

# Arquitetura de um processador

## Fundamentos de Arquitetura de Computadores

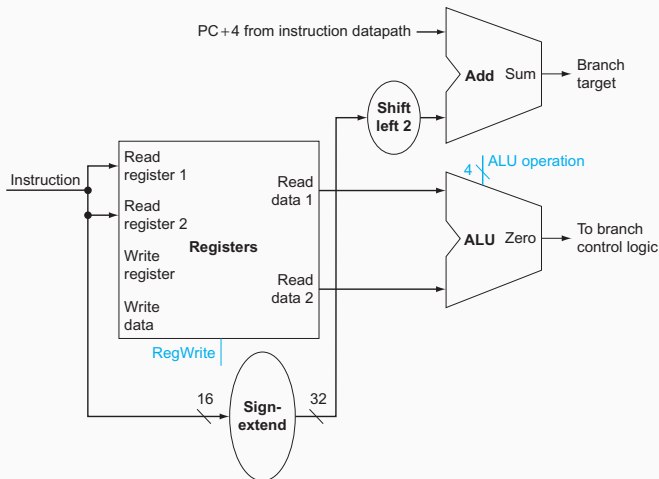
---

Prof. John Lenon C. Gardenghi

Faculdade do Gama  
Universidade de Brasília

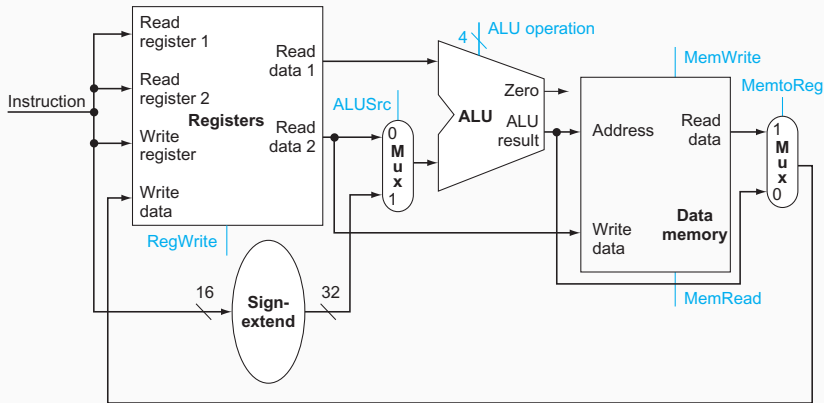
# Caminho de dados para desvio condicional

Exemplo: beq \$t0, \$t1, label.



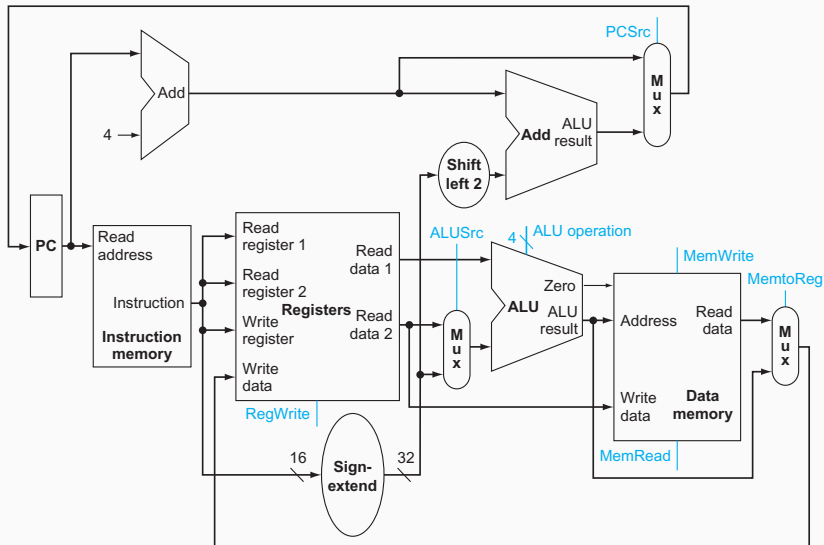
Extraído de [1, Figura 4.9].

# Caminho de dados simples



Extraído de [1, Figura 4.10].

# Caminho de dados simples - com busca de instrução



Extraído de [1, Figura 4.11].

Identificador	Operação
0000	and
0001	or
0010	add
0110	sub
0111	slt

Identificação das operações na ULA.

