Universidade Federal do Rio Grande do Sul
Instituto de Informática

Organização de Computadores

Aula 12

Pipeline Pt. 3

“Soluções para os conflitos no Pipeline”

Introdução

- **Nas últimas duas aulas vimos como o pipeline permite acelerar o funcionamento dos processadores.**

- **Vimos também que se de um lado o aceleramento teórico pode chegar ao número de estágios, existem diversos conflitos que prejudicam o desempenho pleno do pipeline:**

  - **Conflitos de Recursos**

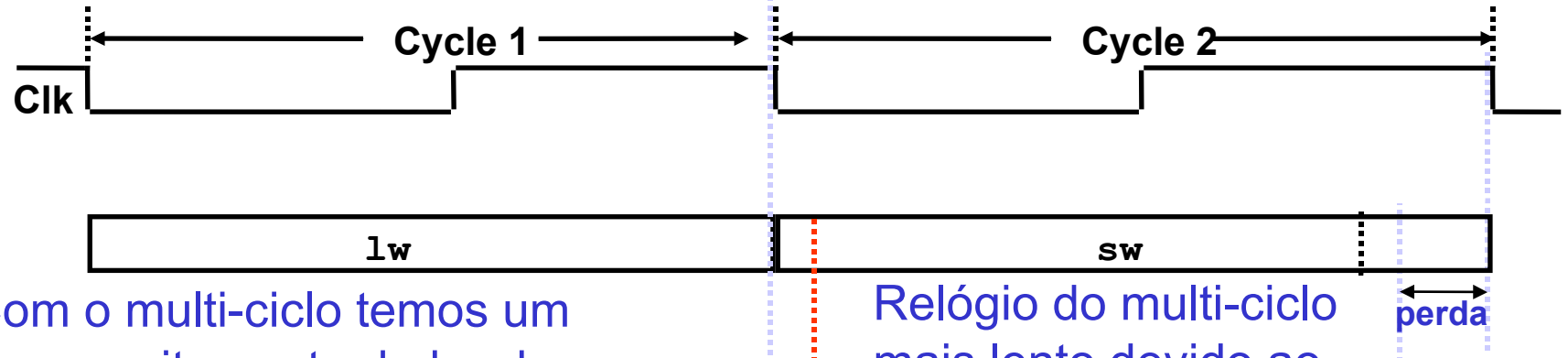
  - **Instruções de Desvios**

  - **Dependências de Dados**

Revisão sobre Ciclos

Implementação Mono-Ciclo:

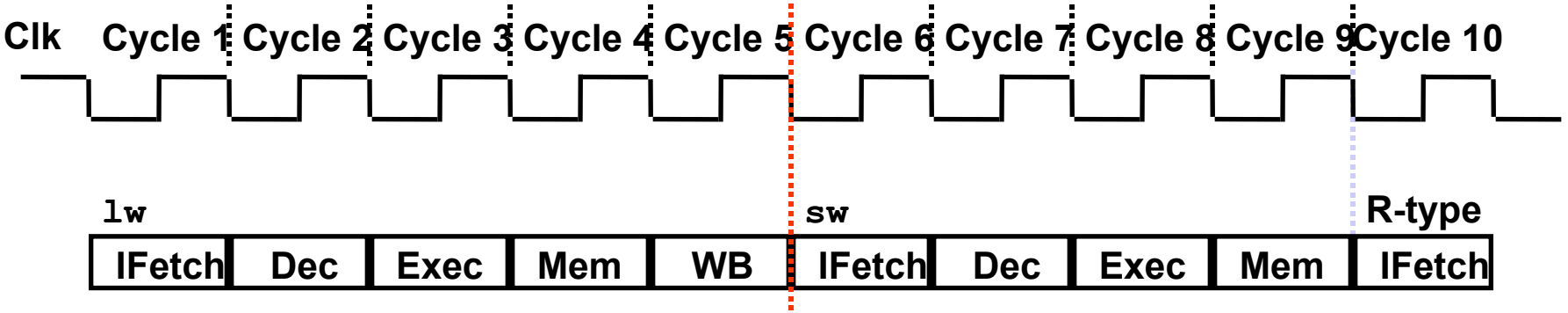
No multi-ciclo podemos ter instruções com diferentes quantidades de ciclos



Com o multi-ciclo temos um reaproveitamento de hardware

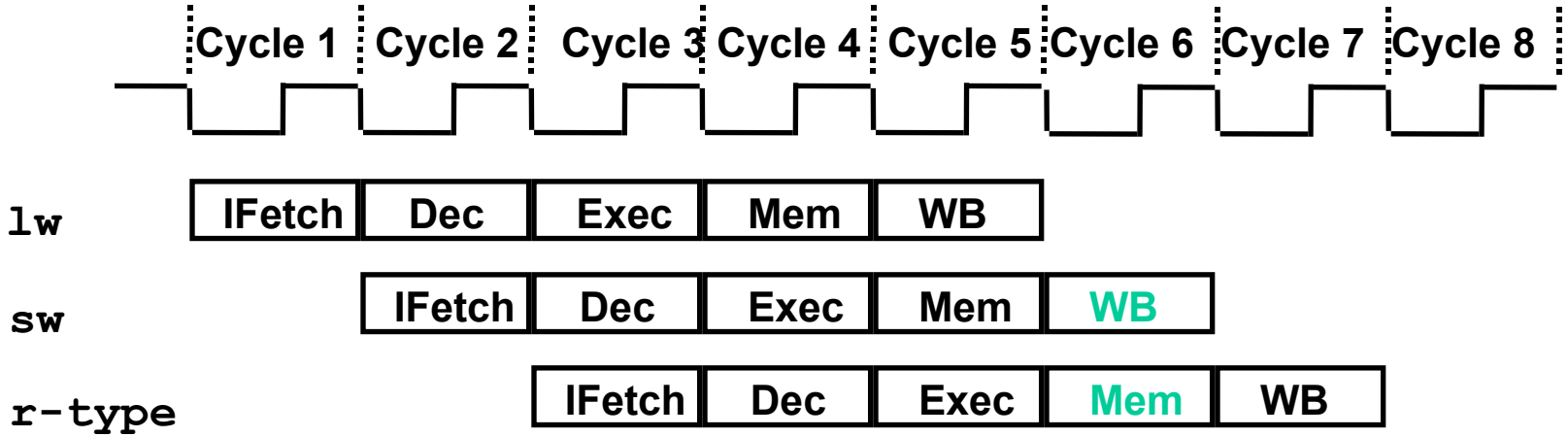
Relógio do multi-ciclo mais lento devido ao custo dos registradores

Implementação Multi Ciclo:



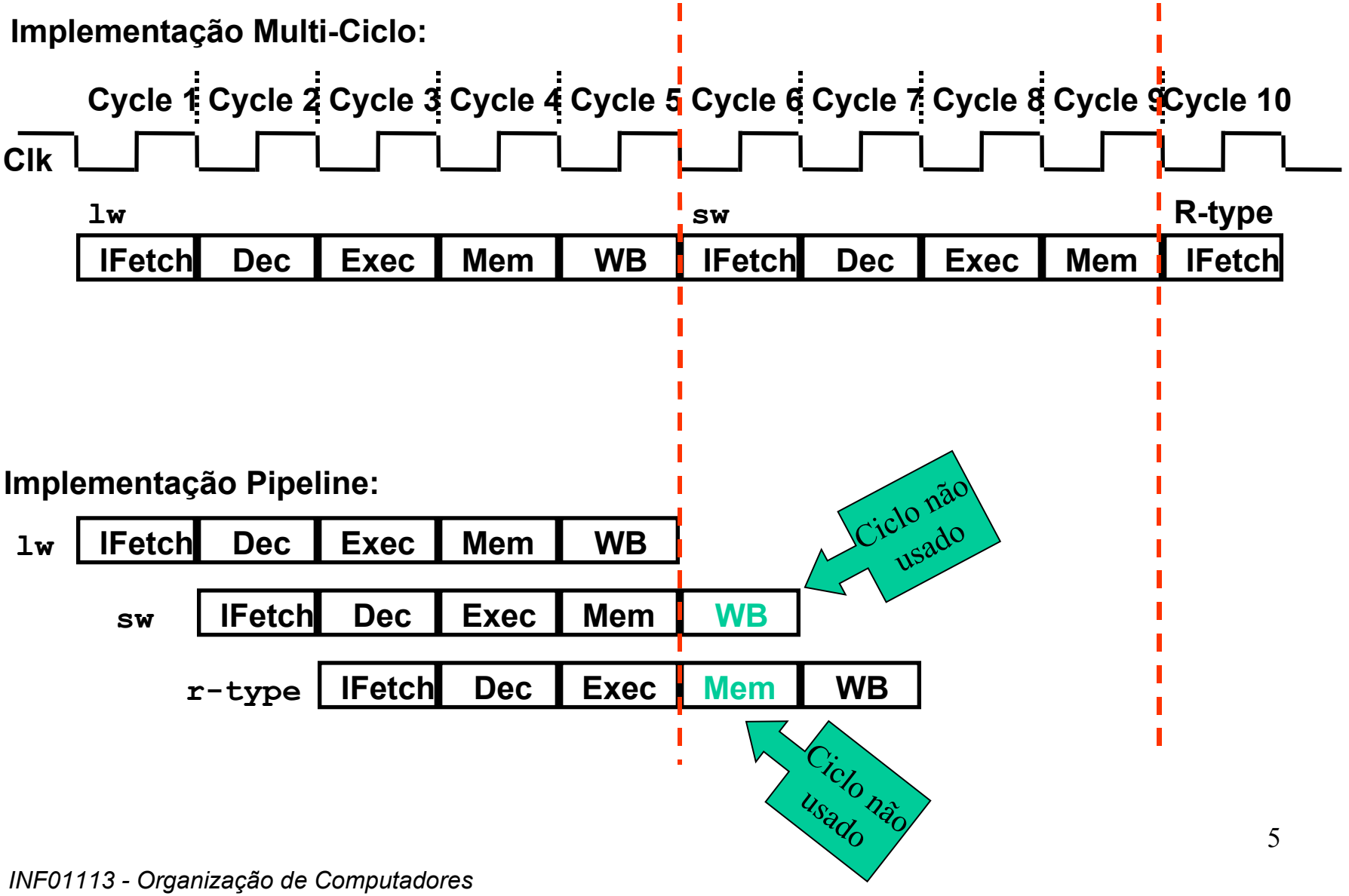
Emprego de Pipeline no Processador MIPS

- **Iniciar a próxima instrução antes da corrente ter sido concluída:**
  - **Aumenta a vazão total de trabalho executado num determinado tempo.**
  - **Tenta aproveita dispositivos que estariam sem utilização.**
  - **A latência da instrução não é reduzida (tempo para a execução de uma instrução do seu início até seu fim) .**



