

Relatório 6 - Simulação de Arquivos VHDL

Patrik Loff Peres (20103830)
Universidade Federal de Santa Catarina (UFSC)
Departamento de Engenharia Elétrica e Eletrônica (DEEL)

I. INTRODUÇÃO

Neste laboratório foram simulados somador, flip-flop tipo T e um multiplexador 4 para 1 a partir de arquivos VHDL.

II. SOMADOR

Foi ajustado o *testbench* do somador de 4 bits para testar todas as 256 possibilidades de entrada, porém o simulador não consegue compilar a biblioteca IEEE.STD_LOGIC_UNSIGNED, dificultando a geração de um *testbench* para 256 possibilidades.

```
--for i in 0 to 15 loop
--  for j in 0 to 15 loop
--    A <= std_logic_vector(to_unsigned(i,4));
--    B <= std_logic_vector(to_unsigned(j,4));
--    wait for 10 ns;
--  end loop;
--end loop;
```

Fig. 1: parte do VHDL da *testbench* atualizado

Sendo assim, manteve-se o *testbench* com algumas das possíveis combinações de entrada.

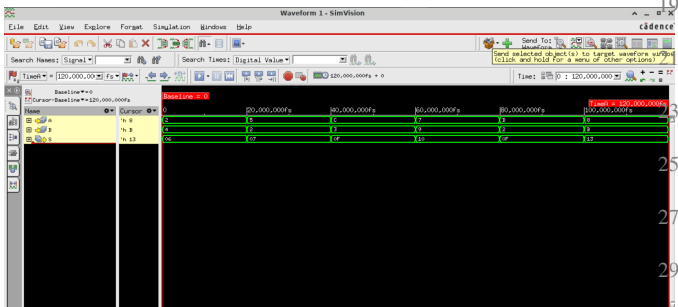


Fig. 2: Simulação do Somador

III. FLIP FLOP T

O procedimento de simulação a partir de arquivos VHDL foi aplicado ao Flip Flop tipo T como mostra a figura 3.

IV. MULTIPLEXADOR 4 PARA 1

Primeiramente foi criado um código VHDL de um multiplexador quatro para um (MUX4to1).

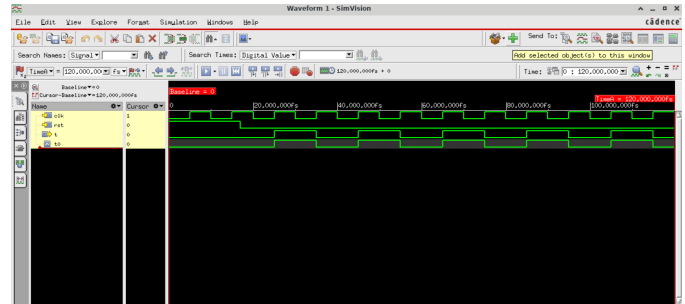


Fig. 3: Simulação Flip Flop T

```
a : in STD_LOGIC;
b : in STD_LOGIC;
c : in STD_LOGIC;
d : in STD_LOGIC;
-- Linhas de selecao
sel : in STD_LOGIC_VECTOR (1 downto 0)
;
-- Saida
y : out STD_LOGIC

);
end mux4to1;

architecture Behavioral of mux4to1 is
begin
  process (a, b, c, d, sel)
  begin
    case sel is
      when "00" =>
        y <= a;
      when "01" =>
        y <= b;
      when "10" =>
        y <= c;
      when "11" =>
        y <= d;
      when others =>
        y <= '0'; -- Valor padrao para
                    casos nao esperados
    end case;
  end process;
end Behavioral;
```

Com o mux4to1 feito, foi criado o *testbench* que inclui todas as 64 possíveis combinações de entrada.

```
Library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use work.all;

entity tb_mux4to1 is -- entity
  declaration
end tb_mux4to1;
```

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity mux4to1 is
5   Port (
6     -- Entradas
```

```

10 architecture arq_tb of tb_mux4to1 is
12 component mux4to1 is
14 port(
16     a : in STD_LOGIC;
16     b : in STD_LOGIC;
18     c : in STD_LOGIC;
18     d : in STD_LOGIC;
20     sel : in STD_LOGIC_VECTOR (1 downto 0)
22     ;
22     y : out STD_LOGIC
24 );
24 end component;
26 signal T_a,T_b,T_c,T_d: std_logic := '0';
26 signal T_sel: std_logic_vector (1 downto 0) :=
28     "00";
28 signal T_y: std_logic;
30 begin
32     U1: mux4to1 port map(T_a,T_b,T_c,T_d,T_sel,
34         T_y);
36
38     T_a <= not T_a after 5 ns;
38     T_b <= not T_b after 10 ns;
40     T_c <= not T_c after 20 ns;
42     T_d <= not T_d after 40 ns;
44
46     process
48     begin
48         wait for 80 ns;
48         T_sel <= "01";
50         wait for 80 ns;
50         T_sel <= "10";
52         wait for 80 ns;
52         T_sel <= "11";
54         wait for 80 ns;
56         end process;
58 end arq_tb;

```

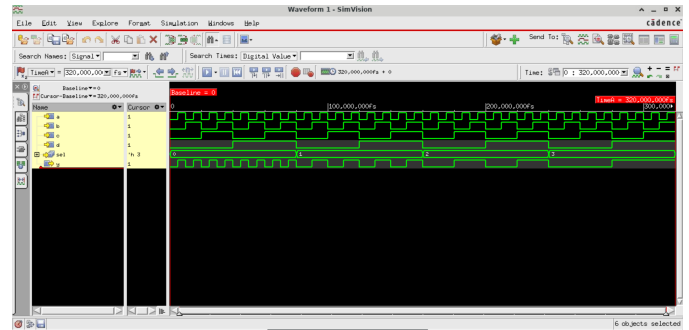


Fig. 4: Simulação Multiplexador

Também, atualizando o arquivo *file_list.f* para os nomes e caminhos corretos de todos os arquivos envolvidos, foi executado a simulação que gerou o resultado da figura 4 que mostra o resultado correto e esperado do multiplexador para todas as combinações.