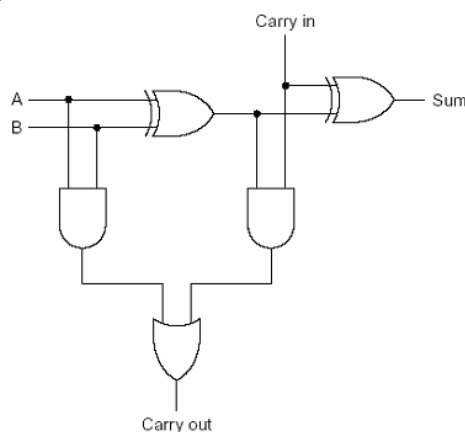


**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO**
26 Gennaio 2007

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

ESERCIZIO 1 (8 punti)

- 1) (3 punti) Scrivere la tabella di verità e le espressioni booleane della somma (Sum) e del riporto in uscita (Carry out) di un Full Adder. Esprimere con A e B i bit da sommare e con C il riporto in ingresso (Carry in).
- 2) (5 punti) Mostrare, **usando le leggi dell'algebra booleana**, che esse sono equivalenti alle espressioni derivanti dalla figura:



Soluzione.

- 1) La tabella di verità richiesta è:

A	B	C	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Da cui si ricava:

$$Sum = A \oplus B \oplus C \quad (1)$$

$$CarryOut = AB + AC + BC \quad (2)$$

- 2) Le espressioni che si ricavano dalla figura sono:

$$Sum = A \oplus B \oplus C \quad (3)$$

$$CarryOut = (A \oplus B) \cdot C + AB \quad (4)$$

E' evidente l'equivalenza di (1) e (3). Per quanto riguarda (2) e (4), riscriviamo la (2) applicando la legge associativa della somma booleana:

$$CarryOut = (A + B) \cdot C + AB \quad (2)'$$

La (2)' è del tutto equivalente alla (4) per C=0. Per C=1 essa si può scrivere:

$$CarryOut = A + B + AB = A + B \quad (2)''$$

che si ottiene mettendo in evidenza A tra gli addendi A e AB.

Analogamente la (4) si può scrivere:

$$CarryOut = \overline{A}\overline{B} + \overline{A}B + AB = A(\overline{B} + B) + \overline{A}B = (A + \overline{A}) \cdot (A + B) = A + B$$

Che si ottiene applicando la legge distributiva della somma rispetto al prodotto. Come si vede quest'ultima espressione è del tutto equivalente alla (2)'.

ESERCIZIO 2 (9 punti)

La memoria di un calcolatore è organizzata con una gerarchia a due livelli cache-primaria (gli indirizzi sono espressi in esadecimale).

- 1) (3 punti) Si ricavi, dalla sequenza di accessi mostrati in figura, l'hit ratio della cache. Si calcoli l'hit ratio della primaria per questa gerarchia a due livelli cache-primaria.

Byte	Recuperato in cache
0F	Sì
AA	Sì
10	
A1	Sì
FA	
90	Sì
65	Sì
80	
B1	Sì
CC	Sì

- 2) (3 punti) Sapendo che il tempo medio di accesso alla cache è pari a 4 ns, e quello alla primaria è pari a 40 ns, calcolare il tempo medio di accesso alla gerarchia a due livelli.
- 3) (3 punti) Indicare di quanto dovrebbe incrementare l'hit ratio della cache per portare il tempo medio di accesso a 8 ns.

Soluzione.

- 1) Dalla definizione di gerarchia di memoria a due livelli, i byte recuperati in cache devono trovarsi anche in primaria, per cui $H_p = 1$.
 Dei dieci accessi in tabella, sette sono 'hit' in cache, per cui $H_c = 0.7$.
- 2) Applicando la formula della gerarchia di memoria a due livelli, si ha:
 $T_m = T_c + (1 - H_c) \cdot T_p = 4 + 0.3 \cdot 40 = 16$ ns.
- 3) Indichiamo con DH_c l'incremento assoluto di hit ratio soddisfacente il requisito:
 $T_m = T_c + (1 - (H_c + DH_c))T_p = T_m(\text{old}) - DH_c T_p = 16 - 40 \cdot DH_c = 8$
 Da cui:
 $DH_c = 8/40 = 0.2$
 Quindi l'hit ratio di cache dovrebbe diventare 0.9.

