

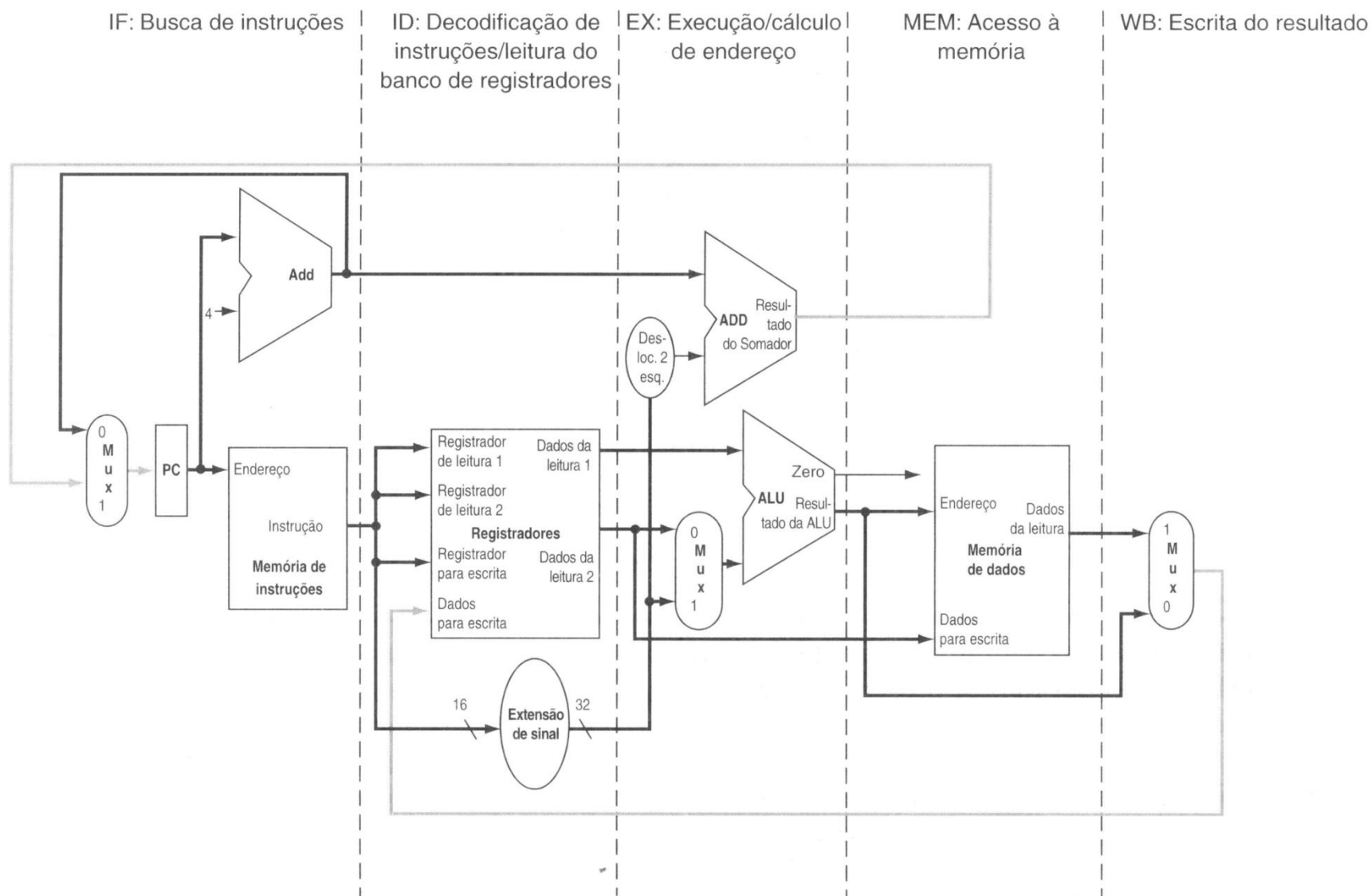
[Aula 11] Pipeline do MIPS 2 – Caminho de dados e controle

Prof. João F. Mari
joaof.mari@ufv.br

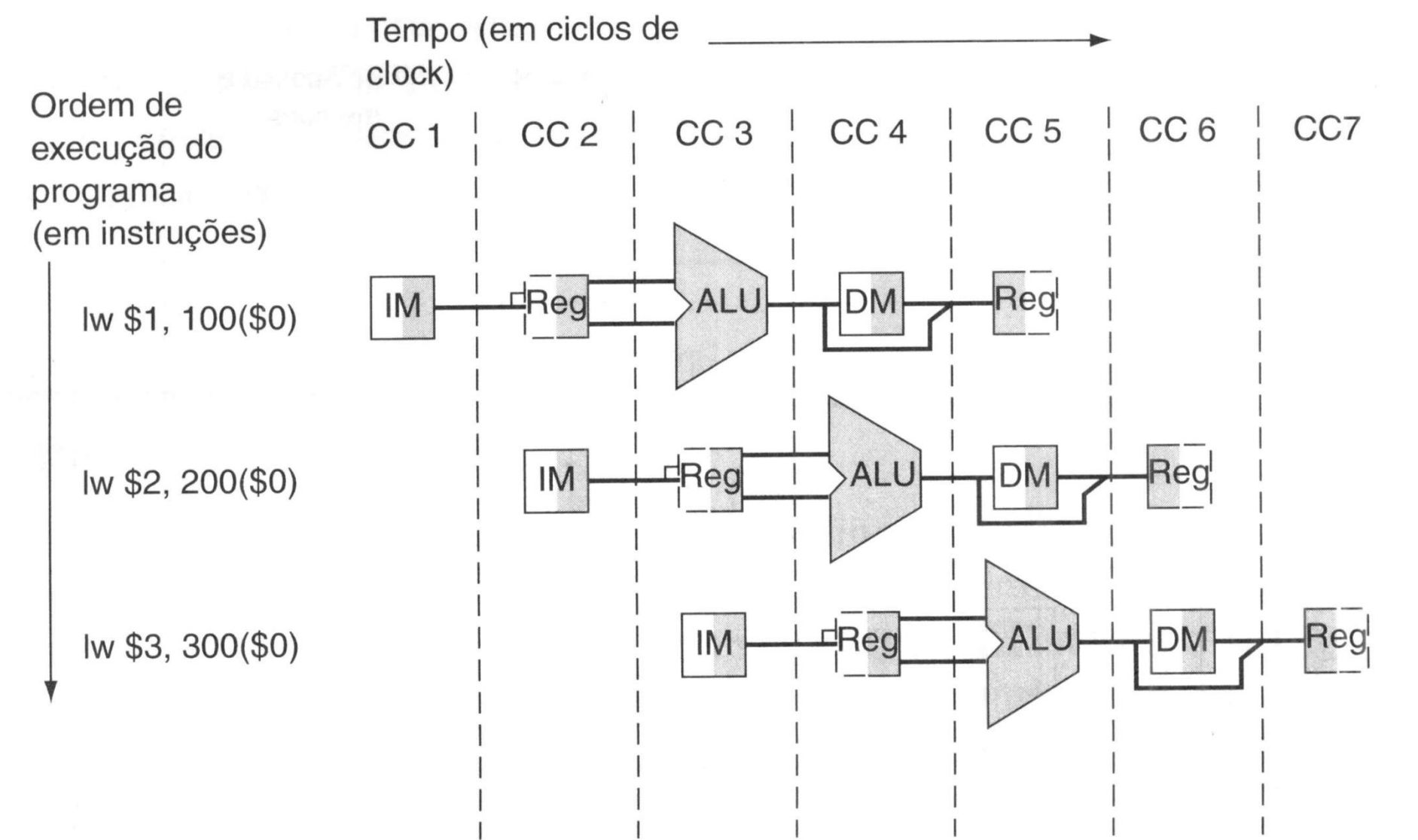
Divisão de instrução em cinco estágios

- Divisão de uma instrução em cinco estágios
 - Cinco níveis de pipeline
 - Cinco instruções executadas em paralelo (durante o mesmo ciclo de clock)
 - **IF (*Instruction Fetch*)**: Busca de instruções
 - **ID (*Instruction Decode*)**: Decodificação de instruções e leitura do banco de registradores
 - **EX**: Execução ou calculo de endereço
 - **MEM**: Acesso à memória de dados
 - **WB (*Write Back*)**: Escrita do resultado

