

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Anno di immatricolazione \_\_\_\_\_

## Compito di Architettura degli elaboratori - A

La soluzione va riportata in bella su questo foglio, se ci sono problemi di spazio, utilizzate il retro, non sono ammessi altri fogli. I punti sono in trentesimi.

1. [1] Si trasformi l'espressione  $y = (a' + b)'c + a(b'c')' + abd$  in un'espressione di tipo SP indicando le proprietà dell'algebra di commutazione utilizzate.

passo 1 .....  $(a'+b)'c+a(b'c')'+abd$  ..... proprietà ..... De Morgan .....

passo 2 .....  $(a''b')c+a(b''+c'')+abd$  ..... proprietà ..... involuzione .....

passo 3 .....  $ab'c+a(b+c)+abd$  ..... proprietà ..... distributiva .....

passo 4 .....  $ab'c+ab+ac+abd$  ..... proprietà ..... assorbimento .....

passo 5 .....  $ab'c+ab+ac$  ..... proprietà ..... assorbimento .....

passo 6 .....  $ab+ac$  ..... proprietà .....

2. [1] Si effettuino le seguenti conversioni:

(a)  $89_{10} \Rightarrow$  1011001 base 2

(b)  $-49_{10} \Rightarrow$  11001111 base 2 (in complemento a 2 su 8 bit)

(c)  $BB$  (intero con segno in esadecimale su 8 bit)  $\Rightarrow$  -69 base 10

(d)  $1100101_2$  (naturale)  $\Rightarrow$  101 base 10

3. [0.5] Si descrivano brevemente le differenze fra flip-flop latch: .....

..... i latch campionano su un livello, i flip-flop campionano in corrispondenza a un fronte .....

Si descriva sinteticamente il motivo per cui il flip-flop è più largamente utilizzato dei latch nella realizzazione di reti sincrone: .....

..... i flip-flop garantiscono la stabilità dell'uscita per un periodo di clock, i latch solo durante la fase di hold .....

4. [2.5] Si consideri la funzione  $f$  non completamente specificata rappresentata nella seguente mappa di Karnaugh e si determini una espressione SP di costo minimo per  $f$  indicando sulla mappa i raggruppamenti rettangolari utilizzati in tale copertura.

	cd				
ab		00	01	11	10
00		0	0	1	1
01		1	-	0	0
11		-	-	1	1
10		0	1	0	1

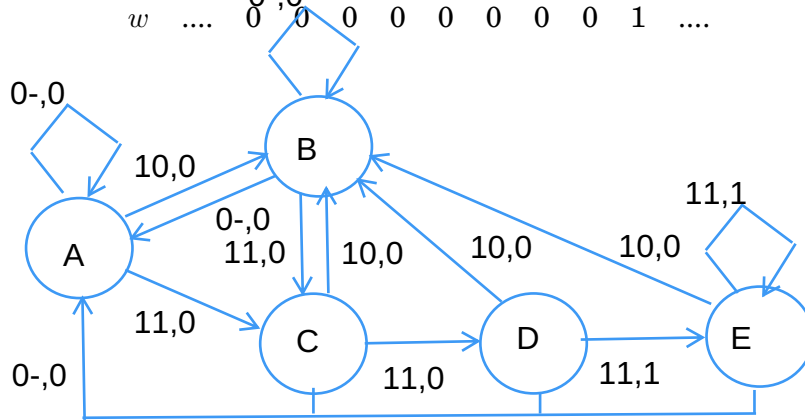
Espressione

$ab+bc'+ac'd+abc'+a'b'c$

5. [4] Si tracci il grafo di transizione dello stato di una FSM con due ingressi  $x$  e  $y$  e un uscita  $w$ . La FSM (di Mealy) ha uscita 0 quando  $x = 0$ , quando  $x = 1$  osserva i bit ricevuti serialmente su  $y$  e porta l'uscita a 1 se gli ultimi 3 bit ricevuti su  $y$  (senza contare quelli ricevuti quando  $x = 0$  e compreso quello corrente) hanno il valore 1.

Esempio di traccia

$x$	....	0	0	0	0	1	1	1	1	1	....
$y$	....	1	0	1	1	1	0	1	1	1	....
$w$	....	0	0	0	0	0	0	0	0	1	....