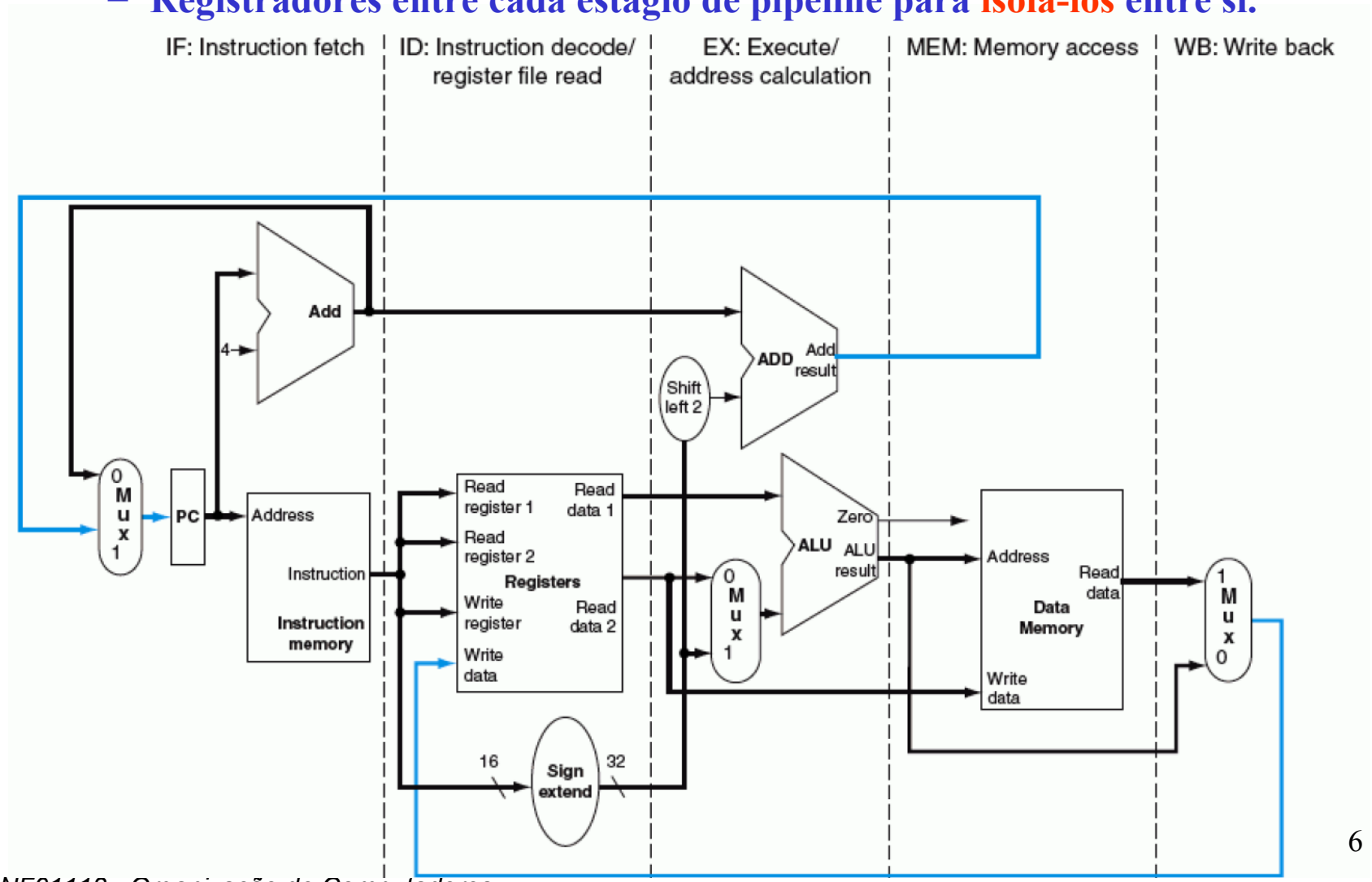
- **Ciclo do relógio (tempo de um estágio de pipeline) é limitado pelo estágio mais lento**
- **Para algumas instruções, alguns ciclos são perdidos (não usados)**

Implementação Multi Ciclo x Pipeline

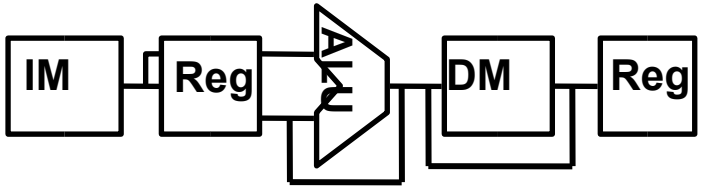


Modificações no Datapath para o MIPS Pipeline

- **O que devemos adicionar ou modificar no caminho dos dados do MIPS?**
  - **Registradores entre cada estágio de pipeline para isolá-los entre si.**

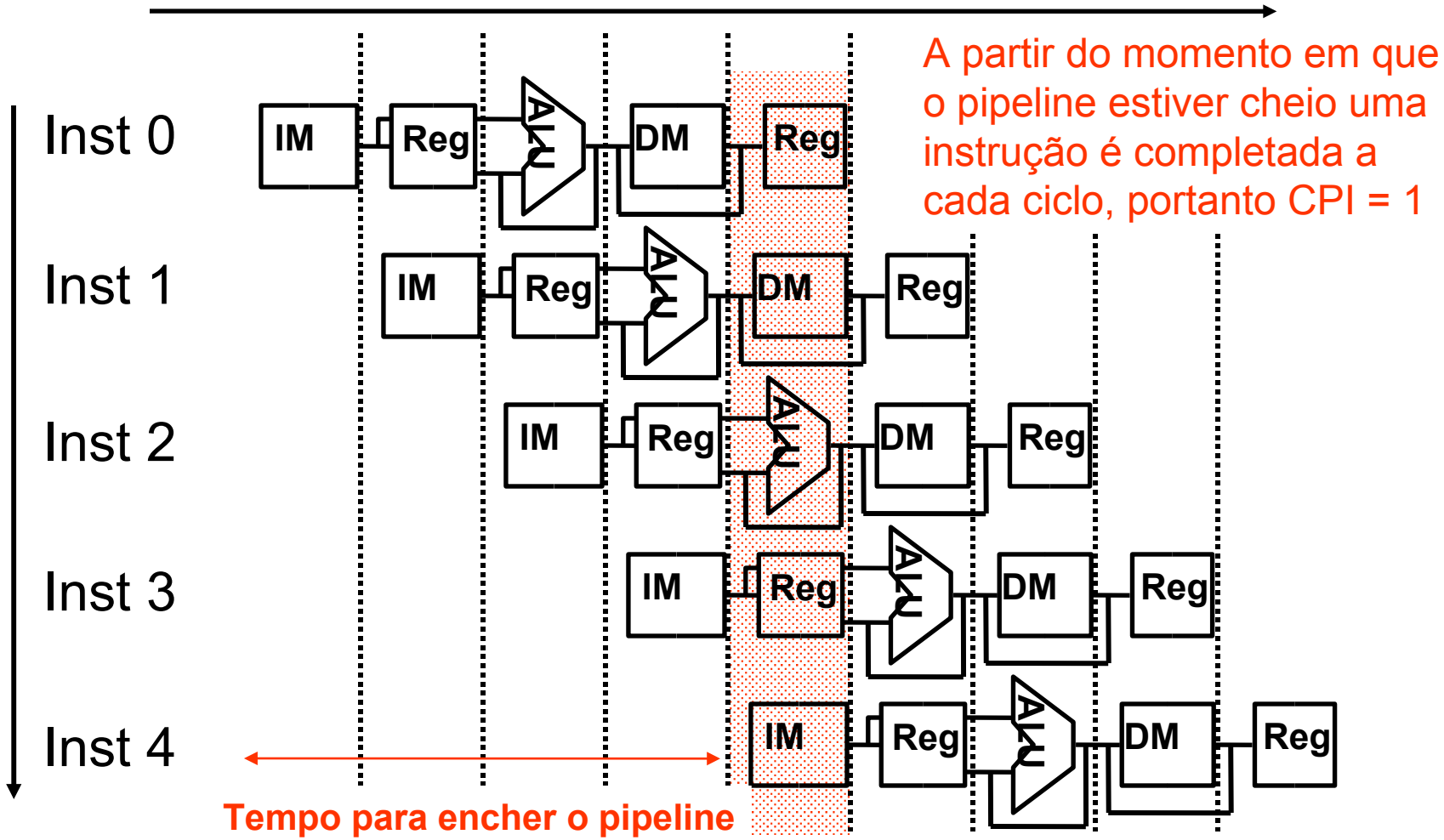


Pipeline da ISA no MIPS



- **O que é fácil**
  - **Todas instruções possuem o mesmo tamanho (32 bits)**
    - **Podem ser buscadas no 1º estágio e decodificadas no 2º estágio**
  - **Poucos formatos de instruções (três) com simetria nos formatos**
    - **Pode começar a ler o arquivo de registradores no 2º estágio**
  - **Operações com memórias ocorrem somente nos loads e stores**
    - **Pode empregar o estágio de execução para calcular o endereço de memória,**
  - **Cada instrução MIPS armazena ao menos um resultado (i.e., troca o estado da máquina) e faz no fim do pipeline (MEM e WB)**
- **O que é difícil**
  - **Conflitos de recursos: O que acontece se temos uma memória só?**
  - **Conflitos de controle : O que acontece com os desvios?**
  - **Conflitos de dados : O que acontece quando a entrada de uma instrução depende da saída da instrução anterior?**

Desempenho do Pipeline



Quais os problemas num Pipeline?

- **Conflitos no Pipeline**

Conflitos de Recursos: uso do mesmo reccurso por duas diferentes instruções no mesmo tempo,

Conflitos de Controle: necessidade de ter uma decisão sobre o fluxo de controle do programa antes que a condição foi avaliada e o novo valor do endereço do PC ter sido calculado,

- **Instruções de Salto condicional.**

Conflitos de Dados: uso de um dado antes de estar pronto, uma instrução precisa de um operando de uma instrução anterior que ainda está no pipeline,

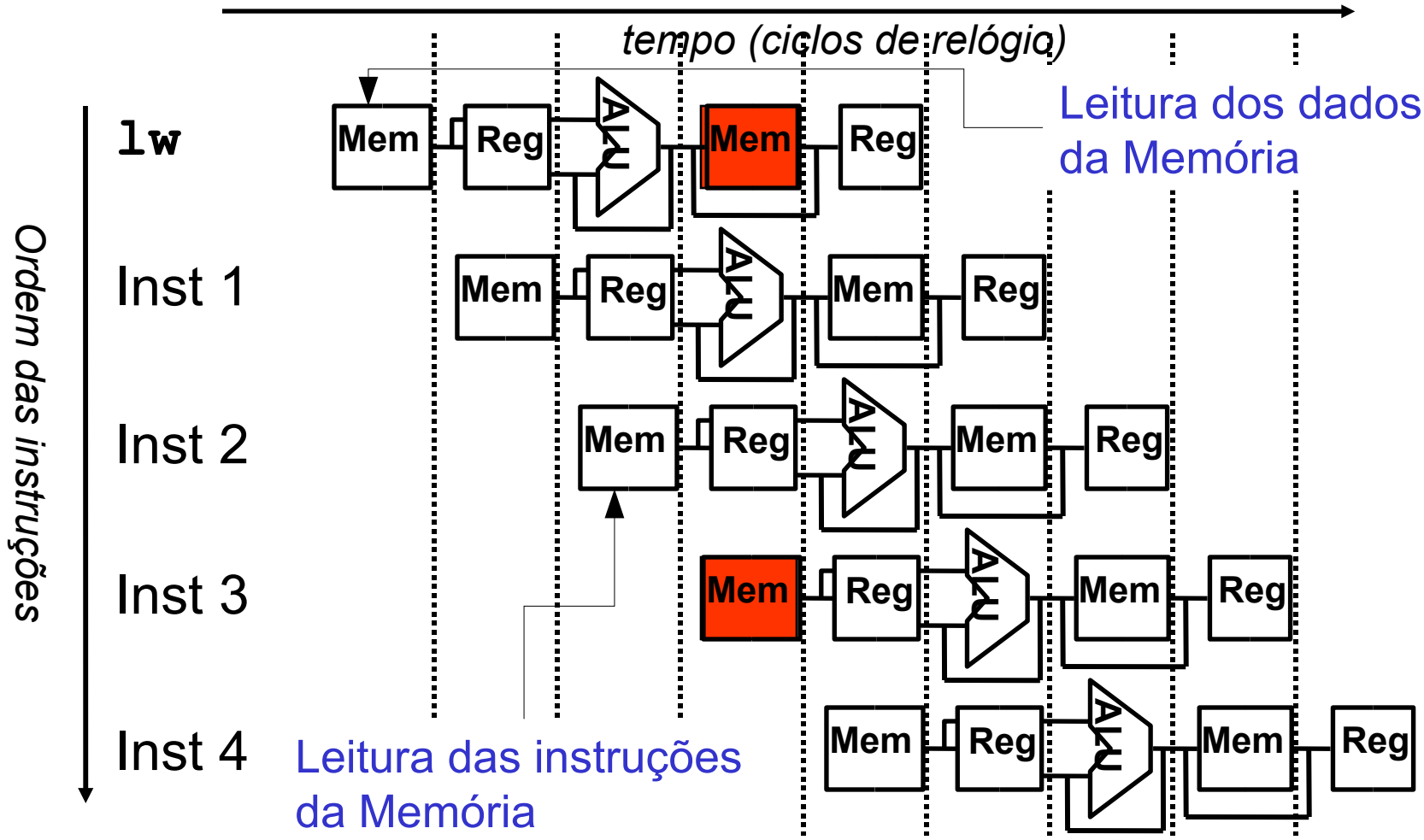
- **Conflitos sempre podem ser resolvidos esperando**