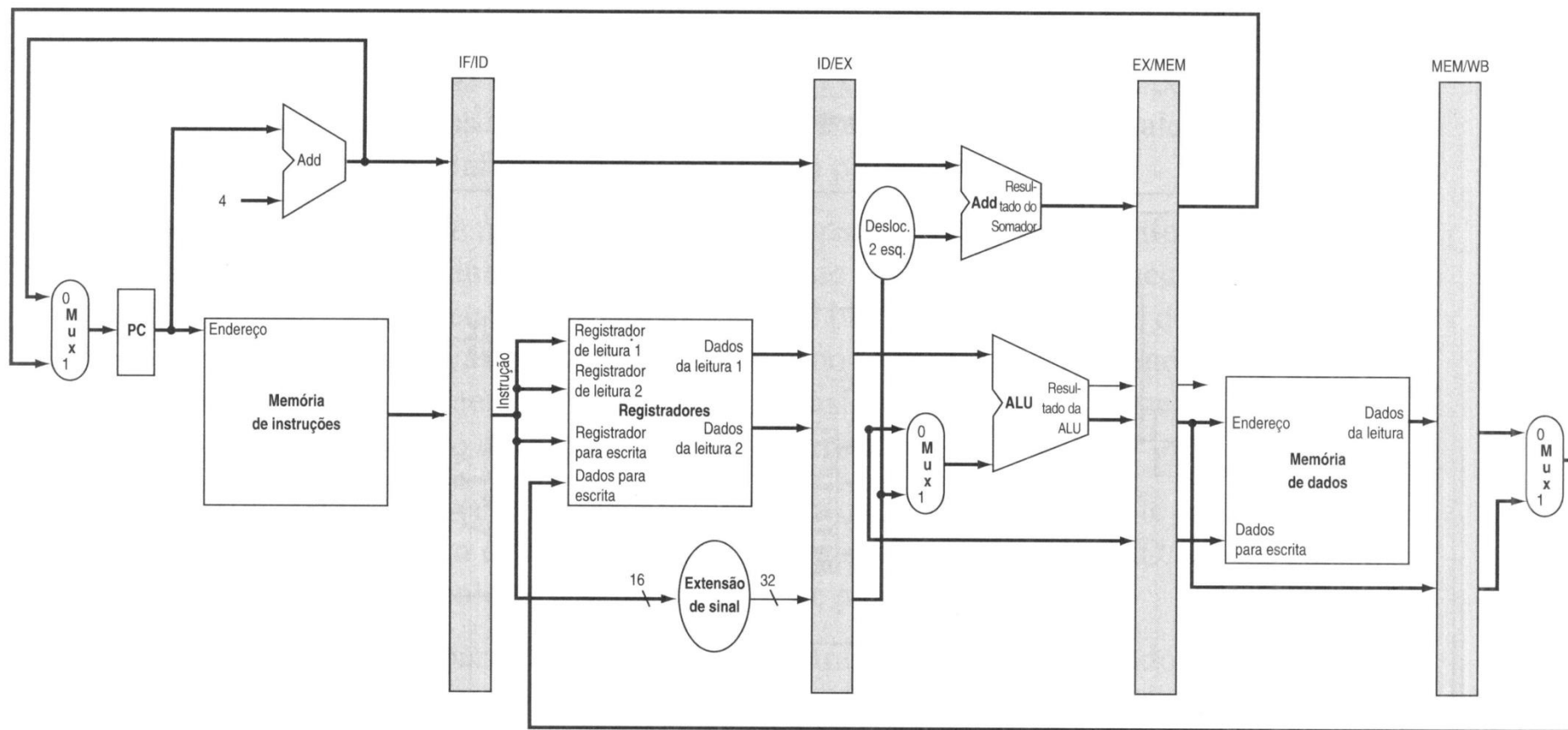
Divisão de instrução em cinco estágios



Divisão de instrução em cinco estágios



Divisão de instrução em cinco estágios

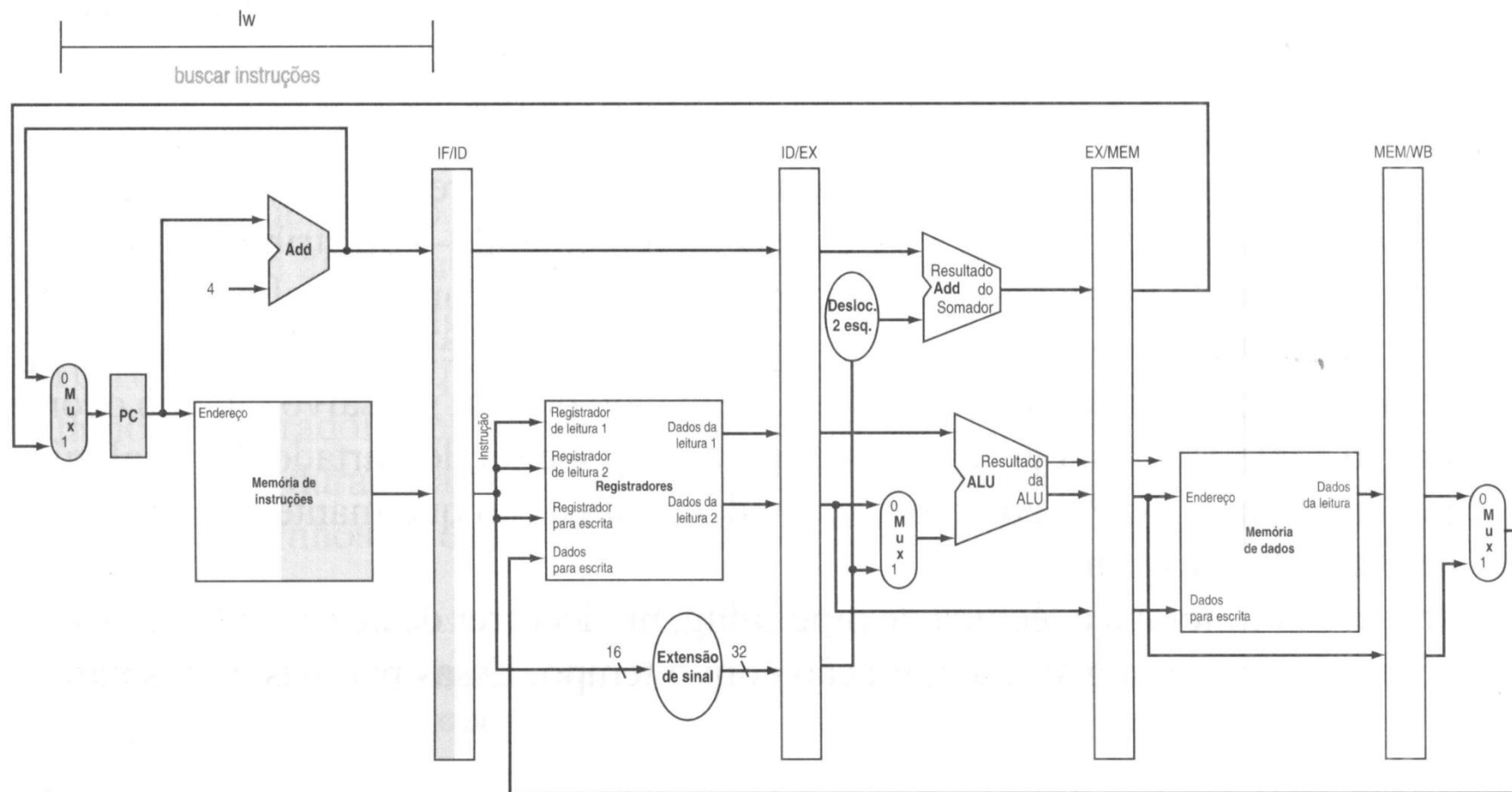


- IF/ID (64 bits) – instrução (32 bits) + endereço do PC++ (32 bits)
- ID/EX (128 bits) – PC++ (32 bits) + Dado da leitura 1 (32 bits) + Dado da leitura 2 (32 bits) + Offset estendido (32 bits)
- EX/MEM (97 bits) – PC++ + desvio (32 bits) + Flag zero (1 bit) + Resultado da ALU (32 bits) + Dado da leitura 2 (32 bits)
- MEM/WB (64 bits) – Dados lidos da memória (32 bits) + Resultado da ALU (32 bits)
 - *Os registradores ainda serão expandidos no futuro!!!*

