

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO
16 Gennaio 2003**

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

Nota: Gli studenti che consegnano e totalizzano un punteggio inferiore a 16 non potranno sostenere la prossima prova scritta.

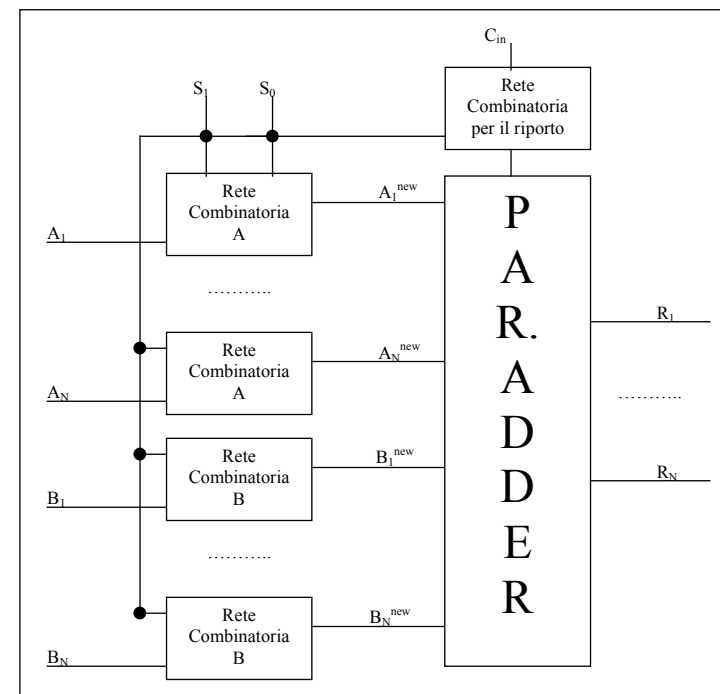
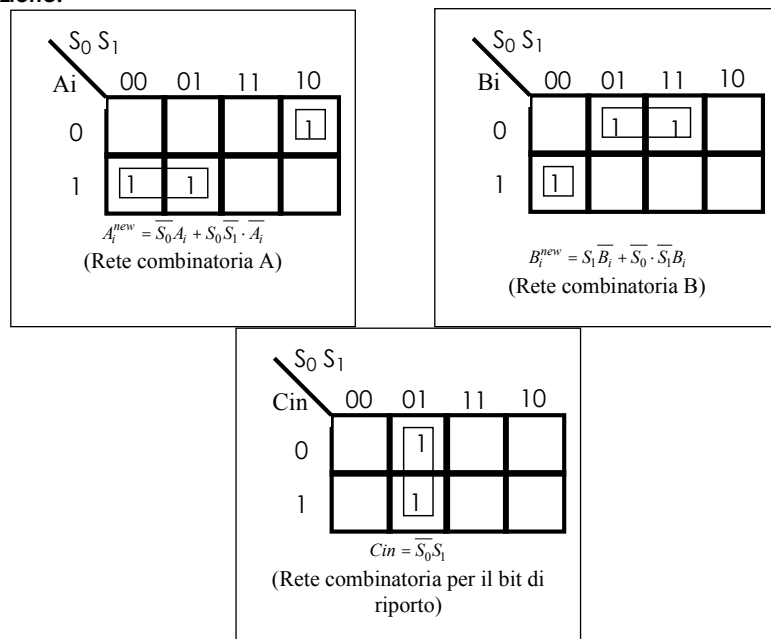
ESERCIZIO 1 (8 punti)

Progettare una ALU con due operandi A e B, che realizzi le seguenti funzioni (l'apice indica la negazione):

s0	s1	Funzione
0	0	A+B
0	1	A-B
1	0	A'
1	1	B'

- (a) (NO: 5 punti – VO: 3 punti) Implementare la ALU utilizzando le opportune reti logiche e un parallel adder
 (b) (3 punti) Mostrare graficamente lo schema della ALU
 (c) (solo VO: 2 punti) Indicare uno schema più efficiente, in termini di tempi di ritardo, di quello presentato nei punti precedenti.

Soluzione.



Per uno schema più efficiente basta sostituire un "carry look-ahead" adder al posto del parallel adder, in modo da evitare il problema dei ritardi dovuti al riporto. E' possibile anche sostituire le varie reti combinatorie con dei multiplexer.