OU

- **O controle do pipeline pode detectar o conflito e tomar medidas para resolve-lo**

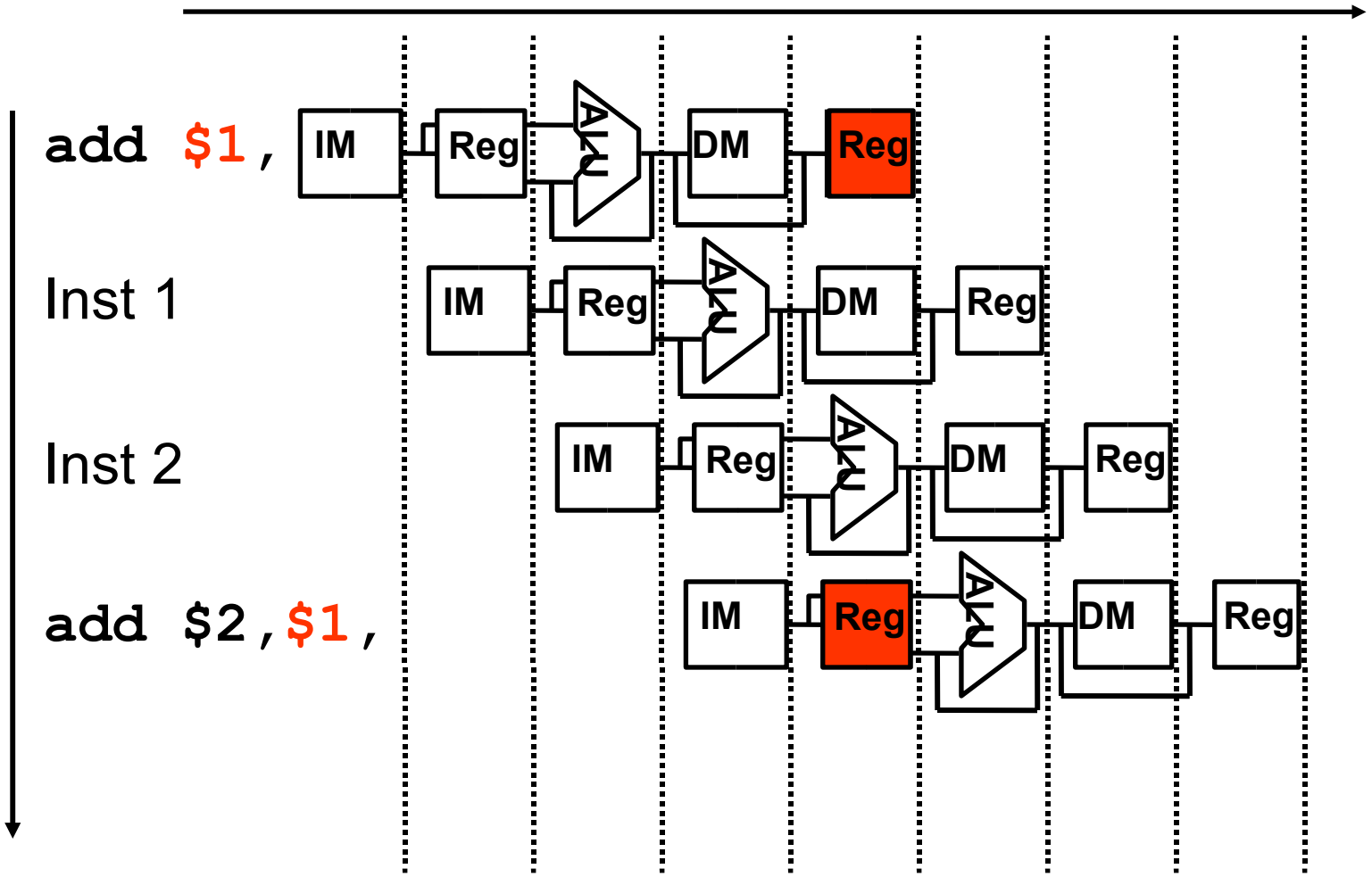
Conflitos de Recursos

Emprego de Memória única

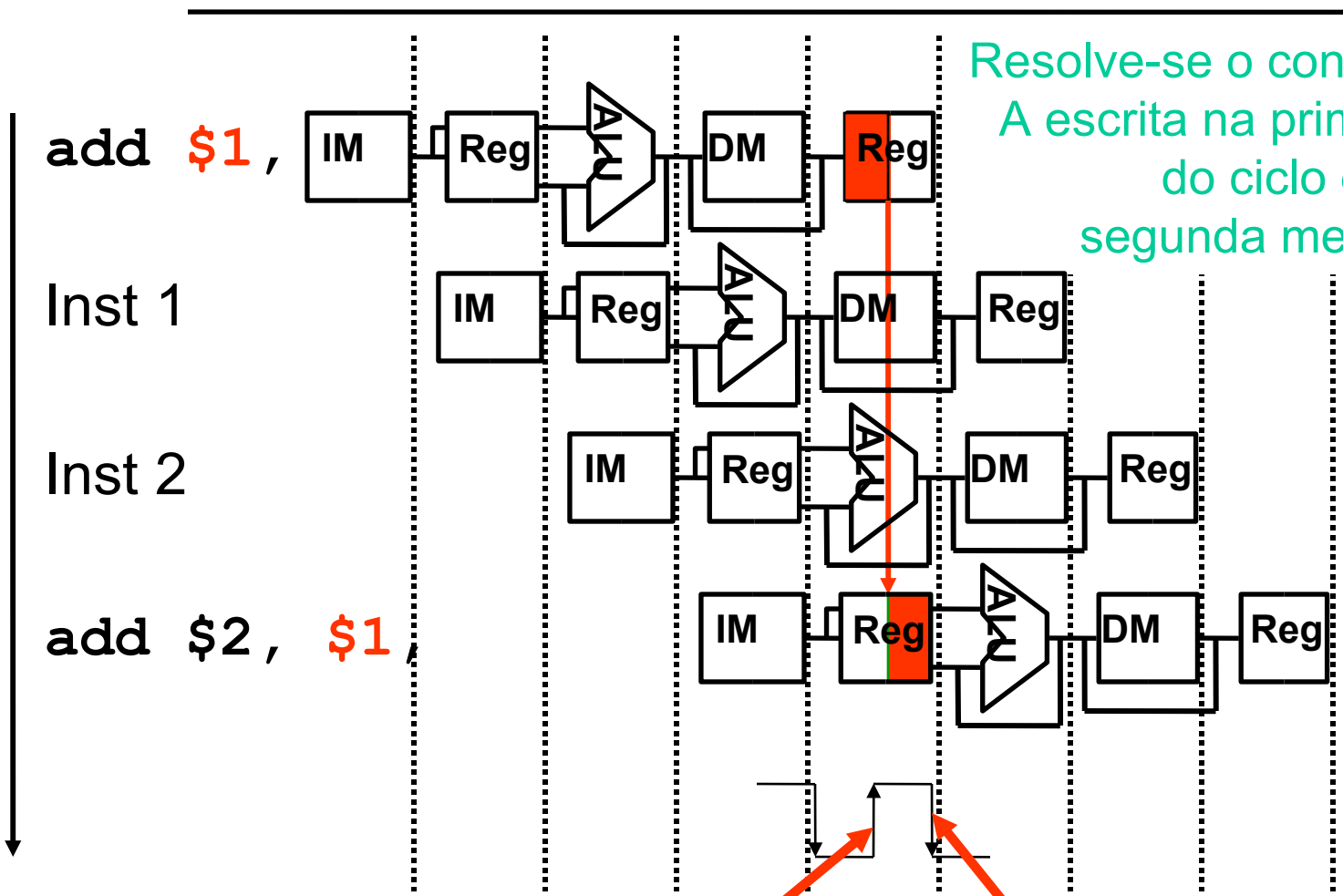


- **Solução: emprego de memórias separadas para dados e instruções**

Acesso a dados nos Registradores



Solução para acesso a dados nos Registradores



Resolve-se o conflito fazendo:
A escrita na primeira metade do ciclo e a leitura na segunda metade do ciclo

