

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  $y = (a + b + d')(abc)'$  in un'espressione di tipo SP indicando le proprietà dell'algebra di commutazione utilizzate.

passo 1 ..... proprietà .....

passo 2 ..... proprietà .....

passo 3 ..... proprietà .....

passo 4 ..... proprietà .....

2. [1] Si effettuino le seguenti conversioni:

(a)  $79_{10} \Rightarrow$  \_\_\_\_\_ base 2

(b)  $-44_{10} \Rightarrow$  \_\_\_\_\_ base 2 (in complemento a 2 su 8 bit)

(c)  $8E$  (intero con segno in esadecimale su 8 bit)  $\Rightarrow$  \_\_\_\_\_ base 10

(d)  $111010_2$  (naturale)  $\Rightarrow$  \_\_\_\_\_ base 10

3. [0.5] Si indichino quali di queste affermazioni sono sempre corrette per una funzione dell'algebra di commutazione non completamente specificata.

- ☐ l'espressione di costo minimo SP si ottiene assegnando a 0 tutte le condizioni di indifferenza
- ☐ l'espressione di costo minimo PS si ottiene assegnando a 1 tutte le condizioni di indifferenza
- ☐ l'espressione di costo minimo SP si ottiene sfruttando eventualmente alcune condizioni di indifferenza per espandere gli implicant della funzione
- ☐ l'espressione di costo minimo SP si ottiene utilizzando tutti gli implicant primi della funzione

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

Espressione

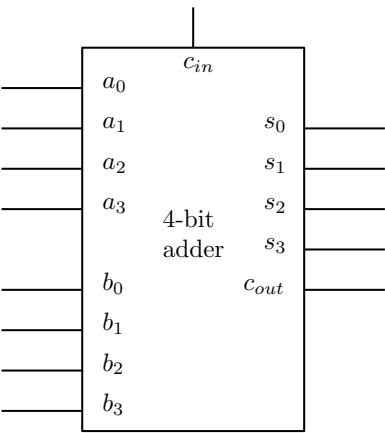
5. [4] Si tracci il grafo di transizione dello stato di una FSM con un ingresso  $x$  e due uscite  $y_1$  e  $y_0$ . La FSM (di Mealy) riceve serialmente su  $x$  delle parole di 3 bit ciascuna e produce in uscita su ciascun bit la codifica binaria del numero di 1 ricevuti fino a quel periodo di clock (nella parola corrente).

Esempio di traccia

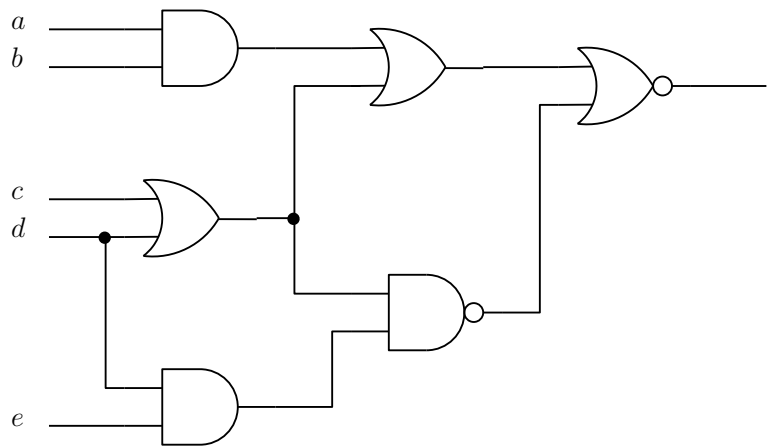
$x$	...	0	1	0	1	1	0	...
$y_1$	...	0	0	0	0	1	1	...
$y_0$	...	0	1	1	1	0	0	...

6. [1.5] Aritmetica binaria

Si utilizzi il sommatore a 4 bit indicato a destra per realizzare una rete che, dati due interi senza segno  $X$  e  $Y$  rappresentati su 3 bit, calcoli l'espressione  $X + 2 * Y + 1$ .  
La soluzione può essere rappresentata nella figura connettendo opportunamente i segnali di  $X$  ( $x_2, x_1$  e  $x_0$ ), quelli di  $Y$  ( $y_2, y_1$  e  $y_0$ ) e le costanti 1 e 0 alle porte di ingresso dell'adder.



