Pipeline Data Hazards

*Warning, warning, warning!*
Are No-ops Really Necessary?

sub $2, $1,$3
and $4, $2,$5
or $8, $3,$6
add $9, $2,$8
slt $1, $6,$7

Pipeline Stalls

• To insure proper pipeline execution in light of register dependences, we must:
  – the hazard
  – the pipeline
Knowing When to Stall

- 6 types of data hazards
  - two reg reads * 3 reg writes

The Pipeline

- What comparisons tell us when to stall?

Stalling the Pipeline

- Once we detect a hazard, then we have to be able to stall the pipeline (insert a bubble).
- Stalling the pipeline is accomplished by
  - (1) preventing the and stages from making progress
    - the ID stage because it cannot proceed until the dependent instruction completes
    - the IF stage because we do not want to lose any instructions.
  - (2) essentially, inserting “” in hardware

- Preventing the IF and ID stages from proceeding
  - don’t write the PC (PCWrite = 0)
  - don’t rewrite IF/ID register (IF/IDWrite = 0)
- Inserting “nops”
  - set all control signals propagating to EX/MEM/WB to zero
Reducing Data Hazards Through Forwarding

Data Forwarding

- The Previous Data Path handles two types of data hazards
  - hazard
  - hazard
- We assume the register file handles the third (hazard)
  - if the register file is asked to read and write the same register in the same cycle, we assume that the allows the write data to be forwarded to the output
  - We’re still going to call that...

Eliminating Data Hazards via Forwarding
Forwarding in Action

Instruction Fetch

add $1, $12, $3
sub $12, $3, $4
add $3, $10, $11

Write Back

Memory Access

CSE 141
Dean Tullsen

Eliminating Data Hazards via Forwarding??

lw $2, 10($1)
and $12, $2, $5
or $13, $6, $2
add $14, $2, $2
sw $15, 100($2)

CSE 141
Dean Tullsen
Eliminating Data Hazards via Forwarding and stalling

lw $2, 10($1)

and $12, $2, $5

or $13, $6, $2

add $14, $2, $2

sw $15, 100($2)

Datapath with Hazard-Detection

if (ID/EX.MemRead and ((ID/EX.RegisterRt = IF/ID.RegisterRs) or (ID/EX.RegisterRt = IF/ID.RegisterRt)))

then stall the pipeline

Hazard Detection

and $4, $2, $5

lw $2, 20($1)

Try this one...

Show stalls and forwarding for this code

add $3, $2, $1

lw $4, 100($3)

and $6, $4, $3

sub $7, $6, $2

add $9, $3, $6
Hazard Detection

Data Hazard Key Points

• Pipelining provides high throughput, but does not handle data dependences easily.
• Data dependences cause data hazards.
• Data hazards can be solved by:
  – software (nops)
  – hardware stalling
  – hardware forwarding
• Our processor, and indeed all modern processors, use a combination of forwarding and stalling.
• ET = IC * CPI * CT