



Hazard e forwarding

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borgnese@dsi.unimi.it

Università degli Studi di Milano

Riferimento al Patterson: 4.7



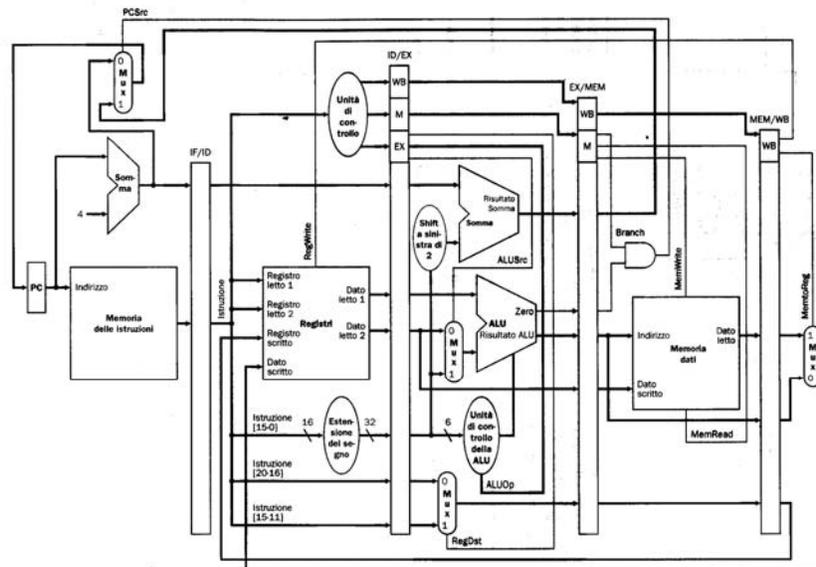
Sommario

Soluzione delle criticità sui dati senza stallo

Soluzione delle criticità sui dati con stallo



CPU con pipeline



Soluzione architetturale della criticità sui dati



La criticità nei dati ha a che fare essenzialmente con la disponibilità di dati corretti.

Identificazione della criticità (funzione del tipo di istruzione e dei registri coinvolti).

Correzione della situazione: propagazione a ritroso (negli stadi della pipeline = in avanti nel tempo) su data-path alternativi dei dati richiesti.



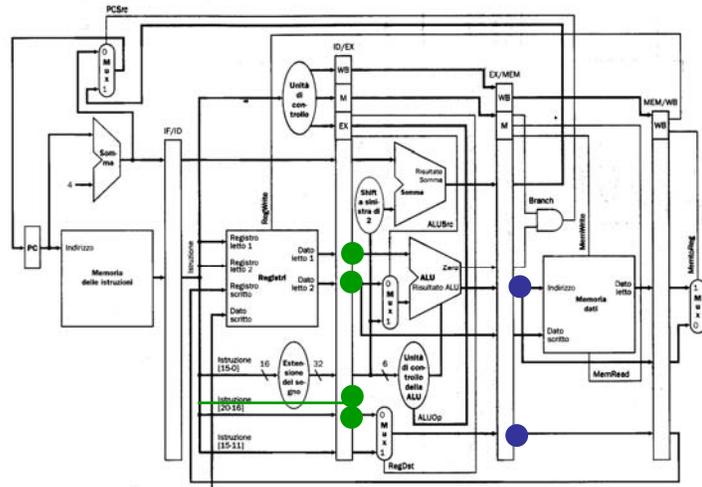
Identificazione delle criticità – EX/MEM



sub \$s2, \$s1, \$s3
add \$t2, \$s2, \$s5

1a. EX/MEM.RegistroRd = ID/EX.RegistroRs

1b. EX/MEM.RegistroRd = ID/EX.RegistroRt



A.A. 2009-2010

ghese



Hazard sui dati: rilevamento della criticità



sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2			
add \$t2, \$s2, \$s5		IF	ID \$s2 and \$s5	EX	MEM s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)	

Rilevo la criticità (dato non corretto) su **and** quando **and** inizia la fase di **EX**. In questo caso il dato corretto si trova all'inizio della fase **MEM** della **sub**.

Questo modo di rilevare la criticità consente di ottenere i datapath più brevi all'interno della CPU.