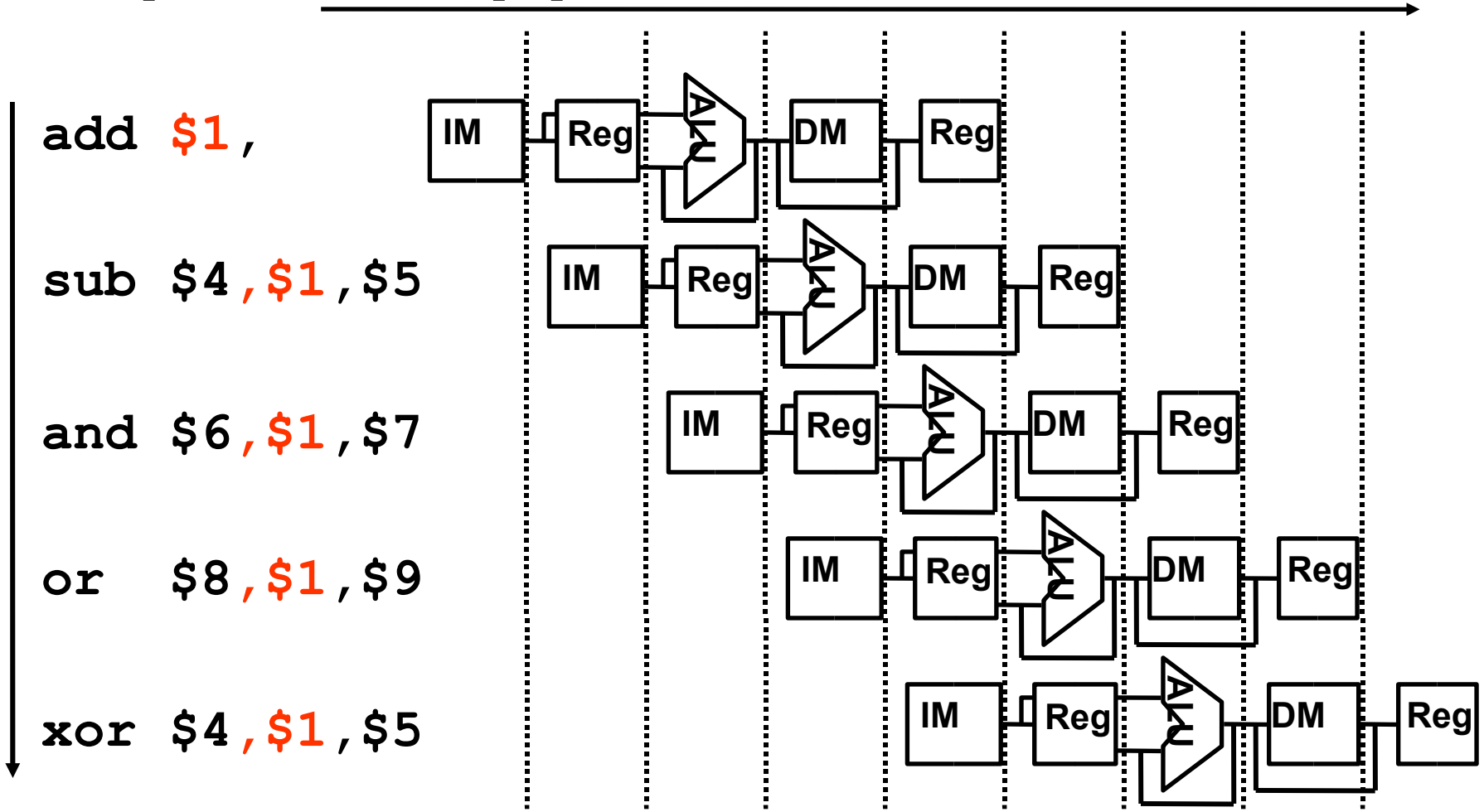
Borda do relógio que controla a escrita no registrador

Borda do relógio que controla a leitura de registrador

Conflitos de Dados

Uso de Registradores podem causar Conflitos de Dados

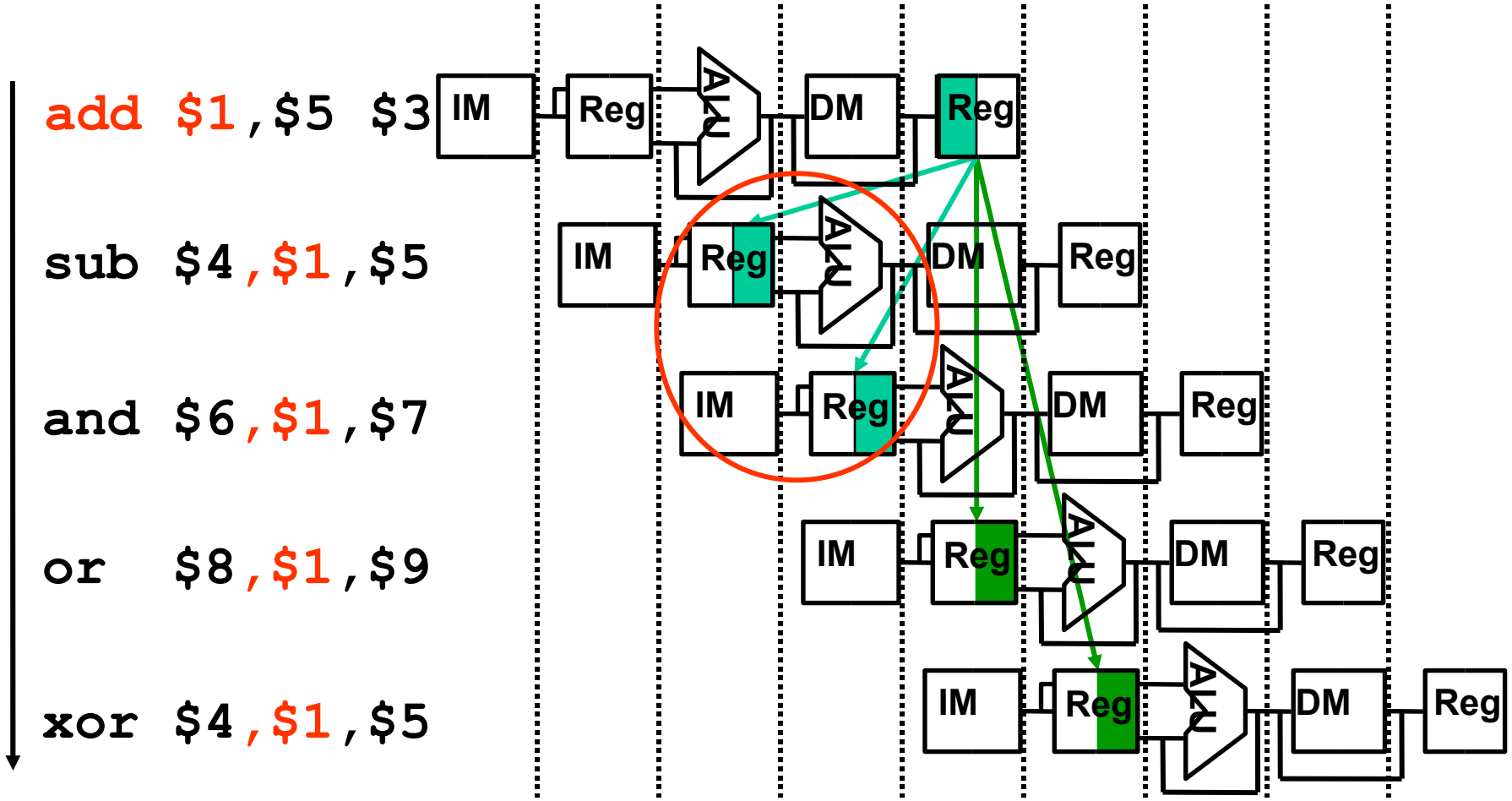
- Dependências no tempo para trás (backward) causam conflitos



- Ler antes de escrever: conflito de dados

Uso de Registradores podem causar Conflitos de Dados

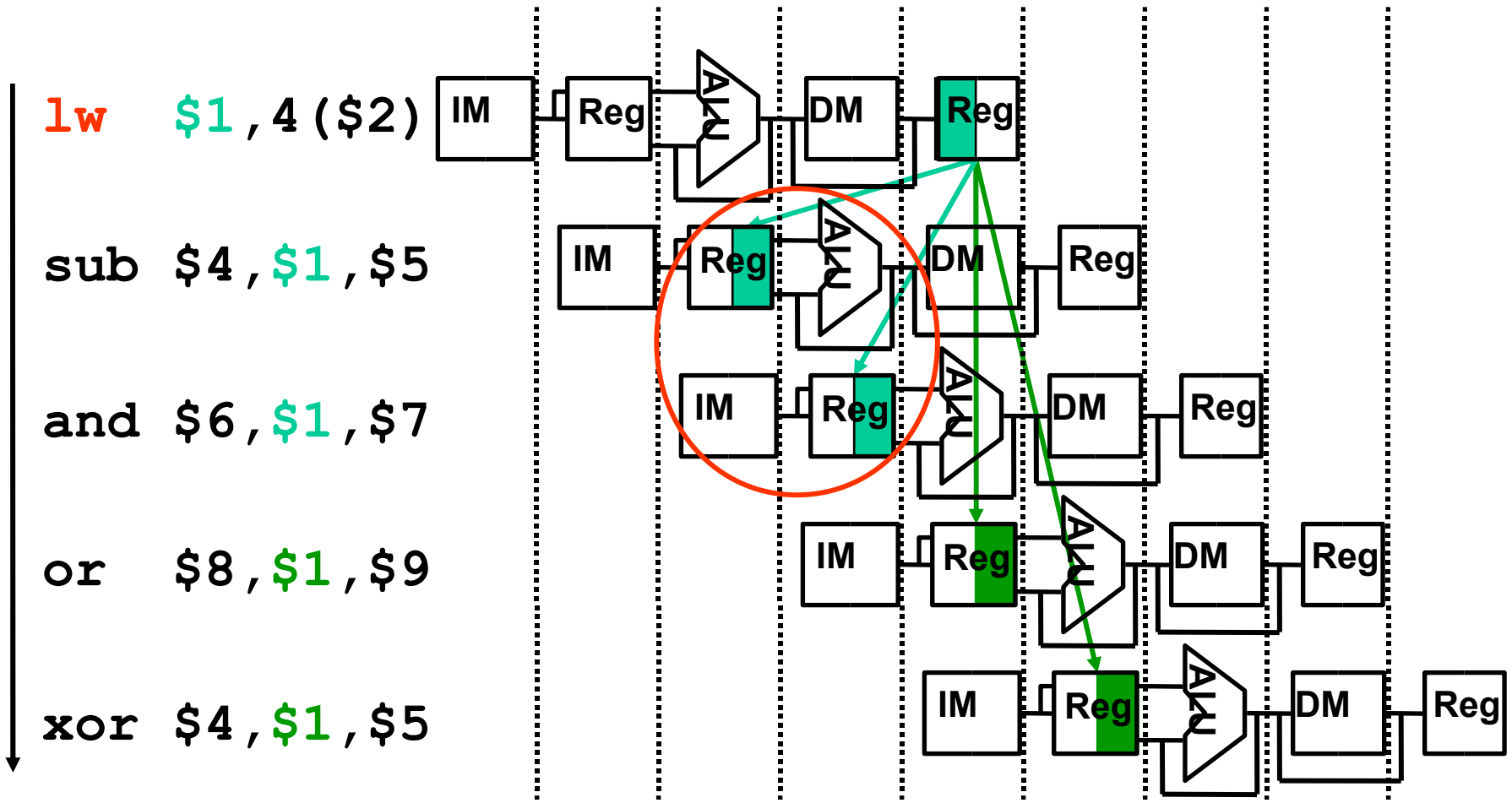
- Dependências no tempo para trás (backward) causam conflitos