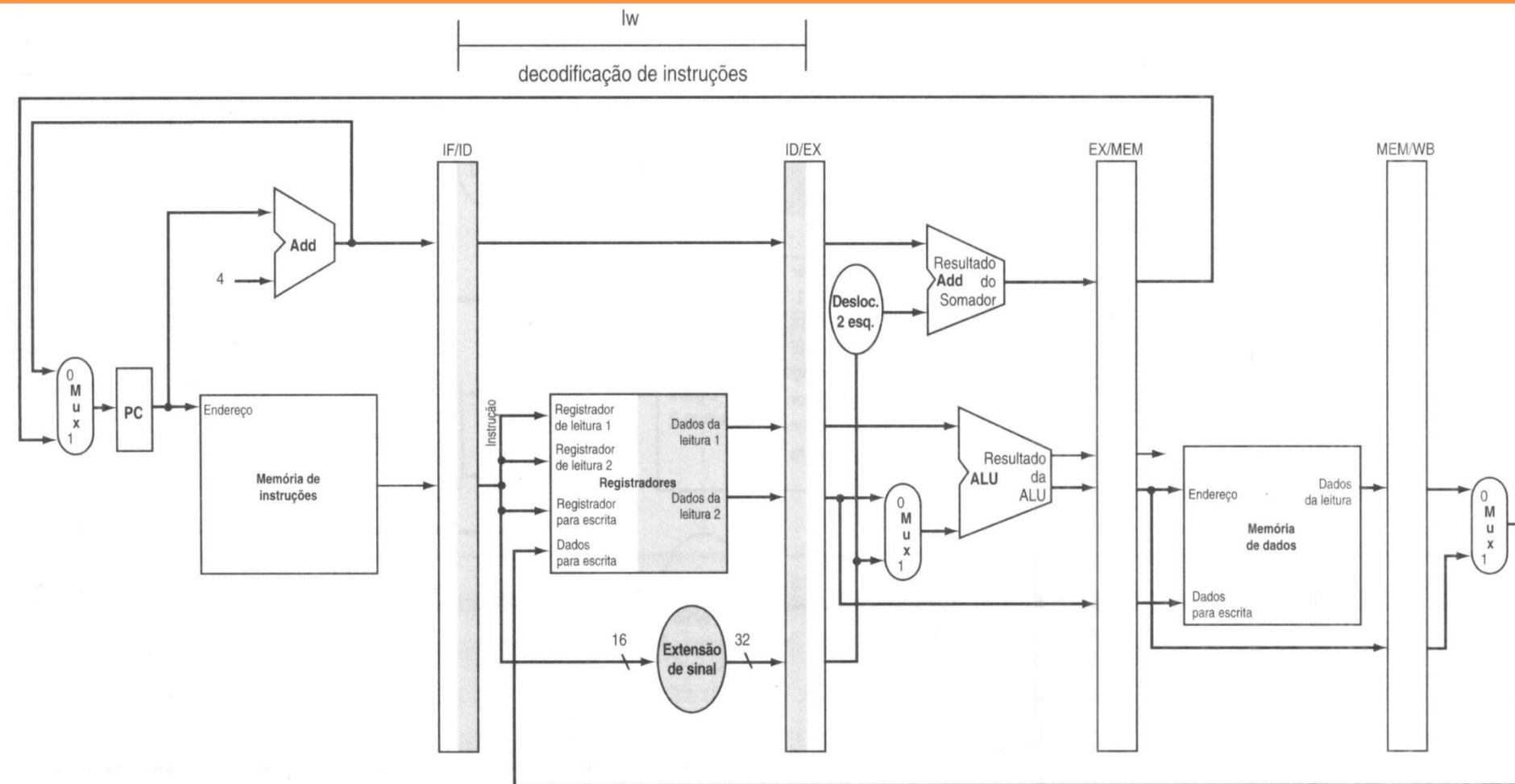
Instrução LW (load word)

- Etapa 1:
 - **IF** - Busca da instrução
- Etapa 2:
 - **ID** - Decodificação de instruções e leitura do banco de registradores
- Etapa 3:
 - **EX** - Execução ou calculo de endereço
- Etapa 4:
 - **MEM** - Acesso a memória
- Etapa 5:
 - **WB** - Escrita de resultado

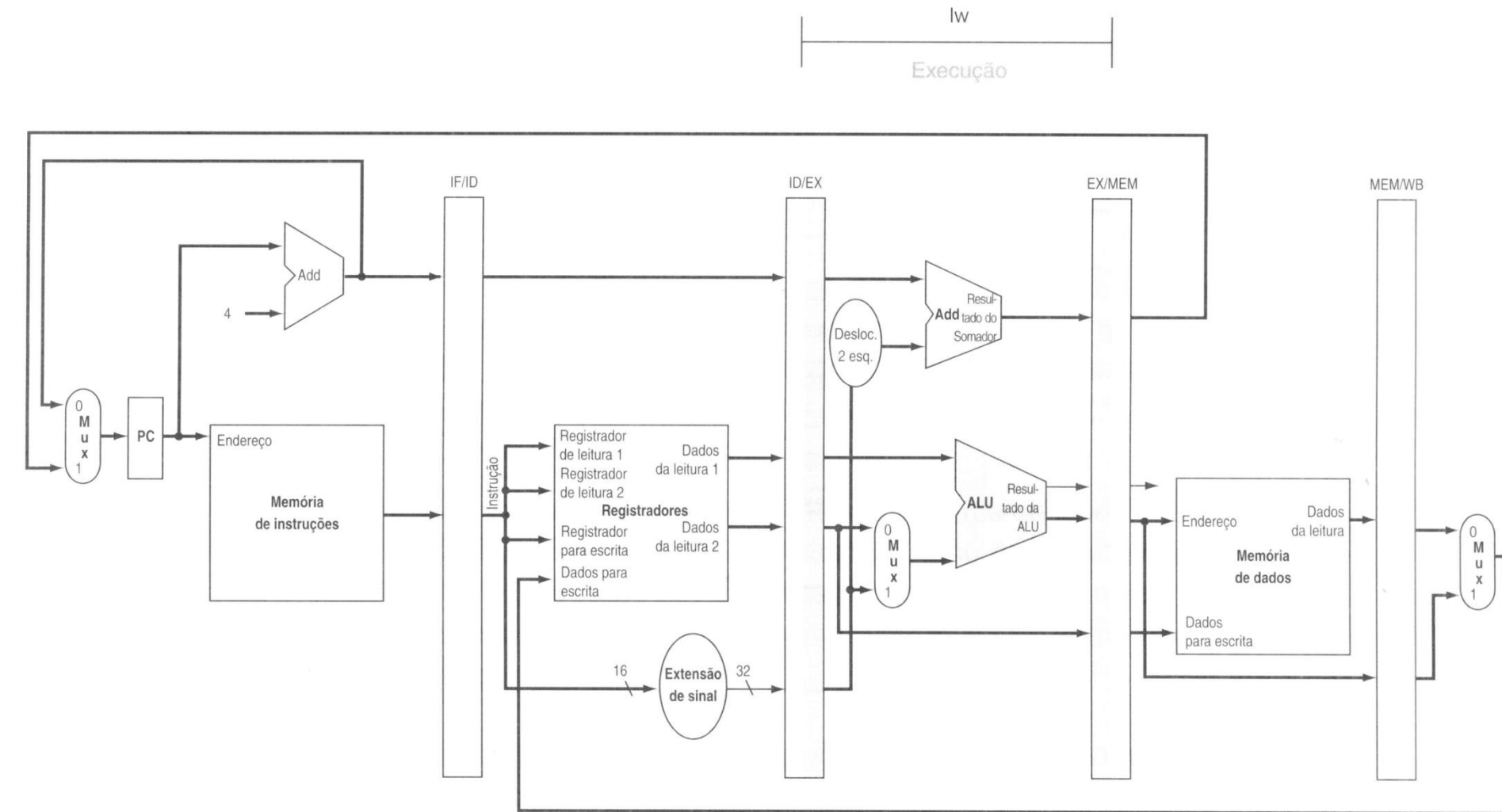
Etapa 1: IF – Busca da instrução



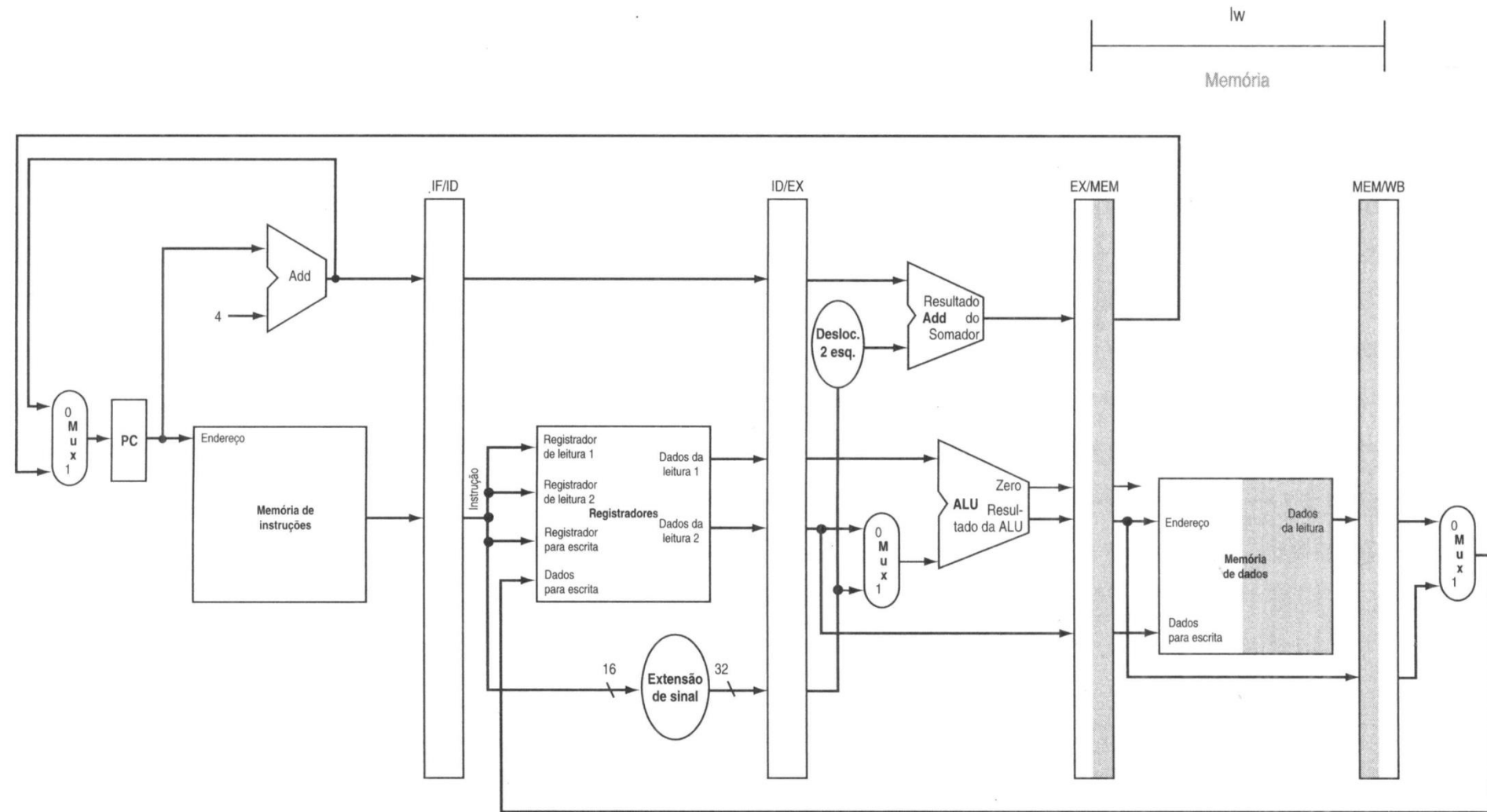
Etapa 2: ID – Decodificação da instrução



