ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

Test-bench para Somador

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

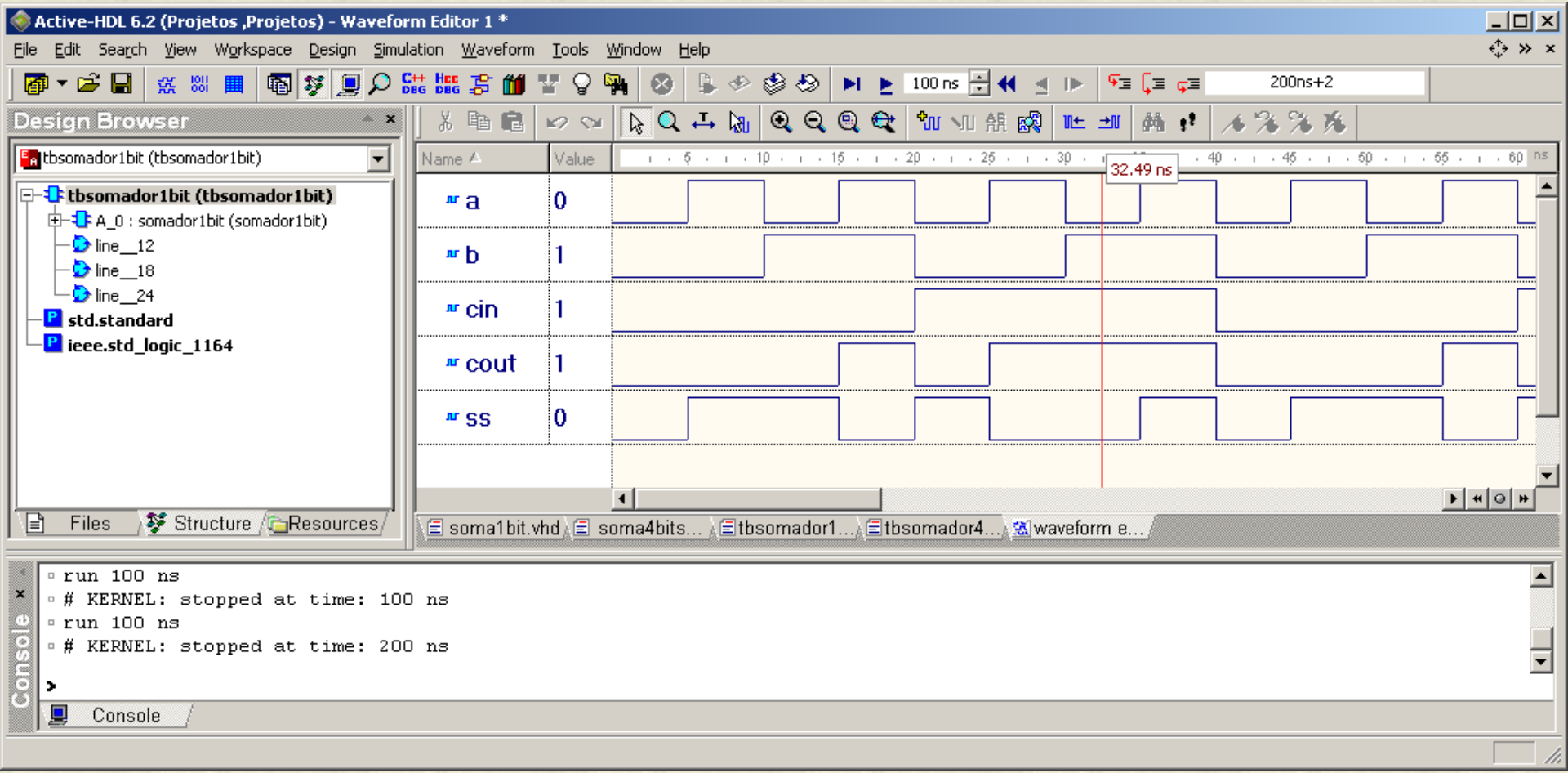
Somador Completo de 1 bit

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity Soma1Bit is
    port
    (
        cin: in std_logic;
        a: in std_logic;
        b: in std_logic;
        cout: out std_logic;
        s: out std_logic
    );
end Soma1Bit;

architecture Somador1Bit of Soma1Bit is
begin
    cout <= (a and b) or (a and cin) or (b and cin);
    s <= a xor b xor cin;
end Somador1Bit;
```

Descrição de um TB para o Somador Completo de 1 Bit

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity TbSomador1Bit is
end TbSomador1Bit;

architecture TbSomador1Bit of TbSomador1Bit is
    signal cin, a, b, cout, ss: std_logic;
begin
    A0: entity Soma1Bit port map(cin=>cin, a=>a, b=>b, cout=>cout, s=>ss);
    process
    begin
        a <= '0', '1' after 5ns;
        wait for 10 ns;
    end process;
    process
    begin
        b <= '0', '1' after 10 ns;
        wait for 20 ns;
    end process;
    process
    begin
        cin <= '0', '1' after 20 ns;
        wait for 40 ns;
    end process;
end TbSomador1Bit;
```