## Controle da ULA

ALUOp	Instrução	funct	ação desejada	controle da ULA
00	load word	xxxxxx	add	0010
00	store word	xxxxxx	add	0010
01	branch equal	xxxxxx	sub	0110
10	add	100000	add	0010
10	sub	100010	sub	0110
10	and	100100	and	0000
10	or	100101	or	0001
10	slt	101010	slt	0111

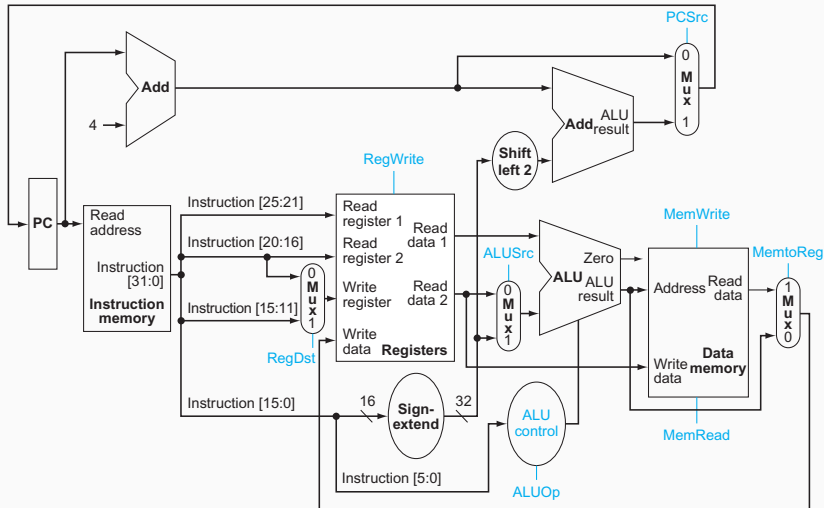
Combinação dos bits ALUOp e do campo funct. Note que sempre que ALUOp vale 10, os bits mais significativos de funct também valem 10.

ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Extraído de [1, Figura 4.13].

Tabela verdade entre os bits ALUOp e o campo funct. Como ALUOp nunca vale 11, então quando um deles valer 1, o outro pode ser desprezado.

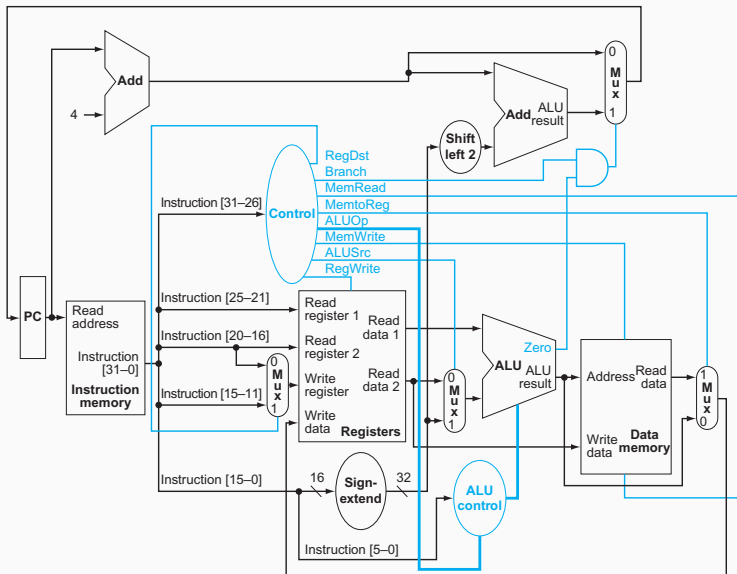
# Projeto do controle central



Extraído de [1, Figura 4.15].



# Projeto do controle central

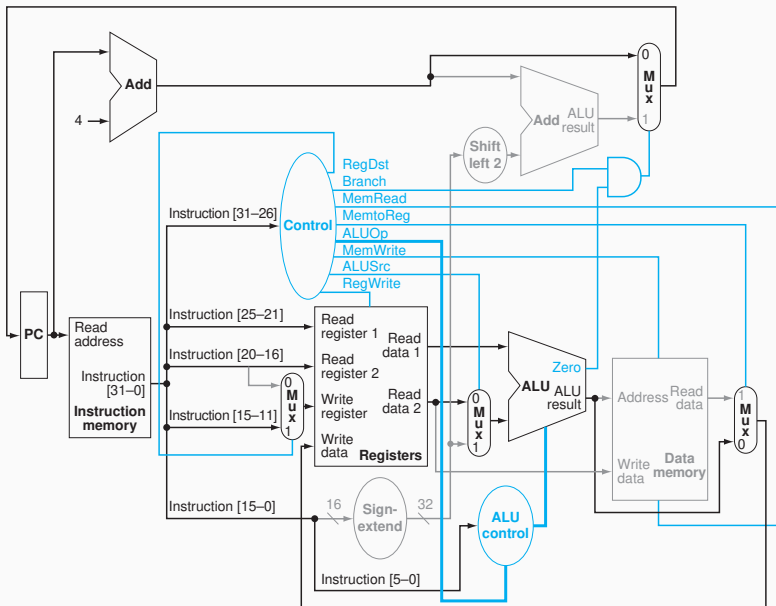


Extraído de [1, Figura 4.17].

