UNIVALI

Laboratório de Sistemas
Embarcados e Distribuídos

Organização do RISC-V:
Visão Geral

Histórico de revisões

Revisão	Data	Responsável	Descrição
0.1	- X -	Prof. Cesar Zeferino	Primeira versão
0.2	03/2016	Prof. Cesar Zeferino	Revisão do modelo e atualização de conteúdo
0.3	05/2020	Prof. Cesar Zeferino	Revisão geral
0.4	12/2022	Felski	Revisão geral da Organização

Observação: Este material foi produzido por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos (LEDS – Laboratory of Embedded and Distributed Systems) da Universidade do Vale do Itajaí e é destinado para uso em aulas ministradas por seus pesquisadores.

Introdução

- **Objetivo**
  - Ter uma visão geral das alternativas de organização do RISC-V
- **Conteúdo**
  - Ciclo de instrução
  - Tipos de organização

Introdução

Bibliografia

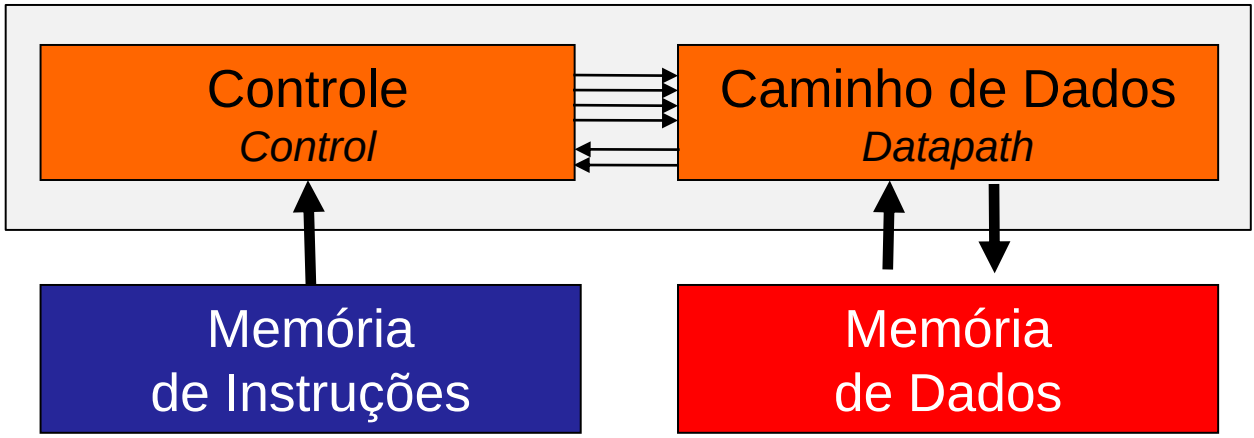
- PATTERSON, David A.; HENNESSY, John L. O Processador. *In*: _____. **Organização e projeto de computadores**: a interface hardware/software. 4. ed. Rio de Janeiro: Campus, 2014. cap. 4.

Introdução

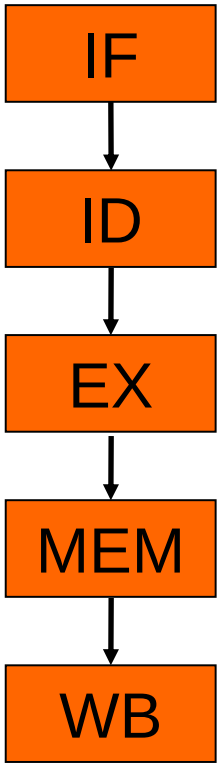
- **A organização de um processador pode ser estruturada em duas partes: Caminho de dados e Controle**

- **Caminho de dados (*data path*)**
  - Parte responsável pela execução das instruções, ou seja, pelo processamento de dados

- **Controle**
  - Parte responsável pela decodificação das instruções e pelo acionamento de todos os componentes do caminho de dados associados à execução de cada instrução



Ciclo de instrução



- **Ciclo de execução de uma instrução no RISC-V**
  - **IF**: Busca da instrução
  - **ID**: Decodificação da instrução e busca de operandos
  - **EX**: Execução ou cálculo do endereço efetivo
  - **MEM**: Acesso à memória ou realização de desvio
  - **WB**: Escrita em registrador

Legenda

IF	= Instruction Fetch
ID	= Instruction Decode
EX	= EXecution
MEM	= MEMory access
WB	= Write-Back

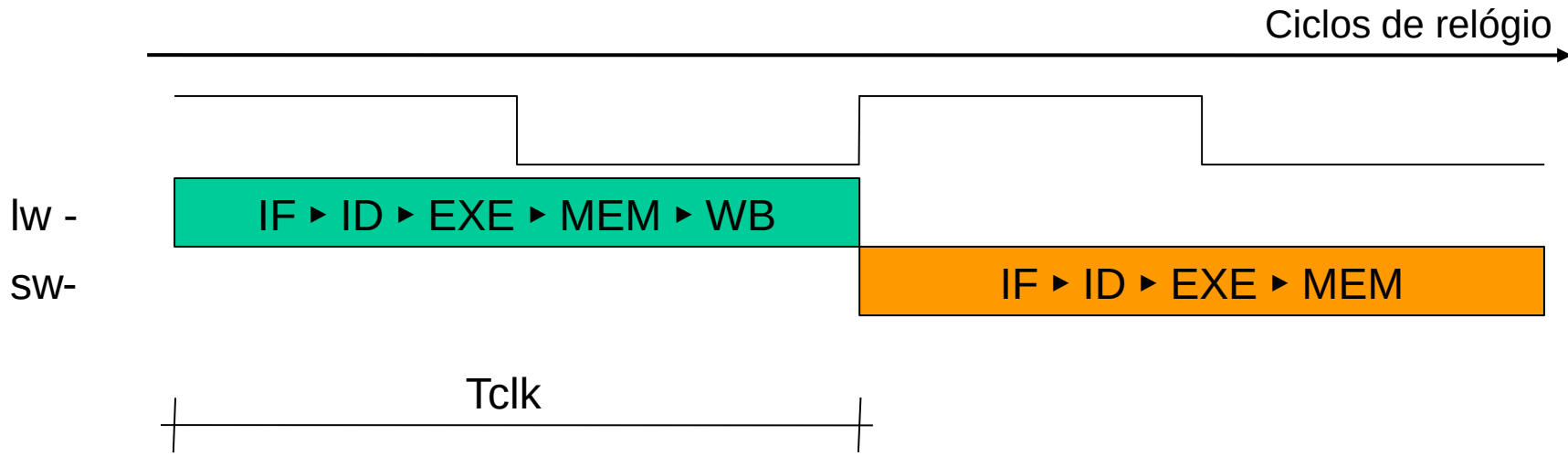
Ciclo de instrução

- **O que é feito em cada ciclo?**
  - **IF**: Lê uma instrução da memória e incrementa o PC (PC+4)
  - **ID**: Decodifica a instrução, busca os registradores (rs e rt) e estende o sinal do operando imediato
  - **EX**: Depende da classe de instrução
    - Lógica e aritmética: realiza a operação
    - Transferência: calcula o endereço da memória
    - Desvio condicional: calcula o endereço de desvio e compara
  - **MEM**: Depende da classe de instrução
    - Transferência: acessa a memória
    - Desvio condicional: atualiza o PC se condição verdadeira
  - **WB**: Depende da classe de instrução
    - Lógica e arimética, load: escreve no banco de registradores

Tipos de organização

Organização Monociclo

- Todas as fases do ciclo de instrução são executadas em um único ciclo de relógio (Tclk)
- Tclk é definido pela instrução mais lenta (load word)
- Todas as instruções consomem um ciclo de relógio (1 ciclo = monociclo)

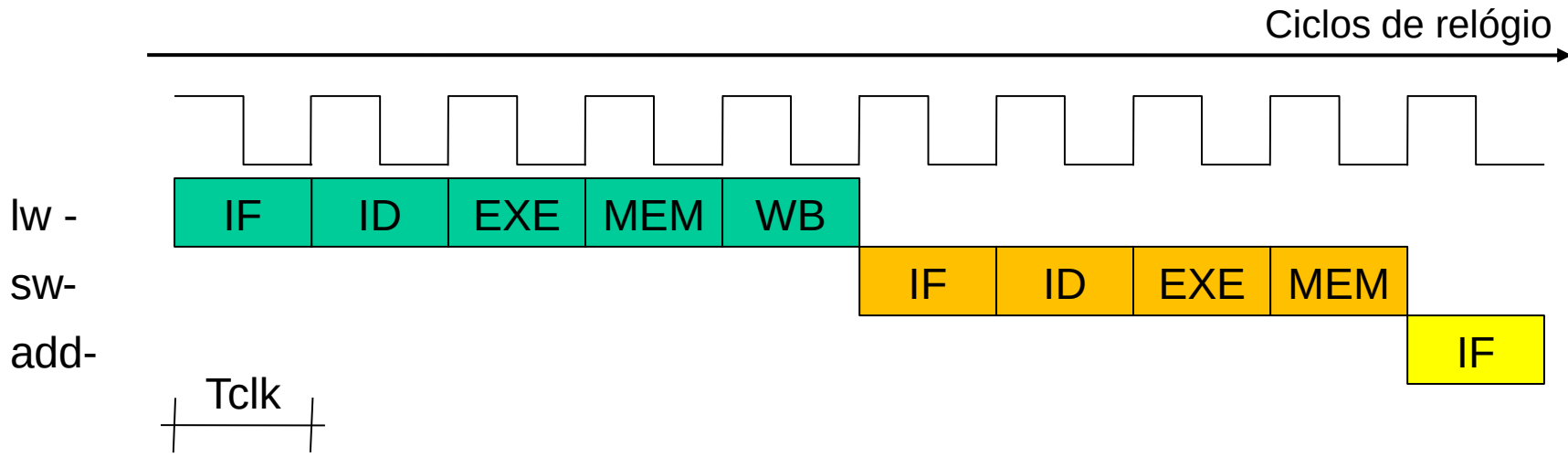


Vantagem: Implementação simplificada

Tipos de organização

Organização Multiciclo

- Cada fase do ciclo de instrução é executada em um Tclk
- Tclk é definido pela fase mais lenta (acesso à memória)
- As instruções consomem vários ciclos de relógio (multiciclo)
- Diferentes instruções consomem diferentes quantidades de ciclos de relógio
- Em um dado ciclo de relógio, apenas uma fase de uma instrução é executada

