

1 O processador

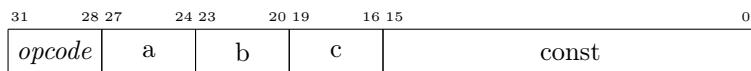
Seu trabalho é completar o circuito de processador¹, conforme o que estudamos em sala. Você deve completar o circuito no simulador Digital. O processador que estamos estudando é uma versão da arquitetura RISC-V.

O RISC-V possui um conjunto reduzido de instruções e vamos implementar apenas 14 delas, conforme a tabela 1. A tabela apresenta cada instrução, seu *opcode* ou código de identificação, e o seu comportamento.

Tabela 1: Conjunto de instruções do processador.

opcode	instrução	comportamento	descrição
0	add c, a, b	$R(c) \leftarrow R(a) + R(b)$	soma
1	sub c, a, b	$R(c) \leftarrow R(a) - R(b)$	subtração
2	xor c, a, b	$R(c) \leftarrow R(a) \oplus R(b)$	ou-exclusivo
3	or c, a, b	$R(c) \leftarrow R(a) \vee R(b)$	disjunção
4	and c, a, b	$R(c) \leftarrow R(a) \wedge R(b)$	conjunção
5	sll c, a, b	$R(c) \leftarrow R(a) << R(b)$	<i>shift left</i>
6	srl c, a, b	$R(c) \leftarrow extZero(R(a) >> R(b))$	<i>shift right logical</i>
7	sra c, a, b	$R(c) \leftarrow extSinal(R(a) >> R(b))$	<i>shift right arith.</i>
8	addi c, a, const	$R(c) \leftarrow R(a) + extSinal(const)$	soma constante aritmética
9	lw c, const(a)	$R(c) \leftarrow M[R(a) + extSinal(const)]$	<i>load from memory</i>
a	sw b, const(a)	$M[R(a) + extSinal(const)] \leftarrow R(b)$	<i>store to memory</i>
b	beq a, b, const	$PC \leftarrow ((R(a) == R(b)? PC + const : PC + 1))$	desvio condicional
c	blt a, b, const	$PC \leftarrow ((R(a) < R(b)? PC + const : PC + 1))$	desvio condicional
d	jal c, const	$PC \leftarrow PC + const, R(c) \leftarrow PC + 1$	<i>jump and link</i>
e	jalr c, const(ra)	$PC \leftarrow R(a) + const, R(c) \leftarrow PC + 1$	<i>jump and link register</i>
-	j const	Pseudo-instrução, equivalente a jal 0, const	salto incondicional

Todas as instruções possuem 32 bits e o mesmo formato, mostrado a seguir:



2 O trabalho

O circuito que você recebeu implementa algumas instruções, mas faltam outras para seu processador funcionar adequadamente. Você deve implementar essas instruções, levando em consideração o projeto estudado em sala de aula. A tabela 2 indica explicitamente quais instruções faltam no seu processador.

Além de alterar o circuito da parte principal do processador, você vai precisar alterar o circuito de controle. Você pode alterar a memória ROM do controle para cumprir com esse objetivo. Se preferir, você também pode criar um circuito de controle com portas lógicas. É permitido usar a função do Digital para criar um circuito a partir da tabela-verdade.

Você **deve** alterar a ULA para realizar as operações de deslocamento (*shift*). É permitido usar, para essa finalidade, um componente disponível no simulador chamado *Barrel shifter*.

Além desse componente, você pode utilizar as portas lógicas, multiplexadores, demultiplexadores, *splitter/merger* (para agrupar ou separar bits), somador. **Não utilize** outros componentes aritméticos como multiplicação ou outros componentes não vistos em sala. Na dúvida, consulte o professor!

Você **não** deve mexer no circuito do Terminal!

