Universidade Federal do Rio Grande do Sul
Instituto de Informática

Organização de Computadores

Aula 11

Pipelines
Segunda parte

Pipelines – Segunda Parte

1. Introdução
2. Dependências verdadeiras
3. Dependências falsas
4. Pipeline interlock
5. Forwarding
6. Exemplos e Estudo de Caso

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 (ex. Valor do registrador) para cálculo de endereço efetivo de operando**

- **Tipos de dependências de dados**
  - **Dependência verdadeira**
  - **Antidependência***
  - **Dependência de saída***

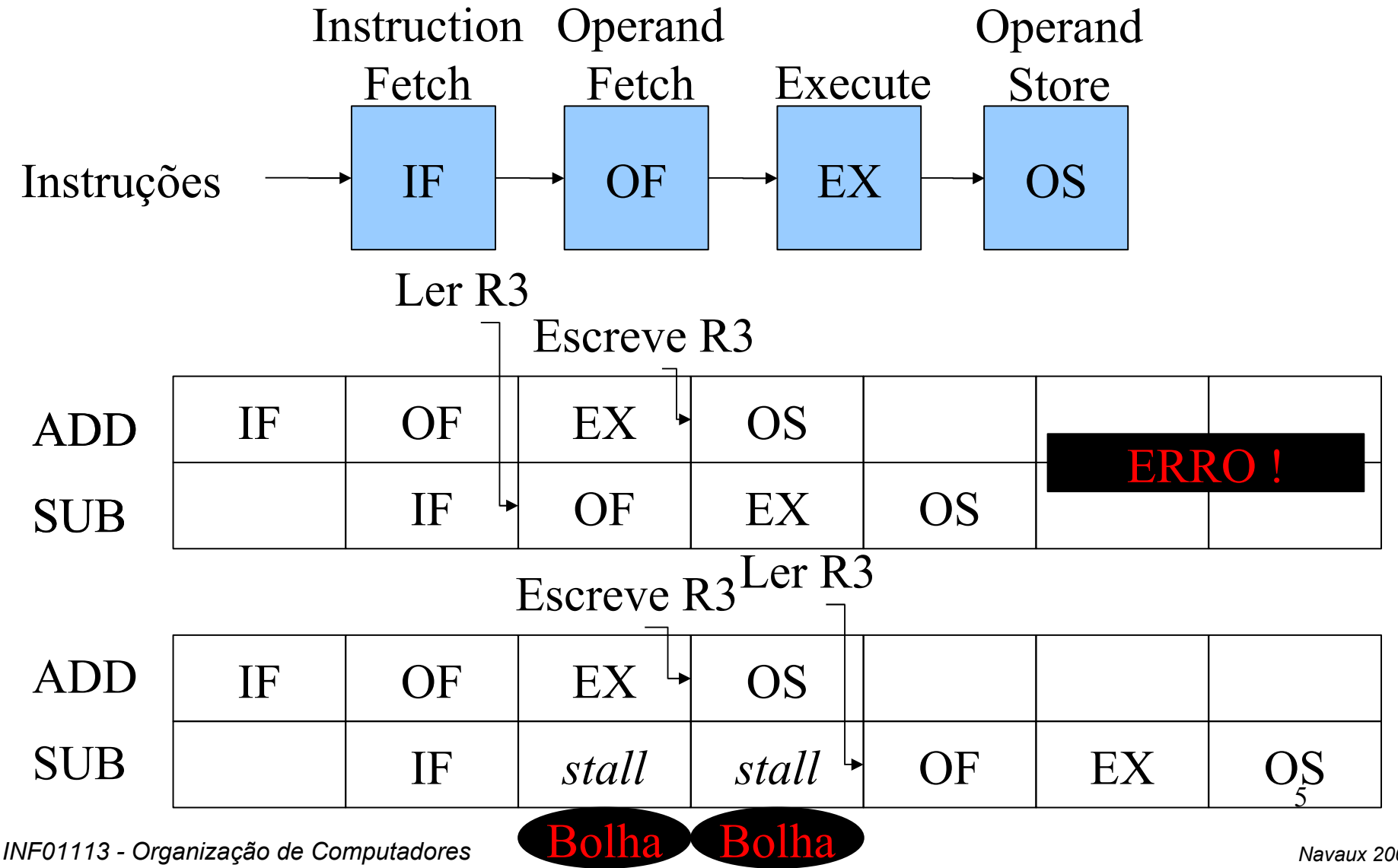
*** Dependências falsas**

Dependências Verdadeiras

- **Dependência Verdadeira**
- **Exemplo**
  - **1. ADD R3, R2, R1 ; R3 = R2 + R1**
  - **2. 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**

- **Problema RAW "Read-after-write"**
- **Pipeline precisa ser parado durante certo número de ciclos**

Dependências verdades



Dependências Falsas

- **Antidependência**
- **Exemplo**
  - **1. ADD R3, R2, R1 ; R3 = R2 + R1**
  - **2. 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**

- **Problema WAR "write-after-read"**
- **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 (próximas aulas)**

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 operando em R3**
- **Instrução 1 tem que escrever seu resultado em R3 antes do que a instrução 3, senão instrução 2 usará valor errado de R3**

- **Problema WAW "write-after-write"**
- **Também só é problema em processadores super-escalares (próximas aulas)**

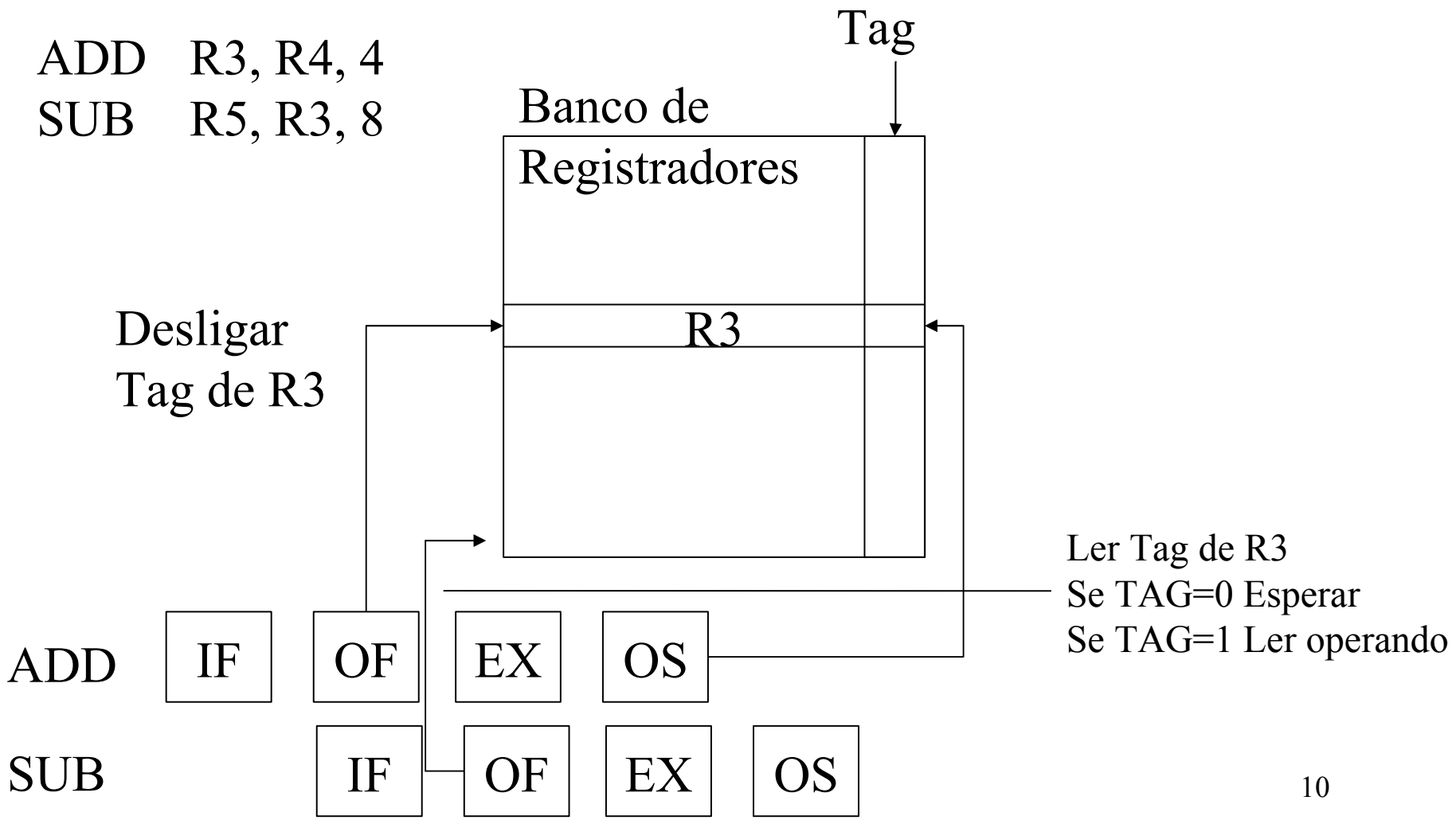
Pipeline Interlock

Como controlar quais registradores estão atualizados e podem ser utilizados em determinado momento ?

