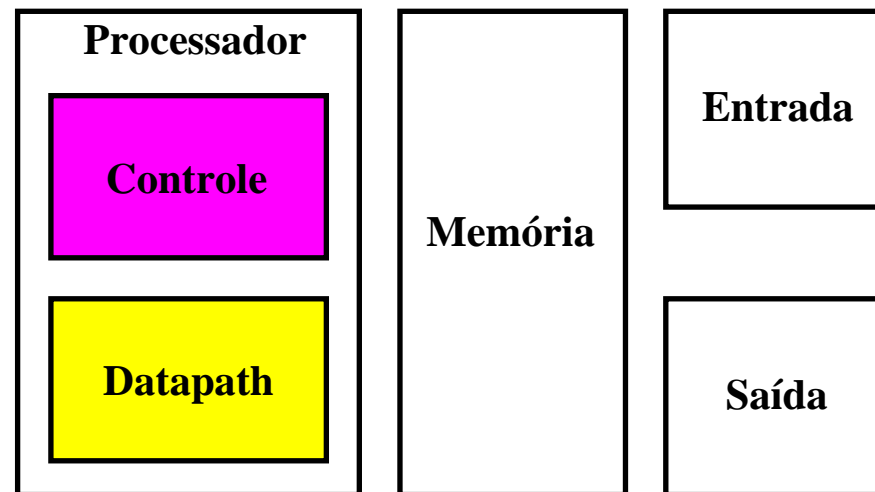
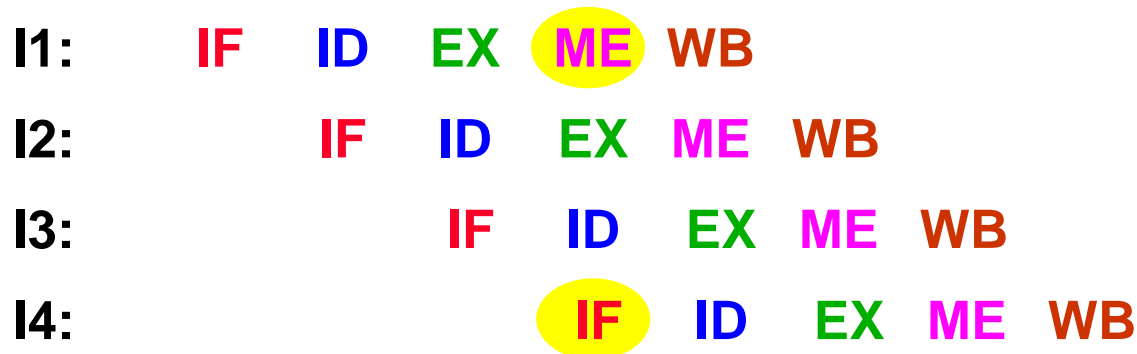


Pipelining: hazards de controle e estrutura do datapath+controlador



Hazards estruturais

- HW não suporta combinação de instruções
 - Exemplo 1: conflito em memória unificada



- Resolvido com duplicação do “bandwidth”
 - IF (cache de instruções)
 - ME (cache de dados)

Hazards estruturais

- HW não suporta combinação de instruções
 - Exemplo 2: conflito no banco de registradores

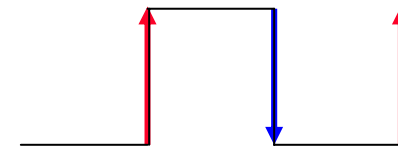
I1: IF ID EX ME **WB**

I2: IF ID EX ME WB

I3: IF ID EX ME WB

I4: IF **ID** EX ME WB

- Resolvido com temporização
 - Escrita no primeiro semi-ciclo
 - » Carga na transição oposta
 - Leitura no segundo semi-ciclo



Hazard de controle

- Associado a desvios condicionais
- A instrução a iniciar execução no próximo ciclo **depende do resultado do teste**
- Mas resultado **não disponível** em tempo
- Problema tem três soluções
 - Pausa até que resultado do teste disponível
 - » “Stall on branch”
 - Previsão do resultado do teste
 - » “Branch prediction”
 - Desvios com retardo
 - » “Delayed branch”

Stall on branch

add \$s0, \$s1, \$s2

beq \$t0, \$t1, L

lw \$t2, \$t3, 300(\$s3)

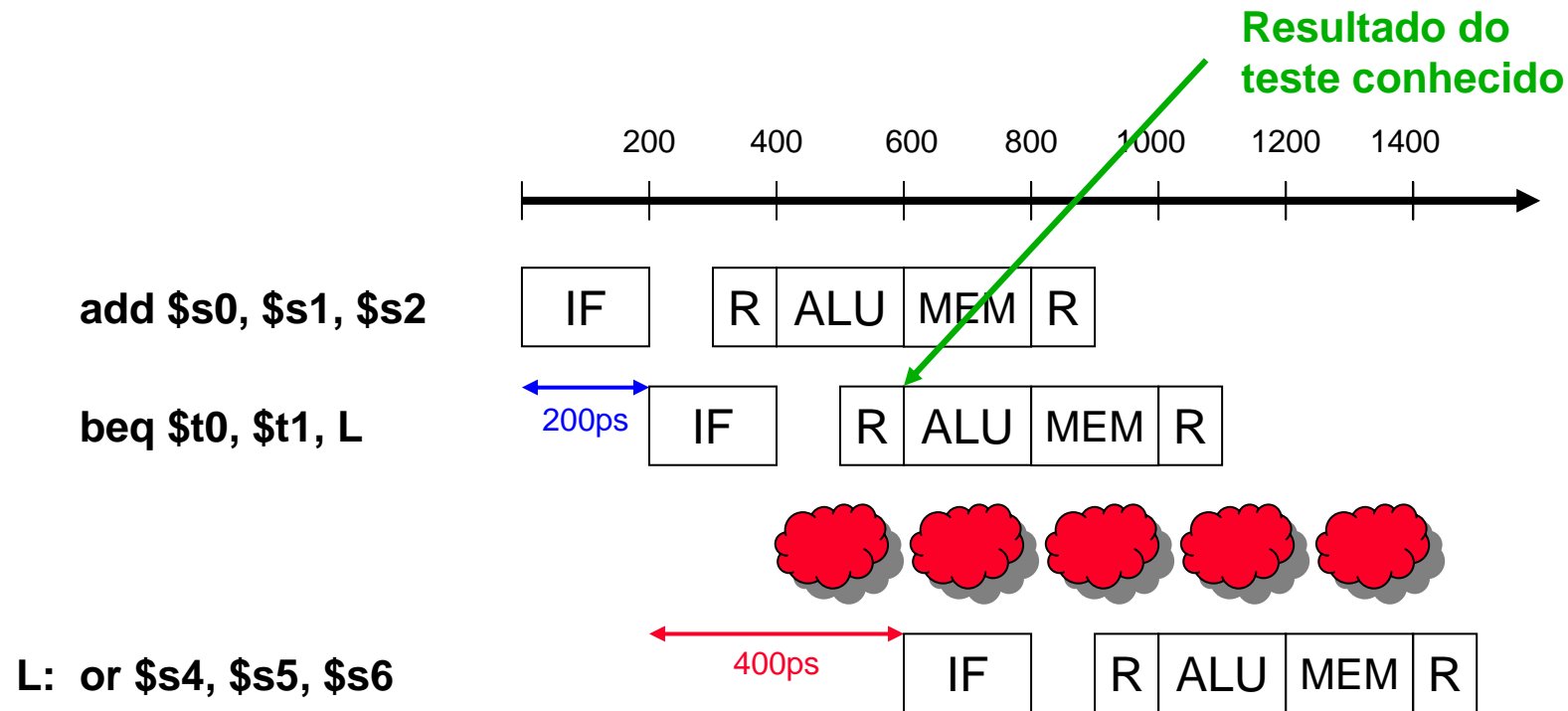
...

L: or \$s4, \$s5, \$s6

Hipótese: teste resolvido no segundo estágio

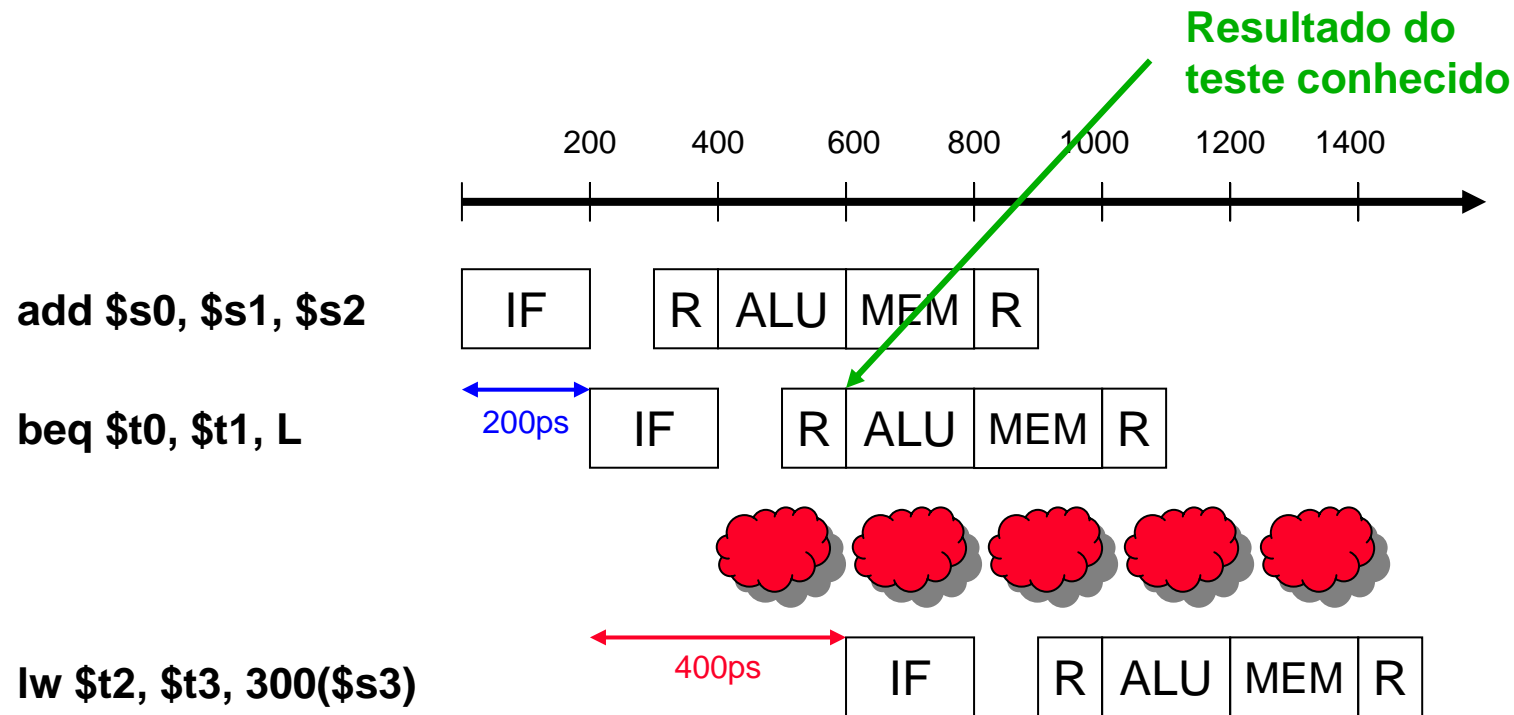
Cenários: desvio **tomado ou **não tomado****

