



**Universidade Federal de Pelotas**

**Instituto de Física e Matemática**

**Departamento de Informática**

**Bacharelado em Ciência da Computação**

# **Arquitetura e Organização de Computadores II**

## **Aula 11**

**2. MIPS pipeline: conflitos por dados e paradas,  
conflitos em desvios condicionais.**

**Prof. José Luís Güntzel**

**[guntzel@ufpel.edu.br](mailto:guntzel@ufpel.edu.br)**

**[www.ufpel.edu.br/~guntzel/AOC2/AOC2.html](http://www.ufpel.edu.br/~guntzel/AOC2/AOC2.html)**

## 2. Organizações do MIPS: pipeline

### ► Conflitos por Dados e Paradas

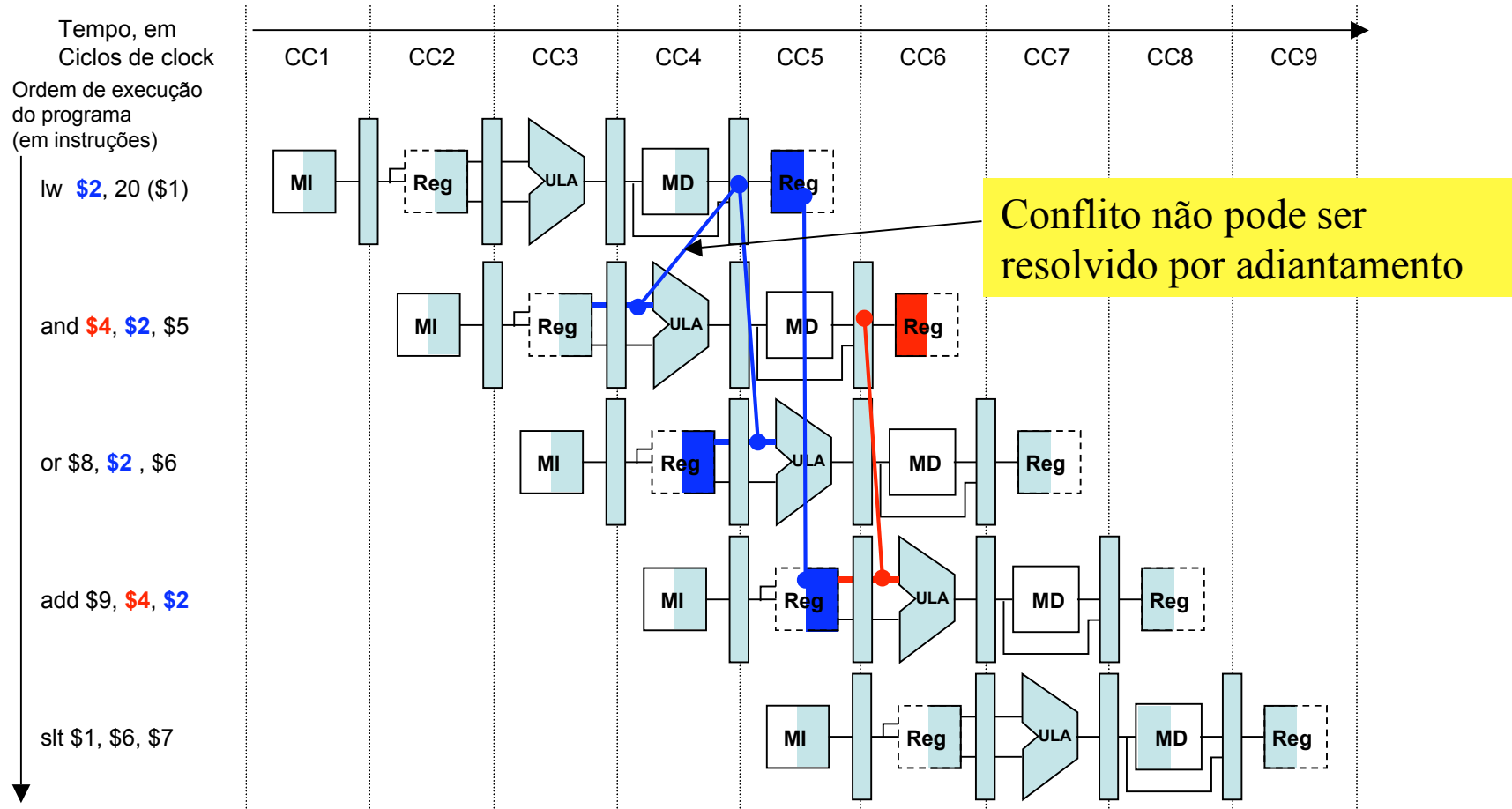
Nem sempre o adiantamento irá resolver um conflito por dados.

