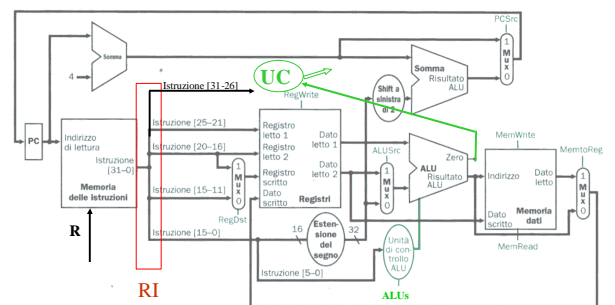
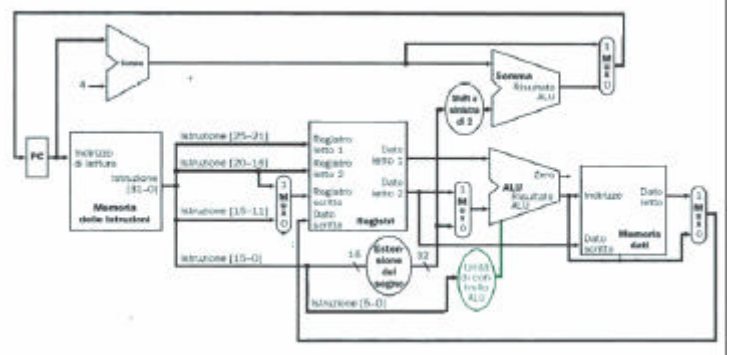
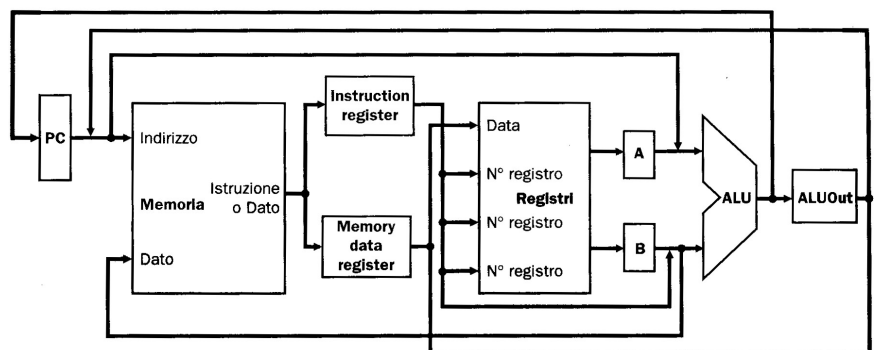


Esercitazione di ricapitolazione

1. Descrivere il ciclo di esecuzione di un'istruzione. Di quali informazioni abbiamo bisogno in ciascuna fase?
2. Descrivere il funzionamento del register file. Come occorre modificarlo per rendere leggibile immediatamente quanto viene scritto?
3. Quali sono i segnali di controllo necessari per fare funzionare questa CPU? Quanti cicli di clock sono necessari per eseguire un'istruzione? Quanto valgono i segnali di controllo per eseguire l'istruzione `lw $s0, 40($s1)`? Quali sono le unità funzionali principali associate alle varie fasi di esecuzione? Indicare sul grafico la quantità che viene calcolata dalle varie unità funzionali.
4. Progettare un controllore della ALU a 2 livelli, sapendo che le operazioni consentite sono: addi, andi, ori, add, sub, and, or, lw, sw, beq. Il primo livello riceverà in input solamente il codice operativo, il secondo livello riceverà in input l'uscita del primo livello ed il contenuto del campo funct.
5. Seguire il data_path ed il control_path per i diversi tipi di istruzioni nella CPU a ciclo singolo riportata qui a fianco. Quali segnali di controllo **non** occorre specificare per un'istruzione di `lw`? E per un'istruzione di `add`? E per un'istruzione di `addi`? Per un'istruzione di `beq`?
6. Specificare quali sono gli operandi su cui agiscono le 3 ALU e da dove provengono. Cosa contiene l'ingresso 0 del Mux del secondo operando della ALU quando sto eseguendo l'istruzione: `add $t0, $t1, $t2`, sapendo che `$t0 = $7` e che `Funct = 0x20`. Cosa contiene l'uscita del Mux all'ingresso della porta di scrittura del register file nell'istruzione `beq $t0, $t1`? E nell'istruzione `sw $t0, 0($t1)`? E nell'istruzione: `add $t0, $t1, $t2`?
7. Quando conviene utilizzare una CPU multi-ciclo e quando una a CPU a ciclo singolo? Quando conviene una CPU con pipeline rispetto ad una CPU a ciclo singolo o multi-ciclo?
8. Come viene gestito un salto dalla CPU? Disegnare il circuito che gestisce i salti.
9. Data la CPU in figura:



- Indicare quali sono i registri ed il loro ruolo.
- Indicare l'inizio e la fine delle cinque fasi del ciclo di un'istruzione.
- Specificare tutti i segnali di controllo necessari al suo funzionamento.
- Specificare anche per quali segnali di controllo si può utilizzare il segnale di clock e per quali non si può utilizzare e perchè.

