UNIVERSIDADE FEDERAL DE VIÇOSA

CAMPUS FLORESTAL

ARTHUR DE BELLIS - 3503
PABLO FERREIRA - 3480
SAMUEL SENA - 3494

CAMINHO DE DADOS MIPS

PARTE I

FLORESTAL
2019

INTRODUÇÃO:

O trabalho a seguir tem como objetivo uma representação do caminho de dados de um processador de 32 bits da arquitetura MIPS.

A implementação a seguir conta com suporte às seguintes instruções:

- ADD
- SUB
- AND
- OR
- NOR
- XOR
- SLT
- BEQ
- BNE
- LW
- SW

A compilação e execução da simulação juntamente com a abertura do arquivo de "*dump"* com o programa *GTKWave* através de algum terminal *linux* se encontra facilitada com o uso de comando "*make"* no terminal, como exemplificado a seguir:

- Para criar o arquivo de simulação com o *iverilog*:

  $ *make*

  # O arquivo de saída "Processor" será gerado.

- Em seguida para executar o arquivo gerado:

  $ *make run*

  # A simulação do processador MIPS será executada e o arquivo de dump "MIPS.vcd" será gerado.

- Para abrir o arquivo de *dump* com o *GTKWave*:

  $ *make gtk*

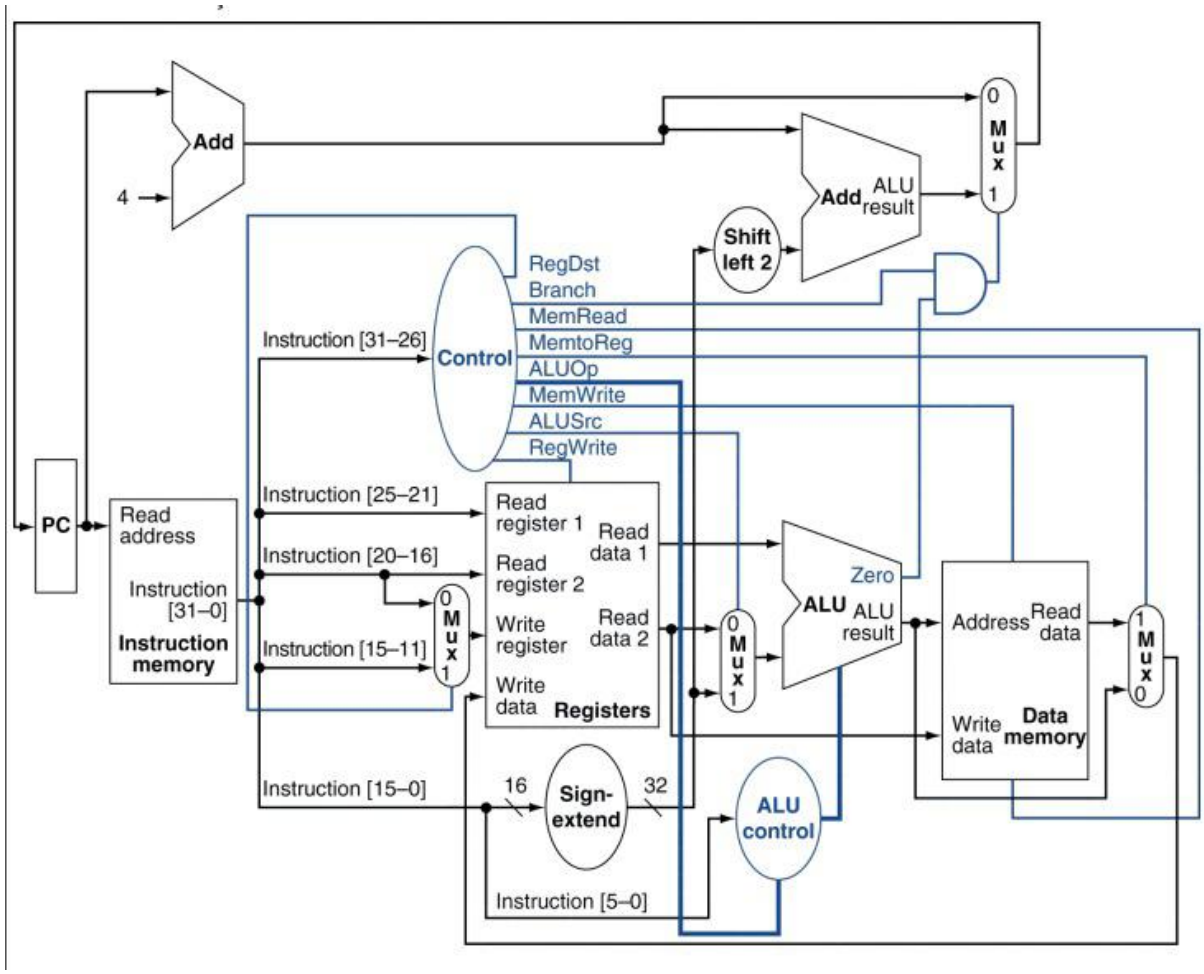
  # O programa GTKWave abrirá e carregará o arquivo dump "MIPS.vcd".

