```vhdl
-----------------------------------------
-- Testbench code:  testbench_simple_dff_circ.vhd
-- History:         04.Feb.2014 - 1st version (dqtm)
-- Function:        testbench example for simple_dff_circ
-----------------------------------------

-- Library & Use Statements
LIBRARY ieee;
USE ieee.std_logic_1164.all;


-- Entity Declaration
ENTITY testbench_simple_dff_circ IS

END testbench_simple_dff_circ;


-- Architecture Declaration
ARCHITECTURE struct OF testbench_simple_dff_circ IS

    -- Component Declaration
    COMPONENT simple_dff_circ
    PORT (
        clock   : in std_logic;
        reset_n : in std_logic;
        data_i  : in std_logic;
        hold_i  : in std_logic;
        buff_o  : out std_logic
        );
    END COMPONENT simple_dff_circ;

    -- Signals & Constants Declaration
    SIGNAL  t_clock     : std_logic;
    SIGNAL  t_reset_n   : std_logic;
    SIGNAL  t_data_i    : std_logic;
    SIGNAL  t_hold_i    : std_logic;
    SIGNAL  t_buff_o    : std_logic;

    CONSTANT  clk_halfp : time := 0.5 us; -- for Fclk=1MHz

-- Begin Architecture
BEGIN
    -----------------------------------------
    -- Instantiation DUT (Device under Test)
    -----------------------------------------
    dut: simple_dff_circ
        PORT MAP(
            clock   => t_clock,
            reset_n => t_reset_n,
            data_i  => t_data_i,
            hold_i  => t_hold_i,
            buff_o  => t_buff_o );

    -----------------------------------------
    -- Clock Generation Process (with wait)
    -----------------------------------------
    clock_gen: PROCESS
    BEGIN
        t_clock <= '0';
        wait for clk_halfp;
        t_clock <= '1';
        wait for clk_halfp;
    END PROCESS clock_gen;

    -----------------------------------------
    -- Stimuli and Check Process (with wait & assert)
    -----------------------------------------
    stimuli: PROCESS
    BEGIN
        -- initialize all inputs and
        -- activate reset_n to initialize the DUT
```

```
72          t_reset_n   <= '0';
73          t_data_i    <= '1';
74          t_hold_i    <= '0';
75          wait for    10*clk_halfp;
76
77          -- release reset_n and wait for 2 clock-periods
78          wait until  t_clock = '0';
79          t_reset_n   <= '1';
80          wait for    2*clk_halfp;
81
82          -- since hold was not active, after clock rising edege
83          -- check that buff_o = data_i
84          wait until  t_clock = '0';
85          assert      (t_buff_o = t_data_i) report "TEST_1: buff_o not equal data_i"
            severity error ;
86          wait for    2*clk_halfp;
87
88          -- change data_i and check that buff_o follows
89          wait until  t_clock = '0';
90          t_data_i    <= '0';
91          wait for    2*clk_halfp;
92          assert      (t_buff_o = t_data_i) report "TEST_2: buff_o not equal data_i"
            severity error ;
93
94          -- now set hold and check that buff_o do not follow data_i changes
95          wait until  t_clock = '0';
96          t_hold_i    <= '1';
97          t_data_i    <= '1';
98          wait for    2*clk_halfp;
99          assert      (t_buff_o /= t_data_i) report "TEST_3: buff_o equal data_i"
            severity error ;
100
101         -- stop simulation
102         wait for    10*clk_halfp;
103         assert      false report "Test programm beendet" severity failure ;
104     END PROCESS stimuli;
105
106
107  -- End Architecture
108 END struct;
109
110
111
112
113
114
115
116
```