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

   passo 1 (a'b'(de)')bc' ..... proprietá De Morgan .....
   passo 2 (a'b'(d'+e'))bc' ..... proprietá De Morgan .....
   passo 3 (d'+e')ab'bc' ..... proprietá commutativa .....
   passo 4 (d'+e')a.0.c' ..... proprietá complemento .....
   passo 5 0 ..... proprietá nullo .....

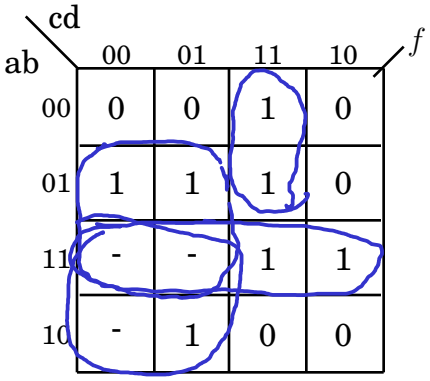
   nota: 0 è un espressione SP comunque andavano bene anche altri passaggi

2. [1] Si effettuino le seguenti conversioni:
   (a) $56_{10} \Rightarrow$ 111000 base 2
   (b) $-110_{10} \Rightarrow$ 10010010 base 2 (in complemento a 2 su 8 bit)
   (c) $B8$ (intero con segno in esadecimale su 8 bit) $\Rightarrow$ -72 base 10
   (d) 100110_2 (naturale) $\Rightarrow$ 38 base 10

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

   ◯ l'espressione di costo minimo SP si puó ottenere negando l'espressione di costo minimo di tipo PS della stessa funzione si otterrebbe il complemento della funzione di partenza
   ◯ l'espressione di costo minimo SP si puó ottenere sostituendo i prodotti con le somme e viceversa in un espressione di costo minimo di tipo PS della stessa funzione
   ⊗ l'espressione di costo minimo SP va calcolata in maniera indipendente dall'espressione PS della stessa funzione si ottiene di solito un altra 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 ragruppamenti rettangolari utilizzati in tale copertura.



Espressione

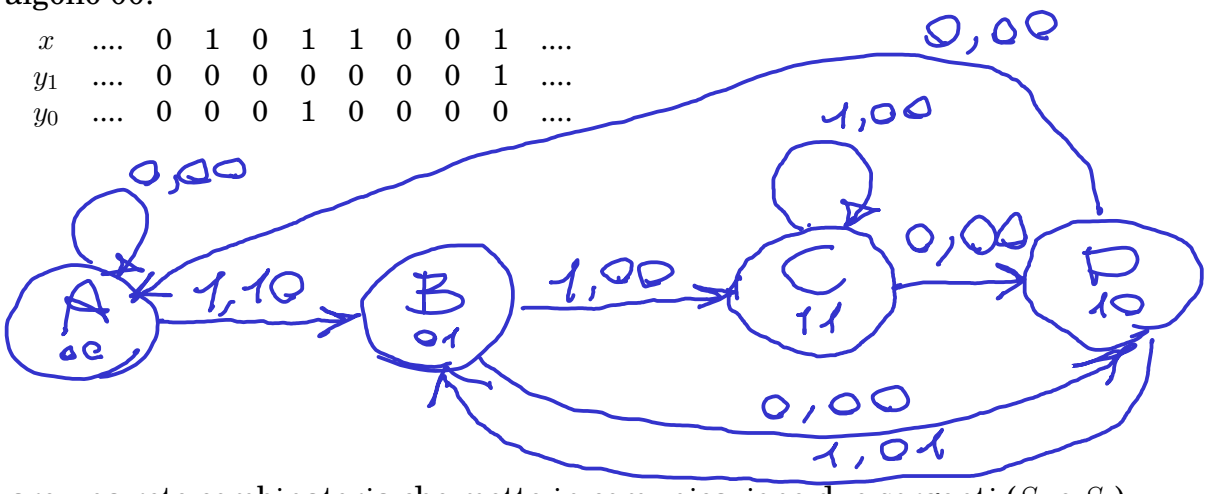
$f = bc' + a'cd + ab + ac'$

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 gli ultimi 3 bit ($x_{k-2}x_{k-1}x_k$) sono pari a 101, le uscite si portano a 01, se tali bit sono pari a 001, le uscite si portano a 10. Negli altri casi, le uscite valgono 00.

