

Architettura degli Elaboratori

Tutorato

a cura di Manuela Flores

19/12/2022 ore 8:30 – 11



Barbara Masucci – Classe: Resto_1

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

DIPARTIMENTO DI ECCELLENZA

domanda 1

Si assuma che le variabili *i* e *k* corrispondano ai registri *\$s0* e *\$s1*. Qual è l'istruzione C corrispondente al seguente codice *assembler* MIPS?

```
Label: bne $s0, $s1, Exit
      sll $s0, $s0, 1
      j Label
```

Exit: ...

A. while(*i*==*k*) *i*=*i**2;

B. if(*i*==*k*) *i*=*i**2;

C. while(*i*!=*k*) *i*=*i**2;

D. Nessuna delle risposte precedenti.



Lezione 13 pag. 3

Prendere decisioni

- Talvolta si ha la necessità di alterare il flusso di un programma **al verificarsi di certe condizioni**
- Il MIPS supporta questa necessità fornendo **istruzioni di salto condizionato**
 - **Branch if Equal: `beq reg1, reg2, L1`**
 - Salta all'istruzione con etichetta L1 se il contenuto del registro reg1 è uguale al contenuto del registro reg2
 - **Branch if Not Equal: `bne reg1, reg2, L1`**
 - Salta all'istruzione con etichetta L1 se il contenuto del registro reg1 è diverso dal contenuto del registro reg2
- Inoltre, il MIPS offre una istruzione di **salto incondizionato**
 - **Jump: `j L1`**
 - Salta all'istruzione con etichetta L1



domanda 2

Il banco di registri (*register file*) come studiato è formato da:

- A. 32 registri ognuno contenente un dato a 32 bit
- B. 2^{32} registri ognuno contenente un dato a 8 bit
- C. 32 registri ognuno contenente un dato a 8 bit
- D. Nessuna delle risposte precedenti.



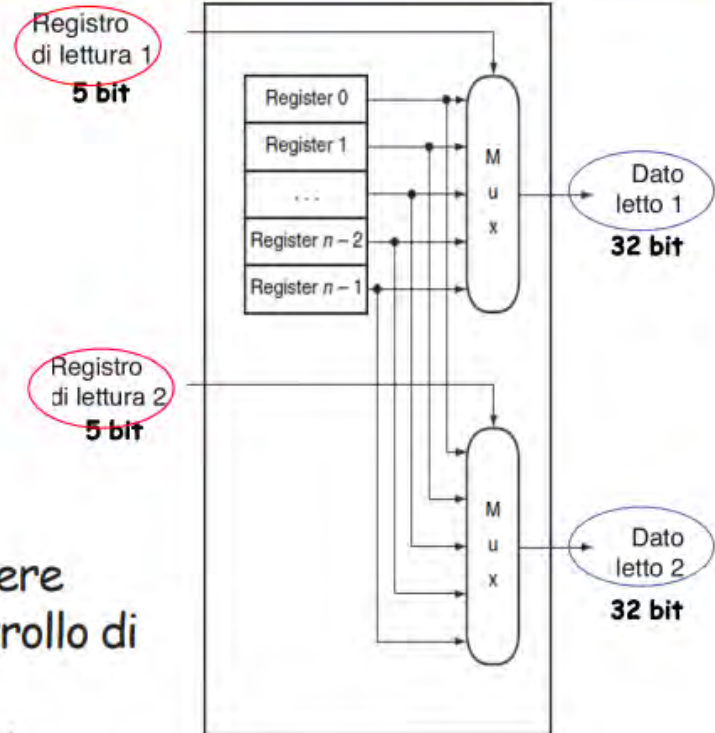
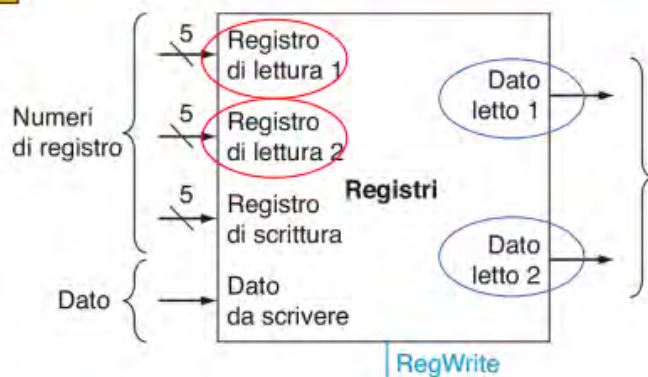
Banco di registri

- Detto anche **Register File**, costituisce l'insieme dei **32 registri del MIPS**
- Ciascun registro è un array di 32 Flip-Flop di tipo D e può essere letto o scritto
 - Per **leggere** un registro occorre specificare in input il **numero del registro** (**Register number**, 5 bit), ottenendo in output il suo contenuto (**Read data**, 32 bit)
 - Per **scrivere** un registro occorre specificare in input il **numero del registro** (**Register number**, 5 bit) in cui scrivere e il **dato da scrivere** (**Write data**, 32 bit)



Lezione 17 pag. 13

Unità funzionali: Banco dei Registri



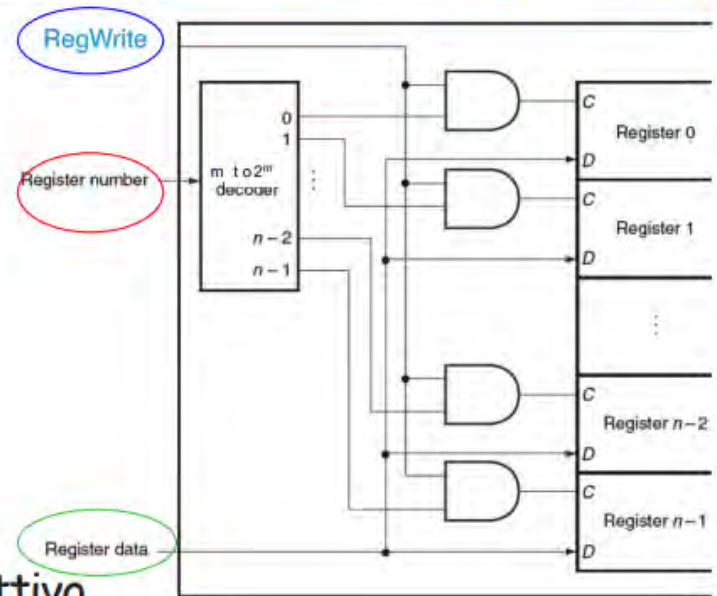
➤ Lettura

- I numeri dei due registri da leggere (5 bit) diventano i segnali di controllo di due MUX $2^5:1$
- Ciascun MUX fornisce in output il contenuto del registro corrispondente (32 bit)



Lezione 17 pag. 14

Unità funzionali: Banco dei Registri



➤ Scrittura

- Il segnale **RegWrite** deve essere attivo
- Il **numero del registro da scrivere** (5 bit) va in input a un decoder 5-to-32 che seleziona il registro da scrivere
- Il **dato da scrivere** (32 bit) va in input ai 32 flip-flop D che costituiscono quel registro (il cui stato viene modificato)



domanda 3

L'Unità di controllo principale del processore a singolo ciclo studiato è

- A. Un circuito combinatorio che riceve in ingresso il codice operativo (*op*) dell'istruzione da eseguire
- B. Un circuito sequenziale che riceve in ingresso il codice operativo (*op*) dell'istruzione da eseguire
- C. Un circuito combinatorio che riceve in ingresso il codice funzionalità (*funct*) dell'istruzione da eseguire
- D. Nessuna delle risposte precedenti



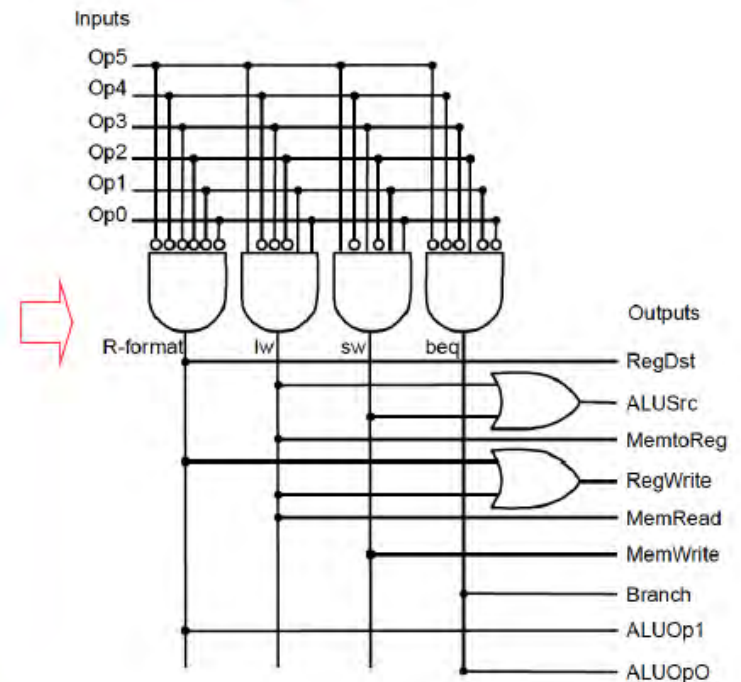
Lezione 19 pag. 25

Unità di controllo Implementazione

Tabella di verità dell'unità di Controllo

	Segnale	form. lw sw beq			
		R	lw	sw	beq
Inputs = Opcode	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
	RegDst	1	0	x	x
Outputs	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Rete Combinatoria realizzabile
tramite PLA



