

# UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE



**Docente:** Alberto Nicolau

**Discentes:** João Vitor Rafael Chrisótomo, Vitor Fassanaro, João Pedro Azevedo

**Projeto de um microcontrolador**

## Codificação das instruções

O projeto busca utilizar uma ISA mais próxima possível de uma ISA RISC pura. Com poucas instruções, instruções de mesmo tamanho e separadas por grupos baseados em funcionalidade.

Instruções da primeira etapa:

MOV - Move de um registrador para outro. Ex: MOV RxRy XXXX XXXX

ADD - Soma valores de 2 registradores e o resultado vai para o primeiro argumento. Ex: ADD RxRy XXXX XXXX

SUB - Subtrai valores de 2 registradores e o resultado vai para o primeiro. Ex: SUB RxRy XXXX XXXX

MVI - Carrega um valor em um registrador de 8bits. Ex: MVI RxXX 0000 0001

IN - Busca um valor em Rin e o coloca em Rx. Ex: IN RxXX XXXX XXXX

OUT - Leva um dado de Rx para Rout. Ex: OUT RxXX XXXX XXXX

Instruções extras da segunda etapa

LD - Carraga em Rx o que está em Ry. Ex: LD RxRy XXXX XXXX

ST - Armazena no endereço apontado por Rx o dado em Ry. Ex: ST RxRy XXXX XXXX

JMP - Vá para end. Ex: JMP XXXX XXXX XXXX

JZ - Se ult==0, vá para end. Ex: JZ XXXX XXXX XXXX

Instruções pensadas pelo grupo

SHL - Shift left. Ex: SHL RxXX XXXX XXXX X

SHR - Shift right. Ex: SHR RxXX XXXX XXXX

Como temos 4 bits para codificar a instrução (sem contar os registradores e os valores), as mesmas podem ser separadas em 4 grandes grupos:

00 - LD, ST, MOV, MVI.

01 - ADD, SUB, SHL, SHR.

10 – Não está em uso ainda

11 - IN, OUT, JMP, JZ.

E com uma codificação arbitrária temos:

0000 - LD

0001 - ST

0010 - MOV

0011 - MVI

0100 - ADD

0101 - SUB

0110 - SHL

0111 - SHR

1000 -

1001 -

1010 -

1011 -

1100 - IN

1101 - OUT

1110 - JMP

1111 - JZ

O projeto possui apenas 4 registradores internos de dados, nossa codificação para os registradores precisa de apenas 2 bits e é feita de maneira simples:

Reg0 - 00

Reg1 - 01

Reg2 - 10

Reg3 - 11

De maneira com que uma instrução ADD Reg0, Reg1 seria escrita da seguinte forma:

0100 0001 0000 0000.

## **Codificação das entradas, saídas e função dos componentes**

### **Multiplexadores**

#### **MUX0**

5 entradas:

- 000 - ALU
- 001 - MUX2
- 010 - Memória de programa
- 011 - Memória de dados
- 100 - Rin (Entrada externa)

#### **MUX1**

4 entradas:

- 00 - Registrador Interno 00
- 01 - Registrador Interno 01
- 10 - Registrador Interno 10
- 11 - Registrador Interno 11

#### **MUX2**

3 saídas:

- 00 - MUX0
- 01 - Registrador de saída
- 10 - ALU

#### **MUX3**

4 entradas

- 00 - Registrador Interno 00
- 01 - Registrador Interno 01
- 10 - Registrador Interno 10
- 11 - Registrador Interno 11

## Unidades lógicas aritméticas

ALU (Só existe um)

4 funções (possuem 3bits para dar folga para futuras instruções):

000 - ADD

001 - SUB

Extras:

010 - SHIFT\_R

011 - SHIFT\_L