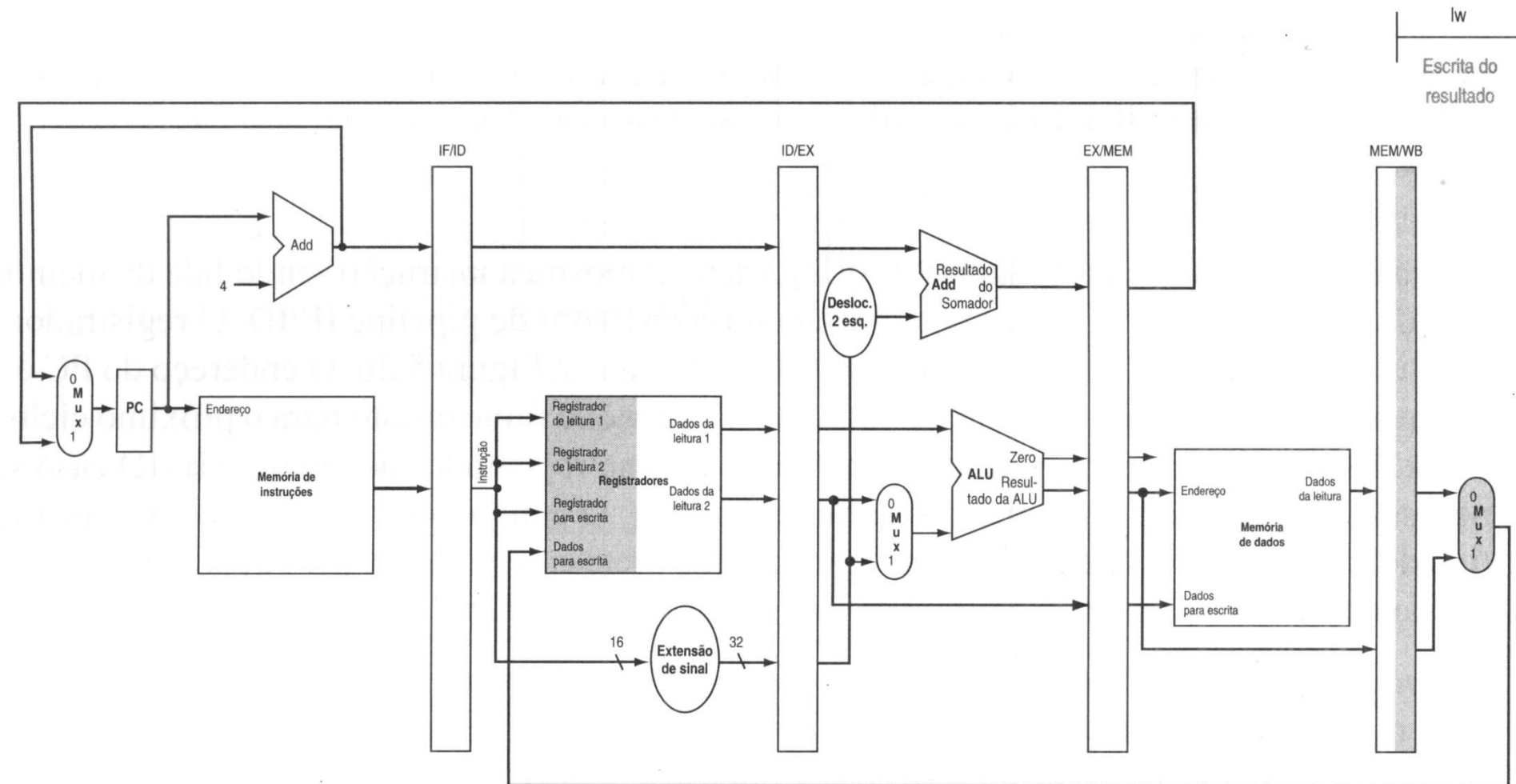
Etapa 3: EX – Execução da instrução (LW)



Etapa 4: MEM – Acesso à memória (LW)



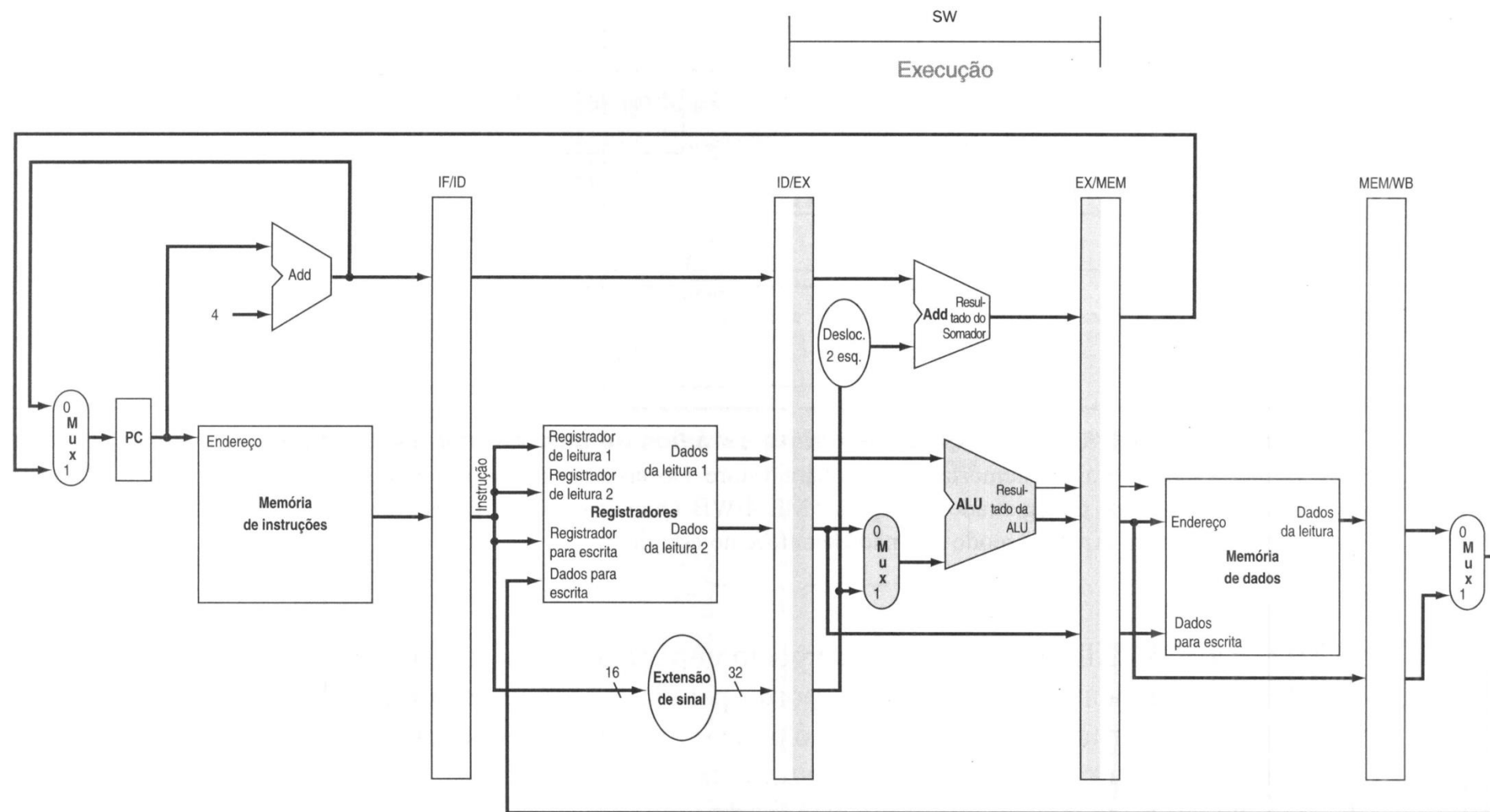
Etapa 5: WB – Escreve nos registradores (LW)