Stall on branch: desvio tomado



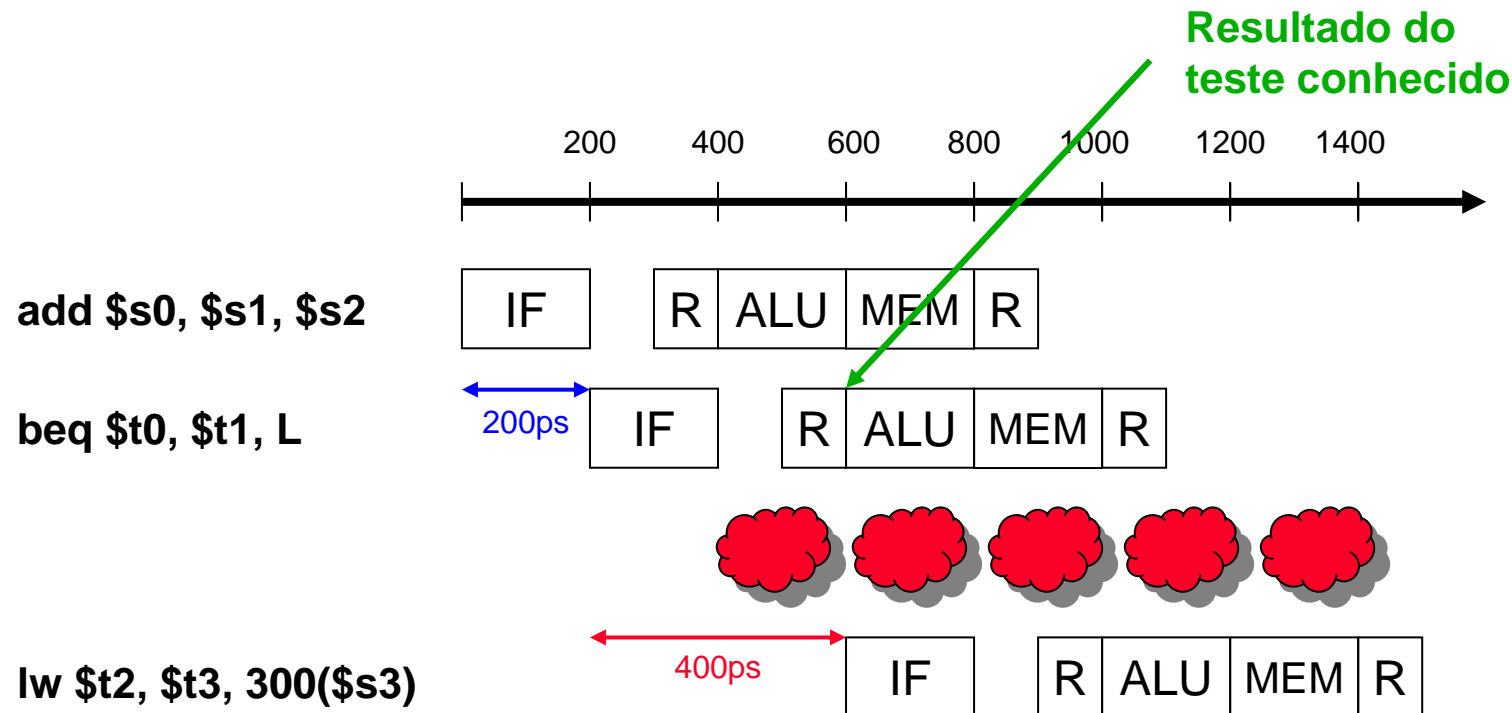
Penalidade de um ciclo

Stall on branch: desvio não tomado



Penalidade de um ciclo

Stall on branch: desvio não tomado



Conclusão: Penalidade de um ciclo **sempre** imposta à instrução que segue o desvio, qualquer que seja o resultado do teste.

Exemplo: se 13% de desvios (SPECint2000), então $CPI = 1,13$

Branch prediction

add \$s0, \$s1, \$s2

beq \$t0, \$t1, L

lw \$t2, \$t3, 300(\$s3)

...