- Ler antes de escrever: conflito de dados

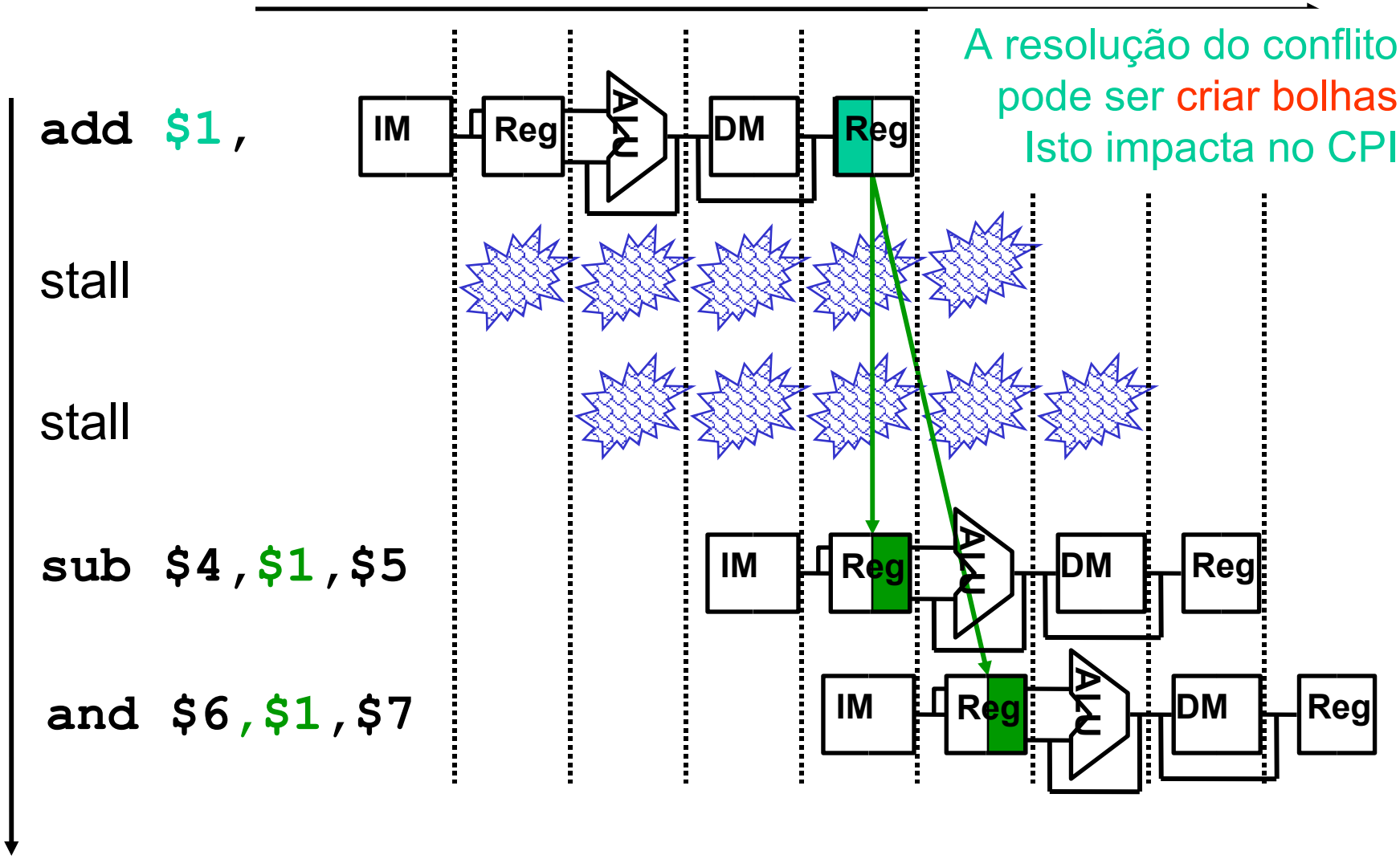
A instrução Load pode causar Conflitos de Dados

- Dependencias para trás (backward) causam conflitos

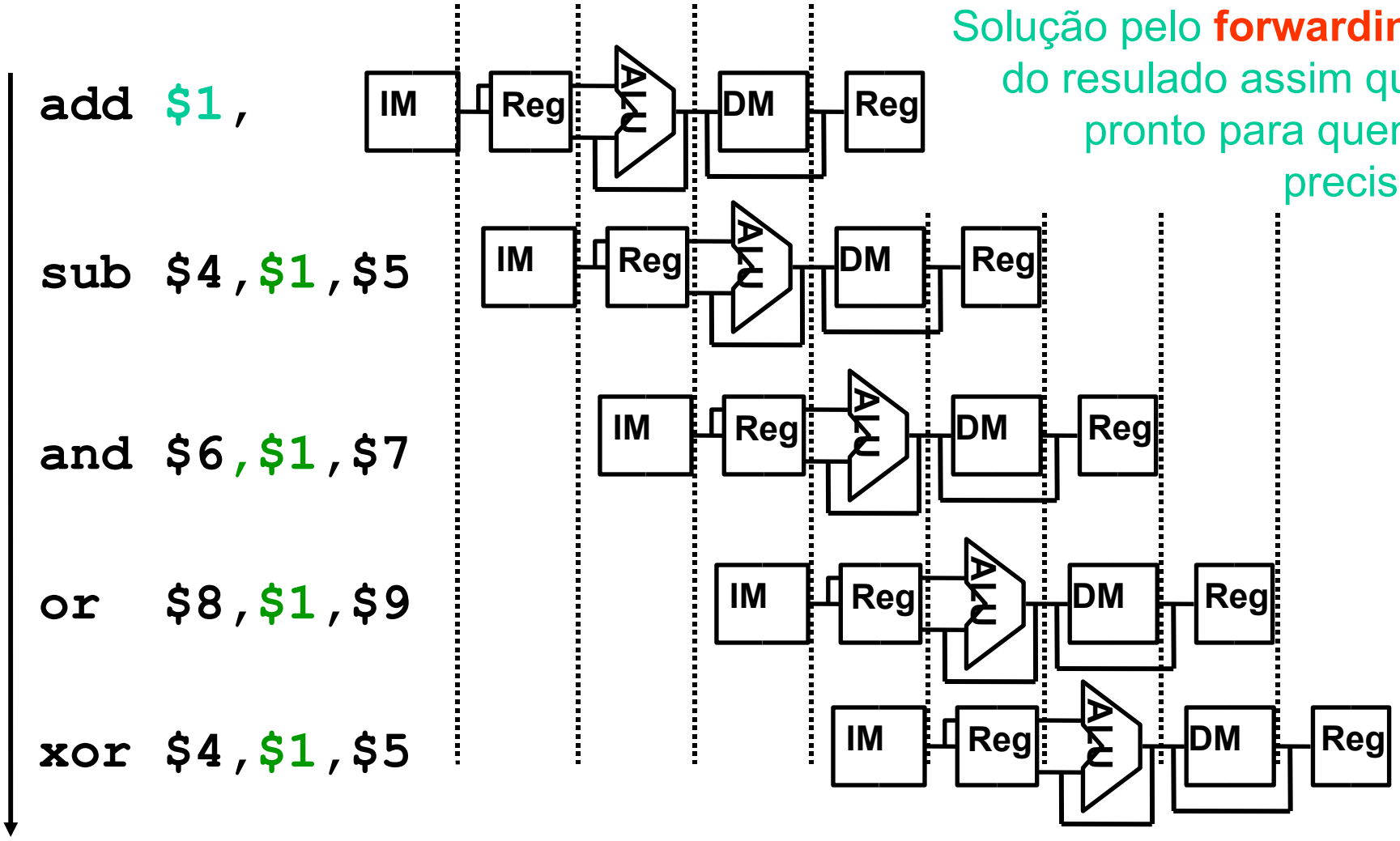


- Uso de Load causa conflito de dados

Uma solução para resolver o Conflito de Dados



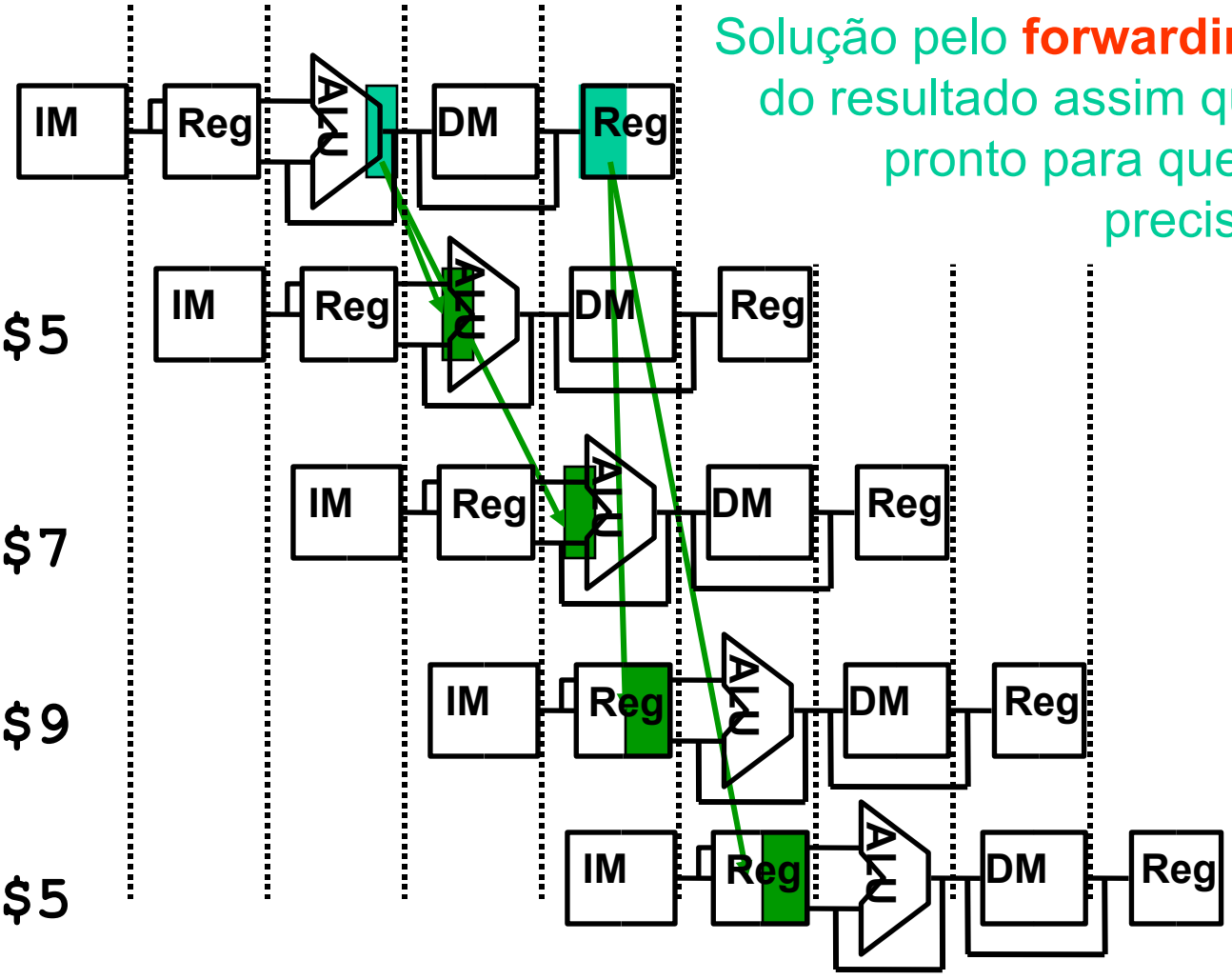
Outra maneira para resolver um conflito de Dados