Pipeline Interlock

- **Método para manter seqüência correta de leituras e escritas em registradores**
- **Tag de 1 bit é associado a cada registrador**
  - **Tag = 0 indica valor não válido Tag = 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**

Pipeline Interlock

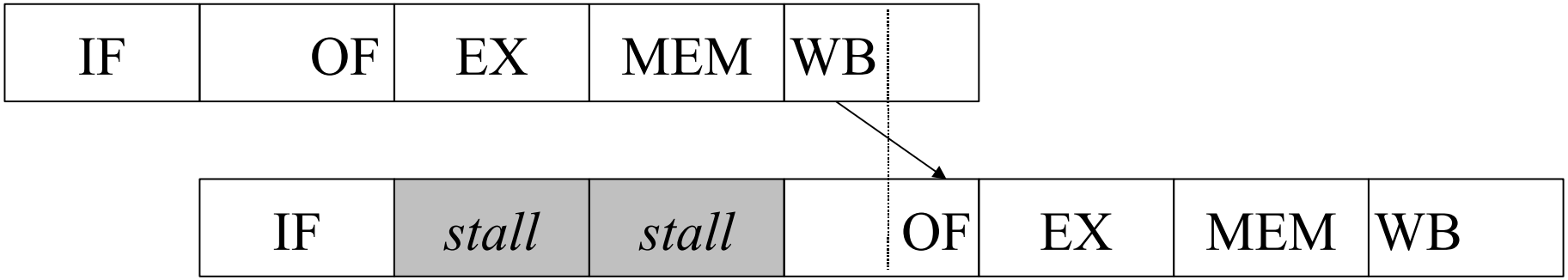


Forwarding

Como podemos adiantar os valores já calculados, para que as próximas instruções não fiquem esperando longos períodos ?

Forwarding

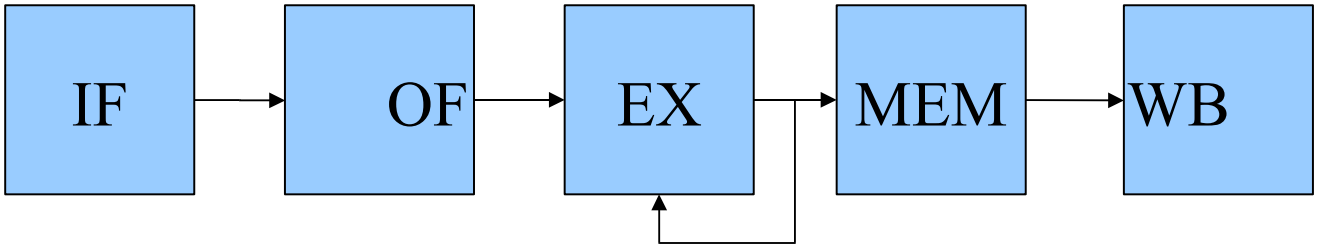
- **Exemplo 1**
  - **ADD R3, R2, R0**
  - **SUB R4, R3, 8**
- **Instrução SUB precisa do valor de R3, calculado pela instrução ADD, ou seja:**
  - **Valor é escrito em R3 por ADD no último estágio WB (write-back)**
  - **Valor é necessário em SUB no terceiro estágio**
  - **Instrução SUB ficará presa por 2 ciclos no 2º estágio do pipeline**

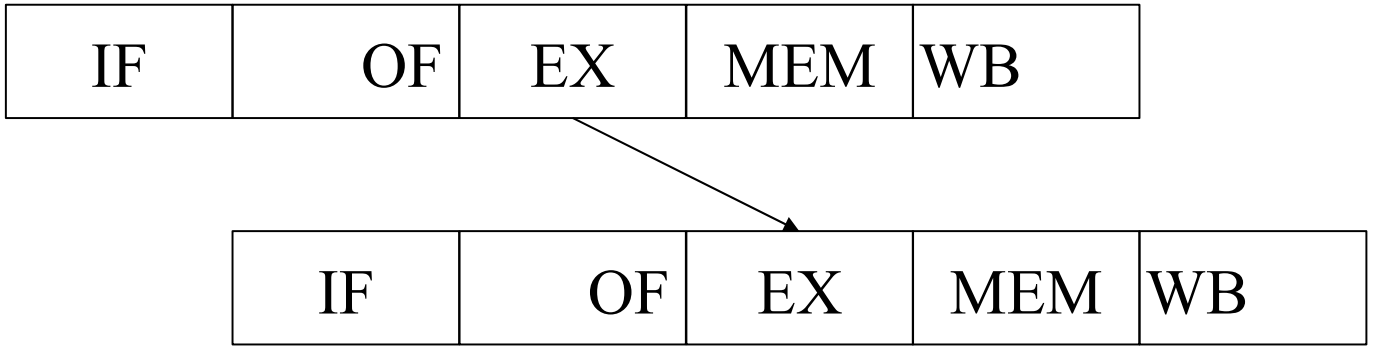


Supõe-se escrita no banco de registradores na primeira metade do ciclo e leitura na segunda metade

Forwarding

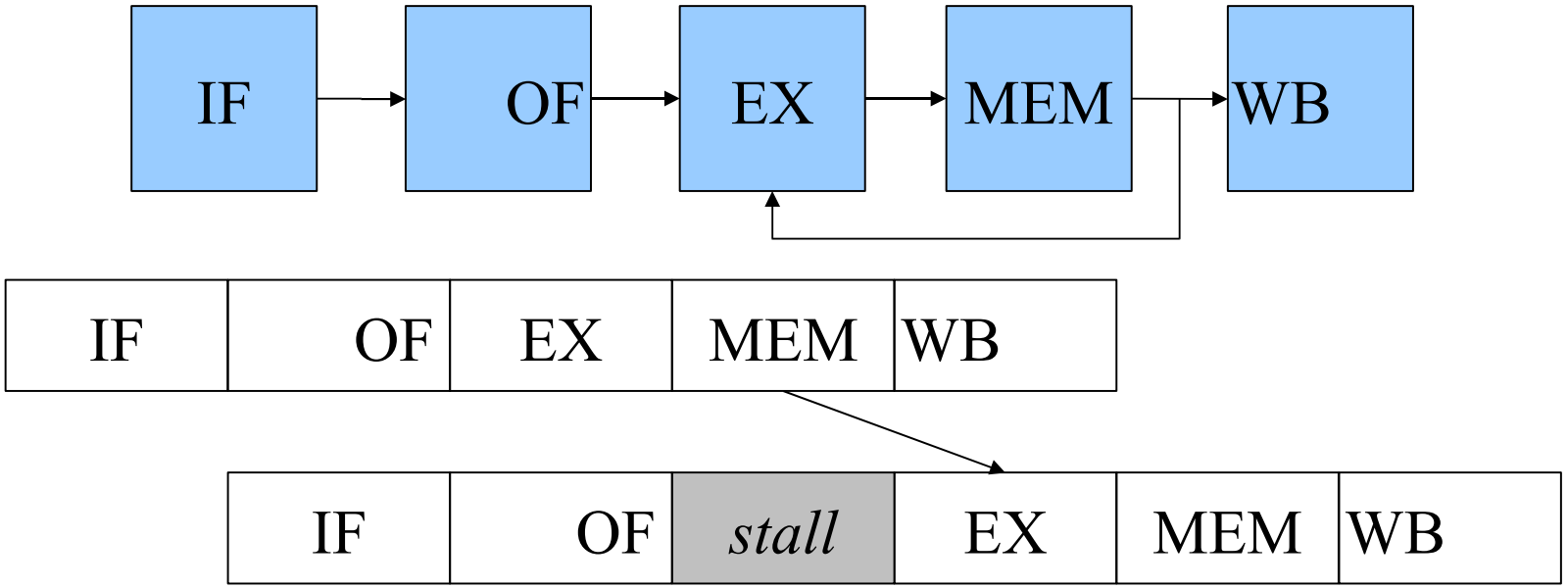
- **Caminho interno dentro do pipeline entre a saída da ALU e a entrada da ALU**
  - **Evita *stall (bolhas)* do pipeline**