**Exemplo:**

lw	\$2, 20(\$1)	# registrador \$2 é escrito
and	\$4, \$2, \$5	# primeiro operando (\$2) depende de lw; registrador \$4 é escrito
or	\$8, \$2, \$6	# primeiro operando (\$2) depende de lw
add	\$9, \$4, \$2	# primeiro operando (\$4) depende de and; segundo operando (\$2) depende de lw
slt	\$1, \$6, \$7	# nenhuma dependencia

## 2. Organizações do MIPS: pipeline

### ► Conflitos por Dados e Paradas



## 2. Organizações do MIPS: pipeline

### ► Conflitos por Dados e Paradas

#### Unidade de Detecção de Conflito

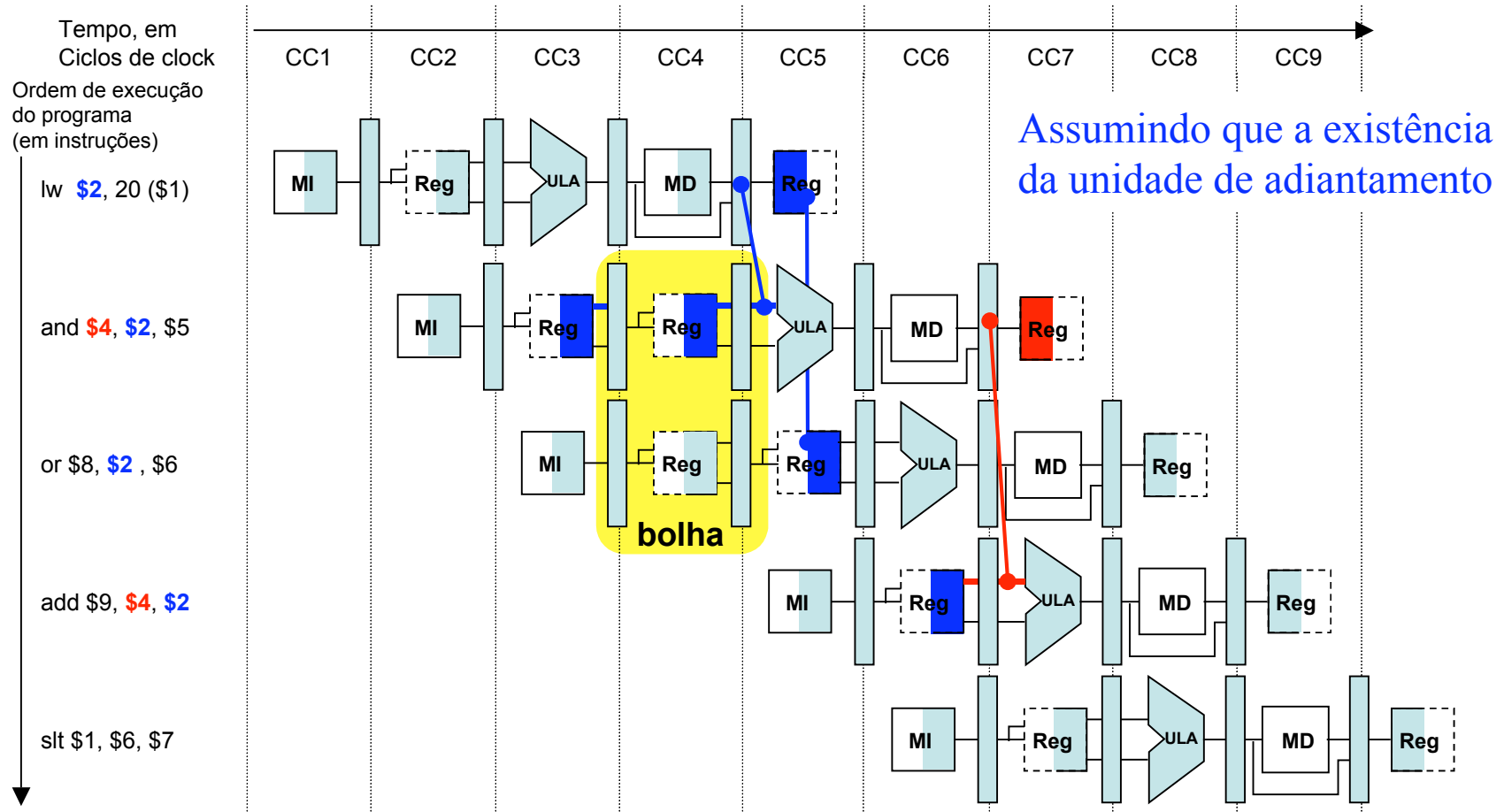
- ❑ Faz o pipeline parar quando houver uma **instrução load word**, seguida de uma instrução que **leia o registrador onde esta instrução de load word escreveu**
- ❑ Vai operar durante o estágio DI, inserindo uma parada entre a instrução load word e o uso de seu resultado
- ❑ **Condição a ser verificada:**