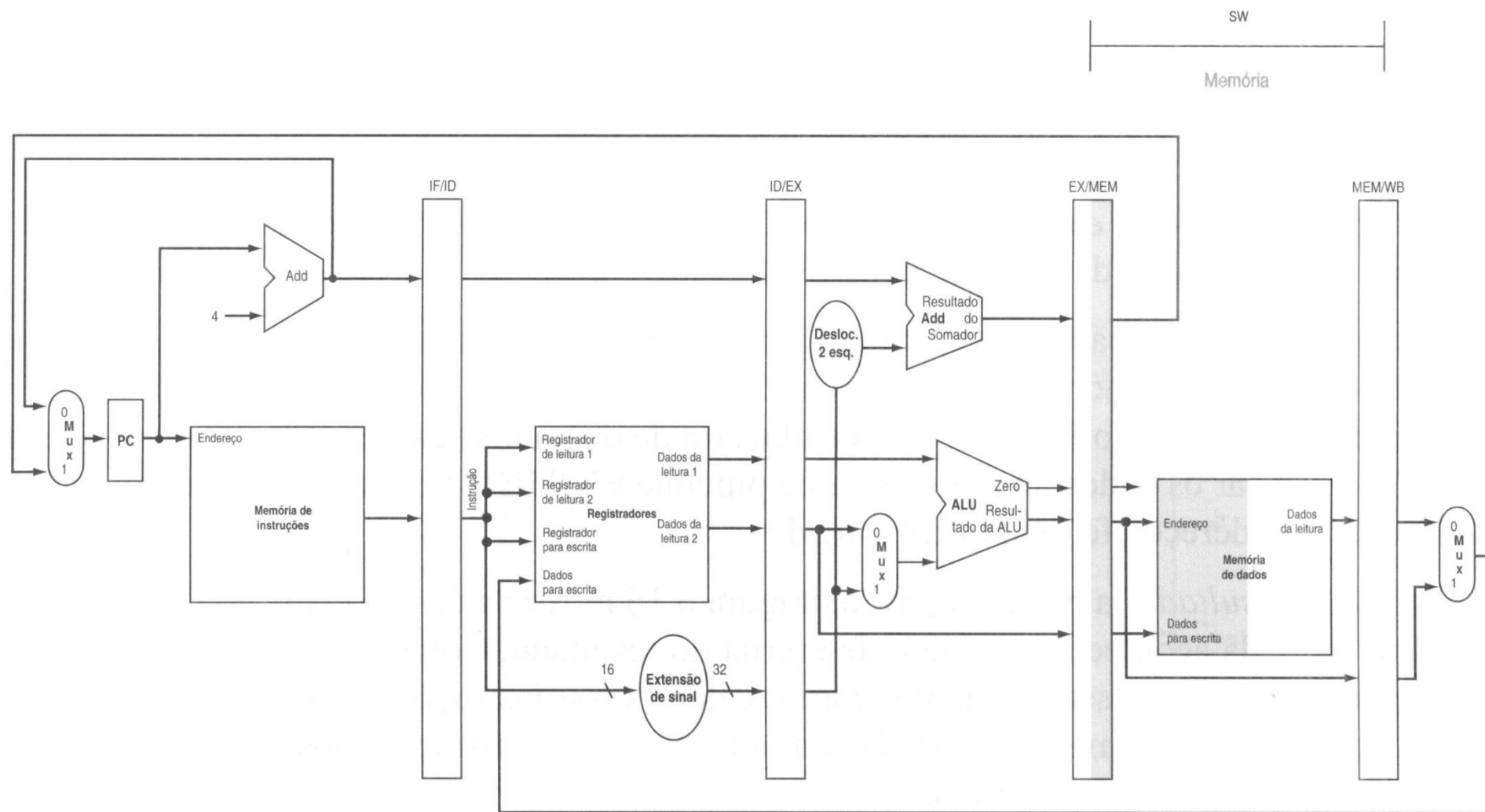
Store Word (SW)

- Etapa 1:
 - **IF** - Busca de instruções
- Etapa 2:
 - **ID** - Decodificação de instruções e leitura do banco de registradores
- Etapa 3:
 - **EX** - Execução ou calculo de endereço
- Etapa 4:
 - **MEM** - Acesso a memória
- Etapa 5:
 - **WB** - Escrita de resultado

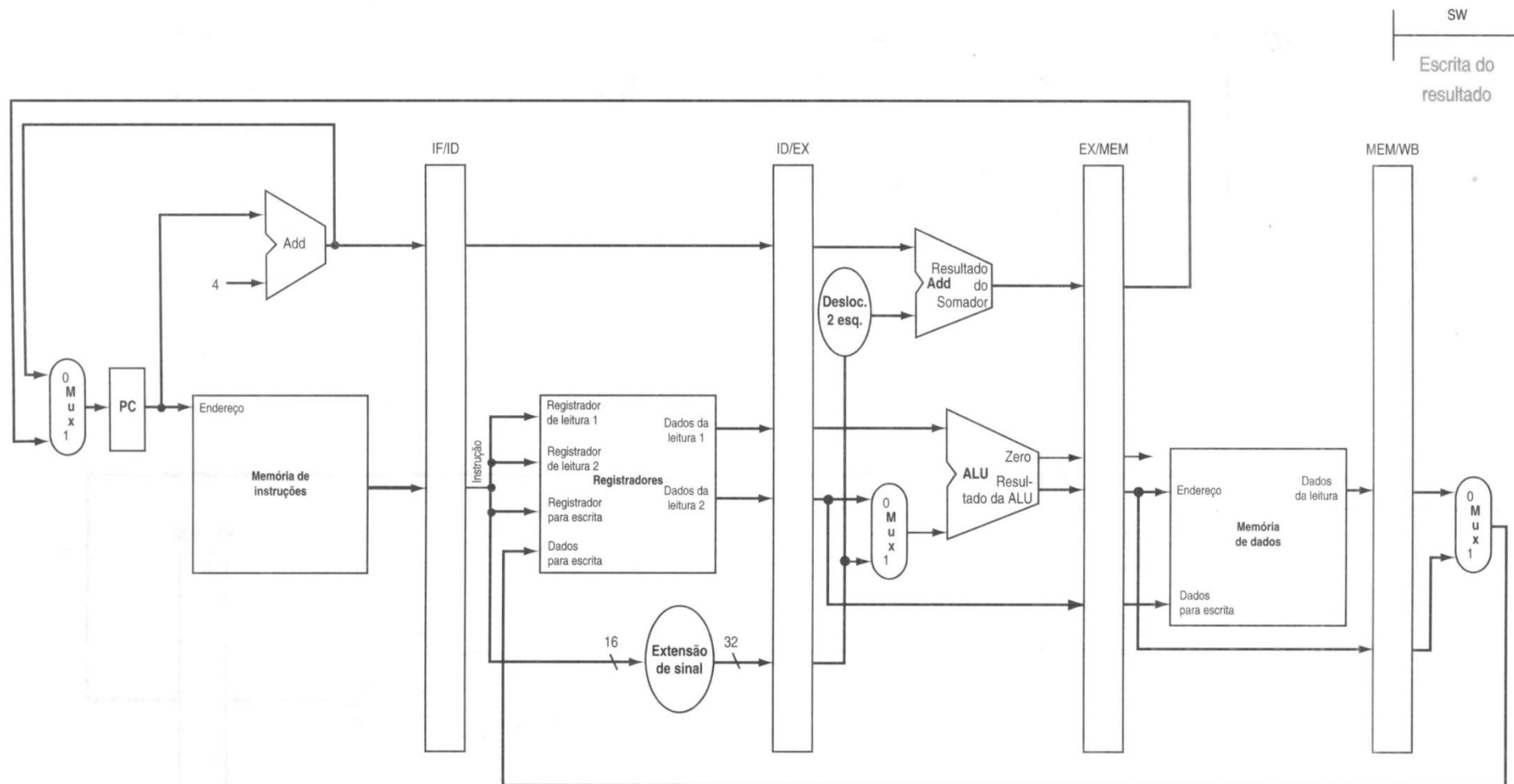
Etapa 3: EX – Execução (SW)



