```
============================
EA999 / SW-3
SEQUENCE OF EXAMPLES
============================

1)  SIMPLE_DFF_CIRC.VHD
    - recall from week-1 to use in testbench example


2)  TESTBENCH_SIMPLE_DFF_CIRC.VHD
    Example for testbench. Check for new elements:
    - COMPONENT DECLARATION     (needed for hierarchical VHDL)
    - INSTATIATION              (needed for hierarchical VHDL)
    - CLOCK_GEN PROCESS         (typical for testbench)
    - STIMULI PROCESS           (typical for testbench)
    - WAIT-FOR/-UNTIL STATEMENT (non synthesisable VHDL)
    - ASSERT/REPORT/SEVERITY    (non synthesisable VHDL)

    Vocabulary:
    - DUT: device under test
    - Behavioural VHDL : non synthesisable code

    - Q: draw corresponding timing diagram



3)  COMPILE.DO
    Script to start the ModelSim simulation. Attention:
    - relative paths to source files
    - vsim command calls testbench entity name
    - requires waveform file wave*.do


4)  TESTBENCH_UART_RX_ONLY_TOP.VHD
    Check for new elements:
    - Syntax for internal probes (TB has access to signals within hierarchy)
    - REPORT STATEMENT      (non synthesisable VHDL)
    - FOR/LOOP STATEMENT
```

```vhdl
-----------------------------------------
-- Block code:  simple_dff_circ.vhd
-- History:     04.Feb.2014 - 1st version (dqtm)
-- Function:    simple 1-DFF circuit for VHDL introduction
-----------------------------------------

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


-- Entity Declaration
ENTITY simple_dff_circ IS
    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 simple_dff_circ;


-- Architecture Declaration
ARCHITECTURE rtl OF simple_dff_circ IS

    -- Signals & Constants Declaration
    SIGNAL buff, next_buff  : std_logic ;

-- Begin Architecture
BEGIN
    -----------------------------------------
    -- Process for combinatorial logic
    -----------------------------------------
    comb_logic: PROCESS(ALL)
    BEGIN
    -- hold or update
        IF hold_i='1' THEN
            next_buff <= buff;
        ELSE
            next_buff <= data_i;
        END IF;
    END PROCESS comb_logic;


    -----------------------------------------
    -- Process for registers (flip-flops)
    -----------------------------------------
    flip_flops : PROCESS(clock, reset_n)
    BEGIN
        IF reset_n = '0' THEN
            buff <= '0';
        ELSIF RISING_EDGE(clock) THEN
            buff <= next_buff ;
        END IF;
    END PROCESS flip_flops;

    -----------------------------------------
    -- Concurrent Assignements
    -- e.g. Assign outputs from intermediate signals
    -----------------------------------------
    buff_o <= buff;

 -- End Architecture
END rtl;

```

```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
```

```
# create work library
vlib work

# compile project files
vcom -2008 -explicit -work work ../../source/testbench_simple_dff_circ.vhd
vcom -2008 -explicit -work work ../../source/simple_dff_circ.vhd


# run the simulation
vsim -novopt -t 1ns -lib work work.testbench_simple_dff_circ

do ../scripts/wave.do
run 30.0 us


```

```vhdl
-----------------------------------------
-- Block code:  testbench_uart_rx_only_top.vhd
-- History:     13.Mar.2018 - 1st version (dqtm)
--                 <date> - <changes>  (<author>)
-- Function: Testbench for uart_rx_only_top in EA999 - Lab2
--
-----------------------------------------

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

-- Entity Declaration
ENTITY testbench_uart_rx_only_top IS

END testbench_uart_rx_only_top ;


-- Architecture Declaration
ARCHITECTURE struct OF testbench_uart_rx_only_top IS

    -- Components Declaration
COMPONENT uart_rx_only_top
    PORT
    (
        CLOCK_50 :  IN   STD_LOGIC;
        GPIO_1_1 :  IN   STD_LOGIC;