Todo processo de compilação, simulação e visualização de *dumpfile* pode também ser realizada manualmente. Logo, basta compilar o arquivo “Main.v” com o *Icarus-Verilog*. Não sendo necessário a instalação do comando *make* caso ele não se encontre pré-instalado no sistema operacional.

DESENVOLVIMENTO:

O desenvolvimento do trabalho foi realizado tendo em vista os conhecimentos adquiridos durante as aulas. Inicialmente a criação dos módulos auxiliares menores e mais simples foi realizada, posteriormente o avanço em direção aos módulos maiores e mais complexos foi sendo alcançado. Por último o módulo principal foi criado e todas as conexões foram efetuadas, tendo como referencia o diagrama a seguir:

Figura 1

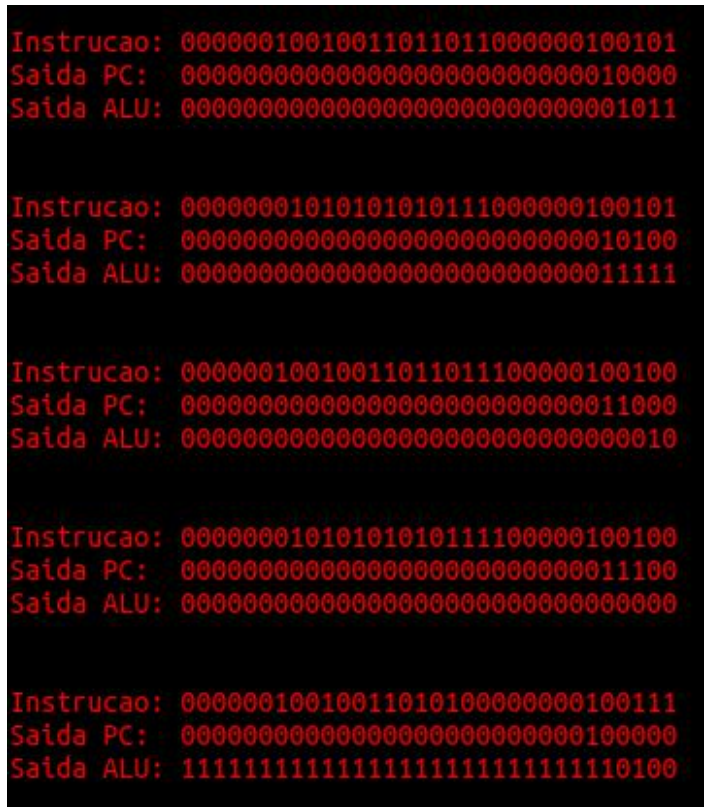


Fonte: Figura 4.17 da 4ª edição do livro

Para o teste de funcionamento do caminho de dados implementado, foram montadas 21 instruções para verificar o comportamento dos circuitos em simulação. Há cerca de duas instruções para cada tipo de instrução suportada pelo projeto, no entanto, o uso de desvios condicionais faz algumas delas serem executadas mais de uma vez.

