



## **Laboratório de Arquitetura e Organização de computadores II**

### **Prática 2**

Alunos: Anélio Gonçalves Caldas e Ryan Eduardo Mansur Vasconcelos

#### **1. Introdução**

Na prática 2, vamos implementar um processador, com uma memória de dados usando a biblioteca ram lpm, e uma memória de instruções usando a biblioteca rom lpm.

#### **2. Objetivos**

O objetivo desta prática é implementarmos os conceitos vistos nas aulas sobre funcionamento de processadores, além de utilizar o que já implementamos anteriormente.

#### **3. Desenvolvimento**

##### **3.1 Projeto**

Para esta prática, foi utilizada uma memória de dados com 256 posições, e cada posição preenchida com uma palavra de 16 bits. Também, foi utilizada uma memória rom de instruções com 32 posições, e cada posição preenchida com uma palavra de 16 bits.

A ideia do projeto, é carregar previamente as instruções na memória de instruções para se lida. A cada instrução será mostrado o resultado dos registradores presentes no processador. As instruções serão carregadas por meio de um arquivo .mif em ambas as memórias.

O projeto segue de acordo com a imagem abaixo.



O formato das instruções para parte 1 do roteiro é dada abaixo.

Instruction	Binary	Decimal
MV R0, #2	0000 1 000 00000010	2050
MV R1, #3	0000 1 001 00000011	2307
ADD R1, R0	0010 0 001 00000 000	8448
MV R2, R0	0000 0 010 00000 000	512
ADD R2, #4	0010 1 010 00000100	10756
SUB R2, R1	0011 0 010 00000 001	12801
MV R3, R2	0000 0 011 00000 010	770
ADD R0,R3	0010 0 000 00000 011	8195
AND R0, R1	0100 0 000 00000 001	16385
SLL R1, R3	1010 0 001 00000 011	41219
SRL R1, R3	1011 0 001 00000 011	45315
MV R2, R1	0000 0 010 00000 001	513
AND R2, #3	0100 1 010 00000011	18947
SLL R0, #2	1010 1 000 00000010	43010
SUB R0, #1	0011 1 000 00000001	14337
SRL R1, #2	1011 1 001 00000010	47362
ADD R1, R1	0010 0 001 00000 001	8449
LD R2, R0	0101 0 010 00000 000	20992
ADD R2, R3	0010 0 010 00000 011	8707
SUB R0, #3	0011 1 000 00000011	14339
SD R2, R0	0110 0 010 00000 000	25088
LD R0, R0	0101 0 000 00000 000	20480
SUB R0, R3	0011 0 000 00000 011	12291

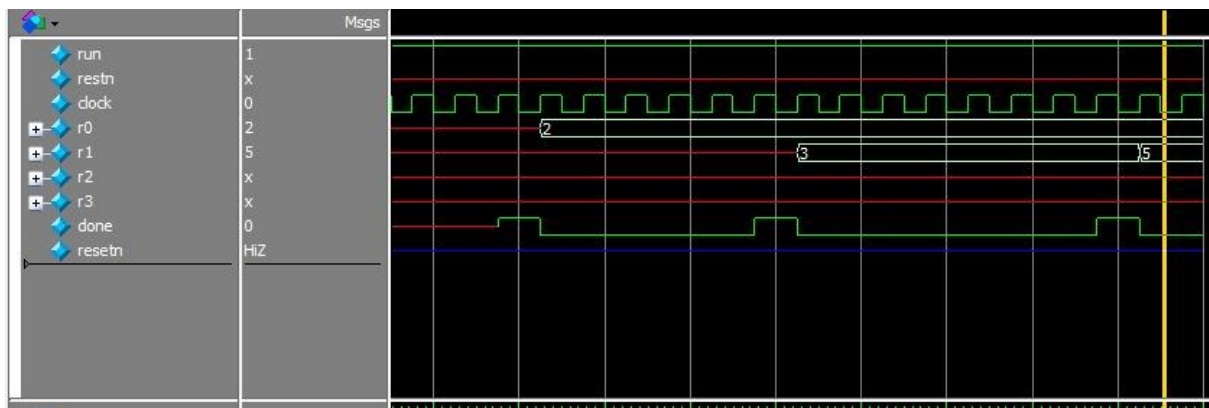
## 3.2 Código

O código em geral foi criado para simular exatamente o processador do arquivo Praticall\_Eng. Foi criado cada bloco um por um. Os módulos ADDR e DOUT não possuem clock no projeto do processador, isso porque esses módulos mandam um endereço de memória para as memórias de dados, usadas para add e sub. O counter também não possui clock, uma vez que ele gera o endereço para ler a memória de instruções.