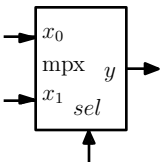
Esempio di traccia

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

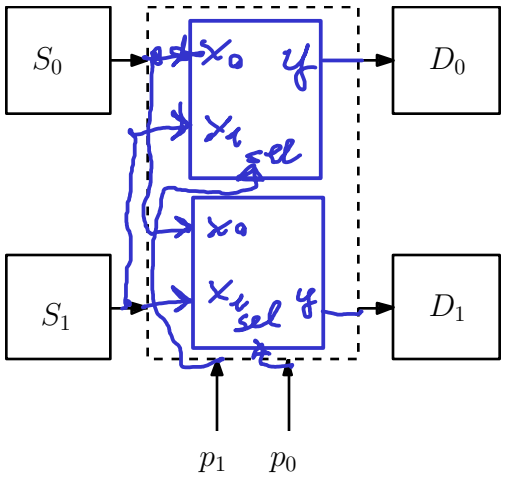


6. [1.5] Si vuole utilizzare una rete combinatoria che mette in comunicazione due sorgenti (S_0 e S_1) con due destinazioni (D_0 e D_1).

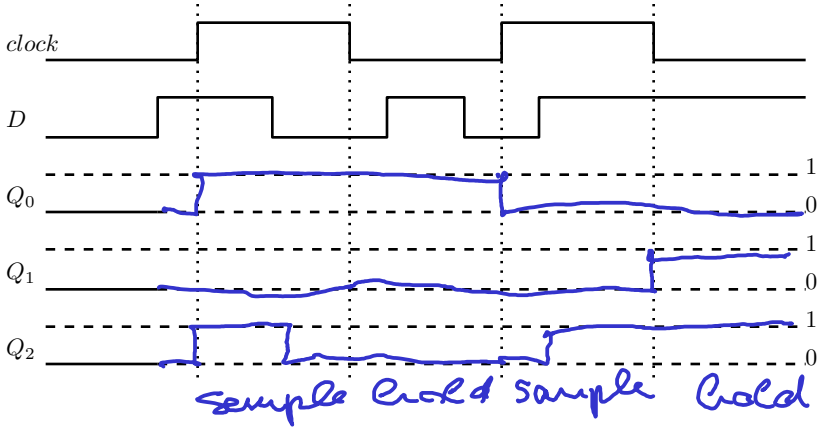
La rete deve consentire tutte le possibili connessioni fra sorgenti e destinazioni sulla base dei valori di p_1 e p_0. Per realizzarla rete si utilizzino multiplexer a due ingressi dati e un bit di selezione come quello indicato in figura. Si tracci lo schema connettendo opportunamente i diversi segnali e poi si riportino in tabella i valori di p_1 e p_0 corrispondenti alle diverse interconnessioni.



D_0	D_1	p_1p_0
S_0	S_0	0 0
S_1	S_1	1 1
S_0	S_1	0 1
S_1	S_0	1 0



7. [1.5] Si considerino 3 diversi elementi di memoria con lo stesso clock e lo stesso ingresso D. Il primo é un flip-flop D edge-triggered che campiona sui fronti di salita (uscita Q_0), il secondo é sempre un flip-flop, ma campiona sui fronti di discesa (uscita Q_1) e il terzo é un D latch trasparente che campiona se il clock e a 1 ed é in hold se il clock é a 0 (uscita Q_2). Si chiede di completare le forme d'onda delle uscite dati il clock e D in figura.



latch →

Nome: _____ Cognome: _____

Compito di Architettura degli elaboratori - B

1. [0.5] Si descrivano i campi dell'istruzione `beq` 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 `beq` : beq $t0, $t4, label

   campo 1: opcode, n. di bit 6, da bit 31 a bit 26, significato identifica l'istruz.