Durante a simulação é exibido na tela a instrução entrada, o valor atual de *PC* e o resultado obtido na saída da *ALU,* como pode ser observado na imagem abaixo:

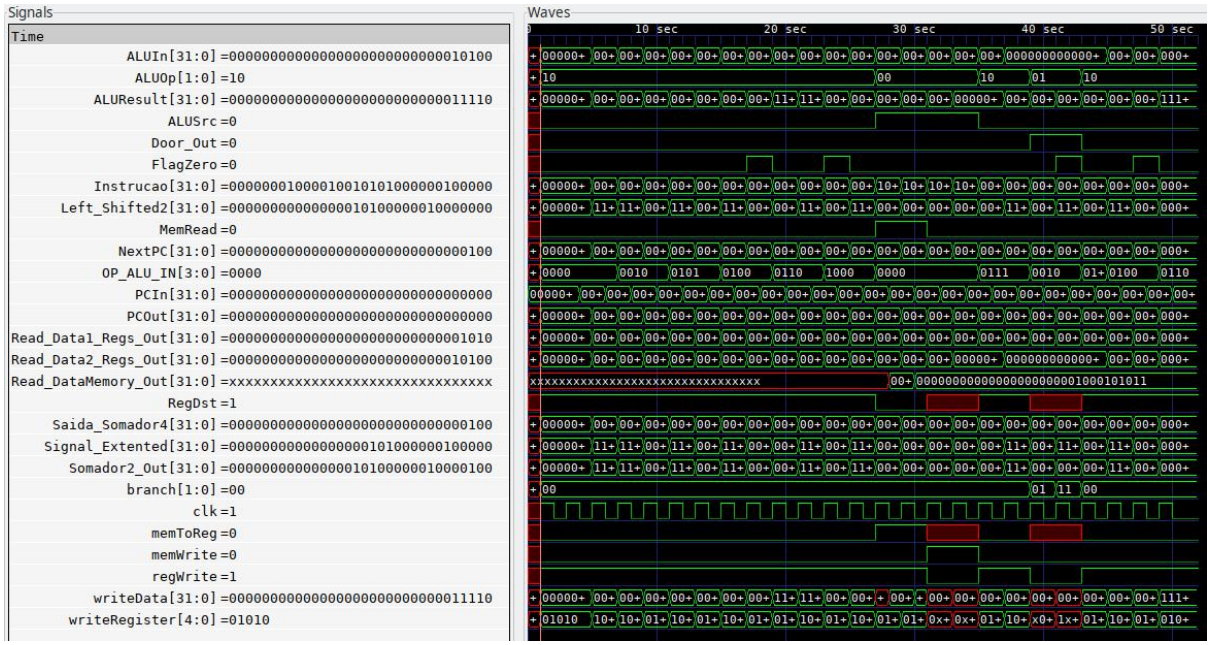
Figura 2



Fonte: Simulação em terminal.

A visualização das formas de ondas e valores das principais entradas e saídas do caminho de dados implementado é uma ótima forma de conferir e analisar o funcionamento do caminho de dados MIPS, como ilustrado na figura a seguir:

Figura 3



Fonte: *GTKWave*

CONSIDERAÇÕES FINAIS:

O respectivo trabalho foi de suma importância para o aprendizado de arquitetura de processamento, além de possibilitar que uma melhor compreensão de como funciona um processador no seu nível mais baixo de abstração.

Agradecimentos ao Professor José Augusto Nacif, por todas as dúvidas tiradas e exemplos dados em sala de aula.

Todo o trabalho encontra-se disponível no seguinte repositório do GitHub:
https://github.com/Globson/MIPS_32bit_Without_Pipeline