

Arquitetura e Organização de Computadores

Caminho de Dados (Parte 2)

Professor: André Luiz Tinassi D'Amato
Disciplina: Arquitetura de Computadores

Câmpus Apucarana



**MINISTÉRIO DA
EDUCAÇÃO**



Previsão Dinâmica de Desvios

- Uma implementação dessa técnica é um buffer de previsão de desvios ou tabela de histórico de desvios.
- Um buffer de previsão de desvios é uma pequena memória indexada pela parte menos significativa do endereço da instrução de desvio.
- A memória contém um bit que diz se o desvio foi tomado recentemente ou não.
Esse esquema simples de previsão de 1 bit tem um problema de desempenho:
 - Mesmo que um desvio quase sempre seja tomado, provavelmente faremos uma
 - previsão incorreta duas vezes, em vez de uma, quando ele não for tomado;
- Exemplo: Considere um desvio de loop que se desvia nove vezes seguidas, depois não é tomado uma vez. Qual é a exatidão da previsão para esse desvio, supondo que o bit de previsão para o desvio permaneça no buffer de previsão?

Previsão Dinâmica de Desvios

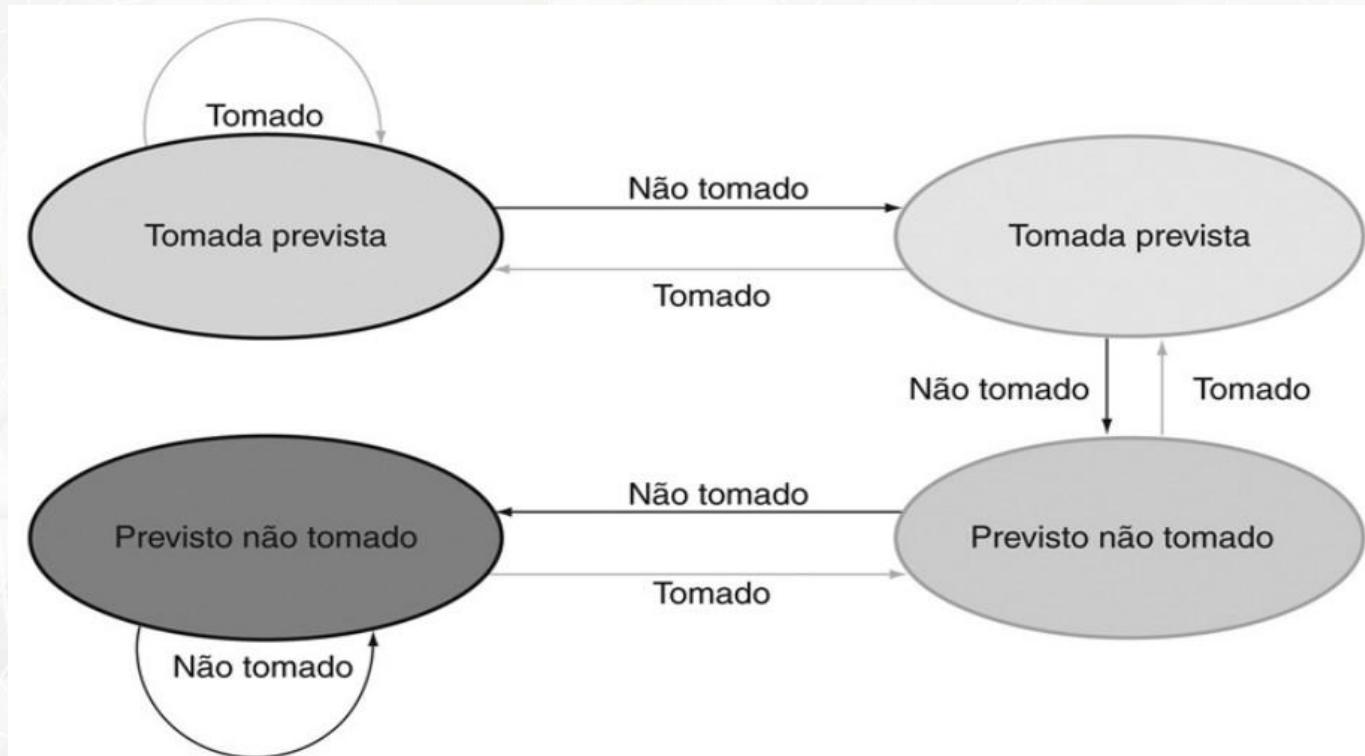
- Uma implementação dessa técnica é um buffer de previsão de desvios ou tabela de histórico de desvios.
- Um buffer de previsão de desvios é uma pequena memória indexada pela parte menos significativa do endereço da instrução de desvio.
- A memória contém um bit que diz se o desvio foi tomado recentemente ou não.
 - - Esse esquema simples de previsão de 1 bit tem um problema de desempenho:
 - Mesmo que um desvio quase sempre seja tomado, provavelmente faremos uma previsão incorreta duas vezes, em vez de uma, quando ele não for tomado;
- Exemplo: Considere um desvio de loop que se desvia nove vezes seguidas, depois não é tomado uma vez. Qual é a exatidão da previsão para esse desvio, supondo que o bit de previsão para o desvio permaneça no buffer de previsão?
 - -R: A exatidão da previsão para esse desvio tomado 90% do tempo é apenas de 80% (duas previsões incorretas contra oito corretas).

Previsão Dinâmica de Desvios

- Uma implementação dessa técnica é um buffer de previsão de desvios ou tabela de histórico de desvios.
- Um buffer de previsão de desvios é uma pequena memória indexada pela parte menos significativa do endereço da instrução de desvio.
- A memória contém um bit que diz se o desvio foi tomado recentemente ou não.
 - - Esse esquema simples de previsão de 1 bit tem um problema de desempenho:
 - Mesmo que um desvio quase sempre seja tomado, provavelmente faremos uma previsão incorreta duas vezes, em vez de uma, quando ele não for tomado;
- Exemplo: Considere um desvio de loop que se desvia nove vezes seguidas, depois não é tomado uma vez. Qual é a exatidão da previsão para esse desvio, supondo que o bit de previsão para o desvio permaneça no buffer de previsão?
 - -R: A exatidão da previsão para esse desvio tomado 90% do tempo é apenas de 80% (duas previsões incorretas contra oito corretas).

Previsão Dinâmica de Desvios

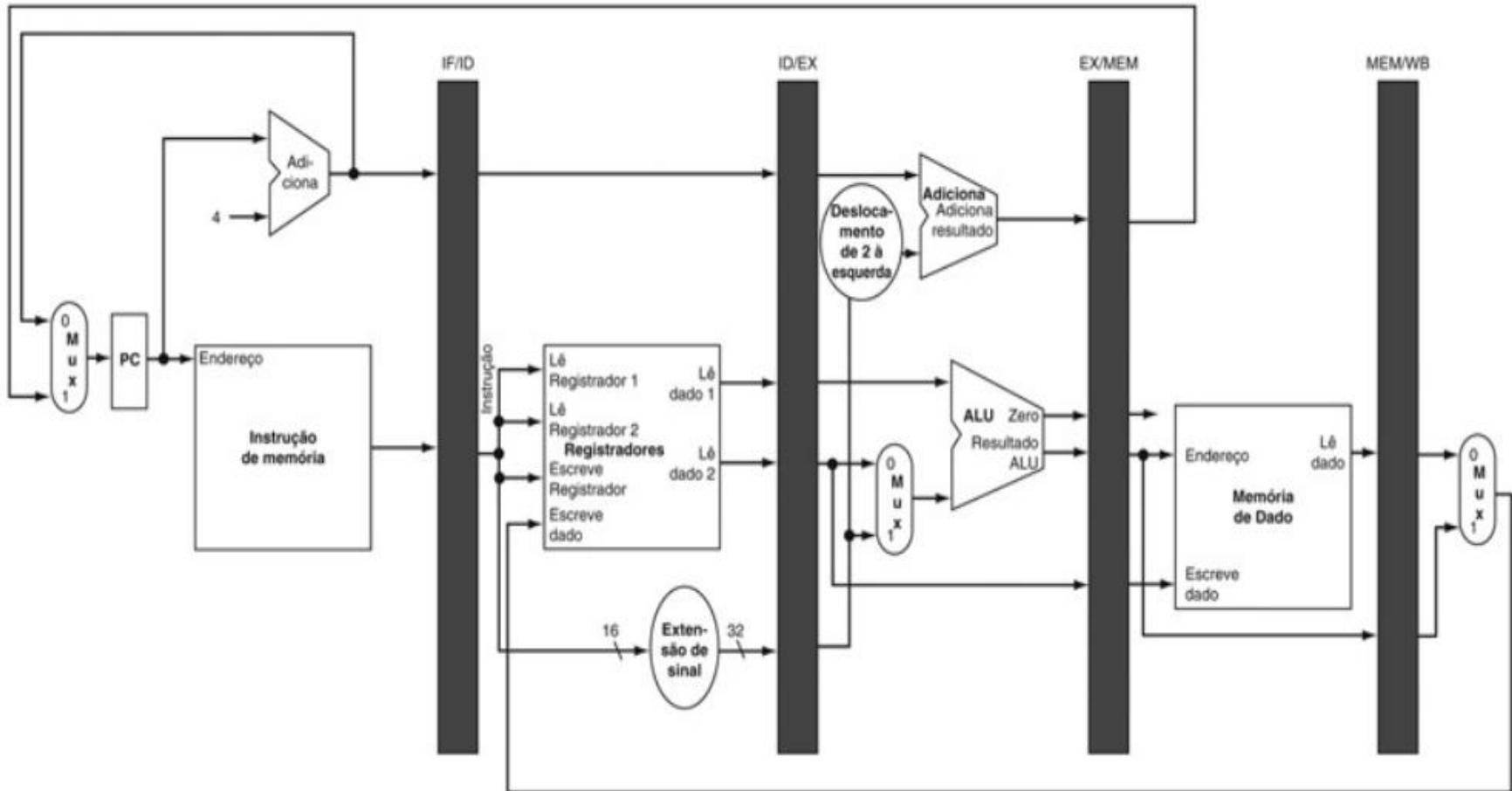
- Alternativa para lidar com os erros: Utilizar 2 bits.