# Definição dos sinais de controle

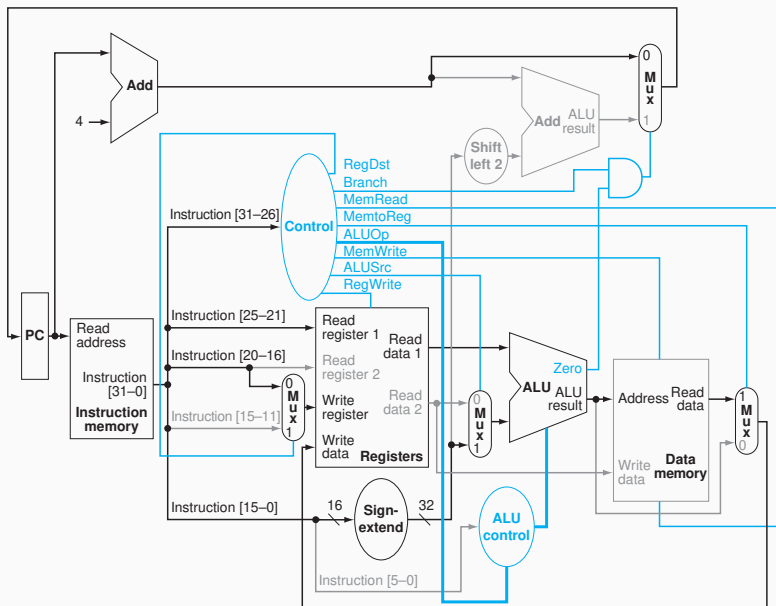
Instrução	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	PCSrc	ALUOp1	ALUOp0
formato R	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	x	1	x	0	0	1	0	0	0
beq	x	0	x	0	0	0	1	0	1

# Operação do caminho de dados - add \$t1, \$t2, \$t3



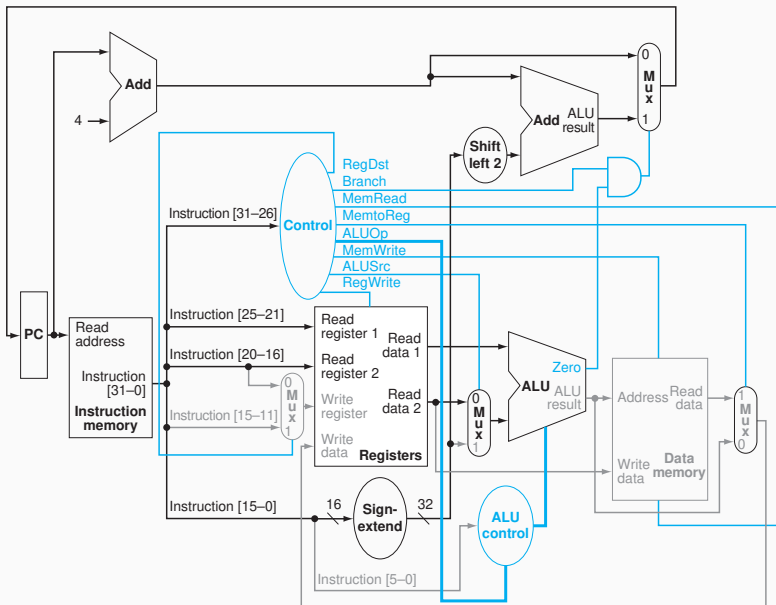
Extraído de [1, Figura 4.19].

# Operação do caminho de dados - load



Extraído de [1, Figura 4.20].

# Operação do caminho de dados - beq

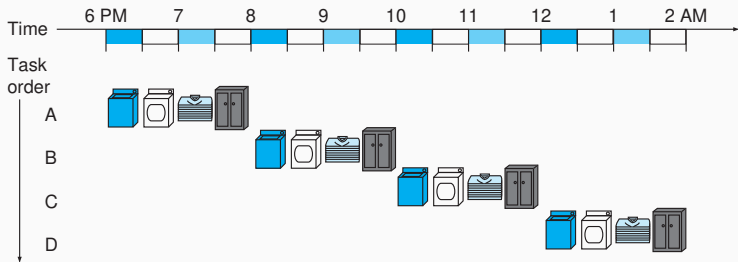


Extraído de [1, Figura 4.21].

