



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO
JOSÉ

RELATÓRIO

Aluno: Ricardo Wagner

Disciplina: Dispositivos lógicos programáveis 2

SÃO JOSÉ, SANTA CATARINA
2025



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO
JOSÉ

INTRODUÇÃO

Esse relatório tem como objetivo descrever a AP4 feita durante as aulas. Nela tínhamos que fazer algumas modificações em relação a atividade anterior para fazer um timer regressivo, que ao zerar fica piscando com os valores 00:00:00. Ele também precisa da lógica de configuração como no exercício anterior.

DESENVOLVIMENTO

Para a realização dessa atividade, parti da implementação do relógio anterior fazendo algumas pequenas mudanças para acertar o funcionamento.

Mudanças no timer:

Mudei a entidade, adicionando a saída “fim”, esse sinal será ligado na máquina de estados para definir quando os displays irão ficar piscando 00:00:00. Ele recebe ‘1’ ao final da contagem.

Figura 1: Nova entidade timer

```
entity timer_hora_carga is
  port(
    clk, reset: in std_logic;
    load : in std_logic;
    hour_i: in std_logic_vector(4 downto 0);
    sec_i,min_i: in std_logic_vector(5 downto 0);
    hour: out std_logic_vector(4 downto 0);
    sec,min: out std_logic_vector(5 downto 0);
    fim: out std_logic
  );
end entity;
```

Fonte: o autor

Mudei a parte do “reset”. Nela coloquei para carregar o valor máximo 23:59:59, ao invés de 00:00:00 como estava anteriormente.

Figura 2: Novo reset do timer

```
process(clk,reset)
begin
    if (reset='1') then
        r_reg <= (others=>'0');
        s_reg <= "111011";
        m_reg <= "111011";
        h_reg <= "10111";
    elsif (clk'event and clk='1') then
        r_reg <= r_next;
        s_reg <= s_next;
        m_reg <= m_next;
        h_reg <= h_next;
    end if;
end process;
```

Fonte: o autor

No divisor de clock, adicionei uma verificação para manter os displays em 00:00:00 piscando ao final da execução.

Figura 3: Correção para parada do timer

```
-- next-state logic/output logic for mod-500000000 counter
r_next <= (others=>'0') when (r_reg=49999999 or load='1' or fim_signal='1') else
    r_reg + 1;
s_en <= '1' when r_reg = 49999999 else
    '0';
```

Fonte: o autor

Na parte final, apenas mudei a lógica de contagem, para contar de maneira regressiva. Além de adicionar a lógica de fim, para mandar o sinal para a máquina de estados.

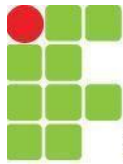


Figura 4: Nova lógica de próximo estado e saída do timer

```
-- next state logic/output logic for second divider
s_next <= unsigned(sec_i) when (load='1') else
    "111011" when (s_reg=0 and s_en='1') else
    s_reg - 1      when s_en='1' else
    s_reg;
m_en <= '1' when s_reg=0 and s_en='1' else
    '0';

-- next-state logic for minute divider
m_next <= unsigned(min_i) when (load='1') else
    "111011" when (m_reg=0 and m_en='1') else
    m_reg - 1      when m_en='1' else
    m_reg;

h_en <= '1' when m_reg=0 and m_en='1' else
    '0';

-- next-state logic for hour divider
h_next <= unsigned(hour_i) when (load='1') else
    "10111" when (h_reg=0 and h_en='1') else
    h_reg - 1      when h_en='1' else
    h_reg;
fim_signal <= '1' when h_reg = 0 and m_reg = 0 and s_reg = 0 else
    '0';

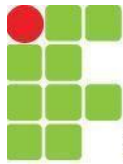
-- output logic
fim <= fim_signal;
sec <= std_logic_vector(s_reg);
min <= std_logic_vector(m_reg);
hour <= std_logic_vector(h_reg);
```

Fonte: o autor

Mudanças na máquina de estados:

Nessa parte adicionei apenas uma verificação. Ela seria para verificar se o timer chegou ao fim.

Adicionei o sinal como entrada na entidade.



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO
JOSÉ

Figura 5: Nova entidade da máquina de estados

```
entity mef is
port (
    clk, reset : in std_logic;
    start, inc, dec, fim_timer : in std_logic;
    hour_in : in std_logic_vector(4 downto 0);
    sec_in, min_in : in std_logic_vector(5 downto 0);
    load : out std_logic;
    hour_out : out std_logic_vector(4 downto 0);
    sec_out, min_out : out std_logic_vector(5 downto 0);
    en_blink_h, en_blink_m, en_blink_s, en_blink : out std_logic
);
end entity;
```

Fonte: o autor

E utilizando para a verificação.

