



Organização de Computadores

Aula 4

Bloco operacional mono-ciclo



Recapitulação

Processador = Caminho de dados + Controle

- **Caminho de Dados:** Realiza operações aritméticas. Ex: Somadores, ULAs.
- **Controle:** Parte do processador que controla o Caminho de dados e outros dispositivos.

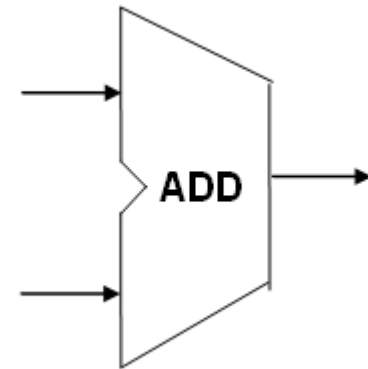
Recapitulação

Tipos de Elementos Lógicos

- Elementos Combinacionais

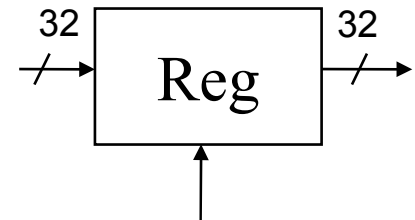
Sua saída depende somente das entradas presentes naquele momento.

Ex: Somadores, Multiplexadores.



- Elementos de Estado

Possui algum tipo de memória interna, ou seja, pode armazenar alguma informação. Ex: Registradores.





Recapitulação

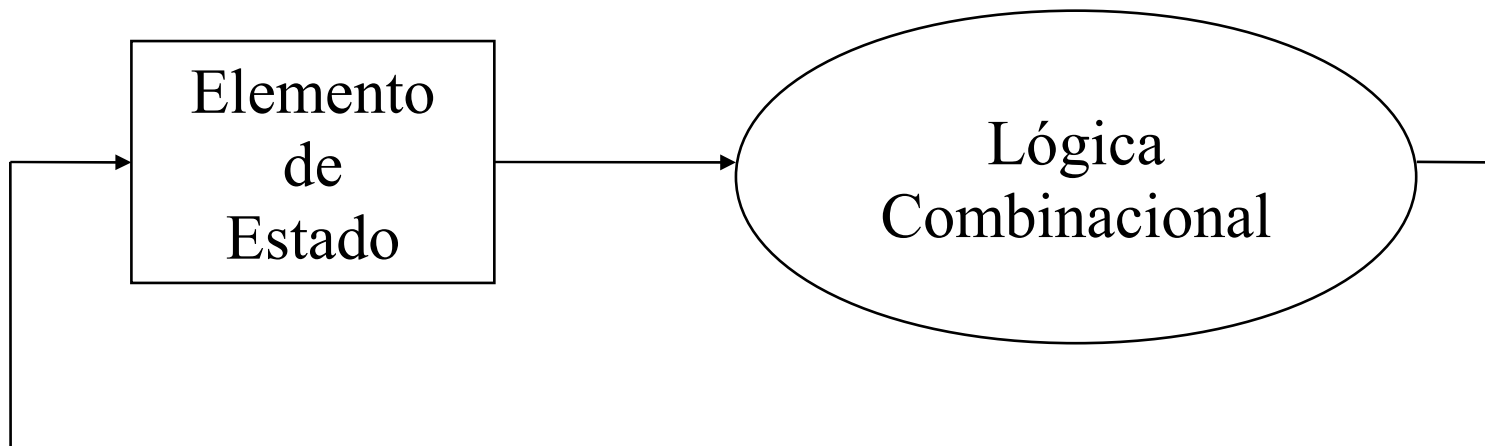
Metodologia de Temporização





Implementação Mono-ciclo

- Toda a instrução será executada em um ciclo de clock



O tempo de ciclo será igual ao tempo gasto na instrução mais demorada.



Bloco Operacional Mono-ciclo

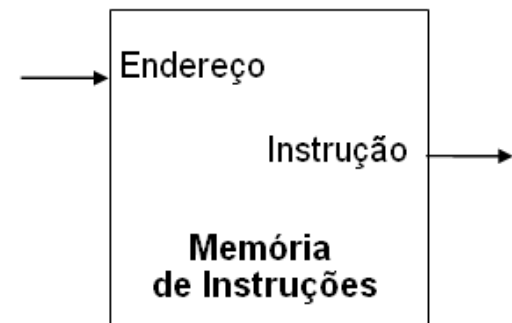
1. Blocos Básicos da Construção do Caminho de Dados
2. Conjunto de Instruções Analisado
 - 2.1. Instruções Aritméticas
 - 2.2. Instruções de Acesso à Memória
 - 2.3. Instruções de Desvio Condicional
 - 2.4. Combinando Instruções
3. Bloco Operacional Completo
4. Cálculo do Período do Ciclo de Relógio
5. Resumo da Aula

1. Blocos Básicos para Construção do Caminho de Dados

- Banco de Registradores

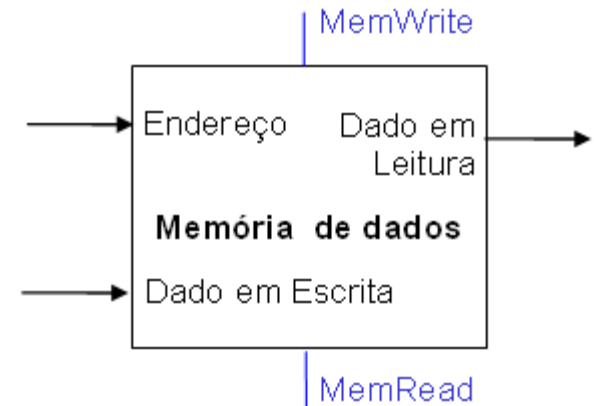


- Memória de Instruções

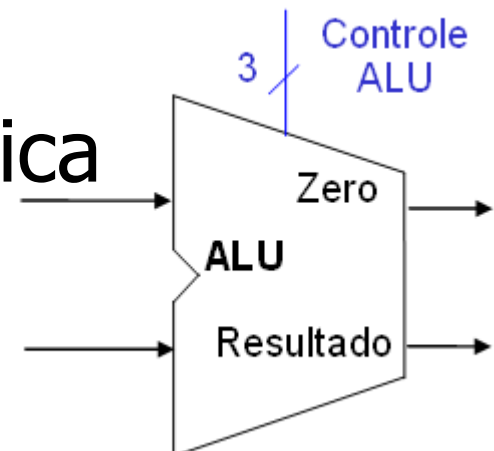


1. Blocos Básicos para Construção do Caminho de Dados

- Memória de Dados



- Unidade Lógica e Aritmética (ULA)