Simulação do Somador Completo de 1 Bit



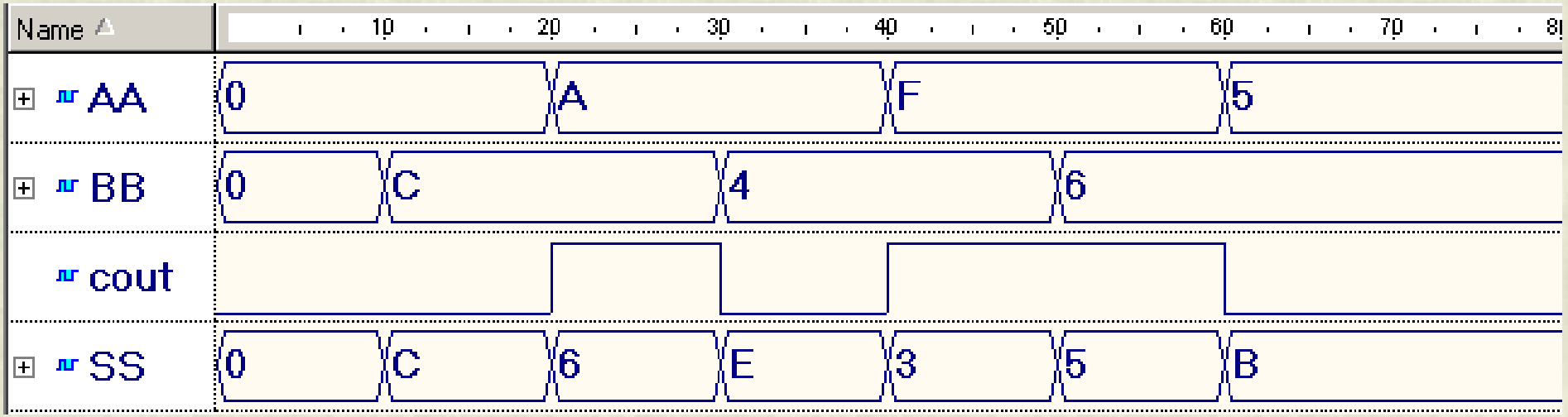
Somador de 4 bits

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity Somador4Bits is
    port(
            A: in std_logic_vector(3 downto 0);
            B: in std_logic_vector(3 downto 0);
            cout: out std_logic;
            S: out std_logic_vector(3 downto 0)
        );
end Somador4Bits;

architecture Somador4Bits of Somador4Bits is
    signal N: std_logic_vector (3 downto 0);
begin
    A0: entity Soma1Bit port map(cin=>'0',  a=>A(0), b=>B(0), cout=>N(0), s=>S(0));
    A1: entity Soma1Bit port map(cin=>N(0), a=>A(1), b=>B(1), cout=>N(1), s=>S(1));
    A2: entity Soma1Bit port map(cin=>N(1), a=>A(2), b=>B(2), cout=>N(2), s=>S(2));
    A3: entity Soma1Bit port map(cin=>N(2), a=>A(3), b=>B(3), cout=>N(3), s=>S(3));
    cout <= N(3);
end Somador4Bits;
```

Descrição do TB do somador de 4 bits

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity TbSomador4Bit is
end TbSomador4Bit;

architecture TbSomador4Bit of TbSomador4Bit is
    signal AA, BB, SS: std_logic_vector(3 downto 0);
    signal cout: std_logic;
begin
   A0:entity Somador4Bits port map(A=>AA,B=>BB,cout=>cout,S=>SS);

   AA <= x"0", x"A" after 20ns, x"F" after 40ns, x"5" after 60ns;
   BB <= x"0", x"C" after 10ns, x"4" after 30ns, x"6" after 50ns;

end TbSomador4Bit;
```

Simulação do somador de 4 bits



Exercício

1. **O que acontece se colocar todos os estímulos de entrada do somador de um bit em um único processo, tal como descrito abaixo? Entenda, simule...**

```
process
 begin
      a <= '0', '1' after 5ns;
      wait for 10 ns;
      b <= '0', '1' after 10 ns;
      wait for 20 ns;
      cin <= '0', '1' after 20 ns;
      wait for 40 ns;
 end process;
```

2. **Altere o test-bench do Somador completo de forma a gerar todas as combinações de todos os estímulos de entrada**
3. **Altere o test-bench acima de forma a testar os somadores implementados ao final da aula 5**