



LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I

AULA 8: definição do caminho de dados (datapath)

Professor: Mateus Felipe Tymburibá Ferreira

Data: 22/07/2021

Aluno: Darmes Araújo Dias

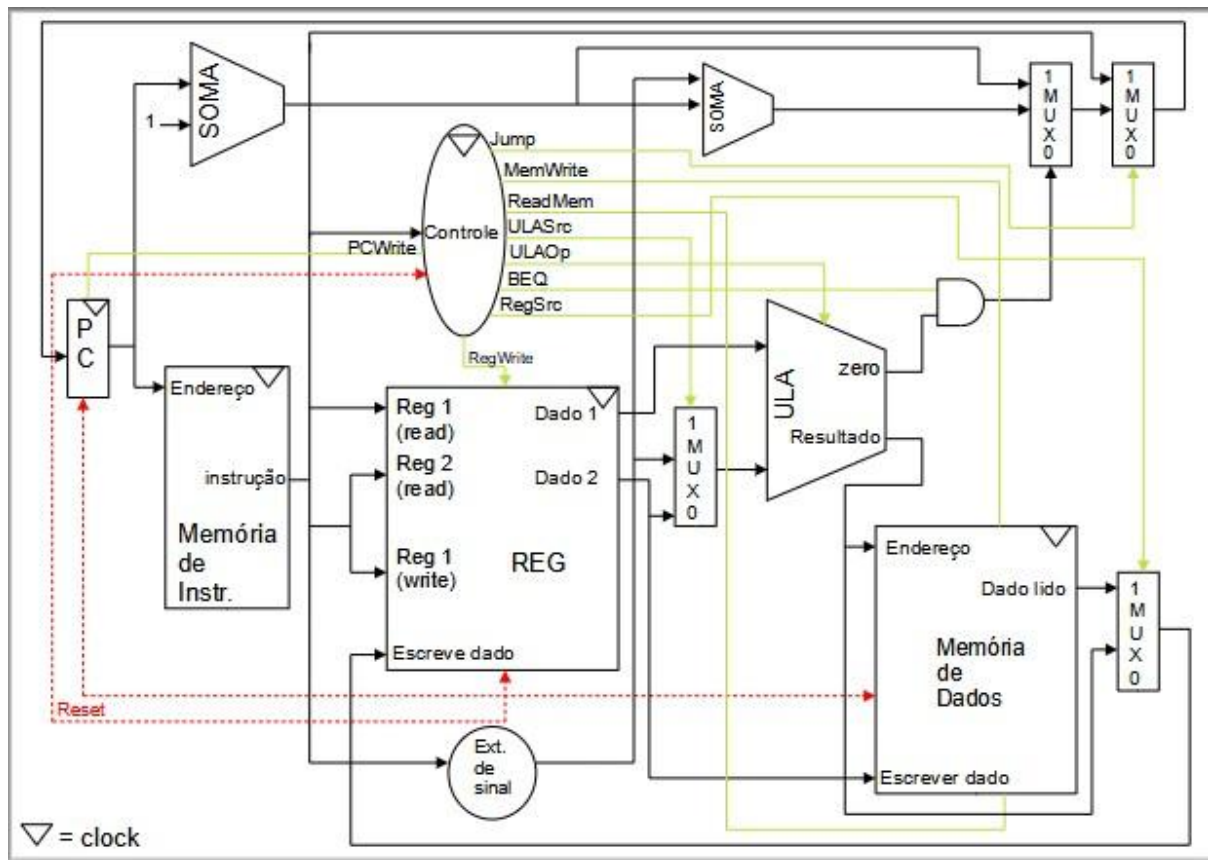
Mudanças no projeto:

Para a arquitetura ficar mais organizada troquei os opcodes das instruções beq, addi e la.

Instrução	opcode	rd	rs	immediate	description
load	000	X	X	XXX	rd = MEM[rs*immediate]
la	001	X	-	LABEL(XXXX)	rd = LABEL
store	010	X	X	XXX	MEM[rs*immediate]=rd
add	011	X	X	rs2(X)	rd = rs+rs2
addi	100	X	-	LABEL(XXXX)	rd = rd + immediate
beq	101	-	-	LABEL(XXXXX)	rd==rs
j	110	-	-	LABEL(XXXXX)	j LABEL
halt	111	-	-	-	PC = PC

(Obs.: Se for preciso eu refaço os passos do relatório anterior, para converter o código em binário.)

1) Apresente o diagrama do caminho de dados projetado para o seu nRisc.



2) Para cada instrução suportada pelo seu processador, liste o caminho que os dados percorrerão até a conclusão da instrução. Nesta etapa do projeto, não é necessário determinar o valor dos sinais de controle para cada instrução. Isso será realizado em uma prática futura.

Instrução	Caminho
Load	PC → Memória de instrução → Extensão de sinal → MUX → ULA → Memória de dado → MUX → Banco de regs.
Store	PC → Memória de instrução → Extensão de sinal → MUX → ULA → Memória de dado
Load Address	PC → Memória de instrução → Extensão de sinal → MUX → ULA → Memória de dado
Add	PC → Memória de instrução → Banco de regs. → MUX → ULA → MUX → Banco de regs.
Addi	PC → Memória de instrução → Banco de regs. → Extensão de sinal → MUX → ULA → MUX → Banco de regs.
Beq	PC → Memória de instrução → Banco de regs. → MUX → ULA → MUX → MUX → PC
J	PC → Memória de instrução → MUX → MUX → PC
Halt	OUT