        KEY_N_0 :   IN   STD_LOGIC;
        HEX0_N :    OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
        HEX1_N :    OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
        LEDR_0 :    OUT  STD_LOGIC
    );
END COMPONENT;


    -- Signals & Constants Declaration
    -- Inputs
    SIGNAL tb_clock     : std_logic;
    SIGNAL tb_reset_n   : std_logic;
    SIGNAL tb_serdata   : std_logic;
    -- Outputs
    SIGNAL tb_ledr      : std_logic;
    SIGNAL tb_hex_0     : std_logic_vector(6 downto 0);
    SIGNAL tb_hex_1     : std_logic_vector(6 downto 0);

    CONSTANT clk_50M_halfp  : time := 10 ns;        -- Half-Period of Clock 50MHz
    CONSTANT baud_31k250_per : time := 32 us;       -- One-Period of Baud Rate
    31.25KHz

    SIGNAL tb_reg0_hi   : std_logic_vector(3 downto 0); -- to check DUT-internal
    signal
    SIGNAL tb_reg0_lo   : std_logic_vector(3 downto 0);

    SIGNAL tb_test_vector : std_logic_vector(9 downto 0); --
    (stop-bit)+(data-byte)+(start-bit) to shift in serial_in

BEGIN
  -- Instantiations
  DUT: uart_rx_only_top
  PORT MAP (
    CLOCK_50    =>  tb_clock ,
    GPIO_1_1    =>  tb_serdata ,
    KEY_N_0     =>  tb_reset_n ,
    HEX0_N      =>  tb_hex_0 ,
    HEX1_N      =>  tb_hex_1 ,
    LEDR_0      =>  tb_ledr
        );

  -- Clock Generation Process
    generate_clock: PROCESS
    BEGIN
```

```
69          tb_clock <= '1';
70          wait for clk_50M_halfp;
71          tb_clock <= '0';
72          wait for clk_50M_halfp;
73      END PROCESS generate_clock;
74
75
76      -----------------------------------------
77      -- VHDL-2008 Syntax allowing to bind
78      --           internal signals to a debug signal in the testbench
79      -----------------------------------------
80      tb_reg0_hi <= <<signal DUT.hexa_hi : std_logic_vector(3 downto 0) >>;
81      tb_reg0_lo <= <<signal DUT.hexa_lo : std_logic_vector(3 downto 0) >>;
82
83
84    -- Stimuli Process
85      stimuli: PROCESS
86      BEGIN
87          -- STEP 0
88          report "Initialise: define constants and pulse reset on/off";
89          tb_serdata <= '1';
90          tb_test_vector <= B"1_0001_0010_0"; -- (stop-bit)+(data-byte)+(start-bit)
91          ----------------
92          tb_reset_n <= '0';
93          wait for 20 * clk_50M_halfp;
94          tb_reset_n <= '1';
95          wait for 100 us;  -- introduce pause to check HW-bug after reset release
96
97          -- STEP 1
98          report "Send (start-bit)+(data-byte)+(stop-bit) with baud rate 31250 (async
            from clk50M)";
99          wait for 200 * clk_50M_halfp;
100
101         -- shift-direction is LSB first
102         -- START-BIT
103         -- BIT-0 (LSB-first of lower nibbl
104         -- BIT-1
105         -- ...
106         -- BIT-7
107         -- STOP-BIT
108         -- SERDATA BACK TO INACTIVE
109         for I in 0 to 9 loop
110             tb_serdata <= tb_test_vector(I);
111             wait for baud_31k250_per;
112         end loop;
113
114         -- STEP 2
115         report "Wait few clock_50M periods and check parallel output";
116         wait for 20 * clk_50M_halfp;
117         assert (tb_reg0_hi = x"1") report "Reg_0 Lower Nibble Wrong" severity note;
118         assert (tb_reg0_lo = x"2") report "Reg_0 Lower Nibble Wrong" severity note;
119         wait for 20 * clk_50M_halfp;
120
121         -- stop simulation
122         assert false report "All tests passed!" severity failure;
123
124     END PROCESS stimuli;
125
126     -- Comments:
127     -- remember to re-save wave.do after setting radix and time ranges
128     -- use >wave>format>toggle leaf names  to display shorter names
129     -- remark syntax with aux debug signal to track in TB internal DUT signals
130
131 END struct;
132
133
```