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**



Exemplos e Estudo de Caso

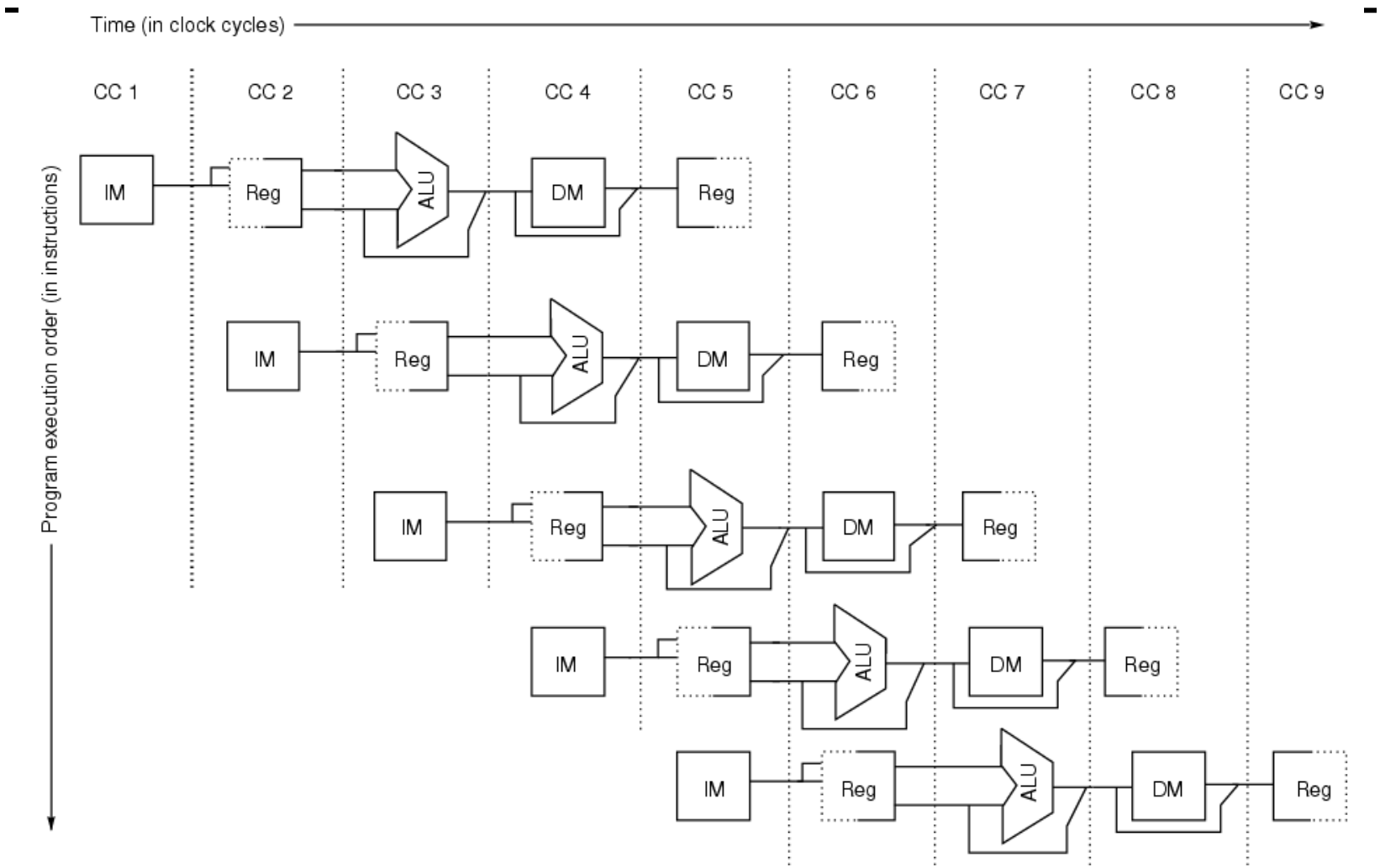
Como esse recurso de forwarding é usado na prática ?

Estudo de Caso:
Pipeline do simulador DLX (MIPS)

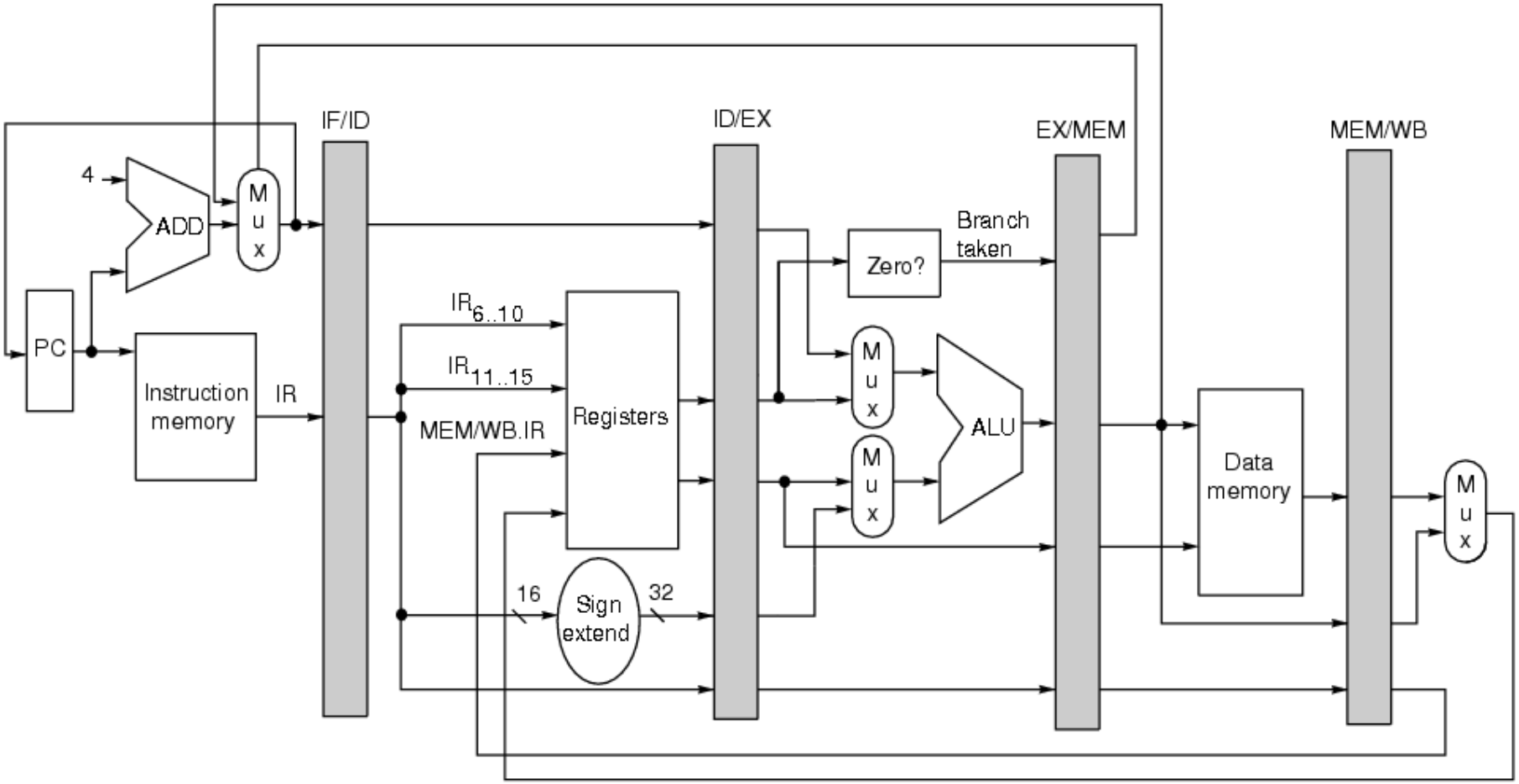
Instrução	Clock 1	2	3	4	5	6	7	8	9
i	IF	ID	EX	MEM	WB				
i + 1		IF	ID	EX	MEM	WB			
i + 2			IF	ID	EX	MEM	WB		
i + 3				IF	ID	EX	MEM	WB	
i + 4					IF	ID	EX	MEM	WB

- **IF – Instruction Fetch**
- **ID – Instruction Decode**
- **EX - Execution**
- **MEM – Memory Access**
- **WB – Write Back**