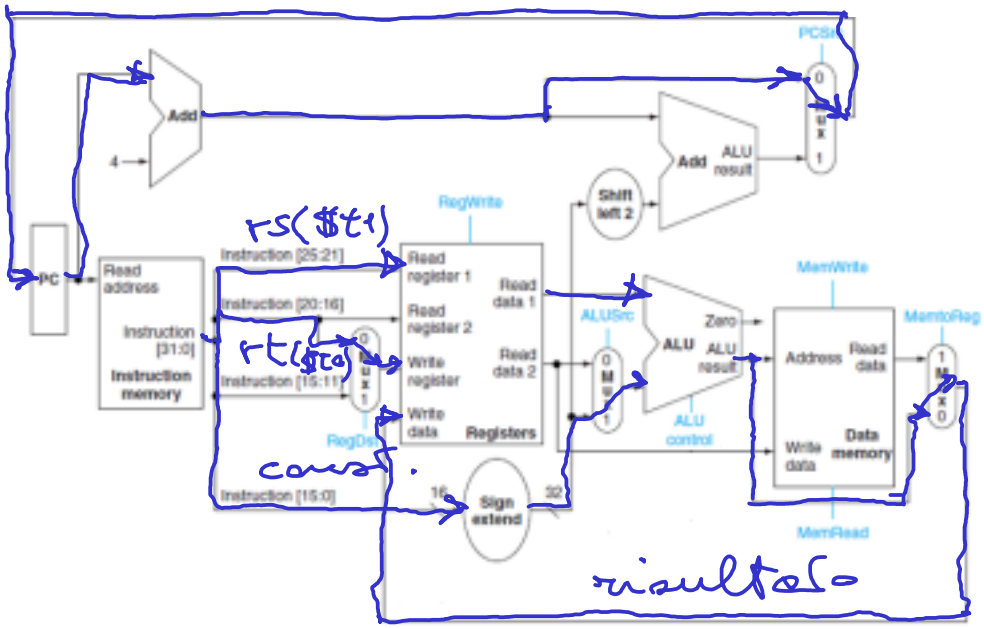
   campo 2: rs, n. di bit 5, da bit 25 a bit 21, significato primo operando

   campo 3: rt, n. di bit 5, da bit 20 a bit 16, significato secondo operando

   campo 4: offset, n. di bit 16 da bit 15 a bit 0, significato offset da cui calc. il BTA

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



3. [1.5] Nell'architettura MIPS si é deciso di avere istruzioni con una lunghezza fissa (32 bit), altre architetture hanno fatto scelte diverse (istruzioni con lunghezza variabile, ad esempio 16, 24 e 32 bit). Si forniscano un vantaggio e uno svantaggio caratteristici della scelta MIPS.

   vantaggio ..... semplicità della rete di controllo che risulta quindi particolarmente veloce

   .....

   svantaggio ..... il codice in memoria istruzioni è meno compatto (esempio: se faccio addi $t0, $t1, 1, i 16 bit della costante non sono sfruttati e quindi potrei sfruttare un istruzione a 24 bit con 8 bit per la costante)

Nome: _____ Cognome: _____

4. [1] Si descrivano brevemente gli hazard di controllo fornendo un possibile esempio di codice assembler che presenta tale problema.

   1) hazard di controllo: avvengono quando un istruzione di branch condizionato ha come operando un valore che non è ancora stato scritto nei registri

   2) esempio di codice

```
sub $t0, $t1, $t2
beq $t0, $t4, label
```

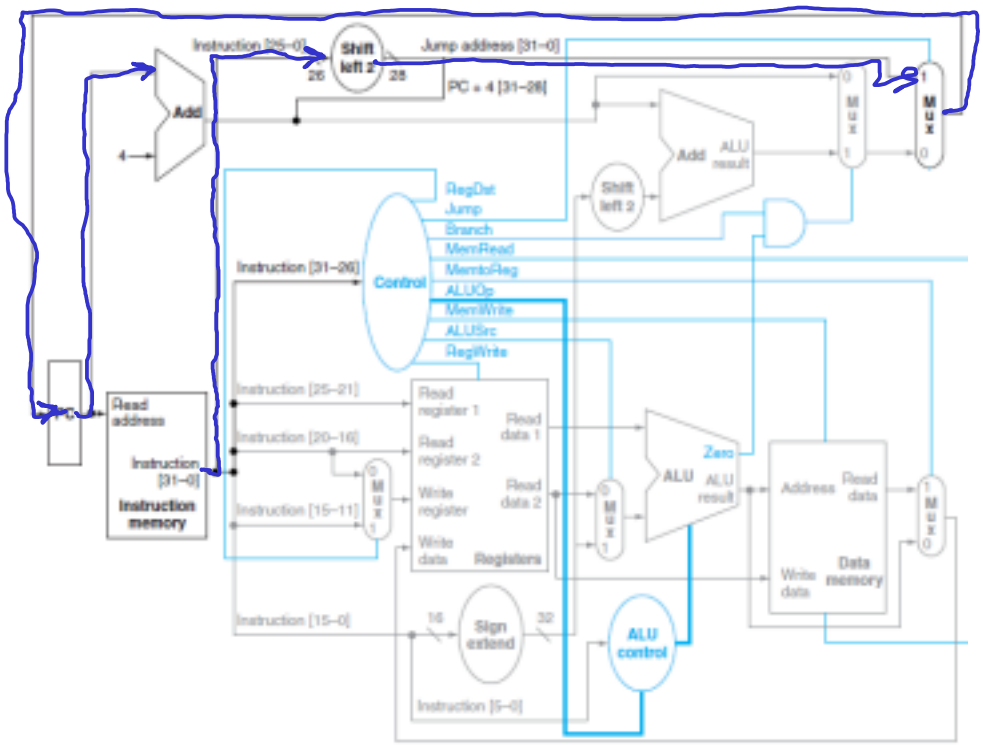
   Si descriva la tecnica piú semplice utilizzabile dal compilatore (senza richiere alcuna modifica alla CPU pipelined) in presenza di un codice iniziale contenente hazard di controllo.

