CI1212 Arquitetura de Computadores - 2025/1 - UFPR

Especificação do Trabalho 1

O aluno deverá escrever, em Assembly de REDUX-V, um programa que some dois vetores de 10 posições fazendo R=A+B. O código deve inicializar os vetores A, B e R. Os vetores devem iniciar logo após as instruções na memória (note que REDUX-V é uma arquitetura Von Neumann). **Você deve implementar a soma com loop para percorrer o vetor**. A lista de instruções é dada na próxima página. **O Assembly deve ser escrito no formato legível pelo emulador EGG (https://github.com/gboncoffee/egg)**.

- O vetor R deve inicializar com zero.
- A = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18}.
- B = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}.

Regras Gerais de Entrega

O código Assembly deve ser entregue através do Moodle da disciplina no formato legível para o emulador EGG, que será utilizado para teste.
Casos não tratados no enunciado deverão ser discutidos com o professor.
Os trabalhos devem ser feitos individualmente. **A cópia do trabalho (plágio), acarretará em nota igual a Zero para todos os envolvidos.**

Os trabalhos deverão ser apresentados de forma oral pelo aluno. A nota irá considerar domínio do tema, robustez da solução e rigorosidade da metodologia.

Descrição da Arquitetura REDUX-V

Informações gerais

1. **Tamanho da palavra**: 8 bits
2. Arquitetura **Load-Store**
3. Memória endereçada a cada 1 Byte
4. Arquitetura **Von Neumann**

Conjunto de instruções (ISA)

Opcode	Tipo	Mnemonic	Nome	Operação
0000	R	brzr	Branch On Zero Register	if (R[ra] == 0) PC = R[rb]
0001	I	ji	Jump Immediate	PC = PC + Imm.
0010	R	ld	Load	R[ra] = M[**R[rb]**]
0011	R	st	Store	M[**R[rb]**] = R[ra]
0100	I	addi	Add Immediate	**R[0]** = **R[0]** + Imm.
0101	Reservado - Instruções adicionais			
0110				
0111				
1000	R	not	Not	R[ra] = not R[rb]
1001	R	and	And	R[ra] = R[ra] and R[rb]
1010	R	or	Or	R[ra] = R[ra] or R[rb]
1011	R	xor	Xor	R[ra] = R[ra] xor R[rb]
1100	R	add	Add	R[ra] = R[ra] + R[rb]
1101	R	sub	Sub	R[ra] = R[ra] - R[rb]
1110	R	slr	Shift Left Register	R[ra] = R[ra] << R[rb]
1111	R	srr	Shift Right Register	R[ra] = R[ra] >> R[rb]

Formatos das instruções

Tipo I

Bits	7	6	5	4	3	2	1	0
	Opcode				Imm.			

Tipo R

Bits	7	6	5	4	3	2	1	0
	Opcode				Ra		Rb	