



Capítulo 4

O Processador

Controle

- Implementação simples acrescentando uma função de controle simples
- Cobre as instruções: lw, sw, beq, add, sub, and, or e slt
- Posteriormente, desenvolveremos o projeto para incluir uma instrução jump (j)

Controle da ULA

- ULA usada para
 - Load/Store: F = adicionar
 - Desvio: F = subtrair
 - Tipo R: F depende do campo de função

ALU control	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set-on-less-than
1100	NOR

Controle da ULA

- Suponha 2 bits ALUOp derivados do opcode
 - A lógica combinacional deriva o controle da ULA (ALU)

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

FIGURE D.2.1 The truth table for the 4 ALU control bits (called **Operation**) as a function of the ALUOp and function code field. This table is the same as that shown in Figure 4.13. Copyright © 2009 Elsevier Inc. All rights reserved

Controle da ULA

ALUOp		Function code fields					
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0
0	1	X	X	X	X	X	X
1	X	X	X	X	X	1	X

a. The truth table for Operation2 = 1 (this table corresponds to the second to left bit of the Operation field in Figure D.2.1)

ALUOp		Function code fields					
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0
0	X	X	X	X	X	X	X
X	X	X	X	X	0	X	X

b. The truth table for Operation1 = 1

ALUOp		Function code fields					
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0
1	X	X	X	X	X	X	1
1	X	X	X	1	X	X	X

c. The truth table for Operation0 = 1

FIGURE D.2.2 The truth tables for three ALU control lines. Only the entries for which the output is 1 are shown.
Copyright © 2009 Elsevier Inc. All rights reserved

Controle da ULA

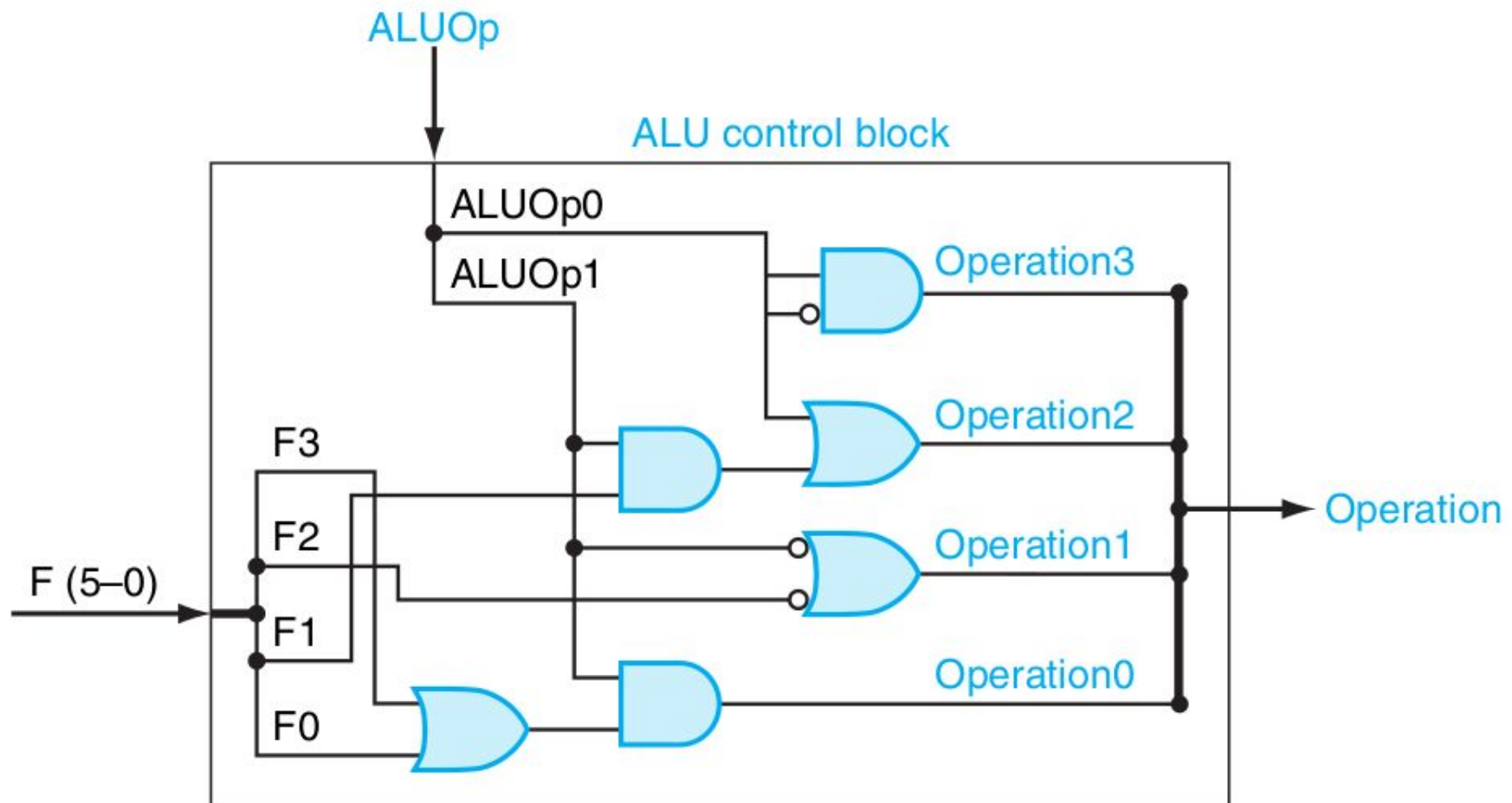


FIGURE D.2.3 The ALU control block generates the four ALU control bits, based on the function code and ALUOp bits. This logic is generated directly from the truth table in Figure D.2.2. Copyright © 2009 Elsevier Inc. All rights reserved

