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 $y = (a + bc)'(d' + (bc)' + e)'$ in un espressione di tipo SP indicando le proprietá dell'algebra di commutazione utilizzate.

   passo 1 ....(a+bc)'(d'+(bc)'+e)'.....proprietá ....De Morgan.....

   passo 2 ....(a'(bc)')(d''(bc)''e')....proprietá ....involuzione.....

   passo 3 ....(a'(bc)')(d(bc)e')....proprietá ....De Morgan.....

   passo 4 ....(a'(b'+c'))(bcde')....proprietá ....distributiva.....

   passo 5 ....(a'b'+a'c')bcde'....proprietá ....distributiva.....

   passo 6 a'b'bcde'+a'bcc'de' = 0 + 0 = 0 proprietá ...complento e nullo.....

2. [1] Si effettuino le seguenti conversioni:
   (a) $49_{10} \Rightarrow$ 110001 base 2
   (b) $-29_{10} \Rightarrow$ 11100011 base 2 (in complemento a 2 su 8 bit)
   (c) $BE$ (intero con segno in esadecimale su 8 bit) $\Rightarrow$ -66 base 10
   (d) $100101_2$ (naturale) $\Rightarrow$ 37 base 10

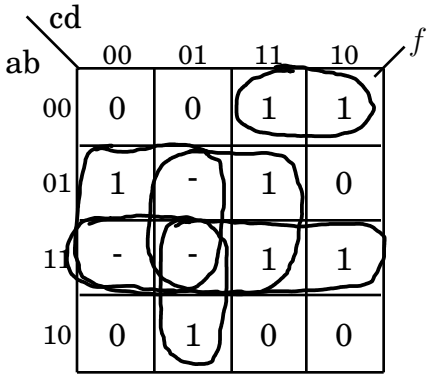
3. [0.5] Si consideri una FSM di Moore e si indichino quali fra le seguenti affermazioni sono sempre corrette.

   ◯ il valore delle uscite dipende da stato presente e ingresso corrente

   Ⓧ il valore delle uscite dipende dallo stato iniziale e dalla sequenza di ingressi applicata alla FSM

   Ⓧ il valore delle uscite dipende dallo stato presente

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 ragruppamenti rettangolari utilizzati in tale copertura.




Espressione

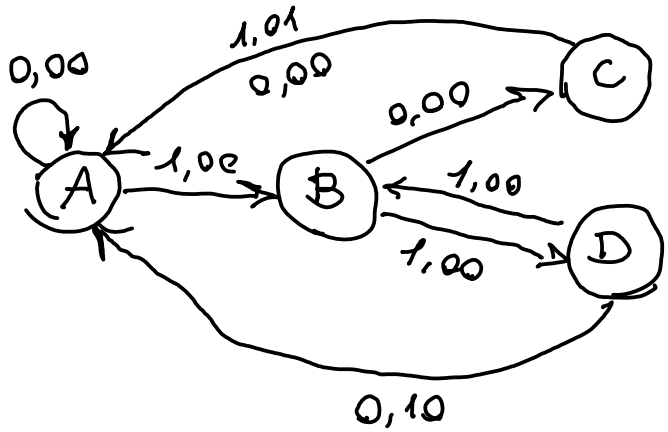
ab + ac'd + bd + bc' + a'b'c

Nome: _____ Cognome: _____

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) osserva i bit ricevuti serialmente su $x$. Se la FSM riceve la sequenza 101 ($x_{k-2}x_{k-1}x_k$), le uscite si portano a 01, se riceve 110, le uscite si portano a 10 (sull'ultimo bit ricevuto in entrambi i casi). Negli altri casi, le uscite valgono 00. Si noti che le due sequenze non possono essere sovrapposte (ad esempio se si riceve 0....01101, le uscite si portano a 10 in quanto si riconosce 110, ma poi tornano a 00 senza considerare la prima occorenza di 101).