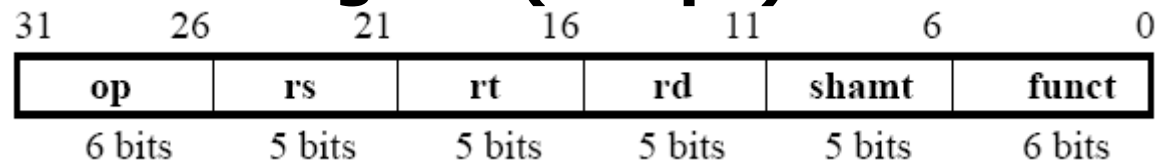
2. Conjunto de Instruções

Subconjunto de instruções do MIPS

- Instruções aritméticas e lógicas (R-Tipo)

add rd, rs, rt

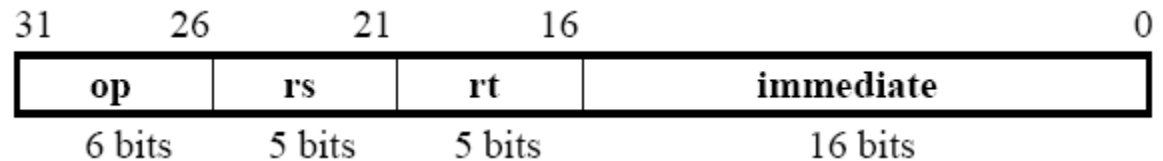
sub rd, rs, rt



- Instruções de acesso à memória (I-Tipo)

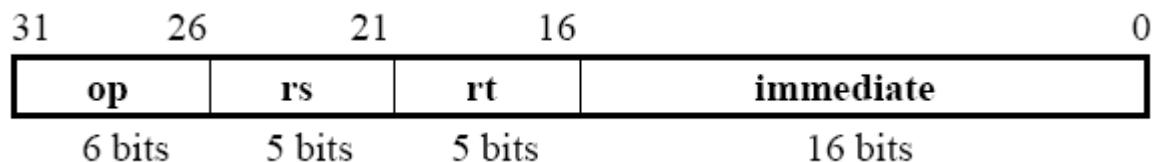
lw rt, rs, imm16

sw rt, rs, imm16



- Instruções de desvio (J-Tipo)

beq rs, rt, imm16





2. Conjunto de Instruções

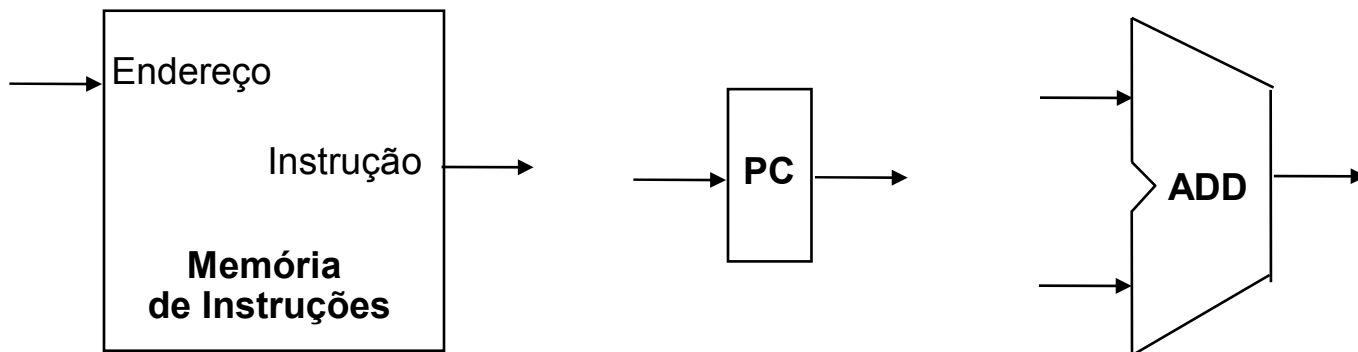
Para todas as instruções, é necessário

- 1.** Ler a instrução na memória que contém os dados, usando o valor do contador de programa como endereço;
- 2.** Ler um ou mais registradores, usando parte da instrução.

2. Conjunto de Instruções (Busca)

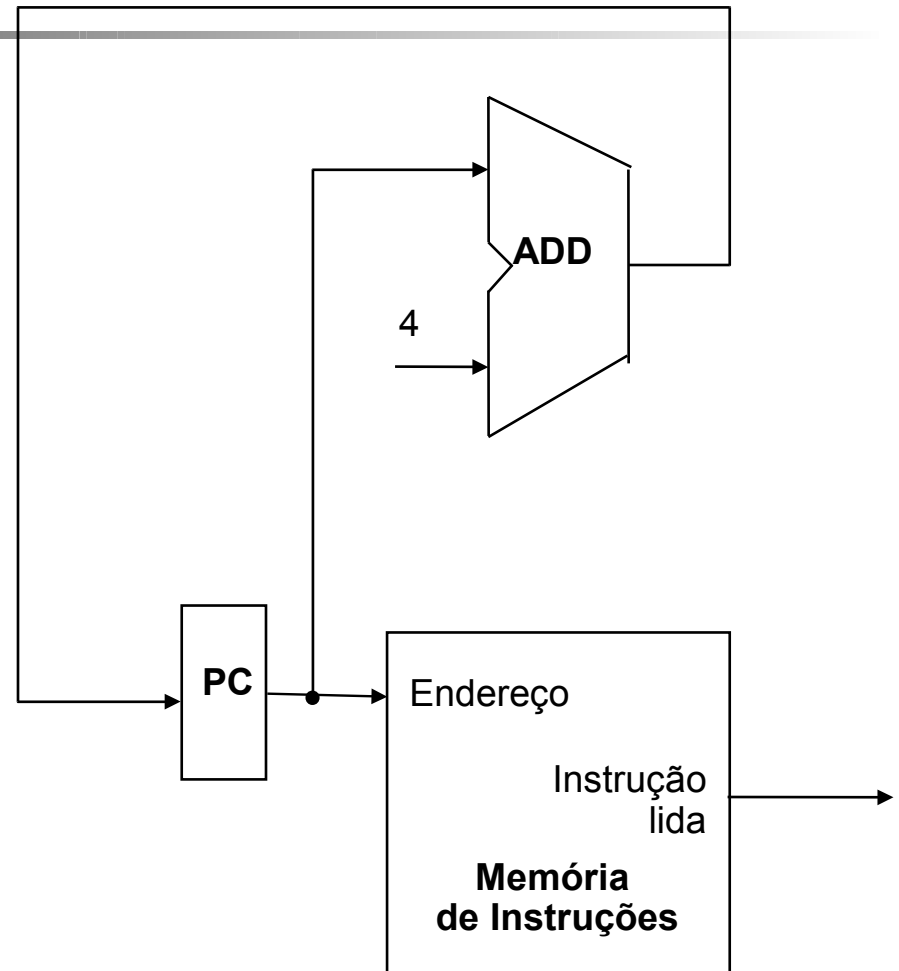
Três elementos são necessários para executar uma busca de instrução:

- a memória onde estão armazenadas as instruções
- o contador de programa (PC) para armazenar o endereço da instrução
- um somador para calcular o endereço da próxima



2. Conjunto de Instruções (Busca)

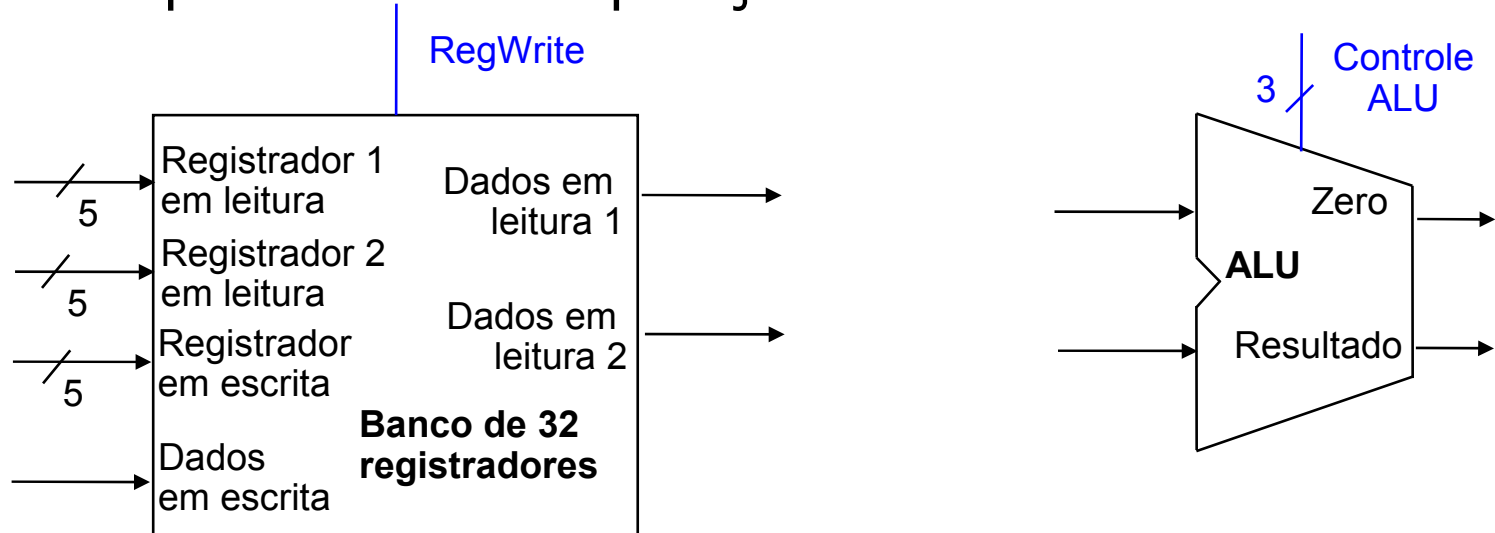
- O contador de programa contém o endereço da instrução em execução
- O endereço da próxima instrução é obtido pela soma de 4 posições ao contador de programa
- A instrução lida é usada por outras porções da parte operativa



2.1. Instruções Aritméticas

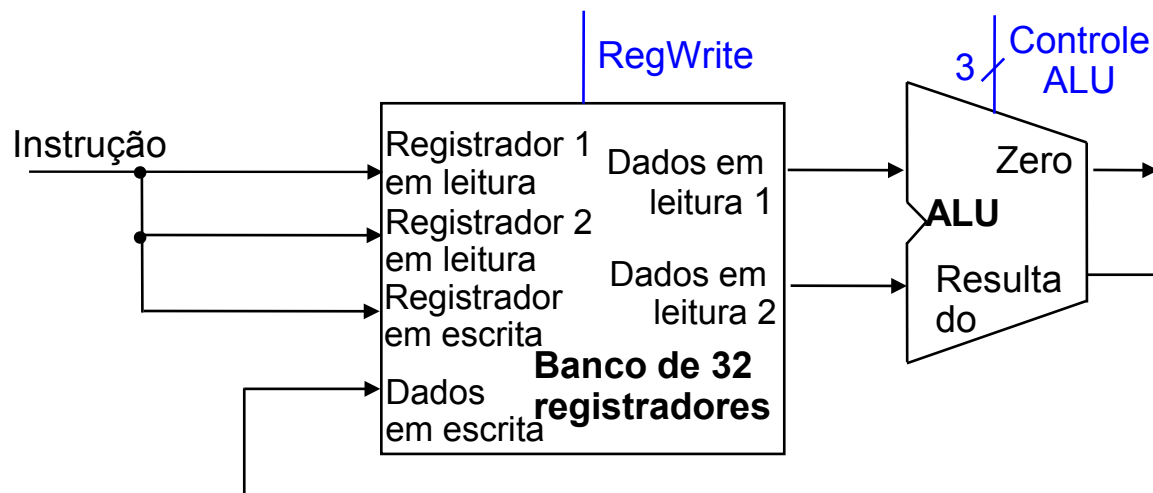
Dois elementos são necessários para a execução de operações aritméticas:

- Um banco de registradores para armazenar os operandos e o resultado das operações
- Uma Unidade Lógica/Aritmética (ALU) que será utilizada para realizar as operações