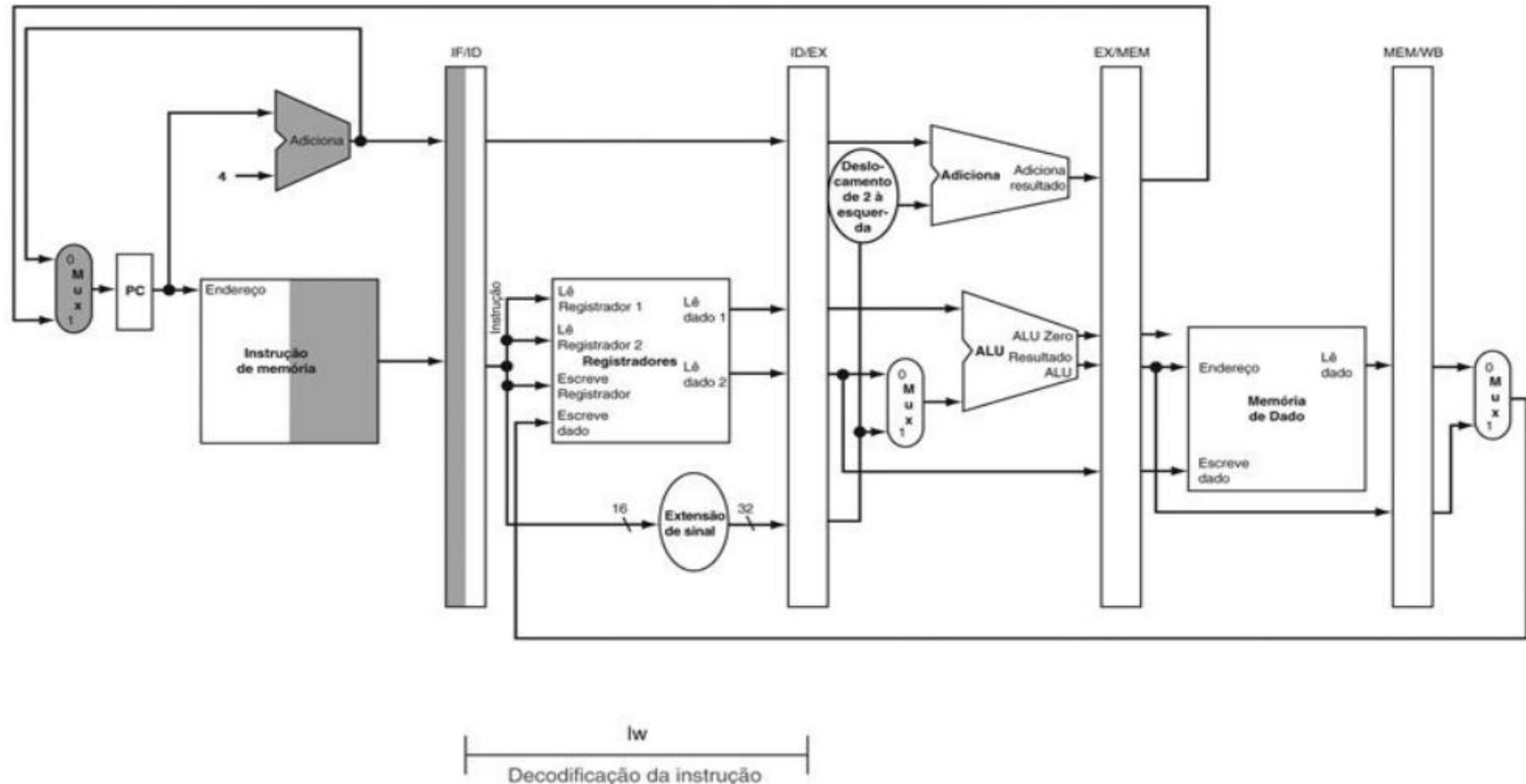
Exercícios de Fixação: Previsão Dinâmica