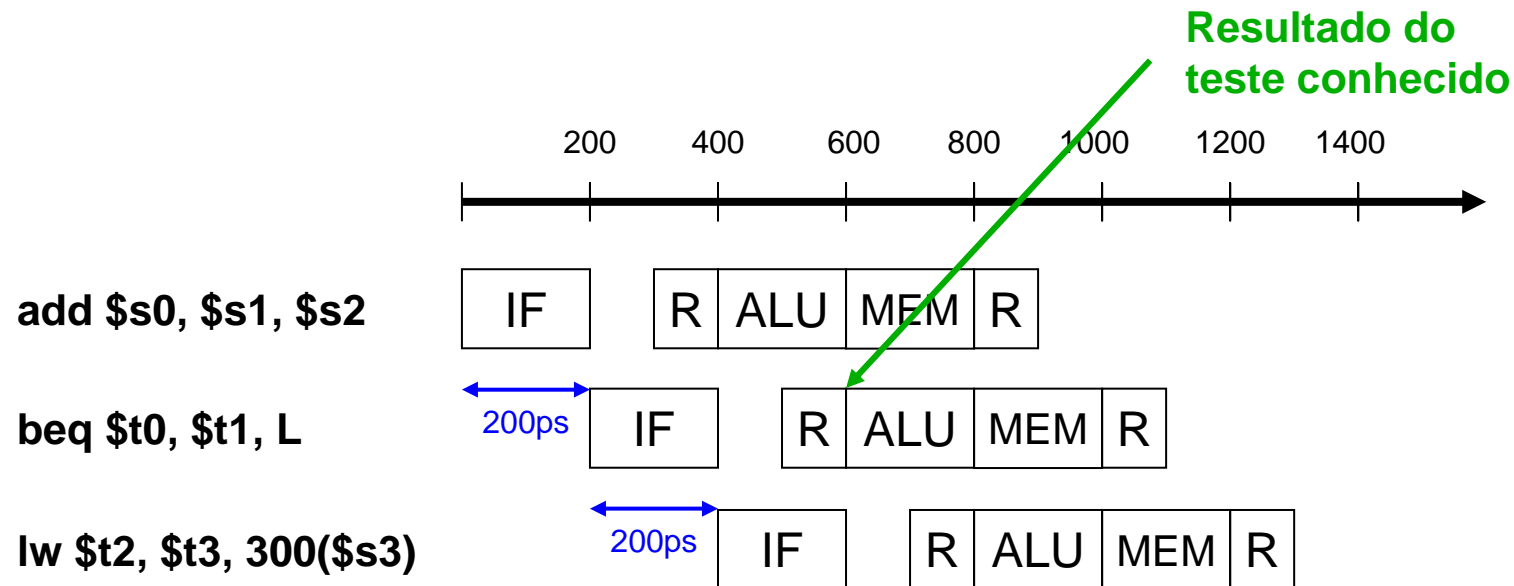
L: or \$s4, \$s5, \$s6

Hipótese: teste resolvido no segundo estágio

Hipótese de previsão: desvio não tomado

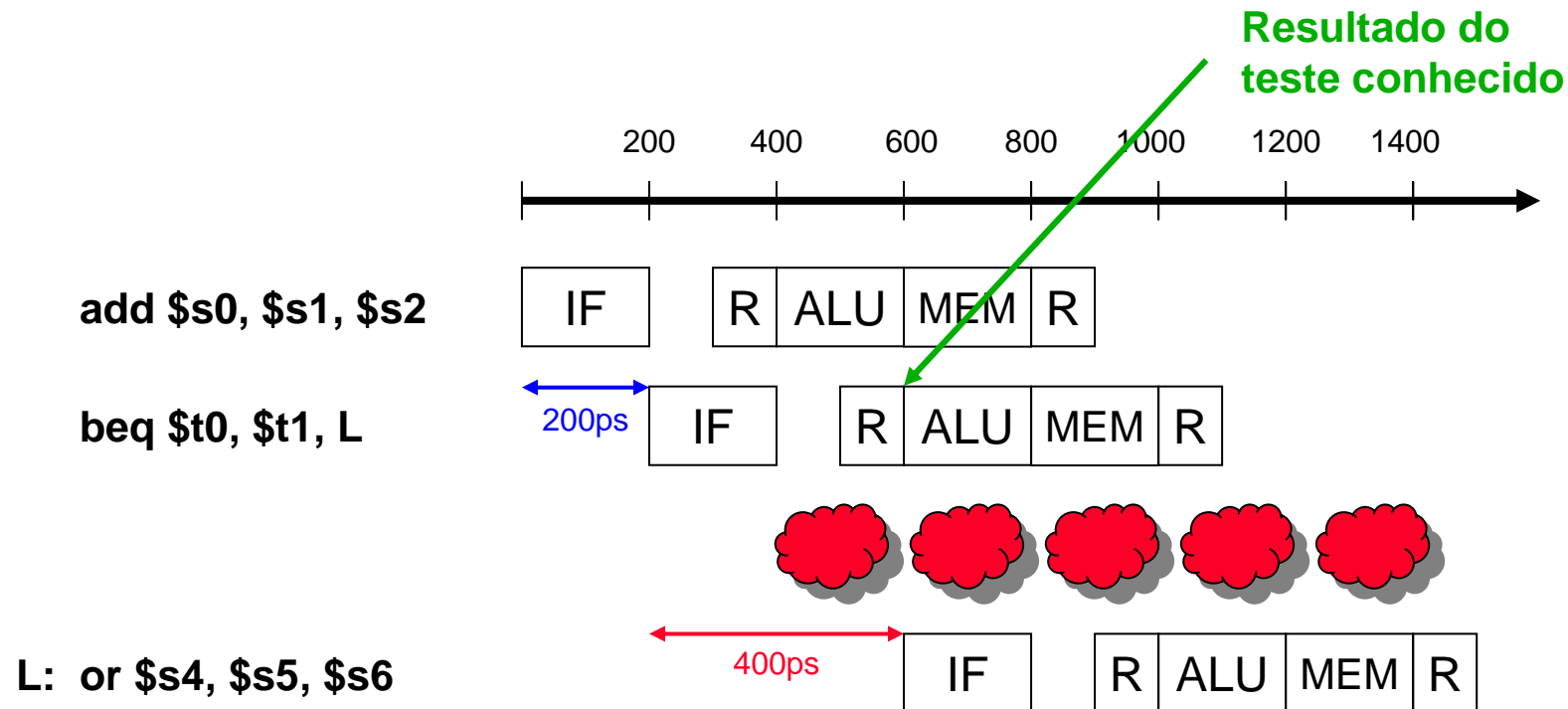
Cenários: previsão correta ou incorreta

Previsão correta



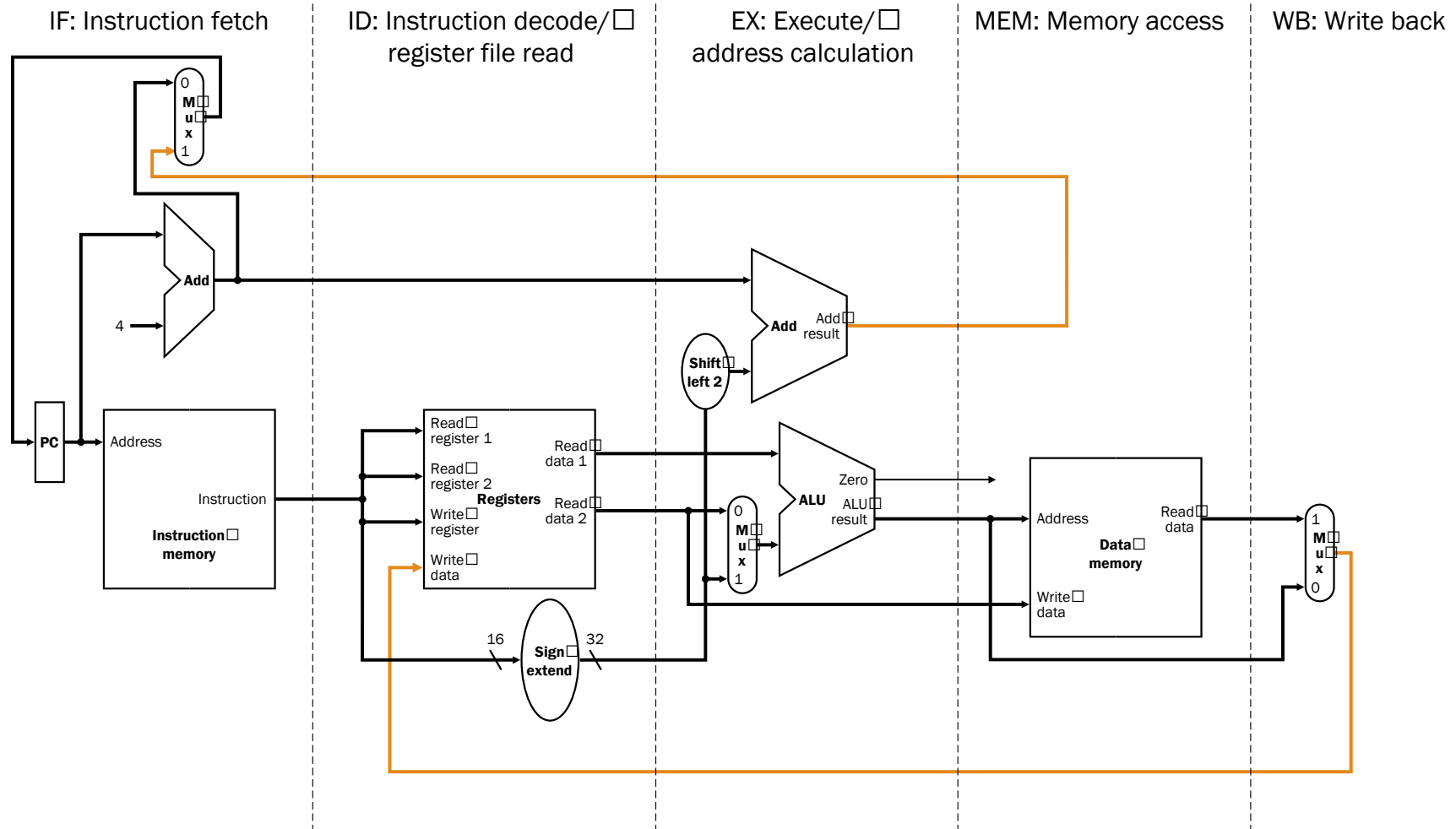
Sem penalidade

Previsão incorreta

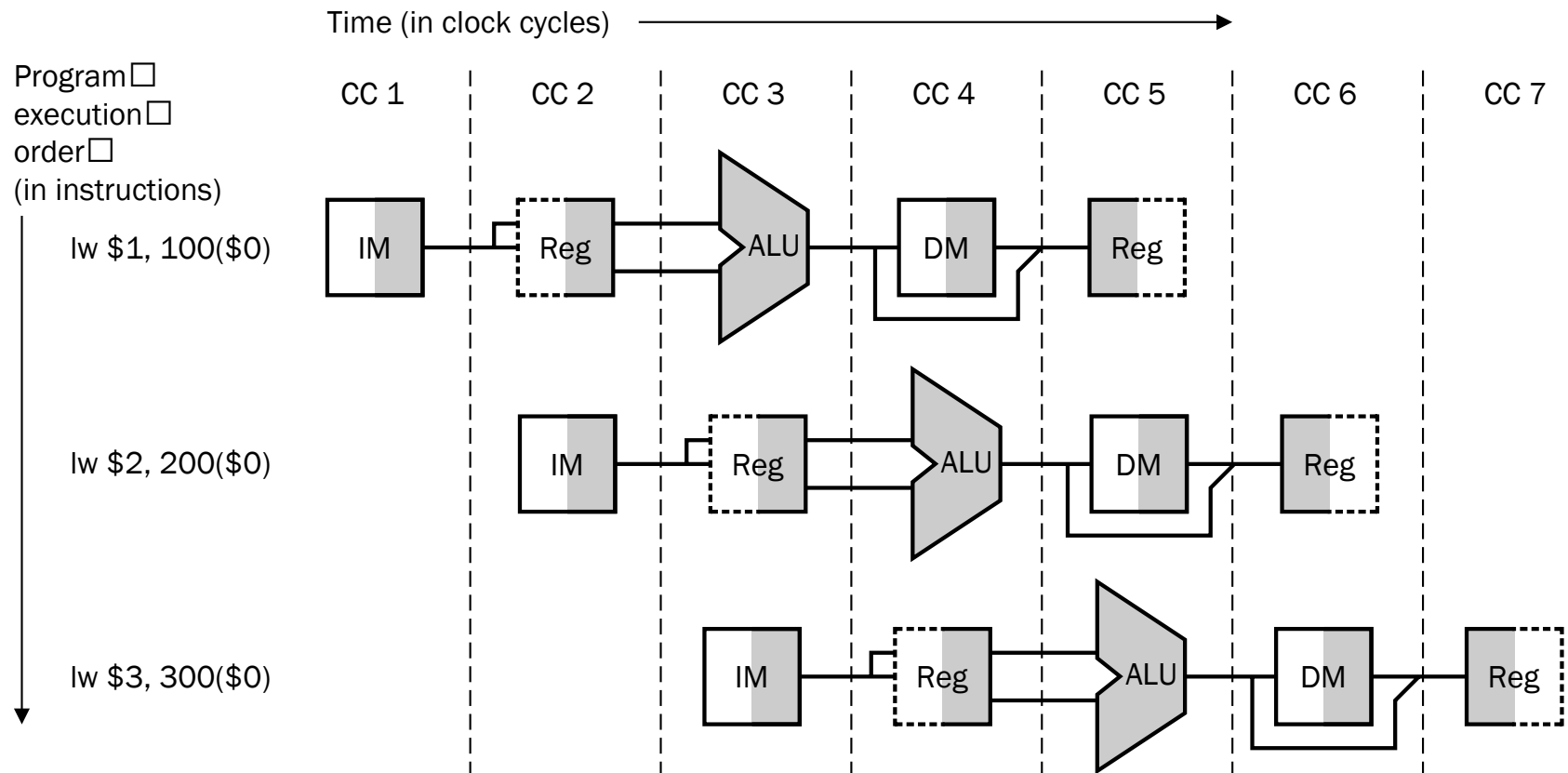


Penalidade de um ciclo imposta **somente** quando previsão incorreta.