6. [1.5] Si descriva sinteticamente il principale problema del sommatore Ripple Carry (sommatore seriale)

Il ritardo nella propagazione del riporto

Si descriva poi l'idea utilizzata nel sommatore Carry Look Ahead che risolve in parte tale problema

Calcolare indipendentemente i riporti utilizzando solo gli ingressi

Si riportino le espressioni del carry propagate ( $p$ ) e del carry generate ( $g$ ) utilizzati nel sommatore CLA

$p(i) = a(i) \text{ xor } b(i)$  e  $g(i) = a(i) \text{ and } b(i)$  dove  $i$  rappresenta l'indice dell'  $i$ -mo bit di ingresso

7. [1.5] Si consideri la seguente espressione del calcolo delle proposizioni e la si traduca in un'espressione equivalente che utilizza invece gli operatori dell'algebra di commutazione:

$$P = (a \rightarrow b) \wedge (a \rightarrow c) \quad P = (a' + b)(a' + c)$$

Si faccia la stessa operazione per l'espressione:

$$Q = (a \rightarrow (b \wedge c)) \quad Q = a' + bc$$

Utilizzando la valutazione delle due espressioni per tutti i valori di  $a, b, c$  o usando delle trasformazioni algebriche si determini se le due espressioni sono equivalenti:

$$P = (a' + b)(a' + c) = a'a' + a'c + a'b + bc = a' + a'c + a'b + bc = a' + bc = Q$$

diversamente, si poteva anche valutare  $P$  e  $Q$  per gli 8 possibili valori di  $a, b$  e  $c$

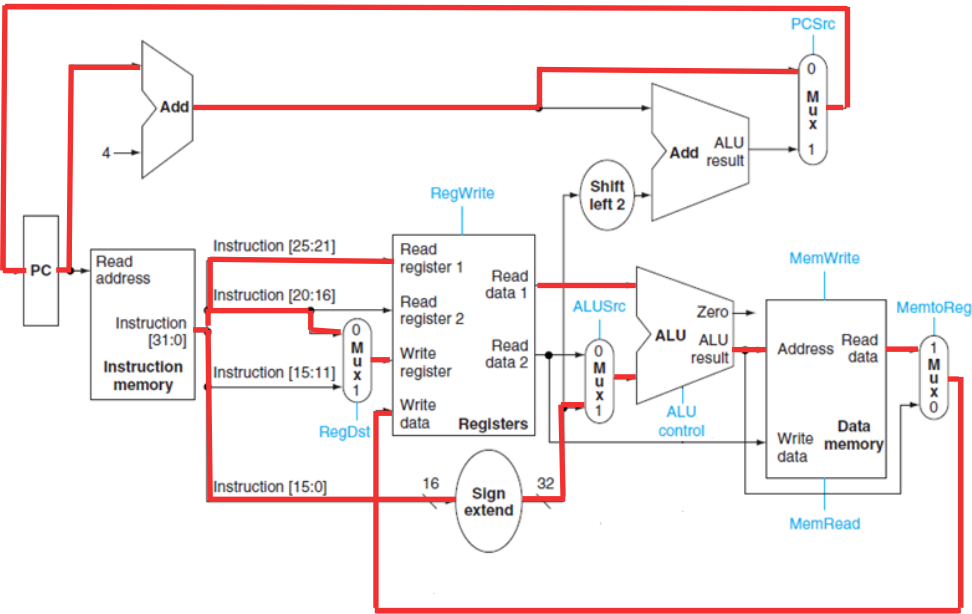
Compito di Architettura degli elaboratori - B

1. [0.5] Si descrivano i campi dell'istruzione or dell'ISA MIPS (numero di bit per campo, posizione dei campi nella parola e significato dei campi)

Esempio di istruzione or : .....or \$t0, \$t1, \$t2.....

campo 1:	opcode	bit, da bit	31	a bit	26	, significato	codice operativo
campo 2:	rs	bit, da bit	25	a bit	21	, significato	primo operando
	rt	bit, da bit	20	a bit	16	, significato	secondo operando
campo 3:		bit, da bit		a bit		, significato	
campo 4:	rd	bit, da bit	15	a bit	11	, significato	risultato
	shamt	bit, da bit	10	a bit	6	, significato	shamt
campo 5:	funct	bit, da bit	5	a bit	0	, significato	operazione svolta
campo 6:		bit, da bit		a bit		, significato	dalla alu

2. [1.5] Si evidenzino (ripassandoli con una matita colorata) i cammini dei dati che vengono attivati durante l'esecuzione di un istruzione di tipo lw \$t0, offset(\$t1) nella CPU MIPS a ciclo singolo illustrata di seguito. Si annotino tali cammini con gli argomenti dell'istruzione.



