

Arquitetura de um processador - Aula 3

Nicolas Chagas Souza

22/08/2022

Controle no caminho de dados

A controladora é a unidade que decide qual caminho uma instrução deve percorrer, a decisão é tomada baseada no campo op da instrução, os 6 primeiros bits da instrução. Essa unidade possui uma tabela verdade, implementada digitalmente, que faz a decisão do caminho a partir da tabela citada.

Controle da ULA

A ULA é utilizada nos seguintes casos, com as respectivas operações:

- Instruções de load e store: operação add;
- Instruções de desvio: operação sub;
- Instruções do tipo R: operação depende do campo funct da instrução.

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

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

Unidade de controle principal

Os sinais de controle derivam da instrução.

Field	0	rs	rt	rd	shamt	funct
Bit positions	31:26	25:21	20:16	15:11	10:6	5:0

a. R-type instruction

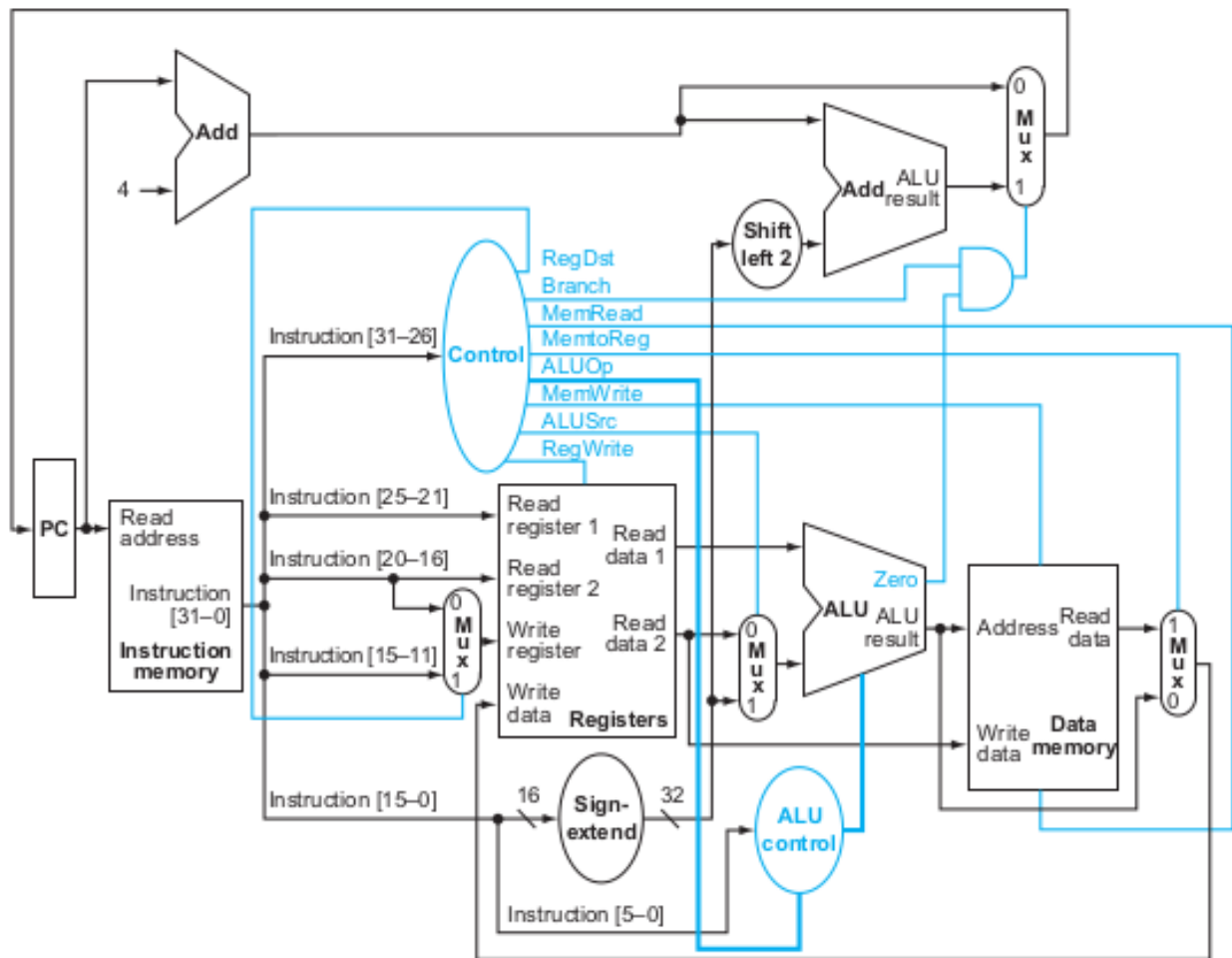
Field	35 or 43	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

b. Load or store instruction

Field	4	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

c. Branch instruction

O primeiro campo é o opcode, o segundo é de leitura, o terceiro é de leitura (exceto para o load, onde é de escrita), o registrador rd e rt é de escrita para as instruções do tipo R e do tipo load.