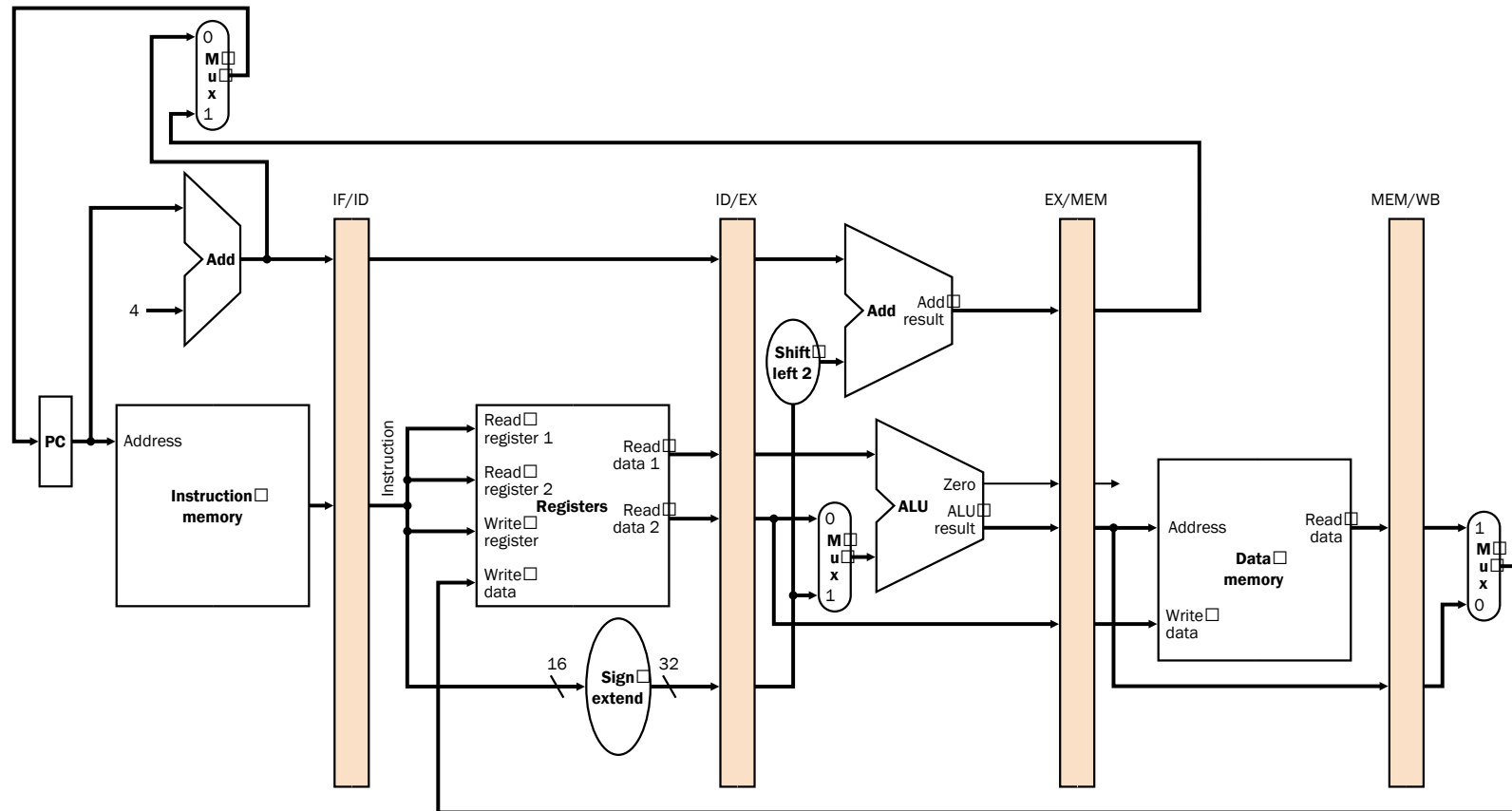
Pipeline: estrutura do datapath



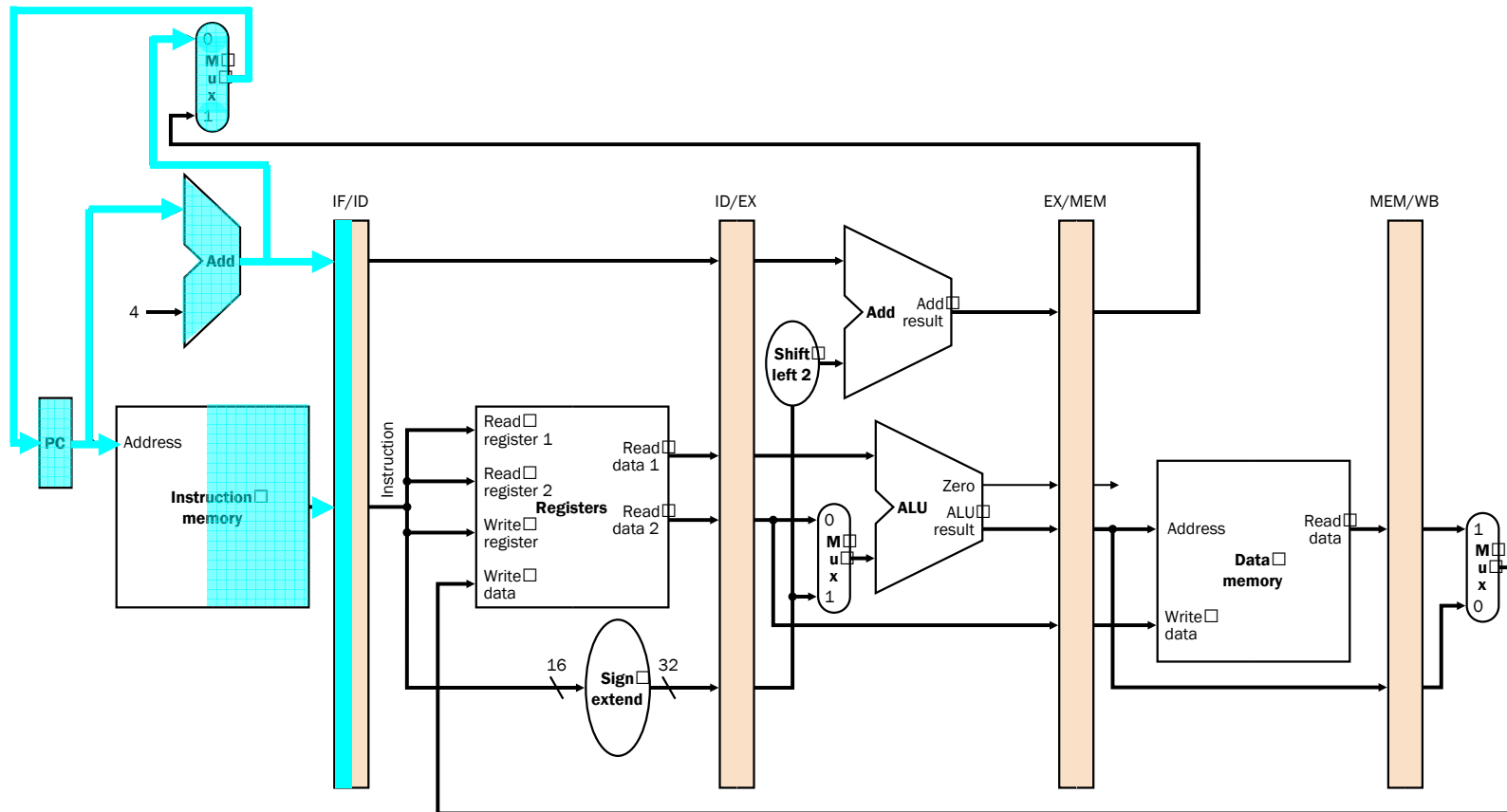
Pipeline: comportamento



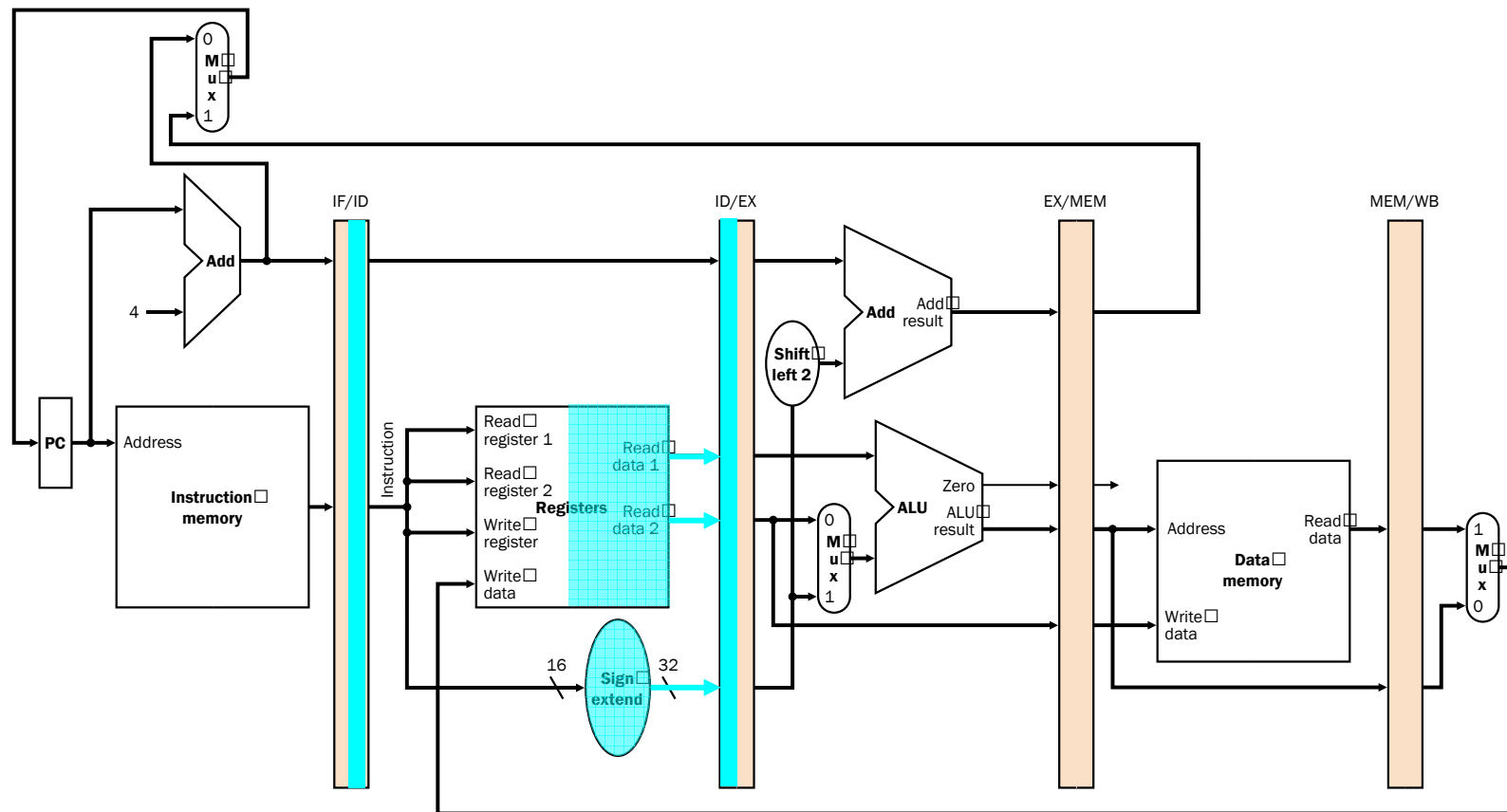
Delimitando os estágios



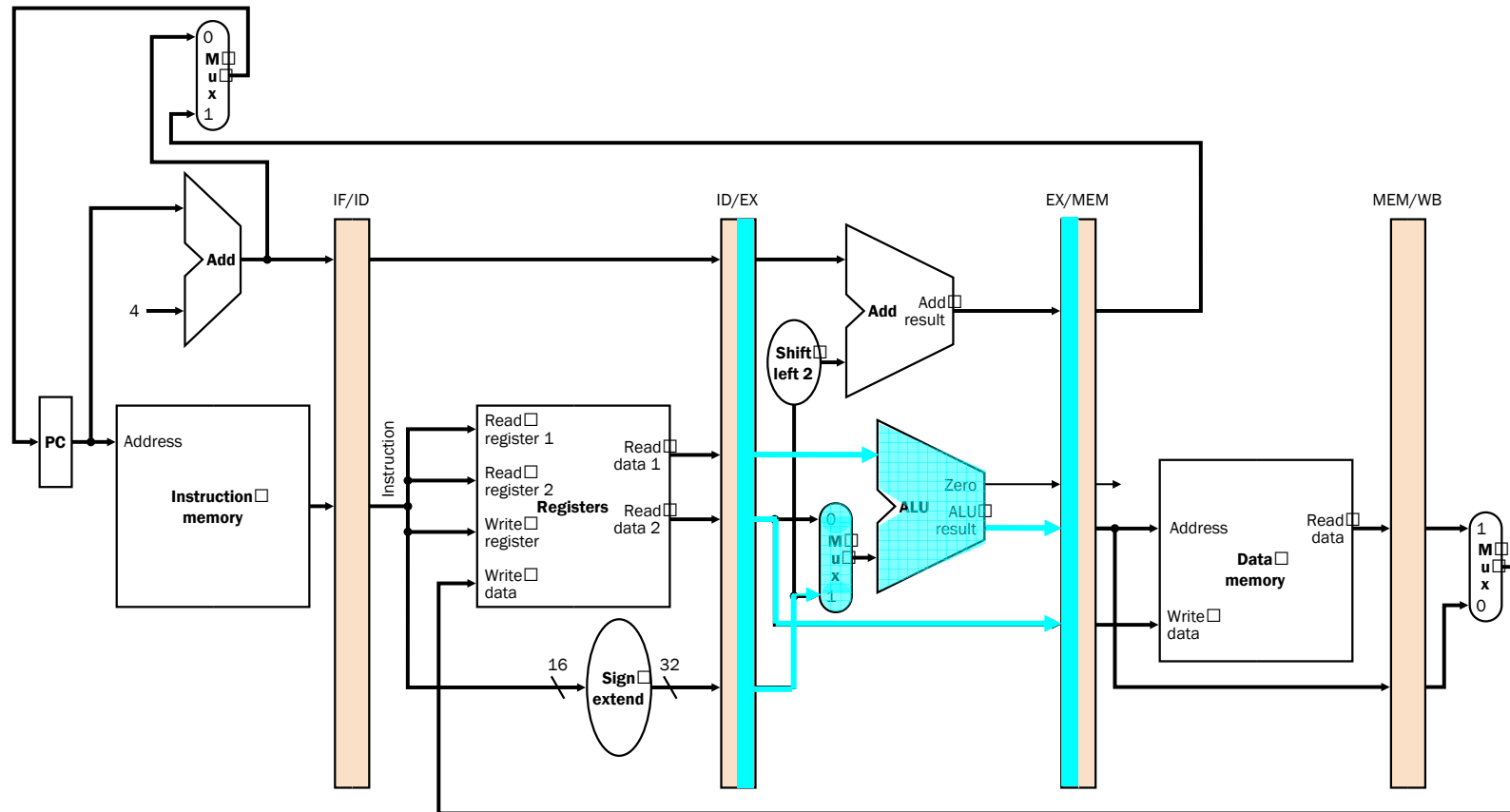