Esempio di traccia

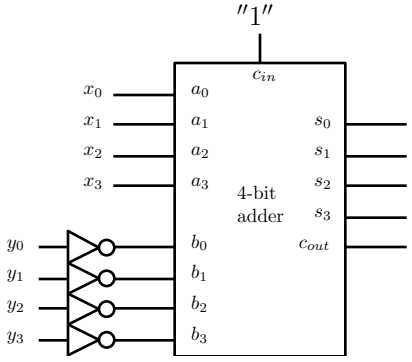
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | .... | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | .... |
| $y_1$ | .... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | .... |
| $y_0$ | .... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | .... |



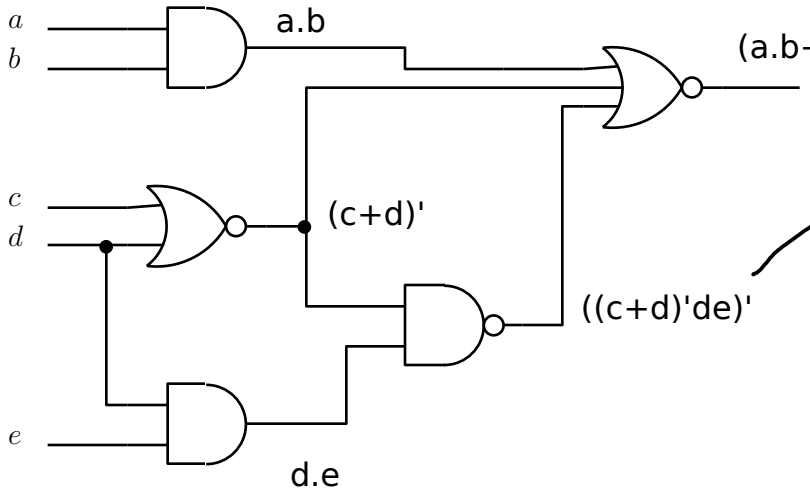

6. [1.5] aritmetica binaria

Si consideri la rete indicata a destra e descriva, giustificando sinteticamente il ragionamento svolto, l'espressione aritmetica dell'uscita ($S = \{s_3, s_2, s_1, s_0\}$ senza $c_{out}$) in funzione dei due ingressi $X$ e $Y$ che si suppone rappresentino due interi con segno rappresentati in complemento a 2.

S = X - Y




7. [1.5] Si calcoli l'espressione di ciascun segnale della rete in funzione degli ingressi $(a, b, c, d, e)$ riportandola sulla figura. Questa rete é il risultato di un qualche errore di progettazione (gate inutili), provate ragionando sulla rete (o anche sull'espressione di uscita) a determinare quale.




con De Morgan diventa
(c+d)''+(de)' =
c+d+d'+e' = 1
in quanto d+d'=1,
quindi un ingresso del NOR e' sempre a 1 e l'uscita e' sempre a 1
tutta la rete e' equivalente alla costante 1

Nome: _____ Cognome: _____

# Compito di Architettura degli elaboratori - B

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

   Esempio di istruzione `sw` : ..... sw $t1, offset($s1) .....

   campo 1: ....opcode....bit, da bit ....31.....a bit ....26....., significato cod. op.....

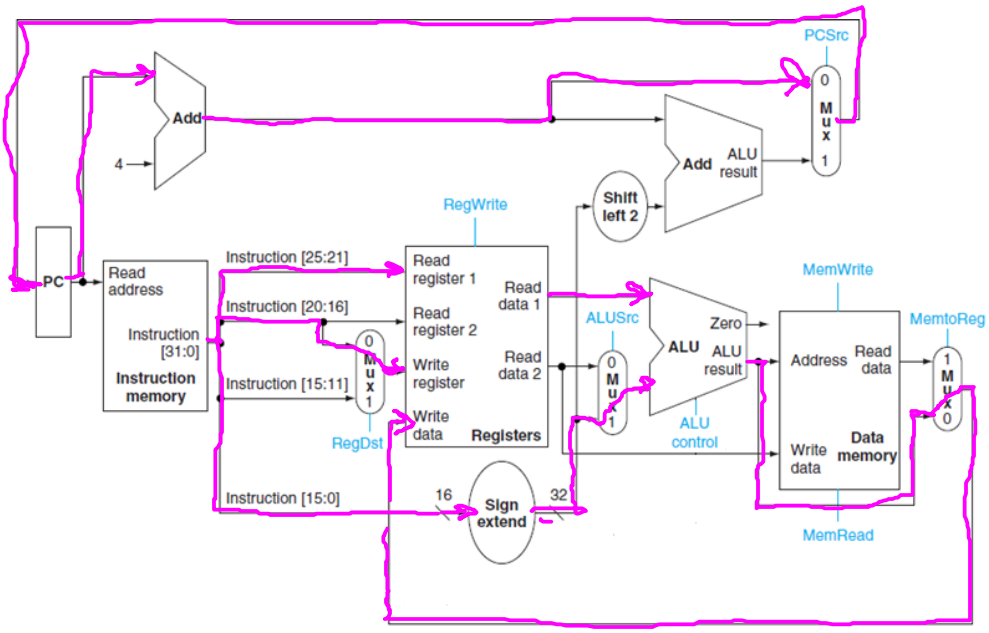
   campo 2: ....rs.....bit, da bit ....25.....a bit ....21....., significato address.....