A.A. 2009-2010

6/39

<http://homes.dsi.unimi.it/~borghese>



Hazard sui dati: formalizzazione della criticità



sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2			
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2		
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)	

IF ((EX/MEM.RegistroRd == ID/EX.RegistroRs) ||
 ((EX/MEM.RegistroRd == ID/EX.RegistroRt)) &&
 (EX/MEM.RegisterWrite)

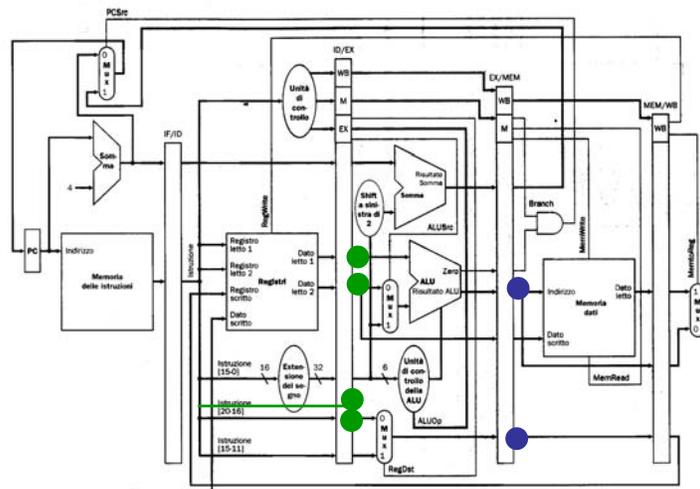


Abbiamo identificato il problema, dobbiamo ora risolverlo



sub \$s2, \$s1, \$s3
 add \$t2, \$s2, \$s5

IF ((EX/MEM.RegistroRd == ID/EX.RegistroRs) ||
 ((EX/MEM.RegistroRd == ID/EX.RegistroRt)) &&
 (EX/MEM.RegisterWrite)





Hazard sui dati: feed-forwarding

sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2				
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		

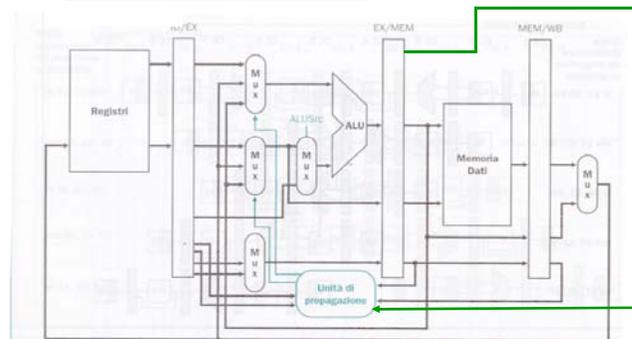
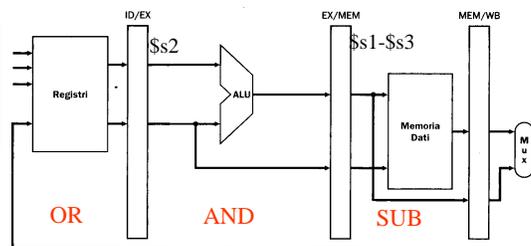
IF (EX/MEM.RegistroRd == ID/EX.RegistroRs) && (EX/MEM.RegisterWrite)
ALUSrcA = <EX/MEM.Data>

IF (EX/MEM.RegistroRd == ID/EX.RegistroRt) && (EX/MEM.RegisterWrite)
ALUSrcB = <EX/MEM.Data>