domanda 4

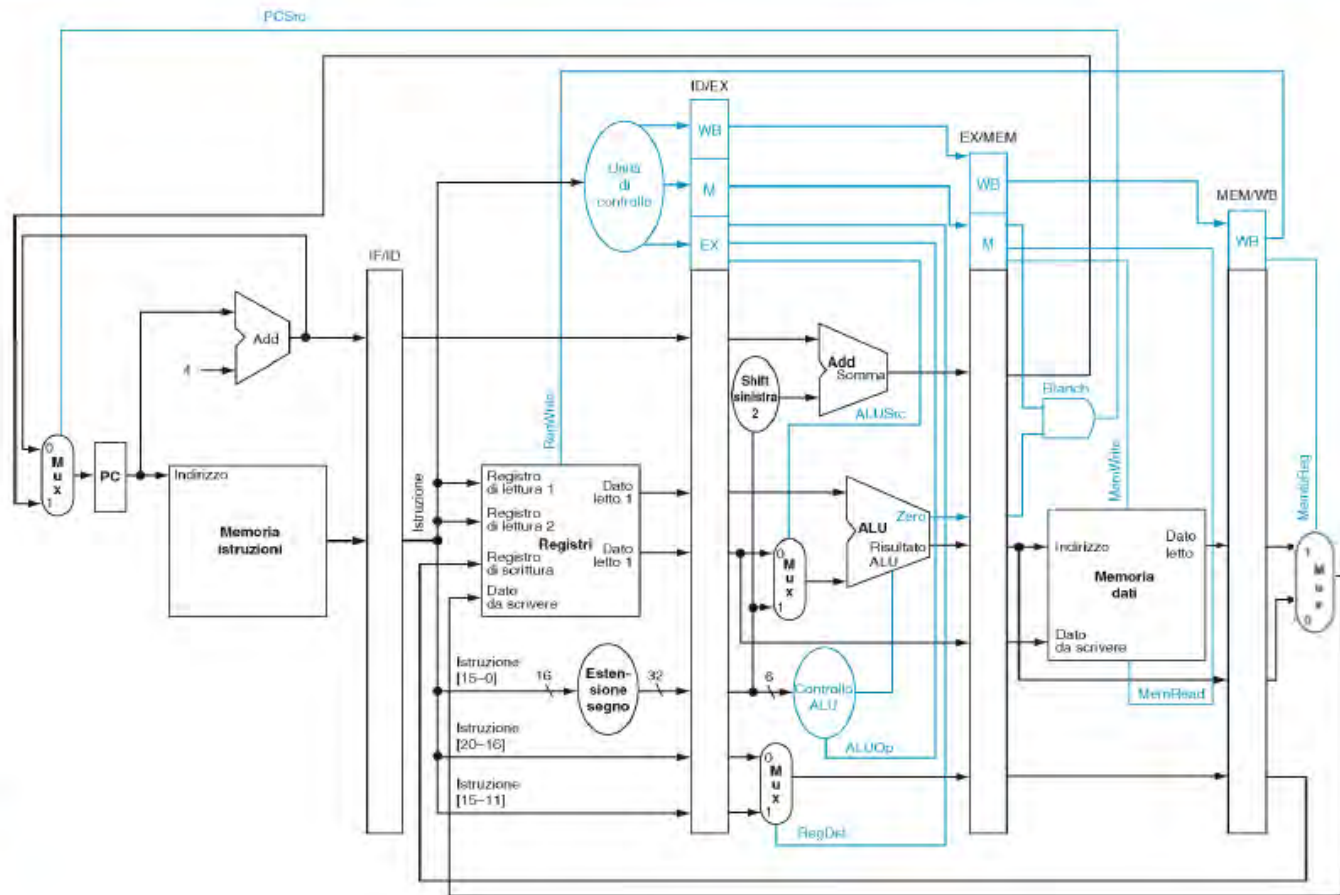
Si consideri l'implementazione studiata del processore con pipeline.
Cosa fa l'istruzione `sw $s1, 24($s2)` nello stadio MEM?

- A. Esegue la somma del contenuto del registro `$s2` con 24
- B. Scrive nel registro `$s1` il dato prelevato dalla memoria
- C. Preleva l'istruzione dalla memoria istruzioni.
- D. Nessuna delle risposte precedenti



Lezione 22 pag. 3

Unità di elaborazione con pipeline e controllo



domanda 5

Nella seguente sequenza di istruzioni per una pipeline a 5 stadi come studiata:

sub \$t0,\$t1,\$t2

and \$s1,\$t0,\$s2

- A. Non vi sono hazard
- B. Vi è un hazard sui dati che può essere risolto con la propagazione
- C. Vi è un hazard sui dati che non può essere risolto con la propagazione
- D. Nessuna delle risposte precedenti



Lezione 22 pag. 18

Rilevare gli hazard

- Le condizioni che generano hazard sui dati sono di 4 tipi diversi
 1. EX/MEM.RegistroRd = ID/EX.RegistroRs
 2. EX/MEM.RegistroRd = ID/EX.RegistroRt
 3. MEM/WB.RegistroRd = ID/EX.RegistroRs
 4. MEM/WB.RegistroRd = ID/EX.RegistroRt
- Quando si verificano i primi due tipi di hazard?
 - Quando uno dei due operandi (rs, rt) di una istruzione che si trova nello stadio ID e deve entrare nello stato EX (ed è quindi presente nel registro ID/EX) dipende dal risultato di una istruzione precedente che viene prodotto nello stadio EX (ed è quindi presente nel registro EX/MEM)



Lezione 22 pag. 21

Propagazione

	Tempo (in cicli di clock) →								
	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7	CC 8	CC 9
Contenuto del registro \$2:	10	10	10	10	10/-20	-20	-20	-20	-20
Contenuto del registro EX/MEM:	X	X	X	-20	X	X	X	X	X
Contenuto del registro MEM/WB:	X	X	X	X	-20	X	X	X	X

Ordine di esecuzione
del programma
(sequenza
delle istruzioni)

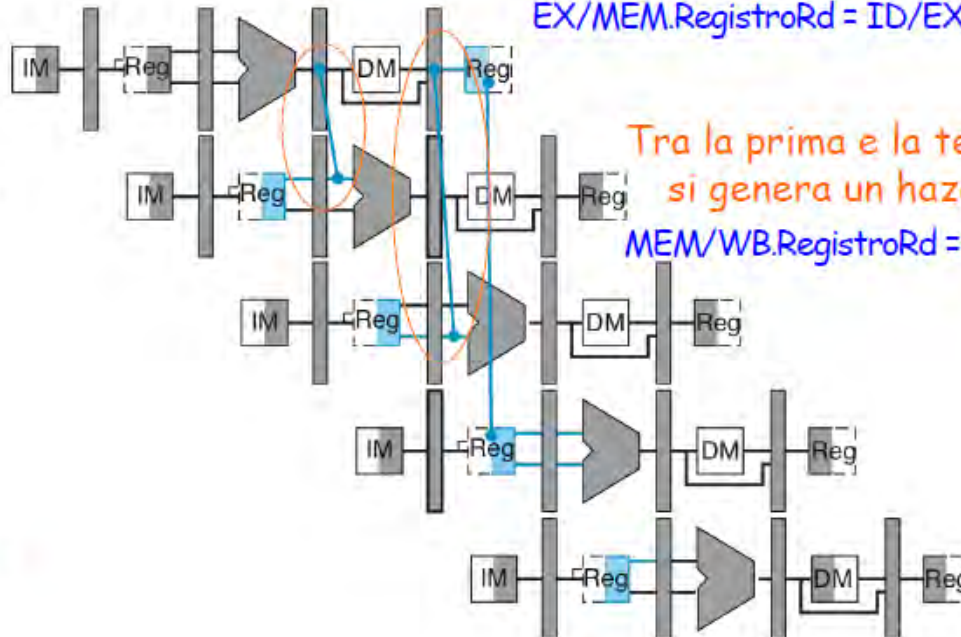
sub \$2, \$1, \$3

and \$12, \$2, \$5

or \$13, \$6, \$2

add \$14, \$2, \$2

sw \$15, 100(\$2)



Tra le prime due istruzioni si genera
un hazard di tipo 1

EX/MEM.RegistroRd = ID/EX.RegistroRs

Tra la prima e la terza istruzione
si genera un hazard di tipo 4

MEM/WB.RegistroRd = ID/EX.RegistroRt



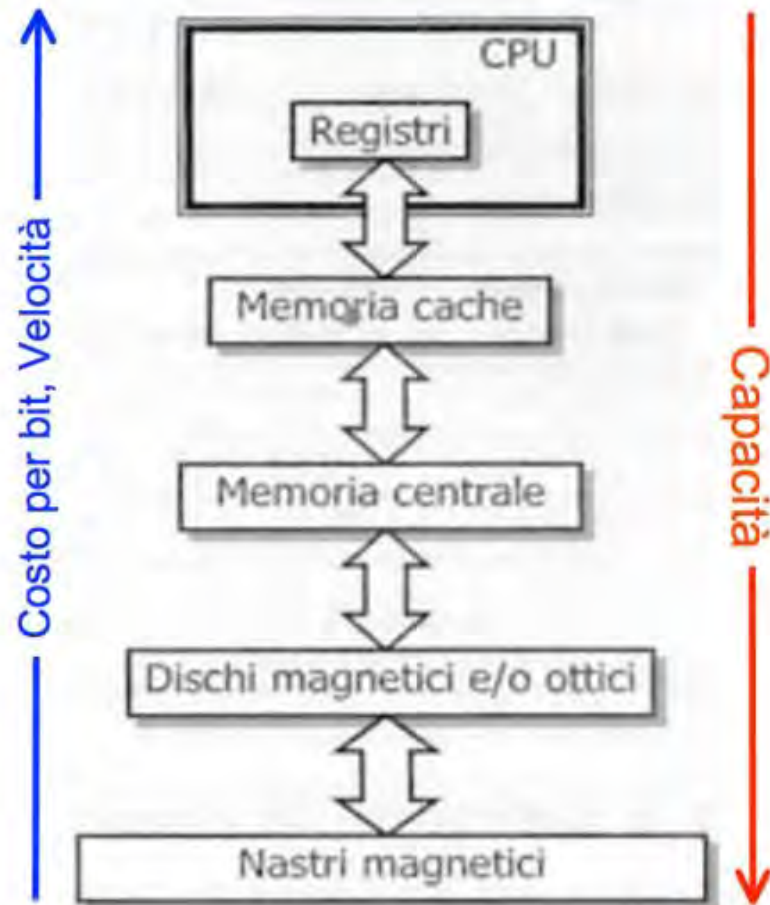
domanda 5bis

In una gerarchia delle memorie troveremo più vicina al processore

- A. Una memoria più veloce e di maggiore capacità
- B. Una memoria più veloce e di minore capacità
- C. Una memoria di minor costo
- D. Nessuna delle risposte precedenti