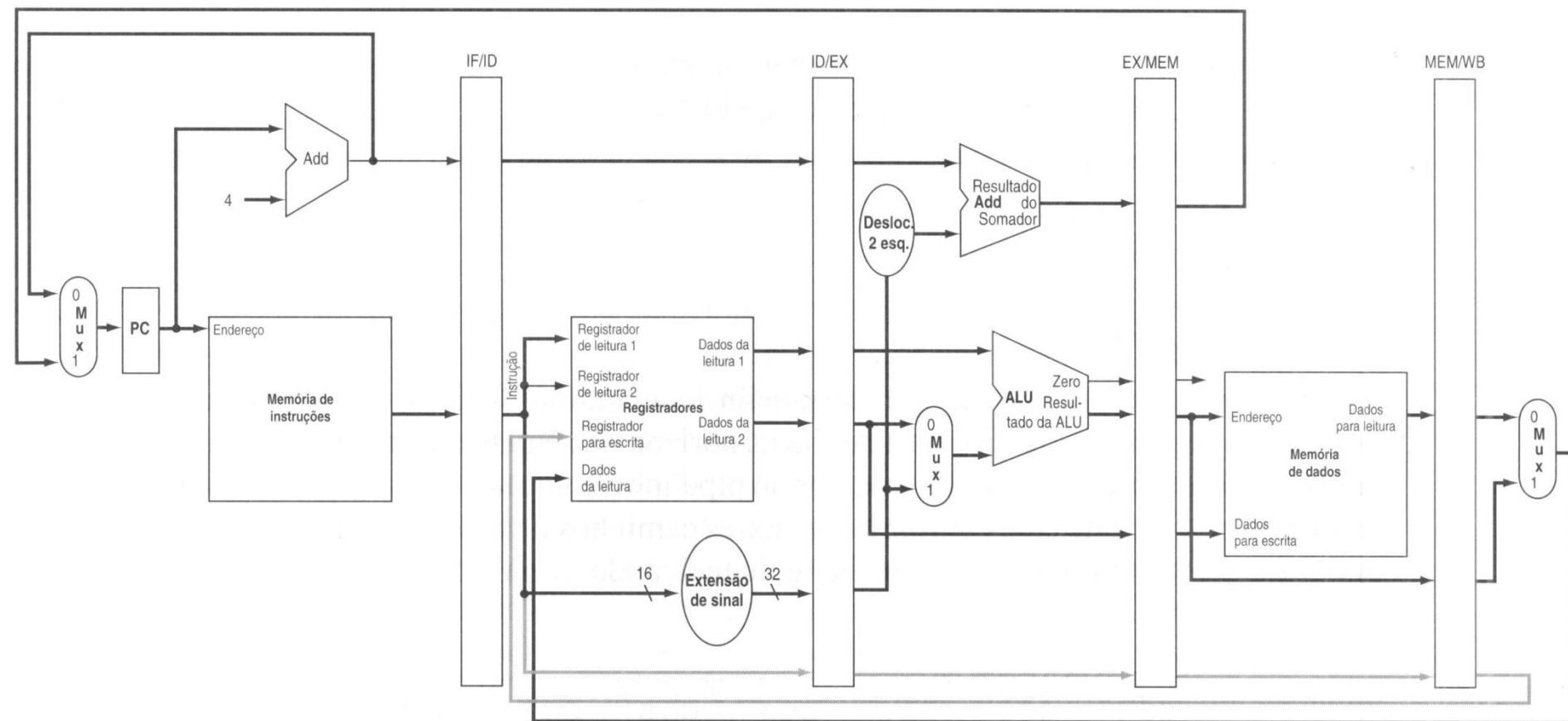
Etapa 4: MEM – Acesso à memória (SW)



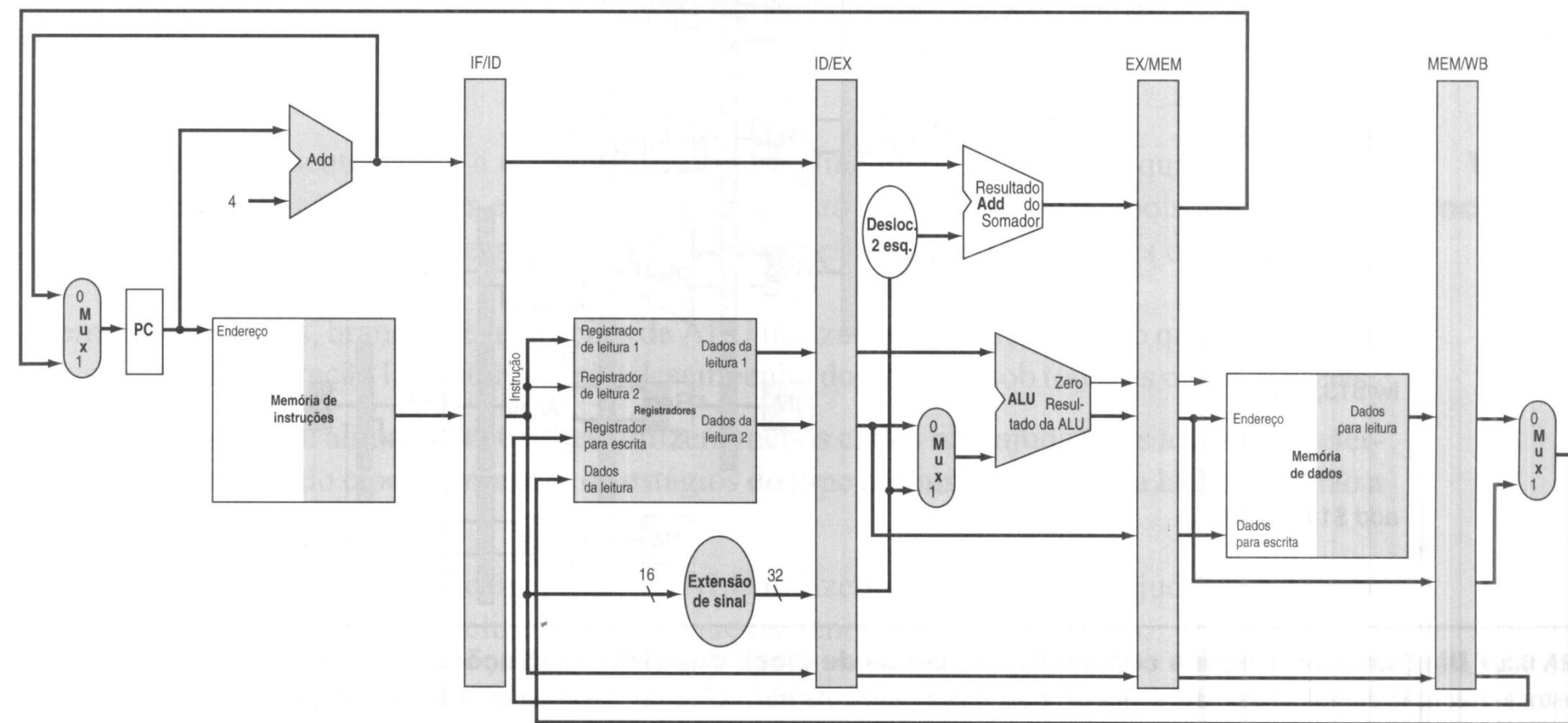
Etapa 5: WB – Escrita nos registradores (SW)