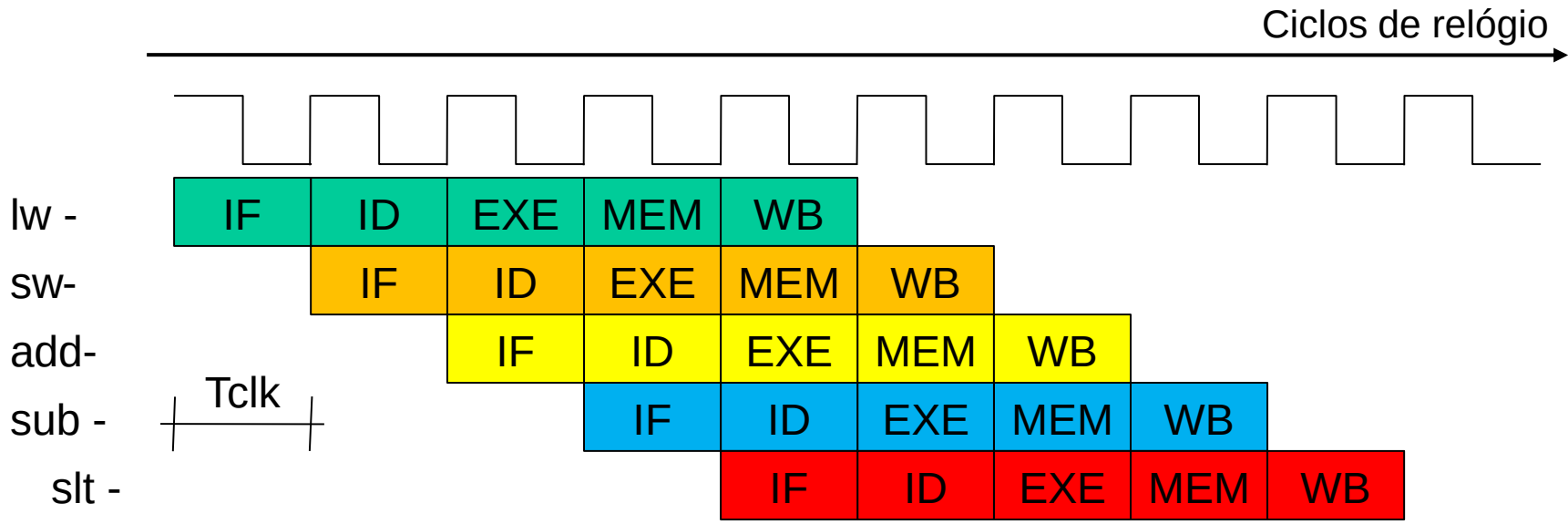


Vantagem: Instruções que usam menos fases são executadas em menos tempo

Tipos de organização

Organização em pipeline

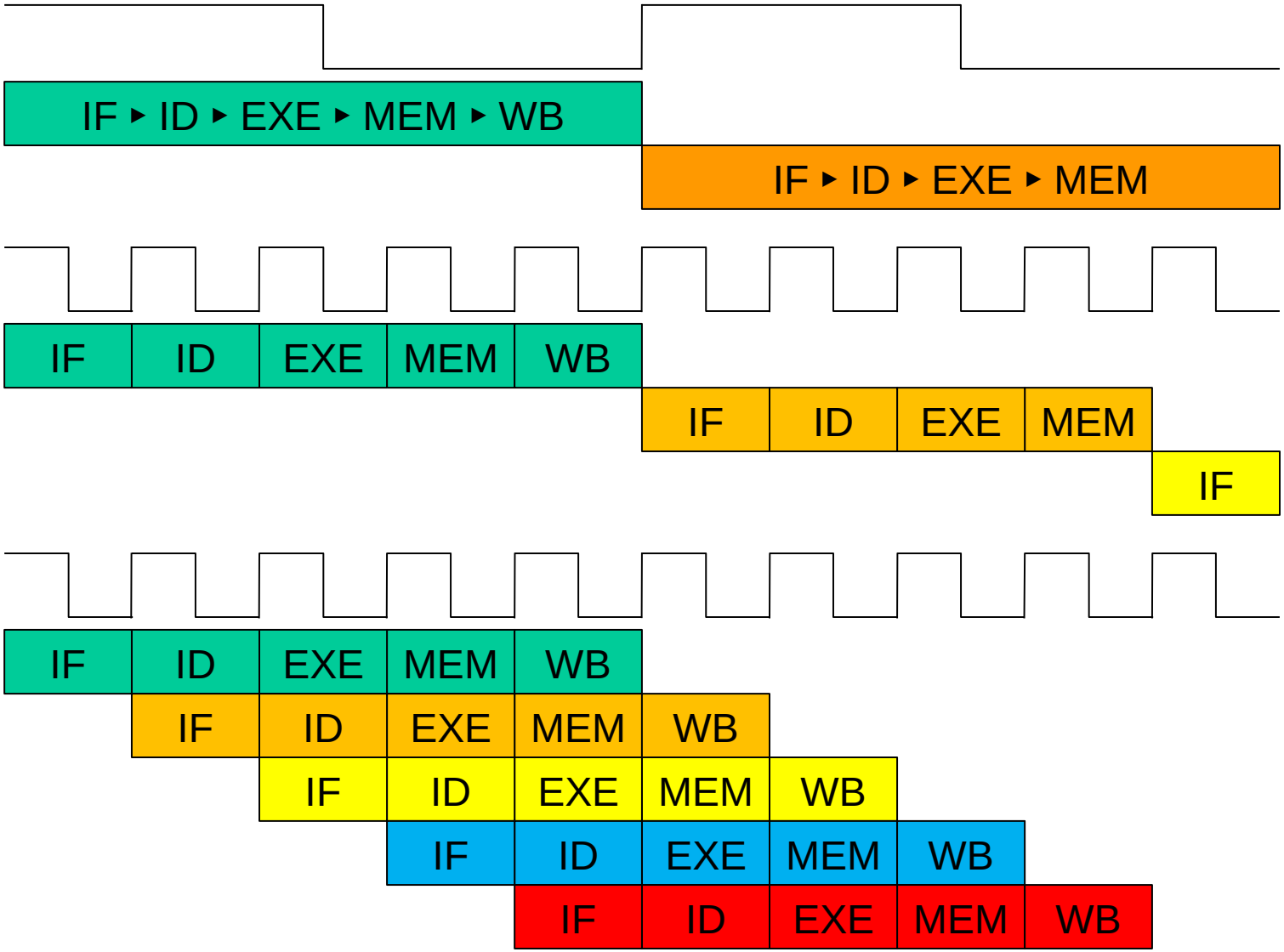
- Cada fase do ciclo de instrução é executada em um Tclk
- Tclk é definido pela fase mais lenta (acesso à memória)
- Todas as instruções gastam o mesmo número de ciclos (5)
- A cada ciclo, uma instrução é concluída e uma nova instrução é iniciada
- Em um dado ciclo é possível ter várias fases diferentes em execução para diferentes instruções