- Disponível no moodle

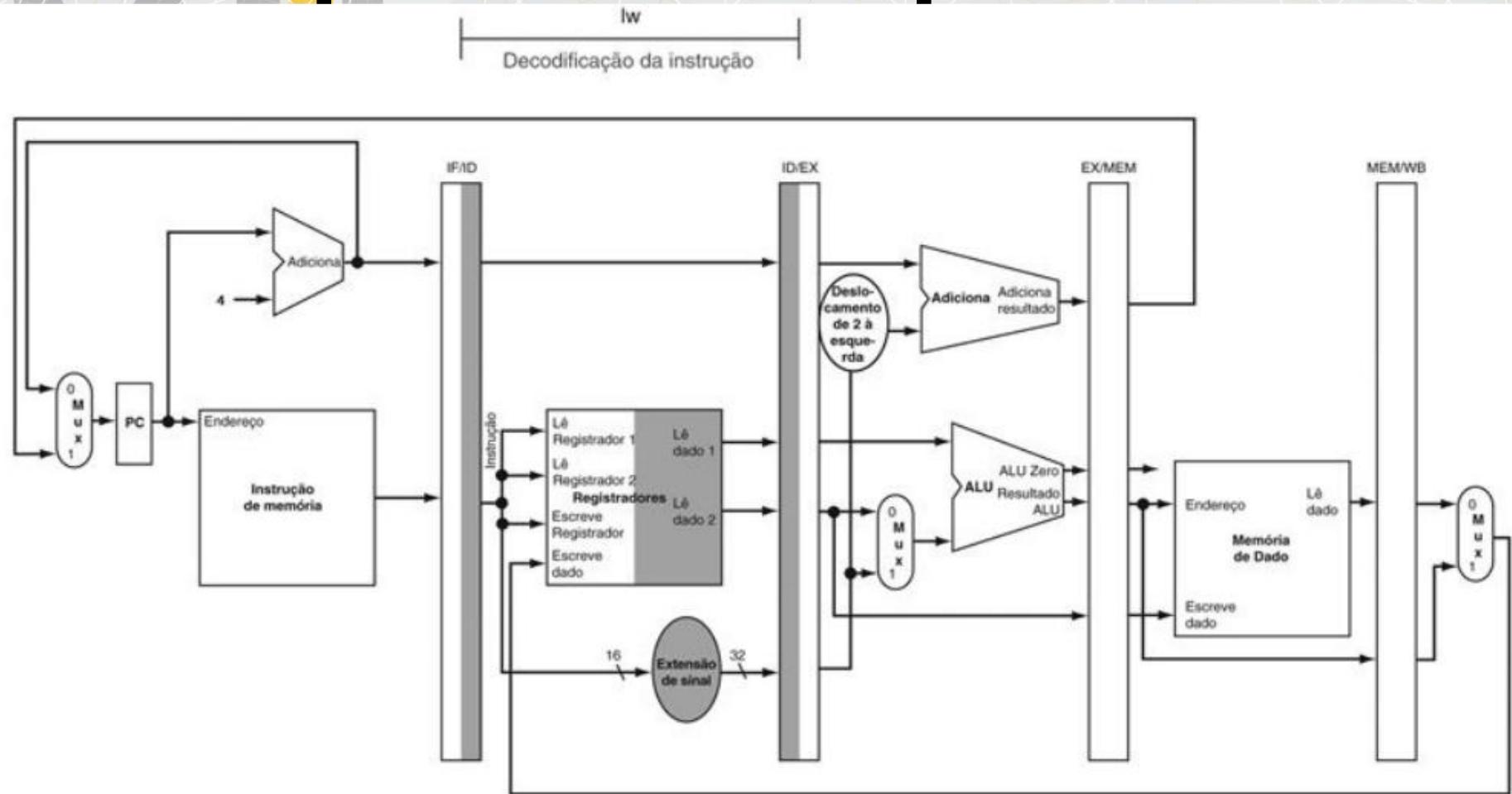
Datapath com Pipeline



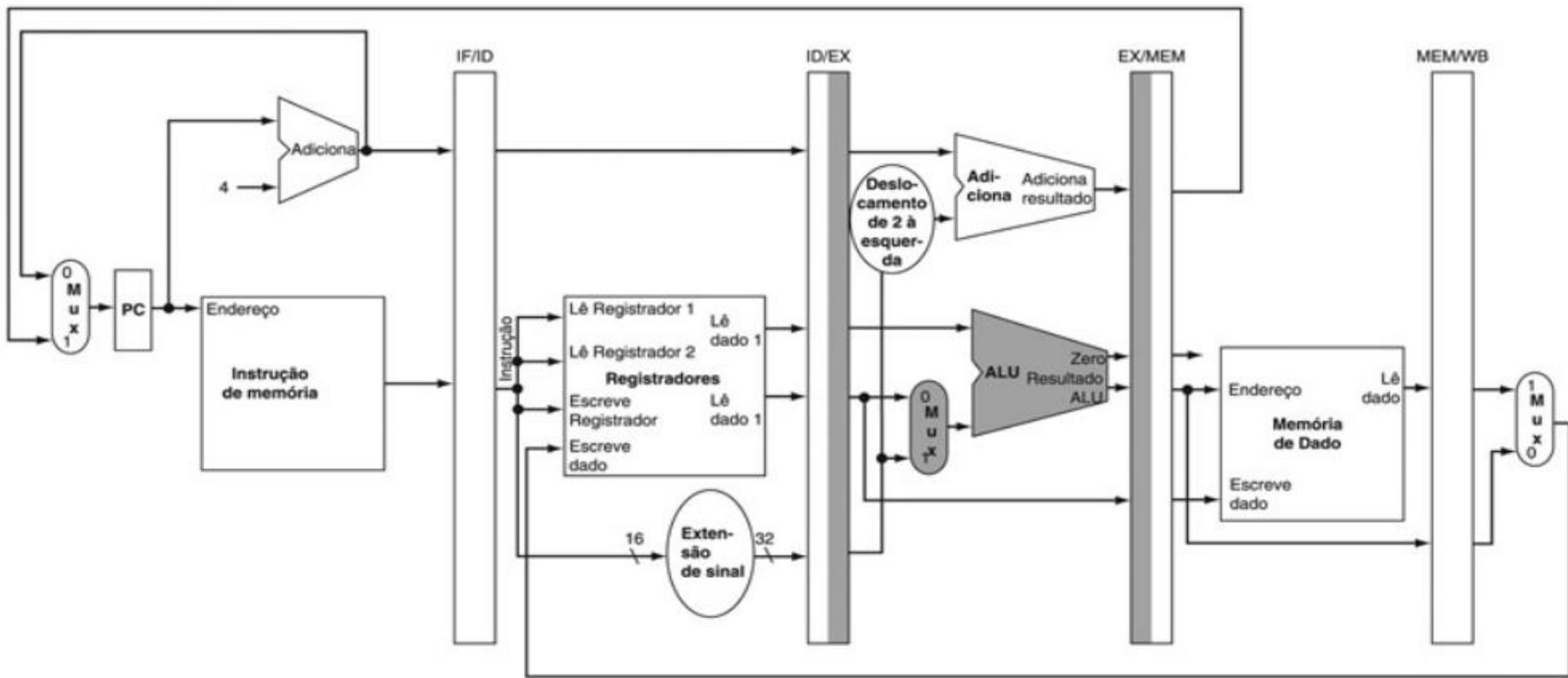
Datapath com Pipeline



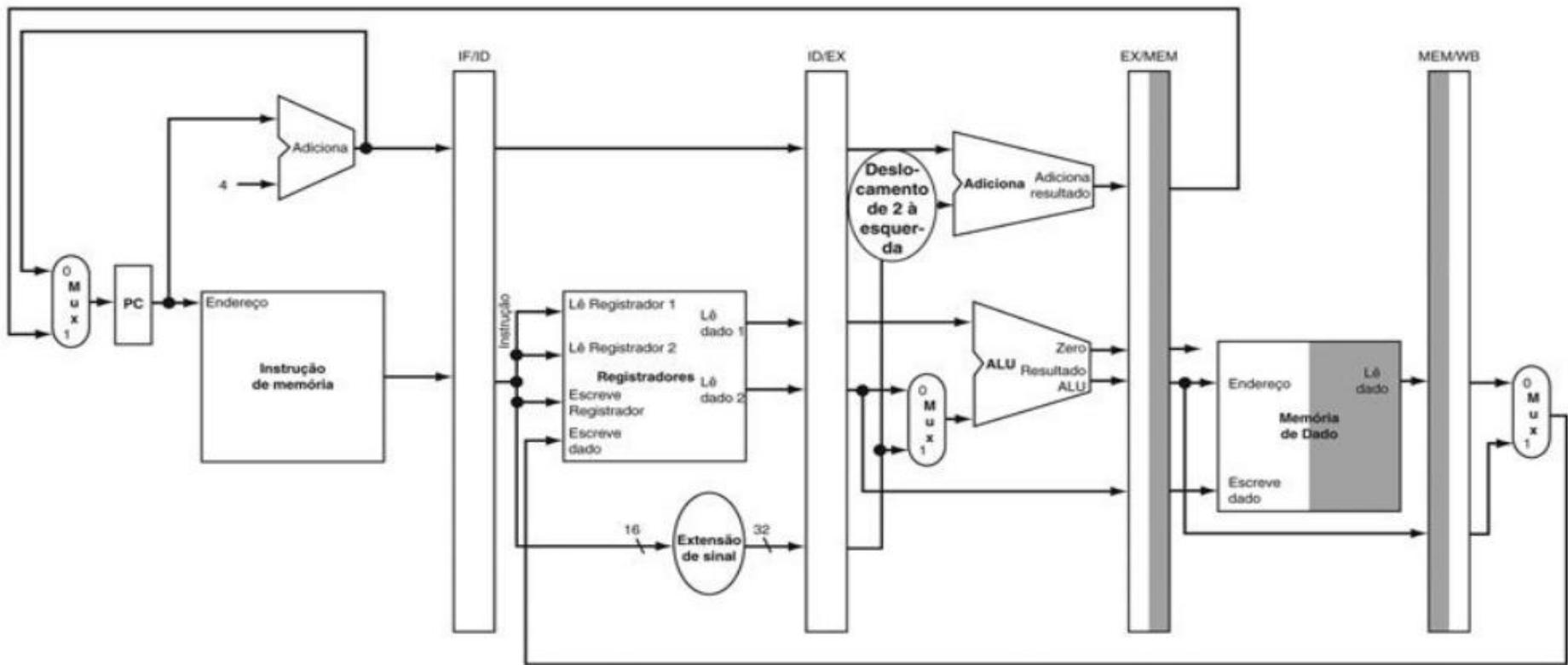
Datapath com Pipeline



Datapath com Pipeline

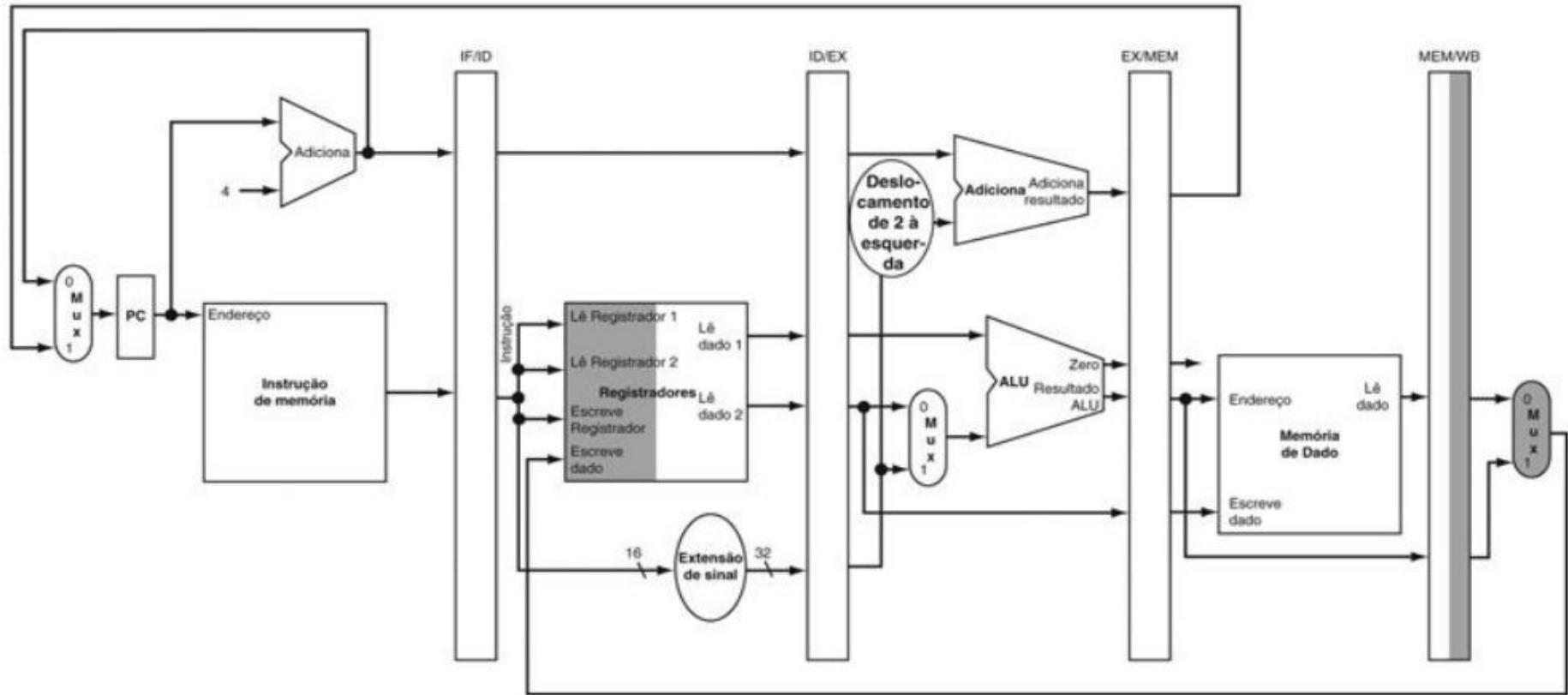


Datapath com Pipeline

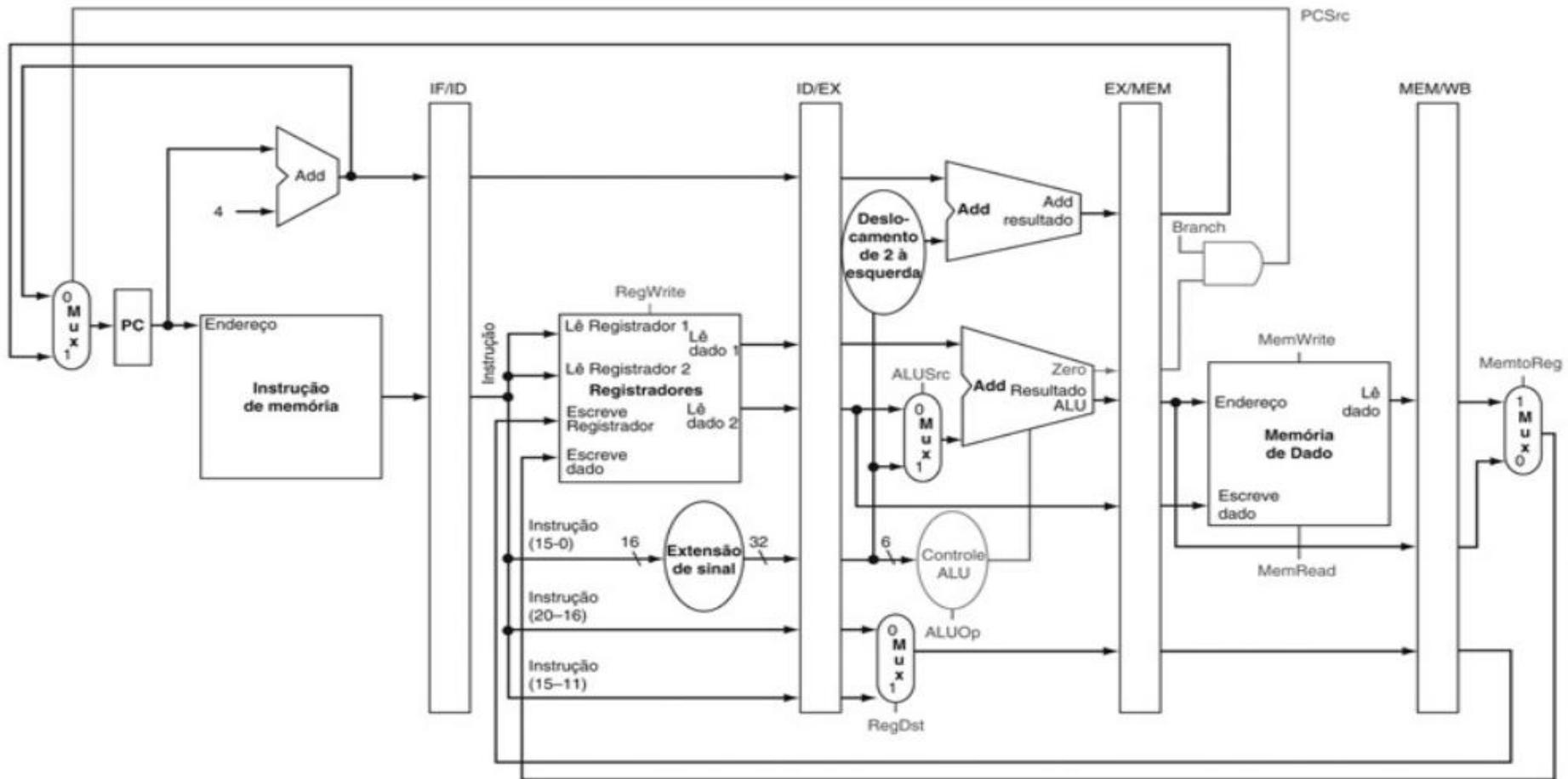


Datapath com Pipeline

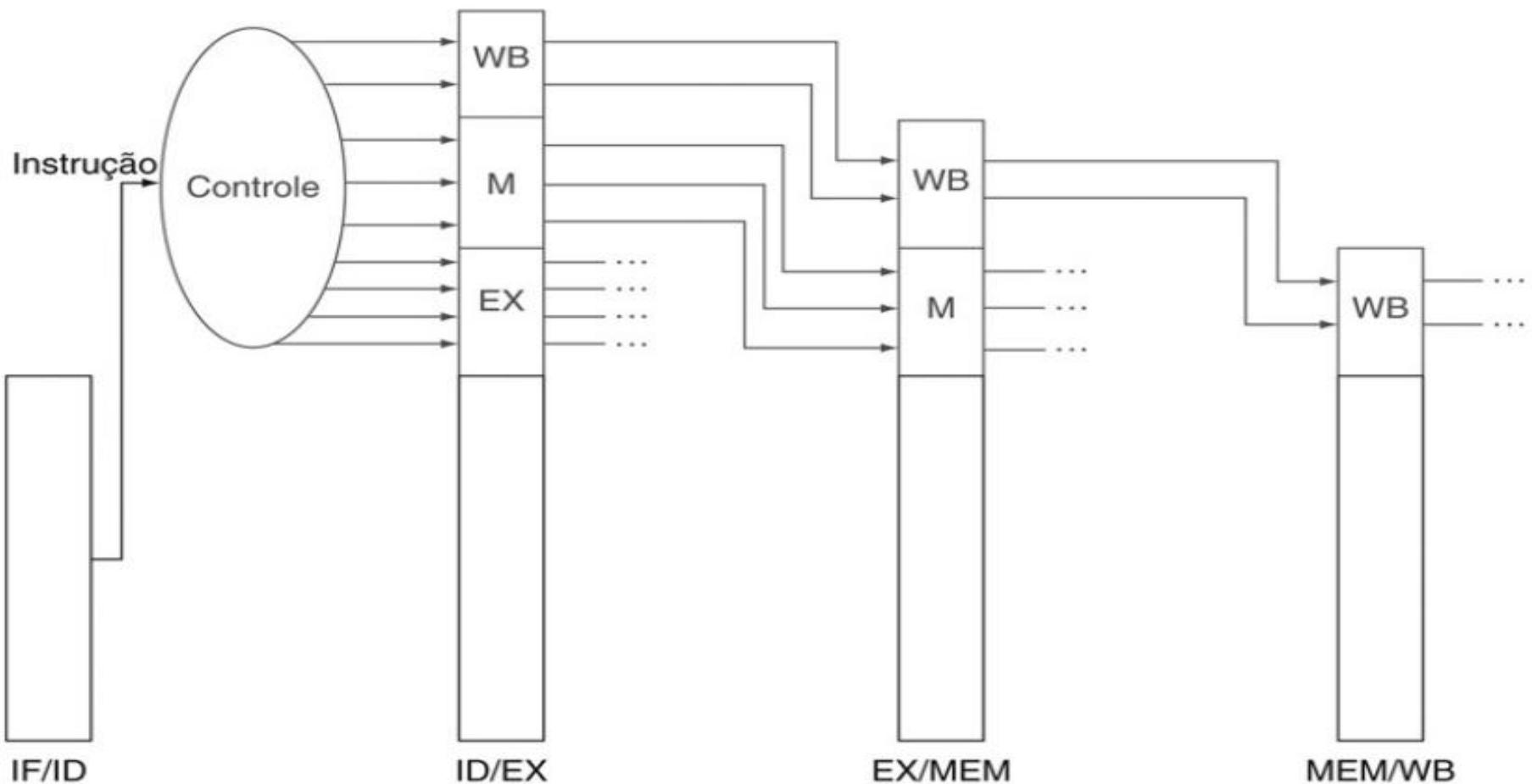