# Pipeline - Lavadora de roupas

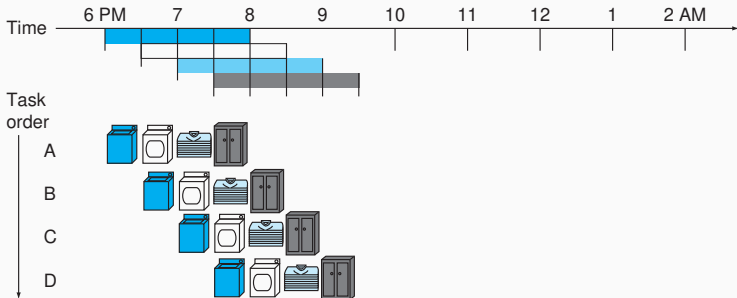
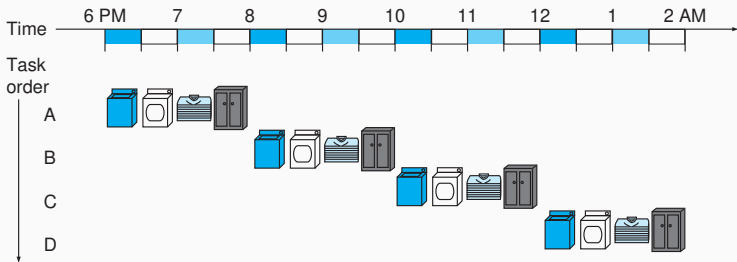
Suponha que você precise lavar 4 trouxas de roupa. Sua tarefa é

1. lavar a roupa na máquina,
2. secar a roupa na secadora,
3. passar a roupa e
4. guardar a roupa.



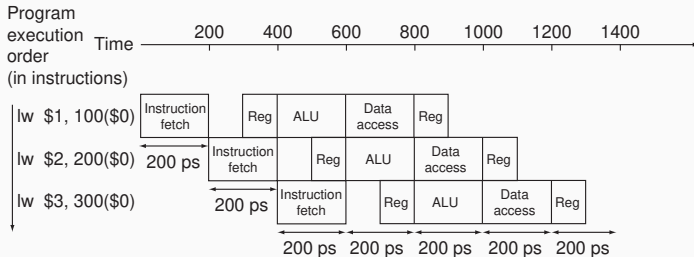
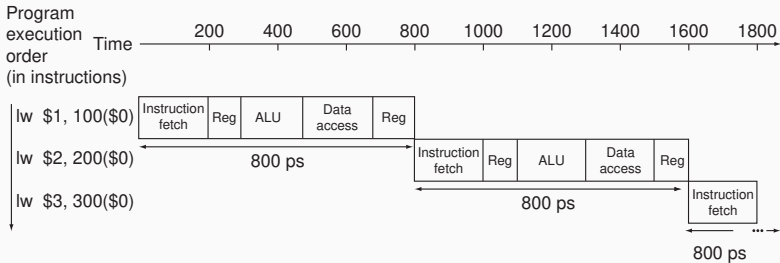
Extraído de [1, Figura 4.25].

# Pipeline - Lavadora de roupas



Extraído de [1, Figura 4.25].

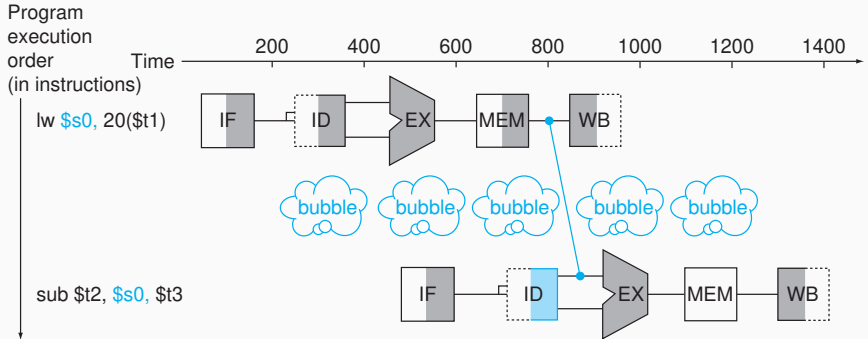
# Pipeline com 3 chamadas de lw



Extraído de [1, Figura 4.27].