Vantagem: Maior vazão e menor tempo de execução

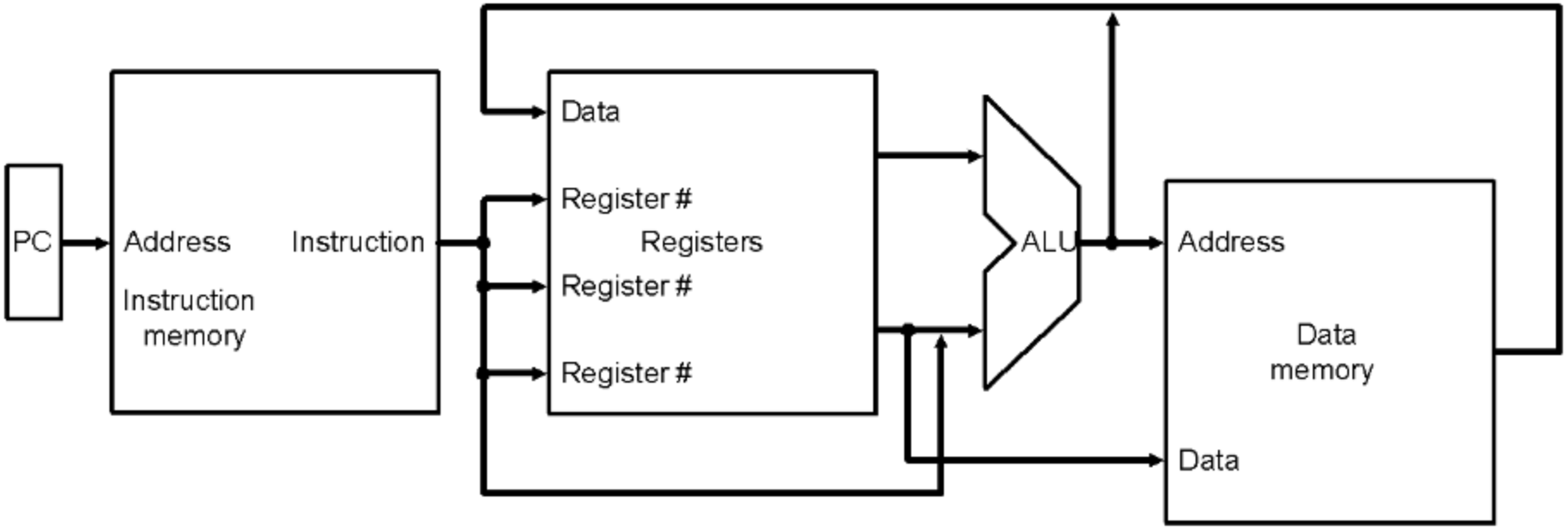
A organização Monociclo

- Comparando



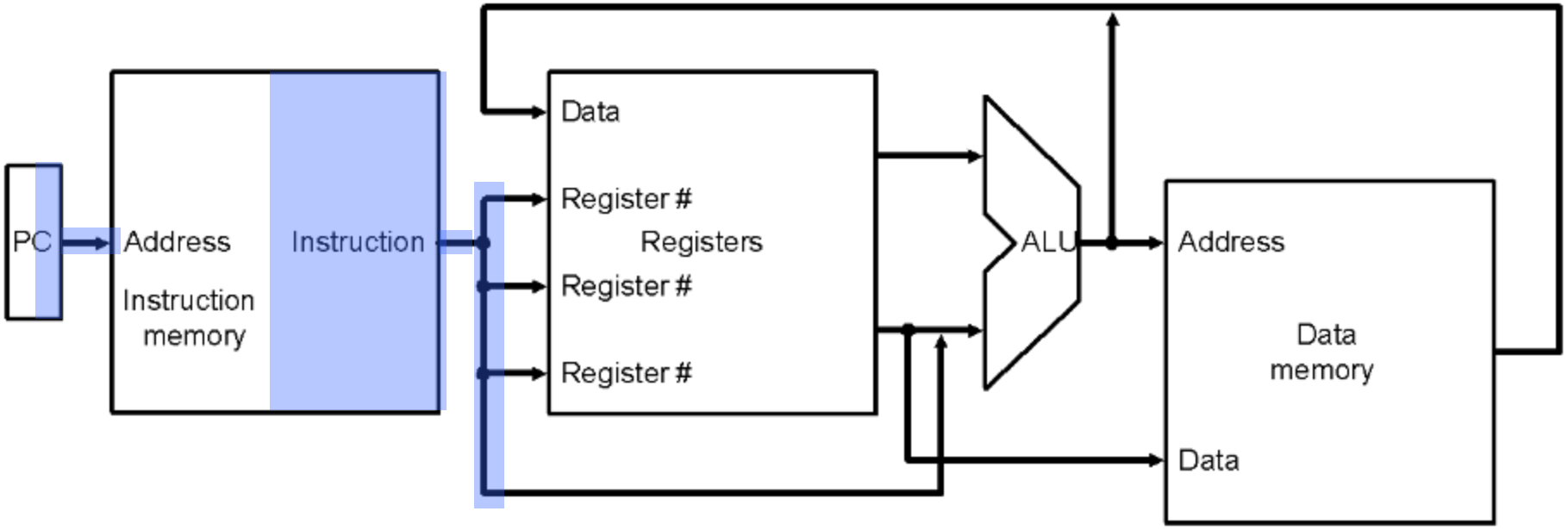
A organização Monociclo

- **Memória de instruções**: Armazena o programa em execução
- **Memória de Dados**: Armazena os dados do programa
- **PC**: Indica um endereço na Memória de Instruções
- **Banco de registradores**: Armazenamento temporário de dados dentro do processador
- **UAL**: Unidade Aritmética Lógica (ULA, UAL ou ALU)



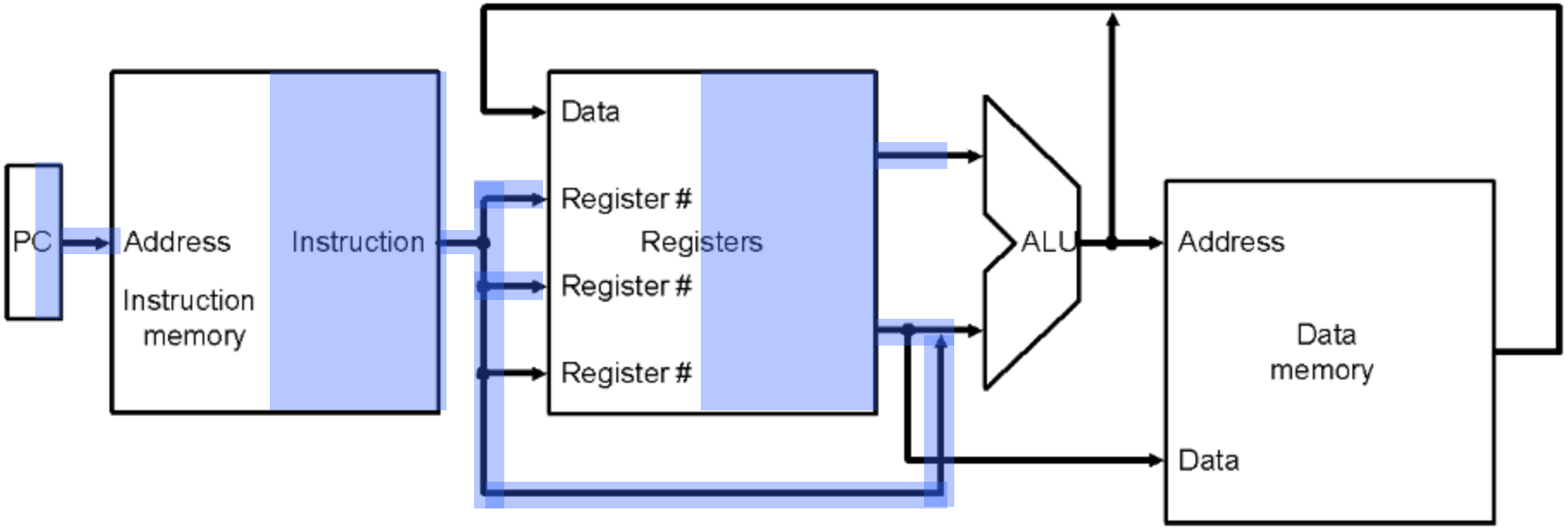
A organização Monociclo

- **IF: Busca da Instrução (Load Word)**
  - É feita a leitura da posição da memória de instruções apontada pelo PC
  - A instrução é disponibilizada na saída da memória, permanecendo até o final do ciclo



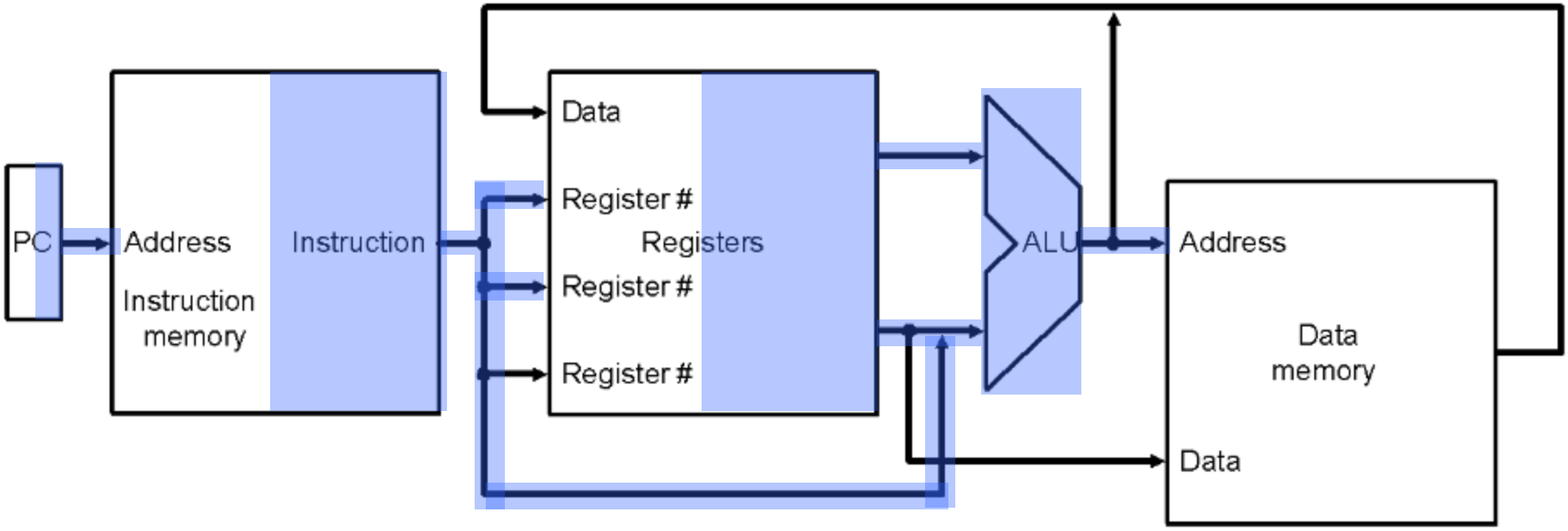
A organização Monociclo

- **ID: Decodificação da instrução (e busca dos operandos)**
  - É feita a decodificação dos campos op e function (se op = 0)
  - Os registradores rs e rt são lidos e é feita a extensão de sinal do campo imed.
  - No caso da instrução lw, apenas os campos rs e imed serão usados com operandos fonte



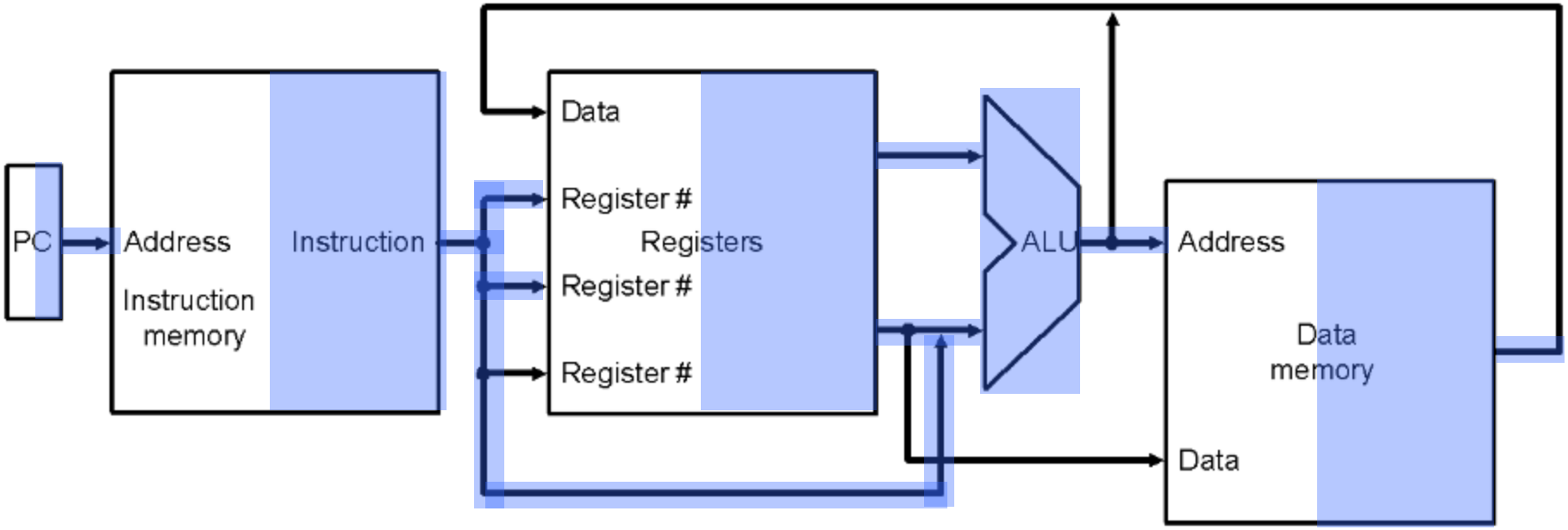
A organização Monociclo

- **EXE: Cálculo do endereço efetivo da memória**
  - A ALU realiza a soma do conteúdo apontado por rs com a extensão de sinal do imed para calcular o endereço a ser lido da memória de dados
  - ALU.result = (rs) + signal_ext(imed)