3. [1] Si elenchino un vantaggio e uno svantaggio nell'utilizzo di un ISA in cui, diversamente dall'ISA MIPS, le istruzioni hanno diverse dimensioni (ad esempio 2 byte, 3 byte e 4 byte).

vantaggio	memorizzazione più compatta dei programmi
svantaggio	rete di fetch e decodifica delle istruzioni più complessa

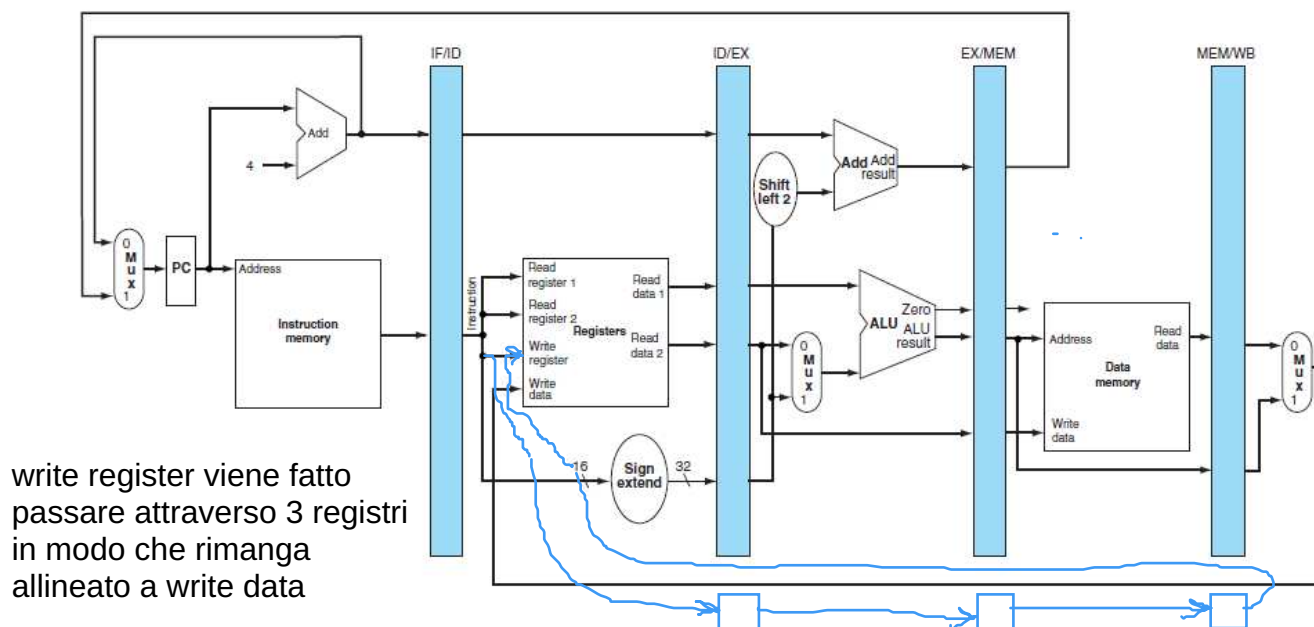
4. [1] Analizzando statisticamente il trasferimento dati di una CPU verso la memoria si osserva una forte località spaziale. Ovvero .....  
 ..... quando si legge o scrive un dato dalla memoria, è molto probabile che i dati adiacenti a questo siano letti o scritti dopo poche istruzioni ..... (definire la località spaziale)

Supponendo di utilizzare una memoria cache a mappatura diretta, su quale parametro si può agire per sfruttare il tipo di località evidenziata, e quali valori si possono tipicamente utilizzare?

..... indipendentemente dal tipo di cache, si possono usare dimensioni dei blocchi (b) .....  
 ..... maggior di 1 (2,4 ....) .....

5. [1.5] Questa versione pipelined della CPU MIPS ha un problema che riguarda gli accessi in scrittura al register file. Si descriva sinteticamente tale problema e si annoti lo schema con una possibile soluzione.

