

UTFPR
Engenharia de Computação
LRCO7A - Lógica Reconfigurável

Atividade 5 - Luzes piscando

Aluno: Deivid da Silva Galvão
Aluno: João Vitor Nakahodo Yoshida
Professor orientador: Marcelo de Oliveira

Dezembro
2024

UTFPR
Engenharia de Computação
LRCO7A - Lógica Reconfigurável

Relatório

Relatório do Trabalho Prático Disciplinar apresentado como requisito parcial à obtenção de nota na disciplina de Lógica Reconfigurável do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Aluno: Deivid da Silva Galvão

Aluno: João Vitor Nakahodo Yoshida

Professor orientador: Marcelo de Oliveira

Dezembro
2024

Conteúdo

1	Introdução	1
2	Implementação	1
3	Resultados	4

1 Introdução

A atividade 5 tem como objetivo implementar um código que acenda um LED por vez, do LED0 ao LED9, e depois do LED9 ao LED0. Incluindo funções como resetar o circuito, pausar o circuito e alterar a velocidade dos LEDs.

2 Implementação

Codigo VHDL

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

-----
entity relat5 is
  — Frequencia do clock , em Hz
  generic ( f_clk : integer := 50_000_000 );
  port (
    clk      : in std_logic;   — Entrada de clock
    reset    : in std_logic;   — Botao de reset (ativa baixo)
    pause    : in std_logic;   — Botao de pausa (ativa baixo)
    — Selecao de velocidade (2 bits)
    velocidade : in std_logic_vector(1 downto 0);
    — LEDs de LED0 a LED9
    leds      : out std_logic_vector(9 downto 0)
  );
end entity;

-----

architecture relat5 of relat5 is
  — Frequencia do clock principal
  constant f_clk_freq : integer := f_clk;
  — Tempo de piscada padrao (500 ms)
  signal tempo_piscar : integer := f_clk / 2;
  — Contador para controlar o tempo
  signal contador : integer range 0 to f_clk_freq - 1 := 0;
  — Indice do LED atual
```

```

    signal led_atual : integer range 0 to 9 := 0;
    — Sinal de controle para acender ou apagar o LED
    signal acender : std_logic := '0';

begin
    — Processamento do contador e controle do LED
    process (clk)
    begin
        if rising_edge(clk) then
            — Reset do circuito
            if reset = '0' then
                contador <= 0;
                led_atual <= 0;
                leds <= (others => '0');
            elsif pause = '0' then
                — Pausa: Se o botao de pausa for pressionado, nao incrementa
                o contador
                null;
            else
                — Ajuste do tempo de piscada conforme a velocidade escolhida
                case velocidade is
                    — 500 ms (padrao)
                    when "01" => tempo_piscar <= f_clk_freq / 2;
                    — 250 ms
                    when "10" => tempo_piscar <= f_clk_freq / 4;
                    — 125 ms
                    when "11" => tempo_piscar <= f_clk_freq / 8;
                    — 500 ms (padrao)
                    when others => tempo_piscar <= f_clk_freq / 2;
                end case;
                — Contador para alternar o estado dos LEDs
                if contador < tempo_piscar then
                    contador <= contador + 1;
                else
                    — Aciona o proximo LED
                    if acender = '1' then
                        — Desliga todos os LEDs
                        leds <= (others => '0');
                        — Acende o LED atual
                        leds(led_atual) <= '1';
                        — Prepara para o proximo ciclo

```

```

        acender <= '0';
    else
        — Altera o estado para 'acender'
        acender <= '1';
    end if;