Gerarchia di memoria



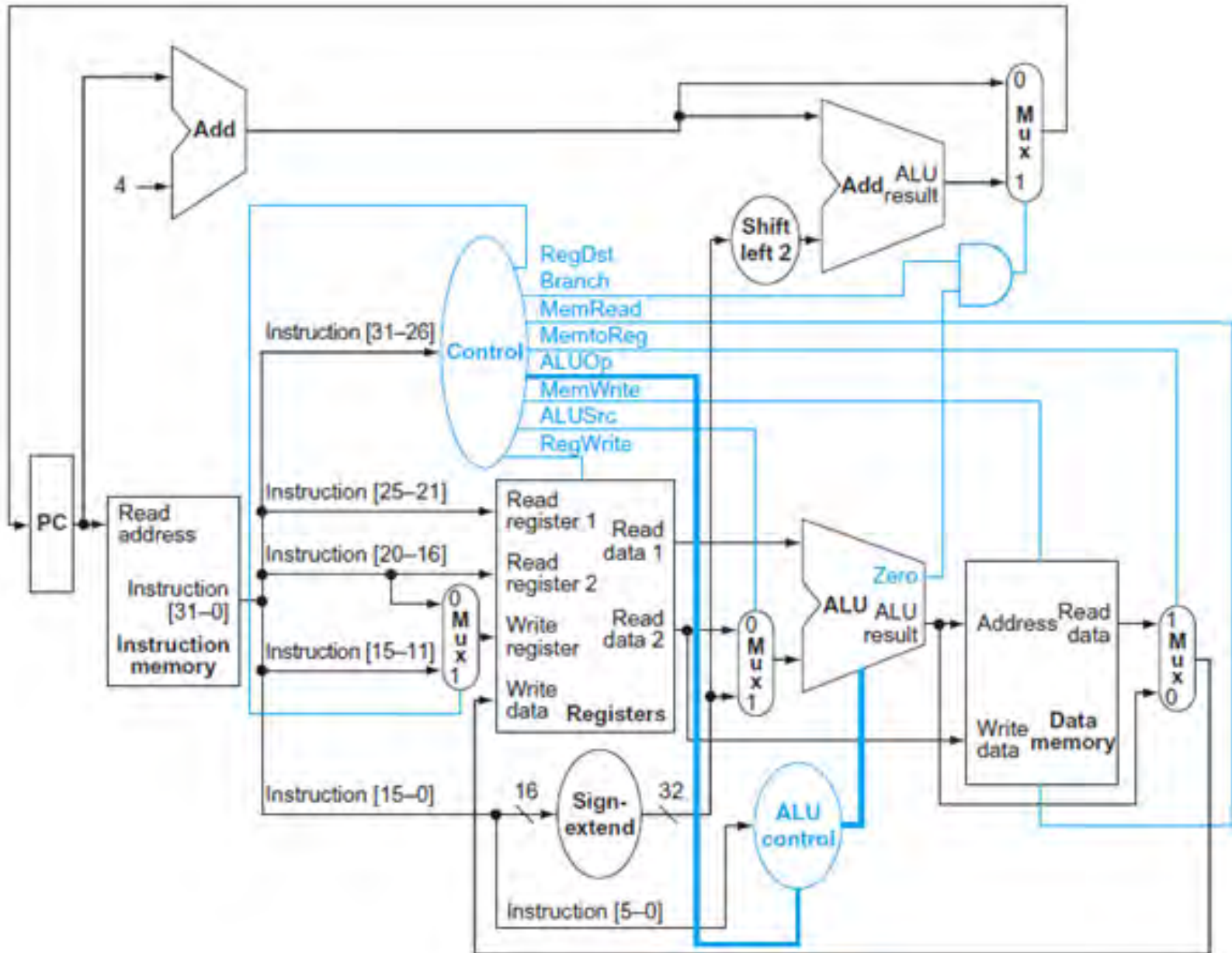
domanda 6

Nello schema di implementazione studiata per il processore MIPS, il *multiplexer* posto all'uscita della memoria dati permette di scegliere fra:

- A. Il secondo dato letto dal banco dei registri e l'estensione del segno dei 16 bit meno significativi dell'istruzione
- B. Il secondo dato scritto nel banco dei registri e l'estensione del segno dei 16 bit più significativi dell'istruzione
- C. Il dato letto dalla memoria dati e l'uscita della ALU.
- D. Nessuna delle risposte precedenti



Lezione 19 pag. 9



domanda 7

La Unità di Controllo dell'ALU riceve in ingresso esclusivamente:

- A. I segnali AluOp1, AluOp0
- B. I 6 bit più significativi dell'Istruzione
- C. Entrambe le risposte precedenti
- D. Nessuna delle risposte precedenti

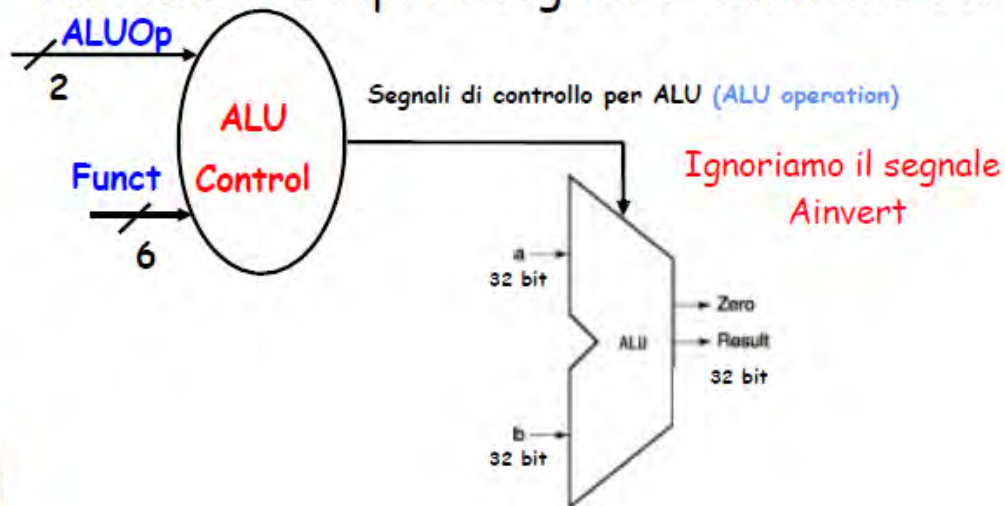


Lezione 18 pag. 5

La ALU Control

Per generare i segnali di controllo per la ALU viene utilizzata una **piccola unità di controllo (ALU Control o Contollore ALU)** che

- Riceve in ingresso il campo **funct** dell'istruzione e un campo di controllo su 2 bit, detto **ALUOp**, che indica l'operazione da eseguire
- Fornisce in output i segnali di controllo della ALU



domanda 7bis

Con riferimento alla ALU (finale, a 32 bit) studiata, come occorre porre, rispettivamente, i segnali di controllo AInvert, BNegate e Operation, per eseguire una sottrazione?

A. 0, 1, 10

B. 0, 1, 11

C. 1, 1, 1

D. Nessuna delle risposte precedenti

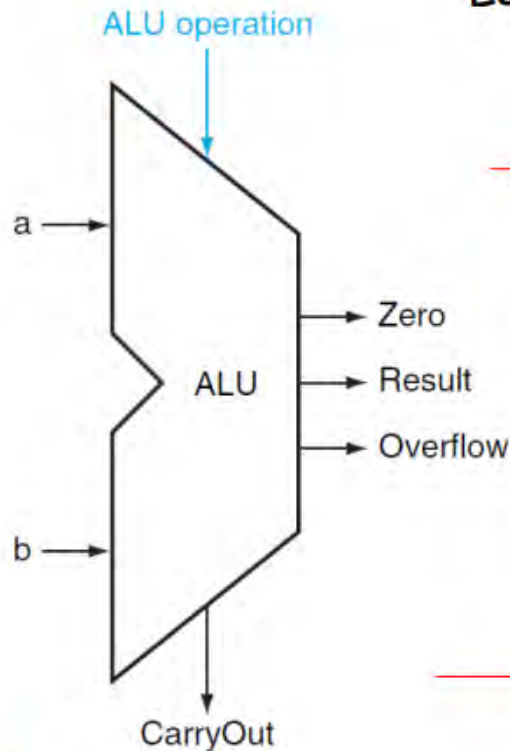


Lezione 18 pag. 3

La ALU del MIPS

La ALU prevede le 6 seguenti combinazioni dei suoi **4 segnali di controllo**

~~Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)~~



ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

Ignoriamo l'istruzione NOR (e il segnale Ainvert) perché non fa parte del set di istruzioni considerate



domanda 8

Il pipelining:

- A. Riduce il tempo per eseguire una singola istruzione
- B. Riduce il throughput
- C. Aumenta il numero di istruzioni eseguibili nell'unità di tempo
- D. Nessuna delle risposte precedenti



Osservazioni

- L'implementazione con pipeline aumenta il **numero di istruzioni contemporaneamente in esecuzione**
- Quindi, la pipeline
 - Aumenta il **throughput**
(numero di istruzioni eseguite nell'unità di tempo)
 - Non diminuisce la **latenza**
(tempo di esecuzione della singola istruzione), che resta sempre la stessa