lw
Write-back



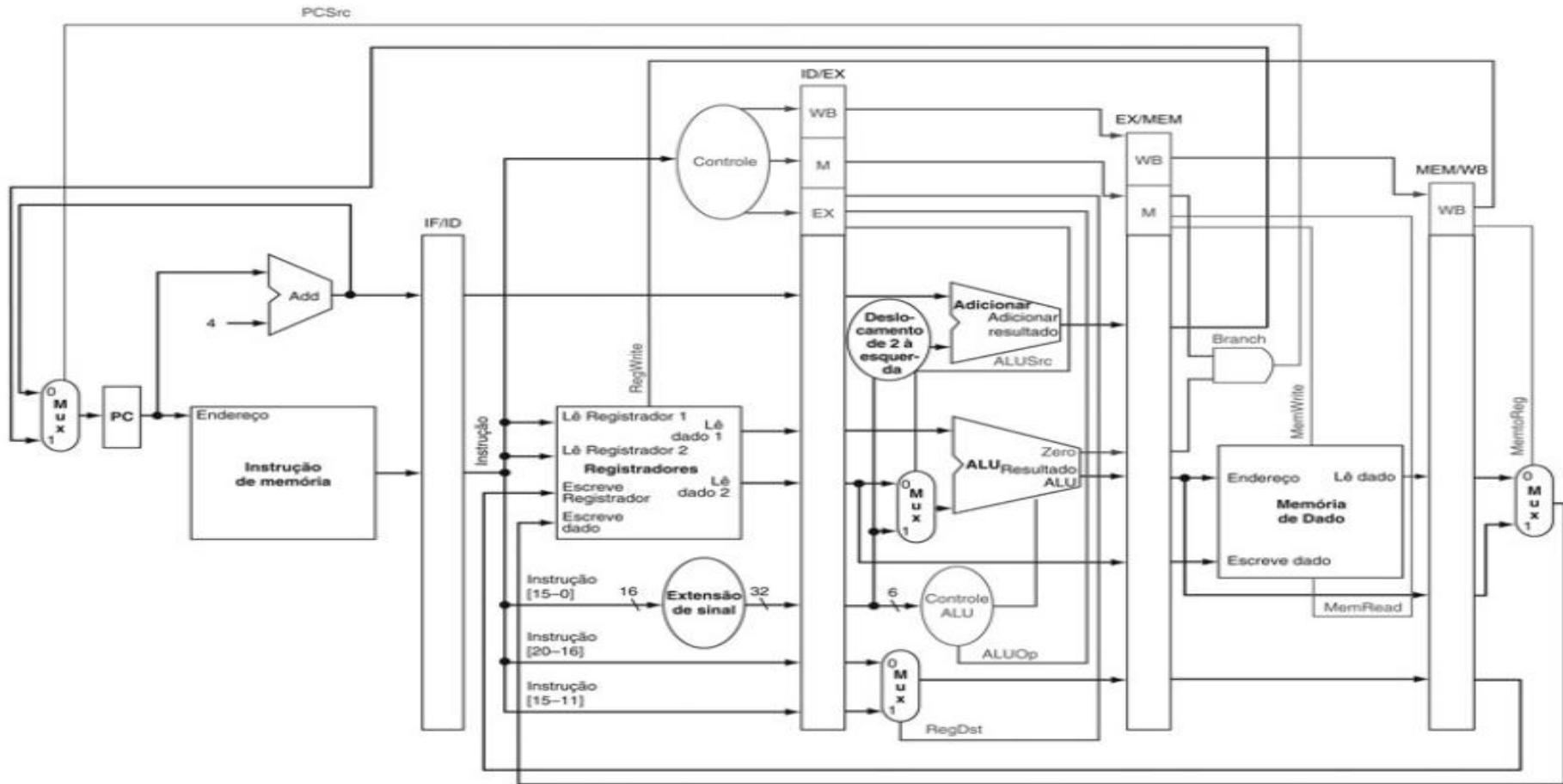
Datapath com Pipeline



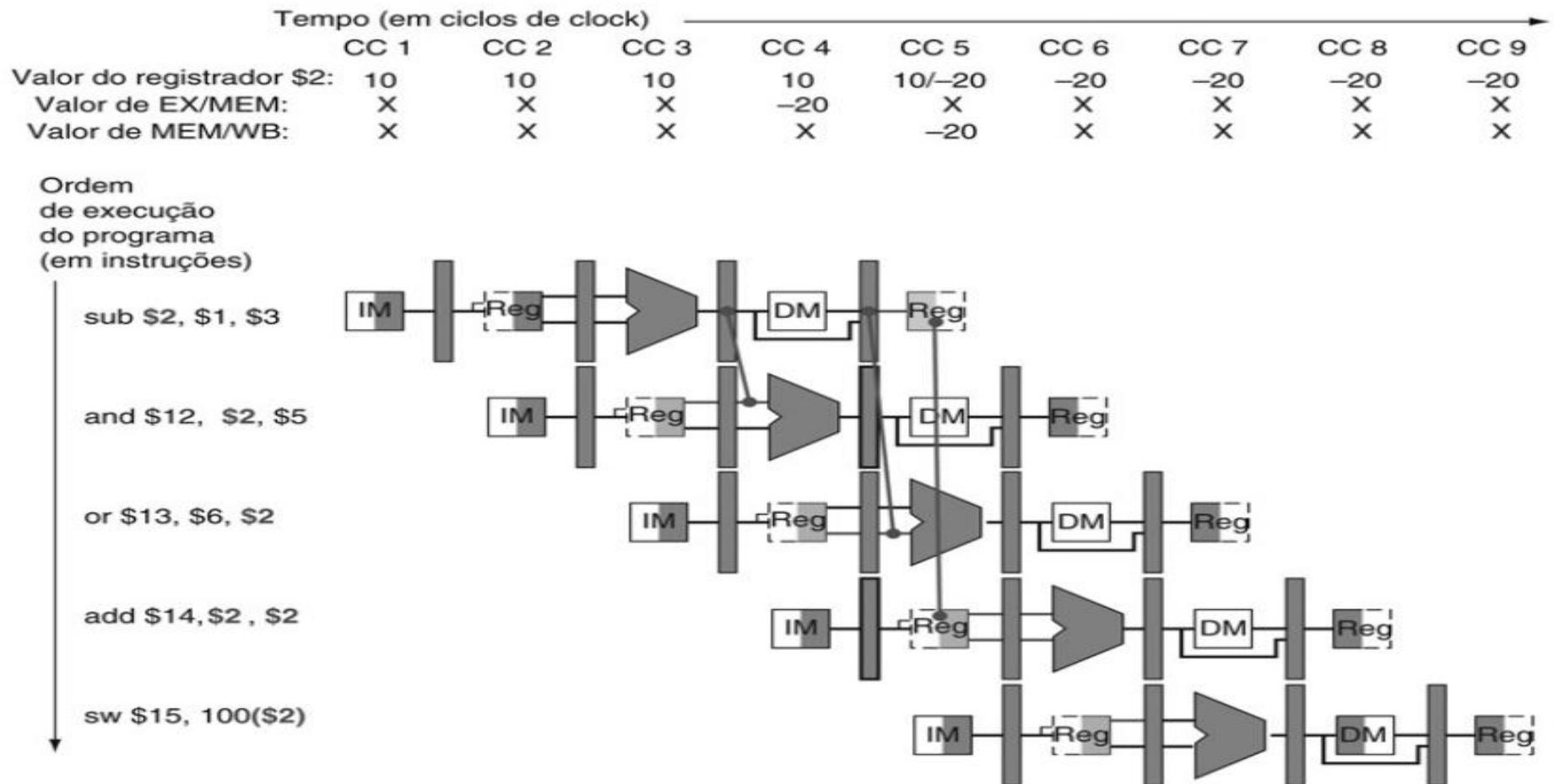
Datapath com Pipeline



Datapath com Pipeline

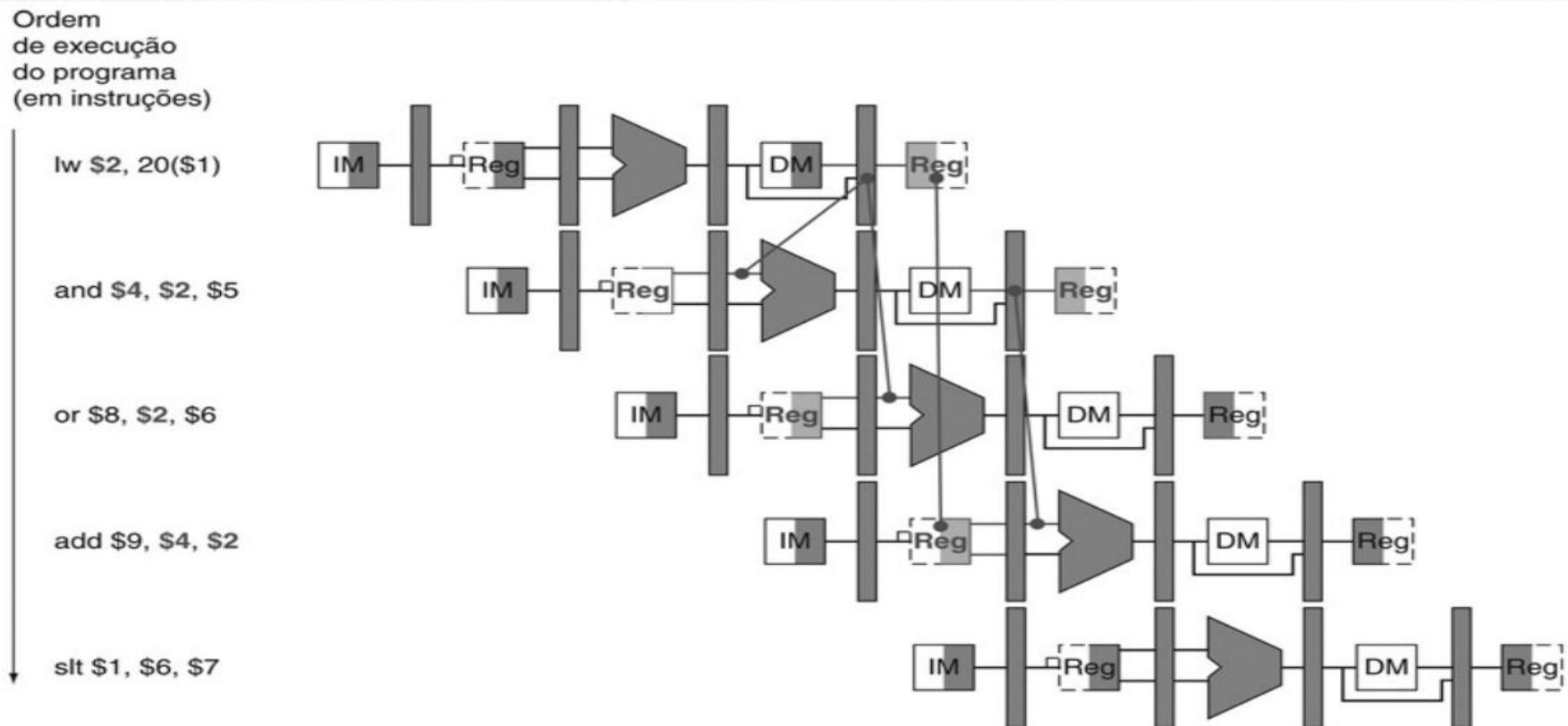


Datapath com Pipeline



Datapath com Pipeline

- O forwarding não pode salvar o dia quando uma instrução tenta ler um registrador após uma instrução **load** que escreve no mesmo registrador.

