

# Trabalho 2 - Organização e Arquitetura de Computadores

Aluno(a): Júlia Yuri Garcia Baba

Matrícula: 19/0057921

## 1. Apresentação

A proposta do trabalho 2 é desenvolver um programa em linguagem C para simular o RV32I, a fim de entender melhor o funcionamento do processador.

## 2. Descrição da implementação

Primeiramente o programa consiste em reutilizar funções já feitas no trabalho 1, além de implementar outras sete funções principais:

- **fetch()**: lê uma instrução da memória e atualiza o pc para apontar para a próxima instrução.
- **decode()**: extrai todos os campos da instrução, sendo esses campos: opcode, rd, rs1, rs2, shamt, funct3 e funct7. Sendo esse processo realizado por meio de deslocamentos lógicos de bits. Portanto, por exemplo, ao extrair o campo rd (5 bits) de uma instrução, sabendo que o opcode possui 7 bits, o registrador(ri) é deslocado 7 bits para direita e depois é feita uma operação AND, resultando no campo da instrução desejada.
- **execute()**: executa a instrução que foi lida pela função fetch() e decodificada por decode(), de acordo com os campos de instrução opcode, funct3 e funct7 que será explicada mais a frente
- **step()**: executa uma instrução do RV32I: fetch(pc), decode(), execute().
- **run()**: executa o programa até encontrar uma chamada de sistema para encerramento, ou até o pc ultrapassar o limite do segmento de código.
- **dump\_mem(int start, int end, char format)**: imprime o conteúdo da memória a partir do endereço start até o endereço end, ambos em índice do vetor de memória. O formato pode ser em hexadecimal ou decimal.
- **dump\_reg(char format)**: imprime o conteúdo do banco de registradores.

Durante o desenvolvimento da função *execute()*, foram implementadas as classes de instruções, de modo geral, para o desenvolvimento de cada classe, com base na tabela de formatos e no guia prático RISC-V, foram se olhando os nomes da instrução e dependendo de qual está em questão os campos de instrução são considerados ou desconsiderados, desse modo:

- **lógico-aritméticas**: executa operações, com ou sem sinal, usando os registradores *rs1* e *rs2* e armazena o registrador *rd*.
- **saltos condições**: se *rs1* é igual a *rs2*, *pc* é somado ao imediato.
- **jumps**: é somado quatro ao *pc* e armazenado no registrador *rd*, em seguida, *pc* é somado ao imediato.
- **acesso à memória**: é feito o uso das funções do trabalho anterior, passando-se como parâmetros *rs1*, o imediato e o *rs2*.
- **chamadas do sistema**: verifica o *breg[17]*, com base em seu valor, imprime o valor de *breg[10]*, como inteiro ou imprime a string, assim como, também pode encerrar o programa.

## 3. Testes e resultados

O maior desafio durante o desenvolvimento do programa foi inicialmente entender os passos da execução, por conta da complexidade do funcionamento do RV32I. Feito isso, o programa pode ser feito ser maiores percalços, apenas se atentando a alguns detalhes, por exemplo, deve-se subtrair quatro de pc, para que assim ele possa passar para próxima instrução, são recorrências pontuais, mas que de certa forma passaram despercebidas à princípio. Contudo, o programa conseguiu passar por todos os testes.