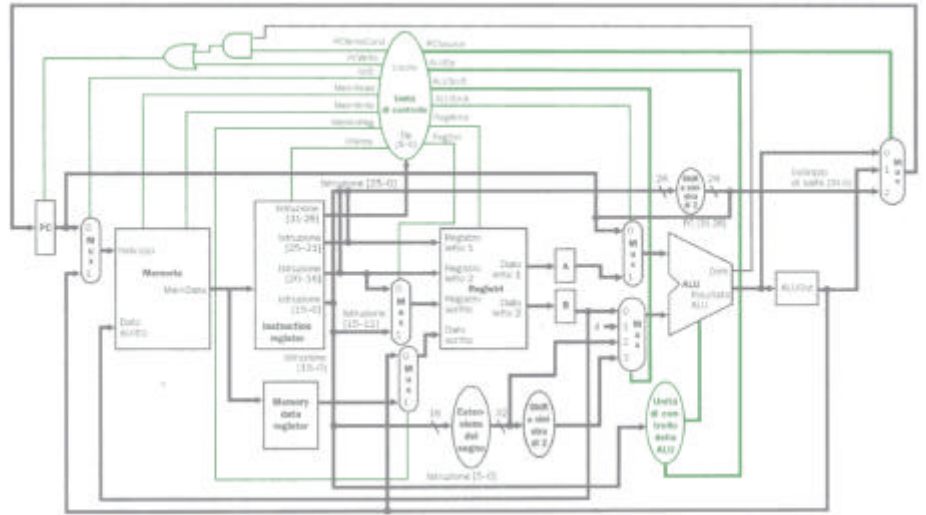


10. Specificare il contenuto (lo stato) dei registri A e B, del registro ALUOut, del Registro Istruzioni, del MDR e del PC al termine del terzo stadio di esecuzione (prima della commutazione del clock) delle istruzioni:

```
lw $t0, 0($t1)
sw $t0, 0($t1)
add $t0, $t1, $t2
beq $t0, $t1, label
addi $t0, $t1, 16
```

Sapendo che i codici operativi sono rispettivamente: 35, 43, 0, 4, 8 e che il campo funct della add è 32.

Specificare tutti i segnali di controllo che sono attivi per le cinque istruzioni nelle diverse fasi di esecuzione. Per quali segnali di controllo è possibile utilizzare il segnale di clock?



11. Cosa succede nel quarto stadio di un'istruzione di branch? Cosa succede nel quinto stadio di esecuzione di una sw? Quando il segnale di scrittura del PC è attivo? Scrivenerne la funzione logica.

12. Sintetizzare la macchina a stati finiti (STG) che controlla la CPU disegnata sopra per le seguenti istruzioni: tipo R, beq, j, lw, sw, addi, subi, sli.

13. Modificare la CPU di cui sopra, perchè sia in grado di gestire le eccezioni di overflow e di codice operativo non valido. Cosa è un'eccezione e un'interruzione? Che ruolo hanno i registri EPC, il registro Causa, stato e la maschera di interruzione? Quali sono le due modalità di risposta ad un'interruzione o eccezione? Cosa è il coprocessore 0?

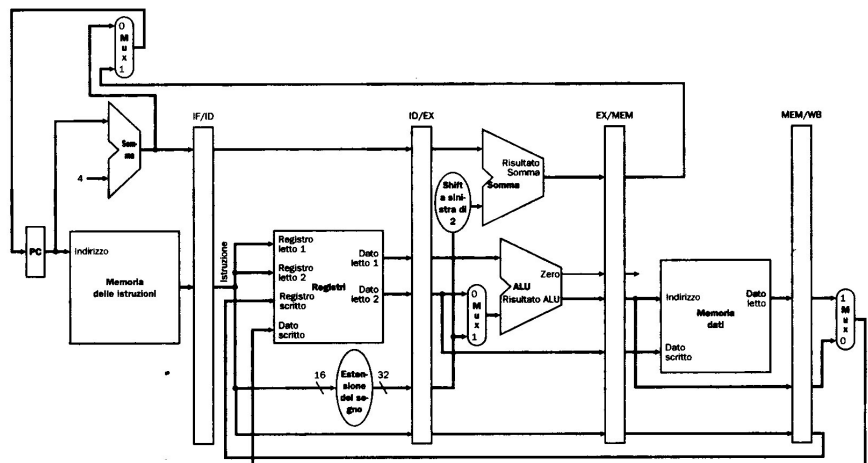
14. Cosa è una CPU con pipe-line? Una pipe-line consente l'esecuzione più veloce di un'istruzione rispetto ad una CPU a singolo ciclo? Di quanto aumenta la velocità di esecuzione in una CPU con pipeline? Una CPU con pipeline richiede più o meno unità funzionali di una CPU a ciclo singolo? E di una CPU multi-ciclo? Motivare le risposte.