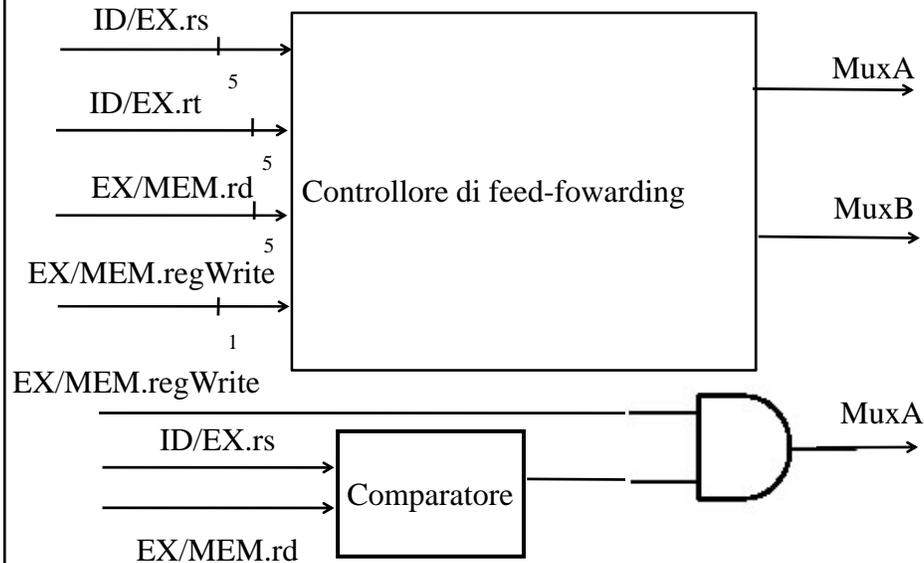
Hazard nei dati: forwarding

sub \$s2, \$s1, \$s3
add \$t2, \$s2, \$s5
or \$t3, \$s6, \$s2





Controllore del feed-forwarding



Posizionamento dei Mux



Il mux che seleziona tra il dato che proviene dal registro ID/EX ed i dati retropropagati dai registri EX/MEM oppure MEM/WB deve essere inserito prima o dopo il mux che seleziona tra contenuto del registro target e l'estensione a 32 bit del campo costante e perché?

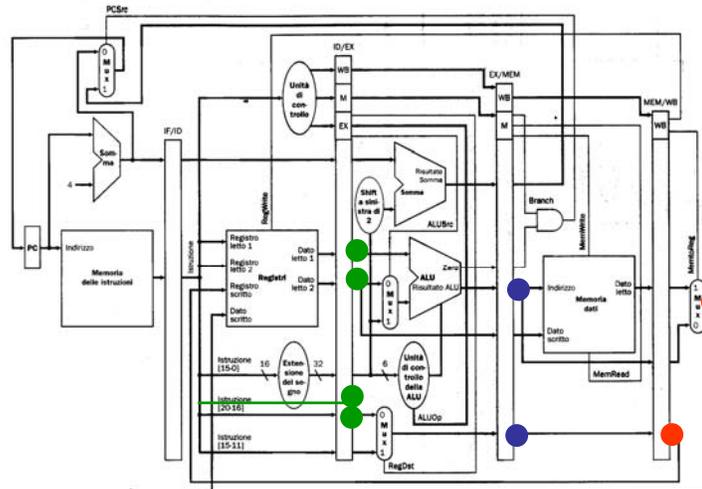


Identificazione delle criticità – MEM/WB



sw \$s1, 100(\$t1)
 sub \$s2, \$s1, \$s3
 add \$t2, \$s2, \$s5
 or \$t3, \$s6, \$s2

2a. MEM/WB.RegistroRd = ID/EX.RegistroRs
 2b. MEM/WB.RegistroRd = ID/EX.RegistroRt



A.A. 2009-2010

ghese



Hazard sui dati: rilevamento della criticità



sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2			
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2		
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)	

Rilevo la criticità (dato non corretto) su **and** quando **and** inizia la fase di **EX**. In questo caso il dato corretto si trova all'inizio della fase **MEM** della **sub**.

→ Rilevo la criticità (dato non corretto) su **or** quando **or** inizia la fase di **EX**. In questo caso il dato corretto si trova all'inizio della fase **WB** della **sub**.

Questo modo di rilevare la criticità consente di ottenere i datapath più brevi all'interno della CPU.

A.A. 2009-2010

14/39

<http://homes.dsi.unimi.it/~borghese>



Hazard sui dati: formalizzazione della criticità



sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2				
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		

IF ((MEM/WB.RegistroRd == ID/EX.RegistroRs) ||
 (MEM/WB.RegistroRd == ID/EX.RegistroRt)) &&
 (MEM/WB.RegisterWrite)



Hazard sui dati: feed-forwarding



sub \$s2, \$s1, \$s3	IF	ID	EX \$s1- \$s3	MEM	WB s->\$s2				
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		

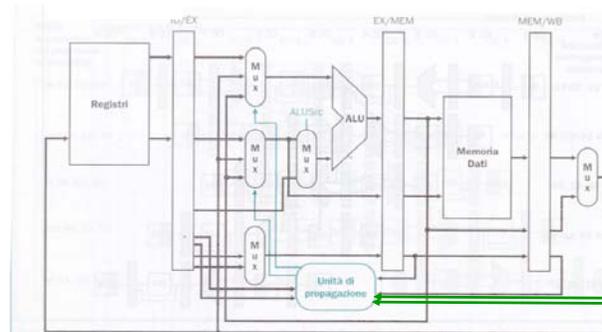
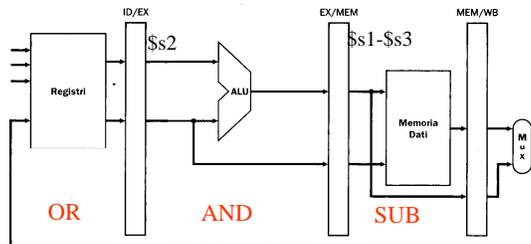
IF (MEM/WB.RegistroRd == ID/EX.RegistroRs) && (MEM/WB.RegisterWrite)
 ALUSrcA = <MEM/WB.Data>

IF (MEM/WB.RegistroRd == ID/EX.RegistroRt) && (MEM/WB.RegisterWrite)
 ALUSrcB = <MEM/WB.Data>



Hazard nei dati: forwarding

```
sub $s2, $s1, $s3
add $t2, $s2, $s5
or $t3, $s6, $s2
```



EX/MEM.RegWrite
MEM/WB.RegWrite



Relazione tra forwarding e contenuto del registro ID/EX

Nel normale funzionamento, il registro ID/EX contiene quanto letto dal Register File.

