

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity tb_cla_tb is -- testbenkentiteter er normalt tomme.
end entity tb_cla_tb;

architecture behavioral of tb_cla_tb is
    -- en komponent er en entitet definert i en annen fil, og som vi vil bruke.
    -- komponentdeklarasjonen må matche entiteten.
    component CLA_top is
        port(
            a, b : in std_logic_vector(31 downto 0);
            cin  : in std_logic;
            sum   : out std_logic_vector(31 downto 0);
            cout  : out std_logic
        );
    end component;

    -- Tilordning av startverdi ved deklarasjon gjøres med :=
    signal tb_a, tb_b, tb_sum: std_logic_vector(31 downto 0);
    signal tb_cin, tb_cout : std_logic;

    -- outputs bør ikke få en startverdi i testbenken, da det kan maskere feil.

begin
    -- instansiering:
    DUT: cla_top
    port map(
        a      => tb_a,
        b      => tb_b,
        cin    => tb_cin,
        sum     => tb_sum,
        cout   => tb_cout
    );

    process
        begin

            tb_a <= "01100011011110100010110101101010";
            tb_b <= "0011010101011110101001001110101111";
            tb_cin <= '1';
            wait for 80 ns;
            assert(tb_sum = "10011000111101001100000100011010") report("tb_s er ulik")
            severity failure;
            assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
            wait for 80 ns;

            tb_a <= "00000110101101011110011001110111";
            tb_b <= "1001011010111101010111101010111";
            tb_cin <= '1';
            wait for 80 ns;
            assert(tb_sum = "10011101011100110100010111001111") report("tb_s er ulik")
            severity failure;
            assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
            wait for 80 ns;
        end process;
    end architecture;

```

```

    tb_a <= "1101011010101010101110101110111";
    tb_b <= "000000000011110101010111111000111";
    tb_cin <= '0';
wait for 80 ns;
    assert(tb_sum = "11010111010100000000110100111110") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "00110101011111110100000101010111";
    tb_b <= "001111111111111100001010101010111";
    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "01110101011111101010101011010111") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "00110101011111110100000101010110";
    tb_b <= "10000000000000000000000000000000";
    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "10110101011111110100000101010111") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "10000000000000000000000000000000";
    tb_b <= "10000000000000000000000000000000";
    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "00000000000000000000000000000001") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "1000000000000000000000101000000000";
    tb_b <= "10000001110001100000011000000000";
    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "00000001110001100010000000000001") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "100000000000000000000011111110000000";
    tb_b <= "011111111111100000000000011111110";

    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "111111111111000011111111111111") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "100000000000000000000010011111110";
    tb_b <= "10000000000000000000000000000000";

    tb_cin <= '1';
wait for 80 ns;
    assert(tb_sum = "0000000000000000000001001111111") report("tb_s er ulik")
severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;
    tb_a <= "11000000000000000000000000000000";

```

```
tb_b <= "10000000000000000000000000000000";

tb_cin <= '1';
wait for 80 ns;
assert(tb_sum = "01000000000000000000000000000001") report("tb_s er ulik")
severity failure;
assert(tb_cout = '1') report("tb_cout er ulik 0") severity failure;
wait for 80 ns;


report("Ferdig!") severity note;
std.env.stop;
end process;
end architecture behavioral;
```