

Instruções:

1. Esta atividade pode ser realizada individualmente ou em **até TRÊS** alunos (no máximo).
2. A defesa da implementação será realizada dia **25/08/2023**, e a data final da postagem do relatório até às **18h59** do mesmo dia **25/08/2023**. Defesa e/ou entrega em atraso **não serão aceitas**.
3. A implementação poderá ser realizada em **qualquer** linguagem de programação, sendo que o código deverá ser entregue (preferencialmente link do GITHUB) no AVA ou Material didático e apresentado.
4. Se forem identificados trabalhos com grau de similaridade que caracterize cópia ou adaptação (autorizadas ou não pelos seus autores originais), a nota dos grupos será a nota de um trabalho dividida pelo número de grupos que entregou esses trabalhos similares.

ATIVIDADE

Elaborar um programa que, mediante a entrada das seguintes informações de duas organizações diferentes (Organização A e Organização B):

- Número de ciclos de cada formato de instrução do RISC-V;
- Tempo de relógio (Clock); e
- Arquivo de memória de instrução (ROM) em linguagem de máquina (binário ou hexadecimal).

Apresente os seguintes resultados:

- O número de ciclos totais gastos por cada programa em cada organização;
- Os ciclos por instrução (CPI) médio para o programa em cada organização;
- Qual organização teve o melhor desempenho e quão melhor foi o desempenho.

O programa deve conseguir classificar todas as instruções presentes na Figura 2.3 do Guia prático do RISC-V, apresentada, a seguir:

31	25	24	20	19	15	14	12	11	7	6	0	
imm[31:12]								rd	0110111			U lui
imm[31:12]								rd	0010111			U auipc
imm[20 10:1 11 19:12]								rd	1101111			J jal
imm[11:0]				rs1	000		rd		1100111			I jalr
imm[12 10:5]		rs2		rs1	000		imm[4:1 11]		1100011			B beq
imm[12 10:5]		rs2		rs1	001		imm[4:1 11]		1100011			B bne
imm[12 10:5]		rs2		rs1	100		imm[4:1 11]		1100011			B blt
imm[12 10:5]		rs2		rs1	101		imm[4:1 11]		1100011			B bge
imm[12 10:5]		rs2		rs1	110		imm[4:1 11]		1100011			B bltu
imm[12 10:5]		rs2		rs1	111		imm[4:1 11]		1100011			B bgeu
imm[11:0]				rs1	000		rd		0000011			I lb
imm[11:0]				rs1	001		rd		0000011			I lh
imm[11:0]				rs1	010		rd		0000011			I lw
imm[11:0]				rs1	100		rd		0000011			I lbu
imm[11:0]				rs1	101		rd		0000011			I lhu
imm[11:5]		rs2		rs1	000		imm[4:0]		0100011			S sb
imm[11:5]		rs2		rs1	001		imm[4:0]		0100011			S sh
imm[11:5]		rs2		rs1	010		imm[4:0]		0100011			S sw
imm[11:0]				rs1	000		rd		0010011			I addi
imm[11:0]				rs1	010		rd		0010011			I slti
imm[11:0]				rs1	011		rd		0010011			I sltiu
imm[11:0]				rs1	100		rd		0010011			I xori
imm[11:0]				rs1	110		rd		0010011			I ori
imm[11:0]				rs1	111		rd		0010011			I andi
0000000		shamt		rs1	001		rd		0010011			I slli
0000000		shamt		rs1	101		rd		0010011			I srli
0100000		shamt		rs1	101		rd		0010011			I srai
0000000		rs2		rs1	000		rd		0110011			R add
0100000		rs2		rs1	000		rd		0110011			R sub
0000000		rs2		rs1	001		rd		0110011			R sll
0000000		rs2		rs1	010		rd		0110011			R slt
0000000		rs2		rs1	011		rd		0110011			R sltu
0000000		rs2		rs1	100		rd		0110011			R xor
0000000		rs2		rs1	101		rd		0110011			R srl
0100000		rs2		rs1	101		rd		0110011			R sra
0000000		rs2		rs1	110		rd		0110011			R or
0000000		rs2		rs1	111		rd		0110011			R and
0000		pred		succ		00000	000	00000	0001111			I fence
0000		0000		0000		00000	001	00000	0001111			I fence.i
0000000000000						00000	000	00000	1110011			I ecall
0000000000001						00000	000	00000	1110011			I ebreak
csr				rs1	001		rd		1110011			I csrrw
csr				rs1	010		rd		1110011			I csrrs
csr				rs1	011		rd		1110011			I csrrc
csr				zimm	101		rd		1110011			I csrrwi
csr				zimm	110		rd		1110011			I csrrsi
csr				zimm	111		rd		1110011			I csrrci

VALIDAÇÃO

Para a validação, serão disponibilizados os códigos para teste no AVA da disciplina além disso serão utilizados códigos inéditos a serem utilizados para validação no dia da apresentação. Esses códigos serão gerados pelo RARS (dump file) e editados para que possua comportamento sequencial. Não sendo necessário, nesse momento, simular possíveis desvios ocorridos no programa.