



---

## Trabalho Prático 2

*Estudo do funcionamento de um processador*

---

**ARQUITETURA DE COMPUTADORES**

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE  
TELECOMUNICAÇÕES E COMPUTADORES

19 de Abril de 2022

## 1 Objetivos

Este trabalho tem como principal objetivo a compreensão do funcionamento de um processador, sendo abordados os seguintes tópicos: codificação de um conjunto de instruções, funcionamento de uma microarquitetura e codificação de programas em linguagem máquina. O ambiente de simulação Logisim é utilizado para consolidação do estudo.

## 2 Especificação do Exercício

Pretende-se completar o projeto de um processador, respeitando o seu modelo de programação e microarquitetura de ciclo único, bem como fazer a validação do projeto usando o ambiente de simulação Logisim e um pequeno programa de teste.

O processador considerado é de 8 bits e tem o seguinte modelo de programação:

- Espaço de endereçamento para código com 256 endereços;
- Espaço de endereçamento para dados também com 256 endereços;
- Quatro registos de uso geral, denominados **r0**, **r1**, **r2** e **r3**;
- Um registo de estado, denominado **CPSR** (do inglês *Current Program Status Register*), que disponibiliza um bit que indica que a última operação realizada deu resultado zero (*flag Z*);
- O conjunto de instruções apresentado na Tabela 1.

Nas instruções apresentadas na Tabela 1, **rd**, **rm** e **rn** representam um dos registos de uso geral do processador, **imm $n$**  simboliza um número natural codificado com  $n$  bits, **label** identifica um endereço na vizinhança de  $\pm 32$  endereços da instrução de salto em causa e **PC** referencia o registo *program counter*.

A microarquitetura do processador está descrita no formato Logisim no ficheiro "ac\_2122v\_tp2.circ", disponível na página de meta disciplina de AC na plataforma Moodle.

Instrução	Descrição	
<b>b rn</b>	Muda a execução para o endereço definido pelo conteúdo de <b>rn</b> .	$PC \leftarrow rn$
<b>bzs label</b>	Muda a execução para o endereço associado ao símbolo <b>label</b> quando a <i>flag Z</i> do registo <b>CPSR</b> apresenta o valor um.	$CPSR.Z == 1 ? PC \leftarrow label : PC \leftarrow PC + 1$
<b>ldr rd, [rn, #imm2]</b>	Copia para <b>rd</b> o conteúdo da posição da memória de dados cujo endereço resulta da soma de <b>rn</b> com o valor da constante <b>imm2</b> .	$rd \leftarrow M[rn + imm2]$
<b>mov rd, #imm4</b>	Estabelece em <b>rd</b> o valor da constante <b>imm4</b> .	$rd \leftarrow imm4$
<b>nor rd, rn, rm</b>	Realiza a função lógica NOR entre os bits da mesma posição de <b>rn</b> e <b>rm</b> , colocando o resultado em <b>rd</b> e atualizando o registo <b>CPSR</b> com a informação do bit Z gerada na ALU.	$rd \leftarrow rn + rm$ $rd == 0 ? CPSR.Z \leftarrow 1 : CPSR.Z \leftarrow 0$
<b>sub rd, rn, #imm2</b>	Subtrai a <b>rn</b> o valor da constante <b>imm2</b> , colocando o resultado em <b>rd</b> e atualizando o registo <b>CPSR</b> com a informação do bit Z gerada na ALU.	$rd \leftarrow rn - imm2;$ $rd == 0 ? CPSR.Z \leftarrow 1 : CPSR.Z \leftarrow 0$
<b>str rd, [rn, rm]</b>	Copia o conteúdo de <b>rd</b> para a posição da memória de dados cujo endereço resulta da soma de <b>rn</b> com <b>rm</b> .	$M[rn + rm] \leftarrow rd$
<b>tst rn, rm</b>	Realiza a função lógica AND entre os bits da mesma posição de <b>rn</b> e <b>rm</b> , atualizando o registo <b>CPSR</b> com a informação do bit Z gerada na ALU.	$rn \cdot rm == 0 ? CPSR.Z \leftarrow 1 : CPSR.Z \leftarrow 0$

Tabela 1: Conjunto de instruções do processador

### 3 Trabalho a Realizar

#### 3.1 Análise da microarquitECTURA

Considere a descrição da microarquitECTURA do processador disponibilizada no ficheiro "ac\_2122v\_tp2.circ".