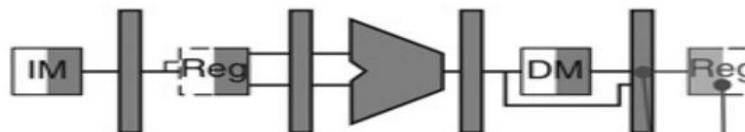


Datapath com Pipeline

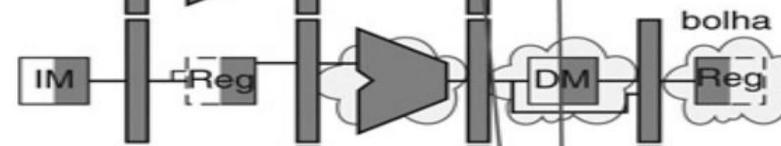
- Os dados ainda são lidos da memória no ciclo de clock 4, enquanto a ALU está realizando a operação para a instrução seguinte. Algo precisa ocasionar um stall no pipeline para a combinação de load seguida por uma instrução que lê seu resultado.

Ordem
de execução
do programa
(em instruções)

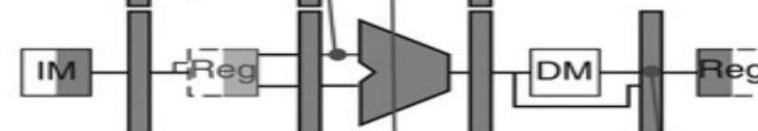
Iw \$2, 20(\$1)



