

```

library ieee;
use ieee.std_logic_1164.all;

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

architecture behavioral of tb_cla_block is
    -- en komponent er en entitet definert i en annen fil, og som vi vil bruke.
    -- komponentdeklarasjonen må matche entiteten.
    component CLA_block is
        port(
            a, b : in std_logic_vector(3 downto 0);
            cin  : in std_logic;
            s    : out std_logic_vector(3 downto 0);
            cout : out std_logic
        );
    end component;

    -- Tilordning av startverdi ved deklarasjon gjøres med :=
    signal tb_a, tb_b, tb_s: std_logic_vector(3 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_block
    port map(
        a    => tb_a,
        b    => tb_b,
        cin  => tb_cin,
        s    => tb_s,
        cout => tb_cout
    );

    process
    begin
        wait for 10 ns;
        tb_a <= "0100";
        tb_b <= "0100";
        tb_cin <= '1';
    wait for 10 ns;
        assert(tb_s = "1001") report("tb_s er ulik 1001") severity failure;
        assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
        wait for 10 ns;
        tb_a <= "0000";
        tb_b <= "0001";
        tb_cin <= '1';
    wait for 10 ns;
        assert(tb_s = "0010") report("tb_s er ulik 0010") severity failure;
        assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
        wait for 10 ns;
        tb_a <= "0101";
        tb_b <= "1010";
    end process
end architecture;

```

```

    tb_cin <= '1';
    wait for 10 ns;
    assert(tb_s = "0000") report("tb_s er ulik 0000") severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 1") severity failure;
wait for 10 ns;
    tb_a <= "0101";
    tb_b <= "1010";
    tb_cin <= '0';
wait for 10 ns;
    assert(tb_s = "1111") report("tb_s er ulik 1111") severity failure;
    assert(tb_cout = '0') report("tb_cout er ulik 0") severity failure;
    wait for 10 ns;
    tb_a <= "1001";
    tb_b <= "1001";
    tb_cin <= '1';
wait for 10 ns;
    assert(tb_s = "0011") report("tb_s er ulik 0011") severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 1") severity failure;
    wait for 10 ns;
    tb_a <= "1001";
    tb_b <= "1001";
    tb_cin <= '0';
wait for 10 ns;
    assert(tb_s = "0010") report("tb_s er ulik 0010") severity failure;
    assert(tb_cout = '1') report("tb_cout er ulik 1") severity failure;
    wait for 10 ns;

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

```