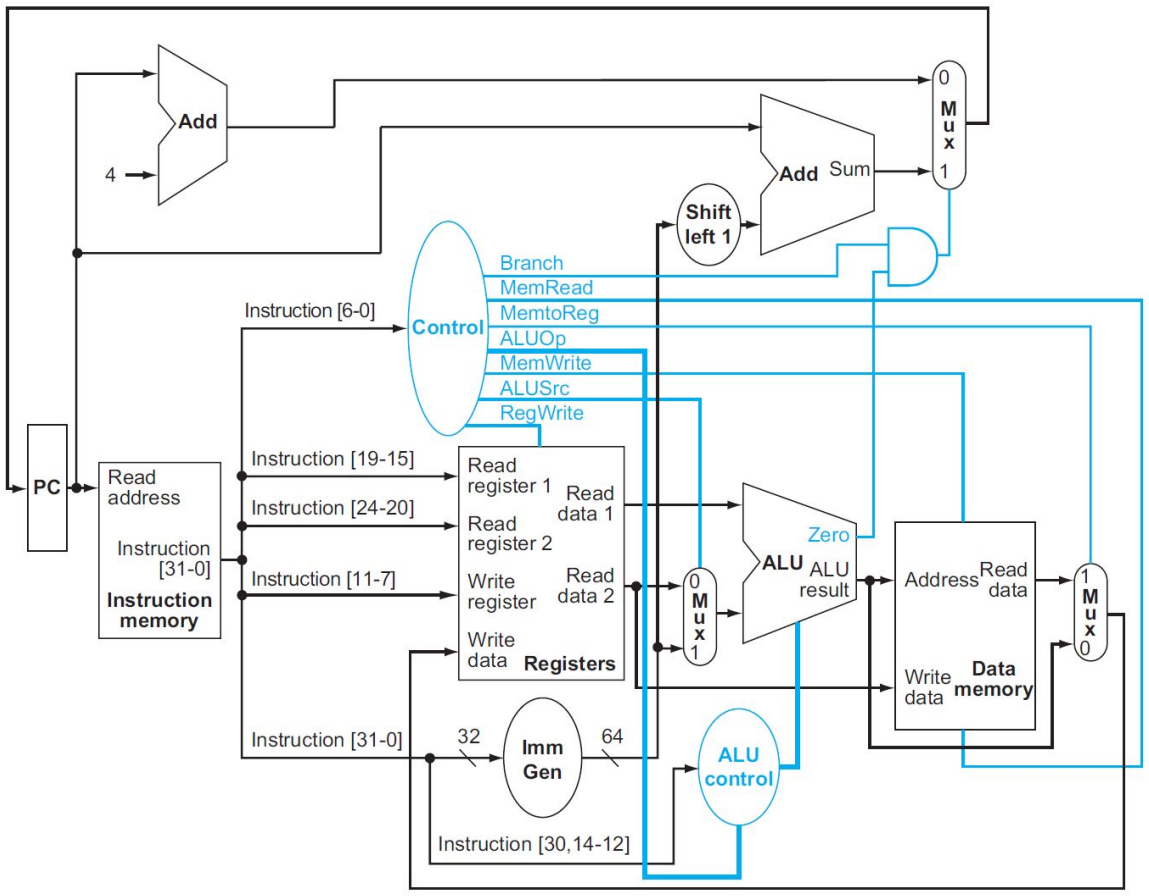
Universidade Federal de Viçosa

Trabalho Prático 03 – Caminho de Dados do RISC-V
CCF 252 – Organização de Computadores I
Ciência da Computação – Campus UFV-Florestal
Prof. José Augusto Miranda Nacif

O trabalho consiste na implementação de uma versão simplificada do caminho de dados do RISC-V. O caminho de dados é apresentado abaixo. Trata-se da Figura 4.17 da edição RISC-V do livro.



Forma de entrega: Github para o código, Classroom para a documentação em PDF e para o vídeo em formato MP4.

O que deve ser entregue:

- Documentação simplificada do trabalho prático em formato SBC;
- Vídeo de apresentação de no máximo 3 minutos;
- Arquivos fonte SystemVerilog/Verilog e arquivos de simulação através do GitHub, como no trabalho anterior.

Será atribuída pontuação extra para os trabalhos que implementarem caminhos de dados que suportem outras instruções.

Critérios de avaliação:

- A documentação deverá conter, pelo menos, introdução, desenvolvimento e considerações finais. Além disso, o texto da mesma deve ser justificado e com referências, todo no formato SBC, caso existam;
- O trabalho deverá ser apresentado em um vídeo de no máximo 3 minutos;
- O caminho de dados do aluno irá passar por casos de testes previamente selecionados pelo professor, de modo que a nota irá sofrer alteração dependendo do resultado obtido;
- Cópias de trabalhos práticos serão exemplarmente punidas. A punição será a mesma para quem copiou e para quem forneceu o trabalho prático.

Na simulação, pelo menos as primeiras 32 posições da memória devem ser exibidas na tela.

Instruções a serem implementadas:

```
ADD, SUB, AND, OR, LD, SD, BEQ
```

Exemplo: Use o simulador RISC-V Interpreter:
https://www.cs.cornell.edu/courses/cs3410/2019sp/riscv/interpreter/#
(O simulador apenas suporta instruções para 1 palavra. Para uso no TP, troque para double, ou seja, lw vira ld, e sw vira sd).

Código:

```
# Colocando valor base na memória
# addi x2, x0, 7
# sw x2, 4(x0)

lw x1, 4(x0)
add x2, x1, x0

add x1, x1, x2
add x1, x1, x2
sub x1, x1, x2
sub x1, x1, x2

beq x1, x2, SAIDA

# Caso o fluxo venha para cá, seu processador está errado
add x1, x1, x1
sw x1, 0(x0)

SAIDA:
```

```
and x1, x1, x2
or x1, x1, x0
sw x1, 0(x0)
```

Memória:
Endereço 32 (4 bytes) = 7

Resultado previsto:
A posição 0 da memória deve ser preenchida com 7.

Observação: podem ser cobrados outros testes parecidos com este na análise do trabalho. Utilize o seu montador (TP 2) para conversão para binário antes de executar no seu RISC-V.