Solução pelo **forwarding** do resulado assim que pronto para quem precisar

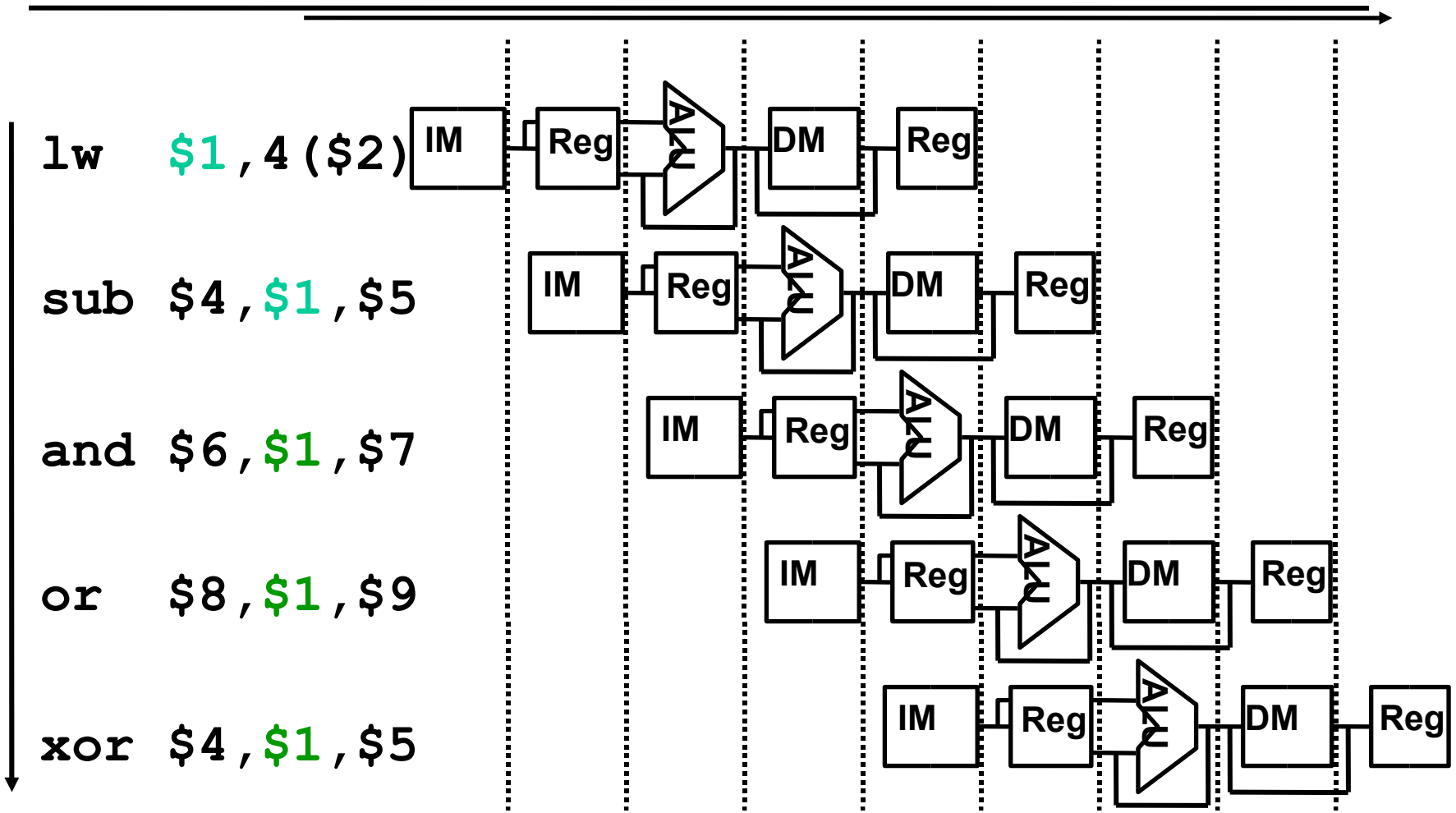
Outra maneira para Resolver um conflito de Dados

```
add $1,
sub $4,$1,$5
and $6,$1,$7
or  $8,$1,$9
xor $4,$1,$5
```