1. Comente a seguinte afirmação: "*A microarquitECTURA do processador é do tipo von Neumann.*"
2. Indique a funcionalidade realizada pelo bloco **Ext**. Justifique a sua resposta com base no funcionamento das instruções **bzs**, **mov**, **ldr** e **sub**.
3. Identifique as operações realizadas pela Unidade Lógica e Aritmética (ALU) do processador e indique o valor do sinal **OP** associado a cada operação.

#### 3.2 Codificação das instruções

Considere a utilização de um código de comprimento fixo e um esquema de codificação uniforme para a codificação do conjunto de instruções apresentado.

1. Apresente um mapa de codificação para o conjunto de instruções, tendo em conta o formato de codificação da instrução **tst** apresentado na Figura 1.

opcode	–	rn	rm
8 7 6	5 4	3 2	1 0

Figura 1: Formato de codificação da instrução **tst**

2. Indique um valor para o código de operação (**opcode**) de cada instrução. Justifique a sua resposta tendo em conta também o funcionamento da ALU.

#### 3.3 Projeto do decodificador de instruções

Considere o subcircuito **Instruction Decoder** disponibilizado no ficheiro "ac\_2122v\_tp2.circ", que implementa o decodificador de instruções da microarquitECTURA proposta para o processador.

1. Usando uma tabela, apresente os valores das saídas deste bloco, em função dos seus sinais de entrada, para o conjunto de instruções apresentado. Explícite os casos de indiferença (*don't care*) e, se aplicável, as saídas obtidas diretamente do código da instrução.
2. Considerando uma implementação baseada, exclusivamente, numa ROM, determine o conteúdo dessa memória. Indique os valores dos endereços e conteúdos em notação hexadecimal.
3. Indique, em bits, a capacidade da memória ROM considerada no ponto 2.

#### 3.4 Codificação de programas em linguagem máquina

Considere o troço de código apresentado na Listagem 1, escrito na linguagem *assembly* do processador.

1. Indique a funcionalidade do troço de código.
2. Traduza o troço de código para código máquina.
3. Pondera-se substituir a instrução **str rd, [rn, rm]** pela instrução **str rd, [rn]**, que realiza a operação  $M[rn] = rd$ .

- (a) Indique, justificando, se é possível fazer esta alteração ao conjunto de instruções mantendo a organização da microarquitetura apresentada.
- (b) Discuta as vantagens e desvantagens desta alteração quanto aos seguintes aspetos: *i)* o endereçamento da memória de dados, *ii)* a densidade do código e *iii)* o impacto no desenho da microarquitetura.

```
        mov    r0, #0
        ldr    r1, [r0, #0]
        nor    r3, r0, r0
        tst    r1, r3
        bzs    skip
        ldr    r2, [r0, #1]
        nor    r1, r1, r1
        nor    r2, r2, r2
        nor    r1, r1, r2
skip:
        mov    r3, #8
        str    r1, [r3, r0]
        mov    r2, #0xc
        b      r2
```

Listagem 1: Programa de teste.

### 3.5 Implementação e simulação no Logisim

Considere a descrição da microarquitetura do processador disponibilizada no ficheiro "ac\_2122v\_tp2.circ".

1. No **Logisim**, implemente o subcircuito **Ext** com base na análise apresentada no ponto 2 da secção 3.1.
2. No **Logisim**, implemente o subcircuito **Interconnections** tendo em conta o resultado indicado no ponto 1 da secção 3.2.
3. No **Logisim**, implemente o subcircuito **Instruction Decoder** com base no projeto apresentado no ponto 2 da secção 3.3.
4. No **Logisim**, programe a memória de código com o código máquina produzido no ponto 2 da secção 3.4. Programe também as primeiras duas posições da memória de dados com os valores três e dez, respetivamente.
5. No **Logisim**, execute o troço de código e, para cada uma das instruções, registre as alterações ocorridas nos registos do processador (**r0** a **r3**, **PC** e **CPSR**) e na memória de dados.

## 4 Avaliação

O trabalho deve ser realizado em grupo e conta para o processo de avaliação da unidade curricular. Na entrega do trabalho, cada grupo deverá submeter na plataforma Moodle os seguintes elementos:

- Um relatório do trabalho realizado, com as respostas às perguntas formuladas no enunciado e as conclusões retiradas do trabalho realizado;
- Nova versão do ficheiro "ac\_2122v\_tp2.circ" com a implementação de todos os subcircuitos desenvolvidos;
- Ficheiros com o conteúdo das memórias de programa e dados para o programa de teste implementado.

**A data limite para a entrega dos trabalhos é nove de maio de 2022.** Após esta entrega, o docente responsável pela lecionação das aulas teórico-práticas combinará com cada grupo de alunos uma data e hora para a realização da apresentação do trabalho.