

## Organização de Computadores

### Aula 11

#### Pipelines segunda parte

INF01113 - Organização de Computadores

## Alguns exercícios (I)

- Dado um processador com 5 estágios de pipeline (F,DR,O,M,W), Tciclo = 2ns, pede-se:
  - quanto tempo será tomado para executar 1B de instruções?
  - qual a aceleração em relação a:
    - uma máquina single-cycle com Tcycle = 8ns;
    - uma máquina multi-cycle com Tcycle = 2ns, CPI = 4;
- Graças a melhora na tecnologia, passa-se o pipeline a 7 estágios (FF,DR,OO,MM,W), com Tciclo = 1ns.
  - Quanto tempo será tomado para executar 1B de instruções?
  - Qual a aceleração em relação a:
    - uma máquina single-cycle com Tcycle = 8ns;
    - uma máquina multi-cycle com Tcycle = 1ns, CPI = 5;

INF01113 - Organização de Computadores

## Alguns exercícios (II)

- O pipe de 5 estágios decide o salto no quarto estágio. A política usada é esvaziar o pipe. Qual a penalidade paga, se 30% das instruções forem branches? Qual o CPI?
- Melhorou-se o processador com uma política de predição de saltos com 85% de acertos. Qual a melhoria em desempenho? Qual CPI se tem?
- Repita o problema para um pipe de 7 estágios, onde a decisão sobre saltos acontece no 5º estágio.

INF01113 - Organização de Computadores

## Pipelines segunda parte

1. Introdução
2. Dependências verdadeiras
3. Dependências falsas
4. Pipeline interlock
5. Forwarding

INF01113 - Organização de Computadores

## 1. Introdução

- problema: instruções consecutivas podem fazer acesso aos mesmos operandos
  - execução da instrução seguinte pode depender de operando calculado pela instrução anterior
- caso particular: instrução precisa de resultado anterior (p.ex. registrador) para cálculo de endereço efetivo de operando
- tipos de dependências de dados
  - dependência verdadeira
  - antidependência
  - dependência de saída

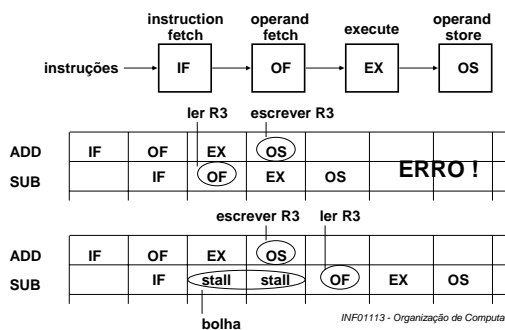
INF01113 - Organização de Computadores

## 2. Dependências verdadeiras

- exemplo
  - ADD R3, R2, R1 ;  $R3 = R2 + R1$
  - SUB R4, R3, 1 ;  $R4 = R3 - 1$
- instrução 2 depende de valor de R3 calculado pela instrução 1
- instrução 1 precisa atualizar valor de R3 antes que instrução 2 busque os seus operandos
- read-after-write hazard*
- pipeline precisa ser parado durante certo número de ciclos

INF01113 - Organização de Computadores

## Dependências verdadeiras



INF01113 - Organização de Computadores

## 3. Dependências falsas

### Antidependência

- exemplo
  - ADD R3, R2, R1 ;  $R3 = R2 + R1$
  - SUB R2, R4, 1 ;  $R2 = R4 - 1$
- instrução 1 utiliza operando em R2 que é escrito pela instrução 2
- instrução 2 não pode salvar resultado em R2 antes que instrução 1 tenha lido seus operandos
- write-after-read hazard*
- não é um problema em pipelines onde a ordem de execução das instruções é mantida
  - escrita do resultado é sempre o último estágio
- problema em processadores super-escalares