15. Modificare la CPU di cui sopra in modo tale che diventi compatibile all'esecuzione in pipe-line (senza gestione di hazard). Modificare la CPU in due modi diversi: tenendo conto dei segnali di controllo e non tenendone conto. Quali diventano i registri? Cosa contengono? Quali passi di esecuzione separano? Da quanti bit sarà costituito ciascun registro? Cos'è uno stallo?

16. Dato lo schema a fianco, quale sarà il contenuto dei registri di pipeline (stato), al termine (prima della commutazione del clock) del terzo stadio dell'istruzione in bold (sub):

```
add $t0, $t1, $t2
sub $t3, $t3, $t5
beq $t6, $t0, 16
add $t0, $t1, $t3
```

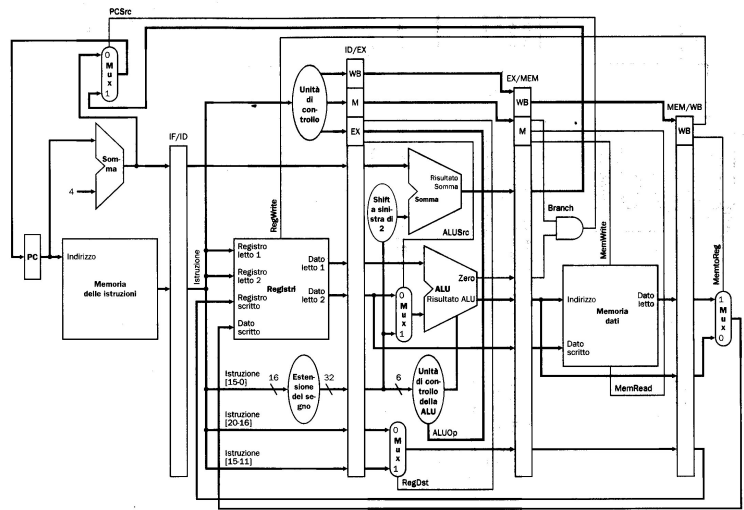
sapendo che \$t0 = 7, i codici operativi di add e sub = 0, beq = 4; il codice Funct della add è 32 e della sub è 34.



17. Dato lo schema a fianco, quale saranno i segnali di controllo attivi e quali indifferenti quando la pipeline si trova al termine (prima della commutazione del clock) del terzo stadio dell'istruzione in bold (sub):

```
add $t0, $t1, $t2
sub $t3, $t3, $t5
beq $t6, $t0, 16
add $t0, $t1, $t3
```

sapendo che \$t0 = 7, i codici operativi di add e sub = 0, beq = 4; il codice Funct della add è 32 e della sub è 34.



18. Cosa rappresenta un hazard? Quando si verifica? Fare un esempio per ogni tipo di hazard.

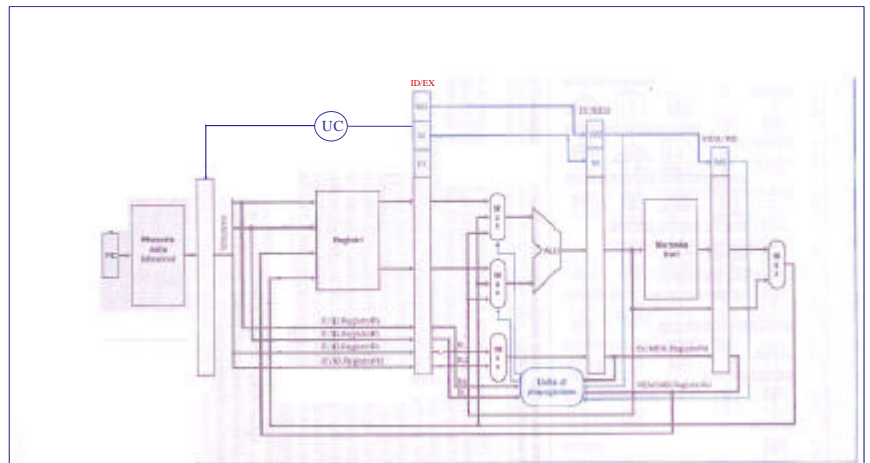
19. Visualizzare con uno schema temporale e con un esempio, quali sono le dipendenze tra le istruzioni che provocano un hazard sui dati o un hazard sul controllo.