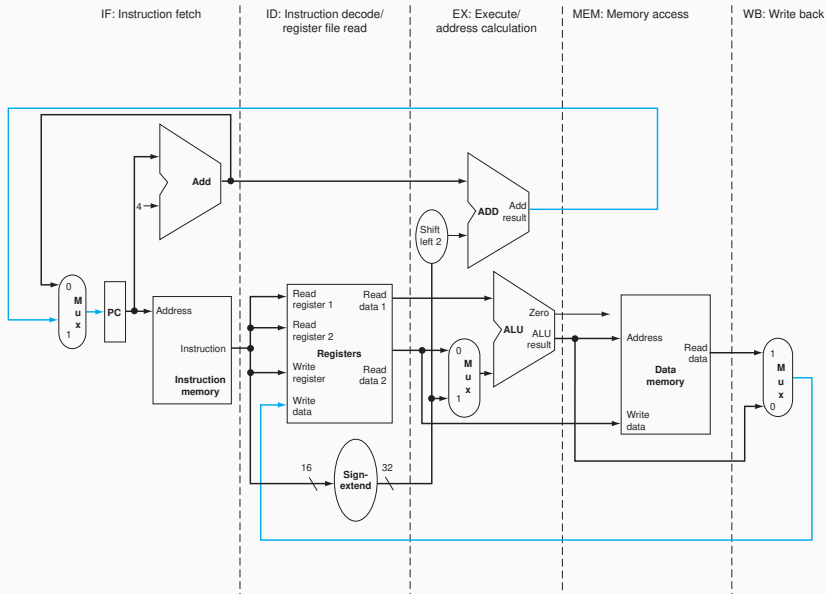


# Pipeline stall

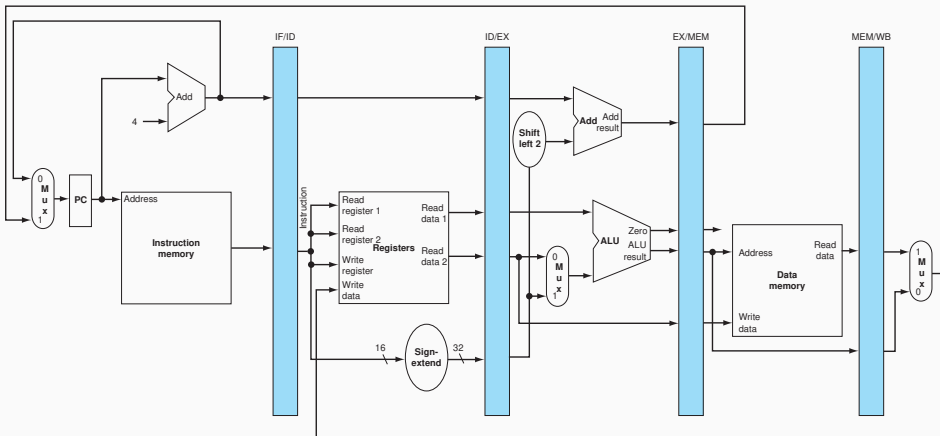


Extraído de [1, Figura 4.30].

# Caminho de dados numa implementação pipeline

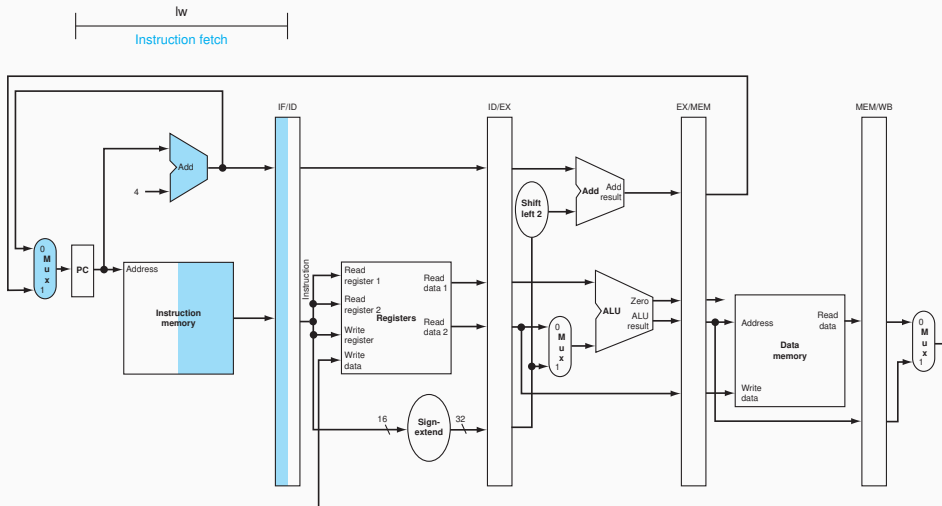


# Caminho de dados numa implementação pipeline



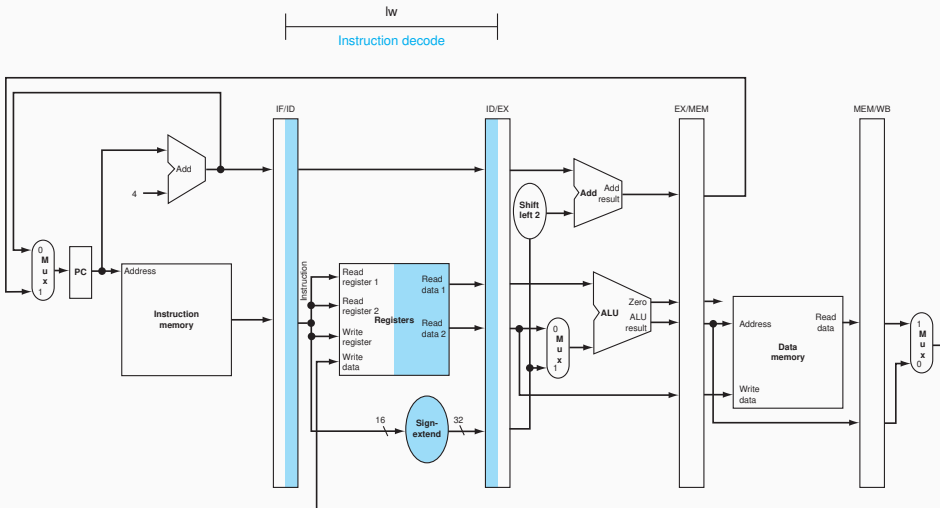
Extraído de [1, Figura 4.35].