A unidade controladora recebe o campo op da instrução [31-26] e a partir dele faz a configuração do caminho de dados com os sinais de controle. A unidade controladora da ULA recebe os campos [5-0] da instrução, que correspondem ao campo funct.

Os outros sinais da unidade de controle são:

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20:16).	The register destination number for the Write register comes from the rd field (bits 15:11).
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, lower 16 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

Os campos MemRead e MemWrite determinam se no final do ciclo haverá escrita, leitura ou nenhuma operação na memória. É necessário utilizar dois sinais, já que existe a possibilidade de nenhuma operação na memória. Não é possível ter MemRead = MemWrite = 1, já que não é possível escrever e ler da memória ao mesmo tempo.

MemtoReg determina qual é o dado que será escrito na memória.

ALUSrc determina de onde vem o operador para a ULA, se vem do campo [15-0] constante ou do Read Data 2. Operações entre 2 registradores.

Valor dos sinais de controle para as operações.

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

Instruções do tipo R:

Instrução load:

Instrução beq:

Problemas de desempenho

A instrução mais demorada determina o período de clock, que é a instrução load.

- Caminho crítico (instrução mais demorada): load;
- Caminho da instrução: memória de instrução → banco de registradores → ULA → memória de dados → banco de registradores;
- Outras instruções são executadas mais rapidamente, e para melhorar esse caminho utiliza-se pipeline.