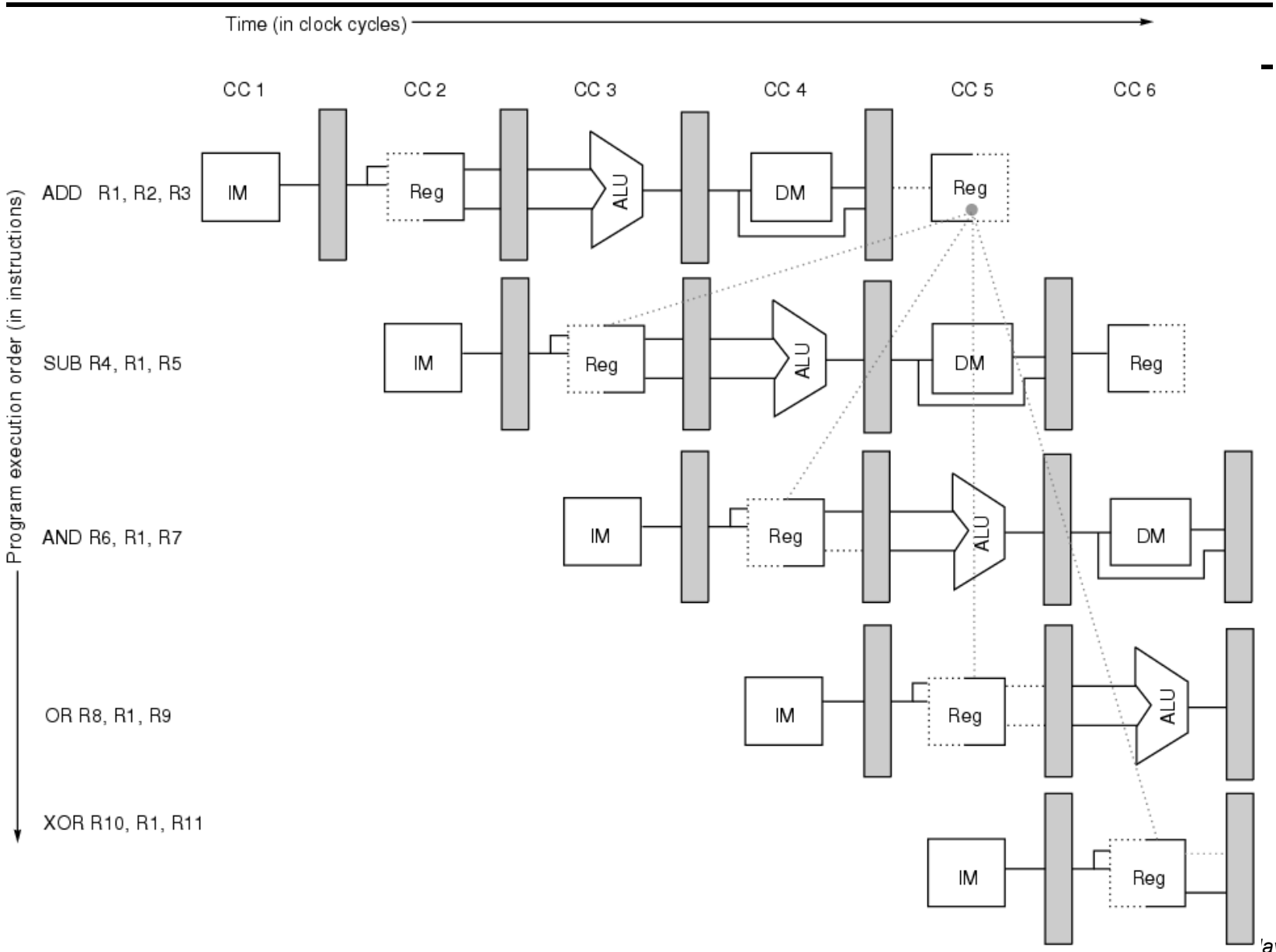
Pipeline do DLX



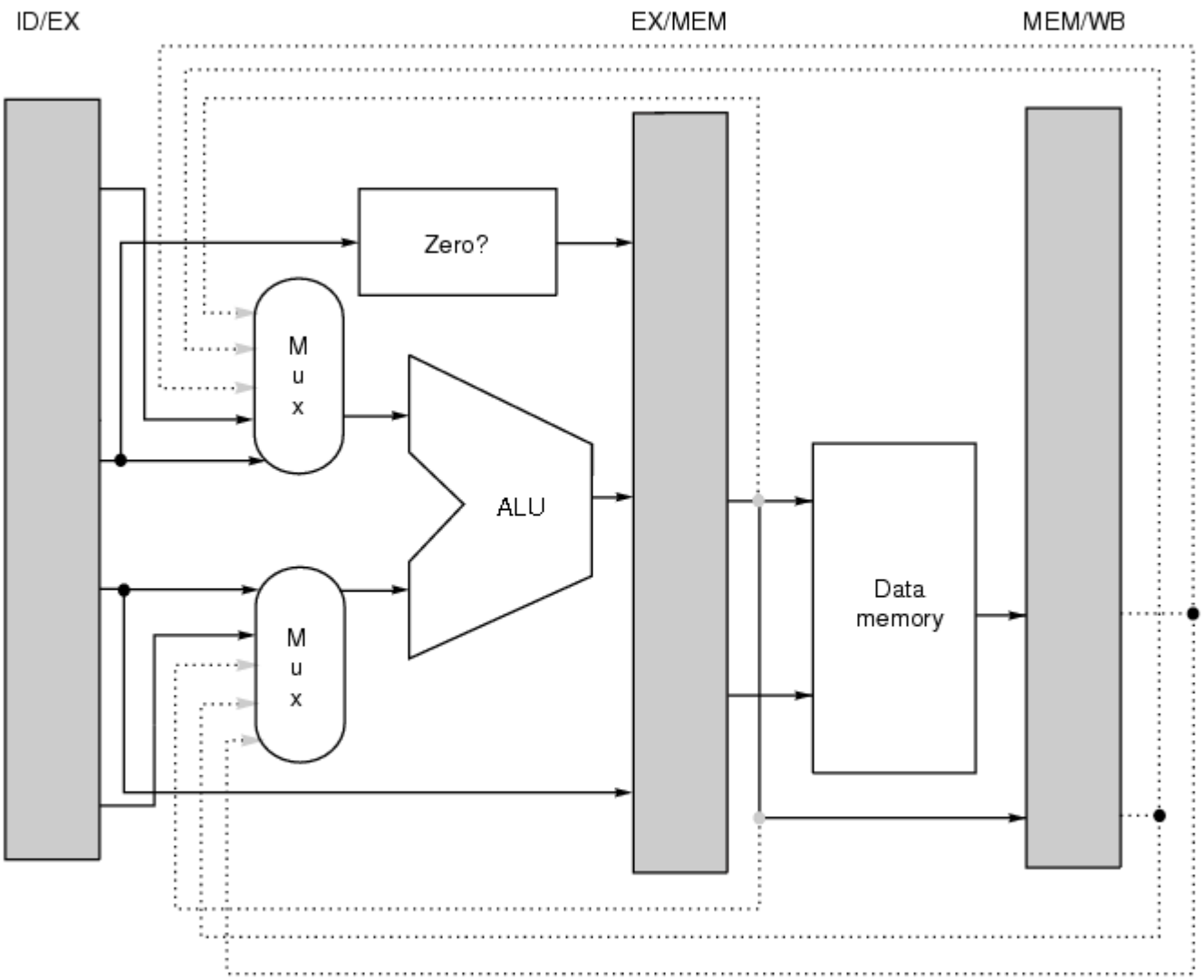
Pipeline do DLX



2. Dependências no Pipeline