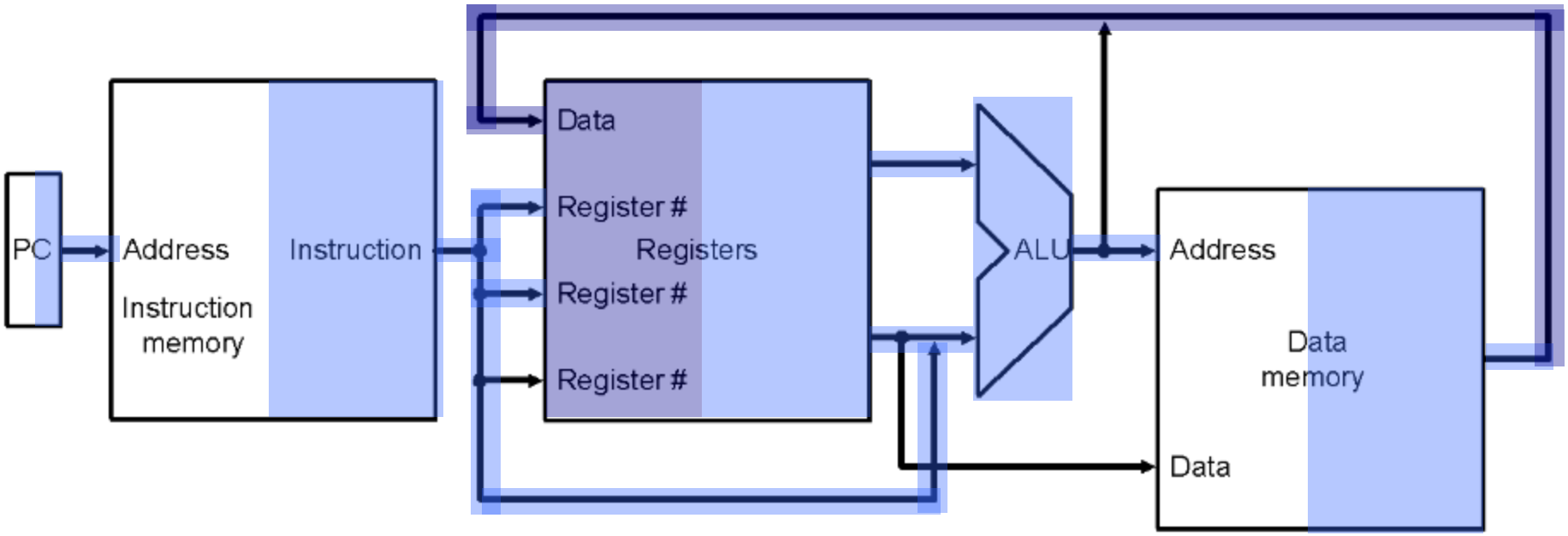
A organização Monociclo

- **MEM: Acesso à memória**
  - É feita a leitura da posição da memória de dados apontado pela saída da ALU



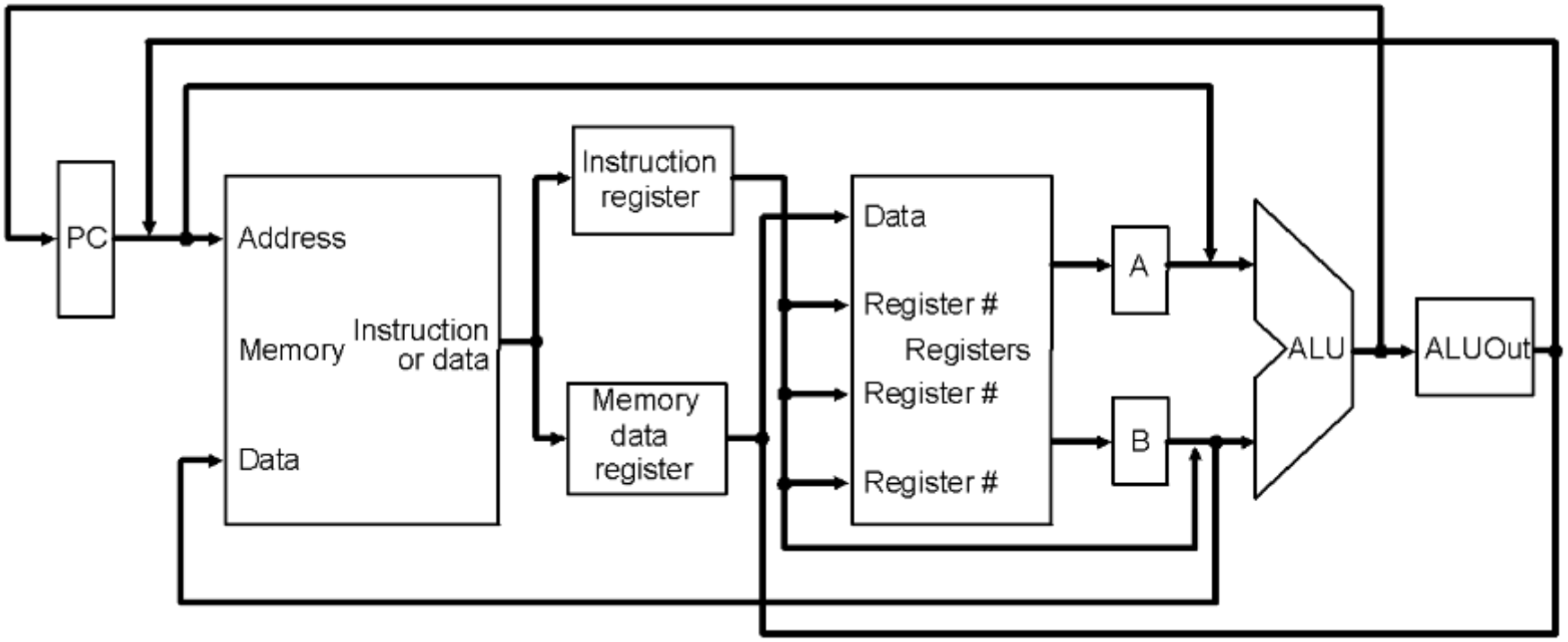
A organização Monociclo

- **WB: Escrita no registrador**
  - O dado lido da memória de dados é escrito no registrador apontado pelo campo rt da instrução



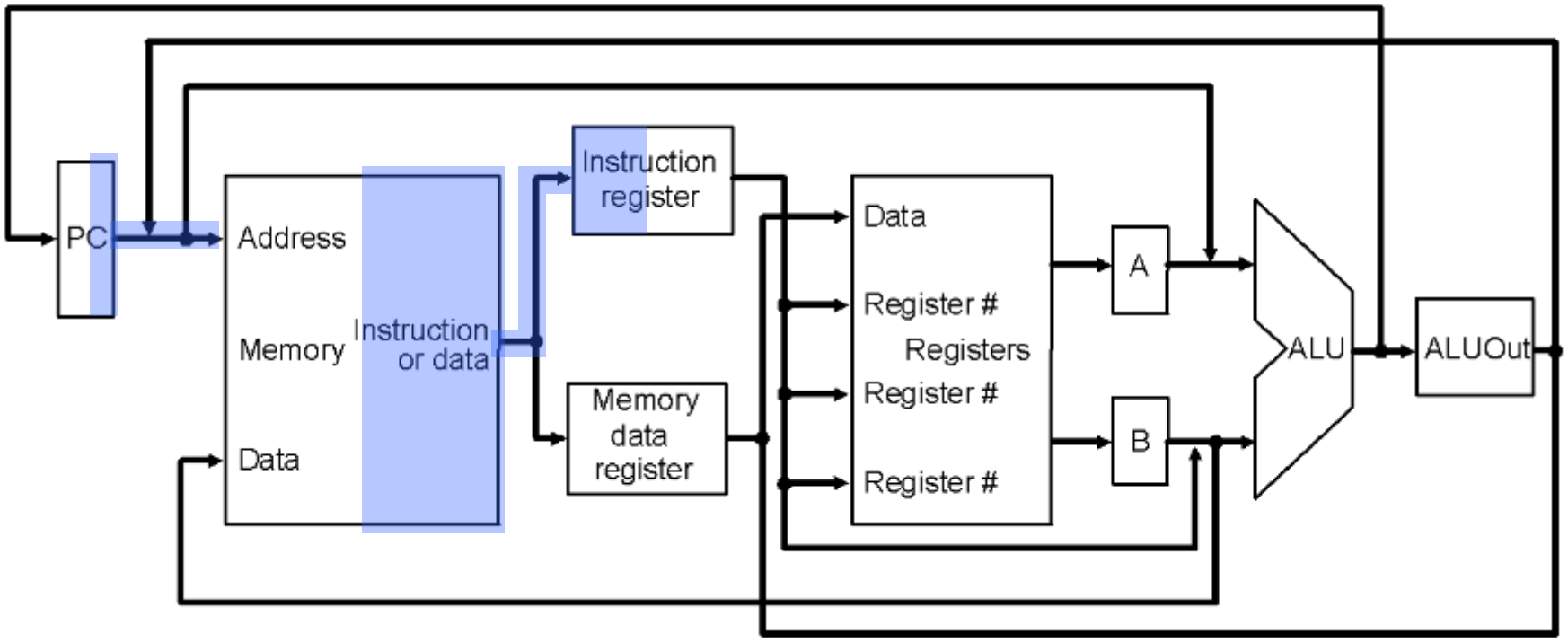
A organização Multiciclo

- **Memória unificada:** Armazena o programa em execução e os dados processados ou a serem processados
- **Registradores de propósito especial:** Mantém os dados lidos em uma fase (ciclo) para uso na fase (ciclo) seguinte. São cinco registradores novos: IR, MDR, ALUOut, A e B