¹<http://gvcc.dev.br/teaching/bcc-arqso/codigos/riscv/index.html>

Tabela 2: Instruções do processador que você deve implementar.

opcode	instrução	comportamento	descrição
5	sll c, a, b	$R(c) \leftarrow R(a) << R(b)$	shift left
6	srl c, a, b	$R(c) \leftarrow extZero(R(a) >> R(b))$	shift right logical
7	sra c, a, b	$R(c) \leftarrow extSinal(R(a) >> R(b))$	shift right arith.
b	beq a, b, const	$PC \leftarrow ((R(a) == R(b)?PC + const : PC + 1))$	desvio condicional
c	blt a, b, const	$PC \leftarrow ((R(a) < R(b)?PC + const : PC + 1))$	desvio condicional
d	jal c, const	$PC \leftarrow PC + const, R(c) \leftarrow PC + 1$	jump and link
e	jalr c, const(ra)	$PC \leftarrow R(a) + const, R(c) \leftarrow PC + 1$	jump and link register

2.1 Nova instrução

Você deve escolher uma das instruções abaixo e implementar no seu processador, usando o *opcode f*, que atualmente não é utilizado.

Tabela 3: Instruções para escolha.

opcode	instrução	comportamento	descrição
f	slt c, a, b	$R(c) \leftarrow ((R(a) < R(b)?1 : 0))$	set on less than
f	slti c, a, const	$R(c) \leftarrow ((R(a) < extSinal(const)?1 : 0))$	set on less than immediate
f	sltiu c, a, const	$R(c) \leftarrow ((R(a) < extZero(const)?1 : 0))$	set on less than immediate unsigned
f	bge a, b, const	$PC \leftarrow ((R(a) >= R(b)?PC + const : PC + 1))$	desvio condicional
f	lui c, const	$R(c) \leftarrow (const << 16)$	load upper immediate

2.2 Critérios de correção

Seu trabalho será corrigido considerando:

- Seu circuito deve funcionar!
- Objetividade e clareza do relatório.
- Uso dos métodos e construções adequados no circuito.
- Domínio do trabalho na defesa.
- Adequação à esta especificação.

Não serão aceitos trabalhos entregues fora do prazo. Trabalhos sem relatório ou que não sejam defendidos não serão considerados!

Plágio e/ou uso de inteligência artificial acarretará no trabalho sendo desconsiderado!

2.3 Relatório

A equipe deve entregar um relatório no formato .pdf, explicando como resolveu os desafios do trabalho.

O relatório deve explicar, para cada instrução adicionada, qual parte do circuito precisou ser modificada e porquê. Lembre-se de explicar as mudanças no circuito de dados e no circuito de controle! Descreva como você alterou o circuito. No caso do controle, explique como gerou o circuito de controle e a justificativa para isso. Por exemplo, explique se usou uma ROM de controle ou um circuito gerado a partir da tabela-verdade.

O relatório deve conter todas as informações necessárias para reproduzir a sua solução. Se você gerou um circuito a partir da tabela verdade, seu relatório deve conter a tabela-verdade utilizada e como o circuito foi gerado a partir dela.

Inclua as referências de pesquisa e código que utilizar para seu trabalho no final de seu relatório.

3 Entrega

Você deverá entregar pelo SUAP, até 06/outubro, um arquivo .zip contendo uma pasta, que por sua vez contém:

- Os arquivos `.dig` do trabalho, conforme nomeados pelo professor.
- Um relatório chamado `relatorio.pdf` contendo o nome completo dos integrantes do grupo e o conteúdo descrito na seção 2.3;
- Uma pasta `exemplos/` que contenha todos os programas que você usou para testar seu processador.
- Se você criou novos programas de teste, inclua estes programas na entrega.

O arquivo compactado descrito acima deve ser nomeado com as iniciais do nome de cada integrante separados por um hífen. Por exemplo: se o Fulano da Silva Sousa fez o trabalho com o João de Souza, o arquivo deve ser nomeado `fss-js.zip`. A pasta, dentro do arquivo compactado, deve ter o mesmo nome, a não ser pela extensão.

O trabalho pode ser feito individualmente ou em duplas.

Histórico das Revisões:

- 06/out/2025 - v1.3: ajuste para refletir a prorrogação de prazo.
- 24/set/2025 - v1.2: imediato da instrução `addi` corrigido.
- 19/set/2025 - v1.1: texto sobre referências.
- 17/set/2025 - v1.0: primeira versão.