Instrução sw: primeiro estágio



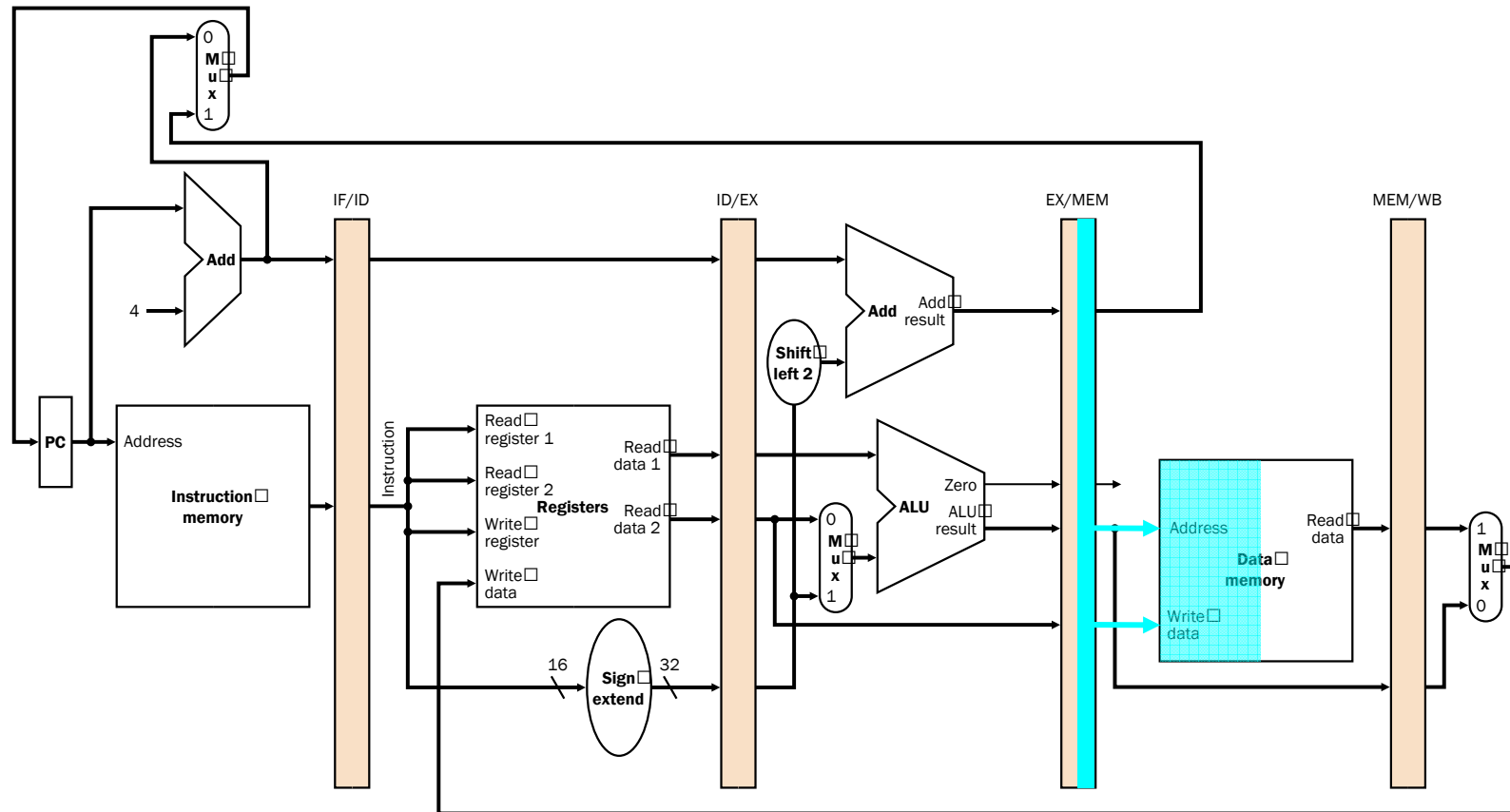
Instrução sw: segundo estágio



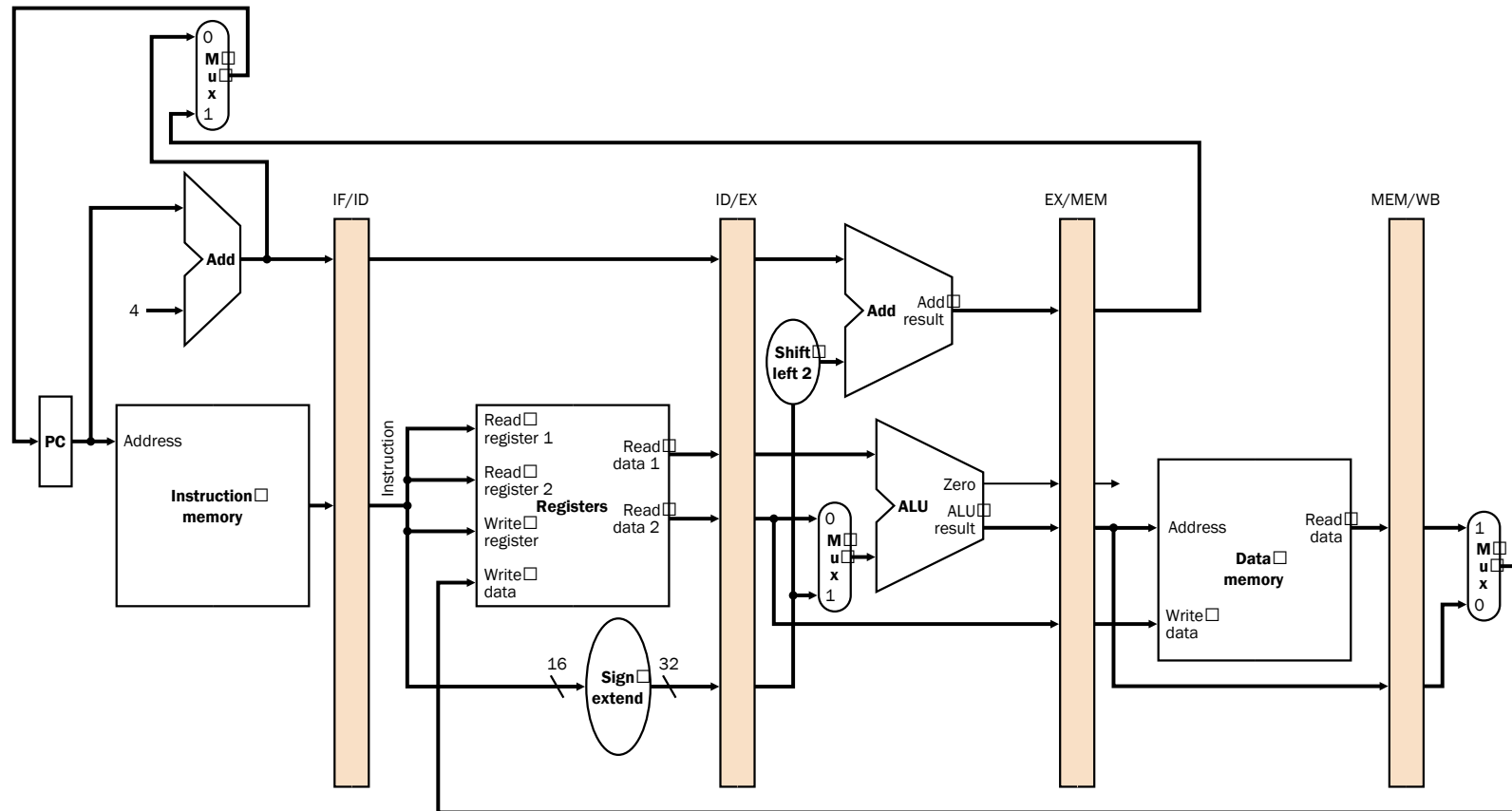
Instrução sw: terceiro estágio



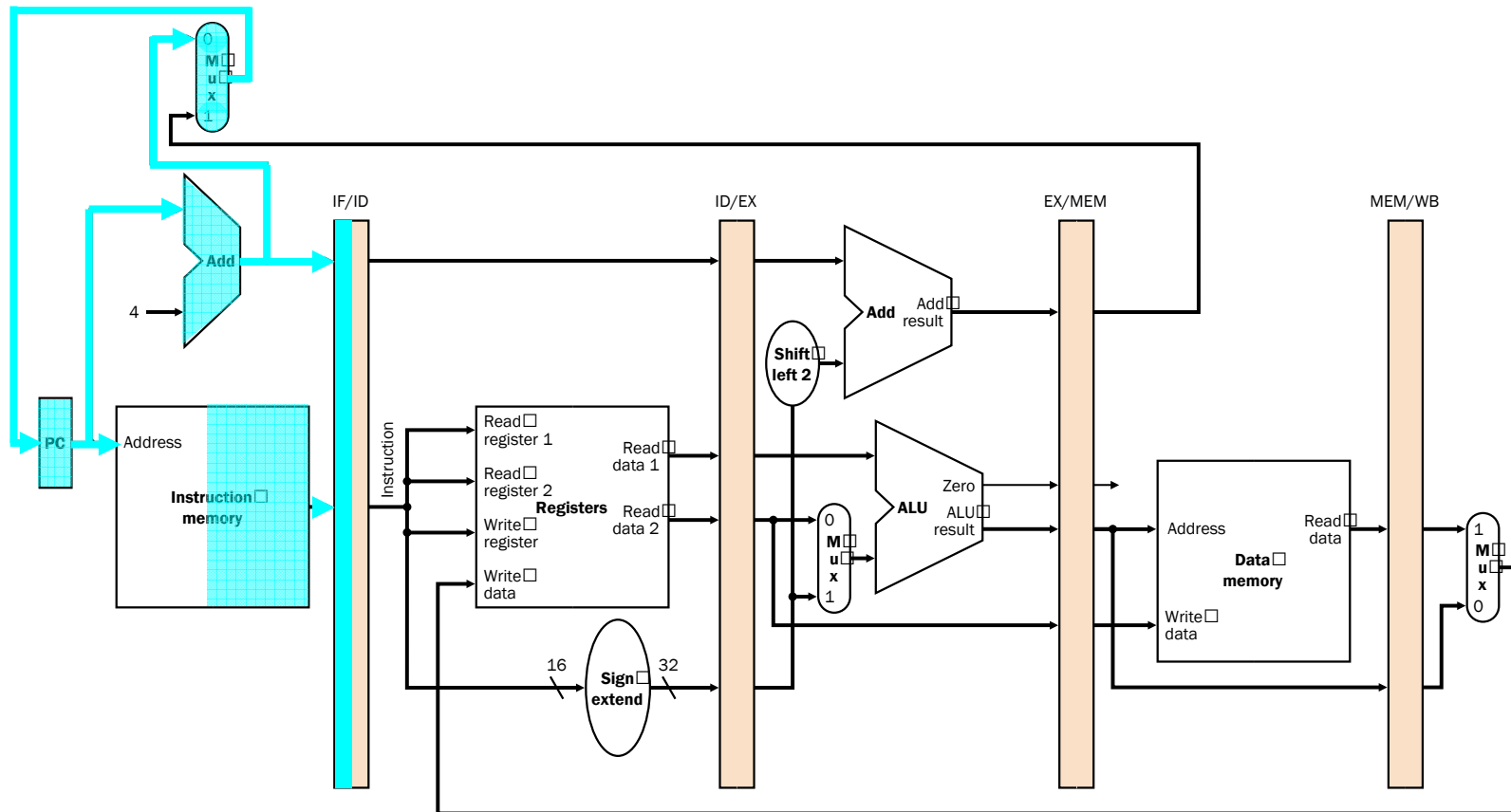
Instrução sw: quarto estágio