Controle da ULA

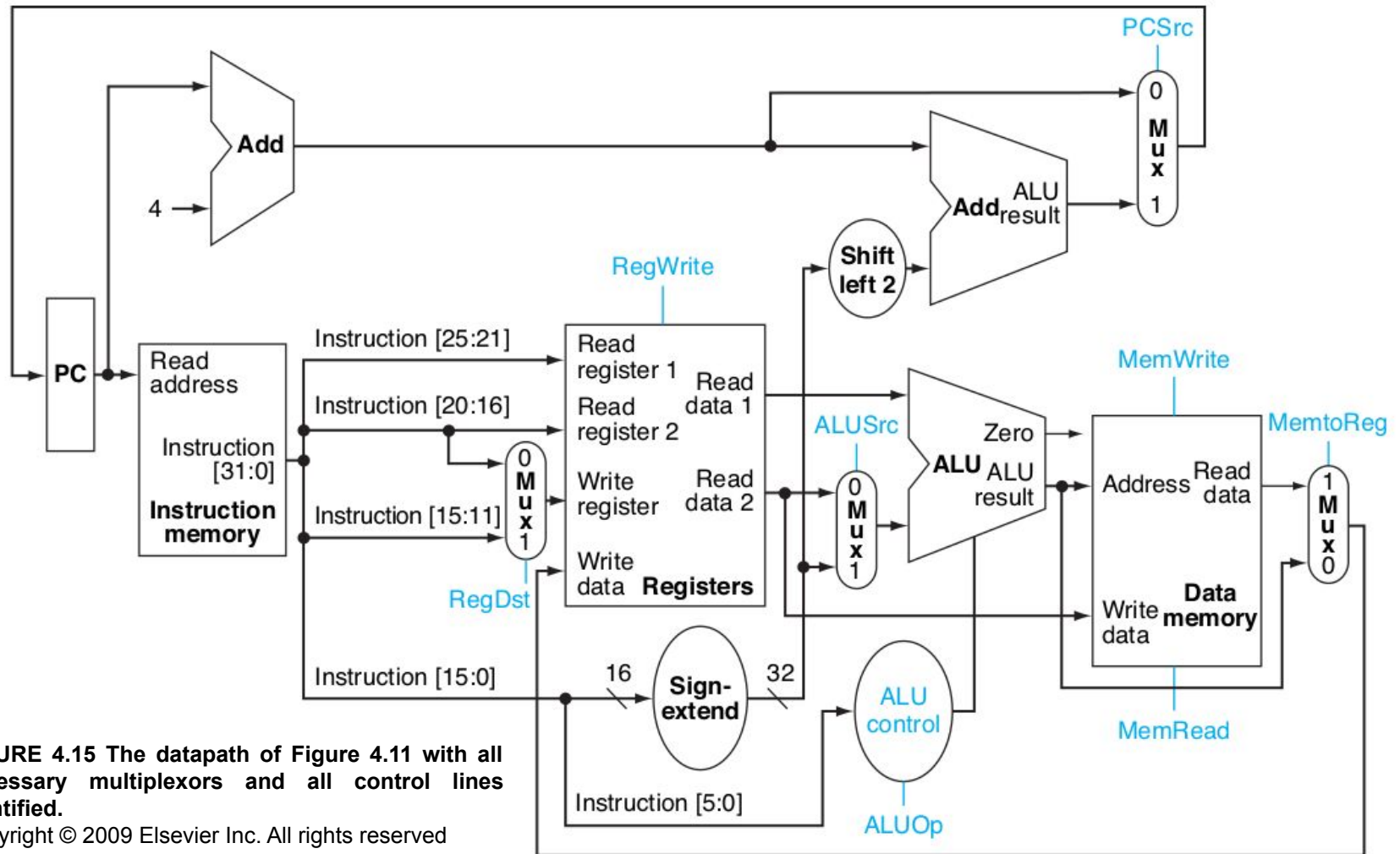


FIGURE 4.15 The datapath of Figure 4.11 with all necessary multiplexors and all control lines identified.

Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

- Para o caminho de dados construído na Figura 4.15 identificar:
 - Os campos de uma instrução
 - As linhas de controle necessárias

Controle da ULA

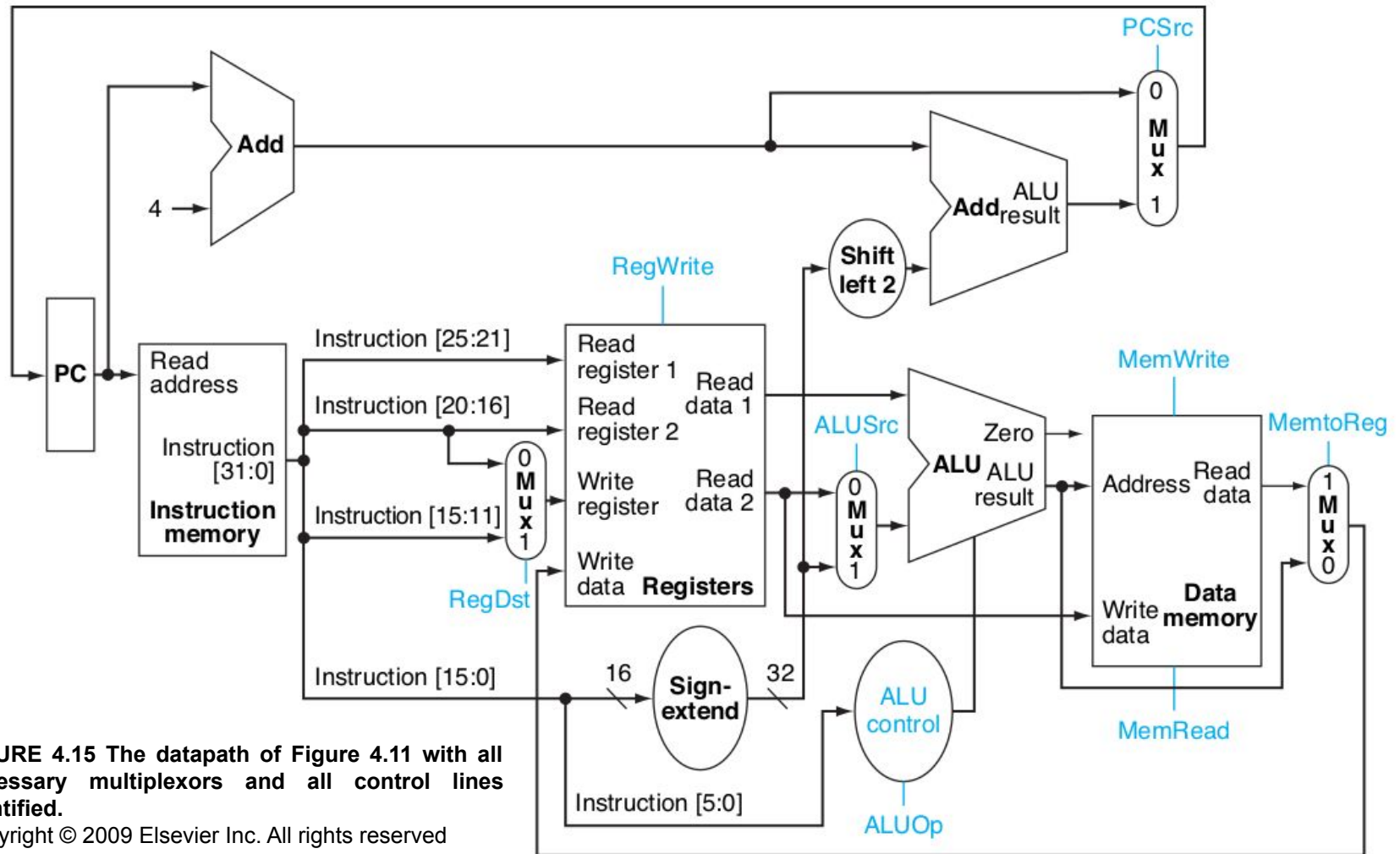


FIGURE 4.15 The datapath of Figure 4.11 with all necessary multiplexors and all control lines identified.

Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

- Os campos de uma instrução
 - Opcode contido nos bits 31:26
 - Iremos nos referir a esse campo como Op[5:0].
 - Os dois registradores a serem lidos sempre são especificados pelos campos rs e rt, nas posições 25:21 e 20:16
 - Isso é verdade para as instruções tipo R, branch equal e store

Controle da ULA

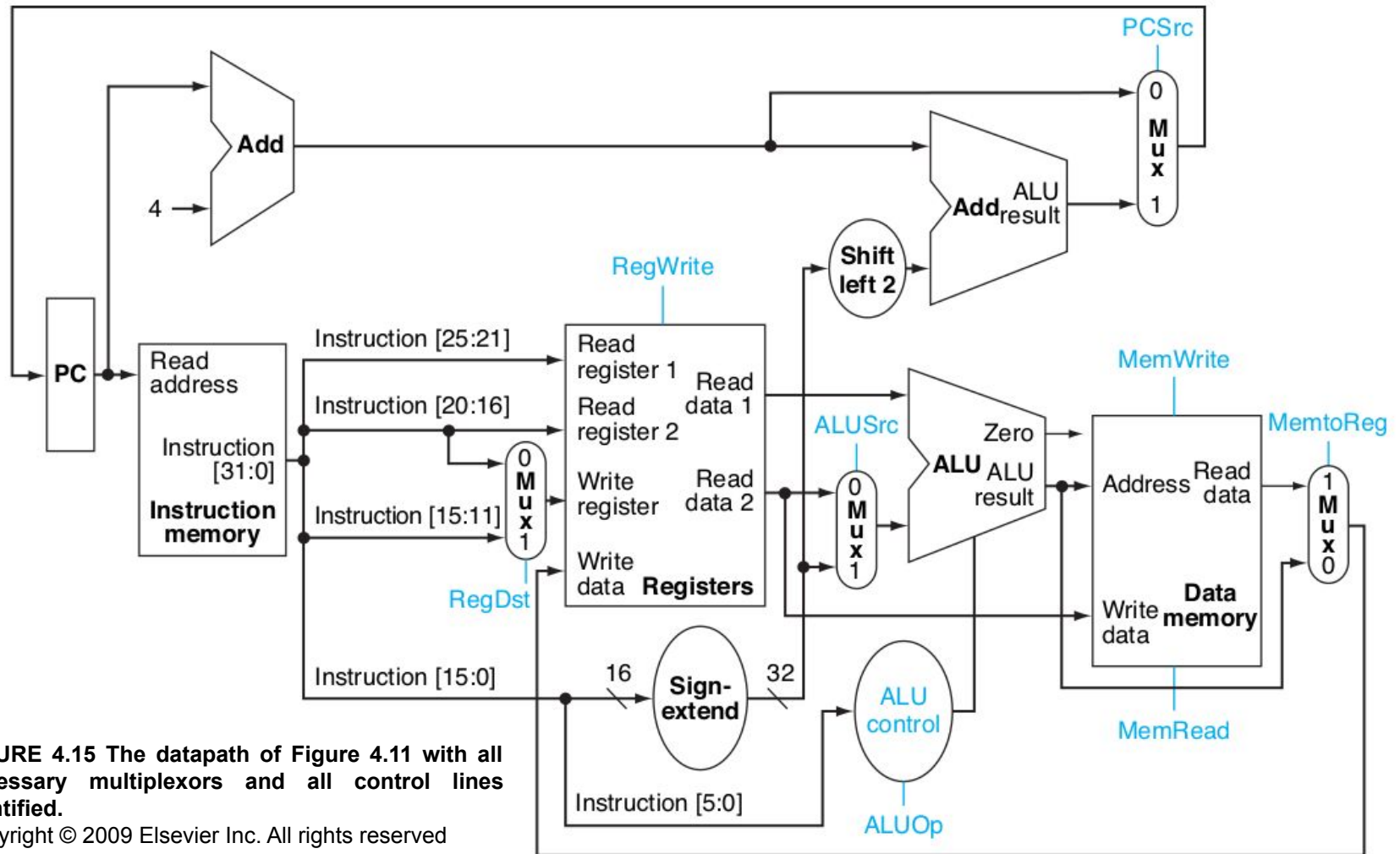


FIGURE 4.15 The datapath of Figure 4.11 with all necessary multiplexors and all control lines identified.