7. [1.5] Si analizzi la seguente rete determinando l'espressione di ciascun segnale in funzione degli ingressi (si annoti la figura), si valuti poi il valore dell'uscita per le configurazioni di ingresso in tabella.



$abcde$	$out$
01111	
10000	
11101	
10010	

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

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

## Compito di Architettura degli elaboratori - B

1. [1] Si descrivano i campi delle istruzioni add e addi dell'ISA MIPS (numero di bit per campo, posizione dei campi nella parola e significato dei campi)

*Nota: non é detto che servano tutti i campi.*

Esempio di istruzione add : .....

campo 1: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 2: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 3: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 4: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 5: ....., n. di bit ....., da bit .... a bit ....., significato .....

Esempio di istruzione addi : .....

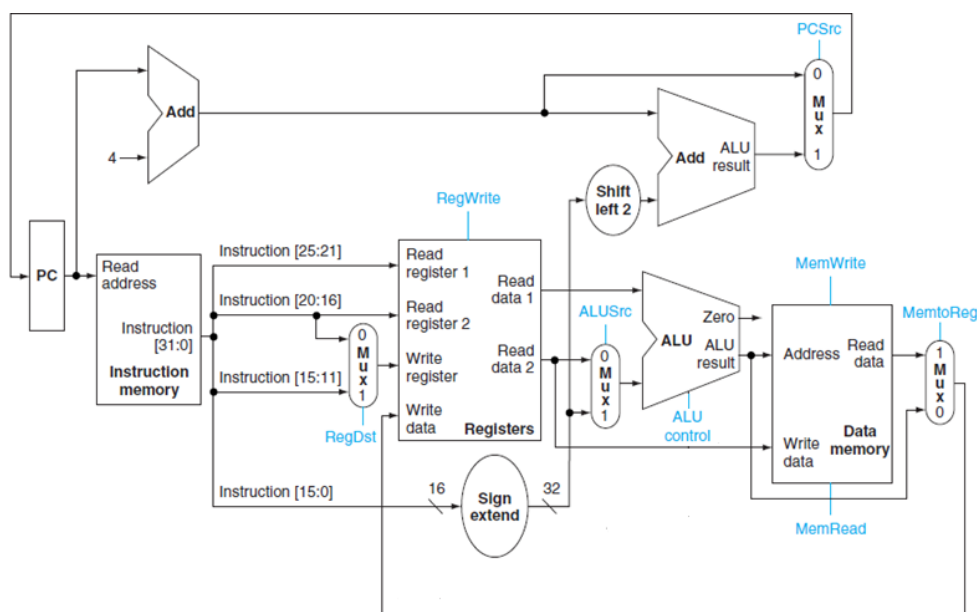
campo 1: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 2: ....., n. di bit ....., da bit .... a bit ....., significato .....

campo 3: ....., n. di bit ....., da bit .... a bit ....., significato .....

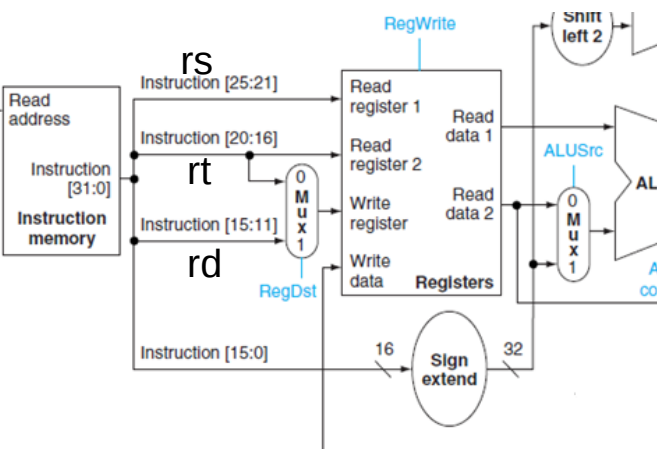
campo 4: ....., n. di bit ....., da bit .... a bit ....., significato .....