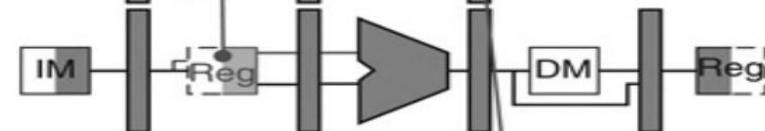
and se torna nop



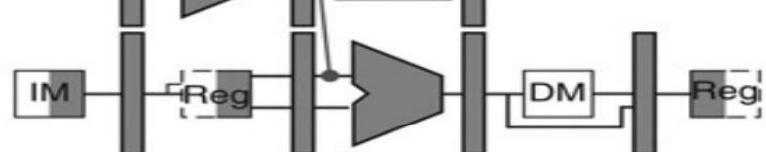
and \$4, \$2, \$5



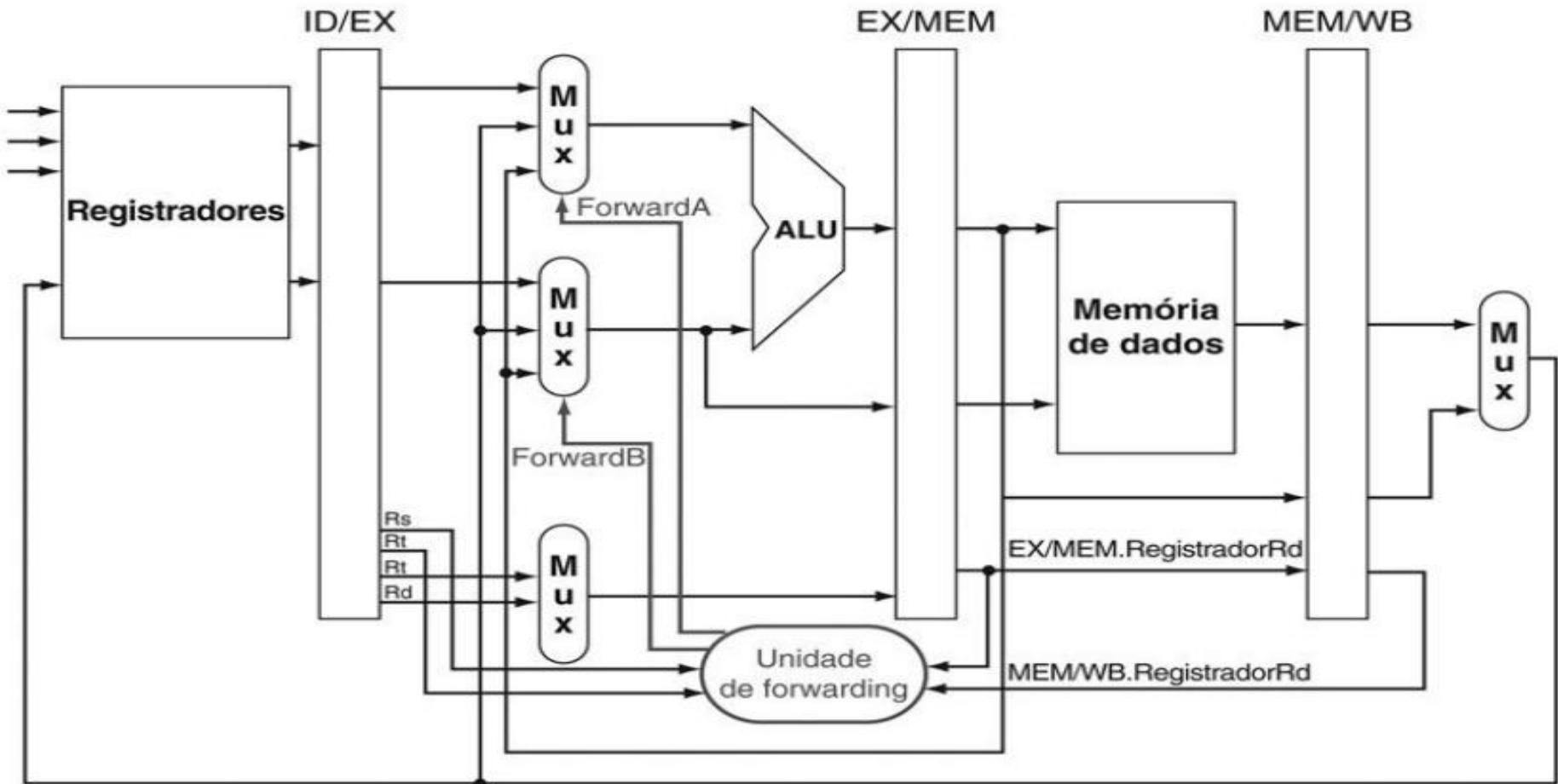
or \$8, \$2, \$6



add \$9, \$4, \$2



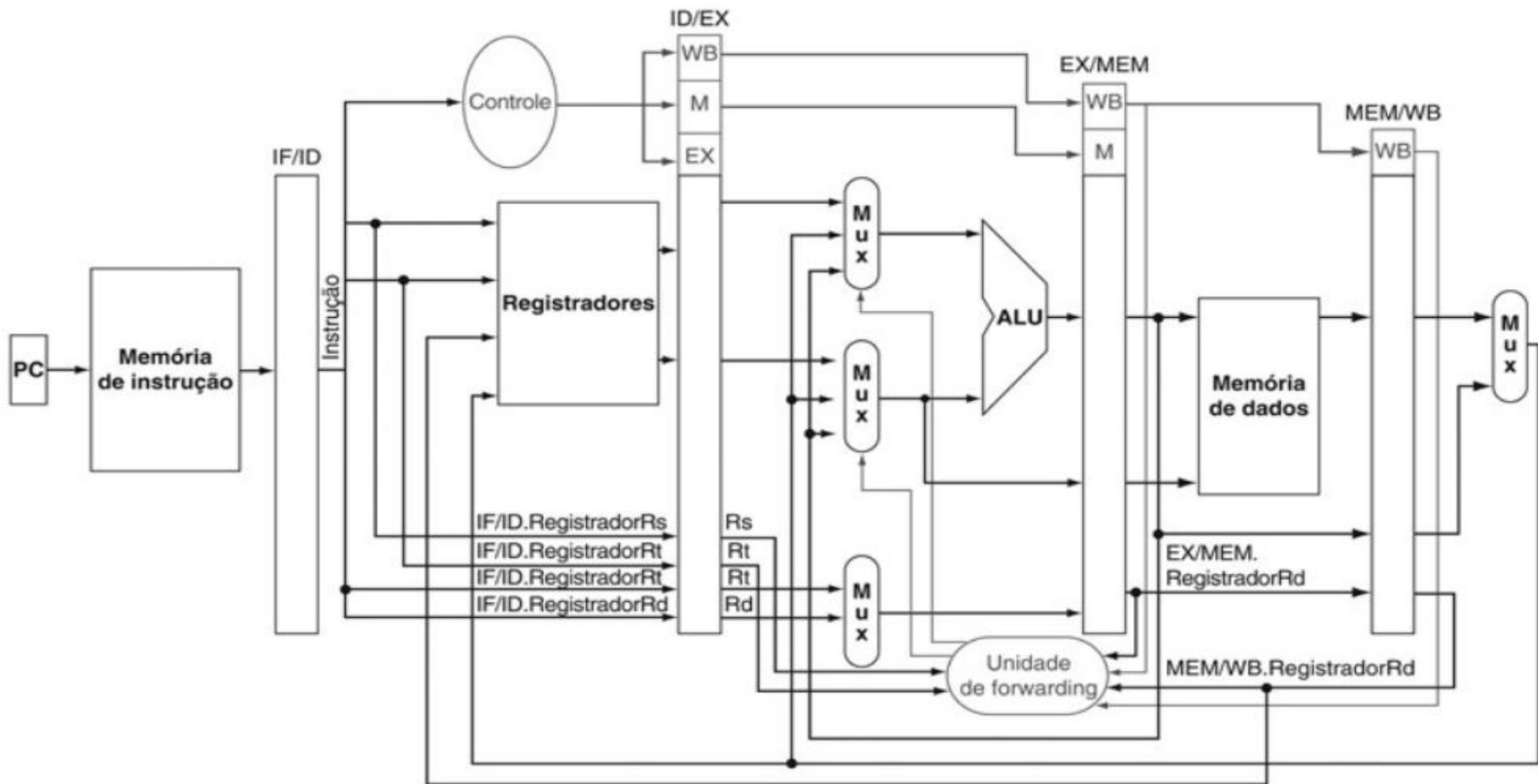
Datapath com Pipeline



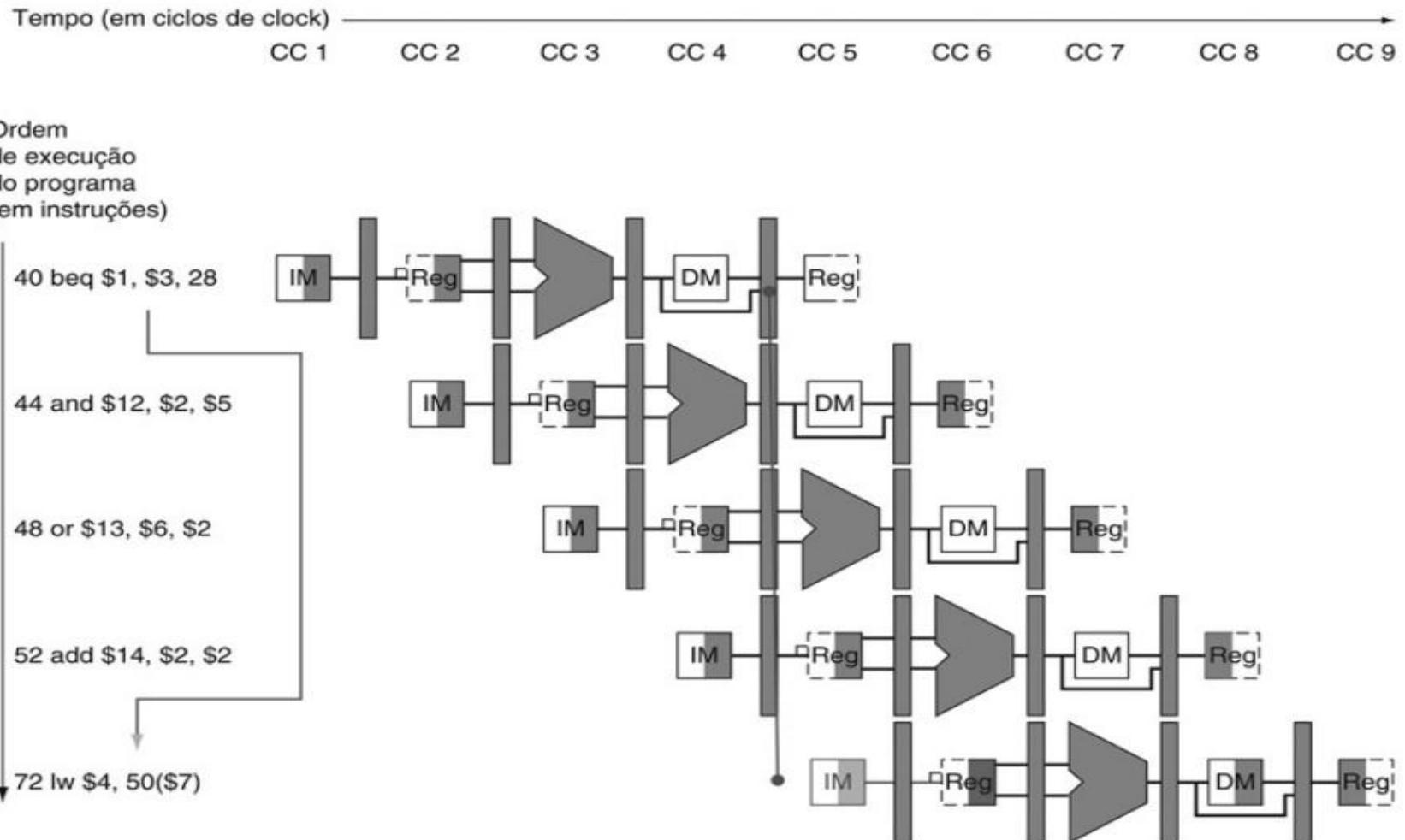
Datapath com Pipeline

Controle do Mux	Origem	Explicação
ForwardA = 00	ID/EX	O primeiro operando da ALU vem do banco de registradores.
ForwardA = 10	EX/MEM	O primeiro operando da ALU sofre forwarding do resultado anterior da ALU.
ForwardA = 01	MEM/WB	O primeiro operando da ALU sofre forwarding da memória de dados ou de um resultado anterior da ALU.
ForwardB = 00	ID/EX	O segundo operando da ALU vem do banco de registradores.
ForwardB = 10	EX/MEM	O segundo operando da ALU sofre forwarding do resultado anterior da ALU.
ForwardB = 01	MEM/WB	O segundo operando da ALU sofre forwarding da memória de dados ou de um resultado anterior da ALU.

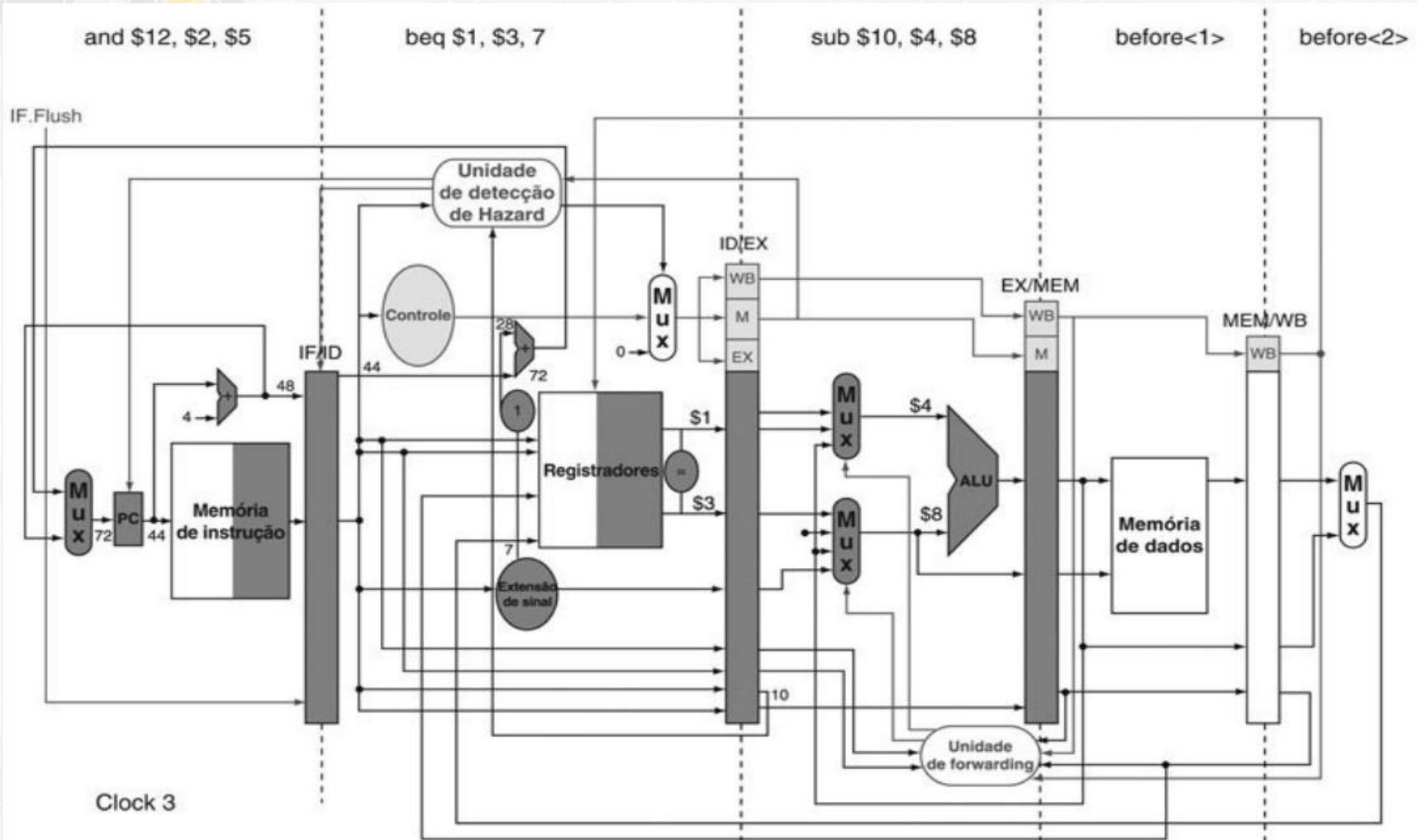
Datapath com Pipeline



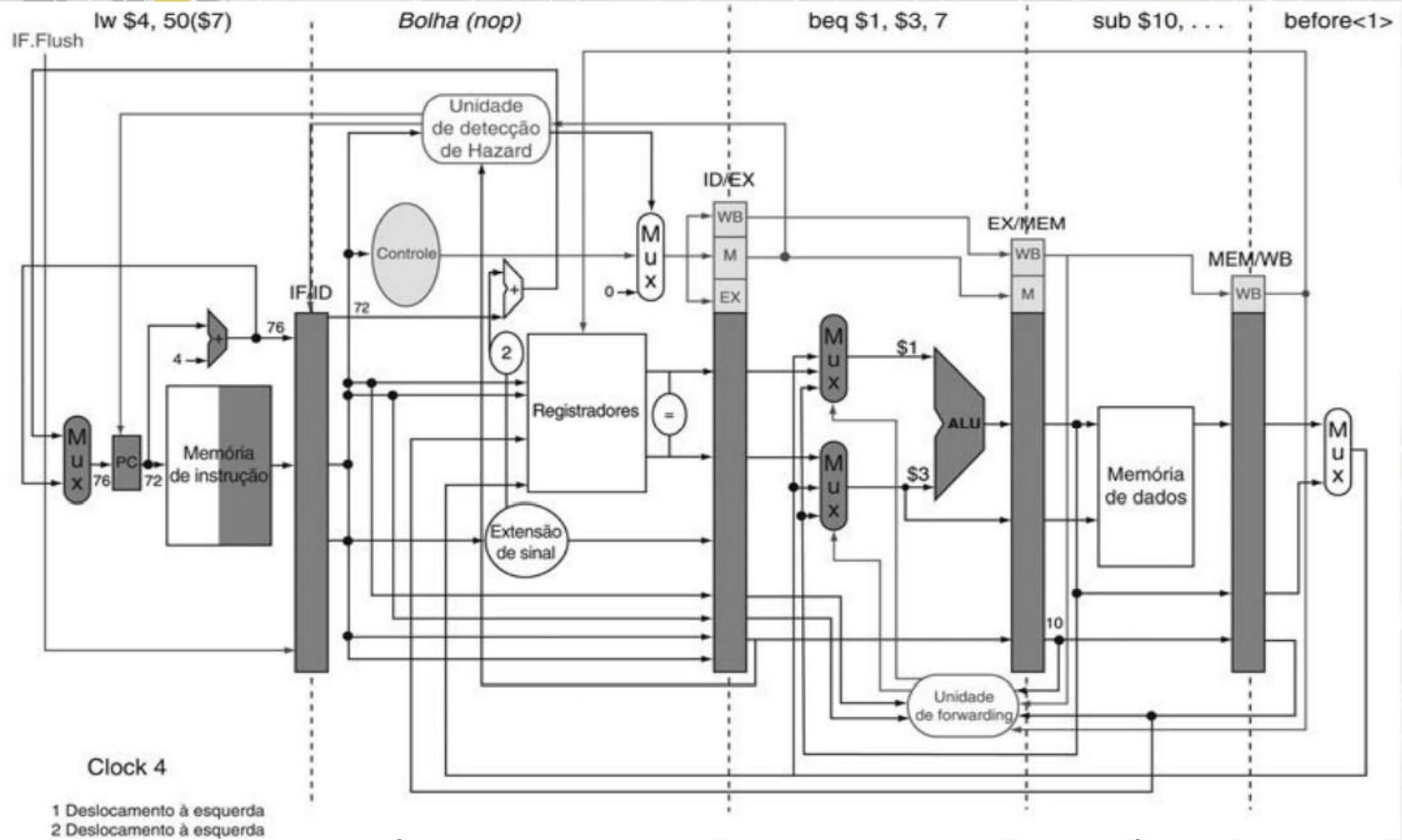
Previsão de Desvio (Hazard C)