Figura 6: Adição de verificação no estado “Parado” da máquina de estados

```
case pr_state is
when Parado =>
    load <= '0';
    if start = '1' then
        nx_state <= Hora;
    elsif fim_timer = '1' then
        en_blink <= '1';
        en_blink_h <= '1';
        en_blink_m <= '1';
        en_blink_s <= '1';
        nx_state <= Parado;
    else
        nx_state <= Parado;
    end if;
```

Fonte: o autor

Mudanças no top level:

Para a parte do top level, adicionei apenas o sinal que “fim_s”, que liga o final do timer à máquina de estados.

Simulações:

Fiz duas simulações distintas nessa parte, uma da configuração e outra da execução.

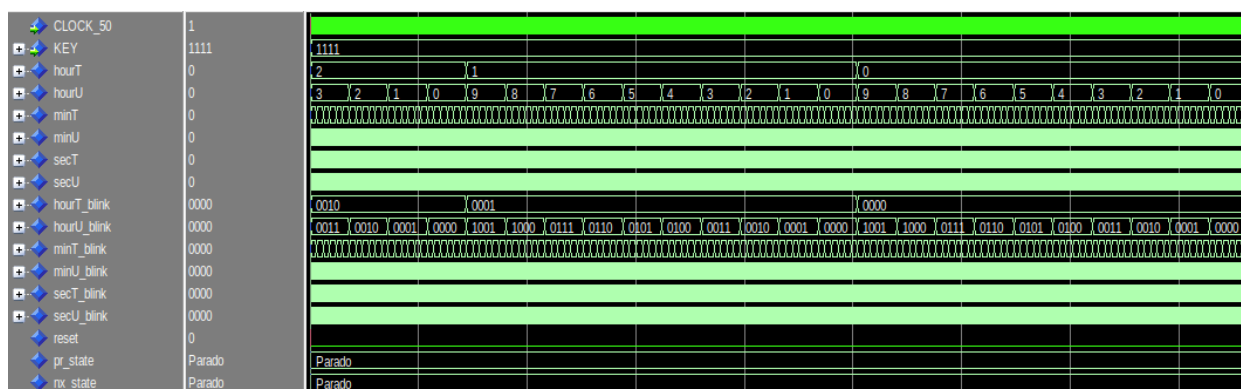
Na simulação pode se notar que os valores de hora, minuto e segundo piscam nos momentos certos.

Fonte: o autor

Na segunda parte, fiz a simulação da execução completa do programa, de 23:59:59 até 00:00:00.

Pode se notar que funciona normalmente as partes das horas nessa imagem abaixo:

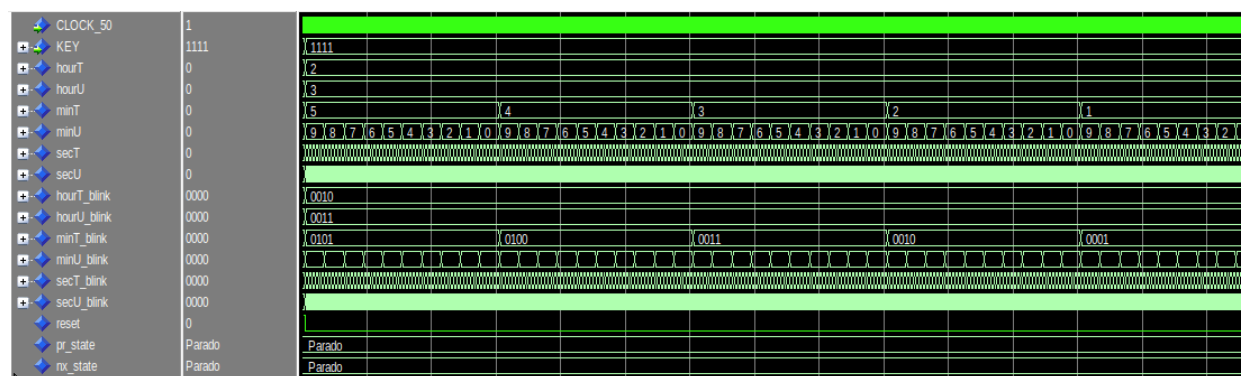
Figura 8: Simulação completa do top level



Fonte: o autor

Aqui temos um recorte da parte dos minutos:

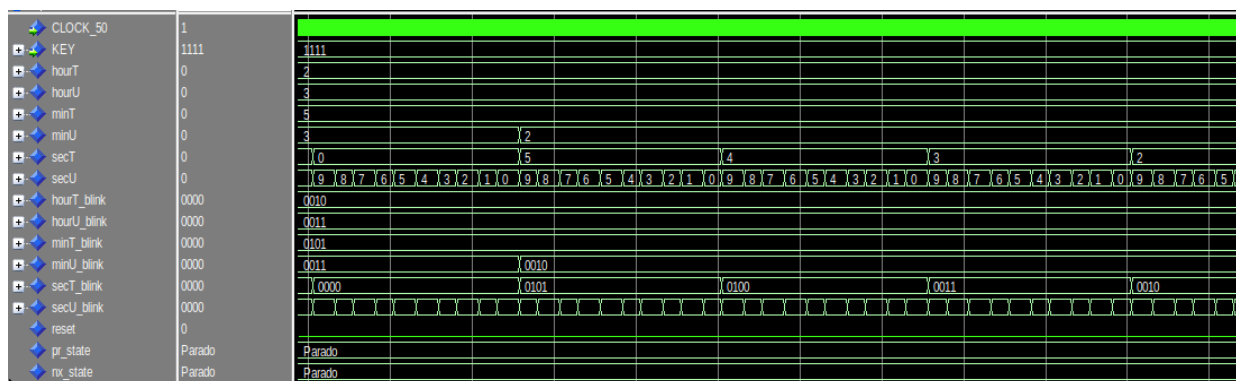
Figura 9: Recorte dos minutos da simulação completa