A organização Multiciclo

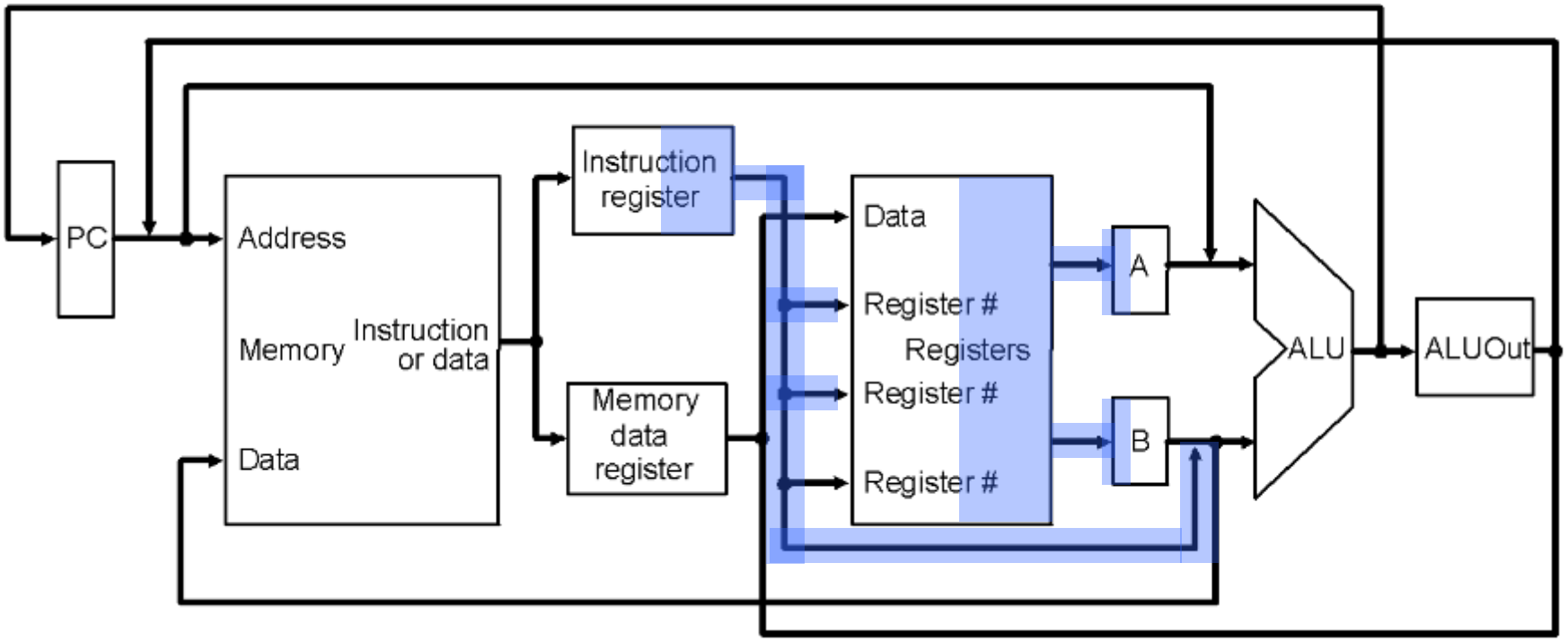
- **IF: Busca da Instrução (Load Word)**
  - A instrução lida da memória unificada é armazenada no registrador de instrução (IR)



IF	ID	EXE	MEM	WB

A organização Multiciclo

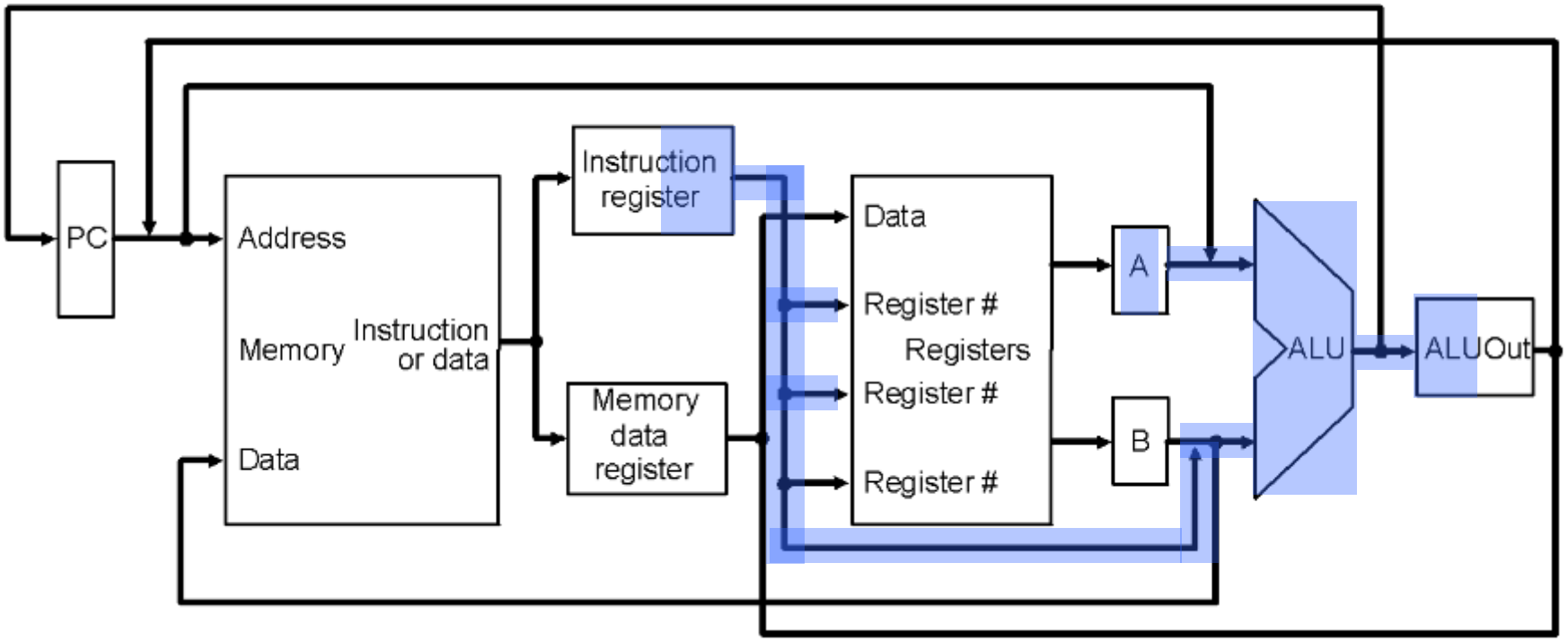
- **ID: Decodificação da instrução e busca dos operandos**
  - É feita a decodificação dos campos op e function (se op = 0)
  - Os registradores rs e rt são lidos e armazenados nos registradores A e B (de entrada da UAL)
  - É feita a extensão de sinal do campo imed



A organização Multiciclo

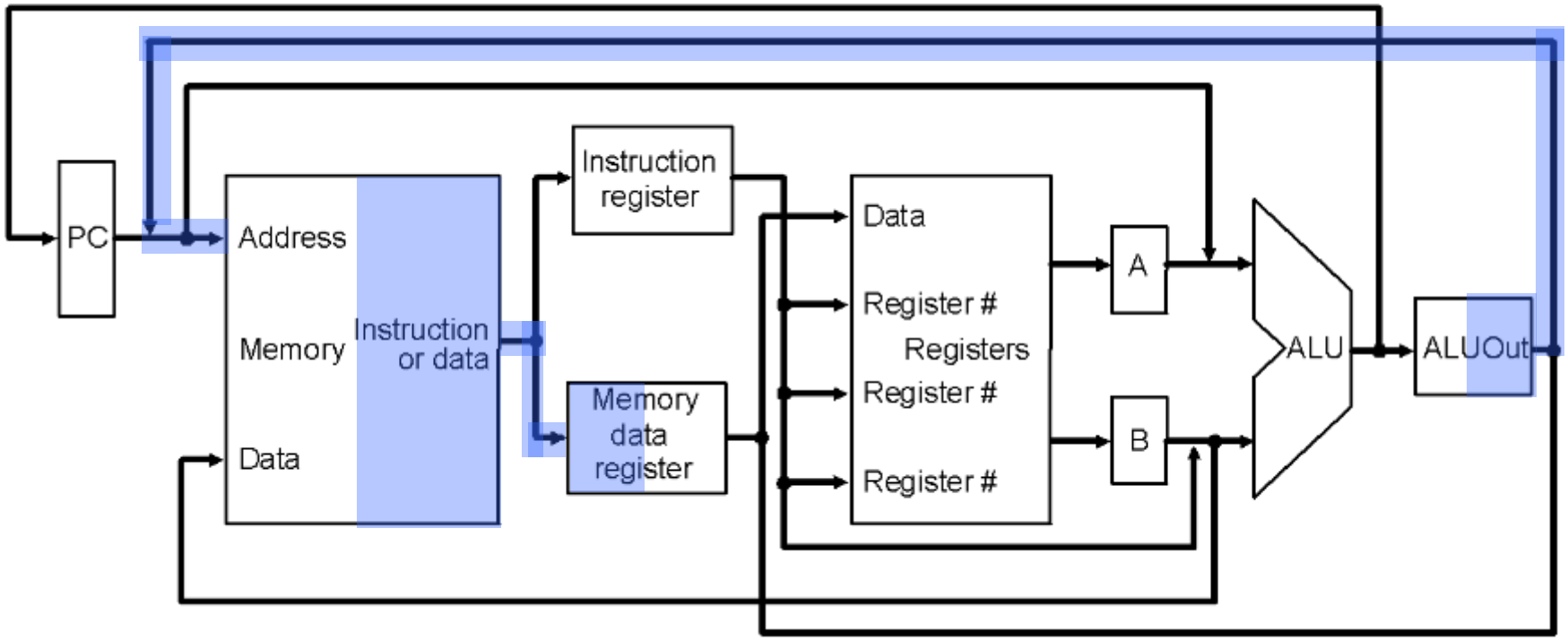
EXE: Cálculo do endereço efetivo

- A ALU realiza a soma do conteúdo armazenado no registrador A com a extensão de sinal do campo imed para calcular o endereço do dado a ser lido da memória
- O resultado é armazenado no registrador UALOut