Quando abbiamo forwarding, quello che viene letto dal registro ID/EX nella fase di esecuzione viene sovrascritto da quanto letto dal registro EX/MEM o MEM/WB.

Nel registro EX/MEM è contenuto il risultato dell'operazione eseguita all'istante precedente.

Nel registro MEM/WB è contenuto il risultato dell'operazione eseguita 2 istanti precedenti.



Controllo Mux ingresso alla ALU



Controllo Multiplexer	Registro Sorgente	Funzione
PropagaA = 00	ID/EX	Il primo operando della ALU proviene dal Register File
PropagaA = 01	EX/MEM	Il primo operando della ALU è propagato dal risultato della ALU per l'istruzione precedente.
PropagaA = 10	MEM/WB	Il primo operando della ALU è propagato dalla memoria o da un'altra istruzione precedente.
PropagaB = 00	ID/EX	Il secondo operando della ALU proviene dal Register File
PropagaB = 01	EX/MEM	Il secondo operando della ALU è propagato dal risultato della ALU per l'istruzione precedente.
PropagaB = 10	MEM/WB	Il secondo operando della ALU è propagato dalla memoria o da un'altra istruzione precedente.



Unità di controllo del forwarding



Deve controllare che la criticità sia effettiva (che l'istruzione precedente scriva il RegisterFile).

E' attiva nella fase di esecuzione (EX) ed implementa le seguenti funzioni:

Dato preso dalla fase MEM:

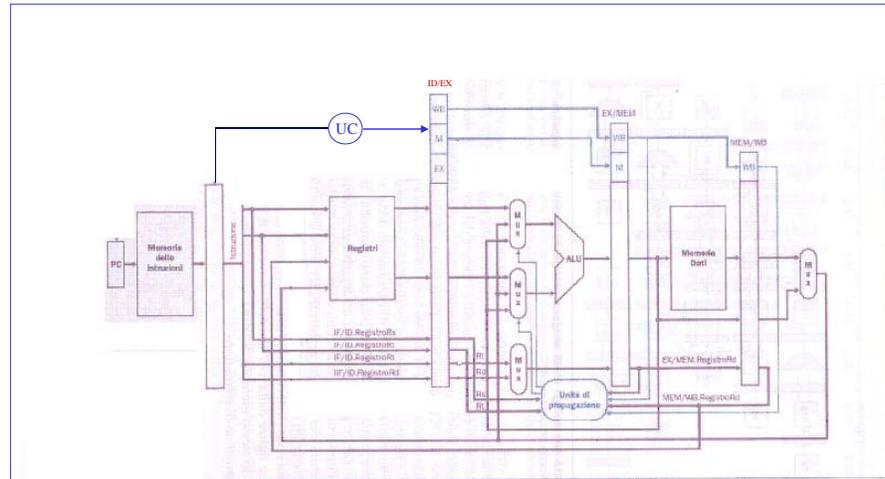
```
IF (ID/EX.RegistroRs == EX/MEM.RegistroRd) AND (EX/MEM.RegWrite)
    ALUSrcA = <EX/MEM.Data>
IF (ID/EX.RegistroRt == EX/MEM.RegistroRd) AND (EX/MEM.RegWrite)
    ALUSrcB = <EX/MEM.Data>
```

Dato preso dalla fase WB:

```
IF (ID/EX.RegistroRs == MEM/WB.RegistroRd) AND (MEM/WB.RegWrite)
    ALUSrcA = <MEM/WB.Data>
IF (ID/EX.RegistroRt == MEM/WB.RegistroRd) AND (MEM/WB.RegWrite)
    ALUSrcB = <MEM/WB.Data>
```



CPU con unità di propagazione



Sommario



Soluzione delle criticità sui dati senza stallo

Soluzione delle criticità sui dati con stallo

Hazard sui dati: lw

lw \$s2, 40(\$s3)	IF	ID	EX \$s3+40	MEM <\$s3+40>	WB s->\$s2				
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		
add \$t4, \$s2, \$s2				IF	ID	EX \$s2 + \$s2	MEM	WB s->\$t4	
sw \$t5, 100(\$s2)					IF	ID	EX \$s2+100	MEM \$t5	WB ->Mem

A.A. 2009-2010 23/39 http://homes.dsi.unimi.it/~borghese

Hazard sui dati: lw, rilevamento della criticità

lw \$s2, 40(\$s3)	IF	ID	EX \$s3+40	MEM <\$s3+40>	WB s->\$s2				
add \$t2, \$s2, \$s5		IF	ID	EX \$s2 and \$s5	MEM	WB s->\$t2			
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		

Il dato corretto per \$s2 è pronto nella lw solamente alla fine della fase **MEM**, ed è perciò utilizzabile solamente a partire dall'inizio della fase di **WB**.

Rilevo la criticità (dato non corretto) su **and** quando **and** inizia la fase di **EX**. In questo caso il dato corretto non è ancora stato prodotto dalla lw.

Rilevo la criticità (dato non corretto) su **or** quando **or** inizia la fase di **EX**. In questo caso il dato corretto si trova all'inizio della fase **WB** della lw.

A.A. 2009-2010 24/39 http://homes.dsi.unimi.it/~borghese



Soluzione mediante stallo



	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇
.....								
lw \$s2, 40(\$s3)	FF (Mem, ALU)	DECOD (RF)	EXEC (ALU)	MEM (MEM)	WB (RF)			
nop		Buco (FF)	Buco (DEC)	Buco (EXEC)	Buco (MEM)	Buco (WB)		
nop			Buco	Buco	Buco	Buco	Buco	
add \$t2, \$s2, \$s5				FF	DEC	EXEC	MEM	

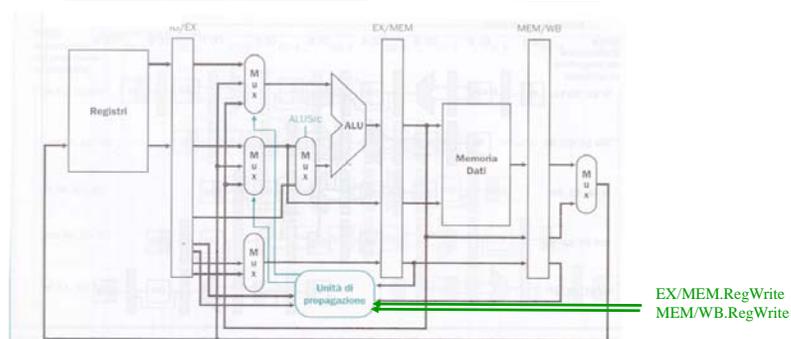
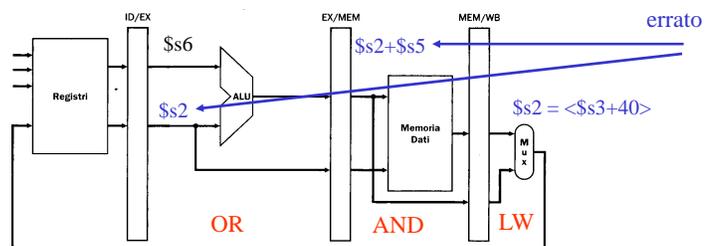
I buchi (o bubble) inducano degli istanti di clock in cui non può essere eseguita l'istruzione successiva → La pipeline va in stallo.



Hazard nei dati: lw, forwarding



lw \$s2, 40(\$s3)
add \$t2, \$s2, \$s5
or \$t3, \$s6, \$s2





Hazard nei dati: lw, unità di propagazione



Dato preso dalla fase WB:

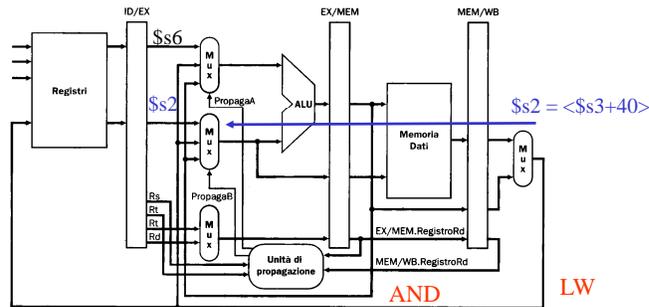
IF (ID/EX.RegistroRs == MEM/WB.RegistroRd) AND (MEM/WB.RegWrite)