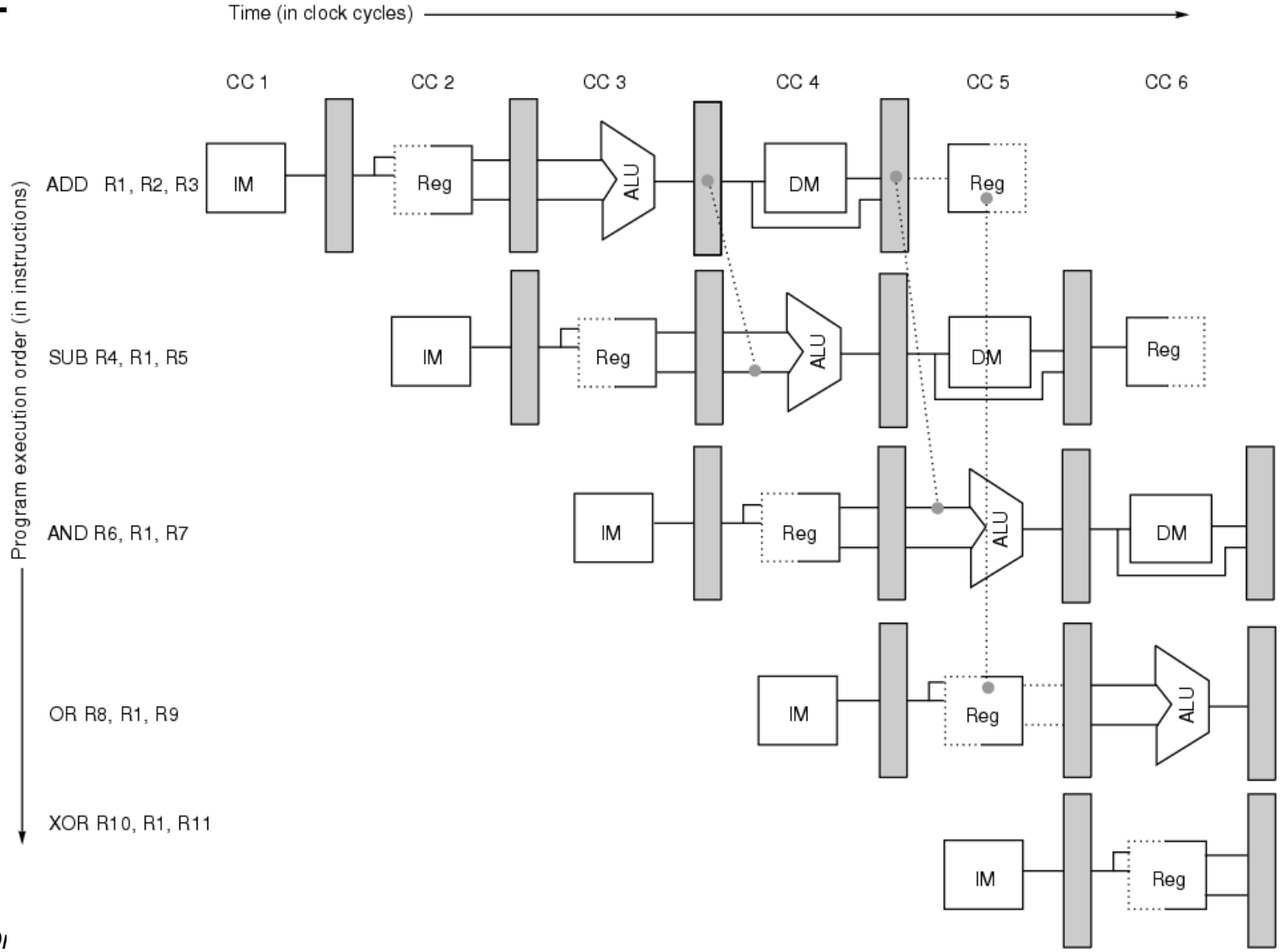


3. Forwarding



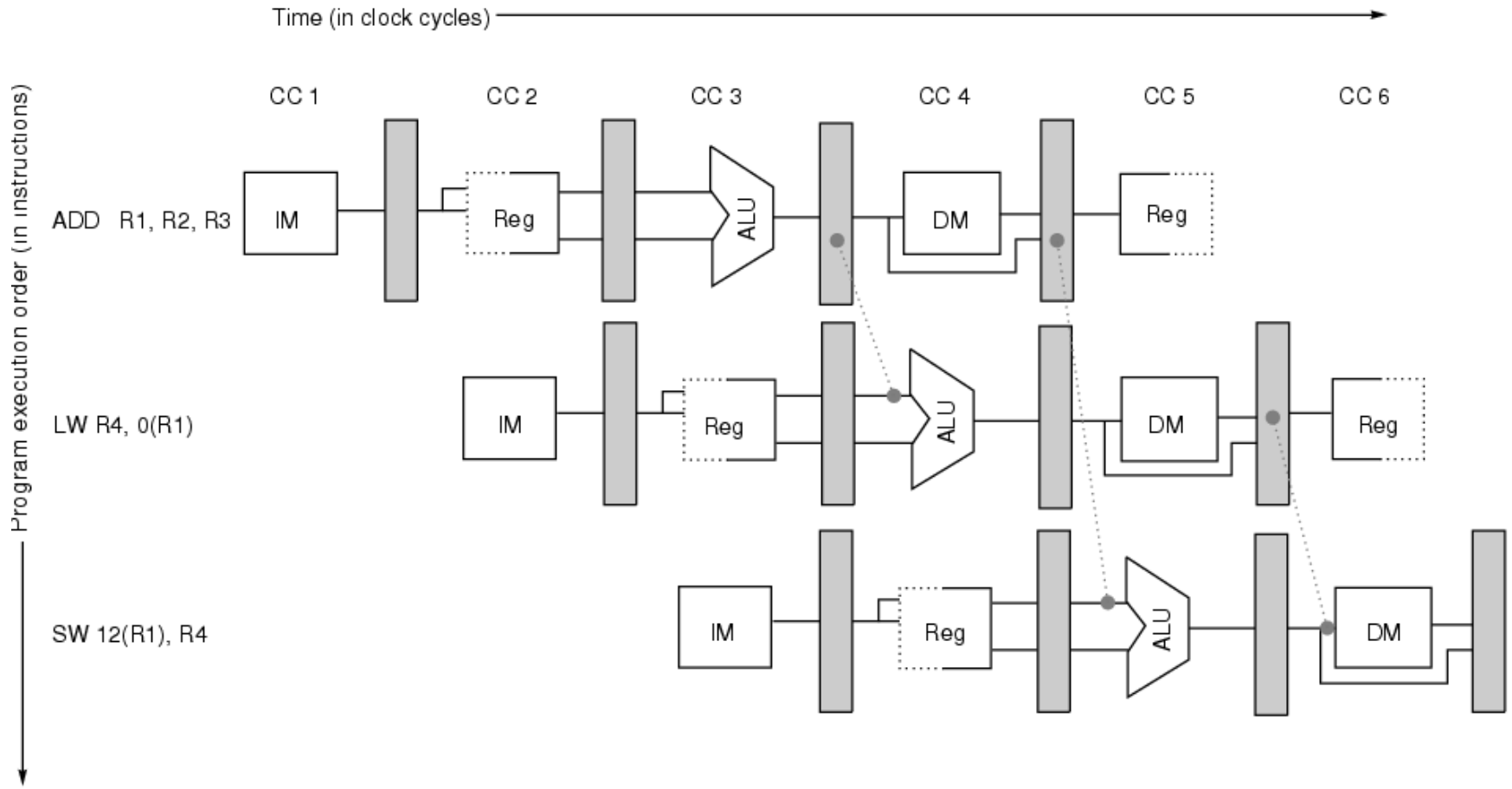
Forwarding

Conflito Lógico-Aritmético x Lógico-Aritmético