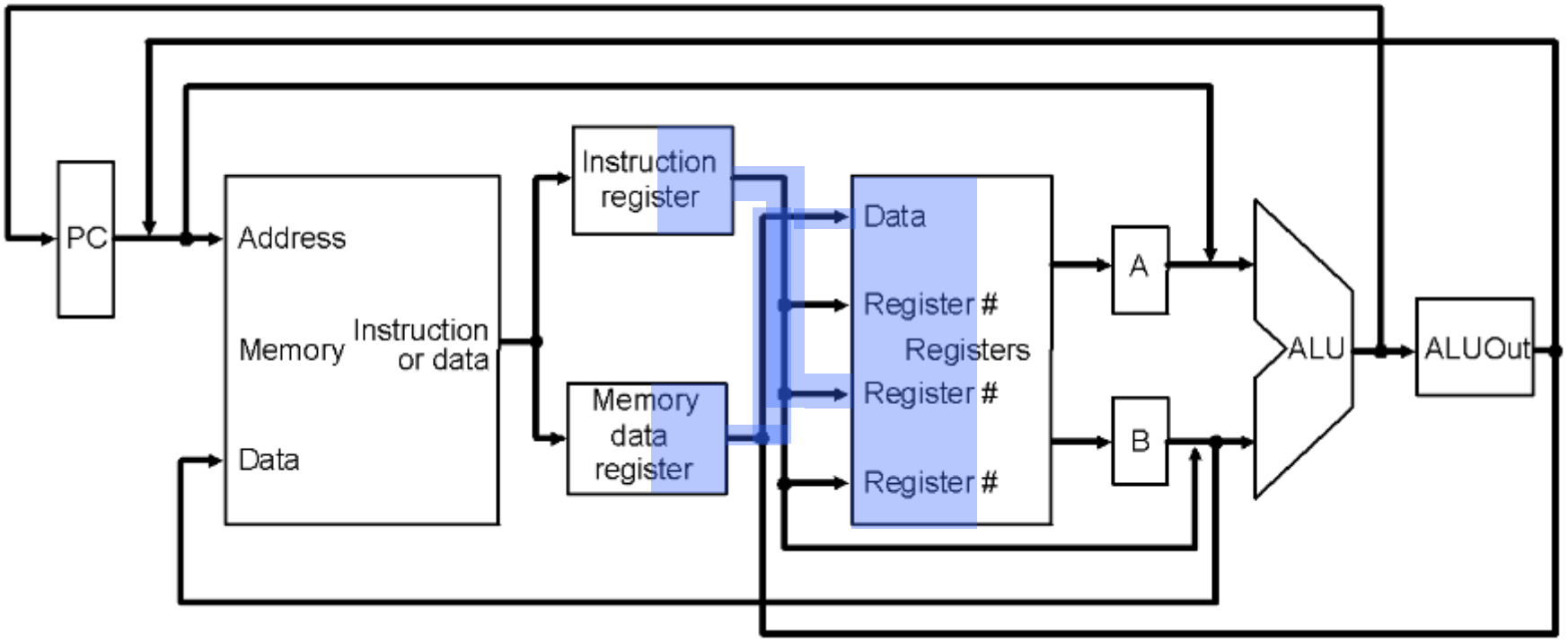
A organização Multiciclo

- **MEM: Acesso à memória**
  - É feita a leitura da posição da memória apontada pelo registrador UALOut
  - O dado lido é escrito no registrado MDR



A organização Multiciclo

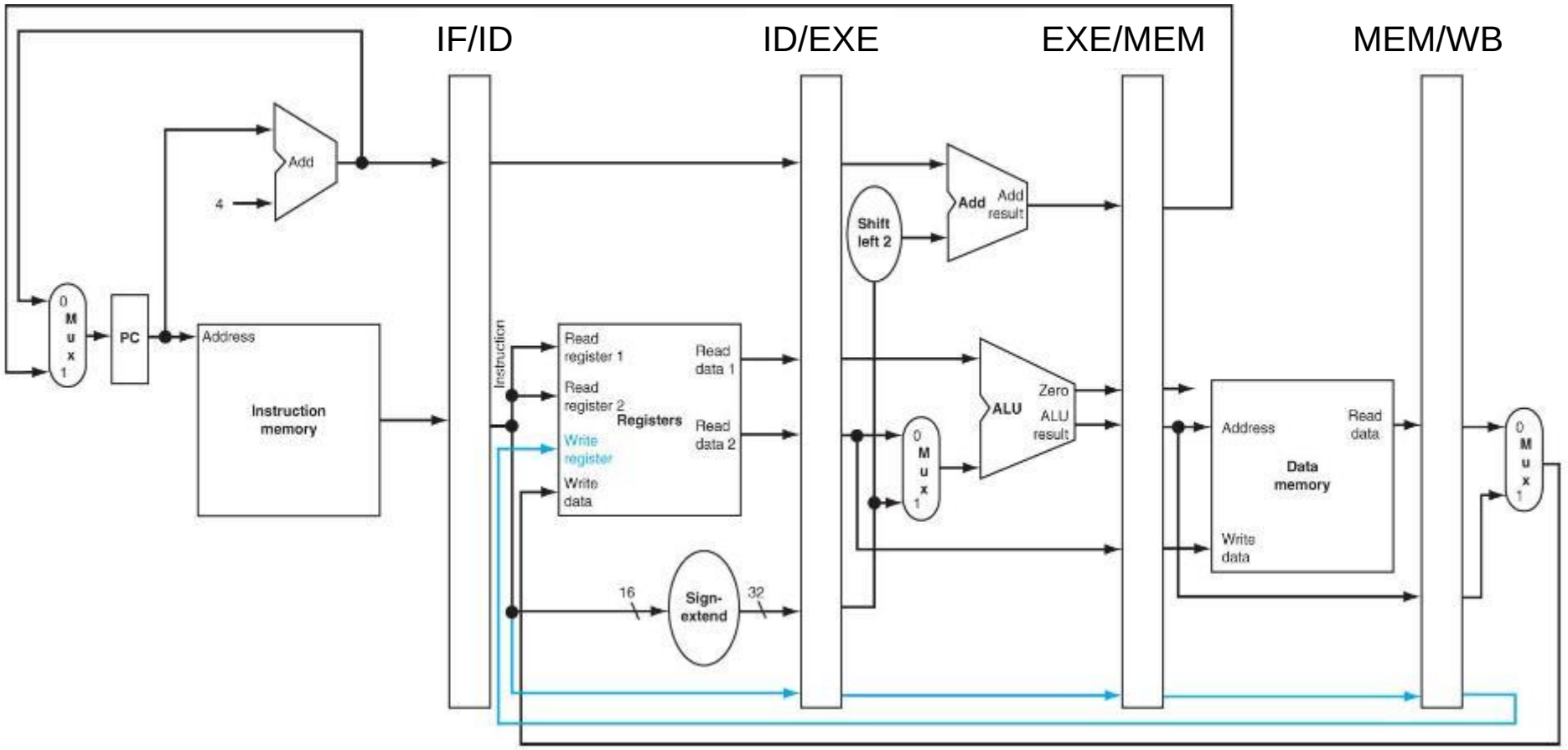
- **WB: Escrita no registrador**
  - O dado armazenado no registrado MDR é escrito no registrador apontado pelo campo rt da instrução



IF	ID	EXE	MEM	WB

A organização em pipeline

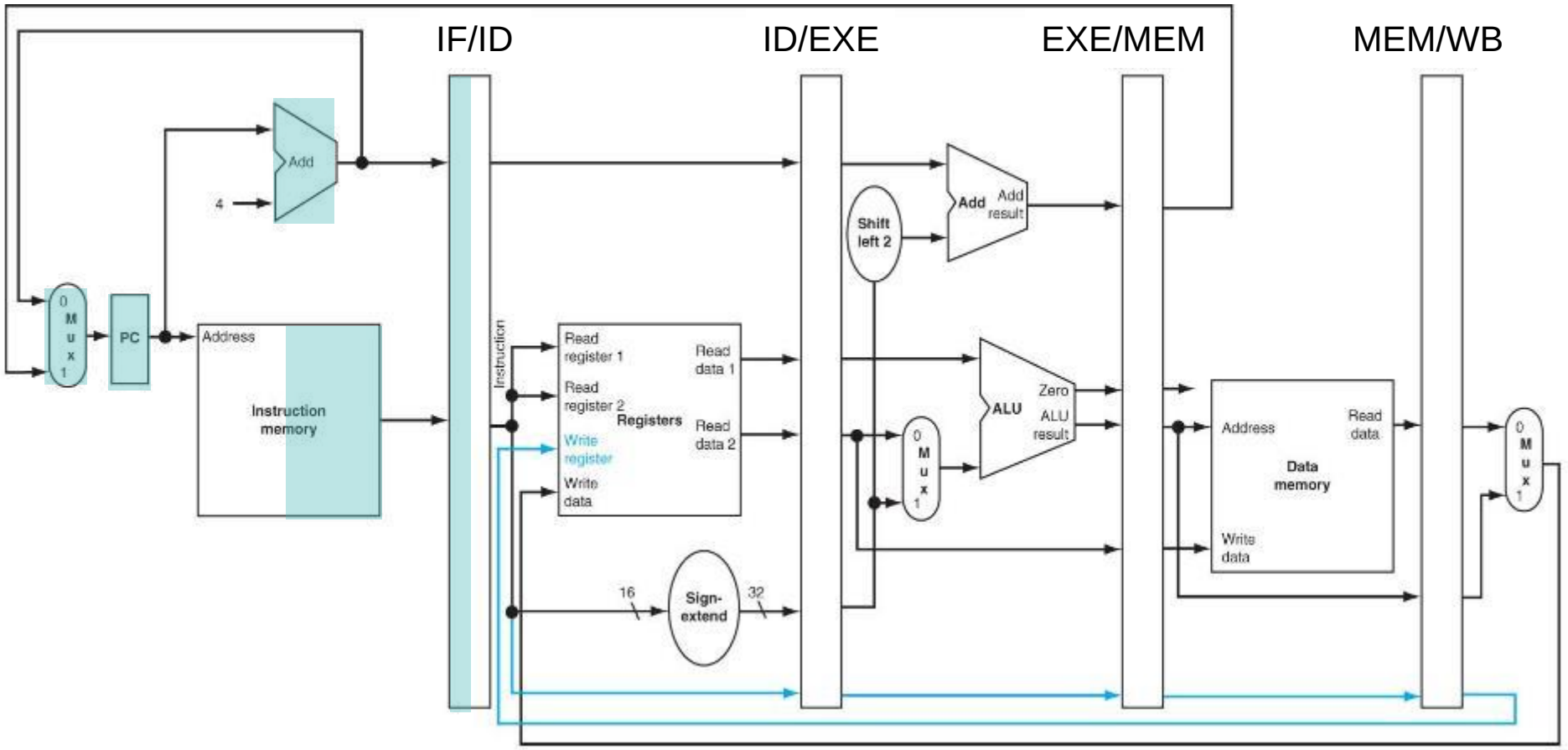
- Organização monociclo + registradores do multiciclo



