

PROVA SCRITTA DEL MODULO DI
CALCOLATORI ELETTRONICI
CORSO DI LAUREA IN INGEGNERIA ELETTRICA ED ELETTRONICA, INGEGNERIA BIOMEDICA
(CFU: 5, 6, 7)
15 gennaio 2015

NOME:

COGNOME:

MATRICOLA:

CFU:

ESERCIZIO 1 (5-6 CFU: 8 punti; 7 CFU: 7 punti)

- (a) (5-6 CFU: 6 punti; 7 CFU: 5 punti) Si progetti una rete logica sequenziale in grado di riconoscere, entro una sequenza di bit X , la sottostringa 1000, ponendo a 1 il corrispondente bit di uscita solo quando si abbia il riconoscimento della sottostringa data. Si implementi la rete utilizzando FF-JK e si minimizzino le espressioni delle funzioni di transizione dello stato mediante le mappe di Karnaugh. Non è necessario disegnare il circuito della rete.
- (b) (2 punti) Derivare un FF-D da un FF-JK motivando chiaramente la risposta.

ESERCIZIO 2 (5-6 CFU: 8 punti; 7 CFU: 7 punti)

Si consideri una memoria primaria costituita da 512 B e una memoria cache costituita da 16 B, con blocchi di 4 B. E' possibile indirizzare il singolo byte.

1. (5-6 CFU: 2 punti; 7 CFU: 1 punto) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nel caso di indirizzamento associativo su insiemi a due vie.
2. (3 punti) Supponendo la cache inizialmente vuota, si considerino le chiamate ai seguenti indirizzi (espressi in decimale): da 0 a 7, da 40 a 47 in questo ordine, per due volte consecutive. Si indichi il contenuto della cache, ovvero quali byte occupano le linee di cache, dopo l'ultima chiamata, utilizzando il metodo di indirizzamento al punto 1.
3. (1 punto) Si calcoli l'hit ratio di cache sulla base delle chiamate al punto precedente.
4. (2 punti) Si consideri infine una gerarchia di memoria a due livelli costituita da cache e primaria. Se il tempo medio di accesso non deve essere superiore a 5 nsec, e il tempo di accesso in cache è pari a 2 nsec mentre quello in primaria è pari a 40 nsec, il metodo di indirizzamento ai punti precedenti soddisfa le specifiche? Se no, ci sono alternative, sempre considerando blocchi da 4 B? Esprimere tutti i tempi **in nanosecondi**.

ESERCIZIO 3 (10 punti)

- (1) (8 punti) Si scriva un frammento di programma Assembler MIPS che consideri un vettore di interi senza segno u (di indirizzo iniziale 1000) la cui dimensione è contenuta nella locazione di indirizzo N , e copi in un altro vettore v (di indirizzo iniziale 5000) i valori di u che risultano strettamente maggiori di un intero contenuto nella locazione di indirizzo M . Alla fine del programma il registro $\$10$ deve contenere il numero di valori copiati. Tale valore dovrà essere trasferito nella locazione di indirizzo K .
- (2) (1 punto) Scrivere quanti interi possono essere memorizzati nel vettore u (ovvero il valore massimo che può essere contenuto nella parola di indirizzo N).
- (3) (1 punto) Illustrare la suddivisione dei campi codice operativo ed operandi per l'istruzione Assembly MIPS `lw`.

ESERCIZIO 4 (5-6 CFU: 7 punti; 7 CFU: 6 punti)

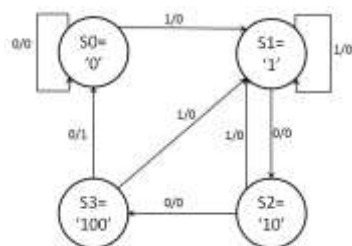
- (1) (5-6 CFU: 4 punti; 7 CFU: 3 punti) Indicare e disegnare uno schema di arbitraggio con tre linee disponibili per cinque periferiche indicando la funzionalità di ciascuna linea. Disegnare e spiegare in dettaglio, passaggio per passaggio, il protocollo di comunicazione che utilizzi tali linee.
- (2) (3 punti) Ipotizzare che i trasferimenti da calcolatore a periferica siano gestiti mediante modulo DMA. La linea dati è di 64 bit e la velocità del processore è pari a 1.4 GHz. Qual è la massima velocità di trasferimento ammissibile, espressa in MB/s?

ESERCIZIO 5 (solo 7 CFU: 3 punti)

Dare la definizione di *throughput rate*.