2.1. Instruções Aritméticas

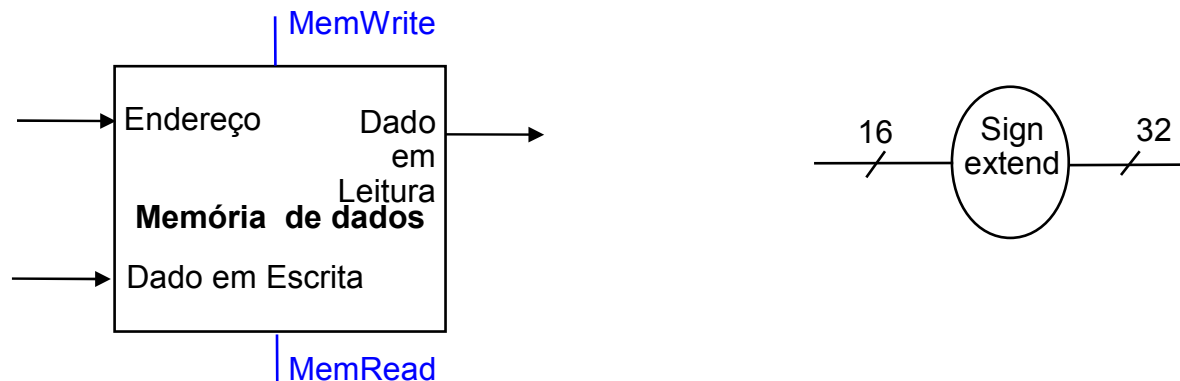
- A instrução (fornecida pelo hardware de busca de instruções) contém o endereço de três registradores
- Dois destes registradores são lidos e passados para a ALU realizar a operação
- O resultado é armazenado em um terceiro registrador
- O controle da ALU determina a operação que será realizada (a partir do campo FNCT da instrução)



2.2. Instruções de Acesso à Memória

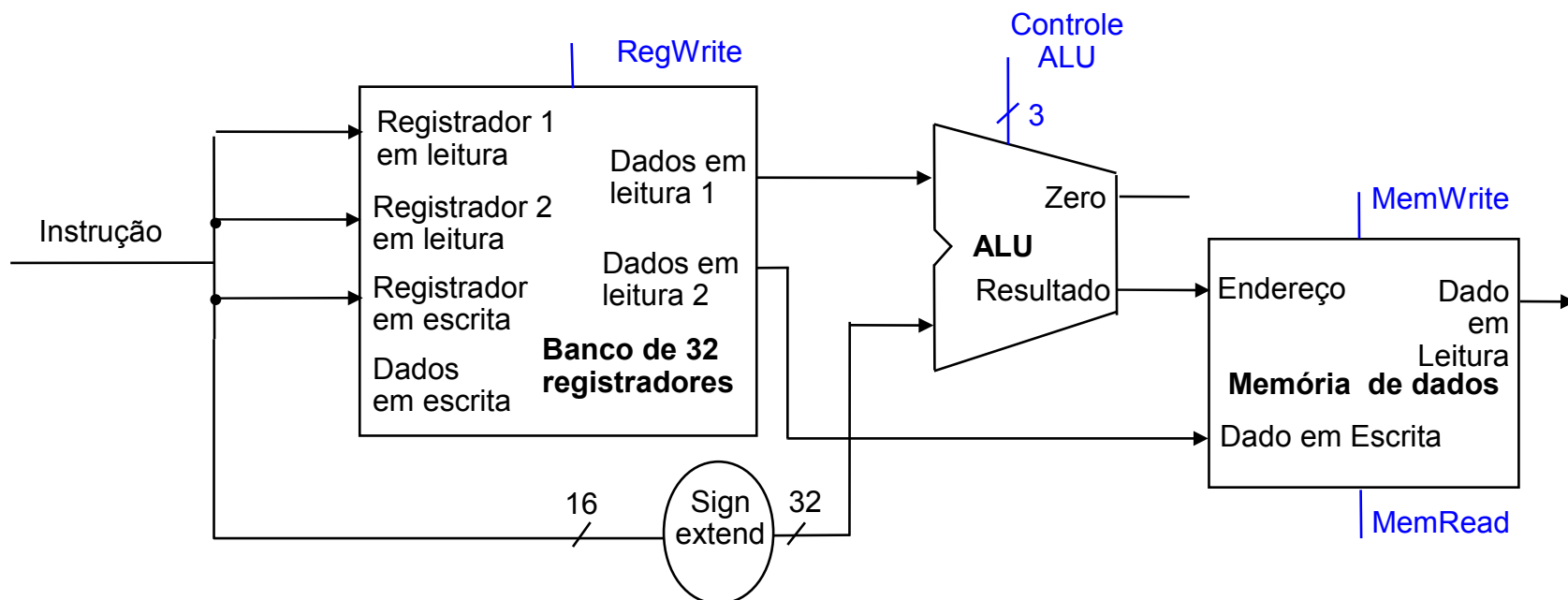
Para executar instruções de acesso à memória do tipo load e store são necessários:

- Uma memória de dados
- Um módulo de extensão de sinal (sign extend) para calcular números negativos e positivos em 32 bits a partir de sua versão 16 bits
- Um banco de registradores (já mostrado)
- Uma ALU (já mostrada)