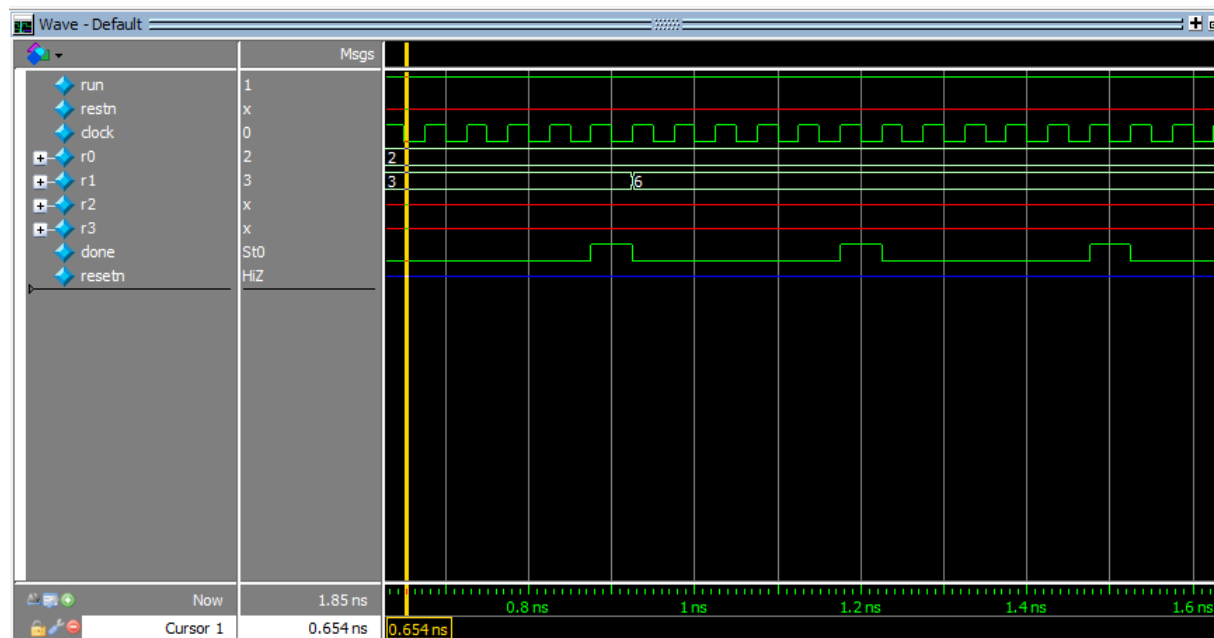
## 3.3 Simulação

Infelizmente a simulação não ocorreu como o esperado, e não foi possível fazer as simulações de b{cond} e SLL e SRL.

As demais instruções foram simuladas em uma versão do código, e podem ser vistas abaixo.



Acima podemos ver a simulação das 3 primeiras instruções do teste 1, instruções de mv e add.



Acima podemos ver a simulação de ld, buscando o dado da memória e escrevendo em R1.

O código dessa simulação é uma versão anterior, e também estará junto com o documento da prática.

## 4. Conclusão

Nesta prática podemos então usar pela primeira vez a memória rom lpm. Ademais, pode-se aprimorar a visão sobre a construção de um processador, como também a análise e desenvolvimento de um projeto para o mesmo. Infelizmente não foi possível concluir a prática como o esperado, porém, o que com o que foi feito neste exercício, pode-se colocar em prática e entender melhor o conteúdo das aulas teóricas.