ESERCIZIO 1

Per prima cosa, occorre scrivere il grafo degli stati che corrisponde al seguente:



Utilizzando FF-JK:

A	B	X	A'	JA	KA	B'	JB	KB	Z
0	0	0	0	0	D	0	0	D	0
0	0	1	0	0	D	1	1	D	0
0	1	0	1	1	D	0	D	1	0
0	1	1	0	0	D	1	D	0	0
1	0	0	1	D	0	1	1	D	0
1	0	1	0	D	1	1	1	D	0
1	1	0	0	D	1	0	D	1	0
1	1	1	0	D	1	1	D	0	1

Semplificando le espressioni delle rete logiche per la transizione dello stato:

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

$$J_A = B\bar{X}$$

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

$$K_A = X + B$$

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

$$J_B = A + X$$

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

$$K_B = \bar{X}$$

L'espressione dell'uscita è invece data da:

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

Per passare da un FF-JK ad un D è sufficiente, per definizione, connettere le uscite J e K' tra loro. Ciò si evince anche dalla tabella di transizione da un JK ad un D:

Q(t)	D	Q(t+1)	J	K
0	0	0	0	D
0	1	1	1	D
1	1	1	D	0
1	0	0	D	1

ESERCIZIO 2

1. Essendo la memoria primaria costituita da $512 \text{ B} = 2^9 \text{ B}$, l'indirizzamento è a 9 bit. Di questi, i due meno significativi compongono l'offset (i blocchi sono di quattro parole). Il numero di linee di cache è invece 4 ($16 \text{ B}/4 \text{ B}$). Nel metodo set associativo a due vie un bit di index è sufficiente per indirizzare i set. Si ottiene dunque un TAG di 6 bit.
2. Per quanto riguarda il metodo set associativo a due vie, il procedimento inizia calcolando la divisione $0/4=0$ con resto 0. L'ulteriore divisione è $0/2=0$ con resto 0. Quindi questa parola, e tutto il resto del blocco (parole 0-3) viene allocato nella prima linea libera dell'insieme 0. Il secondo gruppo di quattro parole, dalla 4 alla 7, viene invece allocato nella prima linea libera dell'insieme 1.
Il secondo nucleo inizia con la parola $40/4=10$ con resto 0. Per calcolare il set index: $10/2=5$ con resto 0. Anch'essa, assieme alle parole 41-43 componenti il blocco, vengono memorizzate nel set 0, ma nella seconda linea libera. Analogamente le parole dalla 45 alla 47 verranno memorizzate nel set 1, seconda linea libera. In questo caso non è stata necessaria alcuna sovrascrittura per cui la seconda chiamata dei numeri di parole le troverà tutte presenti in cache, secondo lo stato finale:

SET ASSOCIATIVO	Offset di parola			
Index di set	0	1	2	3
0	0	1	2	3
	40	41	42	43
1	4	5	6	7
	44	45	46	47

3. Abbiamo 3 hit per ciascuno dei blocchi chiamati al primo ciclo, mentre al secondo ciclo abbiamo tutti hit. Si ottiene dunque:

$$H_c = \frac{3 \cdot 4 + 16}{16 \cdot 2} = \frac{7}{8} = 0.875$$

4. La formula del tempo medio di accesso alla gerarchia di memoria data è:

$$\bar{T} = T_c + (1 - H_c) \cdot T_p$$

Sostituendo i valori di H_c trovati nel precedente esercizio si ha:

$$\bar{T}_{SetAssociativo} = 2 + \frac{1}{8} \cdot 40 = 2 + 5 = 7nsec$$

Il metodo di indirizzamento dato non soddisfa la specifica di un tempo di accesso medio inferiore od uguale a 5 nsec. Considerando che, fissata la dimensione dei blocchi, non v'è altro modo di intervenire sul hit ratio, altri metodi di indirizzamento nel caso in questione non otterrebbero un risultato migliore.

ESERCIZIO 3

Nella nostra soluzione, indicheremo con \$8 il registro che indicizzerà gli elementi di u (indicato genericamente come i), e con \$10 quello che indicizza gli elementi di v (indicato come j). Al termine del frammento, \$10, secondo specifica, conterrà il numero di valori che da u saranno stati copiati in v.

In \$9 depositeremo il valore che si trova nella locazione N, ed in \$5 il valore che si trova nella locazione M. Nel registro \$14 depositeremo inoltre il valore u[j].

Il registro \$12 conterrà, alternativamente, i valori $i*4$, $j*4$ e il valore di verità del confronto $M < u[i]$.