Fonte: o autor

E aqui dos segundos:

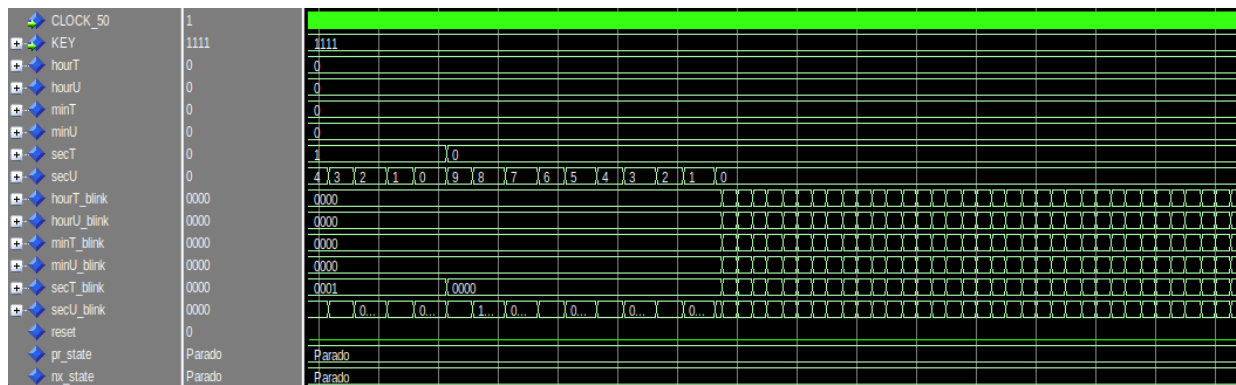
Figura 10: Recorte dos segundos da simulação completa



Fonte: o autor

Ao chegar no final, executei mais algumas vezes para verificar se o valor continuava zero e piscando:

Figura 11: Recorte final da simulação completa

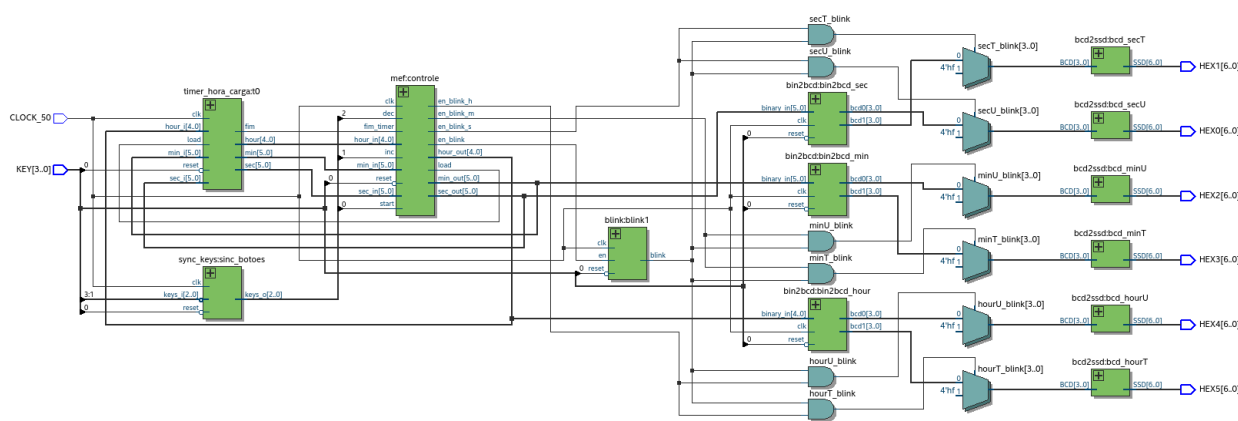


Fonte: o autor

Circuito final:

Com todos os testes realizados, o circuito final ficou assim:

Figura 12: RTL do circuito completo



Fonte: o autor

As principais mudanças estão no bloco timer e na máquina de estados. Além disso, temos o fio que liga a saída “fim” do timer a entrada “fim_timer” na máquina de estados. Mudança necessária para a lógica de piscar ao final da execução.



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO
JOSÉ

CONCLUSÃO

Com a realização dessa atividade, foi possível aprofundar mais o conhecimento sobre máquinas de estado no geral e especificamente no vhdI. Além de, paralelamente, melhorar o conhecimento em vhdI no geral, com as mudanças que teríamos que fazer na implementação do timer regressivo.