load word é a única instrução que lê dados da memória

**Se** (DI/EX.LerMem = 1 **E**  
((DI/EX.RegistradorRt = BI/DI.RegistradorRs ) **OU**  
(DI/EX.RegistradorRt = BI/DI.RegistradorRt )))  
**Então** pára o pipeline por um ciclo de relógio

## 2. Organizações do MIPS: pipeline

### ► Conflitos por Dados e Paradas



## 2. Organizações do MIPS: pipeline

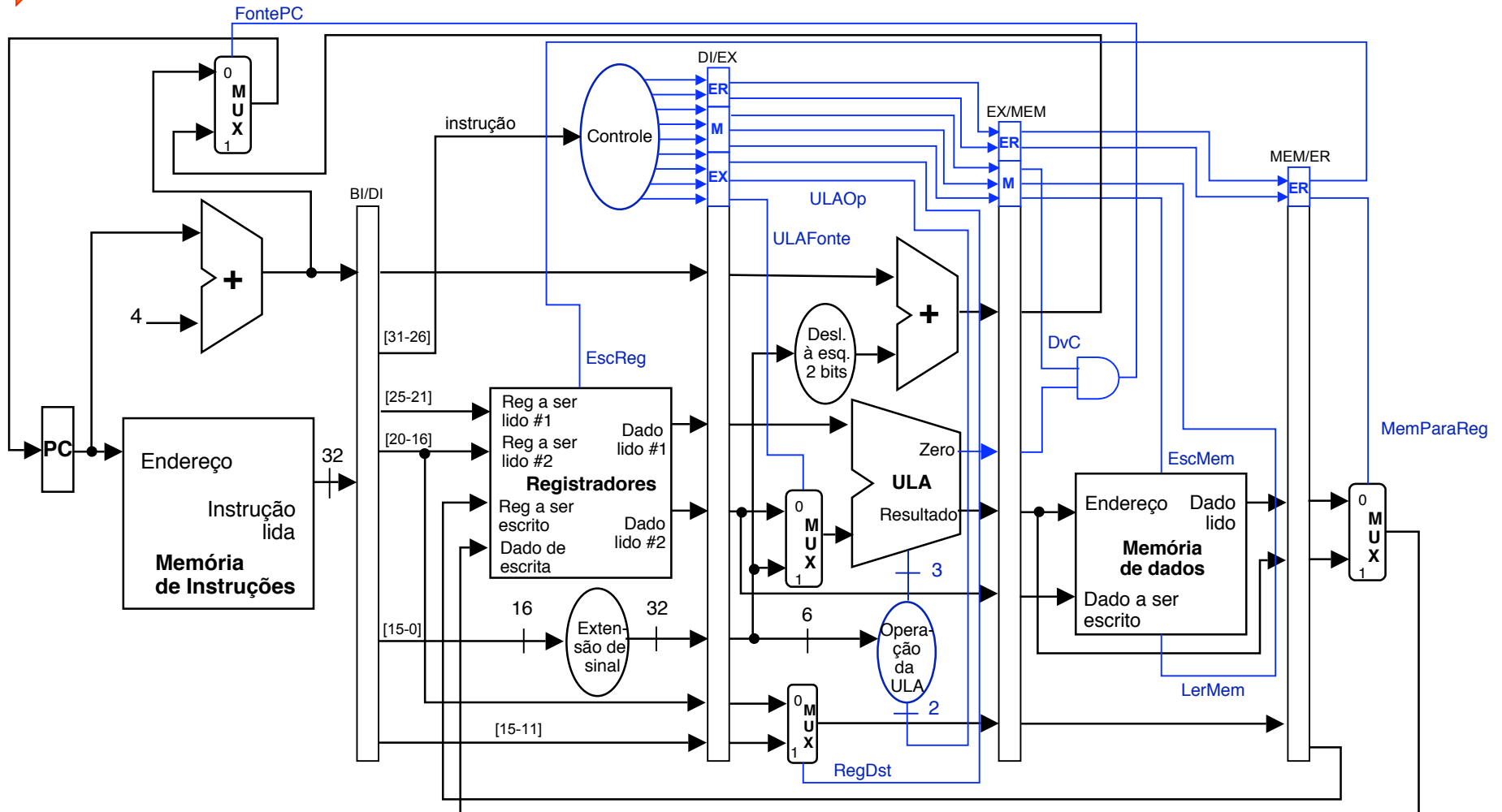
### ► Conflitos por Dados e Paradas

**Trancando o Prosseguimento das Instruções Posteriores a uma Instrução “load word”**

- Se a instrução que está no estágio DI estiver parada, então o estágio BI também precisa parar
- Para impedir o avanço de instruções pelo pipeline, **basta evitar que tanto o PC quanto o registrador BI/DI sejam escritos**
- As condições do item anterior fazem com que, no ciclo de relógio seguinte:
  - A instrução que está no BI seja lida novamente
  - Os registradores lidos no estágio DI serão lidos novamente

## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle



## 2. Organizações do MIPS: pipeline

### ► Conflitos por Dados e Paradas

#### Propagando uma Bolha pelo Pipeline

- Como a instrução load word prossegue pelo pipeline, cria-se uma “bolha” de execução, a qual deve também prosseguir pelo pipeline
- Uma “bolha” deve executar em cada estágio a mesma coisa que uma instrução **NOP** executa
- **NOP:**
  - Todos os sinais de controle em 0 (zero) para os estágios EX, MEM e ER.
  - Estes valores de sinais de controle são passados adiante a cada ciclo de relógio, produzindo o efeito desejado (nenhum registrador ou memória é escrito)



## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle (ou Conflitos de Desvios Condicionais)

Desvio Condicional em Pipeline.