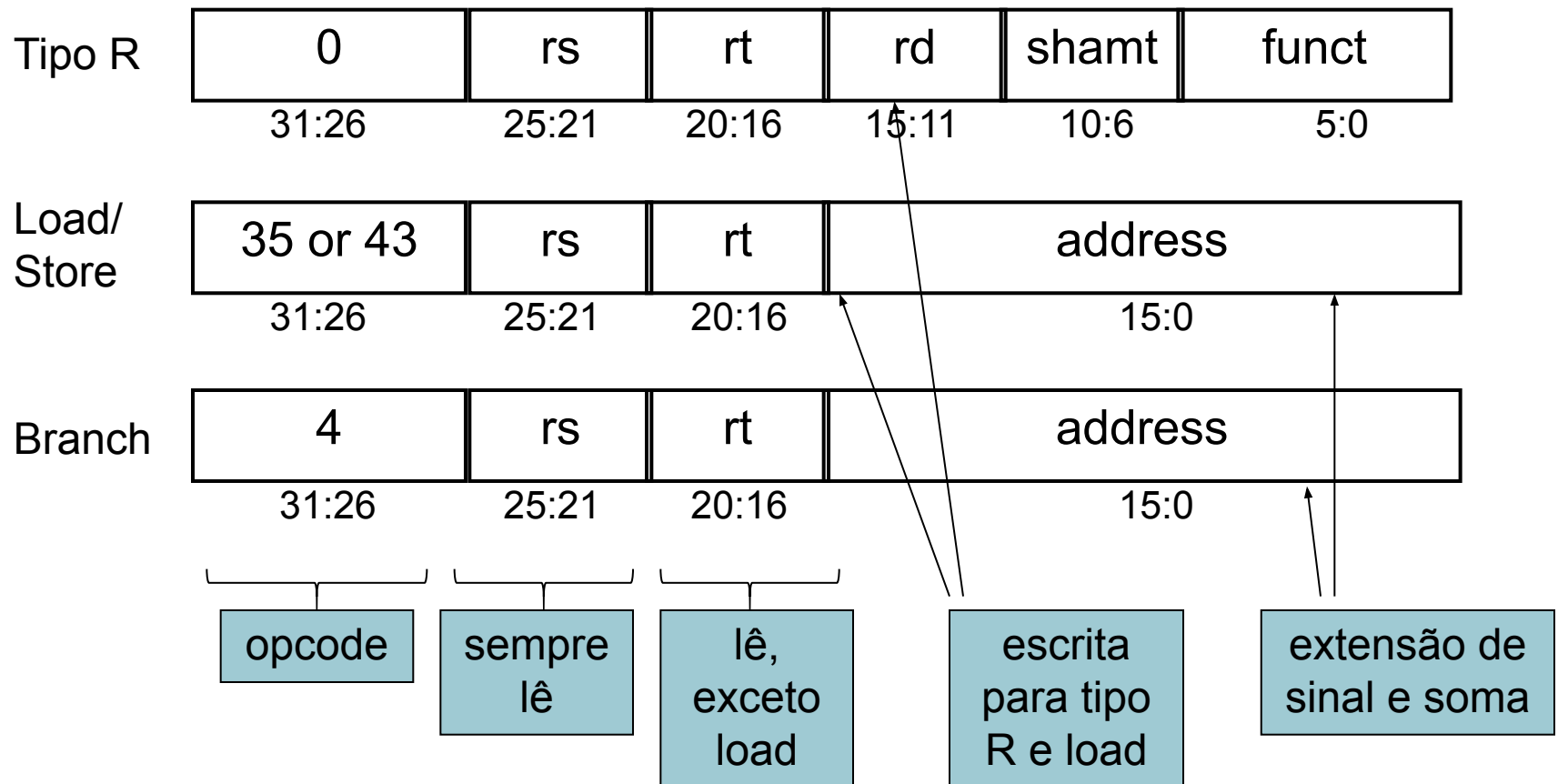
Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

- Os campos de uma instrução
 - Registrador de base para as instruções load e store está sempre nas posições de bit 25:21 (rs)
 - Offset de 16 bits para branch equal, load e store está sempre nas posições 15:0
 - Registrador de destino está em um de dois lugares
 - Para um load, ele está nas posições 20:16 (rt)
 - Para uma instrução tipo R, ele está nas posições 15:11 (rd)

A Unidade de Controle Principal

■ Sinais de controle derivados da instrução



Controle da ULA

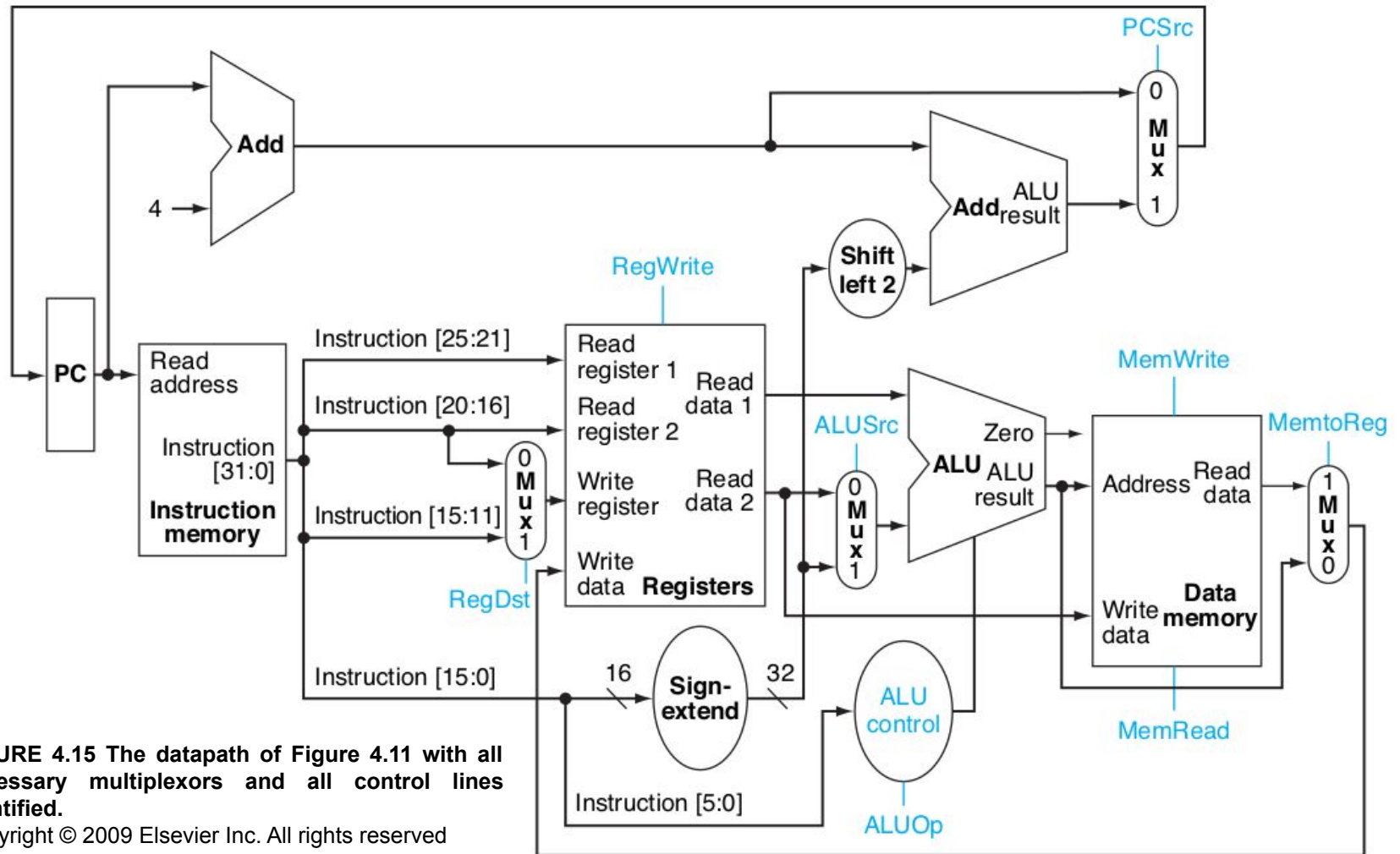


FIGURE 4.15 The datapath of Figure 4.11 with all necessary multiplexors and all control lines identified.