    — Atualiza o indice do LED
    if acender = '0' then
        if led_atual < 9 then
            — Vai de LED0 a LED9
            led_atual <= led_atual + 1;
        else
            — Depois do LED9, volta para o LED0
            led_atual <= 0;
        end if;
    end if;
    — Reseta o contador para o proximo ciclo
    contador <= 0;
end if;
end if;
end process;
end architecture;

```

3 Resultados

Diagrama RTL gerado pelo Quartus Prime de acordo com o código VHDL.

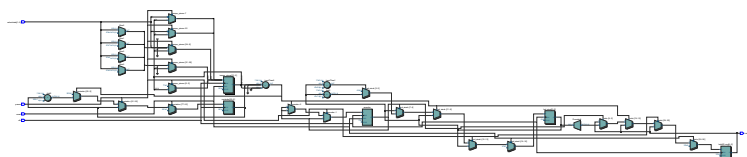


Figura 1: Diagrama RTL

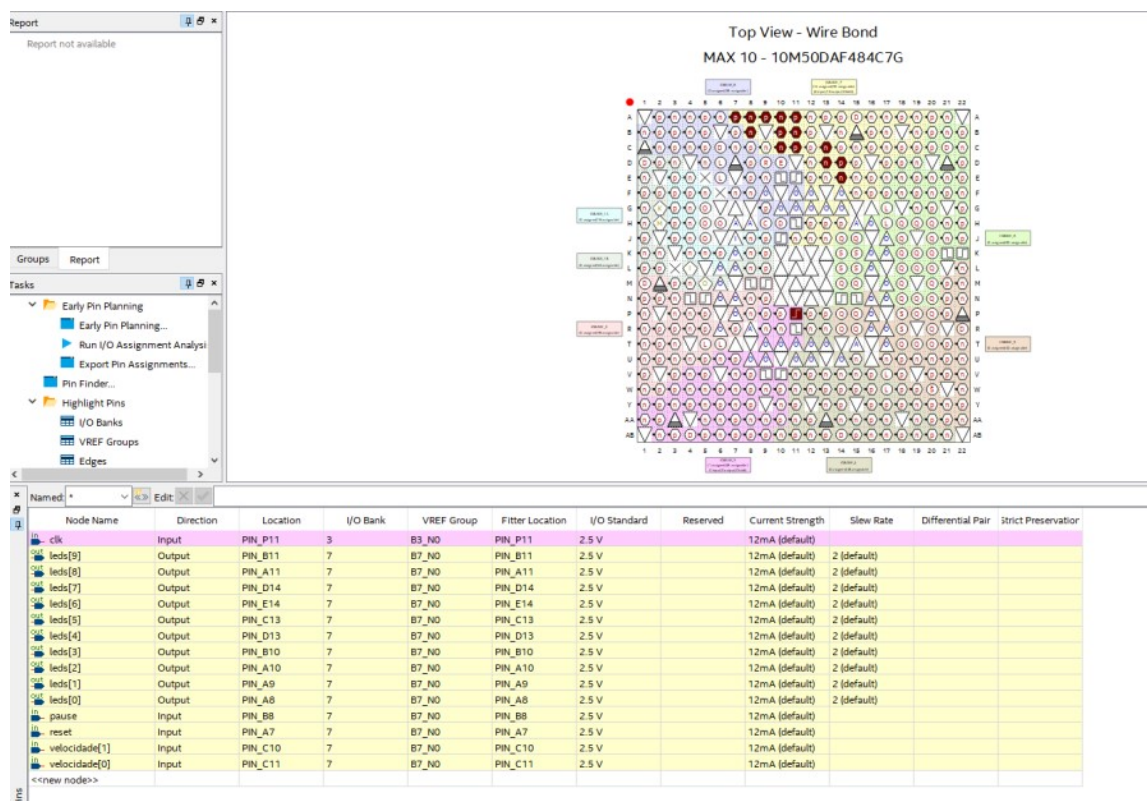


Figura 2: Pin Planner

Resultado do programa na placa.

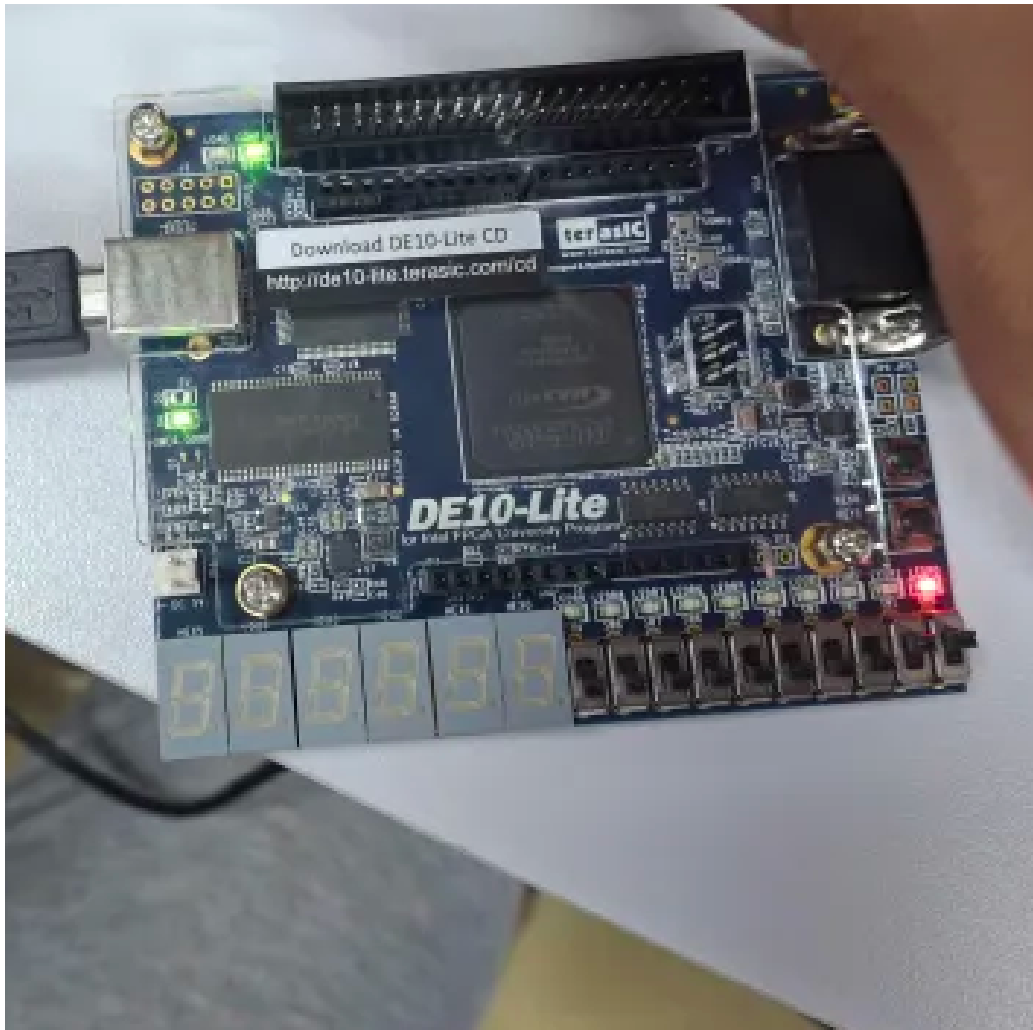


Figura 3: Exemplo na placa



Figura 4: Exemplo na placa

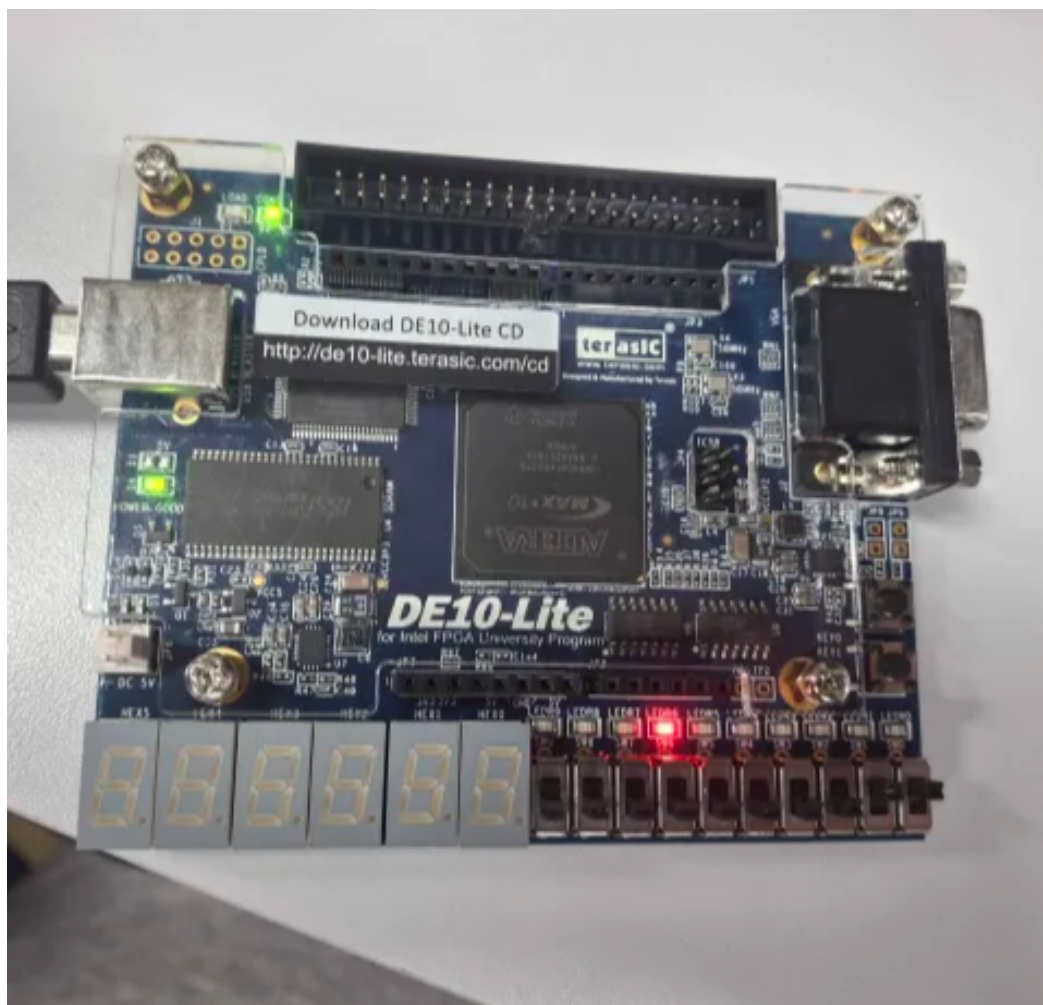


Figura 5: Exemplo na placa