ALUSrcA = <MEM/WB.Data>

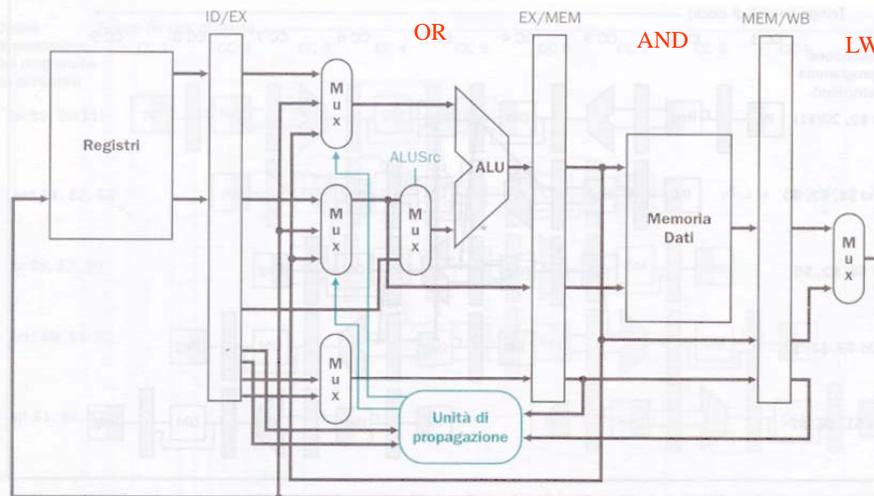
IF (ID/EX.RegistroRt == MEM/WB.RegistroRd) AND (MEM/WB.RegWrite)

ALUSrcB = <MEM/WB.Data>

lw \$s2, 40(\$s3)
add \$t2, \$s2, \$s5
or \$t3, \$s6, \$s2



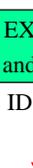
Il forwarding non è sufficiente



Nessuna modifica ma risolve solamente uno dei due problemi della lw.

Hazard sui dati: lw, stallo

lw \$s2, 40(\$s3)	IF	ID	EX \$s3+40	MEM <\$s3+40>	WB s->\$s2				
add \$t2, \$s2, \$s5	IF	ID	EX \$s2 and \$s5	MEM \$s2 and \$s5	WB s->\$t2				
and \$t2, \$s2, \$s5			IF	ID	EX \$s2 and \$s5	MEM	WB (s->\$t2)		



Il dato corretto per \$s2 è pronto nella lw solamente alla fine della fase **MEM**, ed è perciò utilizzabile solamente a partire dall'inizio della fase di **EX**.

Devo bloccare l'esecuzione della and e ripeterla un ciclo dopo, quando è possibile utilizzare il valore corretto del registro \$s2.


Stallo della pipeline

A.A. 2009-2010 29/39 http://homes.dsi.unimi.it/~borghese

Rilevamento della criticità sulla lw

lw \$s2, 40(\$s3)	IF	ID	EX	MEM <\$s3+40>	WB s->\$s2				
add \$t2, \$s2, \$s5	IF	ID	EX \$s2 and \$s5	MEM \$s2 and \$s5	WB s->\$t2				
or \$t3, \$s6, \$s2			IF	ID	EX \$s6 or \$s2	MEM	WB (s->\$t3)		



Il dato corretto per \$s2 è pronto nella lw solamente alla fine della fase **MEM**, ed è perciò utilizzabile solamente a partire dall'inizio della fase di **WB**.

Rilevo questa criticità il prima possibile in modo da mettere in stallo prima possibile la pipeline: nello stadio di decodifica dell'istruzione **AND**.

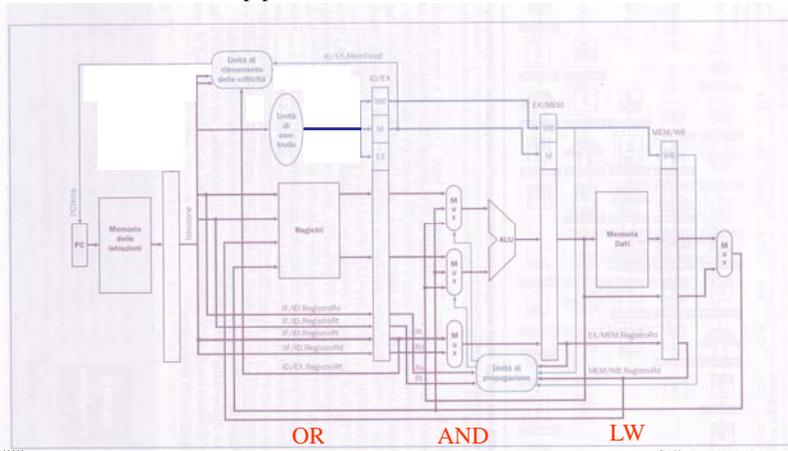
A.A. 2009-2010 30/39 http://homes.dsi.unimi.it/~borghese



