

Relatório

Tópico: Aula 6 - Morse

| | |
|------------------------------|----------|
| Grupo: Maicon Chaves Marques | 14593530 |
| Pedro Calciolari Jardim | 11233668 |
| João Vitor Pereira Candido | 13751131 |

Este projeto em VHDL implementa um sistema que exibe o código Morse de letras selecionadas utilizando LEDs. Ele é controlado por switches e botões, e gera pulsos para representar pontos e traços de acordo com a tabela Morse. Abaixo está uma descrição detalhada dos componentes e funcionamento do sistema:

O sistema utiliza:

- **Um clock:** para sincronizar as operações.
- **Botões (key1 e key0):** para iniciar o processo e resetar o sistema.
- **Interruptores (sw):** para selecionar qual letra em código Morse será exibida.
- **LED (ledr0):** para indicar os pontos e traços do código Morse.
-

Código VHDL:

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
```

```
USE ieee.numeric_std.ALL; -- Usar pacote correto
```

```
ENTITY aula06morse IS
```

```
    PORT (
```

```
        clock    : IN STD_LOGIC;
```

```
        key1     : IN STD_LOGIC;    -- Botão para iniciar
```

```
        key0     : IN STD_LOGIC;    -- Botão de reset
```

```
        sw       : IN STD_LOGIC_VECTOR(2 DOWNTO 0); -- Interruptores SW2-0 para a letra
```

```
        ledr0    : OUT STD_LOGIC    -- LED para exibir Morse
```

```
    );
```

```
END ENTITY;
```

```
ARCHITECTURE behavior OF aula06morse IS
```

```
    -- Definição de parâmetros
```

```
    SIGNAL morse_seq : STD_LOGIC_VECTOR(10 DOWNTO 0); -- Armazena os pontos e traços
```

SIGNAL counter : INTEGER RANGE 0 TO 50000000; -- Contador para gerar pulsos de 0.5 e 1.5s

SIGNAL pulse_0_5s : STD_LOGIC := '0'; -- Pulso de 0.5 segundos

SIGNAL current_bit : INTEGER RANGE 0 TO 11 := 0; -- Índice do bit atual no código Morse

-- Definir constantes de pontos e traços para cada letra

CONSTANT A_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00000011101"; --

A: .-

CONSTANT B_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00101010111"; --

B: -...

CONSTANT C_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "10111010111"; --

C: -. .

CONSTANT D_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00001010111"; --

D: -..

CONSTANT E_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00000000001";

-- E: .

CONSTANT F_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00101110101"; --

F:

CONSTANT G_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00101110111"; --

G: --.

CONSTANT H_morse : STD_LOGIC_VECTOR(10 DOWNT0 0) := "00001010101";

-- H:

-- Processo para selecionar o código Morse baseado nos switches (SW2-0)

BEGIN

PROCESS (key1) -- Adicionado sw à lista de sensibilidade

BEGIN

IF key1 = '0' THEN

CASE sw IS

WHEN "000" => morse_seq <= A_morse; -- Letra A

WHEN "001" => morse_seq <= B_morse; -- Letra B

WHEN "010" => morse_seq <= C_morse; -- Letra C

WHEN "011" => morse_seq <= D_morse; -- Letra D

WHEN "100" => morse_seq <= E_morse; -- Letra E

WHEN "101" => morse_seq <= F_morse; -- Letra F

WHEN "110" => morse_seq <= G_morse; -- Letra G

WHEN "111" => morse_seq <= H_morse; -- Letra H

WHEN OTHERS => morse_seq <= A_morse; -- Padrão (Letra A)

END CASE;

END IF;

END PROCESS;

-- Processo para exibir o código Morse no LEDR0

PROCESS (clock, key0, key1)

BEGIN

```

IF key0 = '0' THEN
    current_bit <= 11;
    counter <= 0;
    pulse_0_5s <= '0'; -- Adicionado reset do pulso
    ELSIF key1 = '0' THEN
        current_bit <= 0;

        counter <= 0;
        pulse_0_5s <= '0'; -- Adicionado reset do pulso
    ELSIF rising_edge(clock) THEN
        IF current_bit < 11 THEN
            IF morse_seq(current_bit) = '1' THEN
                -- Gera pulso de 0.5 segundos
                IF counter < 25000000 THEN
                    pulse_0_5s <= '1';
                    counter <= counter + 1;
                ELSE
                    counter <= 0;
                    current_bit <= current_bit + 1;
                END IF;
            ELSE
                -- Gera pulso de 0.5 segundos para '0'
                IF counter < 25000000 THEN
                    pulse_0_5s <= '0';
                    counter <= counter + 1;
                ELSE
                    counter <= 0;
                    current_bit <= current_bit + 1;
                END IF;
            END IF;
        ELSE
            pulse_0_5s <= '0'; -- Garante que o LED apaga quando todos os bits forem
exibidos
        END IF;
    END IF;
END PROCESS;

ledr0 <= pulse_0_5s;

END ARCHITECTURE;

```

RTL VIEWER DO CIRCUITO:

