

Arquitetura e Organização de Computadores

Turma 3 – 2024-01

Projeto RISC-V Multiciclo

Objetivo: montar e simular uma versão do processador RISC-V multiciclo.

Descrição:

Neste trabalho deve-se desenvolver uma arquitetura RV multiciclo em VHDL. Tomando por base os trabalhos da disciplina, deve-se interligar todos os módulos relativos à parte operativa e à parte de controle realizando a arquitetura ilustrada na figura 1.

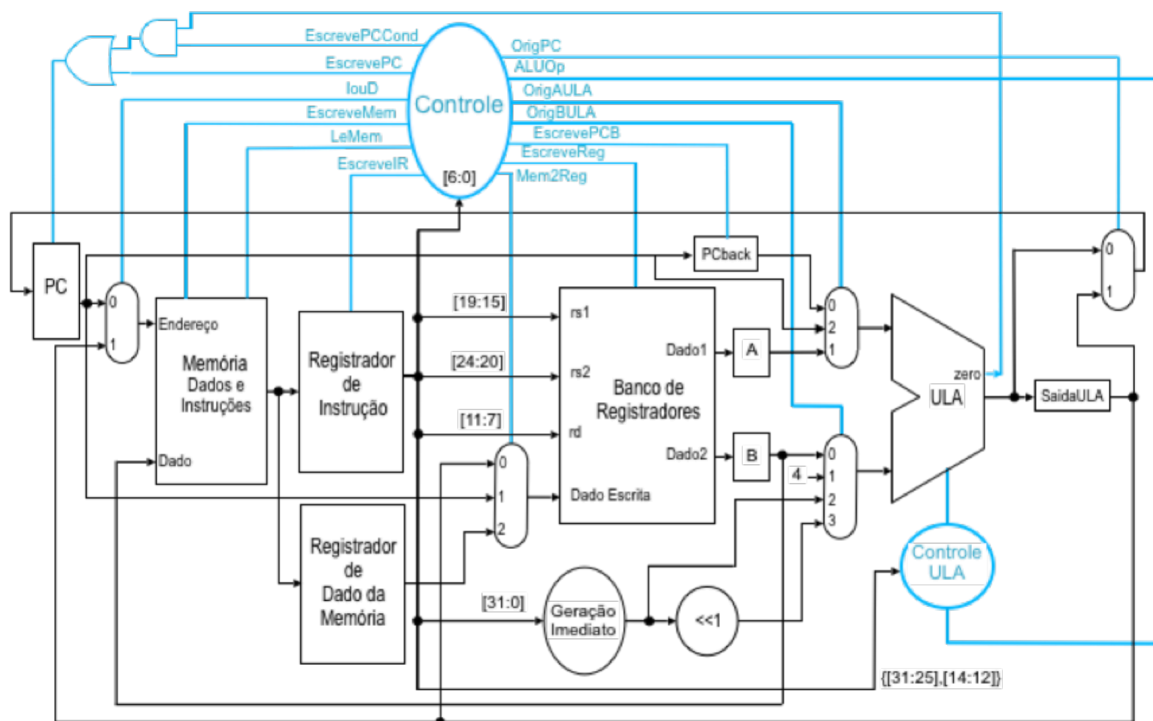


Figura 1. Organização básica do RiscV multiciclo.

O diagrama acima não suporta a execução de todas as instruções do RISC-V. As instruções básicas a serem implementadas são as seguintes:

- LW, SW, ADD, ADDi, SUB, AND, OR, XOR, SLT, JAL, JALR, AUIPC, LUI, BEQ, BNE

Para o projeto da disciplina, os diferentes grupos deverão implementar algumas das instruções adicionais descritas a seguir. X é o banco de registradores.

Grupo Imm:

- ORI: ori rd, rs1, imm32 $X[rd] = X[rs1] \mid \text{imm32}$
- ANDI: andi rd, rs1, imm32 $X[rd] = X[rs1] \& \text{imm32}$
- XORI: xori rd, rs1, imm32 $X[rd] = X[rs1] \wedge \text{imm32}$

Grupo Comp:

- SLTI: slti rd, rs1, imm32 $X[rd] = (X[rs1] < \text{imm32}) ? 1 : 0$
- SLTU: sltu rd, rs1, rs2 $X[rd] = (X[rs1] <_u X[rs2]) ? 1 : 0$
- SLTIU: sltiu rd, rs1, imm32 $X[rd] = (X[rs1] <_u \text{imm32}) ? 1 : 0$

Grupo Shift:

- SLL: deslocamento lógico à esquerda
sll rd, rs1, rs2 $X[rd] = X[rs1] \ll X[rs2]$
- SRL: deslocamento lógico à direita
srl rd, rs1, rs2 $X[rd] = X[rs1] \gg_u X[rs2]$
- SRA: deslocamento aritmético à direita
sra rd, rs1, rs2 $X[rd] = X[rs1] \gg X[rs2]$

Grupo Shift Immediato:

- SLLI: slli rd, rs1, shamt $X[rd] = X[rs1] \ll \text{shamt}$
- SRLI: srl rd, rs1, shamt $X[rd] = X[rs1] \gg_u \text{shamt}$
- SRAI: srai rd, rs1, shamt $X[rd] = X[rs1] \gg \text{shamt}$

Grupo Branch:

- BGE: desvio se maior ou igual
bge rs1, rs2, offset $\text{pc} += \text{sext}(\text{offset}) \text{ se } X[rs1] \geq X[rs2]$
- BGEU: desvio se maior que ou igual, sem sinal
bgeu rs1, rs2, offset $\text{pc} += \text{sext}(\text{offset}) \text{ se } X[rs1] \geq_u X[rs2]$
- BLT: desvio se menor que
blt rs1, rs2, offset $\text{pc} += \text{sext}(\text{offset}) \text{ se } X[rs1] < X[rs2]$
- BLTU: desvio se menor que, sem sinal
bltu rs1, rs2, offset $\text{pc} += \text{sext}(\text{offset}) \text{ se } X[rs1] <_u X[rs2]$

O processador deve ser simulado no ModelSim ou EdaPlayground.

O PC deve ter 32 bits. A memória, como trata-se apenas de simulação, pode ser dimensionada como no simulador, com 4096 palavras de 32 bits. Assim, apenas 12 bits de endereço são necessários para a memória.

A carga de programa para a memória pode ser feita a partir de arquivo texto, como indicado no trabalho de modelagem da memória em VHDL. O código deve ser carregado a partir do endereço 0 e os dados a partir do endereço 0x2000.

A verificação do processador consistirá na execução de programas gerados a partir do RARS.

Entrega: Relatório de implementação, descrevendo os principais pontos do projeto e o código VHDL do processador.

Itens do relatório:

- Introdução
- Descrição do trabalho:
 - Instruções implementadas
 - Descrição da implementação
 - Dificuldades encontradas
- Testes realizados