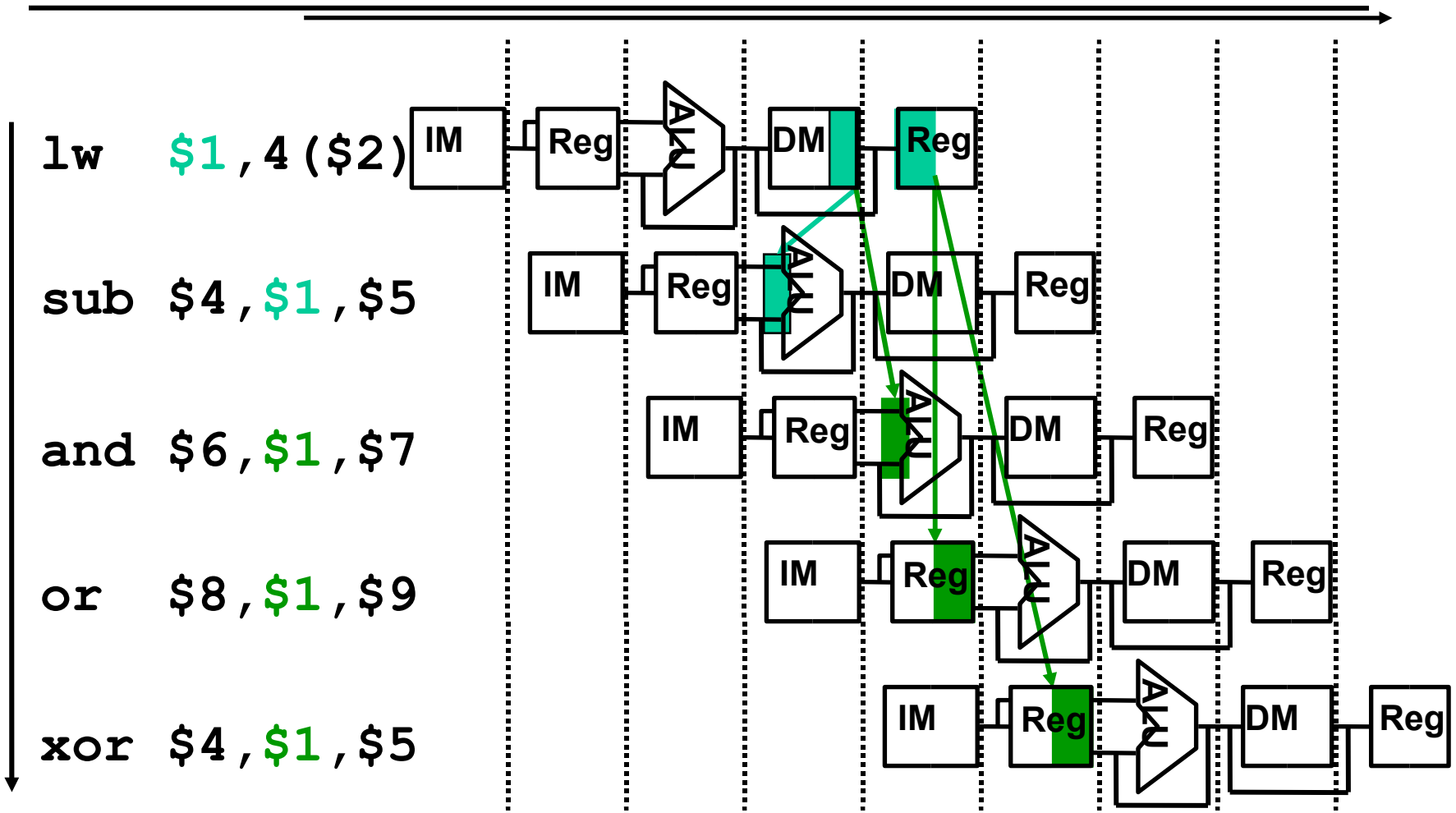


Solução pelo **forwarding** do resultado assim que pronto para quem precisar

Forwarding com Conflito de Dados em Load

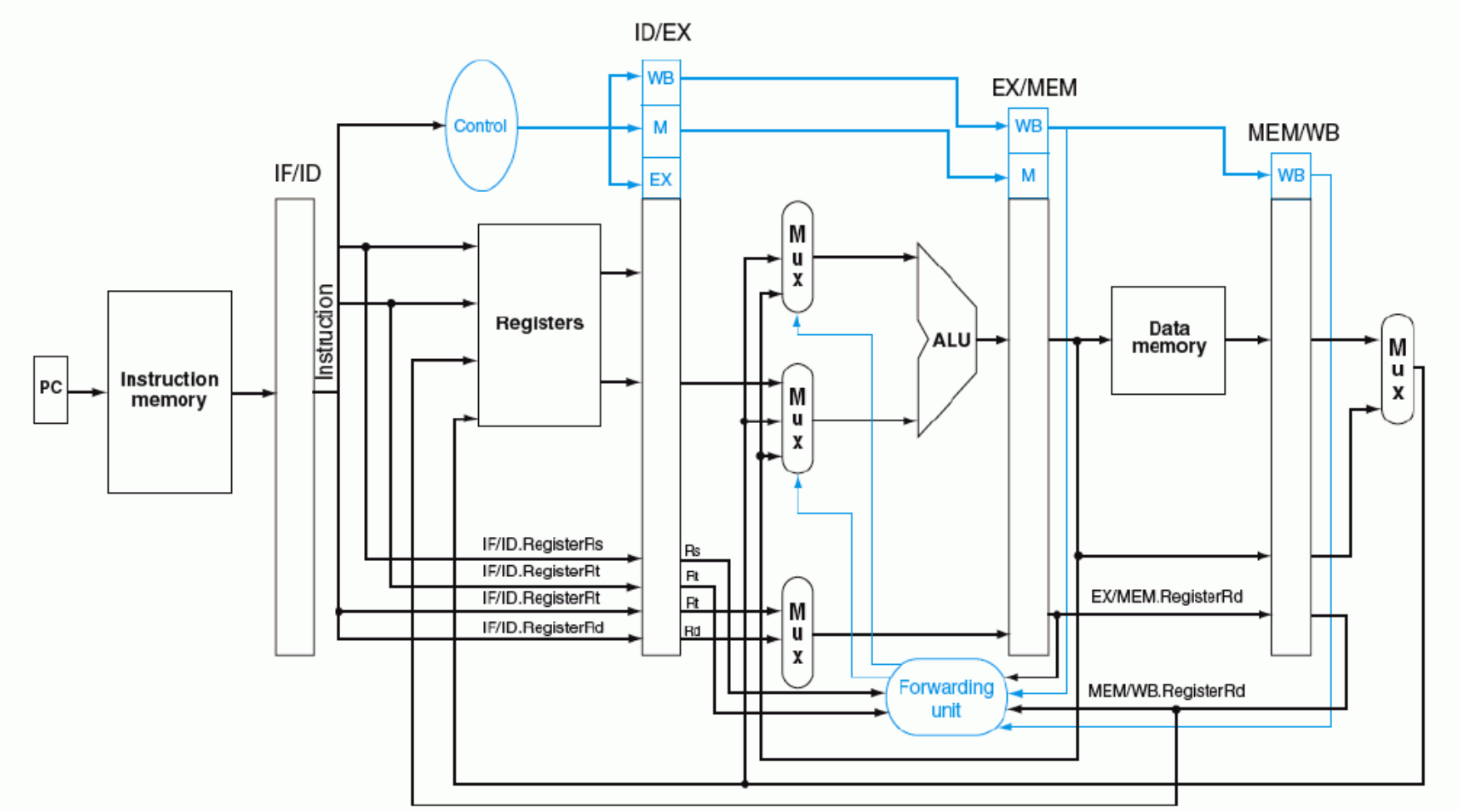


Forwarding com Conflito de Dados em Load



- **Ainda assim precisamos de um ciclo de atraso mesmo com o frowarding**

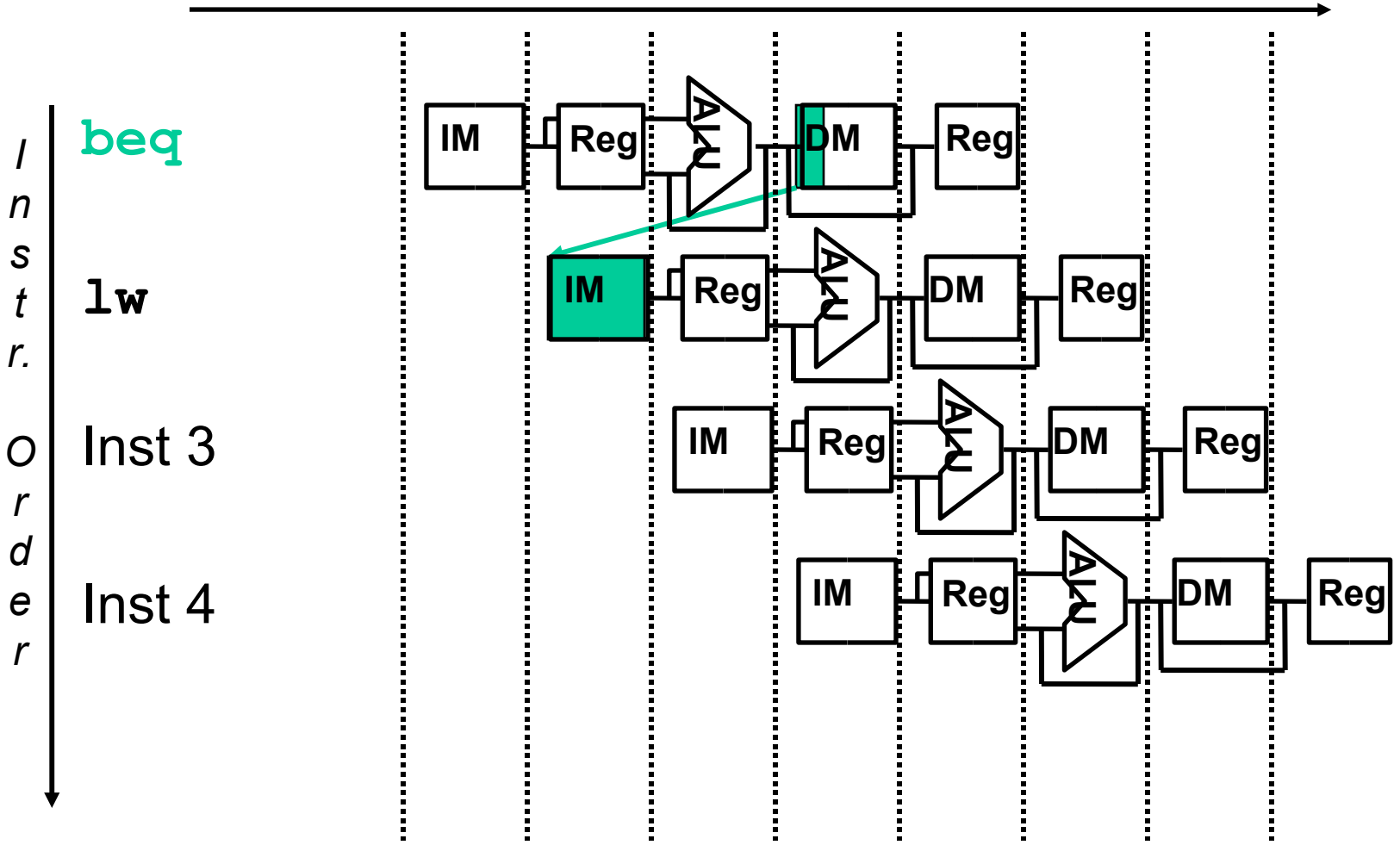
Pipeline do MIPS com Forwarding



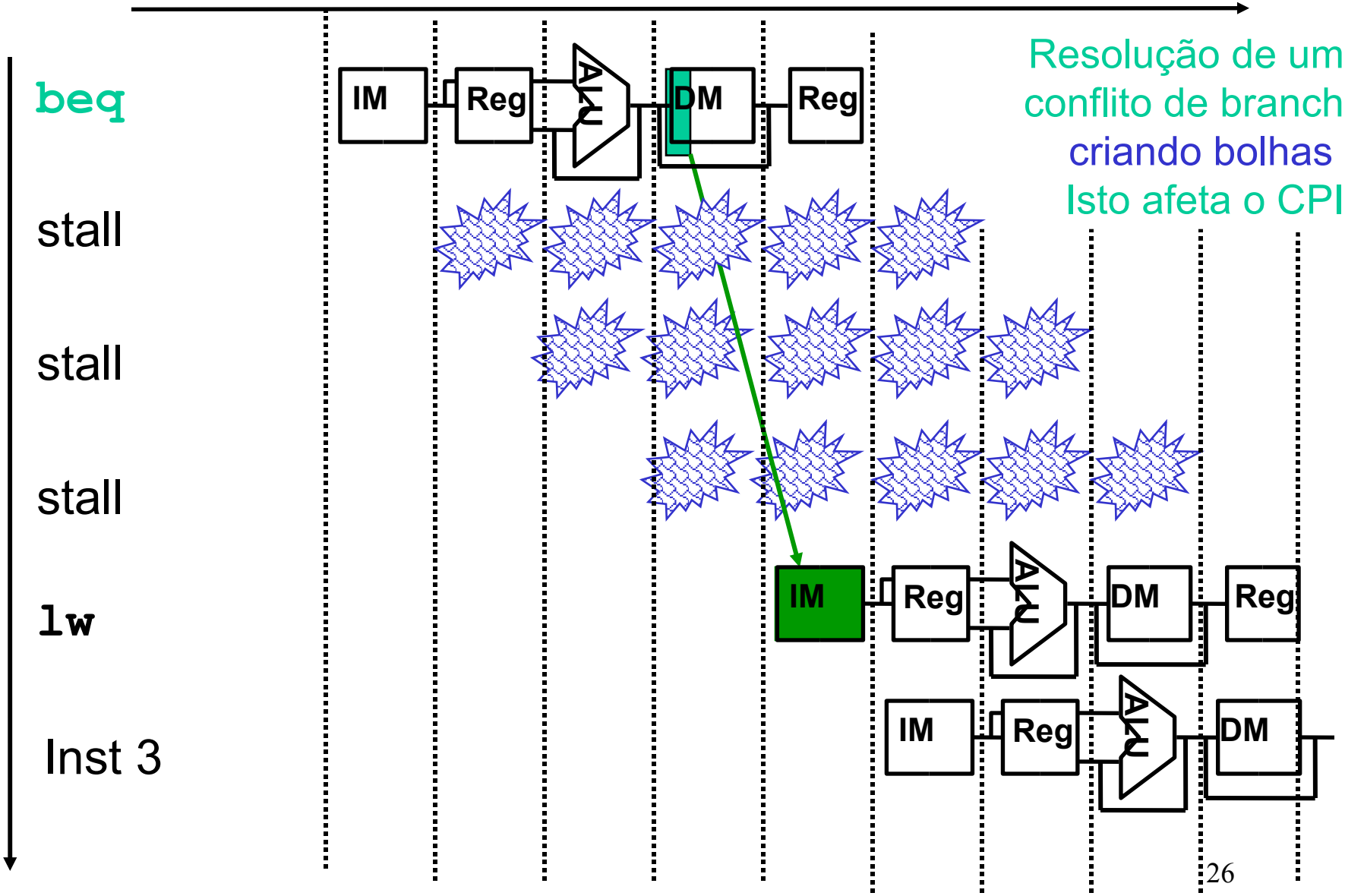
Conflitos de Controle

Intruções de Branch Causam Conflitos de Controle

- Dependencias para trás (backward) no tempo causam conflitos



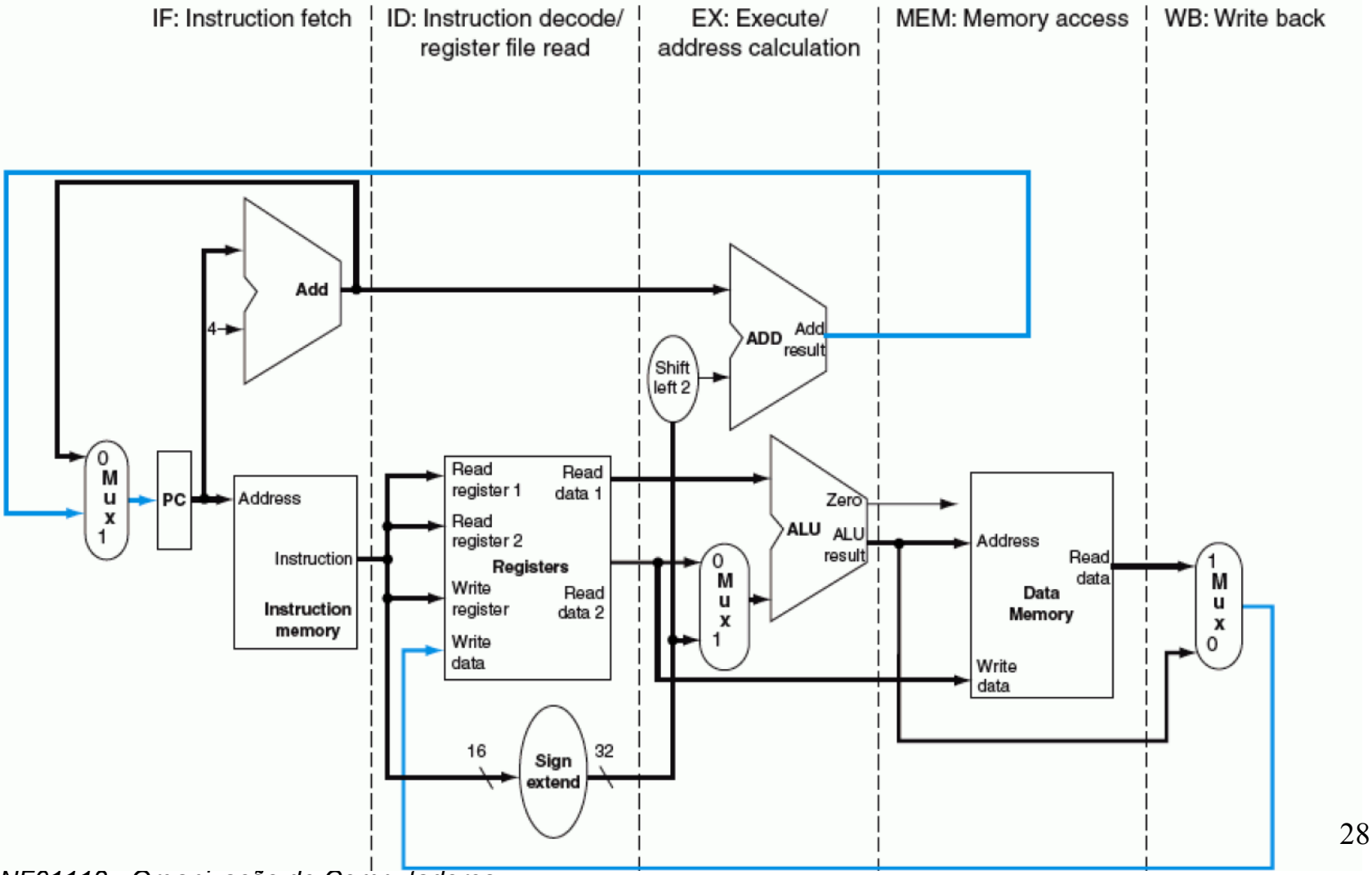
Uma maneira de resolver um Conflito de Controle