Copyright © 2009 Elsevier Inc. All rights reserved

Datapath Com Controle

Nome do sinal	Efeito quando inativo - 0	Efeito quando ativo - 1
RegDst	O número do registrador destino para 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)
RegWrite	Nenhum.	O registrador na entrada Registrador para escrita é escrito com o valor da entrada Dados para escrita.
ALUSrc	O segundo operando da ALU vem da segunda saída do banco de registradores (Read data 2).	O segundo operando da ALU consiste nos 16 bits mais baixos da instrução com sinal estendido.
PCSrc	O PC é substituído pela saída do somador que calcula o valor de PC+4. Obs.: PCSrc = (sinal controle Branch) AND (Zero ULA)	O PC é substituído pela saída do somador que calcula o destino do desvio.
MemRead	Nenhum.	O conteúdo da memória de dados designado pela entrada Address é colocado na saída Read data.
MemWrite	Nenhum.	O conteúdo da memória de dados designado pela entrada Address é substituído pelo valor na entrada Write data.
MemtoReg	O valor enviado à entrada Write data do banco de registradores vem da ALU.	O valor enviado à entrada Write data do banco de registradores vem da memória de dados.



Datapath Com Controle

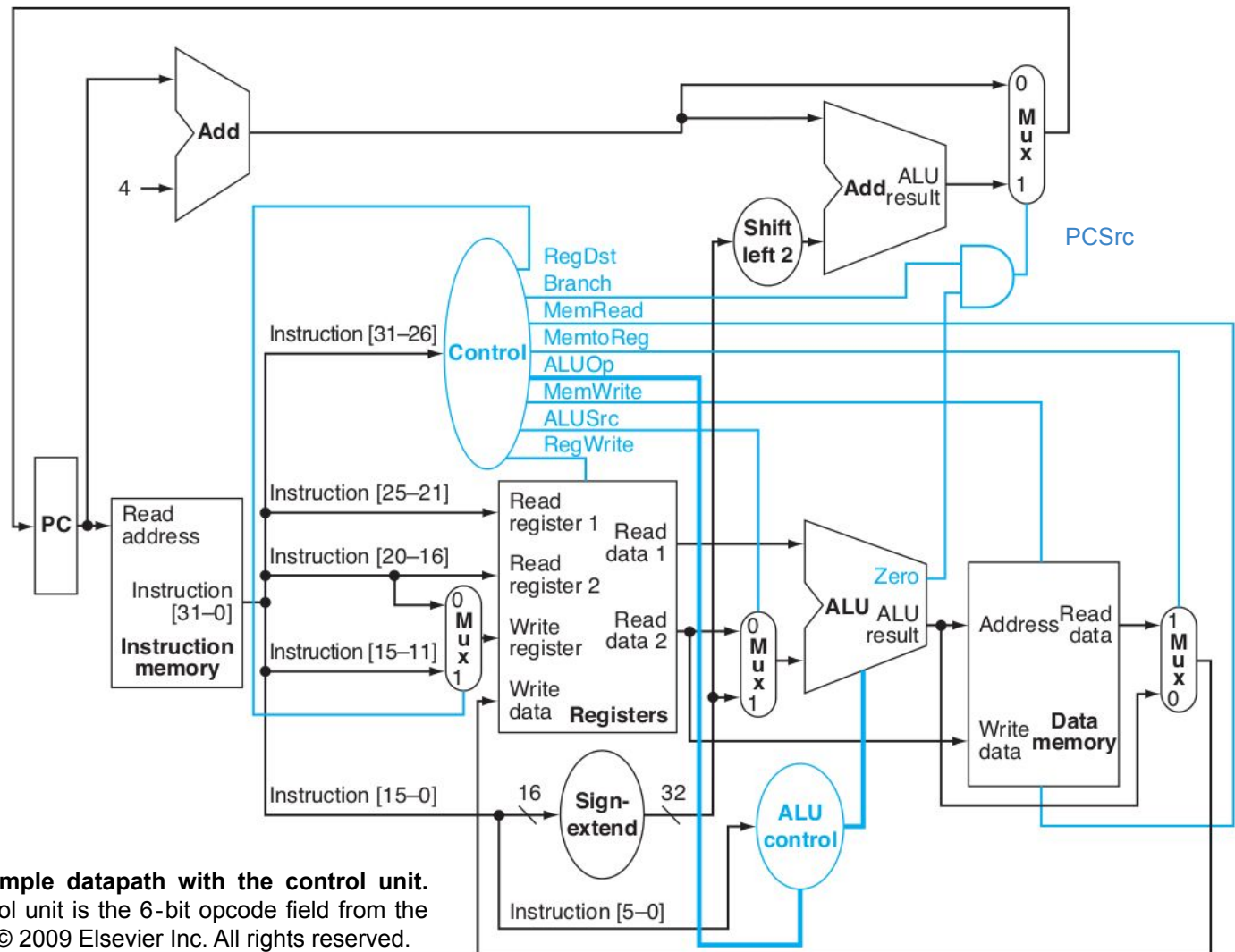


FIGURE 4.17 The simple datapath with the control unit. The input to the control unit is the 6-bit opcode field from the instruction. Copyright © 2009 Elsevier Inc. All rights reserved.