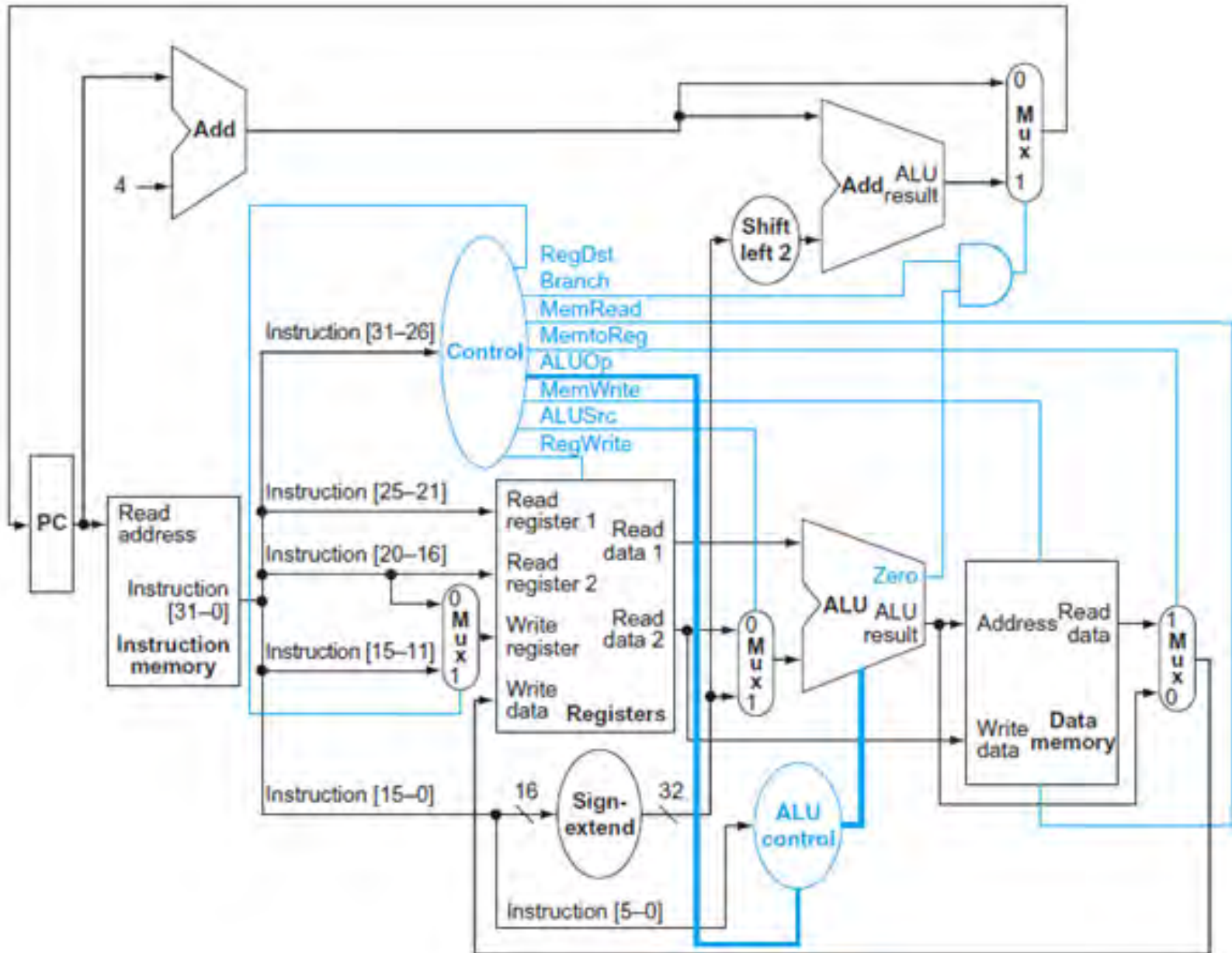


domanda 9

Per eseguire `add` e `lw` il segnale di controllo `RegWrite` deve essere posto, rispettivamente, a:

- A. 1 e 0 B. 0 e 1 C. 0 e 0 D. Nessuna delle risposte precedenti





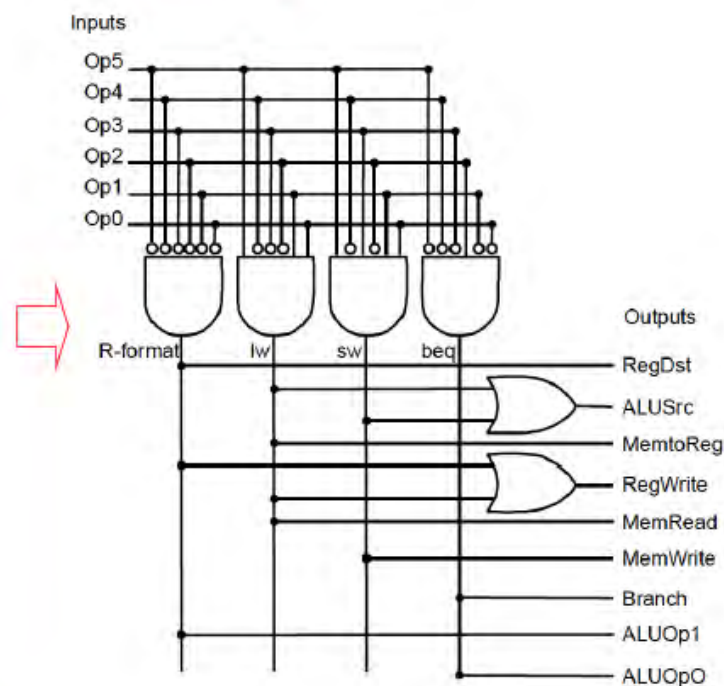
Lezione 19 pag. 25

Unità di controllo Implementazione

Tabella di verità dell'unità di Controllo

Inputs = Opcode	Segnale	form. R	lw	sw	beq
Op5 Op4 Op3 Op2 Op1 Op0	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	x	x
	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1
	ALUOp0	0	0	0	1

Rete Combinatoria realizzabile
tramite PLA



domanda 10

Un banco di registri (register file) ha come uscite:

- A. Il dato letto e il dato scritto (entrambi a 32 bit)
- B. I due dati letti (entrambi a 32 bit)
- C. Il dato letto (32 bit) e un segnale di controllo a 1 bit
- D. Nessuna delle risposte precedenti



Lezione 17 pag. 12

Unità funzionali: Banco dei Registri

Il Banco dei Registri

- E' un insieme di registri che possono essere letti o scritti
- Input:
 - I numeri dei due registri da leggere
 - Il numero del registro da scrivere
 - Il dato da scrivere
- Output:
 - I contenuti (dati) dei due registri letti
- La scrittura di un dato in ingresso viene controllata dal segnale **RegWrite** che deve essere posto a 1 per poter scrivere nel registro



domanda 11

In un flip-flop D se $\text{Clock}=1$ e $D=1$

- A. Lo stato Q sarà uguale allo stato precedente
- B. Lo stato Q sarà 1

- C. Lo stato Q sarà 0
- D. Nessuna delle risposte precedenti.

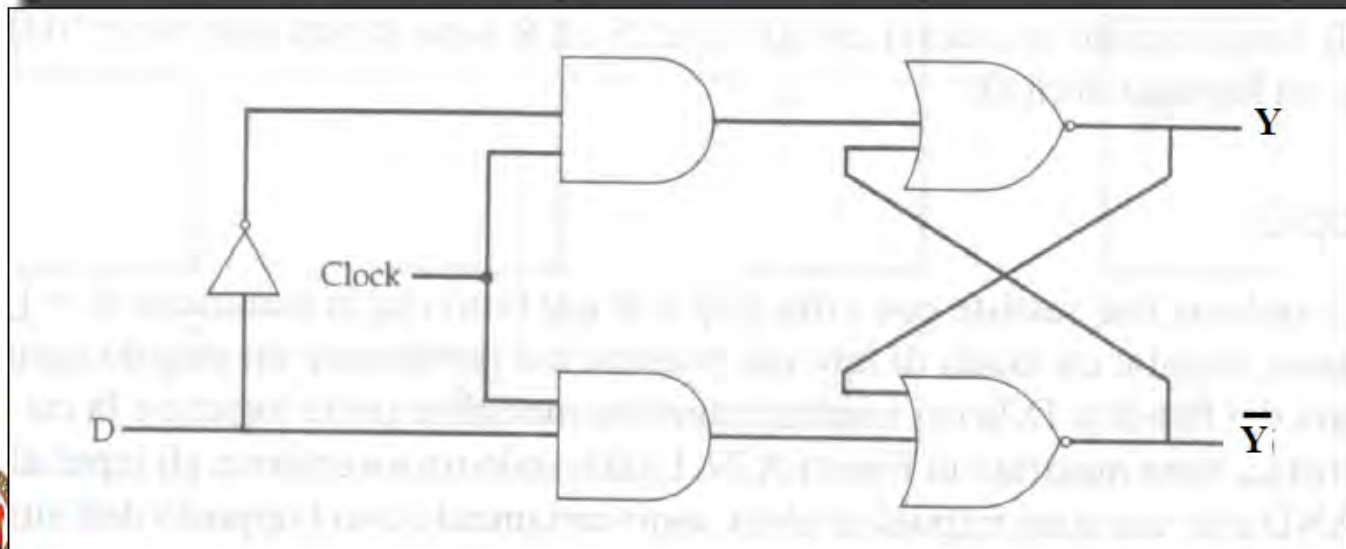


Lezione 16 pag. 24

Flip-Flop D

L'output del Flip-Flop D corrisponde al suo input

- Infatti, quando $D=1$, $S=1$ e $R=0$, quindi il prossimo stato del Flip-Flop (output) sarà $Y=1$
- Analogamente, quando $D=0$, $S=0$ e $R=1$, quindi il prossimo stato del Flip-Flop (output) sarà $Y=0$