ESERCIZIO 3 (8 punti)

Implementare in Assembler MIPS una funzione che, dati l'indirizzo iniziale di un vettore v (in $\$4$) e la sua dimensione (in $\$5$), scriva in $\$6$ il valore 1 se i componenti di v sono in ordine non decrescente, 0 altrimenti.

Soluzione.

```
$8 ← i; $9 ← v[i]; $10 ← v[i+1]  
$11 ← v[i+1] < v[i]
```

```
non_decrescente:  addi $29, $29, -12  
                  sw $8, 0($29)  
                  sw $9, 4($29)  
                  sw $10, 8($29)  
                  subi $5, $5, 1  
                  move $11, $0  
                  addi $6, $0, 1  
                  move $8, $0  
for :             beq $8, $5, exit1  
                  lw $9, 0($4)  
                  lw $10, 4($4)  
                  slt $11, $10, $9  
                  bne $11, $0, exit2  
                  addi $4, $4, 4  
                  addi $8, $8, 1  
                  j for  
exit2:            move $6, $0  
exit1:            lw $8, 0($29)  
                  lw $9, 4($29)  
                  lw $10, 8($29)  
                  addi $29, $29, 12  
                  jr $31
```

ESERCIZIO 4 (8 punti)

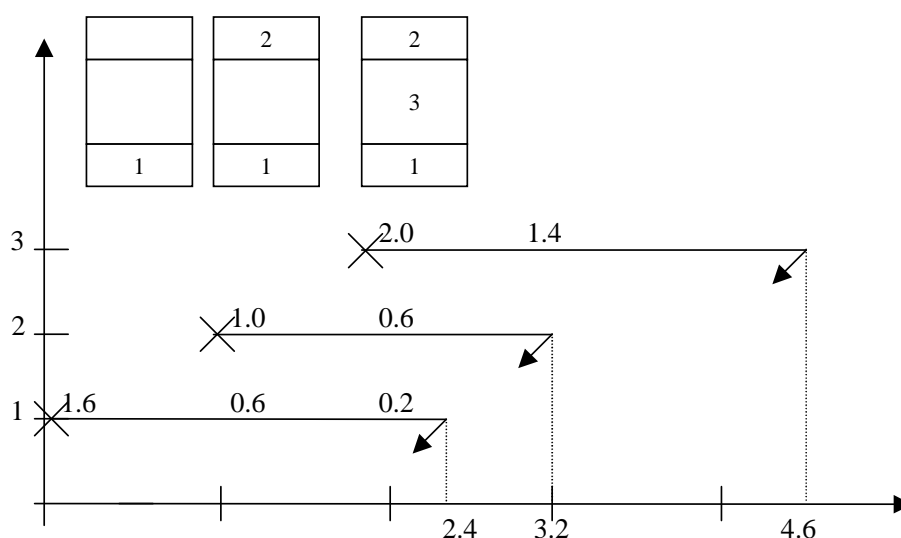
Sia data la seguente lista di processi:

Processo	Tempo di arrivo	Tempo di CPU	Memoria
1	0.0	1.6	140K
2	1.0	1.0	90K
3	1.8	2.0	450K

La memoria è suddivisa (a partire dagli indirizzi "bassi") in tre partizioni statiche di dimensioni 100K, 500K e 150K, e viene gestita con strategia di allocazione Best Fit.

- (6 punti) Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei processi qualora si impieghi la politica di scheduling FIFO multiprogrammata, indicando in quale partizione i processi vengono allocati (memoria inizialmente vuota).
- (2 punti) Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio.

Grafico processo-tempo con politica FIFO multiprogrammata e allocazione Best Fit (gli indirizzi di memoria partono dal basso):



Processo	Arrivo	Fine	Turnaround	WT
1	0.0	2.4	2.4	1.5
2	1.0	3.2	2.2	2.2
3	1.8	4.6	2.8	1.4
Media			2.5	1.7