A Unidade de Controle Principal

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	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

FIGURE 4.18 The setting of the control lines is completely determined by the opcode fields of the instruction. The first row of the table corresponds to the R-format instructions (add , sub , AND , OR , and slt). Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

Control	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

FIGURE D.2.4 The control function for the simple one-clock implementation is completely specified by this truth table. This table is the same as that shown in Figure 4.22. Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

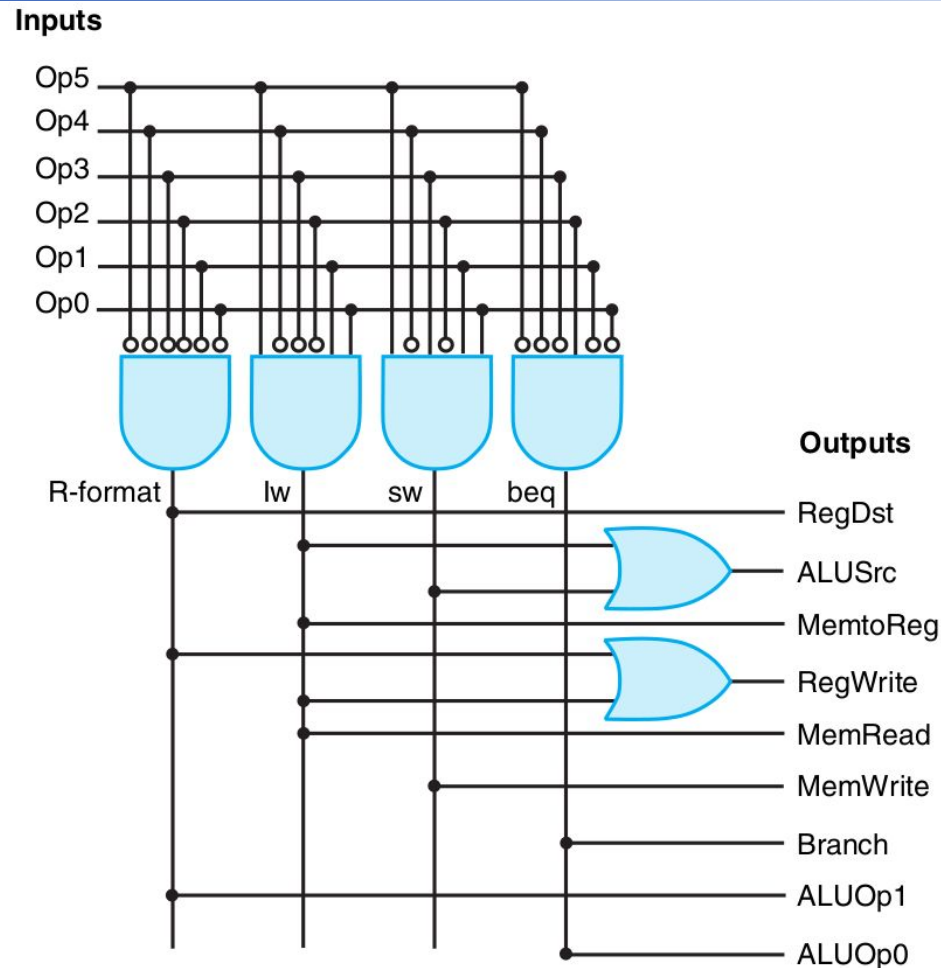


FIGURE D.2.5 The structured implementation of the control function as described by the truth table in Figure D.2.4. The structure, called a programmable logic array (PLA), uses an array of AND gates followed by an array of OR gates. Copyright © 2009 Elsevier Inc. All rights reserved

A Unidade de Controle Principal

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	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

FIGURE 4.18 The setting of the control lines is completely determined by the opcode fields of the instruction. The first row of the table corresponds to the R-format instructions (add , sub , AND , OR , and slt). Copyright © 2009 Elsevier Inc. All rights reserved

Instrução Tipo R

- A Figura 4.19 mostra a operação para uma instrução tipo R, como `add $t1, $t2, $t3`
 - A instrução é buscada e o PC é incrementado
 - \$t2 e \$t3 são lidos do banco de registradores, e a unidade de controle principal calcula a definição das linhas de controle também durante essa etapa
 - A ALU opera nos dados lidos do banco de registradores, usando o código de função (bits 5:0, que é o campo `funct`, da instrução) para gerar a função da ALU
 - O resultado da ALU é escrito no banco de registradores usando os bits 15:11 da instrução para selecionar o registrador de destino (\$t1)

