

Frammento 1.

Si osserva che il registro \$15 è utilizzato come offset per l'accesso ad una locazione di memoria con indirizzo base pari a 1024. Il registro \$14 è invece utilizzato per memorizzare i dati nella locazione di indirizzo $1024 + [\text{\$15}]$ ($[\text{\$x}]$ significa contenuto del registro $\text{\$x}$). A questo valore viene sottratto il contenuto di \$13, e successivamente viene memorizzato in un'area di memoria adiacente alla precedente, in quanto \$15 viene decrementato di un valore pari a 4.

Per tradurre questo frammento in C, è sufficiente ad esempio assegnare a \$13 una certa variabile y e utilizzare un tipo di dato indicizzato come un vettore indicato simbolicamente come $x[i]$, dove i è l'indice memorizzato in \$16. Quindi in C si ottiene l'istruzione:

$$x[i-1] = x[i] + y;$$

Frammento 2.

La presenza dell'istruzione `lui` indica chiaramente che si sta memorizzando nel registro \$16 un valore in modalità immediata eccedente il numero di bit a disposizione. Tale valore equivale a 60_{10} nei 16 bit più significativi, a 2305_{10} in quelli meno significativi. Il valore che si tenta di caricare è dunque: 3934465_{10} . In C il frammento si può tradurre:

$$x = 3934465;$$

Frammento 3.

La presenza dell'istruzione di salto condizionato `bne` e il nome "loop" utilizzato per l'area di memoria contenente l'istruzione a cui il salto fa riferimento suggeriscono inequivocabilmente che il frammento implementi un ciclo di istruzioni. La condizione di terminazione del ciclo si può ricavare osservando i valori contenuti nei registri \$8 e \$21 nell'istruzione `bne`. Il valore in \$21 è utilizzato solo in `bne`, si può quindi ipotizzare si tratti di un valore di riferimento. Il valore in \$8 viene prelevato da un'area di memoria con indirizzo iniziale `save`, mentre l'offset è dato dal valore in \$9. Ricapitolando: la condizione di terminazione di ciclo prevede che il valore contenuto in \$8 e indicizzato da \$9 sia diverso da un valore contenuto in \$21. L'indice viene aggiornato attraverso il valore in \$20. In C il frammento si può tradurre come un ciclo `do-while`:

$$\text{while}(\text{save}[h] \neq k) \quad h = h + c;$$

ESERCIZIO 4

Soluzione.

Nel caso di indirizzamento immediato, il campo indirizzi contiene l'operando; nel caso di indirizzamento diretto il campo indirizzi contiene l'indirizzo di memoria ove risiede l'operando; infine nel caso di indirizzamento indiretto il campo indirizzi contiene l'indirizzo di memoria che a sua volta contiene l'indirizzo di memoria ove risiede l'operando.

Caso (1):

'LOAD 100' carica il valore 100 nell'accumulatore $AC \leftarrow 100$

'ADD 400' somma al contenuto dell'accumulatore il valore in memoria all'indirizzo 400: $AC \leftarrow AC + M[400]$, $AC \leftarrow 100 + 250$. Il contenuto finale dell'accumulatore è 350

Caso (2):

'LOAD 100' carica nell'accumulatore il valore in memoria all'indirizzo 100: $AC \leftarrow M[100]$, $AC \leftarrow 300$

'ADD 400' somma al contenuto dell'accumulatore il valore in memoria all'indirizzo 400: $AC \leftarrow AC + M[400]$, $AC \leftarrow 300 + 250$. Il contenuto finale dell'accumulatore è 550.

Caso (3):

'LOAD 100' carica nell'accumulatore il valore in memoria all'indirizzo 100: $AC \leftarrow M[100]$, $AC \leftarrow 300$

'ADD 400' somma al contenuto dell'accumulatore il valore in memoria all'indirizzo contenuto in 400:

$AC \leftarrow AC + M[M[400]]$, $AC \leftarrow 300 + M[250]$, $AC \leftarrow 300 + 150$. Il contenuto finale dell'accumulatore è 450.