INF01113 - Organização de Computadores

## Dependências falsas

### Dependência de saída

- exemplo
  1. ADD R3, R2, R1 ;  $R3 = R2 + R1$
  2. SUB R2, R3, 1 ;  $R2 = R3 - 1$
  3. ADD R3, R2, R5 ;  $R3 = R2 + R5$
- instruções 1 e 3 escrevem no mesmo R3
- instrução 1 tem que escrever seu resultado em R3 antes do que a instrução 3, senão valor final de R3 ficará errado
- write-after-write hazard*
- também só é problema em processadores super-escalares

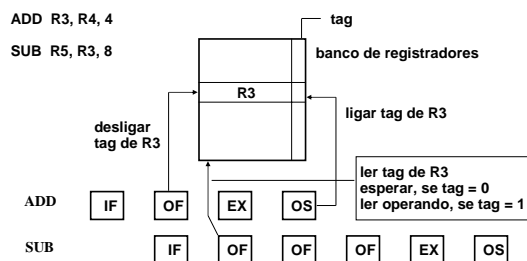
INF01113 - Organização de Computadores

## 4. Pipeline interlock

- método para manter sequência correta de leituras e escritas em registradores
- tag de 1 bit é associado a cada registrador
  - tag = 0 indica valor não válido, = 1 indica valor válido
- se instrução que é buscada e decodificada escreve num registrador, o tag do mesmo é zerado
- tag é ligado quando instrução escreve o valor no registrador
- outras instruções que sejam executadas enquanto tag está zerado devem esperar tag = 1 para ler valor deste registrador

INF01113 - Organização de Computadores

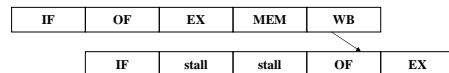
## Pipeline interlock



INF01113 - Organização de Computadores

## 5. Forwarding

- exemplo
  - ADD R3, R2, R0
  - SUB R4, R3, 8
- instrução SUB precisa do valor de R3, calculado pela instrução ADD
  - valor é escrito em R3 por ADD no último estágio WB (write-back)
  - valor é necessário em SUB no terceiro estágio
  - instrução SUB ficará presa por 2 ciclos no 2º estágio do pipeline

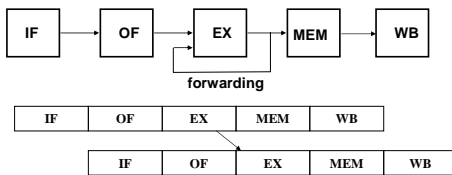


supõe-se escrita no banco de registradores na primeira metade do ciclo e leitura na segunda metade

INF01113 - Organização de Computadores

### Forwarding

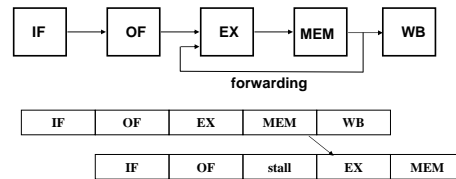
- caminho interno dentro do pipeline entre a saída da ALU e a entrada da ALU
  - evita *stall* do pipeline



INF01113 - Organização de Computadores

### Forwarding

- exemplo 2  
 LOAD R3, 100 (R0)  
 ADD R1, R2, R3
- forwarding: caminho interno dentro do pipeline entre a saída da memória de dados e a entrada da ALU



INF01113 - Organização de Computadores

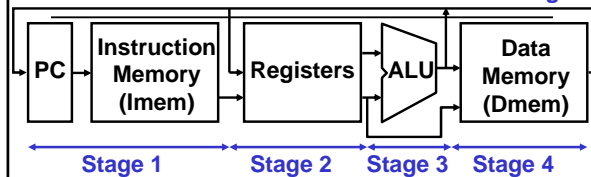
### E no MIPS?

- Como fica o datapath?
- Como fica o controle

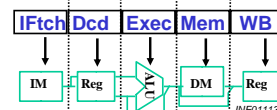
INF01113 - Organização de Computadores

### Review: Single-cycle Datapath for MIPS

Stage 5



- Use datapath figure to represent pipeline



INF01113 - Organização de Computadores