Possiamo dunque sviluppare il nostro frammento come segue:

frammento:	lw \$9, N(\$0)	#\$9 ← N
	lw \$5, M(\$0)	#\$5 ← M
	move \$8, \$0	#\$8 ← 0
	move \$10, \$0	#\$10 ← 0
loop:	beq \$8, \$9, exit_loop	#se i == N esci dal loop
	mul \$12, \$8, 4	#calcolo l'offset per i
	lw \$14, 1000(\$12)	#\$14 ← u[i]
	slt \$12, \$5, \$14	#\$8 ← M < u[i]
	beq \$12, \$0, aggiorna_i	#If confronto fallisce aggiorna, else...
	mul \$12, \$10, 4	#calcolo l'offset per j e memorizzo v[j]
	sw \$14, 5000(\$12)	
	addi \$10, \$10, 1	
aggiorna_i:	addi \$8, \$8, 1	#aggiorno i
	j loop	
exit_loop:	sw \$10, K(\$0)	#salvo K

Poiché i contenuti dei due vettori non devono sovrapporsi, dalla differenza fra le locazioni 5000 e 1000 si ottengono 4000 byte. Poiché ogni parola MIPS è di 4 byte abbiamo che al massimo potrà essere contenuto il valore 1000 nella locazione N.

Le istruzioni Assembly MIPS hanno tutte lunghezza pari a 32 bit.

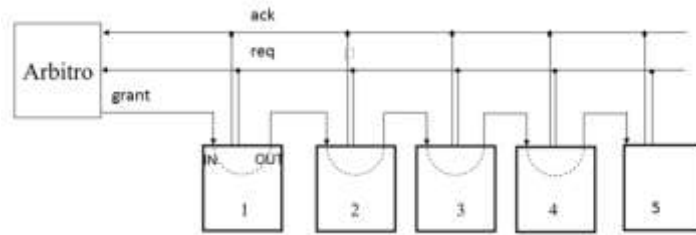
L'organizzazione dei campi per l'istruzione lw è:



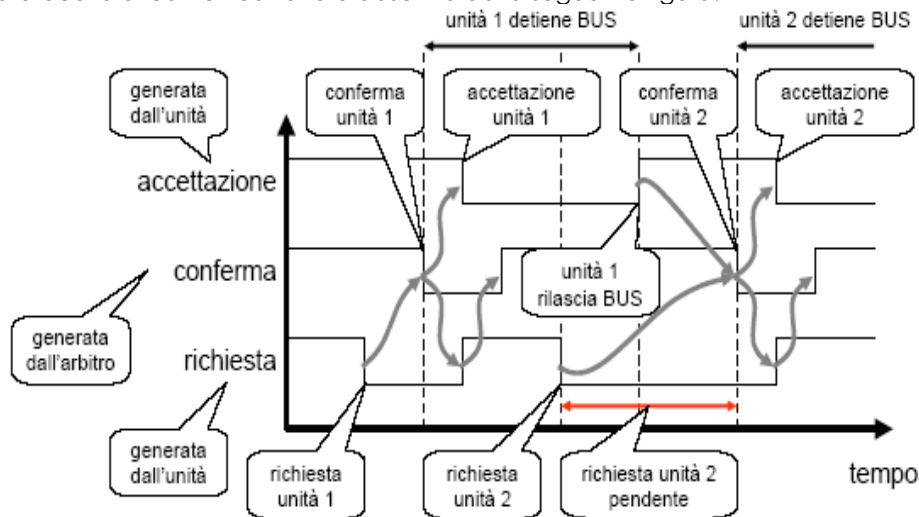
Dove op sta per codice operativo, address è l'indirizzo-etichetta rs è il registro che contiene l'informazione da trasferire e rt è il registro che contiene l'indirizzo che andrà sommato al valore presente in address.

ESERCIZIO 4

- 1) Un possibile schema di arbitraggio è quello centralizzato con tre linee GRANT, ACK, REQ, la prima delle quali attivata dall'arbitro e le altre due dalle periferiche. Lo schema corrispondente è il seguente:



Il relativo protocollo di comunicazione è descritto dalla seguente figura:



In questa figura i segnali sono "attivi bassi". Quando l'Unità 1, ad esempio, compie la richiesta, essa deve attendere dall'arbitro il "grant" ovvero la conferma che, non appena avvenuta, consente alla periferica di attivare la linea "ack" (che significa anche bus "impegnato"). Successivamente l'arbitro disattiva la linea "grant" e la periferica disattiva la linea "req". L'attivazione della linea "ack" impedisce che la richiesta di una seconda periferica, ad esempio la 2, possa essere presa in considerazione dal bus fino a che quella correntemente in possesso non lo rilasci disattivando la linea "ack". A quel punto, l'arbitro riattiverà la linea "grant" consentendo alla prima periferica del festone che ha richiesto il bus di utilizzarlo. Si ricordi infatti che il segnale "grant" si propaga sulle periferiche nell'ordine cablato dalla "daisy chain".

- 2) Nel caso di massima velocità, il DMA deve lavorare in modalità "block transfer" che garantisce un trasferimento dati ad ogni ciclo di clock.
Poiché la linea dati consente il trasferimento di 8 byte, il calcolo è dato da : $1.4 \cdot 10^9$ cicli al secondo $\cdot 8$ byte/ciclo = $11,2 \cdot 10^9 / 2^{20} = 10681$ MB/sec

ESERCIZIO 5

Vedi dispense del corso.