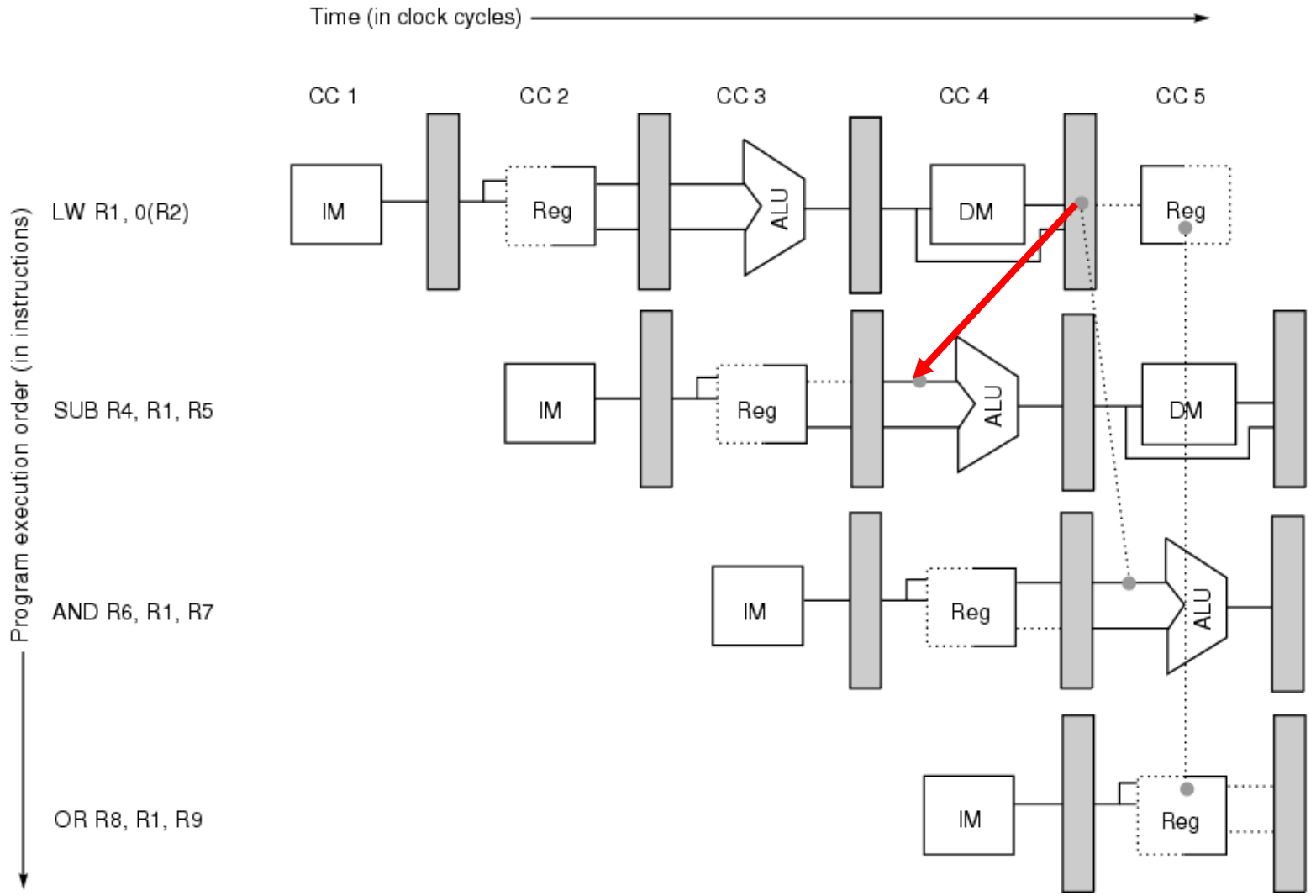


Forwarding

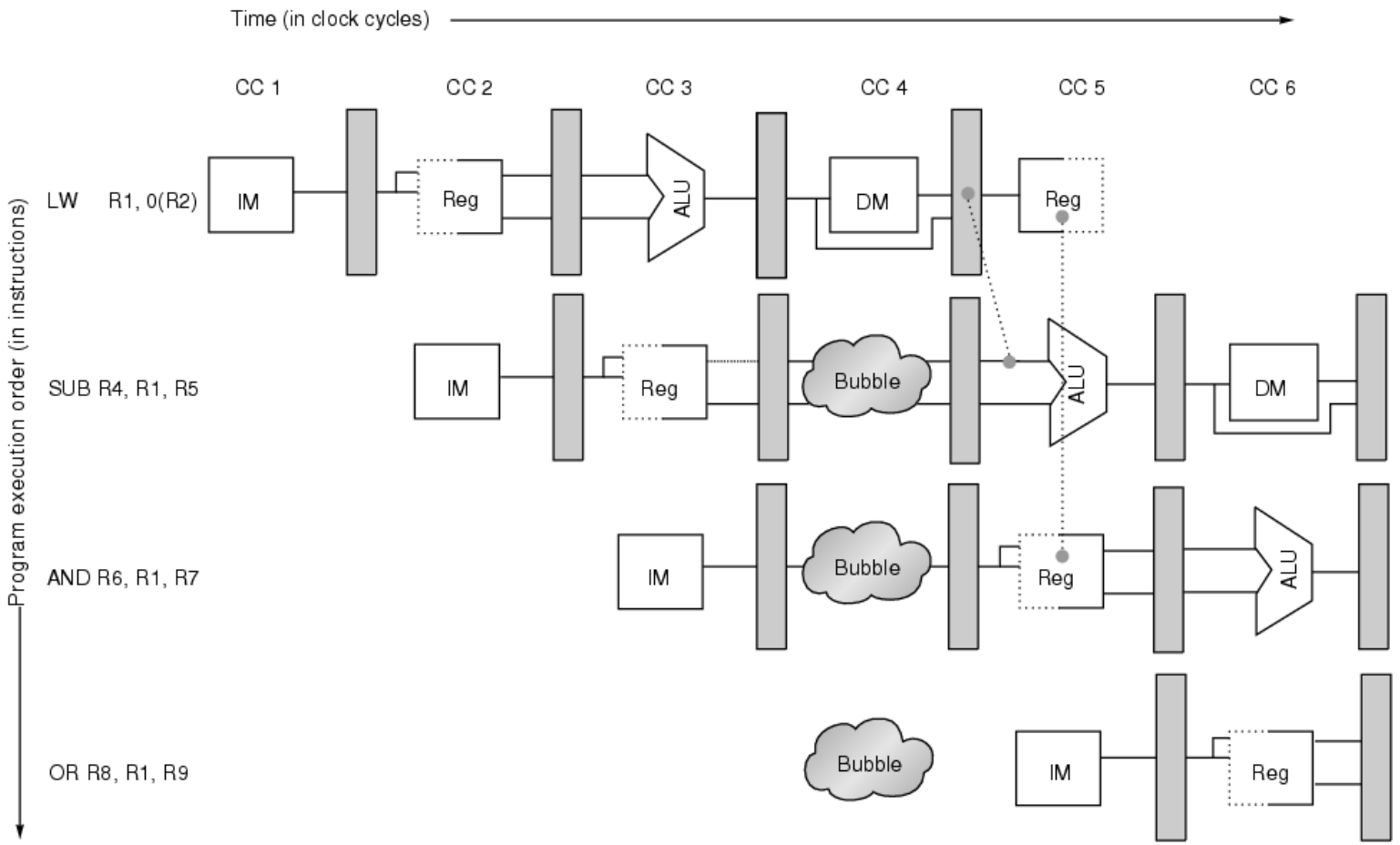
Conflito Lógico-Aritmético x Memória



Forwarding: Qual o problema?

