

CI112 Arquitetura de Computadores - 2023/1 - UFPR

Especificação do Trabalho 1

O aluno deverá desenvolver o projeto de implementação da arquitetura Sagui. Trata-se de uma arquitetura de 8-bits, do tipo load-store endereçada por byte, que foi desenvolvida especialmente para essa disciplina. A organização a ser implementada poderá ser Monociclo. A lista de instruções bem como seus formatos são dados na próxima página.

O trabalho deve ser desenvolvido individualmente. Recomenda-se a seguinte ordem para o desenvolvimento do trabalho:

1. Desenvolvimento com diagrama de caixas do projeto do caminho de dados (datapath) do processador.
2. Desenvolvimento com diagrama de caixas do projeto interno da ULA, pensando no código da ULA a ser usado para cada operação.
3. Desenvolvimento no Logisim Evolution do projeto com componentes e fios.
4. Teste das instruções de forma individual (testar cada instrução separadamente) e depois teste de pequeno código para ver o comportamento com múltiplas instruções.

Regras Gerais de Entrega e Apresentação

A implementação será feita no logisim evolution, onde poderão ser utilizados todos os componentes pré-prontos ali existentes. Cada aluno deverá entregar o diagrama do Sagui bem como o projeto no logisim evolution. A entrega será feita pelo Moodle dividida em duas partes

- Diagrama em PDF contendo o **diagrama do Sagui** e o **projeto da ULA** com eventuais detalhes do projeto.
- Projeto no formato **Logisim Evolution**

As datas limite de **entrega serão sempre às 6h (a.m.) do dia**, impreterivelmente. (não confundir com 18h)

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.

Opcode	Tipo	Mnemonic	Nome	Operação
Controle				
0000	R	brzr	Branch On Zero Register	if (R[ra] == 0) PC = R[rb]
0001	I	brzi	Branch On Zero Immediate	if (R[0] == 0) PC = PC + Imm.
0010	R	jr	Jump Register	PC = R[rb]
0011	I	ji	Jump Immediate	PC = PC + Imm.
Dados				
0100	R	ld	Load	R[ra] = M[R[rb]]
0101	R	st	Store	M[R[rb]] = R[ra]
0110	R	movr	Move Register	R[ra] = R[rb]
0111	I	movh	Move High	R[0] = {Imm. + R[0](3:0)}
1000	I	movl	Move Low	R[0] = {R[0](7:4) + Imm.}
Aritmética				
1001	R	add	Add	R[ra] = R[ra] + R[rb]
1010	R	sub	Sub	R[ra] = R[ra] - R[rb]
Lógica				
1011	R	and	And	R[ra] = R[ra] & R[rb]
1100	R	or	Or	R[ra] = R[ra] R[rb]
1101	R	not	Not	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]

Tipo R							
7	6	5	4	3	2	1	0
opcode				Ra		Rb	

Tipo I							
7	6	5	4	3	2	1	0
opcode				Imm			

