

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO
 18 Giugno 2007

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

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

Si vogliono sommare i seguenti quattro numeri a 8 bit:

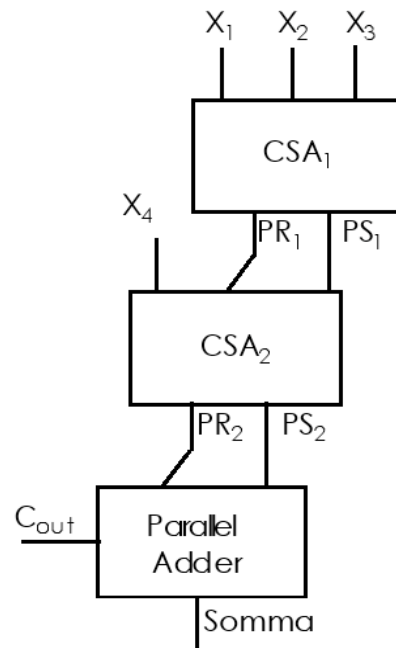
$X_1 = 00100111$, $X_2 = 10101101$, $X_3 = 01100001$, $X_4 = 11010010$.

- (NO: 8 punti, VO: 4 punti) Disegnare lo schema che permette di eseguire tale somma usando due addizionatori del tipo "Carry Save Adder" ed un "Parallel Adder" finale. Gli addizionatori "Carry Save Adder" lavorano su tre operandi. Precisare il valore assunto dalla pseudosomma e dal pseudoriporto all'uscita del primo e del secondo "Carry Save Adder" e le operazioni eseguite per ottenere la somma finale.
- (solo VO: 3 punti) Nell'ipotesi che d sia il ritardo per il calcolo di somma e riporto per un full-adder, calcolare il ritardo del presente sistema, indicando il vantaggio che deriva rispetto all'uso di soli parallel adder.

Soluzione

1.

Schema:



$$PS_i = A_i \oplus B_i \oplus C_i$$

$$PR_i = A_i B_i + A_i C_i + B_i C_i$$

$$S = PS + 2PR$$

X1	00100111	PS1	11101011	PS2	01110011
X2	10101101	2PR1	01001010	2PR2	10010101
X3	01100001	X4	11010010	<hr/> SUM00001000 CARRY 1	
PS1	11101011	PS2	01110011		
PR1	00100101	PR2	11001010		

- Il tempo di ritardo del sistema con l'uso di tre Parallel Adder è $8d+8d+8d=24d$, mentre con il Carry Save Adder equivale a $8d+d+d=10d$.

ESERCIZIO 2 (NO: 9 punti- VO: 8 punti)

- (3 punti) Spiegare in modo chiaro e sintetico le caratteristiche della memoria cache. Indicare quali sono i principali metodi di indirizzamento e spiegarne in dettaglio il funzionamento, specificando come vengono gestiti i campi di un indirizzo di memoria primaria per l'allocazione del blocco in cache.
- (NO: 6 punti – VO: 4 punti) Si consideri una memoria primaria costituita da 128 parole e una memoria cache costituita da 16 parole. Si considerino le seguenti chiamate ad altrettante parole (indirizzi espressi in decimale): 52, 24, 1, 44, 27, 13, 53, 25, 11, 45. Si mostri il contenuto finale della cache nei casi in cui il metodo di indirizzamento sia rispettivamente:
 - (2 punti) completamente associativo, con blocchi di quattro parole;
 - (2 punti) diretto con blocchi di due parole;
 - (2 punti) associativo su insiemi a due vie, e blocchi di due parole.
 Si ipotizzi inoltre che ove necessario il metodo di rimpiazzamento sia FIFO.
- (solo VO: 1 punto) Si calcoli l'hit ratio per ognuno dei metodi al punto precedente.

Soluzione

1.

- Completamente associativo con blocchi di 4 parole.
32 blocchi in memoria primaria e 4 in cache.

	<i>BlockFrame</i> 5 bit					<i>Offset</i> 2 bit				
	52	24	1	44	27	13	53	25	11	45
BF	13	6	0	11	6	3	13	6	2	11
Miss	X	X	X	X		X	X	X	X	X
Hit					X					

Hit Ratio = 1/10

Stato finale della cache:

0				1				2				3			
44	45	46	47	52	53	54	55	24	25	26	27	8	9	10	11

- Diretto con blocchi di 2 parole.
64 blocchi in memoria primaria 8 in cache.