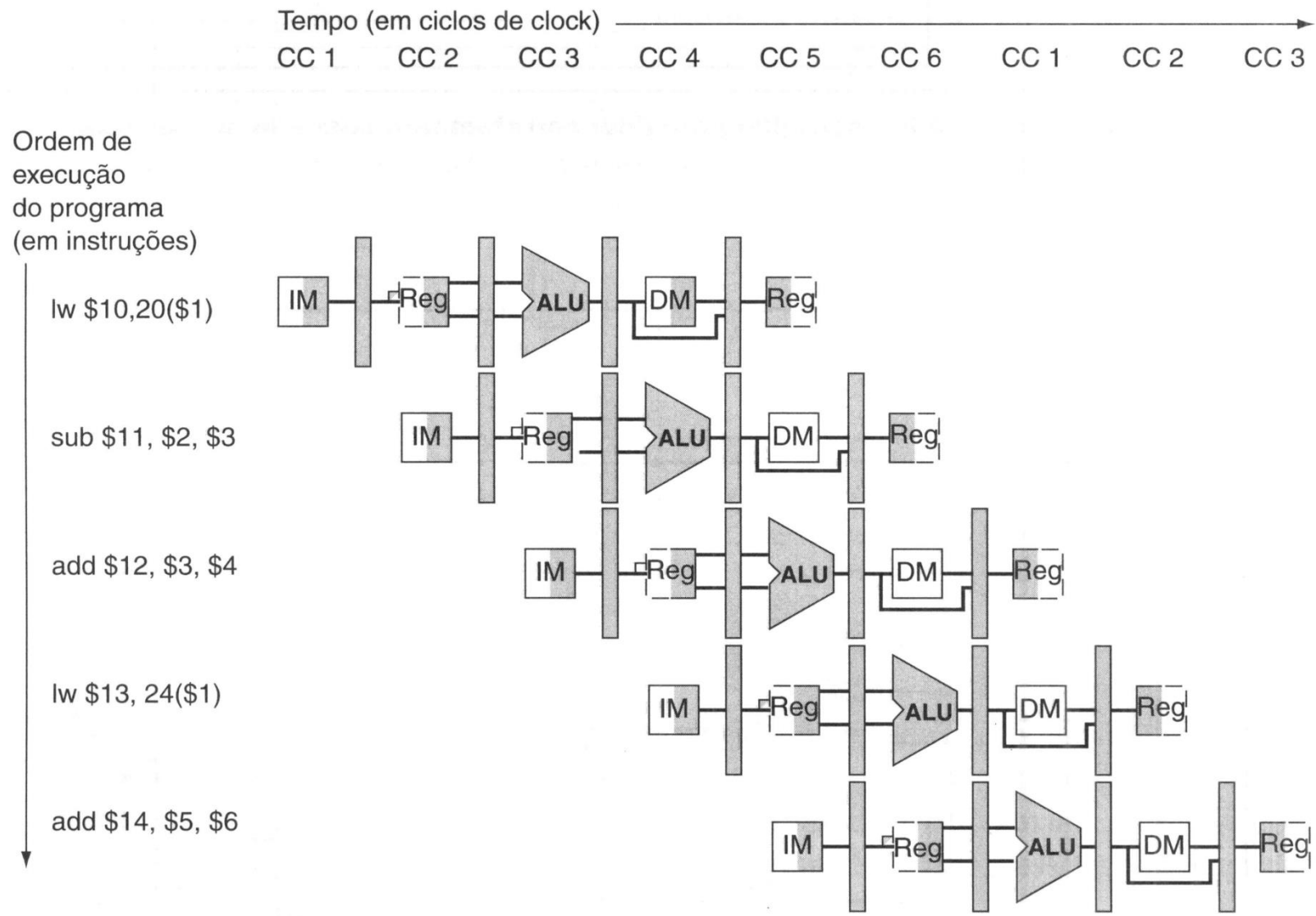
Caminho de dados corrigido – Load Word



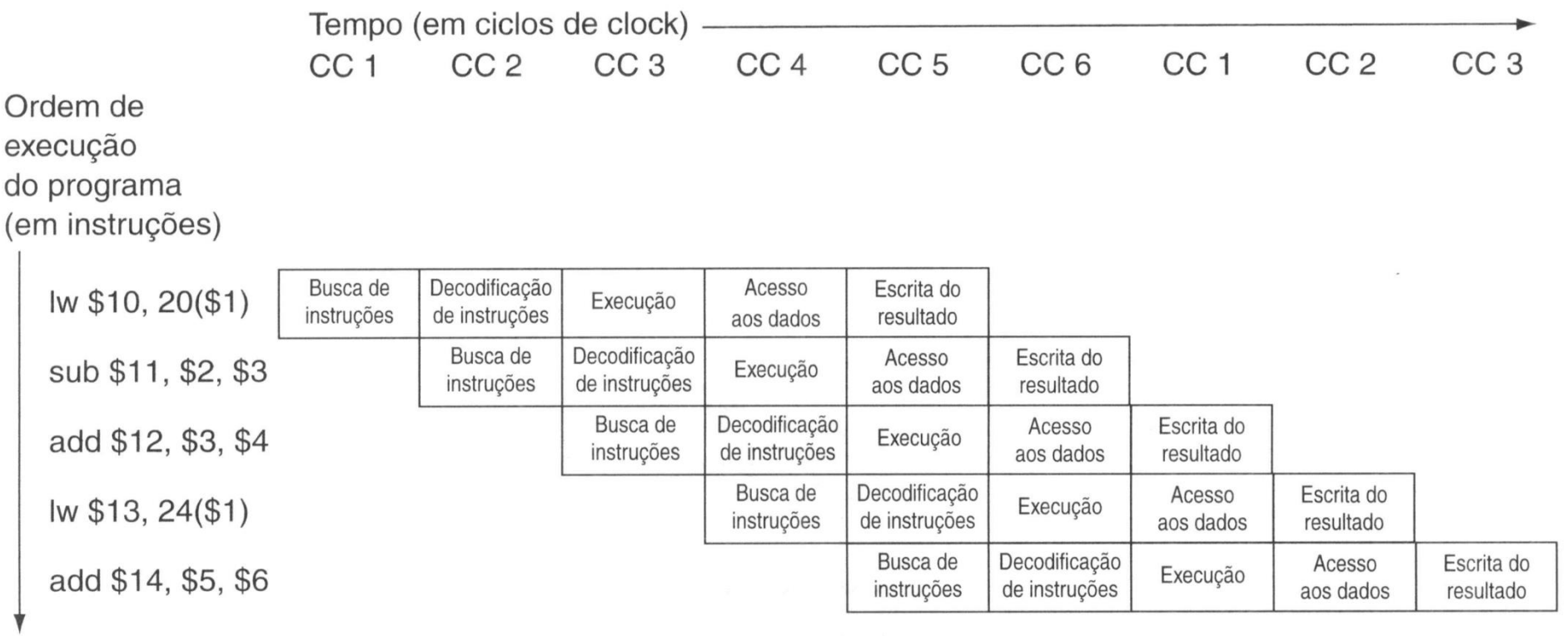
Partes do caminho de dados usado pela instrução LW (cinco estágios)