Rilevamento della criticità della lw

IF [(ID/EX(MemRead)) → Read in fase di EX
 AND
 {[(IF/ID.RegistroRt) == ID/EX.RegistroRt] OR
 [(IF/ID.RegistroRs) == IF/EX.RegistroRt]} EX - lw \$s2, 40(\$s3)
 THEN DEC - add \$t2, \$s2, \$s5

“Metti in stallo la pipeline”



A.A. 2009-2010

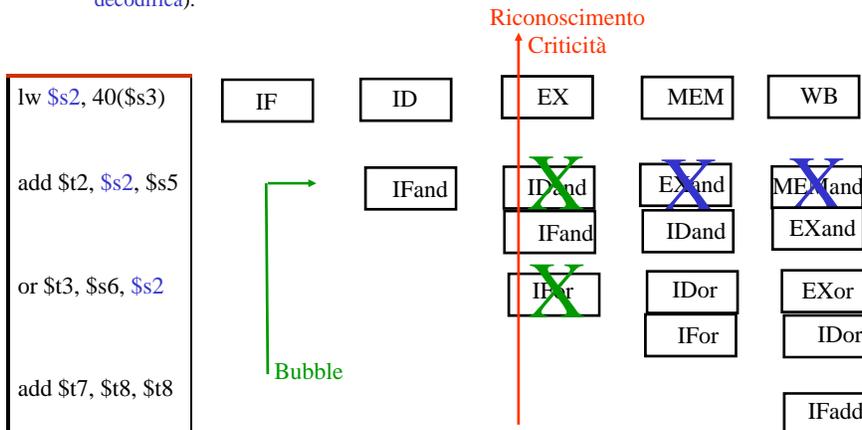
borghese



Stallo della pipeline

Azioni:

- Annullare i segnali di controllo generati nella fase ID per l'esecuzione dell'istruzione (successiva alla lw).
- Ripetere la lettura e la decodifica delle 2 istruzioni successive (ripetere la fase di fetch e decodifica).



