

Unidade de Controle

Aula 17 - Exercício

Bruno Albertini, Edson Gomi e Ricardo Lera

O objetivo deste exercício é implementar a Unidade de Controle (UC) do processador RISC-V. A UC possui duas entradas (a instrução e ALUzero) e seis saídas de 1 bit, anotadas em verde conforme apresentado na Figura 1.

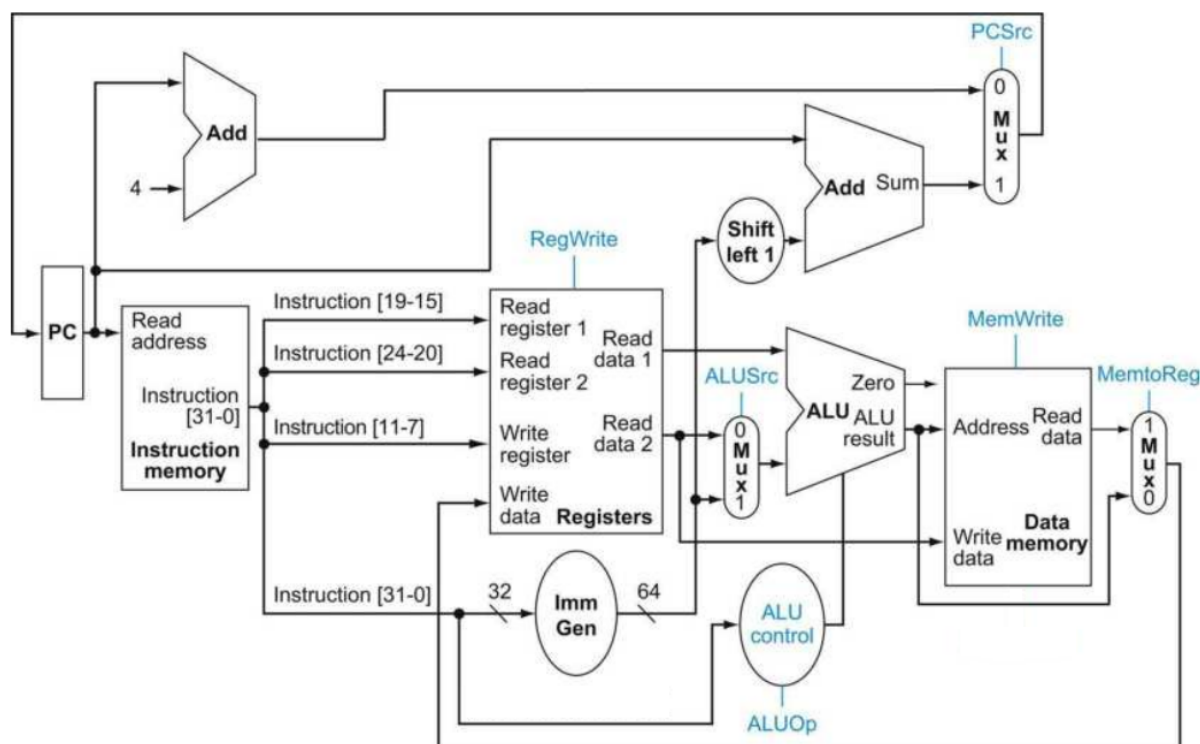


Figura 1: Patterson & Hennessy, pg. 506, alterado

No edisciplinas foi disponibilizado o *template* `control_unit.v`. É necessário completar os três blocos principais do código, conforme descrito a seguir.

Considere também as questões apresentadas. Elas não serão cobradas nesta atividade, mas ajudarão a terminá-la e a compreender o funcionamento do sistema.

- Bloco ALU Control: Define `ALUctl` a partir de `ALUOp` e `funct7[5]` (bit 30 da instrução).
 Questão auxiliar: Por que `ALUctl` depende de `funct7[5]`?
 Obs: `ALUctl` é interpretado pela ALU como *subtração* quando 1 e *adição* quando 0.
- Bloco PC Control: Define `PCsrc` a partir de `branch`, `ALUzero` e `funct3`.
 Questão auxiliar: Qual bit de `funct3` será relevante?

- Bloco OP Control: Define `ALUOp`, `MemtoReg`, `ALUsrc`, `RegWrite`, `MemWrite` e `branch` a partir de `opcode`.

Envie para o juiz somente o módulo `control_unit`. Não será possível testar sua arquitetura em um testbench antes de enviar, portanto, aumentamos a quantidade de submissões.

Detalhes Importantes:

- Na tabela feita em aula o valor X representa “não importa”. Não tente associar o valor literal X a um registrador, utilize 0 ou 1.
- Este sistema deve ter suporte para **ambos** BEQ e BNE.
- O sinal de controle MemRead foi excluído do exemplo no livro por ser redundante com MemWrite.
- A partir desta atividade, as memórias de instrução e de dados são endereçadas por **byte** e não por palavra.

Recursos:

- RISC-V Instruction Set Manual: <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf>
- RISC-V Encoder/Decoder: <https://luplab.gitlab.io/rvcodecjs/>