

Gestione delle criticità sui dati nella pipeline

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

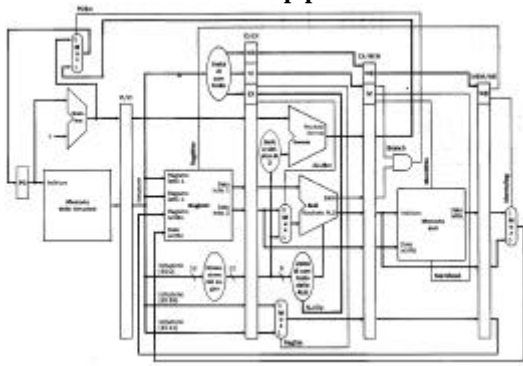
Università degli Studi di Milano

Sommario

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di tipo R.

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di lw.

CPU con pipeline



Criticità nei dati

Dovrei eseguire un'istruzione in cui uno dei dati è il risultato dell'esecuzione di un'istruzione precedente.

Soluzione mediante due tecniche:

Riorganizzazione del codice (compilatore).

Propagazione (forwarding) o scavalco (bypassing).

Hazard nei dati: soluzione tramite compilatore

```

add $s0, $t1, $t1
nop
nop
nop
sub $s1, $s0, $s0
and $t2, $t0, $t1
or $t5, $t3, $t4
add $s2, $s7, $t7
sub $s2, $s0, $t3
sw $t5, 100($s2)
    
```

Spreco di 3 cicli di clock (in modo che la fase IF dell'istruzione sub \$s2, \$s0, \$t3 vada a coincidere con la fase di WB della add \$s0, \$t1, \$t1).

Situazione troppo frequente perché la soluzione sia accettabile.

Hazard sui dati

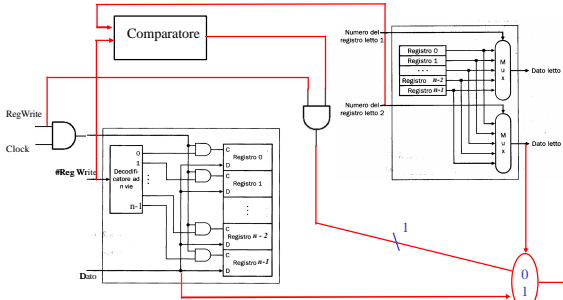
Instruction	IF	ID	EX	MEM	WB				
sub \$s2, \$s1, \$s3					s->\$2				
and \$t2, \$s2, \$s5		ID	EX	MEM					
or \$t3, \$s6, \$s2			ID	EX	MEM	WB			
add \$t4, \$s2, \$s2				ID	EX	MEM	WB		
sw \$t5, 100(\$s2)					IF	EX	MEM	WB	

Con le frecce sono indicate le dipendenze, in blu gli hazard (tra sub e and, sub e or e add).

Il dato in \$s2 viene scritto nel Register File nella fase di WB della sub, è pronto al clock successivo. Non è ancora pronto quando viene effettuata la decodifica della and, della or e della add successiva.

Soluzione criticità sul Register File

Modifichiamo la circuiteria di controllo del Register File.



A.A. 2003-2004

7/33

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

Hazard sui dati

sub \$s2, \$s1, \$s3	IF	ID	EX \$s1-\$s3	MEM	WB s->\$s2				
and \$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	WB \$t5->Mem	

Con le frecce sono indicate le dipendenze, in blu gli hazard (tra sub e and, sub), dopo la modifica del RegisterFile: il dato è disponibile in lettura, già nella prima parte del clock. Il dato in \$s2 viene scritto nel Register File nella fase di WB della sub, è pronto al clock successivo. Non è ancora pronto quando viene effettuata la decodifica della and e della or successiva.

ghese

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).

Propagazione a ritroso (negli stadi della pipeline = in avanti nel tempo) su datapath alternativi dei dati richiesti.