Lezione 16 pag. 25

Flip-Flop D: **Tavola di verità**

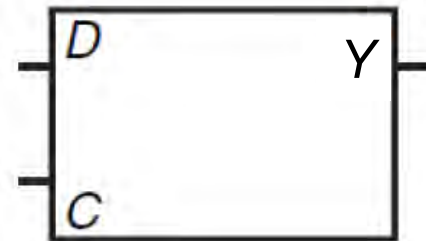
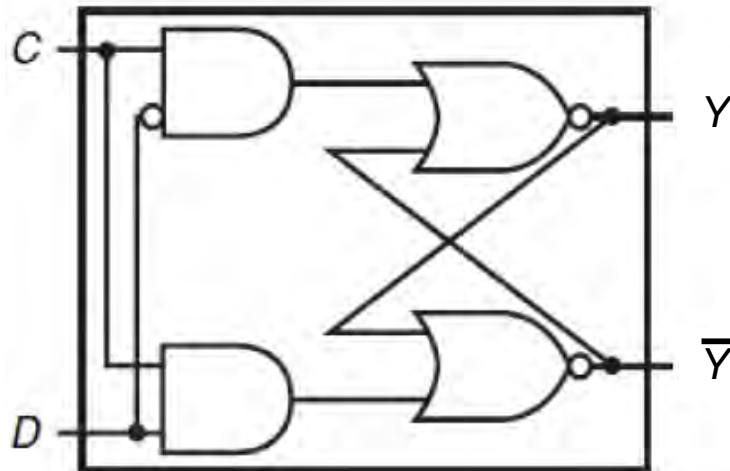
D	S	R	Stato attuale	Prossimo stato
1	1	0	0	1
1	1	0	1	1
0	0	1	0	0
0	0	1	1	0

L'output (prossimo stato) del Flip-Flop D
corrisponde al suo input



Flip-Flop D

- Quando il Clock è 1, cioè il suo fronte è **alto**, il circuito è **aperto** e l'output Y diventa uguale all'input D (il circuito lascia passare il dato D)
- Quando il Clock è 0, cioè il suo fronte è **basso**, il circuito è **chiuso** e l'output Y mantiene il valore che aveva l'ultima volta che il circuito era aperto (il circuito non lascia passare il dato D)



domanda 12

Si consideri la seguente sequenza di istruzioni:

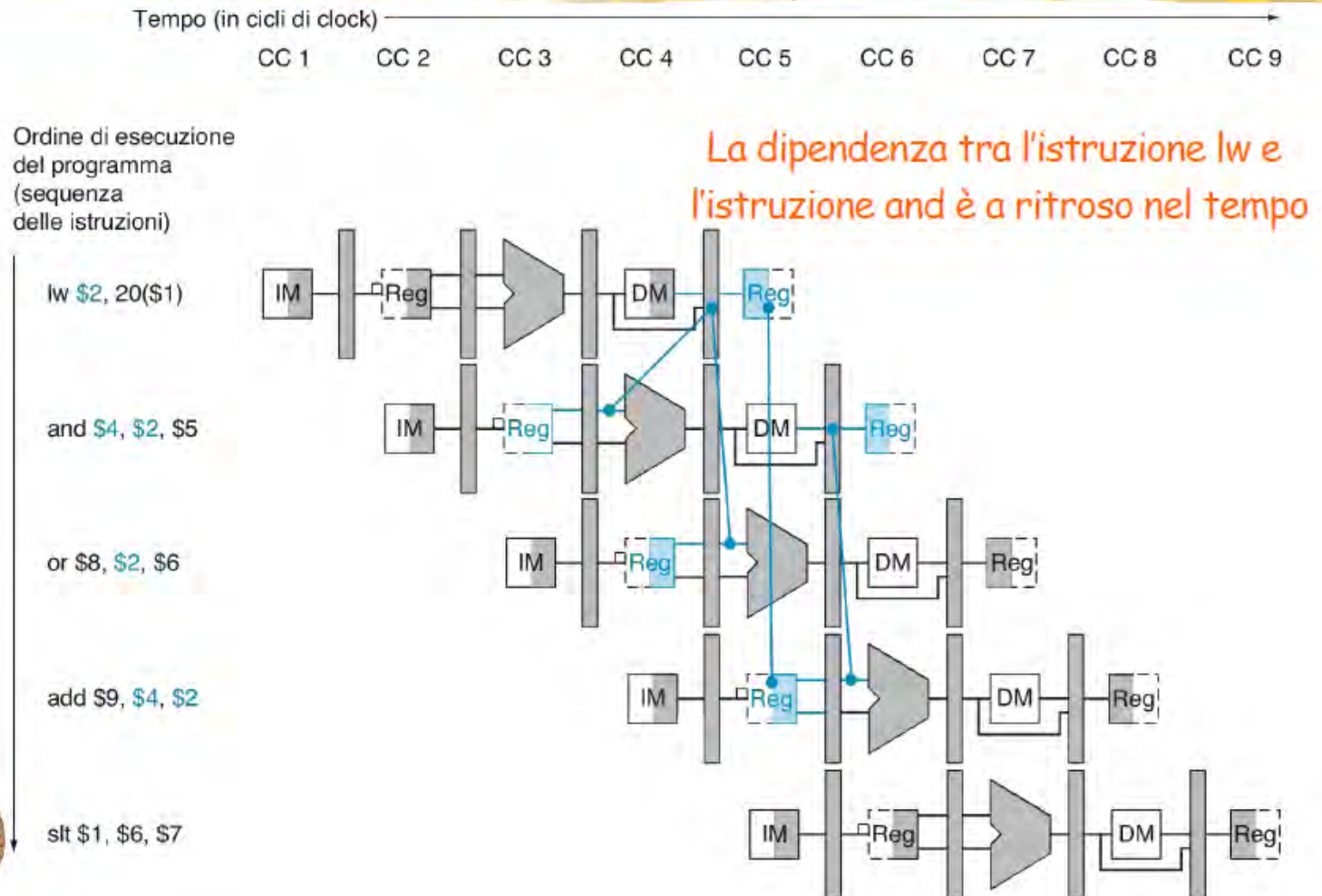
```
lw $s0, 12($s1)
add $s3, $s1, $s0
sub $s1, $s3, $s1
```

- A. Esiste una sola criticità (hazard) sui dati ed è possibile risolverla con la sola propagazione
- B. Esistono più criticità (hazard) sui dati ed è possibile risolverle con la sola propagazione
- C. Esistono più criticità (hazard) sui dati e non è possibile risolverle con la sola propagazione
- D. Esiste una sola criticità (hazard) sui dati e non è possibile risolverla con la sola propagazione



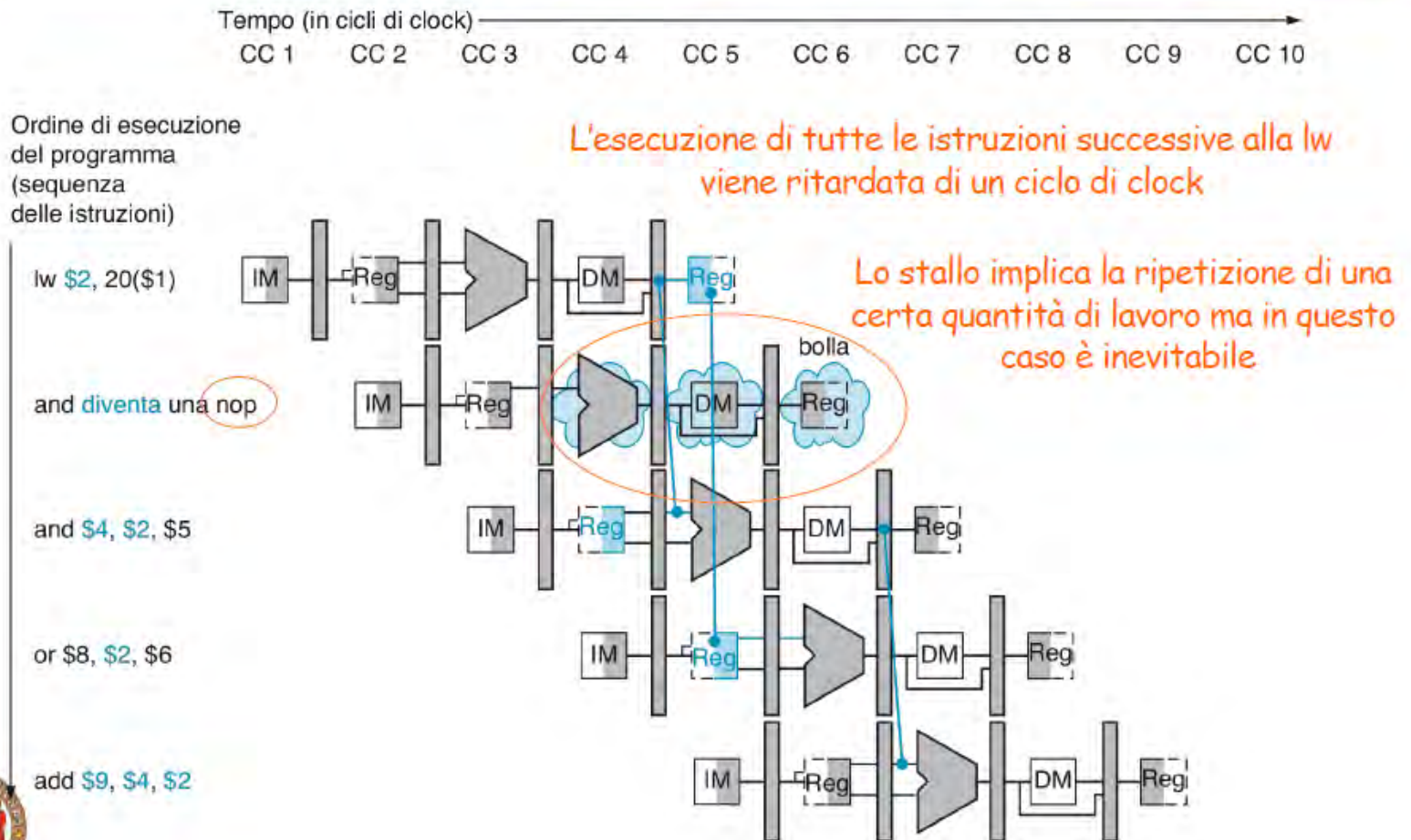
Lezione 22 pag. 26

Propagazione e stallo



Lezione 22 pag. 29

Propagazione e stallo



domanda 13

Si consideri l'implementazione del processore con pipeline.