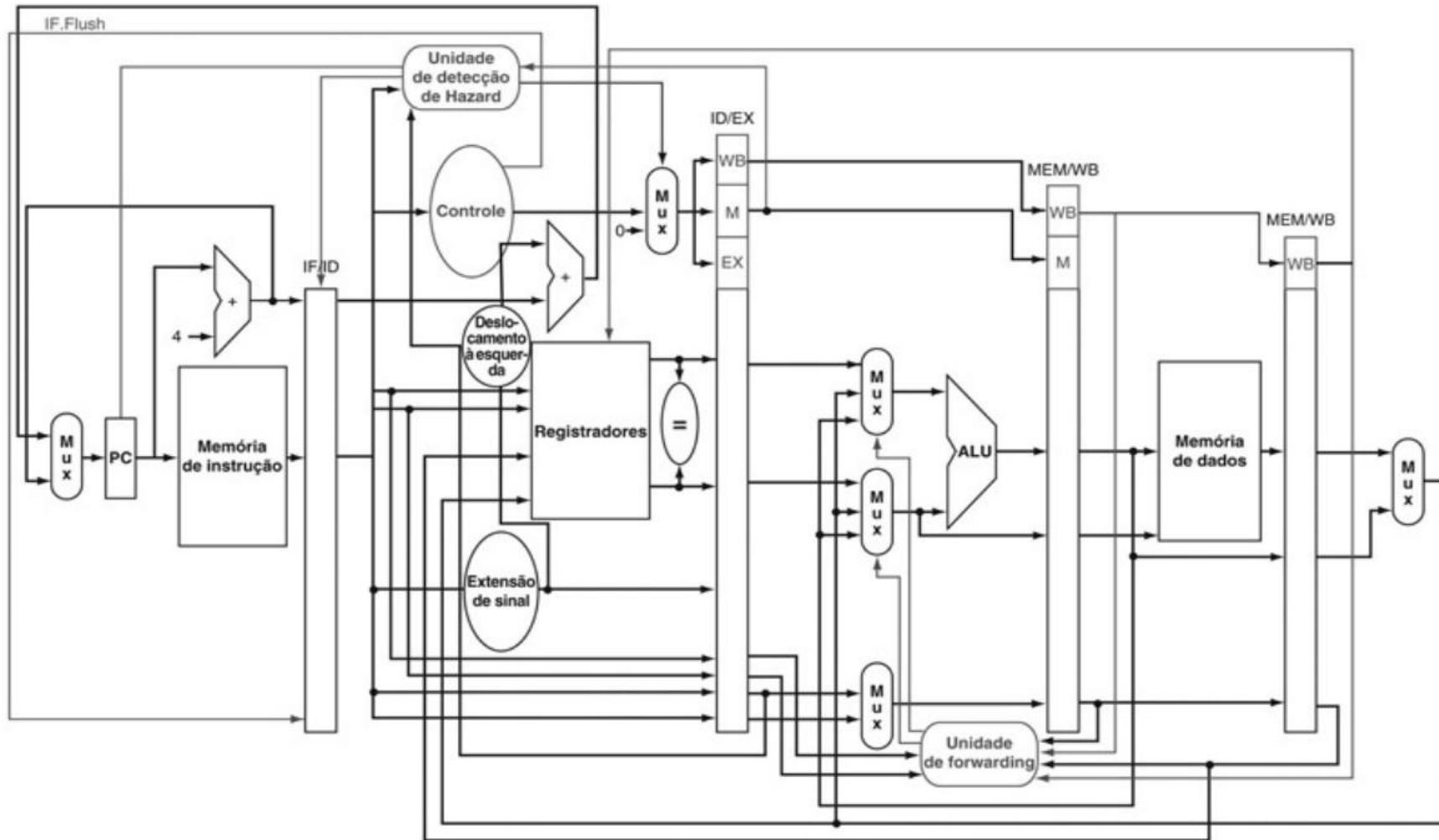
Detecção de Hazard



Detecção de Hazard



Datapath com Previsão



Exceções

- **Exceções:**
 - Também chamada interrupção. Um evento não programado que interrompe a execução do programa; usada para detectar overflow.
- **Interrupção:**
 - Uma exceção que vem de fora do processador. (Algumas arquiteturas utilizam o termo interrupção para todas as exceções.)
- Seguimos a convenção do MIPS, usando o termo exceção para indicar qualquer mudança inesperada no fluxo de controle, sem distinguir se a causa é interna ou externa;

Exceções

Tipo de evento	De onde?	Terminologia MIPS
Solicitação de dispositivo de E/S	Externa	Interrupção
Chamar o sistema operacional do programa do usuário	Interna	Exceção
Overflow aritmético	Interna	Exceção
Usar uma instrução indefinida	Interna	Exceção
Defeitos do hardware	Ambos	Exceção ou interrupção

Tabela: exemplos mostrando se a situação é gerada internamente pelo processador ou se é gerada externamente



Exceções

- As exceções podem ser implementadas via:
 - Registradores:
 - A ação básica que o processador deve realizar quando ocorre uma exceção é salvar o endereço da instrução causadora no contador de programa de exceção (Exception Program Counter — EPC) e depois transferir o controle para o sistema operacional em algum endereço especificado;
 - O sistema operacional pode tomar a ação apropriada, que pode ser fornecer algum serviço ao programa do usuário, tomar alguma ação predefinida em resposta a um overflow ou terminar a execução do programa e informar um erro;
 - Depois de realizar qualquer ação necessária devido à exceção, o sistema operacional pode terminar o programa ou pode continuar sua execução, usando o EPC para determinar onde reiniciar a execução do programa.
 - Vetor de interrupções:
 - Em uma interrupção vetorizada, o endereço ao qual o controle é transferido é determinado pela causa da exceção.

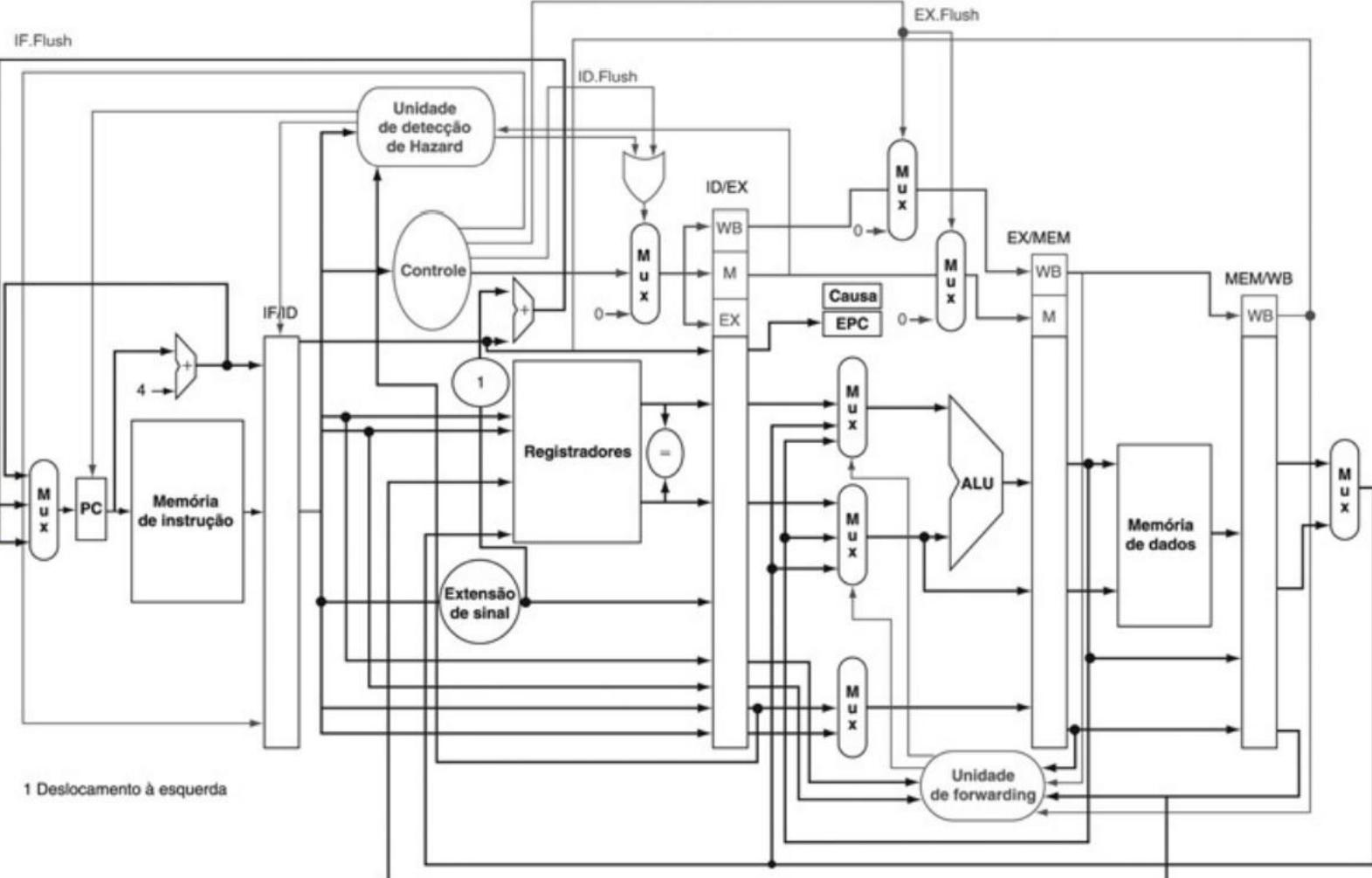
Exceções

- As exceções podem ser implementadas via:
 - Vetor de interrupções:
 - Em uma interrupção vetorizada, o endereço ao qual o controle é transferido é determinado pela causa da exceção;

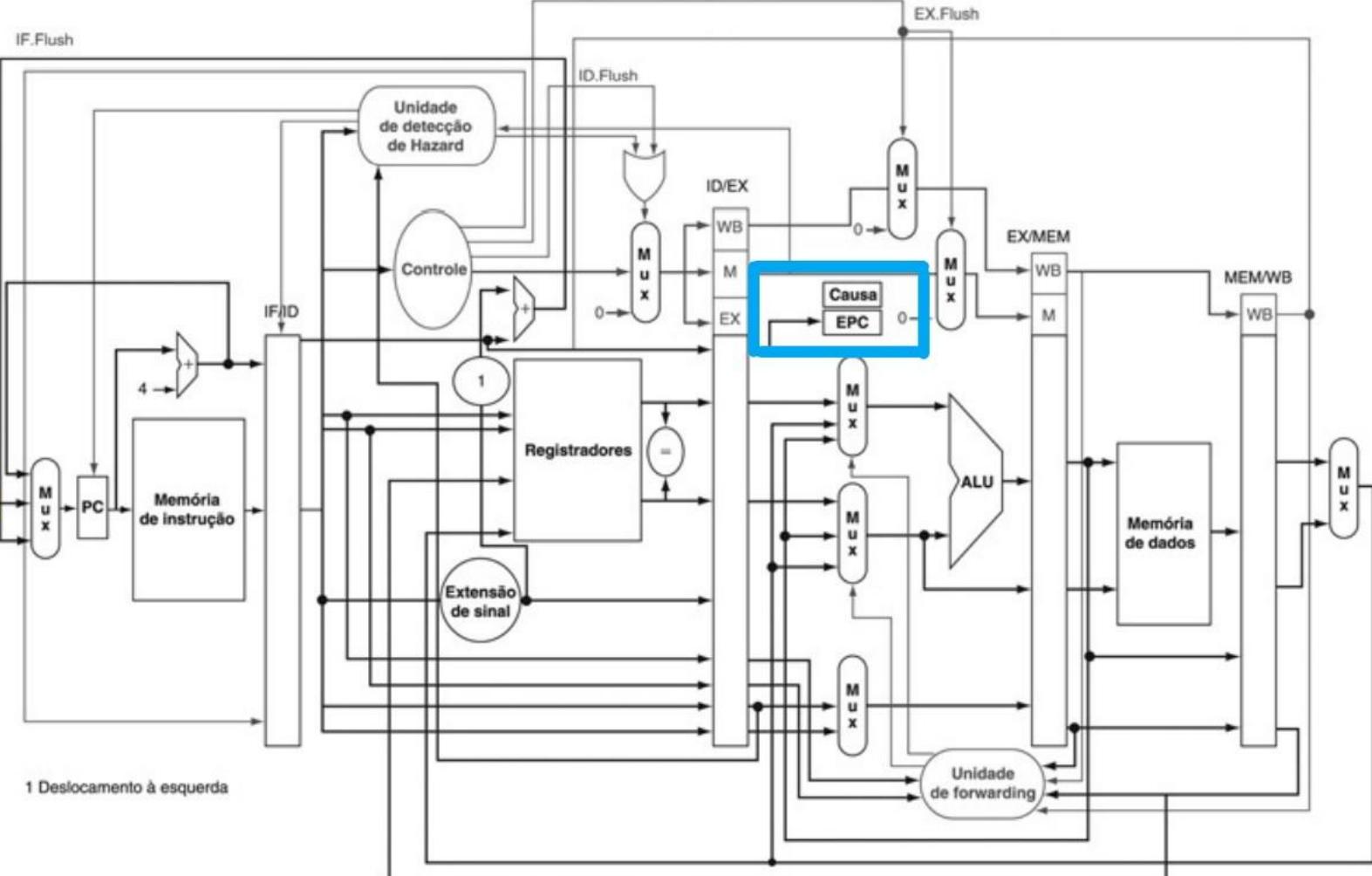
Tipo de exceção	Endereço do vetor de exceção (em hexa)
Instrução indefinida	8000 0000 _{hexa}
Overflow aritmético	8000 0180 _{hexa}

- Vamos supor que estejamos implementando o sistema de exceção utilizado na arquitetura MIPS, com o único ponto de entrada sendo o endereço 8000 0180 hexa, precisaremos acrescentar dois registradores;
 - EPC: Um registrador de 32 bits usado para manter o endereço da instrução afetada. (Esse registrador é necessário mesmo quando as exceções são vetorizadas.);
 - Cause: Um registrador usado para registrar a causa da exceção.