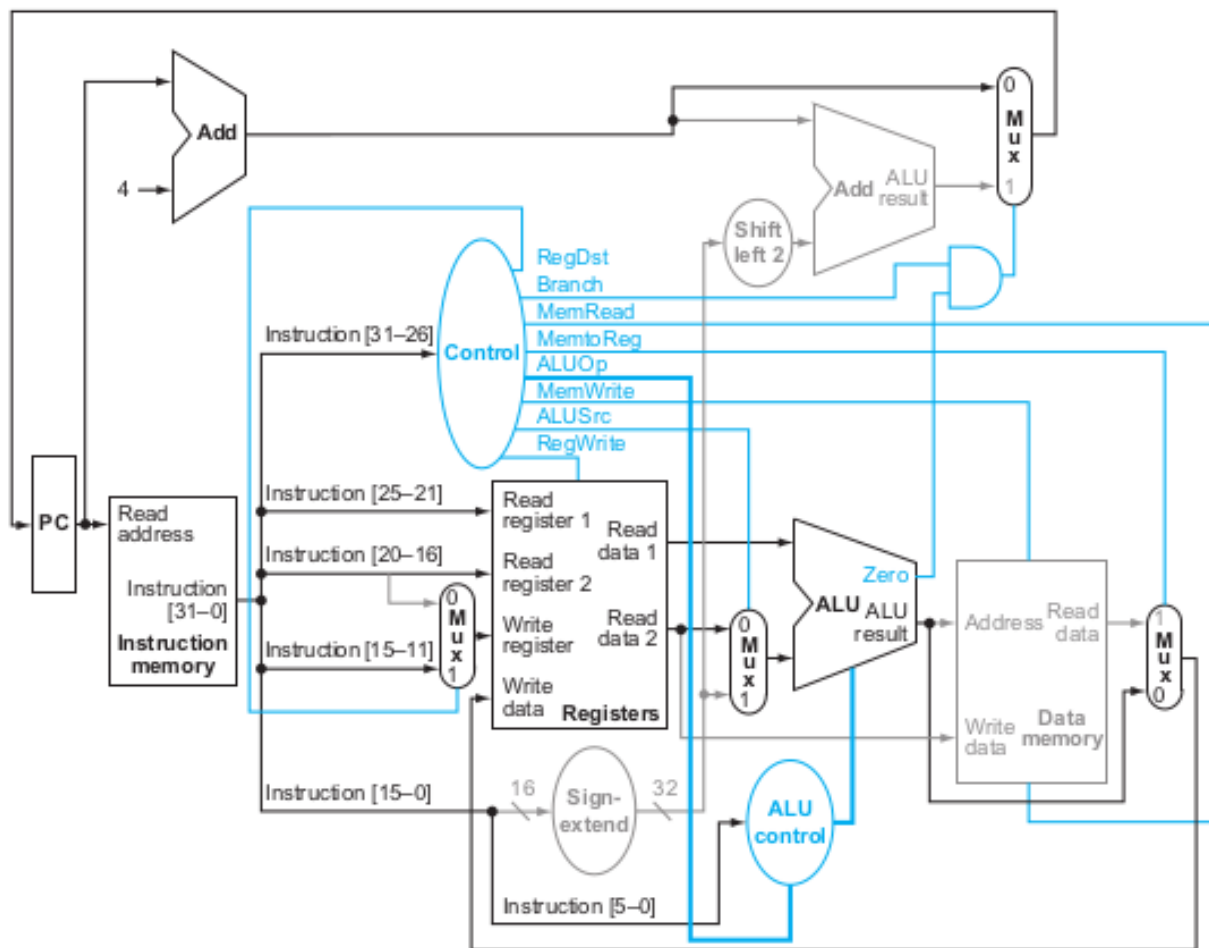


Figure 1: Instruções do tipo R

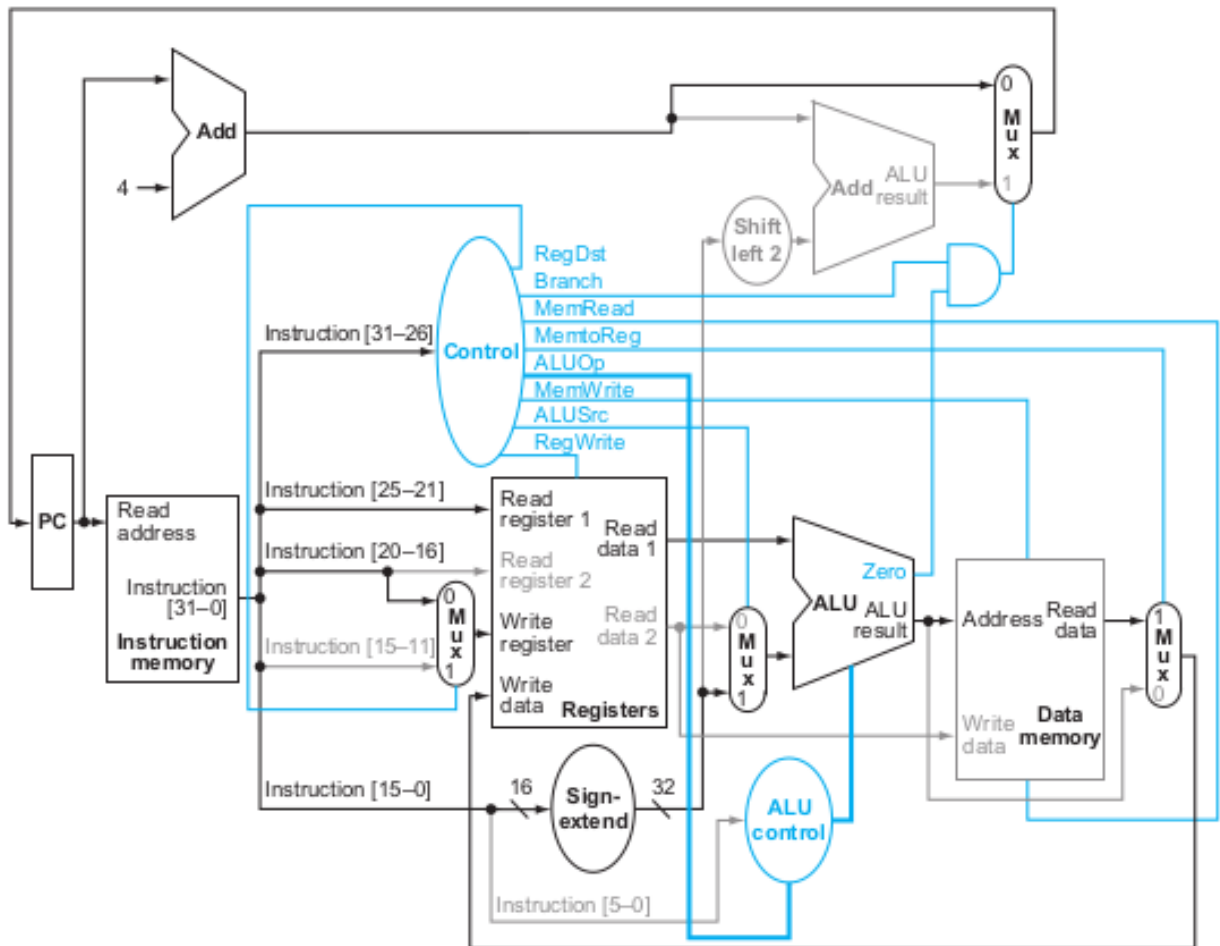