	<i>Tag</i> 3 bit		<i>Cache Index</i> 3 bit		<i>Offset</i> 1 bit					
	52	24	1	44	27	13	53	25	11	45
BF	26	12	0	22	13	6	26	12	5	22
CI	2	4	0	6	5	6	2	4	5	6
Miss	X	X	X	X	X	X			X	X
Hit							X	X		

Hit Ratio = 2/10

Stato finale della cache:

0		1		2		3		4		5		6		7	
0	1			52	53			24	25	10	11	44	45		

- c. Associativo a 2 vie, con blocchi di 2 parole.
64 blocchi in memoria primaria e 2 insiemi in cache di 4 blocchi ciascuno.

Tag	Set Index	Offset
5 bit	1 bit	1 bit

	52	24	1	44	27	13	53	25	11	45
BF	26	12	0	22	13	6	26	12	5	22
SI	0	0	0	0	1	0	0	0	1	0
Miss	X	X	X	X	X	X	X	X	X	
Hit										X

Hit Ratio = 1/10

Stato finale della cache:

0								1							
0		1		2		3		0		1		2		3	
12	13	52	53	24	25	44	45	26	27	10	11				

ESERCIZIO 3 (solo NO: 8 punti)

Si scriva una funzione Assembler MIPS, chiamata `media_k` che, ricevendo in ingresso l'indirizzo iniziale di un vettore in `$4` di dimensione `N` (in `$5`), ed un intero `k < N` (in `$6`), metta in `$7` il valore $(v[k] + v[k+1])/2$. Se $k == N-1$, metta in `$7` il valore $(v[N-1] + v[0])/2$. Si faccia uso di una funzione esistente `media` che, ricevendo due interi in `$4` e `$5`, mette in `$6` il loro valor medio. Si abbia cura di non avere alterato i contenuti di `$4`, `$5`, `$6` all'uscita dalla funzione `media_k`.

Soluzione.

`$8 ← k*4; $9 ← &v[0]+k*4 ; $4-$6 vengono copiati in $10-$12`

```
max_vettori:  addi $29, $29, -24
               sw $8, 0($29)
               sw $9, 4($29)
               sw $10, 8($29)
               sw $11, 12($29)
               sw $12, 16($29)
               sw $31, 20($29)
               move $10, $4
               move $11, $5
               move $12, $6
               muli $8, $6, 4
               add $9, $8, $4
               addi $6, $6, 1
               lw $4, 0($9)
               beq $5, $6, cicla
               lw $5, 4($9)
cont:         jal media
               move $7, $6
               move $4, $10
               move $5, $11
               move $6, $12
               lw $8, 0($29)
               lw $9, 4($29)
               lw $10, 8($29)
               lw $11, 12($29)
               lw $12, 16($29)
               lw $31, 20($29)
               addi $29, $29, 24
               jr $31
cicla:        lw $5, 0($10)
               j cont
```

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	$\text{pop}(2) / \text{pop}(1) \rightarrow \text{push}$	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	$\text{ACC} * M[X] \rightarrow \text{ACC}$		

ACC è il registro accumulatore; $M[X]$ 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: si usi un registro P dove depositare i risultati parziali)

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

ESERCIZIO 4 (8 punti)

1. (NO: 8 punti – VO: 6 punti) Si consideri un sistema operativo che implementi una memoria virtuale a 4 segmenti di uguale dimensione, contenente 16 pagine di 4 KB.

- (2 punti) Qual è la dimensione di ciascun segmento?
- (NO: 6 punti – VO: 4 punti) Indicare dimensione e struttura degli indirizzi logici per ogni Byte indirizzato.

2. (solo VO: 2 punti) Spiegare in modo chiaro e sintetico le differenze fra paginazione e segmentazione.

Soluzione

- Dimensione di ogni segmento: $16 \times 4 \text{ KB} = 64 \text{ KB}$
 -

Segmento	Pagina	Offset
2 bit	4 bit	12 bit

- Vedi le dispense del corso.

ESERCIZIO 5 (solo VO: 5 punti)

Illustrare sinteticamente le caratteristiche e il formato tipico di una istruzione RISC,

Soluzione

Vedi le dispense del corso.