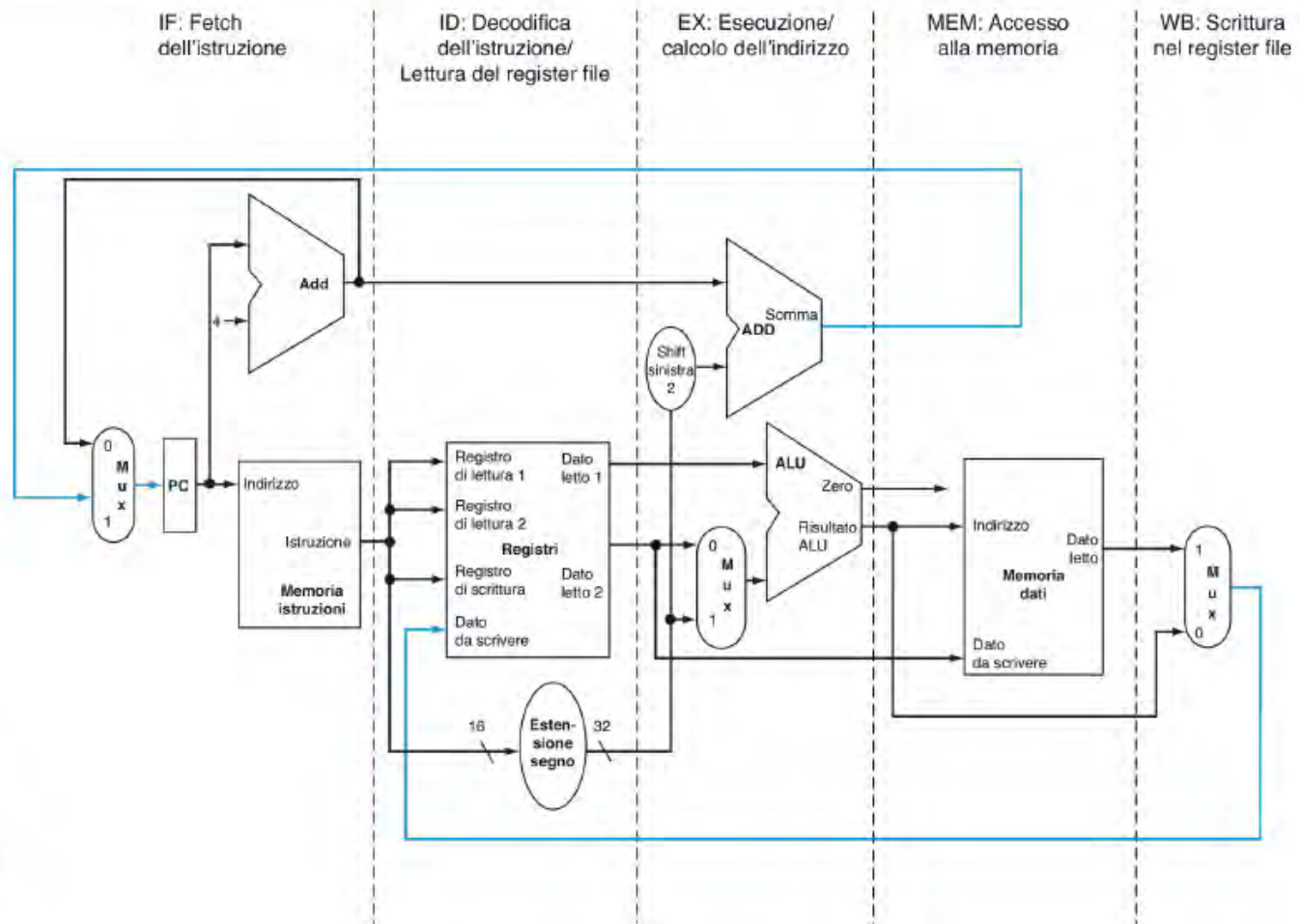
Cosa fa l'istruzione **add \$s0,\$s1,\$s2** nello stadio WB?

- A. Esegue la somma del contenuto dei registri \$s1 e \$s2
- B. Scrive nel registro \$s0 il dato prelevato dalla memoria
- C. Preleva l'istruzione dalla memoria istruzioni
- D. Nessuna delle altre risposte



Lezione 21 pag. 21

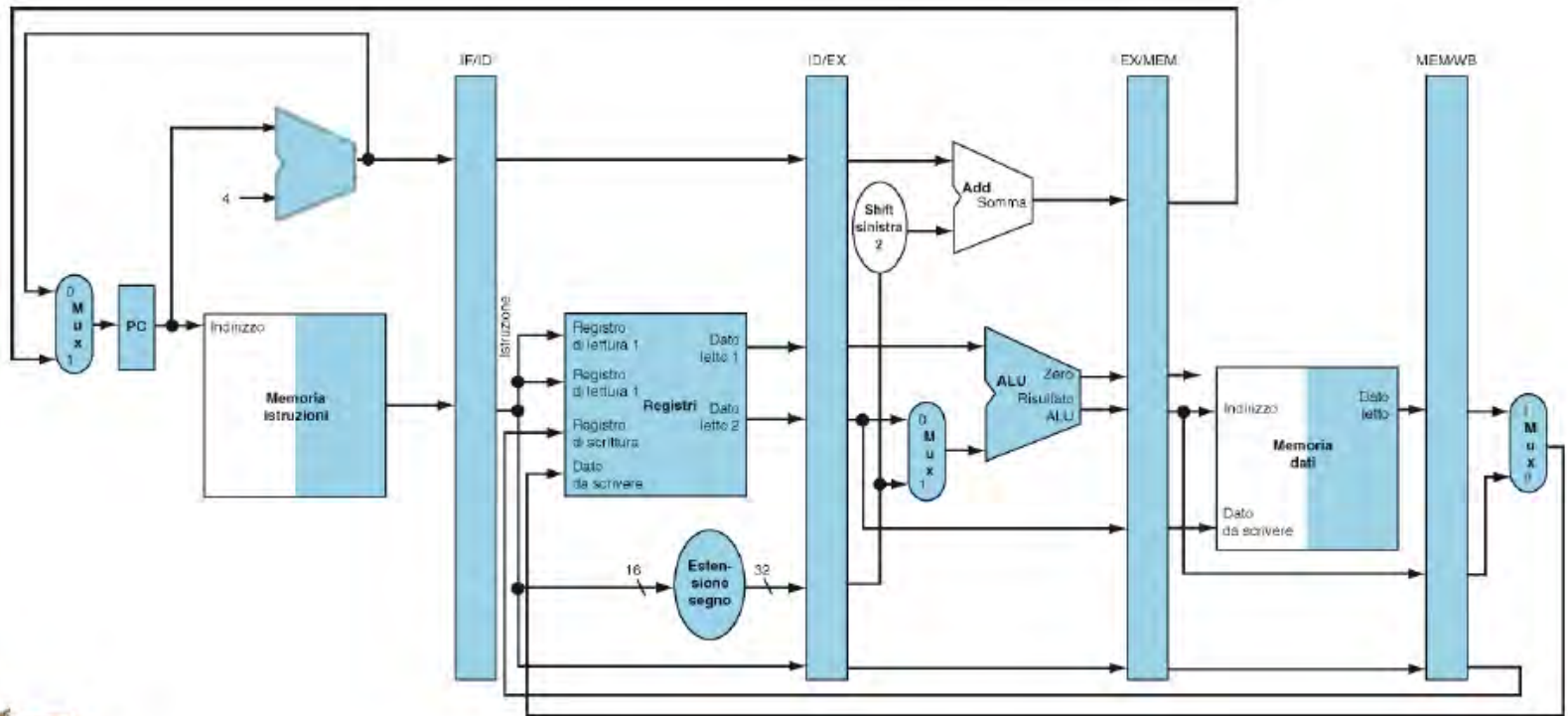
L'unità di Elaborazione a Ciclo Singolo



Lezione 21 pag. 34

Schema corretto per lw

- Mettendo insieme i 5 stadi della pipeline per l'istruzione **lw** si ha



domanda 14

In una RAM (*Random Access Memory*) l'accesso ad ogni locazione avviene in un tempo

- A. Costante per ogni locazione
- B. In un tempo proporzionale alla distanza
- C. In un tempo proporzionale all'indirizzo
- D. Nessuna delle risposte precedenti



Lezione 23 pag. 6

Tipi di accesso

- Nelle **memorie ad accesso casuale**, il tempo di accesso ai dati è indipendente dalla posizione in cui essi sono memorizzati
 - Tipico delle memorie a semiconduttori
 - Esempio: la **memoria principale** (RAM)
- Nelle **memorie ad accesso sequenziale**, il tempo di accesso ai dati dipende dalla posizione in cui essi sono memorizzati
 - Tipico delle memorie magnetiche
 - Esempio: la **memoria secondaria** (dischi e nastri)



domanda 15

Si supponga che i vari stadi della pipeline (come studiata) abbiano le latenze (ritardi) riportate in tabella. Quanto dura un ciclo di clock nel processore con pipeline e senza pipeline, rispettivamente?

IF	ID	EX	MEM	WB
100 ps	100 ps	300 ps	200 ps	200 ps

- A. 100 ps e 900 ps C. 900 ps e 1500 ps
B. 300 ps e 900 ps D. Nessuna delle risposte precedenti.