Figure 2: Instrução load

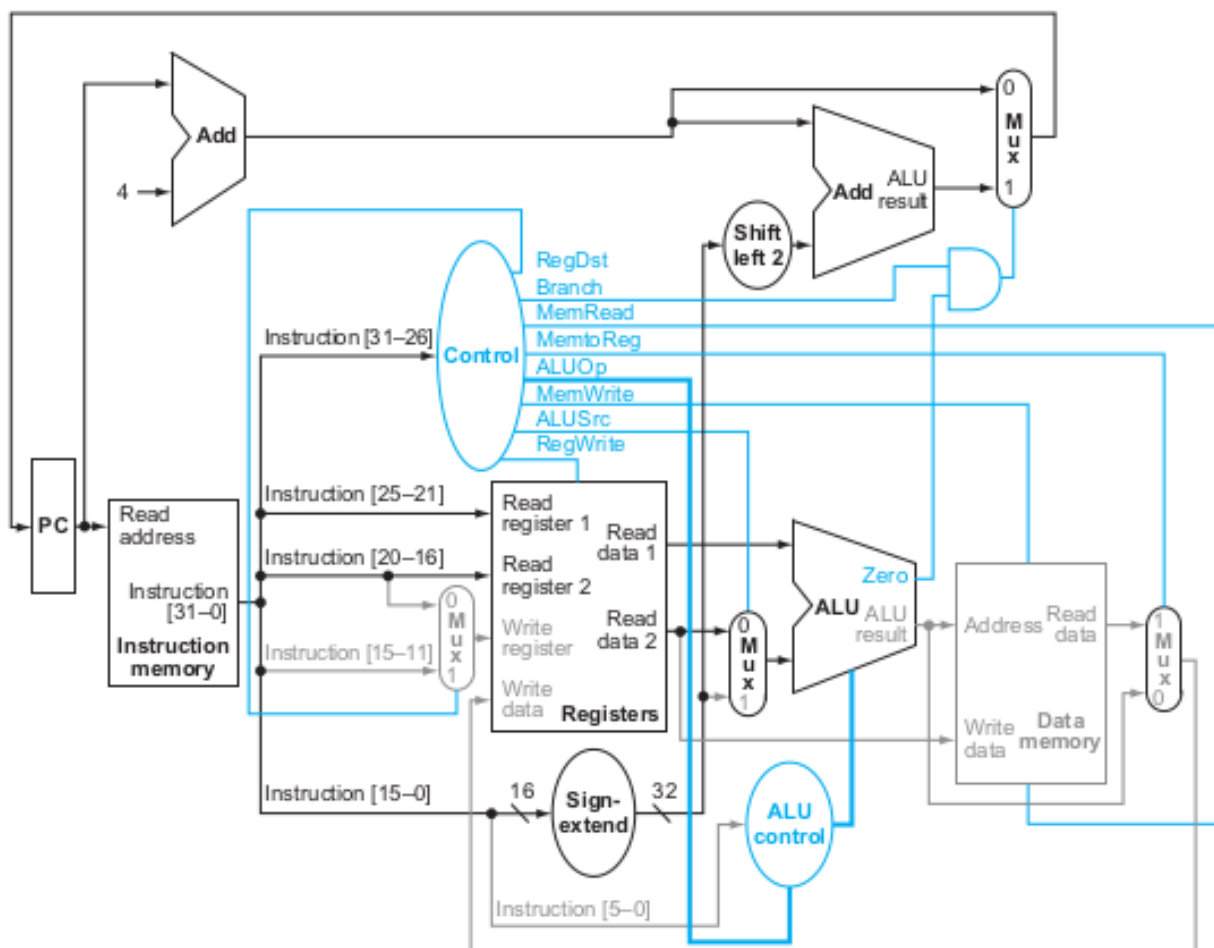


Figure 3: Instrução beq