Exceções



Exceções



Exceções

- Exceção em um computador com pipeline
 - Exemplo:

40 _{hex}	sub	\$11,	\$2,	\$4
44 _{hex}	and	\$12,	\$2,	\$5
48 _{hex}	or	\$13,	\$2,	\$6
4C _{hex}	add	\$1,	\$2,	\$1
50 _{hex}	slt	\$15,	\$6,	\$7
54 _{hex}	lw	\$16,	50(\$7)	

Exceções

80000180 _{hex}	SW	\$26, 1000(\$0)
80000184 _{hex}	SW	\$27, 1004(\$0)
. . .		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4		1				1
2	and \$12, \$2, \$5						
3	or \$13, \$2, \$6						
4	add \$1, \$2, \$1						
5	slt \$15, \$6, \$7						
6	lw \$16, 50(\$7)						

Exceções

80000180 _{hex}	SW	\$26, 1000(\$0)
80000184 _{hex}	SW	\$27, 1004(\$0)
. . .		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4		1				1
2	and \$12, \$2, \$5		2	1			2
3	or \$13, \$2, \$6						
4	add \$1, \$2, \$1						
5	slt \$15, \$6, \$7						
6	lw \$16, 50(\$7)						

Exceções

80000180 _{hex}	SW	\$26, 1000(\$0)
80000184 _{hex}	SW	\$27, 1004(\$0)
• • •		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4	1					1
2	and \$12, \$2, \$5	2	1				2
3	or \$13, \$2, \$6	3	2	1			3
4	add \$1, \$2, \$1						
5	slt \$15, \$6, \$7						
6	lw \$16, 50(\$7)						

Exceções

80000180 _{hex}	SW	\$26, 1000(\$0)
80000184 _{hex}	SW	\$27, 1004(\$0)
• • •		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4	1					1
2	and \$12, \$2, \$5	2	1				2
3	or \$13, \$2, \$6	3	2	1			3
4	add \$1, \$2, \$1	4	3	2		1	4
5	slt \$15, \$6, \$7						
6	lw \$16, 50(\$7)						

Exceções

80000180 _{hex}	SW	\$26, 1000(\$0)
80000184 _{hex}	SW	\$27, 1004(\$0)
. . .		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4	1					1
2	and \$12, \$2, \$5	2	1				2
3	or \$13, \$2, \$6	3	2	1			3
4	add \$1, \$2, \$1	4	3	2			4
5	slt \$15, \$6, \$7	5	4	3		2	5
6	lw \$16, 50(\$7)						

Exceções

80000180 _{hex}	SW	\$26 , 1000(\$0)
80000184 _{hex}	SW	\$27 , 1004(\$0)
• • •		

Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4	1					1
2	and \$12, \$2, \$5	2	1				2
3	or \$13, \$2, \$6	3	2	1			3
4	add \$1, \$2, \$1	4	3	2			4
5	slt \$15, \$6, \$7	5	4	3		2	5
6	sw \$16, 50(\$7)	6	5	4 (Exception)			6

→ PC = 8000 0180 7

→ EPC = 4C+4

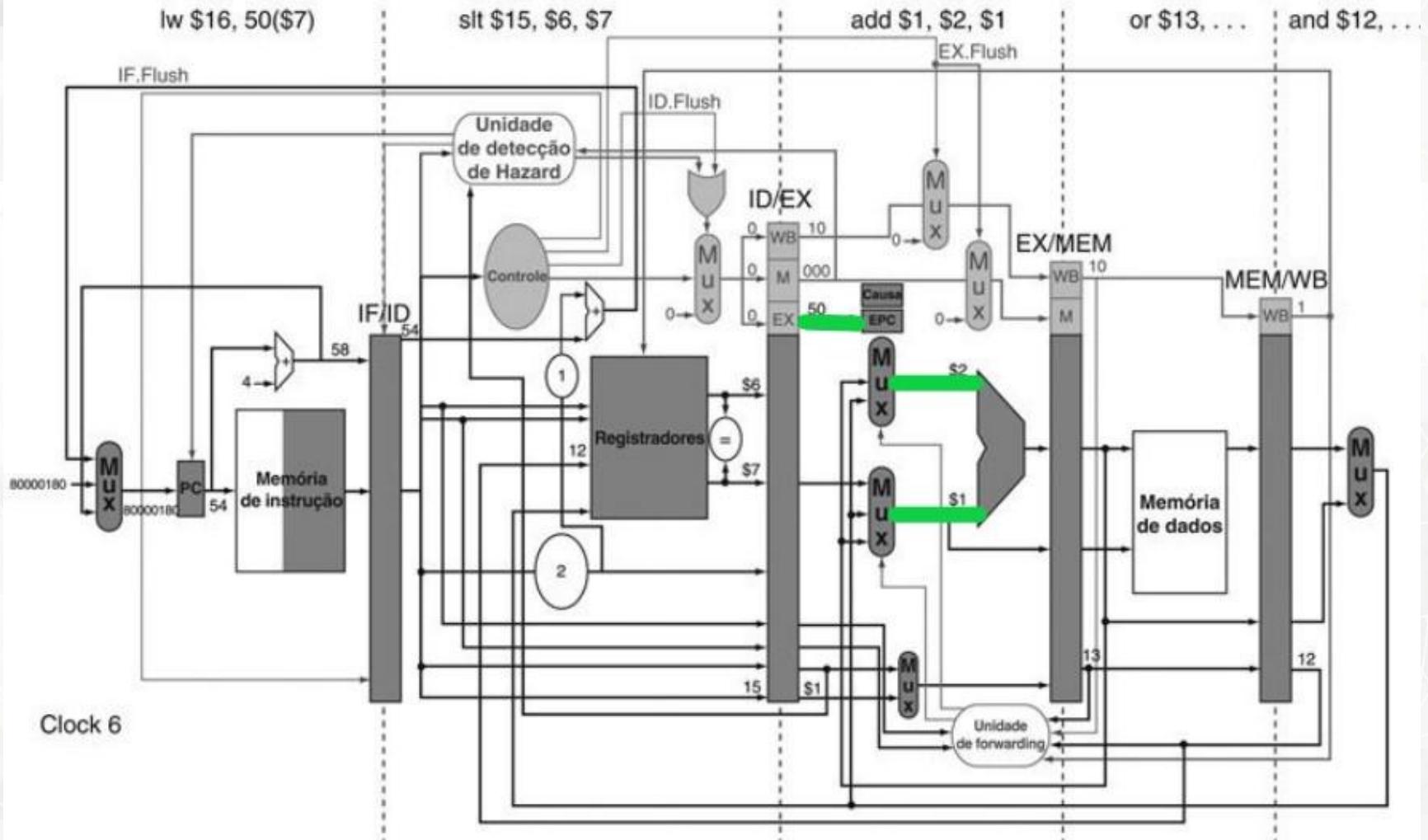
Exceções

80000180 _{hex}	SW	\$26 , 1000(\$0)
80000184 _{hex}	SW	\$27 , 1004(\$0)
• • •		

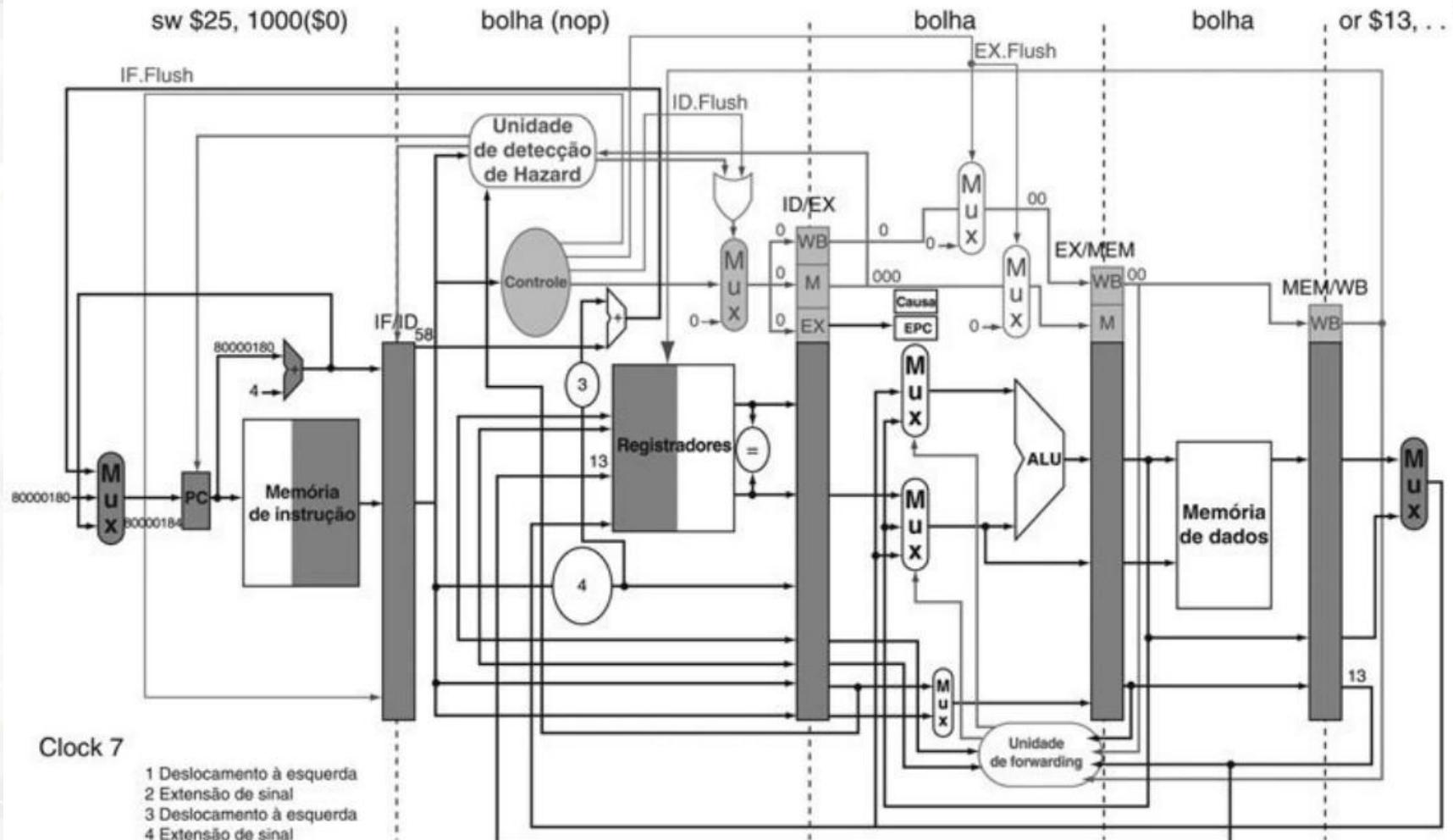
Pipe

		IF	ID	EX	DM	WB	Ciclo
1	sub \$11, \$2, \$4	1					1
2	and \$12, \$2, \$5	2	1				2
3	or \$13, \$2, \$6	3	2	1			3
4	add \$1, \$2, \$1	4	3	2			4
5	slt \$15, \$6, \$7	5	4	3		2	5
6	sw \$16, 50(\$7)	6	5	4 (Exception) (flush)	(flush)	(flush)	6 7

Exceções



Exceções



Referências

- PATTERSON, David A.; HENNESSY, John L. Organização e projeto de computadores: a interface hardware/software, 5. ed. Rio de Janeiro: Elsevier, 2017. ISBN 9788535287936.
- STALLINGS, William.; Arquitetura e Organização de Computadores. 10. ed. São Paulo: Pearson Education do Brasil, 2017. ISBN 9788543020532.
- TANENBAUM, A. S.; AUSTIN, T. Organização Estruturada de Computadores. 6. ed. São Paulo: Pearson Education do Brasil, 2013. ISBN 9788581435398.