Lezione 21 pag. 12

Confronto di prestazioni

- Supponiamo che i **tempi richiesti** per le varie fasi siano rappresentati in tabella
 - Non si considerano i ritardi introdotti dai multiplexer, dall'unità di controllo, dall'accesso al registro PC e dall'estensione del segno

Tipo di istruzione	Letture dell'istruzione	Letture dei registri	Operazione con la ALU	Accesso ai dati in memoria	Scrittura del register file	Tempo totale
Load word (lw)	200 ps	100 ps	200 ps	200 ps	100 ps	800 ps
Store word (sw)	200 ps	100 ps	200 ps	200 ps		700 ps
Formato R (add, sub, and, or, slt)	200 ps	100 ps	200 ps		100 ps	600 ps
Salto condizionato (beq)	200 ps	100 ps	200 ps			500 ps

1 ps (picosecondo) = 10^{-12} s

- Nell'implementazione a ciclo singolo bisogna adeguarsi all'istruzione più lenta (**lw**) per stabilire la durata del ciclo di clock: quindi **800 ps**

Confronto di prestazioni

- Consideriamo una sequenza di **tre** istruzioni **lw**
- L'**implementazione a ciclo singolo** richiede $800 \times 3 = 2400$ ps
- Nell'**implementazione con pipeline**
 - Ciascuno dei 5 stadi impiega un ciclo di clock
 - La durata del ciclo di clock va tarata in base alla durata dello stadio più lento (200 ps), anche se alcuni stadi impiegano meno tempo
 - Ad esempio, la lettura dai registri richiede 100 ps
 - Quindi i 5 stadi della pipeline in sequenza richiedono 5 cicli di clock, per un totale di $200 \times 5 = 1000$ ps
 - Il tempo richiesto per eseguire le tre istruzioni è **1400 ps**
 - Ogni istruzione successiva alla prima fa aumentare il tempo di esecuzione di 200 ps, per cui il tempo richiesto dalle tre istruzioni è
$$200 \times 5 + 200 + 200 = 1400 \text{ ps}$$



domanda 15bis

Si supponga che i vari stadi della pipeline (come studiata) abbiano le latenze (ritardi) riportate in tabella. Quanto dura una singola istruzione nel processore con pipeline e senza pipeline, rispettivamente?

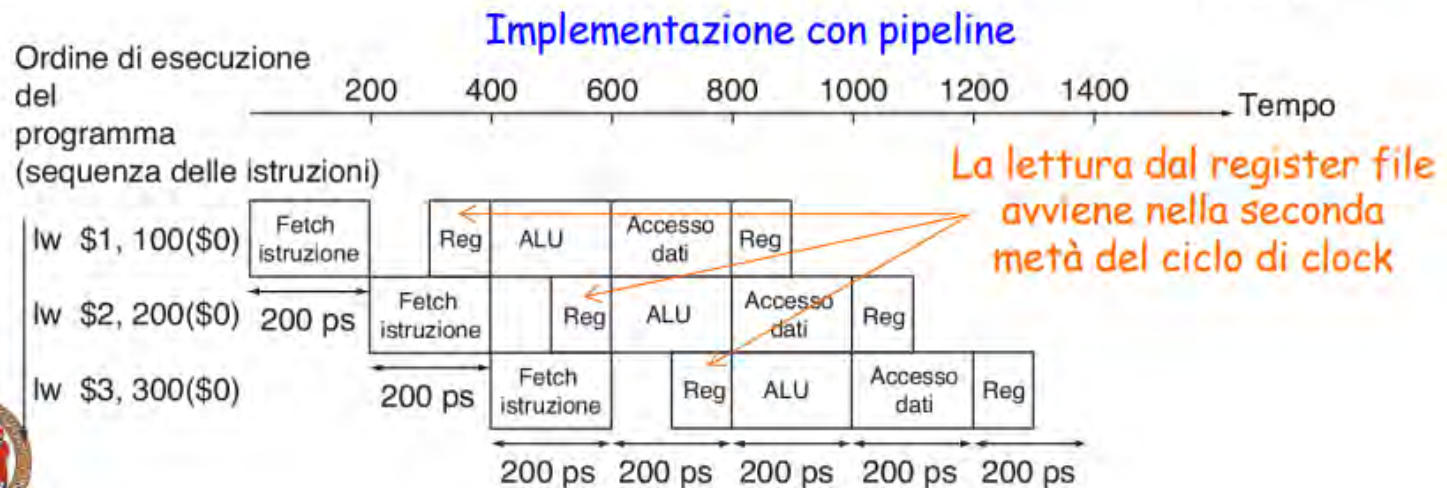
IF	ID	EX	MEM	WB
100 ps	100 ps	300 ps	200 ps	200 ps

- A. 1500 ps e 900 ps C. 900 ps e 1500 ps
B. 300 ps e 900 ps D. Nessuna delle risposte precedenti.



Lezione 21 pag. 14

Confronto di prestazioni



domanda 16

Si consideri la seguente sequenza di istruzioni:

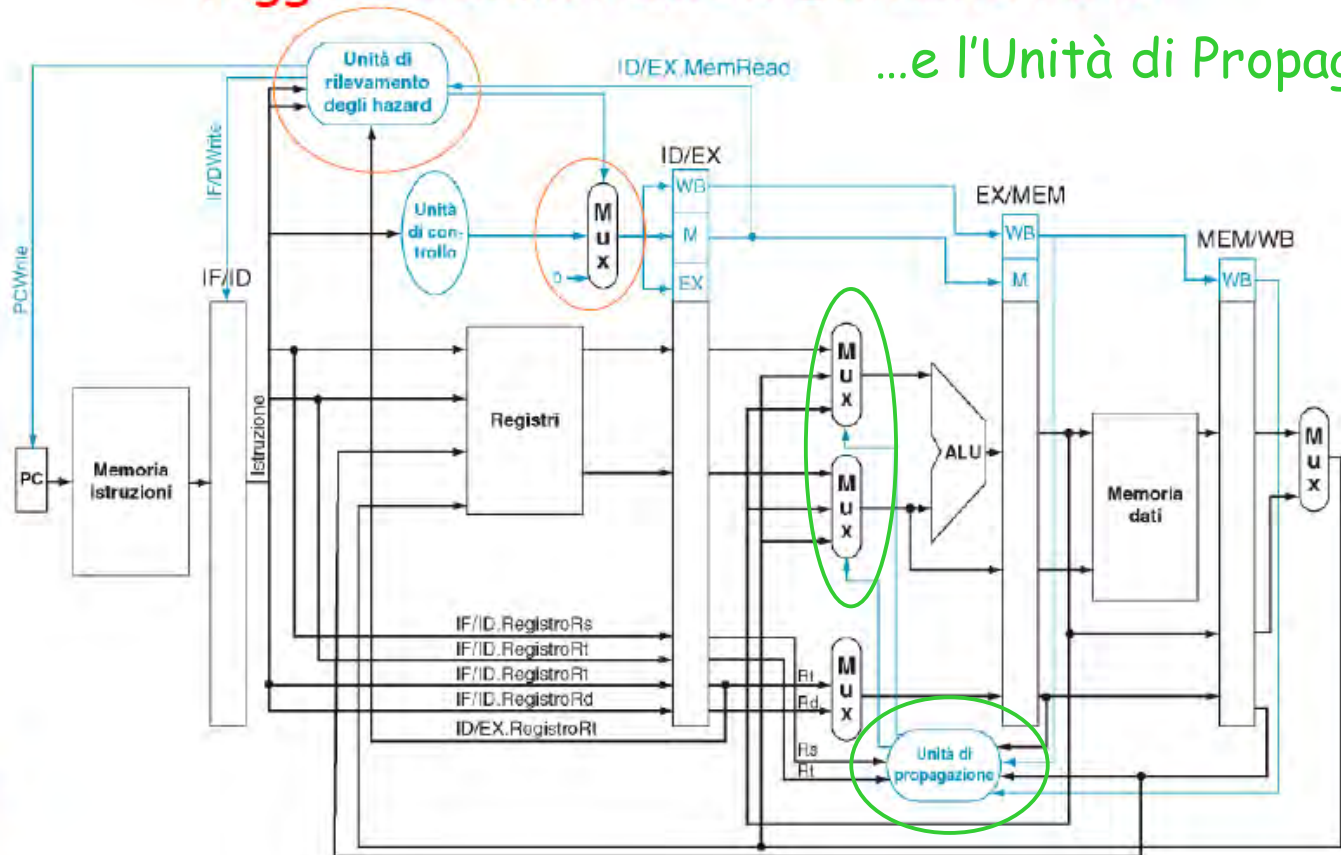
```
lw $s0, 12($s1)
sw $s1, 20($s2)
add $s3, $s1, $s1
sub $s1, $s3, $s1
```

- A. Esiste una sola criticità (hazard) sui dati ed è possibile risolverla con la sola propagazione
- B. Esistono più criticità (hazard) sui dati ed è possibile risolverle con la sola propagazione
- C. Esistono più criticità (hazard) sui dati e non è possibile risolverle con la sola propagazione
- D. Esiste una sola criticità (hazard) sui dati e non è possibile risolverla con la sola propagazione