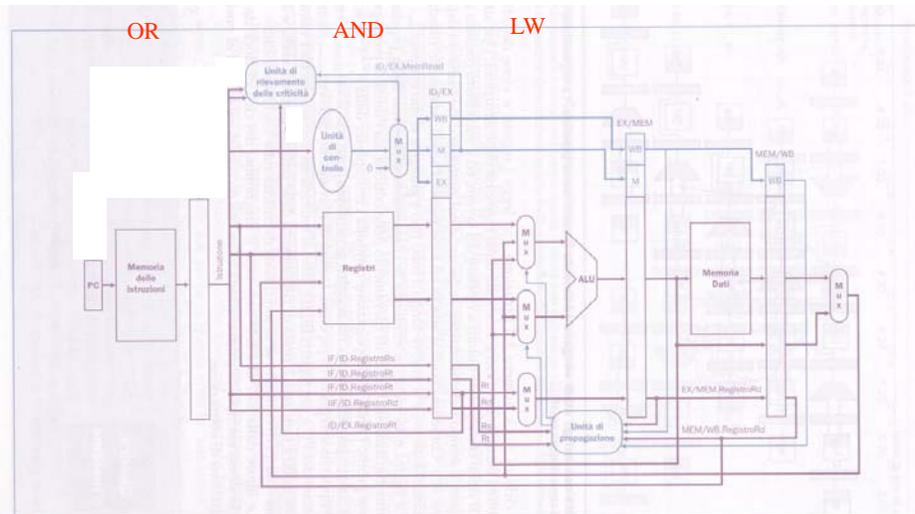
A.A. 2009-2010

32/39

<http://homes.dsi.unimi.it/~borghese>



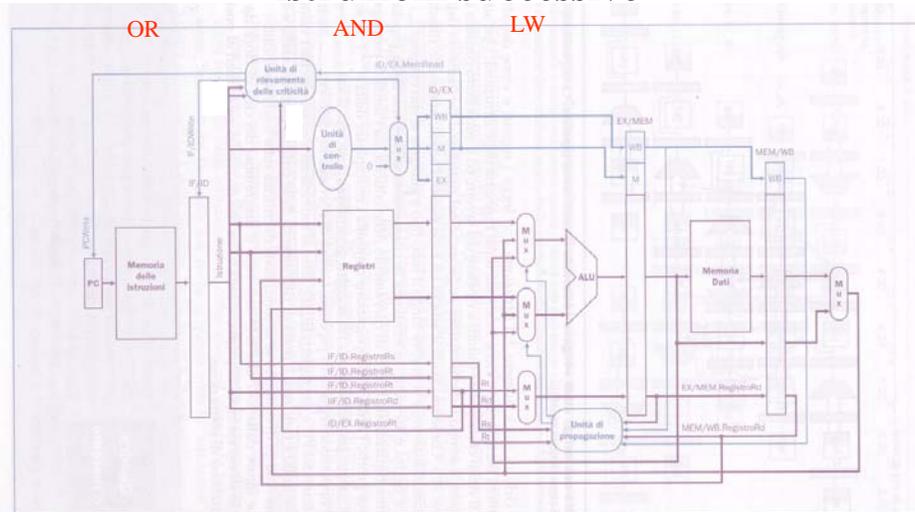
Annullamento dell'istruzione in fase ID (and)



Annullamento dei segnali di controllo associati. Perché invece non annullo la scrittura dei registri ID/EX, EX/MEM e MEM/WB?



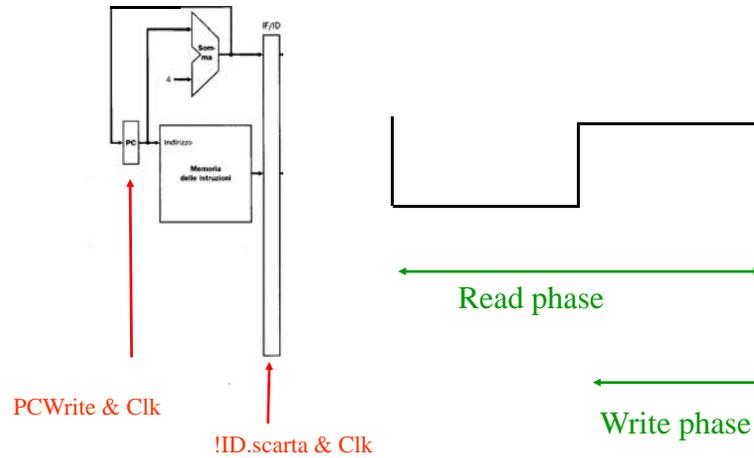
Ripetizione delle fasi ID e IF delle due istruzioni successive



Disabilitazione della scrittura del PC e del registro IF/ID nella fase di Exec della lw.



Disabilitazione della scrittura dei registri



Hp: L'unità di controllo della criticità è in grado di prendere una decisione in tempo utile (prima dell'inizio della fase di Write).



Rilevamento della criticità sulla lw



lw \$s2, 40(\$s3)	IF	ID	EX	MEM	WB				
and \$t2, \$s2, \$s5		IF	ID	EX	MEM	WB			
or \$t3, \$s6, \$s2			IF	ID	EX	MEM	WB		

Annotations: A blue arrow points from the ID stage of the first instruction to the ID stage of the second instruction. A red dotted arrow points from the EX stage of the second instruction to the EX stage of the third instruction. The WB stage of the first instruction is labeled 's->\$s2'. The WB stage of the second instruction is labeled 's->\$t2'. The WB stage of the third instruction is labeled '(s->\$t3)'.

Il dato corretto per \$s2 è pronto nella lw solamente alla fine della fase MEM, ed è perciò utilizzabile solamente a partire dall'inizio della fase di WB.

Rilevo questa criticità il prima possibile in modo da mettere in stallo prima possibile la pipeline: nello stadio di decodifica dell'istruzione AND.

Potrei rilevare la criticità anche nello stadio EX dell'istruzione AND. Quale svantaggio avrei?



Hazard sui dati della lw



1) Rilevamento della criticità

IF [ID/EX.MemRead] AND {[(IF/ID.RegistroRt) == ID/EX.RegistroRt] OR
[(IF/ID.RegistroRs) == IF/EX.RegistroRt]}

2) Correzione del problema -> stallo

2a) faccio eseguire l'istruzione in ID con segnali di controllo a 0: esecuzione fasulla.

2b) inibisco la scrittura dei registri ID e PC.



Hazard nei dati: soluzioni



- Buona scrittura del codice (il programmatore deve conoscere la macchina per scrivere un buon codice!).
- Compilatore efficiente (che riordini il codice).
- Architettura che renda disponibile i dati appena pronti alla fase di esecuzione.
- Accettare uno stallo (non sempre si può evitare).



Sommario



Soluzione delle criticità sui dati senza stallo

Soluzione delle criticità sui dati con stallo