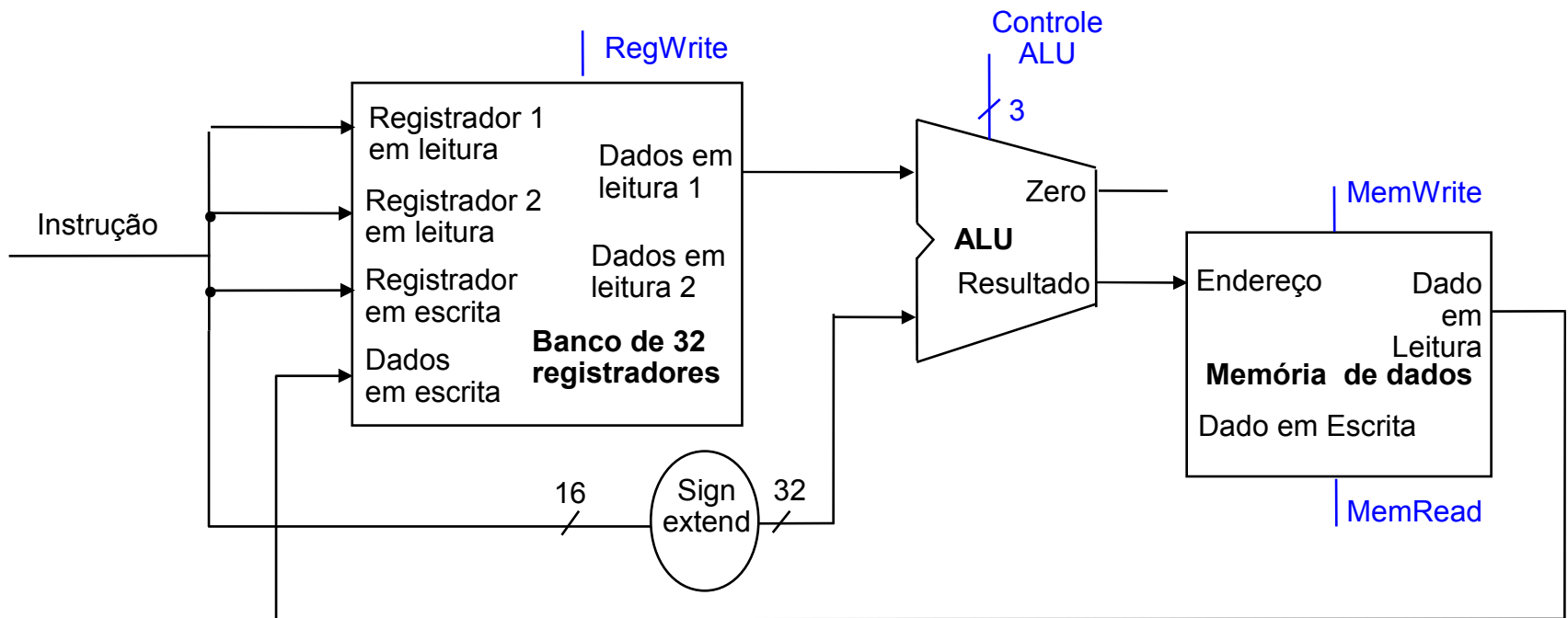
2.2. Instruções de Acesso à Memória (Escrita)

- O endereço de escrita é obtido pela soma de um registrador de base (registrador 1) com um deslocamento de 16 bits estendido para 32 bits
- O registrador 2 é escrito na memória



2.2. Instruções de acesso à memória (Leitura)

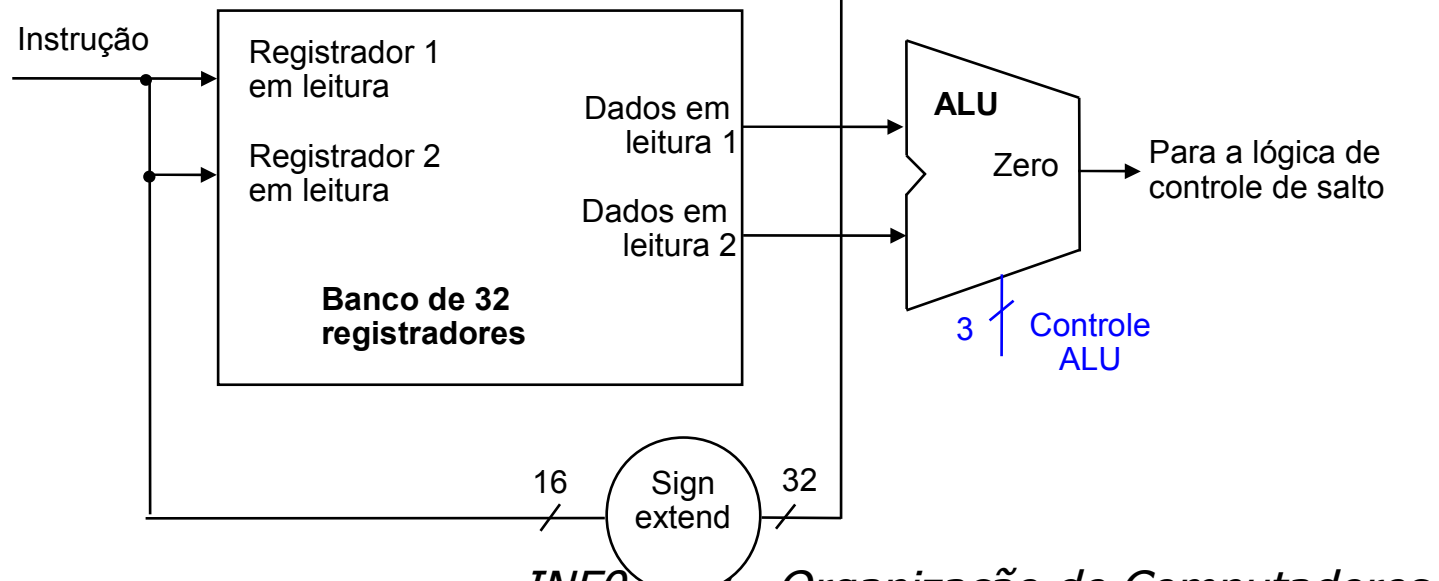
- O processo de leitura é semelhante ao de escrita
- A diferença básica é a existência de um caminho para escrever o valor lido no banco de registradores



2.3. Instruções de Desvio Condicional

Esquema apenas para a implementação de instruções do tipo beq (branch-on-equal)

- Destino é calculado pela soma de um deslocamento com o PC
- Salto é realizado se a diferença entre os registradores 1 e 2 for nula