..... l'indirizzo (write register) del dato da scrivere nel register file arriva prima .....  
 ..... del dato stesso che sta ancora viaggiando nella pipeline .....



6. [0.5] In una CPU pipelined a  $n$  stadi, il periodo di clock viene determinato considerando:

- ☐ la somma dei ritardi di tutti gli stadi
- ☒ il ritardo dello stadio più lento
- ☐ la somma dei ritardi di tali stadi diviso  $n$

dove per ritardo di uno stadio si intende:

- ☒ il ritardo del cammino più lento in tale stadio (ad esempio, nello stadio EX nella figura di sopra, un possibile cammino va da ID/EX a EX/MEM attraverso Shift e Add)
- ☐ la somma dei ritardi di tutti i blocchi utilizzati in tale stadio (ad esempio, ALU, adder, multiplexer e shifter nello stadio EX nella figura di sopra)
- ☐ il ritardo massimo fra quelli di tali blocchi

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

7. [3] Si considerino la versione pipelined della CPU MIPS e questo frammento di codice:

```
add $t0, $s0, $s1
and $s1, $t0, $t4
or  $t1, $s2, $s4
or  $t2, $t1, $t0
```

Si riporti l'esecuzione in sequenza (senza tenere conto degli hazard) nella seguente tabella. Si indichino poi in tale tabella le dipendenze fra i dati che darebbero luogo ad hazard.

*nota: non é detto che servano tutte le righe nelle tabelle*

clock	IF	ID	EX	MEM	WB
1	and t0,s0,s1				
2	and s1,t0,t4	and t0,s0,s1			
3	or t1,s2,s4	and s1,t0,t4	and t0,s0,s1		
4	or t2,t1,st0	or t1,s2,s4	and s1,t0,t4	and t0,s0,s1	
5		or t2,t1,st0	or t1,s2,s4	and s1,t0,t4	and t0,s0,s1
6			or t2,t1,st0	or t1,s2,s4	and s1,t0,t4
7				or t2,t1,st0	or t1,s2,s4
8					or t2,t1,st0

Si mostri l'esecuzione di tale frammento di codice in modo che tali dipendenze siano risolte. Si supponga a questo riguardo di considerare una pipeline priva di bypass a parte quello che rende possibile leggere e scrivere dal register file nello stesso ciclo di clock. Per ridurre il numero di stalli provate invece a riorganizzare il frammento di codice.

clock	IF	ID	EX	MEM	WB
1	and t0,s0,s1				
2	or t1,s2,s4	and t0,s0,s1			
3		or t1,s2,s4	and t0,s0,s1		
4	and s1,t0,t4		or t1,s2,s4	and t0,s0,s1	
5	or t2,t1,st0	and s1,t0,t4		or t1,s2,s4	and t0,s0,s1
6		or t2,t1,st0	and s1,t0,t4		or t1,s2,s4
7			or t2,t1,st0	and s1,t0,t4	
8				or t2,t1,st0	and s1,t0,t4
9					or t2,t1,st0
10					
11					
12					

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

8. [3] Si consideri la seguente memoria cache a mappatura diretta con  $b = 2$  illustrata nella prima tabella. Si descrivano i campi di un indirizzo dell'architettura MIPS utilizzato per accedere a tale memoria.

Risposta:

31

5 4 3    2    1 0  
|    |    |    |  
set block byte  
          offset

La prima tabella mostra anche il suo stato iniziale prima dell'esecuzione della sequenza di istruzioni.

istruzione	hit/miss
lw \$t0, 0x80(\$zero)	miss
lw \$t1, 0x20(\$zero)	miss
lw \$t2, 0x4c(\$zero)	hit
lw \$s0, 0xb84(\$zero)	miss

Si annoti ogni istruzione con il suo esito (hit/miss) e si riporti lo stato finale della cache nella seconda tabella.

Stato iniziale

V	tag	data block 1	data block 0	set
0				7
0				6
0				5
1	0x0..10	M[0xA4]	M[0xA0]	4
0				3
0				2
1	0x0..01	M[0x4c]	M[0x48]	1
0				0

Stato finale

V	tag	data block 1	data block 0	set
				7
				6
				5
1	0x0..00	M[0x24]	M[0x20]	4
				3
				2
1	0x0..01	M[0x4c]	M[0x48]	1
1	0x0..101110	M[0xb84]	M[0xb80]	0