   inserimento di un certo numero di stalli nella pipeline per ritardare la branch

   Si descriva una possibile tecnica utilizzabile per ridurre la perdita di prestazioni dovuta all'utilizzo della prima (si tratta sempre di una tecnica utilizzabile dal compilatore)

   delayed branch: si inseriscono dopo entrambi i rami della branch delle istruzioni che verrebbero eseguite indipendentemente dalla branch in modo da ritardarla, chiaramente non sempre è possibile

5. [1] In questa rappresentazione della micro-architettura della CPU MIPS si individui la parte di data-path che viene attivata durante l'esecuzione di un istruzione del tipo `jump` (ignorate la parte di controllo). Si annoti il relativo cammino ripassandolo con una matita colorata.



Nome: _____ Cognome: _____

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

```
lw $s0, 0($t0)
lw $s1, 4($t0)
or  $t3, $s0, $s1
```

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	lw $s0,0($t0)				
2	lw $s1,4($t0)	lw $s0,0($t0)			
3	or $t3,$s0,$s1	lw $s1,4($t0)	lw $s0,0($t0)	hazard	
4		or $t3,$s0,$s1	lw $s1,4($t0)	lw $s0,0($t0)	
5			or $t3,$s0,$s1	lw $s1,4($t0)	lw $s0,0($t0)
6				or $t3,$s0,$s1	lw $s1,4($t0)
7					or $t3,$s0,$s1
8					

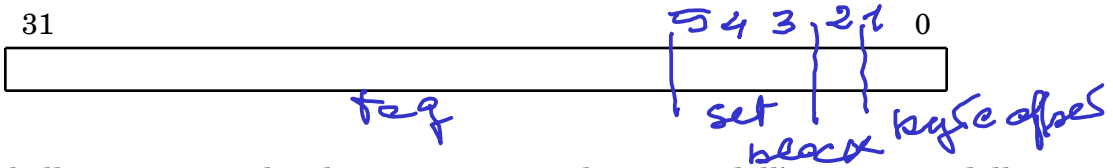
Si mostri l'esecuzione di tale frammento di codice in modo che tali dipendenze siano risolte. Si supponga a questo riguardo di considerare una pipeline che dispone di bypass fra lo stadio di WB e quello di EXE (e in cui é comunque possibile leggere e scrivere dal register file nello stesso ciclo di clock).

clock	IF	ID	EX	MEM	WB
1	lw $s0,....				
2	lw $s1,....	lw $s0,....			
3	stall	lw $s1,....	lw $s0,....		
4	or	stall	lw $s1,....	lw $s0,....	
5		or $t3,$s0,$s1	stall	lw $s1,....	lw $s0,....
6			or $t3,$s0,$s1	stall bypass	lw $s1,....
7				or	stall
8					or ....
9					
10					

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 $t0, 0x80($zero)`	hit
`lw $t1, 0x70($zero)`	miss
`lw $t2, 0x94($zero)`	miss
`lw $s0, 0x90($zero)`	hit

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	data	set
0				7
~~0~~	0x0..01	M[0x74]	M[0x70]	6
0				5
1	0x0..03	M[0xE4]	M[0xE0]	4
0				3
~~0~~	0x0..02	M[0x94]	M[0x90]	2
0				1
1	0x0..02	M[0x84]	M[0x80]	0

stato finale (quello che resta salvo)

Stato finale

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