IF	ID	EXE	MEM	WB

A organização em pipeline

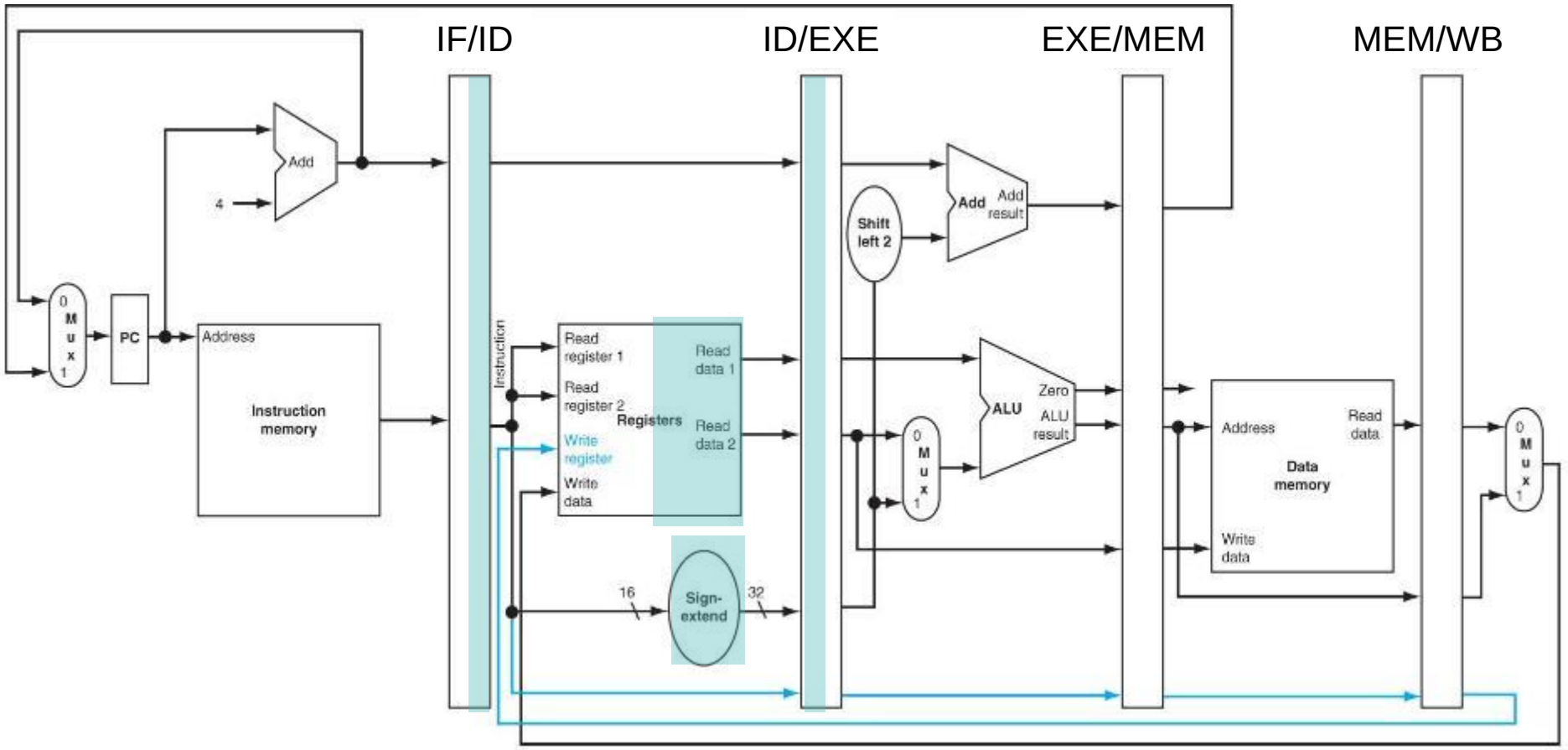
- IF: Busca da instrução (Load Word)



IF	ID	EXE	MEM	WB

A organização em pipeline

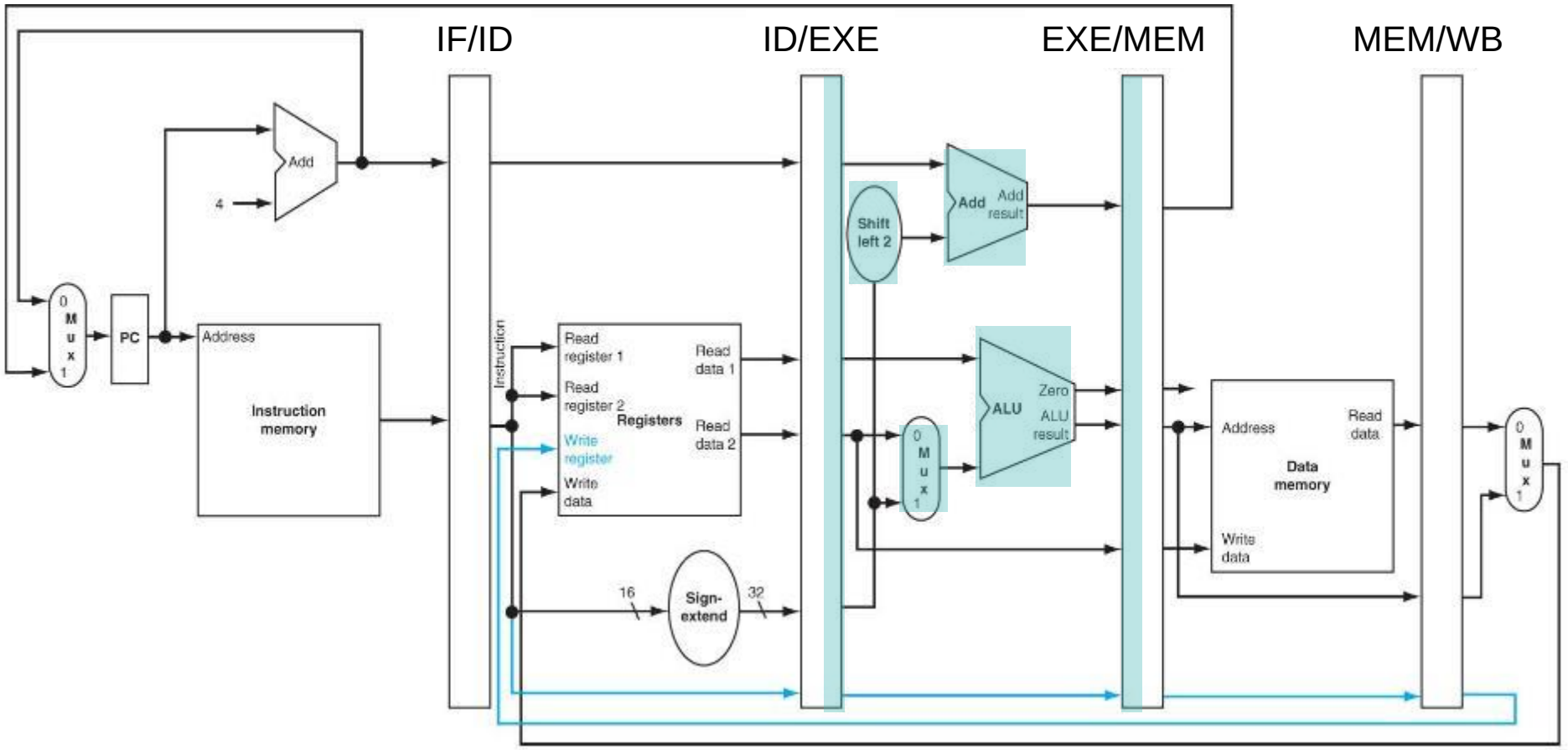
- **ID: Decodificação da instrução e busca dos operandos**



