**UFPB**

**Aluno:** Ronaldo Augusto de Morais Sobrinho

**Matrícula:** 11211401

**Aluno:** José Neto

**Matrícula:** 20171019455

**Professor:** Alisson Brito

**Disciplina:** Arquitetura de Computadores II

**Período:** 2017.1

**Projeto: Simulador – Unidade de Controle**

**Descrição:** O projeto tem o objetivo de simular uma Unidade de Controle, por meio do Programa (Código Fonte em um arquivo .txt), e Memória de Dados (escolhido a partir de um arquivo .txt) para armazenamento/consulta. O simulador conta com algumas instruções pré-programadas para realização da execução desse código. São elas:

**- ADD OP1,OP2,OP3**

Esta instrução irá somar os valores dos Operadores2 e 3 (OP2 e OP3), e armazenar o resultado no endereço de memória do Operador1 (OP1)

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- SUB OP1,OP2,OP3**

Esta instrução irá subtrair o valor do Operador3 no Operador2 (Operador2 – Operador3), e armazenar o resultado no endereço de memória do Operador1 (OP1)

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- MUL OP1,OP2,OP3**

Esta instrução irá multiplicar os valores dos Operadores2 e 3 (Operador2 \* Operador3), e armazenar o resultado no endereço de memória do Operador1 (OP1)

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- DIV OP1,OP2,OP3**

Esta instrução irá dividir o valor do Operador2 pelo Operador3 (Operador2 / Operador3), e armazenar o resultado no endereço de memória do Operador1 (OP1)

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- JMPM OP1,OP2,OP3**

Esta instrução irá “saltar” (jump) para o endereço de memória do Operador1 se o Operador2 for maior que o Operador3 (Operador2 > Operador3). Então, irá dar continuidade a execução do programa, de acordo com o “salto”.

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- JMPL OP1,OP2,OP3**

Esta instrução irá “saltar” (jump) para o endereço de memória do Operador1 se o Operador2 for menor que o Operador3 (Operador2 < Operador3). Então, irá dar continuidade a execução do programa, de acordo com o “salto”.

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, poderão ser Registradores, Endereços de Memória e/ou Constantes.

**- JMPZ OP1,OP2,x**

Esta instrução irá “saltar” (jump) para o endereço de memória do Operador1 se o Operador2 for igual a 0. Então, irá dar continuidade a execução do programa, de acordo com o “salto”.

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- O Operador2, poderá ser Registradores, Endereços de Memória e/ou Constantes.

**- JMP OP1,x,x**

Esta instrução irá “saltar” (jump) para o endereço de memória do Operador1 (de forma incondicional, ou seja, deve-se ter cuidado para não iniciar um loop infinito).

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- Os Operadores2 e 3, deverão ser o caractere x (pois irão assumir valores nulos na operação).

**- MOV OP1,OP2,x**

Esta instrução irá mover para o endereço de memória do Operador1 o valor do Operador2 (seja ele um Registrador ou Endereço de Memória / Constante)..

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- O Operador2, poderá ser um Registrador (ex. R1), um Endereço de memória (ex.<10>) ou uma constante (ex. 10).

- O Operador3, deverá ser o caractere x (pois irá assumir valor nulo na operação).

**- STR OP1,OP2,x**

Esta instrução irá salvar para o endereço de memória do Operador1 o valor do Operador2 (seja ele um Registrador ou Endereço de Memória / Constante)..

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- O Operador2, poderá ser um Registrador (ex. R1), um Endereço de memória (ex.<10>) ou uma constante (ex. 10).

- O Operador3, deverá ser o caractere x (pois irá assumir valor nulo na operação).

**- LOAD OP1,OP2,x**

Esta instrução irá “carregar” para o endereço de memória do Operador1 o valor contido do Operador2 (seja ele um Registrador ou Endereço de Memória / Constante)..

- O Operador1 sempre será um Registrador (para armazenamento do valor resultante).

- O Operador2, poderá ser um Registrador (ex. R1), um Endereço de memória (ex.<10>).

- O Operador3, deverá ser o caractere x (pois irá assumir valor nulo na operação).