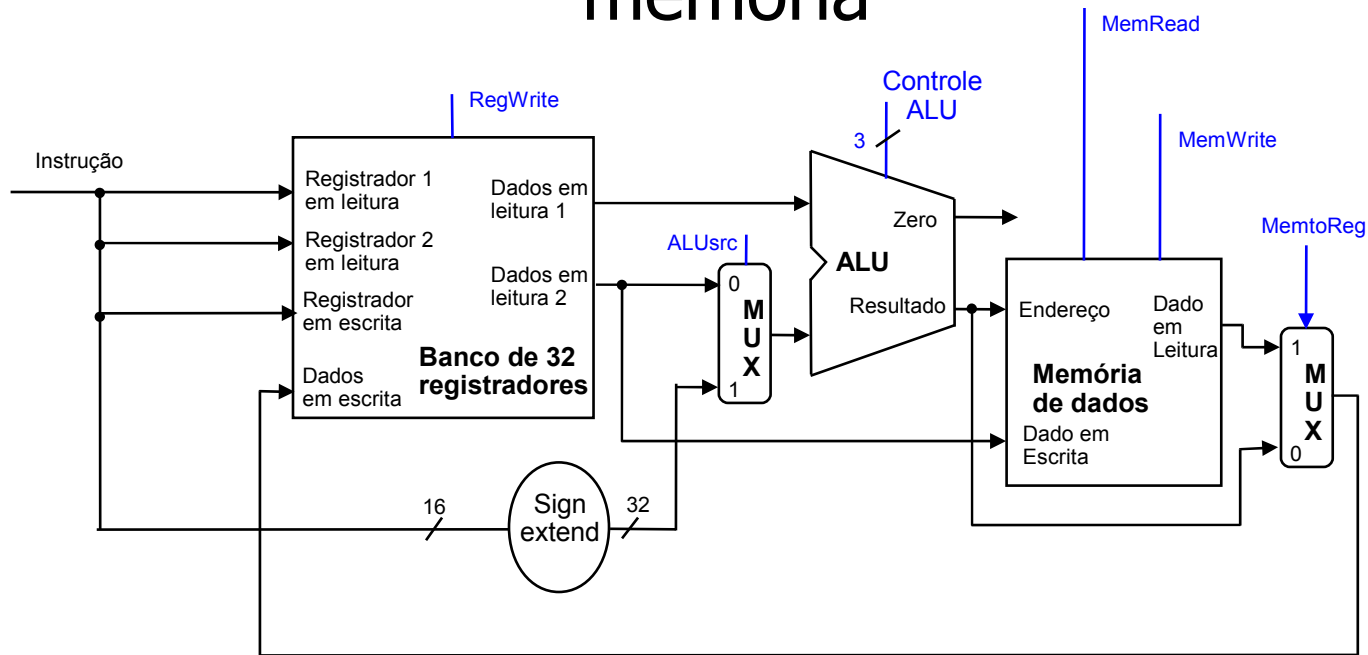


2.4. Combinando Instruções

- Os recursos para as instruções aritméticas e para as instruções de acesso à memória são bastante semelhantes. Seria possível combiná-los em um só?
- A segunda entrada da ALU ...
 - é um registrador, no caso de instruções aritméticas
 - é obtida a partir da extensão dos 16 bits inferiores da instrução, no caso de instruções de acesso à memória
- O valor a ser escrito no registrador destino ...
 - vem da saída da ALU no caso de uma operação aritmética
 - vem da memória no caso de uma instrução de acesso à memória
- Uma parte operativa combinada pode ser obtida através da inserção de multiplexadores nestes pontos