ESERCIZIO 2 (NO: 9 punti – VO: 7 punti)

Sono dati i seguenti formati per la rappresentazione dei numeri in un calcolatore (campo complessivo 28 bit):

- interi senza segno
- reali in virgola fissa con bit di segno, parte intera e sei bit per la parte frazionaria
- reali in virgola mobile con bit di segno, esponente in eccesso 63 (7 bit) e mantissa frazionaria e normalizzata in segno e valore (1.M)

Si richiede:

- 1) (NO: 5 punti – VO: 4 punti) il minimo e massimo valore raggiungibili escluso lo zero e in valore assoluto

- 2) (NO: 4 punti – VO: 3 punti) rappresentare in virgola mobile i numeri 121.25 e 32.75 e sommarli secondo l'algoritmo usato nei calcolatori.

Soluzione.

1)

Interi senza segno: min: 1 – max $2^{28}-1$.

Virgola fissa: min 2^{-6} – max $2^{22}-2^{-6}$

Virgola mobile: min 2^{-63} – max $(2-2^{-20}) \cdot 2^{64}$

2)

$$121.25 = 1.11100101 \cdot 2^6$$

$$32.75 = 1.0000011 \cdot 2^5$$

6 in eccesso 63: 1000101

5 in eccesso 63: 1000100

121.25 1 1000101 11100101000000000000

32.75 1 1000100 0000011000000000000000

Per sommare i due numeri, bisogna innanzi tutto uniformare gli esponenti, poi sommare le mantisse:

$$\begin{array}{r} 1.11100101 + \\ 0.10000011 = \\ \hline 10.01101000 \end{array}$$

E' necessario normalizzare il risultato, che in virgola mobile si rappresenta:

154.00 1 1000110 00110100000000000000

ESERCIZIO 3 (NO: 8 punti - VO: 7 punti)

- (1 punto) Mostrare la suddivisione nei relativi campi TAG, Index, Offset di un indirizzo di memoria primaria di 1Kbyte, disponendo di una cache da 32 byte, secondo il metodo diretto con blocchi da 4 byte.
- (NO: 4 punti - VO: 3 punti) Si consideri la sequenza di riferimenti alla memoria indicati come indirizzi di parole (il primo indirizzo è 0): 389, 719, 387, 697, 306, 308, 194, 198, 699, 310. Indicare i cache *hit* e il contenuto finale della cache, nella configurazione al punto 1. E' sufficiente indicare, per ciascun blocco di cache, l'indirizzo del corrispondente blocco di primaria in esso eventualmente presente.
- (3 punti) Indicare almeno una possibile alternativa per aumentare il numero di hit in cache, senza alterare la dimensione di memoria primaria e di cache e il metodo di indirizzamento adottato.
(Suggerimento. Ricordare il principio di località: chiamate vicine in ordine di tempo e di spazio)

Soluzione:

- < TAG 5 bit> < Cache Index 3 bit> <Offset 2 bit>

- Applicando per ciascuna chiamata le formule per il calcolo di block frame e cache index, si ricavano i seguenti valori per ciascun set:

Indirizzo	389	719	387	697	306	308	194	198	699	310
Blocco primaria	97	179	96	174	76	77	48	49	174	77
Blocco cache	1	3	0	6	4	5	0	1	6	5
Hit									X	X

Il contenuto finale della cache è dunque:

Blocco Cache	0	1	2	3	4	5	6	7
Blocco Primaria	48	49		179	76	77	174	

- La cosa più semplice per ridurre il numero di miss in cache è accrescere la grandezza dei blocchi. Ciò si può notare dal fatto che le richieste in primaria distano spesso meno di otto parole (e.g. prima e terza chiamata, quinta, sesta e ultima chiamata). Ad es. possiamo utilizzare blocchi da 8 byte. In questo caso la cache conterrà solo 4 blocchi, con la seguente sequenza:

Indirizzo	389	719	387	697	306	308	194	198	699	310
Blocco primaria	48	89	48	87	38	38	24	24	87	38
Blocco cache	0	1	0	3	2	2	0	0	3	2
Hit			X			X		X	X	X

Come si nota il numero di hit è passato da 2 a 5.

ESERCIZIO 4 (NO: 8 punti – VO: 6 punti)

Sia data la seguente lista di processi:

Processo	Tempo di arrivo	Tempo di CPU	Memoria
1	0.0	1.2	125K
2	0.4	0.7	25K
3	0.9	1.0	175K

Il sistema utilizza una memoria partizionata in modo statico con quattro partizioni nell'ordine di: 100K, 200K, 50K, 150K. La politica di scheduling dei processi è la FIFO multiprogrammata.