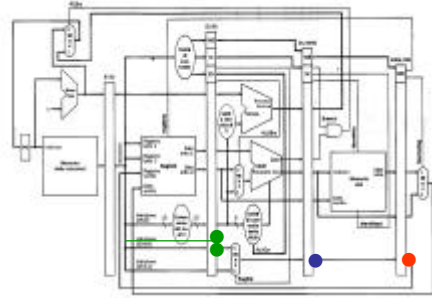
A.A. 2003-2004

9/33

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

Nomenclatura ed identificazione delle criticità

- 1a. EX/MEM.RegistroRd = ID/EX.RegistroRs
- 1b. EX/MEM.RegistroRd = ID/EX.RegistroRt
- 2a. MEM/WB.RegistroRd = ID/EX.RegistroRs
- 2b. MEM/WB.RegistroRd = ID/EX.RegistroRt



A.A. 2003-2004

ghese

Hazard sui dati: rilevamento della criticità

sub \$s2, \$s1, \$s3	IF	ID	EX \$s1-\$s3	MEM	WB s->\$s2				
and \$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. 2003-2004

11/33

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

Hazard sui dati: formalizzazione della criticità

sub \$s2, \$s1, \$s3	IF	ID	EX \$s1-\$s3	MEM	WB s->\$s2				
and \$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)
IF (MEM/WB.RegistroRd == ID/EX.RegistroRt)

A.A. 2003-2004

12/33

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

Hazard sui dati: feed-forwarding

sub \$s2, \$s1, \$s3	IF	ID	EX \$s1-\$s3	MEM	WB s->\$s2			
and \$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)
ALUSrcA = <EX/MEM.RegistroRd>

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

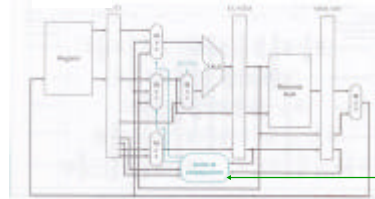
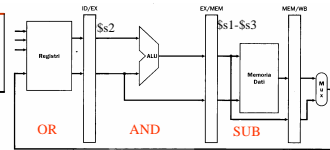
A.A. 2003-2004

13/33

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

Hazard nei dati: forwarding

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



A.A. 2003-2004

14/33

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

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.

A.A. 2003-2004

15/33

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

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.

A.A. 2003-2004

16/33

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

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)
ID/EX.RegistroRs = EX/MEM.RegistroRd

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

Dato preso dalla fase WB:

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

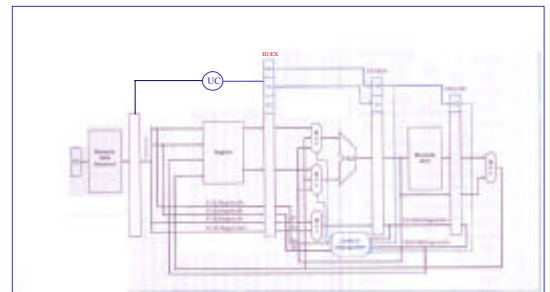
IF (ID/EX.RegistroRt == MEM/WB.RegistroRd) AND (MEM/WB.RegWrite)
ID/EX.RegistroRt = MEM/WB.RegistroRd

A.A. 2003-2004

17/33

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

CPU con unità di propagazione



A.A. 2003-2004

18/33

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

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.
- Inserire una nop.
- Accettare uno stallo (non sempre si può evitare).

A.A. 2003-2004

19/33

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

Sommario

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di tipo R.

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di lw.

A.A. 2003-2004

20/33

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

Hazard sui dati: lw

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

A.A. 2003-2004

21/33

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

Hazard sui dati: lw, rilevamento della criticità

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

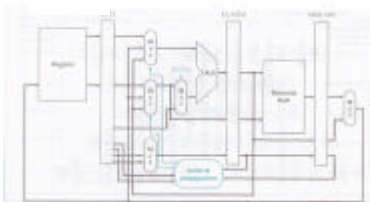
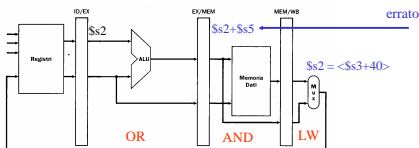
A.A. 2003-2004

22/33

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

Hazard nei dati: lw, forwarding

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



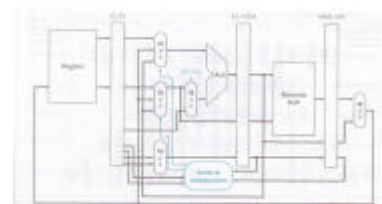
A.A. 2003-2004

23/33

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

Hazard nei dati: lw, unità di propagazione

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



Dato preso dalla fase WB:

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

ID/EX.RegistroRs = MEM/WB.RegistroRd

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

ID/EX.RegistroRt = MEM/WB.RegistroRd

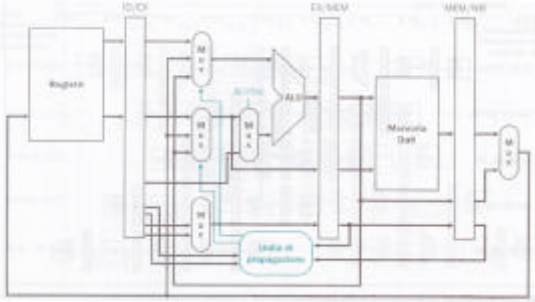
Nessun cambiamento.

A.A. 2003-2004

24/33

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

Il forwarding non è sufficiente



Risolve solamente uno dei due problemi della lw.

Hazard sui dati: lw, stallo

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

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

Stallo della pipeline

Rilevamento della criticità sulla lw

lw \$s2, 40(\$s3)	IF	ID	EX	MEM	WB s->\$s2			
and \$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)		

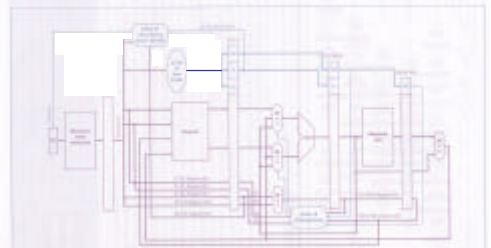
Il dato corretto per \$s2 è pronto nella lw solamente alla fine della fase MEM, ed è perciò utilizzabile solamente a partire dall'inizio della fase di WB.

Rilevo 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?

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]}}
THEN
"Mettilo in stallo la pipeline"

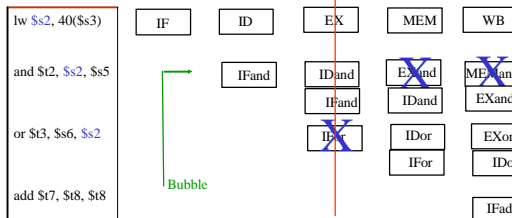


Stallo della pipeline

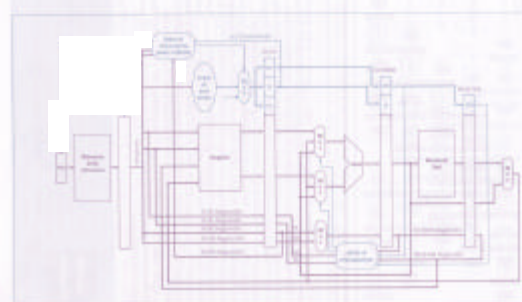
Azioni:

- 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).

Riconoscimento
Criticità

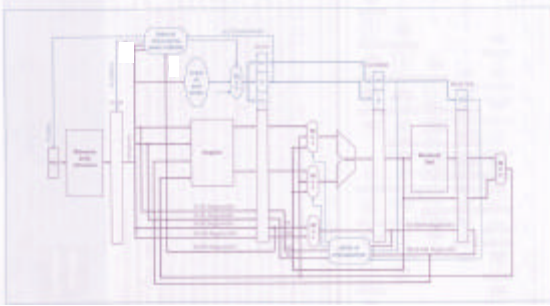


Annullamento dell'istruzione in fase ID



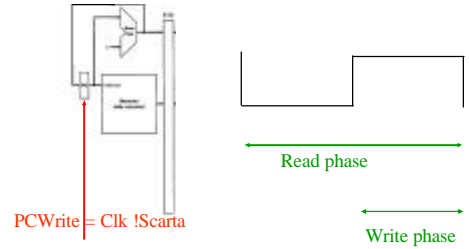
Annullamento dei segnali di controllo associati. Perché invece non annullo la scrittura dei registri ID/EX, EX/MEM e MEM/WB?

Ripetizione delle fasi ID e IF delle due istruzioni successive



Disabilitazione della scrittura del PC e del registro IF/ID.

Disabilitazione della scrittura dei registri



Hp: L'unità di controllo della criticità è in grado di prendere una decisione nella prima metà del ciclo di clock.

Sommario

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di tipo R.

Modifiche alla CPU per la gestione di criticità sui dati, istruzioni di lw.