20. Modificare lo schema della CPU con pipeline riportato sopra per potere gestire: un hazard sui dati dovuto ad istruzioni aritmetico-logiche. Scrivere le condizioni logiche che vengono utilizzate per identificare questo hazard e le funzioni logiche che servono a risolverlo.

21. Modificare lo schema della CPU, in modo tale da potere gestire:

- a) Un hazard dovuto ad un'istruzione di load.
- b) Un hazard sul controllo dovuto ad una beq.

Dare un esempio di codice in cui questi due hazard si verificano. E spiegare la dipendenza tra dati che origina l'hazard.



22. Dato lo schema qui a fianco, specificare il contenuto di tutti i registri ed i segnali di controllo negli stadi 1, 2, 3, 4, 5 di esecuzione della lw:

```
add $t0, $t1, $t2
addi $t0, $t1, 64
beq $t3, $t4, 16
lw $t3, 0($t0)
add $t4, $t5, $t3
add $t6, $t6, $t7
```

sapendo che i codici operativi di add, addi, beq, lw sono rispettivamente: 0, 8, 4, 35. Il codice funct della add è 32 e che \$t0 = \$7.

23. Cos'è il branch prediction buffer? Cosa si intende per pipeline superscalare, superpipeline e scheduling dinamico? Che cosa sono le reorder station? E le reservation station? A cosa serve l'operazione di renaming dei registri? E' corretto dire che internamente la CPU di un Pentium IV ha un'ISA RISC e perchè? Quante cache primarie ha un Pentium IV e perchè?

24. Qual'è il ruolo della memoria? Quali funzioni si possono eseguire sulla memoria? Cosa rappresenta l'altezza e l'ampiezza della memoria e come si calcola la capacità? Come è definita la parola di memoria? Qual'è la relazione tra capacità di memoria e numero di bit di indirizzamento?
25. Cosa esprime il principio di località di una memoria? Cosa contiene una memoria cache? Può il contenuto di una memoria cache essere diverso dal contenuto della memoria principale?
26. Cosa si intende per Hit e Miss? Hit rate e miss rate? Può essere la somma di Miss rate e Hit rate maggiore di 15? Cosa si intende per Write through e Write back in una memoria cache?
27. Data una memoria cache di 64Kbyte ed una RAM di 1Gbyte, a quanti bit devo dimensionare il campo TAG in caso di una memoria cache a mappatura diretta? E in caso di una cache a mappatura diretta a 2 vie?
28. Dato un indirizzo di memoria di 32 bit, specificare il significato dei singoli bit nel caso di utilizzo di una memoria cache con le seguenti caratteristiche:
- Cache a mappatura diretta di 128Kbyte, con linee contenenti 8 parole di 4 byte ciascuna.
 - Cache a 2 vie di 128Kbyte, con 2 banchi e linee contenenti 8 parole di 4 byte ciascuna.
 - Cache associativa di 128Kbyte con linee ciascuna contenente 8 parole di 4 byte ciascuna.
- Per ciascuna delle 3 cache disegnare il circuito di lettura e scrittura.
29. Disegnare le seguenti tre cache:
- Cache a mappatura diretta di 128 byte con linee contenenti 2 parole di 4 byte.
 - Cache a 2 vie di 128 byte con linee contenenti 2 parole di 4 byte.
 - Cache associativa con linee contenenti 2 parole di 4 byte.
- Data l'istruzione `lw $t0, 1024($t1)`, specificare all'interno delle cache a), b), c) dove si trova la parola che deve essere letta?
30. Specificare per una memoria cache a chi viene inviato: il segnale di Miss, il segnale di Hit, il dato letto; e da dove proviene il dato scritto.
31. Cos'è l'interleaving di una memoria? Come è costruita una memoria SRAM? Cosa significa l'acronimo SRAM? Qual'è il ruolo dell'uscita "three-state" in una memoria SRAM? Discuterlo con un esempio. Come vengono gestiti i banchi di memoria?
32. Qual'è il principio di funzionamento di una DRAM? Cosa è una SDRAM? Quando una SDRAM lavora in "burst mode"? Cosa rappresentano i segnali CAS e RAS di una DRAM?
33. Su cosa è basato un codice di rilevazione di errore? Come funziona un codice di correzione di errore di una memoria?
34. Cos'è un bus? Cos'è l'arbitraggio? Cosa rappresenta il segnale di "bus grant"? Descrivere gli schemi di arbitraggio centralizzato e gli schemi di arbitraggio distribuiti (con e senza autoselezione).
35. Descrivere i requisiti di funzionamento dei tre tipi di bus principali: processore-memoria, backplane e I/O. Come viene sincronizzata la trasmissione di dati sul bus?
36. Cosa è il device controller? Quali sono i suoi componenti principali? A cosa serve? Come vengono indirizzate le periferiche?
37. Come viene gestito l'I/O a controllo di programma? Cos'è il polling? Come viene gestito l'I/O tramite interrupt? Cosa rappresenta la maschera di interrupt? Cosa è il DMA? Che cos'è lo "spin lock"?