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 beq \$t0, \$t1, label nella CPU MIPS a ciclo singolo illustrata di seguito. Si annotino tali cammini con gli argomenti dell'istruzione.



3. [1.5] Modifica a ISA MIPS

Si vorrebbe aggiungere un'istruzione simile ad `addi` all'ISA MIPS. Tale istruzione (`addx`) ha i campi: `opcode` (6 bit), `rs` (5 bit) e 21 bit per una costante (`const`). La costante viene sommata a `rs` che è anche la destinazione ( $rs = rs + const$ ). Con riferimento alla porzione di data-path a destra si indichi il motivo per cui tale data-path non supporta tale modifica



.....  
.....  
Si descriva o si disegni cosa bisognerebbe fare per supportare la nuova istruzione  
.....  
.....

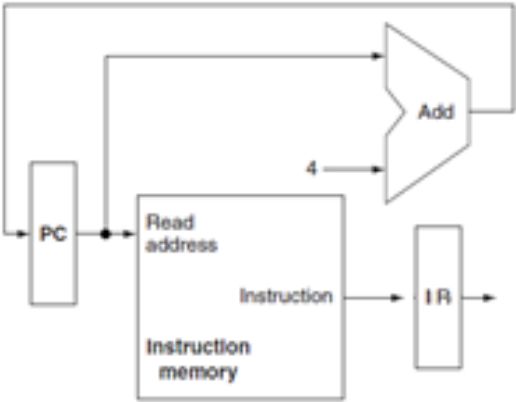
4. [1] Si descrivano due possibili motivi per cui passando da una CPU a ciclo singolo a una sua versione pipelined a 5 stadi, il tempo di esecuzione di un programma non si riduce di un fattore 5 come ci si aspetterebbe nel caso ideale. Questo nonostante il programma non abbia dipendenze dati o branch (si supponga anche che la memoria sia accessibile in un ciclo di clock).

- 1) .....
- 2) .....

5. [1] Ritardo dello stadio di IF

Nella versione pipelined dell'architettura MIPS, lo stadio di IF estrae l'istruzione dalla memoria e incrementa il PC di 4 secondo lo schema illustrato di fianco. Se  $t_{adder} = 0.2ns$  è il ritardo dell'adder e  $t_{mem} = 0.9ns$  quello della memoria, si determini quale è il ritardo (massimo) dello stadio considerato:

.....



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

6. [3] Si considerino la versione pipelined della CPU MIPS e questo frammento di codice:

```
addi $t0, $t0, 64
add $t2, $t0, $t1
or $t3, $t0, $t1
```

Si riporti la loro 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					
2					
3					
4					
5					
6					
7					
8					

Si modifichi 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 fra stadi diversi (in cui é comunque possibile leggere e scrivere dal register file nello stesso ciclo di clock).

clock	IF	ID	EX	MEM	WB
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

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

7. [3] Si consideri la seguente memoria di tipo set associativo a 2 vie con  $b = 1$  illustrata nella prima tabella. Si descrivano i campi di un indirizzo dell'architettura MIPS utilizzato per accedere a tale memoria.

Risposta: 31 0

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

istruzione	hit/miss
lw \$t0, 0xAC(\$zero)	
lw \$t1, 0x70(\$zero)	
lw \$t2, 0x90(\$zero)	
lw \$s0, 0x184(\$zero)	

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

Stato iniziale

V	way-1 tag	data	V	way-0 tag	data	set
0			0			7
0			1	0x0...05	M[0xB8]	6
0			0			5
0			1	0x0...03	M[0x70]	4
0			0			3
0			0			2
0			1	0x0...05	M[0xA4]	1
0			0			0

Stato finale

V	way-1 tag	data	V	way-0 tag	data	set
						7
						6
						5
						4
						3
						2
						1
						0