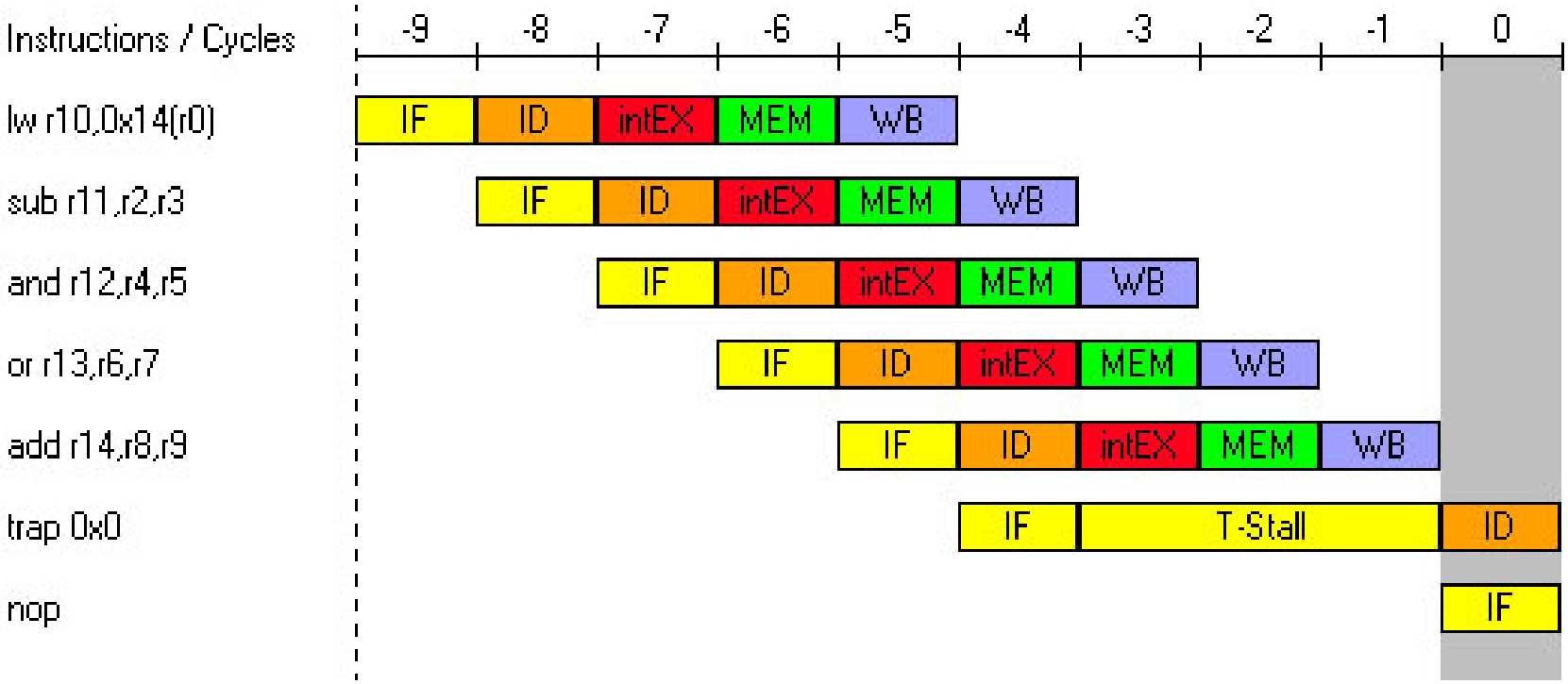


Forwarding: Solução



4. Exemplos

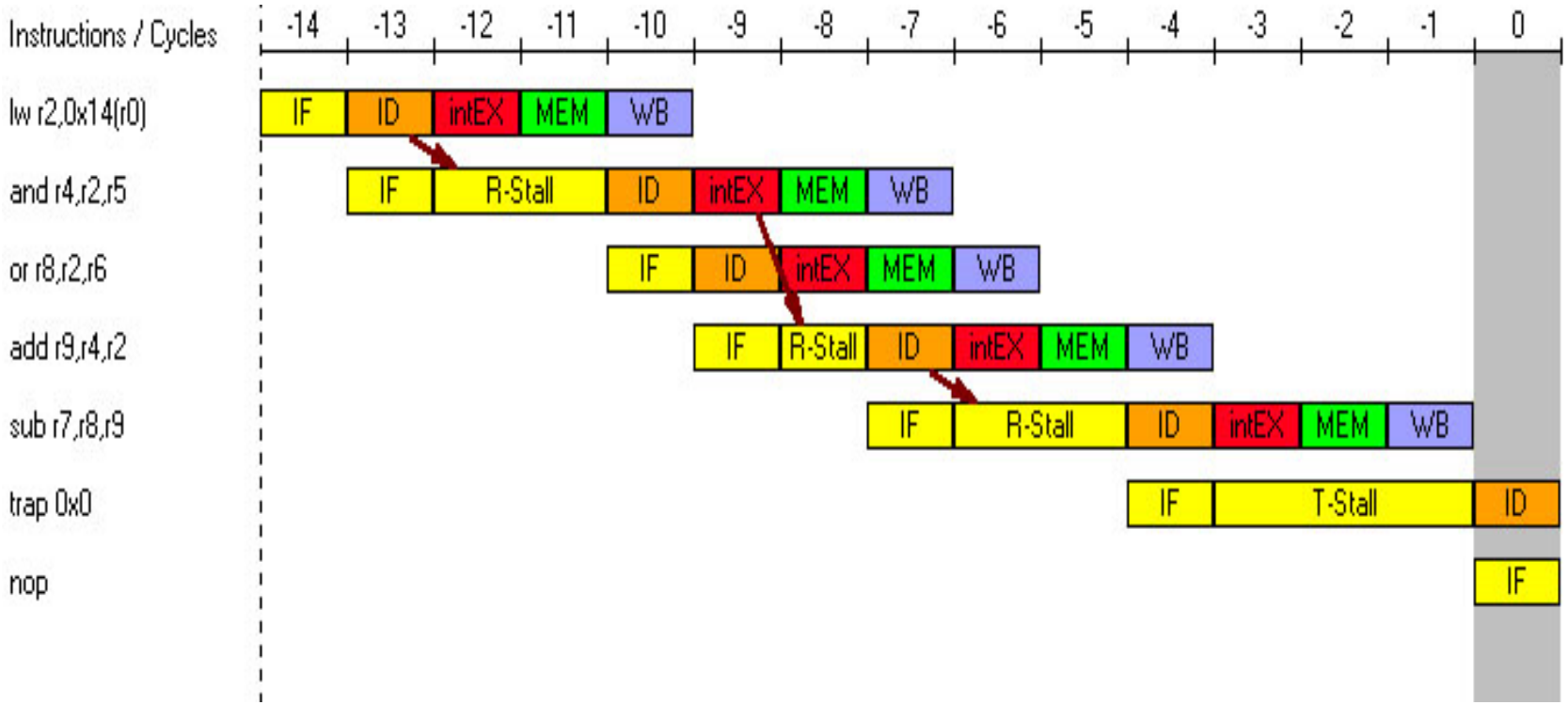
Exemplo 1:Não Existem dependências



Exemplos

Exemplo 2:Forwarding desabilitado.

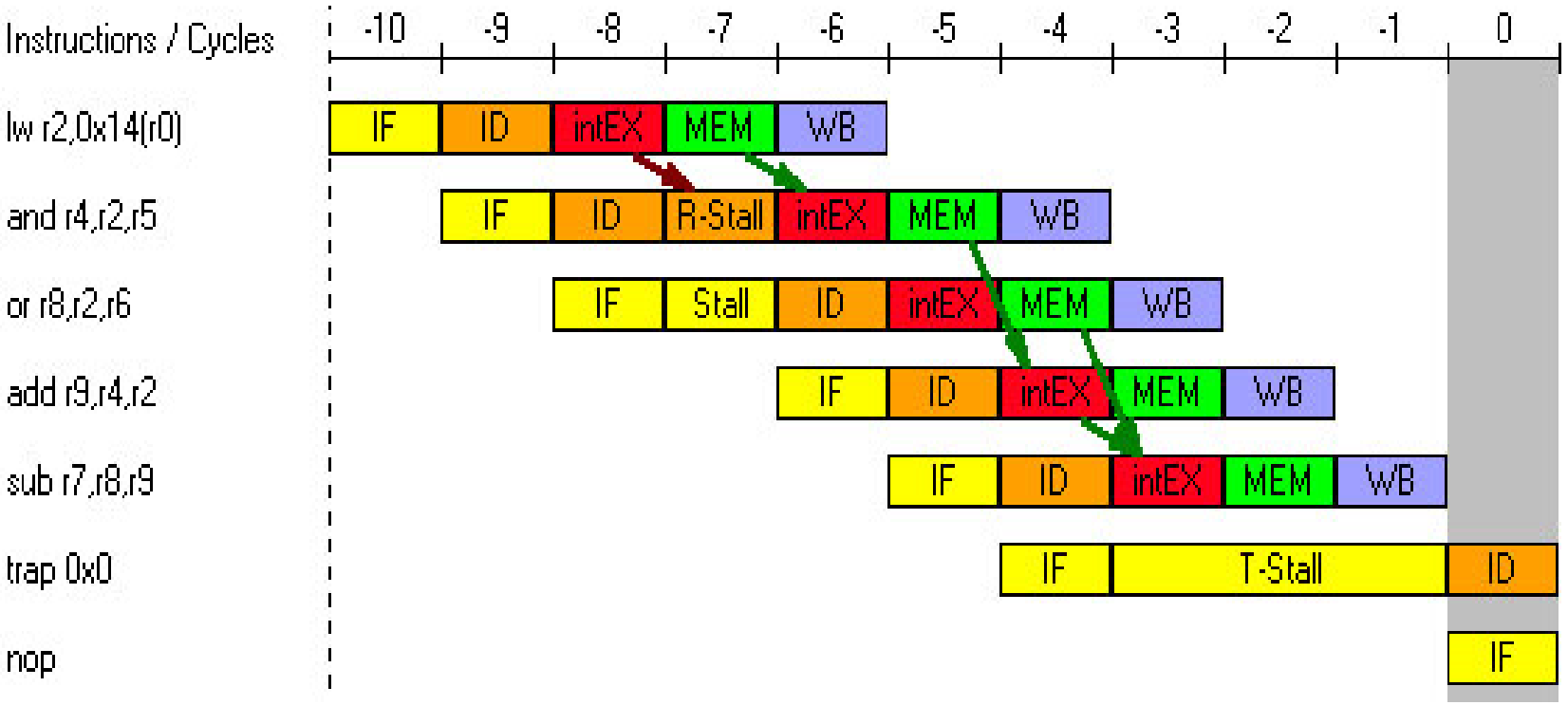
Pergunta: Como ficaria se tivessemos os caminhos de forwarding ?



Exemplos

- **Exemplo 2: Forwarding habilitado.**
- **Com relação Forwarding desabilitado, Speedup = 14/10 = 1.4**

Considerando o clock e os ciclos iguais, qual o speedup em relação ao multi-ciclo ?



Exemplos - Questões

Baseado no exemplo 2:

N=5 (Instruções)
S=5 (Estágios)

Multi-ciclos: ??? ciclos

Pipeline: 14 ciclos **Speedup = ???**

Pipeline c/ forwarding: 10 ciclos **Speedup = ???**

Pipeline ideal: ??? ciclos **Speedup = ???**

Exemplos - Respostas

Baseado no exemplo 2:

N=5 (Instruções)
S=5 (Estágios)