# Operação do caminho de dados pipeline - load



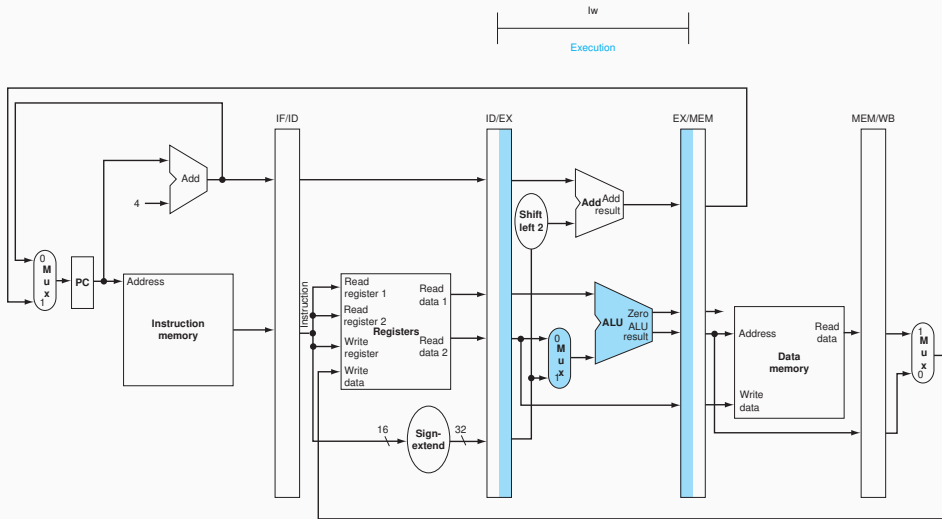
Extraído de [1, Figura 4.36].

# Operação do caminho de dados pipeline - load



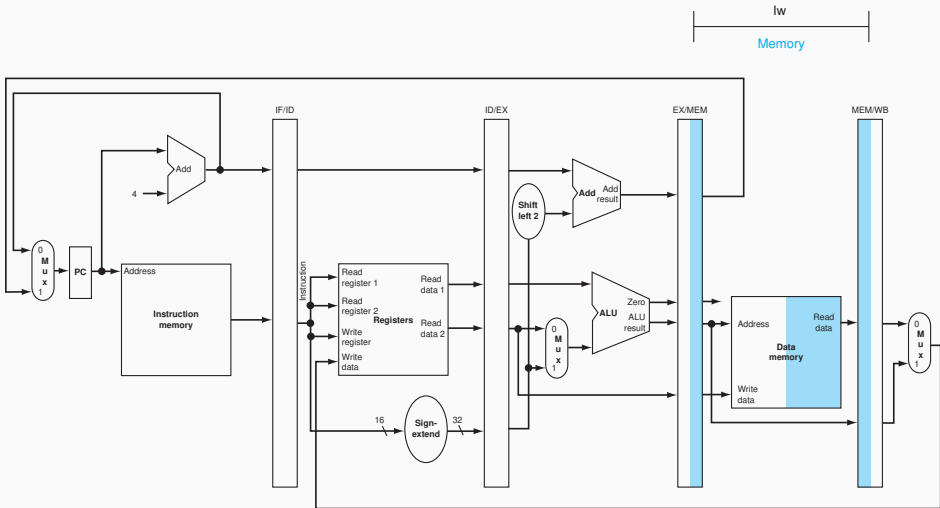
Extraído de [1, Figura 4.36].