Instrução Tipo R

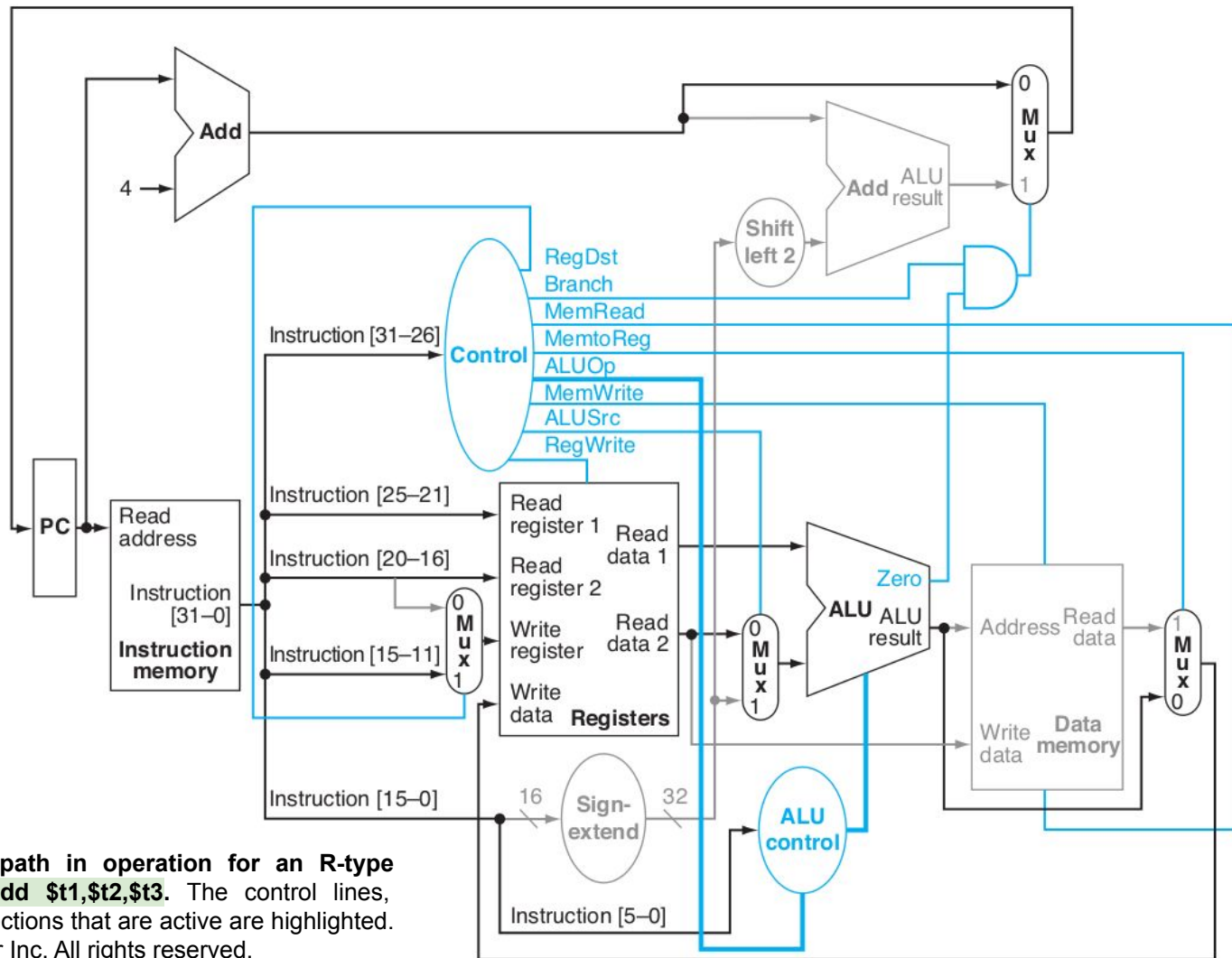


FIGURE 4.19 The datapath in operation for an R-type instruction, such as `add $t1,$t2,$t3`. The control lines, datapath units, and connections that are active are highlighted. Copyright © 2009 Elsevier Inc. All rights reserved.

A Unidade de Controle Principal

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	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

FIGURE 4.18 The setting of the control lines is completely determined by the opcode fields of the instruction. The first row of the table corresponds to the R-format instructions (add , sub , AND , OR , and slt). Copyright © 2009 Elsevier Inc. All rights reserved

Instrução Load

- A Figura 4.20 mostra a operação para uma instrução Load, como `lw $t1, offset($t2)`
 - A instrução é buscada e o PC é incrementado
 - `$t2` é lido do banco de registradores
 - A ALU calcula a soma do valor lido do banco de registradores com os 16 bits menos significativos com sinal estendido da instrução (`offset`)
 - A soma da ALU é usada como o endereço para a memória de dados
 - Os dados da unidade de memória são escritos no banco de registradores; o registrador de destino é fornecido pelos bits 20:16 da instrução (`$t1`)

Instrução Load

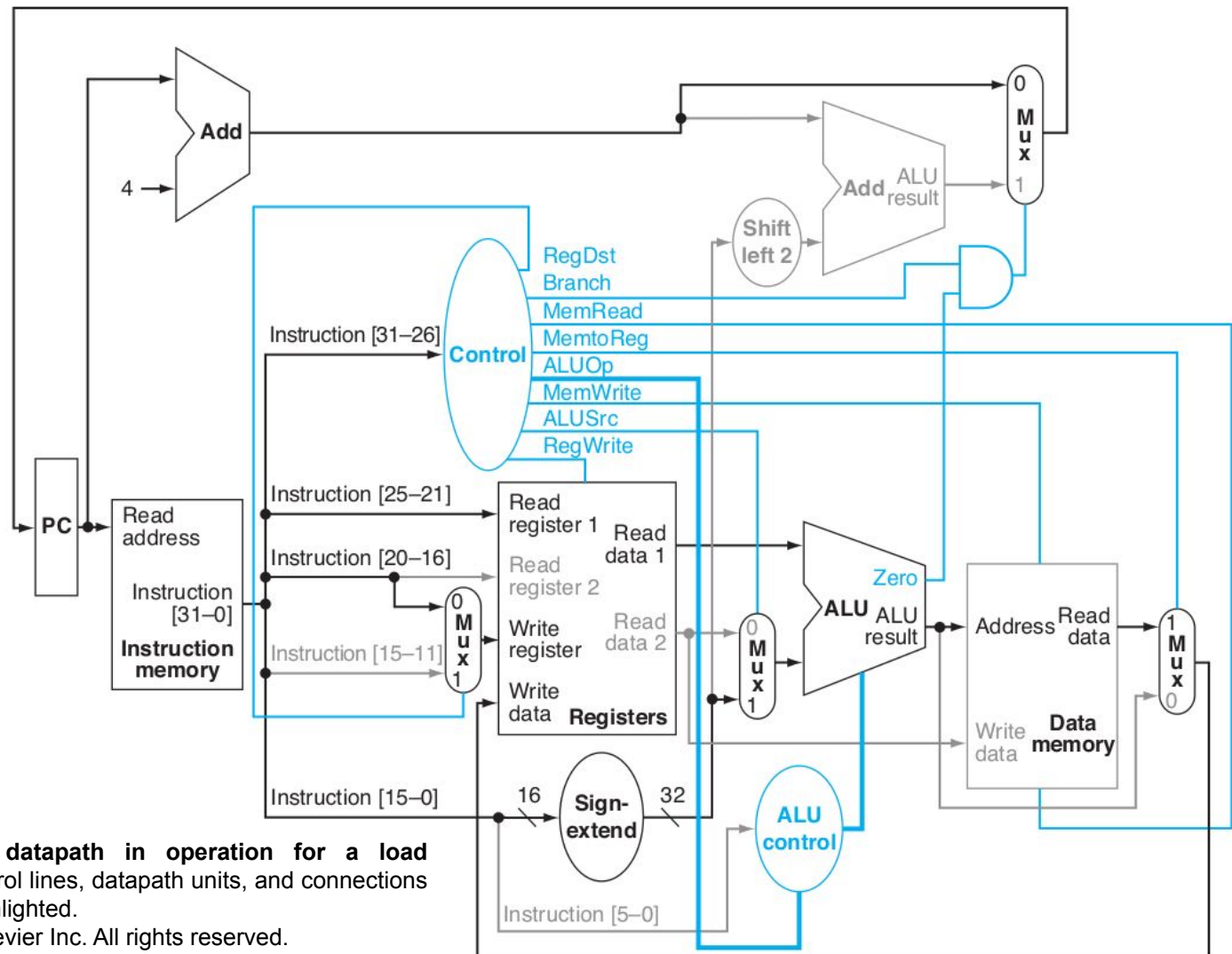


FIGURE 4.20 The datapath in operation for a load instruction. The control lines, datapath units, and connections that are active are highlighted.

Copyright © 2009 Elsevier Inc. All rights reserved.