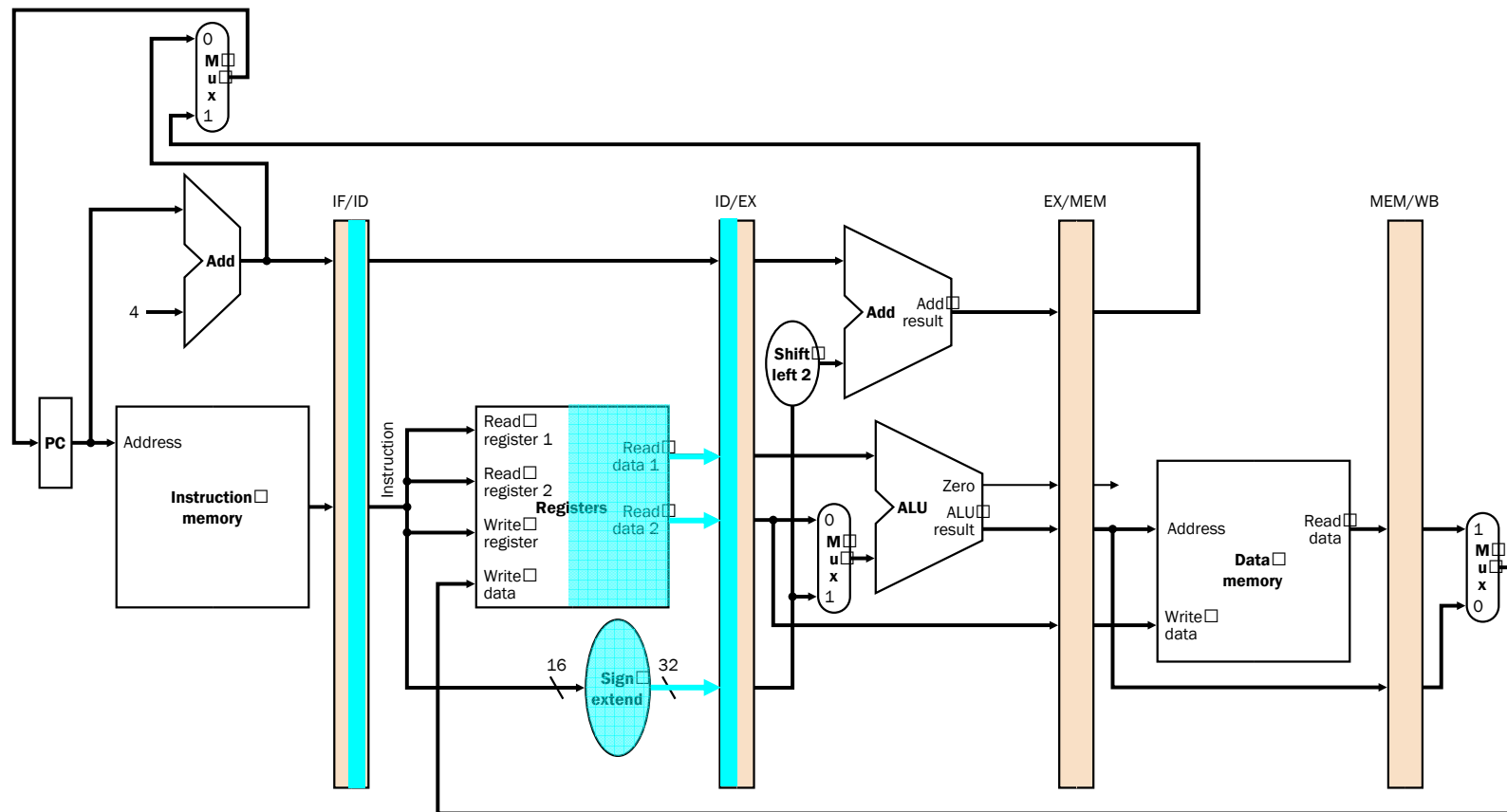
Instrução sw: quinto estágio



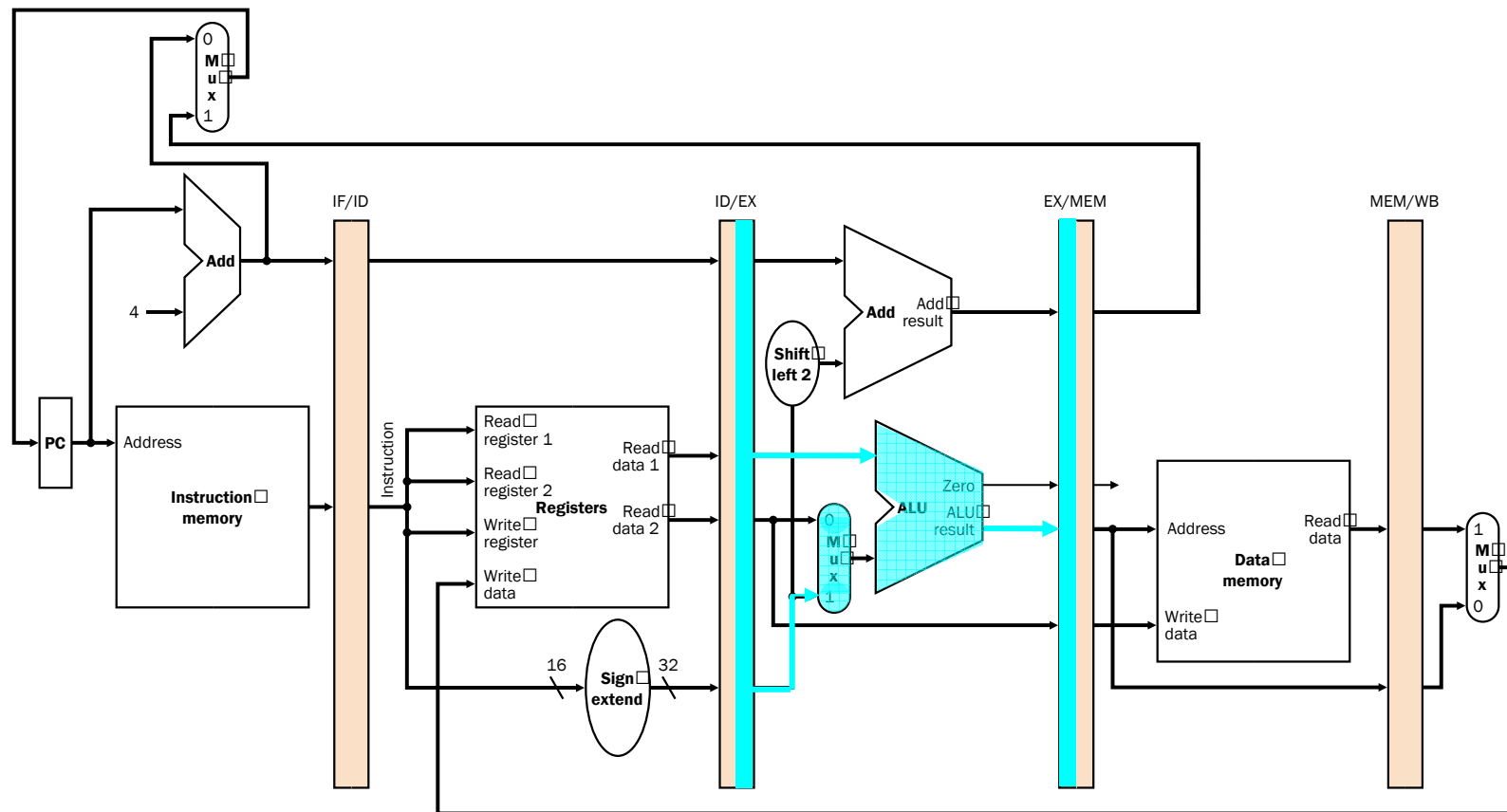
Instrução lw: primeiro estágio



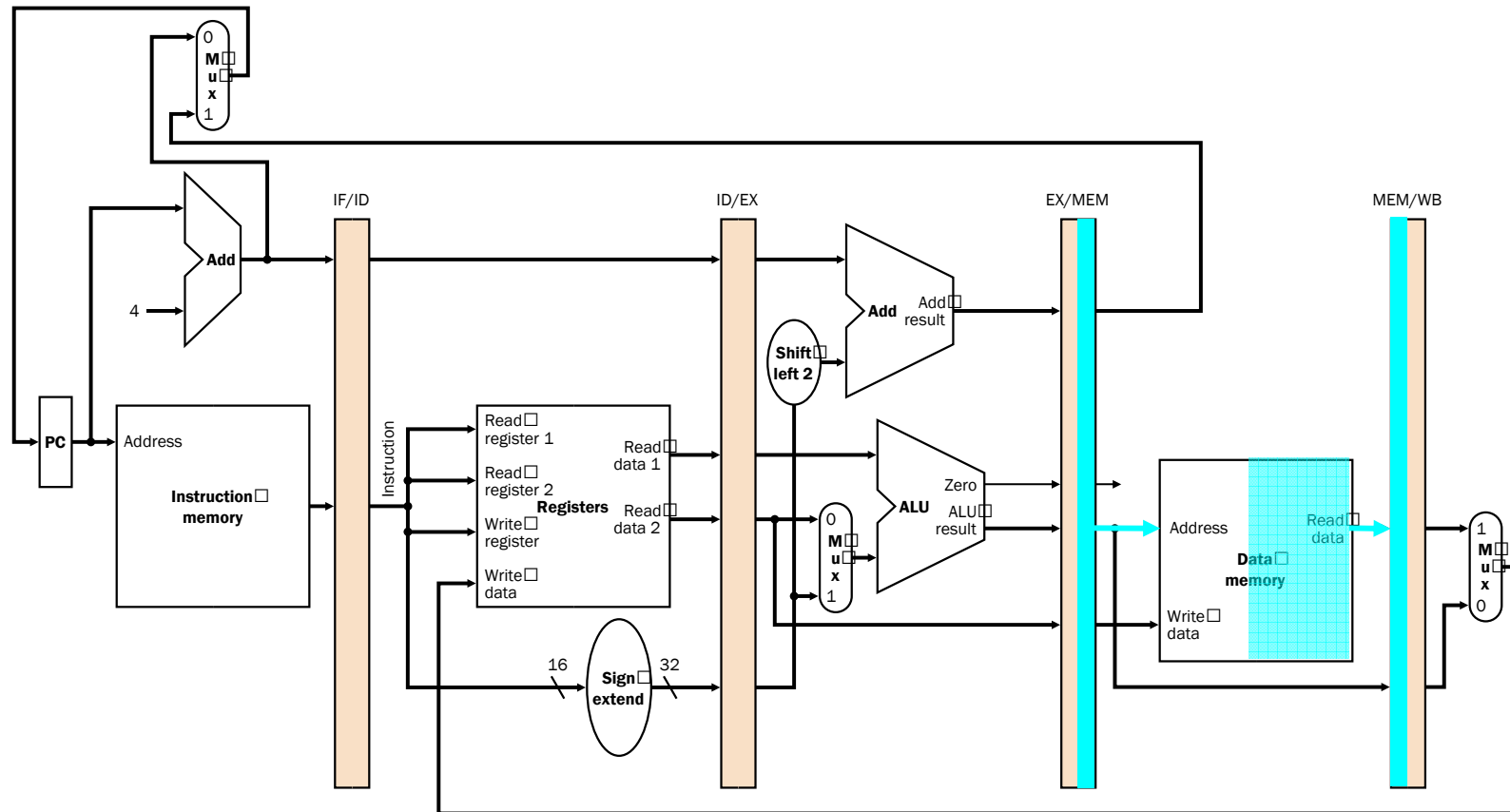
Instrução lw: segundo estágio



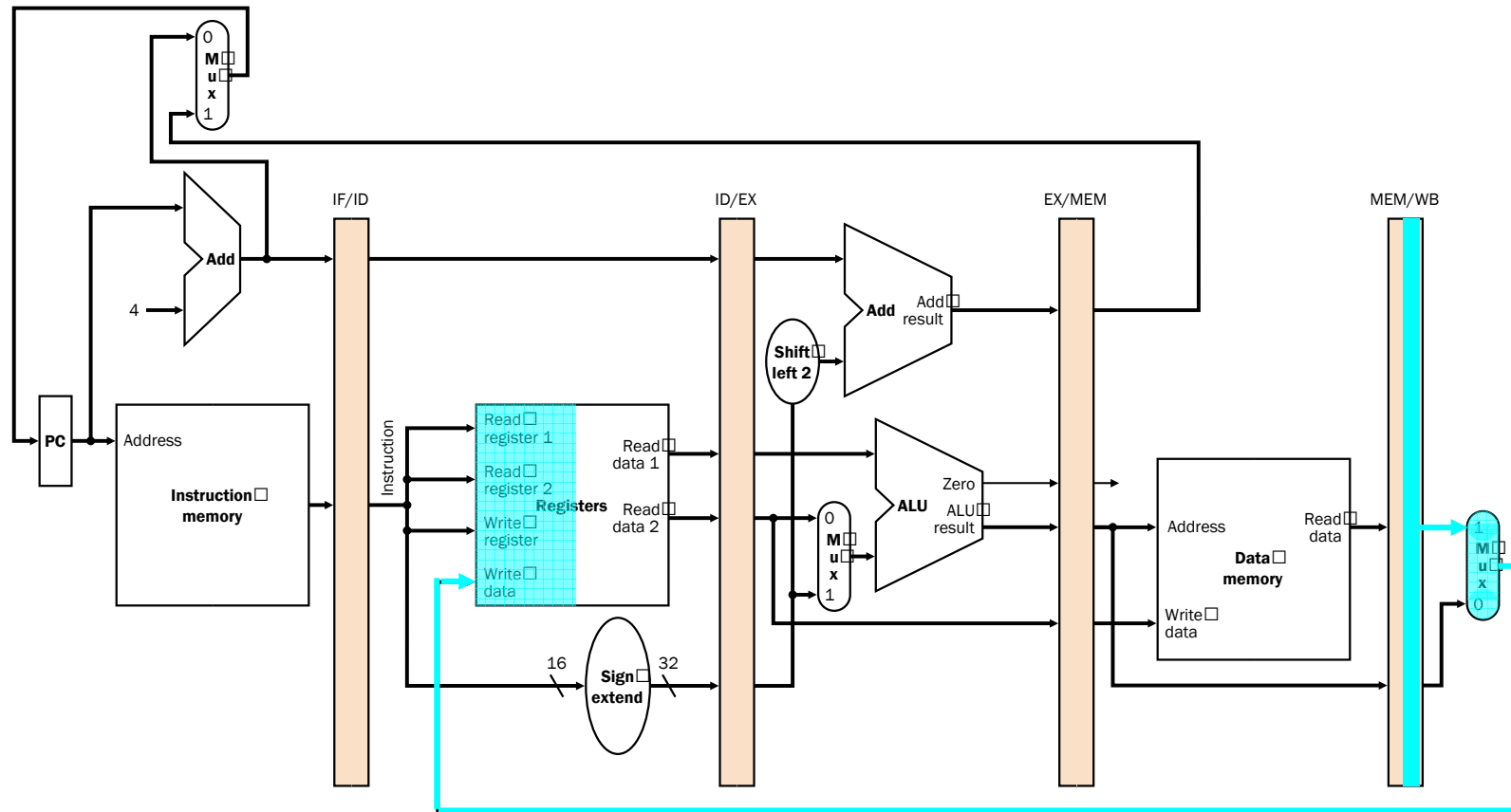