IF	ID	EXE	MEM	WB

A organização em pipeline

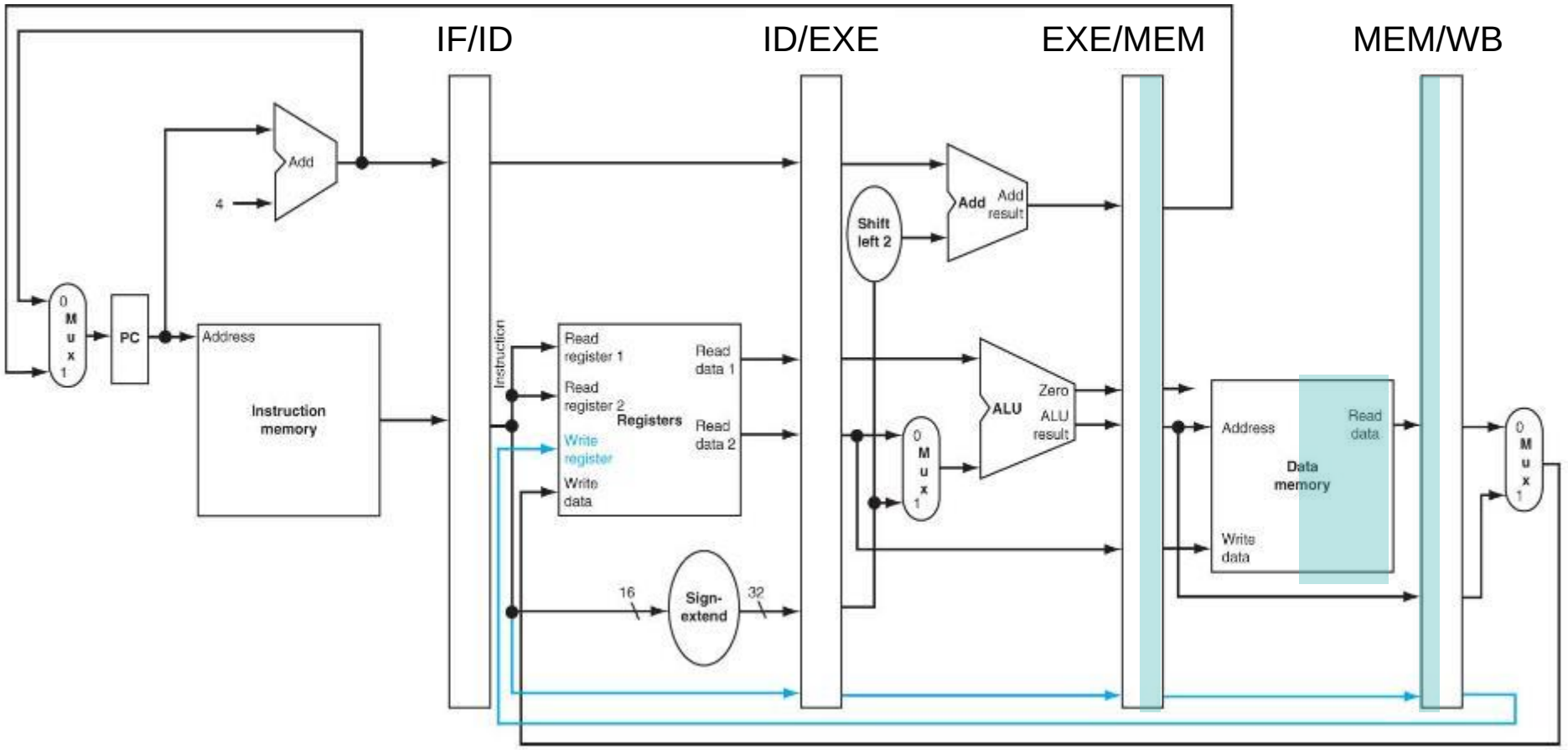
- **EXE: Cálculo do endereço efetivo**



IF	ID	EXE	MEM	WB

A organização em pipeline

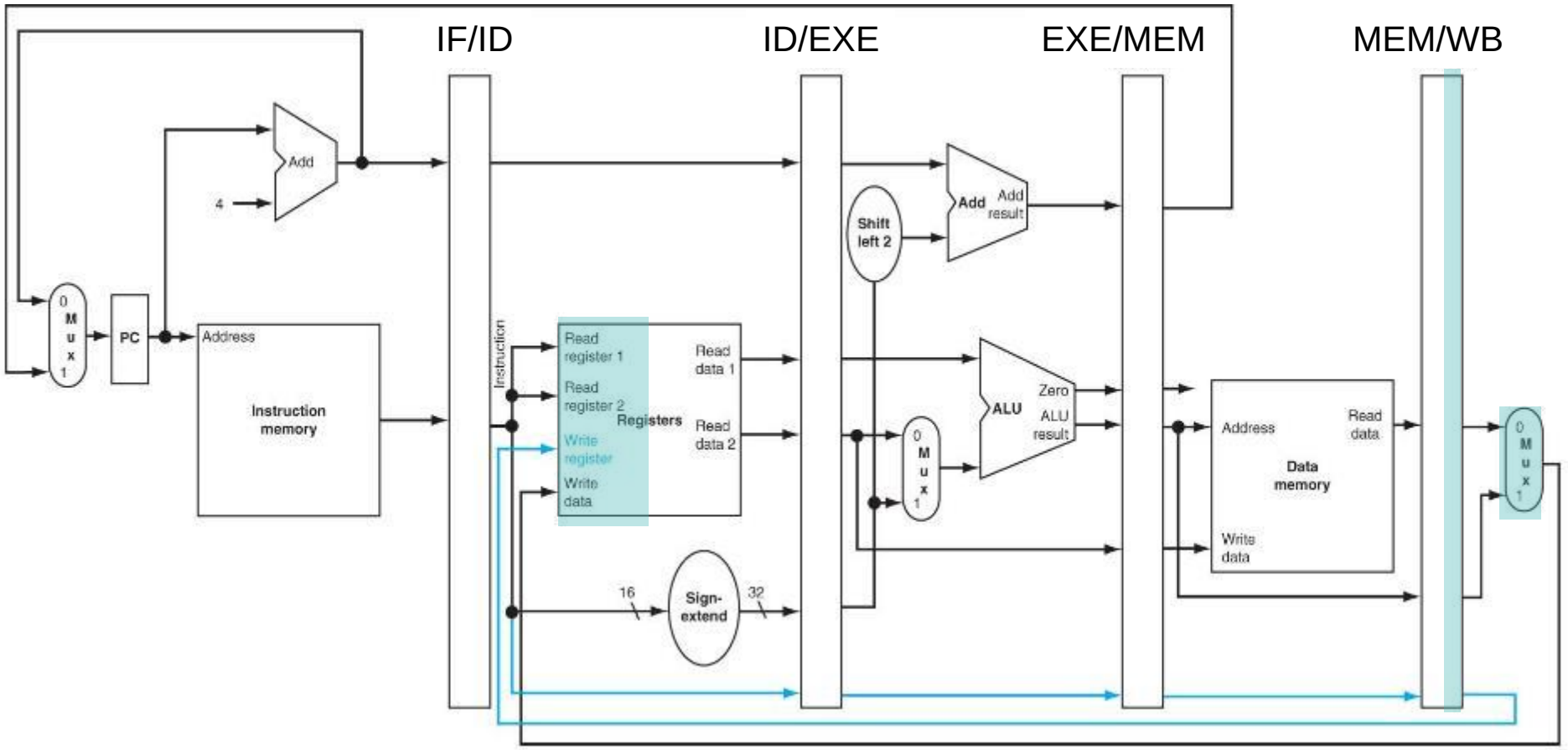
- **MEM: Acesso à memória**



IF	ID	EXE	MEM	WB

A organização em pipeline

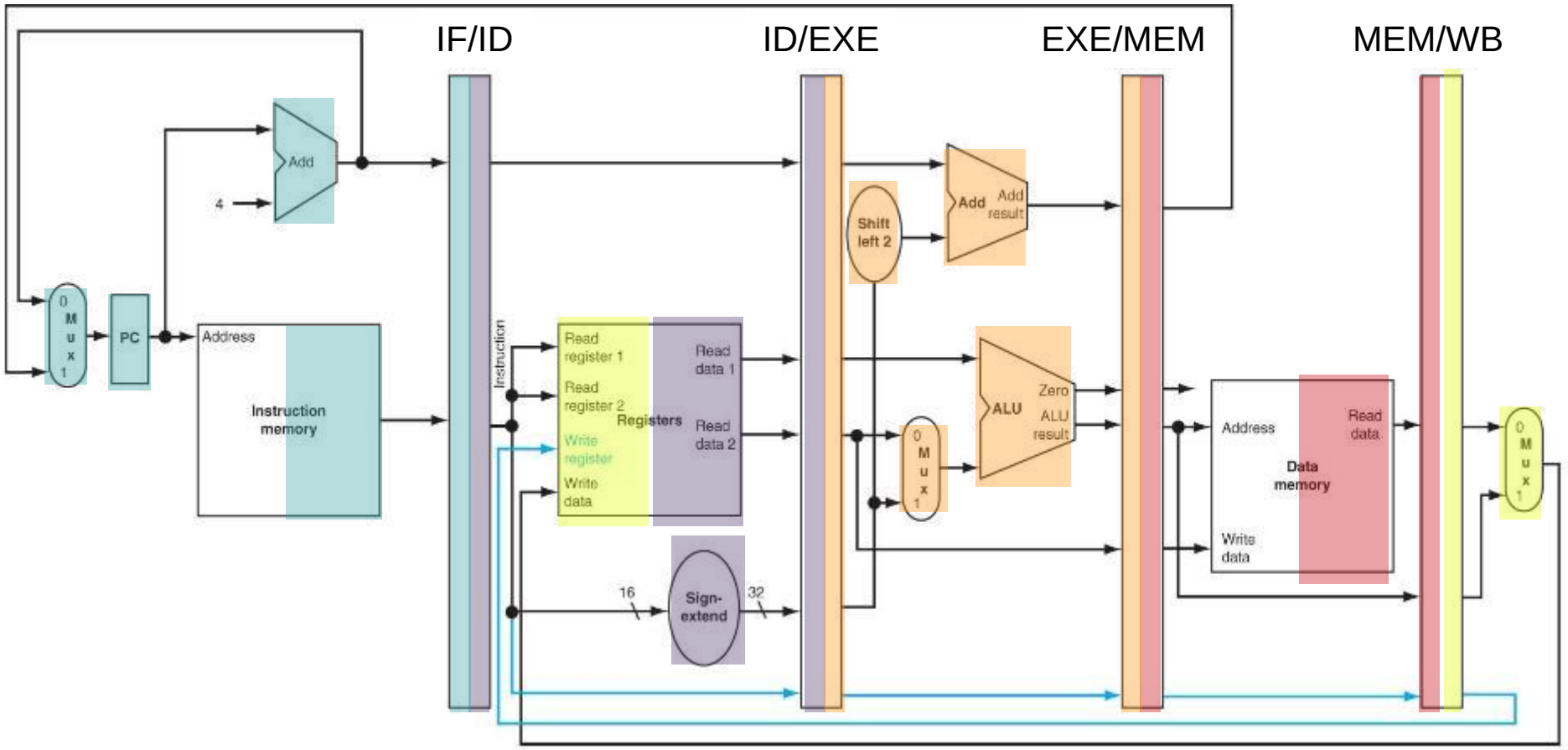
- **WB: Escrita no registrador**



IF	ID	EXE	MEM	WB

A organização em pipeline

- **Cinco instruções executadas simultaneamente (paralelismo temporal)**



IF | ID | EXE | MEM | WB