

Relatório do Trabalho Prático 2

Arquitetura de Computadores

Gabriel Lopes Ferreira
Luiz Junio Veloso Dos Santos
Matheus Luiz Oliveira Spindula
Rebeca Neto

30 de outubro de 2019

1. **Introdução:** O objetivo

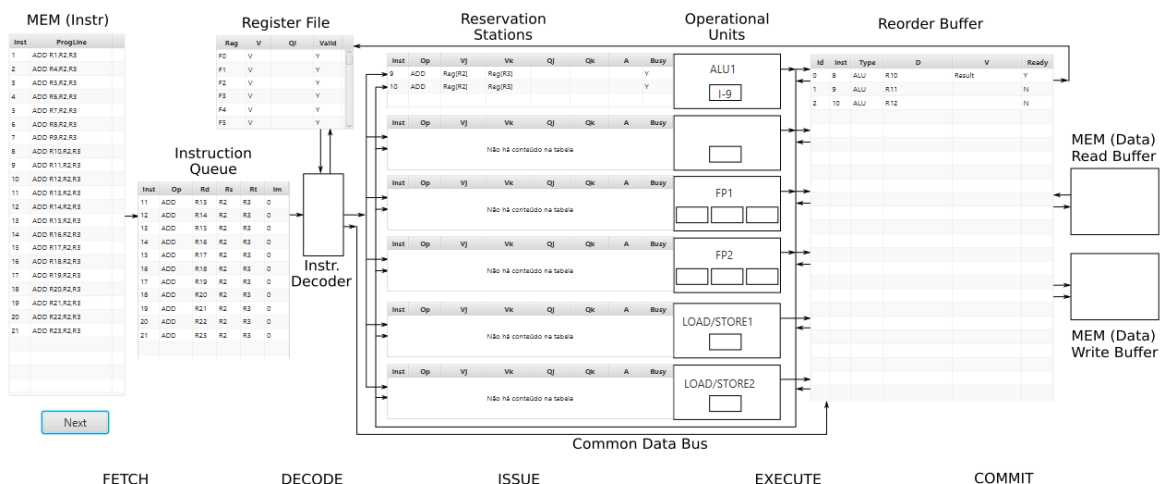
2. **O simulador:**

3. **Análises:**

(a) **Teste 1**

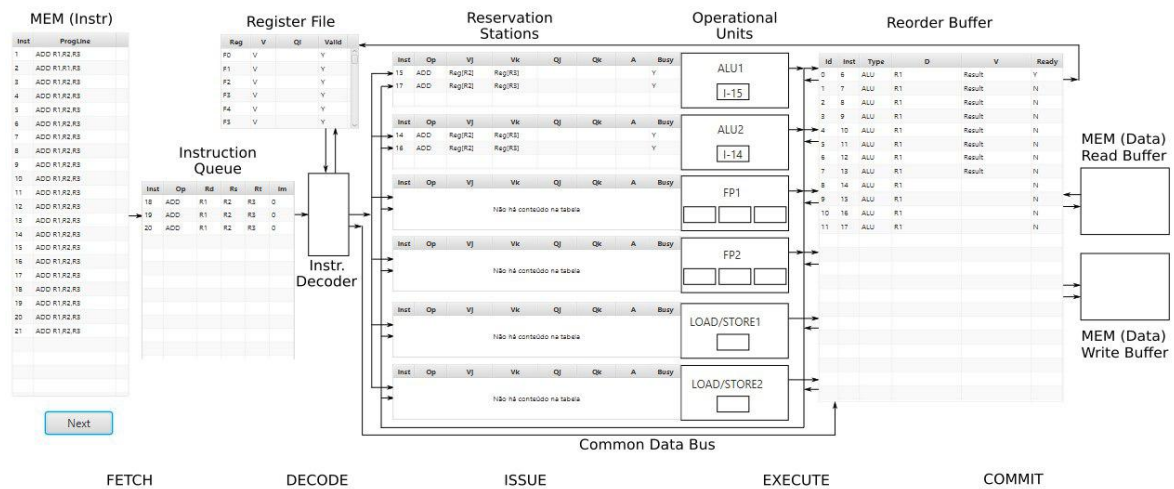
Neste teste, usamos uma arquitetura superescalar com *2-way Pipeline*, *2-line buffer*, um trace com 21 instruções, com todas as instruções idênticas ('ADD R1,R2,R3'), ou seja, todas leem e escrevem no mesmo local, comparamos o impacto de utilizar 1 ALU vs 2 ALU.

Figura 1: Execução com 1 ALU



No teste acima, apesar de muitas instruções serem despachadas para o Instruction Queue, poucas vão para o *Reservation Station* (Janela Distribuída), já que existe somente 1 ALU e sua janela comporta somente 2 instruções e somente uma é executada em cada ciclo. Após sair da ALU o resultado vai para o *Reorder Buffer*, para que possa ser escrito no banco de registradores. Como neste trace todas as instruções escrevem no mesmo local, existe um “acúmulo” de instruções no *buffer* com resultado pronto, porém aguardando uma instrução anterior terminar a escrita.

Figura 2: Execução com 2 ALU



Na segunda parte do teste, foi adicionada uma segunda ALU, isso fez com que um maior número de instruções fossem despachadas para as *Reservation Stations*, gerando um menor acúmulo de instruções no *Instruction Queue*, o fato de ter 2 ALU's faz com que ocorra um paralelismo de instruções, duas em cada janela. Contudo esse paralelismo fez com que houvesse um maior acúmulo de instruções no *Reorder Buffer*.

(b) Teste 2

Neste teste, usamos uma arquitetura com *2-way Pipeline*, *2-line Buffer*

4. Análise do simulador

5. Conclusões finais