# Operação do caminho de dados pipeline - load



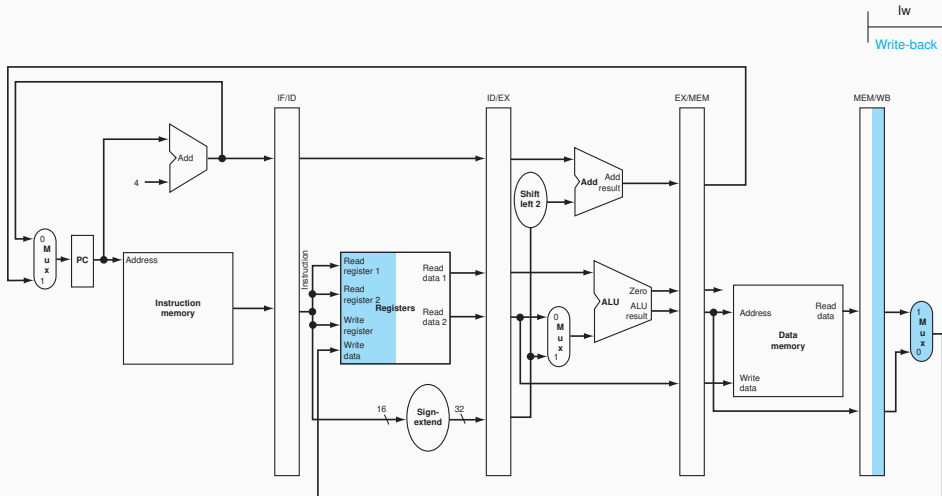
Extraído de [1, Figura 4.37].

# Operação do caminho de dados pipeline - load



Extraído de [1, Figura 4.38].

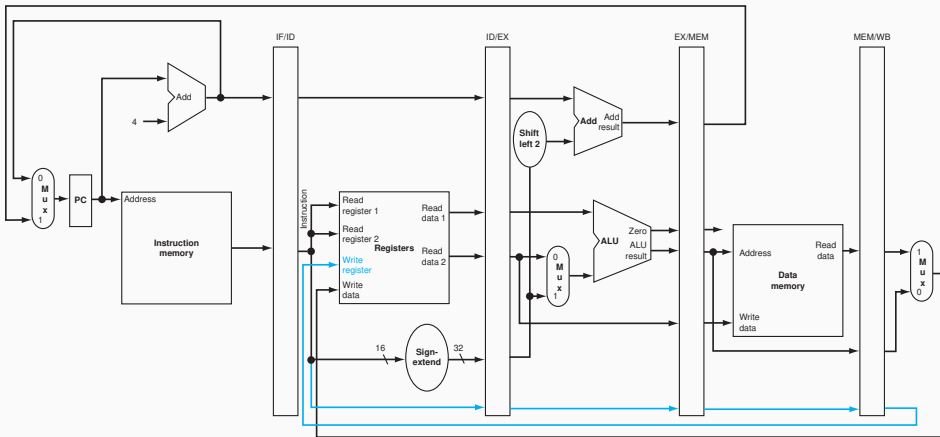
# Operação do caminho de dados pipeline - load



Extraído de [1, Figura 4.38].

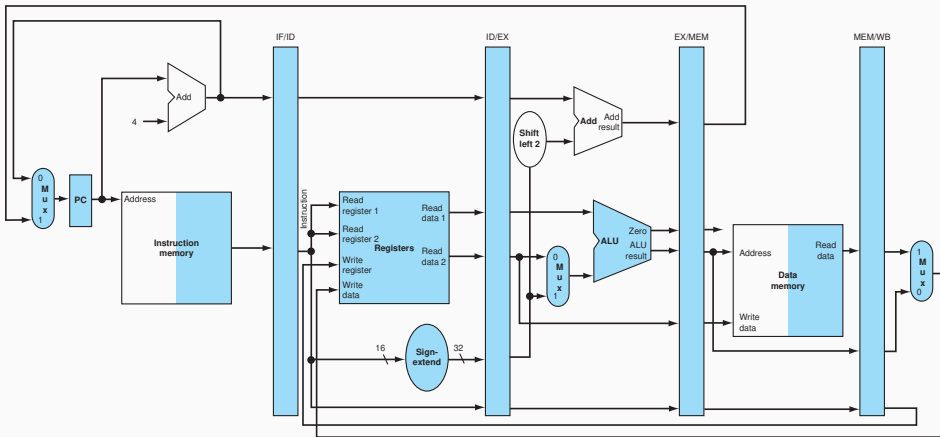


# Caminho de dados numa implementação pipeline



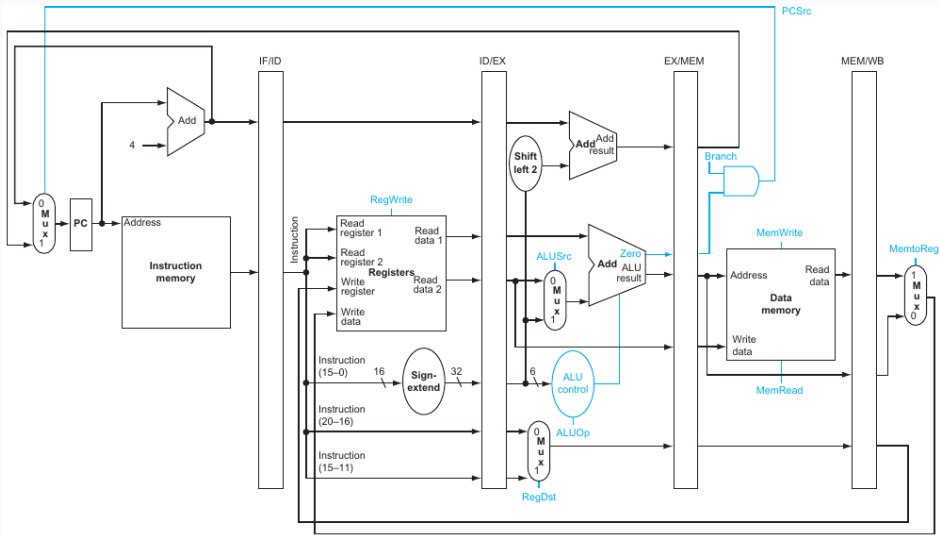
Extraído de [1, Figura 4.41].

# Caminho de dados numa implementação pipeline



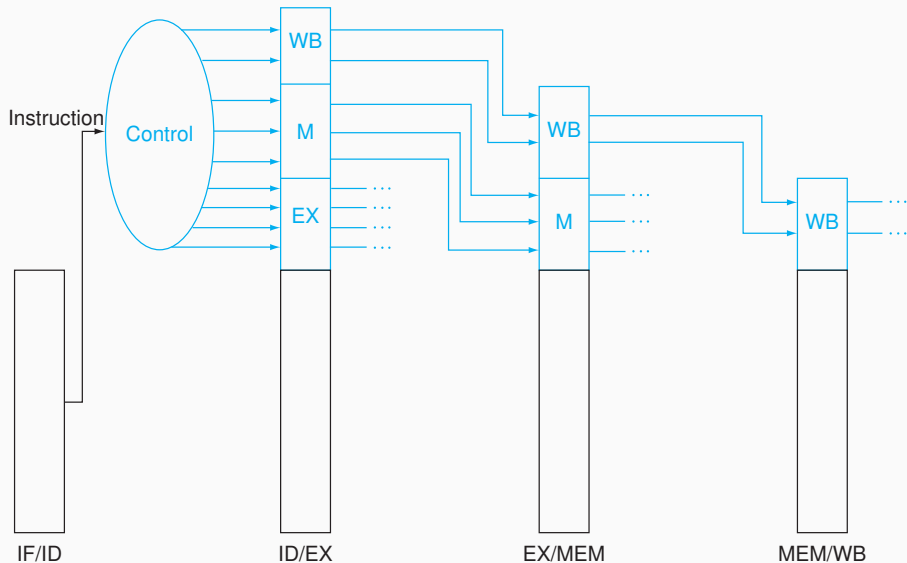
Extraído de [1, Figura 4.42].

# Controle numa implementação pipeline



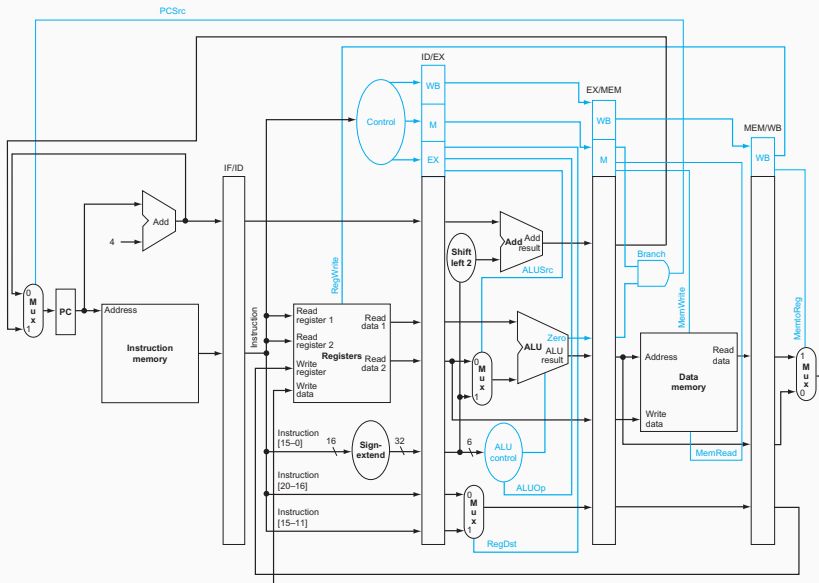
Extraído de [1, Figura 4.46].

# Controle numa implementação pipeline



Extraído de [1, Figura 4.50].

## Caminho de dados completo na implementação pipeline



Extraído de [1, Figura 4.51].



D. A. Patterson e J. L. Hennessy. **Computer and organization design**: the hardware/software interface. 5 ed. Elsevier, 2014.