Propagazione e stallo

...e l'Unità di Propagazione



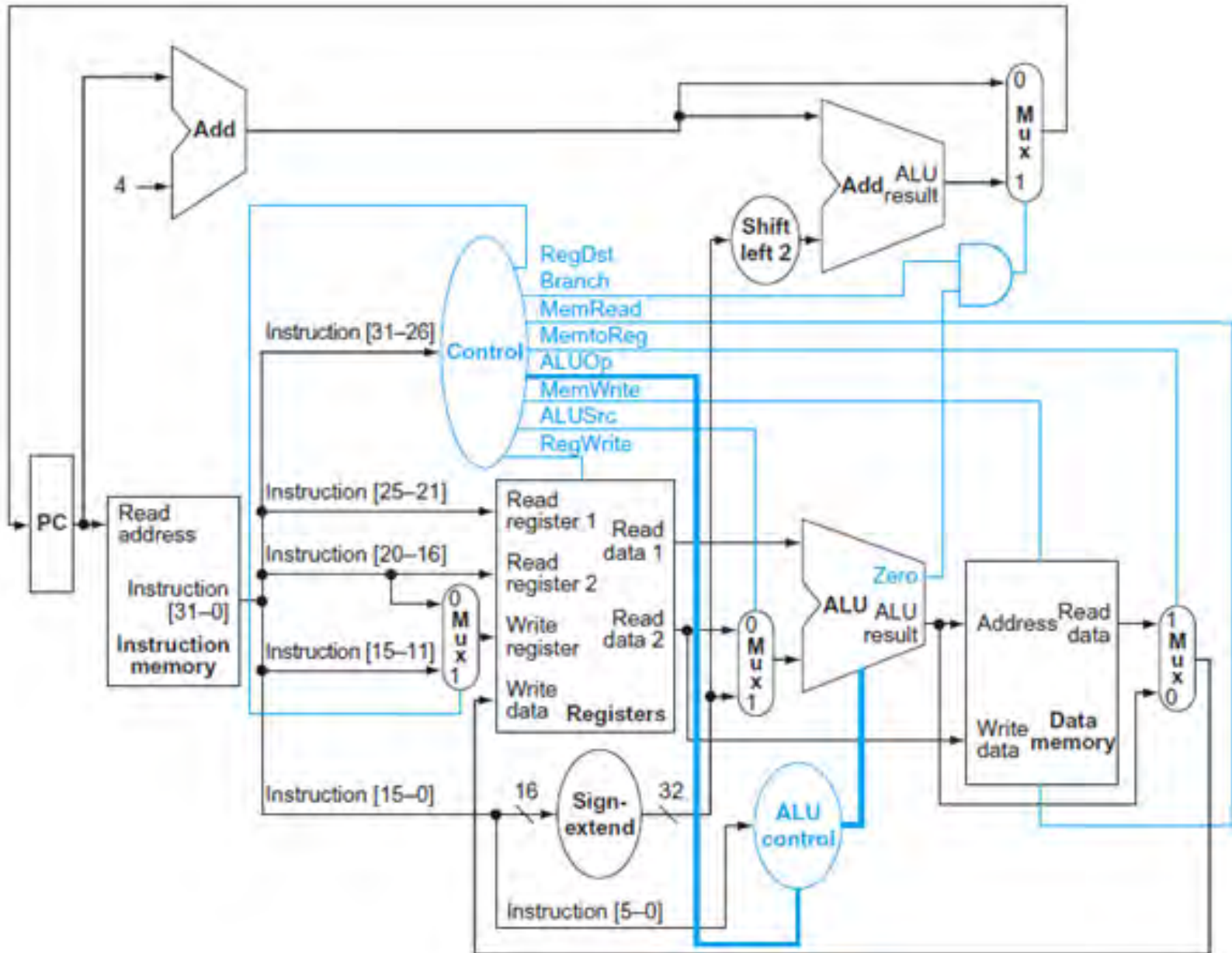
domanda 17

2)

Si consideri l'esecuzione dell'istruzione **sw \$1,152(\$2)** nel processore a singolo ciclo studiato. Le uscite del modulo di estensione del segno e dell'unità di shift a sinistra di 2 sono rispettivamente:

- A. 0000000010011000 e 0000001001100000
- B. 00000000000000000000000010011000 e 0000000000000000000000001001100000
- C. 00000000000000000000000000000010 e 0000000000000000000000000000001000
- D. Nessuna delle precedenti





domanda 18

Volendo saltare 15 istruzioni se il contenuto del registro \$2 è maggiore o uguale di 100, la sequenza di istruzioni da utilizzare è:

- A. `slti $1, $2, 100; beq $1, $0, 15`
- B. `slti $1, $2, 100; bne $1, $0, 15`
- C. `slti $1, $2, 100; bne $2, $0, 15`
- D. Nessuna delle precedenti



Lez. 13 pag. 19

Altri confronti

- Il MIPS offre anche il confronto
"un operando è **minore** di un altro o di una costante?"
 - Il confronto avviene mediante istruzioni che settano a 1 il valore di **registri di flag**
 - **Set if Less Than**
 - `slt $t0, $s3, $s4` #setta \$t0=1 se il contenuto di \$s3
#è minore del contenuto di \$s4
 - `slti $t0, $s3, 10` #setta \$t0=1 se il contenuto di \$s3
#è minore di 10
 - I salti su condizioni di tipo minore uguale o maggiore uguale si ottengono
 - Combinando **slt** con **beq** o **bne** e
 - Usando la costante 0 (disponibile nel registro **\$zero**) per fare i test



domanda 19

Supponendo che il contenuto dei registri \$s0, \$s1 siano rispettivamente 2 e 3 e il contenuto della memoria all'indirizzo 4 sia 10, quali saranno rispettivamente i contenuti di \$s0, \$s1 e della memoria all'indirizzo 4, dopo l'esecuzione della seguente sequenza di istruzioni?

```
bne $s0,$s1,SALTA
addi $s1,$s0,-1
j ESCI
SALTA:
addi $s0,$s1,1
ESCI:
sw $s1,0($s0)
```

- A. 2, 3, 10
- B. 3, 3, 2
- C. 4, 3, 3
- D. Nessuna delle altre risposte



domanda 20

Si assuma che le variabili *i* e *k* corrispondano ai registri *\$s0* e *\$s1*. Qual è l'istruzione C corrispondente al seguente codice *assembler* MIPS?

```
Label: beq $s0, $s1, Exit
      sll $s0, $s0, 2
      j Label
```

Exit: ...

A. while(*i*==*k*) *i*=*i**2;

B. if(*i*==*k*) *i*=*i**2;

C. while(*i*!=*k*) *i*=*i**2;

D. Nessuna delle risposte precedenti.



Lezione 12 pag. 20

Shift logico a sx

- Un effetto collaterale di questa istruzione è che il risultato corrisponde a **moltiplicare l'operando per 2^n**
- Esempio: se il contenuto del registro \$s0 è

0000 0000 0000 0000 0000 0000 0000 1001 cioè 9_{10}

l'istruzione **sll \$t0, \$s0, 4** produce in \$t0



← 0000 0000 0000 0000 0000 0000 0000 1001 ← 0000

che corrisponde a $144_{10} = 9_{10} \times 2^4 = 9_{10} \times 16_{10}$



Altri spunti 1

5)

5 ☐

Durante l'esecuzione di una istruzione `lw` nel processore a singolo ciclo studiato, l'uscita della ALU fornisce

- A. il dato da scrivere nel banco dei registri
- B. il dato da scrivere nella memoria dati
- C. il nuovo valore del PC
- D. Nessuna delle risposte precedenti.

6)

6 ☐

La pipeline migliora

- A. Il numero di istruzioni che vengono eseguite in un determinato tempo (soltanto)
- B. Il tempo di esecuzione di una singola istruzione (soltanto)
- C. Entrambi i valori ai punti A e B
- D. Nessuna delle risposte precedenti

7)

7 ☐

Si consideri l'implementazione studiata del processore con pipeline. Nella fase WB (*Write Back*), il numero del registro di scrittura viene letto

- | | |
|--------------------------------------|--------------------------------------|
| A. Dal campo [15-11] dell'istruzione | C. Dall'uscita della ALU |
| B. Dal registro MEM/WB | D. Nessuna delle risposte precedenti |

Altri spunti 2

1 ☐

1)

Dai 32 bit dell'istruzione prelevata dalla memoria, viene inviato al modulo di estensione del segno il seguente campo:

- A. [20-16] B. [16 -0] C. [15-0] D. Nessuna delle risposte precedenti

2 ☐

2)

Nello schema di implementazione studiata per il processore MIPS, il *mutiplexer* posto al secondo ingresso dell'ALU permette di scegliere fra:

- A. Il secondo dato letto dal banco dei registri e l'estensione del segno dei 16 bit meno significativi dell'istruzione
B. Il secondo dato scritto nel banco dei registri e l'estensione del segno dei 16 bit più significativi dell'istruzione
C. Il dato letto dalla memoria dati e l'uscita della ALU.
D. Nessuna delle risposte precedenti

4 ☐

4)

Si consideri l'implementazione del processore con pipeline studiata.

Cosa fa l'istruzione **sw \$s1, 12(\$s2)** nello stadio WB (Write Back)?

- A. Esegue la somma del contenuto del registro \$s2 con 12
B. Scrive il dato prelevato dal registro \$1 nella memoria dati
C. Scrive il dato prelevato nel registro \$s1
D. Nessuna delle risposte precedenti



Altri spunti 3

2)

2 ☐

La scrittura nel banco di registri (*register file*) avviene tramite

- A. un multiplexer che ha il numero del registro come segnale di controllo
- B. un decoder che ha in input il dato da scrivere
- C. un decoder che ha in input il numero del registro
- D. Nessuna delle risposte precedenti

3)

3 ☐

In *assembler* MIPS la chiamata alla procedura `proc` è realizzata:

- A. Tramite l'istruzione `j proc` e inserendo `jr $ra` dopo l'ultima istruzione della procedura
- B. Tramite l'istruzione `j proc` e inserendo `jal $ra` dopo l'ultima istruzione della procedura
- C. Tramite l'istruzione `jal proc` e inserendo `jr $ra` dopo l'ultima istruzione della procedura
- D. Nessuna delle risposte precedenti

4)

4 ☐

Nello schema di implementazione studiata per il processore MIPS con pipeline, in quale fase opera il modulo di estensione del segno?

- A. IF
- B. ID
- C. EX
- D. Nessuna delle risposte precedenti

|

Altri spunti 4

5)

5 ☐

Durante l'esecuzione di una istruzione `sw` nel processore a singolo ciclo studiato, l'uscita della ALU fornisce

- A. il dato da scrivere nel banco dei registri
- B. il dato da scrivere nella memoria dati
- C. il nuovo valore del PC
- D. Nessuna delle risposte precedenti.

6)

6 ☐

Nello schema di implementazione studiata per il processore MIPS con pipeline, in quale fase opera l'Unità di Controllo principale?

- A. IF
- B. ID
- C. EX
- D. Nessuna delle risposte precedenti

7)

7 ☐

Il tempo di esecuzione di una singola istruzione nel processore con pipeline (come studiato) rispetto al processore a singolo ciclo senza pipeline (come studiato):

- A. È sicuramente inferiore
- B. È uguale
- C. Può essere anche superiore
- D. Nessuna delle precedenti

Riepilogo

- Simulazione seconda prova intercorso
- Prossima lezione prima del preappello di gennaio:
 - 10 gennaio 2023 ore 16:00 - 19:00, stesso Team



In bocca al lupo!