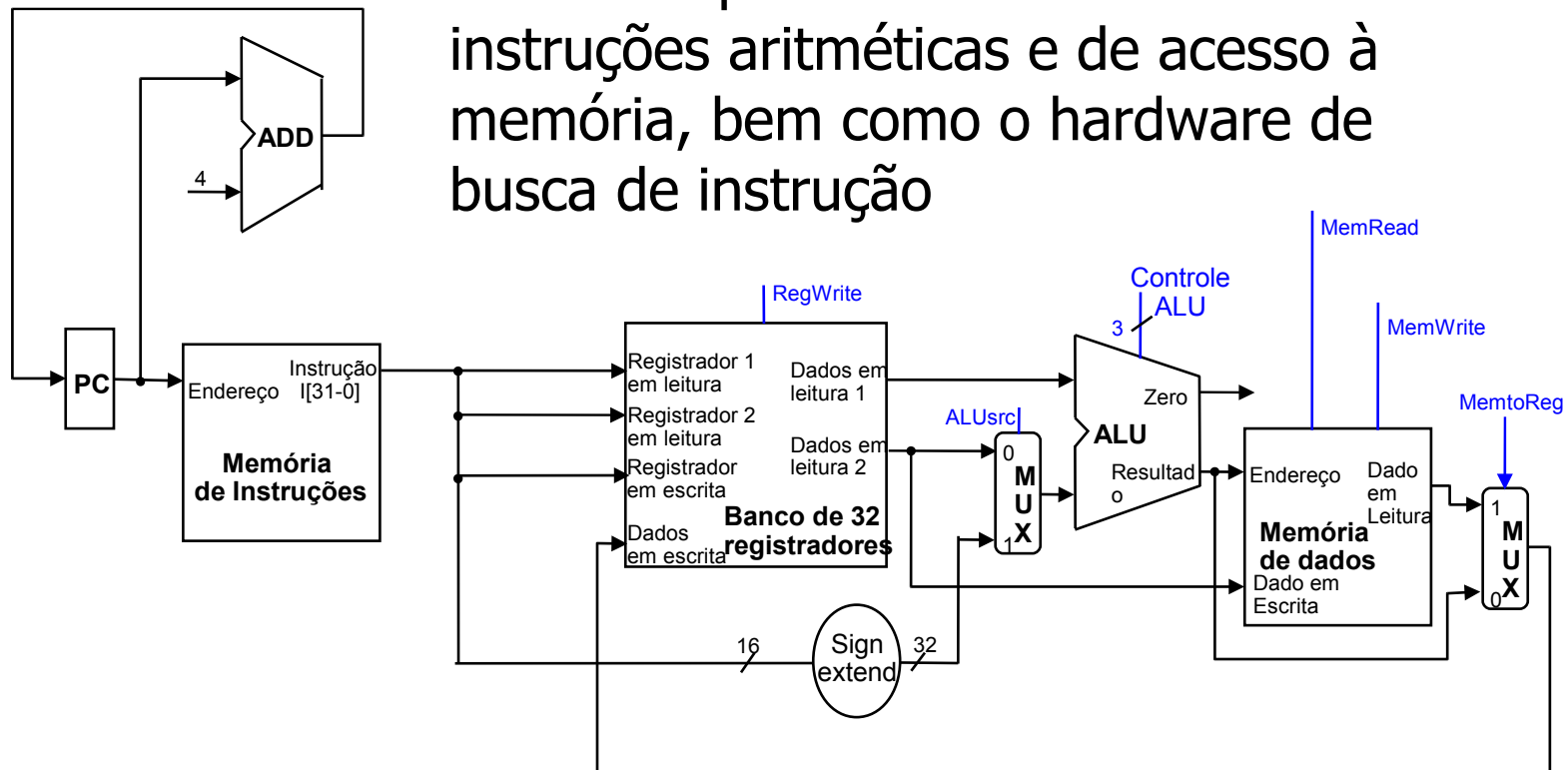
2.4. Combinando Instruções

Bloco operacional considerando instruções aritméticas e de acesso à memória

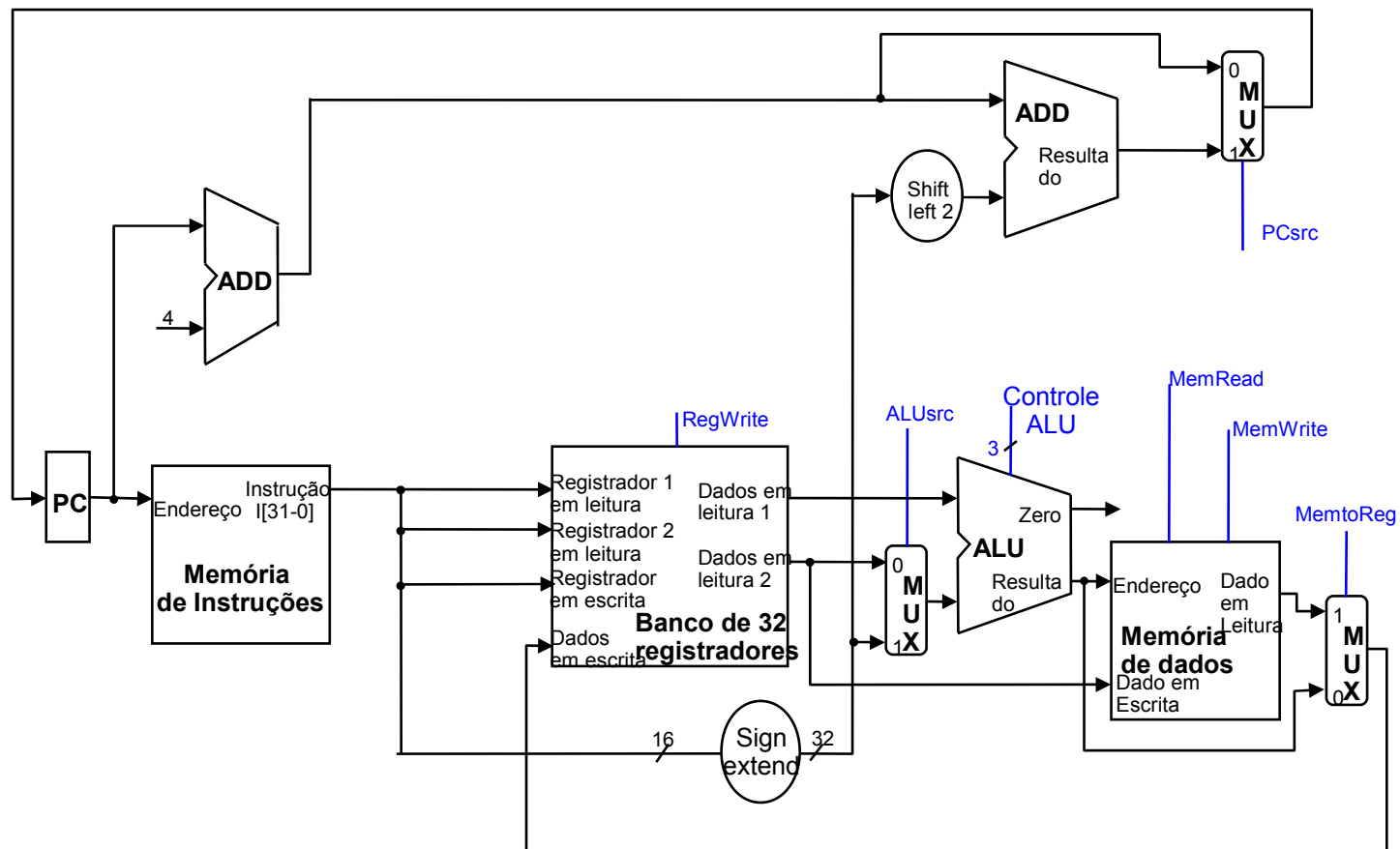


2.4. Combinando Instruções

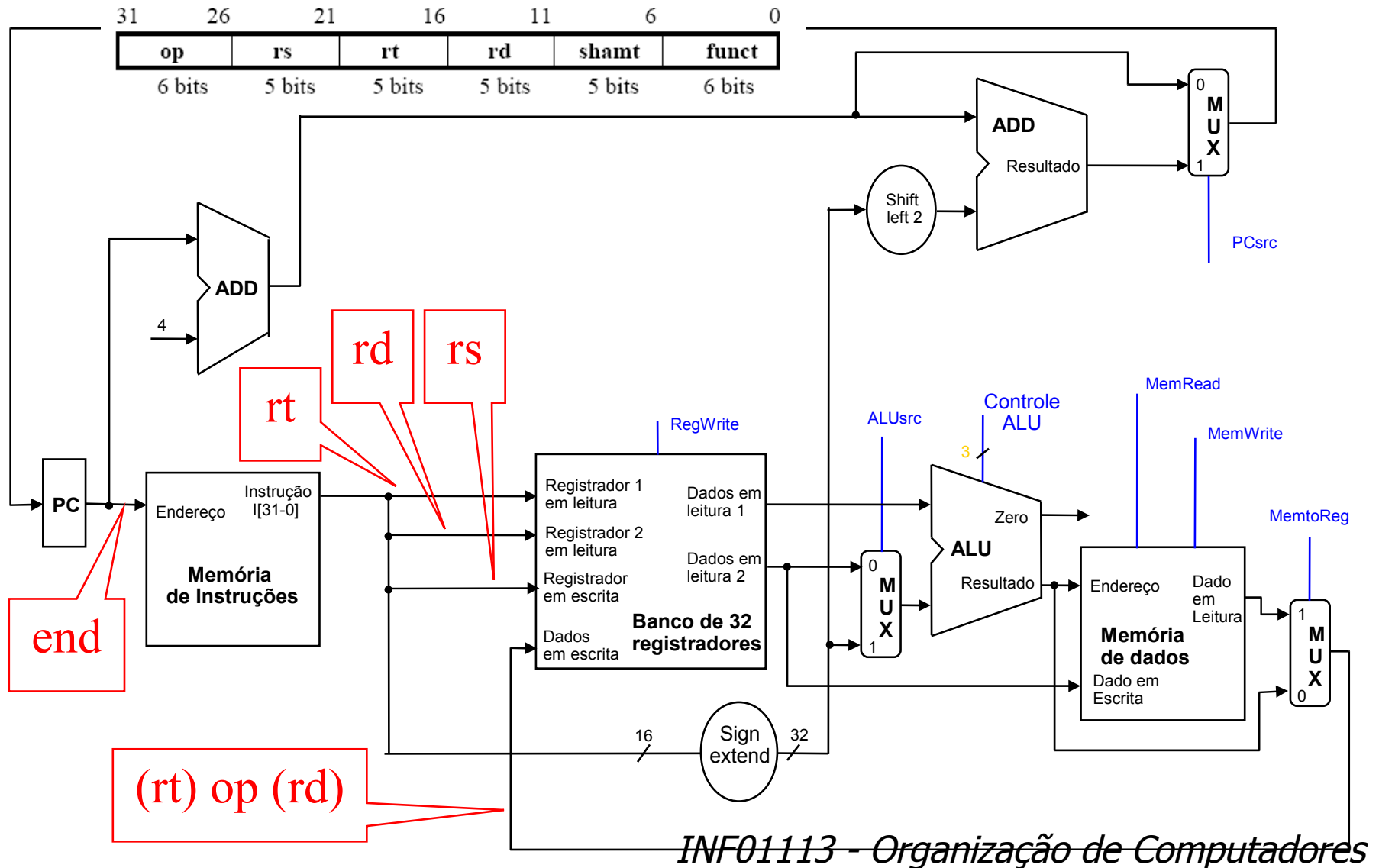
Bloco operacional considerando instruções aritméticas e de acesso à memória, bem como o hardware de busca de instrução