Datapath e Controlpath

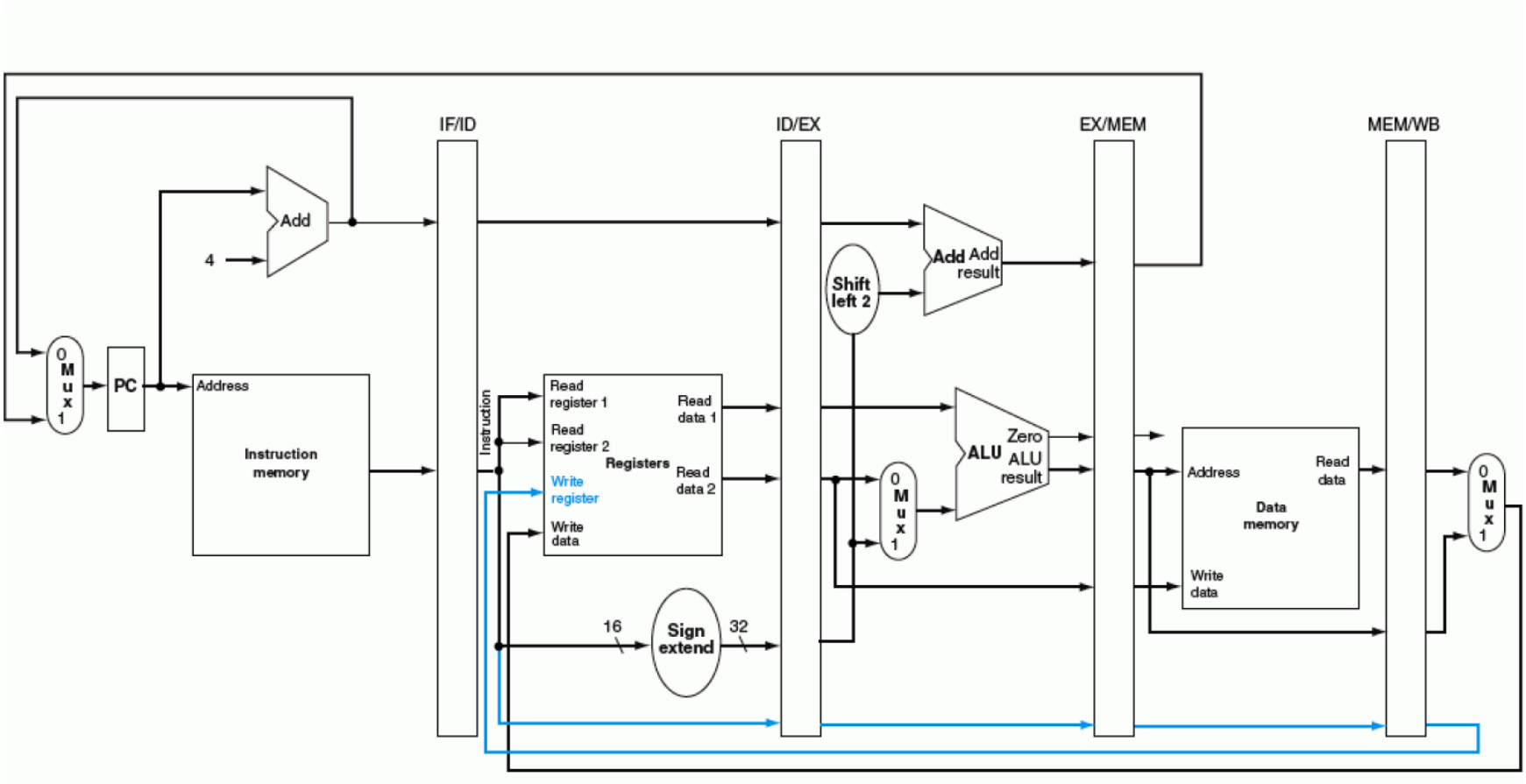
Ajuste no Datapath para Salvar o RegWrite Addr

- **Precisa preservar o endereço do registrador de destino nos Registradores de Estado do Pipeline**



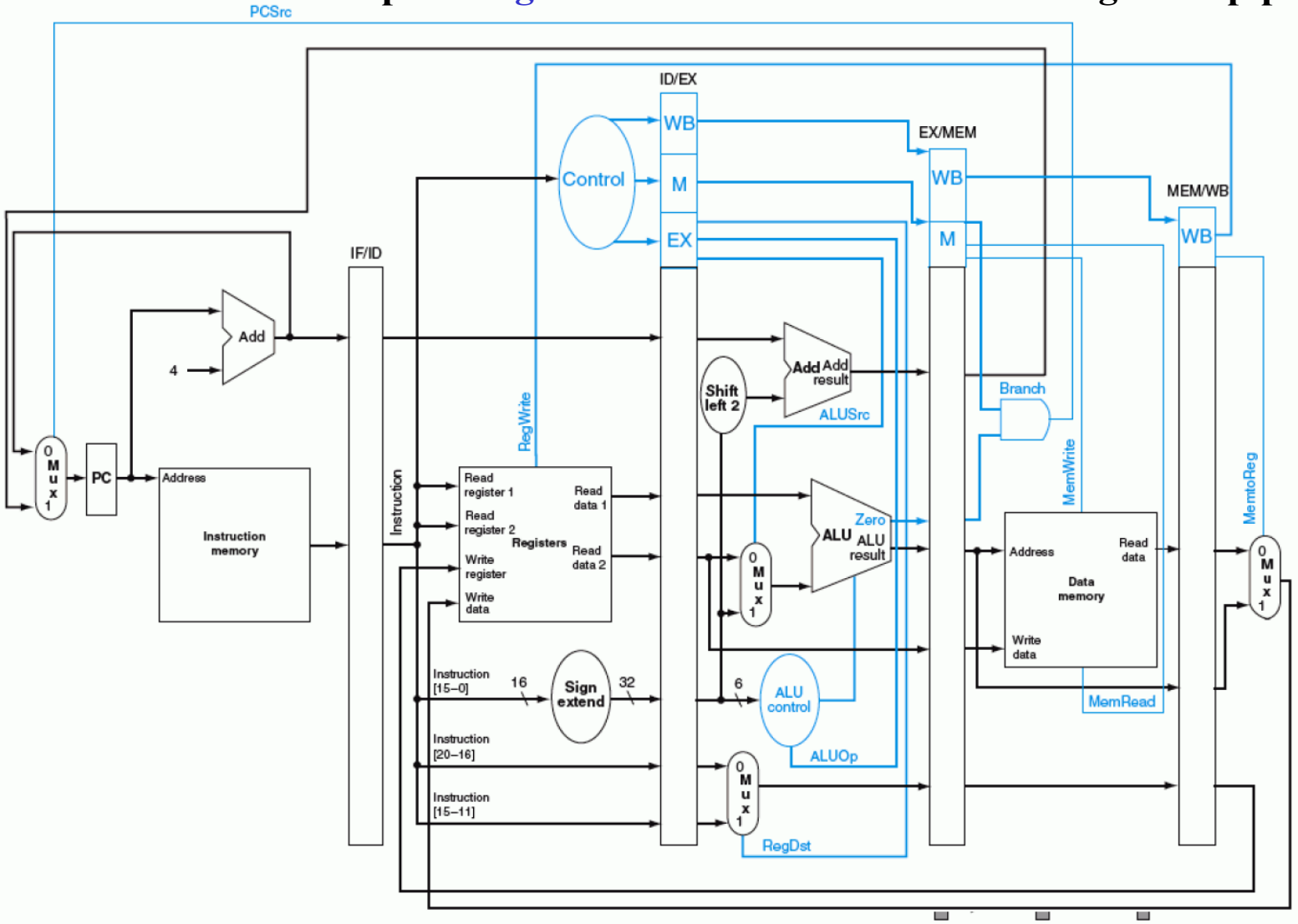
Ajuste no Datapath para Salvar o RegWrite Addr

- **Precisa preservar o endereço do registrador de destino nos Registradores de Estado do Pipeline**



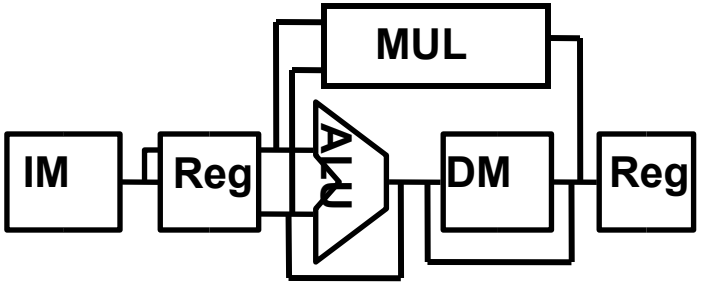
Modificações do Caminho de Controle do MIPS Pipeline

- **Todos sinais de controle são determinados na fase de Decodificação**
  - **São encaminhados pelos registradores de estado entre os estágios do pipeline**

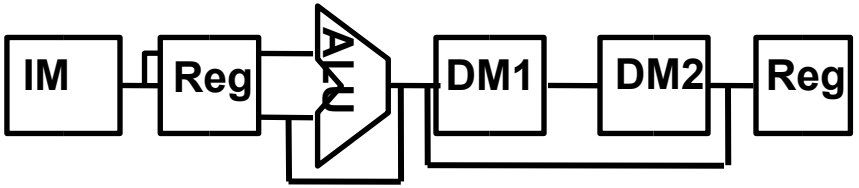


Outras Estruturas de Pipeline Possíveis

- **O que acontece com operações lentas? multiplicação?**
  - **Faz o relógio duas vezes mais lento …**
  - **Ou levar dois ciclos (considerando que não usa o estágio DM)**

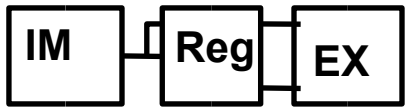


- **O que acontece se o acesso aos dados na memória são duas vezes mais lentos?**
  - **Fazer o relógio duas vezes mais lento ou …**
  - **Fazer o acesso aos dados na memória levar dois ciclos (e manter o mesmo relógio)**



Algumas Alternativas de Pipeline

- **ARM7**

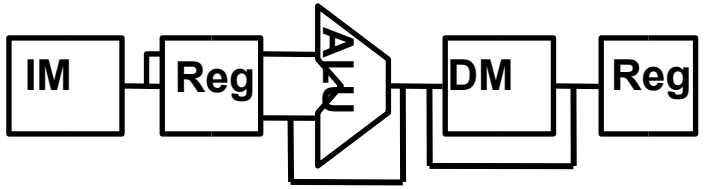


PC update
IM access

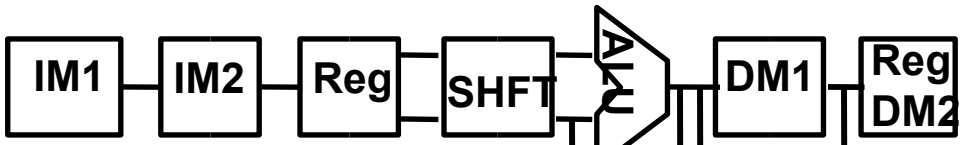
decode
reg
access

ALU op
DM access
shift/rotate
commit result
(write back)

- **StrongARM-1**



- **XScale**



PC update
BTB access
start IM access

IM access

decode
reg 1 access

shift/rotate
reg 2 access

ALU op

start DM access
exception

DM write
reg write

Sumário do Emprego da Técnica Pipeline

- **Todos processadores modernos empregam a técnica pipeline.**

- **Pipeline não ajuda na latência de uma simples tarefa.**
- **Pipeline ajuda na vazão de todo o trabalho e utilização de unidades ociosas.**

  Potencial de aumento do speedup: CPI =1.

  A taxa do Pipeline é limitada pelo estágio do pipeline mais lento.
  - **Estágios de pipeline não balanceados levam a ineficiência.**
  - **O tempo para "encher" o pipeline e o tempo para "esvaziar" podem impactar o aceleramento (speedup) de pipelines profundos com poucas instruções para serem executadas.**

- **É necessário detectar e resolver conflitos:**
  - **Atrasos (stalls) afetam negativamentem o CPI.**
  - **Os atrasos tornam o CPI maior que o valor ideal de 1.**

FIM