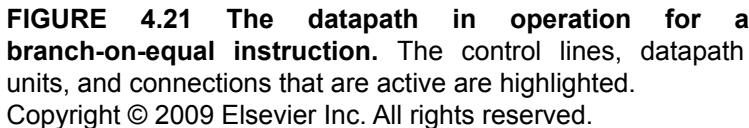
A Unidade de Controle Principal

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	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

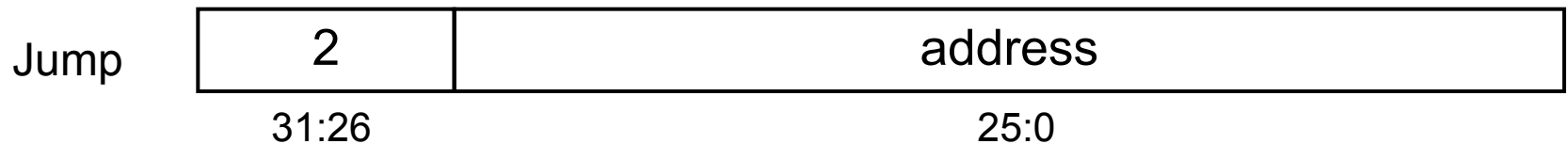
FIGURE 4.18 The setting of the control lines is completely determined by the opcode fields of the instruction. The first row of the table corresponds to the R-format instructions (add , sub , AND , OR , and slt). Copyright © 2009 Elsevier Inc. All rights reserved

Instrução Branch-on-Equal

- A Figura 4.21 mostra a operação para uma instrução Branch, como `beq $t1,$t2,offset`
 - A instrução é buscada e o PC é incrementado
 - `$t1` e `$t2`, são lidos do banco de registradores
 - A ALU realiza uma subtração dos valores de dados lidos do banco de registradores. O valor de `PC + 4` é somado aos 16 bits menos significativos com sinal estendido (offset) deslocados de dois para a esquerda; o resultado é o endereço de destino do desvio
 - O resultado Zero da ALU é usado para decidir qual resultado do somador deve ser armazenado no PC



Implementando Jumps



- Jump usa endereço
- Atualiza o PC com a concatenação de:
 - 4 bits superiores do PC (bits 31:28)
 - 26 bits jump address (bits 25:0)
 - 00_{bin} nos bits menos significativos
- Precisa de um sinal de controle extra decodificado do opcode

Datapath Com Jumps

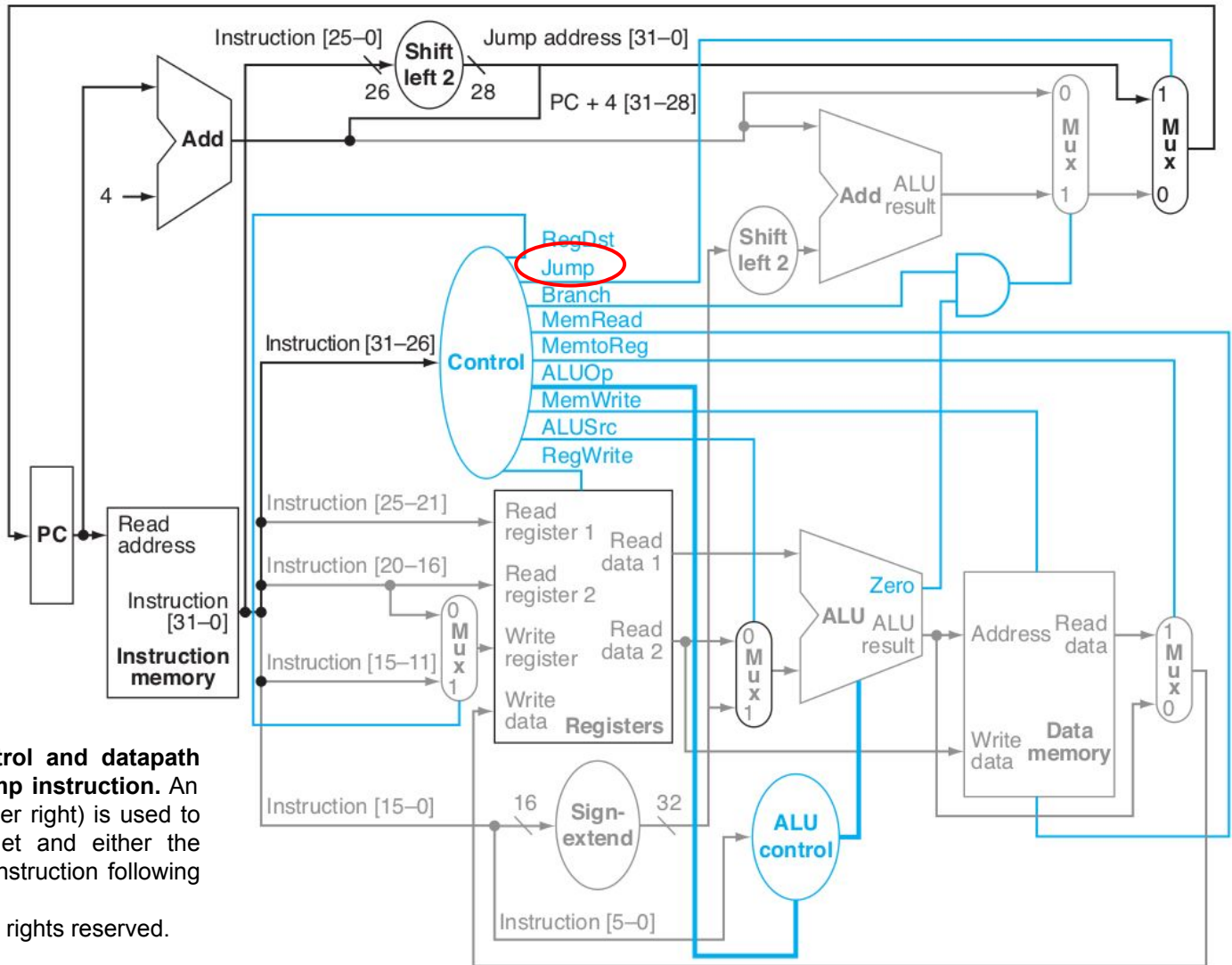


FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one.

Copyright © 2009 Elsevier Inc. All rights reserved.

Referências

- Seções 4.4, Apêndice D - “Organização e Projeto de Computadores - A Interface Hardware/Software, David A. Patterson & John L. Hennessy, Campus, 4 edição, 2013.