- (NO: 4 punti – VO: 3 punti) Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei processi qualora si impieghi la politica di allocazione Best Fit e First Fit, indicando quali partizioni sono coinvolte.
- (NO: 4 punti – VO: 3 punti) Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio nei due casi.

Soluzione.

Grafico processo-tempo con strategia di allocazione Best Fit:

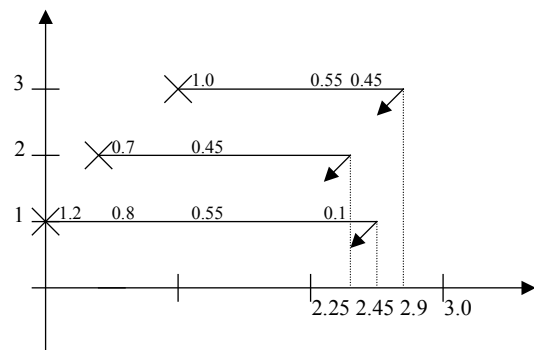
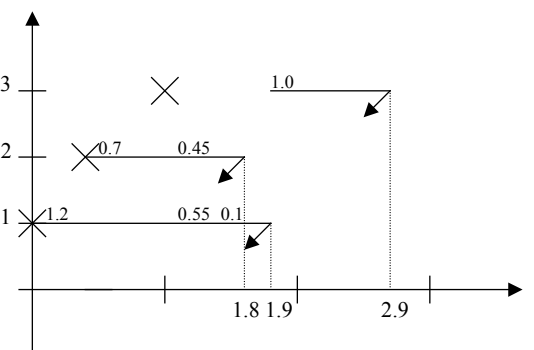


Grafico processo-tempo con strategia di allocazione First Fit



Con la strategia Best Fit tutti e tre i processi vengono allocati ed eseguiti in multi-tasking: il primo nella partizione 150K, il secondo nella partizione da 50K e il terzo nella partizione da 200K.

Con la strategia First Fit, invece, il primo processo viene allocato nella partizione da 200K (la prima in ordine di ricerca), il secondo in quella da 100K. Le ultime due partizioni non sono sufficienti per contenere il terzo processo, che viene così posto in stato di attesa fino a che la partizione da 200K non viene liberata dal primo.

Calcolo di Turnaround e Weighted Turnaround con strategia Best Fit:

Processo	Arrivo	Inizio	Fine	Turnaround	WT
1	0.00	0.00	2.45	2.45	2.04
2	0.40	0.40	2.25	1.85	2.64
3	0.90	1.00	2.90	2.00	2.00
Media				2.10	2.23

Calcolo di Turnaround e Weighted Turnaround con strategia First Fit:

Processo	Arrivo	Inizio	Fine	Turnaround	WT
1	0.00	0.00	1.90	1.90	1.58
2	0.40	0.40	1.80	1.40	2.00
3	0.90	1.90	2.90	2.00	2.00
Media				1.77	1.86

ESERCIZIO 5 (solo VO: 5 punti)

Si consideri una macchina RISC e la seguente sequenza di istruzioni. Si individuino eventuali bolle di ritardo e si risolva il problema adattando il flusso delle istruzioni in modo adeguato. Le lettere indicano altrettante locazioni di memoria.

INDIRIZZO	ISTRUZIONE
100	MOV A,D
101	JMP 103
102	ADD B,I
103	MOV P,Z
104	ADD A,C
105	MOV B,X

Soluzione.

Vi è una bolla di ritardo generata dalla istruzione all'indirizzo 101, che causa un salto all'indirizzo 103. Possiamo introdurre in punti opportuni delle istruzioni NOP (No Operation) oppure riadattare la sequenza secondo il criterio "delayed branch".

INDIRIZZO	NORMAL BRANCH	BRANCH CON NOP	OPTIMIZED "DELAYED BRANCH"
100	MOV A,D	MOV A,D	JMP 103
101	JMP 103	JMP 104	MOV A,D
102	ADD B,I	NOP	ADD B,I
103	MOV P,Z	ADD B,I	MOV P,Z
104	ADD A,C	MOV P,Z	ADD A,C
105	MOV B,X	ADD A,C	MOV B,X
106		MOV B,X	