   campo 3: ....rt.....bit, da bit ....20.....a bit ....16....., significato dest.....

   campo 4: ....offset.....bit, da bit ....15.....a bit .....0....., significato ..offset.....

   campo 5: .....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 `ori $t0, $t1, const` nella CPU MIPS a ciclo singolo illustrata di seguito. Si annotino tali cammini con gli argomenti dell'istruzione.




3. [1.5] Si consideri la CPU MIPS a ciclo singolo e si supponga che il ritardo massimo combinatorio necessario per eseguire un operazione di tipo R sia 8 ns, mentre per eseguire un operazione di tipo I sia di 10 ns. Un progettista modifica l'hardware in modo da rendere piú efficienti le operazioni di tipo R che ora richiedono 7 ns, ma contrariamente alle sue aspettative il tempo di esecuzione di un programma rimane lo stesso. Si spieghi sinteticamente il motivo per cui questo avviene:

   ..... il periodo di clock di una CPU a ciclo singolo dipende dal ritardo massimo .....
   che rimane 10 ns e quindi la velocizzazione non serve a niente
   .....

Nome: _____ Cognome: _____

4. [1] Si scriva una giustificazione sintetica delle seguenti proposizioni che valgono passando dalla versione a ciclo singolo di una CPU alla sua versione pipelined:

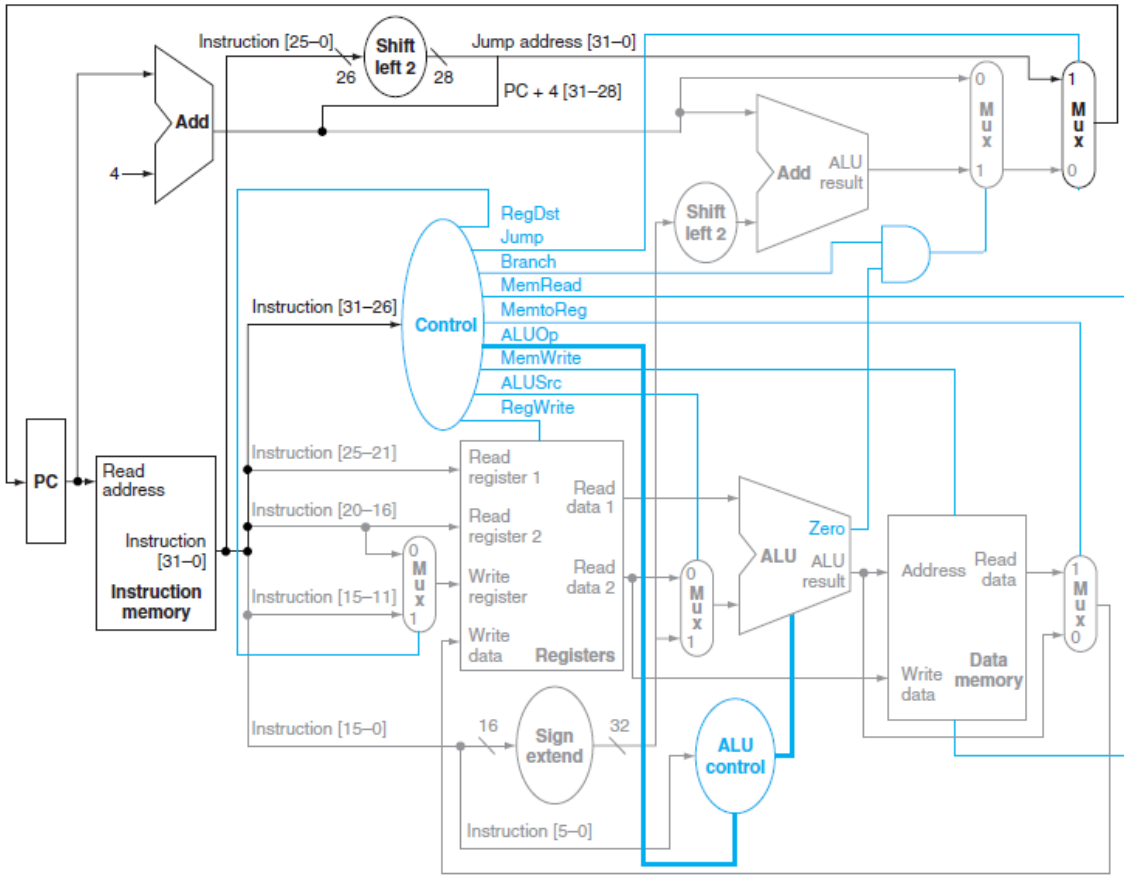
   1) la latenza di una singola istruzione non migliora e puó addirittura peggiorare