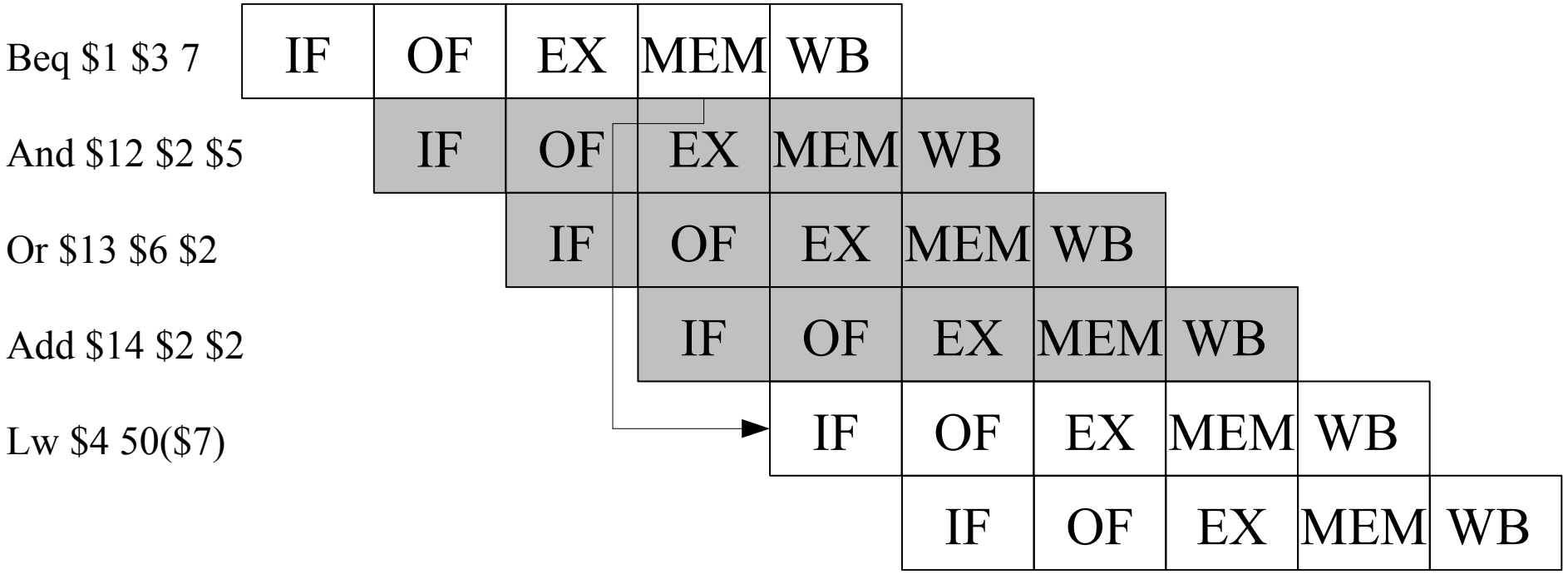
Multi-ciclos: N * S = 5 * 5 = 25 ciclos

Pipeline: 14 ciclos **Speedup = 25 / 14 = 1,78**

Pipeline c/ forwarding: 10 ciclos **Speedup = 25 / 10 = 2,50**

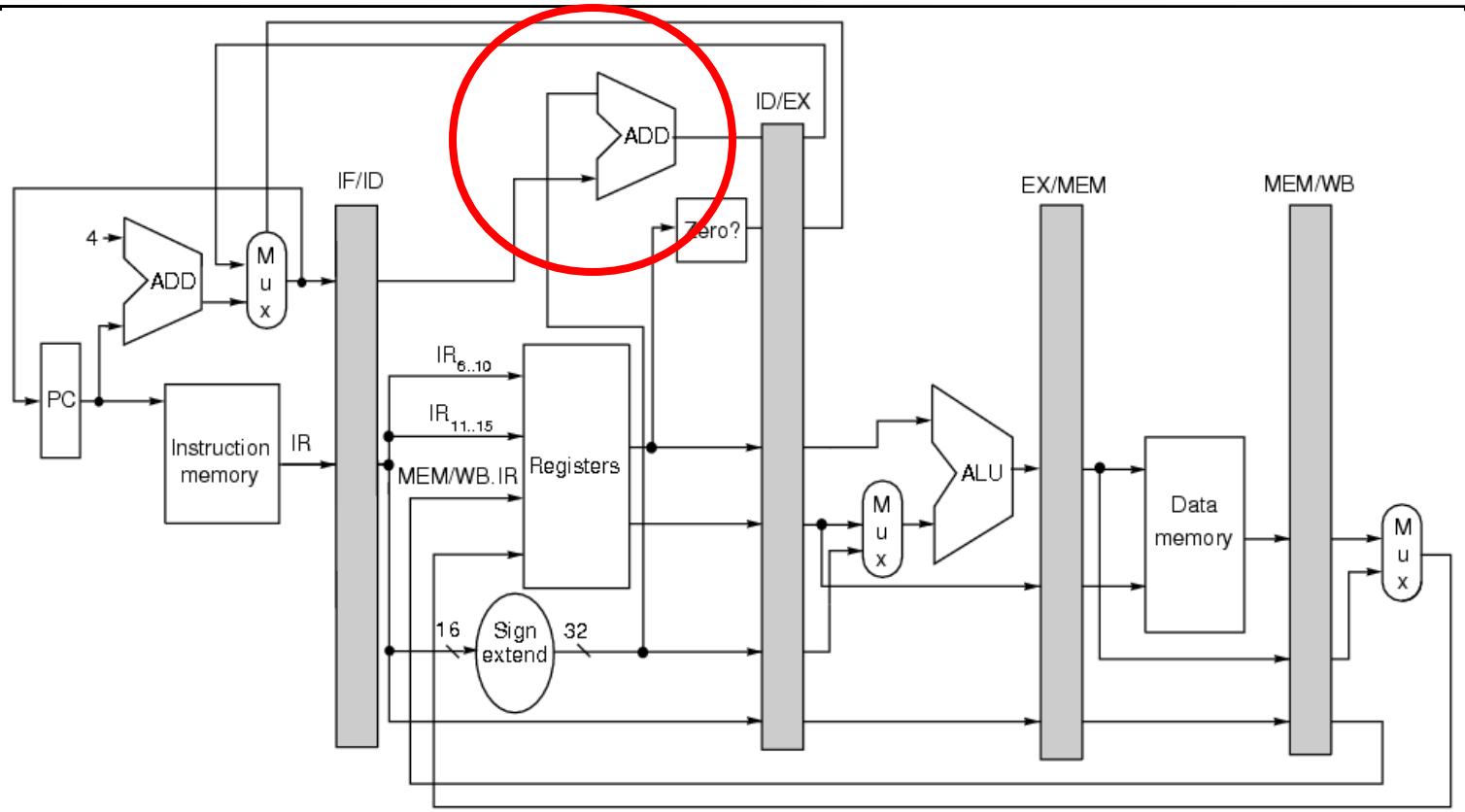
Pipeline ideal: N + (S-1) = 5 + 4 = 9 **Speedup = 25 / 9 = 2,77**

5. Instruções de Desvio no Pipeline



- **Uma instrução de desvio causa um atraso de dois ciclos no pipeline sem forwarding.**
- **Isso ocorre pois o sinal de desvio acontece no estágio da memória !**

Instruções de Desvio no Pipeline



- Os atrasos devido às instruções de desvio podem ser reduzidos
- Através da antecipação do cálculo do endereço alvo do desvio e do teste "Zero?" para a etapa de decodificação do pipeline (ID).
- Qual o custo?

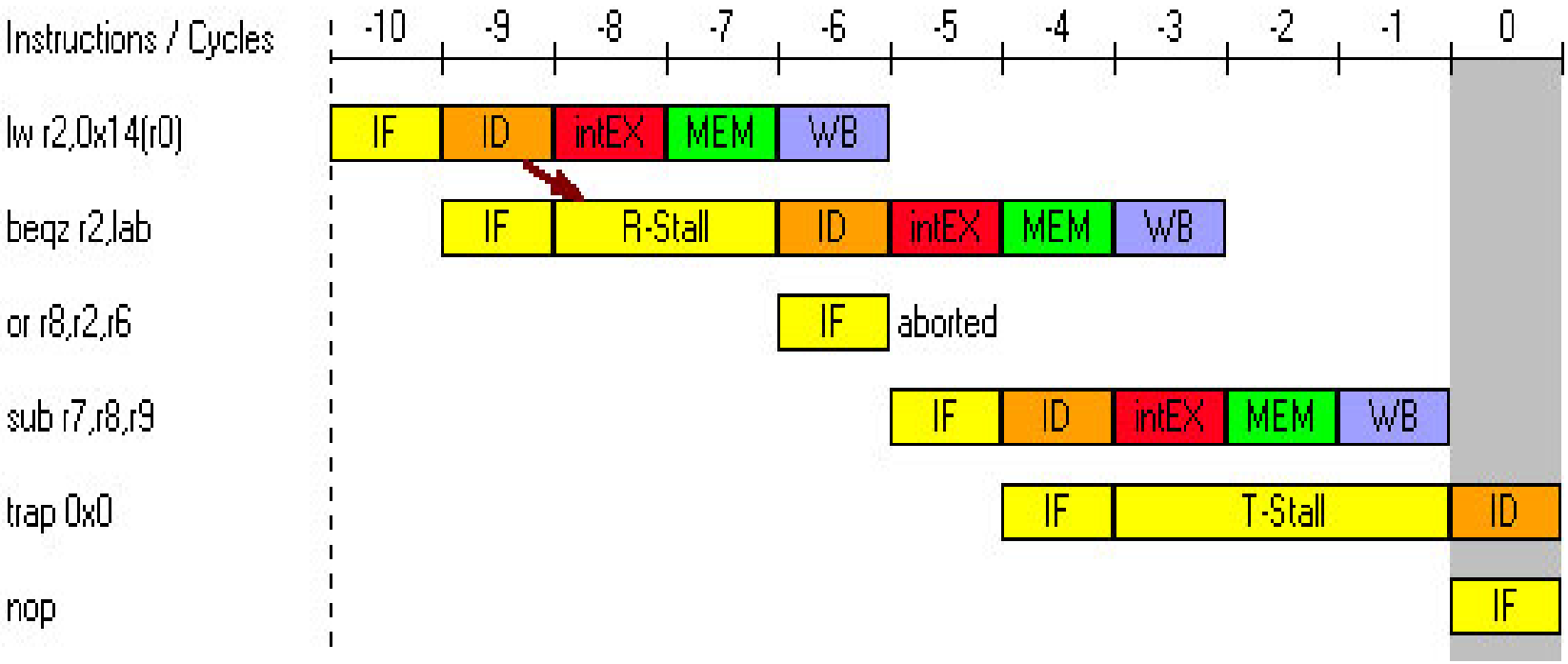
Exemplos

Exemplo 3: **Endereço=20(r0) valor=0**

```
        lw   r2, 20(r0)
        beqz r2, lab
        or   r8, r2, r6
        and  r13, r6, r2
        add  r9, r4, r2
lab:    sub  r7, r8, r9
```

Exemplos

- **Exemplo 3:**



END