Instrução lw: terceiro estágio



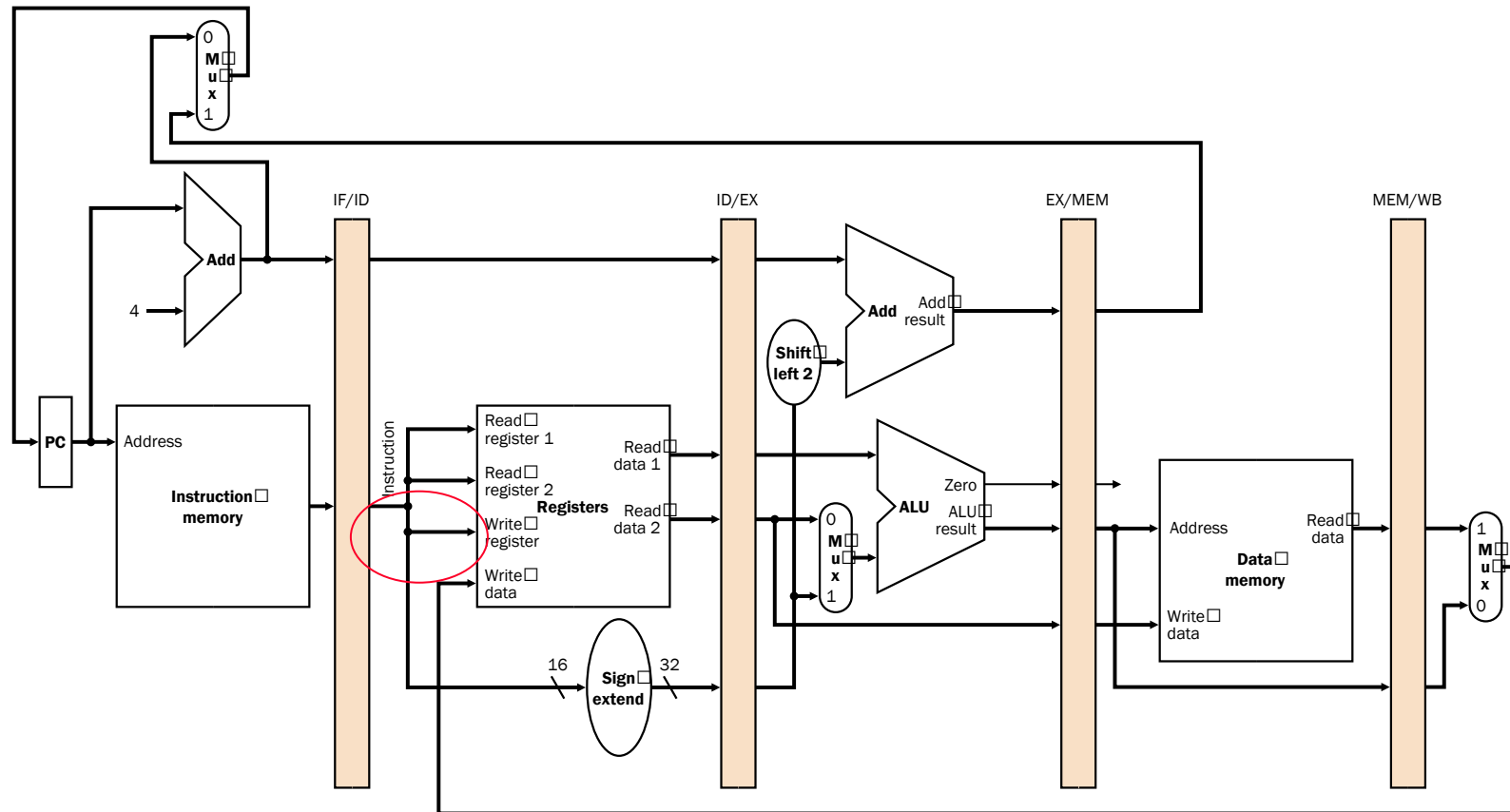
Instrução lw: quarto estágio



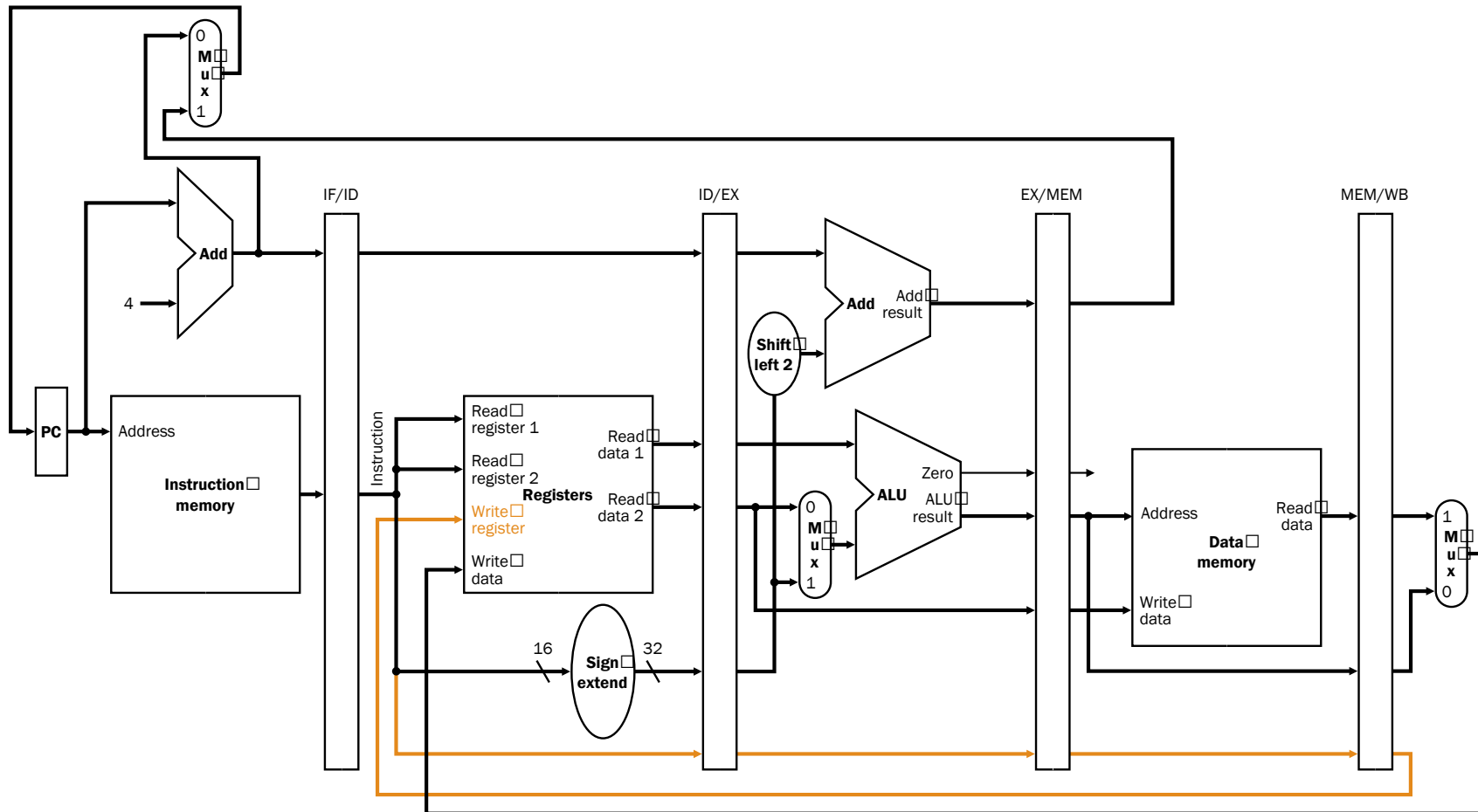
Instrução lw: quinto estágio



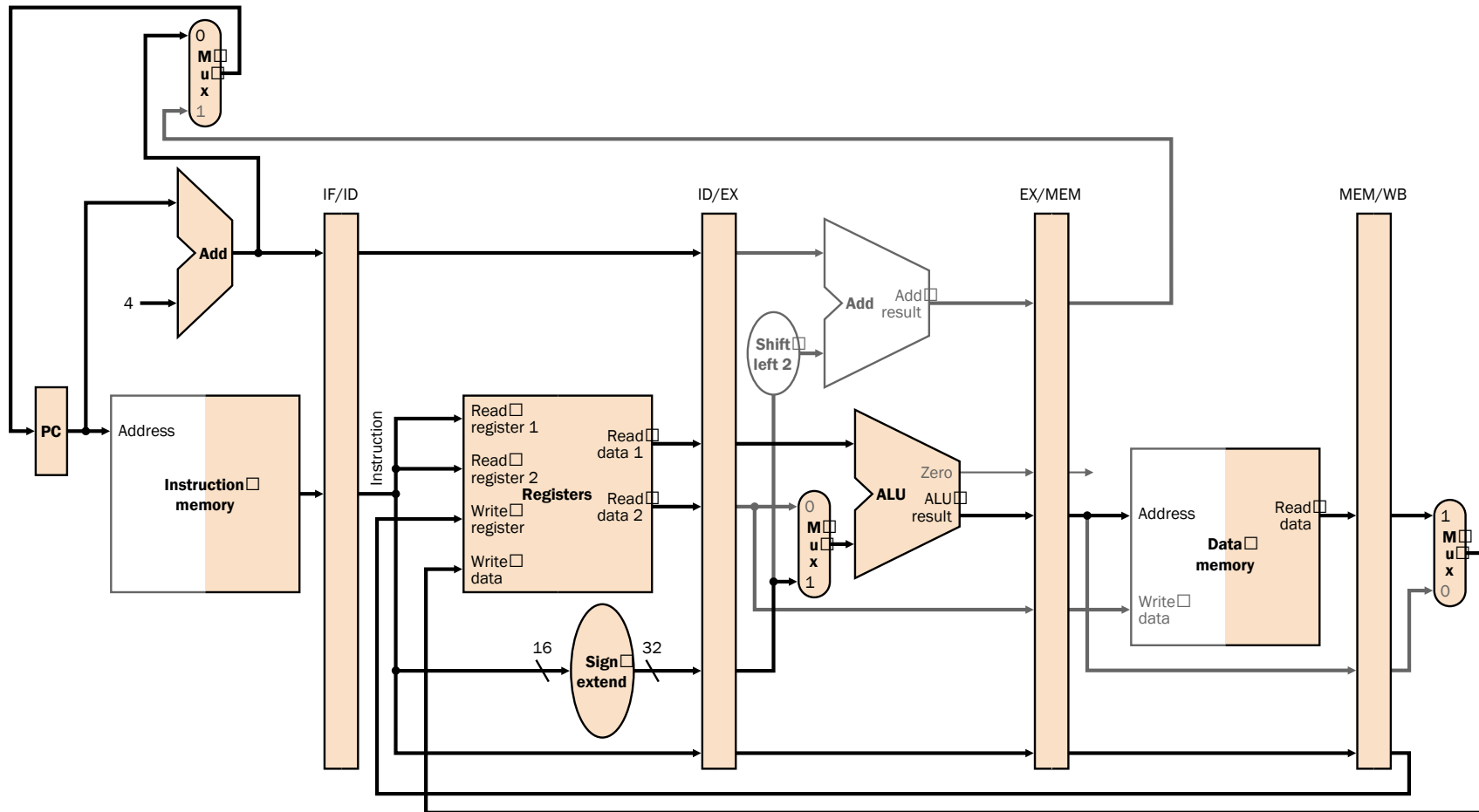
Suporte a lw: há um erro!