   la logica che deve essere "attraversata" dai dati dell'istruzione rimane la stessa e a questa si aggiungono i ritardi dei FF, quindi il tempo per eseguire un istruzione non cambia

   2) la latenza di un programma in generale migliora

   il pipelining consente di aumentare la frequenza di clock e quindi in generale il tempo di esecuzione di un intero programma (che nel caso pipelined non è la somma delle latenze delle singole istruzioni) cala

5. [1] Si determini il valore dei segnali indicati in tabella quando viene eseguita un istruzione `add`. I segnali e il loro significato possono essere dedotti dalla rappresentazione della micro-architettura della CPU MIPS.




| segnale | valore | segnale | valore |
|---|---|---|---|
| RegDst | 1 | Jump | 0 |
| Branch | 0 | MemRead | 0 |
| Memto Reg | 0 | MemWrite | 0 |
| ALUSrc | 0 | RegWrite | 1 |

Nome: _____ Cognome: _____

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

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

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 | add t0,s0,s1 | | | | |
| 2 | add s1,t0,t4 | add t0,s0,s1 | | | |
| 3 | or t1,s2,s4 | add,s1,t0,t4 | add t0,s0,s1 | | |
| 4 | or t2,t1,t4 | or t1,s2,s4 | add s1,t0,t4 | add t0,s0,s1 | |
| 5 | | or t2,t1,t4 | or t1,s2,s4 | add s1,t0,t4 | add t0,s0,s1 |
| 6 | | | or t2,t1,t4 | or t1,s2,s4 | add s1,t0,t4 |
| 7 | | | | or t2,t1,t4 | or t1,s2,s4 |
| 8 | | | | | or t2,t1,t4 |

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.

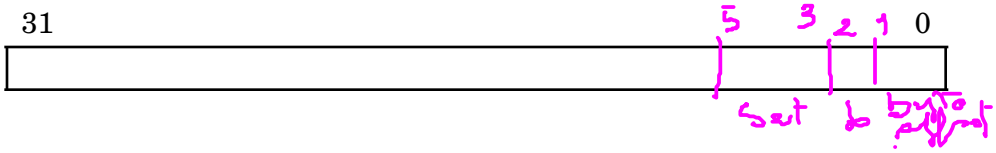
| clock | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | add t0,s0,s1 | | | | |
| 2 | or t1,s2,s4 | add t0,s0,s1 | | | |
| 3 | null | or t1,s2,s4 | add t0,s0,s1 | | |
| 4 | add s1,t0,t4 | null | or t1,s2,s4 | add t0,s0,s1 | |
| 5 | or t2,t1,t4 | add s1,t0,t4 | null | or t1,s2,s4 | add t0,s0,s1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

Si osserva che la terza operazione non ha dipendenze dalla prima e dalla terza e puo' quindi essere anticipata. Questo risparmia un ciclo di clock

Nome: _____ Cognome: _____

7. [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:




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

| istruzione | hit/miss |
|---|---|
| `lw $t0, 0x84($zero)` | miss |
| `lw $t1, 0x70($zero)` | miss |
| `lw $t2, 0xac($zero)` | miss |
| `lw $s0, 0xbc($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..00 | M[0x24] | M[0x20] | 4 |
| 0 | | | | 3 |
| 0 | | | | 2 |
| 0 | | | | 1 |
| 1 | 0x0..00 | M[0xB4]a | M[0xB0] | 0 |

1011 1100

Stato finale

| V | tag | data block 1 | data block 0 | set |
|---|---|---|---|---|
| 1 | 0x0..5 | M[0xbc] | M[0xb8] | 7 |
| 1 | 0x0..1 | M[0x74] | M[0x70] | 6 |
| 1 | 0x0..2 | M[0xac] | M[0xa8] | 5 |
| 1 | 0x0..0 | M[0x24] | M[0x20] | 4 |
| | | | | 3 |
| | | | | 2 |
| | | | | 1 |
| 1 | 0X0..1 | M[0x84] | M[0x80] | 0 |