38. Esercizio. Supponiamo di valutare il costo per una CPU con frequenza pari a 1Ghz per trasferire 64Mbyte di dati. Da: Hard-disk. Trasferimento di 64 byte per accesso. Tempo di CPU per un'operazione di I/O: 200 cicli_clock. Per ogni DMA vengono trasferiti 6,400 parole.

Valutiamo le diverse modalità di trasferimento:

a) a controllo di programma (300 cicli di clock)

b) ad interrupt (400 cicli di clock)

c) mediante DMA (500 cicli clock per l'avviamento e 800 cicli di clock per la chiusura). A controllo di programma: 64Mbyte / 64byte = 1M accessi. Tempo di CPU: 1Maccessi * (200+300)cicli = 500 * 10⁸. Percentuale di sfruttamento della CPU: 50%.

- Mediante interrupt: 64Mbyte / 64 byte = 1M interrupt. Tempo di CPU: 1Minterrupt * (200+400) cicli di clock. Percentuale di utilizzo della CPU: 60%.
- Mediante DMA: 64Mbyte / 6400 byte = 10,000 DMA => 10,000 * (500 + 800) = 13 Mega cicli di clock. Percentuale di utilizzo della CPU: 1,3%.

Ripetete le valutazioni per trasferire dati da tastiera (i quali richiedono una frequenza di 10byte/s, ed un tempo di CPU di 8 cicli_clock / byte).

- A controllo di programma: 10 accessi / s * (200 + 300) cicli_clock = 5,000 cicli_clock. Percentuale di sfruttamento della CPU: 5 / 10⁶.
- Mediante interrupt: 10 accessi / s * (200 + 400) cicli_clock = 6,000 cicli_clock. Percentuale di utilizzo della CPU: 6 / 10⁶.
- Mediante DMA. Non ha senso. Trasferisco 1 byte alla volta.

39. Cosa si intende per rete a commutazione di pacchetto o di circuito? Descrivere lo stack ISO-OSI. Descrivere lo stack TCP-IP. Cosa si intende per collisione di pacchetti? Come viene gestito? Descrivere lo header di IP. Descrivere il frame Ethernet.

40. Descrivere l'utilizzo principali dei registri nelle architetture IA-32. Cosa sono i segmenti? Modalità di indirizzamento dei dati nelle architetture IA-32. Come vengono gestite le operazioni di I/O nelle IA-32? Come funzionano i modificatori delle istruzioni dell'ISA delle architetture IA-32?

41. Cosa è lo SPEC? Cosa è il CPI? Perché sono nati i benchmark? Enunciare la legge di Amdhal. Sotto quali ipotesi riesco ad aumentare la velocità di esecuzione in architetture multi-processori? E la velocità di un server?

42. Si deve valutare un miglioramento di una macchina per l'aggiunta di una modalità vettoriale. La computazione vettoriale è 20 volte più veloce di quella normale. La *percentuale di vettorizzazione* è la porzione del tempo che può essere spesa usando la modalità vettoriale.

- Disegnare un grafico che riporti lo speedup come percentuale della computazione effettuata in modo vettoriale.
- Quale percentuale di vettorizzazione è necessaria per uno speedup di 2?
- Quale per raggiungere la metà dello speedup massimo?

La percentuale di vettorizzazione misurata è del 70%. I progettisti hardware affermano di potere raddoppiare la velocità della parte vettoriale se vengono effettuati significativi investimenti. Il gruppo che si occupa dei compilatori può incrementare la percentuale d'uso della modalità vettoriale.

- Quale incremento della percentuale di vettorizzazione sarebbe necessario per ottenere lo stesso guadagno di prestazioni?
- Quale investimento raccomandereste?