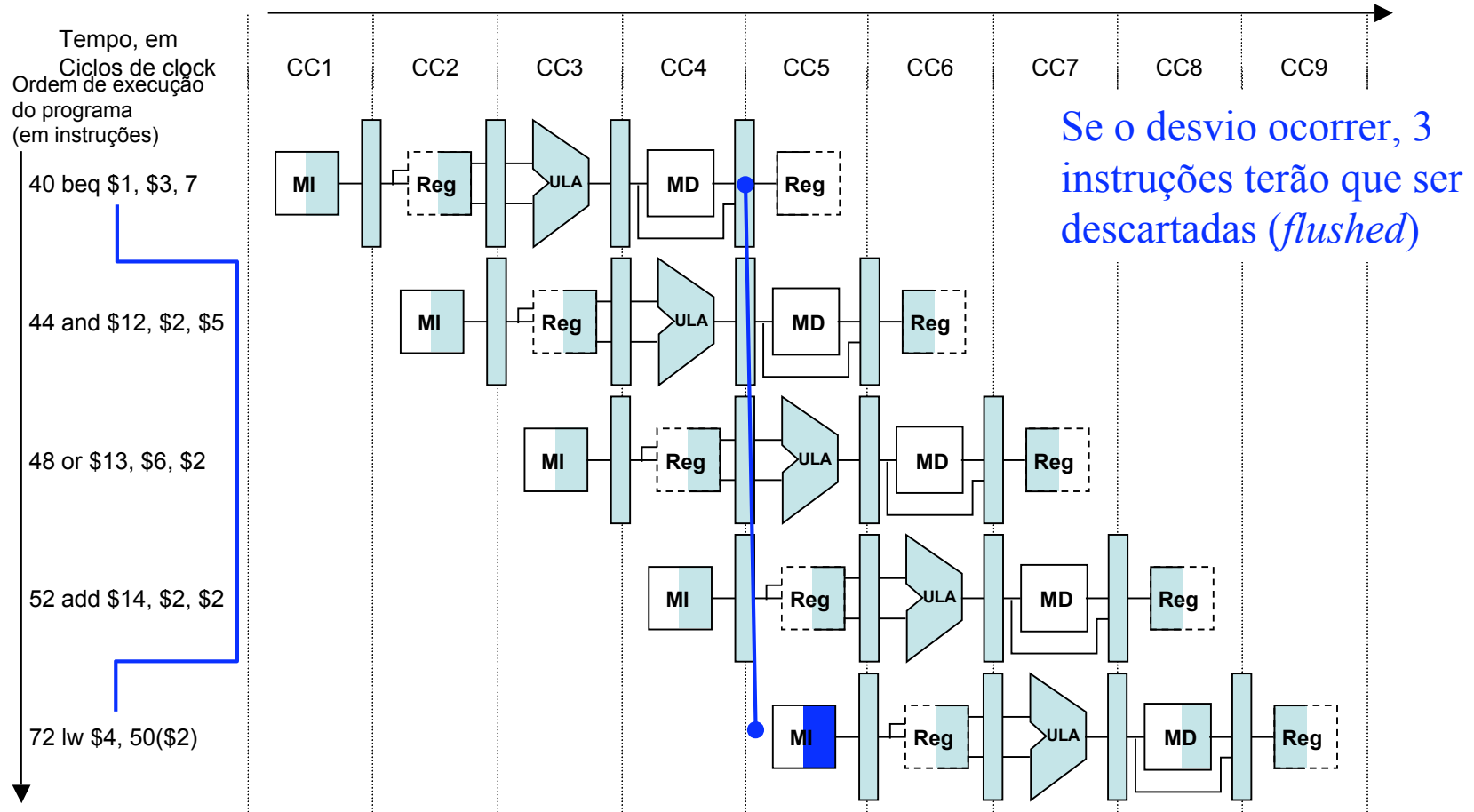
Exemplo:

36	sub	\$10, \$4, \$8	
<b>40</b>	<b>beq</b>	<b>\$1, \$3, 7</b>	# desvio relativo ao PC para $40 + 4 + 7*4 = 72$
44	and	\$12, \$2, \$5	
48	or	\$13, \$2, \$6	
52	add	\$14, \$4, \$2	
56	slt	\$15, \$6, \$7	
...	...		
<b>72</b>	<b>lw</b>	<b>\$4, 50( \$7)</b>	

## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle

Considerando o Bloco Operativo Pipeline Visto Até Aqui...



## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle

- ❑ Conforme já visto anteriormente, a parada no avanço das instruções não é uma solução viável para o desvio condicional
- ❑ Uma alternativa comum é considerar que os desvios condicionais sempre ocorrem, considerando a seqüência normal de execução das instruções
- ❑ Caso o desvio se realize, será necessário descartar as instruções que estiverem sendo buscadas e executadas
- ❑ E a execução deve continuar a partir da instrução armazenada no endereço-alvo do desvio condicional...

## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle

- ❑ Para descartar instruções, basta mudar para 0 os valores originais dos sinais de controle e
- ❑ Também mudar as instruções que estiverem em BI, DI e EX, quando a instrução de desvio condicional chegar ao estágio MEM

## 2. Organizações do MIPS: pipeline

### ► Conflitos de Controle