Representando pipelines graficamente

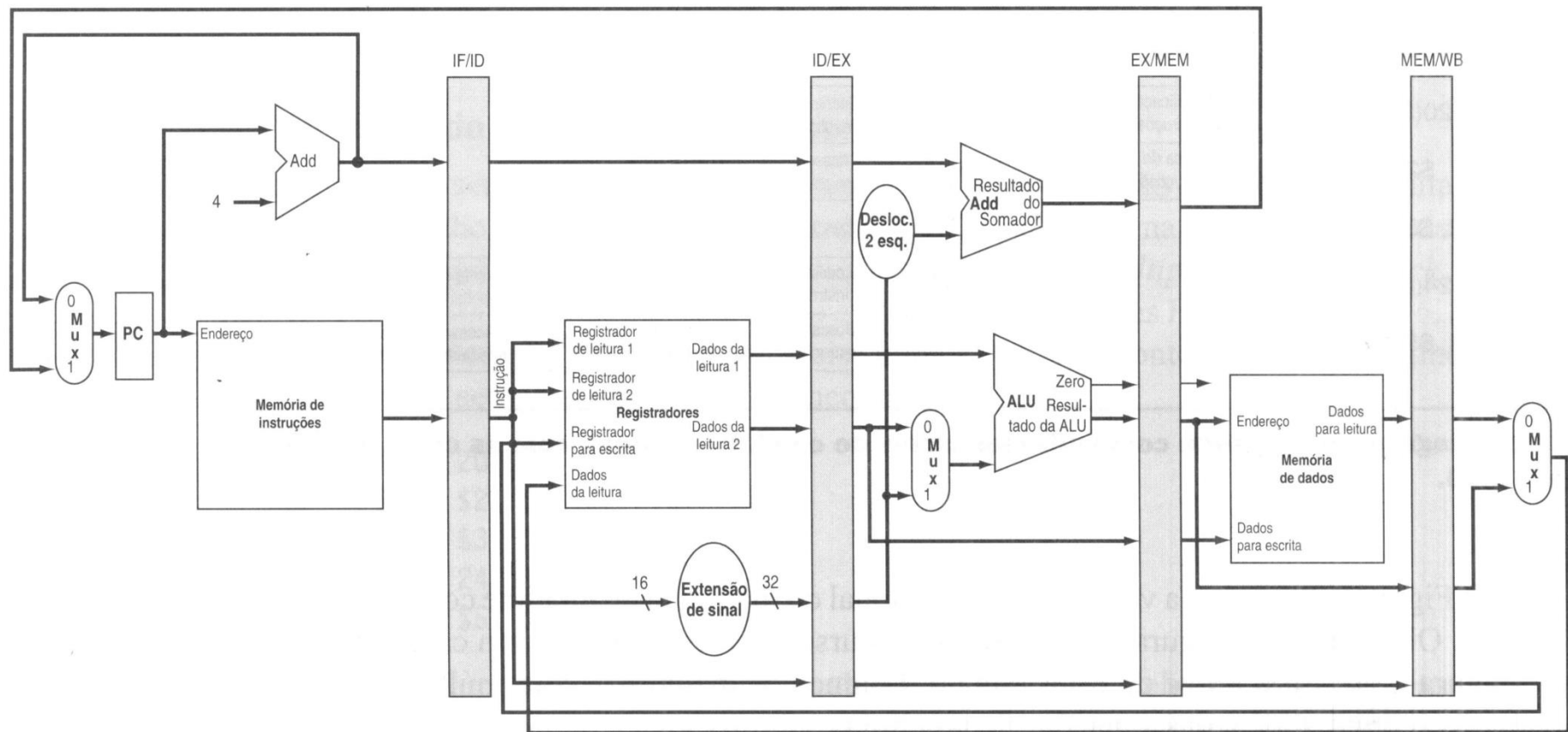


Representando pipelines graficamente



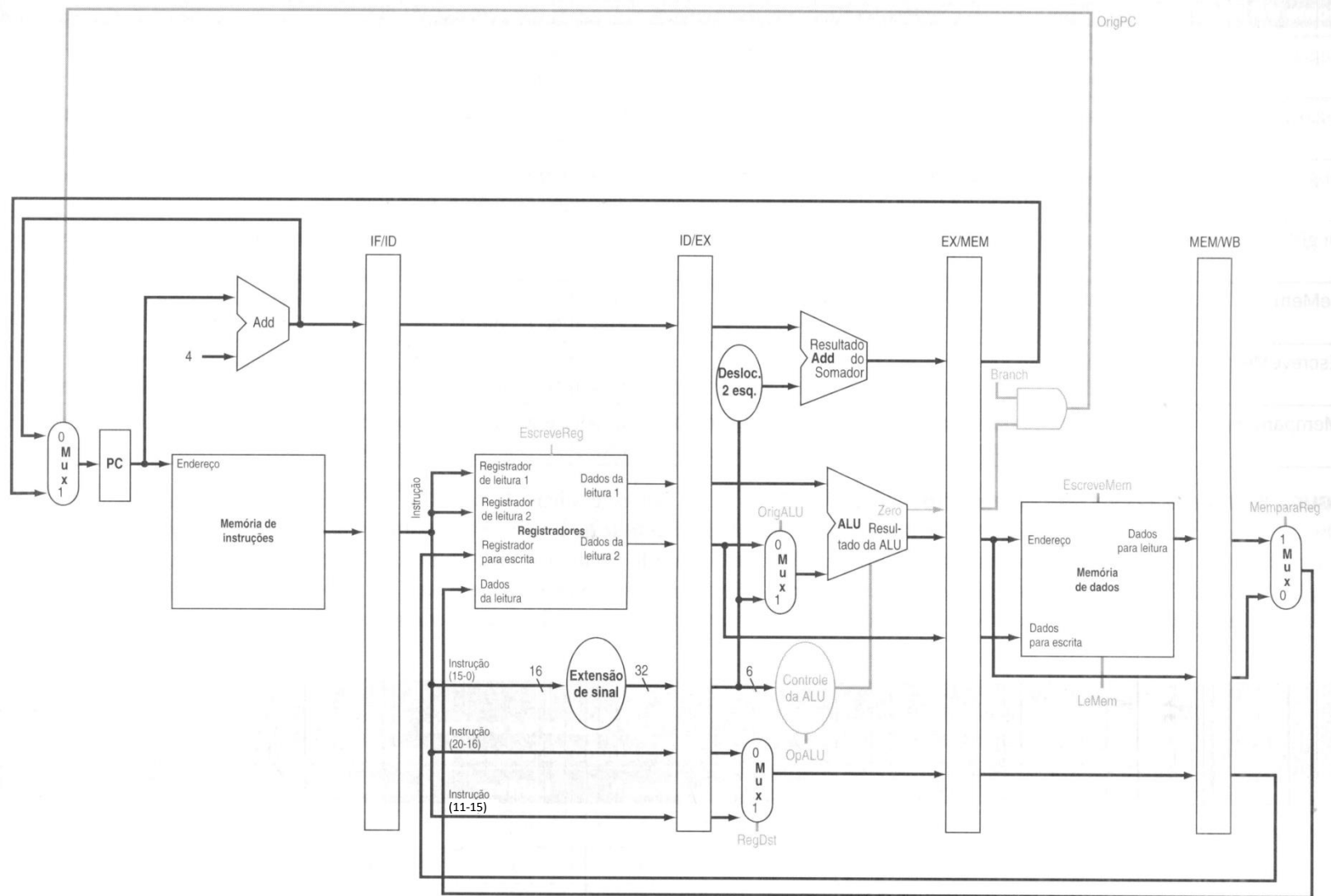
O caminho de dados em um ciclo de clock (CC5)

add \$14, \$5, \$6	lw \$13, 24 (\$1)	add \$12, \$3, \$4, \$11	sub \$11, \$2, \$3	lw \$10, 20(\$1)
Busca de instruções	Decodificação de instruções	Execução	Memória	Escrita do resultado



CONTROLE DE UM PIPELINE

O caminho de dados em pipeline com sinais de controle identificados.



Bits de controle da ALU

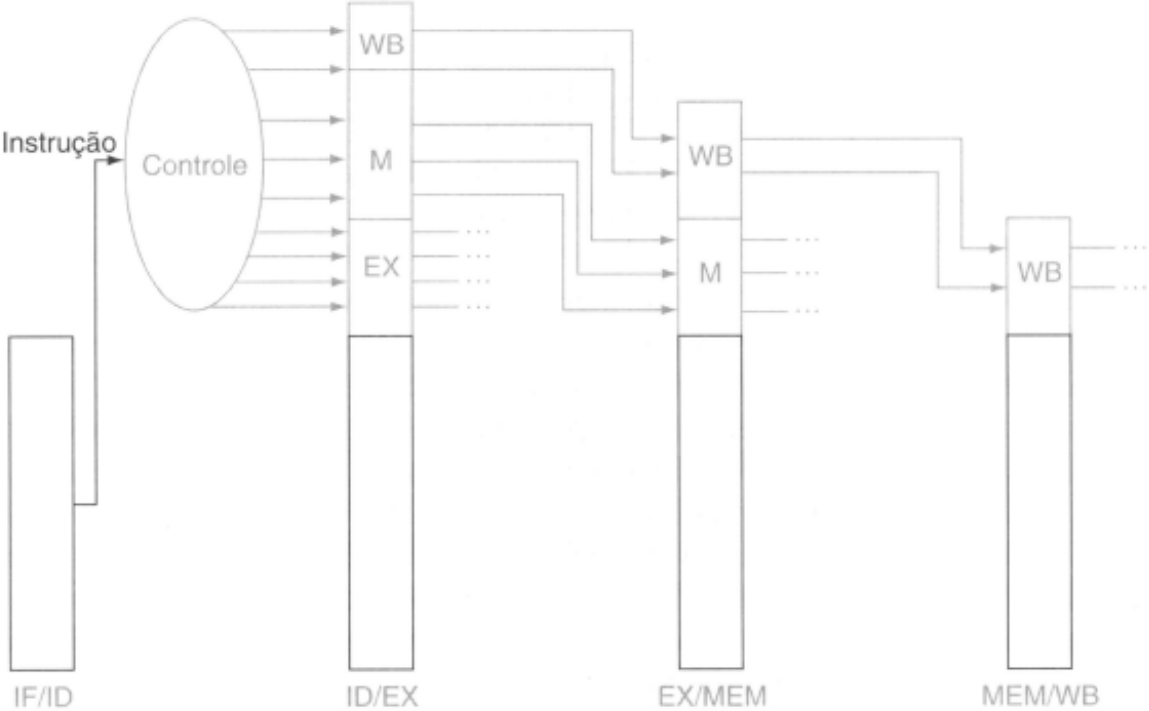
Opcode da instrução	OpALU	Operação da instrução	Campo funct	Ação da ALU desejada	Entrada do controle da ALU
LW	00	load word		add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
tipo R	10	add	100000	add	0010
tipo R	10	subtract	100010	subtract	0110
tipo R	10	AND	100100	AND	0000
tipo R	10	OR	100101	OR	0001
tipo R	10	set on less than	101010	set on less than	0111

Sinais de controle

Nome do sinal	Efeito quando inativo	Efeito quando ativo
RegDst	O número do registrador destino para a entrada Registrador para escrita vem do campo rt (bits 20:16).	O número do registrador destino para a entrada Registrador para escrita vem do campo rd (bits 15:11).
EscreveReg	Nenhum.	O registrador na entrada Registrador para escrita é escrito com o valor na entrada Dados para escrita.
OrigALU	O segundo operando da ALU vem da segunda saída do banco de registradores (Dados da leitura 2).	O segundo operando da ALU consiste nos 16 bits mais baixos da instrução com sinal estendido.
OrigPC	O PC é substituído pela saída do somador que calcula o valor de $PC + 4$.	O PC é substituído pela saída do somador que calcula o destino do desvio.
LeMem	Nenhum.	O conteúdo da memória de dados designado pela entrada Endereço é colocado na saída Dados da leitura.
EscreveMem	Nenhum.	O conteúdo da memória de dados designado pela entrada Endereço é substituído pelo valor na entrada Dados para escrita.
MemparaReg	O valor enviado para a entrada Dados para escrita do banco de registradores vem da ALU.	O valor enviado para a entrada Dados para escrita do banco de registradores vem da memória de dados.

Bits de controle da ALU

Instrução	Linhas de controle do estágio de execução/cálculo de endereço				Linhas de controle do estágio de acesso à memória			Linhas de controle do estágio de escrita do resultado	
	RegDst	OpALU1	OpALU0	OrigALU	Branch	LeMem	EscreveMem	EscreveReg	MemparaReg
Formato R	1	1	0	0	0	0	0	1	0
lw	0	0	0	1	0	1	0	1	1
sw	X	0	0	1	0	0	1	0	X
beq	X	0	1	0	1	0	0	0	X



Controle

- Busca de instruções
- Decodificação de instruções e leitura do banco de registradores
- Execução ou calculo de endereço
 - RegDst, OpALU, OrigALU
- Acesso a memória
 - EscreveMem, LeMem, Branch
- Escrita de resultado
 - MemparaReg, EscreveReg

Bibliografia

1. PATTERSON, D.A; HENNESSY, J.L. **Organização e Projeto de Computadores: A Interface Hardware/Software**. 3a. Ed. Elsevier, 2005.
 - Capítulo 5.
2. Notas de aula do prof. Luciano J. Senger:
 - <http://www.ljsenger.net/classroom.html>
3. Notas de aula da Profa. Mary Jane Irwin
 - CSE 331 Computer Organization and Design
 - <http://www.cse.psu.edu/research/mdl/mji/mjicourses>



FIM

- FIM:
 - **Aula 11** – Pipeline do MIPS 2 – Caminho de dados
- Próxima aula:
 - **Aula 12** – Pipeline do MIPS 3 – Solucionando hazards de dados