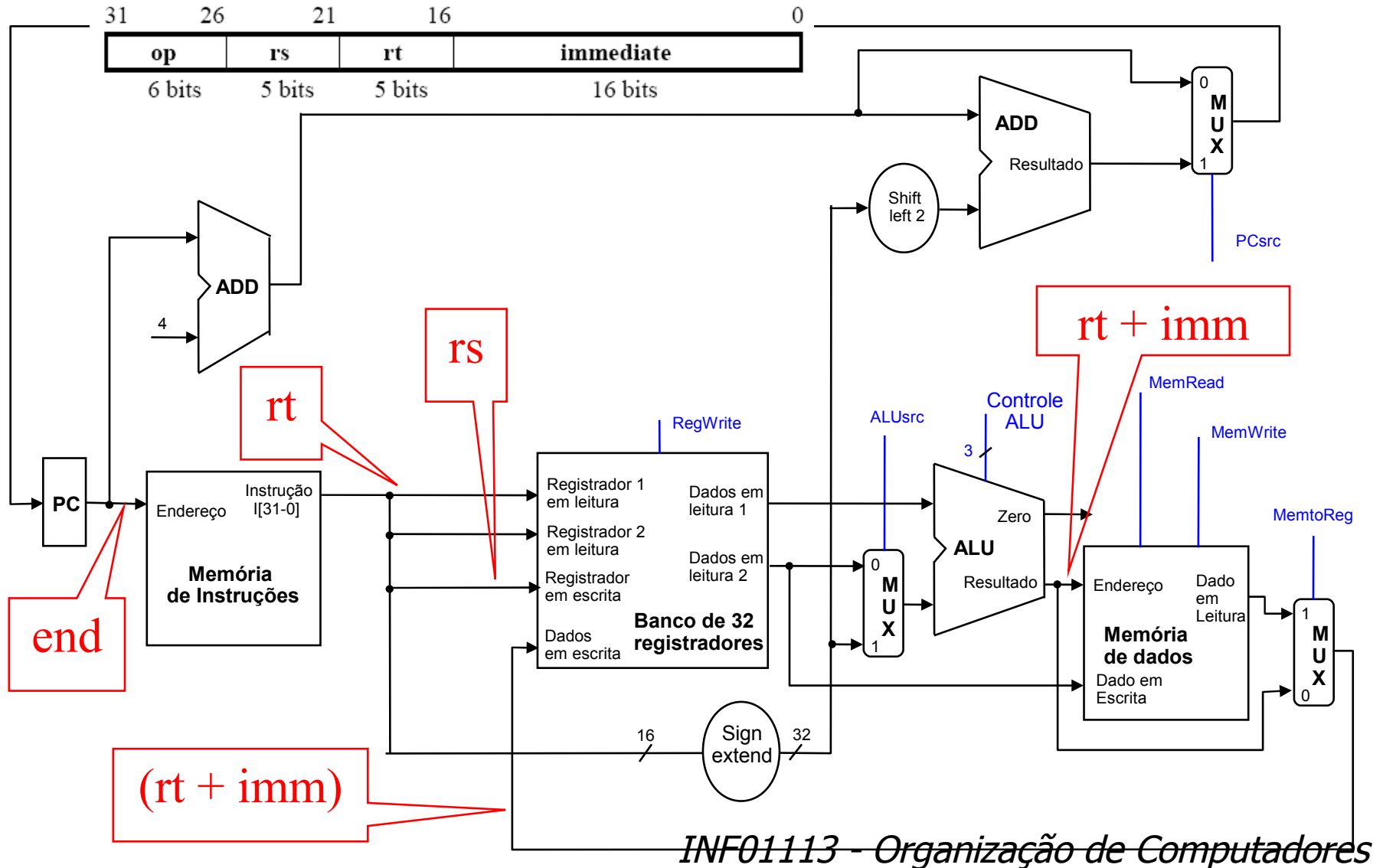
3. Bloco Operacional Completo



3. Bloco Op. (R-Tipo)



3. Bloco Op. (I-Tipo)



5. Cálculo do Período do Ciclo de Relógio

- Ao final de cada ciclo de relógio o PC é carregado com um novo valor
- mudança no valor do PC se propaga através de uma grande lógica combinacional
- Etapas de uma instrução exemplo:
memória de instruções => banco de registradores => ALU =>
memória de dados => banco de registradores
- período do ciclo de relógio deve ser maior do que máximo atraso de propagação através desta lógica combinacional



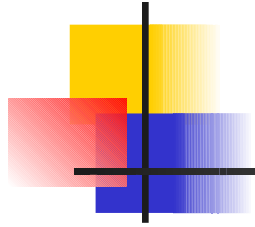
5. Cálculo do Período do Ciclo de Relógio

- memória de instruções => banco de registradores => ALU
=> memória de dados => banco de registradores
- Supondo os seguintes atrasos:
 - memórias: 1 ns
 - ALU: 0.5 ns
 - banco de registradores: 0.5 ns
 - somadores: 0.3 ns
 - demais componentes: atraso desprezível
- período do ciclo de relógio deve ser maior do que 3.5 ns (cerca de 285 MHz)



6. Resumo da Aula

- Caminho de dados da implementação mono-ciclo para o MIPS;
- Vantagens: Um ciclo por instrução, controle simples;
- Desvantagem: Tempo de ciclo muito grande;
- Usem o livro-texto! Ler e exercícios.
- Próxima Aula: Controle para implementação mono-ciclo



FIM