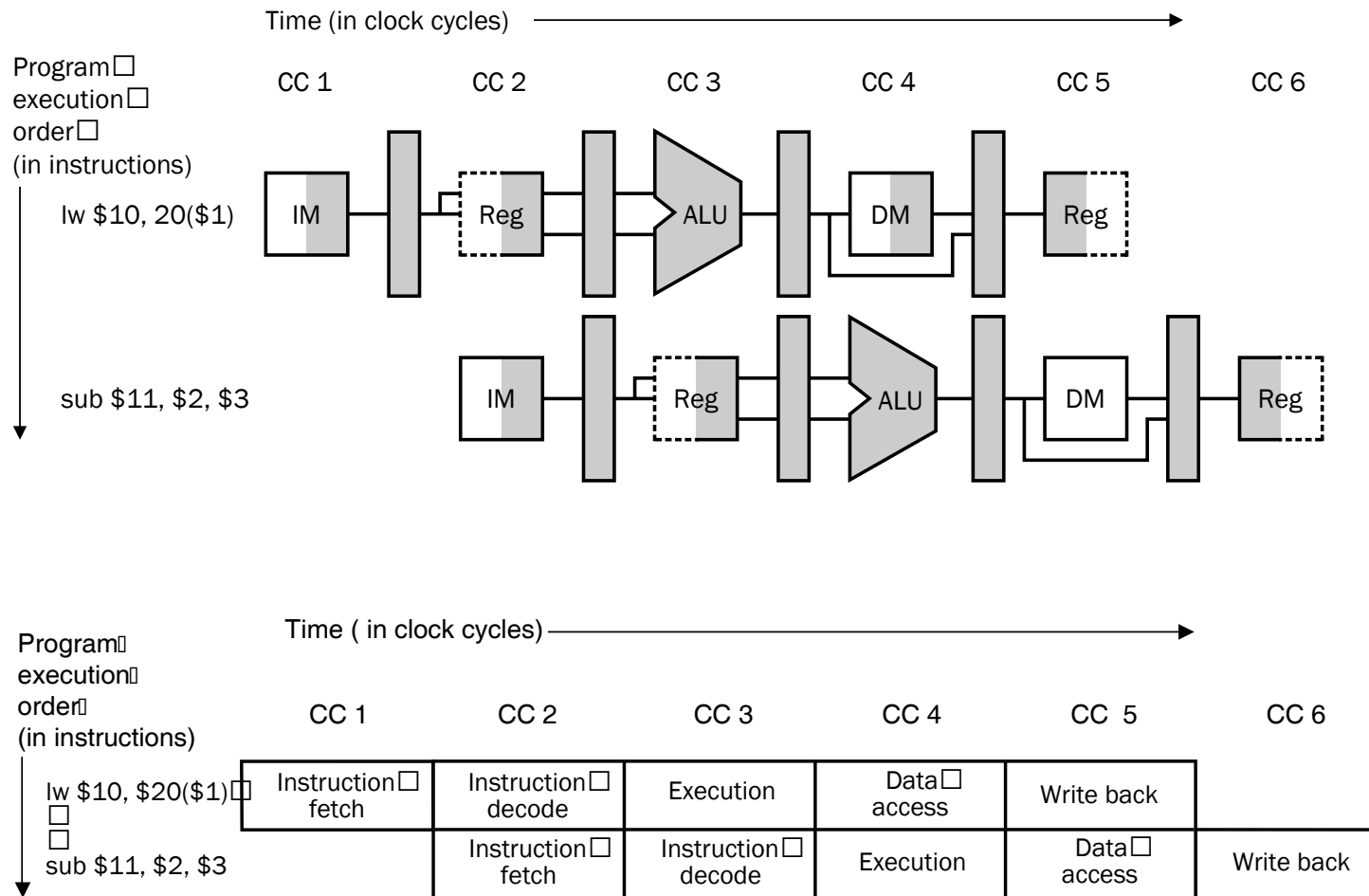
Corrigindo o erro ...



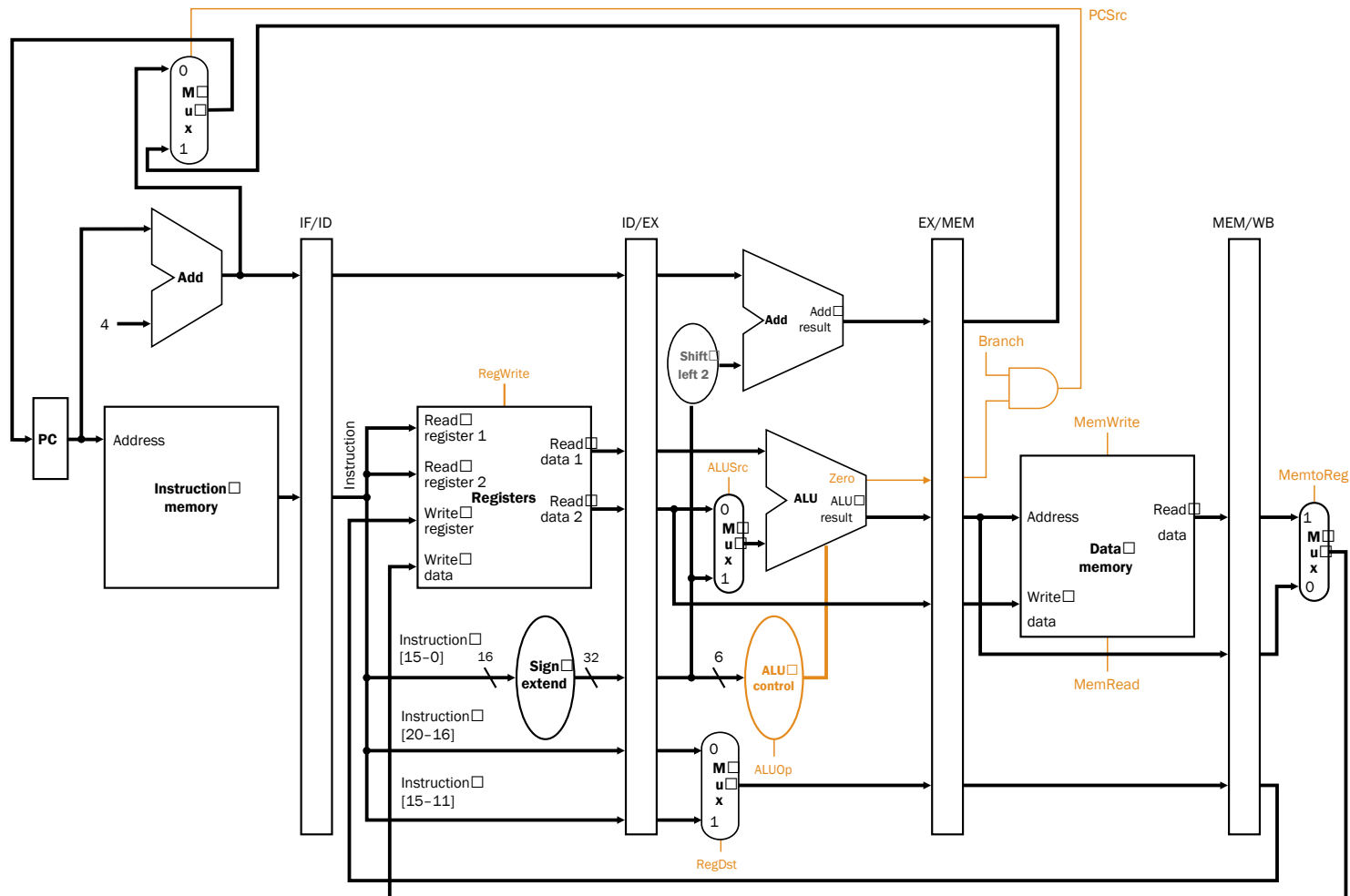
A porção do datapath usada p/ lw



Pipeline: representação gráfica

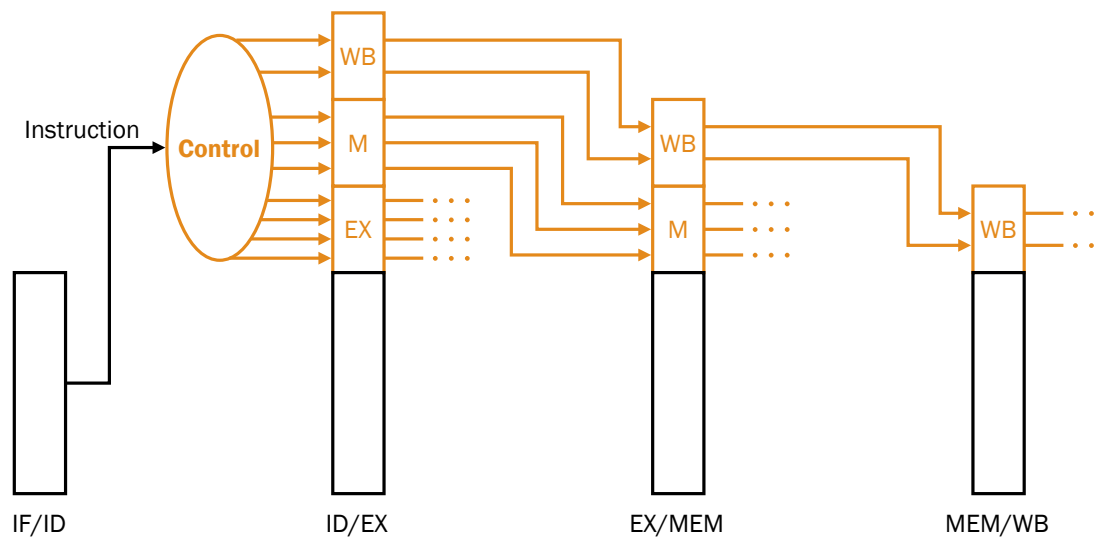


Pipeline: inserindo o controle



Emitindo os sinais de controle

		EX				M		WB	
	Reg	ALU	ALU	ALU	Branch	Mem	Mem	Reg	Memto
	Dst	Op1	Op0	Src		Read	Write	Write	Reg
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



Pipeline: datapath + controlador

